Learning from Human Videos for Robotic Manipulation

Aditya Kannan

CMU-CS-23-124

July 2023

Computer Science Department
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Thesis Committee
Prof. Deepak Pathak, Chair
Prof. Abhinav Gupta

Submitted in partial fulfillment of the requirements for the degree of Master of Science.

Keywords: Dexterous Manipulation, Reward Learning, Learning from Videos, Deep Learning

Abstract

In recent years, many works in Computer Vision and NLP have demonstrated remarkable steps toward generalization through the collection and use of diverse datasets. However, collecting large-scale robot datasets is often difficult due to many reasons including cost, reliance on human supervision, and safety. An alternative approach is to take advantage of the accessibility and wide variety of human videos available on the internet. In this thesis, we investigate two approaches that use human videos for robotic control without relying on robot demonstrations.

In our first work, we use human videos as a prior for dexterous manipulation. Humans are able to perform a host of skills with their hands, from making food to operating tools. In this work, we investigate these challenges, especially in the case of soft, deformable objects as well as complex, relatively long-horizon tasks. However, learning such behaviors from scratch can be data inefficient. To circumvent this, we propose a novel approach, DEFT (**DE**xterous **F**ine-**T**uning for Hand Policies), that leverages human-driven priors, which are executed directly in the real world. In order to *improve* upon these priors, DEFT involves an efficient online optimization procedure. With the integration of human-based learning and online fine-tuning, coupled with a soft robotic hand, DEFT demonstrates success across various tasks, establishing a robust, data-efficient pathway toward general dexterous manipulation.

In our second work, we introduce a method to learn a domain- and agent-agnostic reward function from large-scale egocentric human data. Prior approaches that use human data for reward learning either require a small sample of in-domain robot data in training or need a goal image specified in the robot's environment. In this work, we focus on the setting where only human data is available at training and test time. Our approach trains a multi-task reward function that learns to discriminate between different tasks by observing the changes in the environment. We show that our method has strong performance on three simulation tasks *without* the help of robot demonstrations in training or in-domain goals.

The source code for this thesis document is available at:

https://github.com/adityak77/masters-thesis

Acknowledgments

I have been incredibly fortunate and grateful to have received the opportunity to pursue my thesis. To the countless mentors, friends, and family who invested in me and encouraged me to be bold, I dedicate this work to you.

First, I would like to thank Prof. Deepak Pathak, my thesis advisor, who gave me the opportunity to be in an environment where I could work hard, learn, and succeed. His ambition and encouragement to work on big ideas will always be an inspiration for me. I would also like to thank Prof. Abhinav Gupta for serving on my thesis committee and providing feedback on my thesis.

I would especially like to thank Shikhar Bahl for his guidance and mentorship throughout my time in the lab. I am grateful for his advice, encouragement, and his role in helping me become more independent in my research.

I would also like to thank my collaborators, friends, and everyone else who elevated my experience throughout my time in this lab. This includes (in alphabetical order) Ananye Agarwal, Ellis Brown, Lili Chen, Xuxin Cheng, Shivam Duggal, Zipeng Fu, Konwoo Kim, Alex Li, Edward Li, Pragna Mannam, Russell Mendonca, Mihir Prabhudesai, Ankit Ramchandani, Aravind Sivakumar, Kenny Shaw, Shagun Uppal, Haoyu Xiong, and Yufei Ye.

I am privileged to have had countless mentors early in my life who set me on the path I am today. This includes my middle school teacher Jack Black, who instilled in me a strong work ethic, and my high school physics teacher Michael O'Byrne, who taught me to always keep pushing beyond my comfort zone. Furthermore, I am indebted to Prof. Abraham Flaxman, Prof. Shih-Chieh Hsu, and the PROMYS program for taking a chance on me and giving me the opportunity to actively work on research when I was still in high school. Continuing into undergrad, I would like to thank Prof. Hosein Mohimani, whose guidance cemented my desire to pursue master's research.

I would like to thank all the family and friends who have provided me with incredible support, inspiration, and strength. I owe a great debt to my parents for their love and support without which I surely would not have succeeded in my master's program.

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

Robotic manipulation is an important problem for deploying robots that can perform everyday tasks in the real world. Human videos include numerous examples of humans interacting with objects, which can be a very useful prior for robots. After all, most common objects and tools are built with human use in mind, so observing how humans behave in the world can provide essential information that can be used towards manipulation tasks.

Another important goal in robotics is generalization in terms of environments, tasks, robot embodiments, etc... One of the reasons that other areas, such as Computer Vision and NLP, have shown strong generalization with deep learning is that large, annotated datasets can be collected relatively cheaply for vision and text. On the other hand, collecting demonstrations with robots is difficult because it is expensive, time-consuming, and requires human supervision for managing resets and safety concerns. Additionally, because robots are not widely deployed in society, datasets of large magnitude do not naturally exist on the internet.

Human videos, however, are plentiful on the internet and accessible at scale. Computer vision tools are also advanced enough to be able to reliably extract task-relevant information involving the actions of humans and objects in the scene. As a result, human videos could potentially bring us closer to the grand goal of general-purpose manipulation. Relying on human videos comes with many challenges as well, including the lack of annotated human actions and the embodiment gap between human and robot morphologies. In this thesis, we present two works that aim to overcome these challenges and enable robots to interact with objects by learning from human videos.

- In Chapter 2, we investigate how human videos can be used as a prior for dexterous manipulation. Because most tools and household objects are designed to be manipulated by human hands, an anthropomorphic hand is a natural step to enable human-like interaction with common objects. Our method, DEFT, explores in the real world with a safe and durable soft hand. Because learning a policy from scratch is inefficient, we develop a prior from human videos that learns to predict affordances for grasping the object. Experiments demonstrate that DEFT can perform a variety of challenging tasks effectively.

- In Chapter 3, we investigate how we can learn generalizable reward functions from *only* human videos. While most methods that learn rewards from human videos require either (1) robot demonstrations in training or (2) goal images specified in the robot domain, we show that we can learn a multi-task reward without any in-domain information. In order to bridge the visual morphology gap, our insight is to learn a representation on data where the agents are visually removed. We show strong results on three simulation tasks. In the future, we aim to scale to more tasks.

Chapter 2

Dexterous Fine-Tuning for Hand Policies

2.1 Introduction

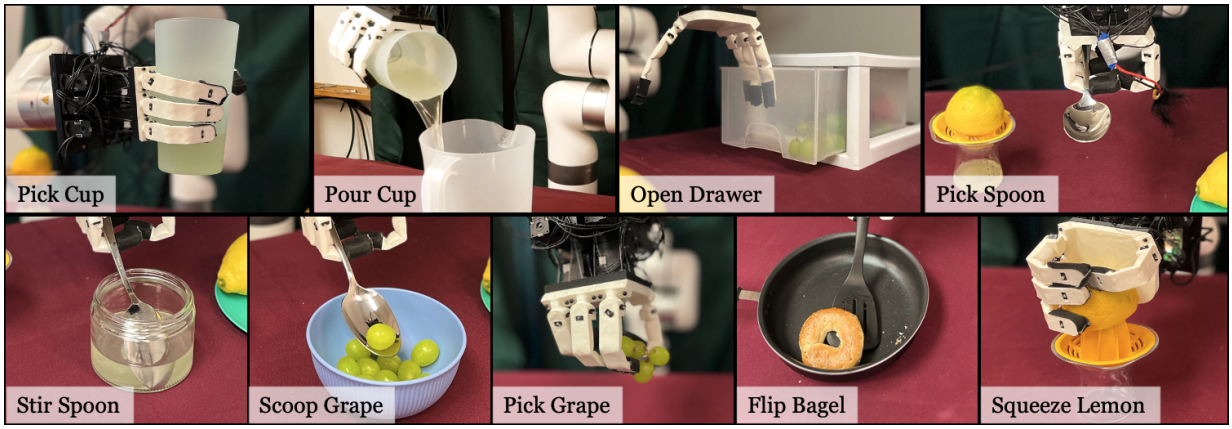

Figure 2.1: We present DEFT, a novel approach that can learn complex, dexterous tasks in the real world in an efficient manner. DEFT manipulates tools and soft objects without any robot demonstrations.

The longstanding goal of robot learning is to build robust agents that can perform long-horizon tasks autonomously. This could for example mean a self-improving robot that can build furniture or an agent that can cook for us. A key aspect of most tasks that humans would like to perform is that they require complex motions that are often only achievable by hands–consider the task of tying shoelaces. Such a task would be impossible with a parallel jaw gripper. Therefore, in this work, we investigate real-world dexterous manipulation, its challenges, and its deployment in the real world.

A key challenge in deploying policies in the real world, especially with robotic hands, is that there exist many failure modes. Controlling a dexterous hand is much harder than end-effectors due to larger action spaces and complex dynamics. This problem becomes even more difficult when contacts with objects are more involved, as there can be many points of contact on a robotic hand. To address this, one option is to *improve* directly

in the real world via *practice*. Traditionally, reinforcement learning (RL) and imitation learning (IL) techniques have been used to deploy hands-on tasks such as in-hand rotation or grasping. To be proficient at any skill a lot of practice or data is needed. This is often the case as setups are built so that it is either easy to simulate in the real world or robust to practice. However, the real world contains tasks that one cannot simulate (such as manipulation of soft objects like food) or difficult settings in which the robot cannot practice (sparse long-horizon tasks like assembly). How can we build an approach that can scale to such tasks?

There are several issues with current approaches for practice and improvement in the real world. Robot hardware often breaks, especially with the amount of contact to learn dexterous tasks like operating tools. We thus investigate using a *soft anthropomorphic hand* [Man+23], which can easily run in the real world without failures or breaking. This soft anthropomorphic hand is well-suited to our approach as it is flexible and can gently handle object interactions. The hand does not get damaged by the environment and is robust to continuous data collection. Due to its human-like proportions and morphology, retargeting human hand grasps to robot hand grasps is made simpler.

Unfortunately, this hand is difficult to simulate due to its softness. Directly learning from scratch is also difficult as we would like to build *generalizable policies*, and not practice for every new setting. To achieve efficient real-world learning, we must learn a prior for reasonable behavior to explore using useful actions. Many previous methods rely on in-domain human demonstrations that are manually collected by a human operator or demonstrator [SBP22; Zha+23; Pom88; Man+21]. Due to recent advances in large-scale computer vision, we propose *leveraging human data to learn priors* for dexterous tasks, and improving on such priors in the real world. We aim to use the vast corpus of internet data to define this prior. What is the best way to combine human priors with online practice, especially for hand-based tasks? When manipulating an object, the first thing one thinks about is where on the object to make contact, and how to make this contact. Then, we think about how to move our hands *after the contact*. In fact, this type of prior has been studied in computer vision and robotics literature as *visual affordances* [FWG15; Bah+23; Goy+22; NFG19; Sha+20; Liu+22a; Liu+22b; WGG17]. In our approach, DEFT, we build dexterous grasp affordances which predict the contact point, hand pose at contact, and post-contact trajectory. To improve upon these, we introduce a sampling-based approach similar to the Cross-Entropy Method (CEM) to fine-tune. Specifically, we tune the robot hand grasp, the pose, and the post-grasp trajectory all in the real world for a variety of tasks. This method enables iterative real-world improvement in less than an hour.

In summary, our approach (DEFT) executes real-world learning on a soft robot hand with only a few trials in the real world. To facilitate this efficiently, we train priors on human motion from internet videos. We introduce 9 challenging tasks (as seen in Figure 2.1) that are difficult even for trained operators to perform: Picking a Cup, Pouring Lemonade, Opening a Drawer, Picking a Spoon, Scooping a Grape, Stirring a Spoon, Picking Grapes, Flipping a Bagel, and Squeezing a Lemon. While our method begins to show good success on these tasks with real-world fine-tuning, more investigation is required to complete these tasks more effectively. Please see our website for details and videos at `http://dexterousfinetuning.github.io`.

2.2 Related Work

2.2.1 Real-world robot learning

Real-world manipulation tasks can involve a blend of classical and learning-based methods. Classical approaches like control methods or path planning often use hand-crafted features or objectives and can often lack flexibility in unstructured settings [Kar+11; KL00; MYB16]. On the other hand, data-driven approaches such as deep reinforcement learning (RL) can facilitate complex behaviors in various settings, although these methods frequently rely on lots of data, privileged reward information and struggle with sample efficiency [KP08; PMA10; Lil+16; Pop+17; Pat+17]. Efforts have been made to scale end-to-end RL [Lev+16; Nai+18; Agr+16; Haa+17; Kal+18; Kal+21] to the real world, but their approaches are not yet efficient enough for more complex tasks and action spaces and are reduced to mostly simple tasks even after a lot of real-world learning. Many approaches try to improve this efficiency such as by using different action spaces [Mar+19], goal relabeling [And+17], trajectory guidance [LK13], visual imagined goals [Nai+18], or curiosity-driven exploration [MBP23].

