Hindawi
Journal of Robotics
Volume 2020, Article ID 3849309, 18 pages
https://doi.org/10.1155/2020/3849309

Research Article

Learning from Demonstrations and Human Evaluative Feedbacks: Handling Sparsity and Imperfection Using Inverse Reinforcement Learning Approach

Nafee Mourad,[1] **Ali Ezzeddine**,[2] **Babak Nadjar Araabi**,[2] **and Majid Nili Ahmadabadi**[1]

[1]*Cognitive Systems Laboratory, School of ECE, College of Engineering, University of Tehran, Tehran, Iran*
[2]*Machine Learning and Computational Modeling Laboratory, School of ECE, College of Engineering, University of Tehran, Tehran, Iran*

Correspondence should be addressed to Nafee Mourad; n.mourad@ut.ac.ir

Received 16 August 2019; Revised 12 November 2019; Accepted 12 December 2019; Published 13 January 2020

Academic Editor: Yangmin Li

Programming by demonstrations is one of the most efficient methods for knowledge transfer to develop advanced learning systems, provided that teachers deliver abundant and correct demonstrations, and learners correctly perceive them. Nevertheless, demonstrations are *sparse* and *inaccurate* in almost all real-world problems. Complementary information is needed to compensate these shortcomings of demonstrations. In this paper, we target programming by a combination of *nonoptimal* and *sparse* demonstrations and a limited number of binary evaluative feedbacks, where the learner uses its own evaluated experiences as new demonstrations in an extended inverse reinforcement learning method. This provides the learner with a broader generalization and less regret as well as robustness in face of sparsity and nonoptimality in demonstrations and feedbacks. Our method alleviates the unrealistic burden on teachers to provide optimal and abundant demonstrations. Employing an evaluative feedback, which is easy for teachers to deliver, provides the opportunity to correct the learner's behavior in an interactive social setting without requiring teachers to know and use their own accurate reward function. Here, we enhance the inverse reinforcement learning (IRL) to estimate the reward function using a mixture of nonoptimal and sparse demonstrations and evaluative feedbacks. Our method, called IRL from demonstration and human's critique (IRLDC), has two phases. The teacher first provides some demonstrations for the learner to initialize its policy. Next, the learner interacts with the environment and the teacher provides binary evaluative feedbacks. Taking into account possible inconsistencies and mistakes in issuing and receiving feedbacks, the learner revises the estimated reward function by solving a single optimization problem. The IRLDC is devised to handle errors and sparsities in demonstrations and feedbacks and can generalize different combinations of these two sources expertise. We apply our method to three domains: a simulated navigation task, a simulated car driving problem with human interactions, and a navigation experiment of a mobile robot. The results indicate that the IRLDC significantly enhances the learning process where the standard IRL methods fail and learning from feedbacks (LfF) methods has a high regret. Also, the IRLDC works well at different levels of sparsity and optimality of the teacher's demonstrations and feedbacks, where other state-of-the-art methods fail.

1. Introduction

The next generation of technologies focuses on the capabilities of artificial intelligent agents to become an integral part of our daily lives. To reach that goal, artificial agents, instead of being preprogrammed, need to be equipped with efficient learning systems to rapidly adapt to novel, dynamic, and complex situations. On top of that, the agents should have the flexibility to be personalized to user preferences, that is, to learn styles and behaviors that their human users prefer and enjoy. Therefore, considering vast individual differences among human beings, in terms of both preferences and technical expertise, the learning systems should be able to learn from nontechnical users with minimum burden on them. A significant body of research has targeted solving this problem, especially by using Learning from Demonstrations (LfD), where the learner agent derives its policy by observing its teacher's demonstrations, and Learning from Feedbacks (LfF), where the teacher provides critiques to indicate the desirability of the learner's actions (see Table 1).

Table 1: Strengths and weaknesses of LfD and LfF approaches.

	Learning from evaluative feedbacks (LfF)	Learning from demonstrations (LfD)
Strengths	(i) Simplicity, where "teachers" teach "learners" without needing detailed knowledge of how to perform the task themselves. It is enough for the teacher to evaluate the outcome (ii) Feedback is not affected by correspondence problem between the learner and the teacher (the physical differences)	(i) Effective transfer learning techniques, where the learner generalizes from teacher's demonstration to state-action mapping in whole space (ii) Speeds up the learning process and reduces the regret, because the teacher is providing the correct action directly (iii) Decreases the learner exploration to get the correct action
Weaknesses	(i) Learning process is slow and needs a large number of teacher's feedbacks. So, it is a boring job (ii) The learner behaves randomly at early learning trials (iii) Inconsistency and errors during providing feedback	(i) The teacher must have a clear policy in her mind to provide demonstrations and must be an expert in doing the task (ii) Demonstrations cannot be easily available in some situations due to danger or low possibility of occurrence (iii) Different physical embodiment and perception between the teacher and the learner (correspondence problem)

In LfD, also known as imitation learning, the learner generalizes the teacher's demonstrations to derive its policy. There exist two major approaches in the LfD framework based on the way that the learner deals with these demonstrations. The first one is the direct approach, termed as behavioral cloning, where the goal is to learn the mapping between states and actions (i.e., policy) in the teacher's demonstrations using a supervised learning technique. This approach suffers from several problems including cascading error issue [1] and being sensitive to the dynamic model of environment [2]. The other approach is known as apprenticeship learning [3] and is usually casted as an inverse reinforcement learning (IRL) problem [4]. In this approach, the policy is derived indirectly by estimating the reward function underlying the teacher's demonstrations, and then a planning algorithm [5] is employed to derive the policy that maximizes the estimated reward function. This approach overcomes the challenges that the preceding one faces [2, 6]. In addition, in this approach, the learner agent not only replicates the observed behavior, but also infers the "reason" behind it [7] and generalizes the demonstrations accordingly. As a result, the learning process becomes transferable, robust to changes in the configuration of the agent and the environment [8–10]. In this paper, we focus on this approach, mainly on the IRL problem. We should note that the IRL is usually used to accomplish two objectives: apprenticeship learning and reward learning, where in the latter gaining the knowledge of reward function is a goal by itself [10, 11].

Most existing works on the IRL assume that (1) the teacher's demonstrations are reliable; i.e., demonstrations are optimal or near-optimal, (2) the teacher's demonstrations are abundant and sufficiently available, and (3) samples of the teacher's policy are provided by demonstrations. In practice, several reasons could be thought of for these assumptions not to hold, which imposes severe limitations on the applicability of IRL in the real world. These reasons include teachers' inability to perform the task optimally, insufficiency and nondiversity of demonstrations due to the dangers for teachers and the burden on them, and poor correspondence between teachers and learners. Moreover, teachers prefer to express their intentions and preferences in multiple modalities rather than just by demonstrations. Consequently, these limitations highly restrict the generalization capability of the standard IRL methods, which leads to poor performance of the learner. Some methods in the literature partially address these nonoptimality and sparsity issues, see Section 2 for details, but do not take into consideration that the nonoptimality may exist in all demonstrations and its amount may be significant rather than being only a noise. Other works tackle these issues by adding another source of information, in addition to teacher's demonstrations, to the learning process. The most recent state-of-the-art works employ reinforcement learning (RL) along with demonstrations [12–14]. These methods require a predefined environmental reward function that should be consistent with the teacher's demonstrations. This somehow necessitates knowing the teacher's reward function a priori, which is not practical in complex situations. Another recent work in this area is our previous method [15], which adds evaluative human feedback information (i.e., right/wrong instructions) to solve the nonoptimality problem in demonstrations. Providing evaluative feedbacks is extremely simpler than constructing the reward function required in RL methods. Nevertheless, Ezzeddine's study [15] uses evaluative feedbacks solely to correct mistakes in the teacher's demonstrations and cannot handle sparsity in demonstrations. A side effect of this limitation is that it decreases the robustness against errors in evaluative feedbacks. In this paper, we successfully handle both sparsity and nonoptimality in demonstrations and evaluative feedbacks. We employ negative evaluative feedbacks to boost alternative actions and employ the learner's own experiences along with the teacher's demonstrations to improve solving IRL problem. This results in faster learning and higher robustness against sparsity and nonoptimality levels.

Motivated by the challenges stated above and in order to leverage learning from humans, we propose a practical approach, called IRLDC, that blends both teacher's task demonstrations and her binary evaluative feedbacks (true/false) into a unified IRL framework. In the presented method, the learning process is done within two phases. In the first phase, the learner acquires its initial skills from the teacher's demonstrations. In the second phase, the learner interacts with the environment and receives binary evaluative feedbacks from the teacher. Here, by taking into account the natural inconsistencies and errors in the teacher's feedbacks, we propose a feedback model coding how the teacher's feedbacks are provided. In addition, the learner takes its own evaluated experiences as new demonstrations. Using these feedbacks and demonstrations, an enhanced version of the IRL is employed to estimate the reward function and the learner policy is revised by using the dynamic programming [16]. The cycle of interact-feedback-update continues until the teacher is satisfied. In summary, the proposed framework contributes in three ways to boost the robustness and the speed of learning:

(i) Developing an IRL framework, which deals with both teacher's demonstrations and evaluative feedbacks at different levels of sparsity, optimality, and inconsistency. This framework, unlike those restricted to demonstrations, is also capable of operating in extreme cases where only erroneous and inconsistent feedback data are available.

(ii) Deriving the teacher's preference model from the noisy and inconsistent feedback data provided by the teacher. For that, we employ a feedback model that incorporates recent and old observations to implicitly handle inconsistencies in providing feedback in addition to handling errors.

(iii) Presenting a new IRL objective function that combines demonstrations and feedbacks as a single optimization problem and allows the teacher's preference model to affect the optimization process when searching for the reward function. In our objective function, the algorithm learns from the incorrect data instead of filtering them out.

The approach presented in this paper can bring notable benefits and possibilities: (1) it can effectively treat the nonoptimality and sparsity in demonstrations and feedbacks; (2) it allows the teacher to express his/her intention and style for solving the task by using two instructive modalities, i.e., demonstrations and evaluative feedbacks; (3) it exploits the complementary and teacher depended on expertise embedded in demonstrations and feedbacks [17, 18] (see Table 1); (4) it is possible to teach the learner by only feedbacks, if needed; (5) being an incremental learning method, the teacher can provide demonstrations at one time or place and provide feedbacks at another; and (6) it is possible to provide demonstrations by one teacher and feedbacks by another.

The rest of this paper is organized as follows: Section 2 discusses and reviews the related works. In Section 4, our IRLDC framework is introduced and formalized. The experimental setup and the results are reported and discussed in Section 5. Finally, Section 6 draws conclusions and discusses future research directions.

2. Related Work

In this section, we describe the closest works to ours, scrutinizing the way they have dealt with nonoptimal and sparse demonstrations in the IRL setting, and how humans can teach learning agents using both modalities, i.e., demonstrations and evaluative feedbacks.

