# RoVi-Aug: Robot and Viewpoint Augmentation for Cross-Embodiment Robot Learning

**Lawrence Yunliang Chen**[*1], **Chenfeng Xu**[*1], **Karthik Dharmarajan**[1], **Richard Cheng**[2], **Kurt Keutzer**[1], **Masayoshi Tomizuka**[1], **Quan Vuong**[3], **Ken Goldberg**[1]
[1]UC Berkeley [2]Toyota Research Institute [3]Physical Intelligence

**Abstract:** Scaling up robot learning requires large and diverse datasets, and how to efficiently reuse collected data and transfer policies to new embodiments remains an open question. Emerging research such as the Open-X Embodiment (OXE) project has shown promise in leveraging skills by combining datasets including different robots. However, imbalances in the distribution of robot types and camera angles in many datasets make policies prone to overfit. To mitigate this issue, we propose RoVi-Aug, which leverages state-of-the-art image-to-image generative models to augment robot data by synthesizing demonstrations with different robots and camera views. Through extensive physical experiments, we show that, by training on robot- and viewpoint-augmented data, RoVi-Aug can zero-shot deploy on an unseen robot with significantly different camera angles. Compared to test-time adaptation algorithms such as Mirage, RoVi-Aug requires no extra processing at test time, does not assume known camera angles, and allows policy fine-tuning. Moreover, by co-training on both the original and augmented robot datasets, RoVi-Aug can learn multi-robot and multi-task policies, enabling more efficient transfer between robots and skills and improving success rates by up to 30%. Project website: https://rovi-aug.github.io.

**Keywords:** Cross-Embodiment Learning, Viewpoint Robust, Data Augmentation

## 1 Introduction

Emerging research in robot learning suggests that scaling up data can help learned policies be more generalizable and robust [1, 2, 3, 4, 5, 6, 7, 8, 9]. However, compared to state-of-the-art foundation models [10] in computer vision (CV) [11, 12, 13, 14] and natural language processing (NLP), the size of robotic data is still several orders of magnitude smaller than those used to train large language and multi-modal models [15, 16, 17, 18, 19]. Collecting real robot data is time-consuming [20, 21, 22] and labor intensive [2, 3, 5, 23, 24], and ensuring data diversity for generalizable policies requires careful balance [25]. Can we more effectively leverage currently available real robot data?

In an unprecedented community effort, the Open-X Embodiment (OXE) project [9] combines 60 robot datasets and finds that co-training can exhibit positive transfer and improves the capabilities of multiple robots by leveraging experience from each other. However, the OXE dataset is highly unbalanced, dominated by a few robot types such as Franka and xArm. Additionally, most datasets have a limited diversity of camera poses. Policies trained on such data tend to overfit to those robot types and viewpoints and need fine-tuning when deploying on other robots or at even slightly different camera angles. To mitigate this issue, a test-time adaptation algorithm, Mirage [26], uses "cross-painting" to transform an unseen target robot into the source robot seen during training, to create an illusion as if the source robot is performing the task at test time. While Mirage can achieve zero-shot transfer on unseen target robots, it has a few limitations: (1) It requires precise robot models and camera matrices; (2) It does not allow policy finetuning; (3) It is limited to small camera pose changes due to depth reprojection error.

8th Conference on Robot Learning (CoRL 2024), Munich, Germany.


Figure 1: Given robot images, RoVi-Aug uses state-of-the-art diffusion models to augment the data and generate synthetic images with different robots and viewpoints. Policy trained on the augmented dataset can be deployed on the target robots zero-shot or further finetuned, exhibiting robustness to camera pose changes.

In this work, we seek to bridge these limitations. Rather than naively co-training on combined data from multiple robots, we aim to more explicitly encourage the model to learn the cross-product of the robots and skills contained in each dataset. We aim to improve the robustness and generalizability of the policy to different robot visuals and camera poses during training instead of relying on an accurate test-time cross-painting pipeline. We propose RoVi-Aug, a robot and viewpoint augmentation pipeline that synthetically generates images with different robot types and camera poses using diffusion models. Through extensive real-world experiments, we show that, by training on robot- and viewpoint-augmented data, RoVi-Aug can zero-shot control different robots with significantly different camera poses compared to the poses seen during training. In contrast to Mirage, RoVi-Aug does not assume known camera matrices and allows policy fine-tuning to increase performance on challenging tasks. Furthermore, by co-training on original and augmented robot datasets, RoVi-Aug can learn multi-robot and multi-task policies and improve finetuning sample efficiency.

This paper makes 3 contributions:

1. RoVi-Aug, a novel approach to robot data augmentation that uses diffusion models to generate trajectories with novel robots and viewpoints;
2. Physical experiments with Franka and UR5 suggesting that robot augmentation enables zero-shot deployment on target robots and viewpoint augmentation improves the robustness of policies to camera pose changes. When combined, they yield policies that work for target robots at camera poses significantly different from those in the initial demonstration data;
3. Experiments suggesting that RoVi-Aug can learn multi-robot multi-task policies and improve the finetuning sample efficiency of a generalist policy on novel robot-task combinations.

## 2 Related Work

### 2.1 Cross-Embodiment Robot Learning

Recognizing the high cost of collecting real robot data, many prior works have studied using other data sources, such as simulation [27, 28, 29, 30, 31, 32], other robot data [33, 34], and human or animal videos [35, 36, 37, 38, 39, 40, 41, 35, 42, 43, 44, 45, 46, 47, 48], to increase sample efficiency and accelerate learning [49]. In a transfer learning setting, one can first pretrain a visual encoder [50], dynamics model [51], or policy [52, 53, 54] and then perform online finetuning using reinforcement learning. In a cross-domain imitation paradigm, methods often involve learning correspondences between the source and target domains [55, 56, 57, 58], and then constructing auxiliary rewards [59, 55, 60, 61] or applying adversarial training [62, 63, 64]. Ghadirzadeh et al. [65] use meta-learning to enable a new robot to quickly learn from few-shot trajectories at test time.

Cross-embodiment learning could also be used to learn more robust and generalizable policies through joint training in a multi-robot multi-task fashion. For example, by training on a family of robots with varying kinematics and dynamics in simulation, robot-conditioned policies [66, 33, 67, 68, 69, 70, 71, 72, 73, 74] are robust to novel morphologies within the range of training

distribution, and modular policies [1, 75, 76, 77, 78] can be more transferrable to different robots and tasks. More recently, many works have also explored training on large and diverse real robot data [79, 80, 81, 82, 24, 23, 83, 84] to learn visual representations [85, 86, 87] and predictive world models [88, 89, 90] and showed that policies trained are more generalizable to new objects, scenes, tasks, and embodiments [2, 3, 4, 6, 7, 8, 91, 92, 93, 94, 95, 96, 97, 98, 99]. In this work, we build on these insights and propose to more explicitly encourage positive transfer between robots and skills by performing data augmentation.

