Giving Robots a Hand: Broadening Generalization via Hand-Centric Human Video Demonstrations

Anonymous authors
Paper under double-blind review

Abstract

Videos of humans performing tasks are a promising data source for robotic manipulation because they are easy to collect in a wide range of scenarios and thus have the potential to significantly expand the generalization capabilities of vision-based robotic manipulators. Prior approaches to learning from human video demonstrations typically use third-person or egocentric data, but a central challenge that must be overcome there is the domain shift caused by the difference in appearance between human and robot morphologies. In this work, we largely reduce this domain gap by collecting hand-centric human video data (i.e., videos captured by a human demonstrator wearing a camera on their arm). To further close the gap, we simply crop out a portion of every visual observation such that the hand is no longer visible. We propose a framework for broadening the generalization of deep robotic imitation learning policies by incorporating unlabeled data in this format—without needing to employ any domain adaptation method, as the human embodiment is not visible in the frame. On a suite of six real robot manipulation tasks, our method substantially improves the generalization performance of manipulation policies acting on hand-centric image observations. Moreover, our method enables robots to generalize to both new environment configurations and new tasks that are unseen in the expert robot imitation data.

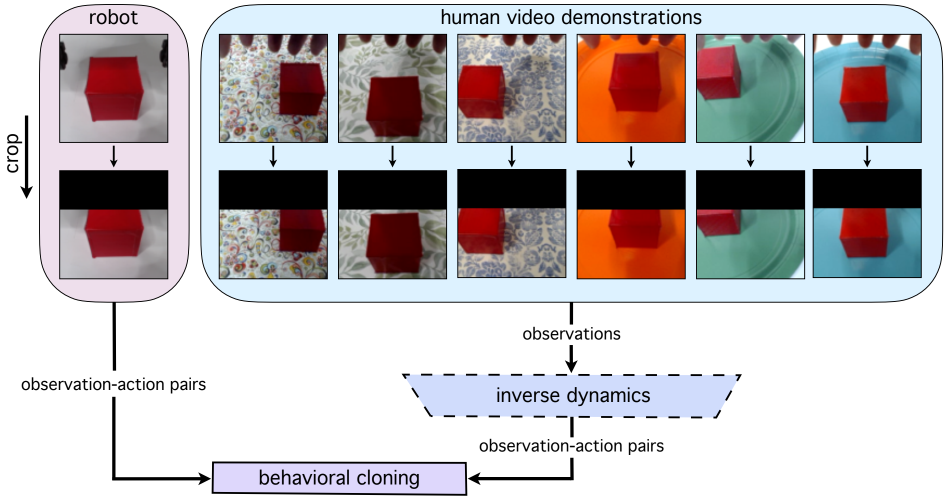

Figure 1: We incorporate diverse hand-centric human video demonstrations to train a behavioral cloning policy that can generalize to new environments and new tasks outside the distribution of expert robot imitation data. Images are cropped to close the domain gap between the human and robot observations. Action labels for human video demonstrations are inferred by an inverse dynamics model that is trained only on robot play data.

1 Introduction

Videos of humans completing tasks can be a beneficial data source in the context of robotic manipulation. First, they eliminate the need to relocate large robotic hardware to varied settings. Second, they lessen the amount of inevitable wear-and-tear that robotic systems accumulate over time. Third, they allow robot learning practitioners to avoid data collection methods such as kinesthetic teaching or robotic teleoperation, which can be physically demanding or involve special tools that require skill and training to use, such as virtual reality headsets (Zhang et al., 2018) and joystick controllers (Jonnavittula & Losey, 2022). The key consequence of these advantages is the relative ease of amassing a broad set of diverse human video demonstrations that can be leveraged to improve the generalization capabilities of vision-based robotic manipulation policies. This potential is particularly enticing because vision-based policies are typically brittle against real-world variation, such as changes in the background, lighting, and object appearances (Julian et al., 2020). However, a central challenge in learning from human video demonstrations is the difference in appearance between human and robot arms and bodies, which creates a distribution shift that must be accounted for.

Prior works in learning from human video demonstrations have aimed to mitigate this domain gap by taking explicit domain adaptation approaches such as human-to-robot image translation (Liu et al., 2018; Smith et al., 2019; Li et al., 2021; Xiong et al., 2021); learning domain-invariant visual representations or reward functions (Yu et al., 2018; Yang et al., 2019; Schmeckpeper et al., 2020; Zhou et al., 2021; Zakka et al., 2022); and leveraging keypoint representations of human and robot states (Das et al., 2020; Xiong et al., 2021). While these works have made progress in training robotic policies from human demonstrations, limitations and difficulties arising from the human-robot domain gap still remain. For example, certain explicit domain adaptation approaches suffer conspicuous visual artifacts when translating human images to robot images (Smith et al., 2019).

One common thread in the aforementioned works is that they operate from a third-person camera perspective, where stark differences in human and robot morphologies cause a prominent distribution shift. In this work, we significantly reduce the domain gap between human and robot data by leveraging hand-centric videos (i.e., clips captured by a wrist-mounted camera), taking the arm and body out of the picture. In the human domain, this involves a demonstrator wearing a camera on their forearm and completing a task with their hand—a process that is quicker and less taxing than kinesthetic teaching and robotic teleoperation. To further close the gap, we crop out a fixed portion of every image observation such that the hand or end-effector is no longer visible, leaving just the environment and objects in view. As a result, we eliminate the need to employ any domain adaptation method and are able to perform end-to-end learning of vision-based manipulation policies directly from human video data, where the actions are inferred by an inverse dynamics model.

The main contribution of this work is the study of a simple, novel method for incorporating diverse hand-centric human video demonstrations that allows a practitioner to improve upon policies trained solely on narrow expert robot demonstrations, while bypassing explicit domain adaptation approaches entirely. Across several real-world robotic manipulation tasks, such as reaching, grasping, pick-and-place, cube stacking, and plate clearing, we observe that our method leads to significant improvements in generalization performance. Our policies generalize to both new environments and new tasks that are outside the distribution of expert robot imitation data. We release the datasets we collect to train the inverse models and imitation learning policies on our project website.

2 Related Work

Imitation learning is a powerful paradigm for training an agent to complete a task by learning a mapping between observations and actions. Traditional approaches to robotic imitation assume access to expert demonstrations collected from the robot's observation and action spaces (Hayes & Demiris, 1994; Atkeson & Schaal, 1997; Argall et al., 2009; Osa et al., 2018). Since collecting expert trajectories with a real robot can be expensive, physically demanding, or require special teleoperation equipment and training (Zhang et al., 2018; Mandlekar et al., 2020), we study the setting of training robotic agents to complete tasks by watching videos of a human demonstrator. One central challenge in this setting is the distribution shift caused by apparent visual differences between human and robot arms and bodies.

