Continuous Inverse Reinforcement Learning with State-wise Safety Constraints for Stable Driving Behavior Prediction

Minglu Zhao and Masamichi Shimosaka[1]

***Abstract*— Inverse reinforcement learning (IRL) is a promising approach for modeling human driving behaviors by learning underlying reward functions from expert demonstrations. While recent studies have incorporated failed demonstrations to improve learning robustness, most existing methods enforce safety constraints only at the trajectory level, which is insufficient for real-world autonomous driving scenarios requiring per-state safety. This paper proposes a novel IRL framework that introduces state-wise safety constraints via a behavior discriminator, which generates safety labels for each state based on environmental context. By integrating the discriminator into the main reward optimization loop, the proposed method avoids additional computational complexity while ensuring safety at every decision point. Experimental results in the CARLA simulator across multiple driving scenarios demonstrate improved performance in both behavior imitation and driving task requirements. The results confirm that enforcing state-wise safety significantly enhances stability and reliability in driving behavior prediction in static contextual environments, providing a viable direction for safer autonomous decision-making.**

I. Introduction

The rapid advancement of autonomous driving technologies has increased the demand for accurate and reliable driving behavior prediction, a key component for ensuring safety and efficiency in real-world traffic scenarios [1]. Learning-based methods for driving behavior prediction have emerged as scalable and efficient solutions. In particular, imitation learning approaches have gained considerable attention and shown promising results in both simulation environments and real-world applications [2], [3].

Among these, inverse reinforcement learning (IRL) has emerged as a prominent technique for modeling human decision-making by learning directly from human demonstrated behaviors [4]. It infers an underlying reward function that serves as a quantitative representation of preferred behaviors, using a data-driven approach. Notably, maximum entropy IRL (MaxEnt IRL) [5] provides a probabilistic framework that captures both optimal and suboptimal behaviors, making it well-suited for modeling the inherent variability in human driving.

However, learning solely from expert demonstrations can lead to limited generalization, especially in safety-critical edge cases [6]. As the proverb "failure teaches success" suggests, incorporating failed demonstrations can help IRL agents learn to intentionally avoid unsafe behaviors [7]. To address this, recent approaches have incorporated both expert and failed demonstrations, leveraging positive feedback from expert demonstration and negative feedback from failed demonstrations [6]–[9]. This dual-feedback strategy enhances the agent's ability to distinguish between desirable and undesirable outcomes, thereby improving the reliability of behavior prediction in safety-critical environments.

However, existing IRL methods that utilize failed demonstrations, commonly referred to as IRLFD, face a major limitation: enforcing safety only at the trajectory level is insufficient for real-world, safety-critical applications [10]. In autonomous driving, for example, many safety requirements are instantaneous and state-dependent. A common case is collision avoidance, where even a single unsafe state near an obstacle can result in failure, despite the rest of the trajectory being safe.

To address the above limitation, reinforcement learning (RL) research has increasingly focused on state-wise safety constraints, which enforce safety at each individual time step [11]; however, directly integrating such constraints into IRL frameworks introduces significant computational complexity. Many existing RL approaches, based on constrained Markov decision processes [12], [13], model safety via soft penalties in the cost function, which often fail to guarantee strict constraint satisfaction. More recent methods using latent barrier functions [14] can represent safety boundaries more effectively, but incorporating them into IRLFD leads to a tri-level optimization process, updating the reward, solving constrained policies, and tuning the barrier function, making the approach computationally expensive and sensitive to hyper-parameters.

An alternative solution has been proposed by SIRLFD [15], which introduces time-series safety labels to represent state-wise constraints without adding an additional optimization loop, but this method becomes computationally impractical in continuous, high-dimensional state spaces due to the cost of labeling each state. Specifically, for a discrete state space, the complexity scales as $\mathcal{O}(nk)$, where n is the resolution per dimension and k is the number of state dimensions. In real-world driving scenarios, the state-space is continuous and high-dimensional, making exhaustive state labeling infeasible without prior knowledge of unsafe regions [14].

To overcome these challenges, this work introduces a novel IRLFD approach that enforces state-wise safety constraints through a safe behavior discriminator, which is jointly optimized within the traditional IRL reward update loop. Inspired by latent barrier modeling [14], we propose a latent label model that generates safety labels based on

[1]The authors are with the Department of Computer Science, Institute of Science Tokyo, Tokyo, Japan. {zhao, simosaka}@miubiq.cs.titech.ac.jp

environmental contexts, allowing the system to learn from limited information while maintaining safety across the continuous state-space. An overview of the proposed framework is illustrated in Fig. 1.

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

- We propose a novel IRLFD framework that introduces state-wise safety constraints via a behavior discriminator, enabling safety enforcement at each state without adding an extra optimization loop.
- Our method targets driving behavior prediction in continuous, high-dimensional state-spaces, improving both stability and safety compared to existing methods.
- We validate the proposed approach in diverse driving scenarios using the CARLA simulator, demonstrating robustness and effectiveness across various driving tasks.

II. Related work

A. IRL from failed demonstrations

Incorporating failed demonstrations is essential for training IRL agents capable of intentionally avoiding unsafe behaviors [6]–[9]; many existing IRLFD methods lack robustness and stability in safety-critical applications.

