HRP: Human Affordances for Robotic Pre-Training

Mohan Kumar Srirama Sudeep Dasari[ψ] Shikhar Bahl[ψ] Abhinav Gupta[ψ]

Carnegie Mellon University

***Abstract*—In order to *generalize* to various tasks in the wild, robotic agents will need a suitable representation (i.e., vision network) that enables the robot to predict optimal actions given high dimensional vision inputs. However, learning such a representation requires an extreme amount of diverse training data, which is prohibitively expensive to collect on a real robot. How can we overcome this problem? Instead of collecting more robot data, this paper proposes using internet-scale, human videos to extract "affordances," both at the environment and agent level, and distill them into a pre-trained representation. We present a simple framework for pre-training representations on hand, object, and contact "affordance labels" that highlight relevant objects in images and how to interact with them. These affordances are automatically extracted from human video data (with the help of off-the-shelf computer vision modules) and used to fine-tune existing representations. Our approach can efficiently fine-tune *any* existing representation, and results in models with stronger downstream robotic performance across the board. We experimentally demonstrate (using 3000+ robot trials) that this affordance pre-training scheme boosts performance by a minimum of 15% on 5 real-world tasks, which consider three diverse robot morphologies (including a dexterous hand). Unlike prior works in the space, these representations improve performance across 3 different camera views. Quantitatively, we find that our approach leads to higher levels of generalization in out-of-distribution settings. For code, weights, and data check: https://www.cs.cmu.edu/~data4robotics/hrp/index.html**

I. Introduction

A truly generalist robotic agent must acquire diverse manipulation skills (ranging from block stacking to pouring) that work with novel objects and remain robust to realistic environmental disturbances (e.g., lighting changes, small camera shifts). Due to the scale of this challenge, the field has trended towards learning these agents directly from data [51, 67], particularly robot trajectories collected either by expert demonstrators or autonomously by the agents themselves (via Reinforcement Learning [86]). Unfortunately, there are innumerable objects/environments, so roboticists cannot tractably collect enough real-world demonstration data and/or design a simulator that captures all this diversity.

One promising solution for this "data challenge" is for the robot to learn a *suitable representation* from Out-Of-Domain (OOD) data that can be transferred into the robotics domain. For example, prior work [64, 70, 57] trained self-supervised image encoders on large scale datasets of human videos (e.g., Ego4D [33]), using standard reconstruction objectives and contrastive learning [65] objectives – e.g., Masked Auto-Encoders [40] (MAE) and Temporal Contrastive Networks [79] (TCN) respectively – developed by the broader learning community. After pre-training, these representations are used to initialize downstream imitation learning [78] algorithms. This formula is extremely flexible, and can substantially reduce the amount of robot data required for policy learning. However, the representations are often only effective when using specific camera views and robot setups. Furthermore, independent evaluations [20, 7] recently showed that these representations cannot improve (on average) over the most obvious baseline – a self-supervised ImageNet representation [40, 21]!

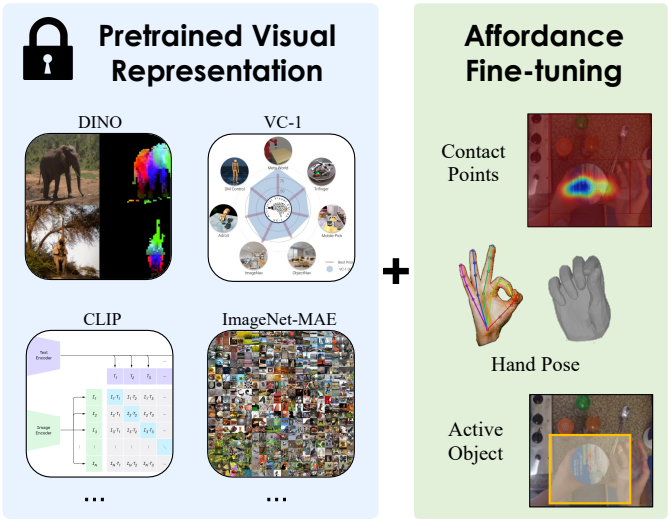

Fig. 1: Pre-trained representations offer a scalable solution to the robotics data bottleneck [64, 70, 57], but existing methods fail to reliably improve over simple baselines like ImageNet [20, 7]. Thus, we present **HRP**, a method that mines affordances (e.g., contact, hand pose, and object labels) from human videos and uses them to improve self-supervised visual encoders. Our best HRP representation consistently outperforms 6 SOTA baselines by $\geq$ **20%** across 5 diverse tasks and 3 camera views.

This result is surprising since robot trajectories and human video sequences share so much common structure: both modalities contain an agent (e.g., human or robot) using their end-effector (e.g., human hand, robot gripper) to manipulate objects in their environment. Ideally, representations trained on this data would learn useful object attributes (e.g., where to grasp a mug), and spatial relationships between the end-effector and target objects. We hypothesize that traditional self-supervised learning objectives are unable to extract this information from human video data, and that explicitly predicting these object/spatial features would result in a stronger robotic representation (i.e., higher down-stream control performance). Our key insight is

ψ Denotes equal advising.

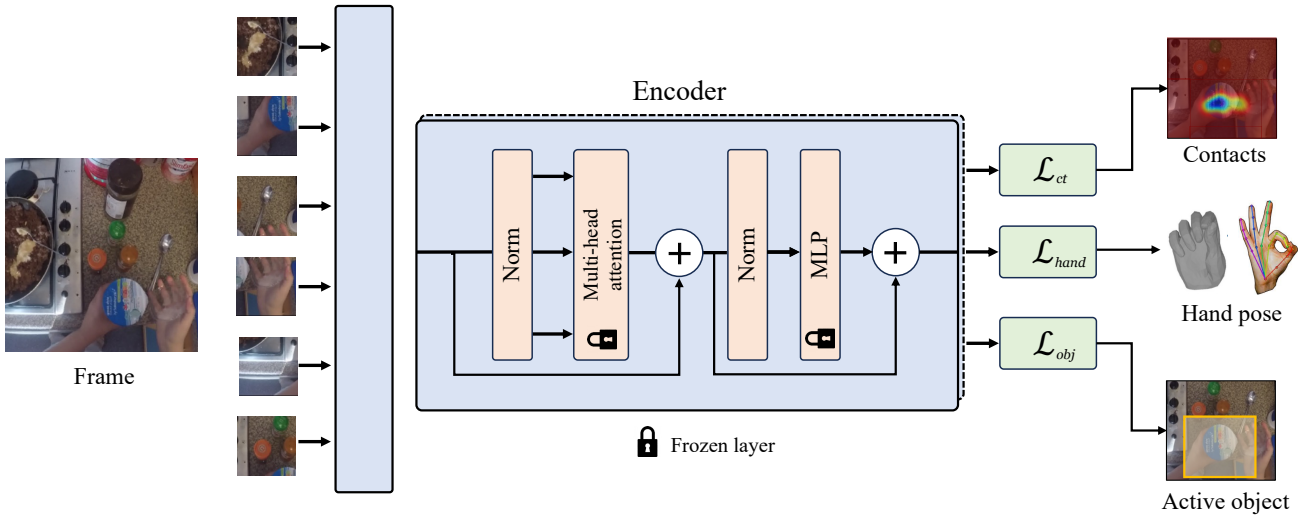

Fig. 2: HRP fine-tunes a pre-trained encoder to predict three classes of human affordance labels via L2 regression. Specifically, the network must predict future contact points, human hand poses, and the target object given an input frame from the video stream. These affordance labels are mined autonomously from a human video dataset [33] using off-the-shelf vision detectors [81]. HRP representations are then fine-tuned to solve downstream manipulation tasks via behavior cloning.

that abandoning self-supervision comes at minimal cost – the necessary object and hand labels can be scalably mined using off-the-shelf vision pipelines.

This paper proposes Human affordances for Robotic Pre-training (HRP), a semi-supervised pipeline to learn effective robotic representations from human video. HRP works in two stages: first, it extracts hand-object "affordance" information – i.e., which objects in the scene are graspable and how the robot should approach them – from human videos using off-the-shelf tracking models [81, 72]. These affordances are then distilled into a pre-existing representation network (e.g., ImageNet MAE [40]), *before* the policy fine-tuning stage. This paradigm allows us to inject useful information into the vision encoder, while preserving the flexibility of self-supervised pre-training – i.e., all labels are automatically generated and the network can be easily slotted into downstream robotic policies/controllers via fine-tuning. To summarize, **we learn stronger robotic representations by predicting object interactions and hand motion from human video dataset images** (see Fig. 1). Our investigations and experiments lead to the following contributions:

1) We present a semi-supervised learning algorithm – HRP– that leverages off-the-shelf human affordance models to learn effective robotic representations from human video. The proposed pipeline strongly outperforms representations learned purely via self-supervision.
2) Applying HRP to 6 pre-existing representations (including ImageNet [21, 40], VC-1 [57], and DINO [8]) substantially boosts robot performance. This conclusion is backed by **3000+ robot trials**, and replicates across 3 camera views, 3 distinct robotic setups, and 5 manipulation tasks!
3) Our ablation study reveals that HRP's three affordance objectives (hand, object, and contact based loss terms) are all critical for effective representation learning.
4) We show that HRP representations generalize across different imitation learning stacks – HRP improves diffusion policy [11] performance by 20%!
5) Our best representation, which increases performance by 20% over State-of-the-Art (SOTA), will be fully open-sourced, along with all code and data.

II. Related Work

Representation Learning in Robotics: End-to-end policy learning offers a scalable formula for acquiring robotic representations: instead of hand-designing object detectors or image features, a visual encoder is directly optimized to solve a downstream robotic task [51]. Numerous works applied this idea to diverse tasks including bin-picking [45, 52, 67], in-the-wild grasping [35, 85], insertion [19, 51], pick-place [6], and (non-manipulation tasks like) self-driving [5, 68, 10]. Furthermore, secondary learning objectives – e.g., dynamics modeling [36, 91], observation reconstruction [63], inverse modeling [17], etc. – can be easily added to improve data efficiency. While this paradigm can be effective, learning purely from robot data requires an expensive data collection effort (e.g., using an arm farm [52, 45], large-scale tele-operation [6], or multi-institution data collection [18, 12]), which is infeasible for (most) task settings.