2.1. Inverse Reinforcement Learning. As previously discussed, the LfD is comprised of two main learning trends: imitation learning (direct approach) and IRL (indirect approach) (see [19, 20]). In the IRL category, there are many approaches that differ in their algorithmic view [8, 17], the objective function they optimize [11, 18, 21, 22], and the challenges they try to solve in the IRL [23–27]. Most of the existing works in this framework assume that demonstrations are abundant and their quality is optimal, which is rarely the case in reality. On the other hand, there are also some methods that slightly relax these assumptions. Bayesian IRL approaches [11, 28, 29] give way to slight deviations from the optimal demonstration assumption, due to the probabilistic nature of the Bayesian approach and the inclusion of teacher's model. Authors in [21] suppose that the suboptimality in demonstrations can occur at a small scale, and they handle this suboptimality by smoothing the constraints of the object function. In [25], it is assumed that demonstrations are locally optimal, but due to this assumption, this work cannot benefit from globally optimal demonstrations, in case they are available. In [30–32], the problem of nonoptimality is handled by using a generative model to learn the optimal demonstrations from a large number of suboptimal ones. Authors in [33] suppose that demonstrations are abundant but noisy, and they pretreat this limited suboptimality by a maximum a posteriori estimation to reconstruct near-optimal demonstrations. In [10], it is assumed that a sparse noise exists in some trajectories in demonstrations, and a model is used to identify and separate noisy trajectories from the reliable ones. Unlike [10], in [34], the authors do not filter out the noisy trajectories; instead, they learn from them, provided that some successful demonstrations are available, which is not always a realistic assumption. All the aforementioned methods cannot deal with real-world cases where demonstrations are sparse and far from optimal (more than noise), and also, in cases where nonoptimality exists in all demonstrations, whereas in this paper, we target learning in such conditions by extending IRL to incorporate teacher's demonstrations and her binary evaluative feedbacks.

2.2. Learning from Evaluative Feedbacks. Learning from feedbacks is another direction for teaching an agent. Out of the different forms that feedbacks can take, here we only focus on binary evaluative feedbacks. Among the large body

of literature on this subject, there are some works that provide an evaluative feedback for each entire trajectory executed by learners (see [35–37]). Using this type of feedback, the majority of works target direct derivation of the optimal policy. On the other hand, in other works, an evaluative feedback is provided for each action and is used either to communicate a numeric reward [38–41] or to transfer the performed action's correctness (true/false) in order to derive the optimal policy. The latter type is used for policy shaping [42, 43], while RL methods [16] are mainly used for policy improvement in the former case. Many recent works emphasize on the effectiveness of policy shaping in comparison with using evaluative feedbacks as a numeric reward function (see [44–46]). Nevertheless, these learning methods are sensitive to nonoptimality of feedbacks. A way to handle this nonoptimality is by employing probabilistic feedback models to deal with errors and sparsities in teacher's feedbacks (see [42, 45, 47]). In our work, human teacher's feedbacks are considered to be binary and evaluative and are provided for each action. We suggest a novel nonprobabilistic feedback model that depends on recent and old observations to handle natural and unavoidable inconsistencies and errors in human's feedbacks. Unlike former approaches, our model implicitly can handle the inconsistency in feedbacks. Contrary to most of the works that directly derive a policy or modify it by using feedbacks, we employ a revised version of the IRL to estimate the teacher's reward function and, hence, generalize the experience of sparse interactions with the teacher to the entire task space.

2.3. Combining Human Demonstrations and Reinforcement Learning. In a more realistic approach, to deal with nonoptimal and sparse demonstrations, most of the state-of-the-art methods combine human demonstrations with the experience of interacting with the environment using reinforcement learning (RL) which requires a critic knowing the reward function [12–14, 48]. Human demonstrations can be used to initiate a policy and then refining it using RL (see Section 5.1 of [20] for a survey). This approach is appealing and results in a good learner performance. However, to learn an acceptable policy, such an approach suffers from the curse of dimensionality and high regret especially in sparse and nonoptimal demonstrations. In addition, this approach does not express the teacher's preferences well and needs to design the environmental reward function consistent with the mentor's behavior.

Our work differs from this approach, where we focus on leveraging learning from human data by combining her sparse as well as nonoptimal demonstrations and error-prone correct/wrong evaluative feedbacks. The human evaluative feedback is different in its nature from the environmental reinforcement signal (see [39, 49]). In addition, the goal of this approach is to derive the optimal policy directly, whereas our work follows IRL approach to derive the reward function underlying the task, which results in less regret due to the inherent generalization capability of IRL.

2.4. Combining Human Demonstrations and Feedbacks. Different combinations of human demonstrations and feedbacks are used in the literature to accelerate and enhance the learning process or to allow teachers to provide information using different modalities. Human feedbacks that are combined with the teacher's demonstrations mainly take the following forms: corrective action, advice preferences, and evaluation of performed actions (evaluative feedback). Corrective action feedback is used in interactive learning systems [50] and in the active learning setting [8, 17, 51]; for providing this kind of feedback, the teacher should be able to provide an optimal action, which is not available in most realistic cases. Advice preference feedback is a kind of prior knowledge for solving the task [52, 53] and is usually combined with other types of learning. Since advice preference feedback is provided by domain experts, its use is restricted to those cases wherein experts are available. In human evaluative feedback, or critique, the human teacher provides evaluative critiques to indicate the desirability of the performed action. This kind of feedback is simple and requires minimal information from the teacher.

In this work, we use binary evaluative feedbacks along with demonstrations. A limited number of works have been done within this setting [54–56]. The most close approach to ours, in terms of human information and feedbacks, is the work of [55, 56]. However, the work of [55] employs a supervised learning method while we use the IRL and extract human preferences. In addition, nonoptimality or sparsity of human demonstrations as well as erroneous feedbacks is not considered in [55, 56]. Recently, a new published paper from our group managed to treat the nonoptimality in demonstrations (within a certain limit) in the presence of feedbacks and abundant demonstrations [15]. But it failed to handle sparsity and high levels of nonoptimality in demonstrations. Also, the feedback error which can be dealt with was very limited. In addition, the nonoptimality and inconsistencies in data were filtered out instead of learning from them. All these issues are successfully handled in this paper.

3. Problem Formulation

The underlying decision-making process of an IRL agent learning from human demonstrations is modeled by a Markov decision process (MDP) without a reward function (MDP/R). MDP/R is a 4-tuple (S, A, T, γ) where S is a set of states in the environment and A is a set of actions available to the learner. Moreover, the transition model T: $S \times A \times S \longrightarrow [0, 1]$ denotes the transitioning probabilities between states; $T(s, a, s') = \Pr(s_{t+1} = s' \mid a_t = a, s_t = s)$ where s is the current state, a is the performed action, and s' is the next state. In our case, this model is preknown. This assumption is realistic in many cases, like when we learn a new task or a novel style in a familiar environment. Furthermore, $\gamma \in [0, 1]$ is a discount factor.

The aim of an IRL problem is to extract the reward function R: $S \times A \longrightarrow \mathbb{R}$, which assigns a real-valued reward for executing action a in state s. Usually, the number of states

is too large. Therefore, for the reward function to admit a practical representation and to allow recovering it from fewer number of demonstrations, the reward is represented as a function of state-action's features; $R(s,a) = f(\varphi(s,a))$, where φ: $S \times A \longrightarrow \mathbb{R}^m$ is a known m-dimensional state-action feature function. As in other research studies [3, 24, 33, 34, 57], here we use a linear function, i.e., $R_\theta(s,a) = \theta^T \varphi(s,a)$, where θ is the weighting vector of features.

Given a reward function, in general, solving a MDP involves obtaining a policy, π: $S \times A \longrightarrow [0,1]$, where $\pi(s,a)$ is the probability of choosing action a in state s, that maximizes the expected return, i.e. $E[\sum_{t=0}^{\infty} \gamma^t R_\theta(s_t, a_t) \mid \pi, T]$. The optimal state value function can be computed recursively using the Bellman equation as $V_\theta^*(s) = \max_{a \in A} [R_\theta(s,a) + \gamma \sum_{s' \in S} T(s,a,s') V_\theta^*(s')]$. Similarly, the optimal state-action value function can be recursively computed as $Q_\theta^*(s,a) = R_\theta(s,a) + \gamma \sum_{s' \in S} T(s,a,s') V_\theta^*(s')$. Also, the optimal state value function can be written in terms of the state-action value function as $V_\theta^*(s) = \max_{a \in A} Q_\theta^*(s,a)$. Thus, the optimal state-action value function becomes

$$Q_\theta^*(s,a) = R_\theta(s,a) + \gamma \sum_{s' \in S} T(s,a,s') \max_{b \in A} Q_\theta^*(s',b). \tag{1}$$

Typically, the IRL seeks for the reward function underlying the demonstration set of the task. This demonstration set is generated according to a certain teacher's policy. Similar to the formulation used in many IRL methods, the demonstrations are represented by a set of trajectories $\mathcal{D} = \{\tau^i\}_{i=1}^K$, where K is the number of trajectories and a trajectory is defined as a set of state-action pairs $\tau^i = \{(s_1, a_1), (s_2, a_2), \ldots\}$. We should note that, in our framework, we use the demonstrations $\mathcal{D}_{\mathrm{E}}$ to indicate the demonstrations provided by the teacher and the demonstrations $\mathcal{D}_{\mathrm{L}}$ to indicate the demonstrations collected from the leaner motion. In this paper, the learner is provided by nonoptimal and sparse (i.e., insufficient number of) demonstrations to estimate the reward function. Taking these assumptions into consideration, one can easily deduce that the traditional IRL alone cannot lead to learning the optimal policy.

The likelihood of the demonstration data $\mathcal{D}$ given the reward function R_θ is defined as $L(\mathcal{D} \mid \theta) = \prod_{\tau \in \mathcal{D}} \prod_{(s,a) \in \tau} [\pi_\theta(s,a)]$. Similar to other works [11, 58], our learning process is not sensitive to the trajectories in the demonstration dataset. It depends on the (s,a) pairs in the demonstration dataset regardless of the trajectory they belong to; thus, the likelihood function can be written as

$$L(\mathcal{D} \mid \theta) = \prod_{(s,a) \in \mathcal{D}} [\pi_\theta(s,a)]. \tag{2}$$

The policy $\pi_\theta(s,a)$ is a stochastic policy, defined by the Boltzmann distribution:

$$\pi_\theta(s,a) = \frac{e^{\alpha Q_\theta(s,a)}}{\sum_{a' \in A} e^{\alpha Q_\theta(s,a')}}, \tag{3}$$

where α controls randomness in the policy.

In our work, we utilize the Bayesian IRL approach (see [8, 11, 28, 29]). More specifically, we adopt the maximum likelihood IRL suggested by [29]. The maximum likelihood IRL (MLIRL) works as follows: given the demonstration dataset $\mathcal{D}$, we seek for the reward function R_θ that maximizes the likelihood of the demonstration data (equation (2)). To that end, a recursive gradient ascent optimization tool is used. First, we take an arbitrary value for θ, and then π_θ is computed by solving the MDP and using equation (3). After that, the likelihood of the demonstrated data (equation (2)) and the gradient of θ is computed. Thereby, θ is updated, and so on (see Figure 1(b)).

4. The Proposed Learning Method

In this section, we present our proposed framework, called IRLDC. In the following, we discuss the detailed framework and delineate the learning and optimization process.

Our framework targets learning by a mixture of sparse as well as nonoptimal demonstrations and human binary evaluative feedbacks, where the learner uses its own evaluated experiences as new demonstrations in an extended IRL method. The learning process starts by providing some demonstrations from the teacher (sparse and/or nonoptimal) for the learner to initialize its policy. Next, the learner interacts with the environment and acquires binary evaluative feedbacks from the teacher. Such feedbacks indicate the desirability of the learner's actions. By taking into account possible inconsistencies and errors in issuing and receiving feedbacks, the learner derives the teacher's preference model. This model is used to revise the estimated reward function by solving a single optimization problem. The cycle of interact-feedback-update continues until the teacher is satisfied.

4.1. IRLDC Framework. Our IRLDC framework includes two main stages: (1) the demonstration stage and (2) the feedback stage. The general framework is shown in Figure 1(a) and it is described procedurally in Algorithm 1.

