Population-Guided Imitation Learning

David S. Hippocampus[*]
Department of Computer Science
Cranberry-Lemon University
Pittsburgh, PA 15213
hippo@cs.cranberry-lemon.edu

Abstract

Learning to imitate expert behavior is a challenging problem, especially in environments with high-dimensional, continuous observations and unknown dynamics. It includes imitation learning from demonstrations (ILfD) and imitation learning from observations (ILfO). The simplest methods in imitation learning are behavior cloning (BC) and behavior cloning from observations (BCO), for ILfD and ILfO respectively. But BC suffers from the problem of distribution shift, while in ILfO, the inverse dynamic model heavily depends on the current policy, without sufficient generalization to the expert state distribution. Since there is no easily-specified reward function available, exploration is more important for imitation learning than regular RL. In this paper, and we propose population-based exploration techniques for imitation learning, which are simple to implement and improve sample efficiency significantly. And we find that population-based exploration can have much more performance improvement than that in regular RL problems. In ILfD, to enlarge the overall search region, we propose to use Stein Variational Gradient Descent (SVGD) to generate the multiple policies, and attenuate distribution shift by RL with intrinsic rewards. In ILfO, additionally, in order to produce more diverse state-action pairs to make the inverse dynamic model generalize better, we introduce neuro-evolution (NE) to further augment the exploration capability of learning policies. We find these population-based exploration techniques can have more performance improvement in imitation learning, than regular RL problems. Here the intrinsic rewards are simply generated by random network distillation (RND), trained over expert states. The proposed frameworks provide the imitation agent both the intrinsic intention of the demonstrator and better exploration ability, which is critical for the agent to outperform the demonstrator. With experiments of ILfD and ILfO over various difficult Atari games and MuJoCo environments, the proposed exploration-augmented methods show significant performance improvement, especially achieving 5X better sampling efficiency, compared with previous popular imitation learning methods.

1 Introduction

Imitation Learning (IL) [17] is a framework of reinforcement learning [29], where the agent has access to an optimal, reward-maximizing expert for the underlying environment. This access is usually provided via a dataset of trajectories where each observed state is annotated with the action produced by the expert policy. This is a powerful learning framework in contrast to standard reinforcement learning since not all tasks of interest admit easily-specified reward functions. This IL with expert actions available is often termed as imitation learning from demonstrations (ILfD). However, most of

[*] Use footnote for providing further information about author (webpage, alternative address)—*not* for acknowledging funding agencies.

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

state-of-the-art imitation learning methods do not have satisfactory performance in many difficult environments, e.g., Atari games, especially suffering from sampling efficiency.

The provision of expert action labels can often be laborious or incur significant cost due to the instrumentation used when recording expert actions. Actually a vast number of rich observation-only data sources are available in practice for imitation learning. Lots of recent work have explored the more natural problem formulation, where an agent must recover an imitation policy from a dataset containing only expert observation sequences [5, 15, 28, 30, 31]. While this Imitation Learning from Observations (ILfO) setting has tremendous potential in practice, such as enabling an agent to learn to play games from watching video clips by expert players, its performance is still not satisfactory in applications with continuous high-dimensional observations. In this paper, we investigate how to use population-based exploration techniques to advance the state-of-the-art performance of both ILfD and ILfO.

Among IL methods, behavior cloning (BC) is an elegant approach whereby agents are trained to directly mimic the behaviors of an expert rather than optimizing a reward function [10, 22, 23, 30]. It basically consists of training a policy to predict the expert's actions from states in the demonstration data using supervised learning, which has the simplest model and implementation complexity among IL methods. While appealingly simple, BC suffers from the problems of error accumulation and covariate shift, where the distribution over states observed at execution time can differ from the distribution observed during training. Minor errors initially produced can be accumulated and become amplified as the policy encounters states further and further [2, 10, 23].

The equivalence of BC in ILfO is behavior cloning from observation (BCO) introduced in [30], which leverages state-action trajectories collected under a random policy to train an inverse dynamics model for inferring the action responsible for a transition between two consecutive states in trajectories. With this inverse dynamic model at hand, the observation-only demonstration data can be converted into the more traditional dataset of state-action pairs over which standard BC can be applied. However, in addition to covariate shift, the inverse dynamic model is trained by collecting rollouts in the environment with the current policy, where the actions corresponding to state transitions of expert dataset are infeasible to obtain [5, 34].

