Visual Imitation Made Easy

Sarah Young[1] **Dhiraj Gandhi** [2] **Shubham Tulsiani**[2]
Abhinav Gupta[2,3] **Pieter Abbeel**[1] **Lerrel Pinto**[1,4]
[1]University of California, Berkeley [2]Facebook AI Research
[3]Carnegie Mellon University [4]New York University
`Project Page`

Abstract: Visual imitation learning provides a framework for learning complex manipulation behaviors by leveraging human demonstrations. However, current interfaces for imitation such as kinesthetic teaching or teleoperation prohibitively restrict our ability to efficiently collect large-scale data in the wild. Obtaining such diverse demonstration data is paramount for the generalization of learned skills to novel scenarios. In this work, we present an alternate interface for imitation that simplifies the data collection process while allowing for easy transfer to robots. We use commercially available reacher-grabber assistive tools both as a data collection device and as the robot's end-effector. To extract action information from these visual demonstrations, we use off-the-shelf Structure from Motion (SfM) techniques in addition to training a finger detection network. We experimentally evaluate on two challenging tasks: non-prehensile pushing and prehensile stacking, with 1000 diverse demonstrations for each task. For both tasks, we use standard behavior cloning to learn executable policies from the previously collected offline demonstrations. To improve learning performance, we employ a variety of data augmentations and provide an extensive analysis of its effects. Finally, we demonstrate the utility of our interface by evaluating on real robotic scenarios with previously unseen objects and achieve a 87% success rate on pushing and a 62% success rate on stacking. Robot videos are available at our project website.

Keywords: Imitation learning, Generalization, Visual observations.

1 Introduction

A powerful technique to learn complex robotic skills is to imitate them from humans [1, 2, 3, 4]. Recently, there has been a growing interest in learning such skills from visual demonstrations, since it allows for generalization to novel scenarios [5, 6]. Prominent works in Visual Imitation Learning (VIL) have demonstrated utility in intricate manipulation skills such as pushing, grasping, and stacking [5, 7]. However, a key bottleneck in current imitation learning techniques is the use of interfaces such as kinesthetic teaching or teleoperation, which makes it harder to collect large-scale manipulation data. But more importantly, the use of such interfaces leads to datasets that are constrained to be in restrictive lab settings. Resulting in-lab demonstrations often contain little to no variability in objects or environments which severely limits the generalizability of the learned skills in novel, previously unseen situations [8].

It is thus important to find a way to simplify data collection for imitation learning to allow both data collection at scale and real world diversity. What we need is a cheap interface (for prevalence), which can be intuitively controlled (for efficiency). Interestingly, one of the cheapest 'robots' that is highly prevalent, easy to control, and requires little to no human training is the reacher-grabber depicted in Fig. 1. This assistive tool is commonly used for grasping trash among other activities of daily living and has recently been shown to be a scalable interface for collecting grasping data in the wild by Song et al. [9]. However, unlike teleoperation [5] or kinesthetic [10] interfaces where the demonstrations are collected on the same platform as the robot, assistive tools are significantly different from robotic manipulators. Song et al. [9] bridges this gap by first extracting grasp points from demonstrations and then transferring them to robot in order to achieve closed-loop grasping of novel objects. A key problem, however, still lies in scaling this to enable imitation of general robotics tasks. One

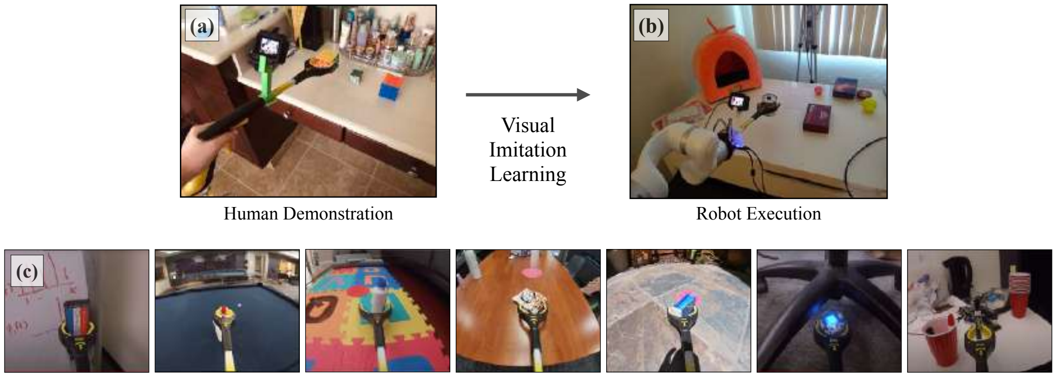


Figure 1: In this work we present a framework for visual imitation learning, where demonstrations are collected using commercially available reacher-grabber tools (a). This tool is also instrumented as an end-effector and attached to the robot (b). This setup allows us to collect and learn from demonstration data across diverse environments (c), while allowing for easy transfer to our robot.

possible solution is to extract full tool configuration and learn a mapping between grabber and the robot hardware. An alternative is to run domain adaptation based techniques for transfer. However, effectively using such techniques in robotics is still an active area of research [6, 11]. Instead, why not simply use the assistive tool as an end-effector?

