Towards Sample-efficient Apprenticeship Learning from Suboptimal Demonstration

Letian Chen, Rohan Paleja, Matthew Gombolay

College of Computing, Georgia Institute of Technology
letian.chen@gatech.edu, rohan.paleja@gatech.edu, matthew.gombolay@cc.gatech.edu

Abstract

Learning from Demonstration (LfD) seeks to democratize robotics by enabling non-roboticist end-users to teach robots to perform novel tasks by providing demonstrations. However, as demonstrators are typically non-experts, modern LfD techniques are unable to produce policies much better than the suboptimal demonstration. A previously-proposed framework, SSRR, has shown success in learning from suboptimal demonstration but relies on noise-injected trajectories to infer an idealized reward function. A random approach such as noise-injection to generate trajectories has two key drawbacks: 1) Performance degradation could be random depending on whether the noise is applied to vital states and 2) Noise-injection generated trajectories may have limited suboptimality and therefore will not accurately represent the whole scope of suboptimality. We present *Systematic* Self-Supervised Reward Regression, S3RR, to investigate systematic alternatives for trajectory degradation. We carry out empirical evaluations and find S3RR can learn comparable or better reward correlation with ground-truth against a state-of-the-art learning from suboptimal demonstration framework.

Introduction

The field of Learning from Demonstration (LfD) seeks to democratize access to robotic assistance by empowering end-users to teach robots by demonstrations, avoiding the need for an army of engineers to manually program and test robots for every task (Seraj and Gombolay 2020). Due to the democratized access via LfD, applications ranging from healthcare (Gombolay et al. 2018), manufacturing (Wang and Gombolay 2020), auto-pilot (Pan et al. 2018), to assistive jobs (Bhattacharjee et al. 2019) have been able to take advantage of expert users in robot learning. However, demonstrations from real end-users are typically suboptimal due to limited cognitive abilities (Newell and Simon 1972; Nikolaidis et al. 2017), inhibiting robot learning frameworks from using direct imitation learning (Chernova and Thomaz 2014; Ross, Gordon, and Bagnell 2011; Paleja et al. 2020), which learns a direct mapping from a state to the demonstrated action, and therefore only imitates the behavior rather than discovering the latent objective of the demonstrator.

Inverse Reinforcement Learning (IRL) (Ng, Russell et al. 2000; Chen et al. 2020) on the other hand, infers the demonstrator's underlying objective. With the objective represented as a reward function, Reinforcement Learning (RL) can find a policy accomplishing the desired goal. However, similar to imitation learning, classic IRL approaches (Abbeel and Ng 2004; Gombolay et al. 2016) rely on the assumption that the demonstration provided is optimal. Maximum-entropy IRL (Ziebart et al. 2008) and Bayesian IRL (Ramachandran and Amir 2007) relax this to stochastic optimality, but in general, it cannot produce a much better policy than the suboptimal demonstrations, limiting the applicability of LfD with naïve users.

Drawing inspiration from Preference-based Reinforcement Learning (Wirth et al. 2017), D-REX (Brown, Goo, and Niekum 2019) and SSRR (Chen et al. 2020) have developed successful learning from suboptimal demonstration frameworks that utilize a performance degradation across noisy trajectories to recover the demonstrator's latent objective. SSRR (Chen et al. 2020) set the state-of-the-art by first utilizing self-supervision to create new trajectories via noise-injection while simultaneously estimating the reward for each trajectory. Next, under the assumption that increased noise will further degrade the trajectory performance, a noise-performance relationship is developed across the generated trajectories, which is then utilized to learn an idealized reward function. Following the learning of a reward function, policy learning occurs via RL to generate a policy that can outperform the demonstrations gathered from suboptimal end-users.

