Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2024.0429000

Stable Inverse Reinforcement Learning via Leveraged Guided Motion Planner for Driving Behavior Prediction

Minglu Zhao [1] and Masamichi Shimosaka [1]

[1]Department of Computer Science, Institute of Science Tokyo, Tokyo, Japan. (e-mail: {zhao, simosaka}@miubiq.cs.titech.ac.jp)

Corresponding author: Minglu Zhao (e-mail: zhao@miubiq.cs.titech.ac.jp).

This work was partially supported by JST SPRING Japan Grant Number JPMJSP2180, and JSPS KAKENHI Grant Numbers 21H03517, 23H00214 and 24K03015.

ABSTRACT Driving behavior prediction has become increasingly important, owing to rapid advancements in autonomous vehicle technologies. Inverse reinforcement learning (IRL) has emerged as a leading approach domain, as it allows the inference of underlying reward functions from human-driving demonstrations, enabling the modeling of complex behaviors. Among various IRL approaches, maximum-entropy IRL (MaxEnt IRL) has gained prominence in driving-behavior modeling owing to its applicability to continuous state-space problems. In continuous state-space MaxEnt IRL, stochastic motion planners are commonly employed during training to approximate the partition function because integrating over the entire state-space is computationally infeasible. However, traditional motion planners often fail to efficiently explore the state-space surrounding human-demonstrated paths, leading to inaccurate approximations of the partition function. This study proposes a novel, stable MaxEnt IRL framework that integrates an IRL-aware motion planner incorporating a guided exploration and exploitation process to efficiently sample high-quality trajectories. The proposed approach leverages two distributions derived from human-demonstrated paths to balance broad state-space exploration and targeted exploitation of relevant regions. Experiments conducted in a driving simulator demonstrate that the proposed method outperforms existing IRL methods in stability and accuracy, enhancing the prediction of driving behaviors and showcasing the potential of IRL for achieving human-like decision-making in autonomous driving.

INDEX TERMS Inverse reinforcement learning, Guided motion planning, Driving behavior prediction

I. INTRODUCTION

The rapid development of autonomous driving systems, designed to mitigate car accidents [1], [2], has fueled a growing need for accurate driving-behavior prediction. Many accidents are rooted in the complexity and variability of human-driving behavior, particularly when humans control vehicles in challenging environments [3]. Accurate driving-behavior prediction necessitates not only understanding driver actions but also modeling the underlying decision-making processes to enhance safety.

Inverse reinforcement learning (IRL) has emerged as prominent technique for modeling this decision-making process. It captures intricate driving behaviors by learning directly from human-driving demonstrations [4], [5], providing a data-driven approach for learning reward functions—indicators of preferred behaviors—without manual specification.Various IRL methods have been developed over the past two decades, including max-margin IRL [6] and Bayesian IRL [7].

Among these approaches, maximum-entropy IRL (MaxEnt IRL) [8], [9] has garnered significant attention owing to its suitability for continuous state-space problems [10]. This capability is crucial for driving-behavior prediction because discretizing the state-space often disrupts the balance between accuracy and computational cost [11]. Recent research has successfully applied MaxEnt IRL to autonomous driving in continuous state-space environments [12], [13].

A key challenge in continuous state-space MaxEnt IRL is accurately approximating the partition function. Calculating the reward function requires an intractable integral over all possible paths [14]. This partition function represents the sum of probabilities for all possible paths, a critical element in normalizing the probability distribution of behaviors. Direct computation of this integral is computationally prohibitive in continuous environments owing to the infinite state-space. Thus, researchers have explored various approximation methods, including Laplace approximation [14] and importance sampling [15].

Despite these developments, the performance of existing approximation methods remains inadequate for practical driving-behavior modeling. Most approaches employ efficient motion planners to sample paths based on the current reward function. However, the quality of these sampled paths and stability of the approximation remain insufficient in the context of driving-behavior prediction. Among the motion planners discussed in the literature, rapidly exploring random tree (RRT) [16] demonstrates potential for modeling driving behaviors [17], [18] but falls short in complex driving scenarios.

Recent advancements have focused on enhancing efficiency by introducing guided distributions to traditional motion planners [19]–[23]. These *guided motion planners* use distributions derived from human-demonstrated paths to steer the exploration, as depicted in Fig. 1b. Unlike traditional motion planners that explore the state-space randomly, guided motion planners concentrate on more relevant regions, generating feasible paths that resemble human-demonstrated ones.

However, integrating guided motion planners into IRL poses difficulties in generating optimal paths owing to the evolving nature of reward functions during training. Motion planners in IRL approximate the partition function by sampling paths that maximize the total reward. However, because the reward function is iteratively refined, paths generated in the early training phases may be suboptimal. This can result in inaccurate partition function approximations, potentially causing the divergence of the gradient of the reward function and hindering the training process.

To address these limitations, this paper proposes a novel, stable MaxEnt IRL framework incorporating an IRL-aware guided motion planner to enhance both the accuracy and stability of partition function approximation in continuous state-spaces. To facilitate optimal path generation, even with preliminary reward functions, a mechanism is introduced to redirect exploration toward demonstrations when deviations occur, as shown in Fig. 4. Two specific distributions are used to balance broad state-space exploration and targeted exploitation: a broad distribution for wider exploration and a focused distribution to guide the search toward critical regions near the demonstrations. By efficiently sampling high-quality paths using this IRL-aware guided motion planner, the proposed model enhances the optimization of the reward function, leading to more stable and accurate driving-behavior predictions.

The main contributions of this study can be summarized as follows:

- A novel continuous state-space MaxEnt IRL approach leveraging an IRL-aware guided motion planner is proposed for enhanced stability and accuracy is proposed. Two types of guided distributions are used to balance broad state-space exploration with targeted exploitation.
- The proposed method, validated in simulated driving scenarios, demonstrates improved effectiveness compared with conventional IRL methods.

The remaining paper is organized as follows: Section II reviews the related literature. Section III outlines the problem setting for sampling-based IRL and highlights the limitations of traditional motion planners in the IRL domain. Section IV details the proposed algorithm. Section V describes the experimental setup, presents the results, and discusses the limitations of this study. Finally, Section VI presents the concluding remarks.

II. RELATED WORK

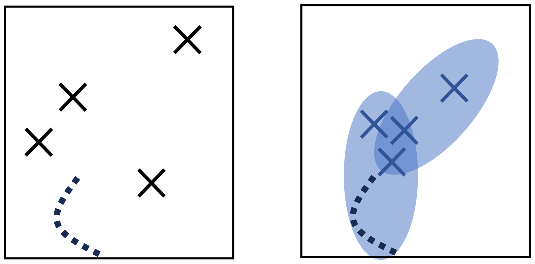
This section provides a review of relevant literature on continuous state-space IRL and guided motion planning methods.

A. CONTINUOUS STATE-SPACE IRL

Continuous IRL aims to optimize reward functions within continuous state-spaces, eliminating the need for discretization. However, the intractability of the integral in the partition function necessitates the approximation of the function within the IRL model [14]. Several approaches have been developed to address this approximation challenge. Shiarlis et al. [17] obtained the highest-rewarded path without planning all possible paths; however, relying on a single path is insufficient for capturing the suboptimal nature of real-world human-driving behaviors. Finn et al. [15] employed importance sampling to approximate the partition function, but their method exhibited instability, requiring numerous trial-and-error iterations. Hosoma et al. [18] introduced RRT as a motion planner to aid importance sampling. Nevertheless, a naive motion planner struggles to explore the entire state-space efficiently, leading to unstable partition function approximations. These methods typically involve a trade-off between the approximation accuracy and the number of exploration iterations. Therefore, this study is aimed at enhancing the solution efficiency by leveraging the advantages of guided motion planners to achieve a more accurate and stable partition function approximation.

B. GUIDED MOTION PLANNING

As illustrated in Fig. 1a, traditional motion planners face significant challenges in efficiently exploring the vastness of continuous state-spaces. In contrast, guided motion planning methods offer a promising solution by incorporating human-driving demonstrations for guide-path generation. Fig. 1b illustrates how guided motion planners leverage probabilistic distributions to sample the next state more efficiently. Several studies have built upon this foundational concept. For example, Ye et al. [20] extracted motion features from human demonstrations and sampled paths based on a statistical estimation of these features. Cheng et al. [24] developed a sampling heuristic to minimize the Kullback–Leibler-divergence between sampled paths and demonstrations. Wang



(a) Traditional motion planner. (b) Guided motion planner.

FIGURE 1: Comparison of traditional and guided motion planning. Black dots indicate the currently planned trajectory and crosses mark the candidates of the next states. Blue circles highlight the guided regions derived from human-demonstrated trajectories. (a) A traditional motion planner randomly samples the next state across the entire state-space, often resulting in low-quality trajectories. (b) A guided motion planner samples the next state using guided distributions, enabling the generation of higher-quality, feasible trajectories near demonstrations.

et al. [23] provided a non-parametric solution employing Gaussian mixture regression within a traditional RRT motion planner to generate high-quality paths. Zwane et al. [25] applied a safety penalty to the reward function and sampled paths using Gaussian processes.

However, these existing guided motion planners do not inherently guarantee the generation of high-quality paths when applied to the IRL training process, where the reward functions are iteratively updated. An IRL-aware guided motion planner requires an additional mechanism to guide the exploration back toward the demonstrations when the generated paths deviate significantly.

III. FORMULATION OF SAMPLING-BASED IRL

This section outlines the problem setting for MaxEnt IRL in a continuous state-space and reviews existing approaches for approximating the partition function in such settings.