2.2.2 Learning from Human Motion

The field of computer vision has seen much recent success in human body and object interaction with deep neural networks. The human hand is often parametrized with MANO, a 45-dimensional vector [RTB17] of axes aligned with the wrist, and a 10-dimensional shape vector. MANOtorch from [Yan+21] aligns it with the anatomical joints. Many recent works detect MANO in monocular video [Wan+20; Kan+17; RSJ21]. Some also detect objects as well as the hand together [Sha+20; YGT22]. We use FrankMocap to detect the hand for this work.

There are many recent datasets including the CMU Mocap Database and Human3.6M [Ion+13] for human pose estimation, 100 Days of Hands [Sha+20] for hand-object interactions, FreiHand [Zim+19] for hand poses, Something-Something [Goy+17a] for semantic interactions. ActivityNet datasets [FN15], or YouCook [Das+13] are action-driven datasets that focus on dexterous manipulation. We use these three datasets: [Gra+22] is a large-scale dataset with human-object interactions, [Liu+22c] for curated human-object interactions, and [Dam+18] which has many household kitchen tasks. In addition to learning exact human motion, many others focus on learning priors from human motion. [Ma+22a; Nai+22b] learn general priors using contrastive learning on human datasets.

2.2.3 Learning for Dexterous Manipulation

With recent data-driven machine learning methods, roboticists are now beginning to learn dexterous policies from human data as well. Using the motion of a human can be directly used to control robots [Han+20a; SSP22; Mul22]. Moving further, human motion in internet datasets can be retargeted and used directly to pre-train robotic policies [SBP22; MG22]. Additionally, using human motion as a prior for RL can help with learning skills

that are human-like [Raj+17; Pen+18; MG21]. Without using human data as priors, object reorientation using RL has been recently successful in a variety of settings [And+20; Che+22b]. Similar to established work in robot dogs which do not have an easy human analog to learn from, these methods rely on a lot of training data collected in simulation along with zero-shot transfer [Aga+23; Mar+22].

2.2.4 Soft Object Manipulation

Manipulating soft and delicate objects in a robot's environment has been a long-standing problem. Using the torque output on motors, either through measuring current or through torque sensors, is useful feedback to find out how much force a robot is applying [Yos85; AS91]. Coupled with dynamics controllers, these robots can learn to not apply too much torque to the environment around them [LP17; LT17; Kha87]. A variety of touch sensors [Si+23; YDA17; Bhi+21; Sun+19] have also been developed to feel the environment around it and can be used as control feedback.

Soft robotics, like our robot hand, inherently have compliant properties that make them sensitive to the environment [RT15; WTB18]. However, this introduces other difficult design challenges. Soft materials can change properties and are difficult to manufacture. Soft robots, including our robot hand, often do not know the end-effector tip position in a closed-loop manner [But+01; Bau+22].

2.3 Background

2.3.1 DASH: Dexterous Anthropomorphic Soft Hand

Recently introduced, DASH (Dexterous Anthropomorphic Soft Hand) [Man+23] is a four-fingered anthropomorphic soft robotic hand well-suited for machine learning research use. We use the DASH hand on the end-effector of our arm for this work.

The DASH hand's human-like size and form factor allows us to retarget human hand grasps to robot hand grasps easily and perform human-like grasps. Each finger is actuated by 3 motors connected to string-like tendons, which deform the joints closest to the fingertip (DIP joint), the middle joint (PIP joint), and the joint at the base of the finger (MCP joint). There is one motor for the finger to move side-to-side at the MCP joint, one for the finger to move forward at the MCP joint, and one for PIP and DIP joints. The PIP and DIP joints are coupled to one motor and move dependently. While the motors do not know the end-effector positions of the fingers, we learn a mapping function from pairs of motor angles and visually observed open-loop finger joint angles. These models are used to command the finger joint positions learned from human grasps.

2.3.2 Retargeting MANO to Soft Hand

In order to use human hand poses as a prior for the end effector joints, we need to first detect hand poses with a model such as MANO [RTB17], and then develop an effective method to retarget it to soft hand joints.

For MANO parameters, the axis of each of the joints is rotation aligned with the wrist joint and translated across the hand. However, our robot hand operates on forward and side-to-side joint angles. To translate the MANO parameters to the robot fingers, we extract the anatomical consistent axes of MANO using MANOTorch. Once these axes are extracted, each axis rotation represents twisting (not possible for human hands), bending, and spreading. We then match these axes to the robot hand. The spreading of the human hand's fingers (side-to-side motion at the MCP joint) maps to the side-to-side motion at the robot hand's base joint. The forward folding at the base of the human hand (forward motion at the MCP joint) maps to the forward motion at the base of the robot hand's finger. Finally, the bending of the other two finger joints on the human hand, PIP and DIP, map to the robot hand's PIP and DIP joints. While the thumb does not have anatomically the same structure, we map the axes in the same way.

Other approaches [SSP22] rely on creating an energy function to map the human hand to the robot hand. However, because the soft hand is similar in anatomy and size to a human hand, it does not require energy functions for accurate retargeting.

2.4 Fine-Tuning Affordance for Dexterity

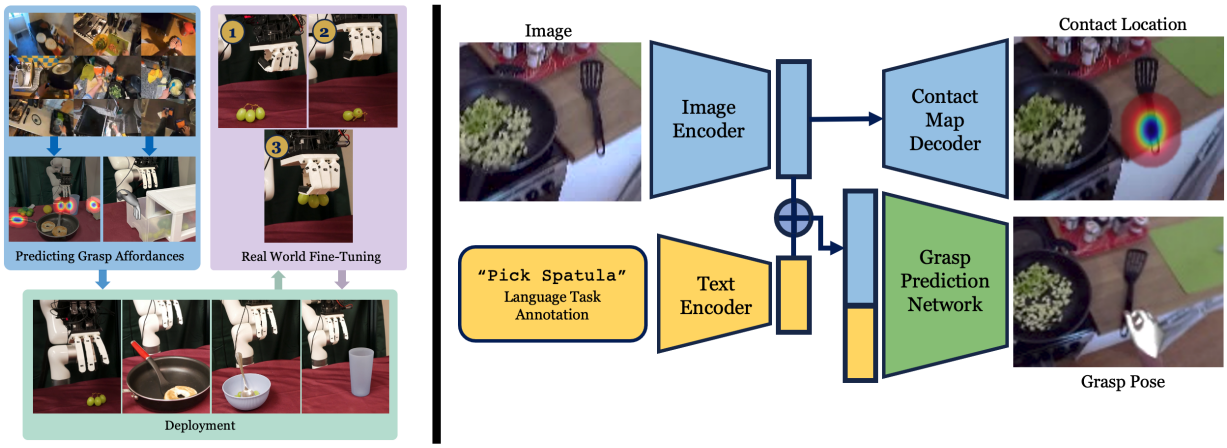

Figure 2.2: **Left:** DEFT consists of two phases: an affordance model that predicts grasp parameters followed by online fine-tuning with CEM. **Right:** Our affordance prediction setup predicts grasp location and pose.

The goal of DEFT is to learn useful, dexterous manipulation in the real world that can generalize to many objects and scenarios. DEFT learns in the real world and fine-tunes robot hand-to-object interaction using only a few samples. However, without any priors on what is useful behavior, the robot will explore the action space inefficiently. Especially with a high-dimensional robotic hand, we need a strong prior to effectively explore the real world. We train an affordance model on human videos to learn what are reasonable behaviors the robot should perform.

2.4.1 Learning grasping affordances

To learn from dexterous interaction in a sample efficient way, we use human hand motion as a prior for robot hand motion. We aim to answer the following: (1) What useful, actionable information can we extract from the human videos? (2) How can human motion be translated to the robot embodiment to guide the robot? In internet videos, humans frequently interact with a wide variety of objects. This data is especially useful in learning object affordances. Furthermore, one of the major obstacles in manipulating objects with few samples is accurately grasping the object. A model that can perform a strong grasp must learn *where* and *how* to grasp. Additionally, the task objective is important in determining object affordances–humans often grasp objects in different ways depending on their goal. Therefore, we extract three pieces of information from human videos: the grasp location, the human grasp pose, and the task.

Given a video clip $V = \{v_1, v_2, \ldots, v_T\}$, the first frame v_t where the hand touches the object is found using a pre-trained, off-the-shelf hand-object detection model [Sha+20]. Similar to previous approaches [Bah+23; Goy+22; Liu+22a; NFG19], a set of contact points are extracted to fit a Gaussian Mixture Model (GMM) with centers $\mu = \{\mu_1, \mu_2, \ldots, \mu_k\}$. Detic [Zho+22] is used to obtain a cropped image v_1' containing just the object in the initial

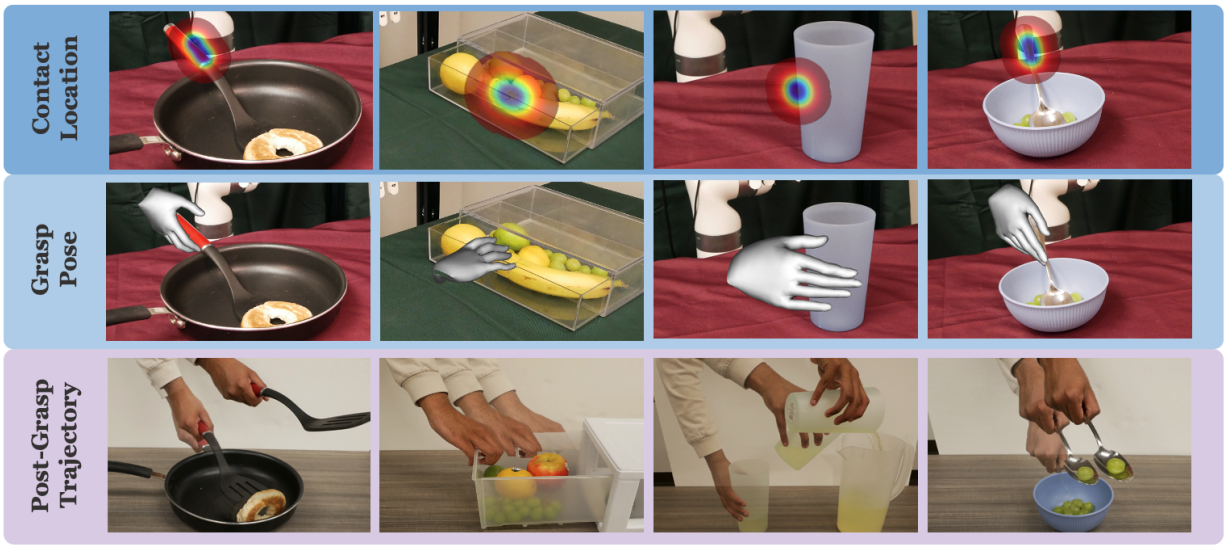

Figure 2.3: We produce three priors from human videos: the contact location (**top row**) and grasp pose (**middle row**) from the affordance prior; the post-grasp trajectory (**bottom row**) from a human demonstration of the task.

frame v_1 to condition the model. We use Frankmocap [RSJ21] to extract the hand grasp pose P in the contact frame v_t as MANO parameters. We also obtain the wrist orientation θ_{wrist} in the camera frame. This guides our prior to output wrist rotations and hand joint angles that produce a stable grasp. Finally, we acquire a text description T describing the action occurring in V.

We extract affordances from three large-scale, egocentric datasets: Ego4D [Gra+22] for its large scale and the variety of different scenarios depicted, HOI4D [Liu+22b] for high-quality human-object interactions, and EPIC Kitchens [Dam+18] for its focus on kitchen tasks similar to our robot's. We learn a task-conditioned affordance model f that produces $(\hat{\mu}, \hat{\theta}_{\text{wrist}}, \hat{P}) = f(v_1', T)$. We predict $\hat{\mu}$ in similar fashion to [Bah+23]. First, we use a pre-trained visual model [Nai+22a] to encode v_1' into a latent vector z_v. Then we pass z_v through a set of deconvolutional layers to get a heatmap over v_1' and use a spatial softmax to estimate $\hat{\mu}$.

