A Benchmark Comparison of Imitation Learning-based Control Policies for Autonomous Racing

Xiatao Sun, Mingyan Zhou, Zhijun Zhuang, Shuo Yang, Johannes Betz, Rahul Mangharam

2023 IEEE Intelligent Vehicles Symposium (IV) | 979-8-3503-4691-6/23/$31.00 ©2023 IEEE | DOI: 10.1109/IV55152.2023.10186780

***Abstract*— Autonomous racing with scaled race cars has gained increasing attention as an effective approach for developing perception, planning and control algorithms for safe autonomous driving at the limits of the vehicle's handling. To train agile control policies for autonomous racing, learning-based approaches largely utilize reinforcement learning, albeit with mixed results. In this study, we benchmark a variety of imitation learning policies for racing vehicles that are applied directly or for bootstrapping reinforcement learning both in simulation and on scaled real-world environments. We show that interactive imitation learning techniques outperform traditional imitation learning methods and can greatly improve the performance of reinforcement learning policies by bootstrapping thanks to its better sample efficiency. Our benchmarks provide a foundation for future research on autonomous racing using Imitation Learning and Reinforcement Learning.**

I. INTRODUCTION

A. Motivation

In motorsport racing, it all boils down to the ability of the driver to operate the racecar at its limits. Expert race drivers are extremely proficient in pushing the racecar to its dynamical limits of handling, while accounting for changes in the vehicle's interaction with the environment to overtake competitors at speeds exceeding 300 km/h. Autonomous racing emphasizes driving vehicles autonomously with high performance in racing conditions, which usually involves high speeds, low reaction times, operating at the limits of vehicle dynamics, and constantly balancing safety and performance [1]. While the goal of autonomous racing is to outperform human drivers through the development of perception, planning and control algorithms, the performance with learning-based approaches is still far from parity. The goal of this paper is to benchmark a variety of imitation learning (IL) approaches that are used directly and for bootstrapping reinforcement learning (RL).

In the past few years, autonomous racing cars at different scales such as Roborace [2], Indy Autonomous Challenge [3], and Formula Student Driverless [4], reduced-scale platforms like F1TENTH [5] have been developed. Reduced-scale platforms with on-board computation and assisted with algorithm development in simulation enable rapid development with lower cost for research and educational purposes.

Autonomous racing has traditionally followed the perception–planning–control modular pipeline. A recent shift towards the end-to-end learning paradigm for autonomous vehicles is showing promise in terms of scaling across common driving scenarios and navigating rare operation contexts [6]. Autonomous racing provides a perfect setting for evaluating end-to-end testing approaches as it clearly specifies the trade-off between safety and performance. However, the difficulties in sim-to-real transfer and ensuring safety remain open for further study [1].

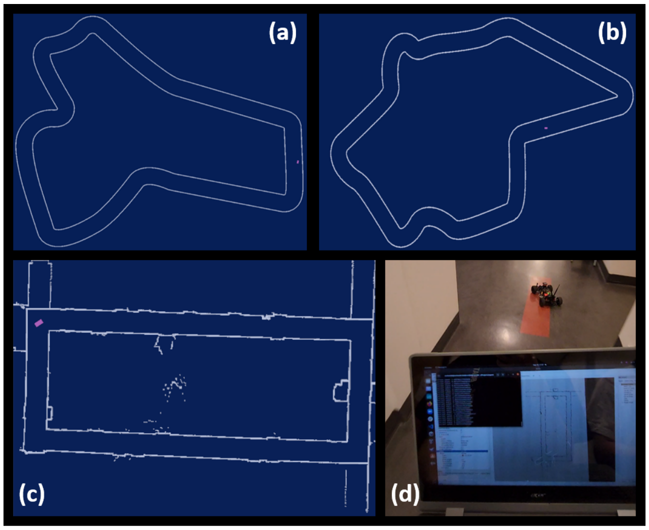

Fig. 1: (a) The training map for evaluations and comparisons in simulation. (b) The unseen map for testing the generalizability of learned policies. (c) The training map for the real-world comparison is generated using LiDAR scans of the real environment. (d) F1TENTH vehicle is controlled by the learned policy.

