RRT-based maximum entropy inverse reinforcement learning for robust and efficient driving behavior prediction

Shinpei Hosoma[1], Masato Sugasaki[1], Hiroaki Arie[2], and Masamichi Shimosaka[1]

Abstract— **Advanced driver assistance systems have gained popularity as a safe technology that helps people avoid traffic accidents. To improve system reliability, a lot of research on driving behavior prediction has been extensively researched. Inverse reinforcement learning (IRL) is known as a prominent approach because it can directly learn complicated behaviors from expert demonstrations. Because driving data tend to have a couple of optimal behaviors from the drivers' preferences, i.e., sub-optimality issue, maximum entropy IRL has been getting attention with their capability of considering suboptimality. While accurate modeling and prediction can be expected, standard maximum entropy IRL needs to calculate the partition function, which requires large computational costs. Thus, it is not straightforward to apply this model to a high-dimensional space for detailed car modeling. In addition, existing research attempts to reduce these costs by approximating maximum entropy IRL; however, a combination of the efficient path planning and the proper parameter updating is required for an accurate approximation, and existing methods have not achieved them. In this study, we leverage a rapidly-exploring random tree (RRT) motion planner. With the RRT planner, we propose novel importance sampling for an accurate approximation from the generated trees. This ensures a stable and fast IRL model in a large high-dimensional space. Experimental results on artificial environments show that our approach improves stability and is faster than the existing IRL methods.**

I. INTRODUCTION

In recent years, advanced driver assistance systems (ADAS) have improved the safety of one's driving, for example, adaptive cruise control (ACC [12]) and traffic sign recognition (TSR [23]). Many cars have been equipped with these technologies, which reduced the rate of traffic accidents [7]. Researchers extensively investigate ADAS, aiming to further improve the safety of automatic driving systems. Driving behavior prediction is crucial for the robustness of ADAS, with methods such as model predictive control [1] or reinforcement learning [15].

In driving behavior predictions via model predictive control, the sequence of control inputs is obtained by maximization of sum of rewards, an indicator of the preferred behaviors. Therefore, the careful design of reward function is necessary. Meanwhile, designing reward functions tends to be complicated due to the complex nature of human factors in driving, such as slow acceleration or comfortable steering.

Inverse reinforcement learning (IRL), inverse problem of reinforcement learning, is known as a promising approach to solving this problem. This is because the reward functions can be directly learned from demonstration data without a complex design. Among IRL methods researched in the last 20 years [14], [24], maximum entropy IRL [24] is known to be one of the common frameworks thanks to its theoretical properties. The property is known as "sub"-optimality; i.e., the model could learn not only optimal behaviors but also suboptimal ones.

Notably, the human demonstrations highly tend to be tentative even if the preference of the demonstration is consistent, and their behaviors naturally include multiple "sub"-optimal decisions. The suboptimality-reflected reward recovery of maximum entropy IRL is suitable for expressing these noisy driving behaviors. In the last decade, techniques based on maximum entropy IRL for driving behavior prediction have been actively explored in acceleration / deceleration modeling [17], [18].

Despite its popularity in maximum entropy IRL, its original form is not so suitable for generic driving behavior modeling because it assumes the state space is discretized. In other words, the applied driving scenario is quite limited within a reasonable computational cost. For more generic driving behavior modeling, non-holonomic motion dynamics should be treated, i.e., not only the x-y positions, but also velocity, angle, and angular velocity should be handled together. This setting causes exponential growth of the number of discrete states; therefore, it is not so feasible to directly apply the discrete type maximum entropy IRL [19].

Towards more practical usage of maximum entropy IRL, recent advances in maximum entropy IRL expand the models into continuous settings, i.e., continuous optimal control [8]. The techniques were born in machine learning literature, then the successful results in continuous maximum entropy IRL are reported in the robotics and autonomous driving literature [21], [22] in recent years.

In the literature, the way on approximations of the log-partition function in maximum entropy IRL is known to be the key to its success due to the intractable integral on reward functions over the possible trajectories [8]. Various types of approximation spanning from Laplace approximation [8], importance sampling [4], and soft-max approximation with perceptron training [16] are proposed.

Though various types of algorithms are presented in the literature, their performance is not sufficient for practical driver behavior modeling. Most of the methods with innovative way on approximated log partition function assume to use of proper efficient motion samplers from the current given reward function; however, its stability and efficiency

[1]The authors are with the Department of Computer Science, Tokyo Insititute of Technology, Tokyo, Japan. E-mail: {hosoma, sugasaki, simosaka}@miubiq.cs.titech.ac.jp

[2]The author is a researcher at the DENSO Corporation, Tokyo, Japan. E-mail: {hiroaki.arie.j8k}@jp.denso.com

is not fully considered for the driving behavior prediction problem.

In this paper, we pursue a more reliable approach to continuous maximum entropy IRL by seeking the best combination of proper approximation on log-partition function in maximum entropy IRL, and efficient and robust motion sampler in driving behavior prediction. Throughout the motion generators presented in the literature, we focus on rapidly-exploring random tree (RRT) [6] as one of the efficient and robust motion samplers in this paper.