In this work, we propose an alternate paradigm for providing and learning from demonstrations. As seen in Fig. 1 (a,c), the user collects Demonstrations using Assistive Tools (DemoAT) to solve a task. During the collection of this data, visual RGB observations are collected from a camera mounted on the DemoAT tool. Given these visual demonstrations, we extract tool trajectories using off-the-shelf Structure from Motion (SfM) methods and the gripper configuration using a trained finger detector. Once we have extracted tool trajectories, corresponding skills can be learned using standard imitation learning techniques. Particularly, we employ simple off-the-shelf behavior cloning. Finally, these skills can be transferred to a robot that has the same tool setup as the end-effector. Having the same end-effector as the demonstration tool coupled with a 6D robotic control (Fig. 1 (b)) allows for a direct transfer of learning from human demonstrations to the robot.

To study the effectiveness of this tool, we focus on two challenging tasks: (a) non-prehensile pushing [12, 13], and (b) prehensile stacking [14, 15]. For both tasks, we collect 1000 demonstrations in multiple home and office environments with various different objects. This diversity of data in objects and environments allows our learned policies to generalize and be effective across novel objects. Empirically, we demonstrate a baseline performance of 62.5% in pushing and 29.2% in stacking with naive behavioral cloning on our robot with objects previously unseen in the demonstrations. We employ random data augmentations such as random crops, jitter, cuts, and rotations to significantly improve pushing performance to 87.5% and stacking performance to 62.5%. Finally, we analyze the effects of diversity to demonstrate the need for large-scale demonstration data in the wild.

In summary, we present three key contributions in this work. First, we propose a new interface for visual imitation learning that uses assistive tools to gather diverse data for robotic manipulation, including an approach for collecting grabber 3-D trajectories and gripper transitions. Second, we demonstrate the utility of this framework on pushing and stacking previously unseen objects, with a success rate of 87.5% and 62.5% respectively. Finally, we present a detailed study on the effects of data augmentations in learning robotic skills, and demonstrate how the combination of random 'crops', 'rotations' and 'jitters' significantly improve our policies over other augmentations. Our platform design, software, and data will be publicly released and is attached in the supplementary material.

2 Related Work

In this section, we briefly discuss prior research in the context of our work. For a more comprehensive review of imitation learning, we point the readers to Argall et al. [16].

Interfaces for Imitation: In imitation learning, a robot tries to learn skills from demonstrations provided by the expert. There are various interfaces through which these demonstration can be recorded. One option is teleoperation, in which the human controls the robot using a control interface. This method has been successfully applied to a large range of robotic tasks including flying a robotic helicopter [17], grasping objects [18, 19], navigating robots through cluttered environments [20, 21, 22], and even driving cars [23]. Teleoperation has been successful in solving a wide variety of tasks because of the availability of control interfaces through which human operators can perform high-quality maneuvers. However, it is challenging to devise such interfaces for robotic manipulation. Kinesthetic demonstrations, in which the expert actively controls the robot arm by exerting force on it, is an effective method [24, 10] of collecting robot manipulation demonstrations for playing ping pong [25] and cutting vegetables [26]. However, for visuomotor policies, which map from pixels to actions, these demonstrations are inappropriate due to the undesirable appearance of human arms. One way to overcome this problem is by mounting an assistive tool on the robot end effector that is being used to record demonstration in isolation [9]. We take this idea a step further by using it as an end-effector on the robot as well. This eliminates the domain gap between the human-collected demonstrations and the robot executions, which enables easier imitation.

Behavior Cloning in Imitation: Behavior cloning is the simplest form of imitation learning, where the agent learns to map observations to actions through supervised learning. It has been successfully applied in solving a wide range of tasks including playing games [27], self-driving [23], and navigating drones through cluttered environments [22]. However, it has not been widely applicable to learning visuomotor policies for robotic manipulation tasks due to unwanted visual artifacts collected in kinesthetic demonstrations. To overcome this problem, Zhang et al. [5] propose a Virtual Reality (VR) setup to collect robot manipulation data. They showed that behavior cloning can be used to learn complex manipulation tasks, such as grasping and placing various objects. There have also been recent efforts to imitate from visual demonstrations collected from a different space e.g. from a different viewpoint or an agent with a different embodiment [6, 11] from the robot. This is a promising direction as it allows for data collection outside the lab. However, learning from such demonstrations is still an active research problem, as there is a significant domain gap between training and testing. In our setup, we use behavior cloning to learn challenging tasks such as pushing and stacking. But instead of relying on a costly VR setup which can only be deployed in constrained lab environments, we rely on cheap assistive tools to collect diverse data in the wild. Further, we eliminate the domain gap present in previously mentioned lines of work by attaching the same tool on the robot to match the demonstration and imitation space.

