Deep Sensorimotor Control by Imitating Predictive Models of Human Motion

Himanshu Gaurav Singh[1], **Pieter Abbeel**[1], **Jitendra Malik**[1], **Antonio Loquercio**[2]
[1]UC Berkeley [2]University of Pennsylvania

Abstract

As the embodiment gap between a robot and a human narrows, new opportunities arise to leverage datasets of humans interacting with their surroundings for robot learning. We propose a novel technique for training sensorimotor policies with reinforcement learning by imitating predictive models of human motions. Our key insight is that the motion of keypoints on human-inspired robot end-effectors closely mirrors the motion of corresponding human body keypoints. This enables us to use a model trained to predict future motion on human data *zero-shot* on robot data. We train sensorimotor policies to track the predictions of such a model, conditioned on a history of past robot states, while optimizing a relatively sparse task reward. This approach entirely bypasses gradient-based kinematic retargeting and adversarial losses, which limit existing methods from fully leveraging the scale and diversity of modern human-scene interaction datasets. Empirically, we find that our approach can work across robots and tasks, outperforming existing baselines by a large margin. In addition, we find that tracking a human motion model can substitute for carefully designed dense rewards and curricula in manipulation tasks. Code, data and qualitative results available at https://jirl-upenn.github.io/track_reward/

1 Introduction

Training robot policies using datasets of humans interacting with their surroundings is a promising approach to scaling robot learning. Indeed, there is no shortage of such datasets, thanks to recent advances in 3D vision (1; 2; 3; 4), industrial augmented and virtual reality devices, *e.g.*, Meta Oculus or Apple Vision Pro, and specifically designed tooling (5; 6). However, leveraging these datasets to train effective sensorimotor robot policies remains an open challenge.

A common strategy to address this challenge is *kinematic retargeting*, which maps human motions to a robot's embodiment. This process is typically performed *independently for each sample* via gradient-based optimization, subject to the robot's kinematic and dynamic constraints. The resulting sensorimotor trajectories can then be used to (pre-)train robot policies through imitation learning (7; 8; 9; 10; 11). However, kinematic retargeting methods often *assume access to a simulated replica of the environment* to evaluate feasibility, e.g., collision checking, which in turn requires accurate 3D scene reconstruction. These requirements are particularly challenging for manipulation tasks, which involve contact-rich interactions and dynamic, non-static scenes. While recent techniques address some of these issues (12; 13), they still need to make scene and/or task-specific approximations to the full robot dynamics. Because of these factors, retargeting methods require significant effort when mapping large human-interaction datasets.

Another approach to training robot policies from datasets of humans interacting with a scene is demonstration-guided reinforcement learning (RL). One line of work (14; 15; 16; 7) uses adversarial rewards to match state distributions between the human and the robot. However, adversarial objectives are unstable to train and are therefore generally employed within small-scale and well-curated

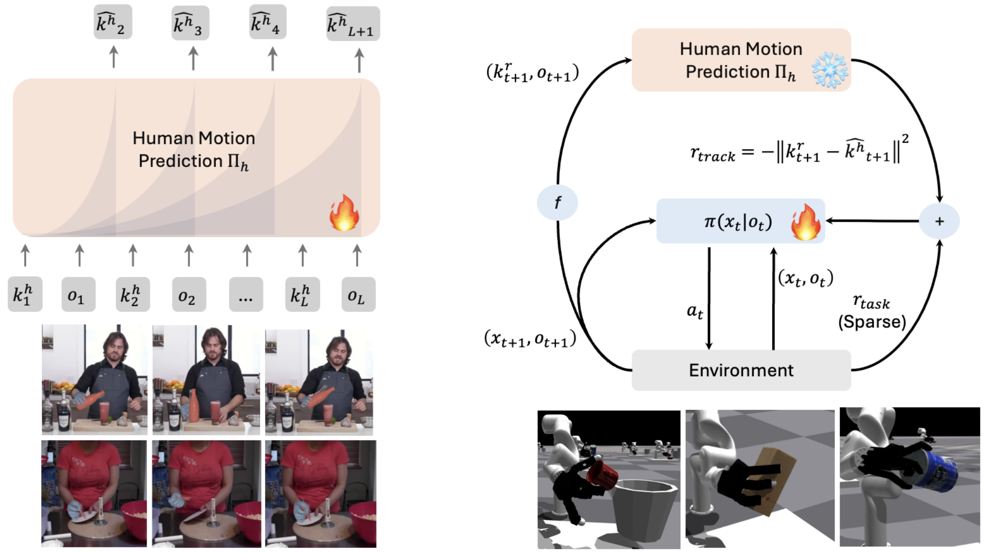

Figure 1: Method. We use a dataset of humans interacting with their scene to train a motion prediction model Π_h. Such model, instantiated as a causal transformer, takes as input a history of previous 3D keypoints $k^h_{t:t-L}$, *i.e.*, the location of the human's fingertips, and observations $o_{t:t-L}$, *i.e.*, the objects' pointcloud, to predict the future human keypoint location $\hat{k}^h_{t+1}$. For anthropomorphic robots, thanks to the abstraction of keypoints, Π_h can be used on robot data despite being trained on human data. Therefore, Π_h can predict likely human motions while training a policy π_θ on a downstream task. A reward to track such motions, r_{track}, provides an additional training signal to the otherwise sparse task reward r_{task}.

datasets. Another line of work combines online policy gradient with imitation losses on offline expert trajectories (17; 18). However, to be used for gradient computation, expert trajectories need to contain robot action annotations, which are absent in datasets of human activities and can only be recovered via retargeting.

In this paper, we introduce a simple yet scalable approach for learning from datasets of humans interacting with their environment. We revisit the core observation of kinematic-retargeting based approaches: modern robotic embodiments are often designed to mimic human form factors (19). As a result, the motion of keypoints on the end-effectors of human-inspired robots closely resembles the motion of keypoints on the human body (20), despite fundamental differences in their action spaces. We use this observation for a novel insight: a predictive model trained on human keypoints can be applied *zero-shot* to robot data. First, we train a predictive model, Π_h, to estimate the future locations of keypoints on the human body based on a history of previous scene observations and keypoint locations. Then, we use RL in simulation to maximize a task reward while tracking the predicted future human keypoints. These predictions are obtained by feeding a history of the robot's observations and keypoint locations into Π_h. Figure 1 provides an overview of our approach.

In contrast to the conventional paradigm, our approach offers several key advantages: (i) It removes the need to replicate the environment in which humans interact within a simulator. Indeed, generalization of the predictive model to the scene in simulation removes the need for accurate *real-to-sim*. (ii) It decouples human data from the robot policy learning loop, meaning the potentially large human dataset is not required during policy training; (iii) It enables the automatic selection of the most suitable skill within the human dataset's simplex to solve the task at hand, eliminating the need for manually defined selection procedures (21; 22).

Similar to previous approaches, our method requires a mapping between human and robot keypoints (see Figure 2). However, we find that this mapping is straightforward to design and remains effective across different tasks and embodiments.

