Preprint. Under review. 1–18

# Zero-Shot Robot Manipulation from Passive Human Videos

**Homanga Bharadhwaj**[1,2], **Abhinav Gupta**[1], **Shubham Tulsiani**[1,*], **Vikash Kumar**[2,*]
[1] *The Robotics Institute, Carnegie Mellon University*
[2] *Meta AI*
[*] *Equal Contribution*

## Abstract

Can we learn robot manipulation for everyday tasks, only by watching videos of humans doing arbitrary tasks in different unstructured settings? Unlike widely adopted strategies of learning task-specific behaviors or direct imitation of a human video, we develop a a framework for extracting agent-agnostic action representations from human videos, and then map it to the agent's embodiment during deployment. Our framework is based on predicting *plausible* human hand trajectories given an initial image of a scene. After training this prediction model on a diverse set of human videos from the internet, we deploy the trained model *zero-shot* for physical robot manipulation tasks, after appropriate transformations to the robot's embodiment. This simple strategy lets us solve coarse manipulation tasks like opening and closing drawers, pushing, and tool use, without access to *any* in-domain robot manipulation trajectories. Our real-world deployment results establish a strong baseline for action prediction information that can be acquired from diverse arbitrary videos of human activities, and be useful for zero-shot robotic manipulation in unseen scenes. [1]

**Keywords:** Learning from human videos, robot manipulation

## 1. Introduction

We humans effortlessly perform a plethora of manipulation tasks in our everyday lives, for example opening cabinets, cutting vegetables, pouring coffee, turning knobs, etc. A common goal in the rapidly growing area of (data-driven) robot learning is to develop agents that can similarly perform diverse tasks in unstructured settings. Deep reinforcement learning based methods Kalashnikov et al. (2021) provide a framework that allows robots to continually improve at performing generic tasks by optimizing a corresponding reward. However, the sample (in)efficiency, the need for online interactions, and the difficulty in designing rewards and environment resets typically narrows their application to specific tasks in structured environments. An alternate approach is to directly learn action policies from robot demonstrations with experts Pomerleau (1989); Finn et al. (2017). While demonstrations across varied settings and tasks can in principle allow learning the desired diverse behaviors, these are typically collected either through tele-operation or kinesthetic teaching and are thus difficult to scale. Moreover, both the online interactions and the robot demonstrations used in these approaches apriori require task definitions, and are restricted to lab settings with little variation, thus being a far cry from the diverse data required to train generalist robots effective in unknown unstructured environments.

Is there an alternate source of data that can enable learning for robot manipulation? In this work, we show that videos of humans interacting with objects as they accomplish everyday tasks serve as a

1. hbharadh@cs.cmu.edu Videos https://sites.google.com/view/human-0shot-robot

© H.B.A.G.S.T.K. .

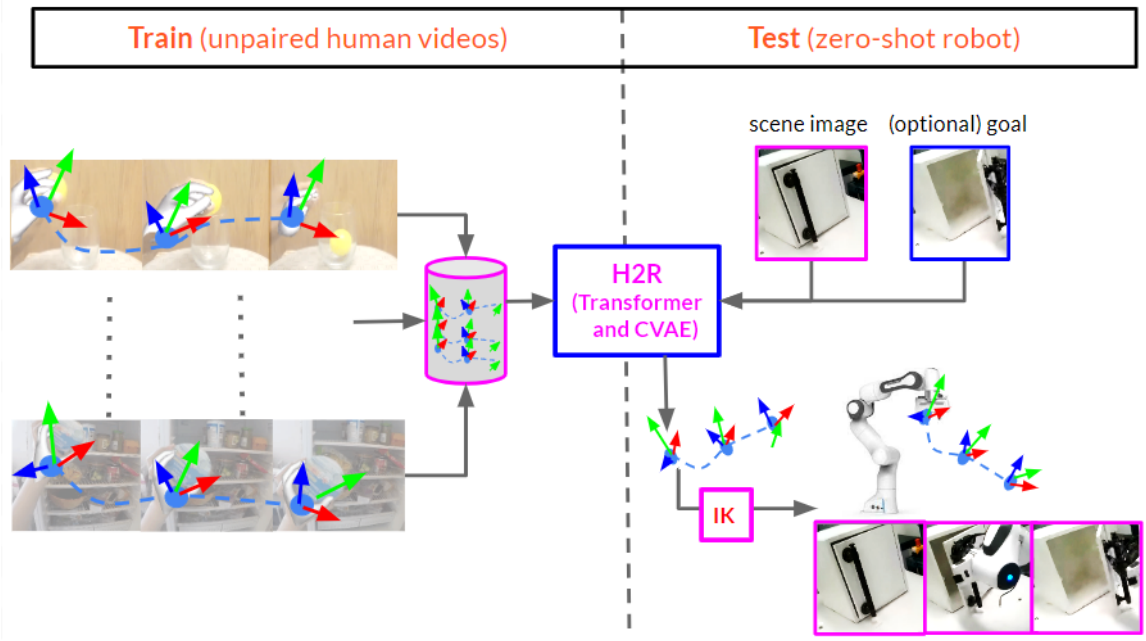


Figure 1: Overview of the proposed framework for zero-shot robot manipulation from passive human videos. Our approach is based on hand trajectory prediction given an image of a scene, from human videos on the web, and transforming the predictions for zero-shot robot manipulation given an unseen scene in the lab.

readily available source of such large-scale data. Specifically, we show that modeling the observed motion of the human hands across human-object interaction videos allows *zero-shot* prediction of the actions a robot should take to achieve goals in its environment. We develop a framework where given an initial image, the model learns to predict the sequence of motions of a human hand acting in the scene in subsequent frames. In particular, the predictions involve predicting a 6D pose of a point on the center of the palm of the human hand, over a 2 second future horizon from the initial image. Since there are several different plausible actions possible in scene, we make the model stochastic, and train it with diverse egocentric videos from existing datasets like Epic-Kitchens Damen et al. (2018). Due to the scale and diversity of the human videos, the prediction model allows us to perform zero-shot action prediction in different physical environments with novel objects. We call such human videos *passive* as they haven't been actively collected for the purpose of robot learning with any task-level segregation, and our framework *zero-shot* because there are no in-lab demonstrations or fine-tuning. Fig. 1 shows an overview of our framework for zero-Shot robot manipulation from passive Human videos

We develop two versions of the framework: un-conditioned, and goal-conditioned. The un-conditioned model predicts plausible trajectories given a scene, and corresponds to realizing the different affordances of objects in a scene that can be helpful for downstream exploration. The goal-conditioned model is further prompted with a goal image to predict trajectories that reach the specified goal, and is helpful in achieving targeted behavior. To manipulate a robot with the predictions, we transform them to the end-effector space of the robot, and use a simple IK controller to execute the motion of the trajectory. We show that this simple approach works reliably well

with around 40-60% success rate for unconditionally manipulating objects like toasters, drawers, bowls etc. (see Fig. 3) and with around 30-40% success rate for goal-conditioned manipulation. Finally, we note that incorporating some kind of fine-tuning with environment-specific data would definitely lead to better performance, for fine-grained manipulation tasks, but our current framework is a surprisingly strong baseline of what can be done zero-shot from robot-agnostic human data.

