# Assisted Inverse Reinforcement Learning

**Parameswaran Kamalaruban**
LIONS, EPFL

**Rati Devidze**
MPI-SWS

**Teresa Yeo**
LIONS, EPFL

**Trisha Mittal**
MPI-SWS

**Volkan Cevher**
LIONS, EPFL

**Adish Singla**
MPI-SWS

## Abstract

We study the problem of inverse reinforcement learning (IRL) with the added twist that the learner is assisted by a helpful teacher. More formally, we tackle the following question: How could a teacher provide an informative sequence of demonstrations to an IRL agent to speed up the learning process? We prove rigorous convergence guarantees of a new iterative teaching algorithm that adaptively chooses demonstrations based on the learner's current performance. Extensive experiments with a car driving simulator environment show that the learning progress can be speeded up drastically as compared to an uninformative teacher.

## 1 Introduction

Imitation Learning (IL), also known as Learning from Demonstrations, enables a learner to acquire new skills by observing a teacher's behavior. It plays an important role in many real-life learning settings, including human-to-human interaction [1, 2], and human-to-robot interaction [3, 4, 5, 6].

IL has been extensively studied in the context of designing efficient learning algorithms for a given set of demonstrations. The two popular approaches for IL include (i) behavioral cloning, which directly replicates the desired behavior [7], and (ii) inverse reinforcement learning (IRL), which infers the reward function explaining the desired behavior [8]. Despite recent advances in designing efficient IRL algorithms, there is little work on how to generate an optimal sequence of demonstrations.

Motivated by applications of intelligent tutoring systems to teach sequential decision-making tasks, such as surgical training or car driving, we study IL from the viewpoint of a teacher in order to best assist an IRL agent. Our work is inspired by real-life pedagogical settings, where it is evident that a carefully chosen demonstrations and tasks can considerably accelerate the learning progress [9]. For instance, consider a real-life scenario where a driving instructor wants to teach a student certain driving skills. The instructor can easily identify the mistakes/weaknesses of the student (e.g., unable to do rear parking), and then carefully choose tasks that student should perform, along with demonstrations to rectify any mistakes. However, the notion of helpful teaching is still not well understood computationally.

This work answers the following algorithmic challenge: How could a teacher provide an informative sequence of demonstrations/tasks to an IRL agent to speed up the learning process? Mathematically, we build an iterative machine teaching (IMT) framework for sequential decision-making tasks in *Markov Decision Processes* (MDP). An IMT framework assumes a "gradient" learner, and then teacher assists this learner by choosing the next training instance based on the current "parameter" of the learner [10].

The upshot of our approach is that we can transfer improvements in learning convergence from the IMT framework to the IRL setting. We note that, in a supervised learning setting (i.e., regression and classification tasks), [10] obtains an exponential improvement in convergence rate as compared to stochastic teacher who picks examples at random (more concretely, $\mathcal{O}\left(\log \frac{1}{\epsilon}\right)$ in comparison to $\mathcal{O}\left(\frac{1}{\epsilon^2}\right)$ to get $\epsilon$ close to the target).

Learning by Instruction Workshop, NIPS 2018. Do not distribute.

### 1.1 Our Approach

In this paper, we consider a learning agent ("learner") who is implementing an online IRL algorithm. In particular, we study an online variant of the popular IRL algorithm, namely Maximum Causal Entropy (MCE) IRL algorithm [11, 12, 13]).

Our key insight in designing our teaching algorithm is to reduce the problem of teaching a policy to that of obtaining an optimal hyperparameter in the learner's solution space. As in the IMT framework, we consider the teacher who knows the learning dynamics including learner's current parameters at any given time. We then prove rigorous convergence guarantees of our teaching algorithm and show that it can significantly reduce the number of demonstrations required to achieve a desired performance of the learner.

We perform extensive experiments in a synthetic learning environment inspired by a car driving simulator. While our model is conceptually simple, it is powerful enough to showcase significant improvements in speeding up the learning progress. In particular, our results show that the sequence of demonstrations picked by our algorithm exhibits a natural, interpretable curriculum: The teaching starts from "easier" tasks (e.g., driving on a free highway and avoiding HOV lane) and then proceeds to more "challenging" tasks (e.g., maintaining safe distance to pedestrians).

## 2 Problem Setup

### 2.1 Environment

Consider an MDP represented by $\mathcal{M} := (\mathcal{S}, \mathcal{A}, T, \gamma, P_0, \phi, R)$. The sets of possible states and actions are denoted by $\mathcal{S}$ and $\mathcal{A}$ respectively. $T : \mathcal{S} \times \mathcal{S} \times \mathcal{A} \to [0, 1]$ captures the state transition dynamics, i.e., $T(s' \mid s, a)$ denotes the probability of landing in state $s'$ by taking action $a$ from state $s$. Here $\gamma$ is the discounting factor, and $P_0 : \mathcal{S} \to [0, 1]$ is an initial distribution over states $\mathcal{S}$. We consider the reward functions $R : \mathcal{S} \to \mathbb{R}$ which are linear w.r.t. feature vector over states $\phi : \mathcal{S} \to \mathbb{R}^d$, e.g., $R(s) = \langle w, \phi(s) \rangle$ for some $w$ s.t. $\|w\|_1 \leq 1$.

