IMITATION LEARNING BY REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Imitation learning algorithms learn a policy from demonstrations of expert behavior. We show that, for deterministic experts, imitation learning can be done by reduction to reinforcement learning with a stationary reward. Our theoretical analysis both certifies the recovery of expert reward and bounds the total variation distance between the expert and the imitation learner, showing a link to adversarial imitation learning. We conduct experiments which confirm that our reduction works well in practice for continuous control tasks.

1 INTRODUCTION

Typically, reinforcement learning (RL) assumes access to a pre-specified reward and then learns a policy maximizing the expected average of this reward along a trajectory. However, specifying rewards is difficult for many practical tasks (Atkeson & Schaal, 1997; Zhang et al., 2018; Ibarz et al., 2018). In such cases, it is convenient to instead perform Imitation Learning (IL), learning a policy from expert demonstrations.

There are two major categories of Imitation Learning algorithms: behavioral cloning and inverse reinforcement learning. Behavioral cloning learns the policy by supervised learning on expert data, but is not robust to training errors, failing in settings where expert data is limited (Ross & Bagnell, 2010). Inverse reinforcement learning (IRL) achieves improved performance on limited data by constructing reward signals and calling an RL oracle to maximize these rewards (Ng et al., 2000).

The most versatile IRL method is adversarial IL (Ho & Ermon, 2016; Li et al., 2017; Ghasemipour et al., 2020), which minimizes a divergence between the distribution of data produced by the agent and provided by the expert. Adversarial IL learns a representation and a policy simultaneously by using a non-stationary reward obtained from a discriminator network. However, training adversarial IL combines two components which are hard to stabilize: a discriminator network, akin to the one used in GANs, as well as a policy, typically learned with an RL algorithm with actor and critic networks. This complexity makes the training process very brittle.

There is a clear need for imitation learning algorithms that are simpler and easier to deploy. To address this need, Wang et al. (2019) proposed to reduce imitation learning to a single instance of reinforcement learning problem, where reward is defined to be one for state-action pairs from the expert trajectory and zero for other state-action pairs. A closely related, but not identical, algorithm has been proposed by Reddy et al. (2020) (we describe the differences in Section 5). However, while empirical performance of these approaches has been good, they enjoy no performance guarantees at all, even in the asymptotic setting where expert data is infinite.

Contributions We fill in this missing justification for this algorithm, providing the needed theoretical analysis. Specifically, in Sections 3 and 4, we show a total variation bound between the expert policy and the imitation policy, providing a high-probability performance guarantee for a finite dataset of expert data and linking the reduction to adversarial imitation learning algorithms. For stochastic experts, we describe how the reduction fails, completing the analysis. Moreover, in Section 6, we empirically evaluate the performance of the reduction as the amount of available expert data varies.

2 PRELIMINARIES

Markov Decision Process An average-reward Markov Decision Process (Puterman, 2014; Feinberg & Shwartz, 2012) is a tuple (S, A, T, R, s_1), where S is a state space, A is the action space, $T : S \times A \to P(S)$ is the transition model, $R : S \times A \to [0, 1]$ is a bounded reward function and s_1 is the initial state. Here, we write $P(X)$ to denote probability distributions over a set X. A stationary policy $\pi : S \to P(A)$ maps environment states to distributions over actions. A policy π induces a Markov chain P_π. In the theoretical part of the paper, we treat MDPs with finite state and action spaces. Given a starting state s_1 of the MDP, and a policy π, the limiting distribution over states is defined as

$$\rho_S^\pi(s) = \mathbb{1}(s_1)^\top \lim_{N\to\infty} \frac{1}{N} \sum_{i=0}^{N} P_\pi^i, \tag{1}$$

where $\mathbb{1}(s_1)$ denotes the indicator vector. We adopt the convention that the subscript S indicates a distribution over states and a no subscript indicates a distribution over state-action pairs. We denote $\rho^\pi(s, a) = \rho_S^\pi(s)\pi(a|s)$. While the limit in equation 1 is guaranteed to exist for all finite MDPs (Puterman, 2014), without requiring ergodicity, in this paper we consider policies that induce ergodic chains. Correspondingly, we drop the dependence on the initial state from the notation. The expected per-step reward of π is defined as

$$V^\pi = \lim_{N\to\infty} E_\pi \left[J_N/N \mid s_1\right] = E_{\rho^\pi} R(s, a), \tag{2}$$

where $J_N = \sum_{t=1}^{N} R(s_t, a_t)$ is the total return.

Expert Dataset Assume that the expert follows an ergodic policy π_E. Denote the corresponding distribution over expert state-action pairs as $\rho^E(s, a) = \rho^{\pi_E}(s, a)$, where we dropped the dependency on the initial state by ergodicity. Consider a finite trajectory $s_1, a_1, s_2, a_2, \ldots, s_N, a_N$. Denote by $\hat{\rho}(s, a) = \frac{1}{N}\sum_{t=1}^{N} \mathbb{1}\left((s, a) = (s_t, a_t)\right)$ the histogram (empirical distribution) of data seen along this trajectory and denote by $\hat{\rho}_S(s) = \frac{1}{N}\sum_{t=1}^{N} \mathbb{1}\left(s = s_t\right)$ the corresponding histogram over the states. Denote by D the support of $\hat{\rho}$, i.e. the set of state-action pairs visited by the expert.

Imitation Learning Learning to imitate means obtaining a policy that mimics expert behavior. This can be formalized in two ways. First, we can seek an imitation policy π_I which obtains a expected per-step reward that is ϵ-close to the expert on any bounded reward signal. Denote the limiting distribution of state-action tuples generated by the imitation learner with $\rho^I(s, a)$. Formally, we want to satisfy

$$\forall R \in S \times A \to [0, 1].\ \ E_{\rho^E}[R] - E_{\rho^I}[R] \leq \epsilon, \tag{3}$$

where ϵ is a small constant. Second, we can seek to ensure that the distributions of state-action pairs generated by the expert and the imitation learner are similar (Ho & Ermon, 2016; Finn et al., 2016). In particular, if we measure the divergence between distributions with total variation, we want to ensure

$$\|\rho^E - \rho^I\|_{\text{TV}} \leq \epsilon. \tag{4}$$

We recall in Appendix C that equation 3 and equation 4 are in fact closely related. We provide further background on adversarial imitation learning and other divergences in Section 5.

3 IMITATION LEARNING BY REINFORCEMENT LEARNING

Algorithm To perform imitation learning, we first obtain an expert dataset D and construct an intrinsic reward

$$R_I(s, a) = \mathbb{1}((s, a) \in D). \tag{5}$$

We can then use any RL algorithm to solve for this reward. We note that for finite MDPs equation 5 exactly matches the algorithm proposed by Wang et al. (2019) as a heuristic.

Guarantee on Imitation Policy The main focus of the paper lies on showing theoretical properties of the imitation policy obtained when solving for the reward signal defined in equation 5. Our formal guarantees hold under three assumptions.