Our experiments demonstrate that incorporating a reward for tracking predictive models of human motion enables robots to learn dexterous manipulation tasks from relatively sparse task rewards, eliminating the need for carefully engineered reward functions. Moreover, we show that the motion predictor Π_h can be applied *zero-shot* across different tasks and embodiments, allowing our approach to outperform existing baselines.

Overall, our approach serves as a stepping stone toward fully leveraging the richness of existing datasets of human activities for sensorimotor policy learning.

2 Related Work

As the embodiment gap between a robot and a human closes, the amount of information that can be learned via observation increases (20), as well as the sources of data robots can learn from (23). In the next paragraphs, we will provide an overview of different ways to learn from human data and map human motion to robot motion using kinematic constraints and/or adversarial losses.

2.1 Cross-Embodiment Transfer

A natural approach to leveraging human data for robot training is to find an abstraction of human motion that eliminates embodiment-specific factors. For robots with an anthropomorphic hand, one such abstraction is the 3D location of the hand's wrist and fingertips (24; 25; 22; 7; 8; 10). Selecting fewer keypoints, *e.g.*only the wrist location, enables transfer between different kinds of end-effectors. However, this often requires additional demonstrations for the transfer between human and robot motion to be effective (26; 27; 28).

Instead of focusing on end-effectors, an alternative approach to map motion between embodiments is to focus on the effects of actions—such as the objects being manipulated (29). This perspective allows for transfer between vastly different embodiments. Indeed, the only requirement is that the outcomes of the actions remain consistent.

A common and data-efficient approach to achieving this is through affordances, which indicate where an object can be interacted with (30; 31; 32). However, while affordance-based methods are data-efficient, they need to handle pre- and post-contact trajectories separately with specifically designed modules. One way to address this limitation is by modeling the object's 3D (33) or 2D motion (34; 35).

Incorporating further contextual cues can add constraints to improve the transfer between embodiment. This can be done by modeling the two-dimensional motion of the entire scene (36; 37; 38). Importantly, mapping human motion and object motion are not mutually exclusive; they can be combined to maintain data efficiency while improving effectiveness for non-anthropomorphic robots (21; 39; 40).

The mapping between keypoints on the human's and robot's end effectors does not necessarily need to be manually defined, but could be learned from data. A common approach to do so is by transforming the entire human demonstration to a robot demonstration via a generative model and map the 2D motion of the rendered robot to actions via reinforcement learning (41; 42). Instead of rendering the video with a different embodiment, another possibility is generating either the full video (43) or a segmented version of it (44) from only the initial observation. Such generated video can then be translated into robot actions by collecting a set of paired demonstrations.

Similarly to our work, these approaches leverage the idea that the data of humans interacting with their surroundings can be translated into sensorimotor robot trajectories. However, they make the additional assumption that the scene between the human and the robot matches. In addition, the mapping is usually learned from data and does not account for the robot kinematic and dynamic constraints. In the next section, we will cover a family of methods that explicitly account for such constraints during motion mapping.

2.2 Kinematic Retargeting

The mapping of human motion to robot motion and actions has traditionally been cast as a constraint optimization problem. Such optimization, however, is challenging when the human interacts with the scene, *e.g.*, picking up an object, as contact dynamics is non-differentiable. Finding effective solutions to this problem has long been studied (45; 46), but it is still an active area of research (13; 12).

Retargeting noisy data acquired from videos adds another layer of complexity, deriving from the noise in humans and objects' pose estimation. One approach to do so is disregarding the robot's and/or objects' dynamics and recovering successful trajectories by sampling (25; 10; 47) or reinforcement learning (7; 48; 49). Other works include the dynamics directly in the optimization (8), which, however, is only feasible when a fast dynamics simulator is available.

Such optimization becomes more challenging when mapping not only end-effectors, *e.g.*, hands, but the whole body (50; 51), or the interactions between different bodies (52). One approach to finding solutions is imposing a reduced structure to the problem, which makes it amenable to neural network optimization (53). However, by simplifying part of the problem, *e.g.*, ignoring the whole-body kinematics and the scene's dynamics, enables retargeting to be used for real-world whole-body control. Such simplifications can be accounted for either with reinforcement learning (54) or via a human teleoperator (55; 56).

A limitation of constrained-based optimization approaches is that motion is mapped from the human to the robot *per sample*. To be effective in cases where there is interaction with a scene, a replica of the environment the human is interacting with is required to constrain the optimization. Our approach proposes a novel way to address this limitation by decoupling human data from robot policy learning.

2.3 Adversarial Losses for Motion Matching

Another way to match the distribution of human and robot states and observations *in expectation* is via adversarial losses (57). This approach has been very successful for character animation (58; 15; 59), but has also been applied to robotics for manipulation (24) and locomotion (14; 16). However, a downside of adversarial losses is that they are challenging to optimize on large datasets, which makes this line of work difficult to scale. Combining adversarial losses with a predictive model of human motion via control hierarchies can address this limitation (60). However, while this was successful for character animation, it still presents challenges when applied in robotics, as downstream controllers might not be available in the first place for the task at hand.

3 Preliminaries

We adopt the standard approach of defining a sensorimotor control task as a discrete-time, finite-horizon, discounted Markov decision process (MDP), represented by the tuple $M = (\mathbb{S}, \mathbb{A}, \mathbb{P}, r, \rho_0, \gamma, T)$. Here, $\mathbb{S}$ is the state space, $\mathbb{A}$ is the action space, $\mathbb{P} : \mathbb{S} \times \mathbb{A} \times \mathbb{S} \rightarrow \mathbb{R}$ is the transition probability distribution, $r : \mathbb{S} \times \mathbb{A} \rightarrow \mathbb{R}$ is the reward function, $\rho_0 : \mathbb{S} \rightarrow \mathbb{R}$ is the initial state distribution, γ is the discount factor, and T is the time horizon.

The objective is to optimize a stochastic policy $\pi_\theta : \mathbb{S} \times \mathbb{A} \rightarrow \mathbb{A}$, parameterized by θ, by maximizing its discounted expected return $R(\pi_\theta) = \mathbb{E}_\tau \left[\sum_{t=0}^{T} \gamma^t r(s_t, a_t) \right]$, where $\tau = (s_0, a_0, \ldots)$ denotes the trajectory of states, actions, and goals encountered during an episode. Specifically, $s_0 \sim \rho_0$ is the initial state sampled from the state distribution, $a_t \sim \pi_\theta(a_t | s_t)$ is the action sampled from the policy, and $s_{t+1} \sim \mathbb{P}(s_{t+1} | s_t, a_t)$ is the next state sampled from the transition distribution.

4 Method

Tasks with a high-dimensional state and action space often require the reward to be carefully crafted to achieve the desired behavior. Motivated by the challenges of designing rewards for complex tasks, we propose a simple approach to bias the agent towards behaviors likely to be performed by humans. Specifically, we shape a sparse task reward r_{task} by incorporating an additive term r_{track}, which incentivizes the robot to follow the predicted motion of a human while doing the same task.