A common strategy among these methods is to optimize behavior by maximizing similarity to expert demonstrations while minimizing similarity to failed ones, typically at the trajectory level. IRLF [7] proposes a trajectory-level similarity metric based on visitation frequencies and optimizes this measure using a dual-gradient strategy during training. One gradient captures positive feedback from expert demonstrations, while the other incorporates negative feedback from failed demonstrations. BIRLF [9] replaces the visitation frequency metric with a halfspace-induced potential measure but retains the trajectory-wise formulation, limiting its ability to address fine-grained safety concerns at the state level.

State-wise methods have emerged in recent years; however, the spatial coverage of demonstrated state sets is often limited, restricting their ability to generalize across diverse or unseen states. MixGAIL [6] introduces a reward function that evaluates the distance between generated and demonstrated states. Similarly, SIRLFD [15] enforces state-wise safety constraints by integrating time-series safety labels into the IRL optimization framework. However, both methods rely heavily on the coverage and quality of the demonstration data. In real-world scenarios where state distribution is sparse or incomplete, these approaches struggle to generalize to unseen or rare states.

B. State-wise safe reinforcement learning

In the field of reinforcement learning, state-wise safe reinforcement learning is an advanced paradigm to enforce safety constraints [10], [11], [14]; however, directly applying these ideas to IRLFD frameworks is nontrivial due to increased computational complexity and integration challenges.

A recent survey [11] categorizes state-wise safety techniques into two groups: those that ensure safety throughout training and those that enforce it only after convergence. For example, state-wise safe proximal policy optimization [10] exemplify the former by incorporating a latent dynamic model and barrier function to ensure safety at each state and train the model effectively. Similarly, Zhan *et al.* [14] employ a latent barrier function to encode state-wise safety constraints, enabling safer exploration and improved policy performance.

While IRL also benefits from enforcing state-wise constraints, existing IRLFD frameworks generally lack efficient mechanisms to incorporate them, especially in high-dimensional, continuous state-spaces. Furthermore, attempts to integrate barrier-based methods into IRLFD often introduce complex tri-level optimization loops, significantly increasing training time and sensitivity to tuning parameters.

III. Problem Formulation

This section introduces the foundational concepts and formal definitions relevant to our approach. Note that this work focuses on the MaxEnt IRL framework, taking the advantage to deal with both optimality and suboptimality.

A. Problem settings

IRL is typically modeled using a finite-horizon Markov decision process. Here, 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 dynamic function T, transiting from the current state $\boldsymbol{s}_t \in \mathcal{S}$ to the next state $\boldsymbol{s}_{t+1} = T(\boldsymbol{a}_t, \boldsymbol{s}_t)$. The state transiting probability is usually represented as $P : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ with $p(\boldsymbol{s}_{t+1}|\boldsymbol{s}_t, \boldsymbol{a}_t)$. A trajectory consists of a time-series of action–state pairs over a time horizon h, such 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 by an immediate reward function $r_{\boldsymbol{w}}(\boldsymbol{a}_t, \boldsymbol{s}_t)$, parameterized by parameter $\boldsymbol{w}$. The IRL model aims to obtain a policy $\pi : \mathcal{S} \times \mathcal{A} \to [0, 1]$ with $\pi(\boldsymbol{s}_t, \boldsymbol{a}_t) = p(\boldsymbol{a}_t|\boldsymbol{s}_t)$ that maximizes the expected future reward.

B. Learning from expert and failed demonstrations

IRLFD method extends the traditional MaxEnt IRL framework by introducing an additional soft constraint to capture negative feedback from failed demonstrations, alongside the standard constraint that promotes imitation of expert behavior. Together, these two constraints guide the learning process to both emulate expert trajectories and explicitly avoid behaviors seen in failed demonstrations.

To enable learning from both expert and failed demonstrations, IRLFD extends the policy structure and reward formulation within the traditional IRL framework. Specifically, during training, three types of policies are considered: the expert policy $\pi^{\mathcal{D}}$ derived from expert demonstrations $\mathcal{D}$, the failed policy $\pi^{\mathcal{F}}$ based on failed demonstrations $\mathcal{F}$, and the learnable policy π, which is optimized throughout training. To explicitly differentiate the learned policy from the failed one, IRLFD utilizes a latent variable $\boldsymbol{z}$ that quantifies their divergence and optimizes this difference to ensure the learned

