LS-IQ: IMPLICIT REWARD REGULARIZATION FOR INVERSE REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent methods for imitation learning directly learn a Q-function using an implicit reward formulation, rather than an explicit reward function. However, these methods generally require implicit reward regularization for improving stability, mistreating or even neglecting absorbing states. Previous works show that a squared norm regularization on the implicit reward function is effective, but do not provide a theoretical analysis of the resulting properties of the algorithms. In this work, we show that using this regularizer under a mixture distribution of the policy and the expert provides a particularly illuminating perspective: the original objective can be understood as squared Bellman error minimization, and the corresponding optimization problem minimizes the χ^2-Divergence between the expert and the mixture distribution. This perspective allows us to address instabilities and properly treat absorbing states. We show that our method, Least Squares Inverse Q-Learning (LS-IQ), outperforms state-of-the-art algorithms, particularly in environments with absorbing states. Finally, we propose to use an inverse dynamics model to learn from observations only. Using this approach, we retain performance in settings where no expert actions are available.

1 INTRODUCTION

Inverse Reinforcement Learning (IRL) techniques have been developed to extract behaviors from expert demonstration robustly and solve the problems of classical Imitation Learning (IL) methods (Ng et al., 1999; Ziebart et al., 2008). Among the recent methods for IRL, the Adversarial Imitation Learning (AIL) approach (Ho & Ermon, 2016; Fu et al., 2018; Peng et al., 2021), which casts the optimization over rewards and policies into an adversarial setting, have been proven particularly successful. These methods, inspired by Generative Adversarial Networks (GANs) (Goodfellow et al., 2014), alternate between learning a discriminator, and improving the agent's policy w.r.t. a reward function, computed based on the discriminator's output. These *explicit reward* methods require many interactions with the environment as they learn both a reward and a value function. Recently, *implicit reward* methods (Kostrikov et al., 2019; Arenz & Neumann, 2020; Garg et al., 2021) have been proposed. These methods directly learn the Q-function, significantly simplifying the policy optimization.

Among the *implicit reward* approaches, the Inverse soft Q-Learning (IQ-Learn) is the current state-of-the-art. This method modifies the distribution matching objective by including a reward regularization term on the expert distribution, which results in a minimization of the χ^2-divergence between the policy and the expert distribution. However, whereas their derivations only considered regularization on the expert's distribution, their practical implementations have shown that regularizing the reward on both the expert and policy distribution achieves significantly better performance.

The contribution of this paper is twofold: First, we show that the resulting objective when using this regularizer minimizes the χ^2 divergence between the expert and a mixture distribution between the expert and the policy. We then investigate the effects of regularizing w.r.t. the mixture distribution on the theoretical properties of IQ-Learn. This divergence is bounded, and we show that this translates to bounds on the Q-function and reward functions, allowing for more robust extrapolation from the training data. Indeed, the resulting objective corresponds to least-squared Bellman error minimization and is closely related to Soft Q-Imitation Learning (SQIL) (Reddy et al., 2020).

Second, we formulate Least Squares Inverse Q-Learning (LS-IQ), a novel IRL algorithm. By following the theoretical insight coming from the analysis of the χ^2 regularizer, we tackle many sources of instabilities of the IQ-Learn approach: the arbitrariness of the Q-function scales, exploding Q-functions targets, and reward bias Kostrikov et al. (2019) i.e., assuming that absorbing states provide the null reward. We derive the LS-IQ algorithm by exploiting structural properties of the Q-function and heuristics based on expert optimality. This results in increased performance on many tasks and, in general, more stable learning and less variance in the Q-function estimation. Finally, we extend the implicit reward methods to the IL from observation setting by training an Inverse-Dynamics Model (IDM) to predict the expert actions, which are no longer assumed to be available. Even in this challenging setting, our approach retains performance similar to the original setting.

Related Work. The vast majority of IRL and IL methods build upon the Maximum Entropy (MaxEnt) IRL framework (Ziebart, 2010). In particular, Ho & Ermon (2016) introduce Generative Adversarial Imitation Learning (GAIL), which applies GANs to the IL problem. While the original method minimizes the Jensen-Shannon divergence to the expert distribution, the approach was extended to general f-divergences (Ghasemipour et al., 2019), building on the work of Nowozin et al. (2016). Among the f-divergences, the Pearson χ^2 divergence was shown to improve the training stability for GANs (Mao et al., 2017) and for AIL (Peng et al., 2021). Kostrikov et al. (2019) introduces a replay buffer for off-policy updates for policy and discriminator. The authors also pointed out the problem of reward bias, which is common in many imitation learning methods. Indeed, AIL methods implicitly assign a null reward to these states, leading to survival or termination biases, depending on the chosen divergence. Kostrikov et al. (2020) improves the previous work introducing recent advances from offline policy evaluation (Nachum et al., 2019). Their method, ValueDice, uses an inverse Bellman operator that expresses the reward function in terms of its Q-function, to minimize the reverse Kullback-Leibler divergence to the expert distribution. Arenz & Neumann (2020) derive a non-adversarial formulation based on trust-region updates on the policy. Their method, O-NAIL, uses a standard soft actor-critic (Haarnoja et al., 2018) update for policy improvement. O-NAIL can be understood as an instance of the more general IQ-Learn algorithm (Garg et al., 2021), which can optimize different divergences depending on an implicit reward regularizer. Garg et al. (2021) also show that their algorithm achieves better performance using the χ^2 divergence instead of the reverse KL. Swamy et al. (2021) provide a unifying perspective on the aforementioned imitation learning approaches, framing them into moment-matching methods, that either match reward or Q-function moments. The authors also introduce ADRIL, which assigns fixed rewards to expert and policy transitions. Using fixed binary reward functions for imitation learning was originally proposed by Reddy et al. (2020) and can be derived from a regularized behavioral cloning objective.

2 PRELIMINARIES

Notation. A Markov Decision Process (MDP) is a tuple $(\mathcal{S}, \mathcal{A}, P, r, \gamma, \mu_0)$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $P : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}^+$ is the transition kernel, $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward function, γ is the discount factor, and $\mu_0 : \mathcal{S} \to \mathbb{R}^+$ is the initial state distribution. At each step, the agent observes a state $s \in \mathcal{S}$ from the environment, predicts an action $a \in \mathcal{A}$ using the policy $\pi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^+$, and transitions with probability $P(s'|s, a)$ into the next state $s' \in \mathcal{S}$, where it receives the reward $r(s, a)$. We define an occupancy measure $\rho_\pi(s, a) = \pi(a|s) \sum_{t=0}^{\infty} \gamma^t \mu_t^\pi(s)$, where $\mu_t^\pi(s') = \sum_{s,a} \mu_t^\pi(s)\pi(a|s)P(s'|s, a)$ is the state distribution for $t > 0$, with $\mu_0^\pi(s) = \mu_0(s)$. The occupancy measure allows us to denote the expected reward under policy π as $\mathbb{E}_{\rho_\pi}[r(s, a)] \triangleq \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)]$, where $s_0 \sim \mu_0$, $a_t \sim \pi(.|s_t)$ and $s_{t+1} \sim P(.|s_t, a_t)$ for $t > 0$. Furthermore, $\mathbb{R}^{\mathcal{S}\times\mathcal{A}} = \{x : \mathcal{S} \times \mathcal{A} \to \mathbb{R}\}$ denotes the set of functions in the state-action space and $\overline{\mathbb{R}}$ denotes the extended real numbers $\mathbb{R} \cup \{+\infty\}$. We refer to the soft value functions as $\tilde{V}(s)$ and $\tilde{Q}(s, a)$, while we use $V(s)$ and $Q(s, a)$ to denote the value functions without entropy bonus.

Inverse Reinforcement Learning as an Occupancy Matching Problem. Given a set of demonstrations consisting of states and actions sampled from an expert policy π_E, IRL aims at finding a reward function $r(s, a)$ from a family of reward functions $\mathcal{R} = \mathbb{R}^{\mathcal{S}\times\mathcal{A}}$ assigning high reward to samples from the expert policy π_E and low reward to other policies. We consider the framework presented in (Ho & Ermon, 2016), which derives the maximum entropy IRL objective from an occupancy matching problem with an additional convex reward regularizer $\psi_\rho : \mathbb{R}^{\mathcal{S}\times\mathcal{A}} \to \overline{\mathbb{R}}$:

$$\max_{r\in\mathcal{R}} \min_{\pi\in\Pi} L_\rho(r, \pi) = \max_{r\in\mathcal{R}} \left(\min_{\pi\in\Pi} -\beta H_\rho(\pi) - \mathbb{E}_{\rho_\pi}[r(s, a)] \right) + \mathbb{E}_{\rho_{\pi_E}}[r(s, a)] - \psi_\rho(r), \tag{1}$$

where $\Pi = \mathbb{R}^{\mathcal{S}\times\mathcal{A}}$ is the space of policies, $H_\rho(\pi)$ is the future cumulative entropy bonus such that $H_\rho(\pi) = \mathbb{E}_{\rho_\pi}[-\log(\pi(a|s))]$, and β is a constant controlling the entropy bonus. Note that the inner optimization is a maximum entropy Reinforcement Learning (RL) objective, for which the optimal policy is given by (Ziebart, 2010):

$$\pi^*(a|s) = \frac{1}{Z_s}\exp\left(\tilde{Q}(s,a)\right), \tag{2}$$

where $Z_s = \int_{\hat{a}} \exp \tilde{Q}(s,\hat{a})$ is the partition function and $\tilde{Q}(s,a)$ is the soft action-value function, which is given for a certain reward function by the soft Bellman operator $(\tilde{\mathcal{B}}^\pi\tilde{Q})(s,a) = r(s,a) + \gamma\mathbb{E}_{s'\sim P(.|s,a)}\tilde{V}^\pi(s')$, where $\tilde{V}^\pi(s') = \mathbb{E}_{a\sim\pi(.|s)}[\tilde{Q}(s,a) - \log\pi(a|s)]$.