To determine $\hat{\theta}_{\text{wrist}}$ and $\hat{P}$, we use z_v and an embedding of the text description $z_T = g(T)$, where g is the CLIP text encoder [Rad+21b]. Because transformers have seen success in encoding various multiple modes of input, we use a transformer encoder $\mathcal{T}$ to predict $\hat{\theta}_{\text{wrist}}, \hat{P} = \mathcal{T}(z_v, z_T)$. Overall, we train our model to optimize

$$\mathcal{L} = \lambda_\mu ||\mu - \hat{\mu}||_2 + \lambda_\theta ||\theta_{\text{wrist}} - \hat{\theta}_{\text{wrist}}||_2 + \lambda_P ||P - \hat{P}||_2 \tag{2.1}$$

At test time, we generate a crop of the object using Segment-Anything [Kir+23] and give our model a task description. The model generates contact points on the object, and we take the average as our contact point. Using a depth camera, we can determine the 3D contact point to navigate to. While the model outputs MANO parameters [RTB17] that are designed to describe human hand joints, we retarget these values to produce similar grasping poses on our robot hand in a similar manner to previous approaches [Han+20b;

SSP22]. In addition to the affordance model f, we collect one demo of the human doing the robot task (Figure 2.3). This demo is used as a prior on the post-grasp trajectory. We extract the task-specific wrist trajectory after the grasp using [RSJ21]. In the fine-tuning stage, we initialize the post-grasp trajectory with this human demonstration. Once we have this prior, how can the robot *improve* upon it?

2.4.2 Fine-tuning via Interaction

Algorithm 1 Fine-Tuning Procedure for DEFT

Require: Task-conditioned affordance model f, task description T, post-grasp trajectory τ, residual policy π. E number of elites, M number of warm-up episodes, N total iterations.

for $k = 1 \dots N$ **do**
  $I_{k,0} \leftarrow$ initial image
  $\xi_k \leftarrow f(I_{k,0}, T)$
  $\epsilon_k = \pi(I_{k,0}, \xi_k)$
  Execute grasp from $\xi_k + \epsilon_k$, then trajectory τ
  Collect reward R_k; reset environment
  if $k > M$ **then**
    Order traj indices $i_1, i_2, \dots, i_k$ based on rewards
    $E \leftarrow \{\epsilon_{i_1}, \epsilon_{i_2}, \dots, \epsilon_{i_E}\}$
    Fit $\pi(.)$ as a VAE to E
  end if
end for

The affordance prior allows the robot to narrow down its learning behavior to a small subset of all possible behaviors. However, these affordances are not perfect and the robot will oftentimes still not complete the task. This is partially due to morphology differences between the human and robot hands, inaccurate detections of the human hands, or differences in the task setup. In order to improve upon this, we practice in an online fashion to optimize the learned skills.

Let the grasp location, wrist rotation, and grasp pose, as well as the trajectory from our affordance prior be ξ. During training we sample noise $\epsilon \sim \mathcal{D}$ where $\mathcal{D}$ is initialized to $\mathcal{N}(0, \sigma^2)$ (for a small σ). We rollout a trajectory parameterized by $\xi + \epsilon$. We collect R_i, the reward for each $\xi_i = f(v_i) + \epsilon_i$ where v_i is the image. After an initial number of M warmup episodes, we rank the rollouts based on R_i and extract sampled noise from the elite trajectories $\{\epsilon_{i_1}, \epsilon_{i_2}, \dots, \epsilon_{i_k}\}$. We fit $\mathcal{D}$ to the elite trajectories to improve the sampled noise.

At test time, we could take the mean values of the top N trajectories for the rollout policy. However, this does not account for the appearance of different objects, previously unseen object configurations, or other properties in the environment. To account for this, we train a VAE [SYL15; RMW14a; RMW14b; KW13] to output residuals δ_j conditioned on an encoding of the initial image $\phi(I_{j,0})$ and affordance model outputs ξ_j from the top ten trajectories. We train an encoder $q(z|\delta_j, c_j)$ where $c_j = (\phi(I_{j,0}), \xi_j)$, as well as a decoder $p(\delta_j|z, c_j)$. At test time, our residual policy $\pi(I_0, \xi)$ samples $z \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and predicts $\hat{\delta} = p(z, (I_0, \xi))$. Then we rollout the trajectory determined by the parameters $\xi + \hat{\delta}$.

2.5 Experimental Setup

Figure 2.4: **Left**: Workspace Setup. We place an Intel RealSense camera above the robot to maintain an egocentric viewpoint, consistent with the affordance model's training data. **Right**: Thirteen objects used in our experiments.

2.5.1 Task Setup

We introduce 9 tabletop tasks: *Pick Cup*, *Pour Cup*, *Open Drawer*, *Pick Spoon*, *Scoop Grape*, *Stir Spoon*, *Pick Grape*, *Flip Bagel*, and *Squeeze Lemon*. For all tasks, we randomize the position of the object on the table, as well as use train and test objects with different shapes and appearances to test for generalization. See Figure 2.4 for a depiction of the robot's workspace and the objects we use in our experiments.

We define the success criteria for each of our 9 tasks as follows:

- Pick Cup: Cup must leave table surface and stay grasped throughout trial.
- Pour Cup: Cup must be grasped throughout trial and also rotate so that the top of the cup is at a lower height than the base.
- Open Drawer: Drawer is initially slightly open so that it can be grasped. By the end of the rollout, the drawer should be at least 1 centimeter more open than it was at the beginning.
- Pick Spoon: The spoon must not be in contact with the table at the end of the trial.
- Stir Spoon: The spoon base must rotate around the jar/pot at least 180 degrees while grasped.
- Scoop Grape: The spoon must have a grape at the end of the trial while being held by the soft hand.
- Pick Grape: All grapes must be held by the hand above the table surface. In particular, if any single grape falls due to a weak stem, this is considered a failure.

- Flip Bagel: The side of the bagel that is facing up at the end of the trial should be opposite the side facing up at the beginning.
- Squeeze Lemon: The lemon should be grasped securely on top of the juicer.

2.5.2 Online Fine-Tuning Setup

To achieve real-world learning with the soft robot hand, we pretrain an internet affordance model as a prior for robot behavior. As explained in Section 2.4, we train one language-conditioned model on all data. At test time, we use this as initialization for our real-world fine-tuning. The fine-tuning is done purely in the real world. An operator runs 10 warmup episodes of CEM, followed by 20 episodes that continually update the noise distribution, improving the policy. We train a residual VAE policy that trains on the top ten CEM rollouts to predict the noise given the image and affordance outputs. Across all our experiments, we collect data for several thousands of rollouts for over **100 hours** of real world data collection.

2.5.3 Datasets and Affordance Model Parameters

We use data from Ego4D [Gra+22], EpicKitchens-100 [Dam+18], and HOI4D [Liu+22b]. After filtering for clips of sufficient length, clips that involve grasping objects with the right hand, and clips that have language annotations, we used 64666 clips from Ego4D, 9144 clips from EpicKitchens, and 2707 clips from HOI4D. In total, we use a dataset of 76517 samples for training our model.

For our contact location model, we use the visual encoder from [Nai+22a] to encode the image as a 512-dimensional vector. We use the spatial features of the encoder to upsample the latent before applying a spatial softmax to return the contact heatmap. This consists of three deconvolutional layers with 512, 256, and 64 channels in that order.

To predict wrist rotation and grasp pose, we use the language encoder from [Rad+21b] to compress the language instruction to a 512-dimensional vector. We concatenate the visual and language latents and pass it through a transformer with eight heads and six self-attention layers. We pass the result of the transformer through an MLP with hidden size 576 and predict a vector of size 48: the first 3 dimensions are the axis-angle rotations; the last 45 dimensions are the joint angles of the hand. These correspond to the parameters output by Frankmocap [RSJ21], which we used to get ground truth hand pose.

We jointly optimize the L2 loss of the contact location μ, the wrist rotation θ_{wrist} and grasp pose P. The weights we used for the losses are $\lambda_\mu = 1.0, \lambda_\theta = 0.1, \lambda_P = 0.1$. We train for 70 epochs with an initial learning rate of 0.0002 and a batch size of 224. We used the Adam optimizer [KB14] with a cosine learning rate scheduler. We trained on a single NVIDIA RTX A6000 with 48GB RAM.

2.5.4 Hardware Setup

We use a 6-DOF UFactory xArm6 robot arm for all our experiments. We attach it to a 16-DOF Soft Hand using a custom, 3D-printed base. We use a single, egocentric RGBD

camera in order to capture the 3D location of the object in the camera frame. We calibrate the camera so that the predictions of the affordance model can be converted to and executed in the robot frame. The flexibility of the robot hand also makes it robust to collisions with objects or unexpected contact with the environment.

2.5.5 Safety

In our fine-tuning experiments, there is a particular focus on the safety of the robot system and the environment. The soft hand allows the policy to perform high-contact manipulation tasks without breaking because of its compliance. Our method takes advantage of its compliance as it performs thousands of iterations in the real world.

While the end-effector is soft, the arm is not. Because it is susceptible to damage when colliding with the environment, we constrain the arm's velocity and ensure that the arm stays above the tabletop. The rollout will be terminated if the arm's dynamics controller senses that the arm collided aggressively with the environment.

2.6 Results

We perform a variety of experiments to answer the following questions: 1) How good is our affordance model? 2) How well can DEFT learn and improve in the real world? 3) How can the experience collected by DEFT be distilled into a policy? 4) How can DEFT be used for complex, soft object manipulation? We investigate the role of the affordance model and real-world fine-tuning in Table 2.1 and Figure 2.5. Then we perform a series of ablations in Table 2.2.

Method	Pick cup		Pour cup		Open drawer		Pick spoon		Scoop Grape		Stir Spoon	
	train	test	train	test	train	test	train	test	train	test	train	test
`Real-World Only`	0.0	0.1	0.2	0.1	0.1	0.0	0.7	0.3	0.0	0.0	0.3	0.0
`Affordance Model Only`	0.1		0.4		**0.5**		0.5		0.0		0.3	
`DEFT`	**0.8**	**0.8**	**0.8**	**0.9**	**0.5**	**0.4**	**0.8**	**0.6**	**0.7**	**0.3**	**0.8**	**0.5**

Table 2.1: We present the results of our method as well as compare them to other baselines: Real-world learning without internet priors used as guidance as well as the affordance model outputs without real-world learning. Together, our method is able to better complete these tasks.

2.6.1 Effect of affordance prior

The human affordance model predicted three items: the contact location, the wrist grasp rotation, and the hand joint pose. In the Real-World Only model, we use a few heuristics in place of each item in the affordance prior and proceed with fine-tuning. For contact location, we detect the object in the scene using a popular object detection model [Kir+23] and let the contact location prior be the center of the bounding box. For the wrist rotation, use a generic rotation with the palm of the hand facing downward. Finally, for the hand joint angles, we fix a half-closed hand as the grasp pose prior.

With these heuristics, the robot has difficulty finding stable grasps consistently across a range of tasks. While the robot is able to navigate to the object, the main obstacle was finding the correct rotation angle for the hand. Hand rotation is very important for many tool manipulation tasks because it requires not only picking the tool but also grasping in a stable manner. While the soft hand shows decent success in grasping a spoon (where the grasp rotation from the heuristic is close to the correct grasp rotation) it is not able to perform well in tasks that require other wrist rotations.

It is possible that with more exploration the Real-World Only model would be able to catch up with DEFT. However, we believe that these results indicate that the human affordance model reduces the number of fine-tuning iterations necessary in the real world.

2.6.2 Zero-shot model execution

We explore the zero-shot performance of our prior with the Affordance Only model. Without applying any online fine-tuning to our affordance model, we rollout the trajectory

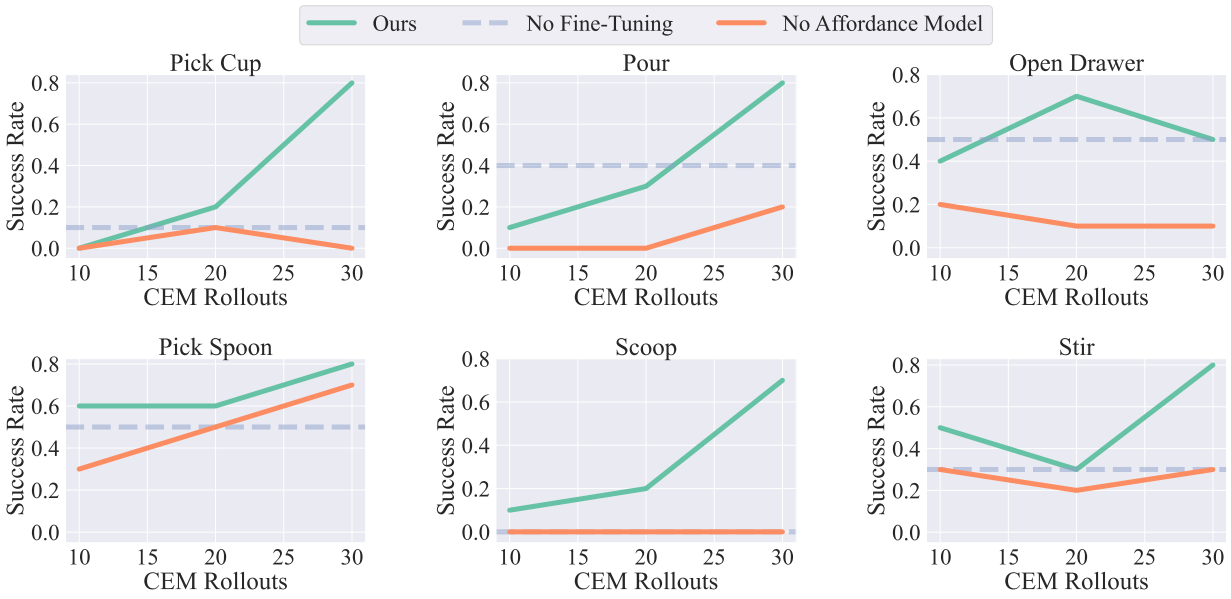

Figure 2.5: Improvement results for 6 tasks: pick cup, pour, open drawer, pick spoon, scoop, and stir. We see a steady improvement in our method as more CEM episodes are collected.