A. PROBLEM SETTING

We define a continuous state-space, $\mathcal{S}$, and a continuous action-space, $\mathcal{A}$. The agent takes an action $\boldsymbol{a}_t \in \mathcal{A}$ at a discrete time t according to the motion dynamics function T, transitioning from the current state $\boldsymbol{s}_t \in \mathcal{S}$ to the next state $\boldsymbol{s}_{t+1} = T(\boldsymbol{a}_t, \boldsymbol{s}_t)$. In driving-behavior modeling, representing a high-dimensional state-space is necessary, including parameters such as vehicle positions, velocities, and angles [18].

A path consists of a time-ordered sequence of action–state pairs over a time horizon h, represented as $\tau = \{(\boldsymbol{a}_1, \boldsymbol{s}_1), \ldots, (\boldsymbol{a}_h, \boldsymbol{s}_h)\}$. The quality of an action taken at a particular state is evaluated using an immediate reward function $r_{\boldsymbol{w}}(\boldsymbol{a}_t, \boldsymbol{s}_t)$, parameterized by the vector $\boldsymbol{w}$. IRL aims to optimize $\boldsymbol{w}$ based on human-driving demonstrations.

B. MAXENT IRL IN A CONTINUOUS STATE-SPACE

Continuous MaxEnt IRL is particularly effective for driving-behavior predictions as it accounts for the inherent suboptimality in human demonstrations. In MaxEnt IRL, the probability of path τ is modeled as [14]

$$p(\tau;\boldsymbol{w}) = \frac{\exp\left(\sum_{(\boldsymbol{a}_t,\boldsymbol{s}_t)\in\tau} r_{\boldsymbol{w}}\left(\boldsymbol{a}_t,\boldsymbol{s}_t\right)\right)}{Z(\boldsymbol{w})}, \tag{1}$$

where $Z(\boldsymbol{w})$ is the partition function, defined as the sum of exponentiated rewards over all possible paths

$$Z(\boldsymbol{w}) = \int \exp\left(\sum_{(\tilde{\boldsymbol{a}}_t,\tilde{\boldsymbol{s}}_t)\in\tilde{\tau}} r_{\boldsymbol{w}}\left(\tilde{\boldsymbol{a}}_t,\tilde{\boldsymbol{s}}_t\right)\right) \mathrm{d}\tilde{\tau}. \tag{2}$$

The loss function is the negative log-likelihood summed over all human-demonstrated paths, defined as follows:

$$\begin{aligned} L(\boldsymbol{w}) &= \sum_{\tau\in D_{demo}} -\log p(\tau;\boldsymbol{w}) \\ &= \sum_{\tau\in D_{demo}} \left(\log Z(\boldsymbol{w}) - \sum_{(\boldsymbol{a}_t,\boldsymbol{s}_t)\in\tau} r_{\boldsymbol{w}}\left(\boldsymbol{a}_t,\boldsymbol{s}_t\right)\right). \end{aligned} \tag{3}$$

C. PARTITION FUNCTION APPROXIMATION USING A MOTION PLANNER

Among the various approaches for partition function approximation, Finn et al. [15] introduced an importance sampling method, aiming for greater reliability. This method approximates the partition function using a set of paths sampled from a motion planner as follows:

$$Z(\boldsymbol{w}) \approx \frac{1}{|D_{samp}|} \sum_{\tau\in D_{samp}} \frac{\exp\left(\sum_{(\boldsymbol{a}_t,\boldsymbol{s}_t)\in\tau} r_{\boldsymbol{w}}\left(\boldsymbol{a}_t,\boldsymbol{s}_t\right)\right)}{q(\tau)}, \tag{4}$$

where $q(\tau)$ represents the auxiliary density function employed for path sampling and D_{samp} denotes the set of paths sampled from $q(\tau)$.

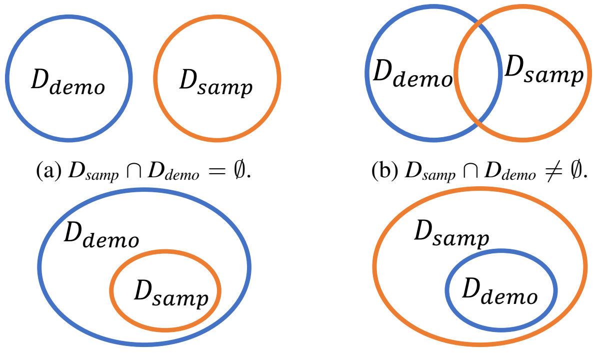
Hosoma et al. [18] applied an RRT planner to enhance the efficiency of path sampling in a continuous state-space. The density function $q(\tau)$ for path τ is defined as

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

where D_{all} is the set of all possible paths within the state-space. A subset of these paths, denoted as D_{samp}, is sampled using $q(\tau)$ to approximate the partition function described in (4).

Using the approximated partition function, the gradient of the MaxEnt IRL loss function can be expressed as

$$\nabla_{\boldsymbol{w}} L(\boldsymbol{w}) = \frac{1}{W} \sum_{\tau\in D_{samp}} \frac{\exp\left(r_{\boldsymbol{w}}\left(\tau\right)\right)}{q(\tau)} \frac{\mathrm{d}r_{\boldsymbol{w}}(\tau)}{\mathrm{d}\boldsymbol{w}} - \sum_{\tilde{\tau}\in D_{demo}} \frac{\mathrm{d}r_{\boldsymbol{w}}(\tilde{\tau})}{\mathrm{d}\boldsymbol{w}}, \tag{6}$$



(a) $D_{samp} \cap D_{demo} = \emptyset$.

(b) $D_{samp} \cap D_{demo} \neq \emptyset$.

(c) $D_{samp} \subset D_{demo}$.

(d) $D_{demo} \subset D_{samp}$.

FIGURE 2: Venn diagrams illustrating the relationships between the sets of demonstrated and sampled trajectories.

where W is defined as

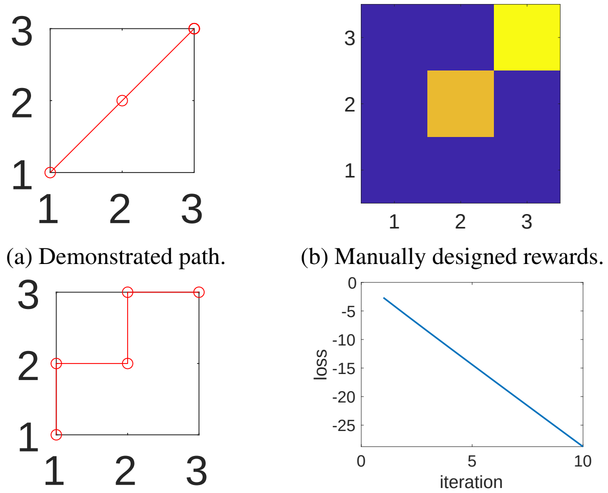
$$W = \sum_{\tau \in D_{samp}} \frac{\exp\left(r_{\boldsymbol{w}}\left(\tau\right)\right)}{q(\tau)}. \tag{7}$$

The marginal reward for a path, defined as $r_{\boldsymbol{w}}(\tau) = \sum_{(\boldsymbol{a}_t, \boldsymbol{s}_t) \in \tau} r_{\boldsymbol{w}}\left(\boldsymbol{a}_t, \boldsymbol{s}_t\right)$, reflects the rewarded *visitation frequencies* of the action–state pairs along that path. This gradient effectively quantifies the difference between the expected visitation frequencies, inferred from the sampled paths, and empirical visitation frequencies present in the demonstrations. The loss function converges when the learned visitation frequencies align with those observed in the demonstrations, thus ensuring that the agent's behavior mirrors the demonstrated actions.

D. LIMITATIONS OF TRADITIONAL MOTION PLANNERS FOR IRL

When coupled with traditional motion planners, continuous IRL struggles to accurately model the decision-making process owing to inadequate motion transition probabilities in the sampled paths. Traditional planners require extensive time to thoroughly explore the continuous state-space. Without sufficient exploration, the generated paths fail to adequately represent the demonstrated ones. However, importance-sampling-based partition function approximation assumes that the density function $(q(\tau))$ of all sampled paths sufficiently represents the distribution of all possible paths, including those demonstrated. Consequently, inadequate motion transitions in sampled paths generated by traditional planners lead to inaccurate partition function approximations, introducing instability into the IRL model.

Fig. 2 illustrates the consequences of inadequate motion transition probabilities based on four logical relationships between the demonstrated and sampled sets of paths. In Fig. 2a, a significant mismatch exists between the sampled and demonstrated paths. This discrepancy not only biases the partition function approximation but also leads to mismatched visitation frequencies between the two sets. Because the gradient of the loss function, defined in (6), aims to align the visitation frequencies of both sets, such a discrepancy induces large divergences in the gradient, destabilizing the IRL model. Fig. 2b depicts a scenario with non-disjoint sets of D_{samp} and D_{demo}. Even in this case, if the probability distribution of the sampled paths, $p(\tau_{samp})$, differs significantly from that of the demonstrated paths, $p(\tau_{demo})$, the partition function approximation remains biased. Conversely, Fig. 2c illustrates a scenario where the IRL model overfits the demonstrations. Finally, Fig. 2d presents the ideal case: $p(\tau_{samp})$ sufficiently covers $p(\tau_{demo})$, enabling effective convergence through gradient-based optimization.



(a) Demonstrated path.

(b) Manually designed rewards.

(c) Generated path.

(d) Loss function.

FIGURE 3: Results from a simplified toy IRL model using a traditional motion planner, illustrating instability.