Garg et al. (2021) transform Equation 1 from reward-policy space to $\tilde{Q}$-policy space using the *inverse* soft Bellman operator $(\tilde{\mathcal{T}}^\pi\tilde{Q})(s,a) = \tilde{Q}(s,a) - \gamma\mathbb{E}_{s'\sim P(.|s,a)}\tilde{V}^\pi(s')$ to get a one-to-one correspondence between the reward and the $\tilde{Q}$-function:

$$\max_{r\in\mathcal{R}}\min_{\pi\in\Pi} L_\rho(r,\pi) = \max_{\tilde{Q}\in\tilde{\Omega}}\min_{\pi\in\Pi} \mathcal{J}_\rho(\tilde{Q},\pi)\,, \tag{3}$$

where $\tilde{\Omega} = \mathbb{R}^{\mathcal{S}\times\mathcal{A}}$ is the space of $\tilde{Q}$-functions. Further, they use Equation 2 to extract the optimal policy $\pi_{\tilde{Q}}$ given a $\tilde{Q}$-function to drop the inner optimization loop in Equation 1 such that:

$$\begin{aligned}\max_{\tilde{Q}\in\tilde{\Omega}} \mathcal{J}_\rho(\tilde{Q},\pi_{\tilde{Q}}) = \max_{\tilde{Q}\in\tilde{\Omega}}\; & \mathbb{E}_{\rho_{\pi_E}}\left[\tilde{Q}(s,a) - \gamma\mathbb{E}_{s'\sim P(.|s,a)}[\tilde{V}^\pi(s')]\right] - \beta H_\rho(\pi_{\tilde{Q}}) \\ & - \mathbb{E}_{\rho_\pi}\left[\tilde{Q}(s,a) - \gamma\mathbb{E}_{s'\sim P(.|s,a)}[\tilde{V}^\pi(s')]\right] - \psi_\rho(r).\end{aligned} \tag{4}$$

Practical Reward Regularization. Garg et al. (2021) derive a regularizer enforcing an L2 norm-penalty on the reward on state-action pairs from the expert, such that $\psi_{\pi_E}(r) = c\,\mathbb{E}_{\rho_{\pi_E}}\left[r(s,a)^2\right]$ with c being a regularizer constant. However, in continuous action spaces, this regularizer causes instabilities. In practice, Garg et al. (2021) address this instabilities by using the regularizer to the mixture

$$\psi_\rho(r) = \alpha\, c\,\mathbb{E}_{\rho_{\pi_E}}\left[r(s,a)^2\right] + (1-\alpha)\, c\,\mathbb{E}_{\rho_\pi}\left[r(s,a)^2\right]\,, \tag{5}$$

where α is typically set to 0.5. It is important to note that this change of regularizer does not allow the direct extraction of the policy from Equation 1 anymore, as it introduces another term depending on the policy. Prior work did not address this issue. In the following sections, we will provide an in-depth analysis of this regularizer, allowing us to address the aforementioned issues and derive the correct policy update. Before we introduce our method, we use Proposition A.1 in Appendix A.1 to change the objectives L_ρ and $\mathcal{J}_\rho$ from expectations under occupancy measures to expectations under state-action distributions d_{π_E} and d_π, from now on denoted as L and $\mathcal{J}$ for brevity.

3 Least Squares Inverse Q-Learning

In this section, we introduce our proposed imitation learning algorithm, which is based on the occupancy matching problem presented in Equation 1 using the regularizer defined in Equation 5. We start by giving an interpretation of the resulting objective as a χ^2 divergence between the expert distribution and a mixture distribution of the expert and the policy. We then show that the regularizer allows us to cast the original objective into a Bellman error minimization problem. This novel objective allows us to stabilize training by bounding the Q-function targets, provide fixed targets for the Q-function on expert states, and properly treat absorbing states. Further, we introduce an entropy-regularization critic, which tackles the aforementioned problem of the regularizer and allows us to learn an implicit soft Q-function of which the policy can be extracted using Equation 1. Finally, we propose to use an IDM to solve the imitation learning from observations problem.

3.1 Interpretation as a Statistical Divergence

Ho & Ermon (2016) showed that their regularizer results in a Jensen-Shannon divergence minimization between the expert's and the policy's state-action distribution. Similarily, Garg et al. (2021) showed the that their regularizer $\psi_{\pi_E}(r)$ results in a minimzation of the χ^2 divergence. However, the regularizer presented in Equation 5 is not investigated yet. We show that this regularizer results

in minimizing a χ^2 divergence between the expert's state-action distribution and a mixture distribution between the expert and the policy. Therefore, we start with the objective presented in Equation 1 yet under state-action distributions instead of occupancy measures, and note that strong duality

$$\max_{r\in\mathcal{R}} \min_{\pi\in\Omega} L = \min_{\pi\in\Omega} \max_{r\in\mathcal{R}} L\,, \tag{6}$$

follows straightforwardly from the minimax theorem (Von Neumann, 1928) as $-H(\pi)$, $-\mathbb{E}_{d_\pi}[r(s,a)]$ and $\psi(r)$ are convex in π, and $-\mathbb{E}_{d_\pi}[r(s,a)]$, $\mathbb{E}_{d_{\pi_E}}[r(s,a)]$ and $\psi(r)$ are concave in r (Ho & Ermon, 2016). We express the χ^2 divergence between the expert's distribution and the mixture distribution using its variational form,

$$2\chi^2\big(d_{\pi_E} \,\|\, \underbrace{\tfrac{d_{\pi_E}+d_\pi}{2}}_{\tilde{d}}\big) = \sup_r 2\left(\mathbb{E}_{d_{\pi_E}}\left[r(s,a)\right] - \mathbb{E}_{\tilde{d}}\left[r(s,a) + \tfrac{r(s,a)^2}{4}\right]\right) \tag{7}$$

$$= \sup_r \mathbb{E}_{d_{\pi_E}}\left[r(s,a)\right] - \mathbb{E}_{d_\pi}\left[r(s,a)\right] - c\mathbb{E}_{d_{\pi_E}}\left[r(s,a)^2\right] - c\mathbb{E}_{d_\pi}\left[r(s,a)^2\right]\,, \tag{8}$$

where $c = \frac{1}{4}$ is the regularizer constant. Now, for $r^* \in \mathcal{R}$, the original objective from Equation 1 becomes an entropy-regularized χ^2 divergence minimization problem:

$$\max_{r\in\mathcal{R}} \min_{\pi\in\Omega} L = \min_{\pi\in\Pi} 2\chi^2\big(d_{\pi_E} \,\|\, \tfrac{d_{\pi_E}+d_\pi}{2}\big) - \beta H(\pi)\,. \tag{9}$$

Unlike the χ^2-divergence between the agent's and the expert's distribution, the χ^2-divergence to the mixture distribution is bounded. When the expert and the policy distribution do not share much support, optimizing an unbounded divergence can result in unbounded rewards that are difficult to estimate, significantly destabilizing the training. Note that the χ^2-divergence between the expert and the mixture is upper bounded by the χ^2-divergence between the expert and the policy, as shown in Proposition A.2.

3.2 A Reinforcement Learning Perspective on Distribution Matching

In the following, we present a novel RL perspective to understand the effect of the objective function defined in Equation 4. Indeed, for the regularizer defined in Equation 5, we can interpret this objective as an entropy-regularized least squares problem, as shown by the following proposition:

Proposition 3.1 *Let $\psi(r_{\tilde{Q}}) = c\,\mathbb{E}_{\tilde{d}}\left[r_{\tilde{Q}}(s,a)^2\right]$ with $\tilde{d}(s,a) = \alpha d_{\pi_E}(s,a) + (1-\alpha)d_\pi(s,a)$, then $\arg\max_{\tilde{Q}\in\tilde{\Omega}} \mathcal{J}(\tilde{Q},\pi_{\tilde{Q}})$ is the solution to the entropy-regularized least squares minimization problem $\arg\min_{\tilde{Q}\in\tilde{\Omega}} \mathcal{L}(\tilde{Q},\pi_{\tilde{Q}})$ with:*

$$\mathcal{L}(\tilde{Q},\pi_{\tilde{Q}}) = \alpha\mathbb{E}_{d_{\pi_E}}\left[\big(r_{\tilde{Q}}(s,a) - r_{\max}\big)^2\right] + (1-\alpha)\mathbb{E}_{d_{\pi_{\tilde{Q}}}}\left[\big(r_{\tilde{Q}}(s,a) - r_{\min}\big)^2\right] + \frac{\beta}{c}H(\pi_{\tilde{Q}})\,, \tag{10}$$

where $r_{\max} = \frac{1}{2\alpha c}$ and $r_{\min} = -\frac{1}{2(1-\alpha)c}$.

The proof is provided in Appendix A. The resulting objective in Equation 10 is very similar to the one in the Least Squares Generative Adversarial Networks (LSGANs) (Mao et al., 2017) setting, where $r_{\tilde{Q}}(s,a)$ can be interpreted as the discriminator, $r_{\max}$ can be interpreted as the target for expert samples, and $r_{\min}$ can be interpreted as the target for samples under the policy π. For $\alpha = 0.5$ and $c = 1$, resulting in $r_{\max} = 1$ and $r_{\min} = -1$, Equation 10 differs from the discriminator's objective in the LSGANs setting only by the entropy term.

From Equation 10 it is straightforward to give an RL perspective on the distribution matching problem under the regularizer $\psi(r)$. Therefore, after applying the inverse soft Bellman operator to the implicit reward $r_{\tilde{Q}}(s,a)$ and rearranging the terms,

$$\begin{aligned}\mathcal{L}(\tilde{Q},\pi_{\tilde{Q}}) =& \alpha\mathbb{E}_{d_{\pi_E}}\left[\Big(\tilde{Q}(s,a) - (r_{\max} + \gamma\mathbb{E}_{s'\sim P(.|s,a)}[\tilde{V}^\pi(s')]\Big)^2\right] \qquad (11)\\ &+ (1-\alpha)\mathbb{E}_{d_{\pi_{\tilde{Q}}}}\left[\Big(\tilde{Q}(s,a) - (r_{\min} + \gamma\mathbb{E}_{s'\sim P(.|s,a)}[\tilde{V}^\pi(s')])\Big)^2\right] + \frac{\beta}{c}H(\pi_{\tilde{Q}})\\ =& \alpha\,\delta^2(d_{\pi_E}, r_{\max}) + (1-\alpha)\,\delta^2(d_\pi, r_{\min}) + \frac{\beta}{c}H(\pi_{\tilde{Q}})\,, \qquad (12)\end{aligned}$$

with the squared soft Bellman error δ^2. This result implies that Equation 4 solves an RL problem, with fixed rewards $r_{\max}$ and $r_{\min}$ for the expert and the policy. The resulting algorithm shows strong relations to the SQIL algorithm, in which also fixed rewards are used for the expert and the policy. However, SQIL uses $r_{\max} = 1$ and $r_{\min} = 0$ as fixed reward targets, which is not feasible in our setting for $\alpha < 1$. While the entropy term appears to be another difference, it is important to note that it does not affect the critic update, where $\pi_{\tilde{Q}}$ is fixed. The entropy is maximized by extracting the maximum entropy policy using Equation 2, which is also done in SQIL. This reinforcement learning perspective allows us to properly treat absorbing states, as shown in the following sections.

