Mirage: Cross-Embodiment Zero-Shot Policy Transfer with Cross-Painting

Lawrence Yunliang Chen*[1], Kush Hari*[1], Karthik Dharmarajan*[1], Chenfeng Xu[1], Quan Vuong[2], Ken Goldberg[1]
[1] UC Berkeley [2] Google DeepMind
https://robot-mirage.github.io

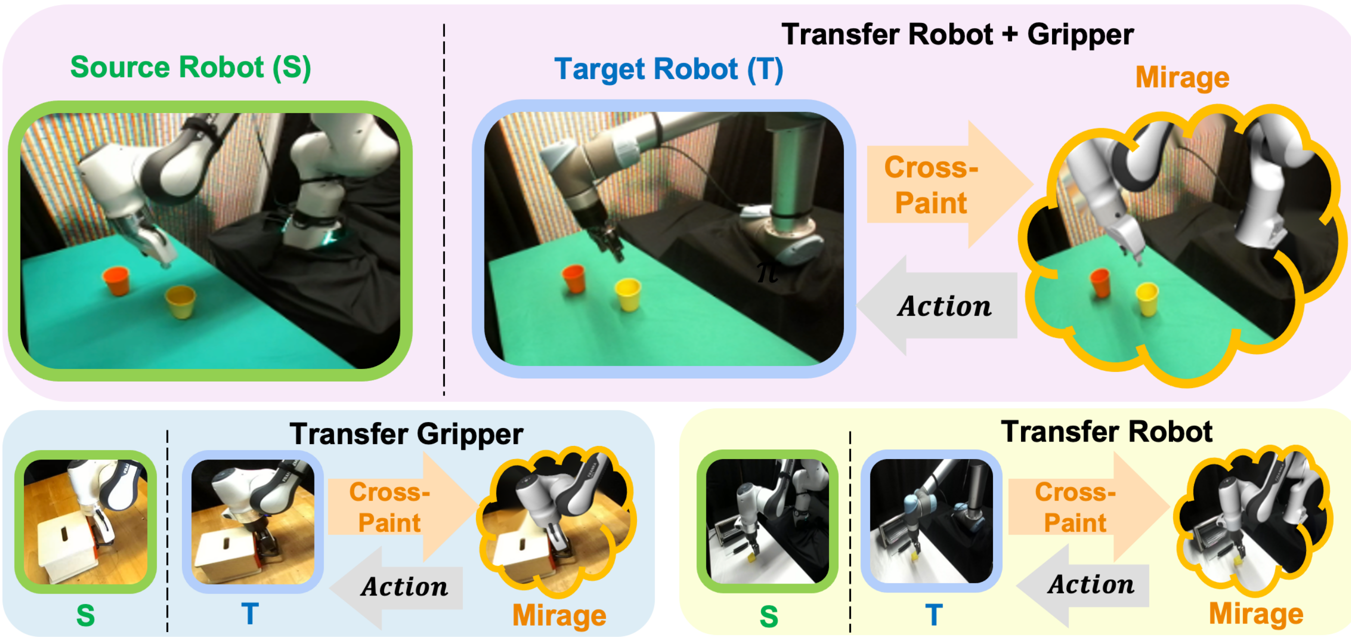

Fig. 1: Overview of Mirage. We study zero-shot transfer of policy across embodiments. Assume there is a policy trained on a source robot (left). At test time, with an unseen target robot (middle), Mirage performs "cross-painting"—masking out the target robot in the image and inpainting the source robot at the same end effector pose—using robot URDFs and a renderer. By creating an illusion as if the source robot were performing the task (right), Mirage queries the source policy with the cross-painted image to obtain the source robot's action. The target robot then uses a forward dynamics model to obtain the desired end effector pose in the target robot frame and executes the steps with a blocking controller. Mirage can successfully zero-shot transfer policies across the same robots with different grippers (bottom left), different robots with the same gripper (bottom right), and different robots with different grippers (top).

***Abstract*—The ability to reuse collected data and transfer trained policies between robots could alleviate the burden of additional data collection and training. While existing approaches such as pretraining plus finetuning and co-training show promise, they do not generalize to robots unseen in training. Focusing on common robot arms with similar workspaces and 2-jaw grippers, we investigate the feasibility of zero-shot transfer. Through simulation studies on 8 manipulation tasks, we find that state-based Cartesian control policies can successfully zero-shot transfer to a target robot after accounting for forward dynamics. To address robot visual disparities for vision-based policies, we introduce Mirage, which uses "cross-painting"—masking out the unseen target robot and inpainting the seen source robot—during execution in real time so that it appears to the policy as if the trained source robot were performing the task. Despite its simplicity, our extensive simulation and physical experiments provide strong evidence that Mirage can successfully zero-shot transfer between different robot arms and grippers with only minimal performance degradation on a variety of manipulation tasks such as picking, stacking, and assembly, significantly outperforming a generalist policy.**

I. Introduction

Consider a scenario where substantial efforts are invested in training a vision-based policy on a Franka robot arm for a manipulation task. The conventional paradigm of policy transfer to another robot often involves collecting new data on the target robot and finetuning the pretrained policy or retraining with a combination of datasets. Collecting data and training policies for each task and physical robot embodiment is time-consuming and expensive. An exciting conjecture is that demonstration data from different robot arms can be combined to learn policies that are robust or more generalizable to different robots, such as Franka, Universal, ABB, KUKA, and Fanuc arms [19]. While scaling up shows great promise in in-distribution embodiments, transferring policies to unseen embodiments remains elusive.

In this work, we ask the question: Is it possible to achieve policy transfer without any target robot data? This poses several challenges, as outlined in prior work [110], stemming from variations in kinematic configuration, control scheme, camera viewpoint, and end-effector morphology. However, commonalities exist among many robots in practice. Despite differences in joint numbers, many robots popular in open-sourced datasets [19] such as the Franka, xArm, Sawyer, Kuka iiwa, and UR5 have similar workspaces and can be controlled in the Cartesian space of the end effector with millimeter-level accuracy. Similarly, while grippers may vary in appearance, most use parallel jaws with similar shapes (but differing dimensions).

Our key insight is that, for robots with similar workspaces and many quasi-static tasks, the unseen target robot can achieve relatively high success rates by directly querying the policy trained on the seen source robot without the need for fine-tuning. To do so, we set the source robot to the same pose as the target robot using state information. The high success rates hold for policies that are both open-loop and closed-loop, both state-based and image-based, and trained using imitation learning and reinforcement learning.

We present Mirage (Fig. 1), a novel cross-embodiment policy transfer method that can zero-shot transfer policies trained on one source robot to an unseen target robot. Mirage decouples vision and control, allowing gaps between the source and target robot to be addressed separately. To address visual differences, Mirage employs "cross-painting" during execution, masking out the target robot and inpainting the source robot at the same pose, creating a mirage for the policy. Importantly, Mirage only requires knowledge of robot base coordinate frames for visual and state alignment. Mirage does not assume any demonstration data on the target robot or paired images or trajectories between the source and unseen target robots. To handle differences in control gains, Mirage pairs the source robot policy with a forward dynamics model and executes the action predicted by the policy on the target robot with a high-gain or blocking controller to accommodate varying control frequencies.

Through extensive experiments on 9 manipulation tasks in both simulation and real across 6 different robot and gripper setups, we show that Mirage, despite its simplicity, is remarkably effective at transferring policies, achieving zero-shot performance significantly higher than a state-of-the-art generalist model [73]. To the best of our knowledge, this is the first demonstration on real robots of zero-shot transfer of visual manipulation skills beyond pushing tasks [41].

To summarize, our key contributions are:

1) A systematic simulation study analyzing the challenges and potential for policy transfer between grippers and arms;
2) Mirage, a novel zero-shot cross-embodiment policy transfer method that uses cross-painting to bridge the visual gap and forward dynamics to bridge the control gap;
3) Physical experiments with Franka and UR5 demonstrating that Mirage successfully transfers between robots and grippers on 4 manipulation tasks, suffering only minimal performance degradation from the source policy and significantly outperforming a state-of-the-art generalist model.

II. Related Work

A. Transfer across embodiments

Can data and models from various sources, including other robots, tasks, environments [86, 117], and modalities be transferred to new robots with differences in morphology, control, and sensors? While some cases pose significant disparities, such as a small soft robot with tactile sensors versus a humanoid robot with RGB cameras, commonly used robot arms share similarities in workspaces, grippers, and camera-based manipulation tasks.

Transferring control dynamics. Previous work in control has investigated cross-robot transfer of dynamics models [18, 34, 39, 78] and alignment-based transfer [6, 64, 65] is a commonly used technique for trajectory tracking, where input-output data points for both robots are collected and then aligned using manifold alignment. The transformed data from the source robot can then be used as initialization to accelerate learning of the dynamics model of the target robot. Mirage does not focus on trajectory tracking. We fit a dynamics model to the source robot trajectory and use a high-gain or blocking controller on the target robot during execution.

Transfer learning and cross-domain imitation methods use finetuning to accelerate the learning process by leveraging data from other robots [42, 83, 99, 123]. For example, assuming access to a policy of the source robot, one can use Reinforcement Learning (RL) to finetune the visual encoder [82, 98], the value function [54], or the policy [59, 60] on the target robot. When learning from videos of other other agents without access to actions [3, 7, 61, 85, 85, 96, 104, 106, 108, 114], methods such as explicit retargeting [75, 95], inverse RL [12, 116], and goal-conditioned RL on a learned latent space [121] have been explored. Assuming isometry between domains, Fickinger et al. [30] train RL in the target domain by using a trajectory in the source domain as a pseudo-reward. Many other works assume data for both source and target robots on a proxy task, learn correspondences if not already available [2, 52, 79, 119], and then use the learned latent space or paired trajectories as auxiliary rewards [2, 36, 56, 91] or for adversarial training [32, 38, 111]. Unlike these methods, Mirage does not assume access to target robot data on a proxy task or paired trajectories, and directly reuses the source robot policy.

Multi-task and Multi-robot training. If the target robot is unknown but comes from a known distribution, such as the distribution of the length of the arm links, kinematic domain randomization [28] can be used to train a robot-agnostic policy. Alternatively, the robot parameters such as kinematic parameters [112], 6 DOF transforms for each joint [14], and URDFs represented by a stack of point clouds [92] or TSDF volumes [109] can be used to train a robot-conditioned policy. Ghadirzadeh et al. [35] use meta-learning to infer the latent vector for a new robot from few-shot trajectories, and Noguchi et al. [72] treat grasped tools as part of the embodiment to learn to grasp objects with objects. Other work has also leveraged a known robot distribution to train modular policies [22, 33, 47, 107, 120]. For example, Devin et al. [22] train a policy head for each task and each robot, and Furuta et al. [33] train an RL policy for each task and each morphology combination and distill it into a transformer. Others [43, 55, 66, 74, 84, 103] encode robot morphology as a graph for locomotion. While these methods leverage simulation to vary the robots in the known distribution, we do not seek to train a policy with the intention of it being performant on a distribution of robots. Instead, Mirage can transfer a specialized policy trained on only one robot to an unseen target robot that the policy is not intended to work on.

