Understanding and Adopting Rational Behavior by Bellman Score Estimation - Appendix

A PROOFS

We restate the definition of the score-operator here to make it easier to go through the proofs.

Definition 2 *The **Bellman score operator** $\mathcal{G}_{\pi,\theta} : \mathbb{R}^{|\mathcal{X}\times\mathcal{A}|\times\dim(\theta)} \to \mathbb{R}^{|\mathcal{X}\times\mathcal{A}|\times\dim(\theta)}$ for a policy π and reward r_θ is defined on an input $J \in \mathbb{R}^{|\mathcal{X}\times\mathcal{A}|\times\dim(\theta)}$ as*

$$\mathcal{G}_{\pi,\theta} J = \nabla_\theta r_\theta + \gamma P_\pi J \ .$$

For all results we make the following weak assumption.

Assumption 1 *The parametric reward function $r_\theta : \Theta \to \mathbb{R}^{|\mathcal{X}\times\mathcal{A}|}$ is continuously differentiable with respect to θ. The considered MDPs satisfy standard regularity conditions from Bertsekas & Tsitsiklis (1995) to guarantee convergence of value iteration.*

We now dive straight into proving our first main result.

Theorem 1 *For all $t = 1, ..., T$ and any matrix $J \in \mathbb{R}^{|\mathcal{X}\times\mathcal{A}|\times\dim(\theta)}$*

$$\nabla_\theta Q^{\text{soft}}_{\theta,t} = \mathcal{G}_{\pi^{\text{soft}}_{\theta,t-1},\theta} ... \mathcal{G}_{\pi^{\text{soft}}_{\theta,0},\theta}(\nabla_\theta Q^{\text{soft}}_{\theta,0})$$
$$\nabla_\theta Q^{\text{soft}}_{\theta,\infty} = \mathcal{G}_{\pi^{\text{soft}}_{\theta,\infty},\theta}(\nabla_\theta Q^{\text{soft}}_{\theta,\infty}) = \lim_{k\to\infty} \mathcal{G}^k_{\pi^{\text{soft}}_{\theta,\infty},\theta} J$$

where, $\nabla_\theta Q^{\text{soft}}_{\theta,0} = \nabla_\theta r_\theta$. Furthermore, the Q-gradient satisfies:

$$\nabla_\theta Q^{\text{soft}}_{\theta,t}(x,a) = \mathbb{E}_{x',a'\sim\rho_{\pi^{\text{soft}}_{\theta,:t},t}(\cdot|x,a)}[\nabla_\theta r_\theta(x',a')]$$
$$\nabla_\theta Q^{\text{soft}}_{\theta,\infty}(x,a) = \mathbb{E}_{x',a'\sim\rho_{\pi^{\text{soft}}_{\theta,\infty}}(\cdot|x,a)}[\nabla_\theta r_\theta(x',a')]$$

Proof We first prove the finite-horizon results. $\nabla_\theta Q^{\text{soft}}_{\theta,0} = \nabla_\theta r_\theta$ holds trivially since $Q^{\text{soft}}_{\theta,0} = r_\theta$.

We start from the definition of the optimal soft Q-function from Eq. 2 and derive the gradient for each dimension of $Q^{\text{soft}}_{\theta,t+1} \in \mathbb{R}^{|\mathcal{X}\times\mathcal{A}|}$.

$$\begin{aligned}
\nabla_\theta Q^{\text{soft}}_{\theta,t+1}(x,a) &= \nabla_\theta\big(r_\theta(x,a) + \gamma\mathbb{E}_{x'\sim P(\cdot|x,a)}[\log\sum_{a'}\exp Q^{\text{soft}}_{\theta,t}(x',a')]\big) \\
&= \nabla_\theta r_\theta(x,a) + \nabla_\theta\gamma\mathbb{E}_{x'\sim P(\cdot|x,a)}[\log\sum_{a'}\exp Q^{\text{soft}}_{\theta,t}(x',a')] \\
&= \nabla_\theta r_\theta(x,a) + \gamma\mathbb{E}_{x'\sim P(\cdot|x,a)}[\nabla_\theta\log\sum_{a'}\exp Q^{\text{soft}}_{\theta,t}(x',a')] \\
&= \nabla_\theta r_\theta(x,a) + \gamma\mathbb{E}_{x'\sim P(\cdot|x,a)}[\frac{\sum_{a'}\nabla_\theta(\exp Q^{\text{soft}}_{\theta,t}(x',a'))}{\sum_{a''}\exp Q^{\text{soft}}_{\theta,t}(x',a'')}] \\
&= \nabla_\theta r_\theta(x,a) + \gamma\mathbb{E}_{x'\sim P(\cdot|x,a)}[\sum_{a'}\frac{\exp Q^{\text{soft}}_{\theta,t}(x',a')}{\sum_{a''}\exp Q^{\text{soft}}_{\theta,t}(x',a'')}\nabla_\theta Q^{\text{soft}}_{\theta,t}(x',a')] \\
&= \nabla_\theta r_\theta(x,a) + \gamma\mathbb{E}_{x'\sim P(\cdot|x,a)}[\sum_{a'}\pi^{\text{soft}}_{\theta,t}(a'|x')\nabla_\theta Q^{\text{soft}}_{\theta,t}(x',a')] \\
&= \nabla_\theta r_\theta(x,a) + \gamma\mathbb{E}_{\substack{x'\sim P(\cdot|x,a)\\ a'\sim\pi^{\text{soft}}_{\theta,t}(\cdot|x)}}[\nabla_\theta Q^{\text{soft}}_{\theta,t}(x',a')] \\
&= (\mathcal{G}_{\pi^{\text{soft}}_{\theta,t},\theta}\nabla_\theta Q^{\text{soft}}_{\theta,t})(x,a)
\end{aligned}$$