In the first stage, the teacher provides a demonstration dataset $\mathcal{D}_{\mathrm{E}}$ (sparse and/or nonoptimal) and the learner uses the IRL algorithm (Figure 1(b)) to estimate the reward function parameter $\theta_{\mathrm{initial}}$ (*line* 02). In the second stage, the learner employs $\theta_{\mathrm{initial}}$ to generate its initial policy (*line* 06). Thereafter, the learner observes the world (gets state s), chooses its action a using the initial policy (*line* 08 and *line* 09) and records its trajectories in $\mathcal{D}_{\mathrm{L}} = \{\tau_{\mathrm{L}}^i\}$, where $\tau_{\mathrm{L}}^i = \{(s_1, a_1), (s_2, a_2), \ldots\}$ (*line* 11). Then, the teacher provides a binary evaluative feedback signal (*line* 10) for the executed action a by $f_a \in \{f^+, f^-\}$ within a certain state s, where f^+ and f^- indicate "good" and "bad" actions, respectively. Note that the teacher may give multiple feedbacks at different times in state s denoted by $f_s = \{f_a\}$. Also, $f = \{f_s\}$ denotes the feedback set given by the teacher.

After M steps of interaction with the environment, the performed demonstrations and the received feedbacks are provided as inputs to our proposed IRL algorithm called the maximum likelihood inverse reinforcement learning with

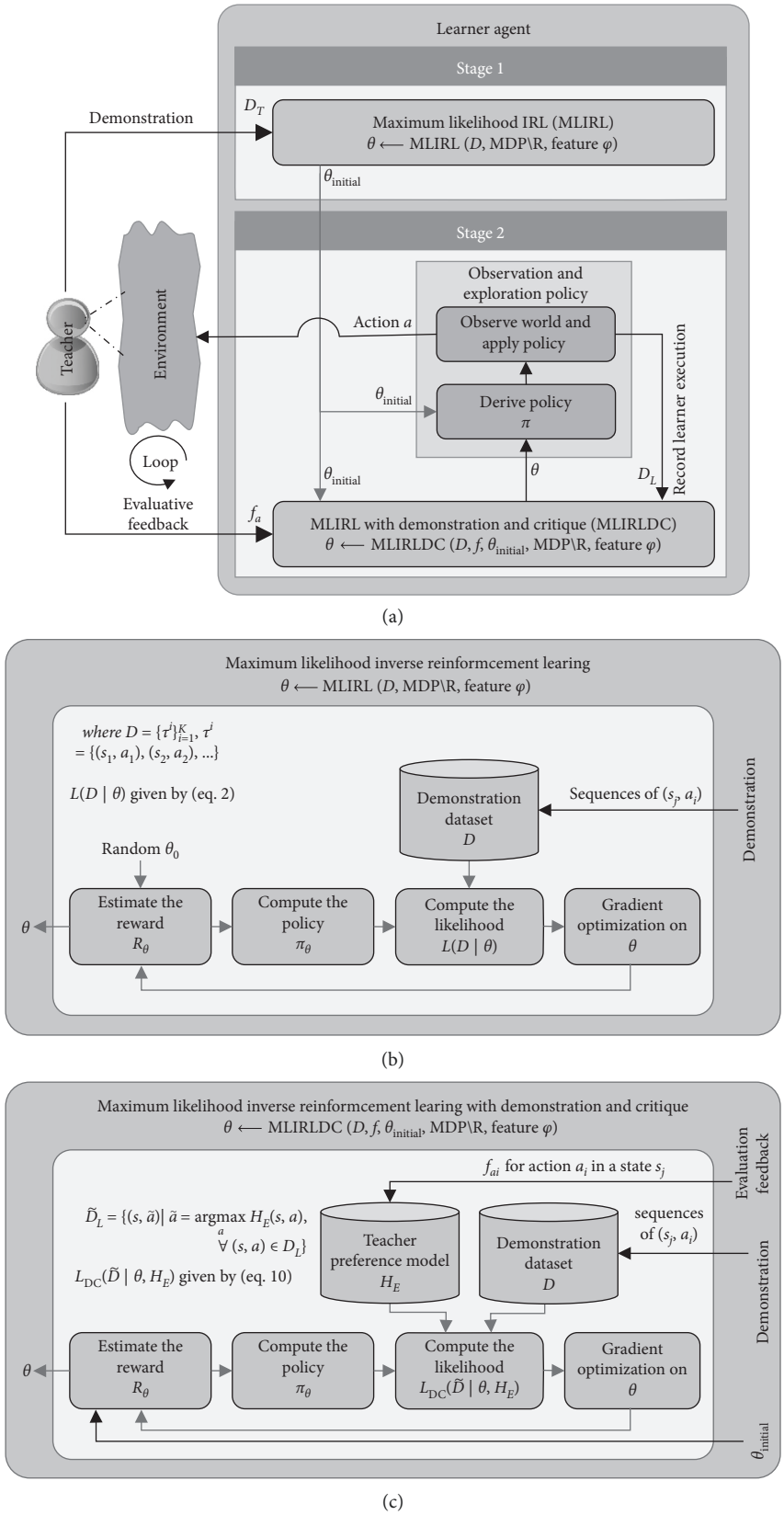


Figure 1: (a) Schematic overview of the IRLDC framework. (b)MLIRL block diagram with only demonstration input. (c) Our MLIRLDC block diagram with demonstration and feedback inputs.

(1) **Input**: MDP/R, feature φ, $\mathscr{D}_{\mathrm{E}}$, and number of interaction steps M
(2) $\theta_{\text{initial}} \longleftarrow \text{MLIRL}(\mathscr{D}_{\mathrm{E}})$
(3) $\theta_{\text{temp}} \longleftarrow \theta_{\text{initial}}$
(4) $\mathscr{D}_{\mathrm{L}} \longleftarrow \varnothing$; $f_s \longleftarrow \varnothing$; $f \longleftarrow \varnothing$
(5) **while** teacher is not satisfied
(6) $\quad \pi_{\text{Learner}} \longleftarrow \text{Solve MDP } (\theta_{\text{temp}})$
(7) $\quad$ **Interact** with the environment for M steps
(8) $\quad\quad s \longleftarrow \text{observe world}(\,)$
(9) $\quad\quad$ execute action a according to π_{Learner}
(10) $\quad\quad$ if teacher critique f_a for (s, a) is received
(11) $\quad\quad\quad \mathscr{D}_{\mathrm{L}} \longleftarrow \mathscr{D}_{\mathrm{L}} \cup (s, a)$
(12) $\quad\quad\quad f_s \longleftarrow f_s \cup f_a$
(13) $\quad\quad\quad f \longleftarrow f \cup f_s$
(14) $\quad$ **end interaction**
(15) $\quad \theta \longleftarrow \text{MLIRLDC}(\mathscr{D}_{\mathrm{L}}, f, \theta_{\text{temp}})$
(16) $\quad \theta_{\text{temp}} \longleftarrow \theta$
(17) **end while**
(18) **Output**: θ

Algorithm 1: IRLDC framework.

demonstration and critique (MLIRLDC). So, the learner uses θ_{initial}, f, and $\mathscr{D}_{\mathrm{L}}$ as inputs for MLIRLDC, to update the reward estimation parameter θ (*line* 15). Using this parameter, the learner updates its policy and executes an action in the environment (*line* 06 and *line* 09). The process of execution and reward function update continues until the teacher satisfaction is attained (*lines* 06–16). We should note that the learner can take different exploration strategies for deriving its policy in the second stage (probabilistic, greedy, and random policies).

As seen in Figure 1(a), in our framework, demonstrations ($\mathscr{D}_{\mathrm{L}}$) are collected from the learner motions in the second stage. On the other hand, the demonstrations provided by the teacher ($\mathscr{D}_{\mathrm{E}}$) are used in the initialization of the MLIRDC and in the initial policy of the learner execution. This allows the learner to operate with diverse combinations of teacher's demonstrations and feedbacks, ranging from demonstrations of any amount or quality, to teacher's feedbacks only.

It is worth mentioning that, from the feedback data, the learner extracts the teacher's preference model $\mathscr{H}_E$, which represents the preferences of the teacher's actions on a certain state s. This preference model is used to weight the likelihood of the demonstrations in the MLIRLDC (Algorithm 2). In the following, first we detail the derivation of $\mathscr{H}_E$ and thereafter describe the MLIRLDC in more details.

4.2. Estimating the Teacher's Feedback Model. Usually, the critique provided by a human teacher is noisy due to the errors in reporting her true assessment (feedback error) and inconsistency in assessing the learner's behavior in a single situation at different times. The inconsistency in feedback can occur due to changes in the teacher's behavior during the teaching process [59], dependency of the teacher's feedback on the current agent policy [44], inconfidency of the teacher in providing feedbacks, and multiple teachers providing feedbacks. Therefore, we use the following feedback model to handle the noise. The feedback model for getting feedback f_{a_j} in state s for performing action a_j is as follows:

$$h\left(s, a_i, f_{a_j} = f^+\right) = \begin{cases} 1-\varepsilon, & \text{if } a_i = a_j, \\ \dfrac{\varepsilon}{|A|-1}, & \text{if } a_i \neq a_j, \end{cases} \quad \forall a_i \in A,$$

$$h\left(s, a_i, f_{a_j} = f^-\right) = \begin{cases} -(1-\varepsilon), & \text{if } a_i = a_j, \\ \dfrac{-\varepsilon}{|A|-1}, & \text{if } a_i \neq a_j, \end{cases} \quad \forall a_i \in A, \tag{4}$$

This model assumes that the teacher determines if the performed action is consistent with her policy π^*, with the probability of error ε (feedback error). If the teacher interprets the learner's action as correct, she gives a positive feedback (f^+) ("good" feedback), so that the action gets a proportion of the "good" feedback equal to $1-\varepsilon$, and each one of the other actions get $\varepsilon/(|A|-1)$. The same model is used for a negative feedback ("bad" feedback). The error ε can also encode the error in the learner's perception of feedback.

The teacher's preference about the agent's action in a certain state is complete and transitive, so we can model it with a utility function $\mathscr{H}(s, a_i \mid f_s)$: $S \times A \longrightarrow \mathbb{R}$:

$$\mathscr{H}\left(s, a_i \mid f_s\right) = \sum_{j=1}^{|f_s|} h\left(s, a_i, f_{a_j}\right), \quad \forall a_i \in A. \tag{5}$$

This utility function is the difference between the number of "good" and "bad" critiques and its value is directly correlated with the teacher's preference for the corresponding action. Equation (5) depends on the history of feedbacks, and therefore, the effect of feedback error and inconsistency in the teacher's critiques are implicitly encoded in that.

(1) **Input**: MDP/R, feature φ, $\mathscr{D}_{\mathrm{L}}$, f, $\theta_{\mathrm{initial}}$, and learning rate β
(2) $\theta \longleftarrow \theta_{\mathrm{initial}}$
(3) compute teacher model $\mathscr{H}_E$ {using equations (4)–(6)}
(4) enhance the demonstration $\mathscr{D}_{\mathrm{L}}$ {using equation (9)}
(5) **while** not converged
(6) compute Q_θ, $\mathrm{d}Q_\theta/\mathrm{d}\theta$ and π_θ {using equations (3) and (12)}
(7) compute $\nabla \log(L_{\mathrm{DC}})$ {using equation (11)}
(8) $\theta \longleftarrow \theta + \beta \nabla \log(L_{\mathrm{DC}})$
(9) **end while**
(10) **Output**: θ

Algorithm 2: MLIRLDC algorithm.

