Learner-aware Teaching: Inverse Reinforcement Learning with Preferences and Constraints

Sebastian Tschiatschek* Microsoft Research setschia@microsoft.com

Ahana Ghosh* MPI-SWS gahana@mpi-sws.org

Luis Haug* ETH Zurich lhaug@inf.ethz.ch

Rati Devidze MPI-SWS rdevidze@mpi-sws.org

Adish Singla MPI-SWS adishs@mpi-sws.org

Abstract

Inverse reinforcement learning (IRL) enables an agent to learn complex behavior by observing demonstrations from a (near-)optimal policy. The typical assumption is that the learner's goal is to match the teacher's demonstrated behavior. In this paper, we consider the setting where the learner has its own preferences that it additionally takes into consideration. These preferences can for example capture behavioral biases, mismatched worldviews, or physical constraints. We study two teaching approaches: *learner-agnostic* teaching, where the teacher provides demonstrations from an optimal policy ignoring the learner's preferences, and *learner-aware* teaching, where the teacher accounts for the learner's preferences. We design learner-aware teaching algorithms and show that significant performance improvements can be achieved over learner-agnostic teaching.

1 Introduction

Inverse reinforcement learning (IRL) enables a learning agent (*learner*) to acquire skills from observations of a *teacher*'s demonstrations. The learner infers a reward function explaining the demonstrated behavior and optimizes its own behavior accordingly. IRL has been studied extensively [Abbeel and Ng, 2004, Ratliff et al., 2006, Ziebart, 2010, Boularias et al., 2011, Osa et al., 2018] under the premise that the learner can and is willing to imitate the teacher's behavior.

In real-world settings, however, a learner typically does not blindly follow the teacher's demonstrations, but also has its own preferences and constraints. For instance, consider demonstrating to an auto-pilot of a self-driving car how to navigate from A to B by taking the most fuel-efficient route. These demonstrations might conflict with the preference of the auto-pilot to drive on highways in order to ensure maximum safety. Similarly, in robot-human interaction with the goal of teaching people how to cook, a teaching robot might demonstrate to a human user how to cook "roast chicken", which could conflict with the preferences of the learner who is "vegetarian". To give yet another example, consider a surgical training simulator which provides virtual demonstrations of expert behavior; a novice learner might not be confident enough to imitate a difficult procedure because of safety concerns. In all these examples, the learner might not be able to acquire useful skills from the teacher's demonstrations.

In this paper, we formalize the problem of teaching a learner with preferences and constraints. First, we are interested in understanding the suboptimality of *learner-agnostic* teaching, i.e., ignoring the learner's preferences. Second, we are interested in designing *learner-aware* teachers who account

*Authors contributed equally to this work.

33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.

for the learner's preferences and thus enable more efficient learning. To this end, we study a learner model with preferences and constraints in the context of the Maximum Causal Entropy (MCE) IRL framework [Ziebart, 2010, Ziebart et al., 2013, Zhou et al., 2018]. This enables us to formulate the teaching problem as an optimization problem, and to derive and analyze algorithms for learner-aware teaching. Our main contributions are:

I We formalize the problem of IRL under preference constraints (Section 2 and Section 3).

II We analyze the problem of optimizing demonstrations for the learner when preferences are *known* to the teacher, and we propose a bilevel optimization approach to the problem (Section 4).

III We propose strategies for adaptively teaching a learner with preferences *unknown* to the teacher, and we provide theoretical guarantees under natural assumptions (Section 5).

IV We empirically show that significant performance improvements can be achieved by learner-aware teachers as compared to learner-agnostic teachers (Section 6).

2 Problem Setting

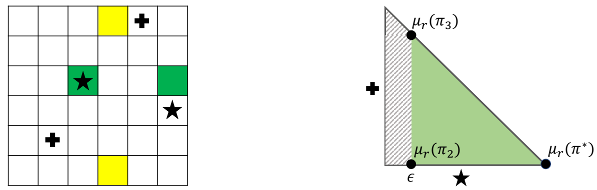
Environment. Our environment is described by a *Markov decision process* (MDP) $\mathcal{M} := (\mathcal{S}, \mathcal{A}, T, \gamma, P_0, R)$. Here $\mathcal{S}$ and $\mathcal{A}$ denote finite sets of states and actions. $T\colon \mathcal{S} \times \mathcal{S} \times \mathcal{A} \to [0,1]$ describes the state transition dynamics, i.e., $T(s'|s,a)$ is the probability of landing in state s' by taking action a from state s. $\gamma \in (0,1)$ is the discounting factor. $P_0 : \mathcal{S} \to [0,1]$ is an initial distribution over states. $R : \mathcal{S} \to \mathbb{R}$ is the reward function. We assume that there exists a feature map $\phi_r\colon \mathcal{S} \to [0,1]^{d_r}$ such that the reward function is linear, i.e., $R(s) = \langle \mathbf{w}_r^*, \phi_r(s)\rangle$ for some $\mathbf{w}_r^* \in \mathbb{R}^{d_r}$. Note that a bound of $\|\mathbf{w}_r^*\|_1 \leq 1$ ensures that $|R(s)| \leq 1$ for all s.

Basic definitions. A *policy* is a map $\pi : \mathcal{S} \times \mathcal{A} \to [0,1]$ such that $\pi(\,\cdot \mid s)$ is a probability distribution over actions for every state s. We denote by Π the set of all such policies. The performance measure for policies we are interested in is the *expected discounted reward* $R(\pi) := \mathbb{E}\left(\sum_{t=0}^{\infty} \gamma^t R(s_t)\right)$, where the expectation is taken with respect to the distribution over trajectories $\xi = (s_0, s_1, s_2, \ldots)$ induced by π together with the transition probabilities T and the initial state distribution P_0. A policy π is *optimal* for the reward function R if $\pi \in \arg\max_{\pi' \in \Pi} R(\pi')$, and we denote an optimal policy by π^*. Note that $R(\pi) = \langle \mathbf{w}_r^*, \mu_r(\pi)\rangle$, where $\mu_r\colon \Pi \to \mathbb{R}^{d_r}$, $\pi \mapsto \mathbb{E}\left(\sum_{t=0}^{\infty} \gamma^t \phi_r(s_t)\right)$, is the map taking a policy to its vector of *(discounted) feature expectations*. We denote by $\Omega_r = \{\mu_r(\pi) : \pi \in \Pi\}$ the image $\mu_r(\Pi)$ of this map. Note that the set $\Omega_r \in \mathbb{R}^{d_r}$ is convex (see [Ziebart, 2010, Theorem 2.8] and [Abbeel and Ng, 2004]), and also bounded due to the discounting factor $\gamma \in (0,1)$. For a finite collection of trajectories $\Xi = \{s_0^i, s_1^i, s_2^i, \ldots\}_{i=1,2,\ldots}$ obtained by executing a policy π in the MDP $\mathcal{M}$, we denote the empirical counterpart of $\mu_r(\pi)$ by $\hat{\mu}_r(\Xi) := \frac{1}{|\Xi|}\sum_i \sum_t \gamma^t \phi_r(s_t^i)$.

An IRL learner and a teacher. We consider a learner L implementing an inverse reinforcement learning (IRL) algorithm and a teacher T. The teacher has access to the full MDP $\mathcal{M}$; the learner knows the MDP and the parametric form of reward function $R(s) = \langle \mathbf{w}_r, \phi_r(s)\rangle$ but does not know the true reward parameter $\mathbf{w}_r^*$. The learner, upon receiving demonstrations from the teacher, outputs a policy π^{L} using its algorithm. The teacher's objective is to provide a set of demonstrations Ξ^{T} to the learner that ensures that the learner's output policy π^{L} achieves high reward $R(\pi^{\mathsf{L}})$.

The standard IRL algorithms are based on the idea of *feature matching* [Abbeel and Ng, 2004, Ziebart, 2010, Osa et al., 2018]: The learner's algorithm finds a policy π^{L} that matches the feature expectations of the received demonstrations, ensuring that $\|\mu_r(\pi^{\mathsf{L}}) - \hat{\mu}_r(\Xi^{\mathsf{T}})\|_2 \leq \epsilon$ where ϵ specifies a desired level of accuracy. In this standard setting, the learner's primary goal is to imitate the teacher (via feature matching) and this makes the teaching process easy. In fact, the teacher just needs to provide a sufficiently rich pool of demonstrations Ξ^{T} obtained by executing π^*, ensuring $\|\hat{\mu}_r(\Xi^{\mathsf{T}}) - \mu_r(\pi^*)\|_2 \leq \epsilon$. This guarantees that $\|\mu_r(\pi^{\mathsf{L}}) - \mu_r(\pi^*)\|_2 \leq 2\epsilon$. Furthermore, the linearity of rewards and $\|\mathbf{w}_r^*\|_1 \leq 1$ ensures that the learner's output policy π^{L} satisfies $R(\pi^{\mathsf{L}}) \geq R(\pi^*) - 2\epsilon$.

Key challenges in teaching a learner with preference constraints. In this paper, we study a novel setting where the learner has its own preferences which it additionally takes into consideration when learning a policy π^{L} using teacher's demonstrations. We formally specify our learner model in the next section; here we highlight the key challenges that arise in teaching such a learner. Given that the learner's primary goal is no longer just imitating the teacher via feature matching, the learner's output policy can be suboptimal with respect to the true reward even if it had access to $\mu_r(\pi^*)$, i.e.,



(a) Environment (b) Set of $\mu_r(\pi)$ vectors

Figure 1: An illustrative example to showcase the suboptimality of teaching when the learner has preferences and constraints. **Environment:** Figure 1a shows a grid-world environment inspired by the object-world and gathering game environments [Levine et al., 2010, Leibo et al., 2017, Mendez et al., 2018]. Each cell represents a state, there are five actions given by "left", "up", "right", "down", "stay", the transitions are deterministic, and the starting state is the top-left cell. The agent's goal is to collect objects in the environment: Collecting a "star" provides a reward of 1.0 and a "plus" a reward of 0.9; objects immediately appear again upon collection, and the rewards are discounted with γ close to 1. The optimal policy π^* is to go to the nearest "star" and then "stay" there. **Preferences:** A small number of states in the environment are distractors, depicted by colored cells in Figure 1a. We consider a learner who prefers to avoid "green" distractors: it has a hard constraint that the probability of having a "green" distractor within a 3x3 neighborhood, i.e., 1-cell distance, is at most $\epsilon = 0.1$. **Feature expectation vectors:** Figure 1b shows the set of feature expectation vectors $\{\mu_r(\pi) : \pi \in \Pi\}$. The x-axis and the y-axis represent the discounted feature count for collecting "star" and "plus" objects, respectively. The striped region represents policies that are feasible w.r.t. the learner's constraint. **Suboptimality of teaching:** Upon receiving demonstrations from an optimal policy π^* with feature vector $\mu_r(\pi^*)$, the learner under its preference constraint can best match the teacher's demonstrations (in a sense of minimizing $\|\mu_r(\pi^\mathsf{L}) - \mu_r(\pi^*)\|_2$) by outputting a policy with $\mu_r(\pi_2)$, which is clearly suboptimal w.r.t. the true rewards. Policy π_3 with feature vector $\mu_r(\pi_3)$ represents an alternate teaching policy which would have led to higher reward for the learner.

the feature expectation vector of an optimal policy π^*. Figure 1 provides an illustrative example to showcase the suboptimality of teaching when the learner has preferences and constraints. The key challenge that we address in this paper is that of designing a teaching algorithm that selects demonstrations while accounting for the learner's preferences.

3 Learner Model

In this section we describe the learner models we consider, including different ways of defining preferences and constraints. First, we introduce some notation and definitions that will be helpful. We capture learner's preferences via a feature map $\phi_c : \mathcal{S} \to [0,1]^{d_c}$. We define $\phi(s)$ as a concatenation of the two feature maps $\phi_r(s)$ and $\phi_c(s)$ given by $[\phi_r(s)^\dagger, \phi_c(s)^\dagger]^\dagger$ and let $d = d_r + d_c$. Similar to the map μ_r, we define $\mu_c \colon \Pi \to \mathbb{R}^{d_c}$, $\pi \mapsto \mathbb{E}\left(\sum_{t=0}^{\infty} \gamma^t \phi_c(s_t)\right)$ and $\mu \colon \Pi \to \mathbb{R}^d$, $\pi \mapsto \mathbb{E}\left(\sum_{t=0}^{\infty} \gamma^t \phi(s_t)\right)$. Similar to Ω_r, we define $\Omega_c \subseteq \mathbb{R}^{d_c}$ and $\Omega \subseteq \mathbb{R}^d$ as the images of the maps $\mu_c(\Pi)$ and $\mu(\Pi)$. Note that for any policy $\pi \in \Pi$, we have $\mu(\pi) = [\mu_r(\pi)^\dagger, \mu_c(\pi)^\dagger]^\dagger$.

Standard (discounted) MCE-IRL. Our learner models build on the (discounted) Maximum Causal Entropy (MCE) IRL framework [Ziebart et al., 2008, Ziebart, 2010, Ziebart et al., 2013, Zhou et al., 2018]. In the standard (discounted) MCE-IRL framework, a learning agent aims to identify a policy that matches the feature expectations of the teacher's demonstrations while simultaneously maximizing the (discounted) causal entropy given by $H(\pi) := H(\{a_t\}_{t=0,1,\dots} \| \{s_t\}_{t=0,1,\dots}) := \sum_{t=0}^{\infty} \gamma^t \mathbb{E}\Big[- \log \pi(a_t \mid s_t)\Big]$. More background is provided in Appendix D.

