DISAGREEMENT-REGULARIZED IMITATION LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

We present a simple and effective algorithm designed to address the covariate shift problem in imitation learning. It operates by training an ensemble of policies on the expert demonstration data, and using the variance of their predictions as a cost which is minimized with RL together with a supervised behavioral cloning cost. Unlike adversarial imitation methods, it uses a fixed reward function which is easy to optimize. We prove a regret bound for the algorithm in the tabular setting which is linear in the time horizon multiplied by a coefficient which we show to be low for certain problems in which behavioral cloning fails. We evaluate our algorithm empirically across multiple pixel-based Atari environments and continuous control tasks, and show that it matches or significantly outperforms behavioral cloning and generative adversarial imitation learning.

1 INTRODUCTION

Training artificial agents to perform complex tasks is essential for many applications in robotics, video games and dialogue. If success on the task can be accurately described using a reward or cost function, reinforcement learning (RL) methods offer an approach to learning policies which has been shown to be successful in a wide variety of applications (Mnih et al., 2015; 2016; Lillicrap et al., 2016; Hessel et al., 2018) However, in other cases the desired behavior may only be roughly specified and it is unclear how to design a reward function to characterize it. For example, training a video game agent to adopt more human-like behavior using RL would require designing a reward function which characterizes behaviors as more or less human-like, which is difficult.

Imitation learning (IL) offers an elegant approach whereby agents are trained to mimic the demonstrations of an expert rather than optimizing a reward function. Its simplest form consists of training a policy to predict the expert's actions from states in the demonstration data using supervised learning. While appealingly simple, this approach suffers from the fact that the distribution over states observed at execution time can differ from the distribution observed during training. Minor errors which initially produce small deviations from the expert trajectories become magnified as the policy encounters states further and further from its training distribution. This phenomenon, initially noted in the early work of (Pomerleau, 1989), was formalized in the work of (Ross & Bagnell, 2010) who proved a quadratic $\mathcal{O}(\epsilon T^2)$ bound on the regret and showed that this bound is tight. The subsequent work of (Ross et al., 2011) showed that if the policy is allowed to further interact with the environment and make queries to the expert policy, it is possible to obtain a *linear* bound on the regret. However, the ability to query an expert can often be a strong assumption.

In this work, we propose a new and simple algorithm called DRIL (Disagreement-Regularized Imitation Learning) to address the covariate shift problem in imitation learning, in the setting where the agent is allowed to interact with its environment. Importantly, the algorithm does not require any additional interaction with the expert. It operates by training an ensemble of policies on the demonstration data, and using the disagreement in their predictions as a cost which is optimized through RL together with a supervised behavioral cloning cost. The motivation is that the policies in the ensemble will tend to agree on the set of states covered by the expert, leading to low cost, but are more likely to disagree on states not covered by the expert, leading to high cost. The RL cost thus pushes the agent back towards the distribution of the expert, while the supervised cost ensures that it mimics the expert within the expert's distribution.

Our theoretical results show that, subject to realizability and optimization oracle assumptions, our algorithm obtains a $\mathcal{O}(\epsilon\kappa T)$ regret bound for tabular MDPs, where κ is a measure which quantifies a tradeoff between the concentration of the demonstration data and the diversity of the ensemble outside the demonstration data. We evaluate DRIL empirically across multiple pixel-based Atari environments and continuous control tasks, and show that it matches or significantly outperforms behavioral cloning and generative adversarial imitation learning, often recovering expert performance with only a few trajectories.

2 Preliminaries

Denote by $\mathcal{S}$ the state space, $\mathcal{A}$ the action space, and Π the class of policies the learner is considering. Let T denote the task horizon and $\pi^\star$ the expert policy whose behavior the learner is trying to mimic. For any policy π, let d_π denote the distribution over states induced by following π. Denote $C(s, a)$ the expected immediate cost of performing action a in state s, which we assume is bounded in $[0, 1]$. In the imitation learning setting, we do not necessarily know the true costs $C(s, a)$, instead we observe expert demonstrations. Our goal is to find a policy π which minimizes an observed surrogate loss ℓ between its actions and the actions of the expert under the induced distribution of states, i.e.

$$\hat{\pi} = \arg\min \mathbb{E}_{s \sim d_\pi}[\ell(\pi(s), \pi^\star(s))] \tag{1}$$

For the following, we will assume ℓ is the total variation distance (denoted by $\|\cdot\|$), which is an upper bound on the $0-1$ loss. Our goal is thus to minimize the following quantity, which represents the distance between the actions taken by our policy π and the expert policy $\pi^\star$:

$$J_{\text{exp}}(\pi) = \mathbb{E}_{s \sim d_\pi}\Big[\|\pi(\cdot|s) - \pi^\star(\cdot|s)\|\Big] \tag{2}$$

The following result shows that if ℓ represents an upper bound on the $0-1$ loss and C satisfies certain smoothness conditions, then minimizing this loss within ϵ translates into an $\mathcal{O}(\epsilon T)$ regret bound on the task cost $J_{\text{C}}(\pi) = \mathbb{E}_{s,a\sim\pi}[C(s, a)]$.

Theorem 1. *(Ross et al., 2011) Let π be such that $J_{\text{exp}}(\pi) = \epsilon$, and $Q^{\pi^\star}_{T-t+1}(s, a) - Q^{\pi^\star}_{T-t+1}(s, \pi^\star) \leq u$ for all $a \in \mathcal{A}, t \in \{1, 2, ..., T\}, d^t_\pi(s) > 0$. Then $J_{\text{C}}(\pi) \leq J_{\text{C}}(\pi^\star) + uT\epsilon$.*

Unfortunately, it is often not possible to optimize J_{exp} directly, since it requires evaluating the expert policy on the states induced by following the *current* policy. The supervised behavioral cloning cost J_{BC}, which is computed on states induced by the expert, is often used instead:

$$J_{\text{BC}}(\pi) = \mathbb{E}_{s \sim d_{\pi^\star}}[\|\pi^\star(\cdot|s) - \pi(\cdot|s)\|] \tag{3}$$

Minimizing this loss within ϵ yields a quadratic regret bound on regret:

Theorem 2. *(Ross & Bagnell, 2010) Let $J_{\text{BC}}(\pi) = \epsilon$, then $J_{\text{C}}(\pi) \leq J_{\text{C}}(\pi^\star) + T^2\epsilon$.*

Furthermore, this bound is tight: as we will discuss later, there exist simple problems which match the worst-case lower bound.