Among the emerging end-to-end approaches, IL and RL are the most promising. IL essentially trains policies to mimic the given expert demonstration [7]. It is shown to outperform supervised machine learning algorithms since those methods suffer from problems including distribution mismatch among datasets and long-term sequential planning [8]. Based on the innovative algorithm Data Aggregation (DAGGER) [8], some interactive IL methods, such as human-gated DAGGER (HG-DAGGER) [9] and expert intervention learning (EIL) [10], use interactive querying to improve the training process and overall performance.

However, imitation learning-based autonomous racing vehicles is only just getting started [6]. Several recent efforts [11, 12] implemented IL on autonomous racing cars, but only for bootstrapping and simplified tests. This work implements and provides a comprehensive comparison among several IL methods in simulation and on the F1Tenth physical racing platform (`https://f1tenth.org`). By making this available as open-source software, we hope to encourage researchers to further the exploration in learning-based controllers for autonomous driving.

All authors are with the University of Pennsylvania, Department of Electrical and Systems Engineering, 19104, Philadelphia, PA, USA. Emails: {`sxt, derekzmy, zhijunz, yangs1, joebetz, rahulm`}`@seas.upenn.edu`

B. Contributions

In this paper, we tackle the problem of IL-based control for autonomous ground robots that will drive with high-speed and high acceleration. This work has three main contributions:

1) We implement 4 different IL algorithms that learn from expert demonstrations;
2) We display results from both simulation and real-world experiments on a small-scale autonomous racing car;
3) We benchmark different algorithms for both direct learning and bootstrapping.

II. RELATED WORK

A. Autonomous Racing

End-to-end approaches for autonomous driving in general, and autonomous racing in specific, replace partial or whole modules of the modular perception-planning-control autonomous software pipeline with data-driven approaches [1]. For instance, [13] combined non-linear model predictive control (NMPC) and deep neural network (DNN) for the trajectory planning, while [14] utilize model-based RL to test their vision-based planning.

Few studies have explored IL for autonomous racing. Deep imitation learning (DIL) [11] trained the DNN policy with an MPC expert using DAGGER and tested it on AutoRally. Additionally, few studies took IL and RL together into account. Controllable imitative reinforcement learning (CIRL) [15], and deep imitative RL (DIRL) [12] initialized the RL policy network with IL before starting exploration. Still, IL was implemented in basic behavioral cloning (BC) rather than interactive IL methods. [16] and [17] loaded the transitions of demonstration into the replay buffer to lead the RL process [12]. Nonetheless, they still use IL as simple demonstrations, and the methods have not been verified in real-world scenarios.

B. Imitation Learning

BC is the most straightforward IL method. Supervised machine learning is used in BC to train the novice policy with the demonstrated expert policy. One of its first applications in autonomous driving from 1988 is ALVINN [18], which could achieve the vehicle-following task on the road with a vehicle equipped with sensors. BC is easy to understand and simple to implement. However, it suffers from the risk of distribution mismatch, covariate shift [19], and compounding errors [8], making it brittle for autonomous racing.

Studies in imitation learning after BC usually could be categorized into direct policy learning (DPL) and inverse reinforcement learning (IRL). DPL primarily emphasizes learning the policy directly [7], while IRL pays more attention to learning the intrinsic reward function [20]. In this paper, we will mainly discuss DPL.

Data Aggregation (DAGGER) allows the novice to influence the sampling distribution by aggregating the expert-labeled data extensively and updating the policy iteratively, which mitigates BC's drawbacks [8]. However, the data-gathering rollouts are under the complete control of the incompletely trained novice rather than interactively querying the expert, which degrades the quality of the sampling and efficiency of data labeling; this even potentially destabilizes the autonomous system [9].