In this paper, we investigate how we use the leaf nodes from the given RRT tree to approximately obtain the log partition function efficiently. Finally, we propose the new RRT based continuous maximum entropy IRL for driving behavior prediction ensuring robustness and efficiency. Contributions of this paper can be summarized as follows:

- In this study, we propose a new type of maximum entropy continuous IRL method with RRT as a path planner in training / inference phase for pursuing robustness and efficiency. Specifically, we employ a new type of motion sampler for obtaining approximated log partition function and its derivative in an importance sampling manner. The motion sampler is quite efficient under the tree is given while the other previous approaches require additional computational cost with on-policy evaluation.
- The experimental results on a couple of driving behavior prediction scenarios show that our model could achieve faster and more stable results than the state-of-the-art methods.

Related work:

IRL for driving behavior prediction: In the last two decades, various trials on driving behavior prediction with IRLs have been presented. Typical examples include avoidance of other cars [19], driving behaviors at unsignalized intersections [17], [18], car following scenarios [22], and lane changing scenarios [21]. Though their report shows prominent successful results in each scenario, the application scenario of each paper is not flexible but fixed.

IRL on continuous settings: The continuous inverse reinforcement learning has been proposed as a practical alternative to the traditional discrete IRL in the literature. In the setting, the model directly optimizes the reward function of the continuous space without discretizing it. Due to the intractability of integral on reward function, i.e. log-partition function, the model requires its approximation. Levine et al. [8] applied Laplace approximation to maximum entropy IRL; however, it highly depends on initial guesses owing to the local approximation. Therefore, the result is prone to instability and is not suitable for driving behavior prediction. Shiarlis et al. [16] and Xin et al. [22] obtained the highest reward path without planning all possible paths for the calculation of the partition function. However, they only referred to a single optimal path; therefore, it is not possible to consider the suboptimality in driving principles. Finn et al. proposed GCL [4], which approximated the partition function by importance sampling. In contrast to the other approaches mentioned above, their approach is theoretically stable from the viewpoint of convergence thanks to the nature of importance sampling. However, it is also reported that the probabilistic control with KL divergence [4] is not so stable [21], therefore many demonstrations and trials is necessary. This stems from the fact that the probabilistic control with model-free approach, i.e., motion dynamics is not given in IRL, requires on-policy evaluation during IRL training process. In contrast, the motion dynamics is given as non-holonomic in the driving behavior prediction, therefore the proper model-based motion planner should be employed instead.

II. Formulation of Continuous IRL

In this section, we describe the problem settings of continuous maximum entropy IRL and the existing approaches on approximated solution.

A. Problem setting of continuous IRL

We define the continuous state space X and the control input space U. When a continuous state $\boldsymbol{x}_t \in X$ and a control input$\boldsymbol{u}_t \in U$ at a discrete time t are given, the agent moves to the next state based on the dynamics function F as $\boldsymbol{x}_{t+1} = F(\boldsymbol{x}_t, \boldsymbol{u}_t)$. To express detailed models, addressing high-dimensional state space is needed with its multiple state variables of nonholonomic dynamics; such as x, y positions, velocity v, and body angle θ. In addition, we assume that the agent gains the below immediate reward $r_{\boldsymbol{w}}$ throughout the dynamics transition parameterized by $\boldsymbol{w}$. Here $r_{\boldsymbol{w}}$ can be modeled by either linear function of $\boldsymbol{w}$ or non-linearly such as neural networks parameterized by $\boldsymbol{w}$. The purpose of IRL is to learn $\boldsymbol{w}$ from the driving behavior demonstration.

B. Maximum entropy IRL for continuous space

Continuous maximum entropy IRL, which has been commonly used in the IRL modeling, is one of the probability models that defines a path $\tau = \{(\boldsymbol{x}_1, \boldsymbol{u}_1), (\boldsymbol{x}_2, \boldsymbol{u}_2) \dots (\boldsymbol{x}_T, \boldsymbol{u}_T)\}$ as probability as follows:

$$p(\tau; \boldsymbol{w}) = \frac{\exp\Big(\sum_{\boldsymbol{x}_t, \boldsymbol{u}_t \in \tau} r_{\boldsymbol{w}}(\boldsymbol{x}_t, \boldsymbol{u}_t)\Big)}{Z(\boldsymbol{w})},$$
$$Z(\boldsymbol{w}) = \int \exp\Big(\sum_{\tilde{\boldsymbol{x}}_t, \tilde{\boldsymbol{u}}_t \in \tilde{\tau}} r_{\boldsymbol{w}}(\tilde{\boldsymbol{x}}_t, \tilde{\boldsymbol{u}}_t)\Big) \mathrm{d}\tilde{\tau}. \tag{1}$$