3.3 Treatment of Absorbing States

IQ-Learn treats absorbing states by adding an indicator ν in front of the discounted value function in the soft inverse Bellman:

$$(\tilde{\mathcal{T}}^{\pi}_{iq}\tilde{Q})(s,a) = \tilde{Q}(s,a) - (1-\nu)\gamma\mathbb{E}_{s'\sim P(.|s,a)}\tilde{V}^{\pi}(s')\,, \tag{13}$$

where $\nu = 1$ if s' is a terminal state. Hence, applying the soft inverse Bellman operator to Equation 10 will regress the Q-function of transitions into absorbing states towards $r_{\max}$ or $r_{\min}$, respectively. However, based on Equation 10, the implicit *reward* of absorbing states should be regressed toward $r_{\max}$ or $r_{\min}$. The corresponding value of absorbing states will instead be given by $\tilde{V}^{\pi} = \frac{r_A}{1-\gamma}$, which results in the following soft inverse Bellman operator:

$$(\tilde{\mathcal{T}}^{\pi}_{lsiq}\tilde{Q})(s,a) = \tilde{Q}(s,a) - (1-\nu)\gamma\mathbb{E}_{s'\sim P(.|s,a)}\tilde{V}^{\pi}(s') - \nu\gamma\frac{r_A}{1-\gamma}\,, \tag{14}$$

where the absorbing state reward r_A corresponds to either $r_{\max}$ or $r_{\min}$, depending on whether the expert or the agent reached the absorbing state. Applying the soft inverse Bellman operator $\tilde{\mathcal{T}}_{lsiq}$ to Equation 10 will correctly result in the Q-function targets $\frac{r_A}{1-\gamma}$ for transitions into absorbing states.

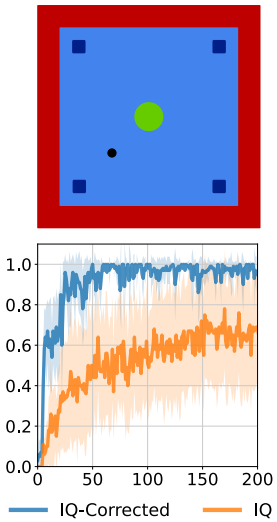

Figure 1: Point mass toy task (top) with success rate plot (bottom). Here, we compare standard IQ-Learn using the modified operator

Consider the toy task depicted in Figure 1 (top), where the black point mass is spawned in either of the four dark blue squares and has to reach the green area in the middle. Once the agent enters the red area, the episode terminates. The expert always takes the shortest path to the green area, never visiting the red area. The operator proposed by IQ-Learn does not sufficiently penalize the agent for reaching absorbing states, preventing the IQ-Learn agent from reaching the goal consistently, as can be seen from the orange graph in Figure 1 (bottom). In contrast, when using our operator $\tilde{\mathcal{T}}_{lsiq}$, the agent solves the task successfully.

3.4 Entropy and Regularization Critic

An RL problem with fixed rewards is a unique setting, which allows us to incorporate more knowledge into the optimization of the hard Q-function. In doing so, we can directly stabilize the training of the latter. However, in the MaxEnt setup, we use $\tilde{Q}$-function—i.e., *soft* action-value functions—which also includes an entropy bonus based on the current policy. To apply these stabilization techniques on the hard Q-function, we express the $\tilde{Q}$-function in terms of a hard Q-function (the critic) and an *entropy critic* $\mathcal{H}^{\pi}(s,a)$, that is, $\tilde{Q}(s,a) = Q(s,a) + \mathcal{H}^{\pi}(s,a)$, where

$$\mathcal{H}^{\pi}(s,a) = \mathbb{E}_{P,\pi}\left[\sum_{t'=t}^{\infty} -\gamma^{t'-t+1}\beta\log\pi(a_{t'+1}|s_{t'+1})\middle| s_t = s, a_t = a\right]. \tag{15}$$

As mentioned before, the regularizer introduced in Equation 5 incorporates yet another term depending on the policy. That is, the inner optimization problem in Equation 1—the term in the brackets—is not purely the MaxEnt problem anymore, but includes the term $-\mathbb{E}_{d_\pi}[r(s,a)^2]$. However, $\tilde{Q}$ does not incorporate this term, which is why the solution of Equation 2 is not the true solution of our objective in Equation 1 when using the regularizer from Equation 5. We need to incorporate the term added by the regularizer into our action-value function. To do so, we learn an additional *regularization critic* $\mathcal{C}(s,a)$, such that $Q^{\dagger}(s,a) = Q(s,a) + \mathcal{H}^{\pi}(s,a) + c(1-\alpha)\mathcal{C}(s,a)$, where

$$\mathcal{C}(s,a) = \mathbb{E}_{P,\pi}\left[\sum_{t'=t}^{\infty}\gamma^{t'-t}r(s_{t'},a_{t'})^2\middle| s_t = s, a_t = a\right]. \tag{16}$$

When now using $Q^\dagger$ in Equation 2 obtain the true solution to the inner minimization problem. In practice, we do not learn two additional critics but one, in which we combine the entropy and regularization critic to a single critic $\mathcal{G}^\pi$. We train the latter independently by minimizing the following Bellman error:

$$\min_{\mathcal{G}^\pi} \delta_{\mathcal{G}}^2 = \min_{\mathcal{G}^\pi} \mathbb{E}_{d_\pi}\left[(\mathcal{G}^\pi(s,a) - (r_Q(s,a)^2 + \mathbb{E}_{\substack{s'\sim P\\ a'\sim\pi}}\left[\gamma(-\beta\log\pi(a'|s') + \mathcal{G}^\pi(s',a')\right]))^2\right]. \tag{17}$$

The introduction of the entropy and regularization critics results in two benefits. First, the entropy critic allows us to switch from optimization in $\tilde{Q}$-space to Q-space. That is, we do not explicitly train the $\tilde{Q}$-function. Instead, it is constructed implicitly using Equation 15. This procedure allows us to stay in the MaxEnt formulation while retaining the ability to operate on the hard Q-function. We replace the soft inverse Bellman operator $(\tilde{\mathcal{T}}\tilde{Q})(s,a)$ with the hard optimal inverse Bellman operator $(\mathcal{T}Q)(s,a) = Q(s,a) - \gamma\mathbb{E}_{s'\sim P(.|s,a)}V^*(s')$, where $V^*(s) = \max_a Q(s,a)$. This change of operator replaces the soft Bellman error in Equation 10 with the hard Bellman error. Note that we also incorporate the adaptions made to operator in the previous section, but skip it from now on in the notation for brevity. Second, the regularization critic can incorporate the regularizer into the optimization of the policy. By merging the penalty in this critic, we can minimize the exact χ^2 divergence between the expert and the mixture distribution as shown in Equation 9.

3.5 An Alternative Formulation for the Expert Residual Minimization

The first term in Equation 12 defines the squared Bellman error minimization problem on the distribution d_{π_E}:

$$\alpha\mathbb{E}_{d_{\pi_E}}\left[(r_Q(s,a) - r_{\max})^2\right]. \tag{18}$$

Due to bootstrapping, this minimization can become challenging, even for a fixed expert policy, as it does not fix the scale of the Q-function unless the trajectory reaches an absorbing state. This problem arises particularly on expert data for cyclic tasks, where we generate trajectories up to a fixed horizon. The lack of a fixed scale increases the variance of the algorithm, affecting the performance negatively. Furthermore, this optimization does not consider properly the effective imbalance of expert data, often following a deterministic behavior and available in limited amount, and policy data, which is more diverse and abundantly available.

Therefore, we propose a modified objective, analyzing the minima of Equation 18. The minimum of this term is achieved when $r_Q(s,a) = r_{\max}\forall(s,a) \sim d_{\pi_E}$. Thus, the objective of this term is to push the reward, on expert trajectories, towards $r_{\max}$. If we consider this minimum, each step of in the trajectory of the expert has the following Q-value:

$$Q^{\pi_E}(s,a) = \sum_{t=0}^{\infty}\gamma^t r_{\max} = \frac{r_{\max}}{1-\gamma} = Q_{\max}, \quad \forall(s,a)\sim d_{\pi_E}. \tag{19}$$

As the objective of our maximization on expert distribution is equivalent to pushing the value of the expert's states and actions towards $Q_{\max}$, we propose to replace the bootstrapping target with the fixed target $Q_{\max}$ resulting in the following new objective:

$$\begin{aligned}\mathcal{L}_{\text{LSIQ}}(Q) =& \alpha\mathbb{E}_{d_{\pi_E}}\left[(Q(s,a) - Q_{\max})^2\right] \\ &+ (1-\alpha)\mathbb{E}_{d_\pi}\left[\left(Q(s,a) - (r_{\min}(s) + \gamma\mathbb{E}_{s'\sim P(.|s,a)}[V^*(s')])\right)^2\right].\end{aligned} \tag{20}$$

Note that the entropy term is now incorporated into $\mathcal{H}^\pi(s,a)$ and optimized using Equation 2 by using the implicit soft $\tilde{Q}$-function defined in Equation 15. This new objective incorporates a bias towards expert data. Therefore it is not strictly equivalent to the original problem formulation. However, it updates the Q-function towards the same ideal target, while providing a much simpler and stable optimization landscape. Empirically, we experienced that this modification, while only justified intuitively, has a very positive impact on the performance of the algorithm. Finally, the bias towards expert optimization is helpful to deal with data imbalance.

3.6 LEARNING FROM OBSERVATIONS

In many real-world tasks we do not have access to expert actions, but only on observations of expert behavior (Torabi et al., 2019b). In this scenario AIL methods, such as GAIfO (Torabi et al., 2019a), can be easily adapted by learning a discriminator only depending on current and next state. Unfortunately, it is not straightforward to apply the same method to implicit rewards algorithms that learn a Q-function. The IQ-Learn method (Garg et al., 2021) relies on a simplification of the original objective to perform updates not using expert actions but rather actions sampled from the policy on expert states. However, this reformulation is not able to achieve good performance on standard benchmarks as shown in our experimental results.

A common practice used in the literature is to train an IDM. This approach has been previously used in behavioral cloning (Torabi et al., 2018; Nair et al., 2017) and for reinforcement learning from demonstrations (Guo et al., 2019; Pavse et al., 2020; Radosavovic et al., 2021). Following the same idea, we generate an observation-only version of our method by training an IDM online and using it for predicting unobserved actions. The approach is described visually in Fig. 2.

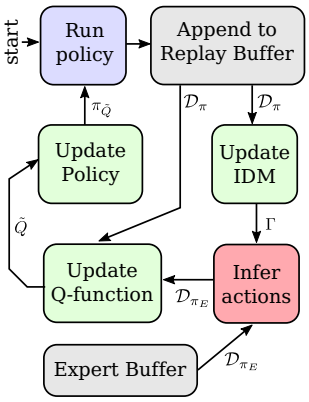

Figure 2: Scheme of the Observation-only approach, including IDM training.

We modify the objective in Equation 20 as follows

$$\mathcal{L}_{\text{LS}}(Q) = \alpha \mathbb{E}_{d_{\pi_E}} \left[\left(Q(s, \Gamma(s, s')) - Q_{\max} \right)^2 \right] + (1-\alpha) \mathbb{E}_{d_\pi} \left[\left(Q(s,a) - (r_{\min}(s) + \gamma \mathbb{E}_{s' \sim P(.|s,a)} [V^*(s')]) \right)^2 \right], \tag{21}$$

with the dynamics model $\Gamma(s, s')$. Notice that the IDM is only used to evaluate the expert actions, while it is trained by minimizing the following loss:

$$\mathbb{E}_{d^\pi, \mathcal{P}} \left[\| \Gamma(s, s') - a \|_2^2 \right], \tag{22}$$

where the expectation is performed on the state distribution generated by the learner policy π. While the mismatch between the training distribution and the evaluation distribution could potentially cause problems, our empirical evaluation shows that on the benchmarks we achieve performance similar to the action-aware algorithm.