To deal with the drawbacks of DAGGER, recent developments in interactive IL involved the human-gated method and robot-gated method. The human-gated techniques, e.g., [9, 21], allow the human supervisor to decide the instants to correct the actions, but continuously monitoring the robot will burden the supervisor. The robot-gated method [22] enables the robot to query the human expert for interventions actively but balancing the burden and providing sufficient information is still difficult [23]. In this paper, we only consider the human-gated method since the robot-gated method is unsuitable for our racing conditions with low reaction time. Using the robot-gated method will probably cause the crashing due to high-speed racing and inertia.

By introducing the gating function, human-gated DAGGER [9] allows the human expert to take control when the condition is beyond the safety threshold and to return the control to novice policy under tolerated circumstances. The interactive property reduces the burden of the expert rather than querying the expert all the time and ensures the efficiency of the training process. Expert intervention learning (EIL) [10] proposed further exploration with implicit and explicit feedback beyond HG-DAGGER. It addresses that any amount of expert feedback needs to be considered, whether intervened or not. EIL records the data into three state-actions datasets and added the implicit loss inferred when the expert decides to intervene.

III. METHODOLOGY

The IL algorithms we implement and compare in this work include BC [24], DAGGER [8], HG-DAGGER [9], and EIL [10]. We use a multi-layer perceptron (MLP) as the learner and a pure pursuit algorithm as the expert for all implemented IL algorithms. We are using a 2D simulation environment (F1TENTH gym) developed for the small-scale autonomous race car [25]. The learner takes the LiDAR scan array o_L as input, whereas the pure pursuit expert takes the x and y coordinates and angular direction of the agent θ as input. Both the learner and expert output steering angle and speed as actions to control the vehicle.

For human-gated IL algorithms that require expert intervention, which are HG-DAGGER and EIL, we define two intervention thresholds γ_v and γ_ω for speed v and steering angle ω, respectively. To mimic the expert intervention using the pure pursuit algorithm, both the learner policy and the pure pursuit output their action based on their observation separately at every step. The pure pursuit will take over the control and provides expert demonstrations whenever the difference of action between the learner policy and the pure pursuit exceeds either of γ_v or γ_ω.

Considering the prominence of RL in learning-based methods for autonomous racing and the potential of IL as a

TABLE I: Evaluations of different learned policies in an unseen simulation environment.

Method	BC	DAGGER	HG-DAGGER	EIL	PPO	BC+PPO	DAGGER+PPO	HG-DAGGER+PPO	EIL+PPO
Distance Traveled (m)	7.84	8.90	12.34	15.89	12.69	151.23	86.49	155.88	150.15
Complete 1 Lap	No	No	No	No	No	Yes	No	Yes	Yes
Bhattacharyya Distance	0.77	0.60	0.12	0.24	1.09	0.59	0.59	0.47	0.43

bootstrapping method for RL, we also implement proximal policy optimization (PPO) [26] to train policies with or without IL bootstrapping to compare the efficiency of various combinations of PPO and different IL algorithms. Before training, the PPO policy can be initialized randomly or bootstrapped using a pre-trained network by IL algorithms with n expert-labeled samples. The pre-trained IL network has the same architecture with the PPO policy. To reduce warbling, avoid crashing and encourage staying close to the center-line of the track, we design a reward function r that incorporates the reward for survival, the penalty for lateral error from the center-line E_l, and penalty for the deviation of steering angle ω as

$$r = -0.02 \cdot \min(1.0, \max(0, \omega)) + \begin{cases} -0.5 & \text{if crashed} \\ -0.02 \cdot E_l & \text{if } E_l > 0.1 \\ 0.02 & \text{otherwise} \end{cases}$$

To transfer the learned policy from simulation to the real world, we add an array of random noise o_R to the LiDAR scan array o_L. o_R and o_L have the same dimension. Each element in o_R is randomly sampled from $[\alpha, \beta]$, where α and β are the lower and upper bounds of the random noise.