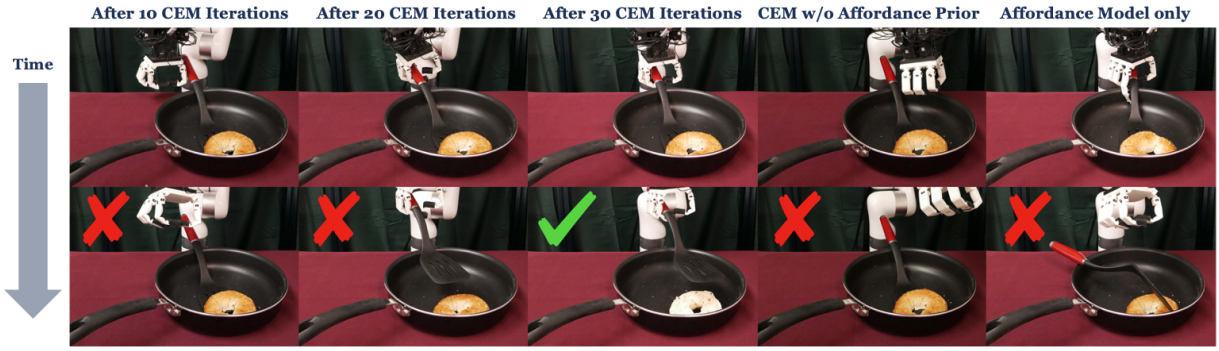

Figure 2.6: Qualitative results showing the finetuning procedure for DEFT. The model learns to hold the spatula and flip the bagel after 30 CEM iterations.

parameterized by the prior. While our model performs decent on simpler tasks, the model struggles on tasks like stir and scoop that require strong, power grasps (shown in Table 2.1). In these tasks, the spoon collides with other objects, so fine-tuning the prior to hold the back of the spoon is important in maintaining a reliable grip throughout the post-grasp motion. Because DEFT incorporates real-world experience with the prior, it is able to sample contact locations and grasp rotations that can better execute the task.

One observation we found is that our zero-shot model sometimes performs better than the residual model trained on the first ten CEM iterations. This is due to DEFT optimizing the grasp parameters only after ten iterations. Our first residual model learns only from random noise, explaining why our zero-shot performance can be stronger than DEFT after a few iterations of fine-tuning.

Method	Pour Cup		Open Drawer		Pick Spoon	
	train	test	train	test	train	test
Reward Function:						
`R3M Reward`	0.0	0.0	0.4	**0.5**	0.5	0.4
`Resnet18 Imagenet Reward`	0.1	0.2	0.3	0.1	0.4	0.2
Policy Ablation:						
`DEFT w/ MLP`	0.0	0.0	0.5	0.0	0.6	0.5
`DEFT w/ Transformer`	0.4	0.5	**0.6**	0.1	0.4	0.5
`DEFT w/ Direct Parameter est.`	0.1	0.1	0.1	0.0	0.3	0.0
`DEFT`	**0.8**	**0.9**	0.5	0.4	**0.8**	**0.6**

Table 2.2: Ablations for (1) reward function type, (2) model architecture, and (3) parameter estimation approach.

2.6.3 Human and automated rewards

Our method queries the operator during the task reset process to assign a continuous score from 0 to 1 for the grasp. Because the reset process requires a human-in-the-loop regardless, this adds little marginal cost for the operator. But what if we would like these rewards to be calculated autonomously?

We use the final image collected in the single post-grasp human demonstration from Section 2.4 as the goal image. We define the reward to be the negative embedding distance between the final image of the rollout and the goal image with either an R3M [Nai+22a] or a ResNet [He+15] encoder. The model learned from ranking trajectories with R3M reward is competitive with DEFT in two of the three tasks we tested on, and performed better than the model that used Resnet18 rewards. Using a third-person camera could potentially improve the rankings because changes in the environment will be more apparent. These results indicate that using a visual reward model can potentially provide reasonable results compared to human rewards.

2.6.4 Model Architecture

We investigate different models and training architectures for the policy trained on the rollouts (Table 2.2). When we replace the conditional VAE with an MLP that predicts residuals, the model has difficulty learning the grasp rotation to effectively pour a cup. This may be because VAEs can compress multi-modal data more effectively (which is useful for our case as our inputs includes location, rotation, and joint angles).

Our transformer ablation is an offline method similar to [Che+21] where in addition to the image and affordance model outputs, we condition on the reward outputs and train a transformer to predict the residual. At test time the maximum reward is queried and the output is used in the rollout. We hypothesize that the reduced performance is because the transformer is a data-hungry architecture. The model may need more real-world data, which can be expensive to collect.

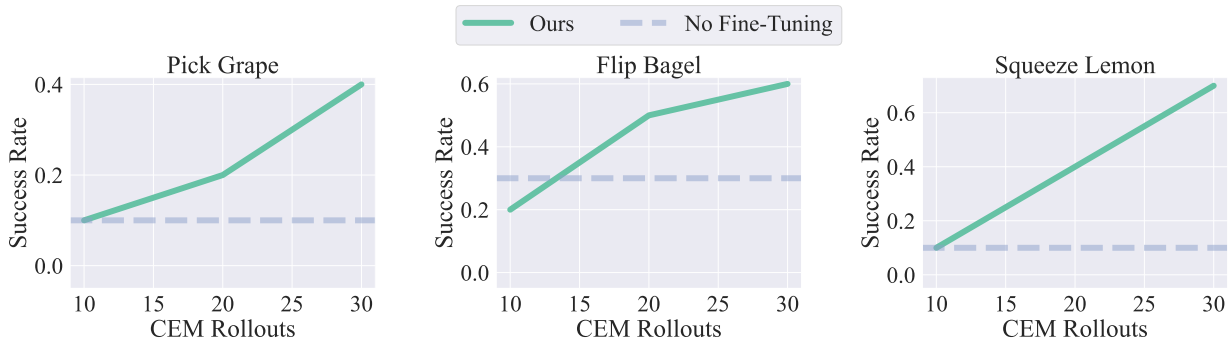

Figure 2.7: We evaluate DEFT on three additional difficult manipulation tasks.

Finally, we train a VAE to directly estimate ξ instead of the residual. This model was unable to effectively distill the information from the affordance prior with neither the diversity of data nor training time allotted. As a result, it often makes predictions that are far from the correct grasp pose.

2.6.5 Performance on complex tasks and soft objects

We investigate the performance of DEFT on more challenging tasks, which involve grasping or manipulating food. Tasks involving soft objects cannot be simulated accurately, so sim2real methods may have difficulty performing these tasks in the real world. Our method does not have that challenge in this setting and DEFT is able to perform reasonably well on these tasks.

Of the three tasks, our method has the most difficulty with the Pick Grape task. This is because grapes are small and require the fingers to curl fully to maintain a stable grasp. A limitation of our hand is that the range of MCP joint does not allow the fingertips to touch its palm, and as a result, it has difficulty in consistently picking small objects.

2.7 Discussion and Limitations

In this thesis, we investigate how to learn dexterous manipulation in complex setups. DEFT aims to learn directly in the real world. In order to efficiently perform real world fine-tuning, we build an *affordance* prior learned from human videos. We are able to practice and improve in the real world via our online fine-tuning approach, enabled by the use of a soft anthropomorphic hand, performing a variety of tasks (involving both rigid and soft objects).

However, there are some limitations to our work. While we are able to learn policies for the high-dimensional robot hand, the grasps learned are not multi-modal and do not capture all of the different grasps humans are able to perform. In particular, we find that the predicted hand poses are all power grasps, which is used even in situations where other grasps might be more appropriate (such as a pinch grasp when picking a grape). We believe that this is mainly caused by noisy hand detections. As these detection models improve, we hope to be able to learn a more diverse set of hand grasps.

Second, during finetuning, resets require human input and intervention. This limits the amount of real-world learning we can do, as the human has to be constantly in the loop to reset the objects. Other works [Che+22a; Gup+21] introduce paradigms that might potentially be useful in helping scale our method for more fine-tuning iterations.

Third, the arm has additional physical limitations. While the DASH hand is soft like the human hand, the robot arm is rigid. As a result, the robot cannot mimic *every* grasp humans make. For example, any underhand grasp that involves sliding the hand underneath an object is not possible with this setup because the arm would collide with the table. A soft arm would enable a wider range of human-like grasps.

Finally, the soft hand's fingers do not curl fully. The soft hand's fingers have a tradeoff between strength and range of motion. The version of the soft hand used for the experiments in this project has high strength, which is useful to pick large, heavier objects. However, this makes grasping smaller objects, such as individual grapes, more difficult. A hand that can have the best of both worlds would facilitate a larger variety of tasks and is a potential area for future research.

Chapter 3

Zero-Shot Rewards from Human Videos

3.1 Introduction

Despite recent advances in robot learning, there are many challenges in advancing towards the long-standing goal of developing generalist robotic agents. To achieve this objective, one must develop a framework that can generalize across diverse environments, tasks, and robot embodiments. Perhaps the most concise way to represent these challenges is the problem of learning a goal-conditioned reward function. Critically, we want to learn and deploy a reward function in a way that is amenable to different tasks and environments.

Traditionally, designing a robotic agent to perform tasks in the real world using reinforcement learning has proved difficult due to the sheer amount of data that needs to be collected in the process of exploring the environment. As a result, much of the advancement in reinforcement learning has come from model-free reinforcement learning in simulated environments. Whereas NLP and vision research has shown greater generalization when scaling to large datasets [Bro+20], the collection and use of large, task-specific data in robot learning has encountered more challenges due to the physical cost of robot deployment. Collecting robot data at scale must provide guarantees for safety, and it requires human supervision for task resets. Many robots are trained in a single environment on a small number of tasks.

Rather than attempting to collect robot interaction data, a growing body of work has focused on exploiting the abundance of internet data, particularly egocentric video, inspired by the success of CLIP [Rad+21a]. The vast quantity of human manipulation data available online could be a useful prior for robot manipulation. In particular, the diversity of the environments and tasks depicted in internet-scale data could improve generalization when learning a goal-specific reward function. With this in mind, many recent works have collected labeled, in-the-wild, egocentric videos of humans manipulating objects and performing various tasks [Gra+22; Goy+17b; Dam+20].

However, there are two immediate challenges in utilizing offline human videos for learning a reward function. Firstly, there is a difference in embodiments when training with human data compared to test time where trajectories are generated in the robot's action space. Transferring rewards across this embodiment gap is necessary to learn a robot pol-

icy. Secondly, there is a difference in the visual appearance of the agents in the human and robot trajectories.

What is the best way to learn a reward representation in the absence of robot data? Our approach to learn a reward function is to mask and inpaint over the agent in a visual scene to provide agent-agnostic data alignment. In visually removing the agent from the scene, we hope to learn a reward representation that focuses on relevant, environment-centric features of the scene that are modified over the course of the trajectory. With the diversity of environments in datasets like Something-Something [Goy+17b] and Ego4D [Gra+22], we believe that we can learn generalizable reward functions across different robotic tasks in simulation environments. Additionally, with this approach, we wish to use only human demonstrations in specifying a goal. In particular, due to our focus on building an agent-agnostic reward function, we investigate only the zero-shot paradigm: no robot demonstrations are used in learning a policy.

We demonstrate that our method shows promise on several tasks in a simulated environment with a Sawyer arm. More investigation is required to scale this method to a diverse range of tasks.