Data Augmentation in Learning: Data augmentation is widely used in machine learning to inject additional knowledge in order to overcome the challenges of overfitting. This technique has been shown to greatly benefit deep learning systems for computer vision. Its use can be found as early as LeNet-5 [28], which was used to classify hand written digits. In AlexNet [29], data augmentations such as random flip and crop were used to improve the classification accuracy. More recently, learning augmentation strategies from data has emerged as a new paradigm to automate the design of augmentation [30, 31, 32]. For unsupervised and semi-supervised learning, several unsupervised data augmentation techniques have been proposed [33, 34]. It has also been extensively used in context of RL, where domain randomization is proposed to transfer learning from simulation to real world [35, 36, 37]. Although the effects of augmentations have been extensively studied in image-based RL [38, 39], to the best of our knowledge, we are the first to study the effects of data augmentations in real-robot applications.

3 Method

In this section, we describe the Demonstrations with Assistive Tools (DemoAT) framework for collecting visual demonstrations, along with our pipeline for imitation learning.

3.1 The DAT imitation framework

Demonstration Tool: Our DemoAT setup is built around a plastic 19-inch RMS assistive tool [40] and a RGB camera [41] to collect visual data. We attach a 3D printed mount above the stick to hold the camera in place. At the base of the reacher-grabber, there is a lever to control the opening and closing of the gripper fingers. To collect demonstrations, a human user uses the setup shown in

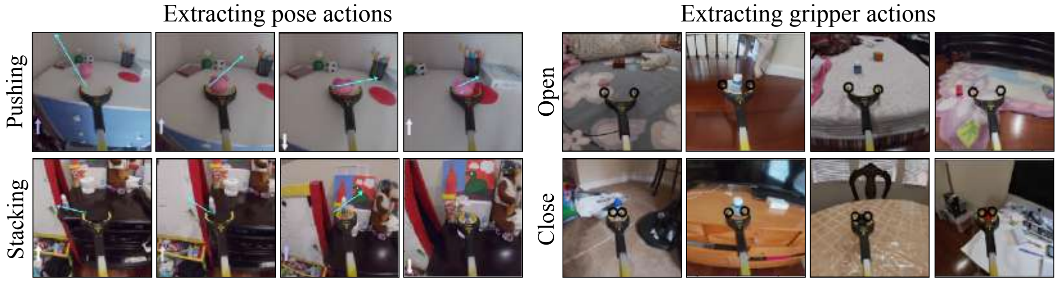


Figure 2: Extracting labels: (a) COLMAP translation arrows are shown for pushing and stacking. The center blue arrow shows movement in the transverse plane of the camera, while the color map arrow in the bottom left corner shows up-down movement. (b) Gripper finger predictions from our finger detection network along with open and close labels for the gripper configuration.

Fig. 1 (a), which allows the user to easily push, grab and interact with everyday objects in an intuitive manner. Examples of demonstrations can be seen in Fig. 1 (c) and Fig. 2. Since a demonstration collected with DemoAT is visual, it can be represented as a sequence of images $\{I_t\}_{t=0}^{T}$.

Robot End-effector: The tool is attached on a 7DoF robot arm with a matching camera and mount setup (Fig. 1 (b)). However, to actuate the fingers, we will need to create an actuating mechanism. Through a compact, lightweight and novel mechanism, we replace the lever from the original reacher grabber tool with a controllable interface. Details on this mechanism are presented in Appendix **??**. While we use an xArm7 robot [42] as our robotic arm, we note that this end-effector setup can be attached to any standard commercial-grade robotic arm.

3.2 Extracting actions from Visual Demonstrations

Although our demonstration tool provides a robust and reliable interface to collect visual demonstrations, our framework in itself does not have explicit sensors to collect information about actions such as the end-effector's motion or the finger locations. For effective imitation learning, this information about the 'actions' taken by the human demonstrator is crucial. To address this, we recover 6DoF poses of the tool using Structure-from-Motion (SfM) reconstruction. Specifically, we use the publicly available COLMAP [43, 44] software for SfM. Once we have the end-effector pose p_t for every image I_t, we extract the relative translation and rotation Δp_t between consecutive frames and use them as the action for training. As SfM only allows us to recover pose up to a scaling factor, we normalize Δp_t across the trajectory to account for this ambiguity.

COLMAP gives us the relative change in pose across frames, however, we also need to obtain the finger configurations for tasks that require moving the fingers. To do this, we use a neural network that extracts the location of the gripper fingers in our observations. This network is trained on a small human-labeled dataset of 155 frames from the DemoAT setup. Given these gripper finger locations predicted by our gripping model, we can generate labels for "close" or "open" states $g_t \in \{0, 1\}$. For this we track the distance between fingers. If distance falls below a threshold, we annotate them as "close", otherwise "open". Through this procedure we can now obtain visual demonstrations with actions a_t, which is represented as $(o_t, a_t = (\Delta p_t, g_{t+1}))_{t=0}^{T}$. Note that the grasping action at a given timestep is the grasp state at the next timestep. Visualizations of actions can be seen in Fig. 2.