To increase data efficiency, prior work applied self-supervised representation learning algorithms on out-of-domain datasets (like Ego4D [33]), and then fine-tuned the resulting representations to solve downstream tasks with a small amount of robot data – e.g., via behavior cloning on ≤ 50 expert demonstrations [64, 57, 70], directly using them as a cost/distance function to infer robot actions [56, 89], or directly pre-training robot policies from extracted human actions. [82, 58, 47]. While this transfer learning paradigm can certainly be effective, it is unclear if these robotic representations [57, 64, 70] provide a substantial boost over

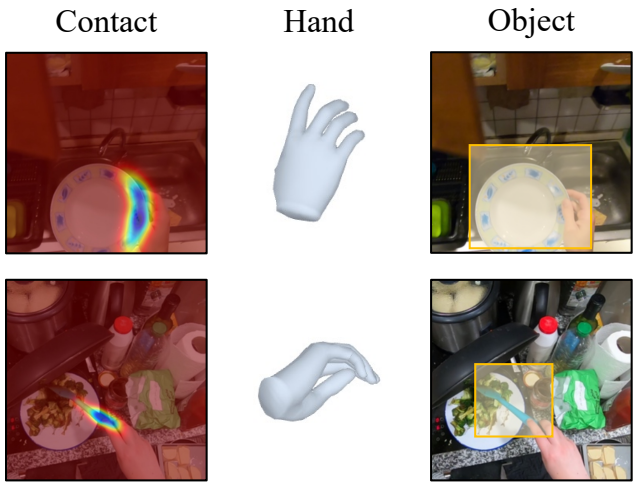

Fig. 3: We extract 3 affordances – contact heatmaps, hand poses and active object bounding boxes – from human videos.

pre-existing vision baselines [20, 7], like ImageNet MAE [40] or DINO [8]. One potential issue is that roboticists often use the same exact pre-training methods from the vision community, but merely apply them to a different data mix (e.g., VC-1 [57] applies MAE [40] to Ego4D [33]). Thus, the resulting representations are never forced to key in on object/agent level information in the scene. This paper proposes a simple formula for injecting this information into a vision encoder, using a mix of hand and object affordance losses, which empirically boost performance on robotic tasks by 25%.

Affordances from Humans: HRP is heavily inspired by the *affordance learning* literature in computer vision [29, 28]. These works use human data as a probe to learn environmental cues (i.e., affordances) that tell us how humans might interact with different objects. These include physical [23, 3, 34, 102, 37, 98, 61] and/or semantic [74, 76] scene properties, or forecast future poses [49, 71, 26, 43, 41, 88, 1, 50, 87, 33, 24, 60, 30] Affordances can also be learned at object or part levels [103, 25, 31, 62, 53, 93]. Usually such approaches leverage human video datasets [33, 14, 16, 13] or use manually annotated interaction data [54, 15, 81]. In addition to these cues, robotic affordances must consider how to move before and after interaction [2, 46]. A simple, scalable way to capture this information is by detecting these cues from human hand poses in monocular video streams [90, 46, 72, 55], which show robots reaching for and manipulating diverse, target objects. Our method combines these three approaches to create a human affordance dataset automatically from human video streams. The labels generated during this process are distilled into a representation and used to improve downstream robotics task performance.

III. Preliminaries

A. Visual Representation Learning

Our goal is to learn a visual encoder network f_θ that takes an input image I and processes it into a low-dimensional vector $f_\theta(I) \in \mathcal{R}^d$. This resulting "embedding vector" would ideally encode important scene details for robotic policy learning – like the number and type of objects in a scene and their relationship to the robot end-effector. In this paper, f_θ is a transformer network (specifically ViT-B [96], with patch size 16 and $d = 768$) parameterized with network weights θ. But to be clear, all our methods are network architecture agnostic.

Self-Supervised Learning: The computer vision community has broadly adopted *self-supervised* representation learning algorithms that can pre-train network weights without using *any* task-specific supervision. This can be accomplished using a *generative learning objective* [22], which trains f_θ alongside a decoder network D that reconstructs the original input image input from the representation. Another common approach is *contrastive learning* [65, 39], which optimizes f_θ to maximize the mutual information between the encoding and the input image (i.e., place "similar" images closer in embedding space). In practice, these methods can learn highly useful features for downstream vision tasks [40, 39], but struggle in robotics settings [20, 7]. Our goal is to inject these features into an existing self-supervised network, with an affordance-driven fine-tuning stage.

B. Extracting Affordance Labels from Human Data

Before we can do any fine-tuning, we must first curate a suitable human affordance dataset $\mathcal{D}_H$. Thankfully this task can be done automatically using off-the-shelf vision modules, applied to a set of $150K$ human-object interaction videos from Ego4D (originally sampled by Nair et al. [64]). These are subsets of larger videos (around 1.2K) videos, which were further broken down into shorter clips. Each clip contains a semantically meaningful action by the human. Each video clip V contains image frames $V = \{I_1, \ldots, I_T\}$ that depict human hands performing tasks and moving around in the scene. From these images, we obtain **contact locations**, **future hand p-oses**, and **active object labels** (examples in Fig. 3) that capture various agent-centric properties (how to move and interact) and environment centric properties (where to interact) at multiple scales, i.e. contact-level and object-level. The following sections detail how each of these labels were generated.

Contact Locations: To extract contact locations for an image I_t (with no object contact), we find the frame $I_j; j > t$ where contact with a given object will begin, using a hand-object interaction detection model [81]. Then, we use I_j to find the active object O_j and the hand mask M_j. The points intersecting M_j and O_j (acquired via skin segmentation) are our contact affordances (C_j). To account for motion between I_t and I_j, we compute the homography matrix between the frames and project those points forward. This is done using standard SIFT feature tracking [99]: $C_t = H_{j,t}C_j$. *In other words, the contact locations denote where in I_t the human will contact in the future.* Note that there could be a different number of points for each contact scenario, which is non-ideal

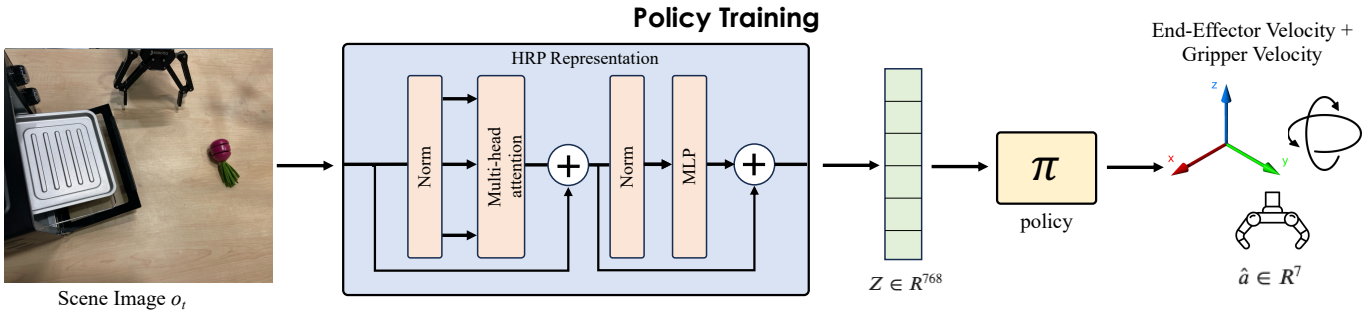

Fig. 4: We present our policy training pipeline, which uses Behavior Cloning (BC) to train policy π, using optimal expert demonstrations. The image observation (o_t) is processed using our HRP representations resulting in a latent vector z. The policy uses z to predict end-effector velocity actions (delta ee-pose/gripper), which are directly executed on the robot during test-time.

for learning. Thus, we fit a Gaussian Mixture Model with $k = 5$ modes on C_t to make a uniform contact descriptor – defined as the means c_t of the mixture model. For more details on extraction, we refer to Appendix F.

Future Hand Poses: This affordance label captures how the human moves next (e.g., to complete a task or reach an object), as the video V progresses. Given a current frame I_t, we detect the human hand's 2d wrist position (h_{t+k}) in a future frame I_{t+k}, where usually $k = 30$ (empirically determined). This is done using the Frank Mocap [72] hand detector. To correctly account for the human's motion, these wrist points are back-projected (again using the camera homography matrix) to I_t to create the final "future wrist label," $h_t = H_{t+k,t}h_{t+k}$.

Active Object Labels: In a similar manner to the contact location extraction, we run a hand-object interaction detection model [81] on V to find the image where contact began I_c. The same detector is used to find the four bounding box coordinates of the object that is being interacted with, which we refer to as the "active object." These coordinates b_c are then projected to every other frame I_t, using the homography matrix (see above). This results in an active bounding box b_t for each image in V.

IV. Introducing HRP

A variety of visual pre-training tasks have been shown to help with downstream robotic performance– ranging from simple ImageNet classification [80] to self-supervised learning on human video [70, 64, 56, 57, 66]. Although these approaches operate on human videos and simple image frames, they fail to explicitly model the rich hand-object contacts depicted. In contrast, we believe explicitly modeling the *affordances* [28] in this data could allow us to learn useful information about the agent's intents, goals, and actions. Indeed, past work has shown that affordances can act as strong prior for manipulation [100, 44, 84, 95, 2, 9, 42, 4] in general. Moreover, this information can be represented in many different formats, such as physical attributes, geometric properties, interactions, object bounding boxes, or motion forecasting. We observe that most tasks of interests humans perform are with their hands. We thus focus on training our model to predict hand-object interactions and hand motion.

We present HRP, a simple and effective representation learning approach that injects hand-object interaction priors into a self-supervised network, f_θ, using an automatically generated human affordance dataset, $\mathcal{D}_H$ (see above for definitions and dataset mining approach). HRP is illustrated in Fig. 2, and the following sections describe its implementation in detail.

A. Training HRP

The initial network f_θ is fine-tuned using batches sampled from the human dataset: $(I_t, c_t, h_t, b_t) \sim \mathcal{D}_H$, where c_t, h_t, and b_t are contact, hand, and object affordances corresponding to image I_t (see Sec. III-B for definitions). Some frames may not include all 3 affordances, so we include 3 mask variables – $m_t^{(c)}, m_t^{(h)}, m_t^{(b)}$ – so the missing values can be ignored during training. We add 3 small affordance modules – p_c, p_h, p_b – on top of f_θ that are trained to regress the respective affordances for I_t. This results in the following three loss functions:

$$\mathcal{L}_{\text{ct}} = ||c_t - p_c(f_\theta(I_t))||_2 \tag{1}$$

$$\mathcal{L}_{\text{hand}} = ||h_t - p_h(f_\theta(I_t))||_2 \tag{2}$$

$$\mathcal{L}_{\text{obj}} = ||b_t - p_b(f_\theta(I_t))||_2 \tag{3}$$

The full loss is:

$$\mathcal{L} = m_t^{(c)}\lambda_{\text{ct}}\mathcal{L}_{\text{ct}} + m_t^{(h)}\lambda_{\text{hand}}\mathcal{L}_{\text{hand}} + m_t^{(b)}\lambda_{\text{obj}}\mathcal{L}_{\text{obj}} \tag{4}$$

Where the λs are hyper-parameters that control the relative weight of each affordance loss. We empirically found $\lambda_{\text{obj}} = 0.05, \lambda_{\text{ct}} = 0.005, \lambda_{\text{hand}} = 0.5$ to be optimal for downstream performance (see Appendix E).