By scaling $\mathscr{H}$ between zero and one, it can be mathematically regarded as a cumulative probability distribution. Subsequently, the teacher's model can be obtained as $\mathscr{H}_E(s, a_i \mid f_s)$: $S \times A \longrightarrow [0, 1]$. Assuming independency among different states, one has

$$\mathscr{H}_E\left(s, a_i \mid f_s\right) = \frac{\mathscr{H}\left(s, a_i\right) + \left(\left|\min_a \mathscr{H}_0(s, a)\right| \times w\right) + k}{\sum_{j=1}^{|A|}\left[\mathscr{H}\left(s, a_j\right) + \left(\left|\min_a \mathscr{H}_0(s, a)\right| \times w\right) + k\right]}, \quad \forall a_i \in A,$$

$$w = \begin{cases} \dfrac{1}{|A| - 1}, & \text{for } H\left(s, a_i\right) < 0 \text{ and } \mathscr{H}_0\left(s, a_i\right) = 0, \\ 1, & \text{otherwise}, \end{cases} \tag{6}$$

where $\sum_{i=1}^{|A|} \mathscr{H}_E(s, a_i \mid f_s) = 1$, and k is a very small number. Also, $\mathscr{H}_0(s, a \mid f_s)$ is defined similarly as equation (5) while considering $\varepsilon = 0$ for the collected feedback dataset f_s. Note that other forms of scaling rather than minimum of $\mathscr{H}_0$ can also be used. This distribution allows the teacher's model to be informative even for actions that do not receive the teacher's critique.

4.3. MLIRLDC Optimization Process and Algorithm. Unlike the majority of IRL algorithms, our proposed IRL algorithm (MLIRLDC) takes demonstrations and evaluative feedbacks as inputs. The implicit assumption in the MLIRL likelihood (equation (2)) is $L(\mathscr{D} \mid \theta) = \prod_{(s,a)\in\mathscr{D}} [\pi_\theta(s, a \in O(s))]$, where $O(s)$ is the correct actions in the state s. We may not have access to the correct action in every state, due to the nonoptimality of the teacher's demonstrations or the absence of them, but we can use the critique data $\{f^+, f^-\}$ which provides a partial evidence for the suitability of action a in state s. Accordingly, we calculate the likelihood using the critique data. To do so, we modify the likelihood model (equation (2)). In the simple case, when there is no inconsistency and error in teacher's feedbacks, we search for θ in a way that:

(i) If the feedback f_a for the pair (s, a) is positive $(f_a = f^+)$, then the action a is exactly correct for that state $(a \in O(s))$; thus, in the likelihood objective function, we must maximize the policy $\pi_\theta(s, a)$.
(ii) If the feedback f_a for the pair (s, a) is negative $(f_a = f^-)$, then the action a is not suitable and exactly wrong for that state $(a \notin O(s))$; thus, in the likelihood objective function, we must maximize $[1 - \pi_\theta(s, a)]$.

As a result, the likelihood objective function of demonstrations given teacher's feedbacks, becomes

$$L_{\mathrm{DC}}\left(\mathscr{D}_{\mathrm{L}} \mid \theta, f\right) = \prod_{\substack{(s,a)\in\mathscr{D}_{\mathrm{L}} \\ \text{where } f_a = f^+}} \pi_\theta(s, a) \prod_{\substack{(s,a)\in\mathscr{D}_{\mathrm{L}} \\ \text{where } f_a = f^-}} (1 - \pi_\theta(s, a)). \tag{7}$$

When the teacher's critiques contain inconsistencies and errors, instead of considering actions that are exactly correct or wrong, we use $\mathscr{H}_E$ (equation (6)) and modify the likelihood (equation (7)) so that the degree of correctness is included:

$$L_{\mathrm{DC}}\left(\mathscr{D}_{\mathrm{L}} \mid \theta, \mathscr{H}_E\right) = \prod_{(s,a)\in\mathscr{D}_{\mathrm{L}}} \left[\pi_\theta(s, a)^{\mathscr{H}_E(s,a)}\right]. \tag{8}$$

The teacher's preference model $\mathscr{H}_E$ affects the optimization process when searching for θ according to its value. If $\mathscr{H}_E(s, a)$ is large $(\mathscr{H}_E(s, a) \longrightarrow 1)$, i.e., a is more likely to be a correct action in state s, the term $\pi_\theta(s, a)^{\mathscr{H}_E(s,a)}$ will highly affect the searching process for parameter θ. In contrast, when $\mathscr{H}_E(s, a)$ is small $(\mathscr{H}_E(s, a) \longrightarrow 0)$, i.e., a is more likely to be a noncorrect action, the term $\pi_\theta(s, a)^{\mathscr{H}_E(s,a)}$ will be large whatever the value of $\pi_\theta(s, a)$ and its effect on the searching process is very low. It means the pair (s, a) will be filtered out from the demonstration $\mathscr{D}_{\mathrm{L}}$. However, in order to fully benefit from the demonstrations and the teacher's preference model $\mathscr{H}_E$ (rather than only filtering out the pair (s, a)), we can learn from the unsuitability of action a in the state s $(\mathscr{H}_E(s, a) \longrightarrow 0)$ by estimating the most likely correct action in that state using the teacher's preference model. Thus, we will firstly enhance the demonstration data $\mathscr{D}_{\mathrm{L}}$ according to the teacher's preference model $\mathscr{H}_E$ as follows:

$$\widetilde{\mathscr{D}}_{\mathrm{L}} = \left\{(s, \widetilde{a}) \mid \widetilde{a} = \arg\max_a \mathscr{H}_E(s, a), \forall (s, a) \in \mathscr{D}_{\mathrm{L}}\right\}. \tag{9}$$

Then, we use the enhanced demonstration (equation (9)) in the likelihood objective function as

$$L_{\mathrm{DC}}\left(\widetilde{\mathscr{D}}_{\mathrm{L}} \mid \theta, \mathscr{H}_E\right) = \prod_{(s,a)\in\widetilde{\mathscr{D}}_{\mathrm{L}}} \left[\pi_\theta(s,a)^{\mathscr{H}_E(s,a)}\right]. \tag{10}$$

The role of the teacher's preference model $\mathscr{H}_E(s,a)$ in equation (9) is to determine the best action in the state s. And its role in equation (10) is to determine the degree of correctness of the action a in the state s.

So, after getting the teacher's preference model $\mathscr{H}_E$, we enhance the demonstration $\mathscr{D}_{\mathrm{L}}$, and then we seek to optimize the objective function (equation (10)) by getting the value of θ such that $\theta^* = \mathrm{argmax}_\theta L_{\mathrm{DC}}(\widetilde{\mathscr{D}}_{\mathrm{L}} \mid \theta, \mathscr{H}_E)$ (Figure 1(c)). To find θ^*, we use a gradient ascent optimization tool:

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}\theta}\log\left[L_{\mathrm{DC}}\left(\widetilde{\mathscr{D}}_{\mathrm{L}} \mid \theta, \mathscr{H}_E\right)\right] &= \sum_{(s,a)\in\widetilde{\mathscr{D}}_{\mathrm{L}}} \frac{\mathscr{H}_E(s,a)}{\pi_\theta(s,a)} \frac{\mathrm{d}\pi_\theta(s,a)}{\mathrm{d}\theta} \\
&= \sum_{(s,a)\in\widetilde{\mathscr{D}}_{\mathrm{L}}} \frac{\mathscr{H}_E(s,a)}{\pi_\theta(s,a)} \frac{1}{B_\theta(s)^2} \\
&\quad \cdot \left[B_\theta(s)\alpha e^{\alpha Q_\theta(s,a)} \frac{\mathrm{d}Q_\theta(s,a)}{\mathrm{d}\theta}\right. \\
&\quad \left. - e^{\alpha Q_\theta(s,a)} \frac{\mathrm{d}B_\theta(s)}{\mathrm{d}\theta}\right],
\end{aligned} \tag{11}$$

where π_θ is given by equation (3), $B_\theta(s) = \sum_{a'} e^{\alpha Q_\theta(s,a')}$ and $\mathrm{d}B_\theta(\mathrm{s})/\mathrm{d}\theta = \sum_{a'} \alpha\, e^{\alpha Q_\theta(s,a')} (\mathrm{d}Q_\theta(s,a')/\mathrm{d}\theta)$.

We should note that the gradient of state-action value function is not differentiable due to the "max" operator in equation (1). To make it differentiable, as in [29], we replace the "max" operator by weighted the state-action values using Boltzmann distribution. Hence, $V_\theta(s) = \sum_{a\in A} \pi_\theta(s,a) Q_\theta(s,a)$ and the state-value function become

$$Q_\theta(s,a) \longleftarrow R_\theta(s,a) + \gamma \sum_{s'\in S} T(s,a,s') \left[\sum_{b\in A} \pi_\theta(s',b) Q_\theta(s',b)\right]. \tag{12}$$

Thus, the state-action value function and its gradient can be computed recursively. The optimization process is summarized in Algorithm 2.

5. Experiments

In our experiments, we assess the performance of our IRLDC framework under different conditions: (i) diverse degrees of demonstration optimality, (ii) different degrees of demonstration sparsity, (iii) lack of demonstration (learning only from feedbacks), (iv) different types of agent policy, and (v) diverse degrees of feedback error.

The experiments are divided into two parts. The first part includes a simulation domain, where the effect of the foresaid aspects is studied. The second part is carried out within two domains to investigate the applicability of our framework for the real human data and real-world problems: highway car driving simulator and a real mobile robot navigation task, both instructed by a human.

In the experiments, the performance evaluation measure is the "expected value" score (EV), to evaluate the optimality level of the learned policy under the "true" reward function. This score is computed by finding the greedy policy π from the learned reward function and then measuring its expected return under the "true" reward function R_T. The "expected value" score of the teacher's policy π_T, derived from the "true" reward function R_T, will be the upper baseline (named "teacher policy") and is used for comparison.

In the literature, the only direct method that uses nonoptimal demonstrations with evaluative feedbacks is our previous work (LfDHF) [15], which we compare with it. Also, we suggest two indirect scenarios to compare our work with LfD methods, which use optimal demonstrations, and LfF approaches:

(i) *Standard* IRL: the standard IRL methods acquire abundant optimal demonstrations, whereas our method employs sparse and nonoptimal demonstrations along with evaluative feedbacks. Therefore, to make a fair comparison, we provide the standard IRL method with the same demonstrations we use in our method as well as a set of optimal demonstrations equivalent to the number of the evaluative feedbacks we employ in IRLDC. Although, according to our assumptions, providing optimal demonstrations might be impractical, we do that just for the comparison purposes. Here, we used MLIRL [29]; other IRL methods also yield similar results in face of sparse and nonoptimal demonstrations.

(ii) *Policy combination*: we derive the policy (π_{demo}) from the provided teacher's demonstrations by using MLIRL to calculate the reward function and then derive the policy by means of dynamic programming [16]. Then, we derive the policy (π_{feedback}) from the provided teacher's feedback (For this method, we use the following settings: the probability of giving explicit and implicit feedbacks is equal, and the feedback error is equal to zero) [45]. Thereafter, we combine the two policies using an idea suggested by [42]:

$$\pi_{\mathrm{Demo\ and\ feedback}}(s,a) = \frac{\left[\pi_{\mathrm{demo}}(s,a) \times \pi_{\mathrm{feedback}}(s,a)\right]}{\left[\sum_{b\in A} \pi_{\mathrm{demo}}(s,b) \times \pi_{\mathrm{feedback}}(s,b)\right]}. \tag{13}$$

In general, we should note that the amount of information provided by optimal demonstrations is more than that of provided by binary evaluative feedbacks. In the following, we relate the information content of these two sources. For $|A|$ number of actions and only one single optimal action per state, by providing optimal demonstrations, the teacher can directly give the optimal action by just

a single interaction per state. By providing binary evaluative feedbacks, the learner may get the optimal action from the first interaction or after $(|A|-1)$ interactions per state. Formally, $i(F \mid s) \in [1, 2, \ldots, |A|-1]$ with a uniform distribution, where $i(F \mid s)$ is the number of feedback interactions per state needed to get the optimal action. Therefore, the average number of feedback interactions per state to get the optimal action will be

$$i(F \mid s)_{\text{average}} = E[\, i(F \mid s)] = \sum_{i(F|s)=1}^{|A|-1} \frac{i(F \mid s)}{|A|-1} = \frac{|A|}{2}. \quad (14)$$

So, the average number of feedback interactions $(i(F)\text{average})$ needed to achieve the same learning performance of optimal demonstration interactions will be

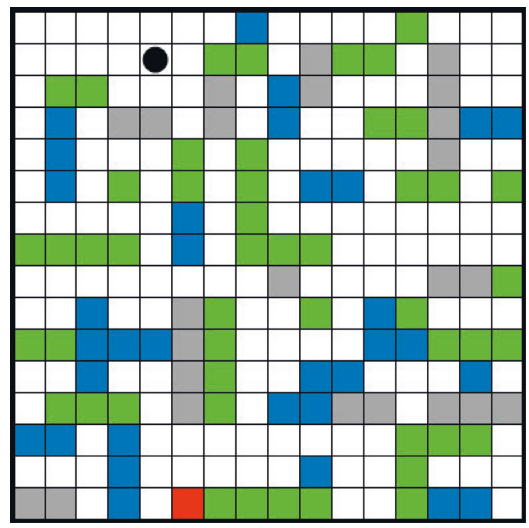
$$i(F)_{\text{average}} = \frac{|A|}{2} \times i(D), \quad (15)$$

where $i(D)$ is the number of state-action pair in demonstrations. In case of error-free feedbacks and nonoptimal demonstrations, the number of required feedbacks will be reduced.

