Inverse Reinforcement Learning with Failed Demonstrations towards Stable Driving Behavior Modeling

Minglu Zhao and Masamichi Shimosaka[1]

***Abstract*— Driving behavior modeling is crucial in autonomous driving systems for preventing traffic accidents. Inverse reinforcement learning (IRL) allows autonomous agents to learn complicated behaviors from expert demonstrations. Similar to how humans learn by trial and error, failed demonstrations can help an agent avoid failures. However, expert and failed demonstrations generally have some common behaviors, which could cause instability in an IRL model. To improve the stability, this work proposes a novel method that introduces time-series labeling for the optimization of IRL to help distinguish the behaviors in demonstrations. Experimental results in a simulated driving environment show that the proposed method converged faster than and outperformed other baseline methods. The results also show consistency for various data balances of the number of expert and failed demonstrations.**

I. Introduction

Autonomous driving is a crucial technology for solving social issues such as labor shortages in an aging society. The intelligent systems of autonomous driving must be capable of handling more complex scenarios in urban environments than highways [2]. Thus, many functionalities such as traffic sign recognition [17] and pedestrian trajectory prediction [12] are necessary for urban environments. In addition to modeling such surrounding traffic, it is important to model the decision-making process of the drivers in order to follow traffic signs and avoid pedestrians.

Driving behavior modeling (DBM) is a technique to model the decision-making process of humans during driving [8]. In recent years, many works have reported that the Markov decision process (MDP) is a powerful method to model driving behaviors [15]. In an MDP, a sequence of optimal driving behaviors is obtained by maximizing the rewards. However, designing the reward function of driving behaviors is nontrivial owing to the complex nature of human behaviors in various driving scenarios [14].

Inverse reinforcement learning (IRL) is a promising approach to the reward design issue. It provides a data-driven solution where the reward functions are directly learned from demonstration data. Many IRL methods have been researched over the last 20 years, such as max margin IRL [9], Bayesian IRL [11] and max entropy IRL [18]. Max entropy IRL is one of the most common frameworks because it is a probabilistic model that can learn not only optimal behaviors but also suboptimal behaviors. However, traditional IRL methods imitate whatever is demonstrated in the data, so typically, only expert demonstrations are fed to an IRL framework.

Humans generally learn through trial and error. When analogizing such human nature to machine learning methods, it follows that failed demonstrations are equally as important as expert demonstrations. To improve learning, a new framework that can not only imitate expert demonstrations but also exploit failed demonstrations is necessary. The advantage of such a framework would be the improved capability of distinguishing the failed behaviors from the expert behaviors based on the learned reward function. In this work, such a framework is called IRL from failed demonstrations (IRLFD). Some successful work based on such an IRLFD framework has been conducted, such as IRL with failures (IRLF) [13] and Bayesian IRL with failures (BIRLF) [16].

However, expert and failed demonstrations generally have some common behaviors that cause instability in the IRL model [5]. Such a problem is also called the *behavior overlapping issue* and usually occurs due to improper demonstration labeling. For example, we usually directly label a demonstration as expert or failed depending on whether any failed behaviors exist in the demonstration. However, apart from the failed behaviors, other behaviors in such demonstrations are sometimes almost identical to the expert behaviors in expert demonstrations. For such common behaviors, the IRLFD method will imitate it if it is taken from an expert demonstration; in contrast, the behavior will not be imitated if it is taken from a failed demonstration. Such a conflict would cause instability in the training process of an IRLFD method.

In this work, we introduce a time-series labeling-based solution to address the behavior overlapping issue and improve the stability of the training process. In doing so, we address a key difficulty in the existing IRLFD framework: the problem is over-constrained as it does not imitate anything from the failed demonstration even if some behaviors are actually good. With the proposed time-series labeling, we relax the constraint so that it will not imitate the failed behaviors but also imitate the expert behavior from a failed demonstration. In addition, we name the proposed framework as stable IRLFD (SIRLFD).

Our contributions are summarized as follows:

- We propose a novel time-series labeling-based IRLFD framework to handle the behavior overlapping issue. This guarantees a behavior-conflict-free training process to improve stability.
- We construct a stable and accurate driving behavior prediction method based on the IRLFD framework.

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

The predicted driving behaviors are generated using a modified graph model for MDP involving time-series labeling.

- We demonstrate the superiority of the proposed SIRLFD method over some conventional approaches through some driving scenarios in a simulated driving environment.

II. Related work

A. IRL for driving behavior modeling

Driving behaviors are typically difficult to model using any handcrafted reward function, which makes IRL a powerful solution to modeling driving behaviors by learning rewards from demonstrations. Various existing works have been presented on using IRL-based methods to model driving behaviors, such as avoiding obstacles [4] and proper velocity control under various weather conditions [10]. However, these methods only utilize expert demonstrations; hence, although those trained driving behavior models can imitate expert behaviors to drive safely, the models cannot intentionally avoid dangers.

B. IRL from failed demonstrations

Failed demonstrations are necessary to avoid dangers. IRL algorithms that also learn from failed demonstrations have already shown notable performance improvement over those that only learn from expert demonstrations [5], [6], [13], [16]. However, the performance of these methods drops significantly with a large number of overlapping behaviors between expert and failed demonstrations.

A common approach in these methods is to maximize the similarity of the generated behaviors to the expert demonstrations. IRLF [13] presents a visiting frequency-based similarity measurement and optimizes such similarity using two gradients in the training process. One gradient aims to imitate the expert demonstrations, whereas another gradient aims to avoid imitating the failed demonstrations. However, with many overlapped behaviors, these two gradients would conflict with each other and result in a significant performance drop. BIRLF [16] introduces a halfspace-induced potential measurement instead of a visiting frequency-based measurement, but the overlapping issue still exists. GPIRL [6] utilizes Gaussian kernel-based measurement, but the performance highly depends on the kernel design, where the design effort is not trivial. Therefore, the behavior overlapping issue must be solved to improve the stability of the gradient-based training process.

