Structured World Models from Human Videos

Russell Mendonca*   Shikhar Bahl*   Deepak Pathak

Carnegie Mellon University

Abstract—In this paper, we tackle the problem of learning complex, general behaviors directly in the real world. We propose an approach for robots to efficiently learn manipulation skills using only a handful of real-world interaction trajectories from *many different settings*. Inspired by the success of learning from large-scale datasets in the fields of computer vision and natural language, our belief is that in order to efficiently learn, a robot must be able to leverage internet-scale, human video data. Humans interact with the world in many interesting ways, which can allow a robot to not only build an understanding of useful *actions* and *affordances* but also how these actions affect the world for manipulation. Our approach builds a structured, human-centric action space grounded in visual affordances learned from human videos. Further, we train a *world model* on human videos and fine-tune on a small amount of robot interaction data without any task supervision. We show that this approach of affordance-space world models enables different robots to learn various manipulation skills in complex settings, in under 30 minutes of interaction. Videos can be found at https://human-world-model.github.io
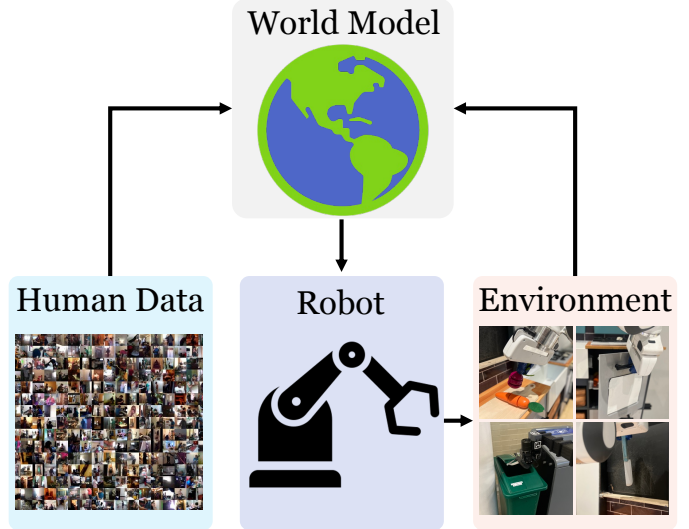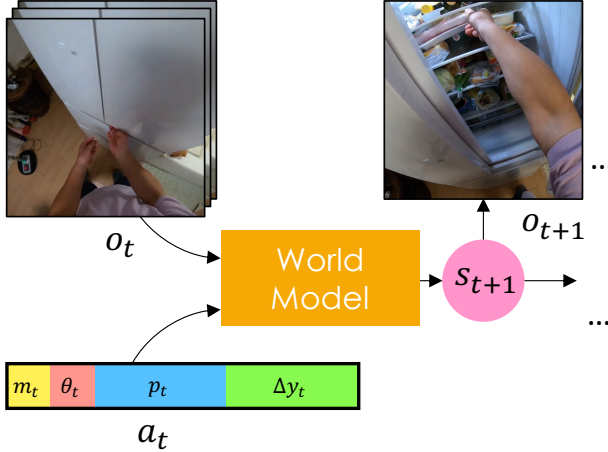
Fig. 1: We present SWIM, an approach for learning manipulation tasks in the real world with only a handful of trajectories and only 30 min of real-world sampling.

I. INTRODUCTION

A truly useful home robot should be general purpose, able to perform arbitrary manipulation tasks, and *get better* at performing new ones as it obtains more experience. How can we build such generalist agents? The current paradigm in robot learning is to train a policy, in simulation or directly in the real world, with engineered rewards or demonstrations directly constructed for the environment. While this has shown successes in lab-based tasks [29, 51, 36], learning is heavily dependent on the structure of the reward. This is not scalable as it is very challenging to *transfer* to new tasks, with different objectives. Often, it is also difficult to obtain ground truth objectives for a task in the real world. For a robot to succeed in the wild, it must not only learn many tasks at a time but also *get better* as it sees more data. How can we build an agent that can take advantage of large-scale experience and multi-task data?

We aim to build *world models* to tackle this challenge. One key observation is that there is commonality between many tasks performed by humans on a daily basis. Even across diverse settings, the environment dynamics and physics share a similar structure. Learning a single *joint* world model, that predicts the future consequences of actions across diverse tasks can thus enable agents to extract this shared structure.

While world models enable learning from inter-task data, they require action information to make predictions about the future. Furthermore, for planning in an environment, the actions need to be relevant to the particular robot. Consequently, world models for robotics have mostly been trained only on data collected directly by a robot [16, 72, 34, 13, 17]. However, the quantity of this data is limited, which is very expensive and cumbersome to collect in the real world. Thus, the benefits of using large datasets as seen in other machine learning areas such as computer vision and language [7, 53] have not been realized for robotics, as no such dataset exists for robotics. However, there is an abundance of *human videos*, performing a very large set of tasks, on the internet. Is there a way to leverage this abundant data to learn world models for robotics, that will enable the robot to predict the consequences of its *actions* in any environment, enabling general-purpose learning?

Due to the large morphology gap between robots and humans, it is challenging to obtain actions from human videos. Thus, previous approaches have mostly focused on learning visual representation features [45, 74] from observations alone. Using internet human videos to train robots requires us to define an action space that is applicable both in the human video domain and for robots. Consider the task of picking up a mug. To perform this task, the low-level signals sent to a person's arm compared to that of a robot would be completely different, and so predictive models in low-level joint space will not transfer well. If the action space instead required predicting the target pose and orientation of the mug handle, with low-level control abstracted away, then target poses used by humans could be utilized by robots as well. Thus, we learn *high-level* structured action spaces that are morphology invariant. For manipulation tasks, predicting a grasp location and post-grasp waypoints is an effective action space since it encourages object interaction. We can train visual affordance networks that
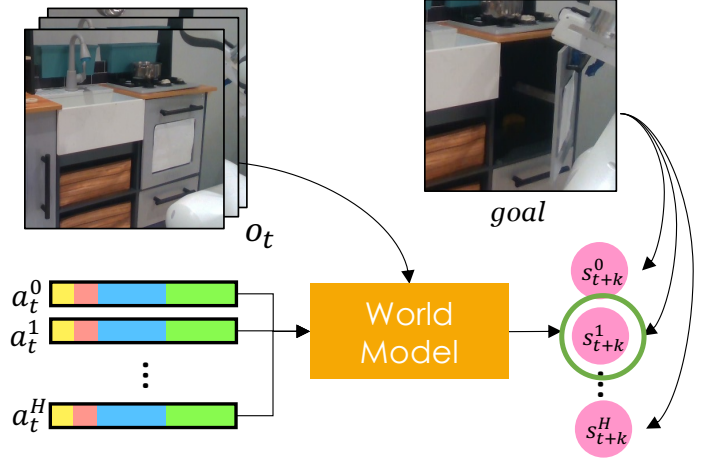
Fig. 2: Overview of SWIM. We first pre-train the world model on a large set of human videos. We finetune this on many robot tasks, in an unsupervised manner, and deploy at test-time in the real world to achieve a given goal. Videos can be found at https://human-world-model.github.io

produce these locations given videos leveraging techniques in computer vision [37, 19, 60, 43].