For $t = 1$, we have $Q^{\text{soft}}_{\theta,1} = \mathcal{G}_{\pi^{\text{soft}}_{\theta,0},\theta}\nabla_\theta Q^{\text{soft}}_{\theta,0}$.

For $t \geq 1$, we assume $\nabla_\theta Q^{\text{soft}}_{\theta,t}(x,a) = \mathcal{G}_{\pi^{\text{soft}}_{\theta,t-1},\theta}...\mathcal{G}_{\pi^{\text{soft}}_{\theta,0},\theta}\nabla_\theta Q^{\text{soft}}_{\theta,0}$. Then, by the above result we have

$$\begin{aligned}\nabla_\theta Q^{\text{soft}}_{\theta,t+1}(x,a) &= \mathcal{G}_{\pi^{\text{soft}}_{\theta,t},\theta}\nabla_\theta Q^{\text{soft}}_{\theta,t} \\ &= \mathcal{G}_{\pi^{\text{soft}}_{\theta,t},\theta}\mathcal{G}_{\pi^{\text{soft}}_{\theta,t-1},\theta}...\mathcal{G}_{\pi^{\text{soft}}_{\theta,0},\theta}\nabla_\theta Q^{\text{soft}}_{\theta,0}\end{aligned}$$

We now prove that the computed gradient is in fact equivalent to the conditional occupancy, i.e $\nabla_\theta Q^{\text{soft}}_{\theta,t}(x,a) = \mathbb{E}_{x',a'\sim\rho_{\pi^{\text{soft}}_{\theta,:t},t}(\cdot|x,a)}[\nabla_\theta r_\theta(x',a')]$ for all finite $t \geq 0$ or $\nabla_\theta Q^{\text{soft}}_{\theta,t} = \rho_{\pi^{\text{soft}}_{\theta,:t},t}\nabla_\theta r_\theta$ in vector notation where $\rho_{\pi^{\text{soft}}_{\theta,:t},t} \in \mathbb{R}^{|\mathcal{X}\times\mathcal{A}|\times|\mathcal{X}\times\mathcal{A}|}$ is overloaded to mean the matrix with each row being a conditional occupancy. (and not the unconditional occupancy)

For $t = 0$, $\nabla_\theta Q^{\text{soft}}_{\theta,0}(x,a) = \nabla_\theta r_\theta(x,a) = \mathbb{1}_{x,a}\nabla_\theta r_\theta = \mathbb{E}_{x',a'\sim\rho_{\pi^{\text{soft}}_{\theta,0},0}(\cdot|x,a)}[\nabla_\theta r_\theta(x',a')]$.

For $t > 0$, we assume $\nabla_\theta Q^{\text{soft}}_{\theta,t} = \rho_{\pi^{\text{soft}}_{\theta,:t},t}\nabla_\theta r_\theta$. Then,

$$\begin{aligned}\rho_{\pi^{\text{soft}}_{\theta,:t+1},t+1}(\cdot|x,a)\nabla_\theta r_\theta &= (\sum_{n=0}^{t+1}\gamma^n \mathbb{1}_{x,a}P^n_{\pi^{\text{soft}}_{\theta,:t+1}})(\nabla_\theta r_\theta) \\ &= (\mathbb{1}_{x,a} + \gamma\mathbb{1}_{x,a}P_{\pi^{\text{soft}}_{\theta,t}}\sum_{n=1}^{t+1}\gamma^{n-1}P^{n-1}_{\pi^{\text{soft}}_{\theta,:t}})(\nabla_\theta r_\theta) \\ &= (\mathbb{1}_{x,a} + \gamma\mathbb{1}_{x,a}P_{\pi^{\text{soft}}_{\theta,t}}\sum_{n=0}^{t}\gamma^{n}IP^{n}_{\pi^{\text{soft}}_{\theta,:t}})(\nabla_\theta r_\theta) \\ &= \nabla_\theta r_\theta(x,a) + \gamma\mathbb{1}_{x,a}P_{\pi^{\text{soft}}_{\theta,t}}\rho_{\pi^{\text{soft}}_{\theta,:t},t}\nabla_\theta r_\theta \\ &= \left(\mathcal{G}_{\pi^{\text{soft}}_{\theta,t},\theta}(\rho_{\pi^{\text{soft}}_{\theta,:t},t}\nabla_\theta r_\theta)\right)(x,a) \\ &= (\mathcal{G}_{\pi^{\text{soft}}_{\theta,t},\theta}\nabla_\theta Q^{\text{soft}}_{\theta,t})(x,a) \\ &= \nabla_\theta Q^{\text{soft}}_{\theta,t+1}(x,a)\end{aligned}$$

Where the last line holds by applying our inductive hypothesis and the previous score operator result.