III. Formulation of IRLFD

This section introduces the base of our proposed method. Specifically, we first describe the problem setting of IRL based on the max entropy IRL framework [18] and a solution based on the Lagrange multiplier method. We then introduce the optimization problem of IRLFD which is extended from IRL with an additional constraint. Note that we concentrate on the max entropy IRL method because it is a probabilistic model and deals with suboptimality.

A. Problem setting of IRL

We define the discrete state space $\mathcal{S}$ and the discrete action space $\mathcal{A}$. When a discrete state $s_t \in \mathcal{S}$ and an action $a_t \in \mathcal{A}$ are given, the agent moves to the next state with a transition probability $P : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ with $p(s_{t+1}|s_t, a_t)$. Note that the transition is Markovian in that the next state only depends on the current state and action. During the transition, the agent obtains an immediate reward $R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ with $R(s_t, a_t)$ at the t-th timestep. We aim to obtain a policy $\pi : \mathcal{S} \times \mathcal{A} \to [0, 1]$ with $\pi(s_t, a_t) = p(a_t|s_t)$ that maximizes the expected future reward. This expected future reward can be expressed as the reward value of being in a particular state and following a given policy over the finite time horizon $\{1, \dots, h\}$,

$$V^{\pi}(s) = \mathbb{E}_{\pi}\left[\sum_{t=1}^{h} R(s_t, a_t)\middle| s_1 = s\right]. \tag{1}$$

(1) is also called a state-value function for policy π. In addition, a behavior is represented as a pair of states and actions. A demonstration consists of a sequence of behaviors; consequently, it is represented as a sequence of state-action pairs, such as $\tau_i = \left\langle \left(s_1^{(\tau_i)}, a_1^{(\tau_i)}\right), \dots, \left(s_{h_i}^{(\tau_i)}, a_{h_i}^{(\tau_i)}\right)\right\rangle$, where τ_i indicates the $i-$th demonstration in the dataset and h_i indicates the time horizon of a demonstration.

The objective of IRL is to learn the reward function given demonstrations of behaviors in the environment, $\mathcal{D}$. The optimal behaviors are predicted based on an immediate reward and an expected future reward. Specifically, to learn from the demonstrations, the learned rewards should be as close to the rewards from demonstrations as possible. We denote $\mathbb{E}_{s\sim\mathcal{D}}\left[V^{\pi_{\mathcal{D}}}(s)\right]$ as a marginal estimation of the state-value function over the states sampled from trajectories. Thus, the optimization problem of a max entropy IRL framework can be expressed as

$$\begin{aligned} \max_{\pi} \quad & H(\mathcal{A}^h||\mathcal{S}^h) \\ \text{s.t.} \quad & \mathbb{E}_{s\sim P}\Big[V^{\pi}(s)\Big] = \mathbb{E}_{s\sim\mathcal{D}}\Big[V^{\pi_{\mathcal{D}}}(s)\Big], \end{aligned} \tag{2}$$

where $H(\mathcal{A}^h||\mathcal{S}^h)$ indicates the causal entropy, which is a conditional entropy of the action sequence $\mathcal{A}^h$ causally conditioned on the state sequence $\mathcal{S}^h$. With the Markovian transition probability, the causal entropy is formed as

$$\begin{aligned} H(\mathcal{A}^h||\mathcal{S}^h) &= \mathbb{E}_{p(\mathcal{S}^h,\mathcal{A}^h)}\left[-\log p(a_t|s_t)\right] \\ &= -\sum_{s_t\in\mathcal{S}^h}\sum_{a_t\in\mathcal{A}^h} p(s_t, a_t)\log p(a_t|s_t), \end{aligned} \tag{3}$$

where,

$$p(s_t, a_t) = p(s_{t-1}, a_{t-1})p(s_t|s_{t-1}, a_{t-1})p(a_t|s_t). \tag{4}$$

Note that $p(a_t|s_t)$ represents the policy π, and $\pi_{\mathcal{D}}$ and π are the expert policy and learned policy, respectively [1].

The Lagrange multiplier method is a common way to solve such constrained optimization problems shown in (2). Let the constraint be a linear relationship with the Lagrange

multiplier w. Then, the Lagrangian function gives

$$L(\pi, w) = H(\mathcal{A}^h||\mathcal{S}^h) + w\left(\mathbb{E}_{s\sim P}\left[V^{\pi}(s)\right] - \mathbb{E}_{s\sim \mathcal{D}}\left[V^{\pi_{\mathcal{D}}}(s)\right]\right). \tag{5}$$

There are two variables to be optimized with the Lagrangian function shown in (5): the policy π and the Lagrangian multiplier w.

First, the policy is optimized through a probabilistic inference process. In this work, we focus on the max entropy IRL method because it is a probabilistic model. Such a probabilistic model can be solved as a probabilistic inference regarding the policy search process [7]. This concept is similar to dynamic programming. We first compute backward messages regarding the reward function from t' to h:

$$\beta_t(s_t, a_t) = \exp\left(\sum_{t=t'}^{h} R(s_t, a_t) p(s_{t+1}|s_t, a_t)\right). \tag{6}$$

We also require messages denoting the state-only probability by integrating out the action:

$$\beta_t(s_t) = \sum_{a_t \in \mathcal{A}} \beta_t(s_t, a_t) p(a_t|s_t). \tag{7}$$

In dynamic programming, the recursive message passing for computing $\beta_t(s_t, a_t)$ proceeds from the last timestep $t = h$ and backward through time to $t = 1$ with the following form:

$$\beta_t(s_t, a_t) = \sum_{s_{t+1}\in \mathcal{S}} \beta_{t+1}(s_{t+1}) p(s_{t+1}|s_t, a_t) \exp\Big(R(s_t, a_t)\Big). \tag{8}$$

We then introduce such backward messages in the log space, and such log-space messages correspond to the state-action and state value functions:

$$\begin{aligned} Q(s_t, a_t) &= \log \beta_t(s_t, a_t), \\ V(s_t) &= \log \beta_t(s_t). \end{aligned} \tag{9}$$

Finally, we can obtain the policy as

$$\pi(s_t, a_t) = \exp\Big(Q(s_t, a_t) - V(s_t)\Big). \tag{10}$$

This proves that the optimal policy can be recovered with a probabilistic inference process.