Including preference constraints. The standard framework can be readily extended to include learner's preferences in the form of constraints on the preference features ϕ_c. Clearly, the learner's preferences can render exact matching of the teacher's demonstrations infeasible and hence we relax this condition. To this end, we consider the following generic learner model:

$$\begin{aligned} \max_{\pi,\ \delta_r^{\text{soft}} \geq 0,\ \delta_c^{\text{soft}} \geq 0} \quad & H(\pi) - C_r \cdot \|\delta_r^{\text{soft}}\|_p - C_c \cdot \|\delta_c^{\text{soft}}\|_p \qquad (1) \\ \text{s.t.} \quad & |\mu_r(\pi)[i] - \hat{\mu}_r(\Xi^\mathsf{T})[i]| \leq \delta_r^{\text{hard}}[i] + \delta_r^{\text{soft}}[i] \ \forall i \in \{1, 2, \dots, d_r\} \\ & g_j(\mu_c(\pi)) \leq \delta_c^{\text{hard}}[j] + \delta_c^{\text{soft}}[j] \ \forall j \in \{1, 2, \dots, m\}, \end{aligned}$$

Here, $g\colon \mathbb{R}^{d_c} \mapsto \mathbb{R}$ are m convex functions representing preference constraints. The coefficients C_r and C_c are the learner's parameters which quantify the relative importance of matching the teacher's demonstrations and satisfying the learner's preferences. The learner model is further characterized by parameters $\delta_r^{\text{hard}}[i]$ and $\delta_c^{\text{hard}}[j]$ (we will use the vector notation as $\delta_r^{\text{hard}} \in \mathbb{R}^{d_r}_{\geq 0}$ and $\delta_c^{\text{hard}} \in \mathbb{R}^{m}_{\geq 0}$). The optimization variables for the learner are given by π, $\delta_r^{\text{soft}}[i]$, and $\delta_c^{\text{soft}}[j]$ (we will use the vector notation as $\delta_r^{\text{soft}} \in \mathbb{R}^{d_r}_{\geq 0}$ and $\delta_c^{\text{soft}} \in \mathbb{R}^{m}_{\geq 0}$). These parameters ($\delta_r^{\text{hard}}$, δ_c^{hard}) and optimization variables (δ_r^{soft}, δ_c^{soft}) characterize the following behavior:

- While a mismatch of up to δ_r^{hard} between the learner's and teacher's reward feature expectations incurs no cost regarding the optimization objective, a mismatch larger than δ_r^{hard} incurs a cost of $C_r \cdot \|\delta_r^{\text{soft}}\|_p$.
- Similarly, while a violation of up to δ_c^{hard} of the learner's preference constraints incurs no cost regarding the optimization objective, a violation larger than δ_c^{hard} incurs a cost of $C_c \cdot \|\delta_c^{\text{soft}}\|_p$.

Next, we discuss two special instances of this generic learner model.

3.1 Learner Model with Hard Preference Constraints

It is instructive to study a special case of the above-mentioned generic learner model. Let us consider the model in Eq. 1 with $\delta_r^{\text{hard}} = 0, \delta_c^{\text{hard}} = 0$, and a limiting case with $C_r, C_c \gg 0$ such that the term $H(\pi)$ can be neglected. Now, if we additionally assume that $C_c \gg C_r$, the learner's objective can be thought of as finding a policy π that minimizes the L^p norm distance to the reward feature expectations of the teacher's demonstration while satisfying the constraints $g_j(\mu_c(\pi)) \leq 0 \; \forall j \in \{1, 2, \dots, m\}$. More formally, we study the following learner model given in Eq. 2 below:

$$\begin{aligned} \min_{\pi} \quad & \|\mu_r(\pi) - \hat{\mu}_r(\Xi^{\mathsf{T}})\|_p \\ \text{s.t.} \quad & g_j(\mu_c(\pi)) \leq 0 \; \forall j \in \{1, 2, \dots, m\}. \end{aligned} \tag{2}$$

To get a better understanding of the model, we can define the learner's constraint set as $\Omega^{\mathsf{L}} := \{\mu : \mu \in \Omega \text{ s.t. } g_j(\mu_c) \leq 0 \; \forall j \in \{1, 2, \dots, m\}\}$. Similar to Ω^{L}, we define $\Omega_r^{\mathsf{L}} \subseteq \Omega_r$ where Ω_r^{L} is the projection of the set Ω^{L} to the subspaces $\mathbb{R}^{d_r}$. We can now rewrite the above optimization problem as $\min_{\pi\colon \mu_r(\pi) \in \Omega_r^{\mathsf{L}}} \|\mu_r(\pi) - \hat{\mu}_r(\Xi^{\mathsf{T}})\|_p$. Hence, the learner's behavior is given by:

(i) *Learner can match:* When $\hat{\mu}_r(\Xi^{\mathsf{T}}) \in \Omega_r^{\mathsf{L}}$, the learner outputs a policy π^{L} s.t. $\mu_r(\pi^{\mathsf{L}}) = \hat{\mu}_r(\Xi^{\mathsf{T}})$.

(ii) *Learner cannot match:* Otherwise, the learner outputs a policy π^{L} such that $\mu_r(\pi^{\mathsf{L}})$ is given by the L^p norm projection of the vector $\hat{\mu}_r(\Xi^{\mathsf{T}})$ onto the set Ω_r^{L}.

Figure 1 provides an illustration of the behavior of this learner model. We will design learner-aware teaching algorithms for this learner model in Section 4.1 and Section 5.

3.2 Learner Model with Soft Preference Constraints

Another interesting learner model that we study in this paper arises from the generic learner when we consider $m = d_c$ number of box-type linear constraints with $g_j(\mu_c(\pi)) = \mu_c(\pi)[j] \; \forall j \in \{1, 2, \dots, d_c\}$. We consider an L^1 norm penalty on violation, and for simplicity we consider $\delta_r^{\text{hard}}[i] = 0 \; \forall i \in \{1, 2, \dots, d_r\}$. In this case, the learner's model is given by

$$\begin{aligned} \max_{\pi,\, \delta_r^{\text{soft}} \geq 0,\, \delta_c^{\text{soft}} \geq 0} \quad & H(\pi) - C_r \cdot \|\delta_r^{\text{soft}}\|_1 - C_c \cdot \|\delta_c^{\text{soft}}\|_1 \\ \text{s.t.} \quad & |\mu_r(\pi)[i] - \hat{\mu}_r(\Xi^{\mathsf{T}})[i]| \leq \delta_r^{\text{soft}}[i] \; \forall i \in \{1, 2, \dots, d_r\} \\ & \mu_c(\pi)[j] \leq \delta_c^{\text{hard}}[j] + \delta_c^{\text{soft}}[j] \; \forall j \in \{1, 2, \dots, d_c\}, \end{aligned} \tag{3}$$

The solution to the above problem corresponds to a *softmax* policy with a reward function $R_{\boldsymbol{\lambda}}(s) = \langle \boldsymbol{w}_{\boldsymbol{\lambda}}, \phi(s) \rangle$ where $\boldsymbol{w}_{\boldsymbol{\lambda}} \in \mathbb{R}^d$ is parametrized by $\boldsymbol{\lambda}$. The optimal parameters $\boldsymbol{\lambda}$ can be computed efficiently and the corresponding softmax policy is then obtained by *Soft-Value-Iteration* procedure (see [Ziebart, 2010, Algorithm. 9.1], [Zhou et al., 2018]). Details are provided in Appendix E. We will design learner-aware teaching algorithms for this learner model in Section 4.2.

4 Learner-aware Teaching under Known Constraints

In this section, we analyze the setting when the teacher has full knowledge of the learner's constraints.

4.1 A Learner-aware Teacher for Hard Preferences: AWARE-CMDP

Here, we design a learner-aware teaching algorithm when considering the learner from Section 3.1. Given that the teacher has full knowledge of the learner's preferences, it can compute an optimal teaching policy by maximizing the reward over policies that satisfy the learner's preference constraints, i.e., the teacher solves a constrained-MDP problem (see [De, 1960, Altman, 1999]) given by

$$\max_{\pi} \quad \langle \mathbf{w}_r^*, \mu_r(\pi) \rangle \quad \text{s.t.} \quad \mu_r(\pi) \in \Omega_r^{\mathsf{L}}.$$

We refer to an optimal solution of this problem as π^{aware} and the corresponding teacher as AWARE-CMDP. We can make the following observation formalizing the value of learner-aware teaching:

Theorem 1. *For simplicity, assume that the teacher can provide an exact feature expectation $\mu(\pi)$ of a policy instead of providing demonstrations to the learner. Then, the value of learner-aware teaching is*

$$\max_{\pi \text{ s.t. } \mu_r(\pi) \in \Omega_r^{\mathsf{L}}} \Big\langle \mathbf{w}_r^*, \mu_r(\pi) \Big\rangle - \Big\langle \mathbf{w}_r^*, \mathrm{Proj}_{\Omega_r^{\mathsf{L}}}\big(\mu_r(\pi^*)\big) \Big\rangle \geq 0.$$

When the set Ω^{L} is defined via a set of linear constraints, the above problem can be formulated as a linear program and solved exactly. Details are provided in Appendix F.

4.2 A Learner-aware Teacher for Soft Preferences: AWARE-BIL

For the learner models in Section 3, the optimal learner-aware teaching problem can be naturally formalized as the following bi-level optimization problem:

$$\max_{\pi^{\mathsf{T}}} \quad R(\pi^{\mathsf{L}}) \quad \text{s.t.} \quad \pi^{\mathsf{L}} \in \arg\max_{\pi} \mathrm{IRL}(\pi, \mu(\pi^{\mathsf{T}})), \tag{4}$$

where $\mathrm{IRL}(\pi, \mu(\pi^{\mathsf{T}}))$ stands for the IRL problem solved by the learner given demonstrations from π^{T} and can include preferences of the learner (see Eq. 1 in Section 3).

There are many possibilities for solving this bi-level optimization problem—see for example [Sinha et al., 2018] for an overview. In this paper we adopted a *single-level reduction* approach to simplify the above bi-level optimization problem as this results in particularly intuitive optimiziation problems for the teacher. The basic idea of single-level reduction is to replace the lower-level problem, i.e., $\arg\max_{\pi} \mathrm{IRL}(\pi, \mu(\pi^{\mathsf{T}}))$, by the optimality conditions for that problem given by the Karush-Kuhn-Tucker conditions [Boyd and Vandenberghe, 2004, Sinha et al., 2018]. For the learner model outlined in Section 3.2, these reductions take the following form (see Appendix G for details):

$$\begin{aligned} \max_{\boldsymbol{\lambda} := \{\boldsymbol{\alpha}^{\text{low}} \in \mathbb{R}^{d_r},\, \boldsymbol{\alpha}^{\text{up}} \in \mathbb{R}^{d_r},\, \boldsymbol{\beta} \in \mathbb{R}^{d_c}\}} \quad & \langle \mathbf{w}_r^*, \mu_r(\pi_{\boldsymbol{\lambda}}) \rangle \\ \text{s.t.} \quad & 0 \leq \boldsymbol{\alpha}^{\text{low}} \leq C_r \\ & 0 \leq \boldsymbol{\alpha}^{\text{up}} \leq C_r \\ \{0 \leq \boldsymbol{\beta} \leq C_c \;\; \texttt{AND} \;\; \mu_c(\pi_{\boldsymbol{\lambda}}) \leq \delta_c^{\text{hard}}\} \;\; \texttt{OR} \;\; & \{\boldsymbol{\beta} = C_c \;\; \texttt{AND} \;\; \mu_c(\pi_{\boldsymbol{\lambda}}) \geq \delta_c^{\text{hard}}\} \end{aligned} \tag{5}$$

where $\pi_{\boldsymbol{\lambda}}$ corresponds to a *softmax* policy with a reward function $R_{\boldsymbol{\lambda}}(s) = \langle \boldsymbol{w}_{\boldsymbol{\lambda}}, \phi(s) \rangle$ for $\boldsymbol{w}_{\boldsymbol{\lambda}} = [(\boldsymbol{\alpha}^{\text{low}} - \boldsymbol{\alpha}^{\text{up}})^{\dagger}, -\boldsymbol{\beta}^{\dagger}]^{\dagger}$. Thus, finding optimal demonstrations means optimization over *softmax* teaching policies while respecting the learner's preferences. To actually solve the above optimization problem and find good teaching policies, we use an approach inspired by the Frank-Wolfe algorithm [Jaggi, 2013] detailed in Appendix G. We refer to a teacher implementing this approach as AWARE-BIL.

5 Learner-Aware Teaching Under Unknown Constraints

In this section, we consider the more realistic and challenging setting in which the teacher T does *not* know the learner L's constraint set Ω_r^{L}. Without feedback from L, T can generally not do better than the agnostic teacher who simply ignores any constraints. We therefore assume that T and L interact in rounds as described by Algorithm 1. The two versions of the algorithm we describe in Sections 5.1 and 5.2 are obtained by specifying how T adapts the teaching policy in each round.

Algorithm 1 Teacher-learner interaction in the adaptive teaching setting

1: Initial teaching policy $\pi^{\mathsf{T},0}$ (e.g., optimal policy ignoring any constraints)
2: **for** round $i = 0, 1, 2, \dots$ **do**
3: Teacher provides demonstrations with feature vector $\mu_r^{\mathsf{T},i}$ using policy $\pi^{\mathsf{T},i}$
4: Learner upon receiving $\mu_r^{\mathsf{T},i}$ computes a policy $\pi^{\mathsf{L},i}$ with feature vector $\mu_r^{\mathsf{L},i}$
5: Teacher observes learner's feature vector $\mu_r^{\mathsf{L},i}$ and adapts the teaching policy

In this section, we assume that L is as described in Section 3.1: Given demonstrations Ξ^{T}, L finds a policy π^{L} such that $\mu_r(\pi^{\mathsf{L}})$ matches the L^2-projection of $\hat{\mu}_r(\Xi^{\mathsf{T}})$ onto Ω_r^{L}. For the sake of simplifying the presentation and the analysis, we also assume that L and T can observe the exact feature expectations of their respective policies, e.g., $\hat{\mu}_r(\Xi^{\mathsf{T}}) = \mu_r(\pi^{\mathsf{T}})$ if Ξ^{T} is sampled from π^{T}.

5.1 An Adaptive Learner-aware Teacher Using Volume Search: ADAWARE-VOL

In our first adaptive teaching algorithm ADAWARE-VOL, T maintains an estimate $\hat{\Omega}_r^{\mathsf{L}} \supset \Omega_r^{\mathsf{L}}$ of the learner's constraint set, which in each round gets updated by intersecting the current version with a certain affine halfspace, thus reducing the volume of $\hat{\Omega}_r^{\mathsf{L}}$. The new teaching policy is then any policy $\pi^{\mathsf{T},i+1}$ which is optimal under the constraint that $\mu^{\mathsf{T},i+1} \in \hat{\Omega}_r^{\mathsf{L}}$. The interaction ends as soon as $\|\mu_r^{\mathsf{L},i} - \mu_r^{\mathsf{T},i}\|_2 \leq \epsilon$ for a threshold ϵ. Details are provided in Appendix C.1.