In this paper we propose **S**tructured **W**orld **M**odels for **I**ntentionality (**SWIM**), which utilizes large-scale internet data to train world models for robotics using structured action spaces. Training the world model in the common high-level structured action space allows it to capture how human hands interact with objects when trying to grasp and manipulate them. This model can then be fine-tuned for robotics settings with only a handful of real-world interaction trajectories. This is because the world model can leverage the actionable representations it was pre-trained with due to the similarity in how the human hands from video data and robot grippers interact with the world. Furthermore, these interaction trajectories for fine-tuning do not require any task supervision and can be obtained simply by executing the visual affordance actions. We note that both pre-training on human videos and finetuning the world model on robot data do not make any assumption on rewards, and this *unsupervised* setting allows us to utilize data relevant for different tasks. This allows the robot to train a *single* world model on all the data, thus enabling us to train generalist agents. In our experiments, we show that we can train such joint world models through two distinct robot systems operating in real-world environments. Finally, we can deploy the fine-tuned world model to perform tasks of interest by specifying a goal image. The world model then plans in the affordance action space to find a sequence of actions to manipulate objects as required by the task.

To summarize, SWIM trains world models for robot control and consists of three stages: 1) Leveraging internet videos of human interactions for pre-training the model, 2) Finetuning the model to the robot setting using reward-free data, 3) Planning through the model to achieve goals. We evaluate this framework on two robot systems – a Franka Arm, and a Hello Stretch

robot. SWIM is able to learn directly, is trained on data from multiple settings and gets better with data from more tasks. We perform a large-scale study across multiple environments and robots and find that SWIM achieves higher success ($\sim$ 2X) than prior approaches while being very sample efficient, requiring less than 30 minutes of real-world interaction data.

II. RELATED WORK

Efficient Real World Robot Learning   Deploying learning-driven approaches on hardware is challenging and requires either large engineering efforts to collect demonstrations [6, 28], many hours of autonomous interactions [29, 30], or simulations [1, 69, 33]. A major constraint of continuous control is the extremely large action space. Prior methods have focused on reducing this search space by using skills or options in a hierarchical manner [10, 47, 3, 68, 12, 46, 12], physical inductive biases [40, 50, 32, 67, 31, 4, 39]. It is also possible to visually ground the action space, by parameterizing each observed location by a 2D [78, 77, 64, 27] or 3D [65] action. While these can speed up learning, we find that our structured action space, based on human-centric visual affordances allows us to not only perform manipulation efficiently but also leverage out-of-domain human/internet videos.

Model-based learning   To tackle the sample efficiency problem in robot learning, prior methods have proposed learning dynamics models, which can later be used to optimize the policy [14, 15, 22, 9, 41, 42]. Such approaches mostly operate and learn in state space, which tends to be low dimensional. In order to deal with the highly complex visual observations from real-world settings, prior methods have used *World Models* [20], which capture dynamics of the *agent* and its *environment*. Such models can plan in image space [18, 16] or fully in imagination space [71, 23, 21, 35]. Such world models have been shown to be useful on a large set of tasks [55], including on hardware [73]. In this paper, we argue that world
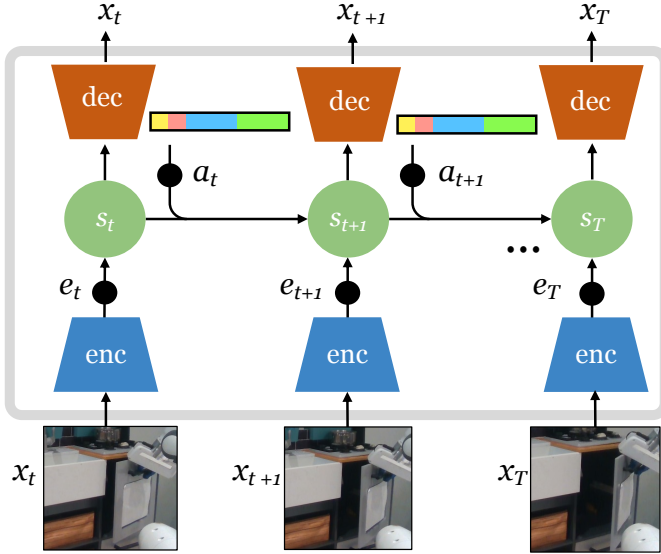
Fig. 3: World Model Training: Images and actions are encoded into a learned feature space that has temporal structure, following the approach from Hafner et al. [24]

models can be helpful in modeling the real world, especially if they can understand how the environment will behave at a high level and model the intentions of the agent.

Visual and Action Pre-Training for Robotics   In order to learn more generalizable and actionable representations, prior methods have learned visual encoders from large-scale human video data, either via video-language contrastive learning [45] or through inpainting masked patches [74, 54]. These representations have been shown to be useful for dynamics models as well [25]. Such approaches focus on the visual complexity of the world but do not encode any behavior information. Some works have incorporated low-level actions from human videos into the learning loop [38, 52, 2, 63], but these are fixed for a specific morphology and use a direct mapping to the robot. In contrast, our approach is able to learn a world model from human videos, incorporating action information, and works in multiple settings.

III. BACKGROUND AND PRELIMINARIES

World Models   These are used to learn a compact state space for control given high-dimensional observations like images. The learned states preserve temporal information, which enables effective prediction and planning [20, 58, 57]. In this work, we use the model structure and training procedure from Dreamer [22, 23, 24], which has the following components:

$$\text{encoder:}\quad e_t = \text{enc}_\phi(x_t) \qquad \text{posterior:}\quad p(s_t|s_{t-1}, a_{t-1}, e_t)$$
$$\text{dynamics:}\quad p(s_t|s_{t-1}, a_{t-1}) \qquad \text{decoders:}\quad p(x_t|s_t), p(r_t|s_t)$$
(1)

Here x_t, a_t, r_t denote the observation, action, and reward at time t, and s_t denotes the learned state space. Note that all these components are parameterized using neural networks. The model is trained by optimizing the ELBO as described in

Dreamer, where the learned features are trained to reconstruct images and rewards and are regularized with a dynamics prior. The reward head decoder is not trained if r_t is not provided. For more details, we refer the readers to Hafner et al. [24].