Past works have addressed this distribution shift in various ways. Some have employed explicit domain adaptation techniques such as human-to-robot context translation (Liu et al., 2018; Sharma et al., 2019) and pixel-level image translation (Smith et al., 2019; Li et al., 2021; Xiong et al., 2021), commonly using generative models like CycleGAN which can learn mappings between domains given unpaired data (Zhu et al., 2017). Other works have explicitly specified the correspondences between human and robot embodiments and behaviors by, e.g., employing pose and object detection techniques inspired by computer vision research (Yang et al., 2015; Nguyen et al., 2018; Ramirez-Amaro et al., 2017; Kumar et al., 2022) and learning keypoint-based state representations of human and robot observations (Das et al., 2020; Xiong et al., 2021). Some have taken a more implicit approach and sought to learn domain-invariant visual representations or reward functions that are useful for learning to solve downstream tasks (Sermanet et al., 2016; 2018; Yu et al., 2018; Yang et al., 2019; Mees et al., 2020; Schmeckpeper et al., 2020; Chen et al., 2021; Zhou et al., 2021; Nair et al., 2022; Zakka et al., 2022). Yet another class of works used robotic end-effectors more closely resembling the human hand (e.g., Allegro Hand) to train dexterous manipulation policies via hand pose estimation and kinematic retargeting (Handa et al., 2020; Qin et al., 2021; Arunachalam et al., 2022; Sivakumar et al., 2022; Qin et al., 2022). In contrast to most of these works, we avoid the need to apply any explicit domain adaptation or human-robot correspondence tracking method by utilizing hand-centric visual data in which the human or robot embodiment is not visible in the demonstrations. We also train policies that can generalize to new settings and tasks without having to learn intermediate representations or reward functions. Further, we use a parallel-jaw robotic end-effector despite it being visually and kinematically dissimilar to the human hand.

One key advantage of our approach in using hand-centric human video demonstrations is the ease of collecting data in a wide range of scenarios, making it easier and quicker to gather visually and behaviorally diverse demonstrations. Related prior works that also pursued this objective are Young et al. (2020) and Song et al. (2020), which amassed diverse data using "reacher-grabber" tools. To minimize domain shift, these tools were attached to the ends of robot arms or engineered to closely resemble real parallel-jaw end-effectors. In contrast, we collect demonstrations with the human hand, which is faster and more flexible than a reacher-grabber tool, and test our policies directly on a robot with a structurally dissimilar gripper. Further, our lightweight hand-centric camera configuration for human demonstrations is simple to assemble and has nearly zero cost (aside from purchasing the camera itself), while the reacher-grabber tool proposed by Song et al. (2020) requires more sophisticated assembly and costs approximately $600 USD.

3 Preliminaries

Observation and action spaces. The observation spaces of the robot and human, $\mathcal{O}^r$ and $\mathcal{O}^h$ respectively, consist of hand-centric RGB image observations $o^r \in \mathcal{O}^r, o^h \in \mathcal{O}^h$. The robot's action space $\mathcal{A}^r$ has four dimensions, consisting of 3-DoF end-effector position control and 1-DoF gripper control. We assume that the human's action space $\mathcal{A}^h$ is the same as the robot's: $\mathcal{A}^h = \mathcal{A}^r$.

Problem definition. Our objective is to train an imitation learning policy using a dataset that combines broad hand-centric human video data with narrow robot demonstrations. By incorporating broad human video data, our goal is for the policy to generalize better than one that is trained solely on the robot dataset. While broad data can improve generalization along a number of axes, we specifically aim to improve performance in terms of environment generalization and task generalization. We define **environment generalization** as the ability to execute a learned manipulation task in a new environment outside the distribution of expert robot imitation data. We define **task generalization** as the ability to execute a new, typically long-horizon task when the expert robot demonstrations only perform an easier, shorter-horizon task.

To achieve each type of generalization, we train a policy on narrow expert robot demonstrations from one environment and task and more diverse human demonstrations that cover a broader distribution of environments or tasks. For instance, consider the task of grasping a cube in the environment generalization setting. Here, the robot demonstrations might perform cube grasping in only one environment, while the human demonstrations perform it in various environments. The goal would be to have the robot generalize to the environments covered by the human demonstrations. Now consider the task of cube pick-and-place in the task generalization setting. Here, the robot demonstrations might perform cube grasping, while the human demonstrations perform pick-and-place, which is a more difficult and longer-horizon task than grasping. The goal would be to have the robot learn to successfully execute pick-and-place even though the task was never demonstrated in the expert robot demonstrations.

4 Learning from Hand-Centric Human Video Demonstrations

We address the problem of improving the generalization of robotic manipulation policies by leveraging unlabeled human video demonstrations captured from a hand-centric camera perspective. In this section, we discuss each module of our overall framework (Figure 1). We first collect hand-centric human demonstrations of a task with a simple low-cost setup (Section 4.1). We then label the human video demonstrations with actions using an inverse dynamics model trained on robot "play" data (Section 4.2). Afterwards, we utilize the human demonstrations to train generalizable imitation learning policies (Section 4.3).

4.1 Hand-Centric Video Data Collection

Data collection setup. Prior approaches to visual imitation from human demonstrations typically use videos collected from the third-person or egocentric camera perspective, which exhibit a substantial distribution shift caused by visual differences between human and robot morphologies. We instead propose to use the hand-centric camera perspective to mitigate this domain gap. As shown in Figure 2, we secure an RGB camera to a human demonstrator's forearm via two rubber bands, and the demonstrator is immediately ready to collect demonstration videos for completing a task. While more secure ways of fastening the camera exist, we find that this simple configuration is sufficient for our purposes and only takes a few seconds to prepare. For the robot domain, the same camera is mounted onto a Franka Emika Panda robot arm via an L-bracket assemblage (see Figure 2). We use a virtual reality controller (Oculus Quest) to teleoperate the robot while collecting play data for the inverse dynamics model (Section 4.2) and expert demonstrations for the imitation learning policy (Section 4.3). Sample images captured by the hand-centric cameras are shown in Figure 3.

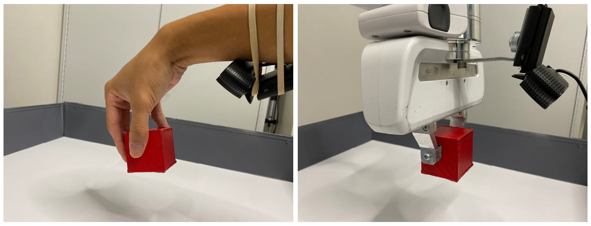

Figure 2: Human and robot hand-centric camera configurations. Fastening a USB camera on a human arm only involves two rubber bands. Mounting a camera on a Franka Emika Panda robot arm involves L-brackets, washers, and screws.

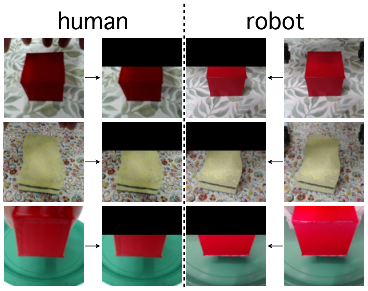

Figure 3: Sample image observations captured by the hand-centric human and robot cameras. We crop the top 36% of every image[1] in both domains.

Cropping out the hand and end-effector. To further close the domain gap between the human and robot domains, we propose masking a fixed region of all image observations o^h, o^r captured by the hand-centric human and robot cameras to hide the agent's embodiment. Specifically, we capture images of size 100×100 and zero out the top 36 rows of pixels with a script, removing the human hand and robotic end-effector from the images entirely; we denote the resulting human and robot observations as $\bar{o}^h, \bar{o}^r$, respectively. This transformation is shown in the middle two columns of Figure 3. We train the inverse dynamics models and imitation learning policies (discussed in subsequent sections) solely on images that are cropped in this manner so that we can directly utilize diverse human demonstration data, while avoiding the need to perform explicit domain adaptation.