## 2. Related Works

We discuss prior works that attempt understanding of human activities in videos through detection of hand-poses, semantic tasks, visual feature extraction, and activity forecasting. We follow this with a discussion of papers that learn robot manipulation skills from videos of humans and robots performing different activities.

**Scaling Human-Object Interaction Understanding.** Understanding human activities has recently received a lot of interest with development of large-scale datasets Goyal et al. (2017); Das et al. (2013); De la Torre et al. (2009); Grauman et al. (2022) that involve recording videos of humans doing cooking related activities in their kitchen Damen et al. (2018); Li et al. (2018), short clips involving manipulating objects Goyal et al. (2017), and more diverse and long clips of activities both inside and outside homes Grauman et al. (2022); Shan et al. (2020).

Based on these large datasets of human videos, several prior works have focused on understanding human-object interactions Shan et al. (2020); Ye et al. (2022). Specifically, prior work has investigated object pose estimation Kehl et al. (2017); Rad and Lepetit (2017); Xiang et al. (2018); Hu et al. (2019); He et al. (2020), hand pose estimation Zimmermann and Brox (2017); Iqbal et al. (2018); Spurr et al. (2018); Ge et al. (2019); Baek et al. (2019); Boukhayma et al. (2019); Hasson et al. (2019); Kulon et al. (2020); Liu et al. (2021), full body pose estimation Rong et al. (2020), and hand-object joint pose estimation Ye et al. (2022); Garcia-Hernando et al. (2018); Hampali et al. (2020); Liu et al. (2021); Chao et al. (2021). These efforts have been aptly complemented by datasets of 3D scans of real objects – like YCB Calli et al. (2015), and Google Obbject Scans Downs et al. (2022) that humans typically interact with. Another line of work has investigated learning interaction hotspots Nagarajan et al. (2019); Liu et al. (2022); Goyal et al. (2022) from videos, and predicting plausible grasps Mo et al. (2021); Brahmbhatt et al. (2019).

Building upon these developments, which focused primarily on visual understanding, our work focuses on *closing the vision-robotics loop* by using large passive datasets of human videos. We learn plausible hand trajectories for interaction with objects, and deploy them for real robot manipulation. In the next sub-section we outline how our framework differs from robot learning approaches that learn manipulation skills from videos.

**Robot Manipulation from Videos** Recent developments in robotics has extensively focused on using increasingly unstructured data. Visual imitation learning approaches aim to learn control policies from datasets of visual observations and robot actions Finn et al. (2017); Young et al. (2020a); Mandlekar et al. (2018). Behavior cloning Pomerleau (1989); Bain and Sammut (1995); Ross and Bagnell (2010); Bojarski et al. (2016); Torabi et al. (2018a) and inverse reinforcement learning Russell (1998); Ng et al. (2000); Abbeel and Ng (2004); Ho and Ermon (2016); Fu et al. (2017); Aytar et al. (2018); Torabi et al. (2018b); Liu et al. (2020) are two popular approaches in this regime, but are difficult to scale to unseen scenes as they require high quality in-domain expert robot

trajectories (typically) with a human controlling the robot. To alleviate the need for collecting high quality robot data, some approaches have used videos of humans doing things Schmeckpeper et al. (2019); Chang et al. (2020); Schmeckpeper et al. (2020); Shaw et al.; Shao et al. (2020); Song et al. (2020); Young et al. (2020b) for learning control policies either through imitation of reinforcement learning. However, for imitation, data needs to be collected through special hardware interfaces by humans in different scenes for visual imitation Young et al. (2020b); Song et al. (2020) which is hard to scale. For the RL based approaches, the frameworks require simulation environments for online interactions with resets and rewards to provide feedback during learning Shao et al. (2020); Schmeckpeper et al. (2020). Compared to these approaches, we do not require any specially collected data, or simulators for learning, thereby generalizing zero-shot to unknown tasks.

Other works have investigated learning robot motions through a direct imitation of a corresponding human video in the scene Peng et al. (2018); Pathak et al. (2018); Sharma et al. (2018); Sieb et al. (2020); Sivakumar et al. (2022); Garcia-Hernando et al. (2020); Xiong et al. (2021); Sermanet et al. (2018); Sharma et al. (2019); Smith et al. (2019); Peng et al. (2020); Bahl et al. (2022); Qin et al. (2021). Some of these approaches Smith et al. (2019); Xiong et al. (2021) although do not require large-scale datasets, requre near-perfect alignment in the poses of the robot and human arms. Recent works Bahl et al. (2022) have alleviated the need for this perfect alignment, through in-painting of humans from the scene to construct a reward function, but they still require per-task online fine-tuning through RL, which is expensive in the real-world. DexMV Qin et al. (2021) combines a vision and simulation pipeline to effectively translate human videos to dextrous hand motions, but requires several in-lab human videos for training and cannot utilize existing human videos on the web.

Compared to these prior works, we extract agent-agnostic representations in the form of hand-trajectories from human videos on the internet. Instead of having a video stream to mimic directly, we learn a scene-conditioned trajectory prediction model that relies on a single image observation as input and can be directly used for robot manipulation zero-shot. This enables generalization of manipulation capabilities to unseen scenes, without apriori notions of task specification.