Now we prove the infinite-horizon results. Notice that the Bellman score operator $\mathcal{G}_{\pi^{\text{soft}}_{r,\infty},\theta}$ is simply the standard Bellman backup operator with a multi-dimensional reward $\nabla_\theta r_\theta(x,a)$, i.e the i^{th} column output of $\mathcal{G}_{\pi,\theta}$ computes the Q-values for an agent acting according to π with the reward set to the i^{th} column of ∇r_θ. Thus, the standard contraction mapping proofs directly apply to show that $\mathcal{G}_{\pi^{\text{soft}}_{\theta,\infty},\theta}$ is a max-norm contraction along each of it's output columns (Bertsekas & Tsitsiklis, 1995). This proves that $\mathcal{G}_{\pi^{\text{soft}}_{\theta,\infty},\theta}$ converges to a unique fixed point for any starting matrix J. We will now show that infinite-horizon Q-gradient satisfies the fixed-point equation $\nabla_\theta Q^{\text{soft}}_{\theta,\infty} = \mathcal{G}_{\pi^{\text{soft}}_{\theta,\infty},\theta}(\nabla_\theta Q^{\text{soft}}_{\theta,\infty})$ by first showing that it is equivalent to the infinite-horizon conditional occupancy. To show this we use the implicit function theorem. Consider the function f defined as:

$$f(\theta, Q) = Q - \mathcal{T}_{\mathcal{H}}Q$$

where $\mathcal{T}_{\mathcal{H}}$ is the Bellman optimality operator defined in Eq. 2. Then, $Q^{\text{soft}}_{\theta,\infty}$ is implicitly defined as a function of θ at the point $f(\theta, Q) = 0$. It trivially holds that f is continuously differentiable in θ since it is linear in r_θ and r_θ is assumed to be continuously differentiable with respect to θ. We may thus apply the implicit function theorem (Bai et al., 2019) to compute the gradient of the fixed point

with respect to θ.

$$
\begin{aligned}
\nabla_\theta Q^{\text{soft}}_{\theta,\infty} &= -(\nabla_Q f(\theta, Q))^{-1}|_{Q=Q^{\text{soft}}_{\theta,\infty}} \nabla_\theta f(\theta, Q^{\text{soft}}_{\theta,\infty}) \\
&= -[I - \nabla_Q \mathcal{T}_{\mathcal{H}} Q|_{Q=Q^{\text{soft}}_{\theta,\infty}}]^{-1}(-\nabla_\theta r_\theta) \\
&= [I - \nabla_Q \mathcal{T}_{\mathcal{H}} Q|_{Q=Q^{\text{soft}}_{\theta,\infty}}]^{-1} \nabla_\theta r_\theta \\
&= \sum_{n=0}^{\infty} (\nabla_Q \mathcal{T}_{\mathcal{H}} Q|_{Q=Q^{\text{soft}}_{\theta,\infty}})^n \nabla_\theta r_\theta \\
&= \sum_{n=0}^{\infty} \gamma^n \begin{pmatrix} P(x_1|x_1,a_1)\frac{\exp Q^{\text{soft}}_{\theta,\infty}(x_1,a_1)}{\sum_a \exp Q^{\text{soft}}_{\theta,\infty}(x_1,a)} & \cdots & P(x_{|\mathcal{X}|}|x_1,a_1)\frac{\exp Q^{\text{soft}}_{\theta,\infty}(x_{|\mathcal{X}|},a_{|\mathcal{A}|})}{\sum_a \exp Q^{\text{soft}}_{\theta,\infty}(x_{|\mathcal{X}|},a)} \\ \vdots & \vdots & \vdots \\ P(x_1|x_{|\mathcal{X}|},a_{|\mathcal{A}|})\frac{\exp Q^{\text{soft}}_{\theta,\infty}(x_1,a_1)}{\sum_a \exp Q^{\text{soft}}_{\theta,\infty}(x_1,a)} & \cdots & P(x_{|\mathcal{X}|}|x_{|\mathcal{X}|},a_{|\mathcal{A}|})\frac{\exp Q^{\text{soft}}_{\theta,\infty}(x_{|\mathcal{X}|},a_{|\mathcal{A}|})}{\sum_a \exp Q^{\text{soft}}_{\theta,\infty}(x_{|\mathcal{X}|},a)} \end{pmatrix}^n \nabla_\theta r_\theta \\
&= \sum_{n=0}^{\infty} \gamma^n \begin{pmatrix} P(x_1|x_1,a_1)\pi^{\text{soft}}_{r,\infty}(a_1|x_1) & \cdots & P(x_{|\mathcal{X}|}|x_1,a_1)\pi^{\text{soft}}_{r,\infty}(a_{|\mathcal{A}|}|x_{|\mathcal{X}|}) \\ \vdots & \vdots & \vdots \\ P(x_1|x_{|\mathcal{X}|},a_{|\mathcal{A}|})\pi^{\text{soft}}_{r,\infty}(a_1|x_1) & \cdots & P(x_{|\mathcal{X}|}|x_{|\mathcal{X}|},a_{|\mathcal{A}|})\pi^{\text{soft}}_{r,\infty}(a_{|\mathcal{A}|}|x_{|\mathcal{X}|}) \end{pmatrix}^n \nabla_\theta r_\theta \\
&= (\sum_{n=0}^{\infty} \gamma^n P^n_{\pi^{\text{soft}}_{r,\infty}}) \nabla_\theta r_\theta \\
&= \rho_{\pi^{\text{soft}}_{\theta,\infty}} \nabla_\theta r_\theta
\end{aligned}
$$