3.2 Related Work

3.2.1 Reward Learning

While many works have used Reinforcement Learning (RL) with great success [Sil+17; Vin+19; Kum+21; Aga+23], RL assumes access to a reward function which may not always be easy to define or optimize for. Inverse RL is the field for defining reward functions from demonstrations and has many prior work [Zie+08; RBZ06; WOP15; Lev18]. Many recent works have aimed to generalize to scenarios where humans provide desired outcomes [Fu+18; Sin+19]. However, these works aim to learn rewards in single-task settings. Our work concentrates on the multi-task setting, where we hope to learn a reward function that generalizes to many tasks. In particular, we wish to learn a function that can generate rewards for many tasks given one human demonstration per task.

3.2.2 Pre-training Representations for Control

A natural approach to address multi-task generalization is pre-training on large datasets. Indeed, many works leverage large-scale data to pre-train representations for control [Par+22; Cui+22; Nai+22a; Ma+22b; SK21; Rad+22]. To create generalized image representations, some have trained on a wide variety of images to take advantage of the natural supervision that's abundantly available on the internet [Rad+21a]. Many works have adopted these universal embeddings for downstream robotics tasks [SMF22; Jan+21]. Others have combined language embeddings with human videos to learn representations that can be used for robot manipulation tasks [Sha+21]. Similarly, [Nai+22a] applies time contrastive learning [Ser+18], video-language alignment [Nai+21], and applies a sparsity penalty to learn a robust representation from Ego4D [Gra+22]. This representation can in turn be utilized as a distance function for rewards.

Self-supervised representation learning from unlabeled videos encodes semantic and geometric understanding of diverse actions to define reward functions for reinforcement learning. Self-supervised representation learning from images has seen significant progress in recent years [Pat+16; Che+20; Car+21; He+22; Gri+20], approaching the performance of fully supervised methods on ImageNet [Rus+14]. Given the success of representation learning in the image domain, recent works adopt similar approaches for learning from video for downstream robotics applications [SLA20]. [Rad+22] trains a single masked autoencoder [He+22] on diverse, in-the-wild videos. This unified vision policy is then adapted for downstream control, significantly outperforming CLIP. [Ser+18] proposes a time-contrastive approach learning approach from multi-view video, where different views of a robot action at the same timestamp are close in feature space, while temporally offset views should be repelled in feature space. [Sch+20] directly incorporates videos collected by humans by adding videos to the replay buffer and directly performing RL on the observational data. Combining offline data with in-domain data boosts performance in the environment. [Zak+21] addresses the generalization across agents by training representations using a cycle-consistency loss by learning from other agents demonstrating the same task and is robust to differences in shape, action, end-effector, and dynamics.

3.2.3 Robot Learning from Human Videos

Because collecting robot demonstrations can be costly, many works have proposed using human data, which is both plentiful and easily accessible on the Internet. In recent years, many large-scale annotated human datasets have been collected, such as Something-Something [Goy+17a], Ego4D [Gra+22], EpicKitchens [Dam+20], YouCook [Das+13], and ActivityNet [FN15]. Instead of learning an intermediate representation, some works directly learn from unstructured human videos [BGP22; SBP22; Bah+23].

Other works train representations [Nai+22a; Rad+22; Ma+22b; Rad+21a] that use human data. For example, VIP [Ma+22b] casts representation learning from human videos as an offline goal-conditioned reinforcement learning problem to create a temporally smooth embedding for novel tasks. [Tia+21] learns a visual dynamics model as well as a dynamical distance function to be used for downstream tasks. However, these approaches require a goal image with the robot at test time.

There are other methods that do not require in-domain goals by learning a classifier [Sha+21; CNF21]. DVD [CNF21] trains a discriminator to determine if two videos are performing the same task by learning from in-the-wild videos of humans. The learned feature representation can be used as a reward function for downstream robot control. However, DVD still requires in-domain robot rollouts in training to learn reward functions for downstream tasks. In our work, we build upon DVD and aim to do away with robot demonstrations at both training and test time.

3.3 Zero-Shot Rewards from Human Videos

We aim to learn a multi-task reward function by learning from human demonstrations. For each task $\mathcal{T}_i$, the reward function $\mathcal{R}$ will be conditioned on one of K human demonstrations $\mathcal{D}_i = \{d_{i,j}\}_{j=1}^{K}$ specifying the desired task.

Then, given the robot environment's state space $\mathcal{S}^r$, action space of the robot $\mathcal{A}^r$, and robot dynamics $f^r(s'|s,a)$, we want to solve the MDP parameterized by $(\mathcal{S}^r, \mathcal{A}^r, f^r, \mathcal{R})$ for a task $\mathcal{T}_i$.

Figure 3.1: Training Overview: Human data is processed as follows. We generate masks for human arms and hands in videos from the Something-Something dataset [Goy+17b]. Videos that do not have high-quality masks are discarded. The remaining videos are inpainted based on their masks. We take this set of inpainted human videos and learn a discriminator to capture the functional features of the video.

3.3.1 Learning Agent-Agnostic Representations

Our work addresses the embodiment gap by visually removing the agent from the scene at both training and test time. There are a few potential inpainting methods we can use when given masks to remove the agent (human or robot) from a video. To generate the masks for the robot arm, we used Detectron2 [Wu+19], a state-of-the-art segmentation method, and fine-tuned it for detecting our robot. However, we found Detectron2 to be unreliable in detecting human hands and arms in egocentric data. We believe this is because Detectron2 was trained on third-person views of humans where most of the body is in the scene. For masking human hands and arms in human videos, we use EgoHOS [Zha+22], a method trained on egocentric videos from datasets like Ego4D [Gra+22] and Epic-Kitchens [Dam+20].

While this method provides more consistent masks, there are still videos that have few or noisy masks due to occlusion or difficult lighting conditions. We refine our dataset by

removing videos where fewer than half of the frames have human masks. This removes only one-fifth of our dataset, so we still keep a large majority of our data for training.

For inpainting, we considered two types of methods. We can utilize a video inpainting method such as Copy and Paste Networks [Lee+19] or E2FGVI [Li+22]. Another option for inpainting is to inpaint frame-by-frame, such as with Stable Diffusion [Rom+21]. We found that video inpainting methods perform better as they have access to context for occlusions that singular images do not have. Stable Diffusion also requires many denoising iterations that reduce its speed.

3.3.2 Addressing Visual Domain Gap

While training on a large dataset with a variety of objects, lighting, and poses will provide some generalization, we conjecture that it might not be sufficient to generalize to simulation environments, which look significantly different from the real world. As a result, we account for the visual differences between simulation and real-world settings. In particular, we use Stable Diffusion [Rom+21], a pre-trained image-text generative model, to create synthetic videos by augmenting existing videos from the dataset.

To augment a video, we take individual frames from the video and generate a novel set of frames. We guide this image-to-image translation using text conditioning. That is, given a text prompt T and video V, for each frame $v_1, v_2, \ldots, v_n \in V$ we use Stable Diffusion as a function g to map it to new frame $v_i' = g(v_i, T)$. In order to promote generalization to simulation environments, we use prompts T such as `"OpenGL style rendering"`, and `"Unity style rendering"` to to encourage the output frames to match the desired style.

Because this approach generates a video by transforming individual frames, this does not necessarily maintain the temporal consistency of the video. In order to keep the videos coherent, we restrict the noise added to individual frames to be sampled within a reasonable range for each video.

3.3.3 Reward Training

The reward function we want to learn is $\mathcal{R}(s_{1:T}, \mathcal{D}_i)$. Here, for each task $\mathcal{T}_i$, we have a set of human demonstrations $\mathcal{D}_i$.

We will use a pretrained inpainting method ϕ^h for human inpainting and ϕ^r for robot inpainting. During training, we can apply ϕ^h to each demonstration and train the DVD encoder f_{enc} and similarity function f_{sim} as described in their paper [CNF21].

To summarize the training method in DVD, we train $\mathcal{R}$ as follows. We use a pretrained encoder f_{enc} and a learned similarity function f_{sim} to generate the pairwise reward:

$$\mathcal{R}(d_i, d_j) = f_{\text{sim}}(f_{\text{enc}}(\phi^h(d_i)), f_{\text{enc}}(\phi^h(d_j))).$$

When learning $\mathcal{R}$ at training time, note that f_{enc} and ϕ^h are pretrained and frozen. To learn f_{sim}, we sample videos $d_i^1, d_i^2, d_j \in \mathcal{D}$. We minimize the cross-entropy loss below:

$$\mathcal{L} = \mathbb{E}_{(d_i^1, d_i^2, d_j) \sim \mathcal{D}}[\log(\mathcal{R}(d_i^1, d_i^2)) + \log(1 - \mathcal{R}(d_i^1, d_j)]$$

Note that this is essentially contrastive loss over demonstrations from different tasks. Above, d_i^1 is the anchor, d_i^2 is a positive sample and d_j is the negative sample.

3.3.4 Task Execution

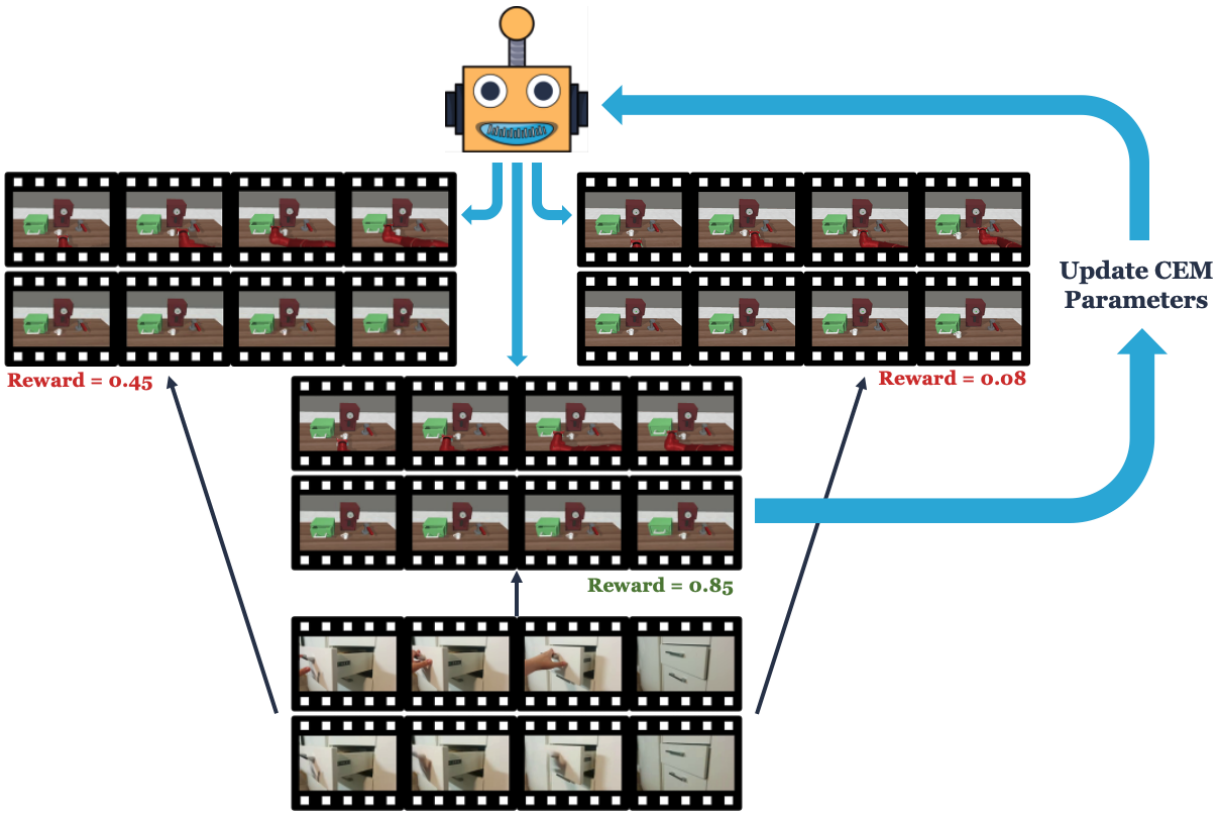

Figure 3.2: At test time, we take as input a set of human demonstrations of the same task. Using trajectories sampled from CEM, we use our reward function to judge the similarity between every (demo, trajectory) pair. We use CEM to optimize this reward function to generate trajectories that are most functionally similar to the demos.

At test time, we are given a set of human demonstrations $\mathcal{D}_i = \{d_{i,j}\}_{j=1}^{K}$ for a specific task. We have a reward function $\mathcal{R}$ that can take in a robot trajectory $s_{1:T}$ and a single human demonstration $d_{i,j}$. We calculate the reward with the corresponding inpainting function, as follows:

$$\mathcal{R}(s_{1:T}, d_{i,j}) = f_{\text{sim}}(f_{\text{enc}}(\phi^r(s_{1:T})), f_{\text{enc}}(\phi^h(d_{i,j})))$$

We can optimize for this reward function using zeroth order methods like Cross-Entropy Method (CEM) or Model Predictive Path Integral (MPPI) control. For our method, we implement CEM. CEM keeps track of a trajectory distribution parameterized by a Gaussian for each action $a_i \sim \mathcal{N}(\mu_i, \sigma_i)$ for $i = 1 \ldots T$. CEM then takes multiple samples from this distribution and retains the samples (elites) with the largest rewards to update the distribution for the next iteration. Then from this distribution, CEM rolls out a trajectory to act out an episode.