Hand-Object Interactions from Human Videos   In this paper, we leverage human videos to learn world models. Throughout the paper, we will refer to a set of visual affordances. These visual affordances comprise of the hand trajectory h_t in image space (normalized to a 0-1 range), and object locations (o_t). We obtain human hand-object information (h_t, o_t) for each frame using the 100 Days of Hands [60] detector model, trained on many hours of youtube videos. These can then be used to identify where on the object the hand makes contact p^g, and we sample the hand position from a later frame in the video to obtain p^{pg}. Here p^g and p^{pg} denote the grasp and post-grasp pixel respectively and specify the visual affordance space.

IV. WORLD MODELS FROM HUMAN VIDEOS

A. Visual Affordances as Actions

One of the key challenges is defining what the actions should be from human videos, most of which just contain image observations. Action information is essential for world models since they are required to learn dynamics and make predictions about the future. Furthermore, we need to define actions in a manner that is *transferable* from the human video domain to robot deployment settings. Following previous work that studies human-to-robot transfer for manipulation [5, 61, 8, 75, 66, 63, 62, 59, 76], we use the human hand motion in the videos to inform the action space. This is because we are focused on performing manipulation tasks, and the manner in which humans interact with objects using their hands contains useful information that can be transferred to robot end-effectors.

Structured Actions from Videos   We note that the videos of humans interacting with objects often consist of the hand moving to a point on the object, performing a grasp, and then manipulating the object. After obtaining the grasp pixel p^g and post-grasp pixels p^{pg}, using computer vision techniques similar to [19, 37, 43] from the video clip, we use these to train $\mathcal{G}_\phi$, which distills these labels into a neural network model conditioned on the first frame of the video clip. This model thus learns affordances associated with objects in the scene, by modeling how humans interact with them.

Transfer to Robot Scene   When dealing with 2D images, there is an inherent ambiguity regarding depth, which is required to map to a 3D point. To overcome this, we utilize the ability of most robots to be equipped with depth cameras to generate physically plausible interactions. From the RGBD observation, we can obtain the depth d_t^g at the image-space point p_t^g, and also sample the post-grasp depth d_t^{pg} within some range of the environment surface. This can then be projected into 3D coordinates in the robot frame, using hand-eye calibration, and the robot can attempt to grasp and manipulate objects by moving its gripper to these locations. The affordance action at time t can thus be expressed as $u_t = [p_t, d_t]$, where d_t is the depth corresponding to pixel p_t.
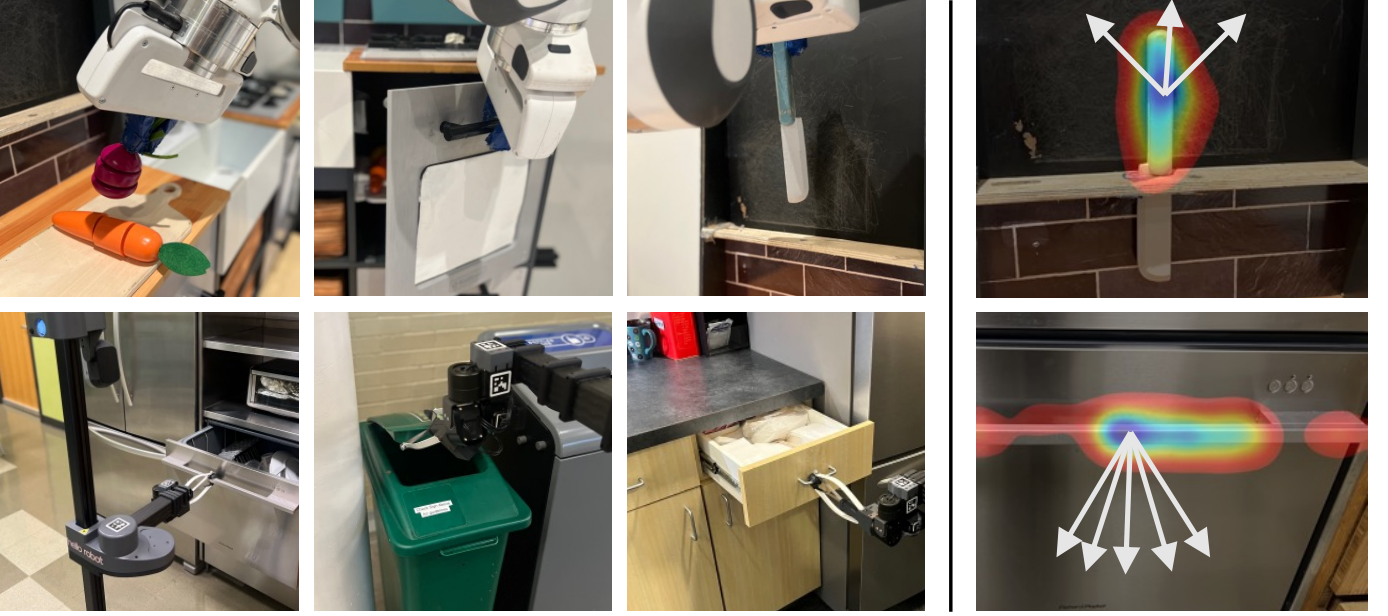
Fig. 4: We evaluate SWIM on six different real-world manipulation tasks on two different robot systems (shown on the left). On the right, we show a sample of the visual affordances from the visual affordance model $\mathcal{G}_\psi$

Hybrid Action Space While visual affordances help structure the action space to increase likelihood of useful manipulation and allow us to learn from human video, they impose restrictions on the full space of end-effector motion. Hence, we adopt a hybrid action space that has the option to execute both the aforementioned visual affordance, as well as arbitrary end-effector Cartesian actions. We append a mode index to denote which type of action should be executed. This enables the robot to benefit both from the structured pixel-space visual affordance actions and the pre-training data in mode (m) 0, and make adjustments using arbitrary end-effector delta actions in mode 1. An action can be described by the following:

$$a_t = [m_t, \theta_t, u_t, \Delta y_t] \qquad (2)$$

Here m_t denotes the mode, θ_t is the rotation of the gripper, u_t is the image-space action ($u_t = [p_t, d_t]$, where p_t are pixel coordinates in the image and d_t is depth), and Δy_t is the Cartesian end-effector action. At a particular timestep, only one out of the image action and Cartesian actions can be executed. If $m_t = 0$, this corresponds to the affordance mode, and so p_t is executed. If $m_t = 1$, then the robot is operating in the Cartesian control mode, and Δy_t is used. Due to our hybrid action space, we can seamlessly switch between training with the visual affordance and Cartesian end-effector action spaces. This allows the robot to leverage the structure from human video and also make adjustments if required using Cartesian actions which are useful for fine-grain control.