By minimizing the negative log-likelihood of the demonstration data, we can obtain the optimal parameter $\boldsymbol{w}^*$. With the gradient-based techniques to obtain $\boldsymbol{w}^*$, we should use the following the loss function and its gradient as follows:

$$L(\boldsymbol{w}) = \sum_{\tau \in D_{\mathrm{demo}}} \Big(\log Z(\boldsymbol{w}) - \sum_{\boldsymbol{x}_t, \boldsymbol{u}_t \in \tau} r_{\boldsymbol{w}}(\boldsymbol{x}_t, \boldsymbol{u}_t)\Big),$$
$$\frac{\mathrm{d}L(\boldsymbol{w})}{\mathrm{d}\boldsymbol{w}} = \sum_{\tau \in D_{\mathrm{demo}}} \Big(\frac{\mathrm{d}}{\mathrm{d}\boldsymbol{w}} \log Z(\boldsymbol{w}) - \sum_{\boldsymbol{x}_t, \boldsymbol{u}_t \in \tau} \frac{\mathrm{d}}{\mathrm{d}\boldsymbol{w}} r_{\boldsymbol{w}}(\boldsymbol{x}_t, \boldsymbol{u}_t)\Big). \tag{2}$$

It should be noted that it is infeasible to exactly obtain the log partition function $Z(\boldsymbol{w})$ and its gradient. In contrast, this intractability of the integral inspires a series of research work on continuous IRL with various approximations.

C. Importance sampling for the reliable approximation

Among a couple of approaches for approximation, Finn et al. [4] employ importance sampling for (1) as

$$Z(\boldsymbol{w}) \approx \frac{1}{|D_{\mathrm{samp}}|} \sum_{\tau \in D_{\mathrm{samp}}} \frac{\exp\left(\sum_{\boldsymbol{x}_t, \boldsymbol{u}_t \in \tau} r_{\boldsymbol{w}}(\boldsymbol{x}_t, \boldsymbol{u}_t)\right)}{q(\tau)}, \tag{3}$$

where $q(\tau)$ depicts the auxiliary density function for motion paths, also used as a motion sampler, and D_{samp} depicts a collection of generated motion paths from density $q(\tau)$. With this assumption, the gradient of (1) can also be approximated as

$$\frac{\mathrm{d}}{\mathrm{d}\boldsymbol{w}} \log Z(\boldsymbol{w}) \approx \frac{1}{G} \sum_{\tau \in D_{\mathrm{samp}}} g(\tau) \sum_{\boldsymbol{x}_t, \boldsymbol{u}_t \in \tau} \frac{\mathrm{d}}{\mathrm{d}\boldsymbol{w}} r_{\boldsymbol{w}}(\boldsymbol{x}_t, \boldsymbol{u}_t). \tag{4}$$

Where we use $g(\tau) = \frac{\exp\left(\sum_{\boldsymbol{x}_t, \boldsymbol{u}_t \in \tau} r_{\boldsymbol{w}}(\boldsymbol{x}_t, \boldsymbol{u}_t)\right)}{q(\tau)}$, and $G = \sum_{\tau \in D_{\mathrm{samp}}} g(\tau)$. It should be noted that the importance sampling could assure the unbiased estimator of $\log Z(\boldsymbol{w})$ and its gradient with a sufficient number of motion samples D_{samp}, therefore the theoretical convergence for obtaining optimality of $\boldsymbol{w}^*$ can be obtained.

However, its efficiency is governed by the accuracy of motion sampler $q(\tau)$ and GCL faces the inefficiency or degraded performance due to the utility of model-free probabilistic control. Specifically, they used a linear Gaussian controller based model-free planner as a distribution $q(\tau)$ to sample paths. In addition, the execution of path planner is required per single motion sample $\tau \in D_{\mathrm{samp}}$, and this causes a lack of scalable training process.

To overcome this limitation, in this research, we pursue a more reliable and efficient motion generator and its sampling process by focusing on the target is driving behavior, i.e., motion dynamics is given.

III. RRT-Based Maximum Entropy IRL

In this research, we attempt efficient path sampling by adopting one of the model-based planners, rapidly-exploring random tree (RRT). Though it seems to be a naive extension from GCL, the key of our algorithm is not only limited to just employing RRTs alternative to the probabilistic control, but also providing an efficient approach to generate motion samples $\tau \in D_{\mathrm{samp}}$ from $q(\tau)$.

A. Efficient path planning with RRT

As previously mentioned, path planners that are used in existing fast IRL methods for driving behavior prediction, only search for a limited part of state space; hence, they cannot work on the many types of tasks. To resolve this problem, we use an RRT-based motion planner [6], [13], [20] for nonholonomic dynamics. RRT [6] has been known as a stable, fast, and versatile planner for several decades.

RRT generates trees covering a diverse region of the state space, and is expected to obtain near-globally optimal results. In addition, owing to the simplicity of its incremental sampling approach, the computational cost is not very large even if it is used in a high-dimensional space.

B. Exploiting tree structures for highly efficient motion sampling

In this research, we need to carefully define the auxiliary probability density function for motion path τ by $q(\tau)$ for reliable and efficient continuous maximum entropy IRL. In its designing process, we should take care of the number of executions of the path planner per objective and its gradient computation.