Accuracy of reconstructed actions: Our method for extracting labels can be noisy. Specifically, COLMAP reconstructions are significantly less accurate in lightly textured, clean, and high dynamic range scenes. However, since our demonstrations are collected in cluttered real-world scenarios, our reconstructions are reasonably accurate for the purposes of learning. Nevertheless, to reduce the effect of noisy action labels, we visually inspect the reconstructions and discard $\sim 6\%$ of aberrant demonstrations. The model we use to generate grasping actions by detecting finger achieves $\sim 95\%$ accuracy on held-out testing set, which is empirically sufficient for downstream learning.

3.3 Imitation from visual demonstrations

Visual behavior cloning: We learn a policy using straightforward behavioral cloning [45, 27]. With the DAT imitation framework, we collect observation-action pairs $D = \{(o_t, a_t)\}$, where o_t is an

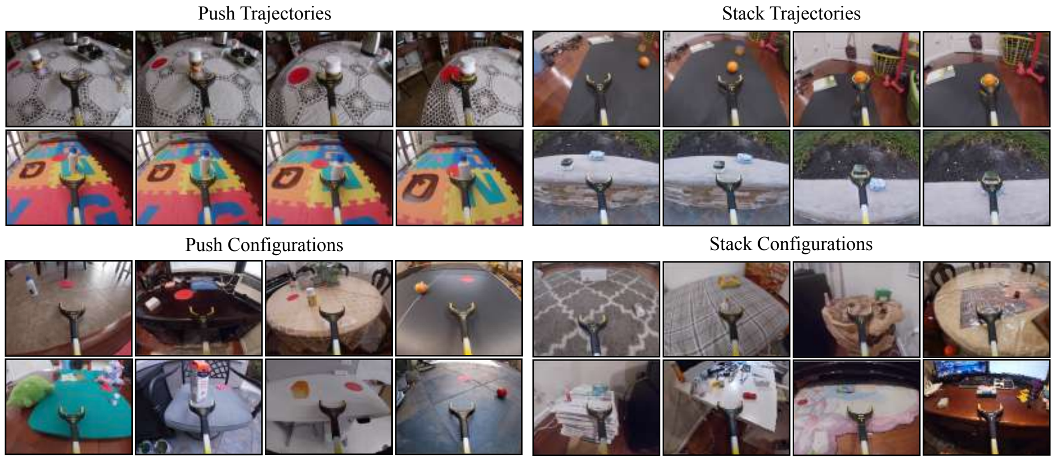


Figure 3: For both Pushing (left) and Stacking (right), we collect 1000 trajectories each with diverse objects and scenes. The top two rows depict 4 frames from single trajectories, while the bottom two rows depicts the variations in environments collected in our dataset.

image and a_t is the action to get from o_t to o_{t+1}. Using supervised learning, our policy learns a function $f(o_t, a_t)$ that maps observations o_t to actions a_t.

The input to the network is a single image $I_t \in \mathbb{R}^{3x224x224}$. The network outputs actions consisting of (a) a translation vector $x_t \in \mathbb{R}^3$ (b) a 6D representation of rotation $w_t \in \mathbb{R}^6$. We train on a 6D rotation representation [46] because it is continuous in the real Euclidean space and thus more suitable for learning as opposed to more commonly used axis-angle and quaternion based representations.

Our network architecture consists of a CNN with a set of fully connected layers. The convolutional part of the network comprises of the first five layers of the AlexNet followed by an additional convolutional layer. The output from the convolutions are fed into a set of three fully connected layers and projected to a 3D translation vector. To obtain predicted rotations, we concatenate the convolutional representation of the image with the predicted translations and feed this through two fully connected layers before projecting to a 6D rotation vector. For tasks that require using the gripper, we train an additional classification model that takes in an image $I_t \in \mathbb{R}^{3x224x224}$ and outputs a gripper open/close label $g_{t+1} \in \{1, 0\}$. Additional details on training are presented in Appendix **??**

Data augmentations for imitations: To improve the performance of our networks with limited data, we experiment with using the following data augmentations in training [38, 39, 47]:

- Color Jitter: Randomly adds up to $\pm 20\%$ random noise to the brightness, contrast and saturation of each observation.
- Crop: Randomly extracts a 224 x 224 patch from an original image of size 240 x 240.
- Cutout-color[rad]: Randomly inserts a colored box of size [10, 60] into the image.
- Rotation: Randomly rotates original image [-5, 5] degrees.
- Horizontal Reflection: Mirrors image across the y-axis. Action labels are reflected as well.

4 Experiments

In this section we describe our experimental evaluations using the DemoAT framework. Specifically, we aim to answer the following key questions: (a) Can DemoAT be used to solve difficult manipulation tasks? (b) How important is the scale and diversity of data for imitation learning in the wild? (c) How important is data augmentation for visual imitation?