In this work, we propose novel and simple population-based exploration techniques to augment the imitation learning, specifically, achieving better-than-demonstrator performance and improved sampling efficiency. In stead of conducting behavior cloning as supervised learning, the expert demonstrations are incorporated into agent's policies in a soft manner via Stein variational gradient descent (SVGD) [14], transforming a population of policies to match the target distribution in expert demonstrations. Here a form of (functional) gradient descent is performed to minimize the KL divergence and drive the weights of policies to fit the true posterior distribution. It mimics a gradient dynamics at the particle level, driving policies towards the high probability areas defined by expert demonstrations. It also has a repulsive force to every policy, preventing all the policies to collapse together at the same time and keeping enough diversity. In order to address the covariate shift, we introduce an RL process, where the agent is interacting with the environment and maximizing the intrinsic rewards received along the trajectory. Here the intrinsic reward is obtained via random network distillation (RND)[4]. Specifically, before the imitation agent starts learning, a random neural network (with fixed weights) is distilled into another neural network, by minimizing the prediction errors over states in the expert dataset, and the reciprocal of its predication error of visited states is used as the intrinsic reward in the RL process. In ILfO, in order to train the inverse dynamic model to generalize better, we further adopt the neuro-evolution [27, 11] technique to produce state-action pairs with bigger diversity.

Compared with previous work, our method has multiple merits. First, without building a specific generative model for expert state distribution in many previous works [1, 25, 35], we use the simple RND technique to address covariate shift. Second the expert demonstrations are fused into policies of imitation agents via SVGD, learning expert demonstrations and gaining diversity at the same time. Third, the neuro-evolution provides the agent a better exploration ability, in a novel and simple approach, advancing the state-of-the-art performance in terms of both performance and sampling efficiency. Finally, we find that, the population-based exploration techniques adopted here can have much more performance gain in imitation learning, compared with regular RL problems.

2 Related Work

2.1 Imitation Learning from Demonstrations

In general the approaches in ILfD fall into two categories: behavior cloning (BC) [10, 22, 23, 21], which optimizes the current policy over the action prediction errors in expert demonstrations, and inverse reinforcement learning (IRL) [1, 16], which infers the reward used by expert to guide the agent policy learning procedure. Recently methods based on adversarial learning have been proposed to tackle the covariate shift of BC [7, 8, 9, 10]. These methods train an RL agent not only to imitate demonstrated actions, but also to visit demonstrated states. Since the true rewards are unknown, a reward function is constructed from the demonstrations and visited trajectories via adversarial learning. However, the alternative training of policy and discriminator can make the learning process unstable, significantly increasing the sampling complexity [3]. Some work solve the imitation learning problem in the frameworks of Q-learning [19, 24]. However, since these methods set the reward based on the appearance of transitions in the expert demonstrations, resulting the problem of sparse reward when few demonstrations are available. Another stream of work [2, 4, 33] uses an extra model, such as random network distillation and disagreement, to estimate the support of the expert's distribution in state-action space, and minimizes an RL cost designed to guide the agent towards the states within the expert's support. But these estimation models increases the model and implementation complexity. And they may not give a good distance between states in replay buffer and those covered by expert demonstrations, especially in high-dimensional cases, which may mislead the agent to wrong states far from expert's support.

2.2 Imitation Learning from Observations

The recent papers in ILfO can be primarily classified into categories, e.g., ILfO with inverse dynamic model [5, 30], and ILfO with GAIL [31, 32]. In the first category, the exact actions for state transition pairs in expert demonstrations are inferred with a learned inverse dynamic model. And the policy is trained to predict the inferred actions given corresponding states from expert demonstrations. In addition to covariate shift, these methods suffer from the insufficient generalization of the inverse dynamic model, producing inaccurate predicted actions for expert state transitions. In the second category, the authors generally follow the idea of GAIL [9] but replace the state-action pairs with state transition pairs. However, methods in this category suffer from the instability of adversarial learning and large amount of samples needed from the environment.

3 Methodology

The primary motivation of this work is to propose population-based exploration techniques to help imitating agents achieve better-than-demonstrator performance with less sample complexity. Our algorithm has three components: i) the RL process with intrinsic reward from RND drives imitating agents towards the expert's state distribution, and ii) the policies of imitating agents learn the expert demonstrations by SVGD, introducing some diversity via the repulsive force in the kernel space, and iii) in ILfO, neuro-evolution (NE) technique is adopted to generate more diverse state-action pairs, increasing generalizability of inverse dynamic model significantly.