Assumption 1. *The expert policy is deterministic.*

Assumption 2. *The expert policy induces an irreducible and aperiodic Markov chain.*

Assumption 3. *The imitation learner policy induces an irreducible and aperiodic Markov chain.*

Assumption 1 is critical for our proof to go through and cannot be relaxed. It is also essential to rationalize the approach of Wang et al. (2019). In fact, no reduction involving a single call to an RL solver with stationary reward can exist for stochastic experts since for any MDP there is always an optimal policy which is deterministic. Assumptions 2 and 3 could in principle be relaxed, to allow for periodic chains, but it would complicate the reasoning, which we wanted to avoid. We note that we only interact with the expert policy via a finite dataset of N samples.

Our main contribution is Proposition 1, in which we quantify the performance of our imitation learner. We state it below and prove it in Section 4.

Proposition 1. *Consider an imitation learner trained on a dataset of size N, which attains the limiting distribution of state-action pairs ρ^I. Under Assumptions 1, 2 and 3, given we have $N \geq \max\left\{\frac{A}{\eta}, \log\left(\frac{2|S|}{\delta}\right)|S|^2\frac{B}{\eta^2}\right\}$ expert demonstrations, with probability at least $1-\delta$, the imitation learner attains total variation distance from the expert of at most*

$$\|\rho^E - \rho^I\|_{TV} \leq \eta. \tag{6}$$

For brevity, we introduced $A = 2C(1+\frac{2C}{1-\alpha})(1-\alpha)^{-1}$ and $B = 4.5\tau_{mix}(1+\frac{2C}{1-\alpha})^2$, where the constants C, α and τ_{mix} reflect the mixing time of the expert policy and are defined in Appendix D. Moreover, with the same probability, for any extrinsic reward function, the imitation learner achieves expected per-step extrinsic reward of at least

$$E_{\rho^I} R_E \geq E_{\rho^E} R_E - \eta. \tag{7}$$

Proposition 1 shows that the policy learned by imitation learner satisfies two important desiderata: we can guarantee both the closeness of the generated state-action distribution to the expert distribution as well as recovery of expert reward. Moreover, the total variation bound in equation 6 links the obtained policy to adversarial imitation learning algorithms. We describe this link in more detail in Section 5.

4 Proof

Structure of Results We begin by showing a result about mixing in Markov chains in Section 4.1. We then give our proof, which has two main parts. In the first part, in Section 4.2, we show that it is possible to achieve a high expected per-step intrinsic reward defined in equation 5. In the second part, in Section 4.3, we show that, achieving a large expected per-step intrinsic reward guarantees a large expected per-step extrinsic reward for any bounded extrinsic reward. In Section 4.4, we combine these results and prove Proposition 1.

4.1 Mixing in Markov Chains

We now show a lemma that quantifies how fast the histogram approaches the stationary distribution. The proof of the lemma relies on standard results about the Total Variation distance between probability distributions as well as McDiarmid's inequality for Markov chains, which we recall in Appendices C and D,

Lemma 1. *The total variation distance between the expert distribution and the expert histogram $\hat{\rho}$ based on N samples can be bounded as*

$$\|\rho^E - \hat{\rho}\|_{TV} \leq \epsilon + C/N\frac{1-\alpha^N}{1-\alpha} \tag{8}$$

with probability at least $1 - 2|S| \exp\left\{-\frac{4\epsilon^2 N}{4.5|S|^2 \tau_{mix}}\right\}$ *(where the probability space is defined by the process generating the expert histogram).*

Proof. We first prove the statement for histograms over the states. Recall the notation $\hat{\rho}_S = \frac{1}{N}\sum_{t=1}^{N} \mathbb{1}\left(s_t = s\right)$. Recall that we denote the stationary distribution (over states) of the Markov chain induced by the expert with ρ_S^E.

Define $f_s(s_1, \ldots, s_N) = \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}(s_i = s)$. For two state sequences $s_1, \ldots, s_N$ and $s'_1, \ldots, s'_N$ we have $f_s(s_1, \ldots, s_N) - f_s(s'_1, \ldots, s'_N) \leq \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}(s_i \neq s'_i)$. Using this property, instantiate Lemma 8 (see Appendix D) with $c_i = 1/N$ for each state separately with f_s as above. Using the fact that $f_s(s_1, \ldots, s_N) = \hat{\rho}_S(s)$, this gives us $P(|\hat{\rho}_S(s) - E\hat{\rho}_S(s)| \leq \epsilon') \geq 1 - 2\exp\left\{-\frac{\epsilon'^2 N}{4.5\tau_{\text{mix}}}\right\}$, where we introduced the notation $E\hat{\rho}_S(s) = 1/N \sum_{i=0}^{N-1} (\mathbb{1}(s_1)^\top P_E^i)(s)$. Applying the union bound over all states, we have $P(\forall s. |\hat{\rho}_S(s) - E\hat{\rho}_S(s)| \leq \epsilon') \geq 1 - 2|S|\exp\left\{-\frac{\epsilon'^2 N}{4.5\tau_{\text{mix}}}\right\}$. This implies $P(\|\hat{\rho}_S - E\hat{\rho}_S\|_1 \leq |S|\epsilon') \geq 1 - 2|S|\exp\left\{-\frac{\epsilon'^2 N}{4.5\tau_{\text{mix}}}\right\}$. Introducing $\epsilon'' = |S|\epsilon'$, this is equivalent to

$$P(\|\hat{\rho}_S - E\hat{\rho}_S\|_1 \leq \epsilon'') \geq 1 - 2|S|\exp\left\{-\frac{\epsilon''^2 N}{4.5|S|^2\tau_{\text{mix}}}\right\}. \tag{9}$$

We now quantify the distance between $E\hat{\rho}$ and the stationary distribution.

$$\begin{aligned}
\|E\hat{\rho}_S - \rho_S^E\|_1 &= \left\|\left(1/N\sum_{i=0}^{N-1} \mathbb{1}(s_1)^\top P_E^i\right) - \rho_S^E\right\|_1 = (1/N)\left\|\sum_{i=0}^{N-1}\left(\mathbb{1}(s_1)^\top P_E^i - \rho_S^E\right)\right\|_1 \\
&\leq (1/N)\sum_{i=0}^{N-1}\left\|\left(\mathbb{1}(s_1)^\top P_E^i - \rho_S^E\right)\right\|_1 \leq (1/N)\sum_{i=0}^{N-1} 2C\alpha^i \\
&= (2C/N)\frac{1-\alpha^N}{1-\alpha}.
\end{aligned} \tag{10}$$

Here, the last inequality follows from Lemma 7 (see Appendix D) and the identity $\|\rho_1 - \rho_2\|_{\text{TV}} = \frac{1}{2}\|\rho_1 - \rho_2\|_1$. We combine equation 9 and equation 10 using the triangle inequality. Using the property $\|\rho_1 - \rho_2\|_{\text{TV}} = \frac{1}{2}\|\rho_1 - \rho_2\|_1$ again and setting $\epsilon = \epsilon''/2$, we obtain

$$\|\rho_S^E - \hat{\rho}_S\|_{\text{TV}} \leq \epsilon + C/N\frac{1-\alpha^N}{1-\alpha} \tag{11}$$

with probability at least $1 - 2|S|\exp\left\{-\frac{4\epsilon^2 N}{4.5|S|^2\tau_{\text{mix}}}\right\}$. Since the expert policy is deterministic, the total variation distance between the distributions of state-action tuples and states is the same, i.e. $\|\rho_S^E - \hat{\rho}_S\|_{\text{TV}} = \|\rho^E - \hat{\rho}\|_{\text{TV}}$. □

4.2 High Expected Intrinsic Per-Step Reward is Achievable

We now want to show that it is possible for the imitation learner to attain a large intrinsic reward. Informally, the proof asks what intrinsic reward the expert would have achieved. We then conclude that a learner specifically optimizing the intrinsic reward will obtain at least as much.