## 3. Approach

Motivated by the intuitive understanding humans have about how a scene can be manipulated in interesting ways, we develop a computational framework that can endow robots with similar manipulation skills, by only observing videos of humans interacting with objects in diverse unstructured settings. Specifically, given an initial image of a scene $o_1$ from a single viewpoint, we learn a model $p(a_{1:T}|o_1)$ that predicts future actions $a_{1:T}$ taken by humans while interacting in the scene. We focus on table-top manipulation setting where $a_{1:T}$ corresponds to plausible 6D poses of a right human hand. In order to make the framework generally applicable for a robot arm with a simple end-effector like a two-finger gripper, we do not model the hand articulation (i.e. how each joint is oriented with respect to the wrist), and focus on predicting only the position and orientation of the palm of the hand. After training this prediction model across diverse internet videos, we apply it to an in-lab setting with no additional fine-tuning on lab data, for manipulating objects like drawers, cabinets, and doors on a table top setting with a fixed camera.

### 3.1. Learning to Predict Future Hand Trajectories

Given an image of a scene, there could be several plausible trajectories that modify objects in the scene. To capture this multi-modality, we develop a stochastic model for $p(a_{1:T}|o_1)$ such that

different samples from the model yield different trajectories. Instead of predicting absolute hand poses at each time-step $h_t$, we predict delta actions $a_t = h_t \ominus h_{t-1}$, where $\ominus$ is an appropriate "difference operator." This minimizes error in predictions by reducing the absolute magnitude of model predictions. In addition, to enforce temporal correlation, the model is auto-regressive such that it takes in previous actions within a trajectory as input while predicting the next action. In the next sub-sections, we describe the prediction model architecture, and the pre-processing of data for training the model.

### 3.1.1. MODEL ARCHITECTURE

The hand prediction model is an Image Transformer Bertasius et al. (2021); Chen et al. (2020), with an Encoder-Decoder architecture, and a final C-VAE hand trajectory prediction head conditioned on the decoder feature outputs. We show an overview of the architecture in Fig. 2. Similar to prior works Furnari and Farinella (2019); Wang et al. (2016), we first extract features offline from the initial image $o_1$, and denote it as $f_1$. The feature $f_1$ is encoded by the transformer encoder $\mathcal{E}$ into a latent code $z_1^e$. The transformer decoder $\mathcal{D}$ conditions on the latent code $z_1^e$ and the previous actions $a_{1:t-1}$ to output the feature for action prediction at time-step $t$, $z_t^d$ .

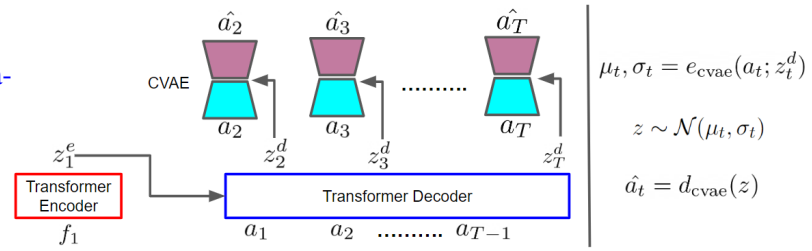


Figure 2: Architecture of the hand trajectory prediction model showing the unconditioned trajectory prediction model given features of the initial scene. On the right we show the process of inference from the model. Details are mentioned in section 3.1.1

During training, a CVAE takes as input the action $a_t$ and conditional context $z_t^d$, and outputs reconstruction $\hat{a}_t$. During inference, we simply sample from the prior of this CVAE, concatenated with the predicted context $z_t^d$, and obtain the sampled delta action $\hat{a}_t$. After obtaining the delta actions we can recover the predicted actions as $\hat{a}_t = \hat{a_{t-1}} \oplus \hat{a}_t \; \forall t > 1$, where $\oplus$ is an appropriate "addition operator." For the goal-conditioned model, the Transformer encoder also takes in a goal image embedding $f_g$ with appropriate positional embedding, and outputs $z_g^e$. We mention specific details of the models in the Appendix.

**Transformer.** The transformer encoder and decoder consist of several stacked blocks of operations that involve attention and MLP with LayerNorm. The decoder blocks have cross-attention with the query and value tokens being the encoded code $z_1^e$. Whereas, in the encoder the attention blocks are all self-attention.

**CVAE.** The CVAE prediction head at each time-step consists of an encoder $e_{\text{cvae}}(\cdot)$ and a decoder $d_{\text{cvae}}(\cdot)$ neural networks. Conditioned on the predicted feature from the transformer decoder $z_t^d$, and the current delta action $a_t$ as input, the CVAE encoder outputs the mean $\mu_t$ and S.D. $\sigma_t$ of a Gaussian distribution. The CVAE decoder samples from this Gaussian and outputs a predicted delta action $\hat{a}_t$. Formally, $\mu_t, \sigma_t = e_{\text{cvae}}(a_t; z_t^d) \qquad z \sim \mathcal{N}(\mu_t, \sigma_t) \qquad \hat{a}_t = d_{\text{cvae}}(z)$

Figure 3: Figure showing a few configurations of the objects we place in the scene for our experiments, with green arrows denoting plausible motions; from left to right in the top row: a door with a vertical hinge, a bowl of fruits, a chopping board with veggies; in the bottom row: a toaster oven with horizontal hinge, a stack of two drawers, and a vegetable strainer.

**Training loss.** The overall training loss is defined in terms of the output of the CVAE per-timestep aggregated over all time-steps $T$ in the prediction horizon.

$$\mathcal{L} = \sum_{t=2}^{T} \left[ ||a_t - \hat{a}_t||^2 - D_{\text{KL}} \left( \mathcal{N}(\mu_t, \sigma_t) || \mathcal{N}(0, 1) \right) \right]$$

This overall loss is backpropagated through the entire prediction model that involves both the Transformer and the CVAE (through re-parameterization), and there is no intermediate supervision for any of the stages. This makes the approach generally applicable to in-the-wild human videos and reduces pre-processing overhead for training data, as described in the next section. In addition, the learned model is task-agnostic since there is no task-specific distinction in the human video clips, and given unseen scenes, it allows performing *plausible* tasks zero-shot.

#### 3.1.2. TRAINING DATA GENERATION

We consider 2 second clips of egocentric videos from Epic-Kitchens that involve people doing everyday household activities, especially in the kitchen, like cooking food, opening cabinets, moving objects from one location to another etc. For the goal-conditioned model, the last image of the clip corresponds to the goal that is input to the model. To obtain ground-truth poses of the hand in the frames within prediction window, we run hand-tracking with an off-the-shelf FrankMocap model Rong et al. (2020). FrankMocap outputs a weak-perspective camera $(t_x, t_y, s)$ and $(x, y, z)$ locations of all the hand joints, and orientation of the hand $(\alpha, \beta, \gamma)$ relative to a canonical hand. We only consider the center of the palm, and record its 6D pose relative to the predicted camera in the beginning of the 2 second window. In summary, the training data consists of pairs $\{(o_1, h_{1:T})\}$ where $o_1$ is the first image of a video clip and $h_t = (x_t, y_t, s_t, t_x, t_y, \alpha, \beta, \gamma)$ is the hand pose and camera parameters at future time $t$.