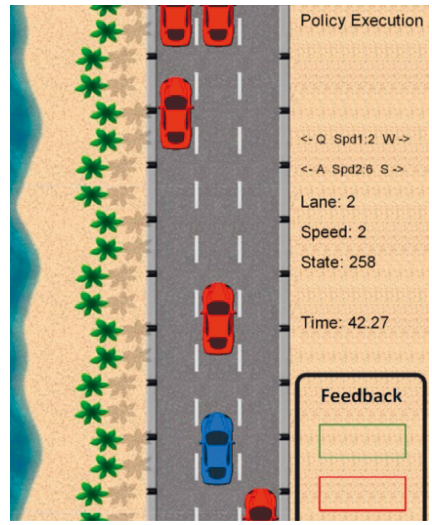
5.1. Simulated Navigation Domain. In this experiment, we consider a simulated navigation task on a 16×16 multifeature grid world, such as in Figure 2(a). The learner robot has five actions for navigation (up, down, left, right, and stay motionless), where each action has 10% chance of failure, leading to one random step move. The purpose of the learner robot is to navigate in the environment by following the teacher's navigation style to reach the goal.

To capture the teacher's navigation style, five features are defined in the environment, namely, ground, puddle, grass, obstacle, and goal, yielding to 5-dimensional binary feature vector φ which is used to characterize each state. For example, a navigation style could be moving in the environment while avoiding the obstacles, with a priority for going through the grass as much as possible; otherwise, it is preferred to pass through the ground rather than over puddle.

The learner's state is represented by its position in the grid which has Markov properties. The reward function is represented by a linear combination of the state's features $(R_\theta(s) = \theta^T \varphi(s))$ and it is unknown to the learner. By manually setting a feature weight vector $\theta_{\text{True}} \longleftarrow \theta$, we obtain a "true reward" function $(R_T = R_{\theta_{\text{True}}})$ which represents a specific teacher's navigation style. Then, we use a planning algorithm to compute the optimal teacher's policy (π_T) for this reward function. Thereafter, the nonoptimal demonstrations are derived by drawing the starting state from a fixed distribution, and the optimal policy is then sampled with a certain chance (degree of nonoptimality $\eta \in [0, 1]$) of selecting a nonoptimal action in each state. Each demonstration is terminated when reaching the goal or after 50 steps are elapsed—among the derived nonoptimal trajectories, we select ones that have the nonoptimality level near to η.



(a)



(b)

Figure 2: (a) Grid world navigation domain. The white, blue, green, gray, and red cells depict the ground, puddle, grass, obstacle, and goal, respectively. The black circle represents the learner robot. (b) A snapshot of our highway car driving simulator.

Similarly, in the execution phase (stage 2; see Section 4.1) the learner agent starts from a state drawn from a certain distribution and terminates its episode when either reaching the goal or after 50 steps. The simulated teacher provides an evaluative feedback after each learner's action, depending on the teacher's policy and feedback error ε.

The simulation is performed using the settings summarized in Table 2. The results of learning in stage 1 of our framework are shown in Figure 3, which show that the agent performance is inversely correlated with the number and nonoptimality degree of demonstrations. The results are averaged over 100 repetitions. The plain lines in the graphs shown in the following pages are the "mean" value of the EV scores and the shaded colored areas are the "standard deviation".

Table 2: Setting used to study different aspects of our framework (IRLDC) in the simulated navigation domain.

Aspect	Demonstration		Feedback error ε	Learner policy type
	Step number	Optimality degree		
Comparison (5.1.1)	100 (s,a)	60% (point A5)	0	Probabilistic
	20 (s,a)	100% (point B2)	0	Probabilistic
Nonoptimal demonstration effect (5.1.2)	100 (s,a)	Different (points: A1–A8, B10, C)	0	Probabilistic
Sparse demonstration effect (5.1.3)	Different (points: B1–B10, C)	100%	0	Probabilistic
Learn only from feedback (5.1.4)	No demo (point C)	—	0	Probabilistic
Effect of the policy execution (5.1.5)	100 (s,a)	60% (point A5)	0	Probabilistic, greedy, random
Effect of feedback error (5.1.6)	100 (s,a)	60% (point A5)	$\varepsilon = 0, 0.1, 0.5$	Probabilistic

Note: learning from the collected feedbacks is done in the batch learning mode.

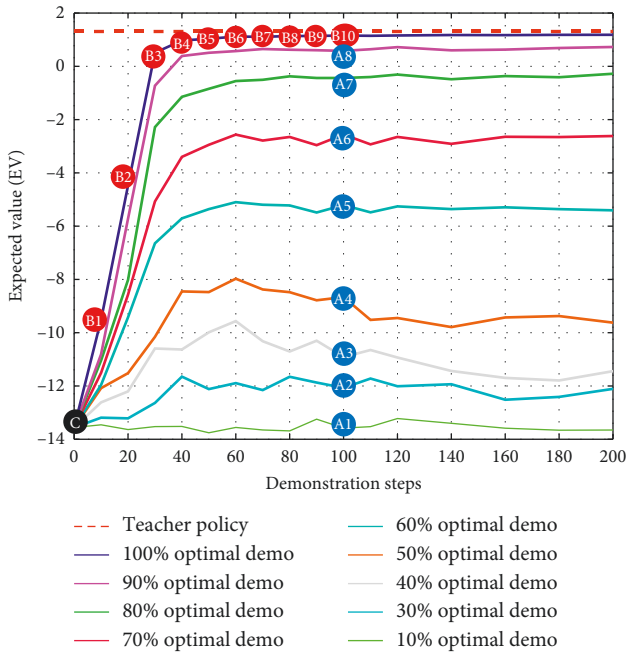


Figure 3: Performance of the standard IRL method (MLIRL) used in the first stage of our framework. The plain curves are the mean of "EV" scores with respect to demonstration steps and nonoptimality degree. The blue, red, and black circles are different initialization settings for stage 2 of our framework.

5.1.1. Comparison with Other Approaches. Figure 4(a) illustrates the performance of our framework in face of nonoptimal demonstrations, where we only need 200 evaluative feedbacks to statistically reach the teacher's performance. This is a reasonable number in comparison with the information transferred by the evaluative feedbacks. Compared to our previous work (LfDHF) [15], our current method (IRLDC) results in very significantly higher learning performance because we use human demonstrations for initialization of our method and employ the learner's own experience trials as new demonstrations. This highly increases sample efficiency and expedites the generalization of experiences. In addition, our previous work filtered out nonoptimal demonstrations, but here we learn from them. In contrast, we see that the "policy combination" method hardly reaches the desired result even if a large number of feedbacks are provided. The results of MLIRL method, which is considered as one of the best methods to deal with nonoptimal demonstrations in an IRL domain, show that nonoptimality has a deep influence on its performance and it requires a large number of additional optimal demonstrations to attain an acceptable result, while providing optimal demonstrations is against our realistic assumption. Therefore, large number of feedbacks and optimal demonstrations cannot resolve the nonoptimality effect within the "policy combination" and "MLIRL" methods, respectively, while within a few number of feedbacks our approach (IRLDC) yielded a much more better result.

Figure 4(b) is the case where a limited number of optimal demonstrations is provided. Having optimal demonstrations, $(\eta \longrightarrow 0)$, IRLDC and MLIRL statistically exhibit a similar performance. For IRLDC, we need $i(f) = 260$ feedbacks, which is equivalent to $i(D) = 100$ extra state-actions in optimal demonstrations in MLIRL. By considering the number of actions $|A|$ equal to five, these values obey equation (15). Our previous work (LfDHF) [15] could not exploit evaluative feedbacks to compensate sparsity in demonstrations. Because it just focused on using human evaluative feedbacks to correct teacher's demonstrations. Regarding the "policy combination" method, it needs a large number of feedbacks to reach an acceptable result. Naturally, the methods employing evaluative feedbacks, i.e., IRLDC and policy combination, show a larger variance.

5.1.2. Nonoptimal Demonstration Effect. According to Figure 5, one can see that, in all cases, the effects of nonoptimality on the learning process can be compensated by using evaluated feedbacks. Nevertheless, it shows that when demonstrations are more misleading than being informative (with optimality degree less than 50%), it is better to use only feedbacks and ignore the demonstrations.

5.1.3. Sparse Demonstration Effect. The relation between the number of required feedbacks increases nonlinearly with the increment of sparsity in demonstrations (see Figure 6(a)). Figure 6(b) indicates that when the optimal demonstration steps (s,a) increase, rapid improvement in the performance

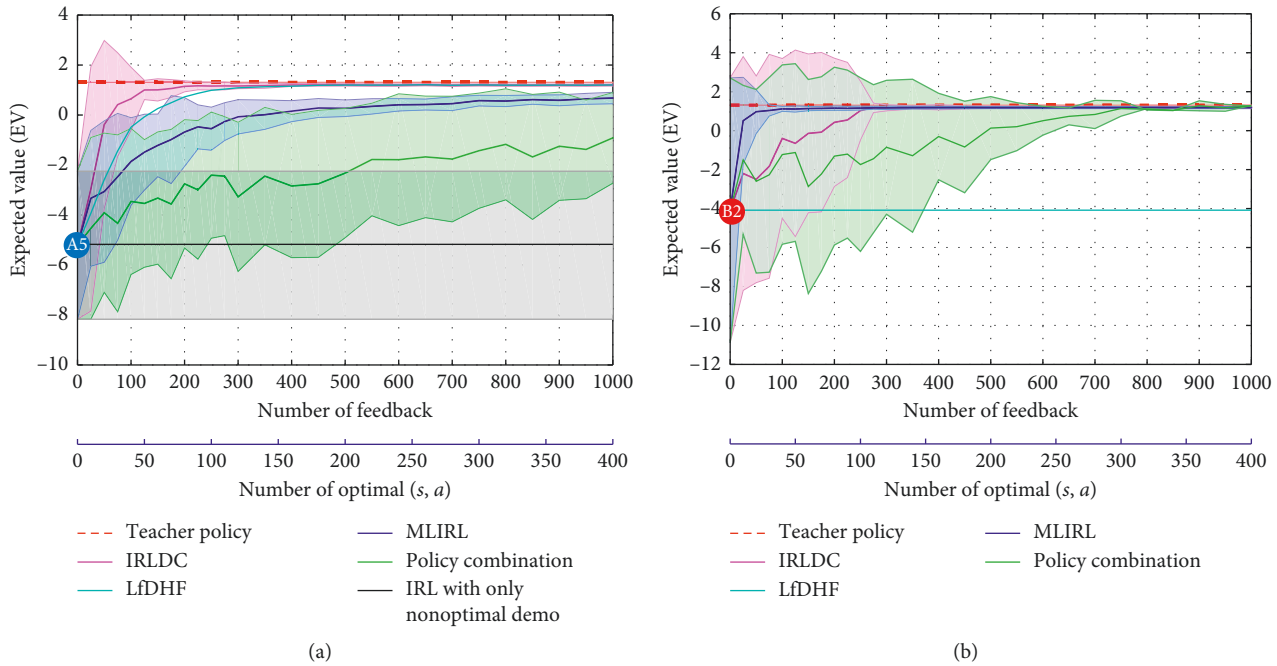


Figure 4: (a) The effect of nonoptimality in demonstrations. The experiment setting: initial nonoptimal demonstration "A5" with 60% optimality and 100 demonstrations in the first stage (see Figure 3). (b) The case where all demonstrations in the first stage are optimal but sparse. The experiment setting: initial sparse demonstration "B2" and 20 demonstrations in the first stage (see Figure 3). Two kinds of data are provided during the experiment: evaluative feedbacks related to the IRLDC, LfDHF, and policy combination (first horizontal axis), and state-action pairs in extraoptimal demonstrations used in the MLIRL (second horizontal axis).