We denote a policy $\pi : \mathcal{S} \times \mathcal{A} \to [0, 1]$ as a mapping from a state to a probability distribution over actions. For any policy $\pi$, the feature expectation vector and value of $\pi$ in the MDP $\mathcal{M}$ (with respect to the initial state $s \in \mathcal{S}$) are defined as follows:

$$\begin{aligned} \mu^{\pi,s} &:= \phi(s) + \gamma \sum_a \pi(a \mid s) \sum_{s'} T(s' \mid s, a) \mu^{\pi,s'} \in \mathbb{R}^d \\ V^{\pi,s} &:= R(s) + \gamma \sum_a \pi(a \mid s) \sum_{s'} T(s' \mid s, a) V^{\pi,s'} \in \mathbb{R} \end{aligned}$$

respectively. Then by taking expectation over $s \in \mathcal{S}$ with respect to the initial state distribution $P_0$, we have $\mu^\pi := \mathbb{E}_{s \sim P_0}[\mu^{\pi,s}]$, and $V^\pi := \mathbb{E}_{s \sim P_0}[V^{\pi,s}]$. Similarly, for any trajectory $\xi = \{(s_\tau, a_\tau)\}_{\tau=0,1,\ldots}$ representing a sequence of state-action pairs when executing a policy in the MDP $\mathcal{M}$, we define the following empirical counterparts of the above defined identities:

$$\begin{aligned} \mu^\xi &:= \sum_{\tau=0}^{\infty} \gamma^\tau \phi(s_\tau) \in \mathbb{R}^d \\ V^\xi &:= \sum_{\tau=0}^{\infty} \gamma^\tau R(s_\tau) \in \mathbb{R} \end{aligned}$$

Then for a collection of trajectories $\Xi = \{\xi_t\}_{t=1,2,\ldots}$, we have $\mu^\Xi := \frac{1}{|\Xi|} \sum_t \mu^{\xi_t}$, and $V^\Xi := \frac{1}{|\Xi|} \sum_t V^{\xi_t}$.

### 2.2 Interaction between Learner and Teacher

We consider a sequential IRL learner (cf., next section for details) who knows the MDP $\mathcal{M}$, but not the reward function $R$, i.e., has only access to $\mathcal{M} \backslash R$. In addition, we have a helpful teacher with full access to the MDP $\mathcal{M}$ who would assist the learner by providing informative demonstrations. For the presentation of the main results, we assume that the teacher knows the learning dynamics including

**Algorithm 1:** Learning via Demonstrations

```
Initialization: initial policy $\pi_1$
for $t = 1, 2, \dots, T$ do
    teacher provides a demonstration $\xi_t$ to the learner
    learner updates the policy $\pi_{t+1}$ using $\pi_t$ and $\xi_t$
end
Output: compute policy $\pi^L$ from $\pi_1, \dots, \pi_{T+1}$
```

learner's current parameters at any given time. The learner receives a sequence of demonstrations $\Xi = \{\xi_t\}_{t=1,2,\dots}$ from the teacher and outputs a policy $\pi^L$ such that $V^{\pi^L}$ is high. The learner-teacher interaction is described formally in Algorithm 1.

### 2.3 Teaching Objective

Let $\pi^E$ be the *target policy* (e.g., an optimal policy in the MDP $\mathcal{M}$) that we want to teach to the learner. The performance of the learner's policy $\pi^L$ (w.r.t. $\pi^E$) in $\mathcal{M}$ is evaluated via the following measures (for some fixed $\epsilon > 0$):

1. $\left|V^{\pi^L} - V^{\pi^E}\right| \leq \epsilon$, ensuring high reward.
2. $\left\|\mu^{\pi^L} - \mu^{\pi^E}\right\|_2 \leq \epsilon$, ensuring robustness for any linear reward functions of the form $R(s) = \langle w, \phi(s) \rangle$ (with some unknown parameter $w$ such that $\|w\|_1 \leq 1$) [14].

If the above conditions are satisfied (which is the objective of the learning agent), we say that the learner's policy $\epsilon$-*approximate* the target policy. In this paper, we study this problem from the viewpoint of a teacher in order to provide a near-optimal sequence of demonstrations $\{\xi_t\}_{t=1,2,\dots}$ to the learner, to achieve the desired objective. The teacher's performance is then measured by the number of demonstrations required to achieve the learner's objective.

## 3 Learner's Model and Algorithms

In this work, we consider Maximum Causal Entropy based Inverse Reinforcement Learning (MCE-IRL, [11, 12]) algorithm for the learner. The MCE-IRL approach can be interpreted (in dual form) as maximizing the causal likelihood of the demonstration data.