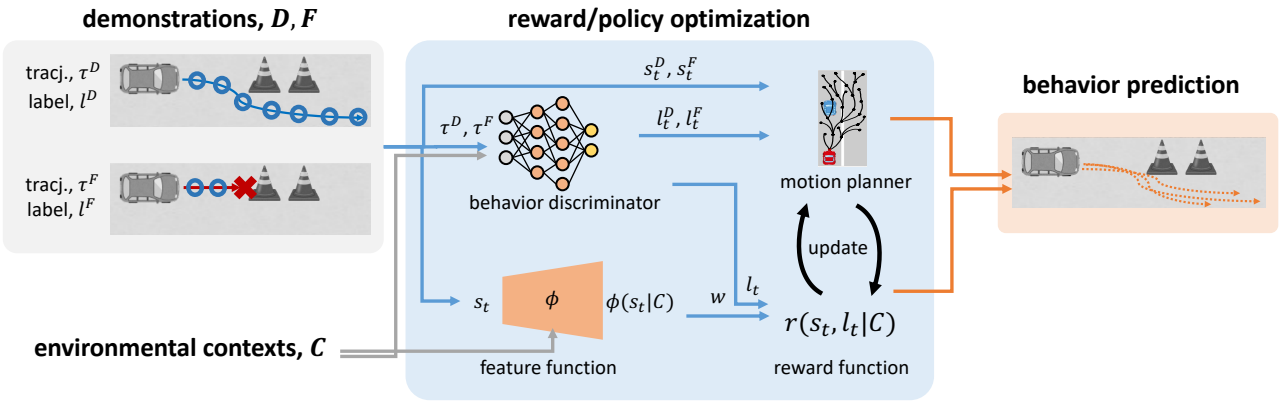


Fig. 1: Overview of the proposed method with state-wise safety constraints. A behavior discriminator learns safety labels from expert and failed demonstrations, which are combined with environmental context features to compute the reward function. The reward, policy and behavior discriminator are updated jointly in a unified optimization loop, enabling safe behavior prediction without incurring additional computational complexity.

behavior moves away from unsafe or undesirable actions. Following the traditional IRL framework, the reward function is represented using a feature mapping $\phi(\boldsymbol{s}_t)$, which encodes geometric and contextual characteristics of the environment. In the IRLFD framework, however, the reward function is modeled as a linear combination of two sets of parameters: one corresponding to positive feedback from expert demonstrations, and the other capturing negative feedback from failed demonstrations.

The trajectory-level optimization problem in IRLFD can be formulated as follows, based on the work of Shiarlis *et al.* [7]:

$$\begin{aligned} \max_{\pi,\boldsymbol{\theta},\boldsymbol{z}} \quad & \mathcal{H}(\mathcal{A}^h||\mathcal{S}^h) + \boldsymbol{\theta}\boldsymbol{z} - \frac{\lambda}{2}||\boldsymbol{\theta}||^2 \\ \text{s.t.} \quad & \mathbb{E}_{\boldsymbol{s}_t\sim p(\tau)}[\phi(\boldsymbol{s}_t)] - \mathbb{E}_{\boldsymbol{s}_t\sim\mathcal{D}}[\phi(\boldsymbol{s}_t)] = \boldsymbol{0} \\ & \mathbb{E}_{\boldsymbol{s}_t\sim p(\tau)}[\phi(\boldsymbol{s}_t)] - \mathbb{E}_{\boldsymbol{s}_t\sim\mathcal{F}}[\phi(\boldsymbol{s}_t)] = \boldsymbol{z}, \end{aligned} \tag{1}$$

where $\mathcal{A}^h$ and $\mathcal{S}^h$ denote the sequences of actions and states in a trajectory of horizon h, and $\mathcal{H}(\mathcal{A}^h||\mathcal{S}^h)$ represents the causal entropy, *i.e.*, the conditional entropy of the action sequence given the state sequence, as defined in (4.13) [16]. The vector $\boldsymbol{\theta}$ parametrizes the weighting of the discrepancy $\boldsymbol{z}$, which captures the divergence between the learned and failed behaviors, and λ is a regularization coefficient.

Zhao *et al.* [15] extend this framework by introducing time-series labels to reformulate the trajectory-level constraints in a state-wise manner, while keeping the original objective function unchanged. In this formulation, a label variable l_t is defined for each time step, indicating the likelihood that the state at time t corresponds to expert or failed behavior, with: $p(l_t = \text{expert}) + p(l_t = \text{failed}) = 1$. The expected feature expectation under the learned policy then becomes: $\mathbb{E}_{(\boldsymbol{s}_t,l_t)\sim p(\tau)}[\phi(\boldsymbol{s}_t)l_t]$. While this reformulated constraint enables state-wise safety enforcement without adding a separate optimization loop, it comes at the cost of increased computational cost due to the need for labeling each state in the demonstration data, especially, in continuous, high-dimensional state-spaces.

IV. Methodology

This section presents our proposed method with state-wise safety constraints, designed for safe and stable behavior prediction in continuous, high-dimensional environments. Unlike prior methods that enforce safety at the trajectory level or rely on exhaustive state labeling, our approach introduces a behavior discriminator that models the safety of individual states within context. This discriminator is jointly optimized within the reward learning process, allowing efficient integration of both expert and failed demonstrations without additional optimization loops. We begin by describing the design of behavior discriminator, followed by its integration into the IRL reward learning process. The framework of the proposed method is illustrated in Fig. 1.

A. Behavior discriminator design

The core idea of our framework is to assign safety-aware labels to individual states using a learned behavior discriminator handle continuous, high-dimensional state spaces. Let $l_t \in \{0, 1\}$ denote the binary safety label at time step t, where $l_t = 1$ indicates a safe (expert-like) state and $l_t = 0$ represents an unsafe (failed) state. The behavior discriminator aims to generate state-wise labels which is formulated as a label function parametric by ψ regarding the current state, $\boldsymbol{s}_t$, and environmental factor $\boldsymbol{c}_t$,

$$l_t := l_\psi(\boldsymbol{s}_t|\boldsymbol{c}_t). \tag{2}$$

Selecting an appropriate model for ψ is critical, particularly in continuous, high-dimensional state-spaces. Prior work typically assumed low-dimensional discrete settings, where safety labels for unknown states are generated by identifying nearest neighbors [15]. However, in high-dimensional spaces, defining a meaningful distance metric becomes non-trivial, and even exhaustive evaluation in a finely discretized state space, while tractable in lower dimensions, quickly becomes computationally infeasible. Moreover, continuous spaces introduce significant epistemic uncertainty, which further complicates reliable safety label predictions [17]. To

address these challenges, we adopt a more expressive model for ψ, specifically, a neural network, capable of capturing complex, diverse behaviors and generalizing across sparse or dispersed state regions.