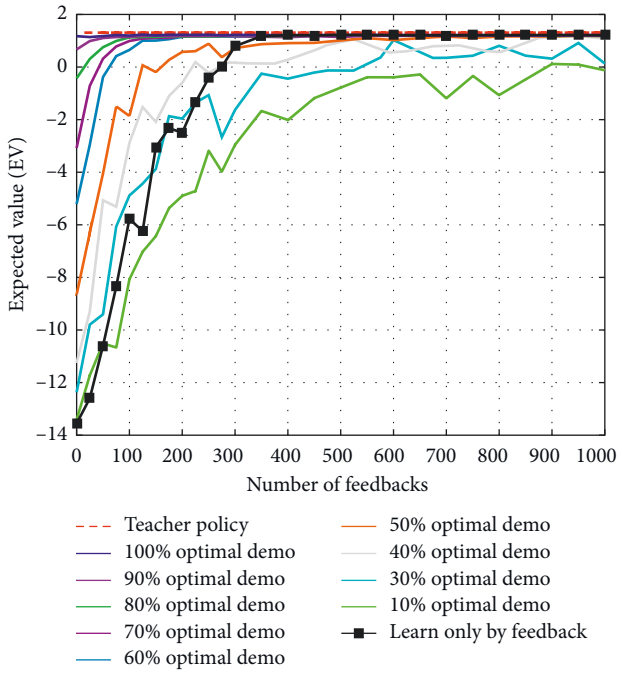


Figure 5: IRLDC's stage-two performance in face of abundant and different demonstration optimality levels in stage-one (points "A1", ..., "A8" and "B10" in Figure 3) and the number of evaluative feedbacks. The black curve has no initial demonstration (point "C" in Figure 3).

occurs. This confirms the intuition that using any amount of optimal demonstrations makes the learning process faster than using only feedbacks. Also, Figure 6(b) depicts that the lack of demonstrations (i.e., sparsity) can be compensated by employing reasonable number of feedbacks.

5.1.4. Learning Only from Feedbacks (No Demonstrations). The performance shown in Figure 7 indicates that, even in the absence of demonstrations, only feedback data are sufficient for the IRLDC to get a good result. Though convergence is slow in the early learning trails, after collecting a sufficient number of feedbacks, the convergence is expedited; this is due to the generalization capability embedded in the IRLDC. This makes IRLDC performance better than that of [45] used in "policy combination" scenario (Figure 4(b)). In addition, learning only from feedbacks obeys equation (15), where it needs $i(F) = 350$ to achieve the same score value of $i(D) = 140$.

5.1.5. Effects of the Learner Policy on Learning Process. In the IRLDC framework, in the first stage, the agent observes demonstrations and then, in the second stage, it uses the gained knowledge to learn interactively. In the second stage, the agent receives feedbacks from the critic and uses that information in its IRL engine to improve its behavior. The agent can use different policies in this stage. Figure 8

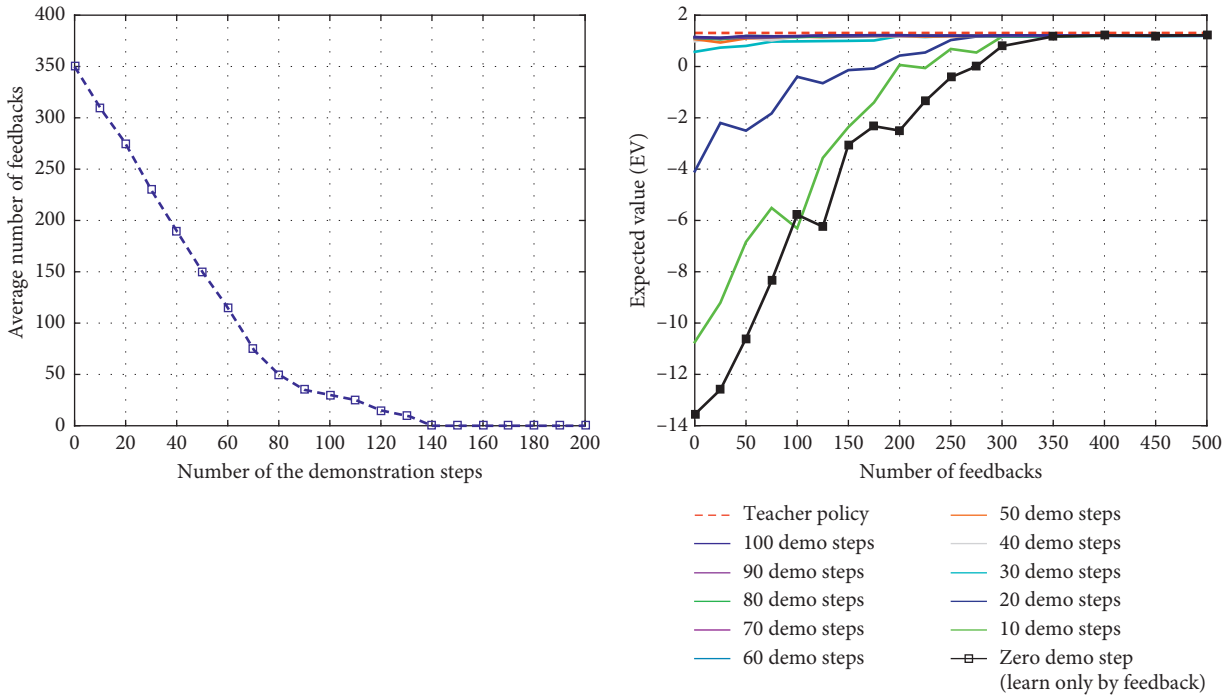


FIGURE 6: (a) The relation between sparsity level of demonstrations in stage-one and the number of feedbacks needed to reach "EV" score equal to 1.17 using IRLDC. (b)IRLDC's stage-two performance in face of optimal and different demonstration sparsity levels in stage-one (point "B1", . . ., "B10" in Figure 3) and the number of evaluative feedbacks. The black curve has no initial demonstration (point "C" in Figure 3).

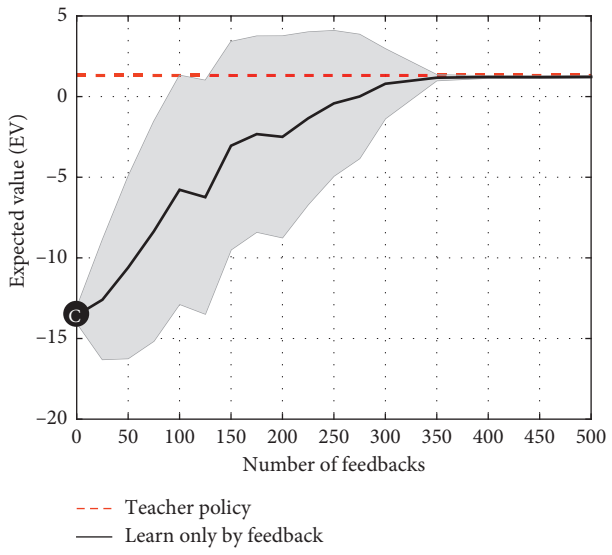


FIGURE 7: The performance of IRLDC framework when learning only from evaluative teacher's feedback. This is the case where no demonstration is available (point "C" in Figure 3).

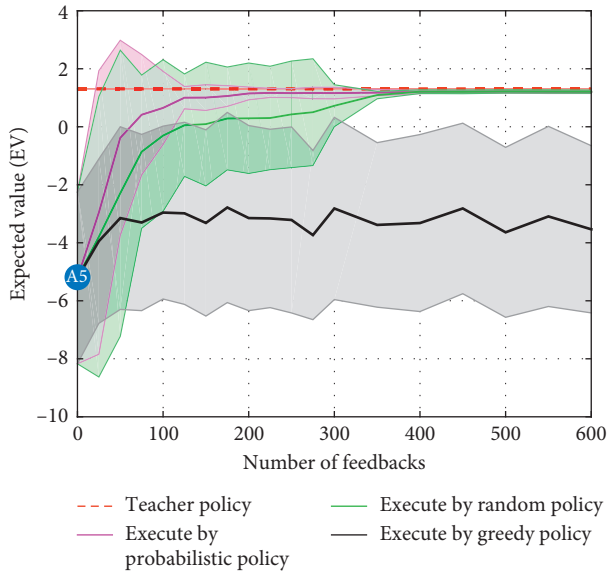


FIGURE 8: The performance of IRLDC under different exploration policy types used in the interactive phase (stage-two) when 100 state-action pairs of 60% optimal demonstrations are given (see Table 2).

compares the performance of the agent against different number of feedbacks using random, probabilistic, and greedy policies. This experiment is done by using batch learning mode for the collected feedbacks. Since the demonstrations are not optimal and sufficient, the agent needs to balance the exploration-exploitation to gain sufficient feedbacks as well as minimizing its regret. The greedy policy is the worst, since it gains information from feedbacks mostly in states where demonstrations are not optimal, and it cannot collect diverse information. In contrast,

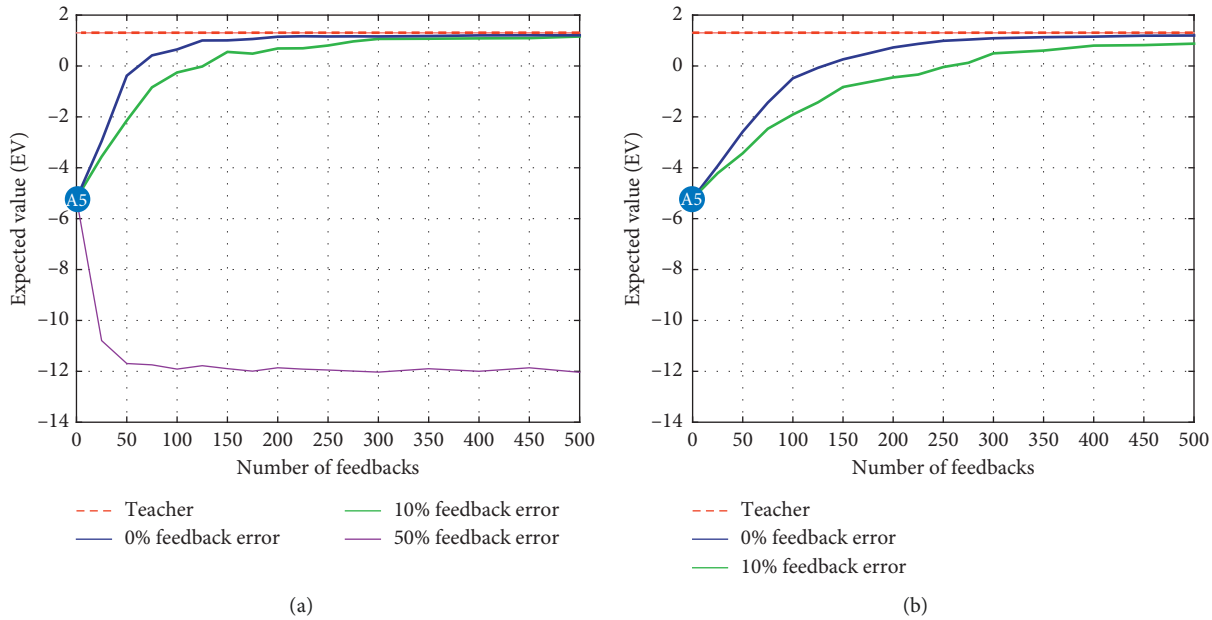


Figure 9: (a) The performance of IRLDC framework under different feedback error values used in the interactive phase when 100 state-action pairs of 60% optimal demonstrations are given (see Table 2). (b) The performance of our previous work (LfDHF) [15] with similar setting to (a).

probabilistic and random policies provide the agent with the chance of facing states not seen in the demonstrations.