To derive r_{track}, we assume the existence of an abstraction level at which the difference between human and robot motion is minimal. For anthropomorphic robots and manipulation tasks, such abstraction is easy to define (Fig. 2). Following prior work on kinematic retargeting (20; 7; 11; 10; 8), we define this abstraction as the 3D positions of fingertips.

The reward r_{track} measures the distance between the robot and 3D human keypoints, *i.e.*,

$$r_{track} = -\|k^h_{t+1} - f(s_{t+1})\|^2 = -\|k^h_{t+1} - k^r_{t+1}\|^2, \tag{1}$$

where $k^h_{t+1} \in \mathbb{B}$ are the keypoints 3D location on the human hand, and $f : \mathbb{S} \mapsto \mathbb{B}$ is the function to compute the robot's keypoints location k^r_{t+1} from its state, *i.e.*, forward kinematics.

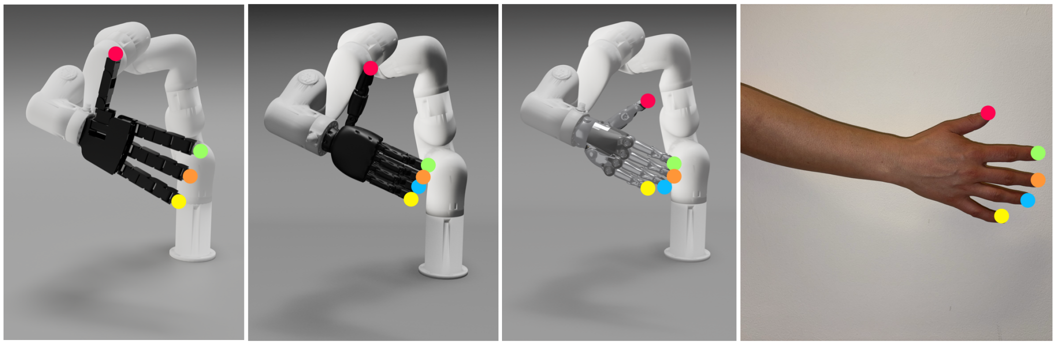

Figure 2: Mapping between human and robot hands Anthropomorphic hands allow an intuitive mapping of robot links to human hand keypoints. It is defined once and remains consistent across tasks. Here, we show the mapping of the human hand to three different morphologies: an Allegro hand (61) (left), an Xhand (62) (center), and an SVH hand (63)(right).

Conceptually, Eq. 1 is straightforward and has been explored in prior works that learn to translate human behavior into that of robot *per-demonstration* (48; 14; 22). However, its application requires human data in an environment and task which *closely matches* that of the robot. This limiting assumption can hinder its application, especially in contact-rich tasks where a suitably accurate re-construction of the human environment is hard.

To address this limitation, we propose to compute an estimate of k^h_{t+1} by training a model of human motion Π_h to predict the most likely future keypoints location given a history of previous keypoints and observations. The key idea of our approach is that, thanks to the abstraction of keypoints, such a model can be trained on human data but evaluated on robot data. Therefore, we can modify Eq. 1 to let the policy π_θ track not the observed human keypoints k^h_{t+1} but the *predicted* keypoints $\hat{k}^h_{t+1}$, generated by passing the robot's state history as an input to Π_h. Formally,

$$r_{track} = -\|\hat{k}^h_{t+1} - k^r_{t+1}\|^2, \tag{2}$$

where $\hat{k}^h_{t+1} = \Pi(f(s_t), \dots, f(s_{t-L}))$, with L being the model's time horizon.

4.1 Human Motion Predictor

Before introducing the details of our approach, we clarify the meaning of state for our setting. We divide the robot's state into two components, *i.e.*, $s_t = [x_t, o_t]$, where $x_t \in \mathbb{R}^d$ is the robot's proprioceptive state and $o_t \in \mathbb{R}^{100 * n_o \times 3}$ is a point cloud of the objects in the scene, n_o being the number of objects. For experiments, d ranges from 20-30 and n_o from 1 to 3. The human keypoints $k^h_t \in \mathbb{R}^{3 \times d}$ are the 3D positions of the d hand and keypoints in the world frame, with d varying according to the robot end-effector morphology.

To train the human motion predictor Π_h, we assume access to a dataset $\mathcal{D} := \{g^{(i)}, (k^{h(i)}_1, o^{(i)}_1, k^{h(i)}_2, k^{h(i)}_2, \dots)\}^n_{i=1}$, which consists of trajectories of human keypoints k^h_t and objects' pointcloud o_t, as well as a goal label g. This dataset contains instances of humans interacting with their surroundings to accomplish the task g.

We instantiate Π_h as a vanilla transformer with causal attention. The transformer has six layers and eight heads. We tokenize keypoints k^h_t with a linear transformation to a 512-dimensional hidden size. The pointcloud o_t is tokenized to the same hidden size by a PointNet encoder (64). We use a context length of $L = 16$, which we find to be sufficient for our setting.

We train Π_h on $\mathcal{D}$ with the mean squared error loss

$$\mathcal{L} = \mathbb{E}_{\tau\sim\mathcal{D},\, g,o,k^h\sim\tau}\left[\|\Pi_h(g, o_{t:t-K}, k^h_{t:t-K}) - k^h_{t+1})\|^2\right].$$

We also experimented with incorporating an additional loss on the observation tokens but found that it provided only marginal benefits. To maintain simplicity, we excluded it from our final model.

To efficiently train Π_h, we employ teacher forcing. However, we find that its naïve application leads to poor performance on closed-loop robot trajectories, since autoregressive inference drifts out of the training distribution over time. Empirically, we observe that this issue can be mitigated by injecting zero-mean Gaussian noise into the input keypoints k_t^h. Importantly, the noise is added only to the input keypoints, and not the targets, ensuring that the model's predictions remain precise.

4.2 Policy Optimization

We train specialized robot policies, π_θ, using model-free reinforcement learning on a set of downstream tasks. For simplicity and efficiency, we implement π_θ as a multi-layer perceptron (MLP) with four layers and a hidden dimension of 128, which is randomly initialized before training. The observation space consists of a point cloud representation of the scene, which is preprocessed by a PointNet encoder, as well as the robot's proprioceptive state.

We use PPO (65) as our reinforcement learning algorithm. This choice is motivated by the observation that PPO is the standard choice for a wide variety of robotics applications (66; 67; 54; 68; 69). We employ an asymmetric actor-critic architecture for optimization. Specifically, we give as extra information to the value function the distance between the fingers and the target's center of mass.

5 Experimental Setup

Our experiments are conducted in the IsaacGym simulator (70) with a PhysX backend, enabling fast, parallel rigid-body physics simulation. Our experiments span three manipulation tasks and three different robot platforms.