Fig. 2b further illustrates the instability of the IRL model using a simple toy example in a two-dimensional, 3×3 discrete state-space, focusing on the issues arising from partially overlapping sets. In this toy model, the challenge of inadequate motion transition probabilities is simulated by restricting the demonstrated motions to diagonal transitions between states, as shown in Fig. 3a, whereas generated motions are limited to four directions: up, down, left, and right, as depicted in Fig. 3c. The objective is to pass through the waypoint at (2,2) and reach the goal at (3,3), consistent with the reward functions illustrated in Fig. 3b. During IRL training, the mismatch in motion transitions and visitation frequencies between these partially overlapping sets causes instability. As shown in Fig. 3d, the loss function diverges in this toy model, highlighting another critical issue for IRL. According to the definition of the loss function in (3), its value should always remain non-negative. This is because the partition function should be larger than the sum of the exponentiated rewards, i.e., $\log Z(\boldsymbol{w}) > \sum_{(\boldsymbol{a}_t, \boldsymbol{s}_t) \in \tau} r_{\boldsymbol{w}}\left(\boldsymbol{a}_t, \boldsymbol{s}_t\right)$, as defined in [8]. Therefore, inadequate motion transitions result in insufficient partition function approximations, leading to an unstable IRL model.

In contrast, guided distributions enable motion planners to sample action–state pairs more closely aligned with the

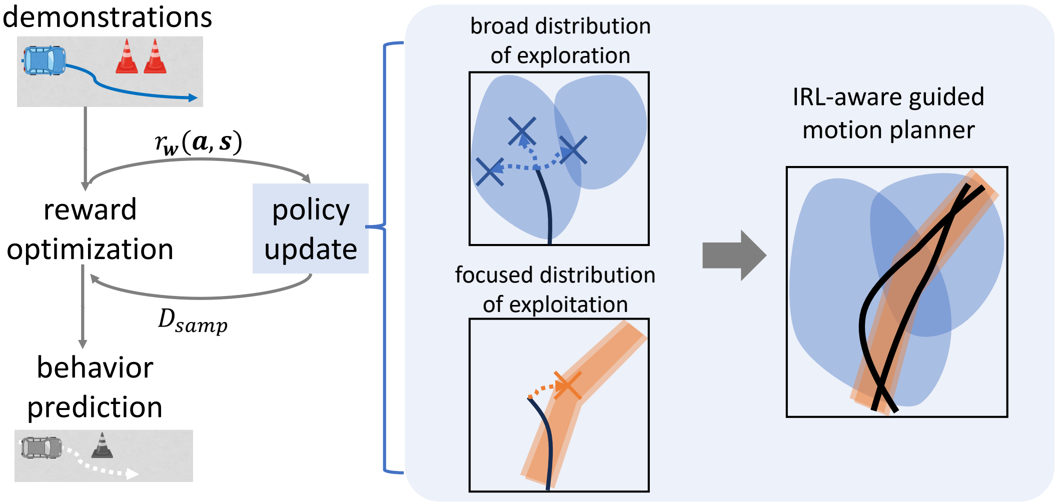

demonstrations
$r_w(a, s)$
reward optimization
policy update
D_{samp}
behavior prediction
broad distribution of exploration
focused distribution of exploitation
IRL-aware guided motion planner

FIGURE 4: Overview of the proposed continuous state-space MaxEnt IRL framework utilizing IRL-aware guided motion planners. The right panel illustrates the use of two distribution types, broad and focused, within the motion planner to balance broad state-space exploration with targeted exploitation. Black lines represent the planned trajectory, while crosses indicate the candidates of the next states. The broad distribution promotes exploration within guided regions, and the focused distribution directs sampling toward demonstrated trajectories. This approach facilitates the generation of high-quality paths for partition function approximation.

demonstrated regions. This improved alignment leads to more accurate state visitation frequencies and, consequently, better model convergence.

IV. IRL LEVERAGING GUIDED MOTION PLANNING

A. OVERVIEW

This study introduces an IRL-aware guided motion planner to achieve stable and accurate partition function approximation within an IRL framework. Our approach builds upon existing methods by incorporating a mechanism specifically targeting exploitation near demonstrations. This crucial addition allows for optimal path generation even with imperfect, evolving reward functions. The core innovation, illustrated in Fig. 4, lies in achieving a dynamic balance between broad exploration and focused exploitation during the path-sampling process. This carefully calibrated balance results in a more stable and reliable IRL model, particularly beneficial for driving-behavior prediction.

The subsequent sections are organized as follows: Section IV-B introduces the general concept of guided motion planners. Section IV-C discusses the specific requirements for a guided motion planner to be considered "IRL-aware". Section IV-D details the design of the proposed IRL-aware guided motion planner. Lastly, Section IV-E summarizes the proposed stable IRL algorithm.

B. MOTION PLANNING WITH GUIDED DISTRIBUTIONS

Traditional motion planners aim to sample paths, $\tau \sim p(\tau)$, according to a distribution, $p(\tau)$, defined by the following Markov property [26]

$$\begin{aligned} p(\tau) &= p(\boldsymbol{a}_{1:h}, \boldsymbol{s}_{1:h}) \\ &= p(\boldsymbol{a}_{1:h-1}, \boldsymbol{s}_{1:h-1}) p(\boldsymbol{s}_h|\boldsymbol{a}_{h-1}, \boldsymbol{s}_{h-1}) p(\boldsymbol{a}_h|\boldsymbol{s}_h), \end{aligned} \tag{8}$$

where $p(\boldsymbol{a}_t|\boldsymbol{s}_t)$ represents the policy. At each timestep t, traditional motion planners randomly sample a new action–state pair $(\boldsymbol{a}_t, \boldsymbol{s}_t)$ to construct a feasible path.

In contrast, guided motion planners [21], [23] leverage prior knowledge, such as human demonstrations, to sample new action–state pairs conditioned on environmental factors: $(\boldsymbol{a}_t, \boldsymbol{s}_t|\boldsymbol{c}_t)$. Here, $\boldsymbol{c}_t$ represents the environmental factors influencing human decision-making. For example, a human driver would stop if the traffic light turned red. Thus, guided motion planners sample paths according to the following probabilistic distribution:

$$p(\boldsymbol{a}_{1:h}, \boldsymbol{s}_{1:h}|\boldsymbol{c}_{1:h}) = \prod_{t=1}^{h} p(\boldsymbol{a}_t, \boldsymbol{s}_t|\boldsymbol{c}_t) p_{\text{guide}}(\boldsymbol{a}_t|\boldsymbol{c}_t) p_{\text{guide}}(\boldsymbol{s}_t|\boldsymbol{c}_t). \tag{9}$$

For clarity, we will use the term *guided distribution* to refer to a probabilistic distribution designed to steer the path-sampling process. Guided distributions encourage the sampling of new action–state pairs in high-reward regions, facilitating the generation of high-quality paths. Notably, actions are assumed to be uniformly sampled (i.e., $p_{\text{guide}}(\boldsymbol{a}_t|\boldsymbol{c}_t) \propto 1$), and the guided distributions within the state-space are explored.

C. IRL-AWARE MOTION PLANNER REQUIREMENTS

As discussed in Section III-D, a critical requirement for an IRL-aware motion planner is to ensure sufficient overlap between the distribution of sampled paths, $p(\tau_{samp})$, and that of demonstrated paths, $p(\tau_{demo})$. This overlap, essential for accurate partition function approximation and stable, gradient-based optimization, stems directly from the fundamental definitions of the partition and loss functions within the IRL framework. The partition function approximation, as defined in (4), depends on a set of paths, D_{samp}, sampled by the motion planner. The probabilities of these sampled paths, $p(\tau_{samp})$, are then used to compute the IRL loss function, defined in (3). The core objective of MaxEnt IRL is to minimize this loss function. This minimization is constrained by the requirement that the expected visitation frequencies of the model match those observed in the demonstration data. This constraint is formally expressed as in (6.1) in [9]:

$$\mathbb{E}_{\tau \sim p(\tau_{samp})}[r_{\mathbf{w}}(\tau)] - \mathbb{E}_{\tilde{\tau} \sim p(\tau_{demo})}[r_{\mathbf{w}}(\tilde{\tau})] = 0. \tag{10}$$

Notably, significant deviations between the sampled paths D_{samp} and demonstrated paths D_{demo} (e.g., if $D_{samp} \cap D_{demo} = \emptyset$, as shown in Fig. 2a), introduce bias into the importance-sampling-based partition function approximation. This bias, in turn, corrupts the gradient calculation, hindering effective optimization of the reward function parameters. This problem is particularly acute in continuous state-spaces, where a dominance of low-quality paths in D_{samp} can cause the model to diverge significantly from the demonstrated behaviors.

Existing guided motion planning methods often fail to meet the above mentioned requirement. The core issue is the tendency of sampled paths to deviate from demonstrations when suboptimal reward functions are used. While IRL relies on the motion planner to sample paths, aimed at maximizing the total reward of visited states, the iterative nature of training means that the reward function is continuously updated. Consequently, the rewards used for sampling at any given stage might still be suboptimal. Although guided motion planners incorporate distributions derived from demonstrations, they often struggle to efficiently explore the state-space surrounding demonstrated paths under these suboptimal reward conditions. This frequently results in the generation of partially disjoint sets of paths (as shown in Fig. 2b), leading to biased partition function approximations and unstable gradient updates.