Lemma 2. *Generating an expert histogram with N points, with probability at least $1 - 2|S|\exp\left\{-\frac{4\epsilon^2 N}{4.5|S|^2\tau_{mix}}\right\}$, we obtain a dataset such that a policy π_I maximizing the intrinsic reward $R_I(s,a) = \mathbb{1}((s,a) \in D)$ satisfies $E_{\rho^I}[R_I] \geq 1 - \epsilon - C/N\frac{1-\alpha^N}{1-\alpha}$, where we used the shorthand notation $\rho^I = \rho^{\pi_I}$ to denote the state-action distribution of π_I.*

Proof. We invoke Lemma 1 obtaining $\|\rho^E - \hat{\rho}\|_{\text{TV}} \leq \epsilon + C/N\frac{1-\alpha^N}{1-\alpha}$ with probability as in the statement of the lemma. First, we prove

$$\sum_{s,a} \rho^E(s,a)\mathbb{1}[\hat{\rho}(s,a) = 0] \leq \epsilon + C/N\frac{1-\alpha^N}{1-\alpha}, \tag{12}$$

by setting $\rho_1 = \rho^E$, $\rho_2 = \hat{\rho}$ and $M = \{(s,a) : \hat{\rho}(s,a) = 0\}$ in equation 26.

Combining

$$1 = \sum_{s,a} \rho^E(s,a) = \sum_{s,a} \rho^E(s,a) \mathbb{1}[\hat{\rho}(s,a) = 0] + \sum_{s,a} \rho^E(s,a) \mathbb{1}[\hat{\rho}(s,a) > 0] \tag{13}$$

and equation 12, we obtain

$$\sum_{s,a} \rho^E(s,a) \mathbb{1}[\hat{\rho}(s,a) > 0] \geq 1 - \epsilon - C/N \tfrac{1-\alpha^N}{1-\alpha}, \tag{14}$$

which means that the expert policy achieves expected per-step intrinsic reward of at least $1 - \epsilon - C/N \frac{1-\alpha^N}{1-\alpha}$. This lower bounds the expected per-step reward obtained by the optimal policy. □

4.3 Maximizing Intrinsic Reward Leads to High Per-Step Extrinsic Reward

We now aim to prove that the intrinsic and extrinsic rewards are connected, i.e. maximizing the intrinsic reward leads to a large expected per-step extrinsic reward. The proofs in this section are based on the insight that, by construction of intrinsic reward in equation 5, attaining intrinsic reward of one in a given state implies agreement with the expert. In the following Lemma, we quantify the outcome of achieving such agreement for l consecutive steps.

Lemma 3. *Assume an agent traverses a sequence of l state-action pairs, in a way consistent with the expert policy. Denote the expert's expected extrinsic per-step reward with $E_{\rho^E} R_E$. Denote by $\tilde{\rho}_t^{p_1}(s,a) = (p_1^\top P_E^t)(s)\pi_E(a|s)$ the expected state-action occupancy at time t, starting in state distribution p_1. The per-step extrinsic reward $\tilde{V}_{p_1}^l$ of the agent in expectation over realizations of the sequence satisfies*

$$\tilde{V}_{p_1}^l = \sum_{s,a} \left(\left(\sum_{t=0}^{l-1} \tfrac{1}{l} \tilde{\rho}_t^{p_1}(s,a) \right) R_E(s,a) \right) \geq E_{\rho^E} R_E - \tfrac{1}{l} \tfrac{2C}{1-\alpha}. \tag{15}$$

Proof. Invoking Lemma 7 (see Appendix D), we have that $\|\tilde{\rho}_t^{p_1} - \rho^E\|_{\text{TV}} \leq C\alpha^t$. Invoking Lemma 6 (see Appendix C), we have that $E_{\rho^E} R_E - E_{\tilde{\rho}_t^{p_1}} R_E \leq 2C\alpha^t$ for any timestep t. In other words, the expected per-step reward obtained in step t of the sequence is at least $E_{\rho^E} R_E - 2C\alpha^t$. The per-step reward in the sequence is at least

$$\begin{aligned} \tilde{V}_{p_1}^l \geq \tfrac{1}{l} \textstyle\sum_{i=0}^{l-1} (E_{\rho^E} R_E - 2C\alpha^i) &= E_{\rho^E} R_E - \tfrac{2}{l} C \textstyle\sum_{i=0}^{l-1} \alpha^i \\ &= E_{\rho^E} R_E - \tfrac{2}{l} C \tfrac{1-\alpha^l}{1-\alpha} \geq E_{\rho^E} R_E - \tfrac{2}{l} C \tfrac{1}{1-\alpha}. \end{aligned}$$

□

In Lemma 3, we have shown that, on average, agreeing with the expert for a number of steps guarantees a certain level of extrinsic reward. We will now use this result to guarantee extrinsic reward obtained over a long trajectory.

Lemma 4. *For any extrinsic reward signal R_E bounded in $[0,1]$, an imitation learner which attains expected per-step intrinsic reward of $E_{\rho^I} R_I$ also attains extrinsic per-step reward of at least*

$$E_{\rho^I} R_E \geq (E_{\rho^I} R_I)(E_{\rho^E} R_E) - \frac{2C}{1-\alpha}(1 - E_{\rho^I} R_I),$$

with probability one, where $E_{\rho^E} R_E$ is the expected per-step extrinsic reward of the expert.

Proof. Consider the imitation learner's trajectory of length T. We will now consider sub-sequences where the agent agrees with the expert. Denote by M_l the number of such sequences of length l. Denote by $\hat{V}^{i,l}$ the per-step extrinsic reward obtained by the agent in the ith sequence of length l.