We adapt tasks from the IsaacGymEnvs (71) benchmark suite to evaluate our approach. Specifically, we test on the following tasks:

- **Grasp and Lift** The robot must grasp an object and hold it at a specified height.
- **Grasp and Lift - Clutter** The goal robot must grasp the specified object amongts multiple distractor objects.
- **Lift and Throw** The agent must first lift the object and then accurately drop it into an adjacent receptacle.
- **Cabinet** This is a dexterous manipulation task in which a robot has to interact with an articulated object. Specifically, the agent has to open the drawer of a cabinet by grasping the handle and pulling it horizontally.

More details about task setup are available in the supplementary. We test our approach on three robotic platforms: (i) a 4-fingered Allegro Hand (61) with 16 degrees of freedom (DoF), (ii) a 5-fingered Xhand (72) with 12 DoF, and (iii) a 5-fingered SVH Hand (63) with 20 DoF. All hands are mounted on a 7-DoF Xarm7 robotic arm.

Note that we use the same Π_h *for all robots and tasks*. Note that we condition Π_h to a categorical label which identifies the object of interest. The mapping between the human and robot hand's keyponts is also quite straightforward and fixed across tasks and embodiments. It corresponds to a fingertip-to-fingertip matching. A visualization of the robot and mapping is available in Figure 2.

We use the DexYCB dataset (73) to train our predictive human motion model. It comprises 1,000 trajectories of human-object interactions in cluttered environments containing up to five objects, with a total of 20 distinct object categories.

6 Results

We design an experimental procedure to answer the following questions: i) Can our approach enable policy learning with relative sparse task rewards? How does it compare to carefully shaped reward functions? ii) Can the same human prediction model Π_h be used across robot embodiments and across tasks? iii) How does our method compare to other demonstration-guided RL approaches using human data (possibly after kinematic retargeting)?

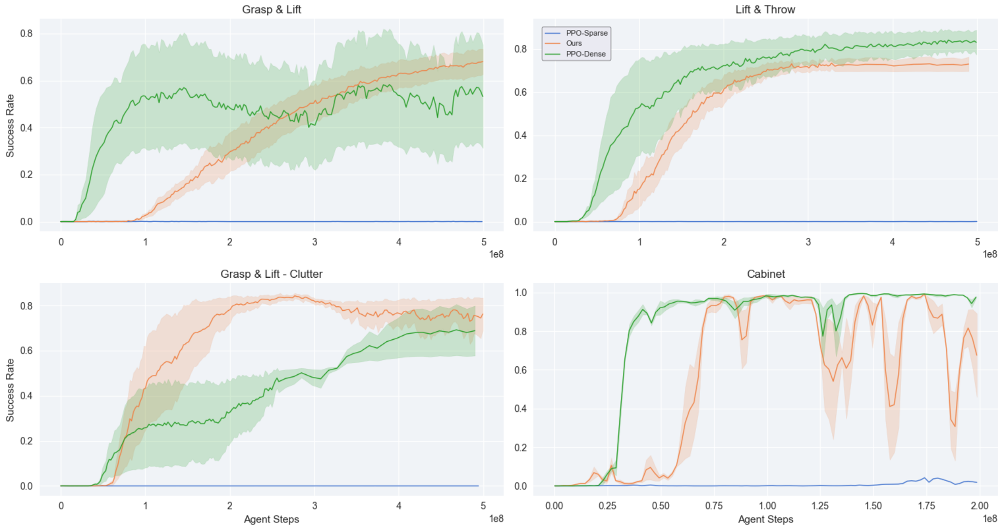

Figure 3: Comparison across tasks Tracking the predictions of Π_h enables tractable reinforcement learning with a sparse task reward. Our approach is comparable to the privileged baseline: *PPO with dense rewards* across all tasks, whereas PPO with the sparse task reward only fails to learn. Runs are averaged across 3 seeds.

6.1 Can tracking Π_h make up for carefully engineered rewards?

Reward design for robot control involves careful tuning *per-task*. Especially for object manipulation, carefully shaped dense rewards often contain a state machine to switch between different behaviors (71). Such task-specific, hand-designed procedure presents a primary bottleneck to scaling RL methods. We now investigate whether the tracking reward r_{track} can substitute for manual reward shaping. To do so, we compare with the following approaches:

1. **PPO-DenseReward**: We utilize the manually engineered dense reward functions provided in IsaacGymEnvs (70) for *Grasp and Lift*, *Lift and Throw*, and *Open Cabinet* tasks. This reward function is a weighted sum of different components. The different reward terms are designed to: guide the hand toward the object, provide a bonus for lifting it a few centimeters, encourage movement toward the goal location after lifting, and provide a success bonus. Note that this reward function is still used in state-of-the-art reinforcement learning works (74). We use this dense reward as a privileged baseline.
2. **PPO-TaskReward**: This baseline uses PPO training solely with a sparse goal reward (r_{rask}). For *Grasp and Lift* and *Lift and Throw* tasks, this reward is:

$$r_{rask} = (obj_z > 0.1) * \text{ReLU}\big(d_t(\text{goal}, \text{obj}) - d_{t-1}(\text{goal}, \text{obj})\big)$$

where $d_t(goal, obj)$ is the distance of the object from the goal at time t, and obj_z is the height of the object from the table. A comparison of this reward to the dense reward is shown in Fig. 4. For the *Open Cabinet* task, the reward is

$$r_{rask} = drawer_x * (1 + c * ReLU(drawer_x >= drawer_{limit}))$$

where $drawer_x$ is the distance to which the drawer body is pulled out and $drawer_{limit}$ is the distance to be pulled for the drawer to be opened.
3. **Ours**: Our method combines trajectory tracking and sparse goal rewards, *i.e.*, using $r_{track} + \lambda * r_{task}$ as the reward function, where λ is a hyperparameter fixed across tasks.

All baselines have an additional reward term penalizing the energy consumed by moving, which we approximate as the absolute sum of all joint velocities. We control for all additional factors, including the model architecture, agent steps, and domain randomization, but tune each baseline independently.

Figure 3 illustrates the result of this experiment. We find that our approach achieves comparable performance with respect to a highly engineered reward function. Conversely, PPO fails to find good solutions when trained exclusively on a sparse task reward.