3.1 Intrinsic Reward

For the simplicity of implementation and model complexity, we adopt random network distillation (RND) [4] to generate intrinsic rewards during the RL process. RND assess state novelty by distilling a random neural network (with fixed weights) into another neural network with the same architecture. For every state, the random network produces random features which are continuous. The second network is trained to reproduce the output of the random network for states in expert demonstrations, before the learning of imitating agents starts. The reciprocal of the l2 norm of prediction error is the intrinsic reward received by the imitating agents. The errors will be high for states out of the expert's state distribution, since the second network never visited them during the training. So the intrinsic reward can drive the policies of imitating agents towards the expert's state distribution.

3.2 Behavior Cloning via Stein Variational Gradient Descent

The expert demonstrations are fused into policies of imitating agents via SVGD [14], balancing between the behavior imitation and diversity among imitating policies. SVGD is a form of functional gradient descent minimize the KL divergence between the distributions of imitating policies and expert demonstrations, and drive the imitating policies to fit the true posterior distribution given the expert state-action pairs. Theoretically, it is to derive a closed form solution for the optimal smooth perturbation direction that gives the steepest descent on the KL divergence within the unit ball of a reproducing kernel Hilbert space (RKHS). Assume there are M imitating policies $\{\pi_{\theta^i}\}$ with weights $\theta^1, \ldots, \theta^M$. And the expert demonstrations form a dataset with state-action pairs, i.e., $\mathcal{D}^E := \{(s_i, a_i)\}$. Given the minibatch $\mathcal{B}$ from the expert dataset $\mathcal{D}^E$, the i-th policy is updated as below,

$$\theta^i \leftarrow \theta^i + \epsilon\hat{\phi}(\theta^i) \tag{1}$$

where

$$\hat{\phi}(\theta^i) = \frac{1}{M}\sum_{j=1}^{M}\left[k(\theta^i, \theta^j)\nabla_{\theta^i}\left(\sum_{(s,a)\in\mathcal{B}} \log \pi_{\theta^i}(a|s)\right) + \nabla_{\theta^i} k(\theta^i, \theta^j)\right] \tag{2}$$

and the kernel $k(x, y)$ is chosen to be the radial base function (RBF), i.e., $\exp(-\frac{1}{h}\|x - y\|_2^2)$. By updating the policy weights with (1), the policies of imitating agents learn the expert behavior with diversity incorporated at the same time.

3.3 Neuro Evolutionary Reinforcement Learning

In order to generate more diverse learning experience for better exploration and generalization, we incorporate Neuro Evolutionary (NE) reinforcement learning [12], a hybrid algorithm that incorporates evolutionary algorithm's (EA) population-based approach to train an RL agent, and transfers the RL agent into the EA population periodically to inject gradient information into the EA. The key insight of NE is that an EA can be used to address the core challenges within DRL without losing out on the ability to leverage gradients for higher sample efficiency.

Specifically, before learning starts, a population of actor networks is initialized with random weights. In addition, a learning actor and critic network are initialized randomly. The population of actors are then evaluated in an episode of interaction with the environment. The fitness for each actor is computed as the cumulative sum of the reward that they receive over the timesteps in that episode. A selection operator then selects a portion of the population for survival with probability commensurate on their relative fitness scores. The actors in the population are then probabilistically perturbed through mutation and crossover operations to create the next generation of actors. A select portion of actors with the highest relative fitness are preserved as elites and are shielded from the mutation step.

3.4 Imitation Learning from Demonstrations

In ILfD, we primarily focus on Atari games, which are most difficult in the field and don't have satisfactory performance until now. Here, we investigate how the proposed population-guided exploration techniques can improve the performance, especially the convergence speed. Since Atari games have discrete actions and states, our algorithm is built on top of PPO [26]. The idea is to apply SVGD [14] in behavior cloning and produce a population of actors mimicking expert behavior with enough diversity. With experience collected by itself and the population of actors, the learning actor gets updated via gradients of PPO objective. The value function of the learning agent is also updated with trajectories of experience from both itself and population of actors, where V-trace target [6] is used to correct the difference between the learning actor and population of actors. In implementation, the learning rate and number of steps of SVGD is carefully tuned to make sure that the population of actors are not too different from the learning actor. The pseudocode of the proposed algorithm is shown in Algorithm 1.