B. Implementation Details

Our affordance dataset ($\mathcal{D}_H$) is at least an order of magnitude smaller than the pre-training image dataset initially used by the baseline representation (e.g., ImageNet has 1M frames v.s. our 150K). To preserve the useful features learned from the larger pre-training distribution, we keep most of the parameters in θ fixed during HRP fine-tuning. Specifically, we only fine-tune the baseline network's normalization layers and leave the rest fixed, which has been shown to be an effective approach [27, 97]. In the case of our ViT-B this amounts to fine-tuning only the LayerNorm parameters γ and β:

$$\text{LayerNorm}(x) = \frac{x - \mu}{\sigma}\gamma + \beta \tag{5}$$

These parameters are fine-tuned to minimize $\mathcal{L}$ using standard back-propagation and the ADAM [48] optimizer.

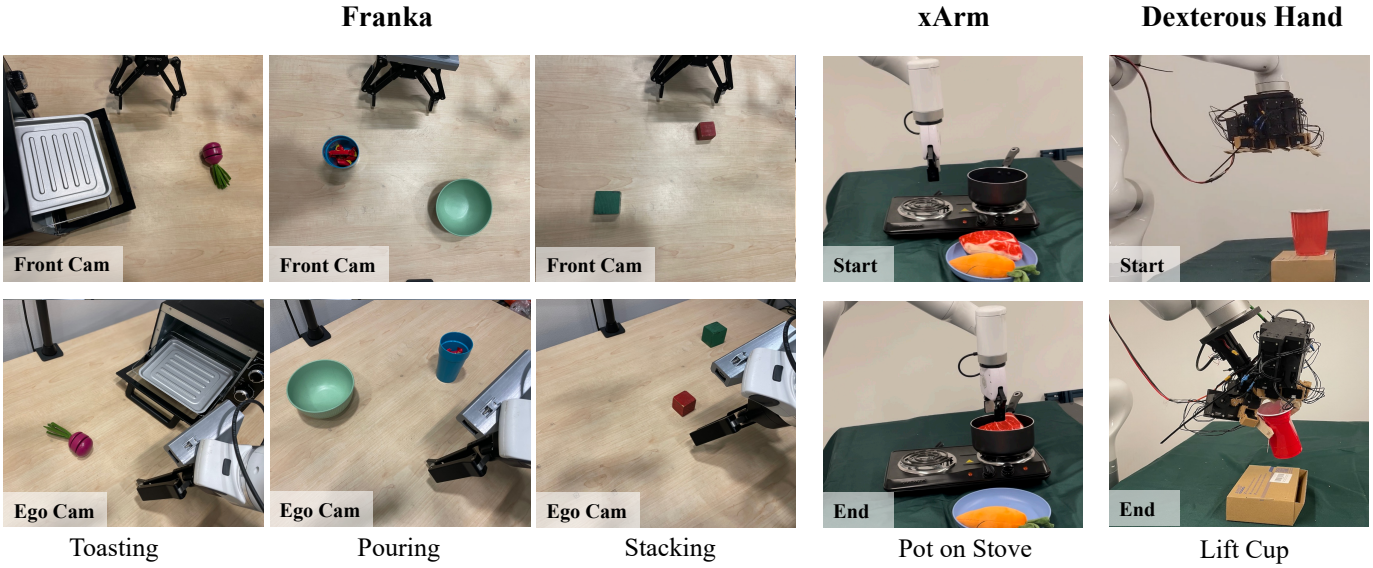

Fig. 5: Our experiments consider 5 unique manipulation tasks, ranging from classic block-stacking to a multi-stage toasting scenario. These tasks are implemented on 3 unique robot setups, including a high Degree-of-Freedom dexterous hand (right). The 3 camera views shown – front, ego, and side views (for xArm/dexterous hand) – are the same views ingested by the policy during test-time. Note that 3 of the tasks consider 2 unique camera views in order to test for robustness!

V. Experimental Details

Our contributions are validated using a simple empirical formula: first, HRP is applied to each baseline model (listed below). Then, (following standard practice [64, 57, 20]) the resulting representation is fine-tuned into a manipulation policy using behavior cloning. Details for each stage are provided below, and the HRP is illustrated in Fig. 2.

Baseline Representations: We chose 6 representative, SOTA baselines from both the vision and robotics communities:

1) **ImageNet MAE** was pre-trained by applying the Masked Auto-Encoders [40] (MAE) algorithm to the ImageNet-1M dataset [21]. It achieved SOTA performance across a suite of vision tasks, and is the first self-supervised representation to beat supervised pre-training. We use the standard Masked Auto Encoder training scheme for this, using hyperparmaeters from MAE [40].
2) **Ego4D MAE** was trained by applying the MAE algorithm to a set of 1M frames sampled from the Ego4D dataset [33]. For consistency with prior work, we use the same 1M frame-set sampled by the R3M authors [64]. We use the standard Masked Auto Encoder training scheme for this, using hyperparmaeters from MAE [40].
3) **CLIP [69]** is a SOTA representation for internet data. It was learned by applying contrastive learning [65] to a large set natural language - image pairs crawled from internet captions. We used publicly available model weights.
4) **DINO [8]** was trained using a self-distillation algorithm that encourages the network to learn local-to-global image correspondences. DINO's emergent segmentation capabilities could be well suited for robotics, and it has already shown SOTA performance in sim [7]. We used publicly available model weights.
5) **MVP [70]** was trained by applying MAEs to a mix of in-the-wild datasets (100 DoH [81], Ego4D [33], etc.). The authors showed strong performance on various manipulation tasks. We used publicly available model weights.
6) **VC-1 [57]** was trained in a similar fashion to MVP, but used a larger dataset mix. It showed strong performance on visual navigation tasks. We used publicly available model weights.

Note that each baseline is parameterized with the same ViT-B encoder w/ patch size 16 (see Sec. III-B), to ensure apples-to-apples comparisons.

Policy Learning: Each representation is evaluated on downstream robotic manipulation tasks, by fine-tuning it into a policy (π) using Behavior Cloning [68, 77, 73]. Note that π must predict the expert action (a_t – robot motor command) given the observation (o_t – input image and robot state): $a_t \sim \pi(\cdot|o_t)$. And π is learned using a set of *50 expert demonstrations* $\mathcal{D} = \{\tau_1, \ldots, \tau_{50}\}$, where each demonstration $\tau_i = [(o_0, a_0), \ldots, (o_T, a_T)]$ is a trajectory of expert observation-action tuples. In our case, π is parameterized by a small 2-layer MLP (p) placed atop the pre-trained encoder $p(f(o_t))$ that predicts a Gaussian Mixture policy distribution w/ 5 modes. Both the policy network and visual encoder are optimized end-to-end (using ADAM [48] w/ $lr = 0.0001$ for 50K steps) to maximize the log-likelihood of expert actions: $\max_{p,f} log(\pi(a_t|p(f(o_t))))$. During test time actions are sampled from this distribution and executed on the robot: $a_t \sim \pi(\cdot|p(f(o_t)))$. This is a standard evaluation formula that closely follows best practices from prior robotic representation learning work [59, 20].

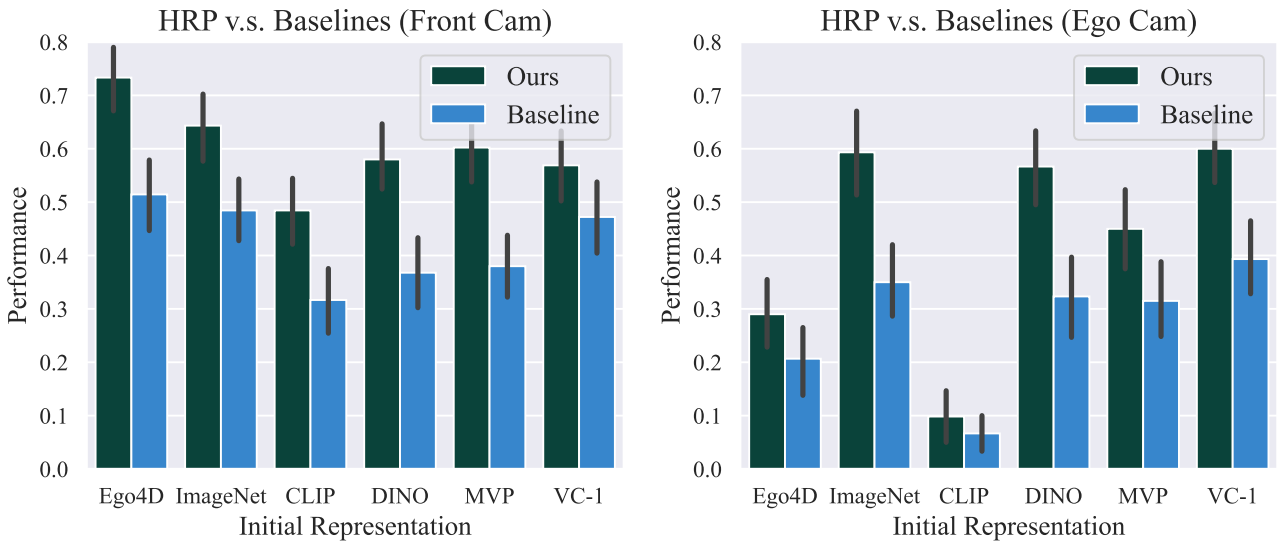

Fig. 6: We apply HRP to 6 different baseline representations and plot how it affects performance on average across the *toasting*, *pouring*, and *stacking* tasks. This evaluation procedure is repeated using two distinct cameras (shown in Fig. 5) in order to test if HRP representation are robust to view shifts. We find that HRP representations consistently and substantially outperform their vanilla baselines, and that this effect holds across both the front (left) and ego (right) cameras. In fact, our strongest representation – ImageNet + HRP– delivers SOTA performance on both views!

Real World Tasks: We fine-tune policies for each representation on the 5 diverse tasks listed below, which are implemented on 3 unique robotic setups, including a dexterous hand (illustrated in Fig. 5). 50 expert fine-tuning demonstrations were collected for each task via expert tele-operation. Note that the stacking, pouring, and toasting tasks were *evaluated twice using different camera views* to test robustness!