Yang et al. [110] study both the control and visual domain gaps when transferring across robots. They use wrist cameras to minimize the observation differences of the robot, learn a multiheaded policy with robot-specific heads that capture separate dynamics, and use contrastive learning to align robot trajectories. In contrast, Mirage works with third-person camera(s) and does not assume any target robot data to jointly train a multi-robot policy. Another closely related work to ours is Robot-Aware Visual Foresight [41]. The authors use the known camera matrix and robot model to mask the robot pixels, train a video-prediction model that only predicts the world pixels, and use visual foresight [25, 31] during execution. Similar to their method, we use robot URDFs to compute robot pixels, but we not only mask out the target robot but also inpaint the source robot. By doing so, we do not impose restrictions on the policy class the source robot can use, allowing Mirage to work with state-of-the-art source robot policies such as diffusion policy [17] and successfully transfer them across real robots beyond pushing tasks [41].

B. *Learning from large datasets*

Beyond targeted effort to transfer robot policies, recent work has also explored use of large and diverse data [21, 26, 27, 29, 49, 58, 88, 102] to train visual encoders [63, 71, 105] and policies that are generalizable to new objects, scenes, tasks, as well as embodiments [1, 4, 10, 11, 15, 23, 46, 46, 48, 62, 76, 80, 89, 90, 93, 94, 97]. Dasari et al. [20] use a large dataset of multiple robots to pretrain a visual dynamics model. Most recently, several works [8, 19, 73, 77] pretrain large transformers and show that co-training or finetuning the models outperform policies trained only on in-domain data. However, RPT [77] uses joint space actions, preventing it from zero-shot transfer. While RT-X [19] and Octo [73] use Cartesian space actions, they do not align action spaces or condition on the robot coordinate frames and kinematics, preventing them from working zero-shot on a new robot setup as they do not know what action space to use. In this work, we conduct experiments that show that leveraging the robot URDFs and aligning the action spaces could enable even a single-robot specialist policy to generalize zero-shot to a different robot.

A work that is closely related to ours is MimicGen [69]. Given novel object poses, they spatially transform human demonstration trajectories in an object-centric way and let the robot follow these generated trajectories to collect new demonstrations. They illustrate that MimicGen can be used to generate rollout data on Sawyer, IIWA, and UR5e arms even when the original demonstrations were on the Franka, and the generated data can be used to learn a target robot policy. In contrast, Mirage does not assume an object-centric framework and uses cross-painting for trained policies, eliminating the requirements for demonstration trajectories on the target robot.

C. *Image inpainting and augmentation in robotics*

Visual domain randomization such as adding distractors and changing the textures and lighting is commonly used to bridge the sim2real gap [101]. Recently, enabled by generative models, researchers have explored image editing such as adding distractors and changing backgrounds as an additional form of augmentation [16, 67, 115]. Black et al. [5] use a video prediction model to generate goal images during execution. Hirose et al. [40] augment the collected data to simulate another robot's observation from a different viewpoint. AR2-D2 [24] renders a virtual AR robot in a scene observed by a camera to replace the hand of a human manipulating objects; the result appears as if the AR robot were manipulating the object. Bahl et al. [3] use a video inpainting model to mask out human hands and robot grippers. Mirage also inpaints the source robot, and can optionally reproject the image into the camera angle the policy is trained on, to create the illusion to the policy as if the source robot were performing the task.

III. Problem Statement

We assume two robot arms, source $\mathcal{S}$ and target $\mathcal{T}$, with parallel-jaw grippers, known URDFs, and known isomorphic kinematics. We assume the manipulation tasks of interest are in both robots' workspace and can be performed by both robots using similar strategies without collision. Our goal is to transfer a trained policy from the source robot to the target robot.

Prior work [110] has found aligning the action and observation spaces can facilitate policy transfer. Mirage leverages the following assumptions and design choices to reduce the gap between robots and enable zero-shot transfer:

1) We assume knowledge of the two robots' coordinate frames. To align the state and action spaces, we follow prior work [110] and use the Cartesian space of the end effector. This allows us to transfer between robots with different numbers of joints and compensate for alternate gripper shapes across embodiments.
2) We assume the target robot has a high-gain or blocking controller that can reach desired poses relatively accurately (within a few millimeters).
3) We assume knowledge of the camera parameters for both domains. This allows us to render robots in a camera pose that is within the distribution of the training image poses.
4) We assume that the background and lighting conditions of the target robot are in the distribution of the source dataset D or that the policy $\pi_{\mathcal{S}}$ is robust to environmental background changes, e.g., using techniques such as background augmentation [16, 67, 115]. This allows us to separate any challenges that arise due to changes in

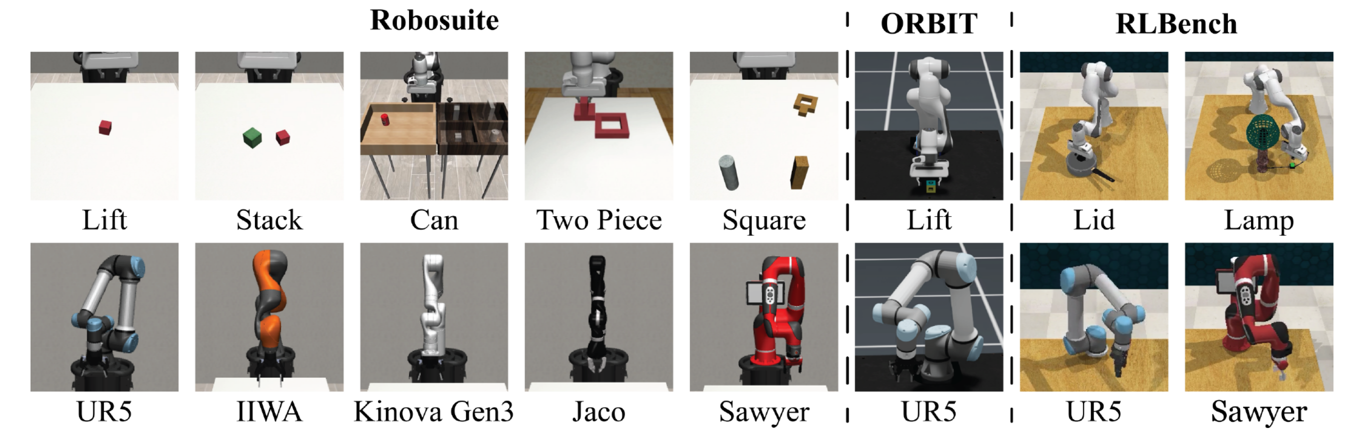

Fig. 2: Simulation Tasks and Robots. The simulation evaluation utilizes the Robosuite simulator with Lift, Stack, Can Pick-and-Place, Two Piece Assembly, and Square Peg Insertion tasks. Additionally, to study other policy classes, we evaluate policy transfers in ORBIT with a block lifting task, and in RLBench with 2 tasks: Lifting a lid, and Pushing a button to turn on a lamp. For all policies, the source robot is the Franka robot as shown in the first row, while the target robots for each of the tasks are shown in the second row.

the background environment and focus on the impact of visual differences between robots on policy performance.

We model each robot manipulation task of interest as a Markov Decision Process. We consider the setting where there is a policy $\pi_{\mathcal{S}}$ trained on a dataset of the source robot $D = \{(s_1^{\mathcal{S}}, o_1^{\mathcal{S}}, a_1^{\mathcal{S}}, ..., s_{H_i}^{\mathcal{S}}, o_{H_i}^{\mathcal{S}})_i^N\}$ consisting of N trajectories, where $s_t^{\mathcal{S}}$ is the proprioceptive state of the robot at timestep t, $o_t^{\mathcal{S}}$ is the third-person camera observation(s), and $a_t^{\mathcal{S}}$ is the action. Given a source policy $a_{t+1}^{\mathcal{S}} = \pi_{\mathcal{S}}(s_t^{\mathcal{S}}, o_t^{\mathcal{S}})$, we would like to transform it into a policy $a_{t+1}^{\mathcal{T}} = \pi_{\mathcal{T}}(s_t^{\mathcal{T}}, o_t^{\mathcal{T}})$ that takes as inputs the states and observations of the target robot without demonstration or finetuning trajectory data on the target robot.

IV. State-Based Transfer Experiments

As a first step in studying the feasibility of zero-shot transfer, we seek to separate the domain gaps of the control from the visuals. To motivate the study, imagine there is a source robot ("oracle") teaching a target robot to perform a task side by side in a duplicate environment. At each time step, the source robot sees the world state $(p_{\mathrm{r}}, p_{\mathrm{o}})$ of the target environment, where p_{r} and p_{o} are the poses of the robot end effector and the objects. Then, it puts its objects and end effector to the same poses, and uses its policy to move its end effector to a new pose p_{r}'. The target robot observes the source robot and also moves its end effector there. In this manner, the target robot mirrors what the source robot does step by step to perform the task. We ask: can the target robot successfully complete a task by querying the source robot's policy in this fashion?

To answer this question, we consider 8 tasks across 3 simulators (Robosuite [122], ORBIT [70], and RLBench [44]) (Fig. 2) with policies trained using imitation learning for Robosuite tasks and reinforcement learning for ORBIT and RLBench. Robosuite and ORBIT policies use closed-loop control (a trajectory consists of >50 time steps), while for RLBench, the policies use open-loop control (a trajectory consists of a few waypoints). For all tasks, we train the source state-based policy on the Franka robot and evaluate the success rates on different target robots using the test-time execution strategy mentioned above.

As the coordinate frames between robots are not necessarily the same, we use the known rigid transform $T_{\mathcal{T}}^{\mathcal{S}}$ between the frames to convert the end-effector and object poses at each time step from the target frame to the source frame $p_{\mathrm{r}}^{\mathcal{S}} = T_{\mathcal{T}}^{\mathcal{S}} p_{\mathrm{r}}^{\mathcal{T}}$ and $p_{\mathrm{o}}^{\mathcal{S}} = T_{\mathcal{T}}^{\mathcal{S}} p_{\mathrm{o}}^{\mathcal{T}}$, before querying the source robot's policy $a^{\mathcal{S}} = \pi_{\mathcal{S}}(p_{\mathrm{r}}^{\mathcal{S}}, p_{\mathrm{o}}^{\mathcal{S}})$. We step the action $a^{\mathcal{S}}$ on the source robot in the simulator to obtain the achieved end effector pose $p_{\mathrm{r}}^{\mathcal{S}'}$, and then use the high-gain or blocking controller on the target robot to reach the equivalent desired pose $p_{\mathrm{r}}^{\mathcal{T}'} = T_{\mathcal{S}}^{\mathcal{T}} p_{\mathrm{r}}^{\mathcal{S}'}$.