B. Structured Affordance-based World Models for Robotics

The overall approach is outlined in Alg. 2. We now describe each of the three phases - 1) World model pre-training on

Algorithm 1 Human Video Data Training

Require: Human Video Dataset $\mathcal{D}$
1: **initialize:** World model $\mathcal{W}$, Affordance model $\mathcal{G}$
2: Process $\mathcal{D}$ into video clips $C^0, ... C^T$
3: Obtain grasp p^g and post grasp p^{pg} pixels for each C^k.
4: Create actions a_t using eq. 2, with mode $m_t = 0$, and randomly sampling depth d_t and rotation θ_t
5: Train $\mathcal{G}_\phi(a^g, a^{pg}|I_0^k)$, where I_0^k is the first frame of C^k
6: Train $\mathcal{W}$ on trajectory sequences $\{(I_0^k, a^g, I_{t1}^k, a^{pg}, I_{t2}^k)\}$
7: **return** $\mathcal{W}, \mathcal{G}$

---

human videos, 2) Unsupervised finetuning with robot data, and 3) Robot deployment to perform a task given a goal image.

Training from Passive Human Videos We first use a large set of human videos, obtained from Epic-Kitchens [11] to both train the world model $\mathcal{W}$, and obtain the visual affordance model $\mathcal{G}_\phi$. This dataset includes around 50k egocentric videos of people performing various manipulation tasks in kitchens. We first process this dataset into a set of short video clips (around 3 seconds). After obtaining the grasp pixel p^g and post-grasp pixels p^{pg} from the video clip, we convert them to our action space (specified in eq. 2), and train $\mathcal{G}_\phi$, as previously described in section IV-A. For video clip k, let I_t^k denote an image frame from the clip at time t. We collect images $I_{t_1}^k$ and $I_{t_2}^k$, where t_1 is the time of the grasp, and t_2 is when the hand is at p^{pg}. $\mathcal{W}$ is then trained on the trajectory sequences:

$$\{I_0^k, a^g I_{t_1}^k, a^{pg} I_{t_2}^k\} \qquad (3)$$

This procedure is outlined in Alg. 1. As described in section IV-A, there are two modes for the actions - either in pixel

(a) World Model pre-training reconstructions

(b) Imagination rollouts at deployment

Fig. 5: a) World Model pre-training reconstructions on Epic-Kitchens dataset [11]. b) Model imagination rollouts for high-reward trajectories. We can see that SWIM can imagine plausible and successful trajectories, for both human and robot data. The first image (highlighted in red) is the original observation by the robot.

Algorithm 2 Overview of SWIM

1: Get $\mathcal{W}, \mathcal{G}$ = Human Video Data **Pre-Training** (Alg. 1)
2: **Finetuning**: Query $\mathcal{G}$ for N_0 iterations to collect robot dataset $\mathcal{R}_\mathcal{D}$ to train $\mathcal{W}$.
3: **Task Deployment:** (Given goal I_g)
4: Rank trajs in $\mathcal{R}_\mathcal{D}$ using I_g. Fit GMM g.
5: **for** traj 1:K **do**
6:    Query N proposals from $\mathcal{G}$, $\{a^g, a^{pg}\}_{1..N}$
7:    Query M proposals from g
8:    Select best proposal using CEM through $\mathcal{W}$
9:    Execute on the robot to reach I_g
10: **end for**

---

space or end-effector space. In order to train on human videos, we consistently set $m_t = 0$ and thus use the image space actions. Since image depth and robot rotation information are not present in the video, we randomly sample values for these components. We include visualizations of the world model predictions on the passive data in Figure 5, and see that the model is able to capture the structure of the data.

Finetuning with Robot Data To use the world model $\mathcal{W}$ for control, we need to collect some in-domain robot data for finetuning. We do so by running the visual affordance model $\mathcal{G}$ to collect a robot dataset R_D, which is then used to train $\mathcal{W}$. We emphasize that this step does not require *any* supervision in the form of task rewards or goals. Hence, we can collect data from diverse tasks in the finetuning step. We see in Fig. 8 that SWIM enables the world model to pick up on the salient features of the robot environment very quickly as compared to models that do not use pre-training on human videos.

Task Deployment After the world model has been fine-tuned on robot domain data, it can be used to perform tasks specified through goal images. The procedure for doing so is outlined in the Task Deployment section in Alg. 2. We collect two sets of action proposals. The first set is obtained by querying the visual affordance model $\mathcal{G}$ on the scene. We also want to leverage our knowledge of trajectories in $\mathcal{R}_\mathcal{D}$ that reach states close to the goal. For this, we create a second set of proposals by fitting a Gaussian Mixture Model to the top

trajectories in $\mathcal{R}_\mathcal{D}$ and sampling from it. We then use the world model to optimize for an action sequence using the standard CEM approach [56], where the initial set of plans is set to be the combined set of action proposals. Ranking the trajectories in $\mathcal{R}_\mathcal{D}$ and running CEM requires rewards, and we can obtain this by measuring the distance to the goal in the world model feature space:

$$r_t = \text{cosine}(f_\mathcal{W}(I_g), f_t)$$

where f_t is the world model feature, and $f_\mathcal{W}$ is the learned feature space of the model. For ranking trajectories in $\mathcal{R}_\mathcal{D}$, $f_t = f_\mathcal{W}(I_k)$ for image k in the dataset. For planning, f_t corresponds to the predicted feature state. In our experiments, we use cosine distance to goal in the feature space from Nair et al. [45] to provide reward for model-free baselines, since they do not have a model, and so we also add this term to our reward by training a reward prediction head to get feature space [45] distance to goal from f_t. In our experiments we run an ablation where we use only the world model feature space, and find that performance for our approach is about the same.

V. EXPERIMENTAL SETUP

A. Environments

Our real-world system consists of two different robots, evaluated over six tasks. Firstly, we use the Franka Emika arm, with end-effector control. This robot acts in a play kitchen environment with multiple tasks that mimic a real kitchen. Specifically, the robot needs to open a cabinet, pick up one of two toy vegetables from the counter and lift a knife from a holder. Note that the knife task is very challenging as it requires fine-grained control from the robot. In order to test SWIM in the wild we also deploy it on a mobile manipulator, the Stretch RE-1 from Hello-Robot. This is a collaborative robot designed with an axis-aligned set of joints and has suction cups as fingertips. We run this robot in real-world kitchens to perform different tasks, including opening a dishwasher, pulling out a drawer and opening a garbage can. The garbage can task is challenging as the area for the robot to grasp onto is quite small. We show images of the environments in Figure 4.