3.5 Imitation Learning from Observations

In ILfO, we use population of actors to augment BCO [30], primarily focusing on high-dimensional MuJoCo environments. Different from behavior cloning in ILfD, in addition to distribution shift,

Algorithm 1: Population-guided ILfD

```
Input :Expert demonstration dataset $\mathcal{D}_E = \{(s_i, a_i)\}$; learning actor $\pi_\theta$ and critic $v_\phi$;
       population of actors $\{\pi_{\theta^i}\}_{i=1}^M$; neural networks in RND $f_{\varphi^1}(\cdot), f_{\varphi^2}(\cdot)$
Initialize $f_{\varphi^1}(\cdot), f_{\varphi^2}(\cdot)$ with random weights
Train $f_{\varphi^2}(\cdot)$ to match the output of $f_{\varphi^1}(\cdot)$ over expert states $s \in \mathcal{D}_E$
Pre-training $\pi_\theta$ by behavior cloning on $\mathcal{D}_E$
for $e = 1, \ldots,$ do
    Initialize actors $\{\pi_{\theta^i}\}_{i=1}^M$ with weights of learning actor $\theta$ perturbed by random noise
    Update $\{\pi_{\theta^i}\}_{i=1}^M$ via SVGD (1) with minibatch from $\mathcal{D}^E$
    Collect trajectories of experience $\tau_0$ with learning actor $\pi_\theta$
    Collect trajectories of experience $\{\tau_i\}_{i=1}^M$ with actors $\{\pi_{\theta^i}\}_{i=1}^M$
    Update learning actor $\pi_\theta$ with trajectories of experience $\{\tau_i\}_{i=0}^M$ via PPO objective,
      where the reward is given by $r(s) = 1/\|f_{\varphi^1}(s) - f_{\varphi^2}(s)\|^2$
    Update learning critic $v_\phi$ with trajectories $\tau_0$, and $\{\tau_i\}_{i=1}^M$ transformed into V-trace
      targets
end
```

BCO suffers from the problem that the inverse dynamic model is only trained by collecting state-actions pairs with current policies, which can be stuck at local areas of state space without enough generalization on the support of expert's state distribution. Here we find that population-based exploration technique is the right tool to tackle this problem. Specifically, we introduce neuro-evolution (NE) reinforcement learning technique [12], where crossover and mutation can transform actors to generate more diverse experience. It is built on DDPG [13] and reward used here is also obtained from RND. Different from conventional NE, we also use SVGD to inject expert demonstrations into selected actors in a Bayesian manner, so as to make evolved actors not far away from the learning actor. By storing generated state-action pairs into replay buffer, the inverse dynamic model can be trained to have more generalizability, giving state-only observations in the expert dataset more accurate action labels. The detailed algorithm is shown in Algorithm 2.

Algorithm 2: Population-guided ILfO

```
Input :Expert state-only dataset $\mathcal{D}_E = \{(s_i, s_{i+1})\}$; learning actor $\pi_\theta$ and critic $v_\phi$;
       population of actors $\{\pi_{\theta^i}\}_{i=1}^M$, with number of elites $K$; neural networks in RND
       $f_{\varphi^1}(\cdot), f_{\varphi^2}(\cdot)$; inverse dynamic model $d_\xi(\cdot, \cdot)$; replay buffer $\mathcal{B}$
Initialize $f_{\varphi^1}(\cdot), f_{\varphi^2}(\cdot)$ and actors $\{\pi_{\theta^i}\}_{i=1}^M$ with random weights
Train $f_{\varphi^2}(\cdot)$ to match the output of $f_{\varphi^1}(\cdot)$ over expert states $s \in \mathcal{D}_E$
for $e = 1, \ldots,$ do
    Collect a trajectory of experience $\tau_0$ with learning actor $\pi_\theta$, and store it into $\mathcal{B}$
    Collect trajectories of experience $\{\tau_i\}_{i=1}^M$ with actors $\{\pi_{\theta^i}\}_{i=1}^M$, and store them into $\mathcal{B}$
    Calculate fitness (reward) for the population $\{\pi_{\theta^i}\}_{i=1}^M$, and select the top-$K$ actors to
      form the elite set $e$
    Select $(M - K)$ actors from the population randomly with replacement to form a set $S$
    Conduct crossover and mutation over sets $e, S$, and update the population of actors
    Update actors in set $S$ via SVGD (1) with minibatch from $\mathcal{D}^E$
    Update learning actor $\pi_\theta$ with minibatches from $\mathcal{B}$ via PPO objective, where the reward is
      given by $r(s) = 1/\|f_{\varphi^1}(s) - f_{\varphi^2}(s)\|^2$
    Update learning critic $v_\phi$ with minibatches from $\mathcal{B}$
    Replace the worst actor in $\{\pi_{\theta^i}\}_{i=1}^M$ with learning actor $\pi_\theta$
end
```

4 Experiments

4.1 Imitation Learning from Demonstrations

We evaluated the proposed IL method on many Atari environments. The expert policy is trained by PPO [26] and generate a butch of expert trajectories stored in $\mathcal{D}_E$. In order to stabilize the training process, the reward is clipped into -1 or 1 based on the threshold, set by the $\beta-$quantile of rewards over all the states in expert's demonstrations [2].