Theorem 2. *Upon termination of* ADAWARE-VOL*, L's output policy π^{L} satisfies $R(\pi^{\mathsf{L}}) \geq R(\pi^{\text{aware}}) - \epsilon$ for any policy π^{aware} which is optimal under L's constraints. For the special case that Ω_r^{L} is a polytope defined by m linear inequalities, the algorithm terminates in $O(m^{d_r})$ iterations.*

5.2 An Adaptive Learner-aware Teacher Using Line Search: ADAWARE-LIN

In our second adaptive teaching algorithm, ADAWARE-LIN, T adapts the teaching policy by performing a binary search on a line segment of the form $\{\mu_r^{\mathsf{L},i} + \alpha \mathbf{w}_r^* \mid \alpha \in [\alpha_{\min}, \alpha_{\max}]\} \subset \mathbb{R}^{d_r}$ to find a vector $\mu_r^{\mathsf{T},i+1} = \mu_r^{\mathsf{L},i} + \alpha_i \mathbf{w}_r^*$ that is the vector of feature expectations of a policy; here $\alpha_{\max} > \alpha_{\min} > 0$ are fixed constants. If that is not successful, the teacher finds a teaching policy with $\mu_r^{\mathsf{T},i+1} \in \arg\min_{\mu_r \in \Omega_r} \|\mu_r - \mu_r^{\mathsf{L},i} - \alpha_{\min}\mathbf{w}_r^*\|_2$. The following theorem analyzes the convergence of L's performance to $\overline{R}_{\mathsf{L}} := \max_{\mu_r \in \Omega_r} R(\mu_r)$ under the assumption that T's search succeeds in every round. The proof and further details are provided in Appendix C.2.

Theorem 3. *Fix some $\varepsilon > 0$ and assume that there exists a constant $\alpha_{\min} > 0$ such that, as long as $\overline{R}_{\mathsf{L}} - R(\mu_r^{\mathsf{L},i}) > \varepsilon$, the teacher can find a teaching policy $\pi^{\mathsf{T},i+1}$ satisfying $\mu_r^{\mathsf{T},i+1} = \mu_r^{\mathsf{L},i} + \alpha_i \mathbf{w}_r^*$ for some $\alpha_i \geq \alpha_{\min}$. Then the learner's performance increases monotonically in each round of* ADAWARE-LIN*, i.e., $R(\mu_r^{\mathsf{L},i+1}) > R(\mu_r^{\mathsf{L},i})$. Moreover, after at most $O(\frac{D^2}{\varepsilon\alpha_{\min}} \log \frac{D}{\varepsilon})$ teaching steps, the learner's performance satisfies $R(\mu_r^{\mathsf{L},i}) > \overline{R}_{\mathsf{L}} - 2\varepsilon$. Here we abbreviate $D := \operatorname{diam} \Omega_r$.*

6 Experimental Evaluation

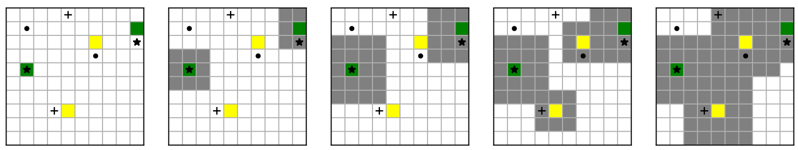
In this section we evaluate our teaching algorithms for different types of learners on the environment introduced in Figure 1. The environment we consider here has three types of reward objects, i.e., a "star" object with reward of 1.0, a "plus" object with reward of 0.9, and a "dot" object with reward of 0.2. Two objects of each type are placed randomly on the grid such that there is always only a single object in each grid cell. The presence of an object of type "star", "plus", or "dot" in some state s is encoded in the reward features $\phi_r(s)$ by a binary-indicator for each type such that $d_r = 3$. We use a discount factor of $\gamma = 0.99$. Upon collecting an object, there is a 0.1 probability of transiting to a terminal state.

Learner models. We consider a total of 5 different learners whose preferences can be described by *distractors* in the environment. Each learner prefers to avoid a certain subset of these distractors.

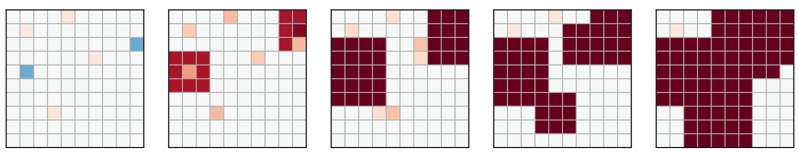
There is a total of 4 of distractors: (i) two “green" distractors are randomly placed at a distance of 0-cell and 1-cell to the “star" objects, respectively; (ii) two “yellow" distractors are randomly placed at a distance of 1-cell and 2-cells to the “plus" objects, respectively, see Figure 2a.

Through these distractors we define learners L1-L5 as follows: **(L1)** no preference features ($d_c = 0$); **(L2)** two preference features ($d_c = 2$) such that $\phi_c(s)[1]$ and $\phi_c(s)[2]$ are binary indicators of whether there is a “green" distractor at a distance of 0-cells or 1-cell, respectively; **(L3)** four preference features ($d_c = 4$) such that $\phi_c(s)[1], \phi_c(s)[2]$ are as for L2, and $\phi_c(s)[3]$ and $\phi_c(s)[4]$ are binary indicators of whether there is a “green" distractor at a distance of 2-cells or a “yellow” distractor at a distance of 0-cells, respectively; **(L4)** five preference features ($d_c = 5$) such that $\phi_c(s)[1], \ldots, \phi_c(s)[4]$ are as for L3, and $\phi_c(s)[5]$ is a binary indicator whether there is a “yellow” distractor at a distance of 1-cell; and **(L5)** six preference features ($d_c = 6$) such that $\phi_c(s)[1], \ldots, \phi_c(s)[5]$ are as for L4, and $\phi_c(s)[6]$ is a binary indicator whether there is a “yellow” distractor at a distance of 2-cells.

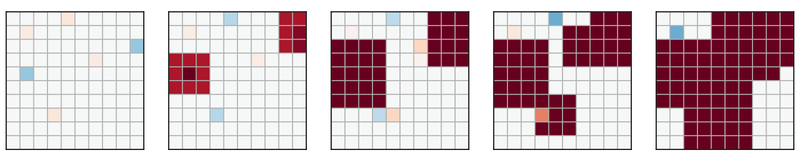
The first row in Figure 2 shows an instance of the considered object-worlds and indicates the preference of the learners to avoid certain regions by the gray area.



(a) Environments and learners’ preferences for 5 different learners L1, . . ., L5



(b) Learners’ rewards inferred from learner-agnostic teacher’s (AGNOSTIC) demonstrations



(c) Learners’ rewards inferred from learner-aware teacher’s (AWARE-BIL) demonstrations

Figure 2: Teaching in object-world environments under full knowledge of the learner’s preferences. Green and yellow cells indicate distractors associated with either “star" or “plus" objects, respectively. Learner’s preferences to avoid cells are indicated in gray. Learner model from Section 3.2 with $C_r = 5$, $C_c = 10$, and $\delta_c^{\text{hard}} = 0$ is considered for these experiments. The learner-aware teacher enable the learner to infer reward functions that are compatible with the learner’s preferences and achieve higher average rewards. In Figure 2b and Figure 2c, blue color represents positive reward, red color represents negative reward, and the magnitude of the reward is indicated by color intensity.

6.1 Teaching under known constraints

In this section we consider learners with soft constraints from Section 3.2, with preference features as described above, and parameters $C_r = 5$, $C_c = 10$, and $\delta_c^{\text{hard}} = 0$ (more experimental results for different values of C_r and C_c are provided in Appendix B.1). Our first results are presented in Figure 2. The second and third rows show the rewards inferred by the learners for demonstrations provided by a learner-agnostic teacher who ignores any constraints (AGNOSTIC) and the bi-level learner-aware teacher (AWARE-BIL), respectively. We observe that AGNOSTIC fails to teach the learner about objects’ positive rewards in cases where the learners’ preferences conflict with the position of the most rewarding objects (second row). In contrast, AWARE-BIL always successfully teaches the learners about rewarding objects that are compatible with the learners’ preferences (third row).

We also compare AGNOSTIC and AWARE-BIL in terms of reward achieved by the learner after teaching for object worlds of size 10×10 in Table 1. The numbers show the average reward over 10 randomly generated object-worlds. Note that AWARE-BIL has to solve a non-convex optimization problem to find the optimal teaching policy, cf. Eq. 5. Because we use a gradient-based optimization approach, the teaching policies found can depend on the initial point for optimization. Hence, we always consider the following two initial points for optimization and select the teaching policy which results in a higher objective value: (i) all optimization variables in Eq. 5 are set to zero, and (ii) the optimization variables are initialized as $\alpha^{\text{low}}[i] = \max\{w_{\boldsymbol{\lambda}}[i], 0\}$, $\alpha^{\text{up}}[i] = \max\{-w_{\boldsymbol{\lambda}}[i], 0\}$, and $\boldsymbol{\beta} = 0$, where $\boldsymbol{w}_{\boldsymbol{\lambda}}$ is as inferred by the learner when taught by AGNOSTIC and $i \in \{1, \ldots, d_r\}$, cf. Section 3.2. From Table 1 we observe that a learner can learn better policies from a teacher that accounts for the learner's preferences.

Table 1: Learners' average rewards after teaching. L1, ..., L5 correspond to learners with preferences as shown in Figure 2. Results are averaged over 10 random object-worlds, $\pm$ standard error

		Learner ($C_r = 5, C_c = 10$)				
		L1	L2	L3	L4	L5
Teacher	AGNOSTIC	7.99 ± 0.02	0.01 ± 0.00	0.01 ± 0.00	0.01 ± 0.00	0.00 ± 0.00
	AWARE-BIL	8.00 ± 0.02	7.20 ± 0.01	4.86 ± 0.30	3.15 ± 0.27	1.30 ± 0.07

6.2 Teaching under unknown constraints

In this section we evaluate the teaching algorithms from Section 5. We consider the learner model from Section 3.1 that uses L^2-projection to match reward feature expectations as studied in Section 5, cf. Eq. 2.[2] For modeling the hard constraints, we consider box-type linear constraints with $\delta_c^{\text{hard}}[j] = 2.5\ \forall j \in \{1, 2, \ldots, d_c\}$ for the preference features, cf. Eq. 3.

We study the learners L1, L2, and L3 with preferences corresponding to the first three object-worlds shown in Figure 2a. We report the results for learner L2 below; results for learners L1 and L3 are deferred to the Appendix B.2.

In this context it is instructive to investigate how quickly these adaptive teaching strategies converge to the performance of a teacher who has full knowledge about the learner. Results comparing the adaptive teaching strategies (ADAWARE-VOL and ADAWARE-LIN) are shown in Figure 3a. We can observe that both teaching strategies get close to the best possible performance under full knowledge about the learner (AWARE-CMDP). We also provide results showing the performance achieved by the adaptive teaching strategies on object-worlds of varying sizes, see Figure 3b.

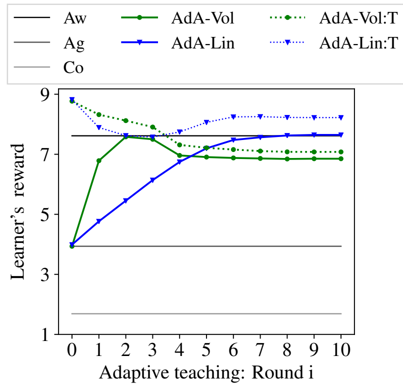
Note that the performance of ADAWARE-VOL decreases slightly when teaching for more rounds, i.e., comparing the results after 3 teaching rounds and at the end of the teaching process. This is because of approximations when learner is computing the policy via projection, which in turn leads to errors on the teacher side when approximating $\hat{\Omega}_r^{\mathsf{L}}$ (refer to discussion in Footnote 2). In contrast, ADAWARE-LIN performance always increases when teaching for more rounds.

7 Related Work

Our work is closely related to algorithmic machine teaching [Goldman and Kearns, 1995, Zhu, 2015, Zhu et al., 2018], whose general goal is to design teaching algorithms that optimize the data that is provided to a learning algorithm. Most works in machine teaching so far focus on supervised learning tasks and assume that the learning algorithm is fully known to the teacher, see e.g., [Zhu, 2013, Singla et al., 2014, Liu and Zhu, 2016, Mac Aodha et al., 2018].

In the IRL setting, few works study how to provide maximally informative demonstrations to the learner, e.g., [Cakmak and Lopes, 2012, Brown and Niekum, 2019]. In contrast to our work, their teacher fully knows the learner model and provides the demonstrations without any adaptation to the learner. The question of how a teacher should adaptively react to a learner has been addressed by [Singla et al., 2013, Liu et al., 2018, Chen et al., 2018, Melo et al., 2018, Yeo et al., 2019, Hunziker et al., 2019], but only in the supervised setting. In a recent work, [Kamalaruban et al., 2019] have studied the problem of adaptively teaching an IRL agent by pro-

[2] To implement the learner in Eq. 2, we approximated the learner's projection onto the set Ω_r^{L} as follows: We implemented the learner based on the optimization problem given in Eq. 3 with a hard constraint on preferences and L^2 norm penalty on reward mismatch scaled with a large value of $C_r = 20$.



(a) Reward over teaching rounds

Teacher \ Env	10×10	15×15	20×20
AWARE-CMDP	7.62 ± 0.02	7.44 ± 0.04	7.19 ± 0.04
AGNOSTIC	3.94 ± 0.09	3.84 ± 0.06	3.95 ± 0.06
CONSERV	1.68 ± 0.01	1.67 ± 0.012	1.62 ± 0.02
ADAWARE-VOL (3^{rd})	7.50 ± 0.14	7.50 ± 0.04	7.29 ± 0.05
ADAWARE-VOL (end)	6.85 ± 0.33	7.06 ± 0.06	6.77 ± 0.08
ADAWARE-LIN (3^{rd})	6.14 ± 0.08	6.28 ± 0.10	6.37 ± 0.08
ADAWARE-LIN (end)	7.64 ± 0.02	7.53 ± 0.03	7.29 ± 0.06

(b) Varying grid-size

Figure 3: Performance of adaptive teaching strategies ADAWARE-VOL and ADAWARE-LIN. **(left)** Figure 3a shows the reward for learner's policy over number of teaching interactions. The horizontal lines indicate the performance of learner's policy for the learner-aware teacher with full knowledge of the learner's constraints AWARE-CMDP, the learner-agnostic teacher AGNOSTIC who ignores any constraints, and a conservative teacher CONSERV who considers all 6 constraints (assuming the learner model L5 in Figure 2). Our adaptive teaching strategies ADAWARE-VOL and ADAWARE-LIN significantly outperform baselines (AGNOSTIC and CONSERV) and quickly converge towards the optimal performance of AWARE-CMDP. The dotted lines ADAWARE-VOL:T and ADAWARE-LIN:T show the rewards corresponding to teacher's policy at a round and are shown to highlight the very different behavior of two adaptive teaching strategies. **(right)** Table 3b shows results for varying grid-size of the environment. Results are reported at $i = 3^{\text{rd}}$ round and at the "end" round when algorithm reaches it's stopping criterion. Results are reported as average over 10 runs $\pm$ standard error, where each run corresponds to a random environment.

viding an informative sequence of demonstrations. However, they assume that the teacher has full knowlege of the learner's dynamics.