D. IRL-AWARE GUIDED MOTION PLANNER DESIGN

This study proposes a novel IRL-aware guided motion planner to address the critical path deviation issue that arises during sampling in standard IRL. Our method extends existing approaches by incorporating a demonstration-targeted exploitation mechanism, which promotes the generation of high-quality paths even when using suboptimal reward functions. The key innovation lies in carefully balancing broad exploration with targeted exploitation during path sampling. This balance ensures that the sampled path set approximates the ideal coverage shown in Fig. 2d, while avoiding the overfitting scenario depicted in Fig. 2c. The proposed approach relies on RRT-based motion planners, given their proven effectiveness and efficiency in continuous state-space problems [18].

The demonstration-targeted exploitation mechanism operates by extending the initial RRT tree nodes with rewards that are discounted based on a predefined factor. In an RRT-based motion planner, tree nodes represent states, edges represent actions, and node values correspond to rewards. The proposed method initializes these nodes using states from the demonstrated paths. By applying discounted rewards, we relax the strict adherence to demonstrations, preventing the overly constrained sampling that would result in the situation observed in Fig. 2c. Without this discounting, sampled paths would remain too close to the demonstrations, limiting the necessary exploration.

To ensure broad exploration, we integrated Gaussian mixture model (GMM)-based methods [23], [25]. Specifically, we employ a Dirichlet process GMM (DPGMM) [27] to effectively model driving behaviors in a high-dimensional state-space. The DPGMM is particularly well-suited for this type of space because it automatically determines the optimal number of clusters, unlike traditional GMMs that require manual tuning. The Dirichlet process serves as a prior for the mixture components. The distribution of mixing coefficients over K components, $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_K)$, can be obtained from a Dirichlet distribution as follows:

$$p(\boldsymbol{\pi}) = \mathrm{Dir}(\boldsymbol{\pi}|\boldsymbol{\alpha}_0), \tag{11}$$

where $\boldsymbol{\alpha}_0$ can be interpreted as a prior number of observations associated with each component of the mixture. The mean $\boldsymbol{\mu}_k$ and precision $\boldsymbol{\Lambda}_k$ of each component follow a Gaussian–Wishart distribution:

$$(\boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k) \sim \mathcal{N}\left(\boldsymbol{\mu}_k|\boldsymbol{m}_k, (\beta_k\boldsymbol{\Lambda}_k)^{-1}\right)\mathcal{W}\left(\boldsymbol{\Lambda}_k|\boldsymbol{W}_k, \boldsymbol{\nu}_k\right), \tag{12}$$

where β_k, $\boldsymbol{m}_k$, $\boldsymbol{W}_k$, and $\boldsymbol{\nu}_k$ are the parameters updated via the expectation-maximization (EM) algorithm, as detailed in (10.60)–(10.63) in [28]. Finally, the path probability density function is defined as

$$\begin{aligned} p_{\text{guide}}(\boldsymbol{s}_t) &= \sum_{k=1}^{K} \boldsymbol{\pi}_k p(\boldsymbol{s}_t|k) \\ &= \sum_{k=1}^{K} \boldsymbol{\pi}_k \mathcal{N}(\boldsymbol{s}_t; \boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k). \end{aligned} \tag{13}$$

Algorithm 1 summarizes the tree-generation process for the proposed IRL-aware RRT-based motion planner. The algorithm requires several inputs: demonstrated path τ_{demo}, action–state space $(\mathcal{A}, \mathcal{S})$, goal region G_{goal}, trained DPGMM parameters $(\boldsymbol{\mu}, \boldsymbol{\Lambda})$, and discount factor γ. A tree $\mathcal{T}$ contains a node set V representing states, an edge set E representing state transitions, and a reward set R representing the cumulative rewards over a sequence of states. The tree initialization (Lines 3–5) involves incorporating the demonstrated path with discounted rewards using γ. In the main loop, the algorithm selects between random sampling (Line 8) or guided sampling (Line 10) to determine the next state. The threshold

Algorithm 1: Tree-generation process of IRL-aware guided RRT-based motion planner

```
Input: $\tau_{demo}, \mathcal{A}, \mathcal{S}, G_{goal}, \boldsymbol{\mu}, \boldsymbol{\Lambda}, \gamma, \Theta$
   // initialization
1  $\boldsymbol{s}_{init} \leftarrow \tau_{demo}$;
2  $V \leftarrow \boldsymbol{s}_{init}, E \leftarrow \emptyset, R \leftarrow r(\boldsymbol{s}_{init})$;
   // tree initialization using
       demonstrations
3  for $\boldsymbol{s}_t \in \tau_{demo}$ do
4      $r_{sum}(\boldsymbol{s}_t) \leftarrow \gamma \sum_{t'=1}^{t} r(\boldsymbol{s}_{t'})$;
5      $\mathcal{T} \leftarrow$ extend$(\boldsymbol{s}_t, e_t, r_{sum}(\boldsymbol{s}_t))$;
   // main loop
6  for $i \leftarrow 1$ to $maxIter$ do
       // guided exploration
7      if $Random(1) < \Theta$ then
8          $\boldsymbol{s}_{rand} \leftarrow$ UniformSample$(\mathcal{S})$;
9      else
10         $\boldsymbol{s}_{rand} \leftarrow$ NormalSample$(\boldsymbol{\mu}, \boldsymbol{\Lambda}, \mathcal{S})$      by (13);
11     $\boldsymbol{s}_{nearest} \leftarrow$ NearestState$(\mathcal{T}, \boldsymbol{s}_{rand})$;
12     $\boldsymbol{s}_{new}, e_{new} \leftarrow$ Steer$(\mathcal{A}, \boldsymbol{s}_{nearest})$;
13     $r_{sum}(\boldsymbol{s}_{new}) \leftarrow r_{sum}(\boldsymbol{s}_{nearest}) + \sum_{\boldsymbol{s} \in e_{new}} r(\boldsymbol{s})$;
       // tree expansion
14     if $\boldsymbol{s}_{new}, e_{new} \in \mathcal{S}$ then
15         $\mathcal{T} \leftarrow$ extend$(\boldsymbol{s}_{new}, e_{new}, r_{sum}(\boldsymbol{s}_{new}))$;
16         if $\boldsymbol{s}_{new} \in G_{goal}$ then
17             return $\mathcal{T} = (V, E, R)$;
18     $i \leftarrow i + 1$;
19 return failure;
```

Θ (Line 7) controls the balance between random and guided sampling. After a new state is sampled, the nearest existing node in the tree is identified (Line 11), and the algorithm steers toward the new state, following nonholonomic vehicle dynamics (Line 12). Notably, the inclusion of demonstrated states in the tree (Line 11) ensures that the planner can correct paths that have deviated, guiding them back toward the demonstrated trajectories, and thus reinforcing guided exploration.

Algorithm 2 summarizes the importance sampling procedure for approximating the partition function using the tree generated in Algorithm 1. Here, N represents the number of paths to be resampled, where $N = |D_{samp}|$. The IRL-aware guided motion planner ensures that the sampled paths are concentrated in regions of interest, thereby improving the quality of the partition function approximation and learning stability.

E. ALGORITHM

Our IRL algorithm combines the strengths of the IRL-aware guided motion planner and importance sampling. This combination yields a more reliable and efficient IRL framework that is particularly well-suited for high-dimensional, continuous state-spaces. The guided approach ensures that the learned reward function accurately captures the decision-making process underlying the demonstrated behaviors, thereby improving performance and stability in complex environments.

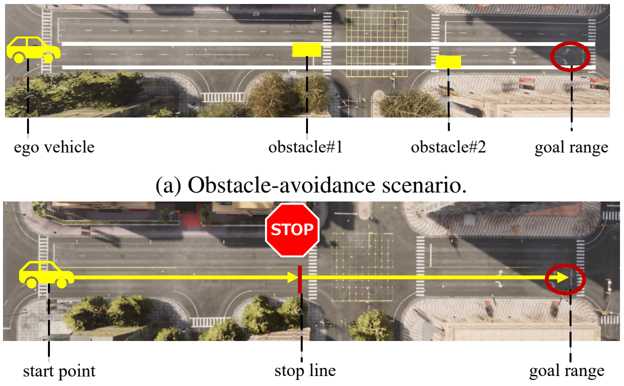
Algorithm 2: Importance sampling by IRL-aware guided motion planner

```
Input: $\mathcal{T}, G_{goal}, N$
// sampling all possible paths
$D_{all} \leftarrow \emptyset$;
for $\boldsymbol{s}_{goal} \in V$ and $\boldsymbol{s}_{goal} \in G_{goal}$ do
    $\tau \leftarrow$ backTrack$(\boldsymbol{s}_{goal}, V, E)$;
    $D_{all} \leftarrow D_{all} \cup \{\tau\}$;
$q \leftarrow$ getNormalizer$(D_{all})$      by (5);
// Resampling
if $|D_{all}| >= N$ then
    $D_{samp} \leftarrow$ UniformSample$(D_{all}, N)$;
    return $q, D_{samp}$;
return failure ;
```

Algorithm 3: Stable IRL via leveraged guided motion planner

```
Input: $D_{demo}, \mathcal{A}, \mathcal{S}, G_{goal}, \boldsymbol{\mu}, \boldsymbol{\Lambda}, \gamma, N$
// initialization
$\boldsymbol{w} \leftarrow$ randomlyInit;
// main loop
for $i \leftarrow 1$ to $maxIter$ do
    for $\tau \in D_{demo}$ do
        // RRT generation by Algorithm 1
        $\mathcal{T} \leftarrow$ guidedRRT$(\tau, \mathcal{A}, \mathcal{S}, G_{goal}, \boldsymbol{\mu}, \boldsymbol{\Lambda}, \gamma)$;
        // Path sampling by Algorithm 2
        $q, D_{samp} \leftarrow$ sampling$(\mathcal{T}, G_{goal}, N)$;
        // parameter update
        $\nabla_{\boldsymbol{w}} L(\boldsymbol{w}) \leftarrow$ getGrad$(q, D_{samp})$      by (6);
        $\boldsymbol{w} \leftarrow$ update$(\boldsymbol{w}, \nabla_{\boldsymbol{w}} L(\boldsymbol{w}))$;
    $i \leftarrow i + 1$;
return $\boldsymbol{w}$;
```