### 3.2. Mapping Trajectories to the Robot's Frame

After training the overall hand pose prediction model, $p_\psi(a_{1:T}|o_1)$ with diverse internet videos, we deploy it directly for robot manipulation tasks in the lab. The robot sees an image of the scene through a fixed camera, and optionally receives a goal-image which is input to the prediction model. In order to use the actions predicted by the model $a_{1:T}$ for moving the robot, we need to transform them to the world coordinate frame of the robot, and considering each action $a_t$ as an end-effector target pose, use a low-level controller for executing the respective motions.

The camera in the scene is calibrated, so the intrinsic matrix $I$ and the extrinsic matrix $[R, T]$ are known. The world coordinates are located at the base of the robot (robot base is at same height as the table top) and the height of the table top from the camera is known and approximately constant. Given scene from the camera $o_1$, the model predicts delta actions $a_{1:T}$ which we convert to absolute actions (described in section 3.1.1), and transform the actions from the camera frame to the world frame of the robot through inverse projection transformation. The prediction horizon is $T = 7$ for our experiments. After obtaining the world coordinates of the action sequence $\{(X_t, Y_t, Z_t, \alpha_t, \beta_t, \gamma_t)\}_{t=1}^{T}$, we use an IK controller to execute the corresponding motion for bringing the end-effector to the desired position and orientation and each time-step. We describe additional details in the Appendix.

## 4. Experiment Design

Through experiments, we aim to understand the following research questions:

- In unknown tasks, how good is our unconditional model in generating diverse but plausible task outcomes?
- In known tasks, how good is our goal-conditional model and action mapping to solve tasks?

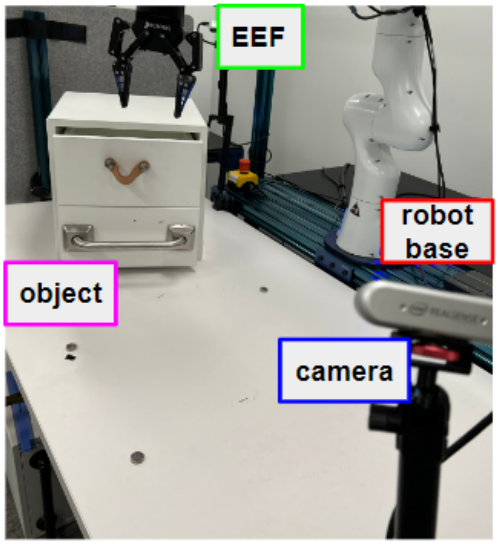


Figure 4: Robot workspace showing the camera, an object in the scene, the Franka arm, and it's end-effector.

Qualitatively, we visualize the diversity and plausibility of trajectories executed for different scenes with the unconditioned model. For quantitative evaluations, we compare success rates of the both the unconditioned model predictions when the robot is placed in different scenes, and the goal-conditioned model when prompted in addition with a desired goal configuration of objects. We compare against a 3D scene flow Vedula et al. (1999); Teed and Deng (2021) baseline that uses RAFT3D Teed and Deng (2021) for predicting scene flow field between the initial and goal images, and uses the dominant flow direction to guide the motion of the robot. To test the importance of training across diverse data, we compare against a version of our method that is trained on only 30% of the total training data. We mention additional details of the setup, objects, tools, and comparisons for the experiments in the Appendix[2].

**Experimental setup**: The workspace shown in Fig. 4 is a white table with a camera in one of the corners, and the robot base on it's opposite edge. In Fig. 3, we show the different objects we place in the scene for our experiments. Note that all of the objects are unseen by definition because the egocentric videos used for training our models are from existing datasets on the web and do not involve any in-lab data.

2. https://sites.google.com/view/human-0shot-robot

Table 1: Quantitative evaluation results for the unconditioned model, with 20 trials for each object. For each object, we measure success differently. For a door, drawer, and toaster, success is when the robot opens the respective object from a closed initial position, or closes the respective object from an open initial position. For the bowl, success is when the bowl is moved from an initial position to a different position without toppling. For the moving of veggies task, success is when at least half the pieces on the table are moved.

| | **Drawer** | **Door** | **Toaster** | **Bowl** | **Veggies** | **Average** |
|---|---|---|---|---|---|---|
| **Our method** | 50% | 45% | 50% | 55% | 50% | 50% |
| **Our method (less data)** | 10% | 10% | 15% | 20% | 15% | 14% |

## 5. Results

In this section, we discuss results for the un-conditional, and goal-conditional models followed by an analysis of failures. We show both qualitative results for visualization, and quantitative evaluations over several trials.

### 5.1. Un-conditional generation results

For the un-conditional generations, given a scene, the model predicts a sequence of actions, which are executed by the robot. We evaluate the model in terms of whether the trajectories executed by the robot correspond to *plausible* interactions in the scene that a human is likely to do. For example, a closed door can be opened, and a bowl of fruits can be moved around on the table i.e. given a scene, there is a distribution of plausible object state changes. Fig. 6 shows visualizations of of predicted trajectories executed by the robot in different initial scenes.

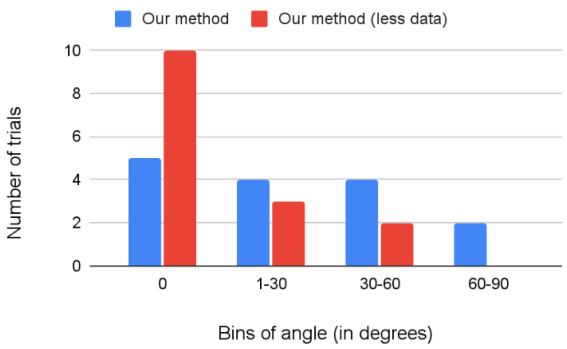


Figure 5: Fine-grained analysis of the door opening task. Histogram shows number of trials (out of 15) that open the door to a certain angle.