Within the area of IRL, there is a line of work on active learning approaches [Cohn et al., 2011, Brown et al., 2018, Brown and Niekum, 2018, Amin et al., 2017, Cui and Niekum, 2018], which is related to our work. In contrast to us, they take the perspective of the learner who actively influences the demonstrations it receives. A few papers have addressed the problem that arises when the learner does not have full access to the reward features, e.g., [Levine et al., 2010] and [Haug et al., 2018].

Our work is also loosely related to multi-agent reinforcement learning. [Dimitrakakis et al., 2017] studied the interaction between agents with misaligned models with a focus on the question of how to jointly optimize a policy. [Ghosh et al., 2019] studied the problem of designing robust AI agent that can interact with another agent of unknown type. However, these works do not tackle the problem of teaching an agent by demonstrations. Another related work is [Hadfield-Menell et al., 2016] which studied the cooperation of agents who do not perfectly understand each other.

8 Conclusions and Outlook

In this paper we considered inverse reinforcement learning in the context of learners with preferences and constraints. In this setting, the learner does not only focus on matching the teacher's demonstrated behavior but also takes its own preferences, e.g., behavioral biases or physical constraints, into account. We developed a theoretical framework for this setting, and proposed and studied algorithms for learner-aware teaching in which the teacher accounts for the learner's preferences for the cases of known and unknown preference constraints. We demonstrated significant performance improvements of our learner-aware teaching strategies as compared to learner-agnostic teaching both theoretically and empirically. Our theoretical framework and our proposed algorithms foster the application of IRL in real-world settings in which the learner does not blindly follow a teacher's demonstrations.

There are several promising directions for future work, including but not limited to: The evaluation of our approach in machine-human and human-machine tasks; extensions of our approach to other learner models; approaches for learning efficiently from a learner's point of view from a fixed set of (potentially suboptimal) demonstrations in the case of preference constraints.

Acknowledgements

This work was supported by Microsoft Research through its PhD Scholarship Programme.

References

[Abbeel and Ng, 2004] Abbeel, P. and Ng, A. Y. (2004). Apprenticeship learning via inverse reinforcement learning. In *ICML*.

[Altman, 1999] Altman, E. (1999). *Constrained Markov decision processes*, volume 7. CRC Press.

[Amin et al., 2017] Amin, K., Jiang, N., and Singh, S. P. (2017). Repeated inverse reinforcement learning. In *NIPS*, pages 1813–1822.

[Boularias et al., 2011] Boularias, A., Kober, J., and Peters, J. (2011). Relative entropy inverse reinforcement learning. In *AISTATS*, pages 182–189.

[Boyd and Vandenberghe, 2004] Boyd, S. and Vandenberghe, L. (2004). *Convex optimization*. Cambridge university press.

[Brown et al., 2018] Brown, D. S., Cui, Y., and Niekum, S. (2018). Risk-Aware Active Inverse Reinforcement Learning. In *Conference on Robot Learning*, pages 362–372.

[Brown and Niekum, 2018] Brown, D. S. and Niekum, S. (2018). Efficient probabilistic performance bounds for inverse reinforcement learning. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

[Brown and Niekum, 2019] Brown, D. S. and Niekum, S. (2019). Machine teaching for inverse reinforcement learning: Algorithms and applications. In *AAAI*.

[Cakmak and Lopes, 2012] Cakmak, M. and Lopes, M. (2012). Algorithmic and human teaching of sequential decision tasks. In *AAAI*.

[Chen et al., 2018] Chen, Y., Singla, A., Mac Aodha, O., Perona, P., and Yue, Y. (2018). Understanding the role of adaptivity in machine teaching: The case of version space learners. In *Advances in Neural Information Processing Systems*, pages 1476–1486.

[Cohn et al., 2011] Cohn, R., Durfee, E., and Singh, S. (2011). Comparing Action-query Strategies in Semi-autonomous Agents. In *AAMAS*, pages 1287–1288, Richland, SC.

[Cui and Niekum, 2018] Cui, Y. and Niekum, S. (2018). Active reward learning from critiques. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6907–6914. IEEE.

[De, 1960] De, G. G. (1960). Les problemes de decisions sequentielles. cahiers du centre d'etudes de recherche operationnelle vol. 2, pp. 161-179.

[Dimitrakakis et al., 2017] Dimitrakakis, C., Parkes, D. C., Radanovic, G., and Tylkin, P. (2017). Multi-view decision processes: the helper-ai problem. In *Advances in Neural Information Processing Systems*, pages 5443–5452.

[Dudík et al., 2007] Dudík, M., Phillips, S. J., and Schapire, R. E. (2007). Maximum entropy density estimation with generalized regularization and an application to species distribution modeling. *Journal of Machine Learning Research*, 8:1217–1260.

[Ghosh et al., 2019] Ghosh, A., Tschiatschek, S., Mahdavi, H., and Singla, A. (2019). Towards deployment of robust AI agents for human-machine partnerships. In *Workshop on Safety and Robustness in Decision Making (SRDM) at NeurIPS'19*.

[Goldman and Kearns, 1995] Goldman, S. A. and Kearns, M. J. (1995). On the complexity of teaching. *Journal of Computer and System Sciences*, 50(1):20–31.

[Hadfield-Menell et al., 2016] Hadfield-Menell, D., Russell, S. J., Abbeel, P., and Dragan, A. (2016). Cooperative inverse reinforcement learning. In *NIPS*.

[Haug et al., 2018] Haug, L., Tschiatschek, S., and Singla, A. (2018). Teaching Inverse Reinforcement Learners via Features and Demonstrations. In *Advances in Neural Information Processing Systems*, pages 8473–8482.

[Hunziker et al., 2019] Hunziker, A., Chen, Y., Mac Aodha, O., Rodriguez, M. G., Krause, A., Perona, P., Yue, Y., and Singla, A. (2019). Teaching multiple concepts to a forgetful learner. In *Advances in Neural Information Processing Systems*.

[Jaggi, 2013] Jaggi, M. (2013). Revisiting Frank-Wolfe: Projection-free sparse convex optimization. In *Proceedings of the 30th International Conference on Machine Learning*, pages 427–435.

[Kamalaruban et al., 2019] Kamalaruban, P., Devidze, R., Cevher, V., and Singla, A. (2019). Interactive teaching algorithms for inverse reinforcement learning. In *IJCAI*, pages 2692–2700.

[Kazama and Tsujii, 2005] Kazama, J. and Tsujii, J. (2005). Maximum entropy models with inequality constraints: A case study on text categorization. *Machine Learning*, 60(1-3):159–194.

[Leibo et al., 2017] Leibo, J. Z., Zambaldi, V., Lanctot, M., Marecki, J., and Graepel, T. (2017). Multi-agent reinforcement learning in sequential social dilemmas. In *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems*, pages 464–473.

[Levine et al., 2010] Levine, S., Popovic, Z., and Koltun, V. (2010). Feature construction for inverse reinforcement learning. In *NIPS*, pages 1342–1350.

[Liu and Zhu, 2016] Liu, J. and Zhu, X. (2016). The teaching dimension of linear learners. *Journal of Machine Learning Research*, 17(162):1–25.

[Liu et al., 2018] Liu, W., Dai, B., li, X., Rehg, J. M., and Song, L. (2018). Towards black-box iterative machine teaching. In *ICML*.

[Mac Aodha et al., 2018] Mac Aodha, O., Su, S., Chen, Y., Perona, P., and Yue, Y. (2018). Teaching categories to human learners with visual explanations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3820–3828.

[Melo et al., 2018] Melo, F. S., Guerra, C., and Lopes, M. (2018). Interactive optimal teaching with unknown learners. In *IJCAI*, pages 2567–2573.

[Mendez et al., 2018] Mendez, J. A. M., Shivkumar, S., and Eaton, E. (2018). Lifelong inverse reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 4507–4518.

[Osa et al., 2018] Osa, T., Pajarinen, J., Neumann, G., Bagnell, J. A., Abbeel, P., Peters, J., et al. (2018). An algorithmic perspective on imitation learning. *Foundations and Trends® in Robotics*, 7(1-2):1–179.

[Ratliff et al., 2006] Ratliff, N. D., Bagnell, J. A., and Zinkevich, M. A. (2006). Maximum margin planning. In *ICML*, pages 729–736.

[Singla et al., 2013] Singla, A., Bogunovic, I., Bartók, G., Karbasi, A., and Krause, A. (2013). On actively teaching the crowd to classify. In *NIPS Workshop on Data Driven Education*.

[Singla et al., 2014] Singla, A., Bogunovic, I., Bartók, G., Karbasi, A., and Krause, A. (2014). Near-optimally teaching the crowd to classify. In *ICML*.

[Sinha et al., 2018] Sinha, A., Malo, P., and Deb, K. (2018). A review on bilevel optimization: from classical to evolutionary approaches and applications. *IEEE Transactions on Evolutionary Computation*, 22(2):276–295.

[Yeo et al., 2019] Yeo, T., Kamalaruban, P., Singla, A., Merchant, A., Asselborn, T., Faucon, L., Dillenbourg, P., and Cevher, V. (2019). Iterative classroom teaching. In *AAAI*, pages 5684–5692.

[Zhou et al., 2018] Zhou, Z., Bloem, M., and Bambos, N. (2018). Infinite time horizon maximum causal entropy inverse reinforcement learning. *IEEE Trans. Automat. Contr.*, 63(9):2787–2802.

[Zhu, 2013] Zhu, X. (2013). Machine teaching for bayesian learners in the exponential family. In *NIPS*, pages 1905–1913.

[Zhu, 2015] Zhu, X. (2015). Machine teaching: An inverse problem to machine learning and an approach toward optimal education. In *AAAI*, pages 4083–4087.

[Zhu et al., 2018] Zhu, X., Singla, A., Zilles, S., and Rafferty, A. N. (2018). An Overview of Machine Teaching. *arXiv:1801.05927*.

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

[Ziebart et al., 2013] Ziebart, B. D., Bagnell, J. A., and Dey, A. K. (2013). The principle of maximum causal entropy for estimating interacting processes. *IEEE Transactions on Information Theory*, 59(4):1966–1980.

[Ziebart et al., 2008] Ziebart, B. D., Maas, A. L., Bagnell, J. A., and Dey, A. K. (2008). Maximum entropy inverse reinforcement learning. In *AAAI*.

A List of Appendices

In this section we provide a brief description of the content provided in the appendices of the paper.

- Appendix B provides additional experimental results (Section 6).
- Appendix C provides additional details on the adaptive teaching strategies (Section 5).
- Appendix D provides background on the (discounted) MCE-IRL problem (Section 3).
- Appendix E provides additional details on the (discounted) MCE-IRL problem with preferences (Section 3.2).
- Appendix F provides the LP formulation for the teacher AWARE-CMDP (Section 4.1).
- Appendix G provides additional details on the bi-level optimization approach for the teacher AWARE-BIL (Section 4.2).

B Experimental Evaluation: Additional Results (Section 6)

B.1 Teaching under known constraints (Section 6.1)

Additional results for teaching under known constraints are presented in Table 2. We observe that AWARE-BIL clearly outperforms AGNOSTIC for most combinations of C_r and C_c. Only for $C_r = 10, C_c = 1$, the teachers AWARE-BIL and AGNOSTIC achieve similar performance because $C_r \gg C_c$, and hence the learner values achieving higher reward more than satisfying its preferences.

Table 2: Learners' average rewards after teaching. L1, . . ., L5 correspond to learners with preferences as shown in Figure 2. Results are averaged over 10 random object-worlds, $\pm$ standard error

		Learner ($C_r = 5, C_c = 10$)				
		L1	L2	L3	L4	L5
Teacher	AGNOSTIC	7.99 ± 0.02	0.01 ± 0.00	0.01 ± 0.00	0.01 ± 0.00	0.00 ± 0.00
	AWARE-BIL	8.00 ± 0.02	7.20 ± 0.01	4.86 ± 0.30	3.15 ± 0.27	1.30 ± 0.07

		Learner ($C_r = 10, C_c = 10$)				
		L1	L2	L3	L4	L5
Teacher	AGNOSTIC	8.34 ± 0.01	0.17 ± 0.02	0.01 ± 0.00	0.01 ± 0.00	0.00 ± 0.00
	AWARE-BIL	8.33 ± 0.01	6.90 ± 0.17	5.03 ± 0.31	3.27 ± 0.28	1.35 ± 0.07

		Learner ($C_r = 10, C_c = 5$)				
		L1	L2	L3	L4	L5
Teacher	AGNOSTIC	8.36 ± 0.01	8.14 ± 0.03	0.01 ± 0.00	0.01 ± 0.00	0.00 ± 0.00
	AWARE-BIL	8.34 ± 0.01	8.13 ± 0.03	5.20 ± 0.29	3.43 ± 0.27	1.69 ± 0.0

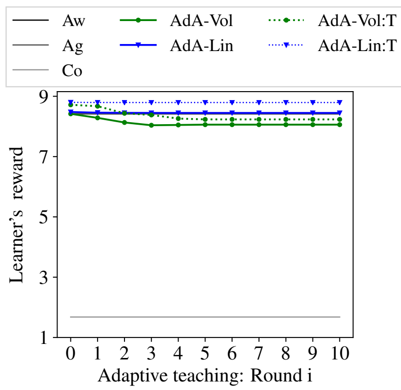
		Learner ($C_r = 5, C_c = 5$)				
		L1	L2	L3	L4	L5
Teacher	AGNOSTIC	7.99 ± 0.02	0.17 ± 0.02	0.01 ± 0.00	0.01 ± 0.00	0.00 ± 0.00
	AWARE-BIL	8.00 ± 0.02	6.64 ± 0.17	4.87 ± 0.30	3.16 ± 0.27	1.31 ± 0.06

		Learner ($C_r = 10, C_c = 1$)				
		L1	L2	L3	L4	L5
Teacher	AGNOSTIC	8.36 ± 0.01	8.39 ± 0.02	8.46 ± 0.02	8.46 ± 0.02	8.49 ± 0.02
	AWARE-BIL	8.33 ± 0.01	8.36 ± 0.03	8.44 ± 0.02	8.44 ± 0.02	8.46 ± 0.02

		Learner ($C_r = 1, C_c = 10$)				
		L1	L2	L3	L4	L5
Teacher	AGNOSTIC	5.67 ± 0.02	0.15 ± 0.02	0.16 ± 0.02	0.11 ± 0.01	0.08 ± 0.01
	AWARE-BIL	5.93 ± 0.02	4.49 ± 0.15	3.56 ± 0.24	2.30 ± 0.22	0.93 ± 0.05

B.2 Teaching under unknown constraints (Section 6.2)

Here, we provide additional experimental results for teaching algorithms from Section 5. In particular, we report on the results for learner L1 and learner L3, similar to the results for learner L2 reported in Section 6.2.