Once the optimal policy π with a given reward function is computed, the Lagrangian multiplier w is updated via gradient descent by noting that

$$\nabla_w L(\pi, w) = \mathbb{E}_{s\sim P}\Big[V^{\pi}(s)\Big] - \mathbb{E}_{s\sim \mathcal{D}}\Big[V^{\pi_{\mathcal{D}}}(s)\Big], \tag{11}$$

where the learned reward $\mathbb{E}_{s\sim P}\Big[V^{\pi}(s)\Big]$ is computed by rolling out policy π. The optimization process terminates once the Lagrangian multiplier w converges to the optimal solution.

B. Optimization problem of IRLFD

IRLFD learns the reward function with two types of demonstrations, failed $\mathcal{F}$ and expert $\mathcal{D}$ demonstrations. Similar to the IRL framework, IRLFD aims to minimize the difference between the learned reward and the reward from the expert demonstrations. Additionally, IRLFD needs to maximize the difference between the learned reward and the reward from failed demonstrations to learn from these failed demonstrations by not imitating them. Consequently, an additional constraint is introduced to the optimization problem with a variable $z \in \mathbb{R}$ to maximize such dissimilarities to the failed demonstrations:

$$\begin{aligned} \max_{\pi, z, \theta} \quad & H(\mathcal{A}^h||\mathcal{S}^h) + \theta z - \frac{\lambda}{2}||\theta||^2 \\ \text{s.t.} \quad & \mathbb{E}_{s\sim P}\Big[V^{\pi}(s)\Big] = \mathbb{E}_{s\sim \mathcal{D}}\Big[V^{\pi_{\mathcal{D}}}(s)\Big] \\ & \mathbb{E}_{s\sim P}\Big[V^{\pi}(s)\Big] - \mathbb{E}_{s\sim \mathcal{F}}\Big[V^{\pi_{\mathcal{F}}}(s)\Big] = z, \end{aligned} \tag{12}$$

where $\pi_{\mathcal{D}}$ and $\pi_{\mathcal{F}}$ indicate the expert and failed policies, respectively. Note that a parameter θ is introduced to guarantee a convex problem, and λ is a constant.

However, this newly added constraint makes the optimization problem ill-posed due to behavior overlapping issues in the demonstrations. Because behaviors are represented as a pair of states and actions, behavior overlapping issues can also be regarded as the overlapping of state-action pairs. During the training process, the agent optimizes the IRLFD problem from many demonstrations. As a result of this optimization, we can obtain a maximized reward function $R(s_t, a_t)$. Note that we are dealing with a reward function over the action and state space instead of a reward value of an entire demonstration R_τ. However, when feeding the expert and failed demonstrations, the entire demonstration is labeled as $l_\tau = l^{\mathcal{D}}$ if it is an expert demonstration and $l_\tau = l^{\mathcal{F}}$ if it is a failed demonstration. Such whole-demonstration labeling ignores the fact that not all state-action pairs in failed demonstrations exhibit failed behaviors.

For example, let $\mathcal{S}^{\mathcal{D}}$ and $\mathcal{A}^{\mathcal{D}}$ represent the state and action space of expert behaviors, and let $\mathcal{S}^{\mathcal{F}}$ and $\mathcal{A}^{\mathcal{F}}$ indicate the state and action space of failed behaviors. Note that the whole space consists of an expert space and a failed space. For example, when considering the state space, we have $\mathcal{S}^{\mathcal{D}} \cup \mathcal{S}^{\mathcal{F}} = \mathcal{S}$ and $\mathcal{S}^{\mathcal{D}} \cap \mathcal{S}^{\mathcal{F}} = \emptyset$; as well as the action space. For any state-action pair in failed demonstrations, sometimes $(s_{t_i}^{(\tau_{\mathcal{F}})}, a_{t_i}^{(\tau_{\mathcal{F}})}) \in (\mathcal{S}^{\mathcal{F}}, \mathcal{A}^{\mathcal{F}})$ when the behavior fails the task; on the other hand, sometimes $(s_{t_j}^{(\tau_{\mathcal{F}})}, a_{t_j}^{(\tau_{\mathcal{F}})}) \in (\mathcal{S}^{\mathcal{D}}, \mathcal{A}^{\mathcal{D}})$ when it shows almost the same behavior as in the expert demonstrations. Therefore, whole-demonstration labeling can introduce a conflict on whether to imitate the same behavior in different demonstrations.

When solving this optimization problem with the Lagrange multiplier method, we have two Lagrange multipliers for two constraints. $w^{\mathcal{D}}$ and $w^{\mathcal{F}}$ represent the weight parameters for the similarities of expert and failed demonstrations, respectively. The Lagrangian function becomes

$$\begin{aligned} L(\pi, z, \theta, w^{\mathcal{D}}, w^{\mathcal{F}}) &= H(\mathcal{A}^h||\mathcal{S}^h) + \theta z - \frac{\lambda}{2}||\theta||^2 \\ &+ w^{\mathcal{D}}\Big(\mathbb{E}_{s\sim P}\Big[V^{\pi}(s)\Big] - \mathbb{E}_{s\sim \mathcal{D}}\Big[V^{\pi_{\mathcal{D}}}(s)\Big]\Big) \\ &+ w^{\mathcal{F}}\Big(\mathbb{E}_{s\sim P}\Big[V^{\pi}(s)\Big] - \mathbb{E}_{s\sim \mathcal{F}}\Big[V^{\pi_{\mathcal{F}}}(s)\Big] - z\Big). \end{aligned} \tag{13}$$

Now, we differentiate the Lagrangian function regarding z

and θ:

$$\begin{aligned}\nabla_\theta L(\pi, z, \theta, w^{\mathcal{D}}, w^{\mathcal{F}}) &= z - \lambda\theta,\\ \nabla_z L(\pi, z, \theta, w^{\mathcal{D}}, w^{\mathcal{F}}) &= \theta - w^{\mathcal{F}}.\end{aligned} \tag{14}$$