In Table 1 we analyze the unconditioned model generations quantitatively, for different objects in the scene with definitions of success criteria different for different objects. For a door, drawer, and toaster, success is when the robot changes objects' state - from open to close or vice versa. For the bowl, success is when the bowl is moved from an initial position to a different position without toppling. From the table we see that success rates vary between 45% and 55% indicating that the model is able to predict trajectories that perform plausible interactions in a scene. Fig. 5, we perform a finer analysis of the model predictions when a closed door is placed in the scene, and observe that over 70% of the trials open the door to non-zero angle, indicating plausible state change of the object.

### 5.2. Goal-conditioned generation results

In addition to executing plausible trajectories in a given scene, we want to understand how effective is the goal-conditioned model in generating trajectories that reach a specified goal from an initial scene. Fig. 7 shows results for trajectories corresponding to different goal images. In Table 2 we

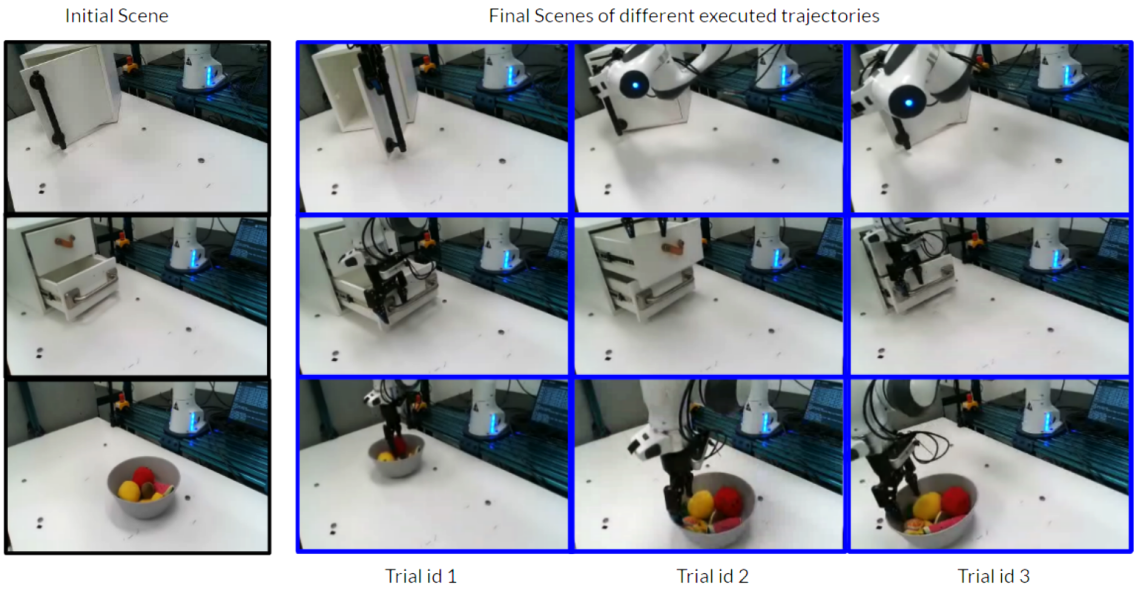


Figure 6: Figure showing final configurations corresponding to different executed trajectories (trial ids 1,2,3) by the unconditional model in the scenes, showing the diversity of plausible behaviors predicted by our model.

show quantitative results for different types of goal images. For each setting, we randomize the initial pose of the object, for example for a 'open door' goal image, the initial position of the door is either closed, or is half-open. We see that the goal-conditioned model succeeds in predicting trajectories that succeed on an average around 37% of the times, which is lower than that of the unconditioned model because the task of reaching a particular goal is more difficult than bringing unconditional changes to a scene. We observe better performance compared to the baselines trained on less data, and the scene flow baseline. In Table 3 we do a finer analysis of our model for a certain scene with stacked drawers and observe that when conditioned on a goal, the goal-conditioned model reaches the final configuration specified by the goal more number of times than the unconditional model reaches the same. This demonstrates that goal-conditioning actually leads to predicted trajectories consistent with the specified goal.

Table 2: Quantitative evaluation results for the goal-conditioned model, with 20 trials for each goal category. Each column denotes the type of goal image specified. In the drawer object, we aggregate results for the top and bottom drawers in the opening and closing tasks, respectively.

| | **Open Drawer** | **Open Door** | **Close Drawer** | **Close Door** | **Move Bowl** | **Move Veggies** | **Average** |
|---|---|---|---|---|---|---|---|
| **Our method** | 35% | 30% | 35% | 40% | 45% | 35% | 37% |
| **Our (less data)** | 10% | 5% | 10% | 15% | 15% | 10% | 11% |
| **3D Scene Flow** | 15% | 15% | 5% | 10% | 15% | 10% | 12% |

Table 3: Given a scene with bottom drawer half open, and top drawer closed, we evaluate 10 trials of the unconditioned model, and count trials (out of 10) that reach a certain final configurations . We compare this to the goal-conditioned model that gets conditioned on a goal corresponding to a final configuration (10 trials per goal).

| Final Config Reached./ Goal | Top Open | Bottom Open | Bottom Close |
|---|---|---|---|
| **Unconditioned (10 trials overall)** | 1/10 | 3/10 | 2/10 |
| **Goal-conditioned (10 trials each)** | 3/10 | 4/10 | 4/10 |