### 3.1 Background on (Batch) MCE-IRL Algorithm

Given a set of expert demonstrations $\Xi = \{\xi_t\}_{t=1,2,\dots}$, the MCE-IRL algorithm attempts to infer the underlying reward function by solving the following problem:

$$\underset{\lambda}{\text{maximize}} \;\; c(\lambda; \Xi) \;:=\; \sum_t \sum_\tau \log P_\lambda(a_{t,\tau} \mid s_{t,\tau}) \tag{1}$$

where

$$P_\lambda(a \mid s) \;=\; \frac{Z_{a|s,\lambda}}{Z_{s,\lambda}}; \quad \log Z_{s,\lambda} \;=\; \log \sum_a Z_{a|s,\lambda}; \quad \log Z_{a|s,\lambda} \;=\; \langle \lambda, \phi(s) \rangle + \gamma \sum_{s'} T(s' \mid s, a) \log Z_{s',\lambda}. \tag{2}$$

Consider that the learner's current parameter is $\lambda_j$, and $\pi^{\lambda_j}$ is the optimal policy for $\mathcal{M} \backslash R$ with $R(s) = \langle \lambda_j, \phi(s) \rangle$ computed via Soft-Value-Iteration procedure, cf. [12, Algorithm. 9.1]. Then, the gradient of the above concave optimization problem is $g_j = \mu^{\Xi} - \mu^{\pi^{\lambda_j}} \in \partial c(\lambda_j; \Xi)$ [12, Eq. 10.1]. The gradient descent update rule is given by:

$$\lambda_{j+1} \;\leftarrow\; \lambda_j - \eta_j g_j \;=\; \lambda_j - \eta_j \left(\mu^{\pi_j} - \mu^{\Xi}\right), \tag{3}$$

where $\pi_j \;:= \pi^{\lambda_j}$ and $\eta_j$ denotes the current learning rate.

**Algorithm 2:** Sequential MCE-IRL [12, 13]

```
Initialization: $\lambda_1$, $\pi_1$
for $t = 1, 2, \dots, T$ do
    teacher provides a demonstration $\xi_t$ with starting state $s_{t,0}$ to the learner
    $\lambda_{t+1} \leftarrow \Pi_\Omega \left[\lambda_t - \eta_t \left(\mu^{\pi_t, s_{t,0}} - \mu^{\xi_t}\right)\right]$
    $\pi_{t+1} \leftarrow$ Soft-Value-Iteration $(\mathcal{M} \backslash R, \lambda_{t+1})$
end
Output: compute policy $\pi^L$ from $\pi_1, \dots, \pi_{T+1}$
```

## 3.2 Sequential MCE-IRL Algorithm

As discussed in the problem setup, our work considers a sequential IRL learner. In fact, the full gradient update rule given in (3) can be naturally extended to an online variant [12, 13], where, at every time step $t$, the learner receives a demonstration $\xi_t$ with starting state $s_{t,0}$. We consider the following online projected gradient descent update:

$$\lambda_{t+1} \;\leftarrow\; \Pi_\Omega \left[\lambda_t - \eta_t \tilde{g}_t\right], \tag{4}$$

where

$$\tilde{g}_t \;=\; \mu^{\pi_t, s_{t,0}} - \mu^{\xi_t}; \qquad \Omega \;=\; \left\{\lambda : \left\|\lambda\right\|_2 \leq r\right\},$$

for large enough $r$ s.t. $\lambda^* \in \Omega$. Then the resulting sequential MCE-IRL algorithm is given in Algorithm 2.

In the following sections, we present different teaching algorithms for the sequential MCE-IRL learner (Algorithm 2), and provide the convergence guarantees for them. The teachers differ in the way they choose the demonstration $\xi_t$ (with starting state $s_{t,0}$) at every time step $t$.

## 4 Agnostic Teaching

In this section, we present a natural agnostic teaching algorithm for the sequential MCE-IRL learner discussed above. We begin by considering the following stochastic optimization formulation of the batch MCE-IRL (1):

$$\min_\lambda c(\lambda) := \mathop{\mathbb{E}}_{\{(s_\tau, a_\tau)\}_\tau \sim (\pi^E, \mathcal{M})} \left[-\sum_\tau \log P_\lambda \left(a_\tau \mid s_\tau\right)\right], \tag{5}$$

where $P_\lambda$ is given by (2).