At first glance, it may seem impossible to learn with $\bar{o}^h, \bar{o}^r$ given that the hand or end-effector is not visible. However, we observe that inverse models trained on data in this format can reasonably infer environment dynamics nonetheless due to the presence of certain visual cues. For example, the grasping and lifting of an object can be inferred even when the gripper is not visible due to visual signals such as the object "locking" into place as it is secured in the hand, the object beginning to levitate, shadows forming underneath the object, and neighboring objects shrinking in size in the hand-centric camera's field of view. Similarly, imitation learning policies can also succeed at various tasks without seeing the hand or end-effector in the frame after slightly modifying the policies' inputs (see Section 4.3 for details). Nonetheless, cropping the image does place some limitations on the tasks that can be performed, which we discuss further in Section 6.

4.2 Action Labeling of Human Video Demonstrations via Inverse Dynamics

Suppose that we have collected a visually diverse set of hand-centric expert human video demonstrations for completing a given manipulation task, $\mathcal{D}^h_{\text{exp}} = \{\bar{o}^h_t\}_{1...M}$, where M is the total number

[1] Note that due to the image cropping, the human demonstrator is not required to shape their hand such that it is visually similar to a parallel-jaw robotic gripper. For instance, their fingers can be splayed out when reaching and grasping objects, as shown in the upper-left image of Figure 3, as long as the human actions correspond to actions that are physically possible on the robot.

of timesteps. Since human video demonstrations only contain sequences of images and lack the actions taken by the human demonstrator to move between states, we cannot train an imitation learning policy on this dataset until we generate action labels for the demonstrations. The inverse dynamics model serves this precise purpose: Given image observations $\bar{o}_t^h$ and $\bar{o}_{t+1}^h$ at timesteps t and $t+1$, the inverse model predicts the action a_t giving rise to the change in observations (Nair et al., 2017; Sharma et al., 2019; Wang et al., 2019; Schmeckpeper et al., 2020; Li et al., 2021). See Appendix A.1.1 for details on the inverse model architecture.

Robot play data. An inverse model can be trained in a variety of ways. Importantly, it needs to be trained on data with sufficient diversity such that it can make accurate predictions on diverse human demonstration data. In this paper, we choose to train the inverse model using visually and behaviorally diverse task-agnostic robot "play" data that is collected in a similar manner as Lynch et al. (2020). To gather play data, a human teleoperator controlling a Franka Emika Panda robot arm executes a diverse repertoire of behaviors in an environment, exploring the observation and action spaces while interacting with objects in the scene. For example, in an environment containing two cubes, the teleoperator may wave the robotic end-effector around, reach towards a cube, grasp and lift up a cube, release and drop the cube, stack one cube on top of the other, and so on. The continuous sequences of observations captured by the hand-centric camera and the actions commanded by the teleoperator are logged and stored into a replay buffer $\mathcal{D}_{\text{play}}^r$ for inverse model training. See Appendix A.3.1 and the project website for examples of play datasets.

The key advantage of using play data is that it is easy to collect meaningful interaction data in large quantities (Lynch et al., 2020) due to the following:

- There is no need to perform frequent resets of the manipulator and objects to some initial state (which is often necessary when collecting expert demonstrations).
- There is no notion of maximum episode length or time limit (allowing a teleoperator to execute a variety of behaviors in a single contiguous stretch of time, pausing only when desired).
- The human teleoperator's knowledge of object affordances leads to interesting interactions with objects (as opposed to a script that executes purely random actions, which leads to slower exploration of the interaction space unless the data collection process is manually biased towards more meaningful interactions, as in Nair et al. (2017)).
- The play behaviors do not have to solve any particular task (which makes it easier to collect play data than expert task-aware demonstrations).

As a result, we can quickly collect a play dataset for a given environment, or set of environments, that is sufficient for training the inverse dynamics model. In addition, a single play dataset could in principle be used to acquire an inverse model that is reused for many different downstream tasks, effectively amortizing the cost of collecting it.

Inverse dynamics model training. After collecting robot play data in a given environment, we now have labeled observation-action-next-observation transitions $(\bar{o}_t^r, a_t^r, \bar{o}_{t+1}^r) \in \mathcal{D}_{\text{play}}^r$. The inverse model, parameterized by θ, takes as input $(\bar{o}_t^r, \bar{o}_{t+1}^r)$ and outputs a prediction $\hat{a}_t^r = f_\theta(\bar{o}_t^r, \bar{o}_{t+1}^r)$. We optimize the parameters θ to minimize the L_1 difference between $\hat{a}_t^r$ and a_t^r for K transitions sampled from the play dataset, using stochastic gradient descent:

$$\mathcal{L}(\hat{a}_t^r, a_t^r; \theta)_{1\ldots K} = \sum_{t=1}^{K} ||\hat{a}_t^r - a_t^r||_1.$$

Labeling human video demonstrations. Once we have trained an inverse model, we run it on all pairs of observations in the expert human demonstration dataset, $(\bar{o}_t^h, \bar{o}_{t+1}^h) \in \mathcal{D}_{\text{exp}}^h$, to automatically generate action labels for the demonstrations. We then have a labeled set of human observation-action pairs, which we denote as $\widehat{\mathcal{D}}_{\text{exp}}^h = \{(\bar{o}_t^h, \hat{a}_t^h)\}_{1\ldots M}$, where M is the total number of such pairs. We use this dataset to train an imitation learning policy, as described in the next section.

4.3 Imitation Learning with Human and Robot Demonstration Videos

Behavioral cloning. Given a dataset of human video demonstrations with inferred action labels $\widehat{\mathcal{D}}_{\text{exp}}^h = \{(\bar{o}_t^h, \hat{a}_t^h)_{1\ldots M}\}$, we train a robotic manipulation policy via behavioral cloning, a supervised learning approach to robotic imitation that learns a mapping between observations encountered by an expert demonstrator and their corresponding actions (Bain & Sammut, 1995). In this case, we treat actions $\hat{a}_t^h$ inferred by the inverse model as "ground truth" labels for the demonstrator's actions.

The behavioral cloning policy π_ϕ takes as input an RGB image observation $\bar{o}_t^h$ and outputs an action $\tilde{a}_t^h$ to best match $\hat{a}_t^h$. We minimize the negative log-likelihood of the predictions to find the optimal policy parameters ϕ^*, using stochastic gradient descent to train the model:

$$\phi^* = \text{arg min}_\phi - \sum_{t=1}^{M} \log \pi_\phi(\tilde{a}_t^h | \bar{o}_t).$$

Conditioning behavioral cloning policy on grasp state. We modify the behavioral cloning policy to be conditioned on an additional binary variable s_t^h representing the grasp state at time t (open/closed). This variable provides proprioceptive information about the manipulator that was removed from the image observations by the image cropping scheme discussed in Section 4.1; without knowing the grasp state, the policy may not be able to discern whether it has already grasped an object and could fail to exit a loop where it continuously attempts to grasp rather than proceeding to complete the task. We automatically estimate s_t^h by setting it as the prior timestep's grasping action, which is inferred by the inverse model when labeling human video demonstrations with actions. We then concatenate s_t^h to the latent image embedding and feed the result into the policy network (see Appendix A.1.2 for model architecture details). The resulting policy is $\pi_\phi(\tilde{a}_t^h | \bar{o}_t^h, s_t^h)$, and we optimize ϕ as described before.