3 Algorithm

Our algorithm is motivated by two criteria: i) the policy should perform similarly to the expert on the expert's data distribution, and ii) the policy should move towards the expert's data distribution if it is away from it. These two criteria are addressed by combining two losses: a standard behavior cloning loss, and an additional loss which represents the variance over actions induced by sampling different policies from the posterior given the demonstration data $\mathcal{D}$. We call this the uncertainty cost, which is defined as:

Algorithm 1 Disagreement-Regularized Imitation Learning (DRIL)

```
Input: Expert demonstration data $\mathcal{D} = \{(s_i, a_i)\}_{i=1}^N$
Initialize policy $\pi$ and policy ensemble $E = \{\pi_e\}$
for $e = 1, E$ do
    Sample $\mathcal{D}_e \sim \mathcal{D}$ with replacement, with $|\mathcal{D}_e| = |\mathcal{D}|$.
    Train $\pi_e$ to minimize $J_{\text{BC}}(\pi_e)$ on $\mathcal{D}_e$ to convergence.
end for
for $i = 1, ...$ do
    Perform one gradient update to minimize $J_{\text{BC}}(\pi)$ using a minibatch from $\mathcal{D}$.
    Perform one step of policy gradient to minimize $C_{\text{U}}^{\text{clip}}(s, a)$.
end for
```

$$C_{\text{U}}(s, a) = \text{Var}_{\pi \sim p(\pi|\mathcal{D})}(\pi(a|s))$$

The motivation is that the variance over plausible policies is high outside the expert's distribution, since the data is sparse, but it is low inside the expert's distribution, since the data there is dense. Minimizing this cost encourages the policy to return to regions of dense coverage by the expert. Intuitively, this is what we would expect the expert policy $\pi^\star$ to do as well. The total cost which the algorithm optimizes is given by:

$$J_{\text{alg}}(\pi) = \underbrace{\mathbb{E}_{s \sim d_{\pi^\star}}[||\pi^\star(\cdot|s) - \pi(\cdot|s)||]}_{J_{\text{BC}}(\pi)} + \underbrace{\mathbb{E}_{s \sim d_\pi, a \sim \pi(\cdot|s)}\Big[C_{\text{U}}(s, a)\Big]}_{J_{\text{U}}(\pi)}$$

The first term is a behavior cloning loss and is computed over states generated by the expert policy, of which the demonstration data $\mathcal{D}$ is a representative sample. The second term is computed over the distribution of states generated by the current policy and can be optimized using policy gradient. More precisely, we approximate the posterior $p(\pi|\mathcal{D})$ by training an ensemble $E = \{\pi_e\}_{e=1}^{|E|}$ of models on different bootstrap samples of the demonstration data. Note that the demonstration data is fixed, and this ensemble can be trained once offline. We then interleave the supervised behavioral cloning updates and the policy gradient updates which minimize the variance of the posterior. The full algorithm is shown in Algorithm 1.

In practice, for the supervised loss we optimize the KL divergence between the actions predicted by the policy and the expert actions, which is an upper bound on the total variation distance. We also found it helpful to use a clipped uncertainty cost:

$$C_{\text{U}}^{\text{clip}}(s, a) = \begin{cases} +1 & \text{if } C_U(s, a) \leq q \\ -1 & \text{else} \end{cases}$$

where the threshold q is a top quantile of the raw uncertainty costs computed over the demonstration data. The threshold q defines a normal range of uncertainty based on the demonstration data, and values outside of this range incur a negative cost.

The RL cost can be optimized using any policy gradient method, in our experiments we used an advantage actor-critic algorithm (Mnih et al., 2016). We note that model-based methods could in principle be used as well if sample efficiency is a constraint.

4 Analysis

4.1 Coverage Coefficient

We now analyze DRIL for tabular MDPs. We will show that, subject to assumptions that the policy class contains an optimal policy and that we are able to optimize costs within ϵ of their global minimum, our algorithm obtains a regret bound which is linear κT, where κ which is quantity specific to the environment and $d_\pi^\star$. Intuitively, κ represents a tradeoff between how concentrated the demonstration data is and how high the variance of the posterior is outside the expert distribution.

Assumption 1. *(Realizability)* $\pi^\star \in \Pi$

Assumption 2. *(Optimization Oracle) For any given cost function J, our minimization procedure returns a policy $\hat{\pi} \in \Pi$ such that $J(\hat{\pi}) \leq \arg\min_{\pi \in \Pi} J(\pi) + \epsilon$*

The motivation behind our algorithm is that the policies in the ensemble agree inside the expert's distribution and disagree outside of it. This defines a reward function which pushes the learner back towards the expert's distribution if it strays away. However, what constitutes inside and outside the distribution, or sufficient agreement or disagreement, is ambiguous. Below we define quantities which makes these ideas precise.

Definition 1. *For any set $\mathcal{U} \subseteq \mathcal{S}$, define the maximum probability ratio between the state distributions induced by the expert policy and by policies in the policy class inside of $\mathcal{U}$ as $\alpha(\mathcal{U}) = \max_{\pi \in \Pi} \sum_{s \in \mathcal{U}} \frac{d_\pi(s)}{d_\pi^\star(s)}$.*

Note that $\alpha(\mathcal{U}) \leq \frac{1}{\min_{s \in \mathcal{U}} d_\pi^\star(s)}$. For a set $\mathcal{U}$, $\alpha(\mathcal{U})$ will be low if the expert distribution has high density inside of $\mathcal{U}$, and the states in $\mathcal{U}$ is reachable by policies in the policy class.

Definition 2. *Define the minimum variance of the posterior outside of $\mathcal{U}$ as $\beta(\mathcal{U}) = \min_{s \notin \mathcal{U}, a \in \mathcal{A}} \mathrm{Var}_{\pi \sim p(\pi|\mathcal{D})}[\pi(a|s)]$.*

We now define the κ coefficient as the minimum ratio of these two quantities over all possible subsets of $\mathcal{S}$.

Definition 3. *We define $\kappa(\mathcal{U}) = \frac{\alpha(\mathcal{U})}{\beta(\mathcal{U})}$, and $\kappa = \min_{\mathcal{U} \subseteq \mathcal{S}} \kappa(\mathcal{U})$.*

We can view minimizing $\kappa(\mathcal{U})$ over different $\mathcal{U} \subseteq \mathcal{S}$ as minimizing a tradeoff between coverage by the expert policy inside of $\mathcal{U}$, and variance of the posterior outside of $\mathcal{U}$. Note that by making $\mathcal{U}$ very small, it may be easy to make $\alpha(\mathcal{U})$ small, but doing so may also make $\beta(\mathcal{U})$ small and $\kappa(\mathcal{U})$ large. Conversely, making $\mathcal{U}$ large may make $\beta(\mathcal{U})$ large but may also make $\alpha(\mathcal{U})$ large as a result.

4.2 Regret Bound

We now establish a relationship between the κ coefficient just defined, the cost our algorithm optimizes, and J_{exp} defined in Equation (2) which we would ideally like to minimize and which translates into a regret bound. All proofs can be found in Appendix A.

Lemma 1. *For any $\pi \in \Pi$, we have $J_{\mathrm{exp}}(\pi) \leq \kappa J_{\mathrm{alg}}(\pi)$*

This result shows that if κ is not too large, and we are able to make our cost function $J_{\mathrm{alg}}(\pi)$ small, then we can ensure $J_{\mathrm{exp}}(\pi)$ is also be small. This result is only useful if our cost function can indeed achieve a small minimum. The next lemma shows that this is the case.

