# Evaluating the Effectiveness of Corrective Demonstrations and a Low-Cost Sensor for Dexterous Manipulation

Abhineet Jain*
*Georgia Institute of Technology*
Atlanta, USA
abhineetjain@gatech.edu

Jack Kolb*
*Georgia Institute of Technology*
Atlanta, USA
kolb@gatech.edu

J.M. Abbess IV
*Georgia Institute of Technology*
Atlanta, USA
jabbess3@gatech.edu

Harish Ravichandar
*Georgia Institute of Technology*
Atlanta, USA
harish.ravichandar@cc.gatech.edu

***Abstract*—Imitation learning is a promising approach to help robots acquire dexterous manipulation capabilities without the need for a carefully-designed reward or a significant computational effort. However, existing imitation learning approaches require sophisticated data collection infrastructure and struggle to generalize beyond the training distribution. One way to address this limitation is to gather additional data that better represents the full operating conditions. In this work, we investigate characteristics of such additional demonstrations and their impact on performance. Specifically, we study the effects of *corrective* and *randomly-sampled* additional demonstrations on learning a policy that guides a five-fingered robot hand through a pick-and-place task. Our results suggest that corrective demonstrations considerably outperform randomly-sampled demonstrations, when the proportion of additional demonstrations sampled from the full task distribution is larger than the number of original demonstrations sampled from a restrictive training distribution. Conversely, when the number of original demonstrations are higher than that of additional demonstrations, we find no significant differences between corrective and randomly-sampled additional demonstrations. These results provide insights into the inherent trade-off between the effort required to collect corrective demonstrations and their relative benefits over randomly-sampled demonstrations. Additionally, we show that inexpensive vision-based sensors, such as LeapMotion, can be used to dramatically reduce the cost of providing demonstrations for dexterous manipulation tasks.**

***Index Terms*—learning from demonstrations, reinforcement learning, dexterous manipulation**

## I. Introduction

Dexterous manipulation often involves the use of high degree-of-freedom robots to manipulate objects. Representative dexterous manipulation tasks include relocating objects, picking up arbitrarily shaped objects, and sequential interactions with articulated objects (e.g. unlatching and opening a door). Indeed, factors such as high-dimensional state space and complex interaction dynamics make these tasks particularly challenging to automate. Classical control methods, that have proven valuable for a variety of manipulation problems, are hard to recruit for dexterous manipulation due to the amount of manual effort required to design controllers in high-dimensional spaces.

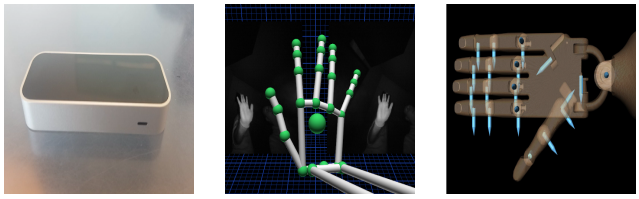

Fig. 1: Left to Right: LeapMotion sensor, Visualization of human hand via LeapMotion, Adroit robot hand simulation in MuJoCo.

Prior work has found success in dexterous manipulation by using self-supervised learning in simulation and transferring learned policies to real robots [1]. Others have utilized demonstrations to improve reinforcement learning [2]. However, these approaches either do not generalize well beyond small workspaces covered during training, or require long training times for highly-specialized tasks.

In this paper, we explore how additional demonstrations that represent the full operating conditions affect performance of the relocation task (described in Fig. 2) in the full operating space. Approaches like demonstration-augmented policy gradient (DAPG) use demonstrations for randomly-sampled start and goal states from a restrictive training distribution. However, we find this approach does not generalize well to task instances in the full operating space. We investigate the effects of corrective versus randomly-sampled additional demonstrations, expecting that corrective demonstrations to areas of policy failure can benefit performance more than randomly-sampled demonstrations.

*These authors contributed equally to this work.

In addition, we explore the feasibility of using a vision-based sensor to record demonstrations for dexterous manipulation tasks. For everyday robots to learn novel manipulation tasks from non-technical humans, it will be useful to reduce the hardware cost of providing demonstrations. The original DAPG paper uses a CyberGlove III [3] wearable sensor glove to record human hand trajectories. However, the CyberGlove III costs around US$13,000. We instead leverage LeapMotion, a US$90 sensor which uses stereo vision to track hand joint positions, and compare the performance of resulting DAPG policies. Our findings validate that the granularity of vision-based joint tracking is effective for training high-performing policies for the relocation task.