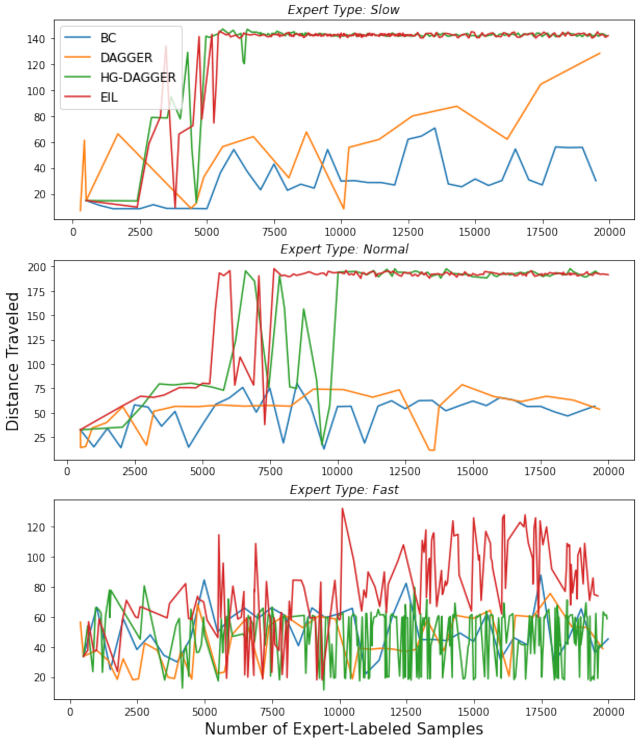

Fig. 2: The distance traveled by each agent with respect to the number of expert-labeled samples.

TABLE II: Elapsed time of IL policies trained with different experts

Expert Type	Expert	BC	DAGGER	HG-DAGGER	EIL
Slow	33.07 s	Failed	34.34 s	33.78 s	33.50 s
Normal	25.04 s	25.35 s	25.85 s	25.06 s	25.22 s
Fast	19.69 s	Failed	Failed	Failed	20.40 s

IV. EXPERIMENTS

A. *Implementation Details*

We implement the IL algorithms using the F1TENTH, a 1/10-scale autonomous racing research platform, for both simulation and real-world scenarios [25]. The maps for training and evaluation in simulation and the real world are shown in Fig. 1. Due to safety considerations, all policies in this work are trained in the F1TENTH gym. We use the same two-layer MLP with 256 hidden units as the learner for all IL algorithms in the comparison. The learning rate is set to 0.001 during training. When training policies using DAGGER, HG-DAGGER, and EIL, the first 500 samples are collected for training initial policies using BC. For HG-DAGGER and EIL, we set the intervention threshold γ_v and γ_ω to 1 and 0.1, respectively. All IL policies are trained using 20k expert-labeled samples. To test the efficiency of bootstrapping, we train different PPO policies for 20k steps. We use IL policies with 3000 expert-labeled samples as the starting point for bootstrapped PPO policies. We let $\alpha = -0.2$ and $\beta = 0.2$ for transferring policies to real world.

B. *Evaluations in Simulations*

During the training of the four IL algorithms, the learned policies are evaluated in terms of distance traveled in the training map after the training of each iteration. We use the number of expert-labeled samples as the independent variable to assess and compare the sample efficiency of different IL algorithms for three reasons. First, the major downside of IL, in general, is its requirement of expert effort with labeling. Moreover, the number of steps in each iteration is not fixed and is uncontrollable for all algorithms except BC. Lastly, implicit samples in EIL are collected at no cost, which makes it unfair to compare with other algorithms in terms of the total number of samples.

As shown in Fig. 2, to further test the ability to imitate the expert's behavior for each IL algorithm, we train different policies using the demonstration of the pure pursuit expert with different speeds. The slow, normal, and fast experts are with an average speed of 4.79 $\mathrm{m/s}$, 6.39 $\mathrm{m/s}$, and 8.24 $\mathrm{m/s}$ respectively. Overall, those IL algorithms with expert intervention, i.e., HG-DAGGER and EIL, have better sample

TABLE III: Evaluations of different learned policies in the real-world environment.