A. Implementation Details

For Robosuite, we choose 5 tasks: Lift, Stack, Can, Two Piece Assembly, and Square. We use Robomimic [68] to train an LSTM policy for each task on the provided demonstration data [69, 122]. We evaluate the performances on 5 different robots, including UR5e, Kuka iiwa, Kinova Gen3, Sawyer, and Jaco. Note that Jaco has a 3-jaw gripper, but we include it for comparison. The source policy predicts delta Cartesian actions and we use the operation space controller [50] on the target robot to servo to the pose the source robot reaches and enforce that the norm of the error from the desired pose (position (in m) and quaternion) is less than 0.015 at each timestep.

For ORBIT, we use the Lift task. We train an RL policy using PPO [87] on the Franka robot and evaluate on the UR5. Similar to Robosuite, we use the absolute pose controller to servo the desired pose at each step during execution. For RLBench, we use Coarse-to-Fine Q-attention [45] to learn the key poses and use its end-effector-pose-via-planning controller to reach the desired pose in an open-loop fashion. The policy also takes in camera observations and the scene point cloud. We study 2 tasks: take the lid off a saucepan ("Unlid Pan") and turn on a lamp ("Lamp On"), and evaluate on the UR5 and Sawyer.

B. Study Results

Table I shows that when the target robots have the same gripper as the source robot, most unseen target robots achieve

Task	Source (Franka)	Franka Gripper on Target Robot					Default Gripper on Target Robot				
		UR5e	IIWA	Kinova Gen3	Sawyer	Jaco	UR5e	IIWA	Kinova Gen3	Sawyer	Jaco*
Lift (Robosuite)	100%	99%	100%	100%	100%	100%	100%	100%	99%	84%	95%
Stack	95%	96%	94%	96%	94%	96%	86%	87%	81%	85%	66%
Can	98%	98%	97%	97%	96%	96%	88%	90%	83%	83%	55%
Two Piece Assembly	96%	91%	89%	88%	92%	90%	89%	70%	92%	90%	34%
Square	81%	74%	72%	34%	83%	21%	69%	48%	49%	75%	1%
Lift (ORBIT)	100%	-	-	-	-	-	89%	-	-	-	-
Unlid Pan	100%	-	-	-	-	-	100%	-	-	90%	-
Lamp On	88%	-	-	-	-	-	64%	-	-	68%	-

TABLE I: **State-Based Policy Transfer Experiment Results.** We evaluate state-based policies trained for each task using a Franka robot across five different robots equipped either with the original Franka gripper or with each target robot's default gripper. Results suggest that most unseen target robots can successfully perform the tasks using the source robot as its guide for where to move its gripper. *Jaco has a 3-jaw gripper, which explains its lower success rates.

very high task success rates. This suggests that the kinematic differences among the robot arms are relatively insignificant. Using the robots' default factory 2-jaw grippers, there is a mild performance drop of around 10-25%, but many target robots can still successfully perform the task using the source robot as its guide for where to move its gripper. In comparison, with a 3-jaw gripper, Jaco's success rates are significantly lower than the others', especially on more challenging tasks, where the grasp configuration required for three jaws is different than the parallel jaw grippers. Comparing the robots, we see that the differences in the performance are roughly consistent across tasks, indicating that gripper properties (e.g., size and friction of the gripper pads) affect how easy it is for the robots to grasp and manipulate objects, but the general task strategy is similar. This holds for policies trained using IL and RL, as well as open loop and closed loop.

Additionally, we notice that the more robust the source policy is, the smaller the performance drop when transferring to other robots. We qualitatively observe that this can be attributed to the extra space the policy leaves between the object and its gripper as well as its retrying behavior. Less robust source policies leave little room for error, while more robust ones tend to retry even if the target robot fails to grasp the object the first time.

To study the impact of differences in the robot controller dynamics, we also experiment with executing the delta action a on the target robot with the non-blocking controller directly instead of using a blocking controller to reach the desired pose $p_{\mathrm{r}}^{\mathcal{T}'}$. Results are included in the Appendix. We see that there is a significant drop in performance, indicating that the difference in the forward dynamics between robots cannot be ignored when transferring policies, and that leveraging a blocking controller on the target robot is an effective way to mitigate this difference.

V. Mirage: A Cross-Embodiment Transfer Strategy for Vision-Based Policies

Motivated by the observation that target robots can successfully perform tasks to a large extent simply by querying a state-based source policy on an oracle source robot that mirrors its pose and obtains the next pose, we seek to extend this strategy to vision-based policies. To transfer vision-based policies, we need to account for the additional difference of robot visuals in addition to the controller forward dynamics.

We propose Mirage, a strategy to zero-shot transfer a trained vision-based policy from the source robot to the target robot. The key idea is "cross-painting": replacing the target robot with the source robot in the camera observations at test time so that it appears to the policy as if the source robot were performing the task.

A. Bridging the Visual Gap

To replace the robots, we leverage the knowledge of the robot URDFs and camera poses to perform cross-painting at test time. Fig. 3 illustrates the pipeline of Mirage. First, given known camera transforms, we can optionally reproject the images from the target domain to the source domain if depth sensing is available at test time. Next, given the image observation $o^{\mathcal{T}}$ and joint angles of the target robot, we use a renderer to determine which image pixels correspond to the source robot and mask out these pixels. Then, we inpaint the missing pixels. We use the fast marching inpainting method [9, 100] for simplicity and speed, but other choices such as off-the-shelf image or video inpainting networks [57, 113] could also work. Another potential approach is to first take an offline picture of the background scene with as much of the target robot arm moved out of the camera frame as possible, and at test time just to fill in the masked-out region with the corresponding pixels from the background image, but this would only apply to fixed third-person cameras. Finally, we use the URDF of the source robot to solve for the joint angles that would put its end effector at the same pose as that of the target robot, render it using a simulator, and overlay it onto the target image. For simulation experiments, we take into account potential occlusions between the robot and objects by comparing the pixel-wise depth values between the camera observation of the scene and the rendered robot. For real experiments, however, we do not use depth due to noise and imprecision in the camera observations. For the gripper, we similarly compute and set the joints of the source robot gripper in the renderer so that its width would roughly match that of the target robot's gripper. We denote this cross-painted image $o^{\mathcal{T}\to\mathcal{S}}$.

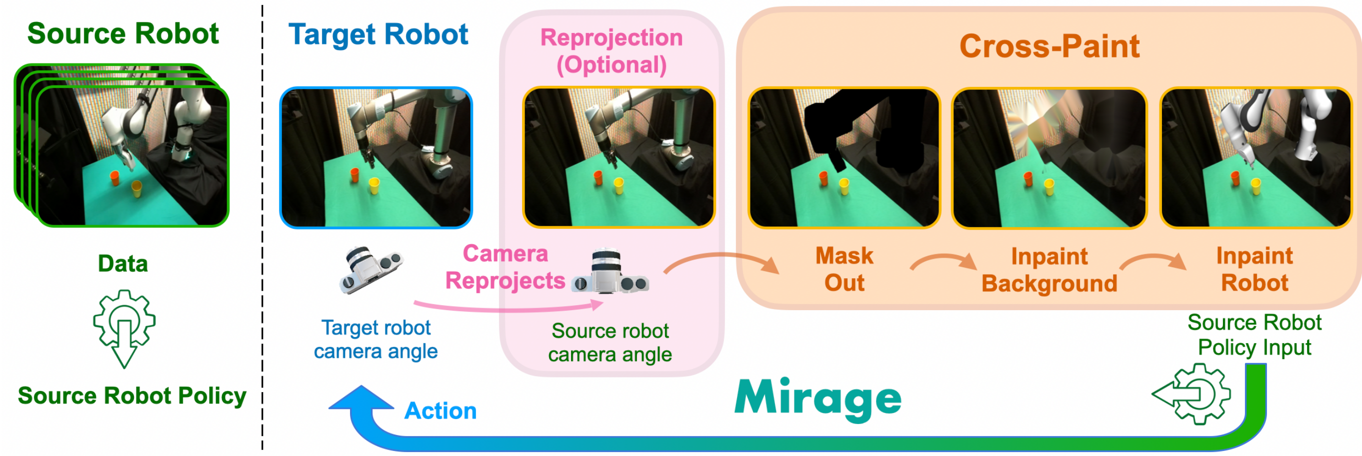

Fig. 3: **Illustration of Mirage's pipeline**. We reproject the camera from the target frame to the source frame if there is a non-negligible camera angle change and then apply cross-painting: (1) use the segmentation mask provided by Gazebo to mask out the target robot, (2) apply the fast marching [100] algorithm to fill in the missing pixels, and (3) overlay Gazebo's rendering of the source robot URDF onto the image. The resulting image is fed into the source robot's policy to obtain the action, which is executed after a coordinate frame transform.

Task	Source (Franka)	UR5e (Franka / Default Gripper)			Kinova Gen3 (Franka / Default Gripper)		
		Oracle	Naive 0-shot	Mirage	Oracle	Naive 0-shot	Mirage
Lift	97%	98% / 98%	55% / 0%	88% / 76%	100% / 97%	48% / 1%	88% / 72%
Stack	97%	97% / 95%	17% / 5%	84% / 80%	99% / 96%	12 % / 2%	76% / 80%
Can	94%	71% / 56%	13% / 1%	68% / 48%	58% / 44%	15% / 2%	40% / 32%
Two Piece Assembly	99%	97% / 99%	16% / 2%	96% / 100%	89% / 91%	32% / 25%	84% / 80%
Square	70%	73% / 69%	22% / 3%	68% / 52%	48% / 32%	8% / 2%	40% / 40%

TABLE II: **Mirage Results on Transferring Vision-Based Policies in Simulation.** For each task and robot arm combination, the Oracle represents the performance of a vision-based policy assuming access to a ground truth rendering of the source robot given the state of the target robot, the Naive 0-shot method directly passes the visual observation of the target robot to the policy, and Mirage uses cross-painting to generate the visual inputs for the policy. For each method, the first number represents the success rate when the target robot uses the source robot (Franka) gripper and the second number corresponds to using the target robot's default gripper (Robotiq gripper).