## II. Related Work

Previous works explore learning dexterous manipulation from demonstrations, using deep neural networks [4] or reinforcement learning techniques [2] [5] [6] to learn task-specific policies. Some contributions learn actuator outputs from raw color and depth information using deep neural networks [4], while others use demonstrations to aid trajectory exploration and learn system dynamics by utilizing linear regression with GMM priors [5]. DAPG uses demonstrations to pre-train a policy and then applies reinforcement learning to improve it. Another contribution aims to learn from suboptimal experts, introducing a policy iteration algorithm called Relative Entropy Q-Learning (REQ), along with an effective exploration technique that intertwines the policy's actions with the demonstrations [6].

A state-of-the-art for solving a Rubik's cube with a robot hand [1] uses automatic domain randomization to generate a distribution of environments to train on. This work uses meta-learning utilizing LSTMs and the asymmetric actor-critic algorithm for training. While the authors claim that their approach transfers well onto real robots, their algorithm is time-intensive, and uses highly engineered reward functions and specialized robot hardware. These characteristics make their work difficult to reproduce and utilize for general dexterous manipulation tasks.

In our work, we build upon DAPG which uses a small number of expert demonstrations to train policies. However, DAPG policies struggle to generalize beyond the demonstration area. To address this, related work has explored autonomously learning the reward function for tasks instead of human-engineering them [7]. Although their work claims to learn better reward functions than GAIL [8] and Adversarial IRL [9], their learned reward function does not improve upon the performance of the original paper on their set of dexterous manipulation tasks. We approach improving DAPG through a human-robot interaction perspective – when access to the full operating space is granted for recording demonstrations, we explore whether strategic demo collection can improve policy performance.

Our work is inspired by DAgger [10], a classical learning from demonstration algorithm that integrates online corrections to mitigate the covariate shift between training and testing. A recent work explores mitigating covariate shift by measuring the induced covariate shift and directly adjusting for it [11]. It uses expert demonstrations that potentially visit all states that the policy will visit, making it infeasible for the dexterous manipulation case. Another work utilizes data augmentation for generating more sample trajectories, along with a correction neural network to preserve the success of these augmented trajectories for adversarial imitation [12]. Our approach aims to utilize the benefits of DAgger without the disadvantages of online corrections. We collect multiple corrective demonstrations at once and retrain using the combined set of original and corrective demonstrations.

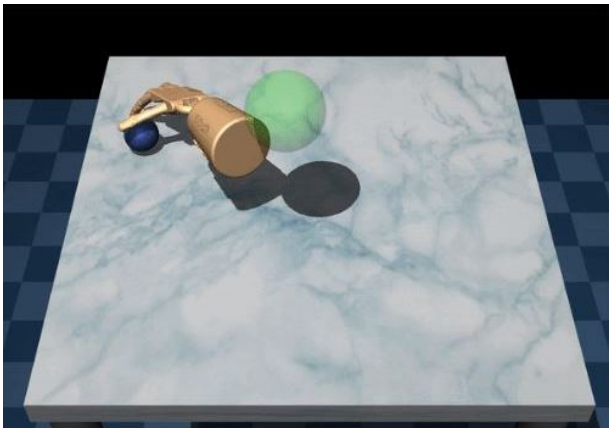

Fig. 2: The relocation task in MuJoCo. This task requires the robot hand to pick up the blue ball from the tabletop and carry it to the green goal region.

## III. Demo-Augmented Policy Gradient (DAPG)

DAPG learns dexterous manipulation via Deep-RL by combining demonstration-based learning with reinforcement learning. DAPG initially trains a policy using behavior cloning, and then tunes the policy using natural policy gradient with a modified loss term, weighing more on the RL policy with increased training iterations. DAPG's loss term penalizes trajectories that are further from the demonstrations. This exploration learns smooth human-like trajectories from a small number of demonstrations, making the algorithm feasible for real world applications.

The original DAPG paper uses expert demonstrations captured using a CyberGlove III [3], a wearable glove embedded with 18-22 sensors. The authors find that their approach results in a high accuracy on four different dexterous manipulation tasks: relocate a ball, open a door, hammer a nail, and orient a pen in-hand. For each of these tasks, the original paper uses 25 demonstrations and trains for 100 iterations. In our paper, we only consider the relocation task.