- **Stacking:** The stacking task requires the robot to pick up the red block and place it on the green block. During test time both blocks' starting positions are randomized to novel locations (not seen in training). A trial is marked as successful if the robot correctly picks and stacks the red block, and half successful if the red block is unstably placed on the green block. This task is implemented on a Franka robot and uses both an Ego and Front camera viewpoint.
- **Pouring:** The pouring task requires the robot to pick up the cup and pour the material (5 candies) into the target bowl. During test time we use novel cups and bowls and place each in new test locations. This task's success metric is the fraction of candies successfully poured (e.g., $2/5$ candies poured $\rightarrow 0.4$ success). This task was also implemented on the Franka using Ego and Front cameras.
- **Toasting:** The toasting task requires the robot to pick up a target object, place it in the toaster oven, and shut the toaster. This is a challenging, multi-stage task. During test time the object type, and object/toaster positions are both varied. A test trial is marked as successful if the whole task is completed, and 0.5 successful if the robot only successfully places the object. This is the final task implemented on Franka w/ Ego and Front camera views.
- **Pot on Stove:** The stove task requires picking up a piece of meat or carrot from a plate and placing it within a pot on a stove. During test time, novel "food" objects are used and the location is randomized. A trial is marked as successful if the food is correctly placed in the pot. This task is implemented on a xArm and uses the side camera view.
- **Hand Lift Cup** This task requires a dexterous hand to reach, grasp, and lift up a deformable red solo up. The hand's high dimensional action space ($\mathcal{R}^{20}$) makes this task especially challenging. A trial is marked successful if the cup is stably grasped and picked. This task is implemented on a custom dexterous hand using a side camera view.

VI. Results

Our experiments are designed to answer the following:

1) **Can HRP improve the performance of the pre-trained baseline networks (listed above)**? Does the effect hold across different camera views and/or new robots? (see Sec. VI-A)
2) Our affordance labels are generated using off-the-shelf vision modules – **does distilling their affordance outputs into a representation (via HRP) work better than simply using those networks as encoders?** (see Sec. VI-B)
3) How does HRP compare against alternate forms of supervision on the same human video dataset? (see Sec. VI-C)
4) How important are each of the three affordance losses for HRP's final performance? And is it really best to only fine-tune the LayerNorms and leave the other weights fixed? (see Sec. VI-D)
5) Can HRP handle scenarios with OOD distractor objects during test time? (see Sec. VI-E)

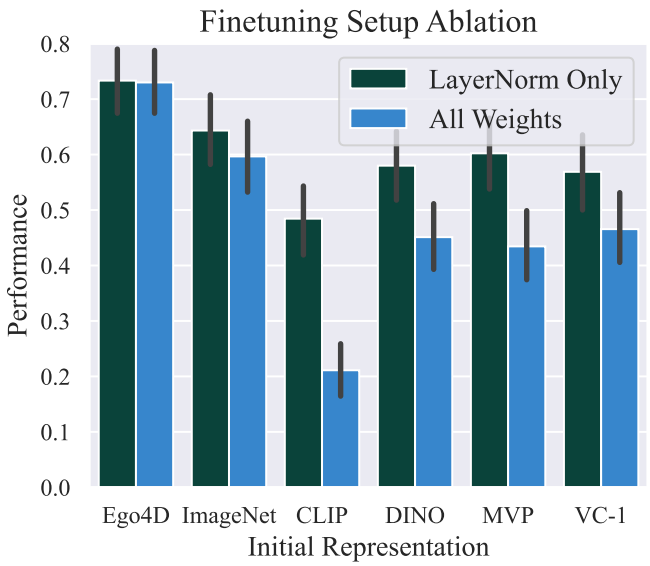

Fig. 7: This chart applies an ablated HRP method (full fine-tuning) to the 6 baseline representations, and compares their average performance v.s. standard HRP representations on the *toasting*, *pouring*, and *stacking* tasks (front cam). We find that LayerNorm only fine-tuning is almost always superior.

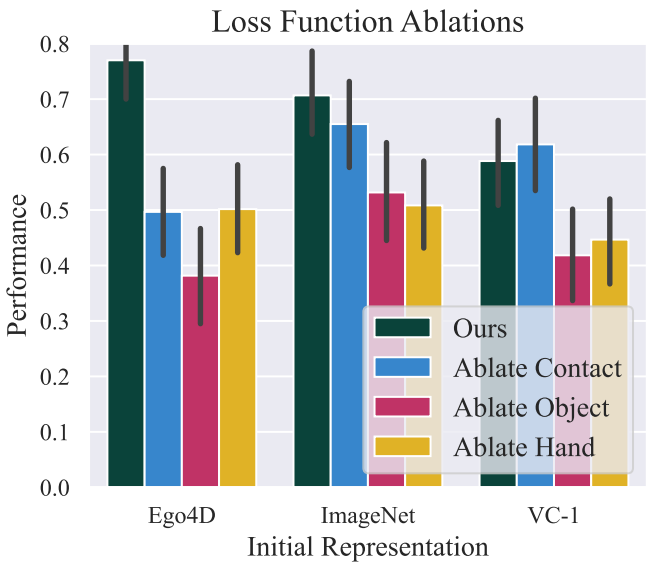

Fig. 8: We drop each of the 3 losses in HRP, and compare the ablated method's average performance (across the *toasting*, *pouring*, *stacking* tasks) against full `HRP` representations. Due to the number of ablations involved, this experiment is only run on the `Ego4D`, `ImageNet`, and `VC-1` base models. We find that the object and hand losses are critical for good performance, but the contact loss only makes a significant impact on the `Ego4D` base model.

6) Can HRP representations work with different imitation learning pipelines, like diffusion policy [11]? (see Sec. VI-F)

Note that all experiments were conducted on real robot hardware, and the models were all tested back-to-back (i.e., using proper A/B evaluation) using 50+ trials per model to guarantee statistical significance. Note that all of our figures and tables report success rates (sometimes averaged across the toasting, stacking, and pouring tasks) alongside std. err. to quantify experimental uncertainty – i.e. success% $\pm$ std. err..

	Teacher ResNet	**HRP Models**		
Front Cam	100DoH [81]	w/ Ego4D	w/ ImageNet	w/ CLIP
Toasting	35% $\pm$ 15%	**83%** $\pm$ 9%	75% $\pm$ 10%	50% $\pm$ 11%
Pouring	34% $\pm$ 13%	**60%** $\pm$ 11%	48% $\pm$ 12%	39% $\pm$ 11%
Stacking	0%	**77%** $\pm$ 10%	70% $\pm$ 11%	57% $\pm$ 11%
Average	35% $\pm$ 10%	**73%** $\pm$ 6%	64% $\pm$ 7%	48% $\pm$ 6%

TABLE I: This table compares 3 representations trained w/ `HRP` against the teacher ResNet [81] that generated our human affordance dataset (see Sec. III-B). We find that the ResNet teacher underperforms even the worst `HRP` representation (fine-tuned from `CLIP`), *even after excluding the stacking task, which it failed on.*

A. Improving Representations w/ HRP

To begin, we evaluate the 6 baseline representations (detailed in Sec. V) on the *toasting*, *pouring*, and *stacking* tasks using the *front camera view*. Then, we apply `HRP` to each of these baselines, and evaluate those 6 new models on the same tasks. Average success rates across all 3 tasks are presented in Fig. 6 (left), and the full table is in the Appendix B. First, this experiment demonstrates that ImageNet MAE is still highly competitive on real-world manipulation tasks when compared to other self-supervised representations from the vision [33, 8], machine learning [69], and robotics communities [92, 57]. Second, we show that `HRP` **uniformly boosts performance** on downstream robotics tasks – i.e., `baseline + HRP > baseline` for every baseline representation considered! Thus, we conclude that the affordance information injected by our method is highly useful for robot learning, and (for now) cannot be learned in a purely self-supervised manner.

Second Camera View: A common critique is that robotic representations perform very differently when the camera view (even slightly) changes. To address this issue, we replicated the first experiment using a radically different *ego view*, where the camera is placed over the robot's shoulder (i.e., on its "head"). While perhaps a more realistic view, it is significantly more challenging due to the increased robot-object occlusion. Average success rates are presented in Fig. 6 (right), and a per-task breakdown is in Appendix C. Note that our findings replicate almost exactly from the front camera view. The ImageNet MAE representation is still competitive with the other baselines, and applying `HRP` uniformly improves the baseline performance. In addition, we find that **HRP injects a higher level of robustness to camera view shifts**, when compared to the baselines. For example, we find that `ImageNet + HRP` performs the same on the ego and front camera, even though the `ImageNet` baseline clearly prefers the front cam. This general effect holds (to varying degrees) across all six baselines!

	Ego4D		ImageNet		CLIP	
	+ HRP	+ Semantic	+ HRP	+ Semantic	+ HRP	+ Semantic
Toasting	**83%** $\pm$ 9%	25% $\pm$ 13%	**75%** $\pm$ 10%	40% $\pm$ 14%	**50%** $\pm$ 11%	20% $\pm$ 13%
Pouring	**60%** $\pm$ 11%	30% $\pm$ 13.4%	**48%** $\pm$ 12%	26% $\pm$ 11%	**39%** $\pm$ 11%	22% $\pm$ 10%
Stacking	**77%** $\pm$ 10%	30% $\pm$ 11%	**70%** $\pm$ 11%	40% $\pm$ 12%	**57%** $\pm$ 11%	30% $\pm$ 13%
Average	**73%** $\pm$ 6%	28% $\pm$ 7%	**64%** $\pm$ 7%	35% $\pm$ 7%	**48%** $\pm$ 6%	24% $\pm$ 7%

TABLE II: We create `Semantic` representations by fine-tuning the `Ego4D`, `ImageNet`, and `CLIP` baselines using a classification loss, instead of HRP's affordance loss. Note that the exact same Ego4D clips (see Sec. III-B) are used during semantic fine-tuning, thanks to object class labels generated automatically by Detic [101]. The sematic representations were evaluated (using the same BC pipeline) on the Toasting, Pouring, and Stacking tasks, and compared against their `HRP` counterparts. Success rates (and standard error) are reported above. We find that the affordance supervision provided by HRP is vastly superior to the semantic alternative.

	Ego4D		ImageNet	
	w/ HRP	Baseline	w/ HRP	Baseline
Pot on Stove	**50%** $\pm$ 17%	40% $\pm$ 16%	**60%** $\pm$ 16%	40% $\pm$ 16%
Hand Lift Cup	**50%** $\pm$ 17%	40% $\pm$ 16%	**50%** $\pm$ 17%	30% $\pm$ 15%

TABLE III: We present results of `Ego4D + HRP` and `ImageNet + HRP`, as well as the respective baselines on the x-Arm (Pot on Stove) and a dexterous hand task (Lift Cup). We see that `HRP` can even boost performance in multiple morphologies, including a high-degree of freedom dexterous hand [83].