B. Bridging the Control Gap

To bridge the difference of the controllers, we use a forward dynamics model to convert the source robot action into the next pose to achieve, and use a high-gain or blocking controller on the target robot to reach the pose. We use the source robot trajectory data D to fit a forward dynamics model f on the transitions: $f(p^{\mathcal{S}}_{\mathrm{r},t}, a^{\mathcal{S}}_t) = p^{\mathcal{S}}_{\mathrm{r},t+1}$. During the target robot's execution, the desired target pose is thus $p^{\mathcal{T}}_{\mathrm{r},t+1} = T^{\mathcal{T}}_{\mathcal{S}} p^{\mathcal{S}}_{\mathrm{r},t+1} = T^{\mathcal{T}}_{\mathcal{S}} f(p^{\mathcal{S}}_{\mathrm{r},t}, a^{\mathcal{S}}_t) = T^{\mathcal{T}}_{\mathcal{S}} f(p^{\mathcal{S}}_{\mathrm{r,t}}, \pi_{\mathcal{S}}(p^{\mathcal{S}}_{\mathrm{r,t}}, o^{\mathcal{T}\to\mathcal{S}}))$.

VI. Vision-Based Policy Transfer Experiments

We aim to answer the following questions:

1) Can cross-painting bridge the visual gap between robots?
2) Can Mirage successfully zero-shot transfer trained vision-based policies from one robot to another?
3) To what extent does each component of Mirage affect the transfer performance?

A. Simulation Experiments

We first study the effect of cross-painting on mitigating the visual gap between robots. We focus on closed-loop policies. Based on Table I, we choose UR5e and Kinova Gen3 as two representative robots that show high potential for reusing the source robot policies. We use the ground-truth forward dynamics and compare the success rates of the target robot using cross-inpainting with no inpainting and ground-truth oracle rendering.

To create a sim-to-sim gap, we choose Gazebo [53] as our renderer. We manually position spotlights in the Gazebo environment to simulate the lighting in Robosuite for similar robot renderings. Similar to Sec. IV, we first train the source robot policies with behavior cloning on the provided demonstration data for each task (200 demos for Lift and Can from Robomimic [68], 1000 demos for the other tasks from MimicGen [69]), using the LSTM architecture with the ResNet-18 visual encoder [68]. The policies utilize 84x84 images, and Mirage operates at approximately 40 Hz to cross-paint the images.

Table II shows the results. Similar to Table I, transferring between robots but with the same gripper has higher success rates than transferring to a different gripper in most cases. For UR5e, Mirage has a 8-22% drop in performance compared to querying the source policy on an oracle rendering. There is a similar gap of at most 25% between using an oracle and using Mirage on Kinova Gen3. In all cases, Mirage significantly outperforms the naive 0-shot performance without any visual gap mitigation. This suggests that cross-painting can effectively bridge the visual differences of the robots.

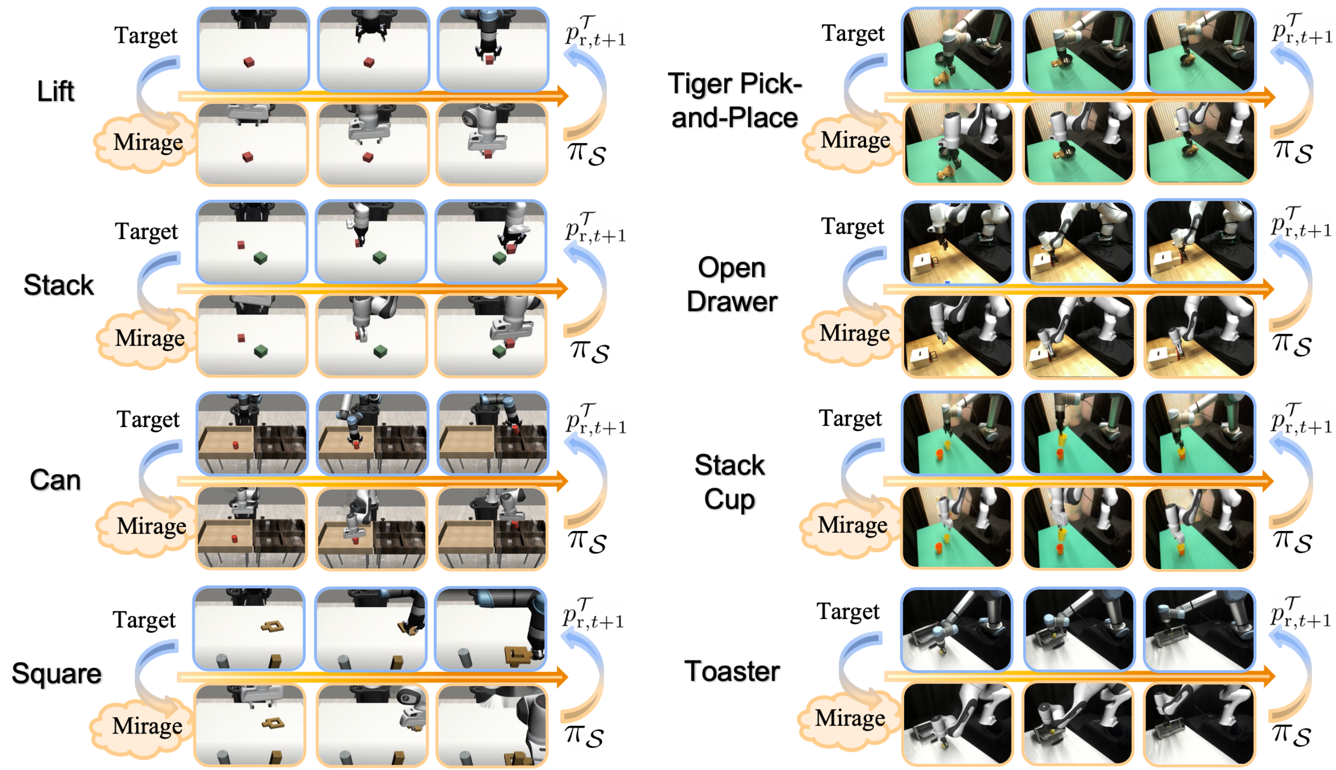

Fig. 4: **Trajectory Rollouts of Simulated (Left) and Real (Right) Tasks.** For each task, the top row shows the actual observations of the target robot during the trajectory rollout, and the bottom row shows the cross-painted images generated by Mirage that are passed to the source robot policy $\pi_{\mathcal{S}}$ to obtain the desired target robot pose $p^{\mathcal{T}}_{r,t+1}$.

Task (Source Gripper)	Source (Franka)		Different Gripper			Different Robot (UR5)		
	Baseline	Octo	Baseline 0-shot	Octo	Mirage	Baseline 0-shot	Octo	Mirage
Tiger Pick-and-Place (Robotiq)	90%	70%	60%	70%	**90%**	50%	10%	**90%**
Open Drawer (Franka)	80%	40%	0%	20%	**70%**	0%	0%	**60%**
Stack Cup (Franka)	80%	30%	10%	20%	**60%**	0%	0%	**50%**
Toaster (Robotiq)	60%	20%	20%	0%	**40%**	0%	0%	**30%**

TABLE III: **Mirage Results on Transferring Vision-Based Policies in Real.** We evaluate Mirage on 4 tasks in 2 settings: **Different Gripper:** Transferring policies between the Franka gripper and the Robotiq 2F-85 gripper on a Franka robot. **Different Robot:** Using the Franka with either gripper as the source robot, and the UR5 robot with the Robotiq gripper as the target robot. **Baseline/Baseline 0-shot:** Separate Diffusion Policy models [17] trained on the source robot data for each task and evaluated on the source robot or zero-shot on the target embodiments. **Octo:** Octo Base model [73] finetuned on the source robot data from all tasks together and evaluated on the source robot or zero-shot on the target embodiments. **Mirage**: Evaluation of zero-shot transfer to the target embodiments using Mirage with the source policy being the corresponding baseline Diffusion Policy models.

B. Physical Experiments

We evaluate Mirage across 3 different embodiments, Franka with Franka and Robotiq 2F-85 grippers, and UR5 with Robotiq 2F-85 gripper. We evaluate on 4 manipulation tasks: (1) Pick up a stuffed animal (tiger) and put it into a bowl [13], (2) open a toy drawer [102], (3) stack one cup into another [13], and (4) put a pepper into a toaster and close its glass door [73]. We select Franka as the source robot. For tasks (2) and (3), the source robot uses the Franka gripper, while for the others, the source robot is equipped with the Robotiq gripper.

We study policy transfer in 2 different settings. The first setting is transferring between grippers only on the Franka robot. The second setting involves robot transfer, where we use the Franka with either gripper as the source robot, and the UR5 with the Robotiq 2F-85 as the target robot. We use one ZED 2 camera positioned from the side for each robot.

For each task, we first use an Oculus controller to collect between 200-400 human demonstration trajectories on the source robot using VR teleoperation at 15 Hz [51]. We then train a separate diffusion policy [17] for each task, which we use as the source policy that we seek to transfer. On the target UR5 robot, we place the camera(s) at similar poses, with up to 4 cm differences from those in the source setup. Similar to Sec. VI-A, we use Gazebo as our renderer. We use a per-dimension linear forward dynamics model and use the demonstration data to fit the regression coefficients.

We compare Mirage to two baselines: naive 0-shot transfer and Octo finetuning [73]. Octo is a recently released state-of-the-art generalist policy for robotic manipulation, pre-trained on 800k robot episodes from the Open X-Embodiment dataset [19], including trajectories collected on Franka and UR5. For both methods, we use the known mapping between the robot coordinate frames to convert the end-effector poses (e.g., there is a 45° offset between the Franka gripper and Robotiq gripper when being installed on the Franka robot). We also use the same fitted forward dynamics model to account for controller differences, and use a blocking controller on the UR5 to reach the desired poses.

In the naive 0-shot transfer baseline, we directly apply the source policy (i.e., the task-specific diffusion policy trained on the source robot only) to the target embodiment for each task, after accounting for the coordinate frame and control differences as mentioned above. For the Octo model, we take the pretrained Base model (93M parameters) and finetune all weights on the combined 4-task demonstration data of the source robot instead of finetuning for each task separately. For both diffusion policy and Octo model, the image observation is the third-person camera view and the proprioception input is the Cartesian position of the end effector and the gripper position. The action output is also the Cartesian position and gripper position. For the Octo model, we use language goal conditioning. We evaluate both baseline models on all 3 robot setups: Source (Franka robot + Franka/Robotiq gripper), Different Gripper (Franka robot + the other gripper), and Different Robot (UR5 + Robotiq gripper).