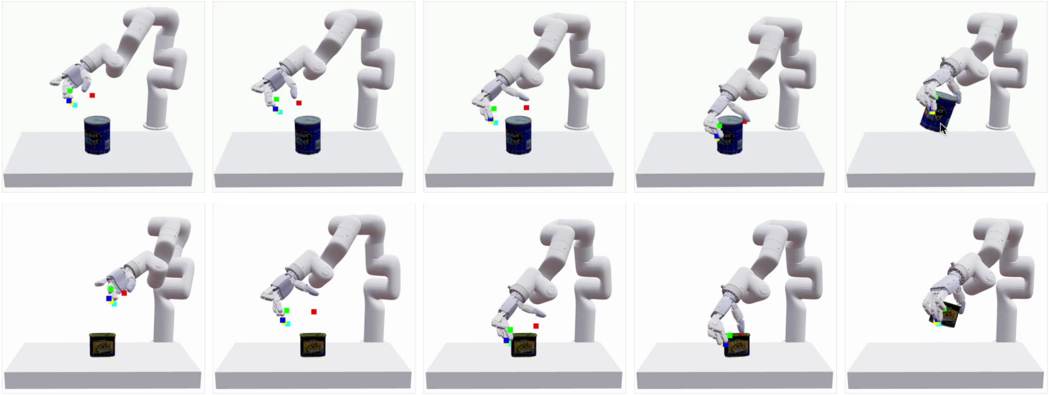

Figure 5: Qualitative Results. A policy trained with our approach successfully picking up two objects. The colored points show the next predictions of Π_h. Such predictions are temporally smooth and guide the policy towards fast grasping and lifting. Better seen at our webpage https://jirl-upenn.github.io/track_reward/.

A more detailed breakdown of task-specific performance reveals several notable observations. First, in the *Grasp and Lift* task, our approach and PPO-Dense achieve similar performance at convergence, though PPO-Dense reaches higher success rates slightly faster. Second, despite the absence of demonstrations for the *Lift and Throw* task in the human dataset, our approach still attains a high success rate, though it performs slightly worse than PPO-Dense. In the *Grasp and Lift-Clutter* task, both methods converge to comparable performance. However, our approach solves the task more quickly, as PPO-Dense requires additional steps to distinguish the target object from distractors—knowledge that our approach automatically inherits from Π_h. For the *Open Cabinet* task too, our approach performs comparably to PPO-Dense.

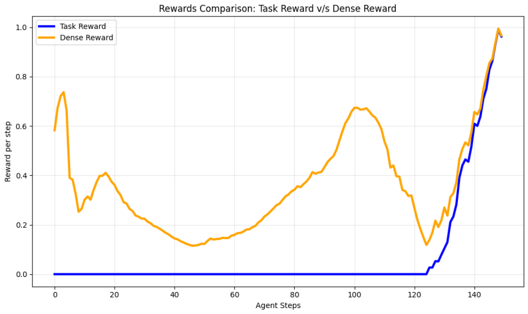

Figure 4: Comparison of sparse and dense step-wise rewards for the *Grasp and Lift task* over expert policy rollouts. The sparse reward activates only after the object is grasped and moves toward the target (≈step 130), while the dense reward provides continuous shaping throughout.

Fig. 5 shows qualitative results of our approach. It also illustrates closed-loop predictions from Π_h based on the trajectory of robot keypoints. These predictions are key to training a policy which is smooth and effective. We refer the reader to the supplementary videos for clearer visualization of the policy behavior.

6.2 Performance across robot embodiments

In this section, we examine whether a human motion predictor can be used to provide tracking rewards across different robot embodiments. To explore this, we fix the task to *Grasp and Lift* while varying the embodiment.

The results of this experiment are shown in Figure 6. Note that we make slight changes to the PPO hyperparameters per embodiment. Specifically, we decrease the maximum learning rate in the adaptive scheduler of PPO as the DOFs in the embodiment increase. This is required because the KL-divergence between policy updates is proportional to the dimension of the action space. We attempt to do the same tuning process for the PPO-TaskReward, but we find it to be unable to learn. In contrast, our approach consistently achieves high success rates, with performance remaining stable across different embodiments.

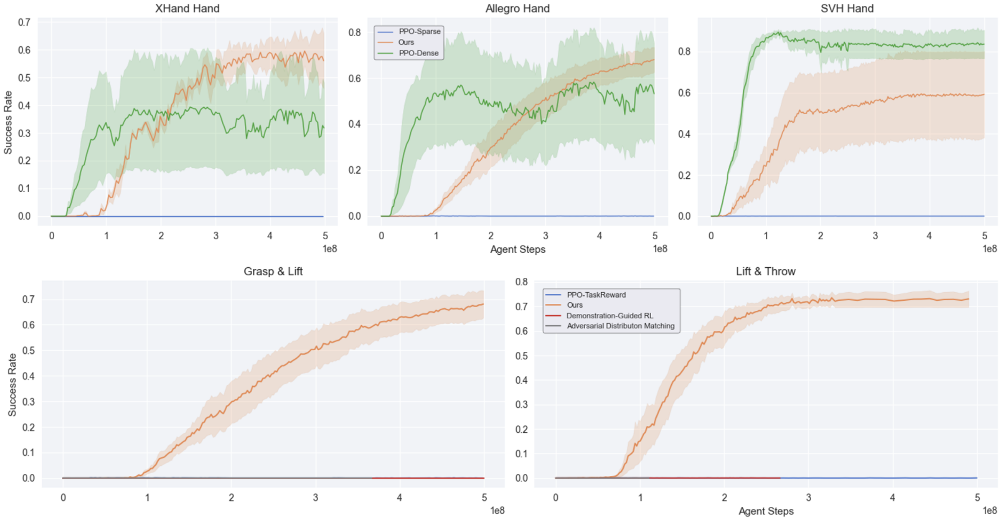

Figure 6: (top) We evaluate our approach on a diverse set of multi-fingered hand embodiments. Our approach performs comparably to the privileged baseline:*PPO with dense rewards* across three such robots. (bottom) Our approach outperforms existing methods that use human data to guide RL exploration, when both are trained with sparse task reward.

6.3 Comparison to baselines

There exists a large body of work that studied the problem of using human data for training robot policies (Sec. 2). From this vast literature, we select a few baselines that are compatible with our problem formulation and setup. Specifically, we focus on approaches that have the following characteristics: (i) They do not divide the task into subroutines, *e.g.*, affordance detection, pre-grasp, and post-grasp motion planning; (ii) they do not require retrieving a specific example from the human dataset and mapping that to robot motion; (iii) they do not require robot demonstrations on the downstream task; (iv) they produce a policy that can be run in isolation from any other training component. Such a filtering process is designed to make the comparison to our approach as fair as possible.

As a result of this filtering process, we compare our approach against the following baselines:

- **Adversarial Distribution Matching.** This baseline trains a policy jointly with a discriminator that distinguishes between trajectory rollouts generated by the policy and those from the offline dataset. An auxiliary reward is added to the sparse task reward r_{task}, encouraging the policy to *fool* the discriminator. In our setting, since the dataset consists of human keypoint trajectories, the discriminator is trained to differentiate between mapped robot keypoint trajectories and human keypoint trajectories. This baseline aligns with prior work on adversarial state distribution matching (57; 15; 7; 16; 60).

- **Retargeting, followed by demonstration-guided RL** This baseline first applies kinematic retargeting to map the human dataset into a set of sensorimotor robot trajectories. A policy is then trained by jointly optimizing r_{task} on on-policy experiences and a supervised learning objective on the retargeted dataset, achieved by assigning high advantage values to these trajectories. This approach represents prior work on combining on-policy learning with off-policy trajectories derived from retargeting of human demonstrations (48; 49; 7; 47; 10; 8).