Scaling to More Robots: Finally, we verify that `HRP` representations can provide benefits on other robotic hardware setups. Specifically, we compare `Ego4D + HRP` and `ImageNet + HRP` versus the respective baselines on the *Pot on Stove* (xARM) and *Hand Lift Cup* (dexterous hand) tasks. Results are presented in Table III. Note that `HRP` representations provide consistent and significant performance during policy learning on these radically different robot setups, which both also use a unique side camera view. This gives us further confidence in HRP's view robustness and demonstrates that these representations are not tied to specific hardware setups, and can scale to complex morphologies like dexterous hands.

B. Distillation w/ HRP Improves Over Label Networks

It is clear that applying `HRP` to self-supervised representations results in a consistent boost. However, the hand, object, and contact affordance labels for `HRP` themselves come from neural networks (see Sec. III-B) – specifically we use the ResNet-101 [38] detector from 100DoH [81] as a label generator for our active object and contact affordance. The hand affordance we use comes from FrankMocap [72], which uses 100DoH [81] as a base model. Thus, does distilling labels from this detector via `HRP` actually provide a benefit over simply using the 100DoH model itself as a pre-trained representation? To test this question, we fine-tune policies on the toasting, pouring, and stacking (front cam) tasks and compare them against `HRP` applied to `ImageNet`, `Ego4D`, and (the weakest model) `CLIP` (see Table I). In all cases, our representation handily beats the 100DoH policy. So while the affordance labels can dramatically boost policy learning (via `HRP`), the source/teacher models are not at all competitive on robotics tasks.

Initialization	w/ HRP	MAE Initialization
Ego4D	**40%** $\pm$ 15%	15% $\pm$ 11%
ImageNet	**40%** $\pm$ 15%	**40%** $\pm$ 15%

TABLE IV: This table compares `Ego4D + HRP` and `ImageNet + HRP` representations against their respective baselines on a *stacking w/ distractors* task. Here the robot must successfully complete the usual stacking task, when extraneous objects (an orange carrot, and a green bowl) are added to the scene. We find that `Ego4D + HRP` improved over its baseline on this task, but `ImageNet + HRP` performed the same as its baseline.

C. Comparing Against Alternate Forms of Supervision

We now analyze if HRP's losses are better suited for robotics tasks than an alternate supervision scheme. To be clear, the previous results already demonstrated that `HRP + Ego4D` out-performed the `Ego4D` baseline by up to 20% (see Fig. 6; left), despite being sourced from the same image data. However, it could be that the additional fine-tuning step with the $100K$ filtered interaction clips is responsible, and the specific affordance losses are not key. To test this, we ran a modified version of HRP using a semantic classification loss, instead of our affordance hand-object losses. The ground-truth labels for each image were obtained using the Detic object detector [101]. We then similarly fine-tuned the `ImageNet`, `Ego4D`, and `CLIP` baseline representation using these labels, and compared them against the respective `HRP` models on the toasting, pouring, and stacking tasks. The results are presented in Table II We find that the `HRP` models perform significantly better on every task. Thus, we conclude that HRP's affordance losses play an important role in boosting performance (i.e., it's not just data or extra fine-tuning).

D. What Design Decisions are Important?

The following section ablates the key components of `HRP` to evaluate their relative importance. First, we apply `HRP` to each of the 6 baseline representations again, but this time none of the weights are kept fixed (see Sec. IV-B). These representations are fine-tuned on the toasting, stacking, and pouring tasks (front cam), and compared against the original `HRP` representations in Fig. 7. Note that fine-tuning all the layers results in a substantial performance hit on average, and this trend is consistent regardless of the base representation!

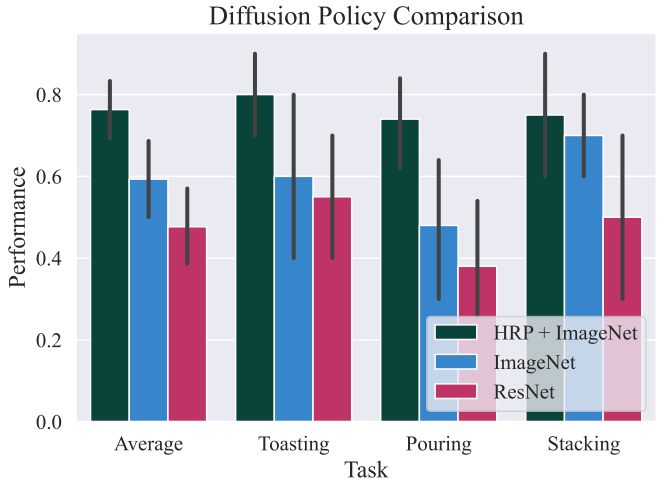

Fig. 9: This figure tests if HRP representations can boost performance when using a radically different imitation learning framework – namely Diffusion Policy [11]. We evaluate diffusion policies (following the U-Net + state action formula described by Chi et. al [11]) on the toasting, pouring, and stacking tasks using 3 different visual encoders: the default ResNet encoder from RoboMimic [59], the `ImageNet + MAE` baseline, and our `HRP + ImageNet` features. We find a clear improvement when using HRP weights, which suggests that HRP is applicable to different imitation learning frameworks!

Thus, we conclude fine-tuning only the layer norms when applying `HRP` is the correct decision.

Next, we ablate each of the affordance losses in Eq. 4, by applying HRP three times: once with $\lambda_{\text{ct}} = 0$, then with $\lambda_{\text{hand}} = 0$, and finally $\lambda_{\text{obj}} = 0$. This process is repeated using 3 different base models; `ImageNet`, `Ego4D`, and `VC-1`. This creates 9 ablated models (3 losses x 3 initializations) that are compared versus the full HRP models on the toasting, pouring, and stacking tasks. The average results are presented in Fig. 8, and the full, per-task breakdown is presented in the Appendix D. We find that removing the object (Eq. 3) and hand (Eq. 2) losses uniformly results in significant performance degradation. Meanwhile, the contact loss (Eq. 1) only provides a significant boost for the `Ego4D` base model but does not affect the others. Thus, we conclude that object and hand losses are critical for our method, while the contact loss is more marginal, most likely due to the fact that the extraction of contacts is a relatively noisy process.

E. Novel Distractors During Test-Time

We evaluate the performance of `HRP` and baseline approaches in OOD settings, by adding extraneous "distractor" objects (an orange carrot and a light green bowl) in the stacking task. The robot must successfully ignore the distractor and complete the task. Results are presented in Table IV. We found that both `ImageNet + HRP` and `ImageNet` had the same level of robustness to distractors. Meanwhile, `Ego4D`'s performance dropped substantially, while `Ego4D + HRP` remained robust. Our hypothesis is that human data by itself does not contain enough information to allow for OOD tasks. However, using `HRP` allows for more focus on task-relevant features, even when the representation is trained on less diverse data.

F. Evaluating w/ Diffusion Policy

Finally, we analyze if `HRP` representations offer improvements when using a radically different imitation learning framework, like diffusion policy [11]. Specifically, we adopt the original U-Net action prediction head and environment setup from Chi et. al. [11], but replace their ResNet visual encoder (inspired from RoboMimic [59]) with our `HRP + ImageNet` ViT-B model. Then we compare this HRP enhanced diffusion policy implementation, against (diffusion agents which use) both the original ResNet encoder and the baseline `ImageNet` ViT-B. Results for the (Franka) stacking, pouring, and toasting tasks are presented in Fig. 9. We find that `HRP + ImageNet` significantly improves over both alternatives (76% for HRP v.s., 56% for Chi et. al.'s implementation [11]), despite using a radically different imitation learning objective/setup! Thus, we conclude that HRP representations can boost performance across different setups.

VII. Discussion and Future Work

In this paper, we investigate human affordances as a strong prior for training visual representations. Thus, we present HRP, a semi-supervised pipeline that extracts contact points, hand poses, and activate objects from human videos, and uses these affordances for fine-tuning representations. HRP improves base model performance drastically, for five different, downstream behavior cloning tasks, across three robot morphologies and three camera views. All components of our approach, including LayerNorm tuning, our three affordances, and our distillation process (from affordance labels to representations) are important for the model's success. One key limitation of this approach is that it has only been tested on imitation settings in this paper. In the future, we hope to not only scale this approach to many more tasks and robot morphologies, but also incorporate HRP in other robot learning paradigms such as reinforcement learning or model based control.

References

[1] Yazan Abu Farha, Alexander Richard, and Juergen Gall. When will you do what?-anticipating temporal occurrences of activities. In *CVPR*, 2018. 3

[2] Shikhar Bahl, Russell Mendonca, Lili Chen, Unnat Jain, and Deepak Pathak. Affordances from human videos as a versatile representation for robotics. 2023. 3, 4, 16

[3] Aayush Bansal, Bryan Russell, and Abhinav Gupta. Marr revisited: 2d-3d alignment via surface normal prediction. In *CVPR*, 2016. 3

[4] Homanga Bharadhwaj, Abhinav Gupta, Vikash Kumar, and Shubham Tulsiani. Towards generalizable zero-shot manipulation via translating human interaction plans. *arXiv preprint arXiv:2312.00775*, 2023. 4

[5] 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 preprint arXiv:1604.07316*, 2016. 2

[6] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, Julian Ibarz, Brian Ichter, Alex Irpan, Tomas Jackson, Sally Jesmonth, Nikhil Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Kuang-Huei Lee, Sergey Levine, Yao Lu, Utsav Malla, Deeksha Manjunath, Igor Mordatch, Ofir Nachum, Carolina Parada, Jodilyn Peralta, Emily Perez, Karl Pertsch, Jornell Quiambao, Kanishka Rao, Michael Ryoo, Grecia Salazar, Pannag Sanketi, Kevin Sayed, Jaspiar Singh, Sumedh Sontakke, Austin Stone, Clayton Tan, Huong Tran, Vincent Vanhoucke, Steve Vega, Quan Vuong, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. Rt-1: Robotics transformer for real-world control at scale. In *arXiv preprint arXiv:2212.06817*, 2022. 2

[7] Kaylee Burns, Zach Witzel, Jubayer Ibn Hamid, Tianhe Yu, Chelsea Finn, and Karol Hausman. What makes pre-trained visual representations successful for robust manipulation? *ArXiv*, 2023. 1, 3, 5

[8] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. *CVPR*, 2021. 2, 3, 5, 7

[9] Matthew Chang, Aditya Prakash, and Saurabh Gupta. Look ma, no hands! agent-environment factorization of egocentric videos. *Advances in Neural Information Processing Systems*, 36, 2024. 4

[10] Dian Chen, Brady Zhou, Vladlen Koltun, and Philipp Krähenbühl. Learning by cheating. In *CoRL*, 2020. 2