For the results in Table 3.1, we use the ground truth dynamics from the simulator when rolling out samples taken from the action distribution. We can also potentially use

a state or visual dynamics function such as [Haf+19] learned in the environment to run CEM efficiently in a closed-loop fashion.

3.3.5 Datasets and Environments

We generate results in Metaworld [Yu+19], with a tabletop environment that has a mug, drawer, and faucet. We train our models using the Something-Something dataset [Goy+17b], an egocentric dataset of humans manipulating different objects according to various tasks. We primarily evaluate our method on three tasks: *Close Drawer*, *Push Mug Away*, and *Move Mug Right*. We train on a dataset of around 10K videos which are labeled with task information. This dataset consists of data for 6 tasks, of which three are the evaluation tasks specified above.

3.4 Experiments

Our experiments aim to answer the following: (1) How effectively does our method perform without any prior knowledge of robot embodiment or robot demonstrations? (2) How does our method compare to previous work? (3) Does domain augmentation improve simulation performance? (4) Does our method improve when trained on additional tasks?

Figure 3.3: Qualitative Outputs: We show the intermediate results of (1) removing the agent and (2) modifying the background with pre-trained text-to-image models.

3.4.1 Comparison to Prior Work

We use two prior works as baselines: DVD [CNF21] and VIP [Ma+22b]. DVD's results involve training on both human and robot videos, so we hypothesize that training only on human videos is not sufficient for learning policies for robots at test time. The DVD reward function does not generalize to unseen robot embodiments. Methods like VIP that pretrain on diverse human datasets generate a reward that represents a distance function to a goal image in the robot's environment. However, we only produce human demonstrations at test time, so we hypothesize this approach will not work well either.

Our results for these baseline methods are in Table 3.1. The methods we show results for are as follows:

- `Shaped Reward`: A human-specified reward function that is shaped for each task.
- `DVD-HR`: The original implementation of DVD, trained on both human and in-domain robot demonstrations.

Model	Close Drawer	Push Mug Away	Move Mug Right	Average
Shaped Reward	0.99	0.95	0.99	0.98
DVD-HR [CNF21]	0.75	0.59	0.06	0.47
Human-only Reward Functions:				
DVD-H [CNF21]	0.0	0.62	0.0	0.21
VIP [Ma+22b]	0.09	0.22	0.14	0.15
Ours	0.65	0.96	0.75	0.79
Ours + Domain Augmentation	0.84	0.96	0.88	0.89

Table 3.1: Fraction of successful iterations for each model with varying data used at training and test time. Each model was averaged over 10 different seeds, each seed being run for 100 CEM iterations.

- `DVD-H`: A modified version of DVD that is limited to training on only human data.
- `VIP`: Using VIP with a goal image from an OOD human video at test time.

In the first two rows, privileged information is available for the reward function (either a manually shaped reward or robot demonstrations are available). When DVD does not have access to in-domain robot data, there is a significant decline in performance in the zero-shot setting. The DVD model struggles to learn a reward function that can transfer to the robot's environment without robot demonstrations. In practice, we see that it only performs one task, even when given demos of other tasks. VIP also does not perform well because it requires a goal image in the robot's setting.

The results of our method are also depicted in Table 3.1. We present our method with and without visual domain augmentation. Both reward functions perform strongly on all three tasks despite not having seen the environment in the training data or at test time. Our method performs significantly better than DVD-H and VIP in the zero-shot setting. Our method even outperforms DVD-HR on two of the three tasks. Overall, we believe that removing the agent encourages the representations learned by our method to more accurately match the functional meaning of the task by focusing on the changes in the environment. Accordingly, our method can perform better than models like DVD-HR.

3.4.2 Effect of visual domain augmentation

As shown in Table 3.1, domain augmentation with Stable Diffusion improves the results of our method on two of the three tasks. Overall, it has an 89% success rate while our method without augmentation has a success rate of 79%. With domain augmentation, our method beats DVD-HR in all three tasks and is moving closer to the performance of the shaped reward. Our method with domain augmentation performs the zero-shot baselines by 68%. Even though Stable Diffusion is not trained on robotics-related data, it is able to provide meaningful priors to enhance training and potentially improve generalization.

More aggressive augmentation could potentially improve the amount of generalization, but could also reduce the temporal consistency of the video as a whole. For sample augmentation outputs, see Figure 3.3.

Figure 3.4: We investigate how adding unrelated tasks in training affects the downstream performance of our method.

Task	Our Method (6 Task Training)	Our Method (20 Task Training)
Close Drawer	0.65	0.41
Push Mug Away	0.96	0.39
Move Mug Right	0.75	0.66
Move Mug Left	0.65	0.31
Pull Mug to Camera	0.01	0.06
Open Drawer	0.25	0.0
Average	0.55	0.31

Table 3.2: Fraction of successful iterations for reward models trained with a varying number of tasks.

3.4.3 Number of Training Tasks

The results that we described in the previous section involve training on 6 tasks. One question we wish to answer is how does the number of tasks we train on affect the performance of the reward function on robotic tasks? In other words, how does the diversity of the training data affect the quality of the reward function learned, and do unrelated videos enhance the reward function learned?

To investigate this, we additionally train a reward function with our method with 3 tasks, 13 tasks, and 20 tasks. All of these variations include the three evaluation tasks in the training set. We evaluate the 4 reward functions with the three original evaluation tasks. We show these results in Figure 3.4. We also show results comparing our method with 6 task training and 20 task training on 3 additional evaluation tasks in Table 3.2.

Based on the results in Figure 3.4, we find that scaling to more tasks in training produces better performance and generalization up to a point. The 3-task reward function performs worse than the 6-task reward function is that there are fewer total videos for training. While both models are trained for the same number of epochs, the number of videos in each epoch is smaller for the model trained with three tasks.

However, after 6 tasks, the success rate plateaus and decreases when going to 13 and then 20 training tasks. This is additionally reflected in Table 3.2 where the 20-task model performs worse on all 6 evaluation tasks. This is likely due to the structure of the reward model: as the discriminator needs to be able to distinguish between more tasks, it might struggle to disambiguate the salient features for the evaluation tasks compared to the other tasks it is trained on. As a result, our method loses robustness when training on *too many* unrelated tasks.

3.5 Conclusion and Limitations

We presented a method building on DVD [CNF21] to create a reward function that is both domain-agnostic and agent-agnostic. Masking and removing the agent is an effective method to force a reward function to learn representations for tasks that involve the behavior of manipulated objects in the scene. We found that using data augmented by pretrained text-image models can further improve the robustness of the reward function at test time. Overall, our method shows promise in learning a reward function that does not rely on prior robot knowledge.

Our method also has several limitations. The core limitation of our approach is that as we train on more tasks, our model has worse performance in evaluation. Ideally, when we train on more tasks, the performance should improve slightly if not stay the same. Currently, we treat training a reward function as a binary classification problem, and this can be improved to account for similarities between similar tasks. Injecting this semantic knowledge via language embeddings would hopefully lead to a smoother representation space that might be more effective at improving with more tasks.

Another limitation of our reward is that it is sparse. While reward functions that use the full video have additional context, it lacks the structure to evaluate partial trajectories and is sparse, unlike image-based reward functions. A reward function that evaluates partial trajectories–perhaps subvideos segmented by *subgoals*–might be able to combine the best of both worlds.

Our data augmentation setup also has room for improvement. To maintain the coherency of the synthetic data generated by the model, we restricted the extent to which we augmented the frames in the video. Perhaps with more text-video generation models being open-sourced as of late, we can modify the style more aggressively while also maintaining consistency across frames. This would potentially allow for better generalization.

Bibliography

This bibliography contains 127 references.

[Aga+23] Ananye Agarwal, Ashish Kumar, Jitendra Malik, and Deepak Pathak. "Legged locomotion in challenging terrains using egocentric vision". In: *Conference on Robot Learning*. PMLR. 2023, pp. 403–415.

[Agr+16] Pulkit Agrawal, Ashvin Nair, Pieter Abbeel, Jitendra Malik, and Sergey Levine. "Learning to poke by poking: Experiential learning of intuitive physics". In: *NIPS* (2016).

[And+17] Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, Pieter Abbeel, and Wojciech Zaremba. "Hindsight Experience Replay". In: *NeurIPS* (2017).

[And+20] OpenAI: Marcin Andrychowicz, Bowen Baker, Maciek Chociej, Rafal Jozefowicz, Bob McGrew, Jakub Pachocki, Arthur Petron, Matthias Plappert, Glenn Powell, Alex Ray, et al. "Learning dexterous in-hand manipulation". In: *IJRR* (2020).

[AS91] Haruhiko Asada and J-JE Slotine. *Robot analysis and control*. John Wiley & Sons, 1991.

[Bah+23] Shikhar Bahl, Russell Mendonca, Lili Chen, Unnat Jain, and Deepak Pathak. "Affordances from Human Videos as a Versatile Representation for Robotics". In: 2023.

[Bau+22] Dominik Bauer, Cornelia Bauer, Arjun Lakshmipathy, Roberto Shu, and Nancy S Pollard. "Towards Very Low-Cost Iterative Prototyping for Fully Printable Dexterous Soft Robotic Hands". In: *2022 IEEE 5th International Conference on Soft Robotics (RoboSoft)*. IEEE. 2022, pp. 490–497.

[BGP22] Shikhar Bahl, Abhinav Gupta, and Deepak Pathak. "Human-to-Robot Imitation in the Wild". In: *CoRR* abs/2207.09450 (2022).

[Bhi+21] Raunaq Bhirangi, Tess Hellebrekers, Carmel Majidi, and Abhinav Gupta. "ReSkin: versatile, replaceable, lasting tactile skins". In: *arXiv preprint arXiv:2111.00071* (2021).

[Bro+20] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. "Language models are few-shot learners". In: *NeurIPS* (2020).

[But+01] Jörg Butterfaß, Markus Grebenstein, Hong Liu, and Gerd Hirzinger. "DLR-Hand II: Next generation of a dextrous robot hand". In: *Proceedings 2001*

ICRA. IEEE International Conference on Robotics and Automation (Cat. No. 01CH37164). Vol. 1. IEEE. 2001, pp. 109–114.

[Car+21] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. "Emerging Properties in Self-Supervised Vision Transformers". In: *CoRR* abs/2104.14294 (2021). arXiv: 2104.14294. URL: https://arxiv.org/abs/2104.14294.

[Che+20] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. "A simple framework for contrastive learning of visual representations". In: *arXiv preprint arXiv:2002.05709* (2020).

[Che+21] Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Michael Laskin, Pieter Abbeel, Aravind Srinivas, and Igor Mordatch. "Decision Transformer: Reinforcement Learning via Sequence Modeling". In: *arXiv preprint arXiv:2106.01345* (2021).

[Che+22a] Annie S Chen, Archit Sharma, Sergey Levine, and Chelsea Finn. "Single-Life Reinforcement Learning". In: *Neural Information Processing Systems 2022*. 2022.

[Che+22b] Tao Chen, Megha Tippur, Siyang Wu, Vikash Kumar, Edward Adelson, and Pulkit Agrawal. "Visual Dexterity: In-hand Dexterous Manipulation from Depth". In: *arXiv preprint arXiv:2211.11744* (2022).

[CNF21] Annie S. Chen, Suraj Nair, and Chelsea Finn. "Learning Generalizable Robotic Reward Functions from "In-The-Wild" Human Videos". In: *CoRR* abs/2103.16817 (2021). arXiv: 2103.16817. URL: https://arxiv.org/abs/2103.16817.

[Cui+22] Yuchen Cui, Scott Niekum, Abhi Gupta, Vikash Kumar, and Aravind Rajeswaran. "Can Foundation Models Perform Zero-Shot Task Specification For Robot Manipulation?" In: *Conference on Learning for Dynamics & Control*. 2022.

[Dam+18] Dima Damen et al. "Scaling Egocentric Vision: The EPIC-KITCHENS Dataset". In: *European Conference on Computer Vision (ECCV)*. 2018.

[Dam+20] Dima Damen et al. *The EPIC-KITCHENS Dataset: Collection, Challenges and Baselines*. 2020. DOI: 10.48550/ARXIV.2005.00343. URL: https://arxiv.org/abs/2005.00343.