	Cabinet	Veg	Knife	Drawer	Dishwasher	Can	Average
No world model:							
`BC-Affordance`	0.32	0.48	0.16	0.56	0.20	0.44	0.36
`BC-Pix`	0.16	0.40	0.00	0.24	0.08	0.12	0.17
No human-centric affordance-based actions:							
`MBRL-single` [24]	0.00	0.28	0.20	0.00	0.04	0.00	0.09
`MBRL-Pix-single`	0.52	0.36	0.16	0.00	0.04	0.04	0.19
No pre-training from human videos:							
`MBRL-Affordance-single`	0.68	0.16	0.40	0.84	0.20	0.36	0.44
`MBRL-Affordance-joint`	0.12	0.36	0.36	0.08	0.20	0.04	0.19
`SWIM`	0.84	0.76	**0.72**	0.92	**0.84**	**0.68**	**0.79**
`SWIM-single`	**0.88**	**0.80**	0.60	**0.96**	0.68	0.56	0.75

TABLE I: Success rates of SWIM and baselines on six different manipulation tasks, over 25 trials.

B. Baselines and Ablations

In order to compare different aspects of SWIM, we run an extensive of baselines and ablations. All world-model-based approaches directly use code from Dreamer [24].

- `MBRL-Affordance`: An important contribution of SWIM is pre-training on human videos. This baseline is similar to SWIM but does not use any human video pre-training, allowing us to test our hypothesis that using human video is important for learning a generalizable world model.
- `MBRL-Pix`: Secondly, we would like to test how much the affordance action space helps the robot. This approach uses the same world model control procedure as SWIM, but does not sample actions using the visual affordance, $\mathcal{G}_\psi$. Instead grasp and post grasp locations are randomly sampled from an image crop around the object.
- `MBRL`: This baseline further removes structure from the action space, and only uses cartesian end-effector actions, without any pixel-space structure, thus $m_t = 1$ (described in section IV-A) for every timestep t. In order to help with sample efficiency, we use a simple heuristic to bootstrap this baseline: we initialize the robot each episode near the center of the detected object, using Detic [79], a state-of-the-art object detector.
- `BC-Affordance`: We would like to test if using world models is critical to performance, or if a simple behavior cloning approach can be effective. This baseline employs a filtered-behavior cloning [48, 49] strategy, in which the top trajectories based on reward (in our case distance to goal) are selected. Since there is no learned world model, we use distance in the feature space from the R3M model [45]. After selecting the top trajectories, we fit a gaussian mixture model and sample from it to obtain actions. These are in the same visual affordance action space used by our approach.
- `BC-Pix`: Uses behavior cloning in the same way as

`BC-Affordance`. The only difference is the action space - this approach randomly samples locations and does not use $\mathcal{G}_\psi$ to obtain actions.

C. Implementation details

Human Video Data Pre-trainig In order to pre-train the world model on human videos, we use the Epic-Kitchens [11] dataset. The dataset is divided into many small clips of humans performing semantic actions. We use the 100 Days of Hands [60] detector to find when an object has been grasped and find post grasp waypoints. Around 55K such clips are used to train the world model. Since we do not have depth or 3D information available, we randomly sample θ_t and the depth component of the image space action p_t.

Affordance Model We show some qualitative examples of the affordances of the human-affordance model ($\mathcal{G}_\psi$) we use in Figure 4. This model has a UNet style encoder-decoder architecture, with a ResNet18 [26] encoder. The final output of the model is h_t and g_t, where g_t is a set of keypoints obtained from a spatial softmax over the network's heatmap outputs, representing the grasp point, and h_t is the post-grasp trajectory of the detected hand.

World Model We use the world model from Hafner et al. [24]. However, in order to handle high-dimensional image inputs we employ NVAE [70] as a stronger visual encoder. While not necessary to train the reward model q_r when finetuning, we empirically found that it added stability to the filtering setup a test time. We leave distilling the latent features into a neural distance function as future work.

Robot Deployment Setup To capture videos and images we use an Intel Realsense D415, to get RGBD images. For each task we collect either 25 or 50 iterations of randomly sampled actions (in human-affordance, random image or Cartesian space), which takes about 30 minutes, finetuning the model on collected data. We obtain feature distance w.r.t. to image goals using the ResNet18 encoder from Nair et al. [44]. We sample around 2K action proposals and use the output of $\mathcal{W}_\phi$ to prune

Fig. 6: Comparison of SWIM and `MBRL-Affordance` for both the single task and jointly trained model. We see a large drop in success when removing pre-training on human videos, especially when dealing with diverse robot tasks.

these. The model outputs are then evaluated (25 times). A human measures success based on a pre-defined metric (i.e. the cabinet should be fully open, etc).

VI. RESULTS

In our experiments we ask the following questions (i) Can we train a *single world model jointly* with data coming from diverse tasks? (ii) Does training the world model on *human video* data help performance? (iii) How important is our structured action space, based on human visual *affordances*? (iv) Are *world models* beneficial for learning manipulation with a handful of samples? (v) Can our approach *continually improve* performance with iterative finetuning?

We present a detailed quantitative analysis of various approaches in Table I. We see that across environment settings and robots SWIM achieves an average success rate of about 80% when using joint models (trained separately for Franka and Hello robot tasks). We also observe strong performance when SWIM is trained on individual tasks, getting an average success of 75%, compared to the next best approaches which only get around 40% success.

Joint World Model A big benefit of SWIM is that it can deal with different sources of data. `SWIM-single` employs a model trained individually for each task. We see that overall the performance improves when sharing data, from the last two rows of Table I. This is likely because there are some similarities across tasks that the model is able to capture. We find this encouraging and hope to scale to more tasks in the future. Further, we see that for the best baseline, `MBRL-Affordance-single`, using all the data jointly to train the world model leads to a major drop in performance (from 44% to 19%). We show this visually in the bar chart in Figure 6, where the effect of pre-training a model on human videos is amplified when dealing with all of the data from all

the tasks. This shows that when dealing with a large set of tasks and diverse data, it is crucial to incorporate human-video pre-training for better performance and generalization. Hence we do not run joint world model experiments for `MBRL` and `MBRL-Pix` since the performance is already quite low (around 10 - 20 %) when the model is trained just on single task data.

Human Video Pre-Training As noted in the previous section, pre-training on human videos is *critical* to being able to effectively train joint world models on multi-task data, as seen in Figure 6. For `MBRL-Affordance` we saw that in many cases the model collapses quickly to a sub-optimal control solution when trained on multiple tasks jointly. To investigate this further, we visualize the image reconstructions from $\mathcal{W}$ within the first minute of training and find that the outputs of SWIM were already very realistic, as compared to those of `MBRL-Affordance`. This can be seen in Figure 8, where the outputs of `MBRL-Affordance` are very pixelated while those of SWIM already capture important aspects of the ground truth, indicating the usefulness of pre-training on human videos.