Our method is inspired by Mirage [26], a recent test-time adaptation algorithm that uses "cross-painting" to achieve cross-embodiment policy transfer by replacing the target robot in the image with a source robot seen during training. While Mirage avoids modifying the source robot policy and enables zero-shot transfer, it has several limitations, such as requiring a fast renderer, precise robot models, and accurate camera calibration. We address these issues by using training time data augmentation with diffusion models trained on randomized robot poses and camera angles, eliminating the need for camera matrix knowledge. Our approach additionally allows zero-shot deployment as well as finetuning or cotraining on additional data to improve the performance and learn multi-robot multi-skill policies that are robust to significant camera angle changes.

### 2.2 Generative Models and Data Augmentation in Robotics

With the significant progress in generative models including large language and multi-modal models [15, 16, 99, 100] and diffusion models [101, 11, 102, 103] trained on Internet-scale data, there is a growing interest in leveraging these models for robotics. For example, prior work has explored using language models for planning [104, 105, 106, 107, 108], control [109], reward specification [110, 111], and data relabeling [112]. Image and video generation models have been used for generative simulation [113, 114], data augmentation [115, 116, 117, 97] and visual goal planning [89, 118]. Our method falls into the data augmentation category. However, unlike prior work that generates distractor objects, backgrounds, and new tasks [115, 116, 117, 97], we use diffusion models to generate alternative robots and camera viewpoints. As such, RoVi-Aug enables trained policies to generalize to different robots with different camera setups.

### 2.3 Viewpoint Adaptation and Viewpoint Robust Policy

Visuomotor control policies that take in images as inputs tend to overfit to the camera angle in the training data, and even small changes between training and testing could severely hurt performance [26, 119]. While using 3D representations [94, 120] alleviates the problem, it requires a calibrated depth camera or multiple views [94, 121], and is more computationally expensive. For mobile robots, Hirose et al. [122] extract a 3D point cloud from the training data and performs re-rendering, and Ex-DoF [123] applies virtual rotation of the robot's 360° camera to augment training data. To improve viewpoint robustness of image-based policies, Sadeghi et al. [119] use a recurrent neural network to understand how actions affect arm movement through history. Seo et al. [124] use many simulated viewpoints to learn a visual representation, whose downstream policy exhibits viewpoint robustness. Instead of pretraining in simulation with diverse rendering, we synthesize novel views of real scenes. SPARTN [125] and DMD [126] use neural radiance fields (NeRFs) and diffusion models, respectively, to generate perturbed viewpoints for wrist cameras, whereas our viewpoint augmentation applies to fixed third-person views.

## 3 Problem Statement

We assume a demonstration dataset $\mathcal{D}^{\mathcal{S}} = \{\tau_1^{\mathcal{S}}, \tau_2^{\mathcal{S}}, ..., \tau_n^{\mathcal{S}}\}$ consisting of $n$ successful trajectories of a source robot $\mathcal{S}$ performing some task. Each trajectory $\tau_i^{\mathcal{S}} = (\{o_{1..H_i}^{\mathcal{S}}\}, \{p_{1..H_i}^{\mathcal{S}}\}, \{a_{1..H_i}^{\mathcal{S}}\})$, where $\{o_1^{\mathcal{S}}, ..., o_{H_i}^{\mathcal{S}}\}$ is a sequence of RGB camera observations, $\{p_1^{\mathcal{S}}, ..., p_{H_i}^{\mathcal{S}}\}$ is the sequence of corresponding gripper poses, and $\{a_1^{\mathcal{S}}, ..., a_{H_i}^{\mathcal{S}}\}$ is the sequence of corresponding robot actions. This dataset can be used to train models with behavior cloning for robot $\mathcal{S}$. Our goal is to augment $\mathcal{D}^{\mathcal{S}}$ into $\mathcal{D}^{\text{Aug}}$ such that we can learn a policy that can be successfully deployed on a different robot $\mathcal{T}$, known as the target robot, with a potentially different camera viewpoint. In this work, we focus on robot arms mounted on a stationary base and assume the grippers are similar in shape and function.

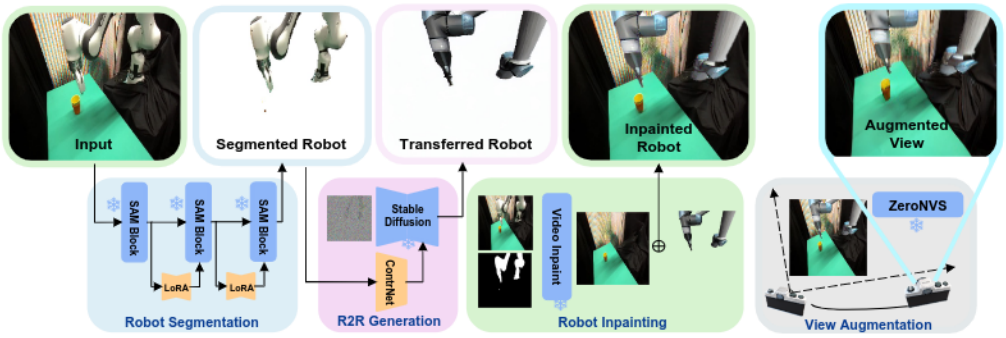


Figure 2: **Overview of the RoVi-Aug pipeline.** Given an input robot image, we first segment the robot out using a finetuned SAM [129] model, then use a ControlNet [130] to transform the robot into another robot. After pasting the synthetic robot back into the background, we use ZeroNVS [131] to generate novel views.

Similar to prior work [7, 26, 127, 128], we use Cartesian control and assume knowledge of the two robots' end effector coordinate frames with respect to their bases (e.g., moving forward corresponds to an increase in the $x$-axis) such that we can use a rigid transformation $T_{\mathcal{T}}^{\mathcal{S}}$ to preprocess the data and align the robots' end effector poses $p^{\mathcal{S}} = T_{\mathcal{T}}^{\mathcal{S}} p^{\mathcal{T}}$ and actions $a^{\mathcal{S}} = T_{\mathcal{T}}^{\mathcal{S}} a^{\mathcal{T}}$ into the same vector space. Thus, for notational convenience, we omit the superscript differentiating gripper poses and actions between the robots. However, the image observations $o^{\mathcal{S}}$ and $p^{\mathcal{T}}$ cannot be easily aligned since the robots may look very different. We do not assume knowledge of the camera matrices in either setup.