4.1 Tasks

To study the use of DemoAT, we look at two tasks, non-prehensile pushing and prehensile stacking. To evaluate our learned policy we use two metrics. First, mean squared error (BC-MSE) between

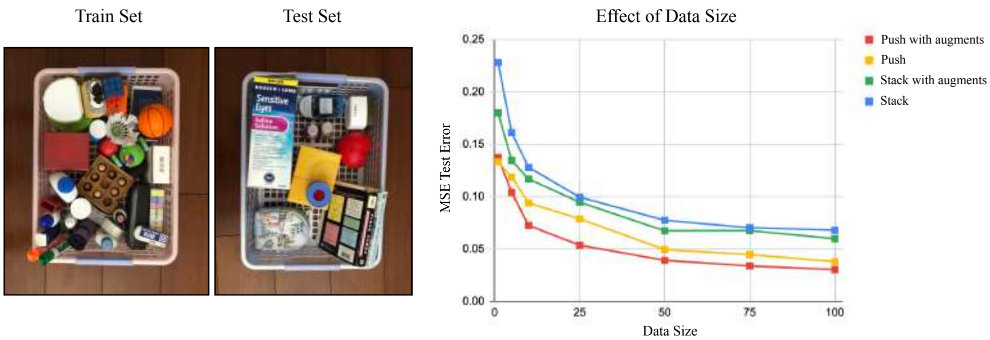


Figure 4: On the left, we show examples of objects used in training and testing for behavioral cloning evaluation. On the right, we evaluate the MSE error on held-out testing objects with and without random data augmentations. Note that as we increase the amount of data, our models improve and achieves lower error.

Table 1: Real robot evaluation results (average success rate): Stacking is split into 2 parts for evaluation: (a) grasping the initial object and (b) stacking the object onto the second object after completing (a).

		Naive BC 100%	BC with augment 100%	BC with augment 50%	BC with augment 10%
Push	reach goal	0.625	**0.875**	0.750	0
Stack	grasp object	0.750	**0.833**	0.792	0
	stack object	0.291	**0.625**	0.416	0

predicted actions and ground truth actions on a set of held-out demonstrations that contain novel objects in novel scenes. This offline measure allows for benchmarking different learning methods. Second, we evaluate on real robot executions on previously unseen objects and measure the fraction of successful executions. This captures the ability of our learned models to generalize on real scenarios.

Non-prehensile Pushing: This task requires the robot to push an object to a red circle by sliding it across the table. Such contact-rich manipulation has been extensively studied and known to be challenging to solve [12, 13]. Particularly in our case, we operate with diverse objects in diverse scenes, which makes accurately manipulating objects difficult. For robotic experiments, we evaluate robotic success rate as $\frac{\#\text{trajectories where object reaches goal}}{\#\text{total trajectories}}$ on a set of 24 different objects unseen in training.

Prehensile Stacking: In this task, the goal is to grasp an object and stack it on top of an equally sized or larger object. We set it up such that the smaller object is always in front of the larger object to reduce ambiguity in learning (Fig. 3). We evaluate robotic success rate as $\frac{\#\text{trajectories where object is grasped and stacked}}{\#\text{total trajectories}}$ on a set of 24 configurations unseen in training.

4.2 Can DemoAT be used for solving difficult manipulation tasks?

To study the utility of our DemoAT framework, we look at both measures of performance, the offline BC-MSE and the real robot success rate. Unless otherwise noted, we train our policies with 100% of training data and using the 'crop'+'jitter' augmentation for pushing and 'rotate'+'jitter' for stacking (Fig. 6). On the BC-MSE metric, we achieve an error of 0.028 on the pushing task and an error of 0.056 on the stacking task. We note that this is more than two orders of magnitude better than random actions, which has error of 0.67 and 0.69 on pushing and stacking respectively. This demonstrates that our policies have effectively learned to generalize to previously unseen demonstrations. Visualizations of how close predicted actions are to ground truth actions are presented in Appendix **??**.

Although our framework results in low BC-MSE error, such offline measures often do not necessarily correspond to effective online robotic performance. However, we demonstrate that our learned policies are robust enough to perform well on our robot. As seen in Table 1, we achieve a success