Next, we study the performance (w.r.t. the loss function $c(\lambda)$) of the sequential MCE-IRL learner when learning from an agnostic/uninformative teacher. This agnostic teacher provides a demonstration at random by executing the policy $\pi^E$ in the MDP $\mathcal{M}$, i.e., $\xi_t \sim (\pi^E, \mathcal{M})$. For this setting, we have,

$$\mathbb{E}_{\xi_t}\left[\tilde{g}_t\right] = \mathbb{E}_{s_{t,0}, \xi_t}\left[\mu^{\pi_t, s_{t,0}} - \mu^{\xi_t}\right] = \mu^{\pi_t} - \mu^{\pi^E} \in \partial c\left(\lambda_t\right).$$

Then for this agnostic teacher (given by Algorithm 3), we have the following convergence guarantee:

**Theorem 1.** *For the agnostic teaching strategy given in Algorithm 3, and $T = \mathcal{O}\left(\frac{1}{\epsilon^2}\right)$, we have* $\mathbb{E}\left[c\left(\frac{1}{T}\sum_{t=1}^T \lambda_{t+1}\right) - \min_\lambda c(\lambda)\right] \leq \epsilon$.

Without any further information about the learner, a teacher can do no better than providing random demonstrations. Next, we present an informative teaching strategy (which exploits the knowledge of the target policy and the learning dynamics), and show that it substantially improves upon the convergence rate given above.

## 5 Assistive Teaching

In this section, we design a teaching strategy to provide a more informative demonstration $\xi_t$ by leveraging the knowledge of the target policy and learner's dynamics. The main idea is to pick the

**Algorithm 3:** Sequential MCE-IRL with Agnostic Teacher

```
Initialization: $\lambda_1$, $\pi_1$
for $t = 1, 2, \dots, T$ do
    teacher provides a demonstration $\xi_t \overset{iid}{\sim} (\pi^E, \mathcal{M})$ to the learner
    $\lambda_{t+1} \leftarrow \Pi_\Omega \left[\lambda_t - \eta_t \left(\mu^{\pi_t, s_{t,0}} - \mu^{\xi_t}\right)\right]$
    $\pi_{t+1} \leftarrow$ Soft-Value-Iteration $(\mathcal{M} \backslash R, \lambda_{t+1})$
end
$\lambda_{\text{avg}} \leftarrow \frac{1}{T}\sum_{t=1}^{T} \lambda_{t+1}$
$\pi^{\lambda_{\text{avg}}} \leftarrow$ Soft-Value-Iteration $(\mathcal{M} \backslash R, \lambda_{\text{avg}})$
Output: policy $\pi^L \leftarrow \pi^{\lambda_{\text{avg}}}$
```

**Algorithm 4:** Sequential MCE-IRL with Assistive Teacher

```
Initialization: $\lambda_1$, $\pi_1$
for $t = 1, 2, \dots, T$ do
    teacher provides a demonstration $\xi_t$ satisfying (7) to the learner
    $\lambda_{t+1} \leftarrow \Pi_\Omega \left[\lambda_t - \eta_t \left(\mu^{\pi_t, s_{t,0}} - \mu^{\xi_t}\right)\right]$
    $\pi_{t+1} \leftarrow$ Soft-Value-Iteration $(\mathcal{M} \backslash R, \lambda_{t+1})$
end
Result: policy $\pi^L \leftarrow \pi_{T+1}$
```

demonstration which minimizes the distance between the IRL agent's current parameter ($\lambda_t$) and the target hyperparameter ($\lambda^*$), i.e., minimize $\|\lambda_{t+1} - \lambda^*\| = \|\lambda_t - \eta_t \tilde{g}_t - \lambda^*\|$. Consider the following unconstrained optimization problem for selecting an informative demonstration at time $t$:

$$\min_{s,\xi} \ \eta_t^2 \left\|\mu^{\pi_t, s} - \mu^{\xi}\right\|^2 - 2\eta_t \left\langle \lambda_t - \lambda^*, \mu^{\pi_t, s} - \mu^{\xi} \right\rangle, \tag{6}$$

where $s$ is the starting state of $\xi$. The optimal solution for the above problem is a synthetic trajectory $\xi_t^{\text{syn}}$ (with starting state $s_{t,0}^{\text{syn}}$) whose feature expectation vector satisfies:

$$\mu^{\xi_t^{\text{syn}}} \ = \ \mu^{\pi_t, s_{t,0}^{\text{syn}}} - \beta_t \left(\lambda_t - \lambda^*\right),$$

for some $\beta_t \in \left[0, \frac{1}{\eta_t}\right]$, and $\left\|\mu^{\xi_t^{\text{syn}}}\right\|_2 \leq h_{\max}\sqrt{d}$.

Ideally, we would like to provide $\xi_t^{\text{syn}}$ as the next demonstration. However, in real-world applications, the teacher would naturally have some constraints (e.g., posed by the physical environment or teacher's policy) on what demonstrations can be provided. The assistive teacher looks for a feasible demonstration $\xi_t$ whose feature expectation vector $\mu^{\xi_t}$ is closest to $\mu^{\xi_t^{\text{syn}}}$. For instance, the teacher could either generate such a feasible demonstration $\xi_t$ on the fly (e.g., in a virtual simulation-based teaching environment) or select it from a pre-defined pool of feasible demonstrations.