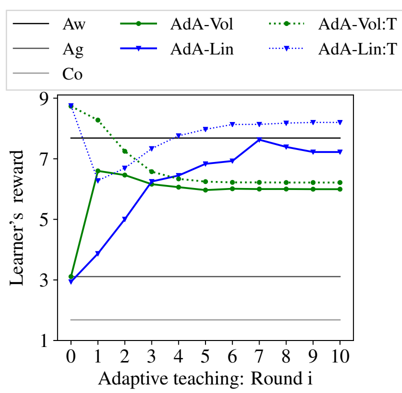


(a) Reward over teaching rounds

Teacher \ Env	10×10	15×15	20×20
AWARE-CMDP	8.42 ± 0.03	8.24 ± 0.05	7.84 ± 0.08
AGNOSTIC	8.42 ± 0.03	8.24 ± 0.05	7.84 ± 0.08
CONSERV	1.68 ± 0.1	1.66 ± 0.01	1.65 ± 0.02
ADAWARE-VOL (3rd)	8.04 ± 0.02	7.83 ± 0.04	7.46 ± 0.07
ADAWARE-VOL (end)	8.06 ± 0.02	7.80 ± 0.08	7.30 ± 0.12
ADAWARE-LIN (3rd)	8.44 ± 0.04	8.23 ± 0.07	8.08 ± 0.08
ADAWARE-LIN (end)	8.44 ± 0.04	8.23 ± 0.07	8.08 ± 0.08

(b) Varying grid-size

Figure 4: Results for learner L1



(a) Reward over teaching rounds

Teacher \ Env	10×10	15×15	20×20
AWARE-CMDP	7.68 ± 0.04	7.35 ± 0.03	7.39 ± 0.09
AGNOSTIC	3.11 ± 0.08	3.12 ± 0.07	3.26 ± 0.14
CONSERV	1.68 ± 0.01	1.65 ± 0.01	1.62 ± 0.01
ADAWARE-VOL (3rd)	6.16 ± 0.42	5.72 ± 0.54	6.39 ± 0.32
ADAWARE-VOL (end)	5.99 ± 0.46	5.38 ± 0.56	6.16 ± 0.31
ADAWARE-LIN (3rd)	6.25 ± 0.20	5.13 ± 0.50	6.15 ± 0.11
ADAWARE-LIN (end)	7.22 ± 0.16	5.83 ± 0.62	7.09 ± 0.07

(b) Varying grid-size

Figure 5: Results for learner L3

C Details for Learner-Aware Teaching under Unknown Constraints (Section 5)

In this appendix, we provide more details on the adaptive teaching algorithms ADAWARE-VOL and ADAWARE-LIN described in Sections 5.1 and 5.2. Recall that both teaching algorithms are obtained from Algorithm 1 by defining the way in which the teacher T adapts the teaching policy based on the learner L's feature expectations μ_r^{L} in past rounds.

C.1 Details for ADAWARE-VOL (Section 5.1)

Estimation of the learner's constraint set. In ADAWARE-VOL, T maintains an estimate $\hat{\Omega}_r^{\mathsf{L},i}$ of L's constraint set, starting with $\hat{\Omega}_r^{\mathsf{L},0} = \Omega_r$. After observing the feature expectations $\mu_r^{\mathsf{L},i}$ of the policy L found in round i, T updates this estimate as follows:

$$\hat{\Omega}_r^{\mathsf{L},i+1} \;:=\; \hat{\Omega}_r^{\mathsf{L},i} \cap \{\mu_r^{\mathsf{L},i} + \nu \in \mathbb{R}^{d_r} \mid \langle \mu_r^{\mathsf{T},i} - \mu_r^{\mathsf{L},i}, \nu \rangle \leq 0\} \tag{6}$$

The set on the right hand side of (6) with which $\Omega_r^{\mathsf{L},i}$ gets intersected is a halfspace containing Ω_r^{L}. This is due to the fact that Ω_r^{L} is convex by assumption, and to our assumption that L's learning algorithm is such that it outputs a policy whose feature expectations $\mu_r^{\mathsf{L},i}$ match the L^2-projection of $\mu_r^{\mathsf{T},i}$ to Ω_r^{L}. Inductively, it follows that $\hat{\Omega}_r^{\mathsf{L},i} \supset \Omega_r^{\mathsf{L}}$ for all i.

In practice, we implement a slightly modified version of the update step in which we intersect $\hat{\Omega}_r^{\mathsf{L},i}$ with a halfspace that is shifted in the direction of $\mu_r^{\mathsf{T},i} - \mu_r^{\mathsf{L},i}$ by a small amount, i.e., we use

$$\{\mu_r^{\mathsf{L},i} + (1-\eta)(\mu_r^{\mathsf{T},i} - \mu_r^{\mathsf{L},i}) + \nu \in \mathbb{R}^{d_r} \mid \langle \mu_r^{\mathsf{T},i} - \mu_r^{\mathsf{L},i}, \nu \rangle \leq 0\}$$

with a step size parameter $\eta \in (0,1)$. This helps make the algorithm more robust to noise in the learner's feature expectations. In our experiments, we used $\eta = 0.9$.

Update of the teaching policy. After updating the estimate of the learner's constraint set to $\hat{\Omega}_r^{\mathsf{L},i}$, T solves a constrained MDP in order to find

$$\pi^{\mathsf{T},i+1} \in \operatorname*{arg\,max}_{\pi, \mu_r(\pi) \in \hat{\Omega}_r^{\mathsf{L},i}} R(\pi).$$

Given that $\hat{\Omega}_r^{\mathsf{L},i}$ is cut out by linear equations, solving the constrained MDP reduces to solving an LP, as described in Appendix F.

Termination of the interaction. The algorithm terminates as soon as the stopping criterion $\|\mu_r^{\mathsf{L},i} - \mu_r^{\mathsf{T},i}\|_2 \leq \epsilon$ is satisfied. Note that $\hat{\Omega}_r^{\mathsf{L},i} \supset \Omega_r^{\mathsf{L}}$ implies that

$$R(\pi^{\mathsf{T},i}) \geq R(\pi^{\text{aware}})$$

for any $\pi^{\text{aware}} \in \arg\max_{\pi, \mu_r(\pi) \in \Omega_r^{\mathsf{L}}} R(\pi)$. Therefore, after termination we have

$$R(\pi^{\mathsf{L},i}) \geq R(\pi^{\text{aware}}) - \epsilon$$

for any policy π^{aware} which is optimal under L's constraints, which is the first statement of Theorem 2.

The second statement of Theorem 2 follows from the fact that if Ω_r^{L} is a convex polytope cut out by m linear inequalities, the number of faces, which is in $O(m^{d_r})$, is an upper bound on the number of iterations of the algorithm, because one face is "eliminated" in each round.

C.2 Details for ADAWARE-LIN (Section 5.2)

In ADAWARE-LIN, T updates the teaching policy $\pi^{\mathsf{T},i+1}$ based on L's feature expectations $\mu_r^{\mathsf{L},i}$ from the previous round. To do so, T uses LINESEARCH (Algorithm 2) to perform a binary search on the line segment

$$\{\mu_r^{\mathsf{L},i} + \alpha \mathbf{w}_r^* \mid \alpha \in [\alpha_{\min}, \alpha_{\max}]\} \subset \mathbb{R}^{d_r} \tag{7}$$

in order to find a vector μ_r that is realizable as the vector of feature expectations of a policy. If the intersection of the line segment (7) with Ω_r is non-empty, it is of the form $\{\mu_r^{\mathsf{L}} + \alpha \mathbf{w}_r^* \mid \alpha \in$

Algorithm 2 LINESEARCH

Require: $\mu_r^{\mathsf{L}}, \alpha_{\min}, \alpha_{\max}, \varepsilon_\alpha, \varepsilon_\mu$.

$\alpha_u \leftarrow \alpha_{\max}, \alpha_l \leftarrow \alpha_{\min}$
while $\alpha_u - \alpha_l > \varepsilon_\alpha$ **do**
  $\alpha \leftarrow (\alpha_u + \alpha_l)/2$
  $\pi^{\mathsf{T}} \leftarrow \mathrm{IRL}(\mu_r^{\mathsf{L}} + \alpha \mathbf{w}_r^*)$
  if $\|\mu_r(\pi^{\mathsf{T}}) - \mu_r^{\mathsf{L}} - \alpha \mathbf{w}_r^*\|_2 > \varepsilon_\mu$ **then**
    $\alpha_u \leftarrow \alpha$
  else
    $\alpha_l \leftarrow \alpha$
if $\|\mu_r(\pi^{\mathsf{T}}) - \mu_r^{\mathsf{L}} - \alpha \mathbf{w}_r^*\|_2 > \varepsilon_\mu$ **then**
  $\pi^{\mathsf{T}} \leftarrow \mathrm{IRL}(\mu_r^{\mathsf{L}} + \alpha_{\min} \mathbf{w}_r^*)$
return π^{T}

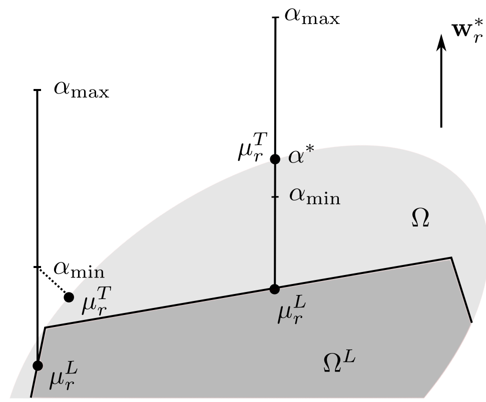


Figure 6

LINESEARCH is the algorithm that T uses in order to find a teaching policy π^{T} provided that the feature expectations of L's current policy are μ_r^{L}. Figure 6 illustrates the two cases may occur: For the right μ_r^{L}, LINESEARCH returns a policy π^{T} whose feature expectations satisfy $\mu_r^{\mathsf{T}} = \mu_r^{\mathsf{L}} + \alpha^* \mathbf{w}_r^*$ such that $\alpha^* > \alpha_{\min}$. For the left μ_r^{L}, LINESEARCH returns a policy π^{T} whose feature expectations satisfy $\mu_r^{\mathsf{T}} \in \arg\min_{\mu_r \in \Omega_r} \|\mu_r - \mu_r^{\mathsf{L}} + \alpha_{\min} \mu_r^{\mathsf{T}}\|$.

$[\alpha_{\min}, \alpha^*]\}$ for some $\alpha^* \leq \alpha_{\max}$ due to the convexity of Ω_r. In that case, LINESEARCH returns a policy with feature expectations

$$\mu_r^{\mathsf{T},i+1} = \mu_r^{\mathsf{L},i} + \alpha_i^* \mathbf{w}_r^*,$$

where α_i^* is the maximal $\alpha \in [\alpha_{\min}, \alpha_{\max}]$ such that $\mu_r^{\mathsf{L},i} + \alpha \mathbf{w}_r^* \in \Omega_r$. If the intersection is empty, LINESEARCH returns a policy with feature expectations

$$\mu_r^{\mathsf{T},i+1} \in \underset{\mu_r \in \Omega_r}{\arg\min} \|\mu_r - \mu_r^{\mathsf{L},i} - \alpha_{\min} \mathbf{w}_r^*\|_2.$$

Figure 6 illustrates the two cases that may occur.

C.2.1 Proof of Theorem 3

In this section, we provide the proof of Theorem 3, which gives a guarantee on the improvement of L's performance in each round of the ADAWARE-LIN algorithm. The assumption we make here is that, in every teaching round, LINESEARCH returns a teaching policy $\pi^{\mathsf{T},i+1}$ such that $\mu_r^{\mathsf{T},i+1} = \mu_r^{\mathsf{L},i} + \alpha_i \mathbf{w}_r^*$ for some $\alpha_i \geq \alpha_{\min}$, where $\alpha_{\min} > 0$ is a fixed constant. It is easy to see that this assumption, together with our assumption on L's algorithm and the convexity of Ω_r^{L}, imply that the change in learner performance

$$\Delta R_i := R(\mu_r^{\mathsf{L},i+1}) - R(\mu_r^{\mathsf{L},i})$$

is non-negative in every teaching round. The following proposition, which will be needed in the proof of Theorem 3, strengthens this statement:

Proposition 1. *Let $\overline{R}_{\mathsf{L}} := \max_{\mu_r \in \Omega_r^{\mathsf{L}}} R(\mu_r)$ be the maximally achievable learner performance. Assume that, in teaching round i, T can find a teaching policy $\pi^{\mathsf{T},i+1}$ whose feature expectations satisfy $\mu_r^{\mathsf{T},i+1} = \mu_r^{\mathsf{L},i} + \alpha_i \mathbf{w}_r^*$ for some $\alpha_i > 0$. Then*

$$\overline{R}_{\mathsf{L}} - R(\mu_r^{\mathsf{L},i}) \leq \Delta R_i + D \cdot \sqrt{\frac{\Delta R_i}{\alpha_i - \Delta R_i}}, \tag{8}$$

where $D = \operatorname{diam} \Omega_r$.

Proof of Proposition 1. Consider the plane $V \subset \mathbb{R}^{d_r}$ spanned by $\mu_r^{\mathsf{L},i}$, $\mu_r^{\mathsf{T},i+1}$ and $\mu_r^{\mathsf{L},i+1}$ and denote by $\tilde{\mu}_r$ the unique point in V with the properties that

(a) $\langle \mathbf{w}_r^*, \tilde{\mu}_r \rangle = \langle \mathbf{w}_r^*, \mu_r^{\mathsf{L},i+1} \rangle$,

(b) $\tilde{\mu}_r$ lies on the same side of the line through $\mu^{\mathsf{L},i}$ and $\mu^{\mathsf{T},i+1}$ as $\mu_r^{\mathsf{L},i+1}$, and

(c) $\tilde{\mu}_r, \mu_r^{\mathsf{T},i+1}$ and $\mu_r^{\mathsf{L},i}$ span a right triangle with $\tilde{\mu}_r$ at the right-angled corner.