[11] Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. In *Proceedings of Robotics: Science and Systems (RSS)*, 2023. 2, 7, 9

[12] Open X-Embodiment Collaboration, Abhishek Padalkar, Acorn Pooley, Ajinkya Jain, Alex Bewley, Alex Herzog, Alex Irpan, Alexander Khazatsky, Anant Rai, Anikait Singh, Anthony Brohan, Antonin Raffin, Ayzaan Wahid, Ben Burgess-Limerick, Beomjoon Kim, Bernhard Schölkopf, Brian Ichter, Cewu Lu, Charles Xu, Chelsea Finn, Chenfeng Xu, Cheng Chi, Chenguang Huang, Christine Chan, Chuer Pan, Chuyuan Fu, Coline Devin, Danny Driess, Deepak Pathak, Dhruv Shah, Dieter Büchler, Dmitry Kalashnikov, Dorsa Sadigh, Edward Johns, Federico Ceola, Fei Xia, Freek Stulp, Gaoyue Zhou, Gaurav S. Sukhatme, Gautam Salhotra, Ge Yan, Giulio Schiavi, Hao Su, Hao-Shu Fang, Haochen Shi, Heni Ben Amor, Henrik I Christensen, Hiroki Furuta, Homer Walke, Hongjie Fang, Igor Mordatch, Ilija Radosavovic, Isabel Leal, Jacky Liang, Jaehyung Kim, Jan Schneider, Jasmine Hsu, Jeannette Bohg, Jeffrey Bingham, Jiajun Wu, Jialin Wu, Jianlan Luo, Jiayuan Gu, Jie Tan, Jihoon Oh, Jitendra Malik, Jonathan Tompson, Jonathan Yang, Joseph J. Lim, João Silvério, Junhyek Han, Kanishka Rao, Karl Pertsch, Karol Hausman, Keegan Go, Keerthana Gopalakrishnan, Ken Goldberg, Kendra Byrne, Kenneth Oslund, Kento Kawaharazuka, Kevin Zhang, Keyvan Majd, Krishan Rana, Krishnan Srinivasan, Lawrence Yunliang Chen, Lerrel Pinto, Liam Tan, Lionel Ott, Lisa Lee, Masayoshi Tomizuka, Maximilian Du, Michael Ahn, Mingtong Zhang, Mingyu Ding, Mohan Kumar Srirama, Mohit Sharma, Moo Jin Kim, Naoaki Kanazawa, Nicklas Hansen, Nicolas Heess, Nikhil J Joshi, Niko Suenderhauf, Norman Di Palo, Nur Muhammad Mahi Shafiullah, Oier Mees, Oliver Kroemer, Pannag R Sanketi, Paul Wohlhart, Peng Xu, Pierre Sermanet, Priya Sundaresan, Quan Vuong, Rafael Rafailov, Ran Tian, Ria Doshi, Roberto Martín-Martín, Russell Mendonca, Rutav Shah, Ryan Hoque, Ryan Julian, Samuel Bustamante, Sean Kirmani, Sergey Levine, Sherry Moore, Shikhar Bahl, Shivin Dass, Shuran Song, Sichun Xu, Siddhant Haldar, Simeon Adebola, Simon Guist, Soroush Nasiriany, Stefan Schaal, Stefan Welker, Stephen Tian, Sudeep Dasari, Suneel Belkhale, Takayuki Osa, Tatsuya Harada, Tatsuya Matsushima, Ted Xiao, Tianhe Yu, Tianli Ding, Todor Davchev, Tony Z. Zhao, Travis Armstrong, Trevor Darrell, Vidhi Jain, Vincent Vanhoucke, Wei Zhan, Wenxuan Zhou, Wolfram Burgard, Xi Chen, Xiaolong Wang, Xinghao Zhu, Xuanlin Li, Yao Lu, Yevgen Chebotar, Yifan Zhou, Yifeng Zhu, Ying Xu, Yixuan Wang, Yonatan Bisk, Yoonyoung Cho, Youngwoon Lee, Yuchen Cui, Yueh hua Wu, Yujin Tang, Yuke Zhu, Yunzhu Li, Yusuke Iwasawa, Yutaka Matsuo, Zhuo Xu, and Zichen Jeff Cui. Open X-Embodiment: Robotic learning datasets and RT-X models, 2024. 2

[13] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. Scaling egocentric vision: The epic-kitchens dataset. In *ECCV*, 2018. 3

[14] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. Scaling egocentric vision: The epic-kitchens dataset. In *European Conference on Computer Vision (ECCV)*, 2018. 3

[15] Ahmad Darkhalil, Dandan Shan, Bin Zhu, Jian Ma, Amlan Kar, Richard Higgins, Sanja Fidler, David Fouhey, and Dima Damen. Epic-kitchens visor benchmark: Video segmentations and object relations. *Advances in Neural Information Processing Systems*, 35:13745–13758, 2022. 3

[16] 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. 3

[17] Sudeep Dasari and Abhinav Gupta. Transformers for one-shot visual imitation. In *Conference on Robot Learning*, pages 2071–2084. PMLR, 2021. 2

[18] Sudeep Dasari, Frederik Ebert, Stephen Tian, Suraj Nair, Bernadette Bucher, Karl Schmeckpeper, Siddharth Singh, Sergey Levine, and Chelsea Finn. Robonet: Large-scale multi-robot learning. *arXiv preprint arXiv:1910.11215*, 2019. 2

[19] Sudeep Dasari, Jianren Wang, Joyce Hong, Shikhar Bahl, Yixin Lin, Austin S Wang, Abitha Thankaraj, Karanbir Singh Chahal, Berk Calli, Saurabh Gupta, et al. Rb2: Robotic manipulation benchmarking with a twist. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021. 2

[20] Sudeep Dasari, Mohan Kumar Srirama, Unnat Jain, and Abhinav Gupta. An unbiased look at datasets for visuo-motor pre-training. In *Conference on Robot Learning*. PMLR, 2023. 1, 3, 5

[21] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 1, 2, 5

[22] Carl Doersch. Tutorial on variational autoencoders. *arXiv preprint arXiv:1606.05908*, 2016. 3

[23] D Eigen and R Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. corr, abs/1411.4734. *arXiv preprint arXiv:1411.4734*, 2014. 3

[24] Antonino Furnari and Giovanni Maria Farinella. Rolling-unrolling lstms for action anticipation from first-person video. *TPAMI*, 2020. 3

[25] Antonino Furnari, Sebastiano Battiato, Kristen Grauman, and Giovanni Maria Farinella. Next-active-object prediction from egocentric videos. *Journal of Visual Communication and Image Representation*, 2017. 3

[26] Jiyang Gao, Zhenheng Yang, and Ram Nevatia. Red: Reinforced encoder-decoder networks for action anticipation. *arXiv preprint arXiv:1707.04818*, 2017. 3

[27] Angeliki Giannou, Shashank Rajput, and Dimitris Papailiopoulos. The expressive power of tuning only the normalization layers. *arXiv preprint arXiv:2302.07937*, 2023. 4

[28] James Jerome Gibson. *The senses considered as perceptual systems*, volume 2. 3, 4

[29] JJ Gibson. The ecological approach to visual perception. *Houghton Mifflin Comp*, 1979. 3

[30] Rohit Girdhar and Kristen Grauman. Anticipative video transformer. In *ICCV*, 2021. 3

[31] Mohit Goyal, Sahil Modi, Rishabh Goyal, and Saurabh Gupta. Human hands as probes for interactive object understanding. In *CVPR*, 2022. 3

[32] 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. 16

[33] 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. 1, 2, 3, 5, 7

[34] Abhinav Gupta, Scott Satkin, Alexei A Efros, and Martial Hebert. From 3d scene geometry to human workspace. In *CVPR*, 2011. 3

[35] Abhinav Gupta, Adithyavairavan Murali, Dhiraj Prakashchand Gandhi, and Lerrel Pinto. Robot learning in homes: Improving generalization and reducing dataset bias. *Advances in neural information processing systems*, 31, 2018. 2

[36] Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. In *International Conference on Learning Representations*, 2020. 2

[37] M Hassanin, S Khan, and M Tahtali. Visual affordance and function understanding: a survey. arxiv. *arXiv preprint arXiv:1807.06775*, 2018. 3

[38] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015. URL http://arxiv.org/abs/1512.03385. 8

[39] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020. 3

[40] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16000–16009, 2022. 1, 2, 3, 5

[41] De-An Huang and Kris M Kitani. Action-reaction: Forecasting the dynamics of human interaction. In *ECCV*, 2014. 3

[42] Wenlong Huang, Chen Wang, Ruohan Zhang, Yunzhu Li, Jiajun Wu, and Li Fei-Fei. Voxposer: Composable 3d value maps for robotic manipulation with language models. *arXiv preprint arXiv:2307.05973*, 2023. 4

[43] Ashesh Jain, Avi Singh, Hema S Koppula, Shane Soh, and Ashutosh Saxena. Recurrent neural networks for driver activity anticipation via sensory-fusion architecture. In *ICRA*, 2016. 3

[44] Yuanchen Ju, Kaizhe Hu, Guowei Zhang, Gu Zhang, Mingrun Jiang, and Huazhe Xu. Robo-abc: Affordance generalization beyond categories via semantic correspondence for robot manipulation. *arXiv preprint arXiv:2401.07487*, 2024. 4

[45] Dmitry Kalashnikov, Alex Irpan, Peter Pastor, Julian Ibarz, Alexander Herzog, Eric Jang, Deirdre Quillen, Ethan Holly, Mrinal Kalakrishnan, Vincent Vanhoucke, et al. Qt-opt: Scalable deep reinforcement learning for vision-based robotic manipulation. *arXiv preprint arXiv:1806.10293*, 2018. 2

[46] Angjoo Kanazawa, Michael J. Black, David W. Jacobs, and Jitendra Malik. End-to-end recovery of human shape and pose. *CoRR*, abs/1712.06584, 2017. URL http://arxiv.org/abs/1712.06584. 3

[47] Aditya Kannan, Kenneth Shaw, Shikhar Bahl, Pragna Mannam, and Deepak Pathak. Deft: Dexterous fine-tuning for real-world hand policies. *CoRL*, 2023. 2

[48] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 4, 5, 16

[49] Hema S Koppula and Ashutosh Saxena. Anticipating human activities using object affordances for reactive robotic response. *TPAMI*, 2015. 3

[50] Tian Lan, Tsung-Chuan Chen, and Silvio Savarese. A hierarchical representation for future action prediction. In *ECCV*, 2014. 3

[51] Sergey Levine, Chelsea Finn, Trevor Darrell, and Pieter Abbeel. End-to-end training of deep visuomotor policies. *The Journal of Machine Learning Research*, 17(1):1334–1373, 2016. 1, 2