Lemma 2. $\min_{\pi \in \Pi} J_{\mathrm{alg}}(\pi) \leq 2\epsilon$

Here ϵ is the threshold specified in Assumption 2. Combining these two lemmas with the previous result of Ross et al. (2011), we get a regret bound which is linear in κT.

Theorem 3. *Let $\hat{\pi}$ be the result of minimizing J_{alg} using our optimization oracle, and assume that $Q_{T-t+1}^{\pi^\star}(s,a) - Q_{T-t+1}^{\pi^\star}(s,\pi^\star) \leq u$ for all $a \in \mathcal{A}, t \in \{1, 2, ..., T\}, d_\pi^t(s) > 0$. Then $\hat{\pi}$ satisfies $J_{\mathrm{C}}(\hat{\pi}) \leq J_{\mathrm{C}}(\pi^\star) + 3u\kappa\epsilon T$.*

Our bound is an improvement over that of behavior cloning if κ is less than $\mathcal{O}(T)$. Note that DRIL does not require knowledge of κ. The quantity κ is problem-dependent and depends on the environment dynamics, the expert policy and the class of policies available to the learner. We next compute κ exactly for a problem for which behavior cloning is known to perform poorly, and show that it is independent of T.

Example 1. *Consider the tabular MDP given in (Ross et al., 2011) as an example of a problem where behavioral cloning incurs quadratic regret, shown in Figure 1. There are 3 states (s_0, s_1, s_2)*

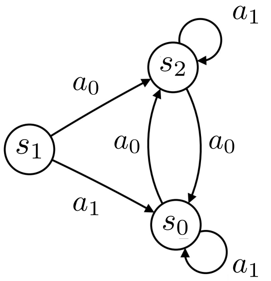


Figure 1: Example of a problem where behavioral cloning incurs quadratic regret.

and two actions (a_0, a_1). *The expert policy is given by* $\pi^\star(s_0) = a_0, \pi^\star(s_1) = a_0, \pi^\star(s_2) = a_1$. *Here* $d_\pi^\star = (0, \frac{1}{T}, \frac{T-1}{T})$. *Writing out* $\kappa(\{s_1, s_2\})$ *yields:*

$$\begin{aligned}\kappa(\{s_1, s_2\}) &= \frac{\alpha(\{s_1, s_2\})}{\beta(\{s_1, s_2\})} \\ &= \frac{\max_{\pi \in \Pi} \sum_{s \in \{s_1, s_2\}} \frac{d_\pi(s)}{d_\pi^\star(s)}}{\min_{a \in \{a_0, a_1\}} \mathrm{Var}(a|s_0)}\end{aligned}$$

For any π, $d_\pi(s_1) \leq \frac{1}{T}$ *and* $d_\pi(s_2) \leq \frac{T-1}{T}$ *due to the dynamics of the MDP, so* $\frac{d_\pi(s)}{d_\pi^\star(s)} \leq 1$ *for* $s \in \{s_0, s_1\}$. *Furthermore, since* s_0 *is never visited in the demonstration data* $\mathrm{Var}_{\pi \sim p(\pi|\mathcal{D})}(\pi(a|s_2))$ *is equal to the variance of a uniform distribution with variance 1, i.e.* $\frac{1}{12}$. *Therefore:*

$$\kappa \leq \kappa(\{s_1, s_2\}) \leq \frac{1+1}{\frac{1}{12}} = 24$$

Applying our result from Theorem 3, we see that our algorithm obtains an $\mathcal{O}(\epsilon T)$ *regret bound on this problem, in contrast to the* $\mathcal{O}(\epsilon T^2)$ *regret of behavioral cloning* [1].

5 Related Work

The idea of learning through imitation dates back at least to the work of (Pomerleau, 1989), who trained a neural network to imitate the steering actions of a human driver using images as input. The problem of covariate shift was already observed, as the author notes: “when driving for itself, the network may occasionally stray from the center of the road and so must be prepared to recover by steering the vehicle back to the center of the road”.

This issue was formalized in the work of (Ross & Bagnell, 2010), who on one hand proved an $\mathcal{O}(\epsilon T^2)$ regret bound, and on the other hand provided an example showing this bound is tight. The subsequent work (Ross et al., 2011) gave an algorithm which obtains linear regret, provided the agent can both interact with the environment, and query the expert policy. Our approach also requires environment interaction, but importantly does not require the ability to query the expert.

Imitation learning has been used within the context of modern RL to help improve sample efficiency (Hester et al., 2018) or overcome exploration (Nair et al., 2017). These settings assume the reward is known and that the policies can then be fine-tuned with reinforcement learning. In this case, covariate shift is less of an issue since it can be corrected using the reinforcement signal.

[1] Observe that a policy with $\pi(a_0|s_1) = 1 - \epsilon T, \pi(a_1|s_1) = \epsilon T, \pi(a_1|s_0) = 1$ has a behavioral cloning cost of ϵ but a regret of ϵT^2.

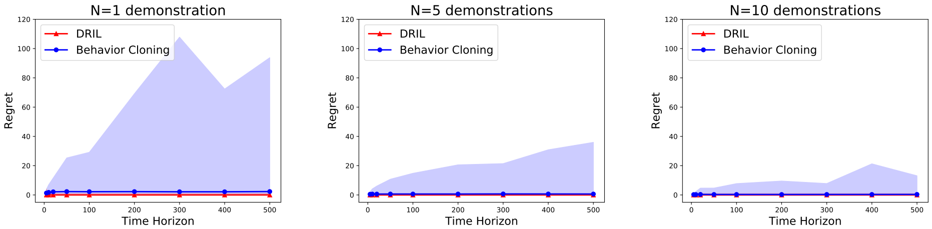


Figure 2: Results on tabular MDP from (Ross & Bagnell, 2010). Shaded region represents range between 5^{th} and 95^{th} quantiles, computed across 500 trials. Behavior cloning exhibits poor worst-case regret, whereas DRIL has low regret across all trials.

The work of (Luo et al., 2019) also proposed a method to address the covariate shift problem when learning from demonstrations when the reward is known, by conservatively extrapolating the value function outside the training distribution using negative sampling. This addresses a different setting from ours, and requires generating plausible states which are off the manifold of training data, which may be challenging when the states are high dimensional such as images. The work of Reddy et al. (2019) proposed to treat imitation learning within the Q-learning framework, setting a positive reward for all transitions inside the demonstration data and zero reward for all other transitions in the replay buffer. However, this requires carefully decaying the reward over time as the policy produces states closer to the expert's distribution. Our approach deals with a similar setting, but uses a fixed reward function.