We say that a teacher is $\Delta_{\max}$-*powerful*, if for every $t$, the teacher can generate a demonstration $\xi_t$ that satisfies:

$$\mu^{\xi_t} \ = \ \mu^{\xi_t^{\text{syn}}} + \delta_t, \tag{7}$$

for some $\delta_t$ s.t. $\|\delta_t\|_2 \leq \Delta_{\max}$. The resulting assistive teaching strategy is given in Algorithm 4.

The following theorem shows that this teaching strategy can significantly speedup the learning progress compared to the agnostic teacher (cf., Theorem 1).

**Theorem 2.** *Given $\epsilon' > 0$, let the teacher be $\Delta_{\max}$-powerful with $\Delta_{\max} := \frac{\epsilon'^2 \beta^2}{4\eta_{\max}[4(1-\beta)r+1]}$ (where $\beta = \min_t \eta_t\beta_t$, and $\eta_{\max} = \max_t \eta_t$). Then for the assistive teaching strategy given in Algorithm 4, and $T = \mathcal{O}\left(\log \frac{1}{\epsilon'}\right)$, we have $\|\lambda_T - \lambda^*\| \leq \epsilon'$.*

Note that the above theorem only guarantees the convergence to the target hyperparameter ($\lambda^*$). In the extended version of this paper, we show that by exploiting the *smoothness* of the MDP $\mathcal{M}$, the convergence to the target hyperparameter guarantees the convergence to the target feature expectation vector.

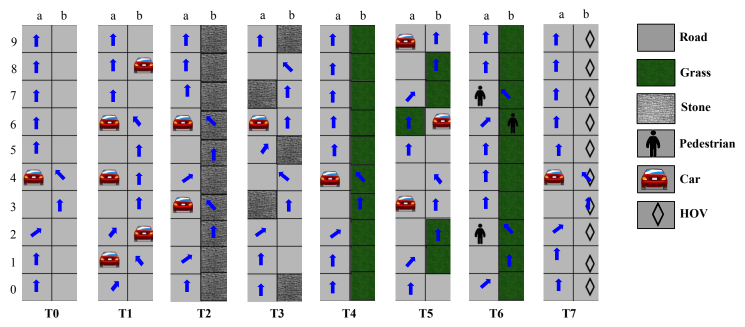


Figure 1: Car simulator environment with 8 distinct lanes (tasks). In any given lane, an agent starts from the bottom-left corner and the goal is to reach the top of the lane. Blue arrows represent the path taken by the teacher's optimal policy.

| $\phi(s)$ | road | stone | grass | car | ped | HOV | car1 | ped1 | nocar1 | noped1 |
|---|---|---|---|---|---|---|---|---|---|---|
| $w$ | 1 | -1 | -0.5 | -5 | -10 | -1 | -2 | -5 | 0 | 0 |

Table 1: Weight vector (without normalization) specifying the linear reward function $R(s) = \langle w, \phi(s)\rangle$. Teacher's optimal policy w.r.t. this reward function is illustrated via the blue arrows in Figure 1 when driving in this environment.

## 6 Experimental Evaluation

In this section, we demonstrate the performance of our algorithm in a learning environment inspired by a car driving simulator application [15].

### 6.1 Environment Setup

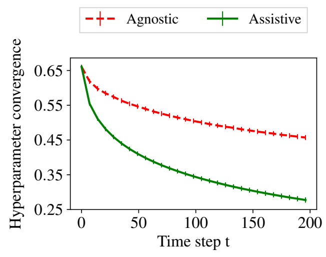
Figure 1 shows the car driving simulator environment that we use in our experiments. It consists of 8 different lanes (henceforth referred as tasks), denoted as T0, T1, T2, T3, T4, T5, T6, and T7. The total number of states in the MDP is 160 where each cell represents a state, and each task is associated with 20 states. There are 8 initial states corresponding to the bottom left cell of each task given by $\{\mathtt{T0}_{0,a}, \mathtt{T1}_{0,a}, \mathtt{T2}_{0,a}, \mathtt{T3}_{0,a}, \mathtt{T4}_{0,a}, \mathtt{T5}_{0,a}, \mathtt{T6}_{0,a}, \mathtt{T7}_{0,a}\}$. The agent can be imagined as driving a car and her goal is to navigate in this environment, starting from an initial state to the top end of the lane. The agent can take three different actions: {left, straight, right}. Action left steers the agent to the left of the current lane (or stay in the same lane if agent is already in the leftmost lane) and action right steers to the right of the current lane (or stay in the same lane if agent is already in the rightmost lane). Irrespective of the action taken by the agent, the agent always move forward. For instance, consider that the agent's current state is the $\mathtt{T2}_{4,a}$ grid cell; then, the agent's next states corresponding these three actions would be $\mathtt{T2}_{5,a}$, $\mathtt{T2}_{5,a}$, and $\mathtt{T2}_{5,b}$. W.l.o.g., we consider that only the agent moves in this environment and rest everything (i.e., other cars and pedestrians) is static.