Human-Affordance Action Space How does the choice of action space affect performance? For this we compare the (single task) model based and BC approaches separately. Comparing `MBRL-single` and `MBRL-Affordance-single` in Table I, we can see that there is a clear benefit in using structured action spaces, with over 5X the success compared to cartesian end effector actions. This fits our hypothesis, as it is very difficult for methods that use low-level actions to find successes in a relatively small number of interaction trajectories. The few successes that `MBRL-single` does see are due to the initialization of the robot close to the object using Detic. Furthermore, we note that this benefit is not simply because the affordance actions are in image space. In both the filtered-BC and world model case, the success rate with the affordance action space is roughly *double* than that of acting in pixel

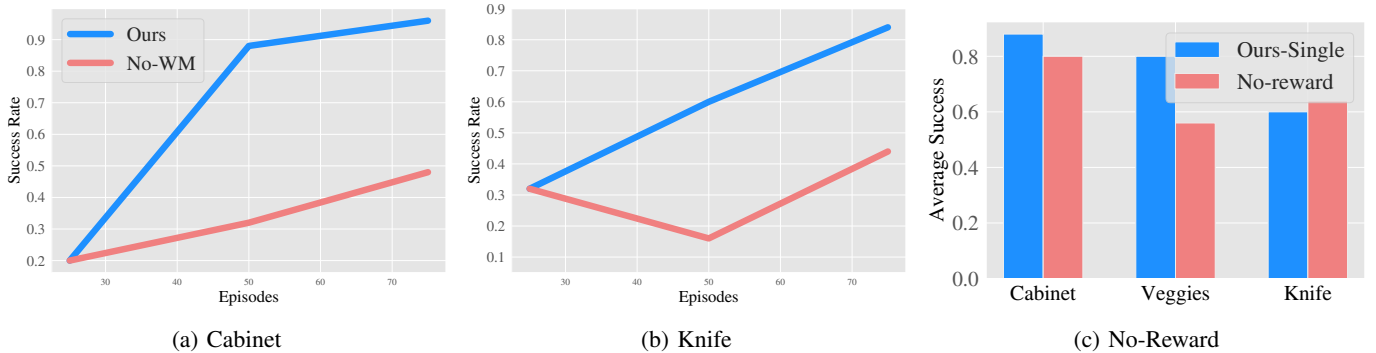(a) Cabinet   (b) Knife   (c) No-Reward

Fig. 7: Continuous improvement (a-b): We see that SWIM continues to improve, achieving high success. (c) Ablating the need for external feature space goal distance at test time.
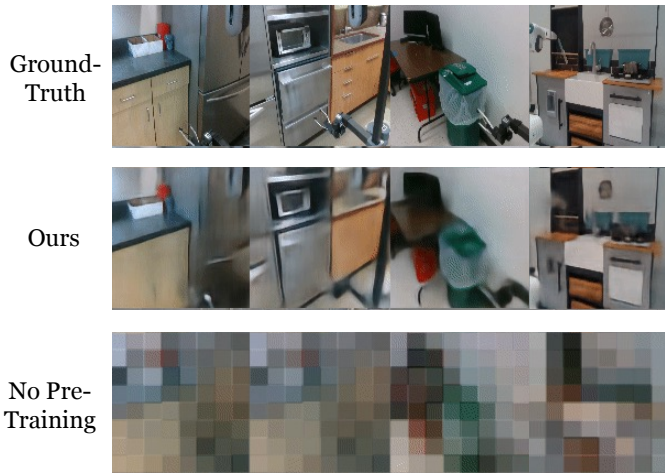

Fig. 8: Image reconstruction using world model features in early training stages for SWIM and `MBRL-Affordance` (which has no pre-training), showing that SWIM can effectively transfer representations from human videos. Note that for our experiments we use models trained to convergence.

space, where target locations are sampled from a random crop around the object. This shows that picking the right action space *and* acting in a meaningful way to collect data can bootstrap learning and lead to efficient control.

Role of World Model   How important is using a world model, and can we achieve good performance by just using the affordance action space? From Table I, we see that the average performance of `BC-Affordance` is not too far behind that of `MBRL-Affordance-single`. However, without a world model, the controller cannot leverage *multi-task* data, both for the pre-training stage to use human videos and for learning the shared structure across multiple robot domains by training a joint world model. Due to these critical reasons discussed previously, SWIM outperforms the best filtered-BC approach by more than a factor of 2.

Continual Improvement   Next, we investigate if SWIM can keep improving using the data that it collects when planning for the task. Since the world model can learn from all the data, we want to test if it can improve its proficiency on the task. Thus, after evaluating $\mathcal{W}$ once, we retrain on the newly collected data (as well as the old data), and re-evaluate the

model. We present the learning curves in Figure 7. We see that SWIM is able to effectively improve performance, and achieves success of over 90%, which is far better than the performance of `BC-Affordance` even after continual training. This is an encouraging sign that SWIM can scale well since it can keep improving its performance with more data to continually learn. In the future, we hope to not only continually finetune, but also add new tasks and settings.

Reward Model   In Figure 7 c) we examine the effect of removing the reward prediction module on planning and find that only using distance in world model feature space is fairly competitive with using both the feature distance and predicted reward. We hypothesize that for the veggies task, it was harder to estimate reward accurately because the free objects tend to move around a lot during training, thus it might take more samples to learn consistent features.

VII. Discussion and Limitations

In this paper, we present SWIM, a simple and efficient way to perform many different complex robotics tasks with just a handful of trials. We aim to build a single model that can learn many tasks, as it holds the promise of being able to continuously learn and improve. We turn to a scalable source of useful data: human videos, from which we can model useful interactions. In order to overcome the morphology gap between robot and human videos, we create a structured action space based on human-centric affordances. This allows SWIM to pre-train a world model on human videos, after which it is fine-tuned using robot data collected in an unsupervised manner. The world model can then be deployed to solve manipulation tasks in the real world. The total robot interaction samples for the system can be collected in just 30 minutes. Videos of SWIM can be found at https://human-world-model.github.io. While SWIM provides a scalable solution and shows encouraging results, some limitations are in the types of actions and tasks that can be performed, as they currently only include quasi-static setups. In future work, we hope to explore different action parameterizations and other types of manipulation tasks. We also hope to scale to many more tasks in order to build a truly generalist agent that can keep on getting better by learning from both passive and active data.