Note that $\mu_r^{\mathsf{L},i+1}$ must lie inside this triangle, i.e., on the red line segment in Figure 7: Otherwise there would a point on the line segment connecting $\mu_r^{\mathsf{L},i+1}$ and $\mu_r^{\mathsf{L},i}$, and hence in Ω_r^{L} by convexity, which is closer to $\mu_r^{\mathsf{T},i+1}$ than $\mu_r^{\mathsf{L},i+1}$, contradicting the fact that $\mu_r^{\mathsf{L},i+1}$ is closest to $\mu_r^{\mathsf{T},i+1}$ among all points in Ω_r^{L}. Denote by $\tilde{\ell}$ the line passing through $\tilde{\mu}_r$ and $\mu_r^{\mathsf{L},i}$.

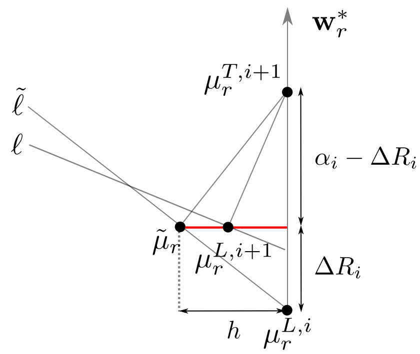


Figure 7: Illustration of the proof of Proposition 1: The smaller the performance increase ΔR_i, the better the upper bound on the gap $\overline{R}_\Omega - R(\mu_r^{\mathsf{L},i})$.

The facts that Ω_r^{L} is convex and that $\mu_r^{\mathsf{L},i+1} = \arg\min_{\mu_r \in \Omega_r^{\mathsf{L}}} \|\mu_r^{\mathsf{T},i+1} - \mu_r\|_2$ imply that Ω_r^{L} must lie on one side of the hyperplane

$$\mu_r^{\mathsf{L},i+1} + (\mu_r^{\mathsf{T},i+1} - \mu_r^{\mathsf{L},i+1})^\perp \subset \mathbb{R}^{d_r}.$$

Therefore, we can upper bound $\overline{R}_{\mathsf{L}}$ in terms of the slope s_ℓ of the line ℓ which arises by intersecting that hyperplane with V:

$$\overline{R}_{\mathsf{L}} \leq R(\mu_r^{\mathsf{L},i+1}) + D \cdot s_\ell = R(\mu_r^{\mathsf{L},i}) + \Delta R_i + D \cdot s_\ell. \tag{9}$$

Note that the slope s_ℓ is upper bounded by the slope $s_{\tilde{\ell}}$ of $\tilde{\ell}$. We have $s_{\tilde{\ell}} = \frac{\Delta R_i}{h}$, where h is the length of the red line segment in Figure 7, and $h = \sqrt{(\alpha_i - \Delta R_i)\Delta R_i}$ by Pythagoras's theorem. Using that, we obtain

$$s_\ell \leq s_{\tilde{\ell}} = \sqrt{\frac{\Delta R_i}{\alpha_i - \Delta R_i}}. \tag{10}$$

The claimed estimate (8) follows by plugging this upper bound for s into (9) and rearranging. $\square$

Proof of Theorem 3.

Proof of Theorem 3. The fact that $R(\mu_r^{\mathsf{L},i+1}) > R(\mu_r^{\mathsf{L},i})$, which is equivalent to $\Delta R_i > 0$, follows immediately from Proposition 1.

We now prove the claimed rate of convergence.

First, using Proposition 1, we note that the assumption that $\overline{R}_{\mathsf{L}} - R(\mu_r^{\mathsf{L},i}) > \varepsilon$ implies that

$$\varepsilon < \Delta R_i + D\sqrt{\frac{\Delta R_i}{\alpha_i - \Delta R_i}}. \tag{11}$$

Using that, we can conclude that

$$\sqrt{\Delta R_i} > \min\{\sqrt{\varepsilon/2}, \varepsilon\sqrt{\alpha_{\min}/(4D^2 + \varepsilon^2)}\}. \tag{12}$$

Indeed, if $\Delta R_i \leq \frac{\varepsilon}{2}$, it follows from (11) that we must have $D \cdot \sqrt{\Delta R_i/(\alpha_{\min} - \Delta R_i)} > \frac{\varepsilon}{2}$, which implies $\sqrt{\Delta R_i} > \varepsilon\sqrt{\alpha_{\min}/(4D^2 + \varepsilon^2)}$. Since we are interested in the behavior as $\varepsilon \to 0$, we assume from now on that ε is so small that $\varepsilon\sqrt{\alpha_{\min}/(4D^2 + \varepsilon^2)} < \sqrt{\varepsilon/2}$, so that (12) becomes

$$\sqrt{\Delta R_i} > \varepsilon\sqrt{\alpha_{\min}/(4D^2 + \varepsilon^2)} =: C_0. \tag{13}$$

Second, we observe that

$$\sqrt{\alpha_i - \Delta R_i} > \sqrt{\frac{\alpha_{\min}}{2}} =: C_1 \tag{14}$$

except in at most $N := \frac{2}{\alpha_{\min}}(\max R|_\Omega - \min R|_\Omega)$ teaching steps. To see that, note that if the claimed inequality, which is equivalent to $\alpha_i - \frac{\alpha_{\min}}{2} > \Delta R_i$, does not hold, performance increases by at least $\Delta R_i \geq \frac{\alpha_{\min}}{2}$ as $\alpha_i > \alpha_{\min}$, and that can happen at most N times.

The inequalities (13) and (14) together imply that we have

$$C_0 \cdot C_1 \leq \sqrt{(\alpha_i - \Delta R_i)\Delta R_i} \tag{15}$$

as long as $\overline{R}_{\mathsf{L}} - R(\mu_r^{\mathsf{L},i}) > \varepsilon$, except in at most N teaching steps. Setting $C := \frac{1}{C_0 \cdot C_1}$, this is equivalent to

$$\sqrt{\frac{\Delta R_i}{\alpha_i - \Delta R_i}} \leq C \Delta R_i \tag{16}$$

Plugging (16) into the bound (8) provided by Proposition 1, we obtain the estimate

$$\frac{1}{1 + CD}(\overline{R}_{\mathsf{L}} - R(\mu_r^{\mathsf{L},i})) \leq \Delta R_i. \tag{17}$$

We have $C = \frac{1}{\varepsilon \alpha_{\min}} \sqrt{2(4D^2 + \varepsilon^2)}$, and hence

$$\frac{1}{1 + CD} = \frac{\varepsilon \alpha_{\min}}{\varepsilon \alpha_{\min} + \sqrt{2(4D^2 + \varepsilon^2)} \cdot D} \geq \frac{1}{1 + \sqrt{10}} \frac{\varepsilon \alpha_{\min}}{D^2} =: \lambda \tag{18}$$

If we had the estimates (17), (18) for *all* teaching steps, we could conclude that the learner performance satisfies $R(\mu_r^{\mathsf{L},i}) > \overline{R}_{\mathsf{L}} - 2\varepsilon$ after at most $O(\frac{D^2}{\varepsilon \alpha_{\min}} \log \frac{D}{\varepsilon})$ teaching steps. One can see that e.g. by comparing the sequence $R_0, R_1, R_2, \ldots$ with the solution $R(t)$ of the ordinary differential equation $\dot{R} = \lambda(\overline{R}_{\mathsf{L}} - R)$, which satifies $\overline{R}_{\mathsf{L}} - R(t) = (\overline{R}_{\mathsf{L}} - R(0)) \exp(-\lambda t)$. Since the number N of teaching steps for which (17), (18) do potentially *not* hold is $O(\frac{D}{\alpha_{\min}})$, we can still make this conclusion. $\square$

D Background on (discounted) MCE-IRL Problem (Section 3)

Our learner models build on the (discounted) Maximum Causal Entropy (MCE) IRL framework [Ziebart et al., 2008, Ziebart, 2010, Ziebart et al., 2013, Zhou et al., 2018]. The results below are based on the MDCE-IRL formulation from [Zhou et al., 2018].

D.1 Primal problem

In the standard (discounted) MCE-IRL framework, a learning agent aims to identify a policy that matches the feature expectations of the teacher's demonstrations while simultaneously maximizing the (discounted) causal entropy of the policy, i.e., the learner solves the following optimization problem:

$$\begin{aligned} \max_{\pi} \quad & H^\gamma(A_{0:\infty} \| S_{0:\infty}) := \sum_{t=0}^{\infty} \gamma^t \mathbb{E}\Big[- \log \pi(a_t \mid s_t) \Big] \\ \text{subject to} \quad & \mu_r(\pi)[i] = \hat{\mu}_r(\Xi^\mathsf{T})[i] \quad \forall i \in \{1, 2, \ldots, d_r\}. \end{aligned}$$

Here, $\mu_r(\pi)[i]$ and $\hat{\mu}_r(\Xi^\mathsf{T})[i]$ denote the scalar values of the i^{th} reward feature. The idea is that without any further information beyond the teacher's demonstrations, the most uncertain solution matching the reward feature expectation of those demonstrations should be preferred.

Formulating this as a minimization problem and spelling out all the constraints, we arrive at the following primal:

$$\min_{\boldsymbol{\pi} = \{\pi_t\}_{t=0}^{\infty}} \; - H^\gamma(A_{0:\infty} \| S_{0:\infty})$$

subject to

$$\mu_r(\pi_t)[i] = \hat{\mu}_r(\Xi^{\mathsf{T}})[i] \quad \forall i \in \{1, 2, \ldots, d_r\}$$
$$\pi_t(a|s) \geq 0 \quad \forall a \in \mathcal{A}, s \in \mathcal{S}, t \geq 0$$
$$\sum_{a \in \mathcal{A}} \pi_t(a|s) = 1 \quad \forall s \in \mathcal{S}, t \geq 0$$
$$\pi_t(a|s) = \pi_{t'}(a|s) \quad \forall a \in \mathcal{A}, s \in \mathcal{S}, t \geq 0, t' \geq 0$$

The last condition ensures that the policy π is stationary.

D.2 Lagrangian relaxation

The Lagrangian relaxation optimization formulation of the above primal problem is given by

$$\mathcal{L}(\boldsymbol{\pi}, \boldsymbol{\lambda}, \boldsymbol{\psi}) = -H^{\gamma}(A_{0:\infty} \| S_{0:\infty}) + \boldsymbol{\lambda}^{\dagger}(\hat{\mu}_r(\Xi^{\mathsf{T}}) - \mu_r(\pi_t)) + \sum_{s,t} \psi_{s,t}(1 - \sum_{a \in \mathcal{A}} \pi_t(a|s))$$

subject to

$$\pi_t(a|s) \geq 0 \quad \forall a \in \mathcal{A}, s \in \mathcal{S}, t \geq 0$$
$$\pi_t(a|s) = \pi_{t'}(a|s) \quad \forall a \in \mathcal{A}, s \in \mathcal{S}, t, t' \geq 0$$

Here, $\boldsymbol{\lambda} \in \mathbb{R}^{d_r}$ and $\boldsymbol{\psi} = \{\psi_{s,t}\}_{\forall s_t}$. Also, $\dagger$ is the transpose operator defined for vectors.

Remark. The Lagrangian relaxation of the optimization problem is not convex in the problem variables because of the term $\boldsymbol{\lambda}^{\dagger}(\hat{\mu}_r(\Xi^{\mathsf{T}}) - \mu_r(\pi_t))$ in the objective function, which is not convex in the variables π_t. However, it can be shown that strong duality holds for both its dual and primal formulations ([Zhou et al., 2018]). The dual formulation is described in Section D.4.

D.3 Parametric form of the policy

For a given $\boldsymbol{\lambda}$, the optimal policy $\pi^{\text{soft}}_{\boldsymbol{\lambda}}(a|s)$ is given by

$$\pi^{\text{soft}}_{\boldsymbol{\lambda}}(a|s) = \frac{\exp(Q^{\text{soft}}_{\boldsymbol{\lambda}}(s,a))}{\exp(V^{\text{soft}}_{\boldsymbol{\lambda}}(s))}$$

where the quantities are defined recursively as follows:

$$Q^{\text{soft}}_{\boldsymbol{\lambda}}(s,a) = \boldsymbol{\lambda}^{\dagger}\mu_r(\pi^{\text{soft}}_{\boldsymbol{\lambda}}(a|s)) + \gamma \sum_{s' \in \mathcal{S}} T(s'|s,a) V^{\text{soft}}_{\boldsymbol{\lambda}}(s')$$
$$V^{\text{soft}}_{\boldsymbol{\lambda}}(s) = \log \sum_{a \in \mathcal{A}} \exp(Q^{\text{soft}}_{\boldsymbol{\lambda}}(s,a))$$

This is shown by taking the derivative of the Lagrangian, $\mathcal{L}(\boldsymbol{\pi}, \boldsymbol{\lambda}, \boldsymbol{\psi})$ w.r.t. the primal variables π_t and equating it to 0, i.e.,

$$\frac{\partial L(\{\pi_t\}_{t=0}^{\infty}, \boldsymbol{\lambda}, \boldsymbol{\psi})}{\partial \pi_t} = 0.$$

For a given $\boldsymbol{\lambda}$, the corresponding softmax policy can be obtained by *Soft-Value-Iteration* procedure (see [Ziebart, 2010, Algorithm. 9.1], [Zhou et al., 2018]).

D.4 Dual problem

For any given $\boldsymbol{\lambda}, \boldsymbol{\psi}$, let $g(\boldsymbol{\lambda}, \boldsymbol{\psi})$ be the optimal value for the optimization problem defined by the Lagrangian relaxation problem in Section D.2. As *strong duality* holds for the (discounted) MCE-IRL problem and its dual counter part, we solve only the following concave dual problem:

$$\underset{\boldsymbol{\lambda} \in \mathbb{R}^{d_r}, \psi_{s,t} \in \mathbb{R}}{\text{maximize}} \quad g(\boldsymbol{\lambda}, \boldsymbol{\psi})$$

D.5 Gradients for the dual variables

As the dual problem is concave, it can be solved using gradient ascent. The gradients of the dual function described in Section D.4 are given by:

$$\begin{aligned}\nabla_{\boldsymbol{\lambda}} \quad & g = \hat{\mu}_r(\Xi^{\mathsf{T}}) - \mu_r(\pi^{\text{soft}}_{\boldsymbol{\lambda}}) \\ \nabla_{\psi_{s,t}} \quad & g = 1 - \sum_{a \in \mathcal{A}} \pi^{\text{soft}}_{\boldsymbol{\lambda}}(a|s)\end{aligned}$$

Here $\pi^{\text{soft}}_{\boldsymbol{\lambda}}$ is the parametric softmax policy described above. The second condition is automatically satisfied because $\pi^{\text{soft}}_{\boldsymbol{\lambda}}$ is a probability distribution.