Assuming the worst case reward on states which are not in any of the sequences (i.e. where the agent disagrees with the expert), the total extrinsic return J_T along the trajectory is at least $\sum_l l \sum_{i=1}^{M_l} \hat{V}^{i,l} = \sum_l l M_l \frac{1}{M_l} \sum_{i=1}^{M_l} \hat{V}^{i,l}$. Dividing by the sequence length, the per-step extrinsic reward is at least

$$\frac{J_T}{T} \geq \textstyle\sum_l \frac{l M_l}{T} \frac{1}{M_l} \sum_{i=1}^{M_l} \hat{V}^{i,l}. \tag{16}$$

Denote by $\Delta_l = \left| \frac{1}{M_l} \sum_{i=1}^{M_l} \hat{V}^{i,l} - \frac{1}{M_l} \sum_{i=1}^{M_l} \tilde{V}^l_{p(s_1^l)} \right|$ the difference between the average extrinsic reward obtained in sequences of length l and its expected value, where we denoted by $p(s_1^l)$ the initial state distribution among sequences of length l. We can now re-write equation 16 as:

$$\frac{J_T}{T} \geq \sum_l \frac{lM_l}{T} \frac{1}{M_l} \sum_{i=1}^{M_l} \tilde{V}^l_{p(s_1^l)} - \sum_l \frac{lM_l}{T} \Delta_l. \tag{17}$$

Using Lemma 3, we can re-write this further

$$\frac{J_T}{T} \geq \sum_l \frac{lM_l}{T} \left(E_{\rho^E} R_E - \frac{1}{l} \frac{2C}{1-\alpha} \right) - \sum_l \frac{lM_l}{T} \Delta_l \tag{18}$$

$$= \sum_l \frac{lM_l}{T} E_{\rho^E} R_E - \sum_l \frac{M_l}{T} \frac{2C}{1-\alpha} - \sum_l \frac{lM_l}{T} \Delta_l. \tag{19}$$

Denote by B the number of timesteps the imitation learner disagrees with the expert. Observe that we have $\sum_l \frac{M_l}{T} \leq \frac{B+1}{T}$ since B bad timesteps can partition the trajectory into at most $B+1$ sub-sequences and $\sum_l M_l$ is the total number of sub-sequences. This gives

$$\frac{J_T}{T} \geq \sum_l \frac{lM_l}{T} E_{\rho^E} R_E - \frac{B+1}{T} \frac{2C}{1-\alpha} - \sum_l \frac{lM_l}{T} \Delta_l. \tag{20}$$

Now, using the fact that the imitation learner policy is ergodic, we can take limits as $T \to \infty$. The left hand side converges to $E_{\rho^I} R_E$ with probability one. On the right-hand side, $\sum_l \frac{lM_l}{T}$ (the fraction of time the imitation learner agrees with the expert) converges to $E_{\rho^I} R_I$ with probability one. Using ergodicity again, $\frac{B+1}{T}$ converges to $1 - E_{\rho^I} R_I$ with probability one.

It remains to prove that the error term $\sum_l \frac{lM_l}{T} \Delta_l$ converges to zero with probability one. First, we will show that, for every l, either M_l is always zero, i.e. the streak length l does not occur in sub-sequences of any length or $M_l \to \infty$ as $T \to \infty$. Indeed, the chain is ergodic, which means that, if we traversed a streak of length l, we have non-zero probability of returning to where the streak begun and then retracing it. Let us call the of set of all ls where $M_l = 0$ by $\mathbb{N} - L_\infty$. Moreover, let us call the set of ls with $M_l \to \infty$ as $T \to \infty$ with L_∞. We have:

$$\sum_{l \in L_\infty} \frac{lM_l}{T} \Delta_l + \sum_{l \in \mathbb{N} - L_\infty} \frac{lM_l}{T} \Delta_l \leq \sup_{l \in L_\infty} \Delta_l. \tag{21}$$

The second term in the sum equals zero with because of the definition of L_∞. The term $\sup_{l \in L_\infty} \Delta_l$ converges to zero with probability one by the law of large numbers, where we use the fact that $M_l \to \infty$ as $T \to \infty$ for $M_l \in L_\infty$. □

4.4 Bringing the Pieces Together

We now show how Lemma 2 and Lemma 4 can be combined to obtain Proposition 1 (stated in Section 3).

Proof of Proposition 1. We use the assumption that

$$N \geq \max \left\{ 2C(1 + \frac{2C}{1-\alpha})(1-\alpha)^{-1} \frac{1}{\eta}, \log \left(\frac{2|S|}{\delta} \right) 4.5|S|^2 \tau_{\text{mix}} \eta^{-2} (1 + \frac{2C}{1-\alpha})^2 \right\}. \tag{22}$$

We will instantiate Lemma 2 with

$$\epsilon = \frac{\eta}{2(1 + \frac{2C}{1-\alpha})}. \tag{23}$$

Equations 22 and 23 imply that $N \geq \log \left(\frac{2|S|}{\delta} \right) 4.5|S|^2 \tau_{\text{mix}} \eta^{-2} (1 + \frac{2C}{1-\alpha})^2 = \log \left(\frac{2|S|}{\delta} \right) 4.5|S|^2 \tau_{\text{mix}} (2\epsilon)^{-2}$. We can rewrite this as $\frac{4\epsilon^2 N}{4.5|S|^2 \tau_{\text{mix}}} \geq \log \frac{2|S|}{\delta}$, which implies $-\frac{4\epsilon^2 N}{4.5|S|^2 \tau_{\text{mix}}} \leq \log \frac{\delta}{2|S|}$. This implies that

$$\delta \geq 2|S| \exp \left\{ -\frac{4\epsilon^2 N}{4.5|S|^2 \tau_{\text{mix}}} \right\}. \tag{24}$$

Moreover, using equation 22 again, we have that $N \geq 2C(1+\frac{2C}{1-\alpha})(1-\alpha)^{-1}\frac{1}{\eta}$. This is equivalent to $\frac{C}{N}\frac{1}{1-\alpha} \leq \frac{1}{2}\frac{\eta}{1+\frac{2C}{1-\alpha}}$. Using equation 23, we obtain $\epsilon + \frac{C}{N}\frac{1}{1-\alpha} \leq \frac{\eta}{1+\frac{2C}{1-\alpha}}$, equivalent to $(\epsilon + \frac{C}{N}\frac{1}{1-\alpha})(1+\frac{2C}{1-\alpha}) \leq \eta$, which implies $(\epsilon + \frac{C}{N}\frac{1-\alpha^N}{1-\alpha})(1+\frac{2C}{1-\alpha}) \leq \eta$. Rewriting this gives

$$\kappa\left(1+\frac{2C}{1-\alpha}\right) \leq \eta, \tag{25}$$

where we use the notation $\kappa = \epsilon + C/N\frac{1-\alpha^N}{1-\alpha}$.

We first show equation 7, which states that we are able to recover expected per-step expert reward. Invoking Lemma 2 and using equation 24, we have that it is possible for the imitation learner to achieve per-step expected intrinsic reward of at least $1-\kappa$ with probability $1-\delta$. Invoking Lemma 4, this implies achieving extrinsic reward of at least

$$E_{\rho^I} R_E \geq E_{\rho^E} R_E (1-\kappa) - \frac{2C\kappa}{1-\alpha}.$$

This implies

$$E_{\rho^E} R_E - E_{\rho^I} R_E \leq E_{\rho^E} R_E - E_{\rho^E} R_E (1-\kappa) + \frac{2C\kappa}{1-\alpha} = \kappa E_{\rho^E} R_E + \frac{2C\kappa}{1-\alpha} \leq \kappa + \frac{2C\kappa}{1-\alpha} \leq \eta,$$

where the last inequality follows from equation 25. Since this statement holds for any extrinsic reward signal R_E, we obtain the total variation bound by invoking Lemma 5 (see Appendix C), getting $\|\rho^E - \rho^I\|_{\text{TV}} \leq \eta$. □

5 Related Work

Behavioral Cloning The simplest way of performing imitation learning is to learn a policy by fitting a Maximum Likelihood model on the expert dataset. While such 'behavioral cloning' is easy to implement, it does not take into account the sequential nature of the Markov Decision Process, leading to catastrophic compounding of errors (Ross & Bagnell, 2010). In practice, the performance of policies obtained with behavioral cloning is highly dependent on the amount of available data. In contrast, our reduction avoids the problem faced by behavioral cloning and provably works with limited expert data.