Generative Adversarial Imitation Learning (GAIL) (Ho & Ermon, 2016) is a state-of-the-art algorithm which addresses the same setting as ours. It operates by training a discriminator network to distinguish expert states from states generated by the current policy, and the negative output of the discriminator is used as a reward signal to train the policy. The motivation is that states which are outside the training distribution will be assigned a low reward while states which are close to it will be assigned a high reward. This encourages the policy to return to the expert distribution if is strays away from it. However, the adversarial training procedure means that the reward function is changing over time, which can make the algorithm unstable or difficult to tune. In contrast, our approach uses a simple fixed reward function. We include comparisons to GAIL in our experiments.

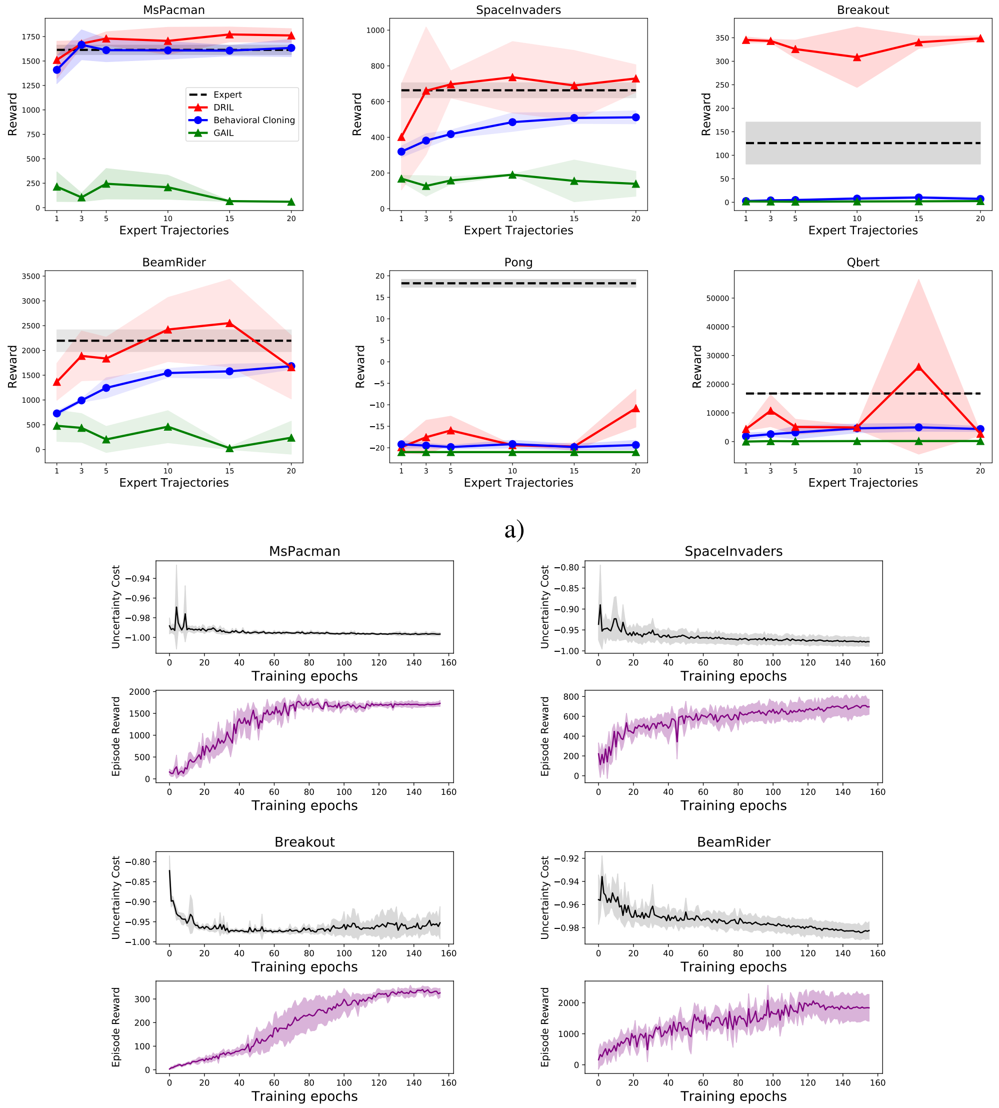
Using disagreement between models in an ensemble to represent uncertainty has recently been explored in several contexts. The works of (Shyam et al., 2018; Pathak et al., 2019) used disagreement between different dynamics models to drive exploration in the context of model-based RL. Conversely, (Henaff et al., 2019) used variance across different dropout masks to prevent policies from exploiting error in dynamics models. Ensembles have also been used to represent uncertainty over Q-values in model-free RL in order to encourage exploration (Osband et al., 2016). Here, we use disagreement between different policies trained on the expert data to address covariate shift in the context of imitation learning.

6 Experiments

6.1 Tabular MDPs

As a first experiment, we applied DRIL to the tabular MDP of (Ross & Bagnell, 2010) shown in Figure 1. We computed the posterior over the policy parameters given the demonstration data using a Dirichlet distribution with parameters determined by visitation counts. For behavior cloning, we sampled a single policy from this posterior. For our method, we sampled 5 policies and used their negative variance to define an additional reward function. We combined this with a reward which was the probability density function of a given state-action pair under the posterior distribution, which corresponds to the supervised learning loss, and used tabular Q-learning to optimize the sum of these two reward functions. This experiment was repeated 500 times for time horizon lengths up to 500 and $N = 1, 5, 10$ expert demonstration trajectories.

Figure 2 shows plots of the regret over the 500 different trials across different time horizons. Although the average performance of BC improves with more expert demonstrations, it exhibits poor



a)

b)

Figure 3: Results on Atari environments. a) Final policy performance for different numbers of expert trajectories. b) Evolution of policy reward and uncertainty cost during training with $N = 5$ trajectories.

worst-case performance with some trials incurring very high regret, especially when using fewer demonstrations. Our method has low regret across all trials, which stays close to constant independantly of the time horizon, even with a single demonstration. This performance is better than that suggested by our analysis, which showed a worst-case linear bound with respect to time horizon.