This proves that $\nabla_\theta Q^{\text{soft}}_{\theta,\infty}(x,a)$ is equivalent to the infinite horizon conditional occupancy. We use this result to show that the fixed point equation for $\mathcal{G}_{\pi^{\text{soft}}_{\theta,\infty},\theta}$ is satisfied by $\nabla_\theta Q^{\text{soft}}_{\theta,\infty}(x,a)$.

$$
\begin{aligned}
(\mathcal{G}_{\pi^{\text{soft}}_{\theta,\infty},\theta} \nabla_\theta Q^{\text{soft}}_{\theta,\infty})(x,a) &= \big(\mathcal{G}_{\pi^{\text{soft}}_{\theta,\infty},\theta}(\rho_{\pi^{\text{soft}}_{\theta,\infty}} \nabla_\theta r_\theta)\big)(x,a) \\
&= \nabla_\theta r_\theta(x,a) + \gamma \mathbb{1}_{x,a} P_{\pi^{\text{soft}}_{\theta,\infty}} \rho_{\pi^{\text{soft}}_{\theta,\infty}} \nabla_\theta r_\theta \\
&= (\mathbb{1}_{x,a} + \gamma \mathbb{1}_{x,a} P_{\pi^{\text{soft}}_{\theta,\infty}} \sum_{n=0}^{\infty} \gamma^n I P^n_{\pi^{\text{soft}}_{\theta,\infty}})(\nabla_\theta r_\theta) \\
&= (\mathbb{1}_{x,a} + \gamma \mathbb{1}_{x,a} P_{\pi^{\text{soft}}_{\theta,\infty}} \sum_{n=1}^{\infty} \gamma^{n-1} P^{n-1}_{\pi^{\text{soft}}_{\theta,\infty}})(\nabla_\theta r_\theta) \\
&= (\sum_{n=0}^{\infty} \gamma^n \mathbb{1}_{x,a} P^n_{\pi^{\text{soft}}_{\theta,\infty}})(\nabla_\theta r_\theta) \\
&= \rho_{\pi^{\text{soft}}_{\theta,\infty}}(\cdot|x,a) \nabla_\theta r_\theta \\
&= \nabla_\theta Q^{\text{soft}}_{\theta,\infty}(x,a)
\end{aligned}
$$

We have thus shown that $\mathcal{G}_{\pi^{\text{soft}}_{\theta,\infty},\theta}$ converges to a unique fixed point and that $\nabla_\theta Q^{\text{soft}}_{\theta,\infty}(x,a)$ satisfies the fixed point equation for $\mathcal{G}_{\pi^{\text{soft}}_{\theta,\infty},\theta}$. Since the MDP satifies the regularity conditions for value-iteration to converge (Assumption 1) and $\mathcal{G}_{\pi^{\text{soft}}_{\theta,\infty},\theta}$ is simply a Bellman backup operator along each of its output columns, repeated application of the infinite-horizon score operator must converge to the true Q-gradient for any starting matrix J.

$$
\nabla_\theta Q^{\text{soft}}_{\theta,\infty}(x,a) = \lim_{k\to\infty} \mathcal{G}^k_{\pi^{\text{soft}}_{\theta,\infty},\theta} J
$$