3.7 PRACTICAL ALGORITHM

We now instantiate a practical version of our algorithm in this section. An overview over our method is shown in Algorithm 1.

Algorithm 1 LS-IQ

Initialize: Q_θ, π_ϕ, $\mathcal{G}_\nu$ and optionally Γ_ω
for step t in $\{1, \ldots, N\}$ **do**
  Sample mini-batches $\mathcal{D}_\pi$ and $\mathcal{D}_{\pi_E}$
  (opt.) Predict actions for $\mathcal{D}_{\pi_E}$ using Γ_ω
  $\{a_1^{\pi_E}, \ldots, a_{|\mathcal{D}_{\pi_E}|}^{\pi_E}\} \leftarrow \Gamma_\omega(\mathcal{D}_{\pi_E})$
  $\mathcal{D}_{\pi_E} \leftarrow \mathcal{D}_{\pi_E} \cup \{a_1^{\pi_E}, \ldots, a_{|\mathcal{D}_{\pi_E}|}^{\pi_E}\}$
  Update the Q-function using Eq. 21
  $\theta_{t+1} \leftarrow \theta_t + \kappa_Q \nabla_\theta [\mathcal{J}(\theta, \mathcal{D}_\pi, \mathcal{D}_{\pi_\mathcal{E}})]$
  (opt) Update $\mathcal{G}$-function using Eq. 17
  $\nu_{t+1} \leftarrow \nu_t - \kappa_\mathcal{G} \nabla_\nu [\mathcal{L}_\mathcal{G}(\nu, \mathcal{D}_\pi)]$
  Update Policy π_ϕ using the KL
  $\phi_{t+1} \leftarrow \phi_t - \kappa_\pi \nabla_\phi [D_{KL}(\pi\phi \parallel \pi_{\tilde{Q}})]$
  (opt.) Update Γ_ω using Eq. 22
  $\omega_{t+1} \leftarrow \omega_t - \kappa_\Gamma \nabla_\omega [\mathcal{L}_\Gamma(\omega, \mathcal{D}_\pi)]$
end for

In practice, we use target networks, which was already suggested by the Garg et al. (2021). However, while the objective in Equation 4 lacked intuition about the usage of target networks—without the regularizer the application of target networks would result in a maximization/minimzation of the Q-function—the objective in Equation 12 is equivalent to a reinforcement learning objective, in which target networks are a well-known tool for stabilization. Additionally to that, we exploit our access to the hard Q-function as well as our fixed reward target setting to calculate the maximum and minimum Q-value possible – $Q_{\min} = \frac{r_{\min}}{1-\gamma}$ and $Q_{\max} = \frac{r_{\max}}{1-\gamma}$ – and clip the output of target network to that range. Note that this also holds for the absorbing states. In doing so, we ensure that the Q-function is always forced to stay in the desired range, which we often experienced not the case in IQ-Learn. Target clipping prevents the explosion of the Q-values that can occur due to the use of neural approximators. This technique allows the algorithm to recover from poor value function estimates and prevents the Q-function from leaving the set of admissible functions for the problem.

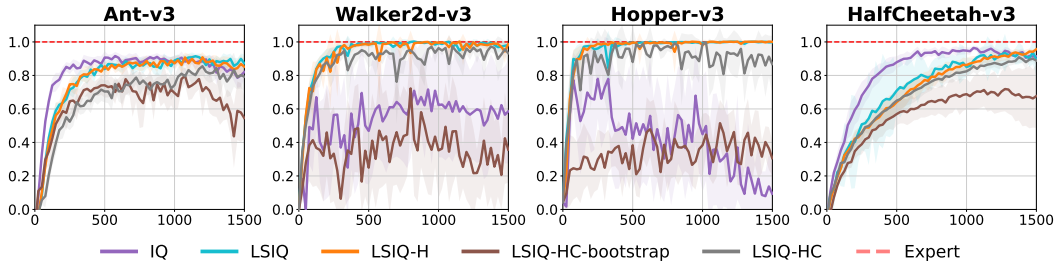

Figure 3: Comparison of different versions of LS-IQ. Abscissa shows the normalized discounted cumulative reward. Ordinate shows the number of training steps ($\times 10^3$). Expert cumulative rewards $\rightarrow$ Hopper:3299.81, Walker2d:5841.73, Ant:6399.04, HalfCheetah:12328.78, Humanoid:6233.45

In continuous action spaces, Z_s is intractable, which is why we can not directly extract the optimal policy using Equation 2. As done in previous work (Haarnoja et al., 2018; Garg et al., 2021), we use a parametric policy π_θ to approximate $\pi_{\tilde{Q}}$ by minimizing the Kullback-Leibler Divergence (KL) $D_{\text{KL}}(\pi_\theta \parallel \pi_{\tilde{Q}})$. In our implementation, we found it unnecessary to use a double-critic update. This choice reduces the computational and memory requirements of the algorithm, making it comparable to Soft-Actor Critic (SAC). Finally, we replace $V^*(s)$ with $V^\pi(s)$ on the policy expectation, as we do not have access to the latter in continuous action spaces.

4 Experiments

We evaluate our method on five MuJoCo environments: Ant-v3, Walker2d-v3, Hopper-v3, HalfCheetah-v3 and Humanoid-v3. We select the following baselines: GAIL (Ho & Ermon, 2016), VAIL (Peng et al., 2019), IQ-Learn (Garg et al., 2021) and SQIL (Reddy et al., 2020). For a fair comparison we reimplemented all methods in the same framework, MushroomRL (D'Eramo et al., 2021), such that the same libraries and RL agents were used. We verified that our implementations achieve comparable results to the original implementations by the authors, where we used their hyperparameters, which were already tuned for the same environments that we use. The original implementation of IQ-Learn tested two different variants of the algorithm, depending on the given environment. We refer to these variants as IQv0—which uses telescoping (Garg et al., 2021) to evaluate the agent's expected return in Equation 4—, and IQ—which directly uses Equation 4—and evaluate both variants on all environments. For our method, we used the same hyperparameters as IQ-Learn, except for the regularizer coefficient c and the entropy coefficient β which we tuned on each environment. For the mixing parameter, we only considered equal mixing, i.e., $\alpha = 0.5$.

In our first experiment, we performed ablations on the different design choices of LS-IQ. We evaluated the following variants: LSIQ-HC uses a (combined) entropy critic and regularization critic, LSIQ-H only uses the entropy critic and LSIQ does not use any additional critic, similar to IQ-Learn. We also evaluate LSIQ-HC without using the fixed Q-function targets on the expert distribution that were described in Section 3.5, and refer to it as LSIQ-HC-bootsrap. We used ten seeds and five expert trajectories for these experiments. We also added IQ as baseline and show the learning curves for four of the environments in Figure 3. The learning curves on the Humanoid environment can be found in Appendix B. Based on these experiments, the additional critics have little effect, whereas fixed Q-function targets significantly increase the performance. It is also interesting to note that IQ-Learn without telescoping, does not perform well on Walker and Hopper, where absorbing states are more likely compared to Ant and HalfCheetah (which always terminates after a fixed number of steps). Our hypothesis is that the worse performance on Walker and Hopper is caused by reward bias, as absorbing states are not sufficiently penalized. IQv0 would suffer less from this problem, as it treats all states visited by the agent as initial states which results in stronger reward penalties for these states.

For our main experiments we only evaluated LSIQ and LSIQ-H, which achieved the strongest performance on all environments. We compare to all baselines for four different numbers of expert demonstrations, namely 1, 5, 10 and 25, and always use five seeds. We performed each experiment with and without action observations. When actions are not available, we used a state transition discriminator (Torabi et al., 2019a) for GAIL and VAIL, and IDMs (see Section 3.6) for IQ-Learn and

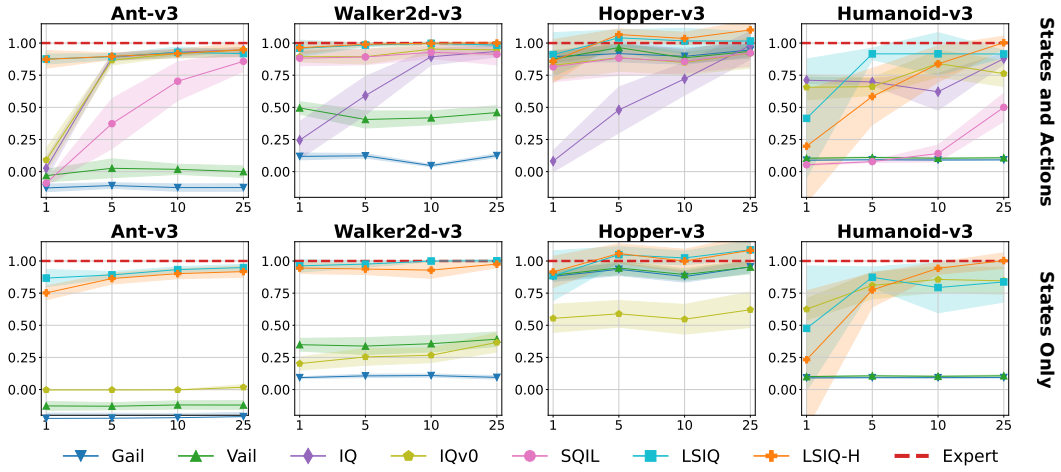

Figure 4: Ablation study on the effect of the number of expert trajectories on different Mujoco environments. Abscissa shows the normalized cumulative reward. Ordinate shows the number of expert trajectories. The first row shows the performance when considering states and action, while the second row considers the performance when using states only. Expert cumulative rewards identical to Figure 3.

LS-IQ. In the learning from observation setting, we did not evaluate SQIL, and we omit the plots for IQ, which did not converge in any environment. Figure 4 shows the final expected return over different numbers of demonstrations for four of the environments. All learning curves, including the HalfCheetah environment, can be found in Appendix B.2 for state-action observations, and in Appendix B.1 for the learning from observation setting. Our experiments show that LS-IQ achieves on par or better performance compared to all baselines, with the notable exception of the Humanoid experiment when only a single expert demonstration is provided. In particular in the learning from observation setting, LS-IQ performs very well by achieving similar return compared to the setting where states and actions are observed.