6.2 Atari Environments

We next evaluated our approach on six different Atari environments. We used pretrained PPO (Schulman et al., 2017) agents from the stable baselines repository (Hill et al., 2018) to generate $N = \{1, 3, 5, 10, 15, 20\}$ expert trajectories. We compared against two other methods: standard behavioral cloning (BC) and Generative Adversarial Imitation Learning (GAIL). Results are shown in Figure 3a. DRIL outperforms behavioral cloning across most environments and numbers of demonstrations, often by a substantial margin. In the worst case its performance matches that of behavior

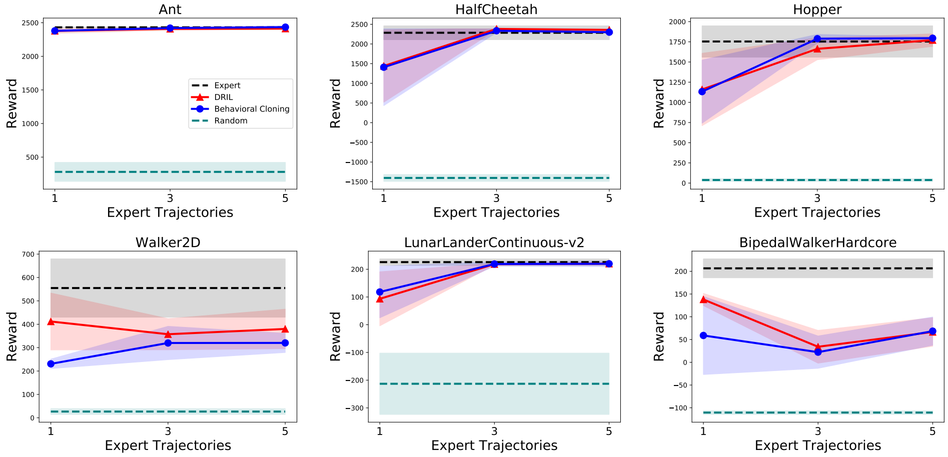


Figure 4: Results on continuous control tasks.

cloning. In many cases, our method is able to match the expert's performance using a small number of trajectories.

Figure 3b shows the evolution of the uncertainty cost and the policy reward throughout training. In all cases, the test reward improves while the uncertainty cost decreases. Interestingly, there is correspondence between the change in the uncertainty cost during training and the gap in performance between behavior cloning and DRIL. For example, in MsPacman there is both a small improvement in uncertainty cost over time and a small gap between behavior cloning and our method, whereas in Breakout there is a large improvement in uncertainty cost and a large gap between behavior cloning and our method. This suggests that the gains from our method comes from redirecting the policy back towards the expert manifold, which is manifested as a decrease in the uncertainty cost.

We were not able to obtain meaningful performance for GAIL on these domains, despite performing a hyperparameter search across learning rates for the policy and discriminator, and across different numbers of discriminator updates. We additionally experimented with clipping rewards in an effort to stabilize performance. These results are consistent with those of (Reddy et al., 2019), who also reported negative results when running GAIL on images. While improved performance might be possible with more sophisticated adversarial training techniques, we note that this contrasts with our method which uses a fixed reward function obtained through simple supervised learning.

6.3 Continuous Control

We next report results of running our method on a 6 different continuous control tasks from the PyBullet[2] and OpenAI Gym (Brockman et al., 2016) environments. We again used pretrained agents to generate expert demonstrations. Results are shown in Figure 4. In these environments we found behavior cloning to be a much stronger baseline than for the Atari environments, and in many tasks it was able to match expert performance using as little as 3 trajectories. Our method exhibits a modest improvement on Walker2D and BipedalWalkerHardcore when a single trajectory is used, and otherwise has similar performance to behavior cloning. The fact that our method does not perform worse than behavior cloning on tasks where covariate shift is likely less of an issue provides evidence of its robustness.

7 Conclusion

Addressing covariate shift has been a long-standing challenge in imitation learning. In this work, we have proposed a new method to address this problem by penalizing the disagreement between an ensemble of different policies sampled from the posterior. Importantly, our method requires

[2] https://github.com/bulletphysics/bullet3/tree/master/examples/pybullet/gym/pybullet_envs/examples

no additional labeling by an expert. Our experimental results demonstrate that DRIL can often match expert performance while using only a small number of trajectories across a wide array of tasks, ranging from tabular MDPs to pixel-based Atari games and continuous control tasks. On the theoretical side, we have shown that our algorithm can provably obtain a low regret bound for tabular problems in which the κ parameter is low.

There are multiple directions for future work. On the theoretical side, extending our analysis to continuous state spaces and characterizing the κ parameter on a larger array of problems would help to better understand the settings where our method can expect to do well. Empirically, there are many other settings in structured prediction (Daumé et al., 2009) where covariate shift is an issue and where our method could be applied. For example, in dialogue and language modeling it is common for generated text to become progressively less coherent as errors push the model off the manifold it was trained on. Our method could potentially be used to fine-tune language or translation models (Cho et al., 2014; Welleck et al., 2019) after training by applying our uncertainty-based cost function to the generated text.

References

Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym, 2016.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1724–1734, Doha, Qatar, October 2014. Association for Computational Linguistics. doi: 10.3115/v1/D14-1179. URL https://www.aclweb.org/anthology/D14-1179.

Hal Daumé, John Langford, and Daniel Marcu. Search-based structured prediction. *CoRR*, abs/0907.0786, 2009. URL http://arxiv.org/abs/0907.0786.

Mikael Henaff, Alfredo Canziani, and Yann LeCun. Model-predictive policy learning with uncertainty regularization for driving in dense traffic. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=HygQBn0cYm.

Matteo Hessel, Joseph Modayil, Hado P. van Hasselt, Tom Schaul, Georg Ostrovski, Will Dabney, Dan Horgan, Bilal Piot, Mohammad Gheshlaghi Azar, and David Silver. Rainbow: Combining improvements in deep reinforcement learning. In *AAAI*, 2018.

Todd Hester, Matej Vecerík, Olivier Pietquin, Marc Lanctot, Tom Schaul, Bilal Piot, Dan Horgan, John Quan, Andrew Sendonaris, Ian Osband, Gabriel Dulac-Arnold, John Agapiou, Joel Z. Leibo, and Audrunas Gruslys. Deep q-learning from demonstrations. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pp. 3223–3230, 2018. URL https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16976.

Ashley Hill, Antonin Raffin, Maximilian Ernestus, Adam Gleave, Rene Traore, Prafulla Dhariwal, Christopher Hesse, Oleg Klimov, Alex Nichol, Matthias Plappert, Alec Radford, John Schulman, Szymon Sidor, and Yuhuai Wu. Stable baselines. https://github.com/hill-a/stable-baselines, 2018.

Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett (eds.), *Advances in Neural Information Processing Systems 29*, pp. 4565–4573. Curran Associates, Inc., 2016.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2014. URL http://arxiv.org/abs/1412.6980. cite arxiv:1412.6980Comment: Published as a conference paper at the 3rd International Conference for Learning Representations, San Diego, 2015.

Ilya Kostrikov. Pytorch implementations of reinforcement learning algorithms. https://github.com/ikostrikov/pytorch-a2c-ppo-acktr-gail, 2018.

Timothy P. Lillicrap, Jonathan J. Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *CoRR*, abs/1509.02971, 2016.

Yuping Luo, Huazhe Xu, and Tengyu Ma. Learning self-correctable policies and value functions from demonstrations with negative sampling. *CoRR*, abs/1907.05634, 2019. URL http://arxiv.org/abs/1907.05634.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, February 2015. ISSN 00280836. URL http://dx.doi.org/10.1038/nature14236.

Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In Maria Florina Balcan and Kilian Q. Weinberger (eds.), *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pp. 1928–1937, New York, New York, USA, 20–22 Jun 2016. PMLR. URL http://proceedings.mlr.press/v48/mniha16.html.