■

Theorem 2 *The gradient of the MaxEntIRL objective of Eq. 6 is*

$$\nabla_\theta \mathbb{E}_{\tau\sim p^*}[\log p_\theta(\tau)] = \mathbb{E}_{x\sim P_0, a\sim\pi^*_T(\cdot|x), a'\sim\pi^{\text{soft}}_{\theta,T}(\cdot|x)}[\nabla_\theta Q_{\pi^*,T}(x,a) - \nabla_\theta Q_{\theta,T}(x,a')] \tag{7}$$

$$= \mathbb{E}_{x,a\sim\rho_{\pi^*,T}}[\nabla_\theta r_\theta(x,a)] - \mathbb{E}_{x',a'\sim\rho_{\pi^{\text{soft}}_\theta,T}}[\nabla_\theta r_\theta(x',a')]\,. \tag{8}$$

Proof We omit the discount factor as it can be absorbed into the reward as $r_{\theta,t}(x,a) = \gamma^t r_\theta(x,a)$.

$$\begin{aligned}
\mathbb{E}_{\tau\sim p^*}[\log p_\theta(\tau)] &= \mathbb{E}_{\tau\sim p^*}[\log P_0(x_0)\prod_{t=0}^{T}\pi_{\theta,T-t}(a_t|x_t)\prod_{t=0}^{T-1}P(s_{t+1}|x_t,a_t)] \\
&= \mathbb{E}_{\tau\sim p^*}[\sum_{t=0}^{T}\log\pi_{\theta,T-t}(a_t|x_t)] + c \\
&= \sum_{x_0,\ldots,a_T} p^*(x_0,a_0,\ldots,x_T,a_T)\big(\sum_{t=0}^{T}\log\pi_{\theta,T-t}(a_t|x_t)\big) + c \\
&= \sum_{t=0}^{T}\sum_{x_0,\ldots,a_T} p^*(x_0,a_0,\ldots,x_T,a_T)\log\pi_{\theta,T-t}(a_t|x_t) + c \\
&= \sum_{t=0}^{T}\sum_{x_t,a_t} p^*(x_t,a_t)\log\pi_{\theta,T-t}(a_t|x_t) + c \\
&= \sum_{t=0}^{T}\mathbb{E}_{x,a\sim p^*_t}[\log\pi_{\theta,T-t}(a|x)] + c
\end{aligned}$$

Now the magic happens when we take gradients:

$$\begin{aligned}
\nabla_\theta\mathbb{E}_{\tau\sim p^*}[\log p_\theta(\tau)] &= \nabla_\theta\Big(\sum_{t=0}^{T}\mathbb{E}_{x,a\sim p^*_t}[\log\pi_{\theta,T-t}(a|x)] + c\Big) \\
&= \sum_{t=0}^{T}\mathbb{E}_{x,a\sim p^*_t}[\nabla_\theta\log\pi_{\theta,T-t}(a|x)] \\
&= \sum_{t=0}^{T}\mathbb{E}_{x,a\sim p^*_t}[\frac{1}{\pi_{\theta,T-t}(a|x)}\cdot\nabla_\theta\pi_{\theta,T-t}(a|x)] \\
&= \sum_{t=0}^{T}\mathbb{E}_{x,a\sim p^*_t}[\frac{1}{\pi_{\theta,T-t}(a|x)}\cdot\nabla_\theta\frac{\exp Q_{\theta,T-t}(x,a)}{\sum_{a'}\exp Q_{\theta,T-t}(x,a')}] \\
&= \sum_{t=0}^{T}\mathbb{E}_{x,a\sim p^*_t}[\frac{1}{\pi_{\theta,T-t}(a|x)}\cdot\sum_{a'}\pi_{\theta,T-t}(a|x)(\mathbb{1}_{a'=a} - \pi_{\theta,T-t}(a'|x))\nabla_\theta Q_{\theta,T-t}(x,a')] \\
&= \sum_{t=0}^{T}\mathbb{E}_{x,a\sim p^*_t}[\nabla_\theta Q_{\theta,T-t}(x,a)] - \mathbb{E}_{\substack{x\sim p^*_t \\ a'\sim\pi_{\theta,T-t}(\cdot|x)}}[\nabla_\theta Q_{\theta,T-t}(x,a')] \\
&= \sum_{t=0}^{T}\mathbb{E}_{x,a\sim p^*_t}[\nabla_\theta r_\theta(x,a) + \mathbb{E}_{\substack{x'\sim P(\cdot|x,a) \\ \pi_{\theta,T-t-1}(a'|x')}}[\nabla_\theta Q_{\theta,T-t-1}(x',a')] - \mathbb{E}_{\substack{s\sim p^*_t \\ a'\sim\pi_{\theta,T-t}(\cdot|x)}}[\nabla_\theta Q_{\theta,T-t}(x,a')] \\
&= \sum_{t=0}^{T}\mathbb{E}_{x,a\sim p^*_t}[\nabla_\theta r_\theta(x,a)] + \mathbb{E}_{\substack{s'\sim p^*_{t+1} \\ \pi_{\theta,T-t-1}(a'|x')}}[\nabla_\theta Q_{\theta,T-t-1}(x,a)] - \mathbb{E}_{\substack{s\sim p^*_t \\ a'\sim\pi_{\theta,T-t}(\cdot|x)}}[\nabla_\theta Q_{\theta,T-t}(x,a')] \\
&= \Big(\sum_{t=0}^{T}\mathbb{E}_{x,a\sim p^*_t}[\nabla_\theta r_\theta(x,a)]\Big) - \mathbb{E}_{\substack{s\sim p^*_0 \\ a'\sim\pi_{\theta,T}(\cdot|x)}}[\nabla_\theta Q_{\theta,T}(x,a')] \\
&= \mathbb{E}_{\substack{x\sim P_0 \\ a\sim\pi^*(\cdot|x)}}[\nabla_\theta Q_{\pi^*,T}(x,a)] - \mathbb{E}_{\substack{x\sim P_0 \\ a\sim\pi_{\theta,T}(\cdot|x)}}[\nabla_\theta Q_{\theta,T}(x,a)]
\end{aligned}$$