The baselines for comparison are standard behavior cloning (BC) [20] and generative adversarial imitation learning (GAIL) [9]. We find that the propose method can outperform both BC and GAIL in all of the evaluated environments. It is already known that GAIL cannot perform well on image-based environments [19], and our method has significant improvement over that.

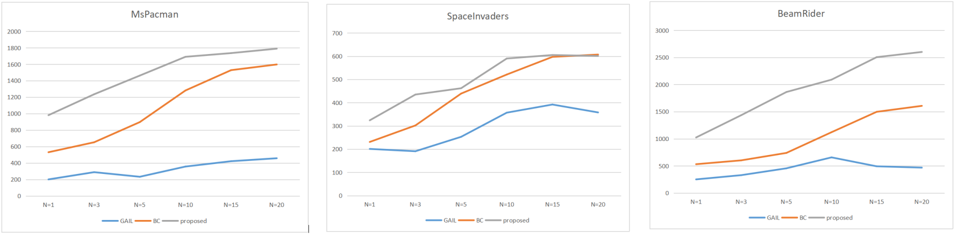

Figure 1: Experiments on Atari games. Average reward vs number of expert trajectories.

4.2 Imitation Learning from Observations

For ILfO, we carry out comparative evaluations over several baselines, including DeepMimic [18], BCO [30], and GAIfO [31]. All experiments are evaluated within fixed steps, where Pendulum and DoublePendulum stop at $5e3$, HalfCheetah stops at $1e7$, and Ant stops at $1e7$. On each task, we run each algorithm over five times with different random seeds. The eventual results are summarized in Table 1, which is averaged over 50 trials of the learned policies.

Table 1: Summary of quantitative results. All results correspond to the original exact reward defined in OpenAI Gym

	Pendulum	DoublePendulum	Hopper	HalfCheetah	Ant
DeepMimic	731.0±19.0	454.4±154.0	2292.6±1068.9	202.6±4.4	-985.3 ± 13.6
BCO	24.9 ± 0.8	80.3 ± 13.1	1266.2 ± 1062.8	4557.2 ± 90.0	562.5 ± 384.1
GAIfO	980.2 ± 3.0	4240.6 ± 4525.6	1021.4 ± 0.6	3955.1 ± 22.1	-1415.0 ± 161.1
Ours	$\mathbf{1000.0 \pm 0.0}$	$\mathbf{8259.1 \pm 0.8}$	$\mathbf{3508.9 \pm 85.9}$	$\mathbf{5495.1 \pm 91.1}$	$\mathbf{5120.1 \pm 64.7}$

References

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

[2] Kiante Brantley, Wen Sun, and Mikael Henaff. Disagreement-regularized imitation learning. In *International Conference on Learning Representations*, 2020.

[3] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018.

[4] Yuri Burda, Harrison Edwards, Amos Storkey, and Oleg Klimov. Exploration by random network distillation. In *International Conference on Learning Representations*, 2019.

[5] Ashley Edwards, Himanshu Sahni, Yannick Schroecker, and Charles Isbell. Imitating latent policies from observation. In *International Conference on Machine Learning*, pages 1755–1763, 2019.

[6] Lasse Espeholt, Hubert Soyer, Remi Munos, Karen Simonyan, Vlad Mnih, Tom Ward, Yotam Doron, Vlad Firoiu, Tim Harley, Iain Dunning, et al. Impala: Scalable distributed deep-rl with importance weighted actor-learner architectures. In *International Conference on Machine Learning*, pages 1407–1416, 2018.

[7] Chelsea Finn, Sergey Levine, and Pieter Abbeel. Guided cost learning: Deep inverse optimal control via policy optimization. In *International conference on machine learning*, pages 49–58, 2016.

[8] Justin Fu, Katie Luo, and Sergey Levine. Learning robust rewards with adversarial inverse reinforcement learning. *arXiv preprint arXiv:1710.11248*, 2017.