B. Optimization problem with state-wise safety constraints

To update the safe behavior discriminator without increasing the overall optimization complexity, we introduce an additional term to the objective function and optimize it jointly within the main reward update loop. This additional term is designed to maximize the entropy of the sampled safety labels conditioned on the environmental context. Let $\mathcal{L}^h$ denotes the set of labels sampled by the behavior discriminator based on a sequence of states over a horizon h, given environmental contexts $\mathcal{C}^h$. Incorporating this term leads to the following augmented optimization problem:

$$\begin{aligned}\max_{\pi,\boldsymbol{\theta},\boldsymbol{z}} \quad & \mathcal{H}(\mathcal{A}^h||\mathcal{S}^h) + \boldsymbol{\theta}\boldsymbol{z} - \frac{\lambda}{2}||\boldsymbol{\theta}||^2 + \mathcal{H}(\mathcal{L}^h|\mathcal{C}^h) \\ \text{s.t.} \quad & \mathbb{E}_{(\boldsymbol{s}_t,l_t)\sim p(\tau)}[\phi(\boldsymbol{s}_t)l_t] - \mathbb{E}_{(\boldsymbol{s}_t,l_t)\sim\mathcal{D}}[\phi(\boldsymbol{s}_t)l_t] = \mathbf{0} \\ & \mathbb{E}_{(\boldsymbol{s}_t,l_t)\sim p(\tau)}[\phi(\boldsymbol{s}_t)l_t] - \mathbb{E}_{(\boldsymbol{s}_t,l_t)\sim\mathcal{F}}[\phi(\boldsymbol{s}_t)l_t] = \boldsymbol{z}.\end{aligned} \tag{3}$$

The optimization problem is then addressed using the method of Lagrange multipliers. To begin, the equality constraints are relaxed, resulting in the following Lagrangian formulation:

$$\begin{aligned}&\mathcal{L}(\pi,\boldsymbol{z},\boldsymbol{\theta},l_t,\boldsymbol{w}^{\mathcal{D}},\boldsymbol{w}^{\mathcal{F}}) := \\ &\mathcal{H}(\mathcal{A}^h||\mathcal{S}^h) + \boldsymbol{\theta}\boldsymbol{z} - \frac{\lambda}{2}||\boldsymbol{\theta}||^2 + \mathcal{H}(\mathcal{L}^h|\mathcal{C}^h) \\ &+ \boldsymbol{w}^{\mathcal{D}^\top}\left(\mathbb{E}_{(\boldsymbol{s}_t,l_t)\sim p(\tau)}[\phi(\boldsymbol{s}_t)l_t] - \mathbb{E}_{(\boldsymbol{s}_t,l_t)\sim\mathcal{D}}[\phi(\boldsymbol{s}_t)l_t]\right) \\ &+ \boldsymbol{w}^{\mathcal{F}^\top}\left(\mathbb{E}_{(\boldsymbol{s}_t,l_t)\sim p(\tau)}[\phi(\boldsymbol{s}_t)l_t] - \mathbb{E}_{(\boldsymbol{s}_t,l_t)\sim\mathcal{F}}[\phi(\boldsymbol{s}_t)l_t] - \boldsymbol{z}\right),\end{aligned} \tag{4}$$

where $\boldsymbol{w}^{\mathcal{D}}$ and $\boldsymbol{w}^{\mathcal{F}}$ denote the Lagrange multipliers associated with the constraints derived from expert and failed demonstrations, respectively.

Next, to eliminate the primary variables, we take the partial derivatives of the Lagrangian with respect to $\boldsymbol{z}$ and $\boldsymbol{\theta}$ and set them to zero. This yields the following optimality conditions:

$$\begin{aligned}&\nabla_{\boldsymbol{z}}\mathcal{L}(\pi,\boldsymbol{z},\boldsymbol{\theta},l_t,\boldsymbol{w}^{\mathcal{D}},\boldsymbol{w}^{\mathcal{F}}) = \boldsymbol{\theta} - \boldsymbol{w}^{\mathcal{F}} = 0 \Rightarrow \boldsymbol{\theta} = \boldsymbol{w}^{\mathcal{F}}, \\ &\nabla_{\boldsymbol{\theta}}\mathcal{L}(\pi,\boldsymbol{z},\boldsymbol{\theta},l_t,\boldsymbol{w}^{\mathcal{D}},\boldsymbol{w}^{\mathcal{F}}) = \boldsymbol{z} - \lambda\boldsymbol{\theta} = 0 \Rightarrow \boldsymbol{z} = \lambda\boldsymbol{\theta}.\end{aligned} \tag{5}$$