5 Conclusion

Inspired by the practical implementation of IQ-Learn, we derive an implicit reward distribution matching algorithm using a squared L2 penalty on the mixture distribution of expert and policy. We show that this regularizer minimizes the χ^2-divergence to the mixture distribution, and results in modified updates for Q-function and policy. Our analysis reveals an interesting connection to SQIL—which was not derived from an adversarial distribution matching objective—and shows that IQ-Learn suffers from reward bias. We build on our insights to propose a novel method, LS-IQ, which uses a modified inverse Bellman operator to address reward bias, target clipping, fixed reward targets for policy samples and fixed Q-function targets for expert samples. We also show that the policy optimization of IQ-Learn is not consistent with regularization on the mixture distribution, and show how this can be addressed by learning an additional critic. In our experiments, LS-IQ outperforms strong baseline methods, in particular when learning from observations, where we train an IDM for predicting expert actions. In future work, we will quantify the bias introduced by the fixed Q-function target and why this heuristic is fundamental for stabilizing learning. We will also analyze the error propagation in the Q-function target and derive theoretical guarantees on the Q-function approximation error.

References

Oleg Arenz and Gerhard Neumann. Non-adversarial imitation learning and its connections to adversarial methods. *arXiv*, August 2020.

Carlo D'Eramo, Davide Tateo, Andrea Bonarini, Marcello Restelli, and Jan Peters. Mushroomrl: Simplifying reinforcement learning research. *Journal of Machine Learning Research*, 22:1–5, 2021.

Justin Fu, Katie Luo, and Sergey Levine. Learning robust rewards with adversarial inverse reinforcement learning. In *Proceeding of the International Conference on Learning Representations*, Vancouver, Canada, April 2018.

Divyansh Garg, Shuvam Chakraborty, Chris Cundy, Jiaming Song, and Stefano Ermon. Iq-learn: Inverse soft-q learning for imitation. In *Proceeding of the Thirty-fifth Conference on Neural Information Processing Systems*, Sydney, Australia, December 2021.

Seyed Kamyar Seyed Ghasemipour, Richard Zemel, and Shixiang Gu. A divergence minimization perspective on imitation learning methods. In *Proceeding of the Conference on Robot Learning*, Osaka, Japan, November 2019.

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. In *Proceeding of the Twenty-eighth Conference on Neural Information Processing Systems*, Montreal, Canada, December 2014.

Xiaoxiao Guo, Shiyu Chang, Mo Yu, Gerald Tesauro, and Murray Campbell. Hybrid reinforcement learning with expert state sequences. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 3739–3746, 2019.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *Proceeding of the International Conference on Machine Learning*, Stockholm, Sweden, July 2018.

Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. In NIPS 2016.

ICLR 2019. *Proceeding of the International Conference on Learning Representations*, New Orleans, United States, May 2019.

ICLR 2020. *Proceeding of the International Conference on Learning Representations*, Virtual, May 2020.

Ilya Kostrikov, Kumar Krishna Agrawal, Debidatta Dwibedi, Sergey Levine, and Jonathan Tompson. Discriminator-actor-critic: Addressing sample inefficiency and reward bias in adversarial imitation learning. In ICLR 2019.

Ilya Kostrikov, Ofir Nachum, and Jonathan Tompson. Imitation learning via off-policy distrbution matching. In ICLR 2020.

Xudong Mao, Qing Li Haoran Xie, Raymond Y.K. Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. In *Proceeding of the IEEE International Conference on ComputerVision (ICCV)*, Venice, Italy, October 2017.

Ofir Nachum, Yinlam Chow, Bo Dai, and Lihong Li. Dualdice: Behavior-agnostic estimation of discounted stationary distribution corrections. In *Proceeding of the Thirty-third Conference on Neural Information Processing Systems*, Vancouver, Canada, December 2019.

Ashvin Nair, Dian Chen, Pulkit Agrawal, Phillip Isola, Pieter Abbeel, Jitendra Malik, and Sergey Levine. Combining self-supervised learning and imitation for vision-based rope manipulation. In *2017 IEEE international conference on robotics and automation (ICRA)*, pp. 2146–2153. IEEE, 2017.

Andrew Ng, Daishi Harada, and Stuart Russell. Policy invariance under reward transformations: Theory and application to reward shaping. In *Proceeding of the International Conference on Machine Learning*, Bled, Slovenia, June 1999.

NIPS 2016. *Proceeding of the Thirtieth Conference on Neural Information Processing Systems*, Barcelona, Spain, December 2016.

Sebastian Nowozin, Botonf Cseke, and Ryota Tomioka. f-gan: Training generative neural samplers using variational divergence minimization. In NIPS 2016.

Brahma S Pavse, Faraz Torabi, Josiah Hanna, Garrett Warnell, and Peter Stone. Ridm: Reinforced inverse dynamics modeling for learning from a single observed demonstration. *IEEE Robotics and Automation Letters*, 5(4):6262–6269, 2020.

Xue Bin Peng, Angjoo Kanazawa, Sam Toyer, Pieter Abbeel, and Sergey Levine. Variational discriminator bottleneck: Improving imitation learning, inverse rl, and gans by constraining information flow. In ICLR 2019.

Xue Bin Peng, Ze Ma, Pieter Abbeel, Sergey Levine, and Angjoo Kanazawa. Amp: Adversarial motion priors for stylized physics-based character control. *ACM Transactions on Graphics*, 40:1 – 20, 2021.

Ilija Radosavovic, Xiaolong Wang, Lerrel Pinto, and Jitendra Malik. State-only imitation learning for dexterous manipulation. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 7865–7871. IEEE, 2021.

Siddharth Reddy, Anca D. Dragan, and Sergey Levine. Sqil: Imitation learning via reinforcement learning with sparse rewards. In ICLR 2020.

Gokul Swamy, Sanjiban Choudhury, J. Andrew Bagnell, and Zhiwei Steven Wu. Of moments and matching: A game-theoretic framework for closing the imitation gap. In *Proceeding of the International Conference on Machine Learning*, Virtual, July 2021.

Faraz Torabi, Garrett Warnell, and Peter Stone. Behavioral cloning from observation. In *Proceeding of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 4950–4957, Stockholm, Sweden, July 2018.

Faraz Torabi, Garrett Warnell, and Peter Stone. Generative adversarial imitation from observation. In *Proceeding of the International Conference on Machine Learning*, Long Beach, California, July 2019a.

Faraz Torabi, Garrett Warnell, and Peter Stone. Recent advances in imitation learning from observation. In *Proceeding of the Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI)*, Macao, China, August 2019b.

John Von Neumann. Zur theorie der gesellschaftsspiele. *Mathematische annalen*, 100(1):295–320, 1928. Translated to English by Sonya Bargman (Von Neumann, 1959).

John Von Neumann. On the theory of games of strategy. *Contributions to the Theory of Games*, 4: 13–42, 1959.

Brian D. Ziebart. *Modeling Purposeful Adaptive Behavior with the Principle of Maximum Causal Entropy*. PhD thesis, University of Washington, 2010.

Brian D. Ziebart, Andrew Maas, J.Andrew Bagnell, and Anind K. Dey. Maximum entropy inverse reinforcement learning. In *Proceeding of the Twenty-Third AAAI Conference on Artificial Intelligence*, Chicago, United States, July 2008.

A PROOFS AND DERIVATIONS

In this section, we present the proofs of this work.

A.1 FROM OCCUPANCY MEASURES TO DISTRIBUTIONS

Based on Proposition A.1, the solution $\arg\max_{Q\in\tilde{\Omega}} \mathcal{J}_\rho(\tilde{Q}, \pi_{\tilde{Q}})$ under occupancy measures equals the solution $\arg\max_{\tilde{Q}\in\tilde{\Omega}} \mathcal{J}(\tilde{Q}, \pi_{\tilde{Q}})$ under state-action distributions. This result allows us to use the following distribution matching problem from now on:

$$\max_{\tilde{Q}\in\tilde{\Omega}} \mathcal{J}(\tilde{Q}, \pi_{\tilde{Q}}) = \max_{\tilde{Q}\in\tilde{\Omega}} \; \mathbb{E}_{d_{\pi_E}}\left[r_{\tilde{Q}}(s,a)\right] - \mathbb{E}_{d_{\pi_{\tilde{Q}}}}\left[r_{\tilde{Q}}(s,a)\right] - \psi(r_{\tilde{Q}}) - \beta H(\pi_{\tilde{Q}})\,, \tag{23}$$

where we introduce the implicit reward $r_{\tilde{Q}}(s,a) = \tilde{Q}(s,a) - \gamma\mathbb{E}_{s'\sim P(.|s,a)}[\tilde{V}^\pi(s')]$ for comprehension, β is a constant controlling the entropy regularization, d_{π_E} is the state-action distribution of the expert, and d_π is the state-action distribution under the policy.

Proposition A.1 *Let $\max_{r\in\mathcal{R}} \min_{\pi\in\Pi} L_\rho(\pi, r)$ be the dual problem of a regularized occupancy matching optimization problem and $L(\pi, r)$ be the Lagrangian of the regularized distribution matching problem. Then it holds that $L_\rho(\pi, r) \propto L(\pi, r)$ and $\mathcal{J}_\rho(\tilde{Q}, \pi_{\tilde{Q}}) \propto \mathcal{J}(\tilde{Q}, \pi_{\tilde{Q}})$.*

Proof of Proposition A.1.

Starting from the definition of the occupancy measure of an arbitrary policy π, we compute the normalizing constant as an integral:

$$\begin{aligned}
\int_{\mathcal{S}\times\mathcal{A}} \rho^\pi(s,a)dsda &= \int_{\mathcal{S}\times\mathcal{A}} \lim_{T\to\infty} \sum_{t=0}^{T-1} \gamma^t \mu_t^\pi(s,a)dsda \\
&= \lim_{T\to\infty} \sum_{t=0}^{T-1} \gamma^t \int_{\mathcal{S}\times\mathcal{A}} \mu_t^\pi(s,a)dsda \\
&= \lim_{T\to\infty} \sum_{t=0}^{T-1} \gamma^t \cdot 1 \\
&= \frac{1}{1-\gamma}
\end{aligned} \tag{24}$$

Now we compute the state-action distribution as:

$$\mu^\pi(s,a) = \frac{\rho^\pi(s,a)}{\int_{\mathcal{S}\times\mathcal{A}} \rho(s,a)dsda} = \frac{\rho^\pi(s,a)}{\frac{1}{1-\gamma}} = (1-\gamma)\rho^\pi(s,a) \tag{25}$$

Thus, we have:

$$\rho^\pi(s,a) = \frac{1}{1-\gamma}\mu^\pi(s,a) \tag{26}$$

Using equation 26 into the definition of the objective we obtain:

$$\mathcal{J}(\pi) = \frac{1}{1-\gamma}\int_{\mathcal{S}\times\mathcal{A}} \mu^\pi(s,a)r(s,a)dsda = \frac{1}{1-\gamma}\mathbb{E}_{\mu^\pi}\left[r(s,a)\right] \tag{27}$$

A derivation similar to equation 27 can be done for the entropy and the regularizer using equation 26. Substituting the derived formulas into equation 1 and collecting the constant $\frac{1}{1-\gamma}$ proves the proposition. ■