[52] Sergey Levine, Peter Pastor, Alex Krizhevsky, Julian Ibarz, and Deirdre Quillen. Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection. *The International journal of robotics research*, 37(4-5):421–436, 2018. 2

[53] Shaowei Liu, Subarna Tripathi, Somdeb Majumdar, and Xiaolong Wang. Joint hand motion and interaction hotspots prediction from egocentric videos. In *CVPR*, 2022. 3, 16

[54] Yunze Liu, Yun Liu, Che Jiang, Kangbo Lyu, Weikang Wan, Hao Shen, Boqiang Liang, Zhoujie Fu, He Wang, and Li Yi. Hoi4d: A 4d egocentric dataset for category-level human-object interaction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21013–21022, 2022. 3

[55] Camillo Lugaresi, Jiuqiang Tang, Hadon Nash, Chris McClanahan, Esha Uboweja, Michael Hays, Fan Zhang, Chuo-Ling Chang, Ming Guang Yong, Juhyun Lee, et al. Mediapipe: A framework for building perception pipelines. *arXiv preprint arXiv:1906.08172*, 2019. 3

[56] Yecheng Jason Ma, Shagun Sodhani, Dinesh Jayaraman, Osbert Bastani, Vikash Kumar, and Amy Zhang. Vip: Towards universal visual reward and representation via value-implicit pre-training. *arXiv preprint arXiv:2210.00030*, 2022. 2, 4

[57] Arjun Majumdar, Karmesh Yadav, Sergio Arnaud, Yecheng Jason Ma, Claire Chen, Sneha Silwal, Aryan Jain, Vincent-Pierre Berges, Pieter Abbeel, Jitendra Malik, et al. Where are we in the search for an artificial visual cortex for embodied intelligence? *arXiv preprint arXiv:2303.18240*, 2023. 1, 2, 3, 4, 5, 7, 16

[58] Priyanka Mandikal and Kristen Grauman. Dexvip: Learning dexterous grasping with human hand pose priors from video. In *Conference on Robot Learning*, pages 651–661. PMLR, 2022. 2

[59] Ajay Mandlekar, Danfei Xu, Josiah Wong, Soroush Nasiriany, Chen Wang, Rohun Kulkarni, Li Fei-Fei, Silvio Savarese, Yuke Zhu, and Roberto Martín-Martín. What matters in learning from offline human demonstrations for robot manipulation. In *Conference on Robot Learning (CoRL)*, 2021. 5, 9, 16

[60] Esteve Valls Mascaro, Hyemin Ahn, and Dongheui Lee. Intention-conditioned long-term human egocentric action forecasting@ ego4d challenge 2022. *arXiv preprint arXiv:2207.12080*, 2022. 3

[61] Austin Myers, Ching L Teo, Cornelia Fermüller, and Yiannis Aloimonos. Affordance detection of tool parts from geometric features. In *ICRA)*, 2015. 3

[62] Tushar Nagarajan, Christoph Feichtenhofer, and Kristen Grauman. Grounded human-object interaction hotspots from video. In *ICCV*, 2019. 3

[63] Ashvin V Nair, Vitchyr Pong, Murtaza Dalal, Shikhar Bahl, Steven Lin, and Sergey Levine. Visual reinforcement learning with imagined goals. *Advances in neural information processing systems*, 31, 2018. 2

[64] Suraj Nair, Aravind Rajeswaran, Vikash Kumar, Chelsea Finn, and Abhinav Gupta. R3m: A universal visual representation for robot manipulation. *arXiv preprint arXiv:2203.12601*, 2022. 1, 2, 3, 4, 5

[65] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. 1, 3, 5

[66] Jyothish Pari, Nur Muhammad, Sridhar Pandian Arunachalam, Lerrel Pinto, et al. The surprising effectiveness of representation learning for visual imitation. *arXiv preprint arXiv:2112.01511*, 2021. 4

[67] Lerrel Pinto and Abhinav Gupta. Supersizing self-supervision: Learning to grasp from 50k tries and 700 robot hours. In *2016 IEEE international conference on robotics and automation (ICRA)*, pages 3406–3413. IEEE, 2016. 1, 2

[68] Dean A Pomerleau. Alvinn: An autonomous land vehicle in a neural network. *Advances in neural information processing systems*, 1, 1988. 2, 5

[69] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. *CoRR*, abs/2103.00020, 2021. URL https://arxiv.org/abs/2103.00020. 5, 7

[70] Ilija Radosavovic, Tete Xiao, Stephen James, Pieter Abbeel, Jitendra Malik, and Trevor Darrell. Real-world robot learning with masked visual pre-training. *CoRL*, 2022. 1, 2, 4, 5

[71] Nicholas Rhinehart and Kris M Kitani. Learning action

maps of large environments via first-person vision. In *CVPR*, 2016. 3

[72] Yu Rong, Takaaki Shiratori, and Hanbyul Joo. Frankmocap: A monocular 3d whole-body pose estimation system via regression and integration. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, pages 1749–1759, October 2021. 2, 3, 4, 8

[73] Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 627–635. JMLR Workshop and Conference Proceedings, 2011. 5

[74] Anirban Roy and Sinisa Todorovic. A multi-scale cnn for affordance segmentation in rgb images. In *ECCV*, 2016. 3

[75] Abraham Savitzky and Marcel JE Golay. Smoothing and differentiation of data by simplified least squares procedures. *Analytical chemistry*, 36(8), 1964. 16

[76] Johann Sawatzky, Abhilash Srikantha, and Juergen Gall. Weakly supervised affordance detection. In *CVPR*, 2017. 3

[77] Stefan Schaal. Is imitation learning the route to humanoid robots? *Trends in cognitive sciences*, 3(6): 233–242, 1999. 5

[78] Stefan Schaal et al. Learning from demonstration. *Advances in neural information processing systems*, pages 1040–1046, 1997. 1

[79] Pierre Sermanet, Corey Lynch, Yevgen Chebotar, Jasmine Hsu, Eric Jang, Stefan Schaal, Sergey Levine, and Google Brain. Time-contrastive networks: Self-supervised learning from video. In *2018 IEEE international conference on robotics and automation (ICRA)*, pages 1134–1141. IEEE, 2018. 1

[80] Rutav M Shah and Vikash Kumar. Rrl: Resnet as representation for reinforcement learning. In *ICML*, 2021. 4

[81] Dandan Shan, Jiaqi Geng, Michelle Shu, and David Fouhey. Understanding human hands in contact at internet scale. In *CVPR*, 2020. 2, 3, 4, 5, 7, 8, 16

[82] Kenneth Shaw, Shikhar Bahl, and Deepak Pathak. Videodex: Learning dexterity from internet videos. In *CoRL*, 2022. 2

[83] Kenneth Shaw, Ananye Agarwal, and Deepak Pathak. Leap hand:low-cost, efficient, and anthropomorphic hand for robot learning. *RSS*, 2023. 8

[84] Mohit Shridhar, Lucas Manuelli, and Dieter Fox. Cliport: What and where pathways for robotic manipulation. In *CoRL*, 2022. 4

[85] Shuran Song, Andy Zeng, Johnny Lee, and Thomas Funkhouser. Grasping in the wild: Learning 6dof closed-loop grasping from low-cost demonstrations. *IEEE Robotics and Automation Letters*, 5(3):4978–4985, 2020. 2

[86] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018. 1

[87] Ruben Villegas, Jimei Yang, Seunghoon Hong, Xunyu Lin, and Honglak Lee. Decomposing motion and content for natural video sequence prediction. *arXiv preprint arXiv:1706.08033*, 2017. 3

[88] Carl Vondrick, Deniz Oktay, Hamed Pirsiavash, and Antonio Torralba. Predicting motivations of actions by leveraging text. In *CVPR*, 2016. 3

[89] Jianren Wang, Sudeep Dasari, Mohan Kumar Srirama, Shubham Tulsiani, and Abhinav Gupta. Manipulate by seeing: Creating manipulation controllers from pre-trained representations. 2023. 2

[90] Jiayi Wang, Franziska Mueller, Florian Bernard, Suzanne Sorli, Oleksandr Sotnychenko, Neng Qian, Miguel A Otaduy, Dan Casas, and Christian Theobalt. Rgb2hands: real-time tracking of 3d hand interactions from monocular rgb video. *ACM Transactions on Graphics (TOG)*, 39(6):1–16, 2020. 3

[91] William Whitney, Rajat Agarwal, Kyunghyun Cho, and Abhinav Gupta. Dynamics-aware embeddings. *arXiv preprint arXiv:1908.09357*, 2019. 2

[92] Tete Xiao, Ilija Radosavovic, Trevor Darrell, and Jitendra Malik. Masked visual pre-training for motor control. *arXiv preprint arXiv:2203.06173*, 2022. 7

[93] Yufei Ye, Xueting Li, Abhinav Gupta, Shalini De Mello, Stan Birchfield, Jiaming Song, Shubham Tulsiani, and Sifei Liu. Affordance diffusion: Synthesizing hand-object interactions. In *CVPR*, 2023. 3

[94] Tianhe Yu, Deirdre Quillen, Zhanpeng He, R. Julian, Karol Hausman, Chelsea Finn, and S. Levine. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *CoRL*, 2019. 16

[95] Andy Zeng, Pete Florence, Jonathan Tompson, Stefan Welker, Jonathan Chien, Maria Attarian, Travis Armstrong, Ivan Krasin, Dan Duong, Vikas Sindhwani, and Johnny Lee. Transporter networks: Rearranging the visual world for robotic manipulation. *CoRL*, 2020. 4

[96] Xiaohua Zhai, Alexander Kolesnikov, Neil Houlsby, and Lucas Beyer. Scaling vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12104–12113, 2022. 3

[97] Bingchen Zhao, Haoqin Tu, Chen Wei, Jieru Mei, and Cihang Xie. Tuning layernorm in attention: Towards efficient multi-modal llm finetuning. *arXiv preprint arXiv:2312.11420*, 2023. 4

[98] Yibiao Zhao and Song-Chun Zhu. Scene parsing by integrating function, geometry and appearance models. In *CVPR*, 2013. 3

[99] Huiyu Zhou, Yuan Yuan, and Chunmei Shi. Object tracking using sift features and mean shift. *Computer vision and image understanding*, 113(3):345–352, 2009. 3

[100] Wenxuan Zhou, Bowen Jiang, Fan Yang, Chris Paxton, and David Held. Hacman: Learning hybrid actor-critic maps for 6d non-prehensile manipulation. In *7th Annual Conference on Robot Learning*, 2023. 4

[101] Xingyi Zhou, Rohit Girdhar, Armand Joulin, Phillip Krähenbühl, and Ishan Misra. Detecting twenty-thousand classes using image-level supervision. *arXiv preprint arXiv:2201.02605*, 2022. 8

[102] Yixin Zhu, Chenfanfu Jiang, Yibiao Zhao, Demetri Terzopoulos, and Song-Chun Zhu. Inferring forces and learning human utilities from videos. In *CVPR*, 2016. 3

[103] Yuke Zhu, Alireza Fathi, and Li Fei-Fei. Reasoning about object affordances in a knowledge base representation. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part II 13*, pages 408–424. Springer, 2014. 3

APPENDIX

A. Robot Controller Details

Franka: We use a 7-DOF Franka Emika Panda robot arm with a parallel gripper, operating in delta end-effector action space. We use a VR-based teleoperation system to collect expert demos on Franka.

xArm: We use a 6-DOF xArm robot arm with a parallel gripper, operating in absolute end-effector action space. We use an off-the-shelf hand tracking system to collect expert demos on xArm.