In this paper, we investigate *systematic* alternatives to self-supervised trajectory generation via noise-injection. Utilizing noise-injection to generate trajectories has two main drawbacks: 1) Noise injection is only effective if the noise is applied in vital states among the trajectory randomly. For example, if the robot is idle for the first two seconds of the demonstration, adding noise during this period will have a very small impact compared with adding noise when the robot is executing a maneuver; 2) Degraded trajectories via noise-injection have increased randomness but may not be able to accurately represent the whole scope of suboptimality. An adversarial/highly sub-optimal policy may perform much worse than a random policy. For example, if the objective is to make a robot cheetah move forward,

the expected performance a random policy could achieve is to not move. However, a policy moving backward yields even lower reward, representing an even further degraded instance in the degradation-performance relationship. We extend our prior work Self-Supervised Reward Regression (SSRR) (Chen, Paleja, and Gombolay 2020), to propose *S3RR, Systematic Self-Supervised Reward Regression* that assesses alternative approaches to generate performance-degraded trajectories. Our proposed trajectory generation approaches allows S3RR to recover more accurate reward functions while avoiding the pitfalls of trajectory generation via noise-injection. We then present empirical results across two simulated robot control tasks and find S3RR can achieve higher reward correlation with ground-truth than SSRR.

Preliminaries

In this section, we introduce Markov Decision Processes, IRL, and SSRR (Chen, Paleja, and Gombolay 2020).

Markov Decision Process – A Markov Decision Process (MDP) (White 1993) M is defined as a 6-tuple $\langle S, A, R, T, \gamma, \rho_0 \rangle$. S and A are the state and action spaces, respectively: $s \in S, a \in A$. $R(s, a)$ defines the reward given to an agent for executing action, a, in state, s. $T(s'|s, a)$ denotes the probability of transitioning from state, s, to state, s', when applying action, a. $\gamma \in [0, 1]$ is the temporal discount factor. $\rho_0(s)$ is the distribution of initial state. A policy, $\pi(a|s)$, defines the probability of the agent taking action, a, in state, s. The objective for RL is to find the optimal policy, $\pi^* = \arg\max_\pi \mathbb{E}_{\tau \sim \pi} \left[\sum_{t=0}^{T} \gamma^t R(s_t, a_t) \right]$, which maximizes cumulative discounted reward in the sampled trajectories, where a trajectory $\tau = \langle s_0, a_0, r_0, \cdots, s_T, a_T, r_T \rangle$ is a sequence generated by the MDP and the policy: $s_0 \sim \rho_0(\cdot), a_t \sim \pi(\cdot|s_t), s_{t+1} \sim T(\cdot|s_t, a_t), r_t \sim R(s_t, a_t), \ \forall t \geq 0$. We consider the maximum entropy version of RL goal (Ziebart 2010), $\pi^* = \arg\max_\pi \mathbb{E}_{\tau \sim \pi} [\sum_{t=0}^{T} \gamma^t R(s_t, a_t) + \alpha H(\pi(\cdot|s_t))]$, which adds an entropy bonus (i.e., $H(\cdot)$) to favor stochastic policies and encourage exploration during the training process.

Inverse Reinforcement Learning – The goal of IRL is to take as input an MDP sans reward function and a set of demonstration trajectories $\mathcal{D} = \{\tau_1, \tau_2, \cdots, \tau_N\}$ and output a recovered reward function R, which denotes the objective trajectories in $\mathcal{D}$ optimize. Our method is based on AIRL (Fu, Luo, and Levine 2018), which casts the IRL problem in a generative-adversarial framework. AIRL consists of a discriminator in the form of a reward function and a generator in the form of a learning policy. The discriminator, D, is given by $D_\theta(s, a) = \frac{e^{f_\theta(s,a)}}{e^{f_\theta(s,a)} + \pi(a|s)}$, where $f_\theta(s, a)$ is the reward function, parameterized by θ, and $\pi(a|s)$ is a policy. The discriminator's objective is to distinguish generated rollouts from demonstrated trajectories via a binary cross entropy, updating the reward function parameter θ via gradient descent. In other words, the reward function learns to provide higher rewards for the demonstration trajectories than the generated policy rollouts. The policy, π, is

trained to maximize the pseudo-reward function given by $\hat{R} = f_\theta(s, a)$, resulting in behaviors close to the demonstrations.