After augmentation, we learn a policy $\pi(a_t | o_t^{\mathcal{T}}, p_t)$ on $\mathcal{D}^{\text{Aug}}$ using imitation learning. At test time, it takes as inputs the observations from the target robot and outputs actions that can be deployed on the target robot. Additionally, by co-training on the original data $\mathcal{D}^{\mathcal{S}}$ as well as $\mathcal{D}^{\text{Aug}}$, we can also obtain a multi-robot policy.

## 4 RoVi-Aug

In this section, we describe RoVi-Aug, an automated pipeline for augmenting and scaling up robot data. Our key insight is that the robot's actions should be invariant to its visual appearances and camera viewpoints. Our robot augmentation pipeline leverages state-of-the-art diffusion models [11, 129] to synthesize alternative robots and novel viewpoints. Fig. 2 illustrates RoVi-Aug pipeline.

### 4.1 Robot Augmentation (Ro-Aug)

Given a sequence of robot image observations $D_i^{\mathcal{S}} = \{o_1^{\mathcal{S}}, ..., o_{H_i}^{\mathcal{S}}\}$, we seek to transform the robot $\mathcal{S}$ in the images into a different robot $\mathcal{T}$ at the same gripper pose, a process known as cross-painting. While Mirage [26] proposes to perform cross-painting using a renderer to compute source robot masks and target robot visuals, it requires precise camera calibration which is unavailable for most open-source datasets. To relax this assumption, we approach cross-painting as an image-to-image translation problem. RoVi-Aug begins by predicting semantic mask on the robot $\mathcal{S}$, which are then extracted and transformed into robot $\mathcal{T}$ using a robot-to-robot (R2R) diffusion model. Meanwhile, the masked regions in the original images are inpainted using a video inpainting network to ensure visual continuity and integrity. Finally, the generated robot $\mathcal{T}$ is pasted back into the background image (see Fig. 2).

**Robot Segmentation.** In order to replace robot $\mathcal{S}$ with robot $\mathcal{T}$ in the image, we first need to detect the robot using semantic segmentation [132, 129, 133, 134]. We find that off-the-shelf segmentation models [129, 135] often fail to accurately segment out the robot, potentially due to the fact that robot images are under-represented in their training data. As such, we finetune a pretrained Segment Anything Model (SAM) [129] using Low-Rank Adaptation (LoRA) [136]. We use simulation to synthetically generate a large dataset of different robot images with corresponding masks, where we randomly sample a wide range of camera and robot poses. We apply brightness augmentation and resizing to simulate different lighting and fields of view. To create diverse backgrounds, we paste the generated robot parts into various background images [137]. By training the LoRA layer on this synthetic dataset, we obtain a mask model capable of handling different robot and camera poses.

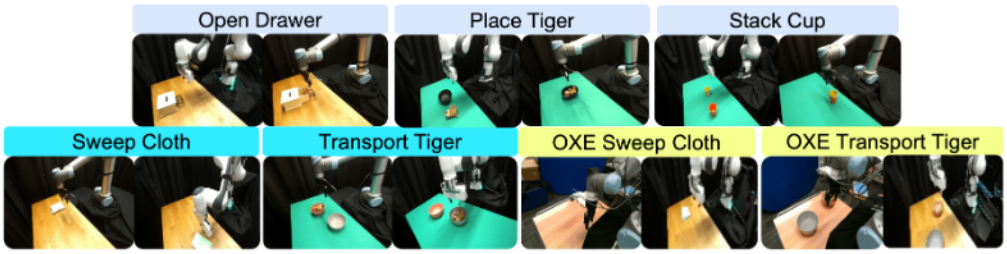


Figure 3: **Tasks used for evaluation.** For each task, on the left is an example training view and robot, and on the right is the different test-time embodiment.


Figure 4: **Evaluated camera views.** For static third-person cameras, we perturb the initial training view by 10 cm translation, 20° rotation and 25 cm translation, 35° rotation. Even when the camera is moving dynamically, RoVi-Aug is able to successfully sweep the cloth.

**Robot-to-Robot (R2R) Generation** Next, we aim to transform the segmented robot $\mathcal{S}$ into robot $\mathcal{T}$. We use an image-to-image diffusion model. Similar to semantic segmentation, training a diffusion model capable of handling various camera and robot poses requires a large dataset of paired images. As collecting paired real robot data is challenging due to the need for precise adjustments of camera and robot poses, we again use simulation to generate pairs of robots at the same randomly sampled robot poses and camera poses, with brightness and resizing augmentations. Inspired by [138, 130], we use a ControlNet [130] to finetune a pretrained Stable Diffusion [11]. Even though we train the model on simulation images, we find that it still performs well on real segmented robot images.

**Robot Inpainting** Inspired by Li et al. [139], after segmenting out robot $\mathcal{S}$ from the image, we inpaint the missing region using a video inpainting model E$^2$FGVI [140]. The final step involves pasting the generated robot $\mathcal{T}$ back to the image. As the R2R diffusion model is trained on simulated robot images, there is a visual gap from the real robot, particularly with the illumination. To prevent the trained policy on the augmented data from overfitting to the synthetic robot visuals, we perform random brightness augmentation to the generated robot before pasting it. We find in our experiments that this randomization significantly helps the performance of the trained policy (Section 5.3).

At the end of the Robot Augmentation pipeline, we obtain a sequence of cross-painted observations with synthesized target robot: $D_i^{\mathcal{S}\to\mathcal{T}} = \{o_1^{\mathcal{S}\to\mathcal{T}}, ..., o_{H_i}^{\mathcal{S}\to\mathcal{T}}\}$.

### 4.2 Viewpoint Augmentation (Vi-Aug)

To increase robustness of the trained policy to camera pose changes, we propose to augment the viewpoints of the images. This is orthogonal to robot augmentation and can be applied to both $D_i^{\mathcal{S}}$ and $D_i^{\mathcal{S}\to\mathcal{T}}$.