Dexterous Hand: We use a 6-DOF xArm robot arm with a custom dexterous hand, operating in absolute end-effector space.

For each task, the expert gets to practice for 30 to 60 mins before collecting the demonstrations. We collect 50 expert demonstrations for each of the tasks.

B. Front Cam: Full Task Performance Breakdown

TABLE V: Front Cam Performance Breakdown

Initial Representation	Method	Toasting	Pouring	Stacking	Avg. (Real)
Ego4D	Baseline	0.58	0.36	0.60	0.51
	Ours	**0.83**	**0.60**	**0.77**	**0.73**
ImageNet	Baseline	0.53	0.45	0.47	0.48
	Ours	**0.75**	**0.48**	**0.70**	**0.64**
CLIP	Baseline	0.28	0.33	0.33	0.32
	Ours	**0.50**	**0.39**	**0.57**	**0.48**
DINO	Baseline	0.38	0.32	0.40	0.37
	Ours	**0.67**	**0.57**	**0.50**	**0.58**
MVP	Baseline	0.27	0.41	0.47	0.38
	Ours	**0.73**	**0.44**	**0.63**	**0.60**
VC1	Baseline	0.52	0.33	0.57	0.47
	Ours	**0.83**	**0.34**	**0.53**	**0.57**

We observe that HRP (Ours) consistently boosts the performance across all three tasks for the front cam.

C. Ego Cam: Full Task Performance Breakdown

TABLE VI: Ego Cam Performance Breakdown

Initial Representation	Method	Toasting	Pouring	Stacking	Avg. (Real)
Ego4D	Baseline	0.2	0.12	0.3	0.21
	Ours	0.2	**0.22**	**0.45**	**0.29**
ImageNet	Baseline	0.3	0.3	0.45	0.35
	Ours	**0.6**	**0.48**	**0.7**	**0.59**
CLIP	Baseline	0.2	0	0	0.07
	Ours	**0.275**	**0.02**	0	**0.1**
DINO	Baseline	0.35	0.32	0.3	0.32
	Ours	**0.45**	**0.7**	**0.55**	**0.57**
MVP	Baseline	0.175	0.32	0.45	0.32
	Ours	**0.3**	**0.4**	**0.65**	**0.45**
VC1	Baseline	0.5	0.28	0.4	0.39
	Ours	**0.55**	**0.6**	**0.65**	**0.6**

We also find that HRP (Ours) consistently boosts the performance across all three tasks for the ego camera.

D. Ablation Breakdown

TABLE VII: Fine-Tuning Ablation Breakdown

Initial Representation	Finetuning Scheme	Toasting	Pouring	Stacking	Avg. (Real)
Ego4D	All Weights	**0.92**	0.51	0.77	0.73
	LayerNorm (Ours)	0.83	**0.60**	0.77	0.73
ImageNet	All Weights	**0.82**	0.34	0.63	0.60
	LayerNorm (Ours)	0.75	**0.48**	**0.70**	**0.64**
CLIP	All Weights	0.23	0.27	0.13	0.21
	LayerNorm (Ours)	**0.50**	**0.39**	**0.57**	**0.48**
DINO	All Weights	0.57	0.39	0.40	0.45
	LayerNorm (Ours)	**0.67**	**0.57**	**0.50**	**0.58**
MVP	All Weights	0.45	0.39	0.47	0.43
	LayerNorm (Ours)	**0.73**	**0.44**	**0.63**	**0.60**
VC1	All Weights	0.52	0.41	0.47	0.47
	LayerNorm (Ours)	**0.83**	**0.34**	**0.53**	**0.57**

TABLE VIII: Loss Ablation Performance Breakdown

Initial Representation	Condition	Toasting	Pouring	Stacking	Avg. (Real)
Ego4D	No Contact	0.65	0.34	0.5	0.50
	No Object	0.425	0.42	0.3	0.38
	No Hand	0.625	0.48	0.4	0.50
	Ours	**0.9**	**0.66**	**0.75**	**0.77**
Imagenet	No Contact	0.625	**0.64**	0.7	0.66
	No Object	0.525	0.52	0.55	0.53
	No Hand	0.525	0.3	**0.7**	0.51
	Ours	**0.8**	0.62	**0.7**	**0.71**
VC-1	No Contact	**0.625**	**0.48**	0.75	**0.62**
	No Object	0.225	0.38	0.65	0.42
	No Hand	0.5	0.44	0.4	0.45
	Ours	0.525	0.44	**0.8**	0.59

Note: do not compare numbers between Table VIII and the other tables. The loss ablation experiments were run on a separate day, so all numbers were re-ran on that day. This was done to ensure a proper A/B comparison between the all methods in this table.

E. Loss Weighting Sweep

We swept through a range of weights for each of the losses to narrow down on a particular set of loss weights for HRP (presented in Table IX). These were based on relative orders of magnitude of the ground truth labels in the dataset. We empirically saw that increasing the loss weights by more than 0.5 negatively affected performance and led to collapse.

TABLE IX: We present the different affordance loss weights we ran sweeps on.

Exp	Loss Weights
HRP	$\lambda_{obj} = 0.05$, $\lambda_{ct} = 0.005$, $\lambda_{hand} = 0.5$
Drop Contact Only	$\lambda_{obj} = 0.05$, $\lambda_{ct} = 0$, $\lambda_{hand} = 0.5$
Drop Object Only	$\lambda_{obj} = 0$, $\lambda_{ct} = 0.005$, $\lambda_{hand} = 0.5$
Drop Hand Only	$\lambda_{obj} = 0.05$, $\lambda_{ct} = 0.005$, $\lambda_{hand} = 0$

F. Data Pipeline Description

To obtain human data, we first extract video clips from Ego4D [32]. Our dataset contains approximately 1200 videos. Each video is broken down semantically into smaller clips by human annotators (as a part of the Ego4D). Our clips are between 1 and 30 seconds. For a given clip, we pass every frame through the 100 DOH model [81], which gives us hand object contact information. These are $\{h_l, h_r, o_l, o_r, c_l, c_r\}$. h are the hand bounding boxes, o are the object bounding boxes (which are in contact with the hand). c are contact variable (i.e. fixed, portable, self or no contact). We only look at contacts with fixed and portable. r or l represents the left or right hand. Active object and hand trajectories used for our representations are directly used. For contact points, it is assumed that at the start of the clip there is no contact, from where we find the frame of first contact t. Since per-frame predictions are noisy, we run a filter [75] over the predictions. From the contact frame, we obtain the hand-bounding box h and object bounding o. Contact points are computed in the intersection of h and o, and the exterior of the hand. This exterior is obtained via skin segmentation (similar to [2, 53]. These contacts can then be projected to previous frames in the clips by the homography matrix H_t obtained via SIFT features.

G. Behavior Cloning Hyper-Parameters

We list the hyper-paramaters that we used for policy training using behavior-cloning in this section. As shown in Figure 4, we pass an image through the learned HRP visual representation to obtain a 768-dimensional latent vector. This latent vector is passed through a two-layer MLP with (512, 512) hidden layer dimensions. To the output of the MLP we apply RELU activation along with dropout regularization with prob=0.2 to estimate the mean (μ), the mixing parameters (ϕ), and the standard deviation (σ) of a Gaussian Mixture Model (GMM) distribution with 5 modes.

We choose GMM model based on prior work [59] that showed its crucial role in increasing BC performance. We use ADAM optimizer [48] with the learning rate set to 1e-4, l2 weight decay also set to 1e-4. We train policy for 50K iterations. We also apply data augmentation (random crop and random blur) for the input images. We use the same set of hyper-parameters for both the real-world and the simulation tasks.

H. Simulation Results

For simulation tasks, we choose 5 tasks from the *Meta-world* [94] benchmark namely: BinPick, ButtonPress, Hammering, Drawer Opening, and Assembly. This benchmark is extensively used by the robot learning community. We used the same camera viewpoint, object sets, and expert demonstrations as used by prior work [57]. We report the average performance on all 5 tasks in table X.

TABLE X: Sim Performance

Initial Representation	Method	MetaWorld Avg Performance
Ego4D	Baseline	**0.656**
	Ours	0.580
ImageNet	Baseline	0.556
	Ours	**0.664**
CLIP	Baseline	**0.444**
	Ours	0.408
DINO	Baseline	0.660
	Ours	**0.664**
MVP	Baseline	0.592
	Ours	**0.640**
VC1	Baseline	0.576
	Ours	**0.648**

I. Evaluation

For each task, we run around 50 trials (per model), at various initial poses (for objects) and with different variations in objects. In every task, about half the trials are from the training distribution and half are from the test. The differences in objects include different colors, shapes, and even semantic differences: for example in the toasting task, plush toys were tested instead of the vegetables used to train. Cups or bowls were tested, instead of mugs that were used to train the pouring task, etc.

Lighting is not controlled between train and test. We did try to run all baselines and methods as closely together as possible to avoid any confounding factors: i.e. for every trial, we ran all the baselines and our method together. Across trials, we allowed for variation in lighting conditions.

The results presented in the paper are the average of the successes, on a scale from 0 to 1. We present the criteria for success in each task:

- **Stacking:** 1 if the robot correctly picks and stacks the red block, and 0.5 if the red block is unstably placed on the green block.
- **Pouring:** The fraction of candies, out of 5, successfully poured (e.g., $2/5$ candies poured $\rightarrow 0.4$ success).
- **Toasting:** 1 if the whole task is completed, and 0.5 successful if the robot only successfully places the object.
- **Pot on Stove:** 1 if the food is correctly placed in the pot.
- **Hand Lift Cup** 1 if the cup is stably grasped and picked.

We also compute the standard error for these trials and show that as our confidence in Tables 1-3, and as an error bar in Figures 5-7.