As in all our previous experiments, we tune each baseline individually. Figure 6 illustrates our results. We find that the baselines fail to leverage the dataset to sufficiently bias the RL exploration required for learning from sparse rewards. While we found this result to be surprising, we would like to stress that the baselines are generally trained on downstream tasks with dense rewards and not previously tested in our setting. Indeed, when we experimented with providing them with dense rewards, we found them to achieve high success rates (See Supplementary 10.2).

7 Conclusions

In this work, we studied how datasets of humans interacting with their environment can be leveraged to train robot policies. While the availability of such datasets continues to grow due to technological advancements, it remains unclear how to effectively extract and utilize this information for policy learning. We demonstrated that a simple modification to a well-established procedure can yield significant benefits. Our approach showed promising results in dexterous manipulation across various robots and tasks. Notably, it enabled successful reinforcement learning in tasks that previously required extensive reward shaping to achieve comparable performance.

Limitations. Our approach shares one key assumption of kinematic retargeting methods, *i.e.*, that a set of keypoints on the human body and on a robot follow similar trajectories, despite significant differences in their action spaces. This limits our work to anthropomorphic robots. While prior work has shown that clever mapping schemes can enable transfer even to robots with significantly different morphologies (27; 26; 42), they still require task-specific robot demonstrations to bridge the embodiment gap. Another limitation of this work is the lack of real-world experiments. While important, our core contribution is orthogonal to sim-to-real transfer, which typically relies on techniques such as domain randomization and sensor calibration. One hypothesis is that mimicking human motions could potentially easen the engineering load of sim-to-real transfer, since the policy inherently learns behaviours feasible in the real world. We believe this to be an interesting line of inquiry for future work.

8 Broader Impact

This paper presents work that aims to advance the fields of Machine Learning and Robotics. There are many potential societal consequences of our work, none of which we feel must be specifically highlighted here.

9 Acknowledgements

This work was supported by the DARPA Machine Common Sense program, the DARPA Transfer from Imprecise and Abstract Models to Autonomous Technologies (TIAMAT) program, the ONR MURI award N00014-21-1-2801 and the InnoHK Clusters of the Hong Kong SAR Government via the Hong Kong Centre for Logistics Robotics. This work was also funded by ONR MURI N00014-22-1-2773. We thank Arthur Allshire, Ankur Handa, Jathushan Rajasegaran and Tara Sadjadpour for helpful discussions. We thank Lorenzo Bianchi and Noemi Aepli for their help with illustrations. Pieter Abbeel holds concurrent appointments as a Professor at UC Berkeley and as an Amazon Scholar. This paper describes work performed at UC Berkeley and is not associated with Amazon.

References

[1] G. Pavlakos, D. Shan, I. Radosavovic, A. Kanazawa, D. Fouhey, and J. Malik, “Reconstructing hands in 3D with transformers,” in *CVPR*, 2024.

[2] S. Goel, G. Pavlakos, J. Rajasegaran, A. Kanazawa, and J. Malik, “Humans in 4d: Reconstructing and tracking humans with transformers,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 14 783–14 794.

[3] G. Wu, T. Yi, J. Fang, L. Xie, X. Zhang, W. Wei, W. Liu, Q. Tian, and X. Wang, “4d gaussian splatting for real-time dynamic scene rendering,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 20 310–20 320.

[4] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis, “3d gaussian splatting for real-time radiance field rendering.” *ACM Trans. Graph.*, vol. 42, no. 4, pp. 139–1, 2023.

[5] V. Guzov, Y. Jiang, F. Hong, G. Pons-Moll, R. Newcombe, C. K. Liu, Y. Ye, and L. Ma, “Hmd2: Environment-aware motion generation from single egocentric head-mounted device,” *arXiv preprint arXiv:2409.13426*, 2024.

[6] L. Ma, Y. Ye, F. Hong, V. Guzov, Y. Jiang, R. Postyeni, L. Pesqueira, A. Gamino, V. Baiyya, H. J. Kim *et al.*, "Nymeria: A massive collection of multimodal egocentric daily motion in the wild," in *European Conference on Computer Vision*. Springer, 2024, pp. 445–465.

[7] Y. Qin, Y.-H. Wu, S. Liu, H. Jiang, R. Yang, Y. Fu, and X. Wang, "Dexmv: Imitation learning for dexterous manipulation from human videos," in *ECCV*, 2022.

[8] H. G. Singh, A. Loquercio, C. Sferrazza, J. Wu, H. Qi, P. Abbeel, and J. Malik, "Hand-object interaction pretraining from videos," *arXiv preprint arXiv:2409.08273*, 2024.

[9] I. Radosavovic, B. Zhang, B. Shi, J. Rajasegaran, S. Kamat, T. Darrell, K. Sreenath, and J. Malik, "Humanoid locomotion as next token prediction," *arXiv:2402.19469*, 2024.

[10] J. Ye, J. Wang, B. Huang, Y. Qin, and X. Wang, "Learning continuous grasping function with a dexterous hand from human demonstrations," *RA-L*, 2023.

[11] K. Shaw, S. Bahl, and D. Pathak, "Videodex: Learning dexterity from internet videos," in *CoRL*, 2022.

[12] A. S. Lakshmipathy, N. Feng, Y. X. Lee, M. Mahler, and N. Pollard, "Contact edit: Artist tools for intuitive modeling of hand-object interactions," *Transactions on Graphics*, 2023.

[13] A. S. Lakshmipathy, J. K. Hodgins, and N. S. Pollard, "Kinematic motion retargeting for contact-rich anthropomorphic manipulations," *arXiv:2402.04820*, 2024.

[14] X. B. Peng, E. Coumans, T. Zhang, T.-W. Lee, J. Tan, and S. Levine, "Learning agile robotic locomotion skills by imitating animals," *arXiv preprint arXiv:2004.00784*, 2020.

[15] X. B. Peng, Z. Ma, P. Abbeel, S. Levine, and A. Kanazawa, "Amp: Adversarial motion priors for stylized physics-based character control," *Transactions on Graphics*, 2021.

[16] A. Escontrela, X. B. Peng, W. Yu, T. Zhang, A. Iscen, K. Goldberg, and P. Abbeel, "Adversarial motion priors make good substitutes for complex reward functions," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 25–32.

[17] A. Rajeswaran, V. Kumar, A. Gupta, G. Vezzani, J. Schulman, E. Todorov, and S. Levine, "Learning complex dexterous manipulation with deep reinforcement learning and demonstrations," *arXiv preprint arXiv:1709.10087*, 2017.

[18] P. J. Ball, L. Smith, I. Kostrikov, and S. Levine, "Efficient online reinforcement learning with offline data," in *International Conference on Machine Learning*. PMLR, 2023, pp. 1577–1594.

[19] M. T. Ciocarlie and P. K. Allen, "Hand posture subspaces for dexterous robotic grasping," *IJRR*, 2009.