Setting both derivatives to zero yields

$$\begin{aligned}z &= \lambda\theta,\\ \theta &= w^{\mathcal{F}}.\end{aligned} \tag{15}$$

Then, (15) is plugged back into the Lagrangian function. As a result, the previous Lagrangian multiplier update equation shown in (11) becomes

$$\begin{aligned}&\nabla_{w^{\mathcal{D}}} L(\pi, w^{\mathcal{D}}, w^{\mathcal{F}}) = \\ &\qquad \mathbb{E}_{s\sim P}\Big[V^{\pi}(s)\Big] - \mathbb{E}_{s\sim \mathcal{D}}\Big[V^{\pi_{\mathcal{D}}}(s)\Big],\\ &\nabla_{w^{\mathcal{F}}} L(\pi, w^{\mathcal{D}}, w^{\mathcal{F}}) = \\ &\qquad \mathbb{E}_{s\sim P}\Big[V^{\pi}(s)\Big] - \mathbb{E}_{s\sim \mathcal{F}}\Big[V^{\pi_{\mathcal{F}}}(s)\Big] - \lambda w^{\mathcal{F}},\end{aligned} \tag{16}$$

where the two Lagrangian multipliers $w^{\mathcal{D}}$ and $w^{\mathcal{F}}$ are updated via gradient descent. During the gradient-based training process, the expert gradient aims to imitate the expert demonstrations, and the failed gradient seeks not to imitate failed demonstrations. However, such overlapped state-action pairs may cause a conflict between the two gradients during training and reduce the performance significantly.

IV. Stable IRLFD

This work aims to overcome the state-action overlapping issue by introducing a novel well-posed constraint to the IRLFD optimization problem. In particular, we are inspired by IRLF [13], where a simple but powerful framework based on the max entropy IRL method is proposed.

A. Dissimilarity measurement with labels

This work introduces a sequence of labels to the dissimilarity measurement to deal with the state-action pair overlapping issue. In other words, instead of labeling the whole demonstration as l_τ, we label each state-action pair in the time series as l_t. The label indicates the possibility of a visited state-action pair exhibiting either expert or failed behavior. For example, $l_t = p\Big((s_t, a_t) \in (\mathcal{S}^{\mathcal{D}}, \mathcal{A}^{\mathcal{D}})\Big)$. The label for failed behavior is denoted as $l^{\mathcal{F}}$ for short; similarly, the label for expert behavior is denoted as $l^{\mathcal{D}}$. Note that this is a two-class label; therefore, the summation on the possibility of exhibiting expert or failed behavior is consistent $p\left(l_t = l^{\mathcal{D}}\right) + p\left(l_t = l^{\mathcal{F}}\right) = 1$. We then introduce time-series labels to the demonstrations so that the demonstration becomes a state-action-label sequence such as $\tau_i = \left\langle \left(s_1^{(\tau_i)}, a_1^{(\tau_i)}, l_1^{(\tau_i)}\right), \ldots, \left(s_{h_i}^{(\tau_i)}, a_{h_i}^{(\tau_i)}, l_{h_i}^{(\tau_i)}\right)\right\rangle$.

The introduced labels also influence the reward function such that the reward function for a state-action pair exhibiting expert behavior is high. The reward function is low for failed behavior. This intuition is represented as a reward function with an additional label variable, $R(s_t, a_t, l_t)$. A reward function with labels affects the state-value function in (1), which gives a state-label-value function:

$$V^{\pi}(s, l) = \mathbb{E}_{\pi}\left[\sum_{t=1}^{h} R(s_t, a_t, l_t)\middle| s_1 = s\right]. \tag{17}$$

Finally, we denote a marginal estimation of the state-label-value function over the states and labels sampled from demonstrations as $\mathbb{E}_{(s,l)\sim\mathcal{D}}\left[V^{\pi_{\mathcal{D}}}(s, l)\right]$. Thus, we can measure the dissimilarity of the learned reward to the failed reward as

$$\mathbb{E}_{(s,l)\sim P}\Big[V^{\pi}(s, l)\Big] - \mathbb{E}_{(s,l)\sim\mathcal{F}}\Big[V^{\pi_{\mathcal{F}}}(s, l)\Big]. \tag{18}$$

Such time-series labels deal with the state-action overlapping issue when measuring dissimilarity because they explicitly distinguish failed behavior from expert behavior at each timestep.

B. Optimization problem with labels

We introduce the proposed dissimilarity measurement with labels to the IRLFD optimization problem:

$$\begin{aligned}\max_{\pi, z, \theta}\quad & H(\mathcal{A}^h||\mathcal{S}^h) + \theta z - \frac{\lambda}{2}||\theta||^2\\ \text{s.t.}\quad & \mathbb{E}_{(s,l)\sim P}\Big[V^{\pi}(s, l)\Big] = \mathbb{E}_{(s,l)\sim\mathcal{D}}\Big[V^{\pi_{\mathcal{D}}}(s, l)\Big]\\ & \mathbb{E}_{(s,l)\sim P}\Big[V^{\pi}(s, l)\Big] - \mathbb{E}_{(s,l)\sim\mathcal{F}}\Big[V^{\pi_{\mathcal{F}}}(s, l)\Big] = z.\end{aligned} \tag{19}$$

Here, we model the driving behaviors with a modified MDP involving labels. Fig. 1 shows the underlying graphical model for an MDP with labels.

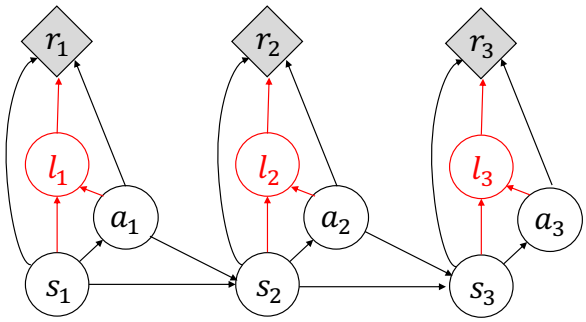


Fig. 1: Underlying graphical model for an MDP with labels. The nodes and edges in black are previously used in probabilistic inference. The nodes and edges in red are introduced in the proposed method. Here, $r_1 = R(s_1, a_1, l_1)$ indicates the reward at the first timestep.