Ashvin Nair, Bob McGrew, Marcin Andrychowicz, Wojciech Zaremba, and Pieter Abbeel. Overcoming exploration in reinforcement learning with demonstrations. *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6292–6299, 2017.

Ian Osband, Charles Blundell, Alexander Pritzel, and Benjamin Van Roy. Deep exploration via bootstrapped DQN. *CoRR*, abs/1602.04621, 2016. URL http://arxiv.org/abs/1602.04621.

Deepak Pathak, Dhiraj Gandhi, and Abhinav Gupta. Self-supervised exploration via disagreement. In *ICML*, 2019.

Dean A. Pomerleau. Alvinn: An autonomous land vehicle in a neural network. In D. S. Touretzky (ed.), *Advances in Neural Information Processing Systems 1*, pp. 305–313. Morgan-Kaufmann, 1989. URL http://papers.nips.cc/paper/95-alvinn-an-autonomous-land-vehicle-in-a-neural-network.pdf.

Siddharth Reddy, Anca D. Dragan, and Sergey Levine. SQIL: imitation learning via regularized behavioral cloning. *CoRR*, abs/1905.11108, 2019. URL http://arxiv.org/abs/1905.11108.

Stephane Ross and Drew Bagnell. Efficient reductions for imitation learning. In Yee Whye Teh and Mike Titterington (eds.), *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pp. 661–668, Chia Laguna Resort, Sardinia, Italy, 13–15 May 2010. PMLR. URL http://proceedings.mlr.press/v9/ross10a.html.

Stephane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In Geoffrey Gordon, David Dunson, and Miroslav Dudk (eds.), *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15 of *Proceedings of Machine Learning Research*, pp. 627–635, Fort Lauderdale, FL, USA, 11–13 Apr 2011. PMLR. URL http://proceedings.mlr.press/v15/ross11a.html.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *CoRR*, abs/1707.06347, 2017. URL http://arxiv.org/abs/1707.06347.

Pranav Shyam, Wojciech Jaskowski, and Faustino Gomez. Model-based active exploration. *CoRR*, abs/1810.12162, 2018.

Sean Welleck, Kianté Brantley, Hal Daumé III, and Kyunghyun Cho. Non-monotonic sequential text generation. *CoRR*, abs/1902.02192, 2019. URL http://arxiv.org/abs/1902.02192.

A Proofs

Lemma 1. *For any $\pi \in \Pi$ we have $J_{\text{exp}}(\pi) \leq \kappa J_{\text{alg}}(\pi)$*

Proof. We will first show that for any $\pi \in \Pi$ and $\mathcal{U} \subseteq \mathcal{S}$, we have $J_{\text{exp}}(\pi) \leq \frac{\alpha(\mathcal{U})}{\beta(\mathcal{U})} J_{\text{alg}}(\pi)$. We can rewrite this as:

$$\begin{aligned} J_{\text{exp}}(\pi) &= \mathbb{E}_{s \sim d_\pi}\Big[\|\pi(\cdot|s) - \pi^\star(\cdot|s)\|\Big] \\ &= \mathbb{E}_{s \sim d_\pi}\Big[\mathbb{I}(s \in \mathcal{U})\|\pi(\cdot|s) - \pi^\star(\cdot|s)\|\Big] + \mathbb{E}_{s \sim d_\pi}\Big[\mathbb{I}(s \notin \mathcal{U})\|\pi(\cdot|s) - \pi^\star(\cdot|s)\|\Big] \end{aligned}$$

We begin by bounding the first term:

$$\begin{aligned} \mathbb{E}_{s \sim d_\pi}\Big[\mathbb{I}(s \in \mathcal{U})\|\pi(\cdot|s) - \pi^\star(\cdot|s)\|\Big] &= \sum_{s \in \mathcal{U}} d_\pi(s)\|\pi(\cdot|s) - \pi^\star(\cdot|s)\| \\ &= \sum_{s \in \mathcal{U}} \frac{d_\pi(s)}{d_{\pi^\star}(s)} d_{\pi^\star}(s)\|\pi(\cdot|s) - \pi^\star(\cdot|s)\| \\ &\leq \Big(\sum_{s \in \mathcal{U}} \frac{d_\pi(s)}{d_{\pi^\star}(s)}\Big)\Big(\sum_{s \in \mathcal{U}} d_{\pi^\star}(s)\|\pi(\cdot|s) - \pi^\star(\cdot|s)\|\Big) \\ &\leq \Big(\sum_{s \in \mathcal{U}} \frac{d_\pi(s)}{d_{\pi^\star}(s)}\Big) \sum_{s \in \mathcal{U}} d_{\pi^\star}(s)\|\pi(\cdot|s) - \pi^\star(\cdot|s)\| \\ &\leq \Big(\sum_{s \in \mathcal{U}} \frac{d_\pi(s)}{d_{\pi^\star}(s)}\Big) \sum_{s} d_{\pi^\star}(s)\|\pi(\cdot|s) - \pi^\star(\cdot|s)\| \\ &= \Big(\sum_{s \in \mathcal{U}} \frac{d_\pi(s)}{d_{\pi^\star}(s)}\Big)\mathbb{E}_{s \sim d_{\pi^\star}}[\|\pi(\cdot|s) - \pi^\star(\cdot|s)\|] \\ &\leq \alpha(\mathcal{U}) J_{\text{BC}}(\pi) \end{aligned}$$

We next bound the second term:

$$\begin{aligned} \mathbb{E}_{s \sim d_\pi}\Big[\mathbb{I}(s \notin \mathcal{U})\|\pi(\cdot|s) - \pi^\star(\cdot|s)\|\Big] &\leq \mathbb{E}_{s \sim d_\pi}\Big[\mathbb{I}(s \notin \mathcal{U})\|\Big] \\ &\leq \mathbb{E}_{s \sim d_\pi}\Big[\mathbb{I}(s \notin \mathcal{U})\frac{\min_{a \in \mathcal{A}} \text{Var}_i[\pi_i(a|s)]}{\beta(\mathcal{U})}\Big] \\ &= \frac{1}{\beta(\mathcal{U})}\mathbb{E}_{s \sim d_\pi}\Big[\mathbb{I}(s \notin \mathcal{U}) \sum_{a \in \mathcal{A}} \pi(a|s)\text{Var}_i[\pi_i(a|s)]\Big] \\ &= \frac{1}{\beta(\mathcal{U})} \underbrace{\sum_{s \notin \mathcal{U}} d_\pi(s) \sum_{a \in \mathcal{A}} \pi(a|s)\text{Var}_i[\pi_i(a|s)]}_{A(\pi)} \end{aligned}$$