In the deterministic dynamics case, this is precisely equal to the contrastive divergence gradient for the energy-based model

$$p_\theta(\tau) = \frac{\exp r_\theta(\tau)}{\sum_{\tau'} \exp r_\theta(\tau')}$$

■

In the case where the reward is a linear function of known features $\mathcal{F} : \mathcal{X} \times \mathcal{A} \to \mathbb{R}^{\dim(\theta)}$, i.e $r_\theta(x, a) = \theta \cdot F(x, a)$, we note that Eq. 8 simplifies into

$$\mathbb{E}_{x,a\sim\rho_{\pi^*,T}}[\mathcal{F}(x, a)] - \mathbb{E}_{x',a'\sim\rho_{\pi_\theta^{\text{soft}},T}}[\mathcal{F}(x, a)]$$

which is the result derived in Theorem 2 of Ziebart et al. (2010) for the simplified feature matching formulation of Maximum Causal Entropy IRL with linear reward models. Our result generalizes Theorem 2 of Ziebart et al. (2010) to the maximum likelihood formulation of MaxEntIRL with non-linear function approximators.

B Experiment Details

GAC details Table 4 shows a list of all GAC specific hyperparameters from Algorithms 2, 3. These hyperparameters were fixed throughout all state and visual control experiments.

For all state control tasks we use `pytorch_sac` (Yarats & Kostrikov, 2020) as the backbone RL in Algorithm 2. For all RL related modules (actor, critic, and etc.), we use their default network architectures and hyperparameters. (see `https://github.com/denisyarats/pytorch_sac`) The learner and expert Q-gradients g_{ψ_L}, g_{ψ_E} are represented by a single network with ReLU activations and two 1024 unit hidden layers. (same archiecture as the critic) The shared gradient network takes as input a one-hot indicator vector to switch between learner and expert. The reward r_θ is represented by a single layer network with 64 hidden units, ReLU activations, and a sigmoid output head to scale the reward between $[0, 1]$.

For all visual control tasks we use DrQ-v2 (Yarats et al., 2022) as the backbone RL and similarly use their default hyperparameters for all RL related modules. (see `https://github.com/facebookresearch/drqv2`) The expert and learner Q-gradients are again represented by a single network with the same architecture as the critic from (Yarats et al., 2022). The indicator vector is injected after the encoding layers. The actor, critic, Q-gradient, and reward all share an encoder but only the critic loss updates it. The implementation code will be released shortly.

Table 4: **GAC Hyperparameters** for both state, visual control problems

Parameter	Value
M: Total number GAC iterations	Until θ convergence
N: Number of score iteration steps per GAC iteration	1
N_θ: Reward update interval	20
N_ψ: Number of Q-gradient update steps per policy update	2
η_r: Reward learning rate	0.00001
η_ψ: Q-gradient learning rate	0.0001
η_ω: score-learning rate	0.0001
α_g: Target Q-gradient mixing rate	0.005
RL: One policy update step in the inner loop of an RL algorithm	SAC (state) DrQ-v2 (visual)

Baselines: For DAC Kostrikov et al. (2019), state-of-the-art results are reported in (Cohen et al., 2021), and so we follow their implementation which is already fine-tuned for DM control environments. GCL (Finn et al., 2016b) and AIRL (Fu et al., 2018) are equivalent to Generative Adversarial Imitation Learning (Ho & Ermon, 2016) with a different parametization of the discriminator Finn et al. (2016a). We use the implementation which at `https://github.com/HumanCompatibleAI/imitation` which builds off of stable-baselines 3 `https://github.com/DLR-RM/stable-baselines3`. Although these baseline implementations are already fine-tuned for DM control out of the box, we do an exhaustive hyperparameter search to maximize their performance. As fully listing out all hyperparameter values for every baseline is not very useful, we will instead release the code upon publication.