We use ZeroNVS [131], a state-of-the-art 3D-aware diffusion model that can zero-shot synthesize 360° view of a scene from a single image. Compared to prior methods [103, 102, 141] that are limited to segmented object with no background, ZeroNVS works with multi-object scenes with complex backgrounds. For each image $o_t \in D_i$, we uniformly sample perturbations $(\tilde{R}_t, \tilde{T}_t) \in SE(3)$ from a box range, where each component in $\tilde{T}_t$ is bounded by an interval. We parametrize $\tilde{R}_t$ with Euler angles and each of those three angles is uniformly sampled within an interval described in Section 5.1. This process produces a resulting image as if the camera were perturbed by the sampled transformation: $o_t^{\tilde{R},\tilde{T}} = f\left(o_t; \tilde{R}, \tilde{T}\right)$, where we use $f$ to denote the camera transformation. We denote the resulting augmented data as $D_i^{\text{Vi-Aug}} = \{o_1^{R_1,T_1}, ..., o_{H_i}^{R_{H_i},T_{H_i}}\}$. We experiment with two strategies for sampling the perturbations: independently sampling random $(\tilde{R}_t, \tilde{T}_t)$ for each image, or applying a consistent random transformation $(\tilde{R}, \tilde{T})$ across the entire trajectory in $D_i$.

### 4.3 Policy Training

After applying robot and viewpoint augmentation, we can train a policy $\pi$ based on the Diffusion Policy architecture [142] on the augmented dataset $\mathcal{D}^{\mathcal{S}\to\mathcal{T}^{\text{Vi-Aug}}}$ and zero-shot deploy the policy on the target robot $\mathcal{T}$. For challenging tasks or when there is a large difference in the dynamics between the robots, we can also collect a small demonstration dataset $\mathcal{D}^{\mathcal{T}}$ on the target robot directly and few-shot finetune $\pi$ on $\mathcal{D}^{\mathcal{T}}$ to further improve policy performance. Alternatively, we can co-train $\pi$ on $\mathcal{D}^{\mathcal{S}^{\text{Vi-Aug}}} \bigcup \mathcal{D}^{\mathcal{S}\to\mathcal{T}^{\text{Vi-Aug}}}$ to obtain a multi-robot policy. Additionally, if we have multiple datasets with different tasks, can mix-and-match the datasets and train a multi-robot multi-task policy. For example, given data $\mathcal{D}_1^{\mathcal{S}}$ and $\mathcal{D}_2^{\mathcal{T}}$ with robot $\mathcal{S}$ performing task 1 and robot $\mathcal{T}$ performing task 2, we can train on the cross-product $\mathcal{D}_1^{\mathcal{S}} \bigcup \mathcal{D}_2^{\mathcal{T}\to\mathcal{S}} \bigcup \mathcal{D}_1^{\mathcal{S}\to\mathcal{T}} \bigcup \mathcal{D}_2^{\mathcal{T}}$ and their viewpoint-augmented versions to obtain a policy that can perform both tasks on both robots. In this way, we efficiently reuse the datasets and explicitly encourage transfer between robots and skills.

## 5 Experiments

### 5.1 Implementation Details

To train our robot segmentation and Robot-to-Robot generation models, we use the Robosuite simulator [143] to generate a large dataset of paired robot images with corresponding masks with randomly sampled robot poses and camera poses (see supplementary material for details). We use 4 robots: Franka, UR5, Sawyer, and Jaco, with 800k images each. We finetune a LoRA layer while keeping SAM frozen with a learning rate of 1e-4 for just one epoch to avoid the overfitting. We train a ControlNet for each robot pair based on Stable Diffusion v1.5 [11] with a learning rate of 1e-4 for 20k steps. During robot inpainting, we randomly sample perturbations of the value channel in the HSV space between -30 and 30.

For view augmentation sampling, $\tilde{T}_x, \tilde{T}_z \in (-0.25\,\text{m}, 0.25\,\text{m})$, $\tilde{T}_y \in (-0.1\,\text{m}, 0.1\,\text{m})$. The $y$ (vertical) direction has a lower translation range, as we have noticed that when moving excessively along the vertical direction, ZeroNVS outputs larger, more distracting artifacts. For rotation, we sample each Euler angle between $\pm 0.1$ radians.

### 5.2 Experiment Setup

We design experiments to answer the following research questions: (1) Can robot augmentation (Ro-Aug) effectively bridge the visual gap between the robots? (2) Can viewpoint augmentation (Vi-Aug) improve policy robustness to camera pose changes? (3) Can policies trained with RoVi-Aug be successfully deployed zero-shot on a different robot with camera changes? (4) Does RoVi-Aug enable multi-robot multi-task training and better facilitate transfer between robots and skills?

| Policies \ Tasks | Franka → UR5 | | | UR5 → Franka | |
|---|---|---|---|---|---|
| | Open Drawer | Place Tiger | Stack Cup | Sweep Cloth | Transport Tiger |
| No Augmentation | 0% | 0% | 0% | 0% | 40% |
| Mirage | 60% | **90%** | **50%** | **100%** | 70% |
| Ro-Aug | **90%** | 80% | 30% | **100%** | **80%** |
| Ro-Aug w/o Bright. Rand. | **90%** | 50% | 10% | 40% | 60% |

Table 1: **Zero-shot physical experiments evaluating robot augmentation.** We evaluate Ro-Aug on 5 tasks in 2 settings with 10 trials each: Learning a policy using Franka demonstration data and evaluating on a UR5, and vice versa. The camera poses are the same. We compare Ro-Aug with 2 baselines and an ablation that does not apply random brightness augmentation during the Ro-Aug pipeline. We see that Ro-Aug achieves comparable zero-shot performance as Mirage.

To answer the first three questions, we study policy transfer between a Franka and a UR5 robot on 5 tasks (Fig. 3): (1) Open a drawer, (2) Pick up a toy tiger from the table and put it into a bowl (Place Tiger), (3) Stack cups, (4) Sweep cloth from right to left, and (5) Transport a toy tiger between two bowls. See the Appendix for more details. For the first three tasks, we collect demonstrations on the Franka, and for the latter two, we collect demonstrations on the UR5. All demonstrations are collected via teleoperation at 15 Hz [5], with 150 trajectories each. A typical trajectory consists

| Policies \ Tasks | Franka → UR5 | | | UR5 → Franka | |
|---|---|---|---|---|---|
| | Open Drawer | Place Tiger | Stack Cup | Sweep Cloth | Transport Tiger |
| 5-Shot | 40% | 30% | 0% | 50% | 40% |
| Ro-Aug + 5-Shot | **100%** | **100%** | **60%** | **100%** | **100%** |
| 10-Shot | 70% | 40% | 50% | 80% | 80% |
| Ro-Aug + 10-Shot | **100%** | **100%** | **80%** | **100%** | **100%** |