Now observe we can decompose the RL cost as follows:

$$
\begin{aligned}
J_{\mathrm{U}}(\pi) &= \mathbb{E}_{s\sim d_\pi, a\sim\pi(\cdot|s)}\Big[\mathrm{Var}_i \pi_i(a|s)\Big] \\
&= \sum_s d_\pi(s) \sum_a \pi(a|s)\Big[\mathrm{Var}_i \pi_i(a|s)\Big] \\
&= \underbrace{\sum_{s\in U} d_\pi(s) \sum_a \pi(a|s)\Big[\mathrm{Var}_i \pi_i(a|s)\Big]}_{B(\pi)} + \underbrace{\sum_{s\notin \mathcal{U}} d_\pi(s) \sum_a \pi(a|s)\Big[\mathrm{Var}_i \pi_i(a|s)\Big]}_{A(\pi)}
\end{aligned}
$$

Putting these together, we get the following:

$$
\begin{aligned}
J_{\mathrm{exp}}(\pi) &\leq \alpha(\mathcal{U}) J_{\mathrm{BC}}(\pi) + \frac{1}{\beta(\mathcal{U})} A(\pi) \\
&= \frac{\alpha(\mathcal{U})\beta(\mathcal{U})}{\beta(\mathcal{U})} J_{\mathrm{BC}}(\pi) + \frac{\alpha(\mathcal{U})}{\alpha(\mathcal{U})\beta(\mathcal{U})} A(\pi) \\
&= \frac{\alpha(\mathcal{U})\beta(\mathcal{U})}{\beta(\mathcal{U})} J_{\mathrm{BC}}(\pi) + \frac{\alpha(\mathcal{U})}{\alpha(\mathcal{U})\beta(\mathcal{U})} A(\pi) \\
&\leq \frac{\alpha(\mathcal{U})}{\beta(\mathcal{U})} J_{\mathrm{BC}}(\pi) + \frac{\alpha(\mathcal{U})}{\beta(\mathcal{U})} A(\pi) \\
&\leq \frac{\alpha(\mathcal{U})}{\beta(\mathcal{U})} \Big(J_{\mathrm{BC}}(\pi) + A(\pi) \Big) \\
&\leq \frac{\alpha(\mathcal{U})}{\beta(\mathcal{U})} \Big(J_{\mathrm{BC}}(\pi) + J_{\mathrm{U}}(\pi) \Big) \\
&= \frac{\alpha(\mathcal{U})}{\beta(\mathcal{U})} J_{\mathrm{alg}}(\pi)
\end{aligned}
$$

Here we have used the fact that $\beta(\mathcal{U}) \leq 1$ since $0 \leq \pi(a|s) \leq 1$ and $\alpha(\mathcal{U}) \geq \sum_{s\in\mathcal{U}} \frac{d_\pi^\star(s)}{d_\pi^\star(s)} \geq 1$ hence $\frac{1}{\alpha(\mathcal{U})} \leq 1$. Taking the minimum over subsets $\mathcal{U} \subseteq \mathcal{S}$, we get $J_{\mathrm{exp}}(\pi) \leq \kappa J_{\mathrm{alg}}(\pi)$.

□

Lemma 2. $\min_{\pi\in\Pi} J_{\mathrm{alg}}(\pi) \leq 2\epsilon$

Proof. Plugging the optimal policy into J_{alg}, we get:

$$
\begin{aligned}
J_{\mathrm{alg}}(\pi^\star) &= J_{\mathrm{BC}}(\pi^\star) + J_{\mathrm{U}}(\pi^\star) \\
&= 0 + \mathbb{E}_{s\sim d_{\pi^\star}, s\sim\pi^\star(\cdot|s)}\Big[\mathrm{Var}_i[\pi_i(a|s)]\Big] \\
&= \mathbb{E}_{s\sim d_{\pi^\star}, s\sim\pi^\star(\cdot|s)}\Big[\frac{1}{|E|}\sum_i \Big(\pi_i(a|s) - \bar{\pi}(a|s)\Big)^2\Big] \\
&\leq \mathbb{E}_{s\sim d_{\pi^\star}, s\sim\pi^\star(\cdot|s)}\Big[\frac{1}{|E|}\sum_i \Big(\pi_i(a|s) - \pi^\star(a|s)\Big)^2 + \Big(\bar{\pi}(a|s) - \pi^\star(a|s)\Big)^2\Big] \\
&= \underbrace{\mathbb{E}_{s\sim d_{\pi^\star}, s\sim\pi^\star(\cdot|s)}\Big[\frac{1}{|E|}\sum_i \Big(\pi_i(a|s) - \pi^\star(a|s)\Big)^2\Big]}_{\mathrm{Term1}} + \underbrace{\mathbb{E}_{s\sim d_{\pi^\star}, s\sim\pi^\star(\cdot|s)}\Big[\Big(\bar{\pi}(a|s) - \pi^\star(a|s)\Big)^2\Big]}_{\mathrm{Term2}}
\end{aligned}
$$

We will first bound Term 1:

$$\begin{aligned}
\mathbb{E}_{s\sim d_{\pi^\star}, s\sim\pi^\star(\cdot|s)}\Big[\frac{1}{|E|}\sum_{i=1}^{|E|}\Big(\pi_i(a|s)-\pi^\star(a|s)\Big)^2\Big] &= \frac{1}{|E|}\mathbb{E}_{s\sim d_{\pi^\star}}\Big[\sum_a \pi^\star(a|s)\sum_{i=1}^{|E|}\Big(\pi_i(a|s)-\pi^\star(a|s)\Big)^2\Big] \\
&\le \frac{1}{|E|}\mathbb{E}_{s\sim d_{\pi^\star}}\Big[\sum_a \pi^\star(a|s)\sum_{i=1}^{|E|}\Big|\pi_i(a|s)-\pi^\star(a|s)\Big|\Big] \\
&\le \frac{1}{|E|}\mathbb{E}_{s\sim d_{\pi^\star}}\Big[\sum_{i=1}^{|E|}\sum_a \Big|\pi_i(a|s)-\pi^\star(a|s)\Big|\Big] \\
&\le \frac{1}{|E|}\sum_{i=1}^{|E|}\mathbb{E}_{s\sim d_{\pi^\star}}\Big[\|\pi_i(\cdot|s)-\pi^\star(\cdot|s)\|\Big] \\
&\le \frac{1}{|E|}\sum_{i=1}^{|E|}\epsilon \\
&= \epsilon
\end{aligned}$$