Method	Expert	BC	DAGGER	HG-DAGGER	EIL	PPO	BC+PPO	DAGGER+PPO	HG-DAGGER+PPO	EIL+PPO
Distance Traveled (m)	61.44	6.44	8.49	37.74	38.04	5.27	6.44	6.29	64.08	39.5
Complete 1 Lap	Yes	No	No	No	No	No	No	No	Yes	No

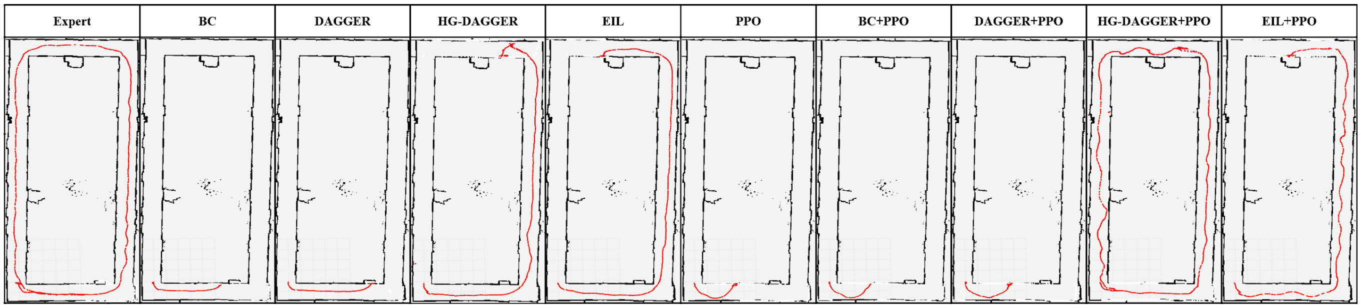

Fig. 3: The movement trajectories of the F1TENTH vehicle on the map under the control of each policy.

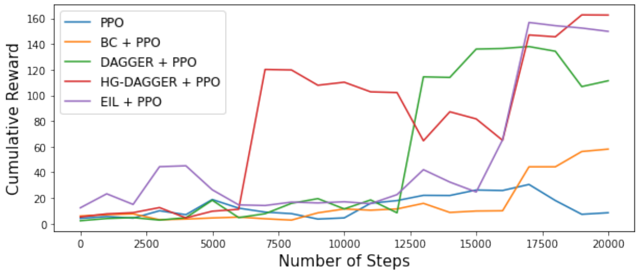

Fig. 4: The cumulative reward with respect to the number of steps during training PPO policies with or without IL bootstrapping.

efficiency since their learned policies travel significantly longer distances than BC and DAGGER.

Although all IL algorithms struggle to learn when using the demonstrations from the fast expert, as shown in Table II, EIL is the only one that can complete one lap. Table II also indicates that the upper limit of the performance of policies learned using IL is from the expert.

Since IL can be combined with RL, we test the bootstrapping efficiency of different IL algorithms for PPO at normal speed. As Fig. 4 suggests, using any IL algorithms can help PPO converge to a better policy as it no longer needs to start from a random policy. DAGGER, HG-DAGGER, and EIL demonstrate considerably better bootstrapping efficiency than BC, thanks to their better sample efficiency.

To evaluate how well the policies generalize, we generate a new (unseen) map, as shown in Fig. 1(b), and perform inference using the policies at normal speed. For each policy, we record the distance traveled and whether it completes one lap or not to evaluate its performance. We also calculate the Bhattacharyya distance [27] for the decision of an expert at every step to evaluate the similarity between learned behaviors and expert behavior. As Table I shows, the combinations of IL and PPO significantly outperform policies only using IL or PPO, with the PPO trained with EIL bootstrapping having the best performance. This indicates that combining IL and RL can efficiently converge to a more generalized policy. Additionally, interactive IL can train policies that are more similar to expert behavior than non-interactive IL and PPO.

C. Evaluations in Real-World Environments

All policies for real-world experiments are at 3 $\mathrm{m/s}$. Fig. 3 shows the results of real-world experiments. For both direct training and bootstrapping, interactive IL methods, which are HG-DAGGER and EIL, can train policies that travel considerably further distances compared with policies from non-interactive methods, which are BC and DAGGER. BC and DAGGER barely help when bootstrapping PPO in real-world experiments. The combination of HG-DAGGER and PPO has the best performance and is the only policy that completes one lap in the real world. Despite incorporating the penalty on steering angle in the reward function, PPO policies have more warbling in their trajectories compared with IL policies, which might result from the difference in floor friction between the gym environment and the real world. The real-world experiments further validate that combining IL and RL yields the best result.