Table 2: **Few-shot physical experiments evaluating robot augmentation.** We apply 5-shot and 10-shot finetuning to policies trained with Ro-Aug and compare them to few-shot policies trained without Ro-Aug. We see that Ro-Aug improves finetuning sample efficiency and exceeds the performance of all policies in Table 1.

| Policies \ Tasks | Place Tiger (Franka → Franka) | | | |
|---|---|---|---|---|
| | Same Angle | 10 cm, 20° | 25 cm, 35° | 35 cm, 45° |
| No Augmentation | **100%** | 0% | 0% | 0% |
| Vi-Aug 10 cm - Consistent | **100%** | 30% | 0% | 0% |
| Vi-Aug 10 cm - Inconsistent | **100%** | 70% | 10% | 0% |
| Vi-Aug 25 cm - Consistent | **100%** | **80%** | 30% | 30% |
| Vi-Aug 25 cm - Inconsistent | 90% | **80%** | 50% | 30% |
| Vi-Aug 40 cm - Consistent | 70% | 70% | **60%** | 20% |
| Vi-Aug 40 cm - Inconsistent | 80% | **80%** | 50% | **40%** |

Table 3: **Physical experiments evaluating viewpoint augmentation.** We compare policies trained with different degrees of camera perturbations (rows). The numbers represent the range of the camera perturbation that $\tilde{T}_x$ and $\tilde{T}_z$ are sampled from. "Consistent/Inconsistent" represents whether the same/different perturbation is applied to each timestep in a trajectory. We evaluate on the same robot but with different test camera angles (columns).

of 75-120 timesteps (5-8 s). We use a ZED 2 camera positioned from the side for each robot. We augment the demonstration data with robot augmentation (Ro-Aug) using the other robot, viewpoint augmentation (Vi-Aug), as well as both (RoVi-Aug), train a diffusion policy, and evaluate on the other robot. All experiments are evaluated with 10 trials each.

To answer the last question, we combine demonstration data from Franka and UR5 for different tasks, perform robot augmentations, and train a multi-robot multi-task diffusion policy. We also select the Berkeley UR5 dataset [144] from the OXE data [9], apply RoVi-Aug to generate synthetic Franka images and finetune a generalist policy, Octo [145], on the augmented datasets. We additionally collect 50 demonstrations on the target robot (Franka) and further finetune Octo-Base in a language goal-conditioned format. We compare whether training Octo on the augmented data improves the finetuning sample efficiency on the downstream tasks.

### 5.3 Results

Table 1 shows the effect of robot augmentation when the camera poses are the same. The policy is deployed zero-shot. We compare Ro-Aug with 2 baselines, no augmentation and Mirage, and an ablation that does not apply random brightness augmentation during the Ro-Aug pipeline. Without robot augmentation, the policy trained on the source robot only barely achieves success on the target robot. On the other hand, Ro-Aug achieves comparable zero-shot performance as Mirage. Additionally, we see that brightness randomization helps performance, suggesting that it effectively prevents the policy from overfitting to the lighting in simulation that the R2R model is trained on.

Table 2 shows the policies trained on Ro-Aug data can be finetuned with 5-10 demonstrations on the target robot to further improve performance. Compared to few-shot policies trained without Ro-Aug, we see that Ro-Aug improves finetuning sample efficiency and exceeds the performance of all policies in Table 1. In contrast, Mirage does not allow finetuning and cannot improve performance on challenging tasks such as cup stacking.

Table 3 evaluates the effect of viewpoint augmentation. We choose the Tiger Place task on the Franka robot and study how different strategies of camera perturbation sampling affect policy robustness. We sample translations $\tilde{T}_x$ and $\tilde{T}_z$ between $\pm 0.1$ m, $\pm 0.25$ m, and $\pm 0.4$ m, and compare consistent perturbation across trajectories or independently on each image. From Table 3, we see that larger variation during augmentation improves policy robustness under severe camera pose changes.

However, the performance decreases under the original camera angle, potentially due to lower density of each camera pose as the sampling range increases. Additionally, inconsistent augmentation seems to slightly outperform consistent augmentation., suggesting potential benefit from more augmentation. We note that the diffusion policy takes in only 2 steps of history, so viewpoint inconsistency may not matter much. Future work can study whether inconsistent augmentation would harm policies that use a longer history. Based on the results, we choose to apply inconsistent augmentation with 25 cm perturbation range for other RoVi-Aug experiments.

| Tasks / Policies | Franka → UR5 | | | | UR5 → Franka | | | |
|---|---|---|---|---|---|---|---|---|
| | Open Drawer | | Place Tiger | | Sweep Cloth | | Transport Tiger | |
| | 10 cm, 20° | 25 cm, 35° | 10 cm, 20° | 25 cm, 35° | 10 cm, 20° | 25 cm, 35° | 10 cm, 20° | 25 cm, 35° |
| Mirage | 50% | 30% | 30% | 20% | **80%** | 30% | 20% | 0% |
| Ro-Aug | 60% | 20% | 30% | 10% | 0% | 0% | 0% | 0% |
| RoVi-Aug | **80%** | **50%** | **70%** | **30%** | **80%** | **40%** | **40%** | **30%** |

Table 4: **Physical experiments evaluating RoVi-Aug on different robots with different camera angles.** The translation and rotation shows the difference in the camera poses between the robots. Mirage uses a policy trained on only the source robot with a test-time cross-painting procedure and depth reprojection to account for camera pose changes. Ro-Aug only applies robot augmentation while RoVi-Aug applies both robot and viewpoint augmentation. For both Ro-Aug and RoVi-Aug, the policy is trained on the augmented data and deployed on the target robot zero-shot.

| | Franka | UR5 |
|---|---|---|
| Place Tiger | 80% | 70% |
| Transport Tiger | 60% | 80% |

Table 5: **Robot-Skill Cross Product.** We train a multi-robot multi-task diffusion policy trained on pooling the Franka Tiger Place data and UR5 Tiger Transport data as well as their RoVi-Aug versions together.

| Tasks / Policies | OXE UR5 → Franka | |
|---|---|---|
| | Sweep Cloth | Transport Tiger |
| Octo-Base | 30% | 20% |
| Octo-Base + RoVi-Aug | **60%** | **40%** |

Table 6: **Octo finetuning from the OXE datasets with 50 in-domain demonstrations for each task.** RoVi-Aug improves finetuning sample efficiency.