Table III shows the results. We can see that, for both gripper transfer and robot (and gripper) transfer, Mirage achieves strong zero-shot performance, significantly outperforming both baselines. Similar to Tables I and II, the stronger the source policy is, the smaller the performance gap is during transfer. Interestingly, unlike Tables I and II, we do not observe that transferring between robots with the same gripper achieves higher performance than transferring between grippers only. We hypothesize that this is due to the minor scene setup differences that cause a natural drop in policy performance. On the other hand, the failure modes we observe on the different robots or grippers are all very similar to those from the source policy on the source robot, such as not dropping the cup high enough during cup stacking or not keeping the gripper low enough when closing the door of the toaster oven.

Comparing the two baselines, task-specific Diffusion policies achieve stronger performance on the source robot it is trained on, while Octo achieves better success rates on most tasks when the grippers are swapped. Still, Octo tends to miss more grasps or gets stuck, resulting in a lower success rate than Mirage by 20% to 50%. When being evaluated on the UR5, neither baselines are able to achieve any success on the 3 more difficult tasks.

C. Sensitivity Analysis of Mirage

We examine the sensitivity of policy performance to the various components of Mirage using the Tiger Pick-and-Place

Component	Factor	Variation	Success Rate
Visual	Calibration Error	~10 pixels	80%
		~30 pixels	50%
	Luminance	+50	90%
		-50	80%
	2 cameras	Source	80%
		Target	70%
	Background	Different	0%
Control	Gain	x2	60%
		x0.5	90%
	Offset	8cm	0%

TABLE IV: **Sensitivity of Mirage Components on Policy Performance on the Tiger Pick-and-Place Task.** For the visual components, we study the effect of camera calibration error, the luminance of inpainted robot rendering, changes in the background between the source and target, and the performance on a 2-camera setup. For the control components, we evaluate sensitivity of Mirage to the control gain of the target robot controller and the z-offset in the proprioceptive values.

task. For the visual components, we study the effect of camera calibration error on the target robot, the luminance of the rendered robot, and background changes between the target and source robots. To study the effect of camera calibration, we add offsets to the masks and inpainted robots. As the calibration error increases, the segmentation masks become less accurate and leave parts of the robot unmasked. This leads to cross-painted images to include remnants of the target robot (see Fig. 5(a)). Table IV shows that large artifacts result in non-negligible policy performance degradation. For the luminance effect, we adjust the inpainted pixels' luminance by an offset amount, and Table IV shows that the trained policy is relatively robust to it. For the background change, we evaluate Mirage under a different background (black) from what the policy is trained on (light background), and we find that Mirage does not mitigate the effect of background changes. Additionally, we add another camera in the front view (see Fig. 6), train a Diffusion Policy on both cameras' observations, and apply Mirage to cross-paint both images during test time. Table IV shows that the target embodiment (UR5 + Robotiq) performance is close to the source, indicating that our method is flexible with the number of cameras.

We also examine the sensitivity to the control pipelines of Mirage. To simulate noises in the forward dynamics model and inaccuracy in the target robot's controller, we adjust the control gain and find that large values affect performance. Additionally, when the target robot's proprioceptive values are shifted due to offsets, the trained policy's performance drastically decreases. This is not surprising as the policy is trained with matching proprioceptive values and image observations, and large offsets in the proprioceptive values correspond to distribution shifts in the state input, which Mirage does not mitigate.

Additionally, we evaluate the robustness of Mirage to changes in camera angles between the source and target robot setups, and in other words, the effectiveness of Mirage's camera

(a) Example of Camera Calibration Error

(b) Example of Camera Reprojection with Large Pose Differences

Fig. 5: (a) An example of camera calibration error resulting in failure to mask the target robot out; (b) An example of the artifacts introduced due to large changes in camera angles.

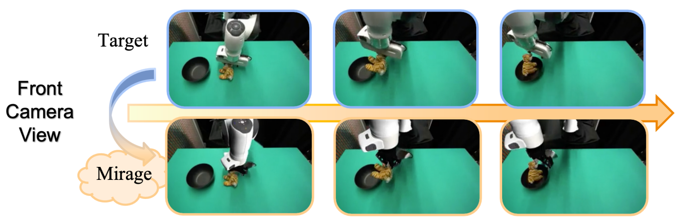

Fig. 6: The front camera angle in the 2-camera experiment in Table IV. **Top:** The actual observations of the target robot during the rollout; **Bottom:** the cross-painted images generated by Mirage.

reprojection step from the target image back to the source camera pose. We select 3 levels of camera pose differences:

- Small Differences: Up to 1 cm in total translation and 5° in total rotation mismatches;
- Medium Differences: Up to 5 cm in total translation and 15° in total rotation mismatches;
- Large Differences: Up to 10 cm in total translation and 15° in total rotation mismatches;

For each level of target camera angles, we compare the performance of the policy with and without the camera reprojection step (Mirage and "w/o reproj." respectively) both on the source robot setup and on the Different Gripper setup. Specifically, for the source robot setup, no cross-painting is needed, so Mirage is effectively just camera reprojection. For the different gripper setup, cross-painting is applied after reprojection, while "w/o reproj." directly applies cross-painting in the target camera pose instead of in the source camera pose.

Table V shows that applying camera reprojection has a 20-30% improvement in the success rates when there are medium to large differences between the camera poses of the source and the target. When the difference increases, there are fewer points projected back to the source image frame, resulting in more missing pixels being inpainted. This leads to blurriness and artifacts close to the boundary of the images (see Fig. 5(b)), making it more likely for the policy to miss the grasp.

VII. Conclusion

We propose Mirage, a novel algorithm for zero-shot transfer of manipulation policies to unseen robot embodiments. Mirage relies on two key ingredients: (1) using robot URDFs and renderers to mitigate the visual differences through cross-painting, and (2) choosing the end effector Cartesian pose as

Camera Pose Change	**Source**		**Different Gripper**	
	w/o reproj.	Mirage	w/o reproj.	Mirage
Small (1 cm + 5°)	90%	90%	90%	90%
Medium (5 cm + 15°)	60%	80%	50%	80%
Large (10 cm + 15°)	30%	50%	20%	40%

TABLE V: **Mirage Effectiveness of Camera Reprojection.** We evaluate Mirage with and without the camera reprojection on the Tiger Pick-and-Place task with 3 levels of camera pose changes: $\leq$ 1 cm and $\leq 5°$, $\leq$ 5 cm and $\leq 15°$, $\leq$ 10 cm and $\leq 15°$. Results suggest that, when there are medium to large differences between camera poses, applying camera reprojection is beneficial.

the state and action spaces and using knowledge of the robot coordinate frames to align the source and target robots. Through both simulation and physical experiments across 9 tasks, we find that Mirage can successfully enable zero-shot transfer across grippers and robot arms, significantly outperforming a state-of-the-art generalist policy.

Limitations and future work. To enable zero-shot transfer, Mirage relies on knowledge of the robots and setups such as the robot URDFs, coordinate frames, and camera matrices. Also, since we use a blocking OSC controller during execution, Mirage is most suitable for quasistatic tasks. Additionally, we do not consider transferring between different backgrounds. Future work can explore combining Mirage with orthogonal approaches such as augmentation and scaling to enable policies to be robust to background changes.

Acknowledgments

This research was performed at the AUTOLab at UC Berkeley in affiliation with the Berkeley AI Research (BAIR) Lab, and the CITRIS "People and Robots" (CPAR) Initiative, and in collaboration with Google DeepMind. The authors are supported in part by donations from Google, Toyota Research Institute, and equipment grants from NVIDIA. L.Y. Chen is supported by the National Science Foundation (NSF) Graduate Research Fellowship Program under Grant No. 2146752. We thank Zehan Ma, Simeon Adebola, Max Fu, Ethan Qiu, and Roy Lin for helping with physical experiments, Karl Pertsch and Sudeep Dasari for support with Octo model finetuning, and Will Panitch and Qianqian Wang for valuable feedback.

References

[1] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob Menick, Sebastian Borgeaud, Andrew Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikolaj Binkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karen Simonyan. Flamingo: a visual language model for few-shot learning, 2022.

[2] Haitham Bou Ammar, Eric Eaton, Paul Ruvolo, and Matthew Taylor. Unsupervised cross-domain transfer in policy gradient reinforcement learning via manifold

alignment. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 29, 2015.

[3] Shikhar Bahl, Abhinav Gupta, and Deepak Pathak. Human-to-robot imitation in the wild. *Robotics: Science and Systems (RSS)*, 2022.

[4] Homanga Bharadhwaj, Jay Vakil, Mohit Sharma, Abhinav Gupta, Shubham Tulsiani, and Vikash Kumar. RoboAgent: Towards sample efficient robot manipulation with semantic augmentations and action chunking. *arxiv*, 2023.

[5] Kevin Black, Mitsuhiko Nakamoto, Pranav Atreya, Homer Walke, Chelsea Finn, Aviral Kumar, and Sergey Levine. Zero-shot robotic manipulation with pre-trained image-editing diffusion models. *arXiv preprint arXiv:2310.10639*, 2023.

[6] Botond Bocsi, Lehel Csató, and Jan Peters. Alignment-based transfer learning for robot models. In *The 2013 international joint conference on neural networks (IJCNN)*, pages 1–7. IEEE, 2013.

[7] Alessandro Bonardi, Stephen James, and Andrew J Davison. Learning one-shot imitation from humans without humans. *IEEE Robotics and Automation Letters*, 5(2):3533–3539, 2020.

[8] Konstantinos Bousmalis, Giulia Vezzani, Dushyant Rao, Coline Devin, Alex X Lee, Maria Bauza, Todor Davchev, Yuxiang Zhou, Agrim Gupta, Akhil Raju, et al. Robocat: A self-improving foundation agent for robotic manipulation. *arXiv preprint arXiv:2306.11706*, 2023.

[9] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.

[10] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, et al. RT-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv preprint arXiv:2307.15818*, 2023.

[11] 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. *Robotics: Science and Systems (RSS)*, 2023.

[12] 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.

[13] Lawrence Yunliang Chen, Simeon Adebola, and Ken Goldberg. Berkeley UR5 demonstration dataset. https://sites.google.com/view/berkeley-ur5/home.

[14] Tao Chen, Adithyavairavan Murali, and Abhinav Gupta. Hardware conditioned policies for multi-robot transfer learning. *Advances in Neural Information Processing Systems*, 31, 2018.