[Das+13] Pradipto Das, Chenliang Xu, Richard F Doell, and Jason J Corso. "A thousand frames in just a few words: Lingual description of videos through latent topics and sparse object stitching". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2013, pp. 2634–2641.

[FN15] Bernard Ghanem Fabian Caba Heilbron Victor Escorcia and Juan Carlos Niebles. "ActivityNet: A Large-Scale Video Benchmark for Human Activity Understanding". In: *CVPR*. 2015, pp. 961–970.

[Fu+18] Justin Fu, Avi Singh, Dibya Ghosh, Larry Yang, and Sergey Levine. "Variational inverse control with events: A general framework for data-driven reward definition". In: *arXiv preprint arXiv:1805.11686* (2018).

[FWG15] David F Fouhey, Xiaolong Wang, and Abhinav Gupta. "In defense of the direct perception of affordances". In: *arXiv preprint arXiv:1505.01085* (2015).

[Goy+17a] Raghav Goyal et al. "The "Something Something" Video Database for Learning and Evaluating Visual Common Sense". In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. Oct. 2017.

[Goy+17b] Raghav Goyal et al. *The "something something" video database for learning and evaluating visual common sense*. 2017. DOI: 10.48550/ARXIV.1706.04261. URL: https://arxiv.org/abs/1706.04261.

[Goy+22] Mohit Goyal, Sahil Modi, Rishabh Goyal, and Saurabh Gupta. "Human Hands as Probes for Interactive Object Understanding". In: *CVPR*. 2022.

[Gra+22] Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, et al. "Ego4d: Around the world in 3,000 hours of egocentric video". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 18995–19012.

[Gri+20] Jean-Bastien Grill et al. "Bootstrap Your Own Latent: A New Approach to Self-Supervised Learning". In: *CoRR* abs/2006.07733 (2020). arXiv: 2006.07733. URL: https://arxiv.org/abs/2006.07733.

[Gup+21] Abhishek* Gupta, Justin* Yu, Tony Z* Zhao, Vikash* Kumar, Aaron Rovinsky, Kelvin Xu, Thomas Devlin, and Sergey Levine. "Reset-Free Reinforcement Learning via Multi-Task Learning: Learning Dexterous Manipulation Behaviors without Human Intervention". In: *International Conference on Robotics and Automation(ICRA)* (2021).

[Haa+17] Tuomas Haarnoja, Haoran Tang, Pieter Abbeel, and Sergey Levine. "Reinforcement learning with deep energy-based policies". In: *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org. 2017, pp. 1352–1361.

[Haf+19] Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. "Dream to Control: Learning Behaviors by Latent Imagination". In: *arXiv preprint arXiv:1912.01603* (2019).

[Han+20a] 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: *ICRA*. 2020.

[Han+20b] 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)*. IEEE. 2020, pp. 9164–9170.

[He+15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep Residual Learning for Image Recognition". In: *CoRR* abs/1512.03385 (2015). arXiv: 1512.03385. URL: http://arxiv.org/abs/1512.03385.

[He+22] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. "Masked autoencoders are scalable vision learners". In: *CVPR*. 2022.

[Ion+13] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. "Human3. 6m: Large scale datasets and predictive methods for 3d human sensing in natural environments". In: *IEEE transactions on pattern analysis and machine intelligence* 36.7 (2013), pp. 1325–1339.

[Jan+21] Eric Jang, Alex Irpan, Mohi Khansari, Daniel Kappler, Frederik Ebert, Corey Lynch, Sergey Levine, and Chelsea Finn. "Bc-z: Zero-shot task generalization with robotic imitation learning". In: *CoRL*. 2021.

[Kal+18] Dmitry Kalashnikov, Alex Irpan, Peter Pastor, Julian Ibarz, Alexander Herzog, Eric Jang, Deirdre Quillen, Ethan Holly, Mrinal Kalakrishnan, Vincent Vanhoucke, et al. "Qt-opt: Scalable deep reinforcement learning for vision-based robotic manipulation". In: *arXiv preprint arXiv:1806.10293* (2018).

[Kal+21] Dmitry Kalashnikov, Jacob Varley, Yevgen Chebotar, Benjamin Swanson, Rico Jonschkowski, Chelsea Finn, Sergey Levine, and Karol Hausman. "MT-Opt: Continuous Multi-Task Robotic Reinforcement Learning at Scale". In: *arXiv preprint arXiv:2104.08212* (2021).

[Kan+17] Angjoo Kanazawa, Michael J. Black, David W. Jacobs, and Jitendra Malik. "End-to-end Recovery of Human Shape and Pose". In: *CoRR* abs/1712.06584 (2017). arXiv: 1712.06584. URL: http://arxiv.org/abs/1712.06584.

[Kar+11] Sertac Karaman, Matthew R Walter, Alejandro Perez, Emilio Frazzoli, and Seth Teller. "Anytime motion planning using the RRT". In: *ICRA*. 2011.

[KB14] Diederik P Kingma and Jimmy Ba. "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980* (2014).

[Kha87] Oussama Khatib. "A unified approach for motion and force control of robot manipulators: The operational space formulation". In: *IEEE Journal on Robotics and Automation* 3.1 (1987), pp. 43–53.

[Kir+23] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. "Segment anything". In: *arXiv preprint arXiv:2304.02643* (2023).

[KL00] James J Kuffner and Steven M LaValle. "RRT-connect: An efficient approach to single-query path planning". In: *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)*. Vol. 2. IEEE. 2000, pp. 995–1001.

[KP08] Jens Kober and Jan Peters. "Policy search for motor primitives in robotics". In: *Advances in neural information processing systems* 21 (2008).

[Kum+21] Ashish Kumar, Zipeng Fu, Deepak Pathak, and Jitendra Malik. "RMA: Rapid Motor Adaptation for Legged Robots". In: *Proceedings of Robotics: Science and Systems*. Virtual, July 2021. DOI: 10.15607/RSS.2021.XVII.011.

[KW13] Diederik P Kingma and Max Welling. "Auto-encoding variational bayes". In: *arXiv preprint arXiv:1312.6114* (2013).

[Lee+19] Sungho Lee, Seoung Wug Oh, DaeYeun Won, and Seon Joo Kim. "Copy-and-Paste Networks for Deep Video Inpainting". In: *CoRR* abs/1908.11587 (2019). arXiv: 1908.11587. URL: http://arxiv.org/abs/1908.11587.

[Lev+16] Sergey Levine, Peter Pastor, Alex Krizhevsky, and Deirdre Quillen. "Learning hand-eye coordination for robotic grasping with large-scale data collection". In: *ISER*. 2016.

[Lev18] Sergey Levine. "Reinforcement learning and control as probabilistic inference: Tutorial and review". In: *arXiv preprint arXiv:1805.00909* (2018).

[Li+22] Zhen Li, Cheng-Ze Lu, Jianhua Qin, Chun-Le Guo, and Ming-Ming Cheng. *Towards An End-to-End Framework for Flow-Guided Video Inpainting*. 2022. DOI: 10.48550/ARXIV.2204.02663. URL: https://arxiv.org/abs/2204.02663.

[Lil+16] Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. "Continuous control with deep reinforcement learning". In: *ICLR* (2016).

[Liu+22a] Shaowei Liu, Subarna Tripathi, Somdeb Majumdar, and Xiaolong Wang. "Joint Hand Motion and Interaction Hotspots Prediction from Egocentric Videos". In: *CVPR*. 2022.

[Liu+22b] Yunze Liu, Yun Liu, Che Jiang, Kangbo Lyu, Weikang Wan, Hao Shen, Boqiang Liang, Zhoujie Fu, He Wang, and Li Yi. "HOI4D: A 4D Egocentric Dataset for Category-Level Human-Object Interaction". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2022, pp. 21013–21022.

[Liu+22c] Yunze Liu, Yun Liu, Che Jiang, Kangbo Lyu, Weikang Wan, Hao Shen, Boqiang Liang, Zhoujie Fu, He Wang, and Li Yi. "HOI4D: A 4D Egocentric Dataset for Category-Level Human-Object Interaction". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2022, pp. 21013–21022.

[LK13] Sergey Levine and Vladlen Koltun. "Guided policy search". In: *ICML*. 2013.

[LP17] Kevin M Lynch and Frank C Park. *Modern robotics*. Cambridge University Press, 2017.

[LT17] Changliu Liu and Masayoshi Tomizuka. "Designing the robot behavior for safe human robot interactions". In: *Trends in Control and Decision-Making for Human-Robot Collaboration Systems*. Springer, 2017, pp. 241–270.

[Ma+22a] Yecheng Jason Ma, Shagun Sodhani, Dinesh Jayaraman, Osbert Bastani, Vikash Kumar, and Amy Zhang. "VIP: Towards Universal Visual Reward and Representation via Value-Implicit Pre-Training". In: *arXiv preprint arXiv:2210.00030* (2022).

[Ma+22b] Yecheng Jason Ma, Shagun Sodhani, Dinesh Jayaraman, Osbert Bastani, Vikash Kumar, and Amy Zhang. *VIP: Towards Universal Visual Reward and Representation via Value-Implicit Pre-Training*. 2022. DOI: 10.48550/ARXIV.2210.00030. URL: https://arxiv.org/abs/2210.00030.

[Man+21] Ajay Mandlekar, Danfei Xu, Josiah Wong, Soroush Nasiriany, Chen Wang, Rohun Kulkarni, Li Fei-Fei, Silvio Savarese, Yuke Zhu, and Roberto Martın-Martın. "What matters in learning from offline human demonstrations for robot manipulation". In: *arXiv preprint arXiv:2108.03298* (2021).

[Man+23] Pragna Mannam, Kenneth Shaw, Dominik Bauer, Jean Oh, Deepak Pathak, and Nancy Pollard. *A Framework for Designing Anthropomorphic Soft Hands through Interaction*. 2023. arXiv: 2306.04784 [cs.RO].

[Mar+19] Roberto Martin-Martin, Michelle A. Lee, Rachel Gardner, Silvio Savarese, Jeannette Bohg, and Animesh Garg. "Variable Impedance Control in End-Effector Space: An Action Space for Reinforcement Learning in Contact-Rich Tasks". In: *IROS* (2019).

[Mar+22] Gabriel B Margolis, Ge Yang, Kartik Paigwar, Tao Chen, and Pulkit Agrawal. "Rapid locomotion via reinforcement learning". In: *arXiv preprint arXiv:2205.02824* (2022).

[MBP23] Russell Mendonca, Shikhar Bahl, and Deepak Pathak. "Alan: Autonomously exploring robotic agents in the real world". In: *arXiv preprint arXiv:2302.06604* (2023).

[MG21] Priyanka Mandikal and Kristen Grauman. "DexVIP: Learning Dexterous Grasping with Human Hand Pose Priors from Video". In: *Conference on Robot Learning (CoRL)*. 2021.

[MG22] Priyanka Mandikal and Kristen Grauman. "Dexvip: Learning dexterous grasping with human hand pose priors from video". In: *Conference on Robot Learning*. PMLR. 2022, pp. 651–661.

[Mul22] From One Hand to Multiple Hands: Imitation Learning for Dexterous Manipulation from Single-Camera Teleoperation. *Qin, Yuzhe and Su, Hao and Wang, Xiaolong*. 2022.

[MYB16] Mustafa Mukadam, Xinyan Yan, and Byron Boots. "Gaussian process motion planning". In: *2016 IEEE international conference on robotics and automation (ICRA)*. IEEE. 2016, pp. 9–15.

[Nai+18] Ashvin V Nair, Vitchyr Pong, Murtaza Dalal, Shikhar Bahl, Steven Lin, and Sergey Levine. "Visual reinforcement learning with imagined goals". In: *NeurIPS*. 2018, pp. 9191–9200.

[Nai+21] Suraj Nair, Eric Mitchell, Kevin Chen, Brian Ichter, Silvio Savarese, and Chelsea Finn. "Learning Language-Conditioned Robot Behavior from Offline Data and Crowd-Sourced Annotation". In: *CoRR* abs/2109.01115 (2021). arXiv: 2109.01115. URL: https://arxiv.org/abs/2109.01115.

[Nai+22a] Suraj Nair, Aravind Rajeswaran, Vikash Kumar, Chelsea Finn, and Abhinav Gupta. "R3M: A Universal Visual Representation for Robot Manipulation". In: *arXiv preprint arXiv:2203.12601* (2022).

[Nai+22b] Suraj Nair, Aravind Rajeswaran, Vikash Kumar, Chelsea Finn, and Abhinav Gupta. "R3m: A universal visual representation for robot manipulation". In: *arXiv preprint arXiv:2203.12601* (2022).