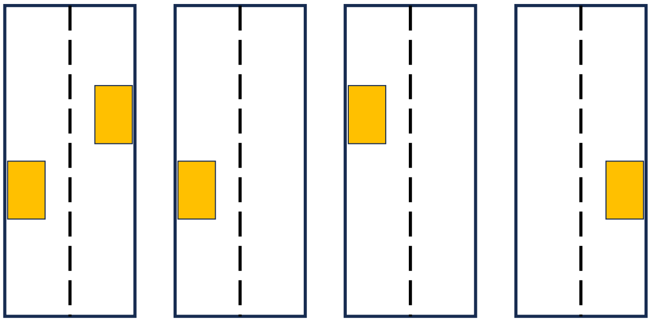
Algorithm 3 provides a comprehensive overview of the gradient-based optimization process for IRL, integrating importance sampling with an IRL-aware guided motion planner. This approach capitalizes on the insights gained from Algorithms 1 and 2, enabling efficient partition function approximation and stable gradient optimization in a continuous state-space. The overall guided IRL algorithm takes the demonstrated paths, D_{demo}, and other parameters necessary for sampling as inputs, as detailed in the pseudocodes of the algorithms. The primary goal of the process is to iteratively update the reward function parameters, $\boldsymbol{w}$, until the expected visitation frequencies from the model align with those observed in the demonstrations.



(a) Obstacle-avoidance scenario.

(b) Stop-line scenario.

FIGURE 5: Bird's-eye view of driving scenarios in the CARLA simulator.



(a) Scene 0. (b) Scene 1. (c) Scene 2. (d) Scene 3.

FIGURE 6: Obstacle configurations for each scene. One or two obstacles (yellow blocks) are placed on a two-lane road. (a) Scene 0: data used in Sections V-E and V-F. (b–d) Scenes 1–3: data used in Section V-G.

V. EXPERIMENTAL RESULTS

This section details the experimental setup. Sections V-A–V-E describe the driving scenarios, comparison methods, evaluation metrics, and learned DPGMM fitting to demonstrated data.

Subsequently, experimental results are presented for two types of driving environments: static and dynamic. Static environments represent fundamental real-world conditions, with environmental factors such as road geometry and obstacle positions assumed to be consistent over time Section V-F presents both quantitative and qualitative experimental results, followed by a detailed analysis of the findings. In contrast, dynamic environments feature variable environmental factors and are categorized into two types based on the driving complexity: contextual dynamic and fully dynamic environments. In contextual dynamic environments, environmental factors are time-invariant during path sampling but may differ between training and evaluation. As described in Section V-G, experiments were conducted to simulate a real-world use case where a trained model was applied to an unknown scenario while the overall road geometry remained unchanged. Fully dynamic environments align more closely to real-world instances, characterized by constantly changing elements, such as obstacles, pedestrians, and other vehicles. Section V-H presents a thorough discussion of the challenges associated with fully dynamic environments and potential extensions to enhance the adaptability of the proposed approach in such conditions.

A. DRIVING SCENARIOS

We simulated two distinct driving scenarios using the CARLA simulator [29], maintaining the same environmental configurations as those employed in [30].

The stop-line scenario, depicted in Fig. 5b, involves two primary tasks: 1) Bringing the vehicle to a complete stop at a designated stop line (zero velocity) and 2) reaching a specified goal area, also with zero velocity. A four-dimensional feature function captures the vehicle position relative to the stop line and its dynamic limitations near the stop line. Twenty demonstrations, representing over 10 min of driving behavior, are collected in this scenario.

The obstacle-avoidance scenario, shown in Fig. 5a, also involves two tasks: 1) Navigating around obstacles on the path and 2) successfully reaching the goal area. A ten-dimensional feature function is used, incorporating two categories of factors: geometric and vehicle dynamics. The geometric factors account for the positions of lanes, obstacles, and road widths, whereas the vehicle dynamics factors encompass velocity constraints, angular velocity, and positioning relative to obstacles. Note that the number and locations of obstacles vary across the experiments presented in Sections V-E, V-F, and V-G. Fig. 6 illustrates the various configurations of obstacles. These different configurations are specifically designed for the dynamic contextual experiments detailed in Section V-G, aimed at evaluating the robustness of the proposed method in unknown scenes. Approximately 30 demonstrations are collected for each scene; consequently, the dataset for the obstacle-avoidance scenario consists of approximately 1 h of demonstrated driving behaviors.

B. STATE-SPACE AND MOTION DYNAMICS

The experiments describes herein model driving behaviors using a five-dimensional state-space, $\boldsymbol{s}_t = (x_t, y_t, \theta_t, v_t, \omega_t)^\top \in \mathbb{R}^5$, where x_t, y_t, θ_t, v_t, and ω_t denote the x-position, y-position, angle, velocity, and angular velocity, respectively. Additionally, a two-dimensional action–space is used, represented as $\boldsymbol{a}_t = (a_t, \alpha_t)^\top$, where a_t and α_t denote acceleration and angular acceleration, respectively. Motion transitions adhere to nonholonomic vehicle dynamics.

C. COMPARISON METHODS

To evaluate the performance of the proposed approach, we compared its performance with that of the following methods:

1) IRL: MaxEnt IRL with RRT as an importance sampling-based motion planner [18].

2) GMM-IRL: MaxEnt IRL with GMM-RRT [23] as a guided motion planner. To effectively manage high-dimensional spaces and automatically determine the number of clusters, this method was implemented as DPGMM-RRT. The threshold for balancing exploration and exploitation in Algorithm 1 was set as $\Theta = 0.5$ following the work of Wang et al. [23], who reported that this value optimally balances guided sampling within target regions while ensuring sufficient exploration of the entire state-space.
3) guideIRL (ours): Proposed method using two types of guided motion planners: GMM-RRT (as shown in Algorithm 1, but without Lines 3–5) and demo-RRT (as shown in Algorithm 1, but without Lines 9–10), as described in Algorithm 3. The discount factor for the reward was set as $\gamma = 0.95$, emphasizing targeted exploitation while still allowing for slight relaxation to avoid overfitting to the demonstrations.
4) demoIRL (ours-ablation): Proposed method using only demo-RRT as a weakly guided motion planner. This model was used for an ablation study to demonstrate the role of GMM guidance.

D. EVALUATION METRICS

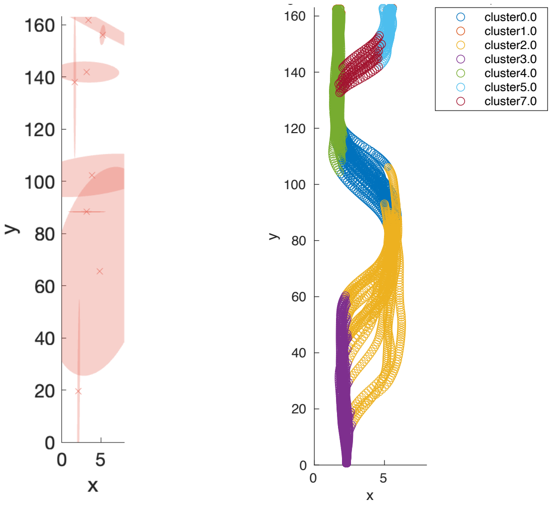
The following metrics were used to assess the quality of the generated paths and the effectiveness of IRL learning outcomes:

1) Path quality metric: This metric is calculated as
$$\log \sum_{\tau \in D_{samp}} \exp\left(r_w(\tau)\right) - \sum_{\tilde{\tau} \in D_{demo}} r_w(\tilde{\tau}). \qquad (14)$$
It measures the difference between the log-sum-exp of rewards across all sampled paths and the total reward of the demonstrated paths.
2) Loss function: The loss is defined as the negative log-likelihood, as defined in (3).
3) Demonstration similarity: This metric is computed using the modified Hausdorff distance (MHD) [31] between the generated and demonstrated paths. Metrics mhd50 and mhd90 represent the 50th and 90th percentile values, respectively, providing insights into the distribution of path distances.
4) Success rate: This metric measures whether the generated path meets the task requirements. For instance, in the stop-line scenario, a path that stops at the stop line but does not reach the goal area would have a success rate of 50%.

E. DPGMM PREPARATION

This section details the update process for the DPGMM parameters and presents the resulting preprocessed data. For intuitive understanding, we visualize the learned DPGMM using the obstacle-avoidance scenario as an example.

The DPGMM parameters were updated to fit the human demonstrations. The mean and precision of each component were determined by four parameters that were iteratively refined during the DPGMM learning process, as defined in (12). We used the maximum likelihood EM algorithm to update these parameters. In the E-step, the lower bound was evaluated using (10.70)–(10.77) from [28]. In the M-step, parameters were updated via (10.60)–(10.63) from [28] to maximize the lower bound of the likelihood. Appendix A describes each parameter and summarizes the parameter tuning results. Four key parameters are required for tuning, and the DPGMM is robust to parameter variations as its performance does not vary significantly across different parameter settings.



(a) Components.