Table 4 evaluates RoVi-Aug on different robots with different viewpoints. We can see that viewpoint augmentation is crucial and Mirage struggles with larger camera pose changes. In contrast, RoVi-Aug can still achieve success when the target robot viewpoint is significantly different from source robot.

To evaluate robot-skill cross-product, we combine the Tiger Place demonstration data from the Franka and Tiger Transport demonstration data from the UR5, as well as their robot-augmented UR5 and Franka versions, and train a multi-robot multi-task diffusion policy. From Table 5, we can see that the policy can successfully execute the two tasks on both robots. Additionally, we evaluate whether RoVi-Aug improves finetuning sample efficiency. From Table 6, we can see that after training Octo on the augmented OXE data, the policy has seen the synthetic target robots performing the tasks, accelerating downstream finetuning of similar tasks.

## 6 Limitations and Future Work

We present RoVi-Aug, a pipeline for robot and viewpoint augmentation that bridges different robot datasets and better facilitates transfers between robots and skills. There are several limitations, which open up possibilities for future work: (1) Our robot augmentation pipeline relies on a sequence of different models so artifacts can cascade. For example, inaccuracies in the robot segmentation stage (e.g., mistakenly segmenting the object out) could lead to bad robot-to-robot generations in the second stage. See the Appendix for more details on artifacts. Additionally, instead of training an R2R diffusion model for each robot pair, future work could explore a unified model that handles multiple pairs. (2) For viewpoint augmentation, future work could improve the quality of novel view synthesis by finetuning the model on robotics data or using video-based models [146]. (3) While we mitigate viewpoint changes in this work, there are also often background changes in practice during cross-embodiment transfer. Future work could combine RoVi-Aug with prior orthogonal approaches such as object, background, and task augmentation [115, 116] to further obtain more generalizable policies. (4) We only demonstrate transfer between stationary robot arms and do not consider very different grippers such as multi-fingered hands. We leave these extensions to future work.

**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 reviewers for valuable feedback.

## References

[1] C. Devin, A. Gupta, T. Darrell, P. Abbeel, and S. 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.

[2] E. Jang, A. Irpan, M. Khansari, D. Kappler, F. Ebert, C. Lynch, S. Levine, and C. Finn. BC-Z: Zero-shot task generalization with robotic imitation learning. In *Conference on Robot Learning (CoRL)*, pages 991–1002, 2021.

[3] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, J. Dabis, C. Finn, K. Gopalakrishnan, K. Hausman, A. Herzog, J. Hsu, et al. RT-1: Robotics transformer for real-world control at scale. *Robotics: Science and Systems (RSS)*, 2023.

[4] B. Zitkovich, T. Yu, S. Xu, P. Xu, T. Xiao, F. Xia, J. Wu, P. Wohlhart, S. Welker, A. Wahid, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In *Conference on Robot Learning*, pages 2165–2183. PMLR, 2023.

[5] A. Khazatsky, K. Pertsch, S. Nair, A. Balakrishna, S. Dasari, S. Karamcheti, S. Nasiriany, M. K. Srirama, L. Y. Chen, K. Ellis, P. D. Fagan, J. Hejna, M. Itkina, M. Lepert, Y. J. Ma, P. T. Miller, J. Wu, S. Belkhale, S. Dass, H. Ha, A. Jain, A. Lee, Y. Lee, M. Memmel, S. Park, I. Radosavovic, K. Wang, A. Zhan, K. Black, C. Chi, K. B. Hatch, S. Lin, J. Lu, J. Mercat, A. Rehman, P. R. Sanketi, A. Sharma, C. Simpson, Q. Vuong, H. R. Walke, B. Wulfe, T. Xiao, J. H. Yang, A. Yavary, T. Z. Zhao, C. Agia, R. Baijal, M. G. Castro, D. Chen, Q. Chen, T. Chung, J. Drake, E. P. Foster, J. Gao, D. A. Herrera, M. Heo, K. Hsu, J. Hu, D. Jackson, C. Le, Y. Li, K. Lin, R. Lin, Z. Ma, A. Maddukuri, S. Mirchandani, D. Morton, T. Nguyen, A. O'Neill, R. Scalise, D. Seale, V. Son, S. Tian, E. Tran, A. E. Wang, Y. Wu, A. Xie, J. Yang, P. Yin, Y. Zhang, O. Bastani, G. Berseth, J. Bohg, K. Goldberg, A. Gupta, A. Gupta, D. Jayaraman, J. J. Lim, J. Malik, R. Martín-Martín, S. Ramamoorthy, D. Sadigh, S. Song, J. Wu, M. C. Yip, Y. Zhu, T. Kollar, S. Levine, and C. Finn. Droid: A large-scale in-the-wild robot manipulation dataset. In *Proceedings of Robotics: Science and Systems*, Delft, Netherlands, 2024.

[6] Y. Jiang, A. Gupta, Z. Zhang, G. Wang, Y. Dou, Y. Chen, L. Fei-Fei, A. Anandkumar, Y. Zhu, and L. Fan. VIMA: General robot manipulation with multimodal prompts. *International Conference on Machine Learning (ICML)*, 2023.

[7] D. Shah, A. Sridhar, A. Bhorkar, N. Hirose, and S. 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.

[8] D. Shah, A. Sridhar, N. Dashora, K. Stachowicz, K. Black, N. Hirose, and S. Levine. ViNT: A Foundation Model for Visual Navigation. In *7th Annual Conference on Robot Learning (CoRL)*, 2023.

[9] O. X.-E. Collaboration, A. Padalkar, A. Pooley, A. Jain, A. Bewley, A. Herzog, A. Irpan, A. Khazatsky, A. Rai, A. Singh, A. Brohan, A. Raffin, A. Wahid, B. Burgess-Limerick, B. Kim, B. Schölkopf, B. Ichter, C. Lu, C. Xu, C. Finn, C. Xu, C. Chi, C. Huang, C. Chan, C. Pan, C. Fu, C. Devin, D. Driess, D. Pathak, D. Shah, D. Büchler, D. Kalashnikov, D. Sadigh, E. Johns,