Then, we substitute (5) back into the Lagrangian and compute its derivative with respect to the policy π at time step t, where $\pi_t = \pi(\boldsymbol{s}_t,\boldsymbol{a}_t)$. This yields the following expression:

$$\begin{aligned}&\nabla_{\pi_t}\mathcal{L}(\pi,l_t,\boldsymbol{w}^{\mathcal{D}},\boldsymbol{w}^{\mathcal{F}}) = \\ &\quad p(\boldsymbol{a}_{1:t-1},\boldsymbol{s}_{1:t})\Big(-\log\pi(\boldsymbol{a}_t,\boldsymbol{s}_t) - 1 + \mathcal{H}(\mathcal{A}^{t+1:h}||\mathcal{S}^{t+1:h}) \\ &\quad + \left(\boldsymbol{w}^{\mathcal{D}} + \boldsymbol{w}^{\mathcal{F}}\right)^\top \mathbb{E}_{p(\boldsymbol{a}_{t+1:h},\boldsymbol{s}_{t+1:h}|\boldsymbol{a}_{1:t-1},\boldsymbol{s}_{1:t})}[\phi(\boldsymbol{s}_t)l_t]\Big).\end{aligned} \tag{6}$$

To compute the policy efficiently in high-dimensional continuous state-spaces, we adopt a trajectory sampling approach inspired by motion planning techniques, as proposed in [18]. In contrast, traditional IRL frameworks, such as that of Ziebart [16] typically solve for the optimal policy using dynamic programming methods, including the Bellman equation and forward-backward message passing algorithms [19]. While effective in discrete settings, these methods become prohibitively expensive in continuous settings due to the need for dense discretization and accurate function approximation. Hence, we adopt sampling-based approach addresses this limitation by generating trajectories that imitate experts more efficiently without relying on exhaustive dynamic programming.

Once the policy is computed for the current optimization iteration, the parameters of reward function, $\boldsymbol{w}^{\mathcal{D}}$ and $\boldsymbol{w}^{\mathcal{F}}$, and the behavior discriminator are updated via gradient descent as follows:

$$\begin{aligned}&\nabla_{\boldsymbol{w}^{\mathcal{D}}}\mathcal{L}(\pi,l_t,\boldsymbol{w}^{\mathcal{D}},\boldsymbol{w}^{\mathcal{F}}) = \\ &\quad \mathbb{E}_{(\boldsymbol{s}_t,l_t)\sim p(\tau)}[\phi(\boldsymbol{s}_t)l_t] - \mathbb{E}_{(\boldsymbol{s}_t,l_t)\sim\mathcal{D}}[\phi(\boldsymbol{s}_t)l_t], \\ &\nabla_{\boldsymbol{w}^{\mathcal{F}}}\mathcal{L}(\pi,l_t,\boldsymbol{w}^{\mathcal{D}},\boldsymbol{w}^{\mathcal{F}}) = \\ &\quad \mathbb{E}_{(\boldsymbol{s}_t,l_t)\sim p(\tau)}[\phi(\boldsymbol{s}_t)l_t] - \mathbb{E}_{(\boldsymbol{s}_t,l_t)\sim\mathcal{F}}[\phi(\boldsymbol{s}_t)l_t] - \lambda\boldsymbol{w}^{\mathcal{F}},\end{aligned} \tag{7}$$

and

$$\begin{aligned}\nabla_{l_t}\mathcal{L}(\pi,l_t,\boldsymbol{w}^{\mathcal{D}},\boldsymbol{w}^{\mathcal{F}}) &= \nabla_{l_t}\mathcal{H}(\mathcal{L}^h|\mathcal{C}^h) \\ &+ \boldsymbol{w}^{\mathcal{D}^\top}\left(\mathbb{E}_{(\boldsymbol{s}_t,l_t)\sim p(\tau)}[\phi(\boldsymbol{s}_t)] - \mathbb{E}_{(\boldsymbol{s}_t,l_t)\sim\mathcal{D}}[\phi(\boldsymbol{s}_t)]\right) \\ &+ \boldsymbol{w}^{\mathcal{F}^\top}\left(\mathbb{E}_{(\boldsymbol{s}_t,l_t)\sim p(\tau)}[\phi(\boldsymbol{s}_t)] - \mathbb{E}_{(\boldsymbol{s}_t,l_t)\sim\mathcal{F}}[\phi(\boldsymbol{s}_t)]\right).\end{aligned} \tag{8}$$

It is important to note that the update of the behavior discriminator is influenced not only by the cross-entropy loss, but also by the reward function. In the gradient expression shown in (8), the first term corresponds to the derivative of the cross-entropy loss. In addition to this, the final two terms capture differences in expected rewards between the generated trajectories and those from expert and failed demonstrations. These additional terms help stabilize the discriminator's learning process and encourage it to generate labels that effectively distinguish expert-like behavior from failed behavior in terms of their associated rewards.

V. Experiments

A. Environmental setup

We evaluated our proposed method using the CARLA simulator [20] in two representative urban driving scenarios: obstacle avoidance and stop-line compliance. These scenarios are commonly used benchmarks for assessing decision-making performance in structured environments. Bird's-eye views of both scenarios are illustrated in Section V-A [15].

In the obstacle-avoidance scenario, the driving tasks are defined as: (1) avoiding collisions with static obstacles, and (2) reaching a designated goal position. For the stop-line scenario, the tasks include: (1) coming to a complete stop near the stop-line within an acceptable distance, and (2) reaching the final goal position. Failed demonstrations are characterized by either collisions with obstacles or by driving through the stop-line at an unsafe speed.

The dataset comprises over 80 trajectories, totaling approximately one hour of driving data. To improve learning performance, the data were downsampled to 3 Hz from

the original 30 Hz sampling rate recorded in the CARLA simulator.