[NFG19] Tushar Nagarajan, Christoph Feichtenhofer, and Kristen Grauman. "Grounded Human-Object Interaction Hotspots from Video". In: *ICCV*. 2019.

[Par+22] Simone Parisi, Aravind Rajeswaran, Senthil Purushwalkam, and Abhinav Kumar Gupta. "The Unsurprising Effectiveness of Pre-Trained Vision Models for Control". In: *International Conference on Machine Learning*. 2022.

[Pat+16] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A. Efros. "Context encoders: Feature learning by inpainting". In: *CVPR*. 2016.

[Pat+17] Deepak Pathak, Pulkit Agrawal, Alexei A. Efros, and Trevor Darrell. "Curiosity-driven Exploration by Self-supervised Prediction". In: *ICML*. 2017.

[Pen+18] Xue Bin Peng, Pieter Abbeel, Sergey Levine, and Michiel Van de Panne. "Deepmimic: Example-guided deep reinforcement learning of physics-based character skills". In: *ACM Transactions On Graphics (TOG)* 37.4 (2018), pp. 1–14.

[PMA10] Jan Peters, Katharina Mulling, and Yasemin Altun. "Relative entropy policy search". In: *Twenty-Fourth AAAI Conference on Artificial Intelligence*. 2010.

[Pom88] Dean A Pomerleau. "Alvinn: An autonomous land vehicle in a neural network". In: *Advances in neural information processing systems* 1 (1988).

[Pop+17] Ivaylo Popov, Nicolas Heess, Timothy Lillicrap, Roland Hafner, Gabriel Barth-Maron, Matej Vecerik, Thomas Lampe, Yuval Tassa, Tom Erez, and Martin Riedmiller. "Data-efficient deep reinforcement learning for dexterous manipulation". In: *arXiv preprint arXiv:1704.03073* (2017).

[Rad+21a] Alec Radford et al. "Learning Transferable Visual Models From Natural Language Supervision". In: *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*. Ed. by Marina Meila and Tong Zhang. Vol. 139. Proceedings of Machine Learning Research. PMLR, 2021, pp. 8748–8763.

[Rad+21b] Alec Radford et al. "Learning Transferable Visual Models From Natural Language Supervision". In: *CoRR* abs/2103.00020 (2021). arXiv: 2103.00020. URL: https://arxiv.org/abs/2103.00020.

[Rad+22] Ilija Radosavovic, Tete Xiao, Stephen James, Pieter Abbeel, Jitendra Malik, and Trevor Darrell. "Real-World Robot Learning with Masked Visual Pre-training". In: *CoRR* abs/2210.03109 (2022).

[Raj+17] Aravind Rajeswaran, Vikash Kumar, Abhishek Gupta, Giulia Vezzani, John Schulman, Emanuel Todorov, and Sergey Levine. "Learning complex dexterous manipulation with deep reinforcement learning and demonstrations". In: *arXiv preprint arXiv:1709.10087* (2017).

[RBZ06] Nathan D. Ratliff, J. Andrew Bagnell, and Martin A. Zinkevich. "Maximum Margin Planning". In: *Proceedings of the 23rd International Conference on Machine Learning*. ICML '06. Pittsburgh, Pennsylvania, USA: Association for Computing Machinery, 2006, pp. 729–736. ISBN: 1595933832. DOI: 10.1145/1143844.1143936. URL: https://doi.org/10.1145/1143844.1143936.

[RMW14a] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. "Stochastic backpropagation and approximate inference in deep generative models". In: *International conference on machine learning*. PMLR. 2014, pp. 1278–1286.

[RMW14b] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. "Stochastic backpropagation and approximate inference in deep generative models". In: *arXiv preprint arXiv:1401.4082* (2014).

[Rom+21] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. *High-Resolution Image Synthesis with Latent Diffusion Models*. 2021. DOI: 10.48550/ARXIV.2112.10752. URL: https://arxiv.org/abs/2112.10752.

[RSJ21] Yu Rong, Takaaki Shiratori, and Hanbyul Joo. "FrankMocap: A Monocular 3D Whole-Body Pose Estimation System via Regression and Integration". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*. Oct. 2021, pp. 1749–1759.

[RT15] Daniela Rus and Michael T Tolley. "Design, fabrication and control of soft robots". In: *Nature* 521.7553 (2015), pp. 467–475.

[RTB17] Javier Romero, Dimitrios Tzionas, and Michael J. Black. "Embodied Hands: Modeling and Capturing Hands and Bodies Together". In: *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)*. 245:1–245:17 36.6 (Nov. 2017).

[Rus+14] Olga Russakovsky et al. "ImageNet Large Scale Visual Recognition Challenge". In: *CoRR* abs/1409.0575 (2014). arXiv: 1409.0575. URL: http://arxiv.org/abs/1409.0575.

[SBP22] Kenneth Shaw, Shikhar Bahl, and Deepak Pathak. "VideoDex: Learning Dexterity from Internet Videos". In: *CoRL*. 2022.

[Sch+20] Karl Schmeckpeper, Oleh Rybkin, Kostas Daniilidis, Sergey Levine, and Chelsea Finn. "Reinforcement Learning with Videos: Combining Offline Observations with Interaction". In: *4th Conference on Robot Learning, CoRL 2020, 16-18 November 2020, Virtual Event / Cambridge, MA, USA*. Ed. by Jens Kober, Fabio Ramos, and Claire J. Tomlin. Vol. 155. Proceedings of Machine Learning Research. PMLR, 2020, pp. 339–354.

[Ser+18] Pierre Sermanet, Corey Lynch, Yevgen Chebotar, Jasmine Hsu, Eric Jang, Stefan Schaal, and Sergey Levine. "Time-Contrastive Networks: Self-Supervised Learning from Video". In: *2018 IEEE International Conference on Robotics and Automation, ICRA 2018, Brisbane, Australia, May 21-25, 2018*. IEEE, 2018, pp. 1134–1141. DOI: 10.1109/ICRA.2018.8462891. URL: https://doi.org/10.1109/ICRA.2018.8462891.

[Sha+20] Dandan Shan, Jiaqi Geng, Michelle Shu, and David F Fouhey. "Understanding human hands in contact at internet scale". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 9869–9878.

[Sha+21] Lin Shao, Toki Migimatsu, Qiang Zhang, Karen Yang, and Jeannette Bohg. "Concept2robot: Learning manipulation concepts from instructions and human demonstrations". In: *The International Journal of Robotics Research* 40.12-14 (2021).

[Si+23] Zilin Si, Tianhong Catherine Yu, Katrene Morozov, James McCann, and Wenzhen Yuan. "RobotSweater: Scalable, Generalizable, and Customizable Machine-Knitted Tactile Skins for Robots". In: *arXiv preprint arXiv:2303.02858* (2023).

[Sil+17] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. "Mastering the game of Go without human knowledge". In: *Nature* 550.7676 (2017), p. 354.

[Sin+19] Avi Singh, Larry Yang, Kristian Hartikainen, Chelsea Finn, and Sergey Levine. "End-to-end robotic reinforcement learning without reward engineering". In: *arXiv preprint arXiv:1904.07854* (2019).

[SK21] Rutav M Shah and Vikash Kumar. "RRL: Resnet as representation for Reinforcement Learning". In: *ICML*. 2021.

[SLA20] Aravind Srinivas, Michael Laskin, and Pieter Abbeel. "Curl: Contrastive unsupervised representations for reinforcement learning". In: *arXiv preprint arXiv:2004.04136* (2020).

[SMF22] Mohit Shridhar, Lucas Manuelli, and Dieter Fox. "Cliport: What and where pathways for robotic manipulation". In: *CoRL*. 2022.

[SSP22] Aravind Sivakumar, Kenneth Shaw, and Deepak Pathak. "Robotic telekinesis: learning a robotic hand imitator by watching humans on Youtube". In: *arXiv preprint arXiv:2202.10448* (2022).

[Sun+19] Subramanian Sundaram, Petr Kellnhofer, Yunzhu Li, Jun-Yan Zhu, Antonio Torralba, and Wojciech Matusik. "Learning the signatures of the human grasp using a scalable tactile glove". In: *Nature* 569.7758 (2019). DOI: 10.1038/s41586-019-1234-z.

[SYL15] K. Sohn, X. Yan, and H. Lee. "Learning Structured Output Representation using Deep Conditional Generative Models". In: *NeurIPS*. 2015.

[Tia+21] Stephen Tian, Suraj Nair, Frederik Ebert, Sudeep Dasari, Benjamin Eysenbach, Chelsea Finn, and Sergey Levine. "Model-Based Visual Planning with Self-Supervised Functional Distances". In: *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. 2021.

[Vin+19] Oriol Vinyals et al. "Grandmaster level in StarCraft II using multi-agent reinforcement learning". In: *Nature* (2019), pp. 1–5.

[Wan+20] Jiayi Wang, Franziska Mueller, Florian Bernard, Suzanne Sorli, Oleksandr Sotnychenko, Neng Qian, Miguel A Otaduy, Dan Casas, and Christian Theobalt. "Rgb2hands: real-time tracking of 3d hand interactions from monocular rgb video". In: *ACM Transactions on Graphics (TOG)* 39.6 (2020), pp. 1–16.

[WGG17] Xiaolong Wang, Rohit Girdhar, and Abhinav Gupta. "Binge watching: Scaling affordance learning from sitcoms". In: *CVPR*. 2017.

[WOP15] Markus Wulfmeier, Peter Ondruska, and Ingmar Posner. "Deep Inverse Reinforcement Learning". In: *CoRR* abs/1507.04888 (2015). arXiv: 1507.04888. URL: http://arxiv.org/abs/1507.04888.

[WTB18] Hongbo Wang, Massimo Totaro, and Lucia Beccai. "Toward perceptive soft robots: Progress and challenges". In: *Advanced Science* 5.9 (2018), p. 1800541.

[Wu+19] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. *Detectron2*. https://github.com/facebookresearch/detectron2. 2019.

[Yan+21] Lixin Yang, Xinyu Zhan, Kailin Li, Wenqiang Xu, Jiefeng Li, and Cewu Lu. "CPF: Learning a Contact Potential Field to Model the Hand-Object Interaction". In: *ICCV*. 2021.

[YDA17] Wenzhen Yuan, Siyuan Dong, and Edward H Adelson. "Gelsight: High-resolution robot tactile sensors for estimating geometry and force". In: *Sensors* 17.12 (2017), p. 2762.

[YGT22] Yufei Ye, Abhinav Gupta, and Shubham Tulsiani. "What's in your hands? 3D Reconstruction of Generic Objects in Hands". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 3895–3905.

[Yos85] Tsuneo Yoshikawa. "Dynamic manipulability of robot manipulators". In: *Transactions of the Society of Instrument and Control Engineers* 21.9 (1985), pp. 970–975.

[Yu+19] Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Avnish Narayan, Hayden Shively, Adithya Bellathur, Karol Hausman, Chelsea Finn, and Sergey Levine. *Meta-World: A Benchmark and Evaluation for Multi-Task and Meta Reinforcement Learning*. 2019. DOI: 10.48550/ARXIV.1910.10897. URL: https://arxiv.org/abs/1910.10897.

[Zak+21] Kevin Zakka, Andy Zeng, Pete Florence, Jonathan Tompson, Jeannette Bohg, and Debidatta Dwibedi. "XIRL: Cross-embodiment Inverse Reinforcement Learning". In: *Conference on Robot Learning, 8-11 November 2021, London, UK*. Ed. by Aleksandra Faust, David Hsu, and Gerhard Neumann. Vol. 164. Proceedings of Machine Learning Research. PMLR, 2021, pp. 537–546.

[Zha+22] Lingzhi Zhang, Shenghao Zhou, Simon Stent, and Jianbo Shi. *Fine-Grained Egocentric Hand-Object Segmentation: Dataset, Model, and Applications*. 2022. DOI: 10.48550/ARXIV.2208.03826. URL: https://arxiv.org/abs/2208.03826.

[Zha+23] Tony Z. Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. *Learning Fine-Grained Bimanual Manipulation with Low-Cost Hardware*. 2023. arXiv: 2304.13705 [cs.RO].

[Zho+22] Xingyi Zhou, Rohit Girdhar, Armand Joulin, Phillip Krähenbühl, and Ishan Misra. "Detecting twenty-thousand classes using image-level supervision". In: *arXiv preprint arXiv:2201.02605* (2022).

[Zie+08] Brian D. Ziebart, Andrew Maas, J Andrew Bagnell, and Anind K Dey. "Maximum entropy inverse reinforcement learning". In: *AAAI*. 2008.

[Zim+19] Christian Zimmermann, Duygu Ceylan, Jimei Yang, Bryan Russell, Max Argus, and Thomas Brox. "Freihand: A dataset for markerless capture of hand pose and shape from single rgb images". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 813–822.