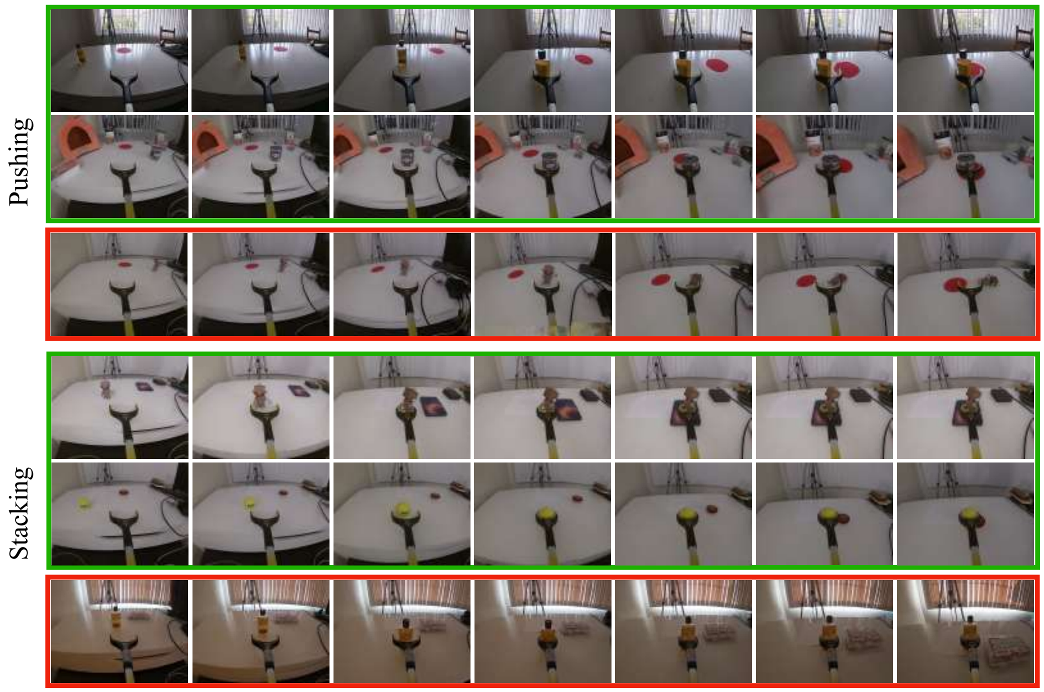


Figure 5: Here we visualize trajectories executed on the robot using our learned pushing and stacking policies trained with augmented data. Successful trajectories are highlighted in green, unsuccessful ones in red.

rate of 87.5% on pushing and 62.5% on stacking previously unseen objects. This demonstrates that our DemoAT framework can indeed solve complex tasks in diverse domains.

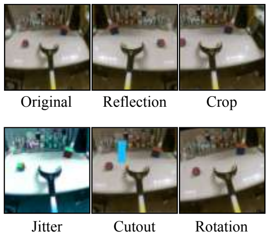
4.3 How important is data for imitation learning in the wild?

A key promise of the DemoAT setup is the ability to collect large-scale, diverse demonstrations. But how important is this diversity of data? To study this, we train policies on different fractions of training data - 1, 5, 10, 25, 50, 75, 100% and evaluate their performance. There are two ways of creating a fractional split, either by sequentially selecting the data or by random selection. The first split will contain more data in the same environment, while the second will contain more diverse data albeit with smaller amounts for each environment. In Fig. 4, we use the sequential split since it better captures the process of collecting data. In Appendix **??**, we present results for random splits.

Behavioral Cloning Evaluation: In Fig. 4 we illustrate the effects of changing dataset size on BC-MSE performance. In both the pushing and stacking task, we see increasing data size significantly improves performance especially in the low-data regime. We note that improvements diminish with larger data on the BC-MSE metric with just $\sim 0.9\%$ performance gain when increasing our training data from 500 to 1000 trajectories.

Real Robot Evaluation: In Table 1, we report the performance of robotic execution on models trained on 10%, 50%, and 100% of the collected data for each task on an unseen test set of 24 different objects. In both tasks, we see that with only 10% of the data (100 trajectories), the robot is unable to even reach the first object. When we increase to 50% of the data, we see a huge improvement and the robot starts to learn to reach the objects and complete the tasks. Particularly, with just 50% of the data, the robot can successfully reach the object 100% of the time in the non-prehensile pushing task. When we evaluate with all our data, we still see considerable performance improvements in completing the tasks, with 12.5% in pushing and 20.9% in stacking. This improvement is significantly higher than what we see with the BC-MSE metric. We hypothesize that since both these tasks require fine-grained manipulation, small improvements in BC-MSE results in large improvements in real-robot performance, especially when the models are already performative.

Data augmentations



Results on pushing

	Rotation	Crop	Color Jitter	Reflection	Cutout-color
None	0.0381				
Rotation	0.0352	0.0340	0.0342	0.0368	0.0334
Crop	0.0340	0.0356	0.0315	0.0365	0.0346
Color Jitter	0.0342	0.0315	0.0355	0.0350	0.0348
Reflection	0.0368	0.0365	0.0350	0.0376	0.0341
Cutout-color	0.0334	0.0346	0.0348	0.0341	0.0398

Results on stacking

	Rotation	Crop	Color Jitter	Reflection	Cutout-color
None	0.0695				
Rotation	0.0636	0.0700	0.0599	0.0646	0.0672
Crop	0.0700	0.0684	0.0721	0.0712	0.0696
Color Jitter	0.0599	0.0721	0.0711	0.0614	0.0648
Reflection	0.0646	0.0712	0.0614	0.0670	0.0733
Cutout-color	0.0672	0.0696	0.0648	0.0733	0.0749

Figure 6: On the left, we show the five data-augmentations used in this work. On the middle and right, we present an analysis of MSE error (lower is better) on test-set using different combinations of data-augmentations for pushing and stacking respectively. In dark green, we highlight the best combinations.

Data Diversity vs Size: To further understand the effects of diversity and size, we run experiments that compare performance on the same fractional split, but different amounts of diversity in the data. Given a quota 100 demonstrations, we train on two splits of data: (A) many observations of the same objects and scenes (B) sparse observations across a diverse set of objects and scenes. We expect that dataset (A) will be better at generalizing to unseen objects, since it sees many different scenes during training. Indeed, we find that the test error for the diverse dataset (A) [0.081] is on average 1.4% higher than that of dataset (B) [0.067]. Analysis on other data splits is presented in Appendix **??**.