[15] Xi Chen, Josip Djolonga, Piotr Padlewski, Basil Mustafa, Soravit Changpinyo, Jialin Wu, Carlos Riquelme Ruiz, Sebastian Goodman, Xiao Wang, Yi Tay, Siamak Shakeri, Mostafa Dehghani, Daniel Salz, Mario Lucic, Michael Tschannen, Arsha Nagrani, Hexiang Hu, Mandar Joshi, Bo Pang, Ceslee Montgomery, Paulina Pietrzyk, Marvin Ritter, AJ Piergiovanni, Matthias Minderer, Filip Pavetic, Austin Waters, Gang Li, Ibrahim Alabdulmohsin, Lucas Beyer, Julien Amelot, Kenton Lee, Andreas Peter Steiner, Yang Li, Daniel Keysers, Anurag Arnab, Yuanzhong Xu, Keran Rong, Alexander Kolesnikov, Mojtaba Seyedhosseini, Anelia Angelova, Xiaohua Zhai, Neil Houlsby, and Radu Soricut. Pali-x: On scaling up a multilingual vision and language model, 2023.

[16] Zoey Chen, Sho Kiami, Abhishek Gupta, and Vikash Kumar. Genaug: Retargeting behaviors to unseen situations via generative augmentation. *arXiv preprint arXiv:2302.06671*, 2023.

[17] 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.

[18] Girish Chowdhary, Tongbin Wu, Mark Cutler, and Jonathan P How. Rapid transfer of controllers between uavs using learning-based adaptive control. In *2013 IEEE International Conference on Robotics and Automation*, pages 5409–5416. IEEE, 2013.

[19] 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. https://arxiv.org/abs/2310.08864, 2023.

[20] 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.

[21] Amaury Depierre, Emmanuel Dellandréa, and Liming Chen. Jacquard: A large scale dataset for robotic grasp detection. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3511–3516. IEEE, 2018.

[22] Coline Devin, Abhishek Gupta, Trevor Darrell, Pieter Abbeel, and Sergey Levine. Learning modular neural network policies for multi-task and multi-robot transfer. In *2017 IEEE international conference on robotics and automation (ICRA)*, pages 2169–2176. IEEE, 2017.

[23] Danny Driess, Fei Xia, Mehdi S. M. Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, Wenlong Huang, Yevgen Chebotar, Pierre Sermanet, Daniel Duckworth, Sergey Levine, Vincent Vanhoucke, Karol Hausman, Marc Toussaint, Klaus Greff, Andy Zeng, Igor Mordatch, and Pete Florence. PaLM-E: An embodied multimodal language model, 2023.

[24] Jiafei Duan, Yi Ru Wang, Mohit Shridhar, Dieter Fox, and Ranjay Krishna. Ar2-d2: Training a robot without a robot. *arXiv preprint arXiv:2306.13818*, 2023.

[25] 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.

[26] 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. In *Robotics: Science and Systems (RSS) XVIII*, 2022.

[27] Clemens Eppner, Arsalan Mousavian, and Dieter Fox. ACRONYM: A large-scale grasp dataset based on simulation. In *2021 IEEE Int. Conf. on Robotics and Automation, ICRA*, 2020.

[28] Ioannis Exarchos, Yifeng Jiang, Wenhao Yu, and C Karen Liu. Policy transfer via kinematic domain randomization and adaptation. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 45–51. IEEE, 2021.

[29] Hao-Shu Fang, Hongjie Fang, Zhenyu Tang, Jirong Liu, Junbo Wang, Haoyi Zhu, and Cewu Lu. RH20T: A robotic dataset for learning diverse skills in one-shot. In *RSS 2023 Workshop on Learning for Task and Motion Planning*, 2023.

[30] Arnaud Fickinger, Samuel Cohen, Stuart Russell, and Brandon Amos. Cross-domain imitation learning via optimal transport. *arXiv preprint arXiv:2110.03684*, 2021.

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

[32] Tim Franzmeyer, Philip Torr, and João F Henriques. Learn what matters: cross-domain imitation learning with task-relevant embeddings. *Advances in Neural Information Processing Systems*, 35:26283–26294, 2022.

[33] Hiroki Furuta, Yusuke Iwasawa, Yutaka Matsuo, and Shixiang Shane Gu. A system for morphology-task generalization via unified representation and behavior distillation. *arXiv preprint arXiv:2211.14296*, 2022.

[34] Shijie Gao and Nicola Bezzo. A conformal mapping-based framework for robot-to-robot and sim-to-real transfer learning. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1289–1295. IEEE, 2021.

[35] Ali Ghadirzadeh, Xi Chen, Petra Poklukar, Chelsea Finn, Mårten Björkman, and Danica Kragic. Bayesian meta-learning for few-shot policy adaptation across robotic platforms. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1274–1280. IEEE, 2021.

[36] Abhishek Gupta, Coline Devin, YuXuan Liu, Pieter Abbeel, and Sergey Levine. Learning invariant feature spaces to transfer skills with reinforcement learning. *arXiv preprint arXiv:1703.02949*, 2017.

[37] 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.

[38] Donald Hejna, Lerrel Pinto, and Pieter Abbeel. Hierarchically decoupled imitation for morphological transfer. In *International Conference on Machine Learning*, pages 4159–4171. PMLR, 2020.

[39] Mohamed K Helwa and Angela P Schoellig. Multi-robot transfer learning: A dynamical system perspective. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4702–4708. IEEE, 2017.

[40] Noriaki Hirose, Dhruv Shah, Ajay Sridhar, and Sergey Levine. Exaug: Robot-conditioned navigation policies via geometric experience augmentation. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4077–4084. IEEE, 2023.

[41] Edward S Hu, Kun Huang, Oleh Rybkin, and Dinesh Jayaraman. Know thyself: Transferable visual control policies through robot-awareness. *arXiv preprint arXiv:2107.09047*, 2021.

[42] Yang Hu and Giovanni Montana. Skill transfer in deep reinforcement learning under morphological heterogeneity. *arXiv preprint arXiv:1908.05265*, 2019.

[43] Wenlong Huang, Igor Mordatch, and Deepak Pathak. One policy to control them all: Shared modular policies for agent-agnostic control. In *International Conference on Machine Learning*, pages 4455–4464. PMLR, 2020.

[44] Stephen James, Zicong Ma, David Rovick Arrojo, and Andrew J Davison. Rlbench: The robot learning benchmark & learning environment. *IEEE Robotics and Automation Letters*, 5(2):3019–3026, 2020.

[45] Stephen James, Kentaro Wada, Tristan Laidlow, and Andrew J Davison. Coarse-to-fine q-attention: Efficient learning for visual robotic manipulation via discretisation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13739–13748, 2022.

[46] 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 *Conference on Robot Learning (CoRL)*, pages 991–1002, 2021.

[47] Pingcheng Jian, Easop Lee, Zachary Bell, Michael M Zavlanos, and Boyuan Chen. Policy stitching: Learning transferable robot policies. *arXiv preprint arXiv:2309.13753*, 2023.

[48] Yunfan Jiang, Agrim Gupta, Zichen Zhang, Guanzhi Wang, Yongqiang Dou, Yanjun Chen, Li Fei-Fei, Anima Anandkumar, Yuke Zhu, and Linxi Fan. VIMA: General robot manipulation with multimodal prompts. *International Conference on Machine Learning (ICML)*, 2023.

[49] 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.

[50] Oussama Khatib. A unified approach for motion and force control of robot manipulators: The operational space formulation. *IEEE Journal on Robotics and Automation*, 3(1):43–53, 1987.

[51] Alexander Khazatsky. Droid: A large-scale in-the-wild robot manipulation dataset. https://github.com/AlexanderKhazatsky/R2D2, 2023.

[52] Kuno Kim, Yihong Gu, Jiaming Song, Shengjia Zhao, and Stefano Ermon. Domain adaptive imitation learning. In *International Conference on Machine Learning*, pages 5286–5295. PMLR, 2020.

[53] N. Koenig and A. Howard. Design and use paradigms for gazebo, an open-source multi-robot simulator. In *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, volume 3, pages 2149–2154 vol.3, 2004. doi: 10.1109/IROS.2004.1389727.

[54] George Konidaris and Andrew Barto. Autonomous shaping: Knowledge transfer in reinforcement learning. In *Proceedings of the 23rd international conference on Machine learning*, pages 489–496, 2006.

[55] Vitaly Kurin, Maximilian Igl, Tim Rocktäschel, Wendelin Boehmer, and Shimon Whiteson. My body is a cage: the role of morphology in graph-based incompatible control. *arXiv preprint arXiv:2010.01856*, 2020.

[56] Balaji Lakshmanan and Ravindran Balaraman. Transfer learning across heterogeneous robots with action sequence mapping. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3251–3256. IEEE, 2010.

[57] Sungho Lee, Seoung Wug Oh, DaeYeun Won, and Seon Joo Kim. Copy-and-paste networks for deep video inpainting. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4413–4421, 2019.

[58] 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.

[59] Xingyu Liu. Meta-evolve: Continuous robot evolution for one-to-many policy transfer. 2022.

[60] Xingyu Liu, Deepak Pathak, and Kris M Kitani. Revolver: Continuous evolutionary models for robot-to-robot policy transfer. *arXiv preprint arXiv:2202.05244*, 2022.

[61] YuXuan Liu, Abhishek Gupta, Pieter Abbeel, and Sergey Levine. Imitation from observation: Learning to imitate behaviors from raw video via context translation. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1118–1125. IEEE, 2018.

[62] Corey Lynch, Ayzaan Wahid, Jonathan Tompson, Tianli Ding, James Betker, Robert Baruch, Travis Armstrong, and Pete Florence. Interactive language: Talking to robots in real time. *IEEE Robotics and Automation Letters*, 2023.

[63] Yecheng Jason Ma, Shagun Sodhani, Dinesh Jayaraman, Osbert Bastani, Vikash Kumar, and Amy Zhang. Vip: Towards universal visual reward and represen-

tation via value-implicit pre-training. *arXiv preprint arXiv:2210.00030*, 2022.

[64] Ndivhuwo Makondo, Benjamin Rosman, and Osamu Hasegawa. Knowledge transfer for learning robot models via local procrustes analysis. In *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*, pages 1075–1082. IEEE, 2015.

[65] Ndivhuwo Makondo, Benjamin Rosman, and Osamu Hasegawa. Accelerating model learning with inter-robot knowledge transfer. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2417–2424. IEEE, 2018.

[66] Ashish Malik. Zero-shot generalization using cascaded system-representations. *arXiv preprint arXiv:1912.05501*, 2019.

[67] Zhao Mandi, Homanga Bharadhwaj, Vincent Moens, Shuran Song, Aravind Rajeswaran, and Vikash Kumar. Cacti: A framework for scalable multi-task multi-scene visual imitation learning. *arXiv preprint arXiv:2212.05711*, 2022.

[68] 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 *arXiv preprint arXiv:2108.03298*, 2021.