Apprenticeship Learning Apprenticeship Learning (Abbeel & Ng, 2004) assumes the existence of a pre-learned linear representation for rewards and then proceeds in multiple iterations. In each iteration, the algorithm first computes an intrinsic reward signal and then calls an RL solver to obtain a policy. Our algorithm also uses an RL solver, but unlike Apprenticeship Learning, we only call it once.

Adversarial IL Modern adversarial IL algorithms (Ho & Ermon, 2016; Finn et al., 2016; Li et al., 2017; Sun et al., 2019) remove the requirement to provide a representation by learning it online. They work by minimizing a divergence between the expert state-action distribution and the one generated by the RL agent. For example, the GAIL algorithm (Ho & Ermon, 2016) minimizes the Jensen-Shannon divergence. It can be related to the total variation objective of equation 4 in two ways. First, the TV and JS divergences obey $2\sqrt{D_{\text{JS}}(\rho^E, \rho^I)} \geq \|\rho^E - \rho^I\|_{\text{TV}}$, so that $D_{\text{JS}}(\rho^E, \rho^I) \leq \epsilon'$ implies $\|\rho^E - \rho^I\|_{\text{TV}} \leq 2\sqrt{\epsilon'}$. This means that GAIL indirectly minimizes the total variation distance. Second, we have $\|\rho^E - \rho^I\|_{\text{TV}} \geq D_{\text{JS}}(\rho^E, \rho^I)$. This means that our algorithm, which minimizes Total Variation, also minimizes the GAIL objective. Both of these properties imply that our reduction converges to a fixpoint with similar properties as GAIL's. This has huge practical significance because our algorithm does not need to train a discriminator.

Another adversarial algorithm, InfoGAIL (Li et al., 2017) minimizes the 1-Wasserstein divergence, which obeys $E_{\rho^E}[R_L] - E_{\rho^I}[R_L] \leq D_{\text{W}^1}(\rho^E, \rho^I)$ for any reward function R_L Lipschitz in the L_1 norm. This is similar to the property defined by equation 3, implied by our Proposition 1, except we guarantee closeness of expected per-step reward for any bounded reward function as opposed to Lipschitz-continuous functions.

Random Expert Distillation Wang et al. (2019) propose an algorithm which, for finite MDPs, is the same as ours. However, they do not justify the properties of the imitation policy formally. Our work can be thought of as complementary. While Wang et al. (2019) conducted an empirical evaluation using sophisticated support estimators, we fill in the missing theory. Moreover, while our empirical evaluation is smaller in scope, we attempt to be more complete, demonstrating convergence in cases where only partial trajectories are given.

The SQIL heuristic SQIL (Reddy et al., 2020) is close to our work in that it proposes a similar algorithm. At any given time, SQIL performs off-policy reinforcement learning on a dataset sampled from the mixture distribution $\frac{1}{2}(\rho^E(s,a) + \rho_{\text{RL}}(s,a))$, where $\rho^E(s,a)$ is the distribution of data under the expert. Since the rewards are one for the data sampled from ρ^E and zero for state-action pairs sampled from ρ_{RL}, the expected reward obtained at a state-action pair (s,a) is given by $\frac{\rho^E(s,a)}{\rho^E(s,a)+\rho_{\text{RL}}(s,a)}$, which is non-stationary and varies between zero and one. The benefit of SQIL is that it does not require a support estimate. However, while SQIL has demonstrated good empirical performance, it does not come with a theoretical guarantee of any kind, making it hard to deploy in settings where we need a theoretical certificate of policy quality.

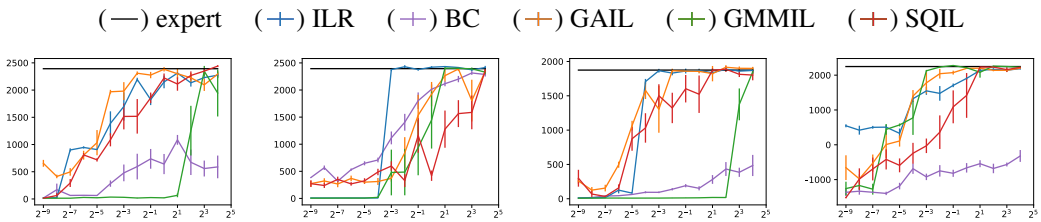
Expert Feedback Loop IL algorithms such as SMILe (Ross & Bagnell, 2010) and DAGGER (Ross et al., 2011) assume the ability to query the expert for more data. While this makes it easier to reproduce expert behavior, the ability to execute queries is not always available in realistic scenarios. Our reduction is one-off, and does not need to execute expert queries to obtain more data.

6 EXPERIMENTS

In this section, we empirically investigate the performance of our reduction on continuous control tasks. Because Wang et al. (2019) have already conducted an extensive evaluation of various ways of estimating the support for the expert distribution, we do not attempt this here. Instead, we focus on two aspects: the amount of expert data needed to achieve good imitation quality and the relationship between intrinsic and extrinsic return.

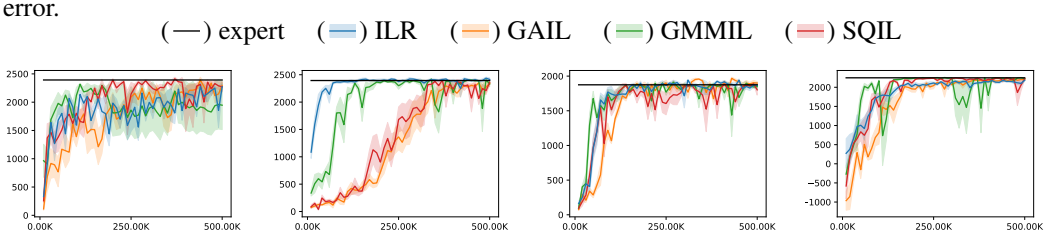
Experimental Setup We use the Hopper, Ant, Walker and HalfCheetah continuous control environments from the `PyBullet` gym suite (Ellenberger, 2018–2019). The expert policy is obtained by training a SAC agent for 2 million steps. We compare 5 methods. ILR denotes our imitation learning reduction, where we defined the reward as $R'_I(s,a) = 1 - \min_{(s',a')\in D} d_{L_2}((s,a),(s',a'))^2$, adapting the discrete reduction to a continuous state-action space. BC denotes a behavioral cloning policy, obtained after training on the expert dataset for 500000 steps. GAIL (Ho & Ermon, 2016) denotes the Generative Adversarial Imitation Learning algorithm. GMMIL (Kim & Park, 2018) denotes a variant of adversarial IL minimizing the MMD divergence. SQIL (Reddy et al., 2020) denotes the SQIL heuristic. In order to ensure a fair comparison among algorithms, we re-implemented all of them using the same basis RL algorithm, SAC (Haarnoja et al., 2018). All imitation learning agents were run for 500000 environment interactions. We repeated our experiments using 5 different random seeds. The remaining details about the implementation and hyperparameters are provided in Appendix A.