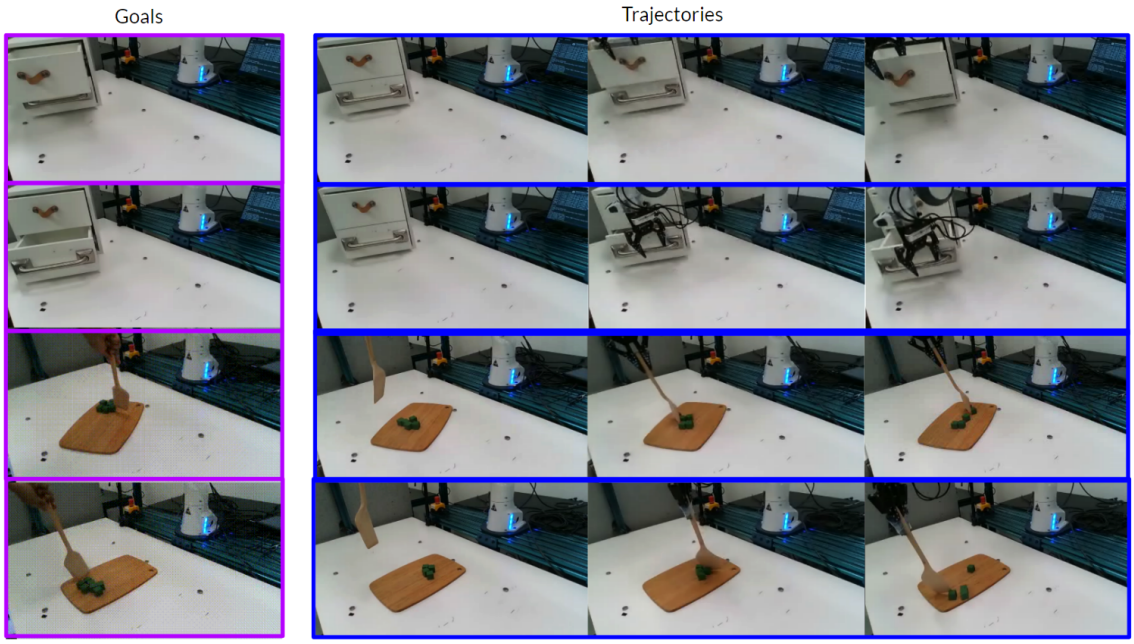


Figure 7: Figure showing different goal-conditioned robot evaluations for goals corresponding to the one shown on the left. Each row corresponds to a sub-sampled trajectory, respectively showing top and bottom drawers being opened, door being opened and closed, and veggies being moved on a table.

### 5.3. Analysis of failures

For the unconditioned model, we observe two primary failure modes: when the robot fails to make contact with the object in the initial time-step (about 40% of all failures) and when it does make contact, but the resulting motion is not feasible and the robot gets stuck (about 60% of all failures). The second failure mode corresponds to a predicted trajectory that doesn't align with the affordances of the respective object, for example trying to pull a door outwards, or pressing against the side of a bowl resulting in the bowl toppling on the table. In addition to the above failures, for the goal conditioned model we observe a third failure mode, where the executed trajectory is plausible (for example a closer drawer is opened) but doesn't correspond to the specified goal (for example the goal shows the top drawer opened, whereas the executed trajectory opens the bottom drawer). This corresponds to about 40% failures of the model.

## 6. Discussion and Conclusion

In this paper we developed an approach for predicting plausible action trajectories from passive human videos, for zero-shot robot interaction given a scene with objects. After learning a model to predict agent-agnostic action trajectories from human videos on the web, we transformed the predictions to the robot's embodiment, and executed the motions zero-shot in a robot workspace without fine-tuning on any in-lab data. For everyday objects like drawers, toaster ovens, doors, and fruit bowls we observed that the predicted trajectories performed plausible interactions with a success rate of around 50%, for example opening a drawer that is closed, and moving a bowl on the table. We further developed a goal-conditioned version of the model that conditions on an initial image of a scene, and a goal image, trained again from egocentric human videos on the internet where the goal image is a final image of a video clip. When the goal-conditioned model is deployed in the robot workspace, we observed around 40% success rate in reaching a specified goal, for example a door fully open, from an initial scene, for example a door half open. We believe our approach for scene-conditioned action trajectory generation would help understand the limits of extracting action representations from passive human videos alone, such that they are useful for robot manipulation zero-shot without any fine-tuning.

## References

Pieter Abbeel and Andrew Y Ng. Apprenticeship learning via inverse reinforcement learning. 2004.

Yusuf Aytar, Tobias Pfaff, David Budden, Thomas Paine, Ziyu Wang, and Nando de Freitas. Playing hard exploration games by watching youtube. In *NeurIPS*, 2018.

Seungryul Baek, Kwang In Kim, and Tae-Kyun Kim. Pushing the envelope for rgb-based dense 3d hand pose estimation via neural rendering. In *CVPR*, 2019.

Shikhar Bahl, Abhinav Gupta, and Deepak Pathak. Human-to-robot imitation in the wild. *arXiv preprint arXiv:2207.09450*, 2022.

Michael Bain and Claude Sammut. A framework for behavioural cloning. In *Machine Intelligence*, 1995.

Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? In *ICML*, volume 2, page 4, 2021.

Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Prasoon Goyal, Lawrence D Jackel, Mathew Monfort, Urs Muller, Jiakai Zhang, et al. End to end learning for self-driving cars. *arXiv*, 2016.

Adnane Boukhayma, Rodrigo de Bem, and Philip HS Torr. 3d hand shape and pose from images in the wild. In *CVPR*, 2019.

Samarth Brahmbhatt, Ankur Handa, James Hays, and Dieter Fox. Contactgrasp: Functional multi-finger grasp synthesis from contact. *arXiv*, 2019.

Berk Calli, Aaron Walsman, Arjun Singh, Siddhartha Srinivasa, Pieter Abbeel, and Aaron M Dollar. Benchmarking in manipulation research: The ycb object and model set and benchmarking protocols. *arXiv*, 2015.

Matthew Chang, Arjun Gupta, and Saurabh Gupta. Semantic visual navigation by watching youtube videos. In *NIPS*, 2020.

Yu-Wei Chao, Wei Yang, Yu Xiang, Pavlo Molchanov, Ankur Handa, Jonathan Tremblay, Yashraj S Narang, Karl Van Wyk, Umar Iqbal, Stan Birchfield, et al. Dexycb: A benchmark for capturing hand grasping of objects. In *CVPR*, 2021.

Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, and Ilya Sutskever. Generative pretraining from pixels. In *International conference on machine learning*, pages 1691–1703. PMLR, 2020.

Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, et al. Scaling egocentric vision: The epic-kitchens dataset. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 720–736, 2018.

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*, pages 2634–2641, 2013.

Fernando De la Torre, Jessica Hodgins, Adam Bargteil, Xavier Martin, Justin Macey, Alex Collado, and Pep Beltran. Guide to the carnegie mellon university multimodal activity (cmu-mmac) database. 2009.

Laura Downs, Anthony Francis, Nate Koenig, Brandon Kinman, Ryan Hickman, Krista Reymann, Thomas B McHugh, and Vincent Vanhoucke. Google scanned objects: A high-quality dataset of 3d scanned household items. *arXiv preprint arXiv:2204.11918*, 2022.

Chelsea Finn, Tianhe Yu, Tianhao Zhang, Pieter Abbeel, and Sergey Levine. One-shot visual imitation learning via meta-learning. In *Conference on robot learning*, pages 357–368. PMLR, 2017.

Justin Fu, Katie Luo, and Sergey Levine. Learning robust rewards with adversarial inverse reinforcement learning. *arXiv*, 2017.