[69] Ajay Mandlekar, Soroush Nasiriany, Bowen Wen, Iretiayo Akinola, Yashraj Narang, Linxi Fan, Yuke Zhu, and Dieter Fox. Mimicgen: A data generation system for scalable robot learning using human demonstrations. In *7th Annual Conference on Robot Learning*, 2023.

[70] Mayank Mittal, Calvin Yu, Qinxi Yu, Jingzhou Liu, Nikita Rudin, David Hoeller, Jia Lin Yuan, Ritvik Singh, Yunrong Guo, Hammad Mazhar, Ajay Mandlekar, Buck Babich, Gavriel State, Marco Hutter, and Animesh Garg. Orbit: A unified simulation framework for interactive robot learning environments. *IEEE Robotics and Automation Letters*, 8(6):3740–3747, 2023. doi: 10.1109/LRA.2023.3270034.

[71] Suraj Nair, Aravind Rajeswaran, Vikash Kumar, Chelsea Finn, and Abhinav Gupta. R3m: A universal visual representation for robot manipulation. In *CoRL*, 2022.

[72] Yuki Noguchi, Tatsuya Matsushima, Yutaka Matsuo, and Shixiang Shane Gu. Tool as embodiment for recursive manipulation. *arXiv preprint arXiv:2112.00359*, 2021.

[73] Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Charles Xu, Jianlan Luo, Tobias Kreiman, You Liang Tan, Dorsa Sadigh, Chelsea Finn, and Sergey Levine. Octo: An open-source generalist robot policy. https://octo-models.github.io, 2023.

[74] Deepak Pathak, Christopher Lu, Trevor Darrell, Phillip Isola, and Alexei A Efros. Learning to control self-assembling morphologies: a study of generalization via modularity. *Advances in Neural Information Processing Systems*, 32, 2019.

[75] Xue Bin Peng, Erwin Coumans, Tingnan Zhang, Tsang-Wei Lee, Jie Tan, and Sergey Levine. Learning agile robotic locomotion skills by imitating animals. *arXiv preprint arXiv:2004.00784*, 2020.

[76] Ilija Radosavovic, Tete Xiao, Stephen James, Pieter Abbeel, Jitendra Malik, and Trevor Darrell. Real-world robot learning with masked visual pre-training. In *Conference on Robot Learning*, 2022.

[77] Ilija Radosavovic, Baifeng Shi, Letian Fu, Ken Goldberg, Trevor Darrell, and Jitendra Malik. Robot learning with sensorimotor pre-training. In *Conference on Robot Learning*, 2023.

[78] Kaizad V Raimalwala, Bruce A Francis, and Angela P Schoellig. A preliminary study of transfer learning between unicycle robots. In *2016 AAAI Spring Symposium Series*, 2016.

[79] Dripta S Raychaudhuri, Sujoy Paul, Jeroen Vanbaar, and Amit K Roy-Chowdhury. Cross-domain imitation from observations. In *International Conference on Machine Learning*, pages 8902–8912. PMLR, 2021.

[80] Scott Reed, Konrad Zolna, Emilio Parisotto, Sergio Gómez Colmenarejo, Alexander Novikov, Gabriel Barth-maron, Mai Giménez, Yury Sulsky, Jackie Kay, Jost Tobias Springenberg, Tom Eccles, Jake Bruce, Ali Razavi, Ashley Edwards, Nicolas Heess, Yutian Chen, Raia Hadsell, Oriol Vinyals, Mahyar Bordbar, and Nando de Freitas. A generalist agent. *Transactions on Machine Learning Research*, 2022. ISSN 2835-8856.

[81] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10684–10695, June 2022.

[82] Andrei A Rusu, Matej Večerík, Thomas Rothörl, Nicolas Heess, Razvan Pascanu, and Raia Hadsell. Sim-to-real robot learning from pixels with progressive nets. In *Conference on robot learning*, pages 262–270. PMLR, 2017.

[83] Gautam Salhotra, I Liu, Chun Arthur, and Gaurav Sukhatme. Bridging action space mismatch in learning from demonstrations. *arXiv preprint arXiv:2304.03833*, 2023.

[84] Alvaro Sanchez-Gonzalez, Nicolas Heess, Jost Tobias Springenberg, Josh Merel, Martin Riedmiller, Raia Hadsell, and Peter Battaglia. Graph networks as learnable physics engines for inference and control. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 4470–4479. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/sanchez-gonzalez18a.html.

[85] Karl Schmeckpeper, Oleh Rybkin, Kostas Daniilidis, Sergey Levine, and Chelsea Finn. Reinforcement learning with videos: Combining offline observations with interaction. In *Conference on Robot Learning*,

pages 339–354. PMLR, 2021.

[86] Ingmar Schubert, Jingwei Zhang, Jake Bruce, Sarah Bechtle, Emilio Parisotto, Martin Riedmiller, Jost Tobias Springenberg, Arunkumar Byravan, Leonard Hasenclever, and Nicolas Heess. A generalist dynamics model for control, 2023.

[87] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[88] Nur Muhammad Mahi Shafiullah, Anant Rai, Haritheja Etukuru, Yiqian Liu, Ishan Misra, Soumith Chintala, and Lerrel Pinto. On bringing robots home, 2023.

[89] Dhruv Shah, Ajay Sridhar, Arjun Bhorkar, Noriaki Hirose, and Sergey Levine. GNM: A general navigation model to drive any robot. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7226–7233. IEEE, 2023.

[90] Dhruv Shah, Ajay Sridhar, Nitish Dashora, Kyle Stachowicz, Kevin Black, Noriaki Hirose, and Sergey Levine. ViNT: A Foundation Model for Visual Navigation. In *7th Annual Conference on Robot Learning (CoRL)*, 2023.

[91] Tanmay Shankar, Yixin Lin, Aravind Rajeswaran, Vikash Kumar, Stuart Anderson, and Jean Oh. Translating robot skills: Learning unsupervised skill correspondences across robots. In *International Conference on Machine Learning*, pages 19626–19644. PMLR, 2022.

[92] Lin Shao, Fabio Ferreira, Mikael Jorda, Varun Nambiar, Jianlan Luo, Eugen Solowjow, Juan Aparicio Ojea, Oussama Khatib, and Jeannette Bohg. Unigrasp: Learning a unified model to grasp with multifingered robotic hands. *IEEE Robotics and Automation Letters*, 5(2):2286–2293, 2020.

[93] Mohit Shridhar, Lucas Manuelli, and Dieter Fox. Cliport: What and where pathways for robotic manipulation. In *Conference on Robot Learning*, pages 894–906. PMLR, 2022.

[94] Mohit Shridhar, Lucas Manuelli, and Dieter Fox. Perceiver-actor: A multi-task transformer for robotic manipulation. In *Proceedings of the 6th Conference on Robot Learning (CoRL)*, 2022.

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

[96] Laura Smith, Nikita Dhawan, Marvin Zhang, Pieter Abbeel, and Sergey Levine. Avid: Learning multi-stage tasks via pixel-level translation of human videos. *arXiv preprint arXiv:1912.04443*, 2019.

[97] Austin Stone, Ted Xiao, Yao Lu, Keerthana Gopalakrishnan, Kuang-Huei Lee, Quan Vuong, Paul Wohlhart, Brianna Zitkovich, Fei Xia, Chelsea Finn, et al. Open-world object manipulation using pre-trained vision-language models. *arXiv preprint arXiv:2303.00905*, 2023.

[98] Yanchao Sun, Ruijie Zheng, Xiyao Wang, Andrew Cohen, and Furong Huang. Transfer rl across observation feature spaces via model-based regularization. *arXiv preprint arXiv:2201.00248*, 2022.

[99] Matthew E Taylor and Peter Stone. Transfer learning for reinforcement learning domains: A survey. *Journal of Machine Learning Research*, 10(7), 2009.

[100] Alexandru Telea. An image inpainting technique based on the fast marching method. *Journal of graphics tools*, 9(1):23–34, 2004.

[101] Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 23–30. IEEE, 2017.

[102] Homer Walke, Kevin Black, Abraham Lee, Moo Jin Kim, Max Du, Chongyi Zheng, Tony Zhao, Philippe Hansen-Estruch, Quan Vuong, Andre He, Vivek Myers, Kuan Fang, Chelsea Finn, and Sergey Levine. Bridgedata v2: A dataset for robot learning at scale, 2023.

[103] Tingwu Wang, Renjie Liao, Jimmy Ba, and Sanja Fidler. Nervenet: Learning structured policy with graph neural networks. In *International conference on learning representations*, 2018.

[104] Chuan Wen, Xingyu Lin, John So, Kai Chen, Qi Dou, Yang Gao, and Pieter Abbeel. Any-point trajectory modeling for policy learning. *arXiv preprint arXiv:2401.00025*, 2023.

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

[106] 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. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7827–7834. IEEE, 2021.

[107] Zheng Xiong, Jacob Beck, and Shimon Whiteson. Universal morphology control via contextual modulation. *arXiv preprint arXiv:2302.11070*, 2023.

[108] Mengda Xu, Zhenjia Xu, Cheng Chi, Manuela Veloso, and Shuran Song. XSkill: Cross embodiment skill discovery. *arXiv preprint arXiv:2307.09955*, 2023.

[109] Zhenjia Xu, Beichun Qi, Shubham Agrawal, and Shuran Song. Adagrasp: Learning an adaptive gripper-aware grasping policy. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4620–4626. IEEE, 2021.

[110] Jonathan Yang, Dorsa Sadigh, and Chelsea Finn. Polybot: Training one policy across robots while embracing variability. *arXiv preprint arXiv:2307.03719*, 2023.

[111] Zhao-Heng Yin, Lingfeng Sun, Hengbo Ma, Masayoshi Tomizuka, and Wu-Jun Li. Cross domain robot imitation with invariant representation. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 455–461. IEEE, 2022.

[112] Chen Yu, Weinan Zhang, Hang Lai, Zheng Tian, Laurent Kneip, and Jun Wang. Multi-embodiment legged robot control as a sequence modeling problem. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7250–7257. IEEE, 2023.

[113] Tao Yu, Runseng Feng, Ruoyu Feng, Jinming Liu, Xin Jin, Wenjun Zeng, and Zhibo Chen. Inpaint anything: Segment anything meets image inpainting. *arXiv preprint arXiv:2304.06790*, 2023.

[114] Tianhe Yu, Chelsea Finn, Sudeep Dasari, Annie Xie, Tianhao Zhang, Pieter Abbeel, and Sergey Levine. One-shot imitation from observing humans via domain-adaptive meta-learning. *Robotics: Science and Systems XIV*, 2018.