Performance as Function of Quantity of Expert Data The plot in Figure 1 shows the performance of various imitation learners as a function of how much expert data is available. The amount of data is measured in episodes, where fractional numbers mean that state-action tuples are taken from the beginning of the episode. Confidence bars represent one standard error. All of the methods except BC work well when given a large quantity of expert data (16 episodes, on the right hand side of the plot). However, for smaller sizes of the expert dataset, the performance of various methods deteriorates at different rates. In particular, GMMIL needs more expert data than other methods on Hopper and Walker, while on Ant ILR achieves top performance with less data than other methods. Across all the environments, ILR is competitive with the best of the other methods. We claim that our method is preferable since it is the simplest one to implement, is the least computationally intensive and does not introduce new hyperparameters beyond those of the RL solver.



(a) Hopper (b) Ant (c) Walker (d) HalfCheetah

Figure 1: Extrinsic return of imitation learners as a function of the number of episodes of demonstration data. For comparison, we also show expert performance. Confidence bars denote one standard error.



(a) Hopper (b) Ant (c) Walker (d) HalfCheetah

Figure 2: Extrinsic return of imitation learner as a function of the number of environment interactions. Imitation learners were trained on 16 expert trajectories. Shaded area denotes one standard error.

Performance as Function of Quantity of Environment Interactions In Figure 2, we examine the progress of the various agents as a function of environment interactions for a dataset containing 16 expert episodes. ILR shows improved sample efficiency on Ant, while behaving similarly to other algorithms on Hopper, Walker and HalfCheetah. Again, we are led to conclude that ILR is preferable to to the other algorithms because of its simplicity. We provide analogous plots for different sizes of the expert dataset in Appendix B.

Intrinsic and Extrinsic Returns are Related in Practice Our reasoning in Section 4.3 shows that achieving a large intrinsic return serves as a performance certificate, guaranteeing a certain level of extrinsic return. We conducted an experiment to determine if this property still holds for our continuous-state relaxation. Figure 3 shows the relationship between the intrinsic return (on the horizontal axis) and the extrinsic return (on the vertical axis), during the first 50K steps of training using the Hopper environment. Each point on the plot corresponds to evaluating the policy 100 times and averaging the results. The plot confirms that there is a clear, close-to-linear, relationship between the intrinsic and extrinsic return.

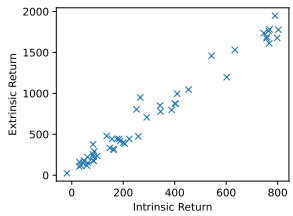


Figure 3: Intrinsic vs Extrinsic returns

7 Conclusions

We have shown that, for deterministic experts, imitation learning can be performed with a single invocation of an RL solver. We have derived a bound guaranteeing the performance of the obtained policy and relating the reduction to adversarial imitation learning algorithms. Finally, we have evaluated the proposed reduction on a family of continuous control tasks, showing that it achieves competitive performance while being comparatively simple to implement.

8 Ethics Statement

Since our most important contribution is a bound stating that the imitation learner closely mimics the expert, the ethical implications of actions taken by our algorithm crucially depend on the provided demonstrations. The most direct avenue for misuse is for malicious actors to intentionally demonstrate

unethical behavior. Second, assuming the demonstrations are provided in good faith, there is a risk of excessive reliance on the provided bound. Specifically, it is still possible that our algorithm fails to recover expert behavior in situations where assumptions needed by our proof are not met, for example if the expert policy is stochastic. In certain settings, the bounds can also turn out to be very loose, for example when the expert policy takes long to mix. This means that one should not deploy our reduction in safety-critical environments without validating the assumptions first. However, the proposed reduction also has positive effects. Specifically, we provide the benefits of adversarial imitation learning, but without having to train the discriminator. This means that training is cheaper, making imitation learning more affordable and research on imitation learning more democratic.

9 Reproducibility Statement

Our main contribution is Proposition 1, for which we provided a complete proof. The only results we rely on are generic properties of the Total Variation distance and of Markov chain mixing, which we recall in our notation in Appendices C and D, together with citations of references where they are derived. Our experimental setup is described in detail in appendix A. Moreover, we make the source code available to reviewers.

References

Pieter Abbeel and Andrew Y Ng. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the twenty-first international conference on Machine learning*, pp. 1, 2004.

Christopher G Atkeson and Stefan Schaal. Robot learning from demonstration. In *ICML*, volume 97, pp. 12–20. Citeseer, 1997.

Benjamin Ellenberger. Pybullet gymperium. `https://github.com/benelot/pybullet-gym`, 2018–2019.

Eugene A Feinberg and Adam Shwartz. *Handbook of Markov decision processes: methods and applications*, volume 40. Springer Science & Business Media, 2012.

Chelsea Finn, Sergey Levine, and Pieter Abbeel. Guided cost learning: Deep inverse optimal control via policy optimization. In *International Conference on Machine Learning*, pp. 49–58, 2016.

Seyed Kamyar Seyed Ghasemipour, Richard Zemel, and Shixiang Gu. A divergence minimization perspective on imitation learning methods. In *Conference on Robot Learning*, pp. 1259–1277. PMLR, 2020.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pp. 1861–1870. PMLR, 2018.

Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. In *Advances in neural information processing systems*, pp. 4565–4573, 2016.

Borja Ibarz, Jan Leike, Tobias Pohlen, Geoffrey Irving, Shane Legg, and Dario Amodei. Reward learning from human preferences and demonstrations in atari. In *Advances in neural information processing systems*, pp. 8011–8023, 2018.

Kee-Eung Kim and Hyun Soo Park. Imitation learning via kernel mean embedding. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

David A Levin, Yuval Peres, and EL Wilmer. Markov chains and mixing times, vol. 107. *American Mathematical Soc*, 799, 2017.

Yunzhu Li, Jiaming Song, and Stefano Ermon. Infogail: Interpretable imitation learning from visual demonstrations. In *Advances in Neural Information Processing Systems*, pp. 3812–3822, 2017.

Andrew Y Ng, Stuart J Russell, et al. Algorithms for inverse reinforcement learning. In *Icml*, volume 1, pp. 2, 2000.

Daniel Paulin et al. Concentration inequalities for markov chains by marton couplings and spectral methods. *Electronic Journal of Probability*, 20, 2015.

Martin L Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

Siddharth Reddy, Anca D. Dragan, and Sergey Levine. SQIL: Imitation learning via reinforcement learning with sparse rewards. In *International Conference on Learning Representations*, 2020. URL `https://openreview.net/forum?id=S1xKd24twB`.

Stéphane Ross and Drew Bagnell. Efficient reductions for imitation learning. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 661–668, 2010.

Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 627–635, 2011.

Wen Sun, Anirudh Vemula, Byron Boots, and Drew Bagnell. Provably efficient imitation learning from observation alone. In *International Conference on Machine Learning*, pp. 6036–6045, 2019.

Ruohan Wang, Carlo Ciliberto, Pierluigi Vito Amadori, and Yiannis Demiris. Random expert distillation: Imitation learning via expert policy support estimation. In *International Conference on Machine Learning*, pp. 6536–6544. PMLR, 2019.