The experiments model driving behavior using a five-dimensional continuous state space, defined as $\boldsymbol{s}_t = (x_t, y_t, \theta_t, v_t, \omega_t)^\top \in \mathbb{R}^5$, where x_t and y_t denote the vehicle's x–y positions, θ_t, v_t, and ω_t indicate the heading angle, velocity, and angular velocity, respectively. The corresponding action space is two-dimensional and given by $\boldsymbol{a}_t = (a_t, \alpha_t)^\top$, where a_t represents acceleration and α_t denote angular acceleration. Motion transitions follow nonholonomic vehicle dynamics.

The behavior discriminator in the proposed method is implemented as a neural network which is composed by three fully connected layers with the dimensions of five (the same dimension as states), 32 and 2 (binary classification). To prevent overfitting and promote generalization, L2 weight regularization is applied to all linear layers, controlled by a regularization coefficient, 10^{-5}. The learning rate for obstacle-avoidance and stop-line scenario is set as 10^{-5} and 10^{-7}, respectively.

The reward function is defined as a linear combination of scenario-specific features: $r_{\boldsymbol{w}}(\boldsymbol{s}_t|\mathcal{C}) = \boldsymbol{w}^\top \phi(\boldsymbol{s}_t|\mathcal{C})$, where $\phi(\boldsymbol{s}_t|\mathcal{C})$ encodes context-aware features based on road geometry, static object locations, and desired velocities within specific regions. In the following experiments, we assume the environmental context $\mathcal{C}$ remains time-invariant over the trajectory horizon, *i.e.*, $\mathcal{C}^h = \mathcal{C}$.

B. Comparison methods

We conducted experiments to compare our proposed method against several baselines within the MaxEnt IRL framework, focusing on their ability to handle various types of demonstrations in continuous, high-dimensional state-spaces.

MaxEntIRL: We implemented the original MaxEntIRL framework [5], augmented with a rapid-random generate tree (RRT)-based motion planner [18] to enable efficient and stable trajectory generation in continuous state-spaces. Two variants were evaluated to assess their behavior under different types of input data: *MaxEntIRL($\mathcal{D}$)* corresponds to the traditional version using only expert demonstrations, while *MaxEntIRL($\mathcal{DF}$)* extends the traditional one to incorporate both expert and failed demonstrations.

IRLF: We extended the original IRLF method [7] by integrating the same RRT-based motion planner to enable trajectory sampling in continuous settings. This method uses both expert and failed demonstrations for learning.

SIRLFD: We implemented the original optimization formulation from SIRLFD [15], replacing the original policy inference mechanism with a motion planner. The original algorithm relies on probabilistic inference via dynamic programming [19], which becomes computationally prohibitive in high-dimensional continuous state-spaces. Moreover, the original method cannot label unseen states not covered in the demonstrations. To address this, we modified it to use a nearest-neighbor search strategy that assigns labels to unknown states based on the closest known state in the demonstration data.

C. Evaluation metrics

We evaluated performance using two criteria: imitation performance, measured by the modified Hausdorff distance (MHD) [21], and driving performance, assessed through task success rate.

MHD was used to quantify the similarity between generated trajectories and the demonstration data. We report both the 50th percentile (mhd50) and 90th percentile (mhd90) values to capture central tendency and worst-case performance, respectively. Since the agent is expected to imitate expert demonstrations while avoiding failed behaviors, MHD values relative to expert demonstrations, referred to as positive MHD, should be minimized. Conversely, MHD values relative to failed demonstrations, negative MHD, should be maximized, reflecting the agent's ability to avoid unsafe or undesirable behaviors.

Driving performance was measured using task success rate, which indicates the percentage of generated trajectories that satisfy the predefined task requirements described in Section V-A. For example, in the stop-line scenario, a trajectory that correctly stops at the designated stop line but fails to reach the final goal region would be counted as 50% successful.

D. Results

1) Quantitative evaluation on imitation and driving performance: To quantitatively evaluate both imitation quality and task performance, we compared our proposed method against several baselines in two driving scenarios: obstacle avoidance and stop-line scenarios. Tables Ia and Ib present the results.

The proposed method, which incorporates state-wise safety constraints through a learned behavior discriminator, consistently outperforms all baselines in both both imitating (measured by MHD) and driving performance (measured by success rate) both scenarios. This demonstrates the effectiveness and robustness of our approach in modeling safe and expert-like driving behavior in continuous, high-dimensional state-spaces.

However, in the stop-line scenario, the imitation performance of our method with respect to failed demonstrations (*i.e.*, negative MHD) was notably reduced. Interestingly, SIRLFD achieved the highest negative MHD values in this scenario, indicating a stronger ability to avoid unsafe behaviors. While both methods enforce state-wise constraints, SIRLFD relies on nearest-neighbor matching for safety labeling, whereas our method uses a neural network-based discriminator. This result suggests that in tasks requiring fine-grained vehicle control, such as precise stopping behavior near a stop line, nearest neighbor labeling may outperform learned label generation when the discriminator lacks sufficient resolution.