[115] Tianhe Yu, Ted Xiao, Austin Stone, Jonathan Tompson, Anthony Brohan, Su Wang, Jaspiar Singh, Clayton Tan, Jodilyn Peralta, Brian Ichter, et al. Scaling robot learning with semantically imagined experience. *arXiv preprint arXiv:2302.11550*, 2023.

[116] Kevin Zakka, Andy Zeng, Pete Florence, Jonathan Tompson, Jeannette Bohg, and Debidatta Dwibedi. Xirl: Cross-embodiment inverse reinforcement learning. In *Conference on Robot Learning*, pages 537–546. PMLR, 2022.

[117] Grace Zhang, Linghan Zhong, Youngwoon Lee, and Joseph J Lim. Policy transfer across visual and dynamics domain gaps via iterative grounding. *arXiv preprint arXiv:2107.00339*, 2021.

[118] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models.

[119] Qiang Zhang, Tete Xiao, Alexei A Efros, Lerrel Pinto, and Xiaolong Wang. Learning cross-domain correspondence for control with dynamics cycle-consistency. *arXiv preprint arXiv:2012.09811*, 2020.

[120] Yifan Zhou, Shubham Sonawani, Mariano Phielipp, Simon Stepputtis, and Heni Amor. Modularity through attention: Efficient training and transfer of language-conditioned policies for robot manipulation. In Karen Liu, Dana Kulic, and Jeff Ichnowski, editors, *Proceedings of The 6th Conference on Robot Learning*, volume 205 of *Proceedings of Machine Learning Research*, pages 1684–1695. PMLR, 14–18 Dec 2023. URL https://proceedings.mlr.press/v205/zhou23b.html.

[121] Yuxiang Zhou, Yusuf Aytar, and Konstantinos Bousmalis. Manipulator-independent representations for visual imitation. *arXiv preprint arXiv:2103.09016*, 2021.

[122] Yuke Zhu, Josiah Wong, Ajay Mandlekar, Roberto Martín-Martín, Abhishek Joshi, Soroush Nasiriany, and Yifeng Zhu. robosuite: A modular simulation framework and benchmark for robot learning. In *arXiv preprint arXiv:2009.12293*, 2020.

[123] Zhuangdi Zhu, Kaixiang Lin, Anil K Jain, and Jiayu Zhou. Transfer learning in deep reinforcement learning: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

Appendix

We provide the following additional results and details in this appendix:

- Section A: State-based transfer experiment results, using the same delta actions on the target robot instead of a blocking controller, and trajectory playback results;
- Section B: Additional implementation details of the cross-painting step of Mirage;
- Section C: Additional experiments using diffusion models to improve the visual qualities of the cross-painted robot.

Appendix A
Delta Control and Trajectory Playback

In Section IV-B Table I, we see that target robots can achieve high task success rates by querying the source robot policy and referencing the source robot for desired poses. Specifically, we use a blocking controller with absolute pose commands. In this section, we compare and experiment with executing the same delta action from the policy directly on the target robot with a non-blocking controller. Table VI shows that there is a significant drop in performance, indicating that the difference in the forward dynamics between robots cannot be ignored when transferring policies and that leveraging a blocking controller on the target robot is an effective way to mitigate this difference.

Additionally, we compare querying a source robot policy with playing back successful source robot trajectories. Since we have shown that executing the same delta actions as the source robot does not transfer well, we use the achieved poses by the source robots in the demonstrated trajectories paired with blocking Cartesian controllers on the target robots. Table VII shows the results. We see that, compared with Table I, the performances are similar to those using the source robot policies. This suggests that these tasks can be achieved by those target robots using the same strategy. In fact, this is similar to the data generation strategy adopted by MimicGen [69], in which the authors roll out the (transformed) trajectories on the target robots and filter out those unsuccessful ones and then train a model on the rest. Comparing playing back human demonstration trajectories and querying policies on the source robot, we observe that in some cases where the learned policies are brittle or suboptimal, they leave less room for error and are less robust to changes in the gripper or tracking errors. On the other hand, good learned policies can potentially adapt to tracking errors, adjust their poses, and retry grasps if necessary, thus outperforming trajectory playback in some cases.

Appendix B
Additional Implementation Details of Mirage

In this section, we detail the inpainting step in the cross-painting process of Mirage, and its sensitivity to calibration errors. Once we have used the segmentation mask provided by Gazebo to black out the target robot, we use Fast Marching [100] algorithm to fill in the black pixels. Specifically, we use a neighborhood radius of 3 pixels and the Telea inpainting method, which is based on a fast marching algorithm [100].

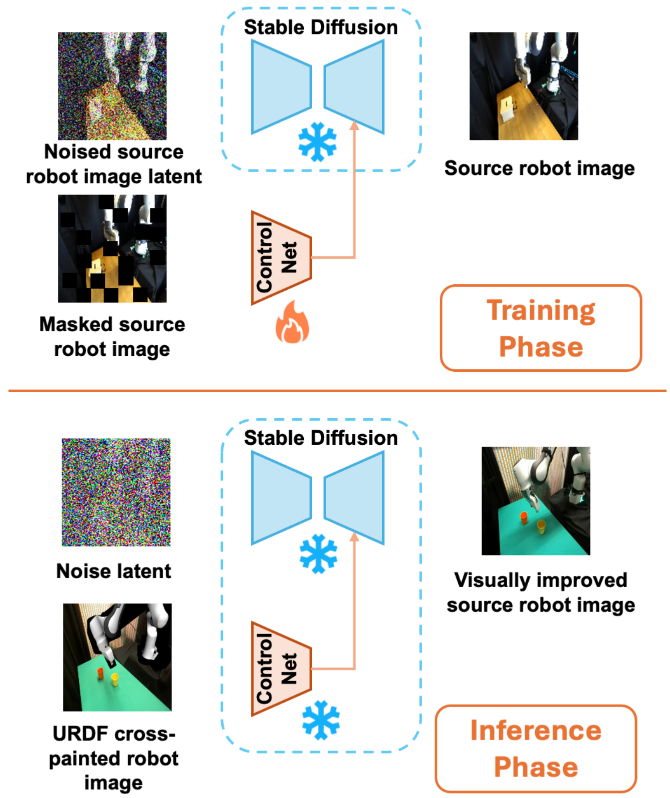

Fig. 7: Robot cross-painting with diffusion generation. We utilize a ControlNet to transfer the pre-trained Stable Diffusion into the robotics domain. The upper part presents the training phase, in which we finetune the parameters of ControlNet while keeping the Stable Diffusion frozen using a masked reconstruction objective. The bottom part presents the inference phase, where a URDF.

While the filled-in pixels appear blurry, they generally match the color scheme of the rest of the image when the calibration error is low. However, with higher calibration errors, the segmentation masks become less accurate, resulting in remnants of the target robot in the cross-painted images (see Fig. 5(a)). To mitigate this issue, we dilate the segmentation masks to ensure that the target robot is completely blacked out even with calibration errors. With moderate to low calibration errors, we dilate the robot arm segmentation mask with a 3x3 kernel for 20 iterations and the gripper segmentation mask for 10 iterations. However, when the calibration error is higher, we dilate the arm and gripper segmentation masks by 40 and 20 iterations respectively. The gripper mask is dilated significantly less than the robot arm mask because the gripper is located near the objects we perform tasks on. Too much dilation would blur the objects that the robot needs to interact with, leading to worse performance.

Appendix C
Improving Visual Qualities with Diffusion Models

Considering the advanced state of diffusion models in generating high-quality images, it is natural to ask whether we can leverage diffusion models to bridge the visual gaps. Specifically, there is still a small visual gap between the

Task	Source (Franka)	Franka Gripper on Target Robot					Default Gripper on Target Robot				
		UR5e	IIWA	Kinova Gen3	Sawyer	Jaco	UR5e	IIWA	Kinova Gen3	Sawyer	Jaco
Lift (Robosuite)	100%	91%	94%	96%	98%	96%	99%	53%	98%	72%	63%
Stack	95%	36%	39%	74%	60%	36%	41%	28%	43%	40%	38%
Can	98%	87%	93%	89%	78%	86%	90%	25%	73%	41%	60%
Two Piece Assembly	96%	44%	65%	54%	43%	41%	44%	45%	48%	29%	15%
Square	81%	10%	15%	3%	5%	9%	6%	1%	2%	5%	0%

TABLE VI: **State-Based Transfer Experiment with Delta Actions.** We evaluate state-based policies trained on a Franka robot for each task across five different robots equipped either with the original Franka gripper or with each target robot's default gripper. The target robot executes the same delta Cartesian action as the source Franka robot instead of reaching the same absolute pose.

Task	Source (Franka)	UR5e	IIWA	Kinova Gen3	Sawyer	Jaco
Lift (Robosuite)	100%	100%	99%	100%	85%	90%
Stack	100%	100%	100%	89%	76%	50%
Can	100%	80%	75%	50%	80%	54%
Two Piece Assembly	100%	98%	90%	94%	93%	69%
Square	100%	73%	43%	27%	82%	2%

TABLE VII: **Success Rates of Target Robots Playing Back the Source Robot Trajectories.** Instead of executing the source robot policy, we evaluate the success rates of target robots performing trajectory playback. Specifically, we use the robot's blocking controller with the achieved pose of the source robot as the target pose instead of performing the same delta actions.

rendered URDF $o^{\mathcal{T}\rightarrow\mathcal{S}}$ and the oracle image $o^{\mathcal{S}}$. However, since the large-scale image-text pair LAION dataset does not contain many robotics images, pre-trained Stable Diffusion [81] models suffer from generalizing to robotics domain zero-shot.

We explore using diffusion models to improve the visual quality of the URDF cross-painted images of Mirage. Specifically, instead of directly finetuning the original Stable Diffusion model, we tune a ControlNet [118] to bridge this gap, as shown in Fig. 7. Additionally, we adopt a self-supervised masking-reconstruction training paradigm [37]. Specifically, during training, the ControlNet takes in a masked image of the source robot $o^{\mathcal{S}}$ and the goal is to reconstruct $o^{\mathcal{S}}$. During inference, we condition the ControlNet on a URDF cross-painted image $o^{\mathcal{T}\rightarrow\mathcal{S}}$ but without the fast marching inpaint step (i.e., leaving the masked out pixels that are not covered by the source robot as black) to generate a visually improved image that is ideally closer to $o^{\mathcal{S}}$.

We qualitatively experiment with and without adding the diffusion model on top of the URDF cross-painted images. While we find that the robot looks visually more realistic than the analytically rendered images, the policy performance is about the same. Since diffusion generation takes a significant amount of time at each time step, we opt not to integrate that part into Mirage pipeline. We leave further investigation of the potential benefit of diffusion models to future work.