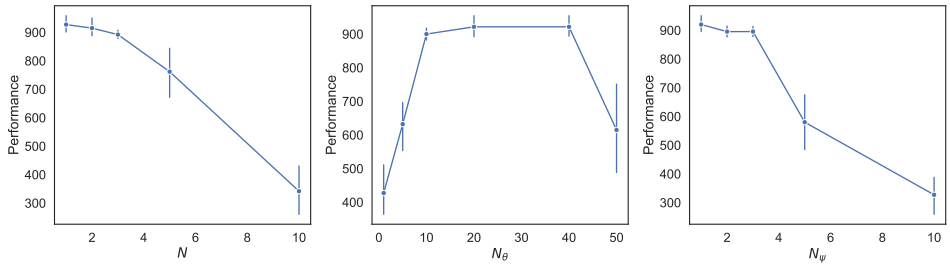

Figure 3: **State-control ablation results** when varying the number of score iteration steps per environment step N, reward update interval N_θ, and the number of Q-gradient update steps per policy update N_ψ. We vary each hyper-parameter while keeping the other parameter values set to those in Table 4. Performance at a given hyper-parameter value is measured as the average task performance across all nine state-control tasks in Table 2 (expert level is ≈ 1000 for all tasks), where ten random seeds are used to evaluate performance on a single task and one demonstration is supplied.

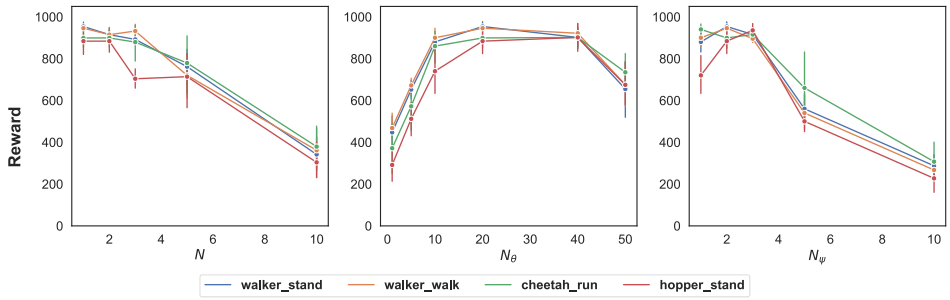

Figure 4: **Visual-control ablation results** when varying the number of score iteration steps per environment step N, reward update interval N_θ, and the number of Q-gradient update steps per policy update N_ψ. We vary each hyper-parameter while keeping the other parameter values set to those in Table 4. Performance at a given hyper-parameter value is measured as the average task performance across ten random seeds. Ten demonstrations were supplied for each task.

For the reward transfer experiments, we compare against AIRL (Fu et al., 2018) and GCL (Finn et al., 2016b). All three models learn state-only reward functions following (Fu et al., 2018) since Theorem 5.2 from Fu et al. (2018) shows that a disentangled reward (which is effective for transfer) must be state-only. Rewards are first learned on the non-perturbed environments using 10 demonstrations, then re-optimized via RL on the perturbed environments.

C Ablation Studies

We've conducted hyperparameter ablation experiments for three main hyperparameters, namely the number of score iteration steps per environment step N, reward update interval N_θ, and the number of Q-gradient update steps per policy update N_ψ. Results are shown in Figure 4. On the walker task, we found that IRL control performance begins to drop meaningfully (more than a standard deviation) past $N > 5$ (default is $N = 1$), which suggests that too many score iteration updates per environment sample collected leads to overfitting and hence poor RL performance. We found that the reward update interval N_θ should be kept within the relatively flexible range of $10 \leq N_\theta \leq 40$ (default is $N_\theta = 20$) in order to avoid drops in IRL performance. We posit that updating the rewards too frequently triggers instability in the RL algorithms that must use the fast changing reward to learn a policy. In contrast, if rewards are updated too infrequently, the algorithm takes an intractable amount of time to converge. Finally, the number of Q-gradient update steps per policy update N_ψ should be kept within $1 \leq N_\psi \leq 5$ (default is $N_\psi = 2$) to avoid IRL performance drop-off. Too many Q-gradient updates per policy update leads overfitting the Q-gradient network to the current policy/replay buffer, and also increases the compute time. Overall we found that our approach was not too sensitive to the choice of hyper-parameters.

D LIMITATIONS AND ADDITIONAL RESOURCES

D.1 LIMITATIONS AND FUTURE WORKS

An important limitation of score-learning (Algorithm 2) is the potential to suffer from the deadly triad (Van Hasselt et al., 2018). The deadly triad issue is the problem that the combination of off-policy learning, function approximations, and bootstrapping can hinder the convergence of value iteration algorithms such as Q-learning. In comparison to standard Q-learning, score-learning may be more heavily impacted by the deadly triad as it seeks to estimate a high dimensional gradient vector $\nabla_\theta Q^{\text{soft}}_{\theta,\infty}$ with the three aforementioned components. The probability of divergence of each dimension of the estimated gradient vector may compound to yield a much higher chance of overall algorithm divergence. We hope to see future works investigate this topic.