SSRR – SSRR learns an idealized reward function from suboptimal demonstrations, consisting of three phases: Phase 1) self-supervised data generation, Phase 2) Noise-Performance characterization, and Phase 3) Reward & Policy learning. Figure 1 top depicts SSRR.

Phase 1) For self-supervised data generation, SSRR utilizes the initial reward, $\tilde{R}$, and the initial policy, $\tilde{\pi}$, inferred from utilizing AIRL on the suboptimal demonstrations. SSRR chooses random actions uniformly (named "injecting noise") to replace the learned policy decision with probability η (Equation 1) to produce "noisy" policies. The trajectories, $\{\tau^i\} \ \forall i$ generated via noise-injection, consist of four elements $\tau^i = \langle \eta^i, \{s_t^i\}, \{a_t^i\}, \{\tilde{r}_t^i\} \rangle$: the noise parameter, η^i, a set of states and actions, $\{s_t^i\}$ and $\{a_t^i\}$, and the corresponding initial reward $\tilde{r}_t^i = \tilde{R}(s_t^i, a_t^i)$.

$$\tau_\eta \sim \pi_\eta(a|s) = \eta U(a) + (1 - \eta)\pi_{\text{AIRL}}(a|s). \quad (1)$$

Phase 2) For noise-performance curve fitting, SSRR empirically characterizes the noise-performance curve. Since SSRR has access to an initial reward estimate for each trajectory, $\tilde{R}(\tau^i) = \sum_t \tilde{R}(s_t^i, a_t^i)$, SSRR regresses a sigmoid function (see Equation 2) towards the cumulative estimated rewards via least squared error minimization. The sigmoid acts as a low-pass filter on the noise-performance relationship from AIRL's initial reward, creating a smooth noise-performance relationship even in the presence of high-frequency neural network output.

$$\sigma(\eta) = \frac{c}{1 + \exp(-k(\eta - x_0))} + y_0. \quad (2)$$

Phase 3) Leveraging the resultant noise-performance curve, SSRR regresses a reward function of trajectory states and actions, as shown in Equation 3. Here, R_θ represents parameters of our idealized reward function. After obtaining an accurate reward function, SSRR applies RL to obtain a policy π^* that outperforms the suboptimal demonstrations.

$$L_{\text{SSRR}}(\theta) = \mathbb{E}_{\tau^i} \left[\left(\left(\sum_{t=0}^{T} R_\theta(s_t^i, a_t^i) \right) - \sigma(\eta^i) \right)^2 \right] \quad (3)$$

Method

In this section, we present *systematic* alternatives to self-supervised trajectory generation via noise-injection: 1) Reduction of demonstrations available to the initial imitation learning framework, 2) Reduction of model capacity within the imitation learning framework, and 3) Increase of policy network weight sparsity during the imitation learning. Relying on a *random* approach such as noise-injection to generate trajectories has two key drawbacks: 1) Performance degradation could be random depending on whether the noise is applied to vital states among the trajectory and 2) Degraded trajectories via noise-injection may not accurately represent the entire scope of suboptimality.
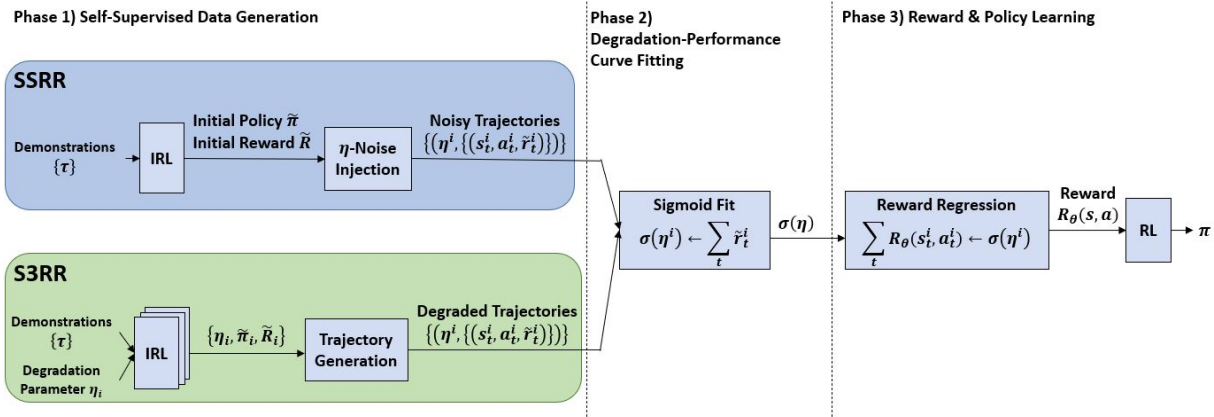
Figure 1: This figure depicts comparison between SSRR and S3RR algorithm pipeline. (Top) SSRR (Bottom) S3RR.