Generalizing beyond narrow robot demonstrations. As discussed in Section 3, we collect and train a behavioral cloning policy on a narrow set of expert robot demonstrations and a broader set of human demonstrations with the goal of generalizing to the environments or tasks covered by the human data. The final objective, given N robot samples and M human samples, is to find

$$\phi^* = \text{arg min}_\phi - \sum_{t=1}^{N} \log \pi_\phi(\tilde{a}_t^r | \bar{o}_t^r, s_t^r) - \sum_{t=1}^{M} \log \pi_\phi(\tilde{a}_t^h | \bar{o}_t^h, s_t^h).$$

5 Experiments

We execute a set of experiments to study whether our framework for incorporating broad hand-centric human video demonstrations can be used to improve the generalization capabilities of a behavioral cloning policy. We focus specifically on environment generalization and task generalization, as defined in Section 3. **First**, to assess environment generalization, we test whether training a behavioral cloning policy on an additional dataset of visually diverse human video demonstrations enables it to generalize to new environments more effectively than training on narrow robot data alone. **Second**, to assess task generalization, we study whether leveraging additional human demonstrations of complex behaviors enables a behavioral cloning policy to generalize to a new, long-horizon task outside the distribution of expert robot demonstrations. **Third**, we ablate key components of our framework, such as cropping the agent's embodiment out of the image observations and conditioning the behavioral cloning policy on grasp state, to study their contributions to the final performance.

5.1 Experimental Setup

As it is difficult to generate realistic human data in simulation, we perform all experiments in the real world. As described in Section 4.1, a human demonstrator collects hand-centric human video demonstrations, and a teleoperator controls a Franka Emika Panda robot arm to collect all robot play data and expert robot video demonstrations. All observations $o^h \in \mathcal{O}^h, o^r \in \mathcal{O}^r$ are RGB images of shape $(3, H, W)$, where $H = W = 100$. Pixel values of raw images range between $[0, 255]$, but we normalize them to lie between $[-0.5, 0.5]$. As discussed in Section 3, we assume the same action space for both the human and robot: 3-DoF position control and 1-DoF gripper control. Each of the three position control actions is a continuous value ranging between $[-1, 1]$, while the gripper action is a binary value (-1: close, 1: open).

5.2 Environment Generalization Experiments

Recall that environment generalization (Section 3) is the ability to complete a learned manipulation task in a new environment lying outside the distribution of expert robot imitation data.

Tasks. When assessing environment generalization, the tasks we test on include reaching a red cube in the presence of different distractor objects, grasping a red cube placed on various environment backgrounds, and clearing different target objects off of a plate. See Figure 4 for a visualization of these tasks and Appendix A.2 for details about each task.

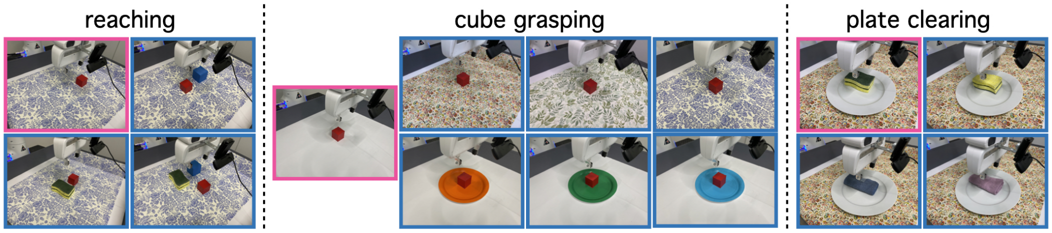

Figure 4: Tasks used for environment generalization experiments. Expert robot demonstrations are collected only in the environment configurations highlighted in pink, while expert human demonstrations are collected in the configurations highlighted in blue.

Table 1: Aggregate environment generalization and task generalization results. Behavioral cloning policies are trained on only expert robot demonstrations ("robot"), expert robot demonstrations and robot play data ("robot + play"), or expert robot and human demonstrations ("robot + human"). The policies are evaluated against environment configurations and tasks not seen in the expert robot demonstrations. Overall, incorporating the human data leads to significantly higher environment and task generalization performance than the other two methods. The average success rates and their 95% confidence intervals are computed by aggregating the results across all tasks and environment configurations in Table 3, and all tasks in Table 4.

	environment generalization		task generalization	
	success rate (%)	95% CI	success rate (%)	95% CI
robot	2.50	$[0.00, 6.02]$	0.00	$[0.00, 0.00]$
robot + play	21.67	$[17.61, 25.73]$	6.67	$[3.40, 9.93]$
robot + human (ours)	**61.67**	$[\mathbf{49.40}, \mathbf{73.93}]$	**63.33**	$[\mathbf{51.56}, \mathbf{75.11}]$

Datasets. For each task, we collect narrow expert robot demonstrations and visually diverse expert human demonstrations. For example, for the cube grasping task, the robot demonstrations are collected from one environment, while the human demonstrations are collected from multiple environments with different backgrounds. We also collect a robot play dataset for an inverse model that is shared with a task generalization experiment involving similar objects. See Appendix A.3 for details on all expert demonstration datasets and robot play datasets.

Methods to compare. As our objective is to study whether incorporating additional diverse human demonstrations achieves increased environment generalization, we compare a behavioral cloning policy trained on both human and robot demonstrations against a baseline policy trained only on robot demonstrations. In addition, to assess whether any improvements in generalization are simply correlated to the increase in dataset size, we also compare against a behavioral cloning policy trained on both expert robot demonstrations and robot play data, as the play datasets are larger than the human demonstration datasets. All policies are trained from scratch, with data points being uniformly sampled from each combined dataset.

Results. Results for the environment generalization experiments are shown in the left half of Table 1. We observe that we can directly incorporate diverse human video demonstrations into policy training to achieve a significant increase in generalization. The policy is able to generalize to new environment configurations outside the distribution of expert robot demonstrations (see fine-grained results in Table 3). To our knowledge, this marks the first time that a real robot policy has been successfully trained end-to-end on hand-centric human demonstrations. On the other hand, in many cases, the policy trained only on a limited set of robot demonstrations fails completely, as shown in Figure 6(c), since the novel out-of-distribution stimuli confuse the policy. In addition, we see that a policy also trained on the full play dataset, which is larger than the set of human demonstrations, does not perform as well as one trained on the human demonstrations, verifying that the increase in generalization performance is not simply a function of training dataset size. Videos of the learned policies are available on our project website.

5.3 Task Generalization Experiments

Recall that task generalization (Section 3) is the ability to complete a new, typically long-horizon task outside the distribution of expert robot demonstrations, which may only perform a simpler, short-horizon task.