Proposition A.2 *Let $\chi^2\big(d_{\pi_E} \,\big\|\, \alpha d_{\pi_E} + (1-\alpha)d_\pi\big)$ be the Pearson χ^2-divergence between the distribution d_{π_E} and the mixture distribution $\alpha d_{\pi_E} + (1-\alpha)d_\pi$, and let $\alpha = 0.5$. Then it holds that:*

$$\chi^2\big(d_{\pi_E} \,\big\|\, \alpha d_{\pi_E} + (1-\alpha)d_\pi\big) \leq (1-\alpha)\chi^2\big(d_{\pi_E} \,\big\|\, d_\pi\big)\,.$$

Proof of Proposition A.2. The proof follows straightforwardly from the joint convexity of f-divergences:

$$D_f\big(\kappa P_1 + (1-\kappa)P_2 \,\big\|\, \kappa Q_1 + (1-\kappa)Q_2\big) \leq \kappa D_f(P_1 \,\|\, Q_1) + (1-\kappa) D_f(P_2 \,\|\, Q_2)\,,$$

where $\kappa \in [0,1]$ is the a mixing coefficient. The χ^2-divergence is an f-divergence with $f(t) = (t-1)^2$. Now using the χ^2-divergence, and let $P = P_1 = P_2 = Q_2$, $D = Q_1$, and $\alpha = (1-\kappa)$:

$$\begin{aligned}
\chi^2\big(P \,\big\|\, \alpha P + (1-\alpha)D\big) \leq& (1-\alpha)\,\chi^2(P \,\|\, D) + \underbrace{\alpha\chi^2(P \,\|\, P)}_{=0} \\
\chi^2\big(P \,\big\|\, \alpha P + (1-\alpha)D\big) \leq& (1-\alpha)\,\chi^2\big(P \,\big\|\, D\big)\,.
\end{aligned}$$

Setting $P = d_{\pi_E}$ and $D = d_\pi$ concludes the proof. ■

Proof of Proposition 3.1. Staring from the objective function in Equation 23, by expanding the expectation and rearranging the terms, we obtain:

$$\begin{aligned}
\mathcal{J}(\tilde{Q}, \pi_{\tilde{Q}}) =& \mathbb{E}_{d_{\pi_E}}\left[r_{\tilde{Q}}(s,a)\right] - \mathbb{E}_{d_\pi}\left[r_{\tilde{Q}}(s,a)\right] - c\,\mathbb{E}_{\tilde{d}}\left[r_{\tilde{Q}}(s,a)^2\right] - \beta H(\pi_{\tilde{Q}}) \\
=& \mathbb{E}_{d_{\pi_E}}\left[r_{\tilde{Q}}(s,a)\right] - \mathbb{E}_{d_\pi}\left[r_{\tilde{Q}}(s,a)\right] \\
&- c\alpha\,\mathbb{E}_{d_{\pi_E}}\left[r_{\tilde{Q}}(s,a)^2\right] - c(1-\alpha)\,\mathbb{E}_{d_\pi}\left[r_{\tilde{Q}}(s,a)^2\right] - \beta H(\pi_{\tilde{Q}}) \\
=& -\mathbb{E}_{d_{\pi_E}}\left[c\alpha\, r_{\tilde{Q}}(s,a)^2 - r_{\tilde{Q}}(s,a)\right] - \mathbb{E}_{d_\pi}\left[c(1-\alpha)\, r_{\tilde{Q}}(s,a)^2 + r_{\tilde{Q}}(s,a)\right] - \beta H(\pi_{\tilde{Q}}) \\
=& -c\Big(\alpha\,\mathbb{E}_{d_{\pi_E}}\left[r_{\tilde{Q}}(s,a)^2 - \tfrac{1}{\alpha c} r_{\tilde{Q}}(s,a)\right] \\
&+ (1-\alpha)\,\mathbb{E}_{d_\pi}\left[r_{\tilde{Q}}(s,a)^2 + \tfrac{1}{(1-\alpha)c} r_{\tilde{Q}}(s,a)\right] + \tfrac{\beta}{c} H(\pi_{\tilde{Q}})\Big)
\end{aligned}$$

Defining $r_{\max} = \frac{1}{2\alpha c}$ and $r_{\min} = -\frac{1}{2(1-\alpha)c}$ and completing the squares we obtain:

$$\begin{aligned}
\mathcal{J}(\tilde{Q}, \pi_{\tilde{Q}}) =& -c\Big(\alpha\,\mathbb{E}_{d_{\pi_E}}\left[r_{\tilde{Q}}(s,a)^2 - \tfrac{1}{\alpha c} r_{\tilde{Q}}(s,a)\right] + (1-\alpha)\,\mathbb{E}_{d_\pi}\left[r_{\tilde{Q}}(s,a)^2 + \tfrac{1}{(1-\alpha)c} r_{\tilde{Q}}(s,a)\right] \\
&+ \tfrac{\beta}{c} H(\pi_{\tilde{Q}})\Big) + \alpha c\,(r_{\max}^2 - r_{\max}^2) + (1-\alpha)c\,(r_{\min}^2 - r_{\min}^2) \\
=& -c\Big(\alpha\,\mathbb{E}_{d_{\pi_E}}\left[r_{\tilde{Q}}(s,a)^2 - \tfrac{1}{\alpha c} r_{\tilde{Q}}(s,a) + r_{\max}^2\right] + (1-\alpha)\,\mathbb{E}_{d_\pi}\left[r_{\tilde{Q}}(s,a)^2\right. \\
&\left. + \tfrac{1}{(1-\alpha)c} r_{\tilde{Q}}(s,a) + r_{\min}^2\right] + \tfrac{\beta}{c} H(\pi_{\tilde{Q}})\Big) + \alpha c\, r_{\max}^2 + (1-\alpha)c\, r_{\min}^2 \\
=& -c\Big(\alpha\,\mathbb{E}_{d_{\pi_E}}\left[\big(r_{\tilde{Q}}(s,a) - \tfrac{1}{2\alpha c}\big)^2\right] + (1-\alpha)\,\mathbb{E}_{d_\pi}\left[\Big(r_{\tilde{Q}}(s,a) + \tfrac{1}{2(1-\alpha)c}\Big)^2\right] \\
&+ \tfrac{\beta}{c} H(\pi_{\tilde{Q}})\Big) + \alpha c\, r_{\max}^2 + (1-\alpha)c\, r_{\min}^2.
\end{aligned}$$

Finally, we obtain the following result:

$$\mathcal{J}(\tilde{Q}, \pi_{\tilde{Q}}) = -c\Big(\alpha\,\mathbb{E}_{d_{\pi_E}}\left[\big(r_{\tilde{Q}}(s,a) - r_{\max}\big)^2\right] + (1-\alpha)\,\mathbb{E}_{d_\pi}\left[\big(r_{\tilde{Q}}(s,a) - r_{\min}\big)^2\right] + \tfrac{\beta}{c} H(\pi_{\tilde{Q}})\Big) - K, \tag{28}$$

where $K = \alpha c\, r_{\max}^2 + (1-\alpha)c\, r_{\min}^2 = \frac{1}{4\alpha c} + \frac{1}{4(1-\alpha)c}$ is a fixed constant.

Comparing Equation 10 with Equation 28 results in

$$\mathcal{J}(\tilde{Q}, \pi_{\tilde{Q}}) + K \propto \mathcal{L}(\tilde{Q}, \pi_{\tilde{Q}}). \tag{29}$$

Given that an affine transformation (with positive multiplicative constants) preserves the optimum, $\arg\max_{\tilde{Q}\in\tilde{\Omega}} \bar{\mathcal{J}}(\tilde{Q}, \pi_{\tilde{Q}})$ is the solution of the entropy-regularized least squares objective $\mathcal{L}(\tilde{Q}, \pi_{\tilde{Q}})$.

■

B ADDITIONAL EXPERIMENT RESULTS

This section contains additional results that have been omitted in the main paper due to space constraints. Figure 5 presents all the plots presented in Figure 3 with the additional Humanoid-v3 results. Figure 6 and Figure 7 correspond to Figure 4 but show all five environments. The corresponding learning curves are shown in Appendix B.2 and B.1.

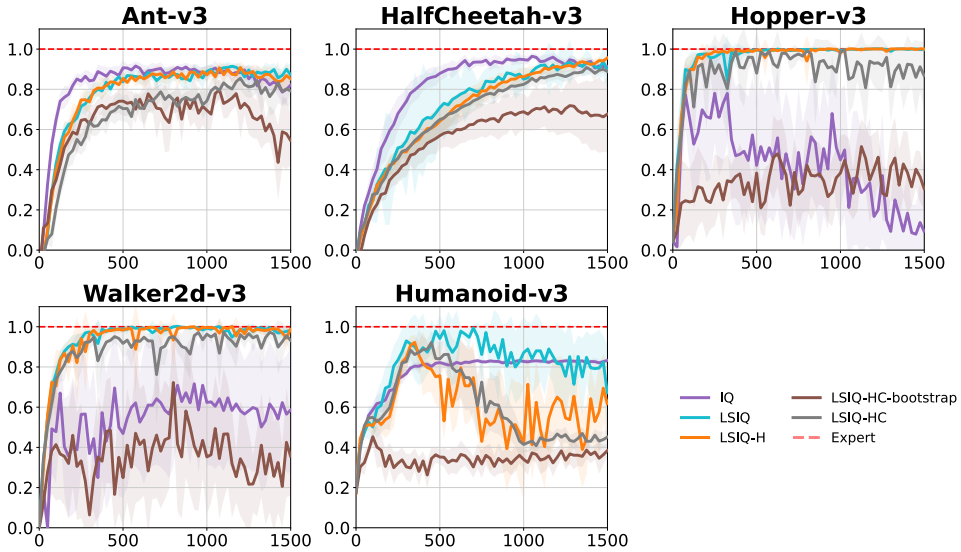

Figure 5: Comparison of different versions of LS-IQ. Now also with the Humanoid-v3 environment. Abscissa shows the normalized discounted cumulative reward. Ordinate shows the number of training steps ($\times 10^3$). Expert cumulative rewards $\rightarrow$ Hopper:3299.81, Walker2d:5841.73, Ant:6399.04, HalfCheetah:12328.78, Humanoid:6233.45