We use $d = 10$ features to define the state space $\phi(s)$. These features are of two types:

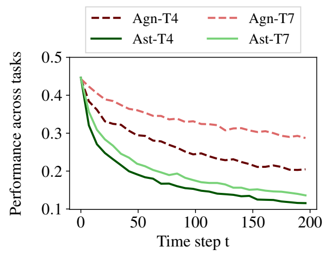
1. features indicating the type of the current grid cell as road, stone, grass, car, ped, and HOV;
2. features providing some look-ahead information about the upcoming grid cells such as if there is a car or pedestrian in the immediate front cell (denoted as car1 and ped1) or otherwise (denoted as nocar1 and noped1)

The weight vector (without normalization) specifying the linear reward function $R(s) = \langle w, \phi(s)\rangle$ is shown in Table 1. Teacher's policy is then computed as the optimal policy w.r.t. this reward function and is illustrated via blue arrows in Figure 1 representing the path taken by the teacher when driving in this environment.

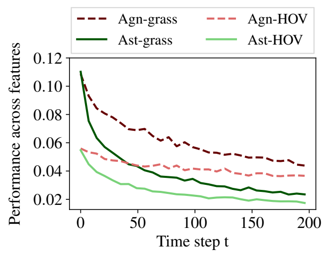
Each of these tasks is associated with different driving skills. For instance, task T0 corresponds to the most basic setup representing a traffic-free highway—it has a very small probability of the presence



(a) Convergence of the learner's hyperparameter $\lambda_t$ to the target $\lambda^*$

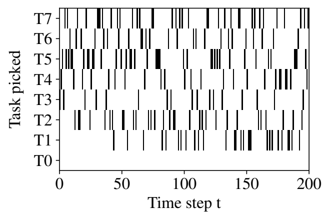


(b) Convergence of the feature expectation $\mu$ for different tasks

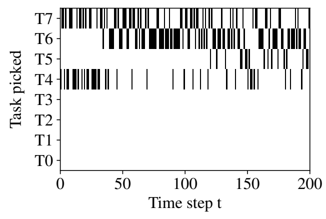


(c) Convergence of the feature expectation $\mu$ for different features

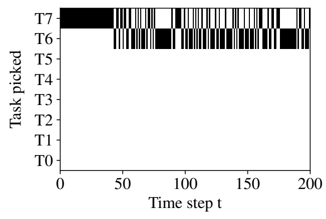
Figure 2: Convergence plots of our assistive teaching algorithm in comparison to the agnostic teacher.



(a) Learner's initial skills: `T0`



(b) Learner's initial skills: `T0`–`T3`



(c) Learner's inital skills: `T0`–`T5`

Figure 3: Teaching curriculum (i.e., the task associated with the picked state $s_{t,0}$ in Algorithm 4) for three different settings depending on the learner's initial skills trained on (a) `T0`, (b) `T0`–`T3`, and (c) `T0`–`T5`.

of another car. However, the task `T1` represents a crowded highway with 0.25 probability of grid cell being a car, and the task `T7` contains an HOV lane that the agent should avoid. Task `T2` has stones on the right lane, whereas task `T3` has a mix of both cars and stones. Similarly, tasks `T4` has grass on the right lane, and `T5` has a mix of both grass and cars. Task `T6` also has pedestrians which the agent should maintain distance to and avoid hitting.

## 6.2 Experimental Results

Next, we report the experimental results where we compare the performance of different teaching algorithms and analyze the teaching curriculum when varying the learner's initial skills. We consider the learning agent implementing a sequential MCE-IRL algorithm (Algorithm 2) and results are reported by averaging over 30 runs.

### 6.2.1 Convergence of algorithms

Here, we consider the following setup: The learner is initially trained based on demonstrations sampled only from the tasks `T0`, `T1`, `T2`, and `T3`. Intuitively, the learner's initial policy $\pi^{\lambda_1}$ (corresponding to hyperparameter $\lambda_1$) possesses skills to avoid collisions with cars and to avoid hitting stones while driving. We expect to teach three major skills to the learner, i.e., avoiding grass while driving (`T4`, `T5`, and `T6`), maintaining distance to pedestrians (`T6`), and not to drive on an HOV lane (`T7`).