Tasks. To assess task generalization, the longer-horizon tasks we test on include stacking a red cube on top of a blue cube, picking-and-placing a red cube onto a green plate, and clearing a green sponge from a plate. See Figure 5 for a visualization of these tasks.

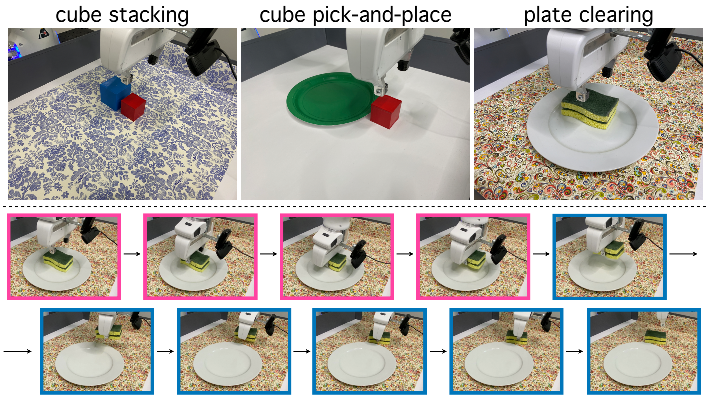

Figure 5: Tasks used in task generalization experiments. Expert robot demonstrations perform an easier, shorter-horizon task, such as grasping (highlighted in pink); expert human demonstrations either perform the full long-horizon task or portions of the task that are missing in the robot demonstrations (highlighted in blue).

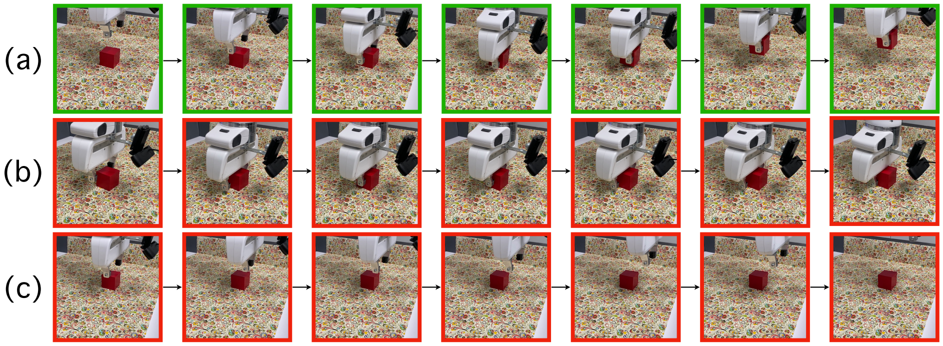

Figure 6: Sample behavioral cloning policy rollouts on the cube grasping environment generalization task. Here, the policies are trained on (a) both robot and human demonstrations, (b) both robot and human demonstrations but without conditioning the policy on grasp state, or (c) only robot demonstrations. Green indicates success; red indicates failure. In (b), the robot does not realize that it has already grasped the cube and repeatedly reattempts to do so, ultimately failing to lift the object. In (c), the robot fails to even reach the cube, as the novel visual stimuli confuse the policy.

Datasets. As in Section 5.2, we collect expert robot demonstrations, human demonstrations, and shared robot play data. The robot demonstrations perform a simple task (e.g., cube grasping), and the human demonstrations perform one of the more difficult, longer-horizon tasks above (e.g., cube stacking). Appendix A.3 gives the full details on all datasets used in the experiments.

Methods to compare. We evaluate the task generalization of the same three behavioral cloning policies discussed in Section 5.2.

Results. As shown in the right half of Table 1, training the behavioral cloning policy on the hand-centric human video demonstrations substantially improves the policy's task generalization performance compared to using robot data alone. Intuitively, a policy trained on robot demonstrations that never perform the desired long-horizon task is incapable of performing the task at test time. On the other hand, a policy that is also trained on robot play data can occasionally execute the desired task since the play dataset contains a collection of behaviors, some of which can be useful for solving the task. However, as the play dataset is task-agnostic, the behavioral cloning policy often struggles to learn one fluid sequence of actions for solving a specific long-horizon task.

5.4 Ablation Experiments

In the next few experiments, we ablate some key components of our framework to observe the effects that they have on the final generalization performance. We test the resulting policies on one representative task from the environment generalization setting (cube grasping) and another from the task generalization setting (plate clearing).

Table 2: Ablation experiments results. We observe that removing either the image cropping or grasp state conditioning generally leads to greatly reduced success rates, validating their important contributions to the final generalization performance. Success rates are aggregated from the finer-grained results in Table 5.

	success rate (%)	95% CI
original method	$\mathbf{54.29}$	$[\mathbf{39.56}, \mathbf{69.01}]$
no image crop	24.29	$[6.21, 42.36]$
no grasp state	28.57	$[9.72, 47.42]$

Training on uncropped images. In the previous experiments, we used the image cropping scheme discussed in Section 4.1 for all observations. Now we remove this cropping transformation entirely to assess whether it is an important component of our framework for leveraging hand-centric human demonstrations to improve generalization. Given uncropped robot play data where the end-effector is now visible, we train an inverse model to predict the dynamics and use the model to infer action labels for uncroppped human demonstrations, regardless of the domain shift caused by apparent visual differences between human hand and robotic gripper. We train a behavioral cloning policy on uncropped versions of the expert robot and human demonstrations used in our previous experiments, and we compare this policy against the policy trained according to our original framework.

Behavioral cloning without conditioning on grasp state. In a separate ablation, we modify the behavioral cloning policy such that it is no longer conditioned on the binary (open/close) grasp state. We reuse the expert robot and human demonstrations from the previous experiments and simply train a new behavioral cloning policy without the grasp state conditioning. We compare this policy against our original conditioned behavioral cloning policy.

Results. A summary of the ablation results are shown in Table 2 (detailed results are shown in Table 5). We generally observe reduced generalization from removing the image cropping scheme and the grasp state conditioning. Qualitatively, the policy often fails to even reach the target object in several cases when using uncropped images; we attribute this to the distribution shift between human and robot observations, leading to inaccurate action predictions from the inverse model. When not conditioning on grasp state, a common failure mode we observe in the cube grasping task is the repeated attempts to grasp the cube rather than lifting it, as the robot does not know that it has already secured the object. An illustration of this behavior is shown in Figure 6(b). Overall, these results indicate that both components of our approach are important to successfully leverage hand-centric human video demonstrations.

6 Conclusion

This work presents a novel yet simple framework for leveraging diverse hand-centric human video demonstrations and displays its potential to expand the generalization capabilities of vision-based manipulators. We utilize the hand-centric camera perspective and an image cropping scheme to largely close the domain gap between human and robot data and bypass explicit domain adaptation entirely. Our experiments show that our framework enables an imitation learning policy to generalize to new environments and new tasks that lie outside the distribution of expert robot demonstrations.