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

[10] Liyiming Ke, Matt Barnes, Wen Sun, Gilwoo Lee, Sanjiban Choudhury, and Siddhartha Srinivasa. Imitation learning as f-divergence minimization. *arXiv preprint arXiv:1905.12888*, 2019.

[11] Shauharda Khadka, Somdeb Majumdar, Tarek Nassar, Zach Dwiel, Evren Tumer, Santiago Miret, Yinyin Liu, and Kagan Tumer. Collaborative evolutionary reinforcement learning. In *International Conference on Machine Learning*, pages 3341–3350, 2019.

[12] Shauharda Khadka and Kagan Tumer. Evolution-guided policy gradient in reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 1188–1200, 2018.

[13] 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. *arXiv preprint arXiv:1509.02971*, 2015.

[14] Qiang Liu and Dilin Wang. Stein variational gradient descent: A general purpose bayesian inference algorithm. In *Advances in neural information processing systems*, pages 2378–2386, 2016.

[15] YuXuan Liu, Abhishek Gupta, Pieter Abbeel, and Sergey Levine. Imitation from observation: Learning to imitate behaviors from raw video via context translation. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1118–1125. IEEE, 2018.

[16] Andrew Y Ng and Stuart J Russell. Algorithms for inverse reinforcement learning. In *Proceedings of the Seventeenth International Conference on Machine Learning*, pages 663–670, 2000.

[17] Takayuki Osa, Joni Pajarinen, Gerhard Neumann, J Andrew Bagnell, Pieter Abbeel, Jan Peters, et al. An algorithmic perspective on imitation learning. *Foundations and Trends® in Robotics*, 7(1-2):1–179, 2018.

[18] Xue Bin Peng, Pieter Abbeel, Sergey Levine, and Michiel van de Panne. Deepmimic: Example-guided deep reinforcement learning of physics-based character skills. *ACM Transactions on Graphics (TOG)*, 37(4):1–14, 2018.

[19] Siddharth Reddy, Anca D Dragan, and Sergey Levine. Sqil: imitation learning via regularized behavioral cloning. *arXiv preprint arXiv:1905.11108*, 2019.

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

[21] Stéphane Ross and J Andrew Bagnell. Agnostic system identification for model-based reinforcement learning. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, pages 1905–1912, 2012.

[22] Stephane Ross and J Andrew Bagnell. Reinforcement and imitation learning via interactive no-regret learning. *arXiv preprint arXiv:1406.5979*, 2014.

[23] 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*, pages 627–635, 2011.

[24] Fumihiro Sasaki, Tetsuya Yohira, and Atsuo Kawaguchi. Sample efficient imitation learning for continuous control. In *International Conference on Learning Representations*, 2019.

[25] Yannick Schroecker, Mel Vecerik, and Jon Scholz. Generative predecessor models for sample-efficient imitation learning. In *International Conference on Learning Representations*, 2018.

[26] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[27] Felipe Petroski Such, Vashisht Madhavan, Edoardo Conti, Joel Lehman, Kenneth O Stanley, and Jeff Clune. Deep neuroevolution: Genetic algorithms are a competitive alternative for training deep neural networks for reinforcement learning. *arXiv preprint arXiv:1712.06567*, 2017.

[28] Wen Sun, Anirudh Vemula, Byron Boots, and J Andrew Bagnell. Provably efficient imitation learning from observation alone. *arXiv preprint arXiv:1905.10948*, 2019.

[29] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[30] Faraz Torabi, Garrett Warnell, and Peter Stone. Behavioral cloning from observation. *arXiv preprint arXiv:1805.01954*, 2018.

[31] Faraz Torabi, Garrett Warnell, and Peter Stone. Generative adversarial imitation from observation. *arXiv preprint arXiv:1807.06158*, 2018.

[32] Faraz Torabi, Garrett Warnell, and Peter Stone. Imitation learning from video by leveraging proprioception. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pages 3585–3591. AAAI Press, 2019.

[33] 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*, pages 6536–6544, 2019.

[34] Chao Yang, Xiaojian Ma, Wenbing Huang, Fuchun Sun, Huaping Liu, Junzhou Huang, and Chuang Gan. Imitation learning from observations by minimizing inverse dynamics disagreement. In *Advances in Neural Information Processing Systems*, pages 239–249, 2019.

[35] Xingrui Yu, Yueming Lyu, and Ivor W Tsang. Intrinsic reward driven imitation learning via generative model. *arXiv preprint arXiv:2006.15061*, 2020.