REFERENCES

[1] OpenAI: Marcin Andrychowicz, Bowen Baker, Maciek Chociej, Rafal Jozefowicz, Bob McGrew, Jakub Pachocki, Arthur Petron, Matthias Plappert, Glenn Powell, Alex Ray, et al. Learning dexterous in-hand manipulation. *IJRR*, 2020. 2

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

[3] Pierre-Luc Bacon, Jean Harb, and Doina Precup. The option-critic architecture. In *AAAI*, 2017. 2

[4] Shikhar Bahl, Mustafa Mukadam, Abhinav Gupta, and Deepak Pathak. Neural dynamic policies for end-to-end sensorimotor learning. In *NeurIPS*, 2020. 2

[5] Shikhar Bahl, Abhinav Gupta, and Deepak Pathak. Human-to-robot imitation in the wild. In *RSS*, 2022. 3

[6] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, et al. Rt-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*, 2022. 2

[7] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *NeurIPS*, 2020. 1

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

[9] Kurtland Chua, Roberto Calandra, Rowan McAllister, and Sergey Levine. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. In *NeurIPS*, pages 4754–4765, 2018. 2

[10] Murtaza Dalal, Deepak Pathak, and Russ R Salakhutdinov. Accelerating robotic reinforcement learning via parameterized action primitives. *NeurIPS*, 2021. 2

[11] 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. 4, 5, 6

[12] Christian Daniel, Gerhard Neumann, Oliver Kroemer, and Jan Peters. Hierarchical relative entropy policy search. *Journal of Machine Learning Research*, 2016. 2

[13] 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. *CoRL*, 2019. 1

[14] Marc Deisenroth and Carl E Rasmussen. Pilco: A model-based and data-efficient approach to policy search. In *ICML*, 2011. 2

[15] Marc Peter Deisenroth, Gerhard Neumann, and Jan Peters. A survey on policy search for robotics. *Found. Trends Robot*, 2013. 2

[16] Frederik Ebert, Chelsea Finn, Sudeep Dasari, Annie Xie, Alex Lee, and Sergey Levine. Visual foresight: Model-based deep reinforcement learning for vision-based robotic control. *arXiv preprint arXiv:1812.00568*, 2018. 1, 2

[17] Frederik Ebert, Yanlai Yang, Karl Schmeckpeper, Bernadette Bucher, Georgios Georgakis, Kostas Daniilidis, Chelsea Finn, and Sergey Levine. Bridge data: Boosting generalization of robotic skills with cross-domain datasets. *arXiv preprint arXiv:2109.13396*, 2021. 1

[18] Chelsea Finn and Sergey Levine. Deep visual foresight for planning robot motion. In *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, pages 2786–2793. IEEE, 2017. 2

[19] Mohit Goyal, Sahil Modi, Rishabh Goyal, and Saurabh Gupta. Human hands as probes for interactive object understanding. In *CVPR*, 2022. 2, 3

[20] David Ha and Jürgen Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2018. 2, 3

[21] Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. *arXiv preprint arXiv:1811.04551*, 2018. 2

[22] Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. *arXiv preprint arXiv:1811.04551*, 2018. 2, 3

[23] Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. *arXiv preprint arXiv:1912.01603*, 2019. 2, 3

[24] Danijar Hafner, Timothy Lillicrap, Mohammad Norouzi, and Jimmy Ba. Mastering atari with discrete world models. *arXiv preprint arXiv:2010.02193*, 2020. 3, 6

[25] Nicklas Hansen, Yixin Lin, Hao Su, Xiaolong Wang, Vikash Kumar, and Aravind Rajeswaran. Modem: Accelerating visual model-based reinforcement learning with demonstrations. *arXiv preprint arXiv:2212.05698*, 2022. 3

[26] 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. 6

[27] Stephen James, Kentaro Wada, Tristan Laidlow, and Andrew J Davison. Coarse-to-fine q-attention: Efficient learning for visual robotic manipulation via discretisation. In *CVPR*, 2022. 2

[28] Eric Jang, Alex Irpan, Mohi Khansari, Daniel Kappler, Frederik Ebert, Corey Lynch, Sergey Levine, and Chelsea Finn. Bc-z: Zero-shot task generalization with robotic

imitation learning. In *CoRL*, 2021. 2

[29] 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. 1, 2

[30] Dmitry Kalashnikov, Jacob Varley, Yevgen Chebotar, Benjamin Swanson, Rico Jonschkowski, Chelsea Finn, Sergey Levine, and Karol Hausman. Mt-opt: Continuous multi-task robotic reinforcement learning at scale. *arXiv preprint arXiv:2104.08212*, 2021. 2

[31] Jens Kober and Jan Peters. Learning motor primitives for robotics. In *ICRA*, 2009. 2

[32] Petar Kormushev, Sylvain Calinon, and Darwin G Caldwell. Robot motor skill coordination with em-based reinforcement learning. In *IROS*, 2010. 2

[33] Ashish Kumar, Zipeng Fu, Deepak Pathak, and Jitendra Malik. Rma: Rapid motor adaptation for legged robots. *arXiv preprint arXiv:2107.04034*, 2021. 2

[34] Alex X Lee, Richard Zhang, Frederik Ebert, Pieter Abbeel, Chelsea Finn, and Sergey Levine. Stochastic adversarial video prediction. *arXiv preprint arXiv:1804.01523*, 2018. 1

[35] Alex X Lee, Anusha Nagabandi, Pieter Abbeel, and Sergey Levine. Stochastic latent actor-critic: Deep reinforcement learning with a latent variable model. *arXiv preprint arXiv:1907.00953*, 2019. 2

[36] Sergey Levine, Chelsea Finn, Trevor Darrell, and Pieter Abbeel. End-to-end training of deep visuomotor policies. *JMLR*, 2016. 1

[37] Shaowei Liu, Subarna Tripathi, Somdeb Majumdar, and Xiaolong Wang. Joint hand motion and interaction hotspots prediction from egocentric videos. In *CVPR*, 2022. 2, 3

[38] 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. 3

[39] Roberto Martin-Martin, Michelle A. Lee, Rachel Gardner, Silvio Savarese, Jeannette Bohg, and Animesh Garg. Variable impedance control in end-effector space: An action space for reinforcement learning in contact-rich tasks. *IROS*, 2019. 2

[40] Katharina Mülling, Jens Kober, Oliver Kroemer, and Jan Peters. Learning to select and generalize striking movements in robot table tennis. *The International Journal of Robotics Research*, 2013. 2

[41] Anusha Nagabandi, Gregory Kahn, Ronald S Fearing, and Sergey Levine. Neural network dynamics for model-based deep reinforcement learning with model-free fine-tuning. *arXiv preprint arXiv:1708.02596*, 2017. 2

[42] Anusha Nagabandi, Kurt Konolige, Sergey Levine, and Vikash Kumar. Deep dynamics models for learning dexterous manipulation. In *CoRL*, 2020. 2

[43] Tushar Nagarajan, Christoph Feichtenhofer, and Kristen Grauman. Grounded human-object interaction hotspots from video. In *ICCV*, 2019. 2, 3

[44] 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. 6

[45] 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, 3, 5, 6

[46] Simone Parisi, Hany Abdulsamad, Alexandros Paraschos, Christian Daniel, and Jan Peters. Reinforcement learning vs human programming in tetherball robot games. In *IROS*, 2015. 2