We will next bound Term 2:

$$\begin{aligned}
\mathbb{E}_{s\sim d_{\pi^\star}, a\sim\pi^\star(\cdot|s)}\Big[\Big(\bar{\pi}(a|s)-\pi^\star(a|s)\Big)^2\Big] &= \mathbb{E}_{s\sim d_{\pi^\star}, s\sim\pi^\star(\cdot|s)}\Big[\Big(\pi^\star(a|s)-\frac{1}{|E|}\sum_{i=1}^{|E|}\pi_i(a|s)\Big)^2\Big] \\
&= \mathbb{E}_{s\sim d_{\pi^\star}, s\sim\pi^\star(\cdot|s)}\Big[\Big(\frac{1}{|E|}\sum_{i=1}^{|E|}\pi^\star(a|s)-\frac{1}{|E|}\sum_{i=1}^{|E|}\pi_i(a|s)\Big)^2\Big] \\
&= \mathbb{E}_{s\sim d_{\pi^\star}, s\sim\pi^\star(\cdot|s)}\Big[\Big(\frac{1}{|E|}\sum_{i=1}^{|E|}\pi^\star(a|s)-\pi_i(a|s)\Big)^2\Big] \\
&\le \mathbb{E}_{s\sim d_{\pi^\star}, s\sim\pi^\star(\cdot|s)}\Big[\frac{1}{|E|^2}|E|\sum_{i=1}^{|E|}\Big(\pi^\star(a|s)-\pi_i(a|s)\Big)^2\Big] (\mathrm{Cauchy-Schwarz}) \\
&= \frac{1}{|E|}\sum_{i=1}^{|E|}\mathbb{E}_{s\sim d_{\pi^\star}, s\sim\pi^\star(\cdot|s)}\Big[\Big(\pi^\star(a|s)-\pi_i(a|s)\Big)^2\Big] \\
&\le \frac{1}{|E|}\sum_{i=1}^{|E|}\mathbb{E}_{s\sim d_{\pi^\star}, s\sim\pi^\star(\cdot|s)}\Big[\Big|\pi^\star(a|s)-\pi_i(a|s)\Big|\Big] \\
&\le \frac{1}{|E|}\sum_{i=1}^{|E|}\mathbb{E}_{s\sim d_{\pi^\star}}\Big[\|\pi^\star(\cdot|s)-\pi_i(\cdot|s)\|\Big] \\
&= \frac{1}{|E|}\sum_{i=1}^{|E|}J_{\mathrm{BC}}(\pi_i) \\
&\le \epsilon
\end{aligned}$$

The last step follows from our optimization oracle assumption: $0 \le \min_{\pi\in\Pi} J_{\mathrm{BC}}(\pi) \le J_{\mathrm{BC}}(\pi^\star) = 0$, hence $J_{\mathrm{BC}}(\pi_i) \le 0 + \epsilon = \epsilon$. Combining the bounds on the two terms, we get $J_{\mathrm{alg}}(\pi^\star) \le 2\epsilon$. Since $\pi^\star \in \Pi$, the result follows.

□

Theorem 1. *Let $\hat{\pi}$ be the result of minimizing J_{alg} using our optimization oracle, and assume that $Q^{\pi^\star}_{T-t+1}(s,a) - Q^{\pi^\star}_{T-t+1}(s,\pi^\star) \leq u$ for all $a \in \mathcal{A}, t \in \{1,2,...,T\}, d^t_\pi(s) > 0$. Then $\hat{\pi}$ satisfies $J(\hat{\pi}) \leq J(\pi^\star) + 3u\kappa\epsilon T$.*

Proof. By our optimization oracle and Lemma 2, we have

$$\begin{aligned} J_{\text{alg}}(\hat{\pi}) &\leq \min_{\pi \in \Pi} J_{\text{alg}}(\pi) + \epsilon \\ &\leq 2\epsilon + \epsilon \\ &= 3\epsilon \end{aligned}$$

Combining with Lemma 1, we get:

$$\begin{aligned} J_{\text{exp}}(\hat{\pi}) &\leq \kappa J_{\text{alg}}(\hat{\pi}) \\ &\leq 3\kappa\epsilon \end{aligned}$$

Applying Theorem 1 from (Ross et al., 2011), we get $J(\hat{\pi}) \leq J(\pi^\star) + 3u\kappa\epsilon T$.

□

B Experimental Details

B.0.1 Atari Environments

All behavior cloning and ensemble models were trained to minimize the negative log-likelihood classification loss on the demonstration data for 500 epochs using Adam (Kingma & Ba, 2014) and a learning rate of $2.5 \cdot 10^{-4}$. For our method, we initially performed a hyperparameter search on Space Invaders over the following values:

Table 1: Hyperparameters (our method)

Hyperparameter	Values Considered	Final Value
Policy Learning rate	$2.5 \cdot 10^{-2}, 2.5 \cdot 10^{-3}, 2.5 \cdot 10^{-4}$	$2.5 \cdot 10^{-3}$
Quantile cutoff	$0.8, 0.9, 0.95, 0.98$	0.98
Number of supervised updates	$1, 5$	1
Number of policies in ensemble	5	5
Gradient clipping	0.1	0.1
Entropy coefficient	0.01	0.01
Value loss coefficient	0.5	0.5
Number of steps	128	128
Minibatch size	4	4

We then chose the best values and kept those hyperparameters fixed for all other environments. All other A2C hyperparameters follow the default values in the repo (Kostrikov, 2018): policy networks consisted of 3-layer convolutional networks with $8-32-64$ feature maps followed by a single-layer MLP with 512 hidden units.

For GAIL, we used the implementation in (Kostrikov, 2018) and replaced the MLP discriminator by a CNN discriminator with the same architecture as the policy network. We initially performed a hyperparameter search on Breakout with 10 demonstrations over the following values:

However, we did not find any hyperparameter configuration which performed better than behavioral cloning.

B.1 Continuous Control

All behavior cloning and ensemble models were trained to minimize the mean-squared error regression loss on the demonstration data for 500 epochs using Adam (Kingma & Ba, 2014) and a learning

Table 2: Hyperparameters (GAIL)

Hyperparameter	Values Considered	Final Value
Policy Learning rate	$2.5 \cdot 10^{-2}, 2.5 \cdot 10^{-3}, 2.5 \cdot 10^{-4}$	$2.5 \cdot 10^{-3}$
Discriminator Learning rate	$2.5 \cdot 10^{-2}, 2.5 \cdot 10^{-3}, 2.5 \cdot 10^{-4}$	$2.5 \cdot 10^{-3}$
Number of discriminator updates	$1, 5, 10$	5
Gradient clipping	0.1	0.1
Entropy coefficient	0.01	0.01
Value loss coefficient	0.5	0.5
Number of steps	128	128
Minibatch size	4	4

rate of $2.5 \cdot 10^{-4}$. Policy networks were 2-layer fully-connected MLPs with tanh activations and 64 hidden units.

Table 3: Hyperparameters (our method)

Hyperparameter	Values Considered	Final Value
Policy Learning rate	$2.5 \cdot 10^{-3}, 2.5 \cdot^{1} 0{-}4, 1 \cdot 10^{-4}, 5 \cdot 10^{-5}$	$2.5 \cdot 10^{-5}$
Quantile cutoff	0.98	0.98
Number of supervised updates	1	1
Number of policies in ensemble	5	5
Gradient clipping	0.1	0.1
Entropy coefficient	0.01	0.01
Value loss coefficient	0.5	0.5
Number of steps	128	128
Minibatch size	4	4