V. CONCLUSION

In this work, we implement four different IL algorithms on the F1TENTH platform to benchmark their performance in the context of autonomous racing. Our experiments show that IL algorithms can train or bootstrap high-performance policies for autonomous racing scenarios. Recent development in interactive IL significantly improves the sample efficiency of policies for autonomous racing. The combination of RL and interactive IL can get the best of both worlds: fast convergence and better generalizability. The interactive imitation learning methods outperform non-interactive methods for both learning directly and bootstrapping due to their improved sample efficiency. Our IL implementations provide a foundation for future research on autonomous racing using IL and RL. Future work will focus on safe human-gated methods for multi-agent autonomous racing, utilization of new network architecture, and better simulation environments to further reduce the sim-to-real gap.

References

[1] J. Betz, H. Zheng, A. Liniger, U. Rosolia, P. Karle, M. Behl, V. Krovi, and R. Mangharam, "Autonomous vehicles on the edge: A survey on autonomous vehicle racing," *IEEE Open Journal of Intelligent Transportation Systems*, 2022.

[2] J. Rieber, H. Wehlan, and F. Allgower, "The roborace contest," *IEEE Control Systems Magazine*, vol. 24, no. 5, pp. 57–60, 2004.

[3] A. Wischnewski, M. Geisslinger, J. Betz, T. Betz, F. Fent, A. Heilmeier, L. Hermansdorfer, T. Herrmann, S. Huch, P. Karle, F. Nobis, L. Ögretmen, M. Rowold, F. Sauerbeck, T. Stahl, R. Trauth, M. Lienkamp, and B. Lohmann, "Indy autonomous challenge - autonomous race cars at the handling limits," in *12th International Munich Chassis Symposium 2021*, P. Pfeffer, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2022, pp. 163–182.

[4] M. Zeilinger, R. Hauk, M. Bader, and A. Hofmann, "Design of an autonomous race car for the formula student driverless (FSD)," 2017.

[5] M. OKelly, H. Zheng, D. Karthik, and R. Mangharam, "F1tenth: An open-source evaluation environment for continuous control and reinforcement learning," in *Proceedings of the NeurIPS 2019 Competition and Demonstration Track*, ser. Proceedings of Machine Learning Research, vol. 123. PMLR, 2020, pp. 77–89. [Online]. Available: http://proceedings.mlr.press/v123/o-kelly20a.html

[6] L. Le Mero, D. Yi, M. Dianati, and A. Mouzakitis, "A survey on imitation learning techniques for end-to-end autonomous vehicles," *IEEE Transactions on Intelligent Transportation Systems*, 2022.

[7] A. Hussein, M. M. Gaber, E. Elyan, and C. Jayne, "Imitation learning: A survey of learning methods," *ACM Computing Surveys (CSUR)*, vol. 50, no. 2, pp. 1–35, 2017.

[8] S. Ross, G. Gordon, and D. 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*. JMLR Workshop and Conference Proceedings, 2011, pp. 627–635.

[9] M. Kelly, C. Sidrane, K. Driggs-Campbell, and M. J. Kochenderfer, "Hg-dagger: Interactive imitation learning with human experts," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 8077–8083.

[10] J. Spencer, S. Choudhury, M. Barnes, M. Schmittle, M. Chiang, P. Ramadge, and S. Srinivasa, "Expert intervention learning," *Autonomous Robots*, vol. 46, no. 1, pp. 99–113, 2022.

[11] Y. Pan, C.-A. Cheng, K. Saigol, K. Lee, X. Yan, E. Theodorou, and B. Boots, "Agile autonomous driving using end-to-end deep imitation learning," in *Robotics: Science and Systems XIV*. Robotics: Science and Systems Foundation, June 2018. [Online]. Available: https://doi.org/10.15607/rss.2018.xiv.056

[12] P. Cai, H. Wang, H. Huang, Y. Liu, and M. Liu, "Vision-based autonomous car racing using deep imitative reinforcement learning," *IEEE Robotics and Automation Letters*, pp. 1–1, 2021.