Antonino Furnari and Giovanni Maria Farinella. What would you expect? anticipating egocentric actions with rolling-unrolling lstms and modality attention. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6252–6261, 2019.

Guillermo Garcia-Hernando, Shanxin Yuan, Seungryul Baek, and Tae-Kyun Kim. First-person hand action benchmark with rgb-d videos and 3d hand pose annotations. *CVPR*, 2018.

Guillermo Garcia-Hernando, Edward Johns, and Tae-Kyun Kim. Physics-based dexterous manipulations with estimated hand poses and residual reinforcement learning. *arXiv*, 2020.

Liuhao Ge, Zhou Ren, Yuncheng Li, Zehao Xue, Yingying Wang, Jianfei Cai, and Junsong Yuan. 3d hand shape and pose estimation from a single rgb image. In *CVPR*, 2019.

Mohit Goyal, Sahil Modi, Rishabh Goyal, and Saurabh Gupta. Human hands as probes for interactive object understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3293–3303, 2022.

Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzynska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fruend, Peter Yianilos, Moritz Mueller-Freitag, et al. The" something something" video database for learning and evaluating visual common sense. In *Proceedings of the IEEE international conference on computer vision*, pages 5842–5850, 2017.

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*, pages 18995–19012, 2022.

Shreyas Hampali, Mahdi Rad, Markus Oberweger, and Vincent Lepetit. Honnotate: A method for 3d annotation of hand and object poses. In *CVPR*, 2020.

Yana Hasson, Gul Varol, Dimitrios Tzionas, Igor Kalevatykh, Michael J Black, Ivan Laptev, and Cordelia Schmid. Learning joint reconstruction of hands and manipulated objects. In *CVPR*, 2019.

Yisheng He, Wei Sun, Haibin Huang, Jianran Liu, Haoqiang Fan, and J. Sun. Pvn3d: A deep point-wise 3d keypoints voting network for 6dof pose estimation. *CVPR*, 2020.

Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. In *NeurIPS*, 2016.

Yinlin Hu, Joachim Hugonot, Pascal Fua, and Mathieu Salzmann. Segmentation-driven 6d object pose estimation. *CVPR*, 2019.

Umar Iqbal, Pavlo Molchanov, Thomas Breuel Juergen Gall, and Jan Kautz. Hand pose estimation via latent 2.5 d heatmap regression. In *ECCV*, 2018.

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. *arXiv preprint arXiv:2104.08212*, 2021.

Wadim Kehl, Fabian Manhardt, Federico Tombari, Slobodan Ilic, and Nassir Navab. Ssd-6d: Making rgb-based 3d detection and 6d pose estimation great again. *ICCV*, 2017.

Dominik Kulon, Riza Alp Guler, Iasonas Kokkinos, Michael M. Bronstein, and Stefanos Zafeiriou. Weakly-supervised mesh-convolutional hand reconstruction in the wild. In *CVPR*, 2020.

Yin Li, Miao Liu, and James M Rehg. In the eye of beholder: Joint learning of gaze and actions in first person video. In *Proceedings of the European conference on computer vision (ECCV)*, pages 619–635, 2018.

Fangchen Liu, Zhan Ling, Tongzhou Mu, and Hao Su. State alignment-based imitation learning. In *ICLR*, 2020.

Shaowei Liu, Hanwen Jiang, Jiarui Xu, Sifei Liu, and Xiaolong Wang. Semi-supervised 3d hand-object poses estimation with interactions in time. In *CVPR*, 2021.

Shaowei Liu, Subarna Tripathi, Somdeb Majumdar, and Xiaolong Wang. Joint hand motion and interaction hotspots prediction from egocentric videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3282–3292, 2022.

Ajay Mandlekar, Yuke Zhu, Animesh Garg, Jonathan Booher, Max Spero, Albert Tung, Julian Gao, John Emmons, Anchit Gupta, Emre Orbay, et al. Roboturk: A crowdsourcing platform for robotic skill learning through imitation. In *Conference on Robot Learning*, pages 879–893. PMLR, 2018.

Kaichun Mo, Leonidas J Guibas, Mustafa Mukadam, Abhinav Gupta, and Shubham Tulsiani. Where2act: From pixels to actions for articulated 3d objects. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6813–6823, 2021.

Tushar Nagarajan, Christoph Feichtenhofer, and Kristen Grauman. Grounded human-object interaction hotspots from video. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8688–8697, 2019.

Andrew Y Ng, Stuart J Russell, et al. Algorithms for inverse reinforcement learning. 2000.

Deepak Pathak, Parsa Mahmoudieh, Guanghao Luo, Pulkit Agrawal, Dian Chen, Yide Shentu, Evan Shelhamer, Jitendra Malik, Alexei A. Efros, and Trevor Darrell. Zero-shot visual imitation. In *ICLR*, 2018.

Xue Bin Peng, Angjoo Kanazawa, Jitendra Malik, Pieter Abbeel, and Sergey Levine. Sfv: Reinforcement learning of physical skills from videos. *TOG*, 2018.

Xue Bin Peng, Erwin Coumans, Tingnan Zhang, Tsang-Wei Lee, Jie Tan, and Sergey Levine. Learning agile robotic locomotion skills by imitating animals. *arXiv*, 2020.

Dean A Pomerleau. Alvinn: An autonomous land vehicle in a neural network. In *NeurIPS*, 1989.

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

Mahdi Rad and Vincent Lepetit. Bb8: A scalable, accurate, robust to partial occlusion method for predicting the 3d poses of challenging objects without using depth. *ICCV*, 2017.

Yu Rong, Takaaki Shiratori, and Hanbyul Joo. Frankmocap: Fast monocular 3d hand and body motion capture by regression and integration. *arXiv preprint arXiv:2008.08324*, 2020.

Stéphane Ross and Drew Bagnell. Efficient reductions for imitation learning. In *AISTATS*, 2010.

Stuart Russell. Learning agents for uncertain environments. 1998.

Karl Schmeckpeper, Annie Xie, Oleh Rybkin, Stephen Tian, Kostas Daniilidis, Sergey Levine, and Chelsea Finn. Learning predictive models from observation and interaction. *arXiv*, 2019.

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

Pierre Sermanet, Corey Lynch, Yevgen Chebotar, Jasmine Hsu, Eric Jang, Stefan Schaal, Sergey Levine, and Google Brain. Time-contrastive networks: Self-supervised learning from video. 2018.

Dandan Shan, Jiaqi Geng, Michelle Shu, and David Fouhey. Understanding human hands in contact at internet scale. In *CVPR*, 2020.