We introduce time-series labels to the probability function of backward messages:

$$\begin{aligned}&\beta_t(s_t, a_t, l_t) = \\ &\qquad \exp\left(\sum_{t=t'}^{h} R(s_t, a_t, l_t) p(s_{t+1}|s_t, a_t)\right) p(l_t|s_t, a_t),\end{aligned} \tag{20}$$

where

$$\exp\left(\sum_{t=t'}^{h} R(s_t,a_t,l_t)p(s_{t+1}|s_t,a_t)\right) = \sum_{s_{t+1}\in\mathcal{S}} \beta_{t+1}(s_{t+1})p(s_{t+1}|s_t,a_t)\exp\Big(R(s_t,a_t,l_t)\Big). \tag{21}$$

Then, the state-marginal probability function gives

$$\beta_t(s_t,l_t) = \sum_{a_t\in\mathcal{A}} \beta_t(s_t,a_t,l_t)p(a_t|s_t)p(l_t|s_t,a_t). \tag{22}$$

We also represent the modified backward messages in log space which corresponds to the state-action and state-value functions:

$$\begin{aligned} Q(s_t,a_t,l_t) &= \log\beta_t(s_t,a_t,l_t), \\ V(s_t,l_t) &= \log\beta_t(s_t,l_t). \end{aligned} \tag{23}$$

Finally, the policy is obtained by

$$\pi(s_t,a_t,l_t) = \exp\Big(Q(s_t,a_t,l_t) - V(s_t,l_t)\Big). \tag{24}$$

By plugging this modified probability function with labels back into the optimization problem, the previous gradients in (16) become

$$\begin{aligned} &\nabla_{w^{\mathcal{D}}} L(\pi, w^{\mathcal{D}}, w^{\mathcal{F}}) = \\ &\quad \mathbb{E}_{(s,l)\sim P}\Big[V^{\pi}(s,l)\Big] - \mathbb{E}_{(s,l)\sim \mathcal{D}}\Big[V^{\pi_{\mathcal{D}}}(s,l)\Big], \\ &\nabla_{w^{\mathcal{F}}} L(\pi, w^{\mathcal{D}}, w^{\mathcal{F}}) = \\ &\quad \mathbb{E}_{(s,l)\sim P}\Big[V^{\pi}(s,l)\Big] - \mathbb{E}_{(s,l)\sim \mathcal{F}}\Big[V^{\pi_{\mathcal{F}}}(s,l)\Big] - \lambda w^{\mathcal{F}}. \end{aligned} \tag{25}$$

We now obtain two conflict-free gradients because the state-action pairs are explicitly split with the proposed labels. These conflict-free gradients represent the key solution to the instability issue of the IRLFD training process.

Algorithm 1 SIRLFD

Input: expert demonstrations $\mathcal{D}$, failed demonstrations $\mathcal{F}$

1: initialize $w^{\mathcal{D}}$ and $w^{\mathcal{F}}$ randomly
2: **repeat**
3: obtain the reward function $R(s_t,a_t,l_t)$ by (27)
4: obtain esitimated state-label-value funtion from expert demonstrations $\mathbb{E}_{(s,l)\sim\mathcal{D}}\left[V^{\pi_{\mathcal{D}}}(s,l)\right]$ by (17)
5: obtain esitimated state-label-value funtion from failed demonstrations $\mathbb{E}_{(s,l)\sim\mathcal{F}}\left[V^{\pi_{\mathcal{F}}}(s,l)\right]$ by (17)
6: optimize the policy π by (20-24), Fig. 1
7: obtain esitimated state-label-value funtion from learned distribution $\mathbb{E}_{(s,l)\sim P}\left[V^{\pi}(s,l)\right]$ by (17)
8: update weight parameters $w^{\mathcal{D}}, w^{\mathcal{F}}$ by (25-26)
9: **until** convergence

Output: $R(s_t,a_t,l_t)$

Algorithm 1 summarizes the method proposed in procedures. First, we initialize the weights randomly (line 1). Notably, the initialization on $w^{\mathcal{F}}$ is trivial if we set the gradient of $\nabla_{w^{\mathcal{F}}} L(\pi, w^{\mathcal{D}}, w^{\mathcal{F}})$ in (25) at zeros, thus yielding the following:

$$w^{\mathcal{F}} = \frac{1}{\lambda}\left(\mathbb{E}_{(s,l)\sim P}\Big[V^{\pi}(s,l)\Big] - \mathbb{E}_{(s,l)\sim\mathcal{F}}\Big[V^{\pi_{\mathcal{F}}}(s,l)\Big]\right). \tag{26}$$

In other words, the optimal $w^{\mathcal{D}}$ is updated incrementally using gradient descent until convergence; meanwhile, the optimal solution of $w^{\mathcal{F}}$ is found analytically. Inside the main loop, we first compute the reward function (line 3) and then compute the expected future rewards by giving the dataset of expert and failed demonstrations (lines 4-5). Notably, a reward function can be represented in multiple ways and a straightforward way is used in this work, as follows:

$$R(s_t,a_t,l_t) = (w^{\mathcal{D}} + w^{\mathcal{F}})\phi(s_t,a_t)l_t, \tag{27}$$

where $\phi(s_t,a_t)$ is a predefined feature function depending on the scenarios. With the reward function, the policy can be obtained via probabilistic inference (line 6), such that the learned expected future rewards can be calculated (line 7). Finally, using the rewards from demonstrations and learned rewards, two weight vectors are updated (line 8). The main loop ends until convergence and the learned reward function is obtained.

V. Experimental Results

The proposed method is evaluated in two driving scenarios. Experiments are conducted to qualitatively evaluate the stability of gradients in the training process and quantitatively evaluate the performance regarding the similarities.

A. Driving scenarios

We use two driving scenarios in this experiment.