The relocation task shown in Fig. 2 requires an agent to grasp and relocate a blue ball from a tabletop to a green region. In the original DAPG paper, task initializations are sampled from a restrictive region on the table top (width of 0.3m). In a pilot study using the authors' CyberGlove demonstrations,

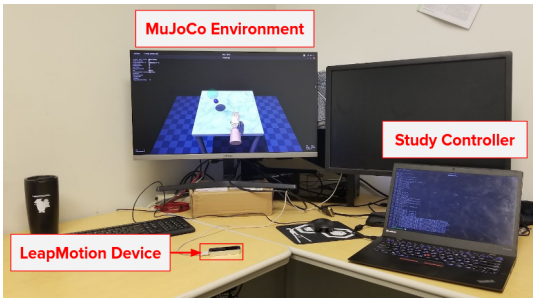


Fig. 3: Demo collection setup; The upwards-facing LeapMotion sensor reads hand joint information, the MuJoCo environment updates the Adroit robot hand joint angles in real time, and the collection is controlled via an external device.

we find that a policy trained solely in the restrictive space for 100 training iterations results in a >95% success rate in the restrictive space, but a 69% success rate in the full operating space (width of 0.5m). When the RL agent is provided access to train in the full operating space, it requires more than 200 training iterations to learn a policy with >95% success rate. Our work aims to reduce the tradeoff between higher sample efficiency and lower performance for the relocation task.

## IV. Methods

In this section, we describe the two sets of experiments we conducted, each aimed to investigate a specific research question (RQ) as described below.

### A. *RQ1: What are the effects of low-cost sensing on learning outcomes?*

Our first set of experiments examine how policies trained with demonstrations from a low-cost vision-based sensor (LeapMotion [13]) compare to policies trained using a wearable glove. A glove's ability to produce high quality demonstrations with on-finger joint tracking makes it advantageous for collecting human demonstrations for complex dexterous manipulation tasks such as rotating objects in-hand. In such situations, external cameras may have difficulty picking up finger movements that are obstructed by the palm or by other fingers. However, for applications aiming to target broader consumer audiences or research labs, the CyberGlove III is expensive.

External vision-based sensors can infer all joint poses for relocation or grasping tasks, potentially reducing hardware costs. To evaluate the performance difference between on-finger and vision-based sensing, we develop an interface to convert input from a LeapMotion hand tracker – a US$90 device which uses stereoscopic vision to estimate hand poses – into demonstrations accessible by the original DAPG implementation. We limit our investigation to the relocation task. The task does not obstruct external visual tracking, and therefore can adequately compare the two sensors. Our demonstration collection environment is shown in Fig. 3.

The LeapMotion's low price-point comes with lower quality demonstrations: 1) The device does not track roll joints for the thumb and little finger, resulting in two fewer DOFs. 2) The device sometimes has difficulty distinguishing between the four fingers, resulting in fingers moving or grasping simultaneously; For the relocation task this behavior is acceptable. 3) The device collects noisy observations due to jittery sensor readings. As such, **RQ1** is indirectly related to the robustness of learning algorithms against suboptimal demonstrations.

TABLE I: Source of demonstrations for each policy.

| Policy | Condition | Rest.-Orig. | Full-Random | Full-Corr. |
|---|---|---|---|---|
| **30O** | – | 30 | – | – |
| **10O+20R** | B | 10 | 20 | – |
| **10O+20C** | A | 10 | – | 20 |
| **20O+10R** | B | 20 | 10 | – |
| **20O+10C** | A | 20 | – | 10 |

"Full-Corr." indicates corrective demonstrations from failures cases of a policy trained on the original demonstration set "Rest.-Orig.".

**H1**: LeapMotion demonstrations will result in policies with rollout success rates that are comparable to CyberGlove III demonstrations for the relocation task, at the cost of sample efficiency.

### B. *RQ2: What is the utility of corrective demonstrations?*

Our second research question is motivated by the practical question of whether a meaningful improvement can be gained by including corrective demonstrations in a data set. We evaluate whether the performance of dexterous manipulation policies for the relocation task can be improved by splitting the training into two stages, labeled as *Condition A*:

**Stage 1**: Collecting demonstrations from randomly-sampled start and goal locations in the restrictive space.
**Stage 2**: Appending additional demonstrations collected from failure cases in the full operating space, using a policy trained in the full space using Stage 1 demonstrations.