4.4 Does augmented data help?

To improve the performance of our learned policies, we employ data augmentations in training. But, how important are these augmentations in imitation learning?

Behavioral Cloning Evaluation: For both pushing and stacking, we compare the application of different augmentations: crop, color jitter, rotate, horizontal reflection, random cutout, and all permutations of two augmentations. We find that data augmentations allow our model to generalize better to unseen objects and scenes on the BC-MSE metric. As shown in Fig. 6, the best augmentation performs 0.7% better than naive behavioral cloning in pushing and 0.9% better in stacking. We note that 'crop'+'jitter' is the most effective augmentation for pushing and 'rotation'+'jitter' for stacking. In both tasks, random color cutout does not work as well. Since we focus on object manipulation tasks, it is likely that random color cutouts block important information such as the gripper fingers or the object, resulting in inaccurate predictions.

Real Robot Evaluation: Our second method of evaluation is to compare the success rates of stacking and pushing with data augmentations to naive behavioral cloning. Table 1 shows that for both tasks we achieve significant improvements with data augmentations. We see the biggest increases in performance in the second part of each task (after the initial object has been reached): a 12.5% improvement for reaching the goal in pushing and a 33.4% improvement in stacking. Interestingly, using augmentations with just 50% of training data surpasses the performance of not using augmentation with 100% of training data on both pushing and stacking. This ability to improve performance in robotics is in line with recent research in RL [38, 39] and computer vision [47].

5 Conclusion

In this paper, we present Demonstrations using Assistive Tools (DemoAT). In contrast to traditional imitation methods that rely on domain adaptation techniques or kinesthetic demonstrations, our proposed method allows for both easy large-scale data collection and direct visual imitation learning. We learn two challenging tasks, non-prehensile pushing and prehensile stacking, and evaluate our methods via two metrics: BC-MSE and robot success rate. We have shown that using a universal reacher-grabber tool that can act as an end-effector for virtually any robot, smarter data collection methods coupled with simple behavior cloning methods and data augmentations can lead to better out of distribution performance. We hope that this interface is a step towards more efficient robot learning, since it opens up directions for wide scale data collection and re-use.

References

[1] J. Piaget. *Play, dreams and imitation in childhood*, volume 25. Routledge, 2013.

[2] A. N. Meltzoff and M. K. Moore. Imitation of facial and manual gestures by human neonates. *Science*, 198(4312):75–78, 1977.

[3] A. N. Meltzoff and M. K. Moore. Newborn infants imitate adult facial gestures. *Child development*, pages 702–709, 1983.

[4] M. Tomasello, S. Savage-Rumbaugh, and A. C. Kruger. Imitative learning of actions on objects by children, chimpanzees, and enculturated chimpanzees. *Child development*, 64(6):1688–1705, 1993.

[5] T. Zhang, Z. McCarthy, O. Jow, D. Lee, X. Chen, K. Goldberg, and P. Abbeel. Deep imitation learning for complex manipulation tasks from virtual reality teleoperation. In *ICRA*, 2018.

[6] B. C. Stadie, P. Abbeel, and I. Sutskever. Third-person imitation learning. *ICLR*, 2017.

[7] Y. Zhu, Z. Wang, J. Merel, A. A. Rusu, T. Erez, S. Cabi, S. Tunyasuvunakool, J. Kramár, R. Hadsell, N. de Freitas, and N. Heess. Reinforcement and imitation learning for diverse visuomotor skills. *CoRR*, abs/1802.09564, 2018. URL http://arxiv.org/abs/1802.09564.

[8] A. Gupta, A. Murali, D. P. Gandhi, and L. Pinto. Robot learning in homes: Improving generalization and reducing dataset bias. In *Advances in Neural Information Processing Systems*, pages 9094–9104, 2018.

[9] S. Song, A. Zeng, J. Lee, and T. Funkhouser. Grasping in the wild: Learning 6dof closed-loop grasping from low-cost demonstrations. *Robotics and Automation Letters (RA-L)*, 2020.

[10] P. Sharma, L. Mohan, L. Pinto, and A. Gupta. Multiple interactions made easy (mime): Large scale demonstrations data for imitation. *CoRL*, 2018.

[11] P. Sermanet, K. Xu, and S. Levine. Unsupervised perceptual rewards for imitation learning. *arXiv preprint arXiv:1612.06699*, 2016.

[12] K. M. Lynch. *Nonprehensile robotic manipulation: Controllability and planning*. Citeseer, 1996.

[13] F. Ruggiero, V. Lippiello, and B. Siciliano. Nonprehensile dynamic manipulation: A survey. *IEEE Robotics and Automation Letters*, 3(3):1711–1718, 2018.

[14] Y. Zhang, B. K. Chen, X. Liu, and Y. Sun. Autonomous robotic pick-and-place of microobjects. *IEEE Transactions on Robotics*, 26(1):200–207, 2009.

[15] F. Furrer, M. Wermelinger, H. Yoshida, F. Gramazio, M. Kohler, R. Siegwart, and M. Hutter. Autonomous robotic stone stacking with online next best object target pose planning. In *ICRA*, 2017.