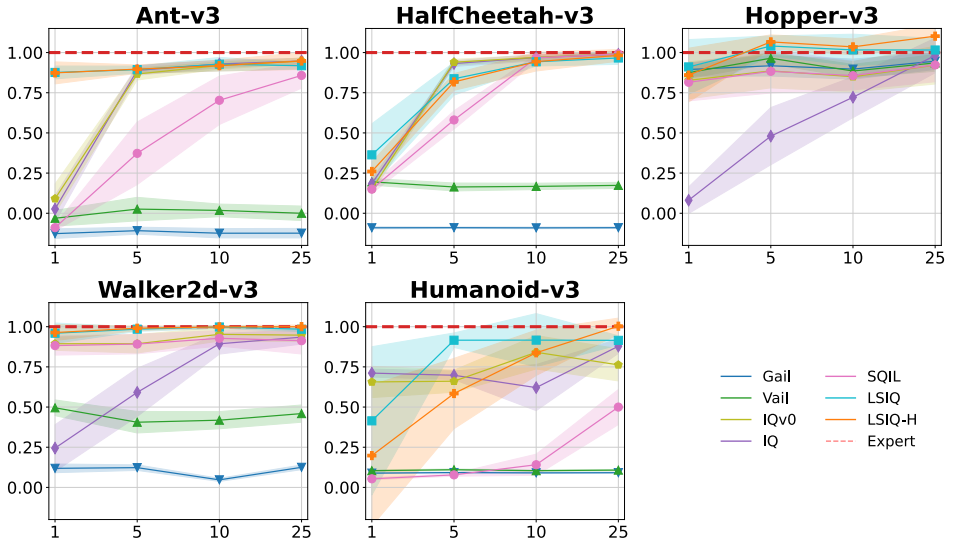

Figure 6: Comparison of the effect of the number of expert trajectories on different Mujoco environments. **States and actions** from the expert are provided to the agent. All plots are added here for the sake of completeness. Abscissa shows the normalized cumulative reward. Ordinate shows the number of expert trajectories. The first row shows the performance when considering states and action, while the second row considers the performance when using states only. Training results are averaged over 5 seeds per agent. The shaded area constitutes the 95% confidence interval. Expert cumulative rewards $\rightarrow$ Hopper:3299.81, Walker2d:5841.73, Ant:6399.04, HalfCheetah:12328.78, Humanoid:6233.45

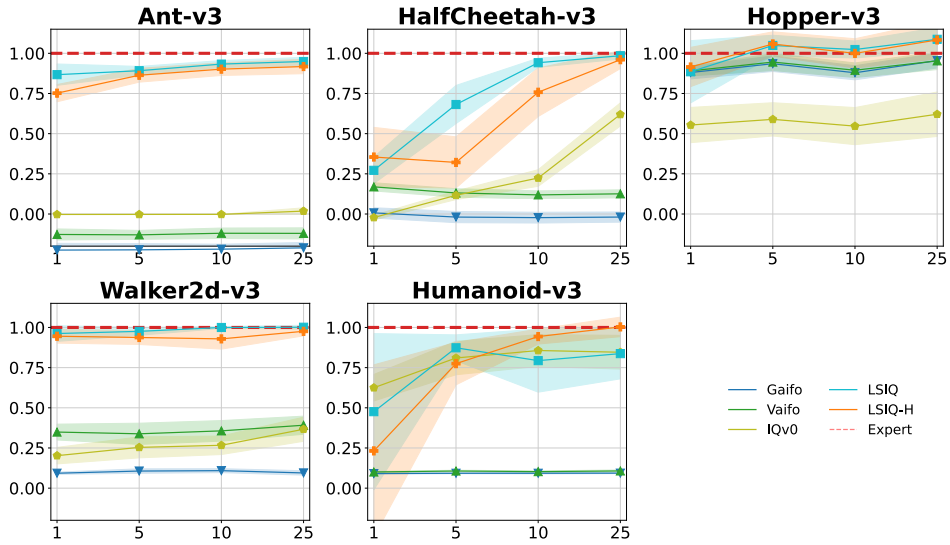

Figure 7: Comparison of the effect of the number of expert trajectories on different Mujoco environments. **Only expert states** are provided to the expert. All plots are added here for the sake of completeness. Abscissa shows the normalized cumulative reward. Ordinate shows the number of expert trajectories. The first row shows the performance when considering states and action, while the second row considers the performance when using states only. Training results are averaged over 5 seeds per agent. The shaded area constitutes the 95% confidence interval. Expert cumulative rewards → Hopper:3299.81, Walker2d:5841.73, Ant:6399.04, HalfCheetah:12328.78, Humanoid:6233.45

B.1 IMITATION LEARNING FROM STATES ONLY – FULL TRAINING CURVES

The learning curves for the learning from observation experiments can be found in Figure 8, 9, 10 and 11 for 1, 5, 10 and 25 expert demonstrations, respectively.

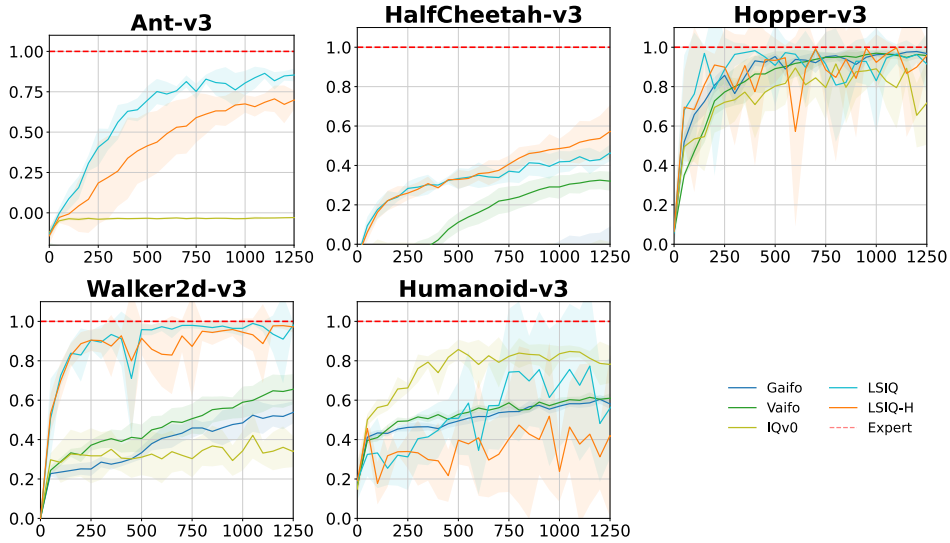

Figure 8: Training performance of different agents on Mujoco Tasks when using **1** expert trajectory consisting of **only states**. Abscissa shows the normalized discounted cumulative reward. Ordinate shows the number of training steps ($\times 10^3$). Training results are averaged over 5 seeds per agent. The shaded area constitutes the 95% confidence interval. Expert cumulative rewards → Hopper:3299.81, Walker2d:5841.73, Ant:6399.04, HalfCheetah:12328.78, Humanoid:6233.45

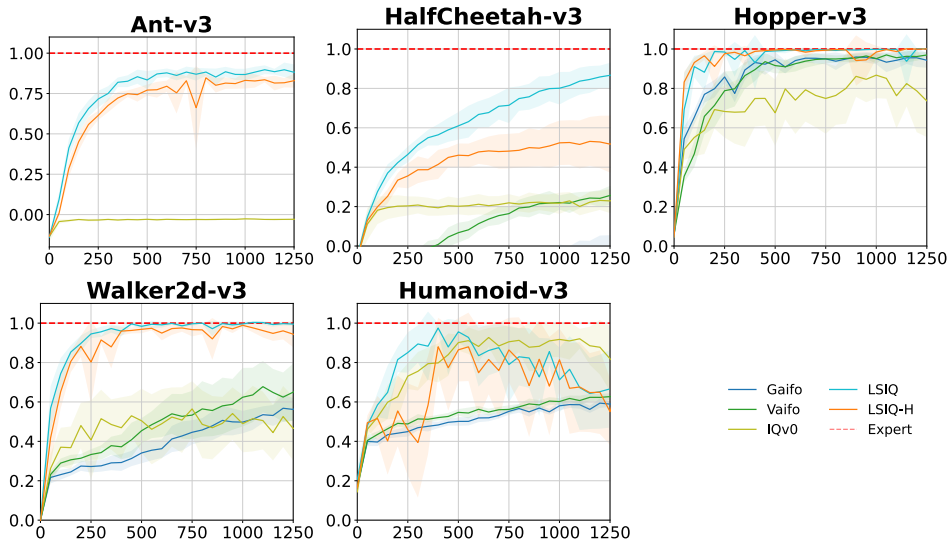

Figure 9: Training performance of different agents on Mujoco Tasks when using **5** expert trajectory consisting of **only states**. Abscissa shows the normalized discounted cumulative reward. Ordinate shows the number of training steps ($\times 10^3$). Training results are averaged over 10 seeds per agent. The shaded area constitutes the 95% confidence interval. Expert cumulative rewards → Hopper:3299.81, Walker2d:5841.73, Ant:6399.04, HalfCheetah:12328.78, Humanoid:6233.45

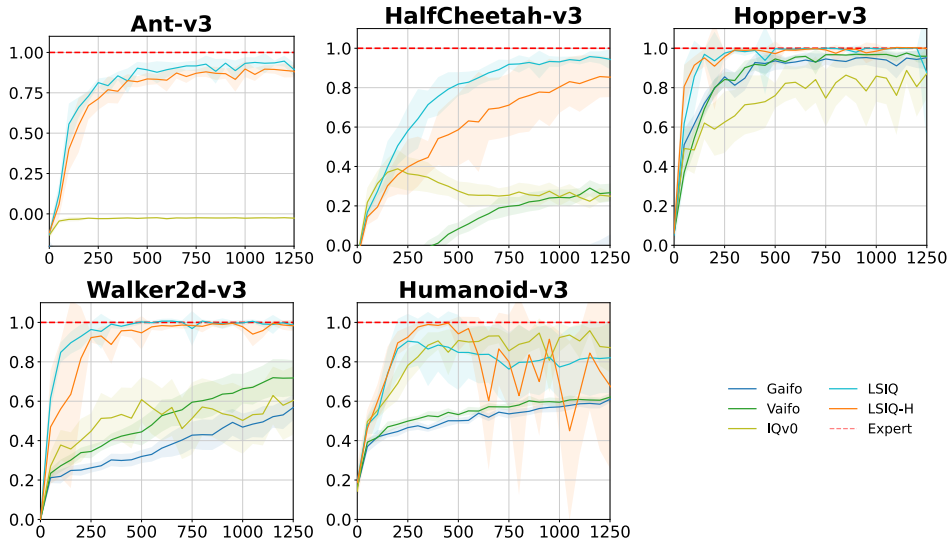

Figure 10: Training performance of different agents on Mujoco Tasks when using **10** expert trajectory consisting of **only states**. Abscissa shows the normalized discounted cumulative reward. Ordinate shows the number of training steps ($\times 10^3$). Training results are averaged over 5 seeds per agent. The shaded area constitutes the 95% confidence interval. Expert cumulative rewards $\rightarrow$ Hopper:3299.81, Walker2d:5841.73, Ant:6399.04, HalfCheetah:12328.78, Humanoid:6233.45