Lin Shao, Toki Migimatsu, Qiang Zhang, Karen Yang, and Jeannette Bohg. Concept2robot: Learning manipulation concepts from instructions and human demonstrations. In *RSS*, 2020.

Pratyusha Sharma, Lekha Mohan, Lerrel Pinto, and Abhinav Gupta. Multiple interactions made easy (mime): Large scale demonstrations data for imitation. *arXiv*, 2018.

Pratyusha Sharma, Deepak Pathak, and Abhinav Gupta. Third-person visual imitation learning via decoupled hierarchical controller. In *NIPS*, 2019.

Kenneth Shaw, Shikhar Bahl, and Deepak Pathak. Videodex: Learning dexterity from internet videos. In *6th Annual Conference on Robot Learning*.

Maximilian Sieb, Zhou Xian, Audrey Huang, Oliver Kroemer, and Katerina Fragkiadaki. Graph-structured visual imitation. In *CoRL*, 2020.

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

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

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

Adrian Spurr, Jie Song, Seonwook Park, and Otmar Hilliges. Cross-modal deep variational hand pose estimation. In *CVPR*, 2018.

Zachary Teed and Jia Deng. Raft-3d: Scene flow using rigid-motion embeddings. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8375–8384, 2021.

Faraz Torabi, Garrett Warnell, and Peter Stone. Behavioral cloning from observation. *arXiv*, 2018a.

Faraz Torabi, Garrett Warnell, and Peter Stone. Generative adversarial imitation from observation. *arXiv*, 2018b.

Sundar Vedula, Simon Baker, Peter Rander, Robert Collins, and Takeo Kanade. Three-dimensional scene flow. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, volume 2, pages 722–729. IEEE, 1999.

Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *European conference on computer vision*, pages 20–36. Springer, 2016.

Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. *arXiv*, 2018.

Haoyu Xiong, Quanzhou Li, Yun-Chun Chen, Homanga Bharadhwaj, Samarth Sinha, and Animesh Garg. Learning by watching: Physical imitation of manipulation skills from human videos. *arXiv*, 2021.

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*, pages 3895–3905, 2022.

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

Sarah Young, Dhiraj Gandhi, Shubham Tulsiani, Abhinav Gupta, Pieter Abbeel, and Lerrel Pinto. Visual imitation made easy. *arXiv*, 2020b.

Christian Zimmermann and Thomas Brox. Learning to estimate 3d hand pose from single rgb images. In *CVPR*, 2017.

# 7. Appendix

## 7.1. Experiment details

For the experiments, we consider five different everyday objects shown in Fig. 3, with a total of ten different semantic tasks possible with these objects. The possible tasks are moving veggies on the chopping board, opening the door, closing the door, opening the top drawer, closing the top drawer, opening the bottom drawer, closing the bottom drawer, opening the toaster oven, closing the toaster oven, and moving bowl of fruits across table. The objects and tasks cover different types of motions like sliding with a tool (chopping board), rotation about vertical hinge (door), rotation about horizontal hinge (toaster), and pushing (bowl of fruits). For the chopping board task, we constrain the motion to be within the plane of the table with vertical motion restricted to within 1 cm from the table (so that the spatula doesn't hit the table and fall off the gripper). Results with these different possible behaviors aim to show the generality of the proposed approach in generating both unconditional (when task is not known apriori) and goal-conditioned (when task is specified with goal image) behaviors.

We consider a single IntelRealsene camera in the scene, with RGB image observation that is used as input to our prediction model. For the goal-conditioned setting, the goal image is obtained with the same camera. The camera is fixed and is calibrated. The robot is a 7DOF Franka Emika Panda arm, operated with an IK controller. The base of the Franka is fixed, and its location is known. The End-Effector is a two-finger adaptive Robotiq gripper.

For the baselines, we consider a scene flow Vedula et al. (1999); Teed and Deng (2021) baseline that uses RAFT3D Teed and Deng (2021) for predicting scene flow field between the initial and goal images, and uses the dominant flow direction to guide the motion of the robot. This baseline uses depth image (RGBD from the same camera) to compute scene flow and so requires more information that our method. To test the importance of training across diverse data, we compare against a version of our method that is trained on only 30% of the total training data, with everything else kept the same for training and evaluation.

## 7.2. Additional details on the Approach

The model architectures are described in section 3.1.1. Fig. 2 shows the un-conditioned model. The goal-conditioned model is very similar, with an additional goal image provided as input, with features $f_g$. This is encoded bu the Transformer encoder into $z_g^e$. There is an additional positional embedding dimension in the encoding to disambiguate between the initial image embedding and the goal image embedding. The rest of the architecture is same as that described in section 3.1.1. For the transformer, the embedding dimension is 512, and dropout rate is 0.1 for all blocks. The CVAE networks for encoder and decoder are implemented as 2-layer MLPs. The architecture is auto-regressive such that prior actions are used as input to predict the current action. As is standard in auto-regressive generation, during training, we feed in the previous ground-truth actions at each time-step, and during inference, we feed in the previous predicted actions.

After training the overall hand pose prediction model, $p_\psi(a_{1:T}|o_1)$ with diverse internet videos, we deploy it directly for robot manipulation tasks in the lab. The robot sees an image of the scene through a fixed camera, and optionally receives a goal-image which is input to the prediction model. In order to use the actions predicted by the model $a_{1:T}$ for moving the robot, we need to transform

them to the world coordinate frame of the robot, and considering each action $a_t$ as an end-effector target pose, use a low-level controller for executing the respective motions.

The camera in the scene is calibrated, so the intrinsic matrix $I$ and the extrinsic matrix $[R, T]$ are known. The world coordinates are located at the base of the robot (robot base is at same height as the table top) and the height of the table top from the camera is known and approximately constant. Given scene from the camera $o_1$, the model predicts delta actions $a_{1:T}$ which we convert to absolute actions (described in section 3.1.1), and transform the actions from the camera frame to the world frame of the robot through inverse projection transformation. The prediction horizon is $T = 7$ for our experiments. After obtaining the world coordinates of the action sequence $\{(X_t, Y_t, Z_t, \alpha_t, \beta_t, \gamma_t)\}_{t=1}^{T}$, we use an IK controller to execute the corresponding motion for bringing the end-effector to the desired position and orientation and each time-step. The IK controller has an error threshold of 10% for position and 20% for orientation so that the robot doens't get stuck trying to reach a difficult predicted pose.