Generally, multiple executions of path planners are required to obtain a set of sampled paths. The linear increase in computational cost prevents us from modeling a fast IRL method despite the rapidity of RRT. To tackle this problem, we focus on leveraging all the tree structures generated by single execution of RRT generation to obtain multiple motion samples efficiently.

In general setting of RRT motion planning, the best path is pursued while the rest possible paths are discarded; however, these "sub"-optimal paths can also be naturally reused for the motion samples as generated samples from $q(\tau)$. This process enables us to collect many paths with only a single execution. Hence, the computational cost of planning becomes smaller than that of the existing sampling-based approach.

Fig. 1 shows concepts of our RRT-based importance sampling approach. If the density function $q(\tau)$ is properly defined, the set of paths can be obtained efficiently.

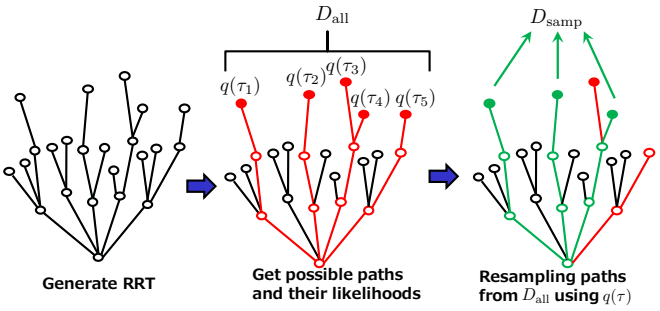


Fig. 1: Importance sampling process based on RRT results

Next, we explicitly formulate the density function $q(\tau)$ to obtain the efficient RRT-based maximum entropy IRL.

C. Fast partition function calculation by RRT-based importance sampling

In this paper, we leverage the fact that the score to obtain the sum of rewards in each path from the root to the leaf in generated RRT can be efficiently calculated via backward calculation. To enhance the efficiency of the motion sampling, we assume all the motion paths whose path length is T as possibly generated. We also assume that the path with a higher sum of rewards is likely to be generated from the given reward function. With this flavor, we define the probability density function $q(\tau)$ over the motion path τ as

$$q(\tau) = \frac{\sum_{\boldsymbol{x}_t, \boldsymbol{u}_t \in \tau} \exp\left(r_{\boldsymbol{w}}(\boldsymbol{x}_t, \boldsymbol{u}_t)\right)}{\sum_{\tilde{\tau} \in D_{\mathrm{all}}} \sum_{\tilde{\boldsymbol{x}}_t, \tilde{\boldsymbol{u}}_t \in \tilde{\tau}} \exp\left(r_{\boldsymbol{w}}(\tilde{\boldsymbol{x}}_t, \tilde{\boldsymbol{u}}_t)\right)}. \tag{5}$$

Using this definition, we can calculate q by assuming that D_{all} contains all the possible paths in the space. Besides, we can use q as a path generator, which probabilistically selects a path from D_{all}, because q is normalized to D_{all}. Therefore, we conduct resampling from D_{samp} using q, and obtain the new sampled set D_{samp}, as of the right side of Fig. 1. Through these algorithms, we can approximate the partition function (3) using q and D_{samp}.

Algorithm 1 summarizes all the processes of our parameter updating method based on importance sampling, where $\boldsymbol{w}_{\mathrm{init}}$, N, and $\mathcal{T}$ denote an initial weight parameter, the maximum number of iterations, and generated tree information, respectively. Thus, we completely formulate an efficient RRT-based maximum entropy IRL on the continuous state space for driving behavior prediction.

Algorithm 1 RRT-based maximum entropy IRL

Input: $\boldsymbol{w}_{\mathrm{init}}, D_{\mathrm{demo}}, N$

$\boldsymbol{w} \leftarrow \boldsymbol{w}_{\mathrm{init}}$
while $i < N$ **do**
  for $\tilde{\tau} \in D_{\mathrm{demo}}$ **do**
    $\boldsymbol{x}_0 \leftarrow \tilde{\tau}$
    $\mathcal{T} \leftarrow \mathrm{generateRRT}(\boldsymbol{x}_0, |\tilde{\tau}|)$
    $D_{\mathrm{all}}, q \leftarrow \mathrm{backtrackTree}(\mathcal{T})$ (Fig. 1 and (5))
    $D_{\mathrm{samp}} \leftarrow \mathrm{resampling}(D_{\mathrm{all}}, q)$ (Fig. 1)
    $\frac{\mathrm{d}L(\boldsymbol{w})}{\mathrm{d}\boldsymbol{w}} \leftarrow \mathrm{getGrad}(D_{\mathrm{samp}}, q)$ ((2) and (4))
    $\boldsymbol{w} \leftarrow \mathrm{update}(\boldsymbol{w}, \frac{\mathrm{d}L(\boldsymbol{w})}{\mathrm{d}\boldsymbol{w}})$
    $i \leftarrow i + 1$
  end for
end while

Output: $\boldsymbol{w}$

D. Comparison to the existing approximated methods

To highlight the advances in our method, we describe the comparison of existing methods on the approximation of the partition function with ours.