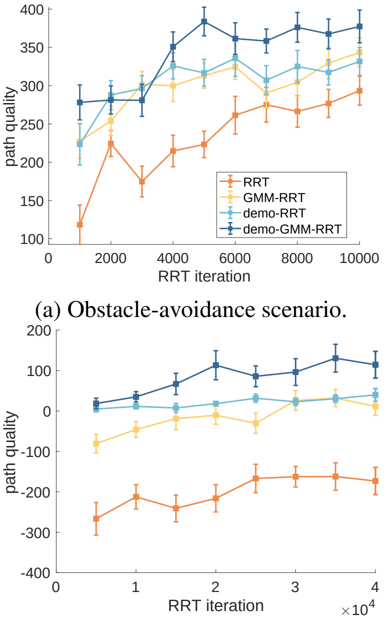
(b) Clustering on states.

FIGURE 7: Visualization of the learned DPGMM on the x-y plane. (a) Gaussian mixture components fitted to the obstacle-avoidance scenario data. (b) State clustering based on the learned DPGMM.

The learned parameters play a crucial role in guided distribution, promoting better exploration during path sampling. For example, Fig. 7a illustrates the learned Gaussian mixture components obtained by fitting the human demonstrations from the obstacle-avoidance scenario. Subsequently, guided planners use these components to sample new states during the path-sampling process, unlike traditional planners that sample new states randomly across the entire state-space. Fig. 7b shows clustering results derived from the training data based on the learned parameters. The DPGMM captures the motion trends of going straight and turning left or right to avoid obstacles. This ensures that newly sampled states maintain the current motion trend, resulting in smoother trajectories without abrupt changes.

F. EXPERIMENTS IN STATIC ENVIRONMENTS

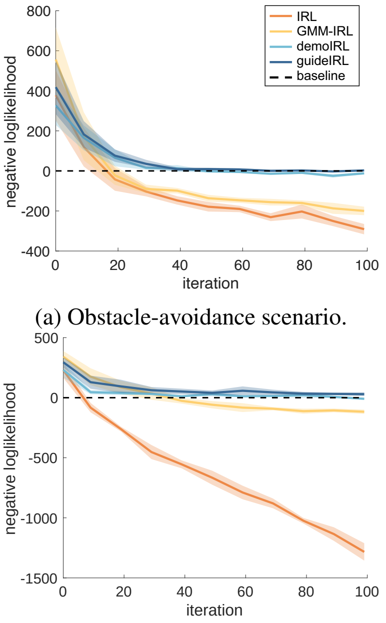
Three quantitative evaluations were performed to demonstrate the performance of various IRL methods that leverage guided motion planners. Additionally, a qualitative evaluation was conducted to clarify the influence of guided planners on the learning outcomes of IRL. For simplicity, these four evaluations were conducted in static environments assuming



(a) Obstacle-avoidance scenario.

(b) Stop-line scenario.

FIGURE 8: Comparison of sampled path qualities using the metric defined in (14). The number of sampled paths, $|D_{samp}|$, is 300 for both scenarios.



(a) Obstacle-avoidance scenario.

(b) Stop-line scenario.

FIGURE 9: Loss function values over iterations during IRL model training following Algorithm 3. The dashed black line at zero serves as a *baseline*; as discussed in Section III-D, values below zero indicate inaccurate partition function approximation.

time-invariant environmental factors $\boldsymbol{c}_t = \boldsymbol{c}$, as indicated in (9). Furthermore, these factors remained consistent across both model training and evaluation ($\boldsymbol{c}_{\text{train}} = \boldsymbol{c}_{\text{eval}}$).

1) Quantitative evaluation 1: path sampling qualities

This experiment evaluated the performance of various motion planners based on the quality of their sampled paths using a manually designed reward function. Fig. 8 presents the sampled path qualities over multiple iterations. The proposed method (demo-GMM-RRT) outperformed the other methods, achieving higher path qualities across nearly all iterations in both scenarios.

Comparing the two scenarios, the sampled path qualities in the obstacle-avoidance scenario (Fig. 8a) are generally higher than those in the stop-line scenario (Fig. 8b). Furthermore, the stop-line scenario requires a significantly larger number of RRT iterations to achieve high-quality path planning. This difference arises because the stop-line scenario demands more precise driving behaviors to execute the stopping action precisely at the stop line. In contrast, the driving behaviors required for obstacle avoidance are more flexible, as a human driver can initiate turning maneuvers at any location sufficiently far from obstacles.

Regarding the performance of individual planners, the traditional RRT motion planner performed poorly in the stop-line scenario, with path quality remaining below zero in every iteration. This suggests that the paths generated by traditional RRT likely correspond to the scenario depicted in Fig. 2a. In the ablation study, where the manually designed reward functions may be considered optimal, GMM-RRT performs well in the obstacle-avoidance scenario but yields negative values in the stop-line scenario. This finding highlights the importance of targeted exploitation in more complex environments. Without it, the sampled paths deviate significantly from the demonstrated paths, as shown in Fig. 2b. Conversely, demo-RRT, which solely employs targeted exploitation without broad exploration, struggles to efficiently sample paths and tends to produce results consistent with the scenario shown in Fig. 2c. Finally, the proposed method achieves the best performance owing to its ability to balance exploitation and exploration, ensuring that the sampled paths align with the scenario shown in Fig. 2d.

2) Quantitative evaluation 2: IRL model training

This experiment examines the loss function over iterations during the IRL model training process. In IRL training, the loss function should decrease monotonically over iterations but must not drop below zero. As discussed in Section III-D, a non-negative loss is a fundamental requirement. A negative loss indicates inaccurate partition function approximation, preventing proper gradient updates. Negative loss values also implicitly indicate poor signal quality in the sampled paths.

Overall, Fig. 9 shows the loss function over iterations, with a baseline defined at zero loss to highlight the issue of negative loss values. In both scenarios, only the proposed method (guideIRL) maintains positive loss values throughout training, with the other methods yielding negative losses at various points. In terms of individual models, the traditional

TABLE 1: Performance metrics for paths generated by the trained IRL models. Values represent the mean and standard deviation across 300 generated paths for each method.

(a) Obstacle-avoidance scenario.

	mhd50↓	mhd90↓	success rate↑
IRL	3.48 ± 0.95	12.21 ± 2.94	$51.82 \pm 7.01\%$
GMM-IRL	2.40 ± 0.49	8.55 ± 1.98	$55.97 \pm 6.15\%$
demoIRL	0.87 ± 0.74	4.88 ± 2.35	$87.29 \pm 6.63\%$
guideIRL	$\mathbf{0.49 \pm 0.36}$	$\mathbf{3.73 \pm 1.80}$	$\mathbf{88.08 \pm 6.77}\%$

(b) Stop-line scenario.

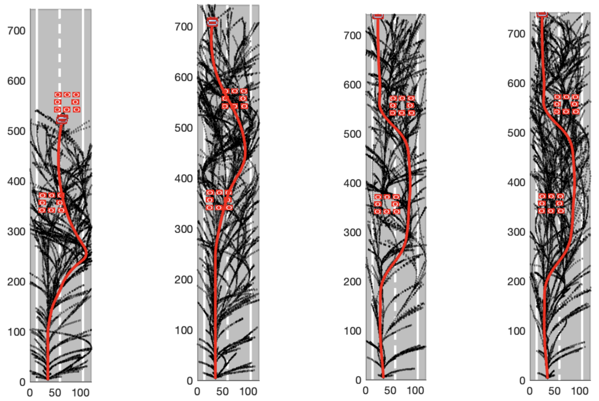
	mhd50↓	mhd90↓	success rate↑
IRL	2.93 ± 1.45	7.07 ± 3.38	$43.14 \pm 4.84\%$
GMM-IRL	5.53 ± 2.95	13.71 ± 6.57	$51.29 \pm 6.29\%$
demoIRL	2.36 ± 1.99	8.93 ± 5.35	$78.79 \pm 8.74\%$
guideIRL	$\mathbf{0.63 \pm 1.11}$	$\mathbf{2.47 \pm 2.84}$	$\mathbf{92.88 \pm 5.53}\%$

IRL method (red lines) exhibits significant instability with diverging loss. This instability is attributable to the path deviation issue, as depicted in Fig. 2a. Although GMM-IRL (yellow lines) outperforms the traditional IRL method, the loss still diverges. This result supports the discussion in Section III-D, indicating that even with a guided motion planner, the sampled paths may not sufficiently cover the demonstrated paths, as shown in Fig. 2b. In the ablation study, although demoIRL demonstrates significantly better performance than GMM-IRL, it still results in negative loss in both scenarios. This finding suggests that the sampled paths in Fig. 2c negatively impact the IRL model. Overall, by balancing broad exploration and targeted exploitation around the demonstrations, the proposed method achieves the best performance and ensures stable IRL learning.

3) Quantitative evaluation 3: IRL evaluation

This experiment evaluates the quality of paths generated by various IRL methods after training. Table 1 presents the results for both the obstacle-avoidance and stop-line scenarios. The proposed method, which balances broad exploration and targeted exploitation in guided motion planning, outperforms other approaches in terms of both path similarity (measured by MHD) and task success rate. The ablation study reveals an interesting finding: demoIRL, which focuses on sampling around the demonstrated regions, outperforms GMM-IRL, which uses DPGMM distributions to broadly explore the entire state-space. This result implicitly suggests that efficient exploitation may be more critical than broad exploration, particularly for complex driving tasks.

In the stop-line scenario, the performance of GMM-IRL is inferior to that of traditional IRL, as this scenario requires precise vehicle control, which broad exploration alone cannot achieve effectively. Additionally, the mhd90 metric for demoIRL is inferior to that of traditional IRL because the demoIRL method relies heavily on the demonstrations but lacks the capability for broader state-space exploration. Nevertheless, integrating the strengths of GMM-IRL and demoIRL into guideIRL results in significant performance improvements. These results underscore the importance of balancing exploitation and exploration during path sampling for effective IRL learning. These results also indicate that the negative effects of the three suboptimal relationships shown in Figures 2a, 2b, and 2c vary depending on the driving environment. Only the relationship depicted in Fig. 2d leads to stable IRL model performance across different scenarios.