Limitations and future work. In our framework, we assume that the human hand and robotic end-effector share the same action space, constraining human demonstrations to only perform actions that are possible on a real robot. For example, we expect the approach to perform poorly if the human performs dexterous in-hand manipulation. Our proposal to crop images such that the embodiment is not visible introduces some limitations as well. For example, if a target object is small enough that it does not appear in the uncropped portion of the image, it may be difficult for the inverse model to infer actions that manipulate the object due to a dearth of sufficient visual cues. Relatedly, actions that do not have any visual effect on objects in the scene (e.g., grasping nothing while the gripper is in mid-air) may be impossible to infer by the inverse model. However, this should not be an issue for most downstream use-cases, as the only actions that have significance when learning a policy are those that have some effect on the agent's environment. Lastly, our method involves collecting a robot play dataset via teleoperation to train the inverse model. While this process is inexpensive, as discussed in Section 4.2, in the future we hope to automate play data collection nonetheless, e.g., by training a behavioral cloning policy on a small play dataset and sampling actions during inference to encourage exploration, as done in Dinyari et al. (2020).

Reproducibility Statement

We discuss in detail our model architectures (Appendix A.1), robotic manipulation tasks (Appendix A.2), and play datasets and expert robot and human demonstration datasets (Appendix A.3) to promote the reproducibility of our work. We also plan to release the play datasets and expert robot and human demonstration datasets we collected on our project website upon publication. Code that interfaces with the real robot is specific to our hardware and network setup; therefore, we plan to release parts of our code that would be compatible with other platforms, also on our project website.

References

Brenna D Argall, Sonia Chernova, Manuela Veloso, and Brett Browning. A survey of robot learning from demonstration. *Robotics and autonomous systems*, 57(5):469–483, 2009. 2

Sridhar Pandian Arunachalam, Sneha Silwal, Ben Evans, and Lerrel Pinto. Dexterous imitation made easy: A learning-based framework for efficient dexterous manipulation. *arXiv preprint arXiv:2203.13251*, 2022. 3

Christopher G Atkeson and Stefan Schaal. Robot learning from demonstration. In *ICML*, volume 97, pp. 12–20, 1997. 2

Michael Bain and Claude Sammut. A framework for behavioural cloning. In *Machine Intelligence 15*, pp. 103–129, 1995. 5

Annie S Chen, Suraj Nair, and Chelsea Finn. Learning generalizable robotic reward functions from" in-the-wild" human videos. *arXiv preprint arXiv:2103.16817*, 2021. 3

Neha Das, Sarah Bechtle, Todor Davchev, Dinesh Jayaraman, Akshara Rai, and Franziska Meier. Model-based inverse reinforcement learning from visual demonstrations. *arXiv preprint arXiv:2010.09034*, 2020. 2

Rostam Dinyari, Pierre Sermanet, and Corey Lynch. Learning to play by imitating humans. *arXiv preprint arXiv:2006.06874*, 2020. 9

Ankur Handa, Karl Van Wyk, Wei Yang, Jacky Liang, Yu-Wei Chao, Qian Wan, Stan Birchfield, Nathan Ratliff, and Dieter Fox. Dexpilot: Vision-based teleoperation of dexterous robotic hand-arm system. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 9164–9170. IEEE, 2020. 3

Gillian M Hayes and John Demiris. *A robot controller using learning by imitation*. University of Edinburgh, Department of Artificial Intelligence, 1994. 2

Ananth Jonnavittula and Dylan P Losey. Communicating robot conventions through shared autonomy. *arXiv preprint arXiv:2202.11140*, 2022. 2

Ryan Julian, Benjamin Swanson, Gaurav S Sukhatme, Sergey Levine, Chelsea Finn, and Karol Hausman. Never stop learning: The effectiveness of fine-tuning in robotic reinforcement learning. *arXiv preprint arXiv:2004.10190*, 2020. 2

Sateesh Kumar, Jonathan Zamora, Nicklas Hansen, Rishabh Jangir, and Xiaolong Wang. Graph inverse reinforcement learning from diverse videos. *arXiv preprint arXiv:2207.14299*, 2022. 2

Jiayi Li, Tao Lu, Xiaoge Cao, Yinghao Cai, and Shuo Wang. Meta-imitation learning by watching video demonstrations. In *International Conference on Learning Representations*, 2021. 2, 5

YuXuan Liu, Abhishek Gupta, Pieter Abbeel, and Sergey Levine. Imitation from observation: Learning to imitate behaviors from raw video via context translation. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1118–1125. IEEE, 2018. 2

Corey Lynch, Mohi Khansari, Ted Xiao, Vikash Kumar, Jonathan Tompson, Sergey Levine, and Pierre Sermanet. Learning latent plans from play. In *Conference on robot learning*, pp. 1113–1132. PMLR, 2020. 5

Ajay Mandlekar, Danfei Xu, Roberto Martín-Martín, Yuke Zhu, Li Fei-Fei, and Silvio Savarese. Human-in-the-loop imitation learning using remote teleoperation. *arXiv preprint arXiv:2012.06733*, 2020. 2

Oier Mees, Markus Merklinger, Gabriel Kalweit, and Wolfram Burgard. Adversarial skill networks: Unsupervised robot skill learning from video. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4188–4194. IEEE, 2020. 3

Ashvin Nair, Dian Chen, Pulkit Agrawal, Phillip Isola, Pieter Abbeel, Jitendra Malik, and Sergey Levine. Combining self-supervised learning and imitation for vision-based rope manipulation. In *2017 IEEE international conference on robotics and automation (ICRA)*, pp. 2146–2153. IEEE, 2017. 5

Suraj Nair, Aravind Rajeswaran, Vikash Kumar, Chelsea Finn, and Abhinav Gupta. R3m: A universal visual representation for robot manipulation. *arXiv preprint arXiv:2203.12601*, 2022. 3

Anh Nguyen, Dimitrios Kanoulas, Luca Muratore, Darwin G Caldwell, and Nikos G Tsagarakis. Translating videos to commands for robotic manipulation with deep recurrent neural networks. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3782–3788. IEEE, 2018. 2

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

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019. 13

Yuzhe Qin, Yueh-Hua Wu, Shaowei Liu, Hanwen Jiang, Ruihan Yang, Yang Fu, and Xiaolong Wang. Dexmv: Imitation learning for dexterous manipulation from human videos. *arXiv preprint arXiv:2108.05877*, 2021. 3

Yuzhe Qin, Hao Su, and Xiaolong Wang. From one hand to multiple hands: Imitation learning for dexterous manipulation from single-camera teleoperation. *arXiv preprint arXiv:2204.12490*, 2022. 3

Karinne Ramirez-Amaro, Michael Beetz, and Gordon Cheng. Transferring skills to humanoid robots by extracting semantic representations from observations of human activities. *Artificial Intelligence*, 247:95–118, 2017. 2

Karl Schmeckpeper, Oleh Rybkin, Kostas Daniilidis, Sergey Levine, and Chelsea Finn. Reinforcement learning with videos: Combining offline observations with interaction. *arXiv preprint arXiv:2011.06507*, 2020. 2, 3, 5

Pierre Sermanet, Kelvin Xu, and Sergey Levine. Unsupervised perceptual rewards for imitation learning. *arXiv preprint arXiv:1612.06699*, 2016. 3

Pierre Sermanet, Corey Lynch, Yevgen Chebotar, Jasmine Hsu, Eric Jang, Stefan Schaal, Sergey Levine, and Google Brain. Time-contrastive networks: Self-supervised learning from video. In *2018 IEEE international conference on robotics and automation (ICRA)*, pp. 1134–1141. IEEE, 2018. 3