The gradient update rule to compute the optimal $\boldsymbol{\lambda}$ is:

$$\boldsymbol{\lambda}_{\text{next}} \leftarrow \boldsymbol{\lambda} - \eta \cdot \left(\mu_r(\pi^{\text{soft}}_{\boldsymbol{\lambda}}) - \hat{\mu}_r(\Xi^{\mathsf{T}})\right)$$

where η is the learning rate.

E Details of (discounted) MCE-IRL Problem with Preferences (Section 3.2)

Here we present the background of the learner model described in Section 3.2. In this setting, the learner's preferences are modeled as linear soft constraints with L1 penalties. We consider the minimization variant of the problem. The results in this section follow directly from the analysis of Maximum Entropy Models under different constraints, as presented in [Kazama and Tsujii, 2005, Dudík et al., 2007] when applied to (discounted) MCE-IRL problem [Ziebart et al., 2013, Zhou et al., 2018]. For brevity, redundant details of the derivations are omitted. The final policy of the learner is given by $\pi^{\text{soft}}_{\boldsymbol{\lambda}}$ and is defined in Section E.3.

E.1 Primal problem

The primal problem is given by

$$\min_{\boldsymbol{\pi}=\{\pi_t\}_{t=0}^{\infty};\ \delta_r^{\text{soft,low}},\ \delta_r^{\text{soft,up}},\ \delta_c^{\text{soft,up}} \geq 0} - H^{\gamma}(A_{0:\infty}||S_{0:\infty}) + \sum_{i=1}^{d_r} C_r \cdot (\delta_r^{\text{soft,low}}[i] + \delta_r^{\text{soft,up}}[i]) + \sum_{j=1}^{d_c} C_c \cdot \delta_c^{\text{soft,up}}[j]$$

subject to

$$\begin{aligned}&\hat{\mu}_r(\Xi^{\mathsf{T}})[i] - \mu_r(\pi_t)[i] \leq \delta_r^{\text{soft,low}}[i] \quad \forall i \in \{1, 2, \dots, d_r\} \\ &\mu_r(\pi_t)[i] - \hat{\mu}_r(\Xi^{\mathsf{T}})[i] \leq \delta_r^{\text{soft,up}}[i] \quad \forall i \in \{1, 2, \dots, d_r\} \\ &\mu_c(\pi_t)[j] \leq \delta_c^{\text{hard}}[j] + \delta_c^{\text{soft,up}}[j] \quad \forall j \in \{1, 2, \dots, d_c\}\end{aligned}$$

Here we have $\delta_r^{\text{soft,low}}, \delta_r^{\text{soft,up}} \in \mathbb{R}^{d_r}$ and $\delta_c^{\text{soft,up}} \in \mathbb{R}^{d_c}$ as the primal optimization slack variables with the constraint that $\delta_r^{\text{soft,low}}, \delta_r^{\text{soft,up}}, \delta_c^{\text{soft,up}} \geq 0$. We also have $C_r > 0, C_c > 0$. $\delta_c^{\text{hard}} \in \mathbb{R}^{d_c}$ is a given constant vector.

Remark. low and up in the superscripts of dual variables represent whether they are variables for lower bound constraints or upper bound constraints.

E.2 Lagrangian relaxation

The Lagrangian relaxation optimization formulation of the primal problem described in Section E.1 is given by

$$\begin{aligned}\mathcal{L}(\boldsymbol{\pi}, \delta_r^{\text{soft,low}}, \delta_r^{\text{soft,up}}, \delta_c^{\text{soft,up}}, \boldsymbol{\lambda}, \boldsymbol{\psi}) = & -H^{\gamma}(A_{0:\infty}, S_{0:\infty}) + (\boldsymbol{\alpha}^{\text{low}} - \boldsymbol{\alpha}^{\text{up}})^{\dagger}(\hat{\mu}_r(\Xi^{\mathsf{T}}) - \mu_r(\pi_t)) \\ & + \boldsymbol{\beta}^{\dagger}\mu_c(\pi_t) \\ & + \sum_{s,t} \psi_{s,t}(1 - \sum_{a \in \mathcal{A}} \pi_t(a|s)) - (\boldsymbol{\alpha}^{\text{low}})^{\dagger}\delta_r^{\text{soft,low}} - (\boldsymbol{\alpha}^{\text{up}})^{\dagger}\delta_r^{\text{soft,up}} \\ & - \boldsymbol{\beta}^{\dagger}\delta_c^{\text{soft,up}} - \boldsymbol{\beta}^{\dagger}\delta_c^{\text{hard}}\end{aligned}$$

$$
\begin{aligned}
&- (\boldsymbol{\rho}^{\text{low}})^{\dagger}\delta_r^{\text{soft,low}} - (\boldsymbol{\rho}^{\text{up}})^{\dagger}\delta_r^{\text{soft,up}} \\
&- \boldsymbol{\sigma}^{\dagger}\delta_c^{\text{soft,up}} \\
&+ \sum_{i=1}^{d_r} C_r \cdot (\delta_r^{\text{soft,low}}[i] + \delta_r^{\text{soft,up}}[i]) + \sum_{j=1}^{d_c} C_c \cdot \delta_c^{\text{soft,up}}[j]
\end{aligned}
$$

subject to

$$
\begin{aligned}
&\pi_t(a|s) \geq 0 \quad \forall a \in \mathcal{A}, s \in \mathcal{S}, t \geq 0 \\
&\pi_t(a|s) = \pi_{t'}(a|s) \quad \forall a \in \mathcal{A}, s \in \mathcal{S}, t, t^{'} \geq 0
\end{aligned}
$$

Here, $\boldsymbol{\alpha}^{\text{low}}, \boldsymbol{\alpha}^{\text{up}}, \boldsymbol{\rho}^{low}, \boldsymbol{\rho}^{up} \in \mathbb{R}^{d_r}$, and $\boldsymbol{\beta}, \boldsymbol{\sigma} \in \mathbb{R}^{d_c}$. We also have non-negativity constraints on the dual variables: $\boldsymbol{\alpha}^{\text{low}}, \boldsymbol{\alpha}^{\text{up}}, \boldsymbol{\beta}, \boldsymbol{\rho}^{\text{low}}, \boldsymbol{\rho}^{\text{up}}, \boldsymbol{\sigma} \geq 0$. A few additional notes:

- For convenience, we will denote the group of dual variables as $\boldsymbol{\lambda} := \{\boldsymbol{\alpha}^{\text{low}}, \boldsymbol{\alpha}^{\text{up}}, \boldsymbol{\beta}, \boldsymbol{\rho}^{\text{low}}, \boldsymbol{\rho}^{\text{up}}, \boldsymbol{\sigma}\}$
- The reward parameter $\boldsymbol{w}_{\boldsymbol{\lambda}} = [(\boldsymbol{\alpha}^{\text{low}} - \boldsymbol{\alpha}^{\text{up}})^{\dagger}, -\boldsymbol{\beta}^{\dagger}]^{\dagger}$ is used to define the learner's reward function $R_{\boldsymbol{\lambda}}(s) = \langle \boldsymbol{w}_{\boldsymbol{\lambda}}, \phi(s) \rangle$.
- $\dagger$ is the transpose operator, defined for vectors.

E.3 Parametric form of the policy

For a given, $\boldsymbol{\lambda} := \{\boldsymbol{\alpha}^{\text{low}}, \boldsymbol{\alpha}^{\text{up}}, \boldsymbol{\beta}, \boldsymbol{\rho}^{\text{low}}, \boldsymbol{\rho}^{\text{up}}, \boldsymbol{\sigma}\}$, the optimal policy $\pi_{\boldsymbol{\lambda}}^{\text{soft}}(a|s)$ is given by

$$
\pi_{\boldsymbol{\lambda}}^{\text{soft}}(a|s) = \frac{\exp(Q_{\boldsymbol{\lambda}}^{\text{soft}}(s,a))}{\exp(V_{\boldsymbol{\lambda}}^{\text{soft}}(s))}
$$

where the quantities are defined recursively as follows:

$$
\begin{aligned}
Q_{\boldsymbol{\lambda}}^{\text{soft}}(s,a) &= (\boldsymbol{\alpha}_{low} - \boldsymbol{\alpha}_{up})^{\dagger}\mu_r(\pi_{\boldsymbol{\lambda}}^{\text{soft}}(a|s)) - \boldsymbol{\beta}^{\dagger}\mu_c(\pi_{\boldsymbol{\lambda}}^{\text{soft}}(a|s)) + \gamma \sum_{s^{'} \in \mathcal{S}} T(s^{'}|s,a) V_{\boldsymbol{\lambda}}^{\text{soft}}(s^{'}) \\
V_{\boldsymbol{\lambda}}^{\text{soft}}(s) &= \log \big(\sum_{a \in \mathcal{A}} \exp(Q_{\boldsymbol{\lambda}}^{\text{soft}}(s,a))\big)
\end{aligned}
$$

This is shown by taking the derivative of the Lagrangian, $\mathcal{L}(\boldsymbol{\pi}, \boldsymbol{\lambda}, \boldsymbol{\psi})$ w.r.t the primal variables, π_t and equating it to 0. i.e.

$$
\frac{\partial L(\{\pi_t\}_{t=0}^{\infty}, \boldsymbol{\lambda}, \boldsymbol{\psi})}{\partial \pi_t} = 0
$$

E.4 Updated Lagrangian

We find the partial derivatives of the Lagrangian defined in Section E.2 w.r.t all the primal variables, $\delta_r^{\text{soft,low}}, \delta_r^{\text{soft,up}}, \delta_c^{\text{soft,up}}$:

$$
\begin{aligned}
&\frac{\partial \mathcal{L}}{\partial \delta_r^{\text{soft,low}}[i]} = 0 \\
&\Rightarrow \alpha^{\text{low}}[i] = C_r - \rho^{\text{low}}[i] \\
\text{Also, } &\frac{\partial \mathcal{L}}{\partial \delta_r^{\text{soft,up}}} = 0 \\
&\Rightarrow \alpha^{\text{up}}[i] = C_r - \rho^{\text{up}}[i] \\
\text{And, } &\frac{\partial \mathcal{L}}{\partial \delta_c^{\text{soft,up}}} = 0 \\
&\Rightarrow \beta[i] = C_r - \sigma[i]
\end{aligned}
$$

The dual variables satisfy $\boldsymbol{\sigma}, \boldsymbol{\rho}^{\text{low}}, \boldsymbol{\rho}^{\text{up}} \geq 0$. Hence, the above conditions translate into the following constraints on the set of dual variables, $\boldsymbol{\alpha}^{\text{low}}, \boldsymbol{\alpha}^{\text{up}}, \boldsymbol{\beta}$:

$$\begin{aligned} 0 \leq \alpha^{\text{low}}[i] \leq C_r \quad \forall i \in \{1, 2, \dots, d_r\} \\ 0 \leq \alpha^{\text{up}}[i] \leq C_r \quad \forall i \in \{1, 2, \dots, d_r\} \\ 0 \leq \beta[j] \leq C_c \quad \forall j \in \{1, 2, \dots, d_c\} \end{aligned}$$

The updated Lagrangian now has these additional constraints and is given by:

$$\begin{aligned} \mathcal{L}(\boldsymbol{\pi}, \delta_r^{\text{soft,low}}, \delta_r^{\text{soft,up}}, \delta_c^{\text{soft,up}}, \boldsymbol{\lambda}, \boldsymbol{\psi}) = &-H^\gamma(A_{0:\infty}, S_{0:\infty}) + (\boldsymbol{\alpha}^{\text{low}} - \boldsymbol{\alpha}^{\text{up}})^\dagger(\hat{\mu}_r(\Xi^\mathsf{T}) - \mu_r(\pi_t)) + \boldsymbol{\beta}^\dagger \mu_c(\pi_t) \\ &+ \sum_{s,t} \psi_{s,t}(1 - \sum_{a \in \mathcal{A}} \pi_t(a|s)) - (\boldsymbol{\alpha}^{\text{low}})^\dagger \delta_r^{\text{soft,low}} - (\boldsymbol{\alpha}^{up})^\dagger \delta_r^{\text{soft,up}} \\ &- \boldsymbol{\beta}^\dagger \delta_c^{\text{soft,up}} - \boldsymbol{\beta}^\dagger \delta_c^{\text{hard}} \\ &- (\boldsymbol{\rho}^{\text{low}})^\dagger \delta_r^{\text{soft,low}} - (\boldsymbol{\rho}^{\text{up}})^\dagger \delta_r^{\text{soft,up}} \\ &- \boldsymbol{\sigma}^\dagger \delta_c^{\text{soft,up}} \\ &+ \sum_{i=1}^{d_r} C_r \cdot (\delta_r^{\text{soft,low}}[i] + \delta_r^{\text{soft,up}}[i]) + \sum_{j=1}^{d_c} C_c \cdot \delta_c^{\text{soft,up}}[j] \end{aligned}$$

subject to

$$\begin{aligned} &\pi_t(a|s) \geq 0 \quad \forall a \in \mathcal{A}, s \in \mathcal{S}, t \geq 0 \\ &\pi_t(a|s) = \pi_{t'}(a|s) \quad \forall a \in \mathcal{A}, s \in \mathcal{S}, t, t^{'} \geq 0 \\ &0 \leq \alpha^{\text{low}}[i] \leq C_r \quad \forall i \in \{1, 2, \dots, d_r\} \\ &0 \leq \alpha^{\text{up}}[i] \leq C_r \quad \forall i \in \{1, 2, \dots, d_r\} \\ &0 \leq \beta[j] \leq C_c \quad \forall j \in \{1, 2, \dots, d_c\} \end{aligned}$$

The set of dual variables becomes $\boldsymbol{\lambda} := \{\boldsymbol{\alpha}^{\text{low}}, \boldsymbol{\alpha}^{\text{up}}, \boldsymbol{\beta}\}$ and $\boldsymbol{\psi} = \{\psi_{s,t}\}_{\forall s_t}$.