We compare the policy from *Condition A* to a policy trained on randomly-sampled demonstrations from both the restrictive space and the full operating space, labeled as *Condition B*. Fig. 5 shows a visual of the tabletop areas for the restrictive and full operating spaces. Table I details the distributions of demonstrations from the restrictive space and the full operating space for each of the five policies we address. The full operating space only encompasses the area outside the restrictive space, causing corrective demonstrations to come from policy rollouts on state initializations not seen in the original demonstrations. While all policies have the same total number of demonstrations, policies with corrective demonstrations require twice the training time. Thus, **RQ2** investigates whether the effort of training two policies – one with the demonstrations from the restrictive space, one after collecting corrective demonstrations – results in a tangible performance improvement over randomly-sampled demonstrations from both spaces. Fig. 4 summarizes our methods, which are formalized below:

For *Condition A*, we first collect a set of randomly-sampled demonstrations $D^O$ in the restrictive space, and train a policy

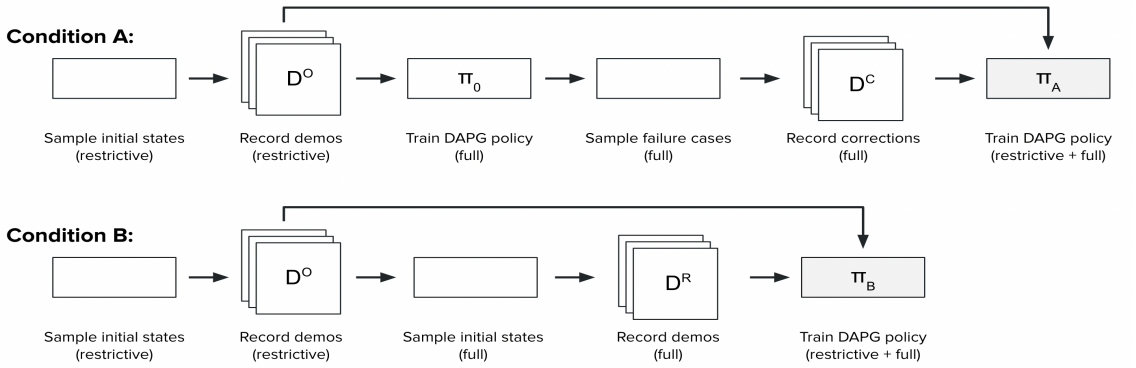


Fig. 4: Flowchart summarizing our approach to investigating RQ2. (Top) shows Condition A using randomly-sampled states from the restrictive space and corrective demonstrations from the full operating space; (Bottom) shows Condition B with randomly-sampled states from the restrictive as well as the full operating space.

$\pi_0$ in the full operating space. On an evaluation set of 1000 initial states from the full operating space, we roll out $\pi_0$ and identify the lowest-performing failure cases. We then record corrective demonstrations $D^C$ for these failure cases, and combine them with the original demonstrations $D^O$ to train a second policy $\pi_A$ in the full operating space. We measure the rollout success ratio for $\pi_A$ using the evaluation set in the full operating space.

We compare $\pi_A$ to the policy from *Condition B*, $\pi_B$, trained using the original demonstrations $D^O$ from the restrictive space, and an additional set of randomly-sampled demonstrations $D^R$ from the full operating space.

We anticipate that by guiding $D^C$ towards weaker areas of $\pi_0$, the resulting $\pi_A$ will have an increased coverage of challenging initial task states in the full operating space, resulting in a higher performance than $\pi_B$. We evaluate a policy's performance by both its rollout success ratio and its sample efficiency. Furthermore, we expect all four policies that include demonstrations from the full operating space to outperform **Policy 30O**, by matter of having access to demonstrations from the full operating space.

**H2**: When the number of additional demonstrations from the full operating space is **higher** than the original demonstrations from the restrictive space, **corrective demonstrations will improve performance** compared to randomly-sampled additional demonstrations.

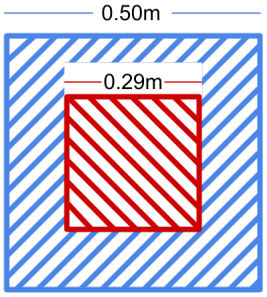


Fig. 5: Depiction of the dimensions for the restrictive (red) space and the full operating (blue) space. Neither space overlaps, and the dimensions of the restrictive space are chosen such that the full operating space is twice as large.