F. Ceola, F. Xia, F. Stulp, G. Zhou, G. S. Sukhatme, G. Salhotra, G. Yan, G. Schiavi, H. Su, H.-S. Fang, H. Shi, H. B. Amor, H. I. Christensen, H. Furuta, H. Walke, H. Fang, I. Mordatch, I. Radosavovic, I. Leal, J. Liang, J. Kim, J. Schneider, J. Hsu, J. Bohg, J. Bingham, J. Wu, J. Wu, J. Luo, J. Gu, J. Tan, J. Oh, J. Malik, J. Tompson, J. Yang, J. J. Lim, J. Silvério, J. Han, K. Rao, K. Pertsch, K. Hausman, K. Go, K. Gopalakrishnan, K. Goldberg, K. Byrne, K. Oslund, K. Kawaharazuka, K. Zhang, K. Majd, K. Rana, K. Srinivasan, L. Y. Chen, L. Pinto, L. Tan, L. Ott, L. Lee, M. Tomizuka, M. Du, M. Ahn, M. Zhang, M. Ding, M. K. Srirama, M. Sharma, M. J. Kim, N. Kanazawa, N. Hansen, N. Heess, N. J. Joshi, N. Suenderhauf, N. D. Palo, N. M. M. Shafiullah, O. Mees, O. Kroemer, P. R. Sanketi, P. Wohlhart, P. Xu, P. Sermanet, P. Sundaresan, Q. Vuong, R. Rafailov, R. Tian, R. Doshi, R. Martín-Martín, R. Mendonca, R. Shah, R. Hoque, R. Julian, S. Bustamante, S. Kirmani, S. Levine, S. Moore, S. Bahl, S. Dass, S. Song, S. Xu, S. Haldar, S. Adebola, S. Guist, S. Nasiriany, S. Schaal, S. Welker, S. Tian, S. Dasari, S. Belkhale, T. Osa, T. Harada, T. Matsushima, T. Xiao, T. Yu, T. Ding, T. Davchev, T. Z. Zhao, T. Armstrong, T. Darrell, V. Jain, V. Vanhoucke, W. Zhan, W. Zhou, W. Burgard, X. Chen, X. Wang, X. Zhu, X. Li, Y. Lu, Y. Chebotar, Y. Zhou, Y. Zhu, Y. Xu, Y. Wang, Y. Bisk, Y. Cho, Y. Lee, Y. Cui, Y. hua Wu, Y. Tang, Y. Zhu, Y. Li, Y. Iwasawa, Y. Matsuo, Z. Xu, and Z. J. Cui. Open X-Embodiment: Robotic learning datasets and RT-X models. IEEE International Conference on Robotics and Automation, 2024.

[10] R. Bommasani, D. A. Hudson, E. Adeli, R. Altman, S. Arora, S. von Arx, M. S. Bernstein, J. Bohg, A. Bosselut, E. Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

[11] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. 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.

[12] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[13] A. Blattmann, T. Dockhorn, S. Kulal, D. Mendelevitch, M. Kilian, D. Lorenz, Y. Levi, Z. English, V. Voleti, A. Letts, et al. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*, 2023.

[14] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009, 2022.

[15] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[16] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[17] H. Liu, C. Li, Q. Wu, and Y. J. Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36, 2024.

[18] T. Brooks, B. Peebles, C. Holmes, W. DePue, Y. Guo, L. Jing, D. Schnurr, J. Taylor, T. Luhman, E. Luhman, C. Ng, R. Wang, and A. Ramesh. Video generation models as world simulators. 2024. URL https://openai.com/research/video-generation-models-as-world-simulators.

[19] C. Saharia, W. Chan, S. Saxena, L. Li, J. Whang, E. L. Denton, K. Ghasemipour, R. Gontijo Lopes, B. Karagol Ayan, T. Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in neural information processing systems*, 35: 36479–36494, 2022.

[20] A. X. Lee, C. M. Devin, Y. Zhou, T. Lampe, K. Bousmalis, J. T. Springenberg, A. Byravan, A. Abdolmaleki, N. Gileadi, D. Khosid, et al. Beyond pick-and-place: Tackling robotic stacking of diverse shapes. In *5th Annual Conference on Robot Learning*, 2021.

[21] A. Herzog, K. Rao, K. Hausman, Y. Lu, P. Wohlhart, M. Yan, J. Lin, M. G. Arenas, T. Xiao, D. Kappler, et al. Deep rl at scale: Sorting waste in office buildings with a fleet of mobile manipulators. *arXiv preprint arXiv:2305.03270*, 2023.

[22] D. Kalashnikov, J. Varley, Y. Chebotar, B. Swanson, R. Jonschkowski, C. Finn, S. Levine, and K. Hausman. Scaling up multi-task robotic reinforcement learning. In *Conference on Robot Learning*, pages 557–575. PMLR, 2022.

[23] H.-S. Fang, H. Fang, Z. Tang, J. Liu, J. Wang, H. Zhu, and C. Lu. RH20T: A robotic dataset for learning diverse skills in one-shot. In *RSS 2023 Workshop on Learning for Task and Motion Planning*, 2023.

[24] N. M. M. Shafiullah, A. Rai, H. Etukuru, Y. Liu, I. Misra, S. Chintala, and L. Pinto. On bringing robots home, 2023.

[25] J. Gao, A. Xie, T. Xiao, C. Finn, and D. Sadigh. Efficient data collection for robotic manipulation via compositional generalization. In *Proceedings of Robotics: Science and Systems (RSS)*, 2024.

[26] L. Y. Chen, K. Hari, K. Dharmarajan, C. Xu, Q. Vuong, and K. Goldberg. Mirage: Cross-embodiment zero-shot policy transfer with cross-painting. In *Proceedings of Robotics: Science and Systems*, Delft, Netherlands, 2024.

[27] P. Christiano, Z. Shah, I. Mordatch, J. Schneider, T. Blackwell, J. Tobin, P. Abbeel, and W. Zaremba. Transfer from simulation to real world through learning deep inverse dynamics model. *arXiv preprint arXiv:1610.03518*, 2016.

[28] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. 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.

[29] Y. Chebotar, A. Handa, V. Makoviychuk, M. Macklin, J. Issac, N. Ratliff, and D. Fox. Closing the sim-to-real loop: Adapting simulation randomization with real world experience. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 8973–8979. IEEE, 2019.

[30] M. Kaspar, J. D. M. Osorio, and J. Bock. Sim2real transfer for reinforcement learning without dynamics randomization. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4383–4388. IEEE, 2020.

[31] S. Nasiriany, A. Maddukuri, L. Zhang, A. Parikh, A. Lo, A. Joshi, A. Mandlekar, and Y. Zhu. Robocasa: Large-scale simulation of everyday tasks for generalist robots. In *Robotics: Science and Systems (RSS)*, 2024.

[32] S. Uppal, A. Agarwal, H. Xiong, K. Shaw, and D. Pathak. Spin: Simultaneous perception, interaction and navigation. *CVPR*, 2024.