Figure 2a shows the convergence of the learner's current hyperparameter $\lambda_t$ to the target hyperparameter $\lambda^*$, i.e., $\|\lambda_t - \lambda^*\|_2$ for both the assistive teacher and the agnostic teacher. In Figure 2b, we consider the convergence of the feature expectation vector of the learner's current policy $\mu^{\pi_t}$ to the target feature expectation vector $\mu^{\pi^{\lambda^*}}$. More specifically, we compute this convergence of the feature expectations separately for the tasks `T4` and `T7`, which is given by $\left\|\mu^{\pi_t,s} - \mu^{\pi^{\lambda^*},s}\right\|_2$ where $s = \texttt{T4}_{0,a}$ and $s = \texttt{T7}_{0,a}$ respectively. Similarly, in Figure 2c, we compare the convergence in feature expectation vectors, but only considering the components associated with the features `grass` and

HOV. These convergence plots illustrate that the assistive teaching algorithm significantly outperforms the agnostic teaching in terms of convergence to the target hyperparameter, as well as in acquiring new skills and improving performance across tasks. In Figure 2b, we see that the gap between the assistive and agnostic teacher is even more drastic for the task T7 compared to the task T4—in general, the assistance becomes even more important when learning more rarely occurring skills.

#### 6.2.2 Teaching curriculum

In Figure 3, we compare the teaching curriculum of the asisstive teacher for different settings depending on the learner's initial skills trained on (a) task T0, (b) tasks T0–T3, and (c) tasks T0–T5. The curriculum here refers to the task associated with the state $s_{t,0}$ picked by the teacher at time $t$ to provide the demonstration (cf., Algorithm 4). In these plots, we can see specific structures and temporal patterns emerging in the curriculum. First, we can see that the teaching curriculum focuses on tasks that help the learner acquire new skills. For instance, in Figure 3b, the teacher primarily picks tasks that provide new skills corresponding to the features grass, HOV, and ped. Second, an interesting temporal pattern in the curriculum can be seen in Figure 3c where the learner needs to acquire two skills about maintaining distance to pedestrians and not driving in the HOV lane. Here, the presence of pedestrians is a rarer event compared to the HOV lane ($10\%$ of grid cells in T6 are ped, compared to $50\%$ HOV grid cells in T7). Consequently, the curriculum begins with the “easier" task T7 by first exclusively teaching the HOV feature, and then over time, additionally includes demonstrations from the more “challenging" task T6.

### 6.3 Related Work

#### 6.3.1 Inverse Reinforcement Learning

In the inverse reinforcement learning (IRL), an agent observes the behavior (batch of demonstrations) of an expert in an MDP environment with an unknown reward function and attempts to infer the weight parameters of the reward function. We refer the reader to a recent survey by [16] providing an algorithmic perspective and recent results on IRL algorithms. While a lot of work in IRL is model-based (cf., [15, 14, 17, 11], recently, new algorithms have been proposed for model-free setting (cf., [18, 19]). [20] have studied the value alignment problem in a game-theoretic setup, and provided an approximate scheme to generate instructive demonstrations for an IRL agent. In our work, we devise a systematic procedure (with convergence guarantees) to choose an optimal sequence of demonstrations, by taking into account the learning dynamics.

A somewhat different but related problem setting is that of *reward shaping* where the goal is to modify/design the reward function to guide the learning agent [21, 22]. There has also been work on designing active/interactive IRL algorithms that focus on reducing the number of demonstrations that need to be requested from a teacher (cf., [23]). However, the key difference in our approach is that we take the viewpoint of a teacher in how to best assist the learning agent by providing an optimal sequence of demonstrations.

#### 6.3.2 Algorithmic Teaching

Another line of research relevant to our work is that of algorithmic teaching. Here, one studies the interaction between a teacher and a learner where the teacher's objective is to find an optimal training sequence to steer the learner towards a desired goal [24, 10]. Algorithmic teaching provides a rigorous formalism for a number of real-world applications such as personalized education and intelligent tutoring systems [25, 26], social robotics [27, 28], and human-in-the-loop systems [29, 30].

[27] have studied the problem of teaching an IRL agent in the batch setting, i.e., the teacher has to provide a near-optimal set of demonstrations at once. They considered the IRL algorithm from [15], which could only result in inferring an equivalent class of reward weight parameters for which the observed behavior is optimal. In a recent work, contemporary to ours, [31] have extended the previous work of [27] by showing that the teaching problem can be formulated as a set cover problem. However, their teaching strategy does not take into account how the learner progresses (i.e., it is non-interactive). In contrast, we study an interactive teaching setting to teach a sequential MCE-IRL algorithm [11, 13]. In another recent work, contemporary to ours, [32] have studied the problem of teaching an IRL agent, however they consider a different setting where there is a model mismatch between the teacher and the learner.

## References

[1] Daphna Buchsbaum, Alison Gopnik, Thomas L Griffiths, and Patrick Shafto. Children's imitation of causal action sequences is influenced by statistical and pedagogical evidence. *Cognition*, 120(3):331–340, 2011.

[2] Patrick Shafto, Noah D Goodman, and Thomas L Griffiths. A rational account of pedagogical reasoning: Teaching by, and learning from, examples. *Cognitive psychology*, 71:55–89, 2014.

[3] Stefan Schaal. Learning from demonstration. In *NIPS*, pages 1040–1046, 1997.

[4] Aude Billard, Sylvain Calinon, Ruediger Dillmann, and Stefan Schaal. Robot programming by demonstration. In *Springer handbook of robotics*, pages 1371–1394. Springer, 2008.