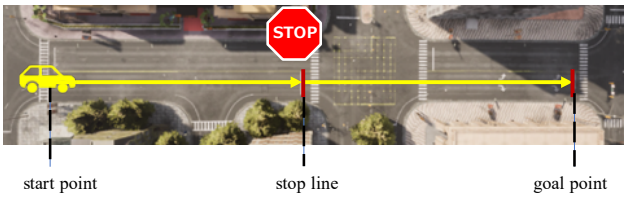


Fig. 2: Stop line scenario on CARLA simulator.

1) Stop line scenario: The first scenario is a stopping-at-stop line task where the scenario design is shown in Fig. 2. There is a stop traffic sign in the middle of the road. The ego vehicle should start from the start point with an initial velocity of 0 km/h, stop at the stop line, and then reach the goal point. The state space contains information on the discrete velocity v and discrete location y of the ego vehicle. Let the location of the stop line be y_s. The expert demonstrations show a stop behavior at the stop line, such as $\exists s_t \in \tau_{\mathcal{D}}, (y_t, v_t) = (y_s, 0)$. On the other hand, the failed demonstrations do not show any stop behavior at the stop line, such as $\forall s_t \in \tau_{\mathcal{F}}, (y_t, v_t) \neq (y_s, 0)$. In this scenario, the feature function is designed as a weighted average of three factors, indicating a stop behavior at the stop line, a

high-velocity range near the stop line, and a stop behavior at the goal point, respectively.

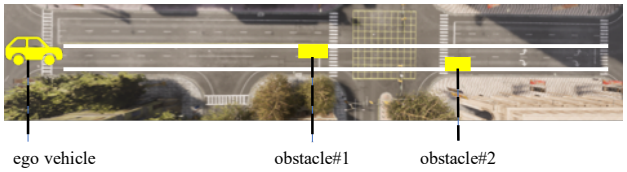


Fig. 3: Scenario of avoiding obstacles on CARLA simulator.

2) Avoiding obstacle scenario: The second scenario involves avoiding two obstacles with the lane-change task; the scenario design is shown in Fig. 3. There are two obstacles on a two-lane road and the ego vehicle should move from left to right while avoiding obstacles and switching to the correct lane. The state space contains the information on the location's x and y coordinates in a three-dimensional map. Let the locations of the obstacles, (x^o, y^o), be in a range of $(x^o, y^o) \in (\mathcal{X}^o, \mathcal{Y}^o)$. The expert demonstrations reveal the behaviors of avoiding all these obstacles, such as $\forall s_t \in \tau_{\mathcal{D}}, (x_t, y_t) \notin (\mathcal{X}^o, \mathcal{Y}^o)$. On the other hand, the failed demonstrations show the behaviors of colliding with the obstacle, such as $\exists s_t \in \tau_{\mathcal{F}}, (x_t, y_t) \in (\mathcal{X}^o, \mathcal{Y}^o)$. In this scenario, the feature function deals with five factors: the first two factors represent the boundaries outside the two-lane road; one factor indicates the boundary between two lanes; and the final two factors represent the locations of two obstacles.

(a) Nonstop at the stop line. (b) Collide to the obstacle.

Fig. 4: Failed demonstrations on CARLA simulator in two scenarios.

B. Data collection

The driving behavior data is collected using the CARLA simulator [3]. Participants were first tasked with a training phase to familiarize themselves with virtual driving in a simulation environment using driving handle controllers, including a steering wheel and pedals. The simulated driving is conducted in a driver's first-person view. For both driving scenarios, 20 expert demonstrations and 20 failed demonstrations are collected. Fig. 4 shows a failed demonstration in both driving scenarios. In the stop line scenario, the failed demonstrations pass the stop line with high velocity. In the avoiding obstacle scenario, the failed demonstrations collide with the obstacle (a stopped vehicle) on the road.

C. Comparison methods

In this work, we evaluate the proposed SIRLFD method by comparing it against the following state-of-the-art IRL methods:

1) max entropy IRL: The max entropy IRL [18] method is a probabilistic approach considering uncertainty in the reward function. This method also provides a well-defined distribution over the demonstrations by matching the feature expectations. We implement the constraint of feature expectations by estimating the state-visiting frequencies. Notably, hereafter, max entropy IRL is simply referred to as IRL.

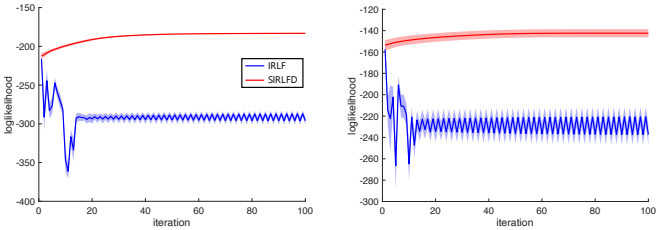
2) IRLF: IRLF [13] introduces an additional constraint to the problem of optimizing the max entropy IRL method. It restricts the conditions of IRL such that IRLF can also learn from the failed demonstrations by not imitating the failed behaviors. This additional constraint is also implemented by the state-visiting frequencies.

D. Evaluation metric

We use two types of likelihoods to evaluate the performance on the similarity between the two types of demonstrations and the recovered trajectories based on the learned reward function. The positive likelihood measures the similarity of the expert demonstrations to the recovered trajectories, such as $L_{\mathcal{D}} = \sum_{\tau_{\mathcal{D}} \in \mathcal{D}} \log p(\tau | w^{\mathcal{D}}, w^{\mathcal{F}}) - \log p(\tau_{\mathcal{D}})$. On the other hand, the negative likelihood measures the similarity of the failed demonstrations to the recovered trajectories, such as $L_{\mathcal{F}} = \sum_{\tau_{\mathcal{F}} \in \mathcal{F}} \log p(\tau | w^{\mathcal{D}}, w^{\mathcal{F}}) - \log p(\tau_{\mathcal{F}})$. Notably, both positive and negative likelihood metrics are reported on a log scale. Furthermore, we use the difference between the positive and negative likelihoods as a performance metric, such as $L = L_{\mathcal{D}} - L_{\mathcal{F}}$, which is used to evaluate how well the model could imitate the expert demonstrations without imitating the failed ones. In other words, it is used to measure the ability to distinguish failed demonstrations from expert demonstrations; hence, the higher the score on this metric, the better is the performance.