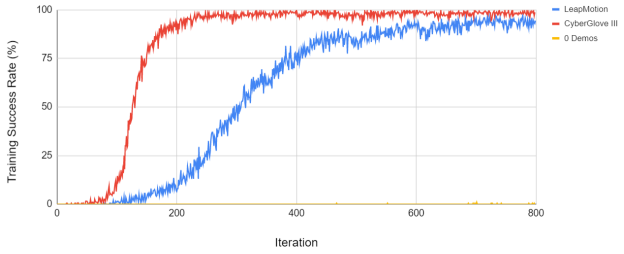


Fig. 6: Rollout success ratios for training 800 iterations. (Blue) Policy trained on 25 LeapMotion demos. (Red) Policy trained on 25 CyberGlove III demos. We see the CyberGlove III policy converges to 100% at close to 200 iterations, while the LeapMotion policy takes more than 600 iterations to reach a similar rollout success ratio. (Yellow) Policy trained on no demonstrations.

**H3**: When the number of additional demonstrations from the full operating space is **lower** than the original demonstrations from the restrictive space, **corrective demonstrations will improve performance** compared to randomly-sampled additional demonstrations.

## V. Results

### A. Evaluating *H1*: Demonstrations from LeapMotion vs. CyberGlove III

TABLE II: Rollout success ratios in full operating space after 800 iterations

| Demos | CyberGlove III | LeapMotion |
|---|---|---|
| 25 | 99.0% | 98.1% |

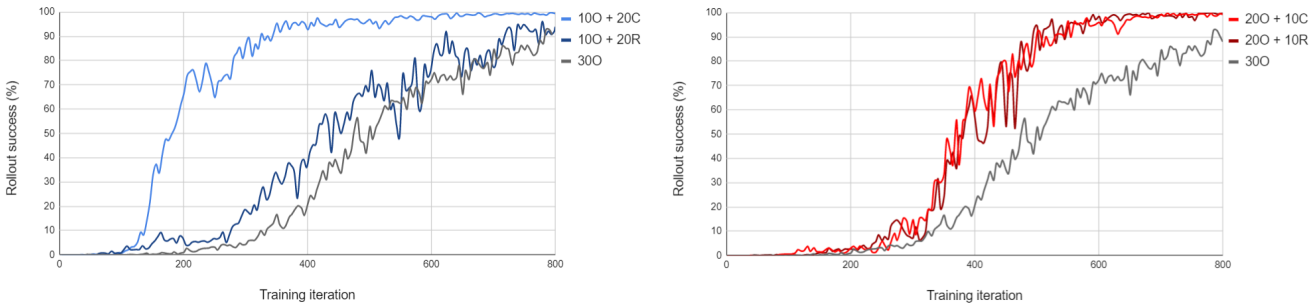


Fig. 7: Rollout success ratios for each policy. (Left) shows policies with a **high** proportion of demonstrations in the full operating space, while (Right) shows policies with a **low** proportion of demonstrations in the full operating space. For comparison with policy trained on demonstrations only from the restrictive space, **Policy 30O** is shown on both plots.

We train a policy for 800 iterations using 25 demonstrations collected from LeapMotion and compare it with a policy trained for 800 iterations using the original 25 demonstrations provided by the authors of DAPG. The results are shown in Table II. We use the same network architecture and hyperparameters as in the original DAPG paper. We observe that the optimal demonstrations from CyberGlove III allow DAPG to converge towards 100% success ratio in around 200 iterations whereas it takes almost 600 iterations for the LeapMotion's policy to reach a comparable performance (see Fig. 6).

***Outcome***: Policies trained on the full operating space using demonstrations from LeapMotion are able to reach comparable success ratios to policies trained with CyberGlove III demonstrations, at the cost of sample efficiency.

### *B. Evaluating **H2**, **H3**: Applying corrective demonstrations in the full operating space*

We train five policies based on the distribution in Table I. Each policy is trained for 800 iterations in order to capture the policy's full plateau of rollout success. Table III compares the rollout success ratio of each policy at different iterations.

We find that all combined policies converge to high rollout success ratio above 90%. However, we see marked differences in their sample efficiencies, as shown in Fig. 7. All policies perform better than **Policy 30O** and are more sample efficient, which is expected given their access to demonstrations from the evaluation set. Furthermore, **Policy 10O+20C** is more sample efficient than **Policy 10O+20R** and has a 5.4% higher rollout success ratio, confirming our hypothesis **H2**. However, we do not find a comparable difference between the sample efficiencies of **Policy 20O+10C** and **Policy 20O+10R**, perhaps due to those policies having a low proportion of additional demonstrations from the full operating space. The findings reject our hypothesis **H3**.