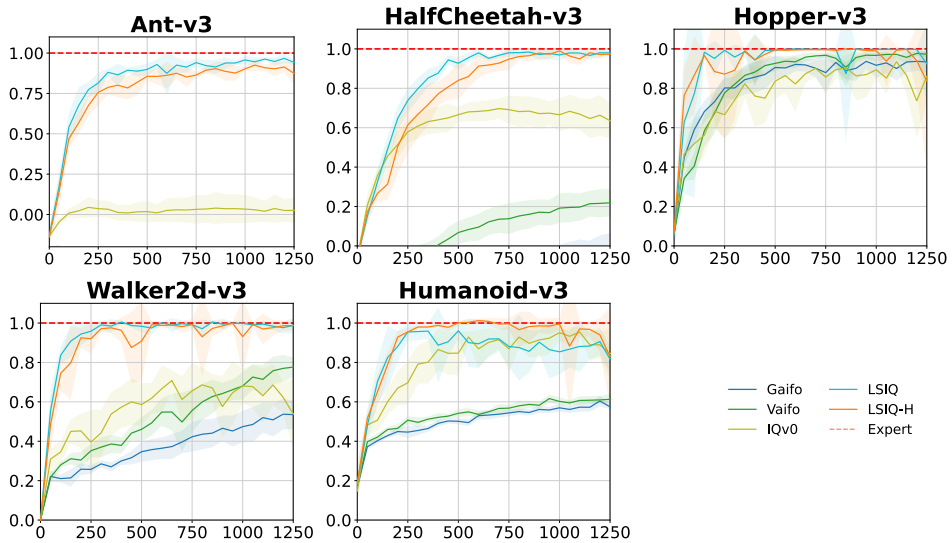

Figure 11: Training performance of different agents on Mujoco Tasks when using **25** expert trajectory consisting of **only states**. Abscissa shows the normalized discounted cumulative reward. Ordinate shows the number of training steps ($\times 10^3$). Training results are averaged over 5 seeds per agent. The shaded area constitutes the 95% confidence interval. Expert cumulative rewards $\rightarrow$ Hopper:3299.81, Walker2d:5841.73, Ant:6399.04, HalfCheetah:12328.78, Humanoid:6233.45

B.2 IMITATION LEARNING FROM STATES AND ACTIONS – FULL TRAINING CURVES

The learning curves for the experiments where states and actions are observed can be found in Figure 12, 13, 14 and 15 for 1, 5, 10 and 25 expert demonstrations, respectively.

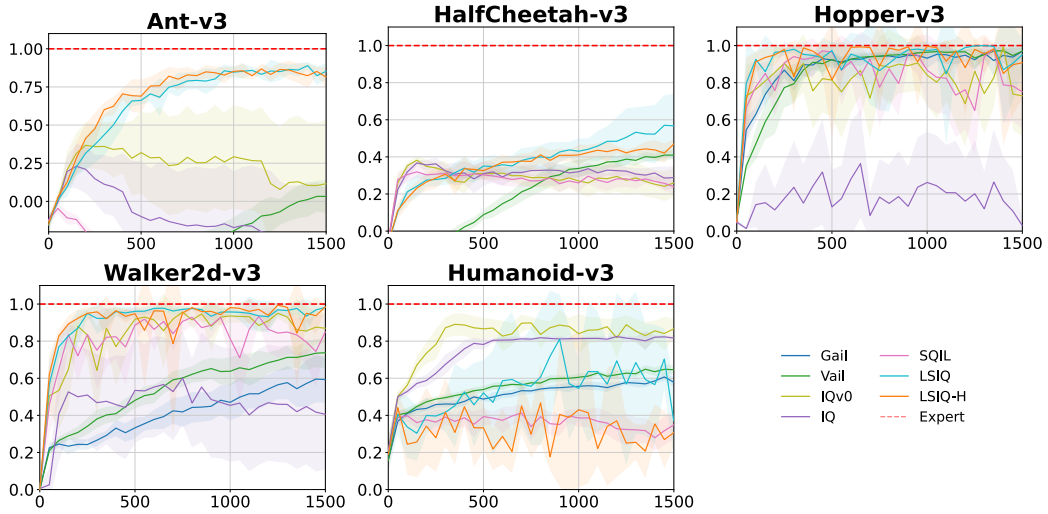

Figure 12: Training performance of different agents on Mujoco Tasks when using **1** expert trajectory. Abscissa shows the normalized discounted cumulative reward. Ordinate shows the number of training steps ($\times 10^3$). Training results are averaged over 5 seeds per agent. The shaded area constitutes the 95% confidence interval. Expert cumulative rewards → Hopper:3299.81, Walker2d:5841.73, Ant:6399.04, HalfCheetah:12328.78, Humanoid:6233.45

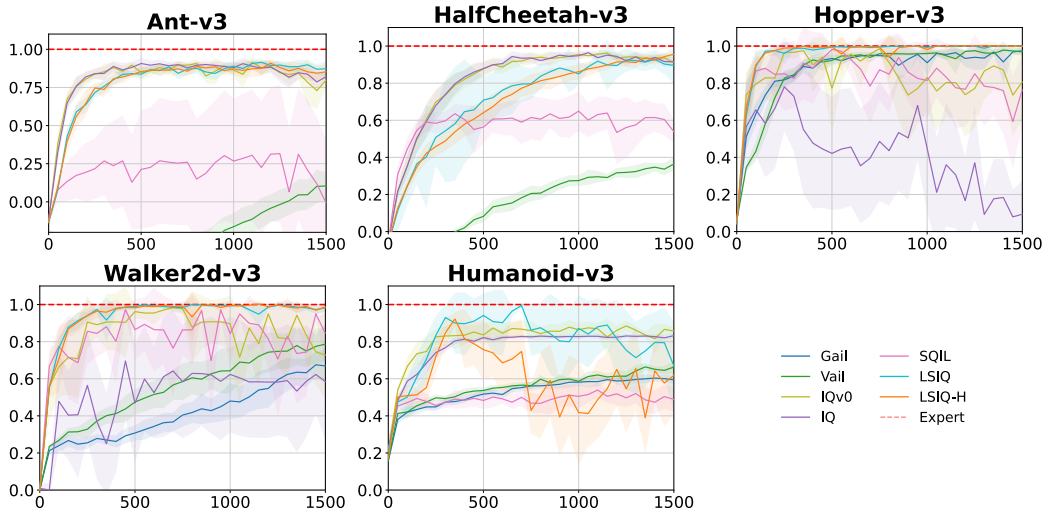

Figure 13: Training performance of different agents on Mujoco Tasks when using **5** expert trajectories. Abscissa shows the normalized discounted cumulative reward. Ordinate shows the number of training steps ($\times 10^3$). Training results are averaged over 10 seeds per agent. The shaded area constitutes the 95% confidence interval. Expert cumulative rewards → Hopper:3299.81, Walker2d:5841.73, Ant:6399.04, HalfCheetah:12328.78, Humanoid:6233.45

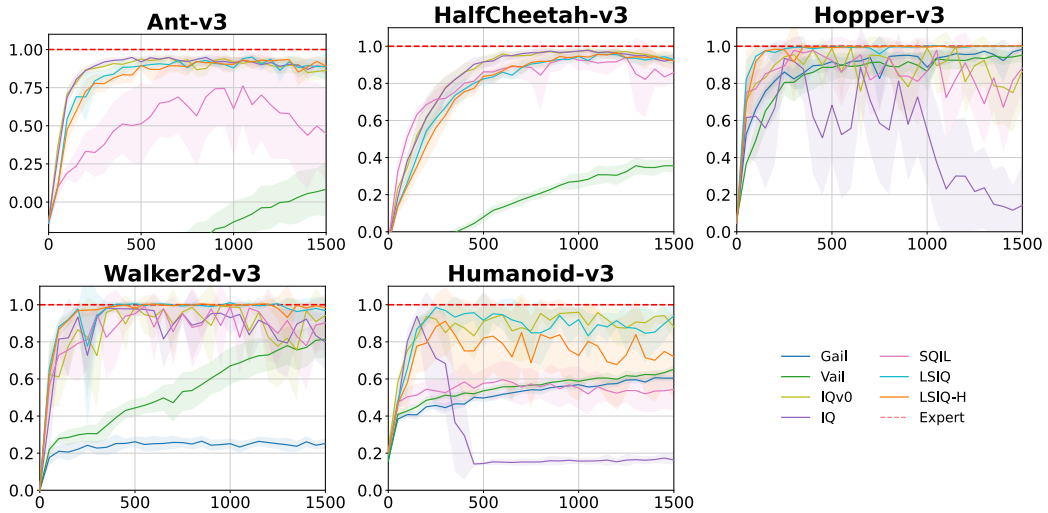

Figure 14: Training performance of different agents on Mujoco Tasks when using **10** expert trajectories. Abscissa shows the normalized discounted cumulative reward. Ordinate shows the number of training steps ($\times 10^3$). Training results are averaged over 5 seeds per agent. The shaded area constitutes the 95% confidence interval. Expert cumulative rewards $\rightarrow$ Hopper:3299.81, Walker2d:5841.73, Ant:6399.04, HalfCheetah:12328.78, Humanoid:6233.45

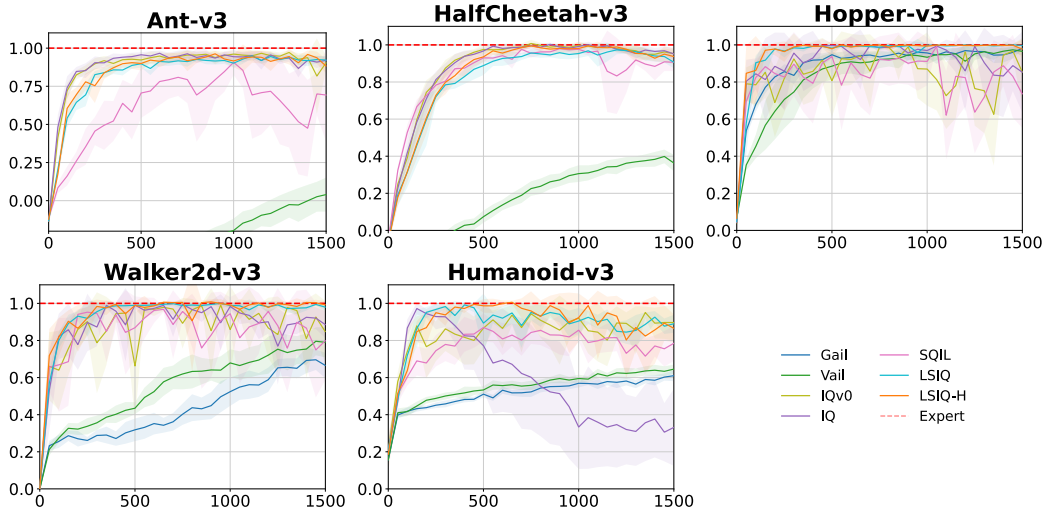

Figure 15: Training performance of different agents on Mujoco Tasks when using **25** expert trajectories. Abscissa shows the normalized discounted cumulative reward. Ordinate shows the number of training steps ($\times 10^3$). Training results are averaged over 5 seeds per agent. The shaded area constitutes the 95% confidence interval. Expert cumulative rewards $\rightarrow$ Hopper:3299.81, Walker2d:5841.73, Ant:6399.04, HalfCheetah:12328.78, Humanoid:6233.45