[13] A. Tătulea-Codrean, T. Mariani, and S. Engell, "Design and simulation of a machine-learning and model predictive control approach to autonomous race driving for the f1/10 platform," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 6031–6036, 2020. [Online]. Available: https://doi.org/10.1016/j.ifacol.2020.12.1669

[14] E. Chisari, A. Liniger, A. Rupenyan, L. V. Gool, and J. Lygeros, "Learning from simulation, racing in reality," *CoRR*, vol. abs/2011.13332, 2020. [Online]. Available: https://arxiv.org/abs/2011.13332

[15] X. Liang, T. Wang, L. Yang, and E. Xing, "Cirl: Controllable imitative reinforcement learning for vision-based self-driving," in *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

[16] Q. Zou, K. Xiong, and Y. Hou, "An end-to-end learning of driving strategies based on ddpg and imitation learning," in *2020 Chinese Control And Decision Conference (CCDC)*, 2020, pp. 3190–3195.

[17] M. Vecerík, T. Hester, J. Scholz, F. Wang, O. Pietquin, B. Piot, N. Heess, T. Rothörl, T. Lampe, and M. A. Riedmiller, "Leveraging demonstrations for deep reinforcement learning on robotics problems with sparse rewards," *CoRR*, vol. abs/1707.08817, 2017. [Online]. Available: http://arxiv.org/abs/1707.08817

[18] D. A. Pomerleau, "Alvinn: An autonomous land vehicle in a neural network," in *Advances in Neural Information Processing Systems*, D. Touretzky, Ed., vol. 1. Morgan-Kaufmann, 1988. [Online]. Available: https://proceedings.neurips.cc/paper/1988/file/812b4ba287f5ee0bc9d43bbf5bbe87fb-Paper.pdf

[19] B. D. Argall, S. Chernova, M. Veloso, and B. Browning, "A survey of robot learning from demonstration," *Robotics and Autonomous Systems*, vol. 57, no. 5, pp. 469–483, 2009. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0921889008001772

[20] S. Arora and P. Doshi, "A survey of inverse reinforcement learning: Challenges, methods and progress," *Artificial Intelligence*, vol. 297, p. 103500, Aug. 2021. [Online]. Available: https://doi.org/10.1016/j.artint.2021.103500

[21] J. Spencer, S. Choudhury, M. Barnes, M. Schmittle, M. Chiang, P. Ramadge, and S. Srinivasa, "Learning from interventions: Human-robot interaction as both explicit and implicit feedback," in *Robotics*, ser. Robotics: Science and Systems, M. Toussaint, A. Bicchi, and T. Hermans, Eds. United States: MIT Press Journals, 2020.

[22] J. Zhang and K. Cho, "Query-efficient imitation learning for end-to-end autonomous driving," *CoRR*, vol. abs/1605.06450, 2016. [Online]. Available: http://arxiv.org/abs/1605.06450

[23] R. Hoque, A. Balakrishna, E. R. Novoseller, A. Wilcox, D. S. Brown, and K. Goldberg, "Thriftydagger: Budget-aware novelty and risk gating for interactive imitation learning," *CoRR*, vol. abs/2109.08273, 2021. [Online]. Available: https://arxiv.org/abs/2109.08273

[24] C. Sammut, *Behavioral Cloning*. Boston, MA: Springer US, 2010, pp. 93–97. [Online]. Available: https://doi.org/10.1007/978-0-387-30164-8_69

[25] J. Betz, H. Zheng, Z. Zang, F. Sauerbeck, K. Walas, V. Dimitrov, M. Behl, R. Zheng, J. Biswas, V. Krovi, and R. Mangharam, "Teaching autonomous systems hands-on: Leveraging modular small-scale hardware in the robotics classroom," 2022.

[26] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[27] K. Fukunaga, "Feature extraction and linear mapping for classification," *Introduction to statistical pattern recognition*, pp. 441–507, 1990.