As discussed in the Preliminaries Section, the SSRR pipeline has three phases consisting of self-supervised data generation, noise-performance curve fitting, and reward & policy learning. In our extension of SSRR, S3RR, *Phase 1* is modified to utilize different performance-degradation methodologies. In this way, instead of learning one initial policy via AIRL and injecting noise to produce performance-degraded trajectories, we learn several AIRL policies, each representing a different level of degradation. A visualization of the modified Phase 1 procedure (representing all three proposed performance-degradation methodologies) of S3RR is shown in Figure 1 bottom. We detail the three different degradation methodologies below:

Reduction of demonstrations available to AIRL

In our first variant, we alter the number of suboptimal demos available to AIRL in each level of degradation. As machine-learning approaches typically benefit from increased numbers of data, a reduction in the amount of available data should lead to a decrease in performance (Hlynsson., Escalante-B., and Wiskott. 2019; Althnian et al. 2021). As such, we learn several policies, each learning from a varying number of suboptimal demos. Each policy is then used to generate performance-degraded trajectories. The generated degradation dataset is then fed into Phase 2 following the SSRR framework to infer sigmoid parameters to fit the degradation-performance curve.

Reduction of Model Capacity within AIRL

In our second variant, we alter the model capacity available to the policy and discriminator networks within AIRL. More specifically, we modify the network sizes to control the model capacities. As highly parameterized networks are typically needed to learn a complete representation of the user's behavior, the performance of machine-learning approaches suffer when the models are under-parameterized (Frankle and Carbin 2019). As such, we learn several policies, each varying in the number of parameters for the AIRL networks. Each learned policy can then be used to generate performance-degraded trajectories, which can then be used

in Phase 2 following the SSRR framework to infer sigmoid parameters to fit the degradation-performance curve.

Increase of Network Weight Sparsity within AIRL

In our third variant, we alter the amount of weight regularization used during training the AIRL policy to control weight sparsity. An increase of L1 regularization (by increasing λ in Equation 4) during training will force the policy network to be sparse (Ma et al. 2019), resulting in a lower performing policy. As such, we learn several policies, each varying in the amount of regularization used during AIRL. Each learned policy can then be used to generate performance-degraded trajectories, which can then be used in Phase 2 following the original SSRR framework to infer sigmoid parameters to fit the degradation-performance curve. Equation 4 shows the policy gradient objective with the added L1 regularization.

$$\phi^* = \arg\max \mathbb{E}_{\tau \sim \pi_\phi}[R(\tau) - \lambda||\phi||_1] \qquad (4)$$

Results

In this section, we show preliminary results with S3RR with two MuJoCo virtual robotic control tasks (Todorov, Erez, and Tassa 2012): HalfCheetah-v3 and Ant-v3, which are common benchmarks in RL and LfD literature. We present both the correlation with the ground-truth reward (in Table 1) and the performance of the learned policies (in Table 2). Each model within S3RR is trained by modifying the respective performance degradation variable. For the reduction of demonstrations available to AIRL, we utilize one, three, six, and ten available demos for training the respective phase 1 policies, with corresponding degradation level $\eta = [1.0, 0.78, 0.44, 0.0]$ by mapping [1,3,6,10] to degradation levels between 1.0 and 0.0. For the reduction of network size within AIRL, we vary the number of layers within the AIRL policy and discriminator network to be one, two, three, four, five for training the respective phase 1 policies with corresponding degradation level $\eta = [1.0, 0.75, 0.5, 0.25, 0.0]$. We keep the number of hidden nodes per layer fixed and choose four nodes for HalfCheetah and eight nodes for Ant on each layer. For the