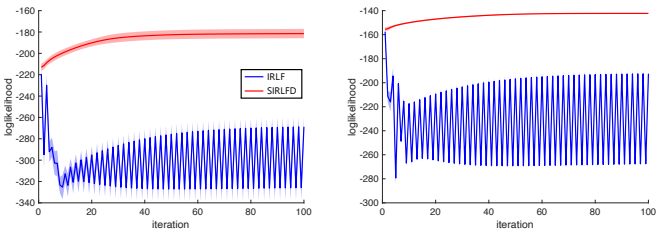
E. Results

First, we qualitatively show the gradient improvement by introducing the overlapping state-action pair issue. Fig. 5 and Fig. 6 show the results of our experiments on the data of the stop line scenario with the data ratio of $|\mathcal{D}| : |\mathcal{F}| = 2 : 1$ and $1 : 2$, respectively. Fig. 5b and Fig. 6b specifically show the negative likelihood. The blue curves representing the results of the IRLF method fluctuate significantly during the training process. The instability in the training process was caused by the overlapping state-action pair issues in the expert and failed demonstrations. Meanwhile, Fig. 5a and Fig. 6a show the positive likelihood. In this case, the fluctuation is less significant than that in the negative likelihood. The fluctuation can be attributed to the overlapped state-action pairs directly influencing the negative gradients, while the positive gradient is influenced indirectly through the learned reward. In addition, both positive likelihood and negative likelihood fluctuate more when the ratio of $|\mathcal{D}| : |\mathcal{F}|$ becomes



(a) Likelihood of $L_{\mathcal{D}}$. (b) Likelihood of $L_{\mathcal{F}}$.

Fig. 5: Likelihood over 100 iterations, for 10 runs with the data ratio of $|\mathcal{D}| : |\mathcal{F}| = 2 : 1$ in the stop line scenario.



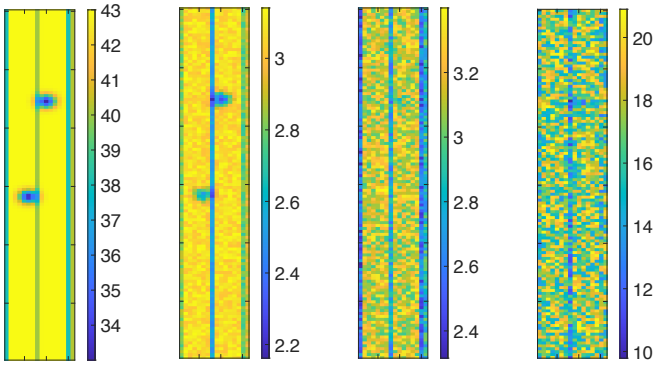
(a) Likelihood of $L_{\mathcal{D}}$. (b) Likelihood of $L_{\mathcal{F}}$.

Fig. 6: Likelihood over 100 iterations, for 10 runs with the data ratio of $|\mathcal{D}| : |\mathcal{F}| = 1 : 2$ in the stop line scenario.

smaller. In other words, the more overlapped the state-action pairs in the data, the worse the performance. The red curves indicate the results obtained by our proposed method, which introduces a time-series labeling mechanism to split the overlapped state-action pairs. This is a compelling solution in that the likelihood could monotonically increase and eventually converge to a higher score.

We then qualitatively show the learned rewards to analyze the characteristics of different methods. Fig. 7 plots the original and learned reward functions for SIRLFD, IRL, and IRLF in the avoiding obstacle scenario. In the original reward function, as shown in Fig. 7a, the reward was low at the locations of two obstacles and the boundaries of two lanes. Fig. 7b shows the learned reward function by our proposed SIRLFD method, which could adequately recover the original reward function by recognizing the obstacles. By contrast, the learned reward function by the IRL method, as shown in Fig. 7c, could only achieve a low reward around the boundaries of lanes but could not distinguish the obstacles on the road. This result proves the importance of an additional constraint that does not aim to imitate failed demonstrations. However, despite the IRLF method including such a constraint in the optimization problem, it still could not distinguish the obstacles as shown in Fig. 7d. This instability could be attributed to the constraint being ill-posed and influenced by the state-action pair overlapping issue significantly such that even with the constraint, the learned reward by IRLF is worse than the learned reward by IRL. Therefore, the state-action pair overlapping issue needs to be solved with a well-posed constraint.

We quantitatively evaluated the performance across dif-



(a) Original (b) SIRLFD (c) IRL (d) IRLF

Fig. 7: Original reward and learned reward functions for the three methods in the avoiding obstacle scenario. Rewards are represented in heat maps. Specifically, two blue circles in the heat map indicate low rewards around two obstacles.

ferent ratios of $|\mathcal{D}| : |\mathcal{F}|$ in both the stop line scenario and avoiding obstacle scenario shown in Table I and Table II, respectively. The mean and standard deviation of the performance were evaluated after 100 iterations for 10 runs. We represent the performance of the methods under different data ratios of $|\mathcal{D}| : |\mathcal{F}|$ while keeping the total size of the full demonstration set $|\mathcal{D} \cup \mathcal{F}|$ fixed. The tables show that the proposed SIRLFD method performed better than IRL and IRLF. Note that IRLF performed poorly compared to IRL in both scenarios because, in both designed driving scenarios, the negative state space set $\mathcal{S}^{\mathcal{F}}$ was a tiny portion of the whole state space $\mathcal{S}$. In other words, the overlapped state-action pairs in both scenarios were significantly large. This result shows that although IRLF has a specific constraint to not imitate failed demonstrations, the performance could have further degraded owing to such an ill-posed problem. Therefore, our proposed method makes it a well-posed problem with the proposed time-series labeling, thereby improving performance significantly.