(a) IRL (b) GMM-IRL (c) demoIRL (d) guideIRL

FIGURE 10: Visualization of generated trees and paths for the obstacle-avoidance scenario. Red rectangles indicate obstacles. Black dots represent tree nodes generated after a fixed number of iterations. The red line shows a generated trajectory connecting the highest-rewarded nodes, subject to nonholonomic vehicle dynamics.

4) Qualitative evaluation: behavior prediction visualization

This experiment provides a visual comparison of the tree nodes and predicted paths generated by various IRL methods. For clarity and ease of understanding, we present the visualization results from the obstacle-avoidance scenario as a representative example. Although the state-space is five-dimensional, we project the results onto the x–y plane for visual clarity. Furthermore, all methods are executed with the same number of iterations (4000). This number is chosen to ensure a trade-off between performance and visual clarity, as shown in Fig. 8a.

Fig. 10 visualizes the predicted behaviors of each method, showing both the generated tree nodes and a representative connected path. The traditional IRL method explores the state-space randomly, resulting in significant inefficiency. Consequently, the nodes are concentrated in intermediate regions, and the generated path fails to reach the goal region. While GMM-IRL demonstrates significantly improved efficiency in exploring the state-space compared with the traditional method, it lacks the precision required to avoid obstacles, leading to collisions. In contrast, demoIRL, which prioritizes sampling in target regions, generates considerably fewer nodes, indicating a more constrained exploration and exploitation strategy. Finally, by combining the broad exploration capability of GMM-IRL with the targeted precision

TABLE 2: Performance metrics for paths generated by the trained IRL model under varying DPGMM representations and training/evaluation scene combinations. Values represent the mean and standard deviation across 300 generated paths. The best and worst performance metrics are boldfaced and presented in parentheses, respectively.

DPGMM type	train/eval scene	mhd50↓	mhd90↓	success rate↑
local	same	$\mathbf{3.46 \pm 1.53}$	$\mathbf{13.86 \pm 3.82}$	$\mathbf{82.73 \pm 8.25\%}$
global large	same	3.63 ± 1.64	15.12 ± 4.09	$80.75 \pm 8.24\%$
	different	4.12 ± 1.86	15.21 ± 4.22	$79.03 \pm 8.69\%$
global small	same	4.03 ± 1.93	15.54 ± 4.16	$79.42 \pm 8.59\%$
	different	4.40 ± 1.82	(16.13 ± 3.96)	$(76.19 \pm 8.49\%)$
local	different	(4.87 ± 2.21)	16.05 ± 4.58	$77.98 \pm 8.39\%$

of demoIRL, the proposed guideIRL method achieves both efficient state-space exploration and the generation of precise, collision-free driving behaviors. This balanced approach yields superior performance in terms of both state-space coverage and path accuracy.

G. EXPERIMENTS IN CONTEXTUAL DYNAMIC ENVIRONMENTS

We conduct evaluations in contextual dynamic environments to further demonstrate the applicability of the proposed method in real-world scenarios. In these experiments, environmental factors, such as the location of obstacles, vary between the training and testing phases ($\boldsymbol{c}_{\text{train}} \neq \boldsymbol{c}_{\text{eval}}$) but remain time-invariant ($\boldsymbol{c}_t = \boldsymbol{c}$, as defined in (9)). This experimental setting simulates a real-world use case where a trained model is applied to an unknown scenario while the overall road geometry remains unchanged. We design three distinct patterns of contextual factors in the obstacle-avoidance scenario, as shown in Fig. 6.

To evaluate the performance of the proposed guideIRL under different levels of prior knowledge, two categories of DPGMM were designed based on the type of data used: global and local DPGMM. In the evaluation using global DPGMM, the DPGMM was trained using a mixture of demonstrations from all available patterns, whereas the IRL model itself was trained only on demonstrations from a single pattern. Furthermore, two types of global DPGMM were considered, differing in the amount of data used from each scene to build the DPGMM model. The *Global large* configuration uses 20 driving paths from each scene, resulting in 60 driving paths. In contrast, the *Global small* configuration adopts twenty, five, and three driving paths from scenes 1, 2, and 3 (Fig. 6), respectively, for a total of 28 driving paths. In the evaluation using *local* DPGMM, both the DPGMM and IRL models were trained using demonstrations from a single, consistent pattern. Twenty driving paths were used to build the local DPGMM for each scene. Notably, these local DPGMMs were also used for evaluating the static environments, as detailed in Section V-F, where the training and evaluation data originate from the same scene.

To evaluate performance under both known and unknown scene conditions, the training and evaluation data were drawn from either the same or different scenes. This experimental setup simulates the real-world situation of deploying a trained model to various unknown scenarios. For example, the guideIRL model could be trained on data from one scene and then tested on data from a completely different scene, allowing us to investigate the extent to which performance would degrade when the model is applied to unknown scenes.

Table 2 summarizes the performance of guideIRL using the three types of DPGMM models and two different configurations of training and evaluation scenes. As expected, local DPGMM with data from the same scene exhibits the best performance, owing to its access to rich scene-specific prior knowledge. Conversely, local DPGMM with data from different scenes shows significant performance degradation, reflecting its lack of prior knowledge relevant to unknown scenes. Although global DPGMM lacks the specificity of local DPGMM for a particular scene, it demonstrates improved performance even when applied to unknown scenes, owing to its broader prior knowledge derived from various scenes. Interestingly, global small DPGMM does not outperform local DPGMM when applied to different scenes. This is because the number of driving data points in certain scenes is insufficient, leading to limited prior knowledge of those specific contexts. This result implicitly highlights a limitation inherent to IRL-based methods: while IRL is a data-driven approach, and although the proposed method with general prior knowledge outperforms those with limited prior knowledge, the quality and quantity of demonstrated data remain essential for achieving safe and stable driving-behavior predictions.

Additionally, the benefits of using global DPGMM over local DPGMM when applying the model to different scenes are significant. For example, the Global large DPGMM configuration on different scenes exhibits an mhd50 metric of 4.12, lower than that of the local configuration (4.87). In practice, considering a path defined solely by (x,y) coordinates, the improvement of 0.75 in the mhd50 metric indicates that the generated path is, on average, 0.75 m closer to the human-demonstrated path at the 50th percentile. While perfect replication of human-demonstrated paths is not necessarily required in real-world applications, such improvements can significantly enhance the safety of generated paths.

In conclusion, the proposed guideIRL method demonstrates robustness to unknown scenes in dynamic contextual environments when using the DPGMM trained with sufficiently general prior knowledge. These results support the applicability of the proposed method to certain real-world scenarios.

H. EXTENSION TO DYNAMIC ENVIRONMENTS

We acknowledge that although the proposed guideIRL method is effective, it is currently limited to time-invariant environmental factors and lacks a mechanism to directly handle time-variant conditions. Although it demonstrates both efficiency and stability in static (Section V-F) and contextual dynamic driving scenarios (Section V-G), real-world environments are inherently dynamic, characterized by constantly changing elements such as obstacles, pedestrians, and other vehicles. This dynamic nature hinders the direct application of the proposed approach to fully dynamic scenarios. Despite the gap between our controlled experimental settings and real-world conditions, the guided IRL method provides a solid foundation for modeling the decision-making process in a stable, IRL-based manner.

Bridging this gap and extending our approach to fully dynamic environments requires enhancements to the guided distribution representation. Equation (9) presents a theoretical formulation of guided distributions, where the sampled action–state pair, $(\boldsymbol{a}_t, \boldsymbol{s}_t)$, should depend on the current environmental factor, $\boldsymbol{c}_t$. This study uses DPGMM as a simple yet interpretable representation of guided distributions to steer exploration. DPGMM is chosen for its simplicity and explainability in modeling driving behaviors [23]. However, this simplicity limits its ability to represent the complexities of more dynamic environments. For instance, adapting to time-variant environmental factors would necessitate an additional mechanism for retraining the DPGMM, potentially destabilizing the IRL model. Moreover, handling time-variant environmental factors requires another mechanism to represent the uncertainty in DPGMM predictions at future time-steps to guarantee driving safety. Additionally, it is important to note that the representation of guided distributions is not limited to Gaussian-based methods. A potential solution involves utilizing more expressive models, such as neural networks, to represent the complex relationships between action–state pairs and dynamic environmental factors.

VI. CONCLUSION

Recently, IRL has been widely recognized for its effectiveness in modeling decision-making processes, particularly owing to its ability to capture complex driving behaviors by learning directly from human demonstrations. However, traditional MaxEnt IRL methods in continuous state-spaces face significant challenges in accurately approximating the partition function, often leading to unstable driving-behavior models. This study addresses this critical issue by introducing an IRL-aware guided motion planner that strategically balances broad exploration with targeted exploitation. This balanced approach improves partition function estimation through the sampling of high-quality paths. Experimental results obtained using a driving simulator demonstrate that the proposed method outperforms existing IRL methods in terms of stability and accuracy in both static and contextual dynamic environments. It not only enhances the stability of IRL methods for predicting driving behaviors but also demonstrates the potential for applying IRL in the field of autonomous driving to achieve human-like decision-making. Furthermore, it offers an explainable solution for analyzing the root causes of suboptimal predictions, which could contribute to traffic accidents.