Pratyusha Sharma, Deepak Pathak, and Abhinav Gupta. Third-person visual imitation learning via decoupled hierarchical controller. *Advances in Neural Information Processing Systems*, 32, 2019. 2, 5

Aravind Sivakumar, Kenneth Shaw, and Deepak Pathak. Robotic telekinesis: learning a robotic hand imitator by watching humans on youtube. *arXiv preprint arXiv:2202.10448*, 2022. 3

Laura Smith, Nikita Dhawan, Marvin Zhang, Pieter Abbeel, and Sergey Levine. Avid: Learning multi-stage tasks via pixel-level translation of human videos. *arXiv preprint arXiv:1912.04443*, 2019. 2

Shuran Song, Andy Zeng, Johnny Lee, and Thomas Funkhouser. Grasping in the wild: Learning 6dof closed-loop grasping from low-cost demonstrations. *IEEE Robotics and Automation Letters*, 5(3):4978–4985, 2020. 3

Angelina Wang, Thanard Kurutach, Kara Liu, Pieter Abbeel, and Aviv Tamar. Learning robotic manipulation through visual planning and acting. *arXiv preprint arXiv:1905.04411*, 2019. 5

Haoyu Xiong, Quanzhou Li, Yun-Chun Chen, Homanga Bharadhwaj, Samarth Sinha, and Animesh Garg. Learning by watching: Physical imitation of manipulation skills from human videos. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 7827–7834. IEEE, 2021. 2

Shuo Yang, Wei Zhang, Weizhi Lu, Hesheng Wang, and Yibin Li. Learning actions from human demonstration video for robotic manipulation. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1805–1811. IEEE, 2019. 2, 3

Yezhou Yang, Yi Li, Cornelia Fermuller, and Yiannis Aloimonos. Robot learning manipulation action plans by" watching" unconstrained videos from the world wide web. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 29, 2015. 2

Sarah Young, Dhiraj Gandhi, Shubham Tulsiani, Abhinav Gupta, Pieter Abbeel, and Lerrel Pinto. Visual imitation made easy. In *Conference on Robot Learning (CoRL)*, 2020. 3

Tianhe Yu, Chelsea Finn, Annie Xie, Sudeep Dasari, Tianhao Zhang, Pieter Abbeel, and Sergey Levine. One-shot imitation from observing humans via domain-adaptive meta-learning. *arXiv preprint arXiv:1802.01557*, 2018. 2, 3

Kevin Zakka, Andy Zeng, Pete Florence, Jonathan Tompson, Jeannette Bohg, and Debidatta Dwibedi. Xirl: Cross-embodiment inverse reinforcement learning. In *Conference on Robot Learning*, pp. 537–546. PMLR, 2022. 2, 3

Tianhao Zhang, Zoe McCarthy, Owen Jow, Dennis Lee, Xi Chen, Ken Goldberg, and Pieter Abbeel. Deep imitation learning for complex manipulation tasks from virtual reality teleoperation. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5628–5635. IEEE, 2018. 2

Yuxiang Zhou, Yusuf Aytar, and Konstantinos Bousmalis. Manipulator-independent representations for visual imitation. *arXiv preprint arXiv:2103.09016*, 2021. 2, 3

Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pp. 2223–2232, 2017. 2

A Appendix

A.1 Model Architectures

In this section, we discuss model architecture details. We implement and train all models using PyTorch (Paszke et al., 2019).

A.1.1 Inverse Dynamics Model Architecture

The inverse dynamics model is a convolutional neural network with 4 convolutional layers followed by 2 feedforward layers. Each convolutional and feedforward layer is followed by a batch normalization layer and ReLU activation layer. For every convolutional layer, the number of convolutional filters is 128, kernel size is 3, stride is 1 (except for the first layer, whose stride is 2), and padding is 0. The latent embedding size of the second feedforward layer is 200. We use early fusion, i.e., two consecutive image observations are concatenated channel-wise and then fed into the first convolutional layer. The full network outputs a 4-DoF action prediction that takes the agent from one observation to the next timestep's observation.

We train every inverse model with random shifts data augmentation. For every pair of 100×100 image observations, we pad each side by 4 pixels and randomly crop a 100×100 region out of the result. The same augmentation is applied to both images in a given pair so as to not perturb the original dynamics captured in the images. We only apply this augmentation with 80% probability, as we found that the resulting model is as accurate as one trained with 100% probability, yet it trains faster because it does not need to compute the augmentation 20 percent of the time.

A.1.2 Behavioral Cloning Policy Network Architecture

The behavioral cloning policy network consists of an image encoder with mostly the same architecture as the inverse model, except that the number of convolutional filters per layer is 32, and the hidden size of the second feedforward layer is 50. Unlike the inverse model, the policy network acts on one image at a time rather than a pair. After the image encoder portion, the policy network consists of an additional two feedforward layers (with a latent dimensionality of 64) representing the policy head. Further, the policy is conditioned on a 1-dimensional grasp state variable as described in Section 4.3; this variable is concatenated with the 50-dimensional latent embedding output by the second feedforward layer of the image encoder, and the resulting 51-dimensional embedding is passed on to the policy head, which outputs a 4-DoF action prediction that best imitates an expert demonstrator's action given some input observation.

As with the inverse model, we apply random shifts data augmentation while training the behavioral cloning policy.

A.2 Tasks

In this section, we discuss the tasks introduced in Section 5 in more detail. The environment generalization tasks include the following:

- **reaching**: The goal is to reach the end-effector towards the red cube. The environment contains just the red cube, the red cube and a blue cube distractor, the red cube and a green sponge distractor, or all three objects. Initial positions of the objects are randomized.
- **cube grasping**: The goal is to grasp the cube and lift it off the ground. The environment only contains on object: the cube. The background can be one of seven: a plain white background, rainbow floral texture, green floral texture, blue floral texture, orange plate, green plate, or blue plate. The initial position of the cube is randomized.
- **plate clearing**: The goal is to grasp a target object resting on a plate, lift it up, transfer it to a location off to the right of the plate, and release it. The target object is either a green sponge, yellow sponge, blue towel, or pink towel. The initial position of the target object is randomized.

We now describe the task generalization tasks:

- **cube stacking**: The goal is to grasp the red cube, lift it up, and stack it on top of the blue cube. Initial positions of the cubes are fixed relative to each other, but vary relative to the background, which is a blue floral texture. Expert robot demonstrations perform cube grasping, while human demonstrations perform full cube stacking or portions of the task that follow the grasping part.

- **cube pick-and-place**: The goal is to grasp the red cube, lift it up, move it over to the green plate, and release it onto the plate. The initial positions of the cube and plate are fixed relative to each other, but vary relative to the background. Expert robot demonstrations perform cube grasping, while human demonstrations perform full cube pick-and-place or portions of the task that follow the grasping part.
- **plate clearing**: The goal is the same as described above for the plate clearing environment generalization task. However, here we only manipulate one target object: the green sponge. The initial position of the sponge is randomized. Expert robot demonstrations perform sponge grasping, while human demonstrations perform full plate clearing or portions of the task that follow the grasping part.

Please see our project website for further details and visualizations of data collected in these tasks and environments.