TABLE III: Policy rollout success ratios

| Policy | Rollout Success [%] 200-iter | 400-iter | 600-iter | 800-iter |
|---|---|---|---|---|
| **30O** | 1.1 | 21.2 | 72.6 | 88.0 |
| **10O+20R** | 7.0 | 36.0 | 75.6 | 93.9 |
| **10O+20C** | 66.6 | 92.7 | 97.9 | 99.3 |
| **20O+10R** | 3.2 | 59.4 | 96.0 | 99.2 |
| **20O+10C** | 2.7 | 55.8 | 98.4 | 100.0 |

***Outcome***: When the proportion of additional demonstrations from the full operating space is higher than the original demonstrations from the restrictive space, we find that corrective additional demonstrations improve the policy's performance for the relocation task compared to randomly-sampled additional demonstrations. When the proportion of additional demonstrations is lower, we find no notable difference between policies learned from *Condition A* and *Condition B*.

## VI. Conclusion

We find that vision-based sensors can be used to collect demonstrations for the relocation task achieving policies with similar success to policies resulting from wearable sensors (see Fig. 6), at the cost of sample efficiency. The result validates our use of the LeapMotion device to collect demonstrations for the relocation task, reducing the setup cost by approximately 140x. Our validation supports **H1**.

We use corrective additional demonstrations to guide policy exploration in the full operating space, and find that corrective demonstrations improve the performance and sample efficiency of policies as compared to those trained on randomly-sampled additional demonstrations. However, this improvement is limited to more additional demonstrations from the full operating space than the original demonstrations from the restrictive space. Our investigation supports **H2** and rejects **H3**. Additionally, the high rollout success ratios of our policies show that DAPG is resilient to suboptimal demonstrations at the cost of sample efficiency.

To further investigate **RQ1**, we plan to improve the correspondence between LeapMotion and MuJoCo. Our pipeline for recording demonstrations using MuJoCo leverages the LeapMotion tracker for data collection and a web server to communicate with MuJoCo. Though this pipeline is functional, some joint mappings are imperfect, limiting the quality of our collected demonstrations.

## VII. Acknowledgements

We thank Nakul Gopalan for his feedback and encouragement towards publishing this work.

## References

[1] I. A. OpenAI, M. Andrychowicz, M. Chociej, M. Litwin, B. McGrew, A. Petron, A. Paino, M. Plappert, G. Powell, R. Ribas *et al.*, "Solving rubik's cube with a robot hand," 2019.

[2] A. Rajeswaran, V. Kumar, A. Gupta, G. Vezzani, J. Schulman, E. Todorov, and S. Levine, "Learning Complex Dexterous Manipulation with Deep Reinforcement Learning and Demonstrations," 2018.

[3] CyberGlove Systems LLC, "CyberGlove III - CyberGlove Systems LLC," http://www.cyberglovesystems.com/cyberglove-iii/.

[4] T. Zhang, Z. McCarthy, O. Jowl, D. Lee, X. Chen, K. Goldberg, and P. Abbeel, "Deep Imitation Learning for Complex Manipulation Tasks from Virtual Reality Teleoperation," *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 5628–5635, 2018.

[5] V. Kumar, A. Gupta, E. Todorov, and S. Levine, "Learning Dexterous Manipulation Policies from Experience and Imitation," 2016. [Online]. Available: http://arxiv.org/abs/1611.05095

[6] R. Jeong, J. T. Springenberg, J. Kay, D. Zheng, Y. Zhou, A. Galashov, N. Heess, and F. Nori, "Learning Dexterous Manipulation from Suboptimal Experts," pp. 1–20, 2020. [Online]. Available: http://arxiv.org/abs/2010.08587

[7] J. Orbik, A. Agostini, and D. Lee, "Inverse reinforcement learning for dexterous hand manipulation," pp. 1–7, 2021.

[8] J. Ho and S. Ermon, "Generative adversarial imitation learning," *Advances in neural information processing systems*, vol. 29, pp. 4565–4573, 2016.

[9] J. Fu, K. Luo, and S. Levine, "Learning robust rewards with adversarial inverse reinforcement learning," *arXiv preprint arXiv:1710.11248*, 2017.

[10] 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.

[11] J. Spencer, S. Choudhury, A. Venkatraman, B. Ziebart, and J. A. Bagnell, "Feedback in imitation learning: The three regimes of covariate shift," *arXiv preprint arXiv:2102.02872*, 2021.

[12] D. Antotsiou, C. Ciliberto, and T.-K. Kim, "Adversarial imitation learning with trajectorial augmentation and correction," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 4724–4730.

[13] LeapMotion, "Ultraleap for Developers," https://developer.leapmotion.com/.