5.1.6. Effect of the Feedback Error. As mentioned in section 0, our model can handle errors and inconsistencies in the feedbacks. Due to space constraints, in this experiment, we only study the effect of feedback error. An insight into Figure 9(a) reveals that the learning performance remains acceptable and the navigation style can be learned even in the presence of noisy feedbacks. It can also be seen that the negative effect of the noise is diminished as the number of feedbacks grows, provided that the noise level is below 50%.

Figure 9(b) illustrates the performance of our previous work (LfDHF) [15] in face of errors in the feedbacks; as the feedbacks' errors increase, the learning performance deteriorates, and as a result, LfDHF needs a large number of feedbacks to attain acceptable results.

5.2. Highway Car Driving Experiment. In this section, we investigate the applicability of our framework with real human data in a dynamic environment. We utilized the car driving experiment that is devised in our previous work [15]. Our task is to navigate the agent car through three busy highway lanes (Figure 2(b)) using five actions: moving left/right, speeding up/down, and no action. The learner agent car moves faster than all of the other cars even at its lowest speed. The state space is constituted of the learner's speed, its lane, and the distribution of other cars on the highway. We consider two driving styles:

Style 1. Giving the highest priority to avoiding collisions with other cars, preferring the middle lane with high speed over the left lane with high speed, and over the right lane with low speed

Style 2. The highest preference is to collide with other cars as possible, and it is preferred to drive at middle lane with high speed

Each of these styles is learned from demonstrations and feedbacks from a real human teacher interacting with the simulator through a keyboard. The nonoptimality in the demonstrations is imposed by assuming that the learner agent perceives the teacher's demonstrations with 30% error, that is, on top of the unmeasurable natural error in human demonstrations and feedbacks. In order to decrease the direct communication between the teacher and the learner, only negative feedbacks are given by the teacher. The pace of the simulator is set in a way that the teacher can conveniently give feedback per decision.

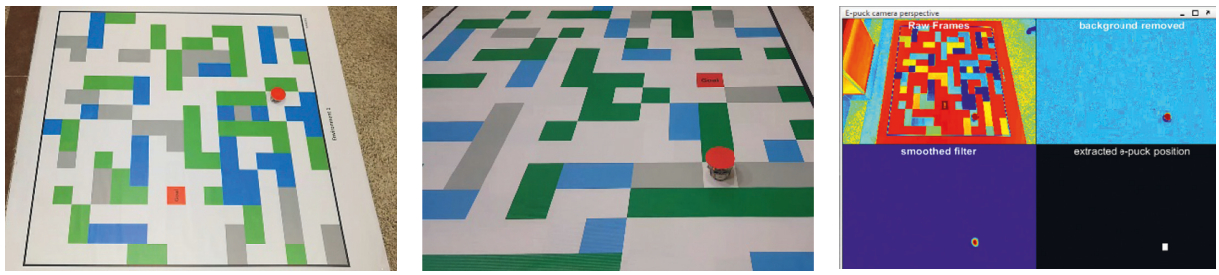
When working with a human teacher, her "true" reward function is not available; instead, a task-specific performance measure is needed for the evaluation purpose [3, 25]. Here, we apply the standard IRL to the teacher's optimal demonstrations and take the extracted reward as a proxy of the "true" reward function.

Table 3 shows the results averaged over 5 independent runs, and $M = 40$ interaction steps with the environment before the learner policy is updated. These results illustrate that the IRLDC with various demonstrations and reasonable number of feedbacks achieves the same performance of the standard IRL given the teacher's optimal demonstrations. A video of this experiment and the learned behavior can be found at http://bit.ly/31FnwGT.

5.3. E-Puck Robot Experiment. Here we use an E-puck educational mobile robot [60] navigating in two environments

Table 3: Learning the driving styles from a human teacher in different conditions.

Driving style	Demonstrations			Number of feedbacks		EV of IRLDC	EV of teacher policy
	Type	Time	Optimality degree	All	Negative		
Style 1	Abundant	120 sec	70%	164	62	16.821	17.164
	Sparse	20 sec	100%	302	84	16.906	
	No demo	—	—	409	239	16.684	
Style 2	Abundant	120 sec	70%	124	48	14.893	15.275
	Sparse	20 sec	100%	218	59	14.969	
	No demo	—	—	306	182	14.877	



(a) (b) (c)

Figure 10: (a) and (b) E-puck robot navigating in two different environments. (c) E-puck external camera perspective used to localize its position.

Table 4: Results of the learned navigation experiment by the E-puck robot in two different environments, with $M = 30$ interaction steps with the environment before the learner policy is updated. Here, in experiments 1 and 3, demonstrations and feedbacks are provided in the same environment, and then, the performance of the learned reward function is examined in the second environment. Experiment 2 is done by providing sparse demonstrations in one environment and feedbacks in another.

Experiment	Demonstration		Environment 1				Environment 2			
			Number of feedbacks		EV of IRLDC	EV of teacher policy	Number of feedbacks		EV of IRLDC	EV of teacher policy
	Type	Number of (s, a)	All feedbacks	Negative feedbacks			All feedbacks	Negative feedbacks		
1	Abundant, nonoptimal	100	135	51	2.735	2.824	—	—	3.145	3.279
2	Sparse, near-optimal	22	—	—	2.749	2.824	239	70	3.157	3.279
3	No demo	—	311	181	2.761	2.824	—	—	3.188	3.279

similar to the one employed in Section 5.1 (see Figure 10). The robot learns the navigation style of the human teacher interacting with it through a keyboard. The robot's odometer and an external camera are used for localization and motion error correction (see Figure 10(c)). The robot has five actions: moving forward/backward, rotating clockwise/counterclockwise, and staying motionless. The transition model is estimated from the previously collected sequences of transition triplets (s, a, s').

The teacher's navigation style is as follows: moving in the environment to reach the goal (the red cell) while avoiding the gray cells, with a priority for going through the green cells as much as possible; otherwise, it is preferred to pass through the white cells rather than the blue ones. Two environments are involved in this experiment (Figures 10(a) and 10(b)), using the same features and state representation, for the following purposes:

(i) Testing the performance of the learned reward function: where the reward function is learned in one environment and evaluated in the second

(ii) Providing demonstrations in one environment and feedbacks in another

We induce nonoptimality in the demonstrations by distracting the attention of the teacher when providing the demonstrations, that is, on top of the unmeasurable natural error in human demonstrations and feedbacks. The "true" reward function is estimated using the standard IRL on the optimal human demonstrations on a simulated version of the task. The feedback protocol and the learner pace setting are similar to the previous section.

The results of this experiment are summarized in Table 4. They show that our framework performs well in the real-world environment as well as when the learned reward

function is generalized to a new environment. Also, the results are consistent with the previous simulated domain. Indeed, the results provide further affirmation that nonoptimal and sparse demonstrations are useful and help the learning process when using them along with evaluative feedbacks. A video of this experiment and the learned behavior can be found at http://bit.ly/31FnwGT.

6. Conclusions

In this paper, we introduced the IRLDC to learn from a mixture of sparse as well as imperfect demonstrations and human evaluative binary feedbacks. Employing these two sources of information, the IRLDC is a practical and convenient tool to program artificial systems in real-world situations, where nonoptimal and sparse human's demonstrations are common and inconsistency as well as error in human's feedbacks is usual. Having the state transition model, the IRLDC estimates the reward function in a single optimization problem in order to generalize the expertise embedded in demonstrations and feedbacks, where standard IRL methods fail in face of sparse and imperfect demonstrations and learning from feedbacks (standard LfF methods) suffers from the curse of dimensionality and high load on human teacher to provide rewards. The closest approach [15] to IRLDC does not benefit from the learner's experiences to improve the learning process and just focuses on using human evaluative feedbacks to correct the teacher's demonstrations and to filter out the nonoptimal ones. These result in failure to face sparsity as well as limited robustness against nonoptimality in demonstrations. In contrast, in IRLDC we use the learner's own experiences as additional demonstrations which enhance sample efficiency and generalization and lead to lower regret and faster learning. In addition, we exploit errors in demonstrations, instead of filtering them out, to improve IRL through giving a higher chance to alternative decisions. These properties make the method faster and highly robust in face of errors in demonstrations and feedbacks.

Comparing to other state-of-the-art methods, which combine demonstrations with RL experience, use corrective actions, or advice preferences, to learn from nonoptimal and sparse demonstrations, we follow a different paradigm to leverage learning from human in order to allow her to simply express her preferences through adding evaluative feedbacks. Unlike the aforementioned rich sources of information, evaluative feedback is simple, offers strengths, and imposes minimum constraints on the teacher during the teaching process. Nevertheless, corrective actions and advice, if available, can be directly used in our IRL model and boost our results further.

We studied the functionality of the IRLDC in three distinct problems: a grid world task, a car driving simulator, and an E-puck mobile robot navigation, where human data are used in the last two cases. The results showed that the addition of feedbacks in our framework exploits well the nonoptimal and sparse demonstrations, when the nonoptimality is below 50%. In addition, the learning was done well in face of intrinsic errors in human feedbacks. Moreover, the IRLDC worked well when programming solely by feedbacks; however, convergence occurred slowly in a linear way.

One of the major assumptions in the IRLDC, as well as in standard IRL methods, is having the state transition model. This assumption is very realistic and prevalent, when learning a new task or style in a known environment. Testing the IRLDC's robustness in face of limited errors in the state transition model is a problem for further studies. Furthermore, we assumed that every decision of the learner can be distinctly evaluated by the teacher. However, this setting is not practical in some situations where the pace of the learner is fast or the effect of multiple decisions is evaluated at once. These situations in turn arises the credit assignment problem [38]. Handling such situations is the next step of this research. In addition, we would like to employ our method in deep neural networks to attain higher generalization in face of more complex problems.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Supplementary Materials

There are two videos in the supplementary materials related to our experiments: Video "Highway car driving task.mp4" describes about the experiment (Section 5.2 in this work) and shows the agent car navigation style during and after the learning phases. Video "E-puck robot navigation task.mp4" describes about the experiment (Section 5.3 in this work) and shows the E-puck mobile robot behavior during and after the learning phases. (*Supplementary Materials*)

References

[1] J. A. Bagnell, *An Invitation to Imitation*, Carnegie Mellon University Field Robotics Center, Pittsburgh, PA, USA, 2015.