[20] M. Ciocarlie, C. Goldfeder, and P. Allen, "Dimensionality reduction for hand-independent dexterous robotic grasping," in *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2007, pp. 3270–3275.

[21] G. Papagiannis, N. Di Palo, P. Vitiello, and E. Johns, "R+ x: Retrieval and execution from everyday human videos," *arXiv preprint arXiv:2407.12957*, 2024.

[22] P. Mandikal and K. Grauman, "Dexvip: Learning dexterous grasping with human hand pose priors from video," in *Conference on Robot Learning*, 2022, pp. 651–661.

[23] J. Romero, "From human to robot grasping," Ph.D. dissertation, KTH Royal Institute of Technology, 2011.

[24] G. Garcia-Hernando, E. Johns, and T.-K. Kim, "Physics-based dexterous manipulations with estimated hand poses and residual reinforcement learning," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 9561–9568.

[25] D. Antotsiou, G. Garcia-Hernando, and T.-K. Kim, "Task-oriented hand motion retargeting for dexterous manipulation imitation," in *Proceedings of the European conference on computer vision (ECCV) workshops*, 2018, pp. 0–0.

[26] S. Kareer, D. Patel, R. Punamiya, P. Mathur, S. Cheng, C. Wang, J. Hoffman, and D. Xu, "Egomimic: Scaling imitation learning via egocentric video," *arXiv preprint arXiv:2410.24221*, 2024.

[27] J. Ren, P. Sundaresan, D. Sadigh, S. Choudhury, and J. Bohg, "Motion tracks: A unified representation for human-robot transfer in few-shot imitation learning," *arXiv preprint arXiv:2501.06994*, 2025.

[28] C. Wang, L. Fan, J. Sun, R. Zhang, L. Fei-Fei, D. Xu, Y. Zhu, and A. Anandkumar, "Mimicplay: Long-horizon imitation learning by watching human play," *arXiv preprint arXiv:2302.12422*, 2023.

[29] Y. Chen, C. Wang, Y. Yang, and C. K. Liu, "Object-centric dexterous manipulation from human motion data," *arXiv preprint arXiv:2411.04005*, 2024.

[30] S. Bahl, R. Mendonca, L. Chen, U. Jain, and D. Pathak, "Affordances from human videos as a versatile representation for robotics," in *CVPR*, 2023.

[31] P. Mandikal and K. Grauman, "Dexterous robotic grasping with object-centric visual affordances," *arXiv preprint arXiv:2009.01439*, vol. 1, no. 2, p. 4, 2020.

[32] Y.-H. Wu, J. Wang, and X. Wang, "Learning generalizable dexterous manipulation from human grasp affordance," in *Conference on Robot Learning*. PMLR, 2023, pp. 618–629.

[33] M. Kokic, D. Kragic, and J. Bohg, "Learning task-oriented grasping from human activity datasets," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3352–3359, 2020.

[34] C. Yuan, C. Wen, T. Zhang, and Y. Gao, "General flow as foundation affordance for scalable robot learning," *arXiv preprint arXiv:2401.11439*, 2024.

[35] M. Xu, Z. Xu, Y. Xu, C. Chi, G. Wetzstein, M. Veloso, and S. Song, "Flow as the cross-domain manipulation interface," *arXiv preprint arXiv:2407.15208*, 2024.

[36] Y. Liu, A. Gupta, P. Abbeel, and S. Levine, "Imitation from observation: Learning to imitate behaviors from raw video via context translation," in *2018 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2018, pp. 1118–1125.

[37] S. Bahl, A. Gupta, and D. Pathak, "Human-to-robot imitation in the wild," in *RSS*, 2022.

[38] H. Bharadhwaj, R. Mottaghi, A. Gupta, and S. Tulsiani, "Track2act: Predicting point tracks from internet videos enables generalizable robot manipulation," *arXiv preprint arXiv:2405.01527*, 2024.

[39] N. Heppert, M. Argus, T. Welschehold, T. Brox, and A. Valada, "Ditto: Demonstration imitation by trajectory transformation," *arXiv preprint arXiv:2403.15203*, 2024.

[40] C. Wen, X. Lin, J. So, K. Chen, Q. Dou, Y. Gao, and P. Abbeel, "Any-point trajectory modeling for policy learning," in *RSS*, 2023.

[41] L. Smith, N. Dhawan, M. Zhang, P. Abbeel, and S. Levine, "Avid: Learning multi-stage tasks via pixel-level translation of human videos," in *RSS*, 2020.

[42] H. Xiong, Q. Li, Y.-C. Chen, H. Bharadhwaj, S. Sinha, and A. Garg, "Learning by watching: Physical imitation of manipulation skills from human videos," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 7827–7834.

[43] H. Bharadhwaj, D. Dwibedi, A. Gupta, S. Tulsiani, C. Doersch, T. Xiao, D. Shah, F. Xia, D. Sadigh, and S. Kirmani, "Gen2act: Human video generation in novel scenarios enables generalizable robot manipulation," *arXiv preprint arXiv:2409.16283*, 2024.

[44] H. Bharadhwaj, A. Gupta, V. Kumar, and S. Tulsiani, "Towards generalizable zero-shot manipulation via translating human interaction plans," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 6904–6911.

[45] C. K. Liu, "Dextrous manipulation from a grasping pose," in *ACM SIGGRAPH 2009 papers*, 2009, pp. 1–6.

[46] Y. Ye and C. K. Liu, "Synthesis of detailed hand manipulations using contact sampling," *ACM Transactions on Graphics (ToG)*, vol. 31, no. 4, pp. 1–10, 2012.

[47] Z. Q. Chen, K. Van Wyk, Y.-W. Chao, W. Yang, A. Mousavian, A. Gupta, and D. Fox, "Dextransfer: Real world multi-fingered dexterous grasping with minimal human demonstrations," *arXiv:2209.14284*, 2022.

[48] X. B. Peng, P. Abbeel, S. Levine, and M. Van de Panne, "Deepmimic: Example-guided deep reinforcement learning of physics-based character skills," *ACM Transactions On Graphics (TOG)*, vol. 37, no. 4, pp. 1–14, 2018.

[49] X. B. Peng, A. Kanazawa, J. Malik, P. Abbeel, and S. Levine, "Sfv: Reinforcement learning of physical skills from videos," *Transactions On Graphics*, 2018.

[50] R. Villegas, D. Ceylan, A. Hertzmann, J. Yang, and J. Saito, "Contact-aware retargeting of skinned motion," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 9720–9729.

[51] Y. Kim, H. Park, S. Bang, and S.-H. Lee, "Retargeting human-object interaction to virtual avatars," *Transactions on Visualization and Computer Graphics*, 2016.

[52] T. Jin, M. Kim, and S.-H. Lee, "Aura mesh: Motion retargeting to preserve the spatial relationships between skinned characters," in *Computer Graphics Forum*, vol. 37, no. 2. Wiley Online Library, 2018, pp. 311–320.