1) Laplace approximation: Levine et al. [8] applied maximum entropy IRL to a continuous space using Laplace approximation. Although it makes learning faster in the high-dimensional space, its dependency on initial guesses and unstable results are caused by the local approximation.

2) The highest reward path approximation: Shiarlis et al. [16] and Xin et al. [22] approximated the partition function $Z(\boldsymbol{w})$ with a path that gained the highest reward as $Z(\boldsymbol{w}) \approx \max_\tau \exp\Big(\sum_{\boldsymbol{x}\in\tau} r_{\boldsymbol{w}}(\boldsymbol{x})\Big)$. This requires only one highest-reward path, whereas (1) needs all possible paths; hence, a significant cost reduction is expected. However, this model is not suitable for noisy driving data because it only refers to the optimal behavior and ignores other suboptimal ones. It should be noted that Shiarlis et al. [16] employ variants of RRT as motion planner during its IRL; however, the combination of approximated log partition function $Z(\boldsymbol{w})$ and motion planner is still important for proper behavior modeling.

3) Approximation with multiple paths: Wu et al. [21] calculated the summation of the set of sampled paths D_{samp} as $Z(\boldsymbol{w}) \approx \sum_{\tau\in D_{\mathrm{samp}}} \exp\Big(\sum_{\boldsymbol{x}\in\tau} r(\boldsymbol{x})\Big)$. Needless to say, this increases when the number of samples $|D_{\mathrm{samp}}|$ increases. therefore, this approximation is not mathematically correct.

GCL presented by Fin et al. [4] is the closest work to our model; however, its motion sampler is quite different. In contrast to the use of guided policy search (GPS [9]) in GCL, our model prefers to employ a more robust motion sampler even if it is specific to vehicle motion dynamics.

IV. Experimental Results

We confirm that our method can perform more stable and faster than existing methods on the multiple driving tasks, using both quantitative and qualitative evaluations.

A. Target scenarios: lane-change & intersection behavior

We use two driving scenarios in the experiment. One is a lane-change task. When obstacles surrounded by cones are placed on the road, we learn behaviors to avoid them by changing lanes.

The second task is the turning task at the intersection. The agent should move from the initial position to the target lanes by steering the car. Using these scenarios, we confirm whether the models stably learn the driving behaviors regardless of the type of task.

B. State space and dynamics settings

In this experiment, we consider a 5-dimensional state space, where a $\boldsymbol{x}_t$ at time step t is expressed as $\boldsymbol{x}_t = (x_t, y_t, \theta_t, v_t, \omega_t)^\top$, where x_t, y_t, θ_t, v_t, and ω_t denote the x-position, y-position, angle, velocity, and angular velocity, respectively. In addition, the input vector is defined by a 2-dimensional vector consisting of acceleration and angular acceleration: $\boldsymbol{u}_t = (a_t, \alpha_t)^\top$. When the state and input at time step t are given, the agent moves to the next state $\boldsymbol{x}_{t+1}$ by following nonholonomic dynamics. ΔT denotes time granularity.

C. Reward representation

To simplify the evaluation protocol, we represent reward function $r_{\boldsymbol{w}}(\boldsymbol{x}_t, \boldsymbol{u}_t)$ as linear mapping function parameterized $\boldsymbol{w}$. Specifically, the reward function can be defined as $r_{\boldsymbol{w}}(\boldsymbol{x}_t, \boldsymbol{u}_t) = \boldsymbol{w}^\top \boldsymbol{\phi}(\boldsymbol{x}_t)$ in the evaluation. It should be noted that the time-variant and nonlinear neural models for r can be used in the evaluation for practical use cases; however, the main objective of this experiment is to confirm the validity of using RRT-based motion sampler in contrast to the other log partition approximation or other motion generators.

In this experiment, we design feature $\boldsymbol{\phi}$ with 11 factors, for example, the center of a lane, obstacles, and roadsides. The number of lanes varies from 2 to 4 and the reward function is time-invariant. For tasks at the intersection, we use additional features: desirable velocity, angular velocity, and positions of target lanes.

For simplicity, we deal with time-invariant settings, where the surrounding environment does not change with the passage of time; however, the planning and learning in our method can work on time-variant settings.

D. Dataset

We artificially generate training data by executing path planners on ground-truth reward maps, and probabilistically choosing paths based on their obtained rewards. In this experiment, we used 100 training paths for lane-change tasks and 45 paths for turning tasks at intersections. In the learning phase, we used 80 paths for training and 20 paths for the test on lane-change tasks. For turning at the intersection, we used 35 paths for training and 15 paths for the test.