[16] B. D. Argall, S. Chernova, M. Veloso, and B. Browning. A survey of robot learning from demonstration. *Robotics and autonomous systems*, 57(5):469–483, 2009.

[17] A. Ng, A. Coates, M. Diel, V. Ganapathi, J. Schulte, B. Tse, E. Berger, and E. Liang. *ISER*, 2004.

[18] P. K. Pook and D. H. Ballard. Recognizing teleoperated manipulations. In *[1993] Proceedings IEEE International Conference on Robotics and Automation*, pages 578–585. IEEE, 1993.

[19] J. D. Sweeney and R. Grupen. A model of shared grasp affordances from demonstration. In *2007 7th IEEE-RAS International Conference on Humanoid Robots*, pages 27–35. IEEE, 2007.

[20] T. I. M. I. H. Inoue, M. Inamura, and H. Inaba. Acquisition of probabilistic behavior decision model based on the interactive teaching method. In *ICAR*, 1999.

[21] W. D. Smart. *Making reinforcement learning work on real robots*. Brown University Providence, 2002.

[22] S. Ross, N. Melik-Barkhudarov, K. S. Shankar, A. Wendel, D. Dey, J. A. Bagnell, and M. Hebert. Learning monocular reactive uav control in cluttered natural environments. In *ICRA*, 2013.

[23] M. Bojarski, D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang, et al. End to end learning for self-driving cars. *arXiv preprint arXiv:1604.07316*, 2016.

[24] B. Akgun, M. Cakmak, K. Jiang, and A. L. Thomaz. Keyframe-based learning from demonstration. *International Journal of Social Robotics*, 4(4):343–355, 2012.

[25] K. Muelling, A. Boularias, B. Mohler, B. Schölkopf, and J. Peters. Learning strategies in table tennis using inverse reinforcement learning. *Biological cybernetics*, 108(5):603–619, 2014.

[26] I. Lenz, R. A. Knepper, and A. Saxena. Deepmpc: Learning deep latent features for model predictive control. In *Robotics: Science and Systems*. Rome, Italy, 2015.

[27] 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*, pages 627–635, 2011.

[28] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 1998.

[29] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[30] E. D. Cubuk, B. Zoph, D. Mane, V. Vasudevan, and Q. V. Le. Autoaugment: Learning augmentation strategies from data. In *CVPR*, 2019.

[31] D. Ho, E. Liang, X. Chen, I. Stoica, and P. Abbeel. Population based augmentation: Efficient learning of augmentation policy schedules. In *ICML*, 2019.

[32] B. Zoph, E. D. Cubuk, G. Ghiasi, T.-Y. Lin, J. Shlens, and Q. V. Le. Learning data augmentation strategies for object detection. *arXiv preprint arXiv:1906.11172*, 2019.

[33] D. Berthelot, N. Carlini, I. Goodfellow, N. Papernot, A. Oliver, and C. A. Raffel. Mixmatch: A holistic approach to semi-supervised learning. In *NeurIPS*, 2019.

[34] Q. Xie, Z. Dai, E. Hovy, M.-T. Luong, and Q. V. Le. Unsupervised data augmentation for consistency training. *arXiv preprint arXiv:1904.12848*, 2019.

[35] F. Sadeghi and S. Levine. Cad2rl: Real single-image flight without a single real image. *arXiv preprint arXiv:1611.04201*, 2016.

[36] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *IROS*, 2017.

[37] L. Pinto, M. Andrychowicz, P. Welinder, W. Zaremba, and P. Abbeel. Asymmetric actor critic for image-based robot learning. *RSS*, 2018.

[38] M. Laskin, K. Lee, A. Stooke, L. Pinto, P. Abbeel, and A. Srinivas. Reinforcement learning with augmented data. *arXiv preprint arXiv:2004.14990*, 2020.

[39] I. Kostrikov, D. Yarats, and R. Fergus. Image augmentation is all you need: Regularizing deep reinforcement learning from pixels. *arXiv preprint arXiv:2004.13649*, 2020.

[40] Grabber pick up tool. `https://www.amazon.com/RMS-Grabber-Reacher-Rotating-Gripper/dp/B00THEDNL8/ref=sr_1_16?dchild=1&keywords=grabber+tool&qid=1595965487&sr=8-16`.

[41] Gopro hero7 silver. `https://gopro.com/en/us/shop/cameras/hero7-silver/CHDHC-601-master.html`.

[42] xarm 7. `https://www.xarm.cc/products/xarm-7-2020`.

[43] J. L. Schönberger and J.-M. Frahm. Structure-from-Motion Revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[44] J. L. Schönberger, E. Zheng, M. Pollefeys, and J.-M. Frahm. Pixelwise View Selection for Unstructured Multi-View Stereo. In *European Conference on Computer Vision (ECCV)*, 2016.

[45] D. A. Pomerleau. Alvinn: An autonomous land vehicle in a neural network. In *Advances in neural information processing systems*, pages 305–313, 1989.

[46] Y. Zhou, C. Barnes, J. Lu, J. Yang, and H. Li. On the continuity of rotation representations in neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5745–5753, 2019.

[47] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*, 2020.