E.5 Dual problem

For any given $\boldsymbol{\lambda}, \boldsymbol{\psi}$, let $g(\boldsymbol{\lambda}, \boldsymbol{\psi})$ be the optimal value for the Lagrangian relaxation problem. Strong Duality holds for both our primal and dual formulations, and the dual optimal policy is also optimal for the primal formulation. Hence, we solve the *concave* dual problem, given by

$$\underset{\boldsymbol{\alpha}^{\text{low}}, \boldsymbol{\alpha}^{\text{up}} \in \mathbb{R}^{d_r}, \boldsymbol{\beta} \in \mathbb{R}^{d_c}, \psi_{s,t} \in \mathbb{R}}{\text{maximize}} \; g(\boldsymbol{\lambda}, \boldsymbol{\psi})$$

subject to

$$\begin{aligned} &0 \leq \boldsymbol{\alpha}^{\text{low}} \leq C_r \\ &0 \leq \boldsymbol{\alpha}^{\text{up}} \leq C_r \\ &0 \leq \boldsymbol{\beta} \leq C_c \end{aligned}$$

where $\boldsymbol{\lambda} := \{\boldsymbol{\alpha}^{\text{low}}, \boldsymbol{\alpha}^{\text{up}}, \boldsymbol{\beta}\}$.

E.6 Gradients for the dual problem

As the dual problem is concave, it can be solved using gradient ascent.
Note that,

$$\nabla_{\psi_{s,t}} g = 1 - \sum_{a \in \mathcal{A}} \pi_{\boldsymbol{\lambda}}^{\text{soft}}(a|s)$$

Here $\pi_{\boldsymbol{\lambda}}^{\text{soft}}$ is the parametric softmax policy described above. This condition is automatically satisfied because $\pi_{\boldsymbol{\lambda}}^{\text{soft}}$ is a probability distribution. For the remaining dual variables, we have the following gradients:

$$\nabla_{\boldsymbol{\alpha}^{\text{low}}} g = \hat{\mu}_r(\Xi^\mathsf{T}) - \mu_r(\pi_{\boldsymbol{\lambda}}^{\text{soft}})$$

$$\nabla_{\boldsymbol{\alpha}^{\text{up}}} \quad g = \mu_r(\pi_{\boldsymbol{\lambda}}^{\text{soft}}) - \hat{\mu}_r(\Xi^{\mathsf{T}})$$
$$\nabla_{\boldsymbol{\beta}}\, g = \mu_c(\pi_{\boldsymbol{\lambda}}^{\text{soft}})$$

The (projected) gradient update rules to compute the optimal value of the dual variables $(\boldsymbol{\alpha}^{\text{low}}, \boldsymbol{\alpha}^{\text{up}}, \boldsymbol{\beta})$ are given by the following:

$$\begin{aligned}
\boldsymbol{\alpha}^{\text{low}}_{\text{next}} &\leftarrow \boldsymbol{\alpha}^{\text{low}} - \eta \cdot (\mu_r(\pi_{\boldsymbol{\lambda}}^{\text{soft}}) - \hat{\mu}_r(\Xi^{\mathsf{T}})) \\
\alpha^{\text{low}}_{\text{next}}[i] &\leftarrow \max(0, \alpha^{\text{low}}_{\text{next}}[i]) \quad \forall i \in \{1, 2, \dots, d_r\} \\
\alpha^{\text{low}}_{\text{next}}[i] &\leftarrow \min(C_r, \alpha^{\text{low}}_{\text{next}}[i]) \quad \forall i \in \{1, 2, \dots, d_r\} \\
\\
\boldsymbol{\alpha}^{\text{up}}_{\text{next}} &\leftarrow \boldsymbol{\alpha}^{\text{up}} - \eta \cdot (\hat{\mu}_r(\Xi^{\mathsf{T}}) - \mu_r(\pi_{\boldsymbol{\lambda}}^{\text{soft}})) \\
\alpha^{\text{up}}_{\text{next}}[i] &\leftarrow \max(0, \alpha^{\text{up}}_{\text{next}}[i]) \quad \forall i \in \{1, 2, \dots, d_r\} \\
\alpha^{\text{up}}_{\text{next}}[i] &\leftarrow \min(C_r, \alpha^{\text{up}}_{\text{next}}[i]) \quad \forall i \in \{1, 2, \dots, d_r\} \\
\\
\boldsymbol{\beta}_{\text{next}} &\leftarrow \boldsymbol{\beta} - \eta \cdot (-\mu_c(\pi_{\boldsymbol{\lambda}}^{\text{soft}})) \\
\beta_{\text{next}}[j] &\leftarrow \max(0, \beta_{\text{next}}[j]) \quad \forall j \in \{1, 2, \dots, d_c\} \\
\beta_{\text{next}[j]} &\leftarrow \min(C_c, \beta_{\text{next}}[j]) \quad \forall j \in \{1, 2, \dots, d_c\}
\end{aligned}$$

where η is the learning rate.

F LP Formulation for the Teacher AWARE-CMDP (Section 4.1)

The problem of finding optimal learner-aware teaching demonstrations for the learner in Section 3.1 with linear preferences can be formulated as the following linear program (based on the linear programming formulation for solving MDPs [De, 1960]):

$$\max_{z} \quad \sum_{s} \sum_{a} z(s,a) \langle \mathbf{w}_r^*, \phi_r(s) \rangle \tag{19}$$
$$\text{s.t.} \quad \sum_{a} z(s',a) = (1-\gamma) P_0(s') + \gamma \sum_{s} \sum_{a} T(s'|s,a) z(s,a) \quad \forall s' \tag{20}$$
$$z(s,a) \geq 0 \quad \forall s,a \tag{21}$$
$$\sum_{s} \sum_{a} z(s,a) \phi_c(s)[j] \leq \delta_c^{\text{hard}}[j] \quad \forall j \in \{1, 2, \dots, d_c\} \tag{22}$$

Here z is a vector of discounted state-action frequencies and $z(s,a)$ refers to state-action frequency for state s and action a. The constraints in (22) are the linear preference constraints. From the optimal solution of the LP, an optimal stochastic policy can be extracted by

$$\pi(s,a) := \frac{z(s,a)}{\sum_{a'} z(s,a')}. \tag{23}$$

G Bi-Level Optimization Approach (Section 4.2)

We only show the formalism for the most general bi-level problem for learners with linear preferences.

G.1 Using Dual (discounted) MCE-IRL formulation for the learner model in Section 3.2

The basic bi-level optimization problem that we aim to solve is the following:

$$\begin{aligned}
\max_{\pi^{\mathsf{T}}} \quad & R(\pi^{\mathsf{L}}) \\
\text{subject to} \quad & \pi^{\mathsf{L}} \in \arg\max_{\pi} \mathrm{IRL}(\pi, \mu(\pi^{\mathsf{T}})).
\end{aligned}$$

We will replace the lower-level problem, i.e., $\arg\max_\pi \text{IRL}(\pi, \mu(\pi^\mathsf{T}))$ with its Karush-Kuhn-Tucker conditions [Boyd and Vandenberghe, 2004, Sinha et al., 2018]. The lower-level problem in its dual formulation is given in Appendix E.5.

Omitting details and replacing $R(\pi_{\boldsymbol{\lambda}}) := \langle \mathbf{w}_r^*, \mu_r(\pi_{\boldsymbol{\lambda}})\rangle$, this yields problems of the following form:

$$\begin{aligned}\max_{\boldsymbol{\lambda}} \quad & \langle \mathbf{w}_r^*, \mu_r(\pi_{\boldsymbol{\lambda}})\rangle \\ \text{subject to:} \quad & \\ & 0 \leq \boldsymbol{\alpha}^{\text{low}} \leq C_r \\ & 0 \leq \boldsymbol{\alpha}^{\text{up}} \leq C_r \\ & 0 \leq \boldsymbol{\beta} \leq C_c \\ & \mu_c(\pi_{\boldsymbol{\lambda}}) \leq (\geq) \delta_c^{\text{hard}}\end{aligned}$$

where $\boldsymbol{\lambda} := \{\boldsymbol{\alpha}^{\text{low}}, \boldsymbol{\alpha}^{\text{up}}, \boldsymbol{\beta}\}$. Here $\pi_{\boldsymbol{\lambda}}$ corresponds to a *softmax* policy with a reward function $R_{\boldsymbol{\lambda}}(s) = \langle \boldsymbol{w}_{\boldsymbol{\lambda}}, \phi(s)\rangle$ for $\boldsymbol{w}_{\boldsymbol{\lambda}} = [(\boldsymbol{\alpha}^{\text{low}} - \boldsymbol{\alpha}^{\text{up}})^\dagger, -\boldsymbol{\beta}^\dagger]^\dagger$. Thus, finding optimal demonstrations means optimization over *softmax* teaching policies while respecting the learner's preferences.

G.1.1 Optimal solution

The cases of the above problem we can observe have to be solved separately and the best solution must be picked. That is, we find the following two solutions: (step i) $\boldsymbol{\lambda}_1^*$, and (step ii) $\boldsymbol{\lambda}_2^*$. Then pick the best $\boldsymbol{\lambda}^*$ in (step iii):

Step i: $\boldsymbol{\lambda}_1^*$ Compute optimal parameters $\boldsymbol{\lambda}_1^*$ by solving the following problem:

$$\begin{aligned}\max_{\lambda} \quad & \langle \mathbf{w}_r^*, \mu_r(\pi_{\boldsymbol{\lambda}})\rangle \\ \text{subject to:} \quad & \\ & 0 \leq \boldsymbol{\alpha}^{\text{low}} \leq C_r \\ & 0 \leq \boldsymbol{\alpha}^{\text{up}} \leq C_r \\ & 0 \leq \boldsymbol{\beta} \leq C_c \\ & \mu_c(\pi_{\boldsymbol{\lambda}}) \leq \delta_c^{\text{hard}}\end{aligned}$$

Step ii: $\boldsymbol{\lambda}_2^*$ Compute optimal parameters $\boldsymbol{\lambda}_2^*$ by solving the following problem:

$$\begin{aligned}\max_{\boldsymbol{\lambda}} \quad & \langle \mathbf{w}_r^*, \mu_r(\pi_{\boldsymbol{\lambda}})\rangle && (24) \\ \text{subject to:} \quad & && (25) \\ & 0 \leq \boldsymbol{\alpha}^{\text{low}} \leq C_r && (26) \\ & 0 \leq \boldsymbol{\alpha}^{\text{up}} \leq C_r && (27) \\ & \boldsymbol{\beta} = C_c && (28) \\ & \mu_c(\pi_{\boldsymbol{\lambda}}) \geq \delta_c^{\text{hard}} && (29)\end{aligned}$$

Step iii: $\boldsymbol{\lambda}^*$ Pick the best solution as

$$\boldsymbol{\lambda}^* = \arg\max_{\boldsymbol{\lambda} \in \{\boldsymbol{\lambda}_1^*, \boldsymbol{\lambda}_2^*\}} \langle \mathbf{w}_r^*, \mu_r(\pi_{\boldsymbol{\lambda}})\rangle$$

This provides the optimal policy for the teacher. The teacher then computes feature expectation of this policy and provide it to the learner.

G.2 Solving the above problem

We adopt a variant of the Frank-Wolfe algorithm [Jaggi, 2013] to solve the problems of the form:

$$\begin{aligned}\max_{\boldsymbol{\lambda}} \quad & R(\pi_{\boldsymbol{\lambda}}) := \langle \mathbf{w}_r^*, \mu_r(\pi_{\boldsymbol{\lambda}})\rangle && (30) \\ \text{subject to:} \quad & && (31)\end{aligned}$$

$$0 \leq \boldsymbol{\alpha}^{\text{low}} \leq C_r \tag{32}$$
$$0 \leq \boldsymbol{\alpha}^{\text{up}} \leq C_r \tag{33}$$
$$0 \leq \boldsymbol{\beta} \leq C_c \tag{34}$$
$$\mu_c(\pi_{\boldsymbol{\lambda}}) \leq (\geq) \delta_c^{\text{hard}} \tag{35}$$

In particular, we take the following steps to optimize the teaching policy $\pi_{\boldsymbol{\lambda}}$:

1. *Initialization.* Find a feasible starting point $\boldsymbol{\lambda}_0$
2. *Optimization.* For $t = 1, 2, \ldots$
   - Compute the gradient $\boldsymbol{g}_t = [\nabla_{\boldsymbol{\lambda}} R(\pi_{\boldsymbol{\lambda}})](\boldsymbol{\lambda}_{t-1})$ of the objective at $\boldsymbol{\lambda}_{t-1}$. In experiments we approximate the gradient using finite-differences.
   - Linearize the constraints $\mu_c(\pi_{\boldsymbol{\lambda}}) \leq (\geq) \delta_c^{\text{hard}}$ at $\boldsymbol{\lambda}_{t-1}$ as $\boldsymbol{b}_t + \boldsymbol{A}_t(\boldsymbol{\lambda} - \boldsymbol{\lambda}_{t-1}) \leq (\geq) \delta_c^{\text{hard}}$, where $\boldsymbol{b}_t = \mu_c(\pi_{\boldsymbol{\lambda}_{t-1}})$ and $\boldsymbol{A}_t = [\nabla_{\boldsymbol{\lambda}} \mu_c(\pi_{\boldsymbol{\lambda}})](\boldsymbol{\lambda}_{t-1})$. Again, we employ finite-differences to approximate this linearization. Clearly, we can reuse computation from the gradient estimation of the objective here to reduce computational demands.
   - Solve the direction-finding subproblem (a linear problem):

$$\max_{\boldsymbol{\gamma}} \quad \langle \boldsymbol{\gamma}, \boldsymbol{g}_t \rangle$$

subject to:

$$\begin{aligned} &0 \leq \boldsymbol{\alpha}^{\text{low}} \leq C_r \\ &0 \leq \boldsymbol{\alpha}^{\text{up}} \leq C_r \\ &0 \leq \boldsymbol{\beta} \leq C_c \\ &\boldsymbol{b}_t + \boldsymbol{A}_{t-1}(\boldsymbol{\lambda} - \boldsymbol{\lambda}_{t-1}) \leq (\geq) \delta_c^{\text{hard}} \end{aligned}$$

     with optimal solution $\boldsymbol{\gamma}_t^*$. Assuming that the linear approximation of the constraints is accurate locally, the directional vector $\boldsymbol{d}_t = \boldsymbol{\gamma}_t^* - \boldsymbol{\lambda}_{t-1}$ is an ascent direction.
   - Perform a line-search from $\boldsymbol{\lambda}_{t-1}$ to $\boldsymbol{\gamma}_t^*$ and let $\boldsymbol{\lambda}_t$ be the point that maximizes the line search.
   - Upon convergence, terminate the For loop.

Upon convergence of the algorithm, the teacher can use the final $\boldsymbol{\lambda}_t$ for teaching.

Remark. Observe that the above algorithm would reduce to the standard Frank-Wolfe algorithm with line-search in the case of linear inequalities only.