E. Evaluation metrics

1) Modified Hausdorff distance: Modified Hausdorff distance (MHD [3]) is often used to evaluate the similarity between test paths and recovered paths on the learned reward function. MHD is an extension of Hausdorff distance to measure the distance of the time-sequential paths. The parameter β was set to 0.5 and 0.9. When the distance between related points was calculated and these values were sorted in ascending order, $\beta = 0.5$ represented the 50th percentile value, and $\beta = 0.9$ represented the 90th percentile value. Subsequently, we refer to them as MHD50 and MHD90.

2) Ground truth reward difference: We adopted another metric to evaluate the accuracy of the learned rewards. Even if the MHD is large, we cannot conclude that the recovered path is poor because of the suboptimality of driving behaviors. When the demonstration path τ_{demo}, a generated path τ_{gen}, and ground-truth weight $\boldsymbol{w}_{\mathrm{gt}}$ are given, the ground-truth reward difference is defined as $\boldsymbol{w}_{\mathrm{gt}}^{\top}\Big(\sum_{\tilde{\boldsymbol{x}}_t \in \tau_{\mathrm{demo}}} \boldsymbol{\phi}(\tilde{\boldsymbol{x}}_t) - \sum_{\boldsymbol{x}_t \in \tau_{\mathrm{gen}}} \boldsymbol{\phi}(\boldsymbol{x}_t)\Big)$. This metric is also often used to evaluate the stability of IRL when we know the ground-truth reward.

3) Calculation time: To evaluate the speed of the IRL methods, we measured the calculation times, and took their average. All methods use the same optimization metric, gradient descent, and the number of iterations is fixed at 30; therefore, we can directly compare the total calculation time.

F. RRT algorithm employed in the experiment

Thanks to the rigorous development of RRT based motion sampler, we need to take care of the selection of RRTs towards reliable and efficient RRT based maximum entropy continuous IRL. Among various kinds of RRT motion planners, we employ variants of template-based RRT generators presented by Ma et al. [11]. It is reported that this algorithm generates natural motion behavior with the combination of the registered motion templates while reducing its computational cost. Therefore, we chose this template-based algorithm in this experiment. It should be noted that we manually implement this algorithm for this experiment and modify it to the parallel computation to obtain the tree.

In the experiment, we use tens of motion templates to generate a tree where the number of nodes for RRT is 25200 on lane-change tasks and 31200 on intersection turning tasks, respectively. For importance sampling, we resampled 100 paths using (5). Specifically, this planner enabled us to obtain $T = 150$ length 100 paths from $q(\tau)$ within 20 [s] under this condition.

G. Comparison methods

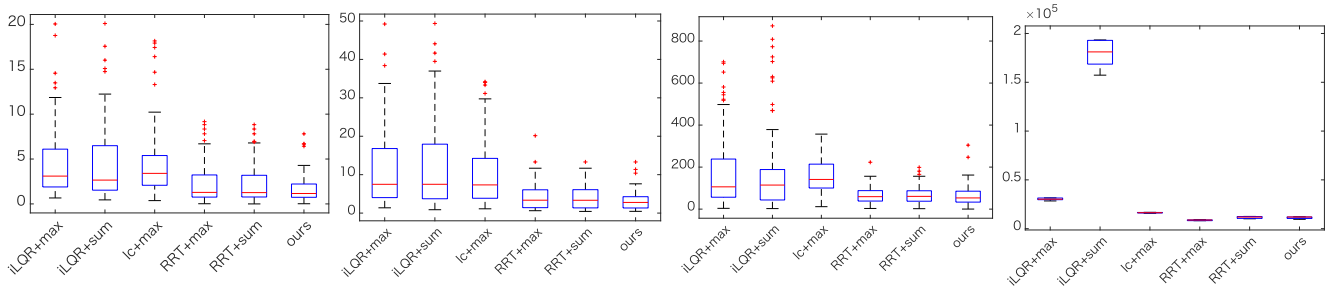
First, we adopted iLQR [2], [10] as a model-based planner. It uses a quadratic approximation around the initial path. In the existing research, Laplace approximation-based IRL [8] is often used, but it requires considerable memory to store the Hessian of each path. Hence, we substitute iLQR-based IRL for this Laplace approximation: 1) combination with the max path approximation (iLQR+max) and 2) with the sum sampling-based method (iLQR+sum). The iteration of iLQR planning is 20, and the sampled path size for the sum sampling is 5. Besides, 3) linear controller-based IRL (lc+max) proposed by Xin et al. [22] is selected. This is the baseline calculation time a very simple planner. We generated 7000 paths at each iteration. We also used 4) RRT with the max-path (RRT+max) and 5) with the sum sampling (RRT+sum) to evaluate the effects of our importance-sampling-based approach. RRT+max is a model similar to RRT*+perceptron by Shialis et al. [16], except that they employ RRT* [5] for holonomic dynamics, and RRT+sum is a model similar to Wu et al. [21] in terms of how log partition function is approximated while their sampling method is based on discrete the elastic band.