[5] Brenna D Argall, Sonia Chernova, Manuela Veloso, and Brett Browning. A survey of robot learning from demonstration. *Robotics and autonomous systems*, 57(5):469–483, 2009.

[6] Sonia Chernova and Andrea L Thomaz. Robot learning from human teachers. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 8(3):1–121, 2014.

[7] Michael Bain and Claude Sommut. A framework for hehavioural claning. *Machine Intelligence 15*, 15:103, 1999.

[8] Stuart Russell. Learning agents for uncertain environments. In *Proceedings of the eleventh annual conference on Computational learning theory*, pages 101–103. ACM, 1998.

[9] Mark K Ho, Michael Littman, James MacGlashan, Fiery Cushman, and Joseph L Austerweil. Showing versus doing: Teaching by demonstration. In *NIPS*, pages 3027–3035, 2016.

[10] Weiyang Liu, Bo Dai, Ahmad Humayun, Charlene Tay, Chen Yu, Linda B. Smith, James M. Rehg, and Le Song. Iterative machine teaching. In *ICML*, pages 2149–2158, 2017.

[11] Brian D Ziebart, Andrew L Maas, J Andrew Bagnell, and Anind K Dey. Maximum entropy inverse reinforcement learning. In *AAAI*, volume 8, pages 1433–1438, 2008.

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

[13] Nicholas Rhinehart and Kris M Kitani. First-person activity forecasting with online inverse reinforcement learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3696–3705, 2017.

[14] Pieter Abbeel and Andrew Y Ng. Apprenticeship learning via inverse reinforcement learning. In *ICML*, page 1. ACM, 2004.

[15] Andrew Y Ng, Stuart J Russell, et al. Algorithms for inverse reinforcement learning. In *Icml*, pages 663–670, 2000.

[16] Takayuki Osa, Joni Pajarinen, Gerhard Neumann, J Andrew Bagnell, Pieter Abbeel, Jan Peters, et al. An algorithmic perspective on imitation learning. *Foundations and Trends® in Robotics*, 7(1-2):1–179, 2018.

[17] Nathan D Ratliff, J Andrew Bagnell, and Martin A Zinkevich. Maximum margin planning. In *ICML*, pages 729–736. ACM, 2006.

[18] Abdeslam Boularias, Jens Kober, and Jan Peters. Relative entropy inverse reinforcement learning. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pages 182–189, 2011.

[19] Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. In *NIPS*, pages 4565–4573, 2016.

[20] Dylan Hadfield-Menell, Stuart J Russell, Pieter Abbeel, and Anca Dragan. Cooperative inverse reinforcement learning. In *NIPS*, pages 3909–3917, 2016.

[21] Jonathan Sorg, Satinder P. Singh, and Richard L. Lewis. Reward design via online gradient ascent. In *NIPS*, pages 2190–2198, 2010.

[22] Andrew Y Ng, Daishi Harada, and Stuart Russell. Policy invariance under reward transformations: Theory and application to reward shaping. In *ICML*, volume 99, pages 278–287, 1999.

[23] Kareem Amin, Nan Jiang, and Satinder P. Singh. Repeated inverse reinforcement learning. In *NIPS*, pages 1813–1822, 2017.

[24] Xiaojin Zhu, Adish Singla, Sandra Zilles, and Anna N. Rafferty. An overview of machine teaching. *CoRR*, abs/1801.05927, 2018.

[25] Anna N Rafferty, Emma Brunskill, Thomas L Griffiths, and Patrick Shafto. Faster teaching via pomdp planning. *Cognitive science*, 40(6):1290–1332, 2016.

[26] Kaustubh R Patil, Xiaojin Zhu, Łukasz Kopeć, and Bradley C Love. Optimal teaching for limited-capacity human learners. In *NIPS*, pages 2465–2473, 2014.

[27] Maya Cakmak, Manuel Lopes, et al. Algorithmic and human teaching of sequential decision tasks. In *AAAI*, 2012.

[28] Maya Cakmak and Andrea L Thomaz. Eliciting good teaching from humans for machine learners. *Artificial Intelligence*, 217:198–215, 2014.

[29] Adish Singla, Ilija Bogunovic, Gábor Bartók, Amin Karbasi, and Andreas Krause. Near-optimally teaching the crowd to classify. In *ICML*, pages 154–162, 2014.

[30] Adish Singla, Ilija Bogunovic, G Bartók, A Karbasi, and A Krause. On actively teaching the crowd to classify. In *NIPS Workshop on Data Driven Education*, 2013.

[31] Daniel S. Brown and Scott Niekum. Machine teaching for inverse reinforcement learning: Algorithms and applications. *CoRR*, abs/1805.07687, 2018.

[32] Luis Haug, Sebastian Tschiatschek, and Adish Singla. Teaching inverse reinforcement learners via features and demonstrations. In *NIPS*, 2018.