Furthermore, the IRLF baseline, which imposes trajectory-level safety constraints, revealed a counterintuitive pattern: its

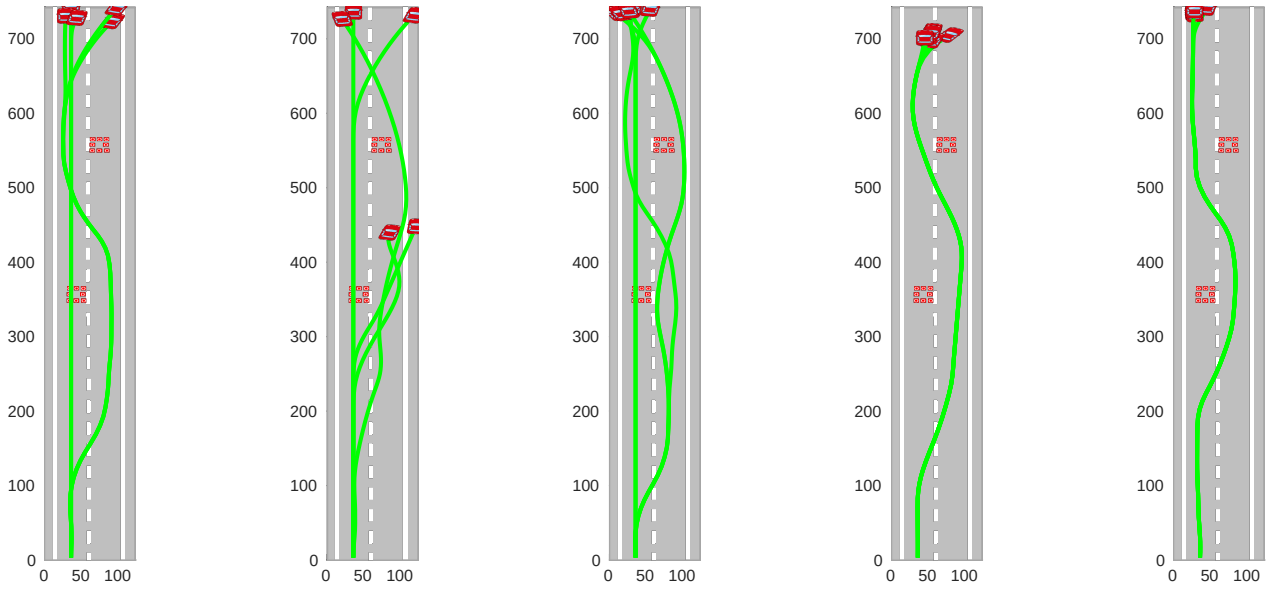
TABLE I: Comparison of imitation and driving performances across two driving scenarios. Imitation performance is measured by trajectory similarity to expert and failed demonstrations, while driving performance is evaluated using task success rate. Each value represents the mean and standard deviation over 300 generated trajectories per method. The best performance metrics are highlighted in bold.

(a) Obstacle-avoidance scenario.

	positive		**negative**		
method	mhd50↓	mhd90↓	mhd50↑	mhd90↑	success rate↑
MaxEntIRL($\mathcal{D}$)	1.11 ± 0.52	4.54 ± 1.44	13.10 ± 4.19	35.16 ± 8.45	$84.37\% \pm 5.55\%$
MaxEntIRL($\mathcal{DF}$)	3.21 ± 1.41	10.10 ± 3.74	9.67 ± 2.84	25.57 ± 6.22	$63.46\% \pm 5.38\%$
IRLF	1.37 ± 0.36	$\mathbf{3.50 \pm 0.69}$	7.62 ± 2.79	38.55 ± 4.40	$79.47\% \pm 5.19\%$
SIRLFD	0.84 ± 0.31	4.32 ± 1.14	12.99 ± 4.25	34.73 ± 8.44	$89.07\% \pm 6.76\%$
ours	$\mathbf{0.62 \pm 0.37}$	3.94 ± 1.53	$\mathbf{13.51 \pm 4.31}$	$\mathbf{42.07 \pm 7.98}$	$\mathbf{90.21\% \pm 5.46\%}$

(b) Stop-line scenario.

	positive		**negative**		
method	mhd50↓	mhd90↓	mhd50↑	mhd90↑	success rate↑
MaxEntIRL($\mathcal{D}$)	13.94 ± 4.85	26.47 ± 7.05	10.86 ± 5.12	19.63 ± 7.32	$35.31\% \pm 9.04\%$
MaxEntIRL($\mathcal{DF}$)	1.33 ± 0.42	5.33 ± 1.78	1.42 ± 0.24	5.33 ± 1.45	$58.93\% \pm 9.14\%$
IRLF	12.11 ± 5.07	22.45 ± 6.78	5.82 ± 3.76	10.18 ± 5.30	$41.30\% \pm 8.28\%$
SIRLFD	12.74 ± 4.91	25.16 ± 7.03	$\mathbf{11.88 \pm 5.42}$	$\mathbf{20.96 \pm 7.69}$	$32.34\% \pm 9.26\%$
ours	$\mathbf{0.78 \pm 0.20}$	$\mathbf{3.33 \pm 0.91}$	1.92 ± 1.02	8.29 ± 2.99	$\mathbf{66.88\% \pm 8.80\%}$



(a) MaxEntIRL($\mathcal{D}$) (b) MaxEntIRL($\mathcal{DF}$) (c) IRLF (d) SIRLFD (e) ours

Fig. 2: Visualization of generated trajectories in the obstacle-avoidance scenario. Red rectangles indicate static obstacles, while green lines represent the top five trajectories with the highest predicted rewards, subject to nonholonomic vehicle dynamics.

imitation performance on failed demonstrations was worse than that of MaxEntIRL($\mathcal{D}$), which uses only expert demonstrations. This outcome implies that trajectory-level feedback may be insufficient for ensuring safe behavior, particularly when failed trajectories include mostly safe behaviors with only brief unsafe segments (*e.g.*, a late-stage failure). These findings underscore the importance of enforcing safety at the state level for fine-grained behavioral fidelity.