Tianhao Zhang, Zoe McCarthy, Owen Jow, Dennis Lee, Xi Chen, Ken Goldberg, and Pieter Abbeel. Deep imitation learning for complex manipulation tasks from virtual reality teleoperation. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1–8. IEEE, 2018.

A Details of Experimental Setup

To ensure a fair comparison, we implemented all imitation learners (ILR, GAIL, GMMIL and SQIL) using the same base reinforcement learning solver (SAC). The implementation of SAC we used as a basis came from the `d4rl-pybullet` repository[1]. The hyperparameters of SAC are given in the table below.

hyperparameter	**value**
actor optimizer	Adam
actor learning rate	3e-4
critic optimizer	Adam
critic learning rate	3e-4
batch size	100
update rate for target network (tau)	0.005
γ	0.99

The policy network and the critic network have two fully connected middle layers with 256 neurons each followed by ReLU non-linearities. The entropy term was not used when performing the actor update after we found that including it had no effect on performance. Instead, a fixed Gaussian noise of 0.01 was used for exploration.

For our behavioral cloning baseline, we performed supervised learning for 500K steps, using the Adam optimizer with learning rate 3e-4. We used the same policy network architecture as the SAC policy network.

ILR does not introduce any additional hyperparameters.

GAIL has the following additional hyperparameters.

[1] https://github.com/takuseno/d4rl-pybullet

hyperparameter	**value**
discriminator optimizer	Adam
discriminator learning rate	3e-4
discriminator batch size	100
discriminator update frequency	32
discriminator training iterations	500

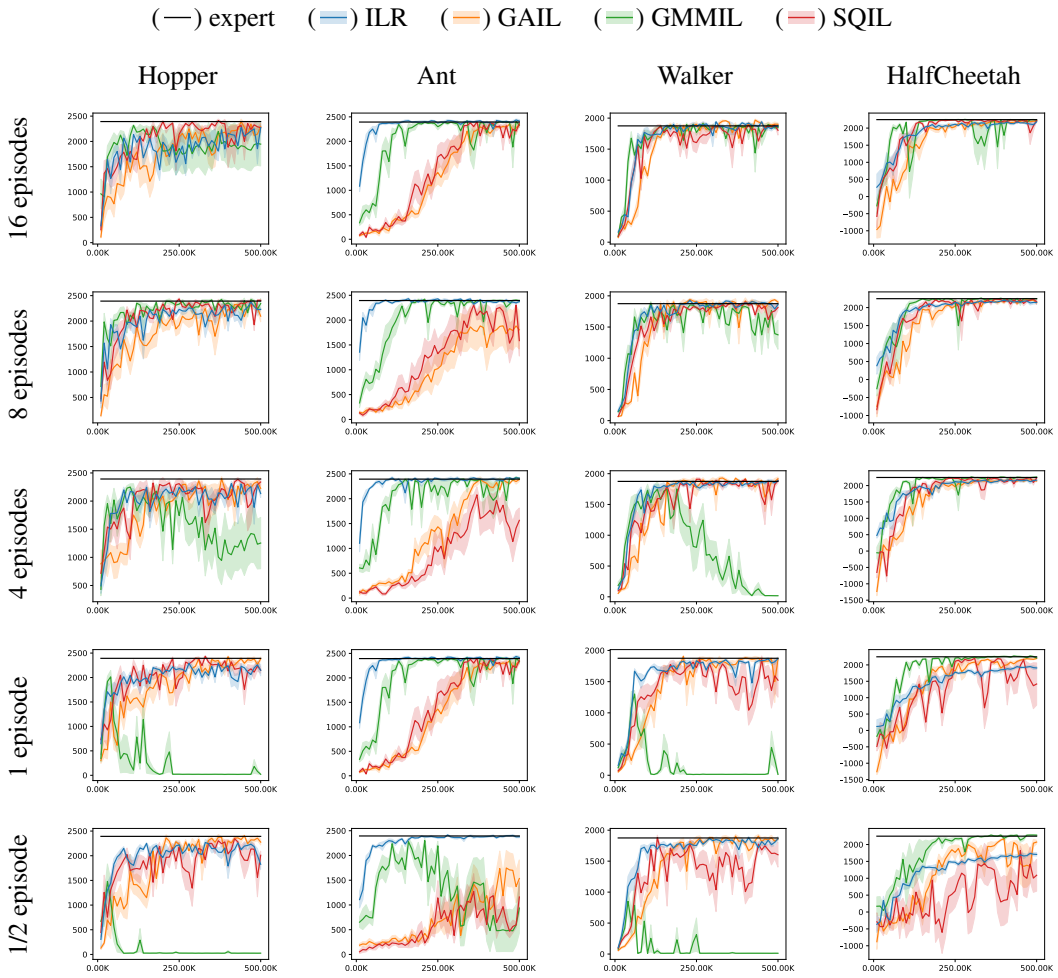
The last two hyperparameters mean that the discriminator network in GAIL is updated every 32 simulation steps, using 500 batches of data.

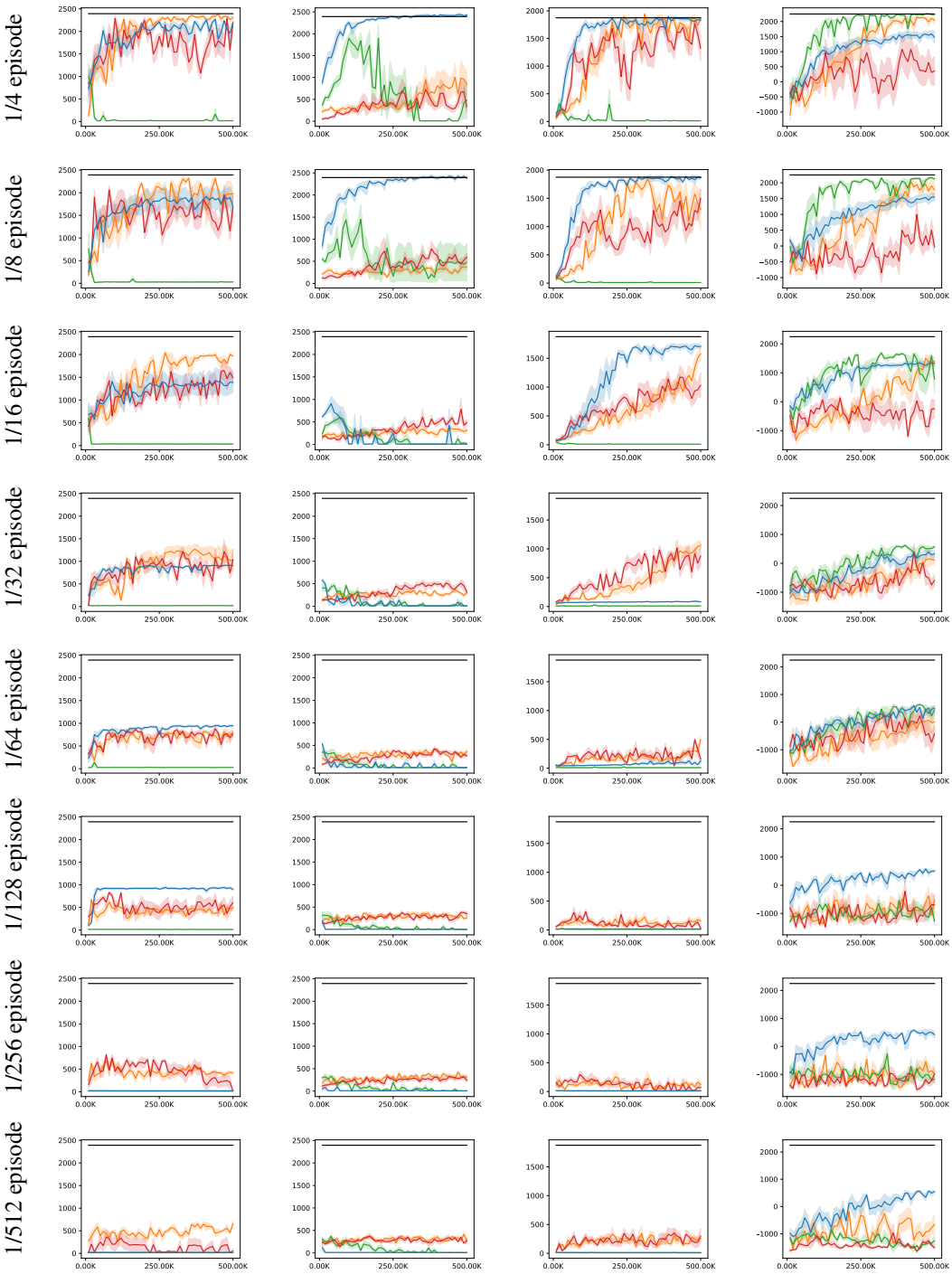
A direct implementation of GMMIL requires looping over the whole replay buffer each time a reward is computed, which makes the algorithm very slow for longer runs. To make it tractable, we we only computed the kernel for 8 points closest to the given state-action pair in the replay buffer and for 8 points in the expert dataset. The data structure used to find the nearest 8 points was updated every 32 steps. Kernel hyper-parameters were automatically estimated using the median heuristic (Kim & Park, 2018) each time the algorithm was run.