Table 1: Learned Reward Correlation Coefficients with Ground-Truth Reward Comparison between S3RR variants and SSRR. Reported results are Mean ± Std. Dev. across five trials. Bold results represent the largest correlation within each domain.

Domain	*S3RR* *Reduction of Demos*	*S3RR* *Model Capacity*	*S3RR* *Weight Sparsity*	SSRR Noisy-AIRL	SSRR AIRL
HalfCheetah-v3	0.936 ± 0.048	0.977 ± 0.014	$\mathbf{0.996 \pm 0.001}$	0.941 ± 0.025	0.917 ± 0.017
Ant-v3	0.941 ± 0.021	0.816 ± 0.109	0.741 ± 0.081	$\mathbf{0.970 \pm 0.006}$	0.615 ± 0.024

Table 2: Ground-Truth Reward of Policy Inferred via S3RR and SSRR. Best correlation reward functions are chosen to optimize via RL out of the trials of S3RR and SSRR. Bold results represent the highest performing policies in each domain.

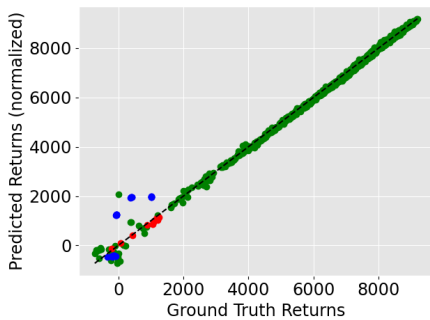| Domain	Demonstration		*S3RR* *Reduction of Demos*	*S3RR* *Model Capacity*	*S3RR* *Weight Sparsity*	SSRR Noisy-AIRL
	Average	Best				
HalfCheetah-v3	803.38	1246.22	572.82 (46%)	4968.29 (398.67%)	**8021.58 (644%)**	2853 (229%)
Ant-v3	1297.59	1559.44	955.94 (61%)	943.91 (61%)	951.52 (61%)	**3944 (253%)**

Figure 2: Reward correlation with Ground-Truth reward in HalfCheetah. Red dots represent demonstrations given to AIRL, blue dots represent synthetic degradation data, green dots are unseen "test" trajectories, and the dotted line reflects perfect correlation. S3RR returns are normalized to the range of ground-truth returns.

increase of network regularization within AIRL, we utilize L1 regularization coefficients of $100, 10, 1, .1, .01$ in training the respective phase 1 policies, with the corresponding degradation level $\eta = [1.0, 0.75, 0.5, 0.25, 0.0]$.

In Table 1, we present the learned reward correlation coefficients with the ground-truth reward across the proposed S3RR variants and prior work, SSRR. Specifically, we benchmark against two variants of SSRR, namely SSRR-AIRL and SSRR-Noisy-AIRL, depicting different initial AIRL training methods (See Chen, Paleja, and Gombolay (2020) for a detailed comparison between AIRL and Noisy-AIRL). As a comparison, SSRR has 21 noise levels equally spaced between 0 and 1. Across two robotic control domains, HalfCheetah and Ant, we see three variants of S3RR achieve higher correlation compared with SSRR-AIRL, illustrating the benefit of *systematic* self-supervised trajectory generation. Comparing with the previous state-of-the-art SSRR-Noisy-AIRL, S3RR achieves better correlation on HalfCheetah and comparable performance on Ant. We hypothesize S3RR could benefit from the Noisy-AIRL intuition by combining the critic training across AIRL runs with different degradation parameters, which creates a more ro-