H. Results

Fig. 2 shows the quantitative evaluation results on lane change tasks. Our proposed method achieves the best result on MHD and the reward difference. As for calculation time, our method works 2.5 and 1.5 times faster than iLQR+max and lc+max, respectively. Fig. 3 shows the result of turning tasks at the intersection. Our proposed method also outperforms the other methods in terms of stability. The computational time is 2.2 and 1.2 times less than iLQR+max and lc+max, respectively. In addition, the computational time of iLQR+sum linearly increases from iLQR+max because of five executions of the planner to sample five paths. On the other hand, our proposed method only requires a 1.5 times larger cost than RRT+max, even though it samples 100 paths.

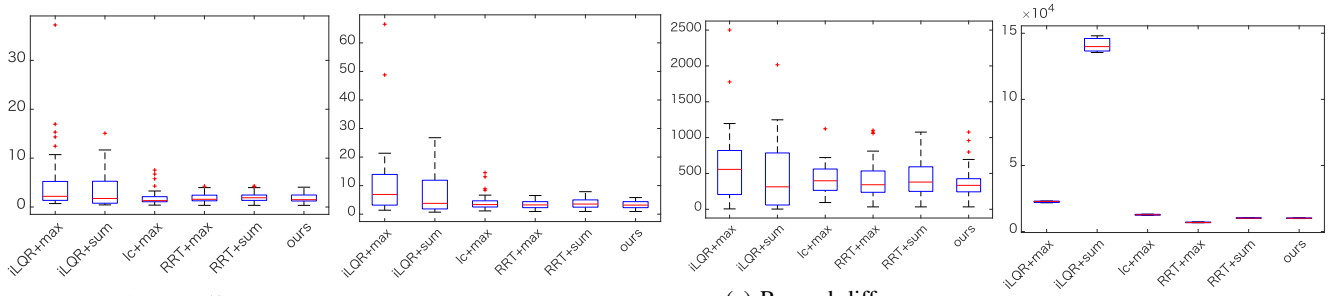
We add Table I and II, which compare the results among RRT-based methods on both tasks to compensate for Fig. 2 and Fig. 3. We can confirm that our importance sampling-based parameter updating is effective and achieves better results on average than other approaches: max-path approximation and sum-sampling-based approximation.

Fig. 4 is the result of the situation when the number of lanes is four. The data include demonstrations that pass the left side of the road as in Fig. 4a. Nevertheless, our method has successfully recovered another optimal behavior that avoided obstacles by the right-side as 4c, owing to the suboptimality-reflected optimization by importance sampling. The max path-based approximation of lc+max cannot learn suboptimal principles as in Fig. 4e because it only



(a) MHD50 (b) MHD90 (c) Reward difference (d) Calculation time [s]

Fig. 2: Results of four evaluation metrics on lane-change tasks



(a) MHD50 (b) MHD90 (c) Reward difference (d) Calculation time [s]

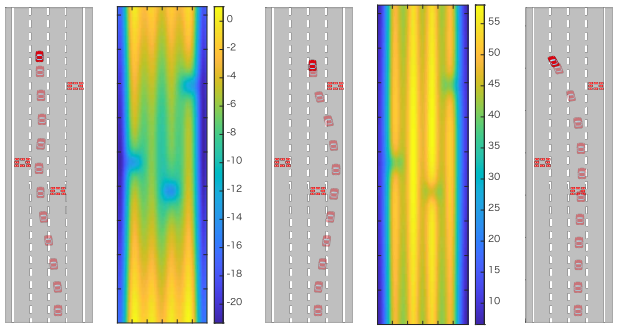
Fig. 3: Results of four evaluation metrics on intersection tasks

TABLE I: Evaluation among RRT-based methods on lane-change tasks

	RRT+max	RRT+sum	ours
MHD50	2.183 ± 2.118	2.087 ± 2.05	$\mathbf{1.744 \pm 1.538}$
MHD90	3.952 ± 3.248	3.831 ± 2.911	$\mathbf{3.449 \pm 2.574}$
reward diff	68.98 ± 42.09	67.04 ± 41.68	$\mathbf{66.49 \pm 50.13}$
time [$\times 10^3$s]	$\mathbf{8.68 \pm 0.46}$	11.53 ± 1.08	11.27 ± 1.10

TABLE II: Evaluation among RRT-based methods on intersection tasks

	RRT+max	RRT+sum	ours
MHD50	1.932 ± 0.977	2.037 ± 0.961	$\mathbf{1.881 \pm 0.949}$
MHD90	3.481 ± 1.477	3.882 ± 1.643	$\mathbf{3.320 \pm 1.237}$
reward diff	414.32 ± 259.74	431.45 ± 231.73	$\mathbf{381.09 \pm 214.54}$
time [$\times 10^3$s]	$\mathbf{6.95 \pm 0.34}$	10.19 ± 0.33	10.12 ± 0.30



(a) behavior (demo) (b) reward (ours) (c) behavior (ours) (d) reward (lc+max) (e) behavior (lc+max)

Fig. 4: Comparison of lane-change behaviors (4 lanes)

uses a single path to update the parameter. lc+max could not recover an effective reward due to the unstable planning; therefore, the predicted behavior passed through the obstacle.