	2:1	1:1	1:2
IRL	-44.90 ± 7.50	-43.65 ± 6.18	-44.29 ± 5.63
IRLF	-70.82 ± 7.31	-56.87 ± 9.65	-62.91 ± 2.69
SIRLFD	$\mathbf{-43.05 \pm 4.64}$	$\mathbf{-37.43 \pm 6.09}$	$\mathbf{-39.96 \pm 5.48}$

TABLE I: Quantitative evaluation of performance, L, across different ratios of $|\mathcal{D}| : |\mathcal{F}|$ in the stop line scenario.

	2:1	1:1	1:2
IRL	-247.88 ± 7.59	-221.80 ± 6.19	-244.98 ± 8.06
IRLF	-474.20 ± 55.37	-442.32 ± 34.78	-600.84 ± 64.42
SIRLFD	$\mathbf{-214.75 \pm 10.37}$	$\mathbf{-216.67 \pm 13.38}$	$\mathbf{-200.13 \pm 9.41}$

TABLE II: Quantitative evaluation of performance, L, across different ratios of $|\mathcal{D}| : |\mathcal{F}|$ in the avoiding obstacle scenario.

We also quantitatively evaluated performance across dif-

	20(1)	20(4)	20(8)
IRL	-41.34 ± 4.50	-45.91 ± 4.70	-45.78 ± 8.89
IRLF	-47.54 ± 11.71	-55.38 ± 9.38	-83.03 ± 15.18
SIRLFD	$\mathbf{-36.99 \pm 6.12}$	$\mathbf{-42.54 \pm 7.06}$	$\mathbf{-40.75 \pm 6.81}$

TABLE III: Quantitative evaluation of performance, L, across different number of $|\mathcal{F}|$ mixed into expert demonstrations in the stop line scenario. The column name indicates the number of expert demonstrations mixed with failed demonstrations $|\mathcal{D}|(|\mathcal{F}|)$.

	20(1)	20(4)	20(8)
IRL	-235.27 ± 17.03	-252.55 ± 9.40	-228.14 ± 8.54
IRLF	-451.98 ± 58.11	-383.65 ± 21.93	-510.37 ± 61.84
SIRLFD	$\mathbf{-207.14 \pm 12.26}$	$\mathbf{-209.52 \pm 11.33}$	$\mathbf{-207.24 \pm 11.23}$

TABLE IV: Quantitative evaluation of performance, L, across different number of $|\mathcal{F}|$ mixed into expert demonstrations in the avoiding obstacle scenario.

ferent numbers of failed demonstrations mixed into the expert demonstrations. This experiment aimed to evaluate the extent to which dataset quality would influence performance because collecting high-quality driving data is highly costly in the real world. By contrast, normal-quality driving data are readily available, such as the daily driving records of normal drivers. We simulated data quality as an expert dataset mixed with some failed demonstrations. In this case, the more mixed failed demonstrations in the expert demonstrations, the worse the quality of the dataset. Table III and Table IV show the results of the stop line scenario and avoiding obstacle scenario, respectively. Our method generally showed the best performance for various data qualities. This result indicates that our method is robust to data quality.

VI. Conclusion

IRL has recently become one of the most prominent approaches for modeling driving behaviors from driving demonstrations. As expert demonstrations cannot help predict driving behaviors to avoid dangers on purpose, failed demonstrations are also introduced to the traditional IRL framework. However, owing to the over-constrained optimization problem, when a large overlapping exists on state-action pairs, the performance reduces significantly. To overcome this limitation, this work proposed a stable IRLFD method that formalizes a novel constraint with time-series labels. Despite its simplicity, the proposed method is more effective and stable. Experimental results on simulated stop line and avoiding obstacle scenarios showed that the proposed method performed better in terms of stability and ability to distinguish expert from failed behaviors. In the future, we plan to develop an automatic label generation algorithm to reduce the labeling cost.

Acknowledgements

This work was supported by JST Grant Number JPMJSP2106, JPMJFS2112, and JSPS KAKENHI Grant Numbers JP21H03517, JP23H00214, JP24K03015.

References

[1] S. Adams, T. Cody, and P. A. Beling, "A survey of inverse reinforcement learning," *Artificial Intelligence Review*, 2022.

[2] D. Coelho and M. Oliveira, "A review of end-to-end autonomous driving in urban environments," *IEEE Access*, 2022.

[3] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," in *Proc. of CoRL*, 2017.

[4] S. Hosoma, M. Sugasaki, H. Arie, and M. Shimosaka, "RRT-based maximum entropy inverse reinforcement learning for robust and efficient driving behavior prediction," in *Proc. of IV*, 2022.

[5] G. Lee, D. Kim, W. Oh, K. Lee, and S. Oh, "MixGAIL: Autonomous driving using demonstrations with mixed qualities," in *Proc. of IROS*, 2020.

[6] K. Lee, S. Choi, and S. Oh, "Inverse reinforcement learning with leveraged gaussian processes," in *Proc. of IROS*, 2016.

[7] S. Levine, "Reinforcement learning and control as probabilistic inference: Tutorial and review," in *arXiv preprint arXiv:1805.00909*, 2018.

[8] N. M. Negash and J. Yang, "Driver behavior modeling toward autonomous vehicles: Comprehensive review," *IEEE Access*, 2023.

[9] A. Y. Ng and S. Russell, "Algorithms for inverse reinforcement learning," in *Proc. of ICML*, 2000.

[10] K. Nishi and M. Shimosaka, "Fine-grained driving behavior prediction via context-aware multi-task inverse reinforcement learning," in *Proc. of ICRA*, 2020.

[11] D. Ramachandran and E. Amir, "Bayesian inverse reinforcement learning," in *IJCAI*, 2007.

[12] L. Shi, L. Wang, C. Long, S. Zhou, M. Zhou, Z. Niu, and G. Hua, "SGCN: Sparse graph convolution network for pedestrian trajectory prediction," in *Proc. of CVPR*, 2021.

[13] K. Shiarlis, J. Messias, and S. Whiteson, "Inverse reinforcement learning from failure," in *Proc. of AAMAS*, 2016.

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

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

[16] X. Xie, C. Li, C. Zhang, Y. Zhu, and S.-C. Zhu, "Learning virtual grasp with failed demonstrations via bayesian inverse reinforcement learning," in *Proc. of IROS*, 2019.

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

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