Future work can be focused on addressing variations in demonstration quality, including the incorporation of failed and dangerous demonstrations by explicitly considering the quality of individual demonstrations during IRL learning. Handling data with mixed qualities is essential for real-world driving tasks, as collecting only expert demonstrations is not always feasible. Addressing this challenge will further enhance the applicability of guideIRL, paving the way for more advanced autonomous driving capabilities.

ACKNOWLEDGMENT

We would like to thank Editage (www.editage.jp) for English language editing.

APPENDIX A: DPGMM PARAMETER CHOICES

PARAMETER DETAILS

DPGMM requires the tuning of four key parameters during training. Three of these are related to the Gaussian–Wishart distribution, as defined in (10.60)–(10.63) in [28]. The fourth parameter, the truncation number, determines the maximum number of clusters. Table 3 summarizes these parameters.

TABLE 3: Key parameters of DPGMM and their descriptions.

Parameter	meaning
T	Truncation number
β	Factor of covariance matrix in Gaussian distribution
$\boldsymbol{W}$	Covariance matrix in Wishart distribution
$\boldsymbol{\nu}$	Degrees of freedom in Wishart distribution

PARAMETER TUNING

Table 4 summarizes the parameter tuning results for the loss function defined in (3). Although a considerable number of parameter combinations are examined during the experiments, only a few representative combinations are listed in the table for clarity. The DPGMM is robust to these parameters as the loss function does not vary significantly across different parameter settings.

Additionally, parameter tuning is conducted separately for each scene described in Section V-A because optimal parameter combinations depend on the driving scenario. The experimental results presented in Sections V-F and V-G are obtained using the best parameter combinations that yielded the optimal performance on the demonstration data.

TABLE 4: DPGMM parameter tuning results for four parameters of DPGMM based on the loss function defined in (3). Note that $\boldsymbol{I}$ represents the identity matrix. The optimal parameter combination yielding the minimum loss is boldfaced.

T	β	$\boldsymbol{W}$	$\boldsymbol{\nu}$	loss↓
10	0.5	$0.1\boldsymbol{I}$	20	205.73
10	0.5	$0.5\boldsymbol{I}$	20	253.77
10	0.8	$0.1\boldsymbol{I}$	20	231.03
10	0.8	$0.5\boldsymbol{I}$	20	249.04
20	0.5	$0.1\boldsymbol{I}$	25	207.59
20	**0.5**	$\mathbf{0.5}\boldsymbol{I}$	**25**	**191.34**
20	0.8	$0.1\boldsymbol{I}$	25	210.94
20	0.8	$0.5\boldsymbol{I}$	25	227.54

REFERENCES

[1] Natnael M. Negash and James Yang. Driver behavior modeling toward autonomous vehicles: Comprehensive review. *IEEE Access*, 11:22788–22821, 2023.

[2] Xinghua Hu and Mintanyu Zheng. Research progress and prospects of vehicle driving behavior prediction. *World Electric Vehicle Journal*, 12(2), 2021.

[3] Daniel Coelho and Miguel Oliveira. A review of end-to-end autonomous driving in urban environments. *IEEE Access*, 10, 2022.

[4] Saurabh Arora and Prashant Doshi. A survey of inverse reinforcement learning: Challenges, methods and progress. *Artificial Intelligence*, 297:103500, 2021.

[5] Stephen Adams, Tyler Cody, and Peter A Beling. A survey of inverse reinforcement learning. *Artificial Intelligence Review*, 2022.

[6] Andrew Y Ng and Stuart Russell. Algorithms for inverse reinforcement learning. In *Proc. of ICML*, 2000.

[7] Deepak Ramachandran and Eyal Amir. Bayesian inverse reinforcement learning. In *IJCAI*, 2007.

[8] Brian D. Ziebart, Andrew Maas, J. Andrew Bagnell, and Anind K. Dey. Maximum entropy inverse reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, volume 8, pages 1433–1438, 2008.

[9] Brian D Ziebart. *Modeling purposeful adaptive behavior with the principle of maximum causal entropy*. Carnegie Mellon University, 2010.

[10] Navid Aghasadeghi and Timothy Bretl. Maximum entropy inverse reinforcement learning in continuous state spaces with path integrals. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1561–1566, 2011.

[11] Masamichi Shimosaka, Junichi Sato, Kazuhito Takenaka, and Kentarou Hitomi. Fast inverse reinforcement learning with interval consistent graph for driving behavior prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, volume 31, 2017.

[12] Zheng Wu, Liting Sun, Wei Zhan, Chenyu Yang, and Masayoshi Tomizuka. Efficient sampling-based maximum entropy inverse reinforcement learning with application to autonomous driving. *IEEE Robotics and Automation Letters*, 5(4):5355–5362, 2020.

[13] Long Xin, Shengbo Eben Li, Pin Wang, Wenhan Cao, Bingbing Nie, Ching-Yao Chan, and Bo Cheng. Accelerated inverse reinforcement learning with randomly pre-sampled policies for autonomous driving reward design. In *Proceedings of the IEEE Intelligent Transportation Systems Conference (ITSC)*, pages 2757–2764, 2019.

[14] Sergey Levine and Vladlen Koltun. Continuous inverse optimal control with locally optimal examples. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 475–482, 2012.

[15] Chelsea Finn, Sergey Levine, and Pieter Abbeel. Guided cost learning: Deep inverse optimal control via policy optimization. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 49–58, 2016.

[16] LAVALLE S. M. Rapidly-exploring random trees : A new tool for path planning. *Computer Science Dept. Oct.*, 98(11), 1998.

[17] Kyriacos Shiarlis, Joao Messias, and Shimon Whiteson. Rapidly exploring learning trees. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 1541–1548, 2017.

[18] Shinpei Hosoma, Masato Sugasaki, Hiroaki Arie, and Masamichi Shimosaka. RRT-based maximum entropy inverse reinforcement learning for robust and efficient driving behavior prediction. In *Proceedings of the IEEE Intelligent Vehicles Symposium (IV)*, pages 1353–1359, 2022.

[19] Matt Zucker, Nathan Ratliff, Anca D Dragan, Mihail Pivtoraiko, Matthew Klingensmith, Christopher M Dellin, J Andrew Bagnell, and Siddhartha S Srinivasa. CHOMP: Covariant hamiltonian optimization for motion planning. *The International Journal of Robotics Research*, 32(9-10):1164–1193, 2013.

[20] Gu Ye and Ron Alterovitz. *Demonstration-Guided Motion Planning*, pages 291–307. Springer International Publishing, 2017.

[21] Takayuki Osa, Amir M Ghalamzan Esfahani, Rustam Stolkin, Rudolf Lioutikov, Jan Peters, and Gerhard Neumann. Guiding trajectory optimization by demonstrated distributions. *IEEE Robotics and Automation Letters*, 2(2):819–826, 2017.

[22] RB Ashith Shyam, Peter Lightbody, Gautham Das, Pengcheng Liu, Sebastian Gomez-Gonzalez, and Gerhard Neumann. Improving local trajectory optimisation using probabilistic movement primitives. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2666–2671, 2019.

[23] Jiankun Wang, Tingguang Li, Baopu Li, and Max Q-H Meng. GMR-RRT*: Sampling-based path planning using Gaussian mixture regression. *IEEE Transactions on Intelligent Vehicles*, 7(3):690–700, 2022.

[24] Richard Cheng, Krishna Shankar, and Joel W Burdick. Learning an optimal sampling distribution for efficient motion planning. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7485–7492, 2020.

[25] Sicelukwanda Zwane, Denis Hadjivelichkov, Yicheng Luo, Yasemin Bekiroglu, Dimitrios Kanoulas, and Marc Peter Deisenroth. Safe trajectory sampling in model-based reinforcement learning. In *Proceedings of IEEE*

International Conference on Automation Science and Engineering (CASE), pages 1–6, 2023.

[26] Martin L. Puterman. Markov decision processes: Discrete stochastic dynamic programming. 2014.

[27] Dilan Görür and Carl Edward Rasmussen. Dirichlet process Gaussian mixture models: Choice of the base distribution. *Journal of Computer Science and Technology*, 25(4):653–664, 2010.

[28] Christopher M Bishop and Nasser M Nasrabadi. *Pattern recognition and machine learning*, volume 4. Springer, 2006.

[29] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. CARLA: An open urban driving simulator. In *Proceedings of the Conference on Robot Learning (CoRL)*, pages 1–16, 2017.

[30] Minglu Zhao and Masamichi Shimosaka. Inverse reinforcement learning with failed demonstrations towards stable driving behavior modeling. In *Proceedings of the IEEE Intelligent Vehicles Symposium (IV)*, pages 2537–2544, 2024.

[31] M-P Dubuisson and Anil K Jain. A modified hausdorff distance for object matching. In *Proceedings of the IEEE International Conference on Pattern Recognition (ICPR)*, volume 1, pages 566–568, 1994.

MINGLU ZHAO received her B.E. and M.E. degrees from Tohoku University in 2020 and 2022, respectively. She is currently pursuing a Ph.D. degree at the Institute of Science Tokyo. Her research interests include inverse reinforcement learning and its applications to driving behavior prediction.

MASAMICHI SHIMOSAKA received his B.E., M.E., and Ph.D. degrees from the University of Tokyo in 2001, 2003, and 2006, respectively. He joined the Institute of Science Tokyo (formerly Tokyo Institute of Technology) as an associate professor in July 2015. Before joining the Institute of Science Tokyo, he was a faculty member at the University of Tokyo from 2006 to 2015. During his doctoral studies, he was financially supported by the Japan Society for the Promotion of Science as a research fellow. His research interests include machine intelligence and ubiquitous computing. He is also a member of the ACM and IEEE.