[2] J. MacGlashan and M. L. Littman, "Between imitation and intention learning," in *Proceedings of the 24th International Conference on Artificial Intelligence*, AAAI Press, Aires, Argentina, July 2015.

[3] P. Abbeel and A. Y. Ng, "Apprenticeship learning via inverse reinforcement learning," in *Proceedings of the Twenty-First International Conference on Machine Learning*, ACM, Alberta, Canada, July 2004.

[4] A. Y. Ng and S. J. Russell, "Algorithms for inverse reinforcement learning (2000)," in *Proceedings of the Seventeenth International Conference on Machine Learning (ICML 2000)*, Stanford, CA, USA, June 2000.

[5] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An introduction*, Vol. 1, MIT Press, Cambridge, UK, 1 edition, 1998.

[6] J. Ho, J. K. Gupta, and S. Ermon, "Model-free imitation learning with policy optimization," 2016, https://arxiv.org/abs/1605.08478.

[7] S. Chernova and A. L. Thomaz, "Robot learning from human teachers," *Synthesis Lectures on Artificial Intelligence and Machine Learning*, vol. 8, no. 3, pp. 1–121, 2014.

[8] M. Lopes, F. Melo, and L. Montesano, "Active learning for reward estimation in inverse reinforcement learning," in *Proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, Springer, Bled, Slovenia, September 2009.

[9] A. Boularias, J. Kober, and J. Peters, *Relative Entropy Inverse Reinforcement Learning*, in *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, Lauderdale, FL, USA, April 2011.

[10] J. Zheng, S. Liu, and L. M. Ni, "Robust bayesian inverse reinforcement learning with sparse behavior noise," in *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, Québec, Canada, July 2014.

[11] D. Ramachandran and E. Amir, "Bayesian inverse reinforcement learning," *Urbana*, vol. 51, p. 61801, 2007.

[12] Y. Gao, *Reinforcement Learning from Imperfect Demonstrations*, https://arxiv.org/abs/1802.05313, 2018.

[13] T. Hester, *Deep Q-Learning from Demonstrations*, https://arxiv.org/abs/1704.03732, 2018.

[14] A. Nair, "Overcoming exploration in reinforcement learning with demonstrations," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, Brisbane, Australia, May 2018.

[15] A. Ezzeddine, "Combination of learning from non-optimal demonstrations and feedbacks using inverse reinforcement learning and Bayesian policy improvement," *Expert Systems with Applications*, vol. 112, pp. 331–341, 2018.

[16] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An introduction*, vol. 1, MIT Press, Cambridge, UK, 2nd edition, 2018, http://www.incompleteideas.net/book/the-book-2nd.html.

[17] R. Cohn, E. Durfee, and S. Singh, "Comparing action-query strategies in semi-autonomous agents," in *Proceedings of the 10th International Conference on Autonomous Agents and Multiagent Systems-Volume 3*, International Foundation for Autonomous Agents and Multiagent Systems, Taipei, Taiwan, July 2011.

[18] S. Levine, Z. Popovic, and V. Koltun, "Nonlinear inverse reinforcement learning with Gaussian processes," in *Proceedings of the Advances in Neural Information Processing Systems, NIPS 2011*, Granada, Spain, September 2011.

[19] B. D. Argall, S. Chernova, M. Veloso, and B. Browning, "A survey of robot learning from demonstration," *Robotics and Autonomous Systems*, vol. 57, no. 5, pp. 469–483, 2009.

[20] A. Hussein, M. M. Gaber, E. Elyan, and C. Jayne, "Imitation learning: a survey of learning methods," *ACM Computing Surveys (CSUR)*, vol. 50, no. 2, pp. 1–35, 2017.

[21] D. Silver, J. A. Bagnell, and A. Stentz, "Learning from demonstration for autonomous navigation in complex unstructured terrain," *The International Journal of Robotics Research*, vol. 29, no. 12, pp. 1565–1592, 2010.

[22] K. K. Budhraja and T. Oates, "Neuroevolution-based inverse reinforcement learning," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC)*, IEEE, Sebastián, Spain, June 2017.

[23] E. Klein, "A cascaded supervised learning approach to inverse reinforcement learning," in *Proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, Springer, Prague, Czech Republic, September 2013.

[24] E. Klein, "Inverse reinforcement learning through structured classification," in *Proceedings of the Advances in Neural Information Processing Systems*, Lake Tahoe, December 2012.

[25] S. Levine and V. Koltun, "Continuous inverse optimal control with locally optimal examples," 2012, https://arxiv.org/abs/1206.4617.

[26] A. Šošić, "Inverse reinforcement learning in swarm systems," in *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems*, International Foundation for Autonomous Agents and Multiagent Systems, São Paulo, Brazil, May 2017.

[27] M. Wulfmeier, P. Ondruska, and I. Posner, "Maximum entropy deep inverse reinforcement learning," 2015, https://arxiv.org/abs/1507.04888.

[28] B. D. Ziebart, "Maximum entropy inverse reinforcement learning," in *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence. Proceedings Cover Image AAAI-08*, Chicago, IL, USA, July 2008.

[29] M. Babes, "Apprenticeship learning about multiple intentions," in *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, Bellevue, WC, USA, June 2011.

[30] A. Coates, P. Abbeel, and A. Y. Ng, "Learning for control from multiple demonstrations," in *Proceedings of the 25th International Conference on Machine Learning*, ACM, Helsinki, Finland, July 2008.

[31] S. Calinon and A. Billard, "Recognition and reproduction of gestures using a probabilistic framework combining PCA, ICA, and HMM," in *Proceedings of the 22nd International Conference on Machine Learning*, ACM, Edinburgh, Scotland, August 2005.

[32] S. Chernova and M. Veloso, "Confidence-based policy learning from demonstration using Gaussian mixture models," in *Proceedings of the 6th International Joint Conference on Autonomous Agents and Multiagent Systems*, ACM, Honolulu, HI, USA, July 2007.

[33] C. Xia and A. El Kamel, "Neural inverse reinforcement learning in autonomous navigation," *Robotics and Autonomous Systems*, vol. 84, pp. 1–14, 2016.

[34] K. Shiarlis, J. Messias, and S. Whiteson, "Inverse reinforcement learning from failure," in *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems*, International Foundation for Autonomous Agents and Multiagent Systems, Singapore, May 2016.

[35] R. Akrour, M. Schoenauer, and M. Sebag, "Preference-based policy learning," in *Machine Learning and Knowledge Discovery in Databases*, pp. 12–27, Springer, Berlin, Germany, 2011.

[36] R. Akrour, "Programming by feedback," in *Proceedings of the International Conference on Machine Learning*, JMLR. org, Beijing, China, January 2014.

[37] A. Jain, "Learning trajectory preferences for manipulators via iterative improvement," in *Proceedings of the Advances in Neural Information Processing Systems (NIPS 2013)*, Lake Tahoe, NV, USA, December 2013.

[38] W. B. Knox and P. Stone, "Interactively shaping agents via human reinforcement: the TAMER framework," in *Proceedings of the Fifth International Conference on Knowledge Capture*, ACM, Redondo Beach, CA, USA, September 2009.

[39] W. B. Knox and P. Stone, "Combining manual feedback with subsequent MDP reward signals for reinforcement learning," in *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: Volume 1*, International Foundation for Autonomous Agents and Multiagent Systems, Toronto, Canada, May 2010.

[40] G. Li, "Using informative behavior to increase engagement in the tamer framework," in *Proceedings of the 2013 International Conference on Autonomous Agents and Multi-Agent Systems*, International Foundation for Autonomous Agents and Multiagent Systems, Saint Paul, MN, USA, May 2013.

[41] N. A. Vien, W. Ertel, and T. C. Chung, "Learning via human feedback in continuous state and action spaces," *Applied Intelligence*, vol. 39, no. 2, pp. 267–278, 2013.

[42] S. Griffith, "Policy shaping: integrating human feedback with reinforcement learning," in *Proceedings of the Advances in Neural Information Processing Systems (NIPS 2013)*, Lake Tahoe, NV, USA, December 2013.

[43] T. Cederborg, "Policy shaping with human teachers," in *Proceedings of the International Joint Conferences on Artificial Intelligence (IJCAI)*, Buenos Aires, Argentina, July 2015.

[44] J. MacGlashan, "Interactive learning from policy-dependent human feedback," 2017, https://arxiv.org/abs/1809.01889.

[45] R. Loftin, "Learning behaviors via human-delivered discrete feedback: modeling implicit feedback strategies to speed up learning," *Autonomous Agents and Multi-Agent Systems*, vol. 30, no. 1, pp. 30–59, 2016.

[46] B. Peng, "A need for speed: adapting agent action speed to improve task learning from non-expert humans," in *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems*, International Foundation for Autonomous Agents and Multiagent Systems, Singapore, May 2016.

[47] K. Judah, "Reinforcement learning via practice and critique advice," in *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence*, Atlanta, GA, USA, July 2010.

[48] B. Kim, "Learning from limited demonstrations," in *Proceedings of the Advances in Neural Information Processing Systems 26 (NIPS 2013)*, Lake Tahoe, NV, USA, December 2013.

[49] W. B. Knox, I. Fasel, and P. Stone, "Design principles for creating human-shapable agents," in *Proceedings of the AAAI Spring Symposium: Agents that Learn from Human Teachers*, Stanford, CA, USA, March 2009.

[50] S. Chernova and M. Veloso, "Interactive policy learning through confidence-based autonomy," *Journal of Artificial Intelligence Research*, vol. 34, no. 1, pp. 1–25, 2009.

[51] F. Melo and M. Lopes, "Multi-class generalized binary search for active inverse reinforcement learning," 2013, https://arxiv.org/abs/1306.3261.

[52] G. Kunapuli, "Guiding autonomous agents to better behaviors through human advice," in *Proceedings of the IEEE 13th International Conference on Data Mining workshops (ICDMW 2013)*, IEEE, Dallas, TX, USA, December 2013.

[53] P. Odom and S. Natarajan, "Active advice seeking for inverse reinforcement learning," in *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems*, International Foundation for Autonomous Agents and Multiagent Systems, Singapore, May 2016.

[54] M. N. Nicolescu and M. J. Mataric, "Natural methods for robot task learning: instructive demonstrations, generalization and practice," in *Proceedings of the Second International Joint Conference on Autonomous Agents and Multiagent Systems*, ACM, Melbourne, Victoria, July 2003.

[55] B. Argall, B. Browning, and M. Veloso, "Learning by demonstration with critique from a human teacher," in *Proceedings of the ACM/IEEE International Conference on Human-Robot Interaction*, ACM, Washington, DC, USA, March 2007.

[56] D. S. Brown, Y. Cui, and S. Niekum, "Risk-aware active inverse reinforcement learning," 2019, https://arxiv.org/abs/1903.09578.

[57] H. B. Suay, "Learning from demonstration for shaping through inverse reinforcement learning," in *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems*, International Foundation for Autonomous Agents and Multiagent Systems, Singapore, May 2016.

[58] D. S. Brown and S. Niekum, "Efficient probabilistic performance bounds for inverse reinforcement learning," in *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, New Orleans, LA, USA, February 2018.

[59] A. L. Thomaz and C. Breazeal, "Teachable robots: understanding human teaching behavior to build more effective robot learners," *Artificial Intelligence*, vol. 172, no. 6-7, pp. 716–737, 2008.

[60] F. Mondada, "The E-puck, a robot designed for education in engineering," in *Proceedings of the 9th Conference on Autonomous Robot Systems and Competitions*, IPCB: Instituto Politécnico de Castelo Branco, Castelo Branco, Portugal, May 2009.