A.3 Datasets

A.3.1 Robot Play Datasets

We collect three robot play datasets and train three corresponding inverse models. Each inverse model is shared across one environment generalization experiment and one task generalization experiment. We discuss the details of each play dataset below:

- **reaching and cube stacking dataset**: We collect $20,000$ timesteps of play data at 5 Hz (approximately 67 minutes) in an environment with a blue floral background and 3 objects: a red cube, a blue cube, and a green sponge. The play data behaviors include waving the end-effector around, reaching towards each object, grasping and lifting up each object, releasing and dropping an object, stacking an object on top of another, and so on. This play dataset is shared for the reaching environment generalization and cube stacking task generalization tasks.
- **cube grasping and cube pick-and-place dataset**: We collect $51,400$ steps of play data at 5 Hz (approximately 171 minutes) in multiple environments containing a red cube, each having a different background that the red cube rests on: a plain white background, rainbow floral texture, green floral texture, blue floral texture, orange plate, green plate, or blue plate. The play data behaviors include waving the end-effector around, reaching towards the cube, grasping and lifting up the cube, releasing and dropping the cube, and so on. This play dataset is shared for the cube grasping environment generalization and cube pick-and-place task generalization tasks.
- **plate clearing environment generalization and task generalization dataset**: We collect $20,000$ steps of play data at 5 Hz (approximately 67 minutes) in multiple environment configurations, each containing a different target object: green sponge, yellow sponge, blue towel, and pink towel. The play data behaviors include waving the end-effector around, reaching towards the objects, grasping and lifting up the objects, releasing and dropping the objects, and so on. This play dataset is shared for both plate clearing environment generalization and task generalization experiments.

Please see our project website for visualizations of these play datasets. We also release the corresponding dataset files on the website.

A.3.2 Expert Demonstration Datasets

In each environment generalization or task generalization experiment, we collect a set of expert robot demonstrations and a set of expert human demonstrations. Below we discuss details of the datasets collected for each experiment. Please refer to Figure 4 and Figure 5 for a visualization of the distribution of environments or tasks that the robot and human datasets are each collected from. All demonstrations are collected at 5 Hz, as is done while collecting the play datasets.

- **reaching (environment generalization)**: We collect 60 robot demonstrations with no distractor objects and 100 human demonstrations with both the blue cube and green sponge as distractors.
- **cube grasping (environment generalization)**: We collect 100 robot demonstrations only in an environment with a plain white background and 20 human demonstrations from each of the following environments: rainbow floral texture, green floral texture, blue floral texture, orange plate, green plate, and blue plate.
- **plate clearing (environment generalization)**: We collect 30 robot demonstrations with just the green sponge as a target object and 20 human demonstrations with each of the following target objects: yellow sponge, blue towel, and pink towel.

- **cube stacking (task generalization)**: We collect 25 robot demonstrations and 130 human demonstrations. The robot demonstrations perform red cube grasping; the human demonstrations perform the cube stacking (stack red cube onto blue cube) or portions of the task that follow the grasping part. For this task, a majority of the human demonstrations do the latter and are thus able to be collected very quickly.
- **cube pick-and-place (task generalization)**: We collect 20 robot demonstrations and 70 human demonstrations. The robot demonstrations perform cube grasping; the human demonstrations perform cube pick-and-place (place cube onto plate) or portions of the task that follow the grasping part.
- **plate clearing (task generalization)**: We collect 40 robot demonstrations and 25 human demonstrations. The robot demonstrations perform sponge grasping; the human demonstrations perform plate clearing (remove sponge off of plate) or portions of the task that follow the grasping part.

Please see our project website for visualizations of these expert demonstration datasets. We also release the corresponding dataset files on the website.

A.4 Detailed Experimental Results

Below are the full experimental results that were aggregated to produce Table 1 and Table 2 in Section 5. All success rates are evaluated over 10 trials, with initial object positions randomized or fixed according to the configurations described in Appendix A.2. Please see our project website for videos of the trained policies.

Table 3: Full environment generalization results. We compare three behavioral cloning policies, each trained on a different set of hand-centric video data: only expert robot demonstrations ("robot"), expert robot demonstrations and robot play data ("robot + play"), or expert robot and human demonstrations ("robot + human"). For each task, the expert robot demonstrations are collected only in the *italicized* environment configuration in the second column; play data and expert human demonstrations are collected in the configurations below the dotted lines. Thus, the non-italicized environment configurations are out-of-distribution with respect to the expert robot demonstrations. Overall, leveraging the hand-centric human demonstration data leads to significantly higher environment generalization performance than using the robot demonstration data alone. Each success rate is computed over 10 test rollouts of the behavioral cloning policy.

		success rate (%)		
task	environment configuration	robot	robot + play	robot + human
reaching	*no distractors (only red cube)*	90	90	90
	+ blue cube distractor	20	20	**90**
	+ green sponge distractor	10	20	**90**
	+ blue cube, green sponge distractors	0	20	**80**
cube grasping	*white background*	90	90	90
	rainbow floral background	0	30	**80**
	green floral background	0	20	**60**
	blue floral background	0	30	**60**
	cube on orange plate	0	20	**40**
	cube on green plate	0	10	**20**
	cube on blue plate	0	20	**50**
plate clearing	*green sponge on plate*	60	**70**	**70**
	yellow sponge on plate	0	30	**40**
	blue towel on plate	0	10	**70**
	pink towel on plate	0	30	**60**

Table 4: Full task generalization results. We compare three behavioral cloning policies, each trained on a different set of hand-centric video data: only expert robot demonstrations ("robot"), expert robot demonstrations and robot play data ("robot + play"), or expert robot and human demonstrations ("robot + human"). For each experiment, the expert robot demonstrations are collected only for the *italicized* task in the second column; expert human demonstrations are collected for the longer-horizon tasks below the dotted lines. Overall, leveraging the hand-centric human demonstration data allows the policies to generalize to tasks that are unseen in the expert robot demonstrations. Each success rate is computed over 10 test rollouts of the behavioral cloning policy.

		success rate (%)		
experiment	task	robot	robot + play	robot + human
1	*cube grasping*	90	90	90
	cube stacking	0	10	**40**
2	*cube grasping*	90	**100**	90
	cube pick-and-place	0	0	**80**
3	*sponge grasping*	100	100	100
	clearing sponge from plate	0	10	**70**

Table 5: Full ablation experiments results. The policy trained on uncropped images fails drastically on the last three environment configurations as it never reaches the cube. The policy that is not conditioned on grasp state often encounters a failure mode in which it repeatedly reattempts to grasp an object even though it has already grasped it. Such a failure mode occurs because the robot cannot see the end-effector. Each success rate is computed over 10 test rollouts of the behavioral cloning policy.

environment generalization

		success rate (%)		
task	environment configuration	original	no image cropping	no grasp state
cube grasping	*white background*	90	40	90
	rainbow floral background	**80**	60	50
	green floral background	**60**	40	40
	blue floral background	**60**	40	20
	cube on orange plate	**40**	0	0
	cube on green plate	**20**	0	10
	cube on blue plate	**50**	0	10

task generalization

		success rate (%)		
experiment	task	original	no image cropping	no grasp state
1	*sponge grasping*	**100**	90	90
	clearing sponge from plate	70	30	70