As score-learning (Algorithm 2) uses a replay buffer for offline-learning, it may also suffer from issues of double-sampling (Zhu & Ying, 2020). As our main experiments were in deterministic Mujoco environments, we may have not run into double-sampling issues. However, it's possible this problem may appear in more complex, stochastic environments and deserves further investigation.

Another limitation of GAC is that it requires more compute time than the baselines. This is mostly due to the computational burden of training an additional Q-gradient network which is updated more frequently than the policy net. We report the compute time benchmarking results in Table 5

Table 5: **GAC Compute Comparison** for both state, visual control problems. Run times are benchmarked on a single A5000 gpu. For state-control (second column), we report the mean and standard deviation of the run time (in hours) averaged across all tasks in Table 2 using ten random seeds per task. We do the same for visual-control (third column) with the tasks in Figure 2. Ten demonstrations are supplied for all tasks

Method	Run Time (state-control)	Run Time (visual-control)
AIRL (Fu et al., 2018)	2.4 ± 0.3	5.5 ± 0.6
DAC (Kostrikov et al., 2019)	2.1 ± 0.4	5.0 ± 0.7
GAC (Ours)	3.7 ± 0.6	6.1 ± 0.5

We find that the GAC requires roughly 1.7 times more time than the baselines on the state-control tasks while 1.2 times more on the visual-control tasks. This is because the Q-gradient and reward networks share the encoder with the actor-critic networks in the visual control experiments.

D.2 DETAILED PROBLEM DEFINITIONS

Here we concretely define the concepts of reward identification, counterfactual predictions, behavior imitation, and behavior transfer. All four problems fall under the broader scope of Learning from Demonstrations.

Reward Identification seeks to infer the underlying reward function of the expert given demonstrations of behavior sampled from the expert. Typically, the expert is assumed to act according to a policy that is optimal with respect to a MDP.

Counterfactual Prediction problems seek to infer the optimal policy of an agent if the reward were different from the reward optimized by the demonstrator. As the demonstrations contain information about the optimal policy solution for one reward, some of that information should be reused in order to more efficiently infer the optimal policy for the counterfactual reward.

Behavior Imitation seeks to infer a policy that matches the demonstrator's behavior policy. Humans frequently learn sports via Imitation Learning, e.g by copying a better athlete's form.

Behavior Transfer seeks to learn a policy that displays the same semantic behavior as the demonstrator under perturbed environmental conditions. For example, given demonstrations of a robot walking on a high-friction surface, we want to learn a policy that enables the robot to walk on a low-friction surface.

D.3 FINITE-HORIZON SCORE ITERATION

Here we give the finite horizon version of the score-iteration algorithm.

Algorithm 4 Score Iteration: Computes the finite-horizon Bellman score via dynamic programming

$\mathcal{M} = (\mathcal{X}, \mathcal{A}, P, P_0, r_\theta, \gamma)$: Markov Decision Process
T: Time-horizon
$\{g^{\text{soft}}_{\theta,t}\}_{t=0}^{T}$: Q-gradient vector
$\{s^{\text{soft}}_{\theta,t}\}_{t=0}^{T}$: Bellman score vector
procedure SCOREITERATION($\mathcal{M}, T$)
  For $\mathcal{M}$, learn the optimal policy $\{\pi^{\text{soft}}_{\theta,t}\}_{t=0}^{T}$
  Set $g^{\text{soft}}_{\theta,0}(x,a) = \nabla_\theta r_\theta(x,a)$ and $s^{\text{soft}}_{\theta,0}(x,a) \leftarrow g^{\text{soft}}_{\theta,0}(x,a) - \sum_{a' \in \mathcal{A}} \pi^{\text{soft}}_{\theta,0}(a'|x) g^{\text{soft}}_{\theta,0}(x,a')$
  for $t = 1, ..., T$ **do**
    for $x \in \mathcal{X}, a \in \mathcal{A}$ **do**
      Update Q-gradient: $g^{\text{soft}}_{\theta,t}(x,a) \leftarrow \nabla_\theta r_\theta(x,a) + \gamma \sum_{x',a'} P_{\pi^{\text{soft}}_{\theta,t-1}}(x',a'|x,a) g^{\text{soft}}_{\theta,t-1}(x',a')$
      Compute Bellman score: $s^{\text{soft}}_{\theta,t}(x,a) \leftarrow g^{\text{soft}}_{\theta,t}(x,a) - \sum_{a'} \pi^{\text{soft}}_{\theta,t}(a'|x) g^{\text{soft}}_{\theta,t}(x,a')$
  return $\{\pi^{\text{soft}}_{\theta,t}\}_{t=0}^{T}, \{g^{\text{soft}}_{\theta,t}\}_{t=0}^{T}, \{s^{\text{soft}}_{\theta,t}\}_{t=0}^{T}$