[53] K. Aberman, P. Li, D. Lischinski, O. Sorkine-Hornung, D. Cohen-Or, and B. Chen, "Skeleton-aware networks for deep motion retargeting," *ACM Transactions on Graphics (TOG)*, vol. 39, no. 4, pp. 62–1, 2020.

[54] M. Ji, X. Peng, F. Liu, J. Li, G. Yang, X. Cheng, and X. Wang, "Exbody2: Advanced expressive humanoid whole-body control," *arXiv preprint arXiv:2412.13196*, 2024.

[55] X. Cheng, J. Li, S. Yang, G. Yang, and X. Wang, "Open-television: Teleoperation with immersive active visual feedback," *arXiv preprint arXiv:2407.01512*, 2024.

[56] Z. Fu, Q. Zhao, Q. Wu, G. Wetzstein, and C. Finn, "Humanplus: Humanoid shadowing and imitation from humans," *arXiv preprint arXiv:2406.10454*, 2024.

[57] F. Torabi, G. Warnell, and P. Stone, "Generative adversarial imitation from observation," *arXiv preprint arXiv:1807.06158*, 2018.

[58] E. Barsoum, J. Kender, and Z. Liu, "Hp-gan: Probabilistic 3d human motion prediction via gan," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2018, pp. 1418–1427.

[59] P. Li, K. Aberman, Z. Zhang, R. Hanocka, and O. Sorkine-Hornung, "Ganimator: Neural motion synthesis from a single sequence," *ACM Transactions on Graphics (TOG)*, vol. 41, no. 4, pp. 1–12, 2022.

[60] G. Tevet, S. Raab, S. Cohan, D. Reda, Z. Luo, X. B. Peng, A. H. Bermano, and M. van de Panne, "Closd: Closing the loop between simulation and diffusion for multi-task character control," *arXiv preprint arXiv:2410.03441*, 2024.

[61] WonikRobotics, "Allegrohand," https://www.wonikrobotics.com/, 2013.

[62] Robotera, "Xhand1," https://www.robotera.com/en/goods1/4.html, 2013.

[63] Shunk, "Svhand," https://schunk.com/us/en/gripping-systems/special-gripper/svh/c/PGR_3161, 2013.

[64] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 652–660.

[65] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv:1707.06347*, 2017.

[66] E. Kaufmann, L. Bauersfeld, A. Loquercio, M. Müller, V. Koltun, and D. Scaramuzza, "Champion-level drone racing using deep reinforcement learning," *Nature*, vol. 620, no. 7976, pp. 982–987, 2023.

[67] J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter, "Learning quadrupedal locomotion over challenging terrain," *Science robotics*, vol. 5, no. 47, p. eabc5986, 2020.

[68] H. Qi, B. Yi, S. Suresh, M. Lambeta, Y. Ma, R. Calandra, and J. Malik, "General in-hand object rotation with vision and touch," in *CoRL*, 2023.

[69] A. Handa, A. Allshire, V. Makoviychuk, A. Petrenko, R. Singh, J. Liu, D. Makoviichuk, K. Van Wyk, A. Zhurkevich, B. Sundaralingam, Y. Narang, J.-F. Lafleche, D. Fox, and G. State, "Dextreme: Transfer of agile in-hand manipulation from simulation to reality," in *ICRA*, 2023.

[70] V. Makoviychuk, L. Wawrzyniak, Y. Guo, M. Lu, K. Storey, M. Macklin, D. Hoeller, N. Rudin, A. Allshire, A. Handa, and G. State, "Isaac gym: High performance gpu-based physics simulation for robot learning," in *NeurIPS Datasets and Benchmarks*, 2021.

[71] A. Petrenko, A. Allshire, G. State, A. Handa, and V. Makoviychuk, "Dexpbt: Scaling up dexterous manipulation for hand-arm systems with population based training," in *RSS*, 2023.

[72] ShadowRobot, "Shadowrobot," https://www.robotera.com/en/goods1/4.html, 2013.

[73] Y.-W. Chao, W. Yang, Y. Xiang, P. Molchanov, A. Handa, J. Tremblay, Y. S. Narang, K. V. Wyk, U. Iqbal, S. Birchfield, J. Kautz, and D. Fox, "Dexycb: A benchmark for capturing hand grasping of objects," *CVPR*, 2021.

[74] J. Singla, A. Agarwal, and D. Pathak, "Sapg: Split and aggregate policy gradients," in *Forty-first International Conference on Machine Learning*.

10 Supplementary Material

10.1 Analysing the behavior learnt by our policies

Visually inspecting the behaviours learnt by the agent provides more insights into the approach. Policy rollouts in simulation for our approach can be found at https://jirl-upenn.github.io/track_reward/. The video rollouts bring out the following insights on our approach:

1. **How does our prediction model guide policy training?** Predictions of our keypoint model direct the robot hand towards reasonable grasp poses of the object.
2. **What is the impact of the hand form factor on learning?** While all robots successfully complete the task, the motion of the robot is more humanlike for hands whose form factor more closely resembles that of humans, such as the X-Hand.
3. **How does our approach perform on multiple tasks?** With the right sparse task reward, our prediction model can guide the robot to do multiple, although similar, tasks.

10.2 Performance of baselines with dense rewards

We find that our baselines, AMP (15) and Demo-guided RL (17), that leverage offline datasets alongside policy gradients fail to learn just with sparse reward. We therefore run these methods with our manually designed dense reward function. Table 1 summarises our results. We find that with dense rewards, our baselines indeed lead to non-trivial success rates. We hypothesize that this reliance on dense rewards comes from the underlying policy gradient based optimization, which is known to perform worse in sparse reward environments.

Approach	**Success Rate**
AMP with dense reward	0.193 ± 0.051
DAPG with dense reward	0.381 ± 0.223

Table 1: Baseline performance with dense rewards.

10.3 Description of our tasks in simulation

- *Grasp and Lift*: The robot must grasp an object and hold it at a specified height for at least 5 seconds. The scene contains a single object, which is randomly placed on a $1m \times 1m$ table. The episode is considered a failure if the object falls off the table, if the robot fails to maintain it at the goal height for the required duration, or if it takes more than 30 seconds to complete the task. Note that this is the simplest task in our benchmark.
- *Grasp and Lift - Clutter*: The goal of this task is similar to the previous one: grasping an object and holding it at a specific height for at least 2.5 seconds. However, instead of a single object, three objects are randomly placed on the table. The agent must correctly identify and pick the target object, specified via a categorical label, while avoiding the other objects, which serve as distractors. The success criteria remain the same as in the previous task.
- *Lift and Throw*: In this task, the agent must first lift the object and then accurately drop it into an adjacent receptacle. The episode is considered a failure if the agent fails to place the object in the bin or exceeds the 30-second time limit. Since our human dataset does not contain instances of grasp-and-drop actions, this task is designed to evaluate whether a policy can still be trained for tasks Π_h was not explicitly trained.