SQIL uses a batch size of 200 (100 for state-action pairs from the expert and 100 for state-action pairs coming from the replay buffer). It does not have additional hyperparameters.

B PLOTS MEASURING SAMPLE EFFICIENCY

In this section, we include plots showing the efficiency of all algorithms measured in terms of environment interactions, for different quantities of available expert data. The plots can be thought as an extension of Figure 1 in the main paper to what is happening during learning, not just at the end.





C TOTAL VARIATION

Consider probability distributions defined on a discrete set X. The total variation distance between distributions is defined as

$$\|\rho_1 - \rho_2\|_{\text{TV}} = \sup_{M \subseteq X} |\rho_1(M) - \rho_2(M)|. \tag{26}$$

In our application, the set X is either the set of MDP states S or the set of state state-action pairs $S \times A$. Below, we restate in our notation a standard result about the total variation distance.

Lemma 5. *If for any vector $v \in [0,1]^{|X|}$ it holds that $E_{\rho_1}[v] - E_{\rho_2}[v] \leq \epsilon$, then we have $\|\rho_1 - \rho_2\|_{TV} \leq \epsilon$.*

Proof. By setting $v' = 1 - v$, it follows that $E_{\rho_2}[v] - E_{\rho_1}[v] \leq \epsilon$ for all v. Therefore $|E_{\rho_1}[v] - E_{\rho_2}[v]| \leq \epsilon$ for all v. For any $M \subseteq X$ on the right-hand side of equation 26, we can instantiate $v(x) = \mathbb{1}(x \in M)$, completing the proof. □

Lemma 6. *If $\|\rho_1 - \rho_2\|_{TV} \leq \epsilon$, then we have $|E_{\rho_1}[v] - E_{\rho_2}[v]| \leq 2\epsilon$ for any vector $v \in [0,1]^{|X|}$.*

Proof. We have

$$|E_{\rho_1}[v] - E_{\rho_2}[v]| = |\textstyle\sum_x (\rho_1(x) - \rho_2(x))v(x)| \leq |\sum_x (\rho_1(x) - \rho_2(x))| \leq \|\rho_1 - \rho_2\|_1.$$

Here, the first inequality follows because elements of v are in the interval $[0,1]$. The statement of the lemma follows from the property $\|\rho_1 - \rho_2\|_{\text{TV}} = \frac{1}{2}\|\rho_1 - \rho_2\|_1$. □

Lemmas 5 and 6 are important because they connect two desiderata defined in equations 3 and 4 of the main paper, which one might pose for imitation learners. Specifically, attaining expected per-step reward close to the expert's is the same (up to a multiplicative constant) as closeness in total variation between the state-action distribution of the expert and the imitation learner.

D Background on Markov Chains

By standard properties of Markov chains, Assumption 2 in Section 3 implies exponential mixing, formalized in the Lemma below.

Lemma 7. *Denote with P_E the Markov chain as induced by the expert policy. There are constants $C > 0$ and $\alpha \in (0,1)$ such that $\max_s' \|\mathbb{1}(s')^\top P_E^t - \rho_S^E\|_{TV} \leq C\alpha^t$. Moreover, defining $\tilde{\rho}_t^{s'}(s,a) = (\mathbb{1}(s')^\top P_E^t)(s)\pi_E(a|s)$, we have $\max_{s'} \|\tilde{\rho}_t^{s'} - \rho^E\|_{TV} \leq C\alpha^t$. The mixing time of the chain τ_{mix}, defined as the smallest t so that $\max_{s'} \|\mathbb{1}(s')^\top P_E^t - \rho_S^E\|_{TV} \leq \frac{1}{4}$ is bounded as $\tau_{mix} \leq \log_\alpha C + \log_\alpha(4)$.*

Proof. For the statement about distributions over states, see Theorem 4.9 in the book by Levin et al. (2017). For the statement about distributions over state-action pairs, we use the fact that the expert policy is deterministic so that $\|\tilde{\rho}_t^{s'} - \rho^E\|_{\text{TV}} = \|\mathbb{1}(s')^\top P_E^t - \rho_S^E\|_{\text{TV}}$. □

In the following Lemma, we recall McDiarmid's inequality for Markov chains.

Lemma 8. *Consider a Markov Chain $s_1, s_2, \ldots, s_N$. For any coefficients c_i and a function f so that $f_s(s_1, \ldots, s_N) - f_s(s_1', \ldots, s_N') \leq \sum_{i=1}^N c_i \mathbb{1}(s_i \neq s_i')$ for all $s_1, \ldots, s_N, s_1', \ldots, s_N'$, we have*

$$P(|f(s_1, s_2, \ldots, s_N) - Ef(s_1, s_2, \ldots, s_N)| \leq \epsilon) \geq 1 - 2\exp\left\{-\frac{\epsilon^2}{4.5\tau_{mix}\|c\|_2^2}\right\},$$

where τ_{mix} is the mixing time of the chain.

Proof. Instantiate Corollary 2.11 of Paulin et al. (2015), where the constant 4.5 appears by plugging in equation (2.9) of Paulin et al. (2015). □