V. Conclusion

Recently, IRL is believed to be one of the prominent approaches for robust reward designing in driving behavior prediction through driver demonstration. Despite the rigorous extension of maximum entropy IRL in continuous setting, this paper clarifies that the exploration towards robust and reliable motion planning and its full exploitation of the planned results still exists. In this paper, we leverage RRT as an efficient motion planner with a highly efficient sampler from the single generated RRT during IRL training. Though the proposed model is still simple, its efficiency and stability could be achieved in contrast to the existing approximation methods. The experimental results on artificial highways and intersections show that the proposed method achieves better performance in terms of stability and speed in multiple driving tasks. Future work includes the validity of our algorithm in time-variant situations where pedestrians pass intersections with time-variant reward functions.

References

[1] A. Bemporad and M. Morari, "Robust model predictive control: A survey." Springer, 1999.

[2] J. Chen, W. Zhan, and M. Tomizuka, "Constrained iterative lqr for on-road autonomous driving motion planning," in *Proc. ITSC*, 2017.

[3] M.-P. Dubuisson and A. K. Jain, "A modified hausdorff distance for object matching," in *Proc. of ICPR*, 1994.

[4] C. Finn, S. Levine, and P. Abbeel, "Guided cost learning: Deep inverse optimal control via policy optimization," in *Proc. of ICML*, 2016.

[5] S. Karaman and E. Frazzoli, "Incremental sampling-based algorithms for optimal motion planning," *Robotics Science and Systems VI*, 2010.

[6] S. M. LaValle, "Rapidly-exploring random trees: A new tool for path planning," 1998.

[7] A. J. Leslie, "Analysis of the field effectiveness of general motors production active safety and advanced headlighting systems," University of Michigan, Ann Arbor, Transportation Research Institute, Tech. Rep., 2019.

[8] S. Levine and V. Koltun, "Continuous inverse optimal control with locally optimal examples," in *Proc. of ICML*, 2012.

[9] ——, "Guided policy search," in *Proc. of ICML*, 2013.

[10] W. Li and E. Todorov, "Iterative linear quadratic regulator design for nonlinear biological movement systems." in *Proc. of ICINCO*, 2004.

[11] L. Ma, J. Xue, K. Kawabata, J. Zhu, C. Ma, and N. Zheng, "A fast RRT algorithm for motion planning of autonomous road vehicles," in *Proc. of ITSC*, 2014.

[12] V. Milanés, S. E. Shladover, J. Spring, C. Nowakowski, H. Kawazoe, and M. Nakamura, "Cooperative adaptive cruise control in real traffic situations," *IEEE Trans. on ITS*, 2013.

[13] A. Perez, R. Platt, G. Konidaris, L. Kaelbling, and T. Lozano-Perez, "LQR-RRT*: Optimal sampling-based motion planning with automatically derived extension heuristics," in *Proc. of ICRA*, 2012.

[14] N. D. Ratliff, J. A. Bagnell, and M. A. Zinkevich, "Maximum margin planning," in *Proc. of ICML*, 2006.

[15] D. M. Saxena, S. Bae, A. Nakhaei, K. Fujimura, and M. Likhachev, "Driving in dense traffic with model-free reinforcement learning," in *Proc. of ICRA*, 2020.

[16] K. Shiarlis, J. Messias, and S. Whiteson, "Rapidly exploring learning trees," in *Proc. of ICRA*, 2017.

[17] M. Shimosaka, T. Kaneko, and K. Nishi, "Modeling risk anticipation and defensive driving on residential roads with inverse reinforcement learning," in *Proc. of ITSC*, 2014.

[18] M. Shimosaka, K. Nishi, J. Sato, and H. Kataoka, "Predicting driving behavior using inverse reinforcement learning with multiple reward functions towards environmental diversity," in *Proc. of IV*, 2015.

[19] M. Shimosaka, J. Sato, K. Takenaka, and K. Hitomi, "Fast inverse reinforcement learning with interval consistent graph for driving behavior prediction," in *Proc. of AAAI*, 2017.

[20] D. J. Webb and J. Van Den Berg, "Kinodynamic RRT*: Asymptotically optimal motion planning for robots with linear dynamics," in *Proc. of ICRA*, 2013.

[21] Z. Wu, L. Sun, W. Zhan, C. Yang, and M. Tomizuka, "Efficient sampling-based maximum entropy inverse reinforcement learning with application to autonomous driving," *IEEE RAL*, 2020.

[22] L. Xin, S. E. Li, P. Wang, W. Cao, B. Nie, C.-Y. Chan, and B. Cheng, "Accelerated inverse reinforcement learning with randomly pre-sampled policies for autonomous driving reward design," in *Proc. of ITSC*, 2019.

[23] F. Zaklouta and B. Stanciulescu, "Warning traffic sign recognition using a hog-based kd tree," in *Proc. of IV*, 2011.

[24] B. D. Ziebart, A. Maas, J. A. Bagnell, and A. K. Dey, "Maximum entropy inverse reinforcement learning," in *Proc. of AAAI*, 2008.