bust discriminator that can generalize to various state spaces generated by different performing policies. S3RR could also benefit from a recent work on end-to-end SSRR (Cui et al. 2021). We leave the exploration for future work. We present a depiction of the correlation between S3RR and the ground-truth reward function in HalfCheetah in Figure 2. In general, we argue there is likely no one degradation method that fits well (Wolpert and Macready 1997), let alone the best, to the data in every domain, as observed in our results. As shown in Table 1, S3RR Weight Sparsity works the best for HalfCheetah, but is the worst for Ant. We find AIRL policy performs well with regularization coefficients of 1 or 100, while with regularization coefficient of 10, the AIRL policy acts randomly. To that end, our paper proposes a general modification to SSRR that opens the possibility to discover effective degradation processes for broader domains to achieve learning from suboptimal demonstration, despite higher computation needs linear to the number of degradation levels.

In Table 2, we present the performance of the policies learned via S3RR and SSRR. The success of reward learning in S3RR with weight sparsity in HalfCheetah is transferred to the policy, yielding a policy performing 544% better than the best demonstration and 415% better than SSRR-Noisy-AIRL. Besides, S3RR with model capacity also achieves better performance than SSRR-Noisy-AIRL. However, policy learning with S3RR on Ant fails drastically, possibly due to the lower reward correlation compared with SSRR-Noisy-AIRL. In future work, We will conduct further analysis to find the cause of the low policy performance.

Conclusion

In this work, we present *Systematic* Self-Supervised Reward Regression, S3RR, to investigate systematic alternatives for trajectory degradation. As alternatives in data generation in imitation learning, we explore 1) a reduction of demonstrations available, 2) a reduction of model learner capacity, and 3) an increase of network weight sparsity. We find that S3RR can achieve higher reward correlation with ground-truth and produce policies performing 544% better than the best suboptimal demonstration and 415% better than the previous state-of-the-art SSRR-Noisy-AIRL.

Acknowledgments

References

Abbeel, P.; and Ng, A. Y. 2004. Apprenticeship Learning via Inverse Reinforcement Learning. In *Proceedings of the International Conference on Machine Learning (ICML)*. ACM.

Althnian, A.; AlSaeed, D.; Al-Baity, H.; Samha, A.; Dris, A. B.; Alzakari, N.; Abou Elwafa, A.; and Kurdi, H. 2021. Impact of Dataset Size on Classification Performance: An Empirical Evaluation in the Medical Domain. *Applied Sciences* 11(2): 796.

Bhattacharjee, T.; Lee, G.; Song, H.; and Srinivasa, S. S. 2019. Towards Robotic Feeding: Role of Haptics in Fork-Based Food Manipulation. *IEEE Robotics and Automation Letters (RA-L)* 4(2): 1485–1492.

Brown, D. S.; Goo, W.; and Niekum, S. 2019. Better-than-Demonstrator Imitation Learning via Automatically-Ranked Demonstrations. In *Proceedings of the Conference on Robot Learning (CoRL)*.

Chen, L.; Paleja, R. R.; Ghuy, M.; and Gombolay, M. C. 2020. Joint Goal and Strategy Inference across Heterogeneous Demonstrators via Reward Network Distillation. In *Proceedings of the International Conference on Human-Robot Interaction (HRI)*.

Chen, L.; Paleja, R. R.; and Gombolay, M. 2020. Learning from Suboptimal Demonstration via Self-Supervised Reward Regression. *ArXiv* abs/2010.11723.

Chernova, S.; and Thomaz, A. L. 2014. *Robot Learning from Human Teachers*. Morgan & Claypool Publishers.

Cui, Y.; Liu, B.; Saran, A.; Giguere, S.; Stone, P.; and Niekum, S. 2021. Aux-AIRL: End-to-End Self-Supervised Reward Learning for Extrapolating beyond Suboptimal Demonstrations. In *Proceedings of the ICML Workshop on Self-Supervised Learning for Reasoning and Perception*.

Frankle, J.; and Carbin, M. 2019. The Lottery Ticket Hypothesis: Finding Sparse, Trainable Neural Networks. In *Proceedings of International Conference on Learning Representations (ICLR)*.