[47] Peter Pastor, Mrinal Kalakrishnan, Sachin Chitta, Evangelos Theodorou, and Stefan Schaal. Skill learning and task outcome prediction for manipulation. In *ICRA*, 2011. 2

[48] Xue Bin Peng, Aviral Kumar, Grace Zhang, and Sergey Levine. Advantage-weighted regression: Simple and scalable off-policy reinforcement learning. *arXiv preprint arXiv:1910.00177*, 2019. 6

[49] Jan Peters and Stefan Schaal. Reinforcement learning by reward-weighted regression for operational space control. In *ICML*, 2007. 6

[50] Jan Peters, Sethu Vijayakumar, and Stefan Schaal. Reinforcement learning for humanoid robotics. In *International Conference on Humanoid Robots*, 2003. 2

[51] Lerrel Pinto and Abhinav Gupta. Supersizing self-supervision: Learning to grasp from 50k tries and 700 robot hours. *ICRA*, 2016. 1

[52] Yuzhe Qin, Yueh-Hua Wu, Shaowei Liu, Hanwen Jiang, Ruihan Yang, Yang Fu, and Xiaolong Wang. Dexmv: Imitation learning for dexterous manipulation from human videos. In *ECCV*, 2022. 3

[53] 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. 1

[54] Ilija Radosavovic, Tete Xiao, Stephen James, Pieter Abbeel, Jitendra Malik, and Trevor Darrell. Real-world robot learning with masked visual pre-training. *arXiv preprint arXiv:2210.03109*, 2022. 3

[55] Scott Reed, Konrad Zolna, Emilio Parisotto, Sergio Gomez Colmenarejo, Alexander Novikov, Gabriel Barth-Maron, Mai Gimenez, Yury Sulsky, Jackie Kay, Jost Tobias Springenberg, et al. A generalist agent. *arXiv preprint arXiv:2205.06175*, 2022. 2

[56] Reuven Rubinstein. The cross-entropy method for combinatorial and continuous optimization. *Methodology and computing in applied probability*, 1:127–190, 1999. 5

[57] Jürgen Schmidhuber. Curious model-building control

systems. In *[Proceedings] 1991 IEEE International Joint Conference on Neural Networks*, pages 1458–1463. IEEE, 1991. 3

[58] Jürgen Schmidhuber. A possibility for implementing curiosity and boredom in model-building neural controllers, 1991. 3

[59] Pierre Sermanet, Kelvin Xu, and Sergey Levine. Unsupervised perceptual rewards for imitation learning. In *RSS*, 2017. 3

[60] Dandan Shan, Jiaqi Geng, Michelle Shu, and David F Fouhey. Understanding human hands in contact at internet scale. In *CVPR*, pages 9869–9878, 2020. 2, 3, 6

[61] Lin Shao, Toki Migimatsu, Qiang Zhang, Karen Yang, and Jeannette Bohg. Concept2robot: Learning manipulation concepts from instructions and human demonstrations. *The International Journal of Robotics Research*, 40(12-14), 2021. 3

[62] Pratyusha Sharma, Deepak Pathak, and Abhinav Gupta. Third-person visual imitation learning via decoupled hierarchical controller. *arXiv preprint arXiv:1911.09676*, 2019. 3

[63] Kenneth Shaw, Shikhar Bahl, and Deepak Pathak. Videodex: Learning dexterity from internet videos. In *CoRL*, 2022. 3

[64] Mohit Shridhar, Lucas Manuelli, and Dieter Fox. Cliport: What and where pathways for robotic manipulation. In *CoRL*, 2022. 2

[65] Mohit Shridhar, Lucas Manuelli, and Dieter Fox. Perceiver-actor: A multi-task transformer for robotic manipulation. *arXiv preprint arXiv:2209.05451*, 2022. 2

[66] Laura Smith, Nikita Dhawan, Marvin Zhang, Pieter Abbeel, and Sergey Levine. Avid: Learning multi-stage tasks via pixel-level translation of human videos. In *RSS*, 2020. 3

[67] F. Stulp, E. A. Theodorou, and S. Schaal. Reinforcement learning with sequences of motion primitives for robust manipulation. *Transactions on Robotics*, 2012. 2

[68] Richard S. Sutton, Doina Precup, and Satinder Singh. Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. *Artificial Intelligence*, 1999. 2

[69] Josh Tobin, Lukas Biewald, Rocky Duan, Marcin Andrychowicz, Ankur Handa, Vikash Kumar, Bob Mc-Grew, Alex Ray, Jonas Schneider, Peter Welinder, et al. Domain randomization and generative models for robotic grasping. In *IROS*, 2018. 2

[70] Arash Vahdat and Jan Kautz. Nvae: A deep hierarchical variational autoencoder. *NeurIPS*, 33:19667–19679, 2020. 6

[71] Théophane Weber, Sébastien Racanière, David P Reichert, Lars Buesing, Arthur Guez, Danilo Jimenez Rezende, Adria Puigdomènech Badia, Oriol Vinyals, Nicolas Heess, Yujia Li, et al. Imagination-augmented agents for deep reinforcement learning. *arXiv preprint arXiv:1707.06203*, 2017. 2

[72] Bohan Wu, Suraj Nair, Li Fei-Fei, and Chelsea Finn. Example-driven model-based reinforcement learning for solving long-horizon visuomotor tasks. *arXiv preprint arXiv:2109.10312*, 2021. 1

[73] Philipp Wu, Alejandro Escontrela, Danijar Hafner, Ken Goldberg, and Pieter Abbeel. Daydreamer: World models for physical robot learning. *arXiv preprint arXiv:2206.14176*, 2022. 2

[74] Tete Xiao, Ilija Radosavovic, Trevor Darrell, and Jitendra Malik. Masked visual pre-training for motor control. *arXiv preprint arXiv:2203.06173*, 2022. 1, 3

[75] Haoyu Xiong, Quanzhou Li, Yun-Chun Chen, Homanga Bharadhwaj, Samarth Sinha, and Animesh Garg. Learning by watching: Physical imitation of manipulation skills from human videos. *arXiv preprint arXiv:2101.07241*, 2021. 3

[76] Kevin Zakka, Andy Zeng, Pete Florence, Jonathan Tompson, Jeannette Bohg, and Debidatta Dwibedi. Xirl: Cross-embodiment inverse reinforcement learning. *arXiv preprint arXiv:2106.03911*, 2021. 3

[77] Andy Zeng, Shuran Song, Stefan Welker, Johnny Lee, Alberto Rodriguez, and Thomas Funkhouser. Learning synergies between pushing and grasping with self-supervised deep reinforcement learning. In *IROS*, 2018. 2

[78] 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. 2

[79] 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. 6