2) Qualitative evaluation on predicted driving behaviors: To provide a qualitative comparison, we visualize the trajectories generated by the proposed method and baseline approaches in the obstacle-avoidance scenario. Although the full state space is five-dimensional, the results are projected onto the x–y plane for visual clarity. Note that all the generated trajectories in Fig. 2 share the same time horizon. Some of the final positions differ due to the randomness of RRT motion planner.

As shown in Fig. 2(e), the proposed method generates the most stable and consistent trajectories, with minimal dispersion and smooth paths that successfully avoid all obstacles while reaching the goal position. In comparison, while the state-wise baseline SIRLFD also avoids collisions, its trajectories tend to pass closer to obstacles and exhibit less smoothness as shown in Fig. 2(d). The advantage of the proposed method highlights the benefits of learning safety labels through a behavior discriminator rather than relying

solely on nearest-neighbor matching.

On the other hand, trajectory-level baselines such as MaxEntIRL($\mathcal{D}$), MaxEntIRL($\mathcal{DF}$), and IRLF as shown in Figs. 2(a)–(c), often generate unsafe or unstable behaviors. For example, several trajectories demonstrate unsafe behavior by either colliding with obstacles or failing to reach the goal due to low-speed. These observations further support the necessity of state-wise safety enforcement and demonstrate that incorporating a behavior discriminator contributes significantly to more reliable and safer behavior prediction in complex driving scenarios.

VI. Conclusion

This paper presented a novel IRL framework that incorporates state-wise safety constraints through a behavior discriminator, addressing critical limitations of trajectory-wise IRL methods in safety-sensitive domains like autonomous driving. By introducing a neural network-based behavior discriminator trained within the main IRL optimization loop, the proposed approach ensures state-wise safety without increasing computational complexity. Extensive experiments in simulated driving scenarios using the CARLA simulator demonstrated that our method outperforms baseline approaches in both imitation and driving performances. The results confirm that enforcing safety at the state level is essential for stable and reliable behavior prediction, particularly in high-dimensional continuous environments. Future work may extend this approach to handle dynamic environmental contexts, such as considering other vehicles and pedestrians, and further improve the generalization capability in unseen scenarios.

Acknowledgements

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

References

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

[2] Mehmet Fatih Ozkan and Yao Ma. Modeling driver behavior in car-following interactions with automated and human-driven vehicles and energy efficiency evaluation. *IEEE Access*, 9:64696–64707, 2021.

[3] Weichao Wang, Lei Jiang, Shiran Lin, Hui Fang, and Qinggang Meng. Imitation learning based decision-making for autonomous vehicle control at traffic roundabouts. *Multimedia Tools and Applications*, 81(28):39873–39889, 2022.

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

[5] 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)*, 2008.

[6] Gunmin Lee, Dohyeong Kim, Wooseok Oh, Kyungjae Lee, and Songhwai Oh. MixGAIL: Autonomous driving using demonstrations with mixed qualities. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020.

[7] Kyriacos Shiarlis, Joao Messias, and Shimon Whiteson. Inverse reinforcement learning from failure. In *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 2016.

[8] Kyungjae Lee, Sungjoon Choi, and Songhwai Oh. Inverse reinforcement learning with leveraged gaussian processes. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016.

[9] Xu Xie, Changyang Li, Chi Zhang, Yixin Zhu, and Song-Chun Zhu. Learning virtual grasp with failed demonstrations via bayesian inverse reinforcement learning. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019.

[10] Changquan Wang and Yun Wang. Safe autonomous driving with latent dynamics and state-wise constraints. *Sensors*, 24(10), 2024.

[11] Weiye Zhao, Tairan He, Rui Chen, Tianhao Wei, and Changliu Liu. State-wise safe reinforcement learning: a survey. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, 2023.

[12] Eitan Altman. *Constrained Markov decision processes*. Routledge, 2021.

[13] Akifumi Wachi and Yanan Sui. Safe reinforcement learning in constrained markov decision processes. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 9797–9806, 2020.

[14] Sinong Zhan, Yixuan Wang, Qingyuan Wu, Ruochen Jiao, Chao Huang, and Qi Zhu. State-wise safe reinforcement learning with pixel observations. In *Proceedings of the Annual Learning for Dynamics & Control Conference (L4DC)*, pages 1187–1201, 2024.

[15] 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.

[16] Brian D. Ziebart. Modeling purposeful adaptive behavior with the principle of maximum causal entropy. 12 2010.

[17] Eyke Hüllermeier and Willem Waegeman. Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods. *Machine learning*, 110(3):457–506, 2021.

[18] 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.

[19] Sergey Levine. Reinforcement learning and control as probabilistic inference: Tutorial and review. In *arXiv preprint arXiv:1805.00909*, 2018.

[20] 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)*, 2017.

[21] 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.