Fu, J.; Luo, K.; and Levine, S. 2018. Learning Robust Rewards with Adverserial Inverse Reinforcement Learning. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Gombolay, M.; Jensen, R.; Stigile, J.; Son, S.-H.; and Shah, J. 2016. Apprenticeship Scheduling: Learning to Schedule from Human Experts. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*.

Gombolay, M.; Yang, X. J.; Hayes, B.; Seo, N.; Liu, Z.; Wadhwania, S.; Yu, T.; Shah, N.; Golen, T.; and Shah, J. 2018. Robotic assistance in the coordination of patient care. *The International Journal of Robotics Research* 37(10): 1300–1316.

Hlynsson., H.; Escalante-B., A.; and Wiskott., L. 2019. Measuring the Data Efficiency of Deep Learning Methods. In *Proceedings of the International Conference on Pattern Recognition Applications and Methods (ICPRAM)*, 691–698. SciTePress.

Ma, R.; Miao, J.; Niu, L.; and Zhang, P. 2019. Transformed l1 regularization for learning sparse deep neural networks. *Neural Networks* 119: 286–298.

Newell, A.; and Simon, H. A. 1972. *Human problem solving*. 9. Prentice-Hall Englewood Cliffs, NJ.

Ng, A. Y.; Russell, S. J.; et al. 2000. Algorithms for inverse reinforcement learning. In *Proceedings of the International Conference on Machine Learning (ICML)*, volume 1, 2.

Nikolaidis, S.; Nath, S.; Procaccia, A. D.; and Srinivasa, S. 2017. Game-Theoretic Modeling of Human Adaptation in Human-Robot Collaboration. In *Proceedings of the International Conference on Human-Robot Interaction (HRI)*. ACM/IEEE.

Paleja, R. R.; Silva, A.; Chen, L.; and Gombolay, M. 2020. Interpretable and Personalized Apprenticeship Scheduling: Learning Interpretable Scheduling Policies from Heterogeneous User Demonstrations. In *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*.

Pan, Y.; Cheng, C.-A.; Saigol, K.; Lee, K.; Yan, X.; Theodorou, E.; and Boots, B. 2018. Agile autonomous driving using end-to-end deep imitation learning. In *Proceedings of Robotics: Science and Systems (RSS)*.

Ramachandran, D.; and Amir, E. 2007. Bayesian Inverse Reinforcement Learning. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, 2586–2591. Morgan Kaufmann Publishers Inc.

Ross, S.; Gordon, G.; and Bagnell, D. 2011. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*.

Seraj, E.; and Gombolay, M. 2020. Coordinated control of uavs for human-centered active sensing of wildfires. In *Proceedings of American Control Conference (ACC)*, 1845–1852. IEEE.

Todorov, E.; Erez, T.; and Tassa, Y. 2012. MuJoCo: A physics engine for model-based control. In *Proceedings of the International Conference on Intelligent Robots and Systems (IROS)*. IEEE/RSJ.

Wang, Z.; and Gombolay, M. 2020. Heterogeneous Graph Attention Networks for Scalable Multi-Robot Scheduling with Temporospatial Constraints. In *Proceedings of Robotics: Science and Systems (RSS)*.

White, D. J. 1993. A survey of applications of Markov decision processes. *Journal of the Operational Research Society* 44(11): 1073–1096.

Wirth, C.; Akrour, R.; Neumann, G.; and Fürnkranz, J. 2017. A Survey of Preference-Based Reinforcement Learning Methods. *Journal of Machine Learning Research* 18: 1–46.

Wolpert, D.; and Macready, W. 1997. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation* 1(1): 67–82.

Ziebart, B. D. 2010. *Modeling purposeful adaptive behavior with the principle of maximum causal entropy*. Ph.D. thesis.

Ziebart, B. D.; Maas, A.; Bagnell, J. A.; and Dey, A. K. 2008. Maximum entropy inverse reinforcement learning. In *Proceedings of the National Conference on Artificial intelligence (AAAI)*, 1433–1438.