[33] T. Chen, A. Murali, and A. Gupta. Hardware conditioned policies for multi-robot transfer learning. *Advances in Neural Information Processing Systems*, 31, 2018.

[34] E. S. Hu, K. Huang, O. Rybkin, and D. Jayaraman. Know thyself: Transferable visual control policies through robot-awareness. In *ICLR 2022 Workshop on Generalizable Policy Learning in Physical World*.

[35] K. Schmeckpeper, O. Rybkin, K. Daniilidis, S. Levine, and C. Finn. Reinforcement learning with videos: Combining offline observations with interaction. In *Conference on Robot Learning*, pages 339–354. PMLR, 2021.

[36] T. Yu, C. Finn, S. Dasari, A. Xie, T. Zhang, P. Abbeel, and S. Levine. One-shot imitation from observing humans via domain-adaptive meta-learning. *Robotics: Science and Systems XIV*, 2018.

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

[38] L. Smith, N. Dhawan, M. Zhang, P. Abbeel, and S. Levine. Avid: Learning multi-stage tasks via pixel-level translation of human videos. *Robotics: Science and Systems*, 2020.

[39] Y. Liu, A. Gupta, P. Abbeel, and S. Levine. Imitation from observation: Learning to imitate behaviors from raw video via context translation. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1118–1125. IEEE, 2018.

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

[41] M. Xu, Z. Xu, C. Chi, M. Veloso, and S. Song. Xskill: Cross embodiment skill discovery. In *Conference on Robot Learning*, pages 3536–3555. PMLR, 2023.

[42] S. Bahl, A. Gupta, and D. Pathak. Human-to-robot imitation in the wild. *Robotics: Science and Systems (RSS)*, 2022.

[43] C. Wen, X. Lin, J. So, K. Chen, Q. Dou, Y. Gao, and P. Abbeel. Any-point trajectory modeling for policy learning. *arXiv preprint arXiv:2401.00025*, 2023.

[44] X. B. Peng, E. Coumans, T. Zhang, T.-W. Lee, J. Tan, and S. Levine. Learning agile robotic locomotion skills by imitating animals. *Robotics: Science and systems*, 2020.

[45] A. Sivakumar, K. Shaw, and D. Pathak. Robotic telekinesis: Learning a robotic hand imitator by watching humans on youtube. *Robotics: Science and Systems*, 2022.

[46] K. Zakka, A. Zeng, P. Florence, J. Tompson, J. Bohg, and D. Dwibedi. Xirl: Cross-embodiment inverse reinforcement learning. In *Conference on Robot Learning*, pages 537–546. PMLR, 2022.

[47] A. S. Chen, S. Nair, and C. Finn. Learning generalizable robotic reward functions from "in-the-wild" human videos. *Robotics: Science and Systems*, 2021.

[48] Y. Zhou, Y. Aytar, and K. Bousmalis. Manipulator-independent representations for visual imitation. *Robotics: Science and Systems*, 2021.

[49] G. Salhotra, I. Liu, C. Arthur, and G. Sukhatme. Bridging action space mismatch in learning from demonstrations. *arXiv preprint arXiv:2304.03833*, 2023.

[50] A. A. Rusu, M. Večerík, T. Rothörl, N. Heess, R. Pascanu, and R. Hadsell. Sim-to-real robot learning from pixels with progressive nets. In *Conference on robot learning*, pages 262–270. PMLR, 2017.

[51] Y. Sun, R. Zheng, X. Wang, A. E. Cohen, and F. Huang. Transfer rl across observation feature spaces via model-based regularization. In *International Conference on Learning Representations*, 2022.

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

[53] X. Liu, D. Pathak, and K. Kitani. Revolver: Continuous evolutionary models for robot-to-robot policy transfer. In *International Conference on Machine Learning*, pages 13995–14007. PMLR, 2022.

[54] X. Liu, D. Pathak, and D. Zhao. Meta-evolve: Continuous robot evolution for one-to-many policy transfer. In *International Conference on Learning Representations*, 2024.

[55] H. B. Ammar, E. Eaton, P. Ruvolo, and M. 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.

[56] K. Kim, Y. Gu, J. Song, S. Zhao, and S. Ermon. Domain adaptive imitation learning. In *International Conference on Machine Learning*, pages 5286–5295. PMLR, 2020.

[57] Q. Zhang, T. Xiao, A. A. Efros, L. Pinto, and X. Wang. Learning cross-domain correspondence for control with dynamics cycle-consistency. In *International Conference on Learning Representations*.

[58] D. S. Raychaudhuri, S. Paul, J. Vanbaar, and A. K. Roy-Chowdhury. Cross-domain imitation from observations. In *International Conference on Machine Learning*, pages 8902–8912. PMLR, 2021.

[59] B. Lakshmanan and R. 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.

[60] A. Gupta, C. Devin, Y. Liu, P. Abbeel, and S. Levine. Learning invariant feature spaces to transfer skills with reinforcement learning. In *International Conference on Learning Representations*, 2022.

[61] T. Shankar, Y. Lin, A. Rajeswaran, V. Kumar, S. Anderson, and J. Oh. Translating robot skills: Learning unsupervised skill correspondences across robots. In *International Conference on Machine Learning*, pages 19626–19644. PMLR, 2022.

[62] D. Hejna, L. Pinto, and P. Abbeel. Hierarchically decoupled imitation for morphological transfer. In *International Conference on Machine Learning*, pages 4159–4171. PMLR, 2020.

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

[64] Z.-H. Yin, L. Sun, H. Ma, M. Tomizuka, and W.-J. Li. Cross domain robot imitation with invariant representation. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 455–461. IEEE, 2022.

[65] A. Ghadirzadeh, X. Chen, P. Poklukar, C. Finn, M. Björkman, and D. 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.

[66] C. Yu, W. Zhang, H. Lai, Z. Tian, L. Kneip, and J. 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.

[67] L. Shao, F. Ferreira, M. Jorda, V. Nambiar, J. Luo, E. Solowjow, J. A. Ojea, O. Khatib, and J. Bohg. Unigrasp: Learning a unified model to grasp with multifingered robotic hands. *IEEE Robotics and Automation Letters*, 5(2):2286–2293, 2020.

[68] Z. Xu, B. Qi, S. Agrawal, and S. Song. Adagrasp: Learning an adaptive gripper-aware grasping policy. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4620–4626. IEEE, 2021.

[69] T. Wang, R. Liao, J. Ba, and S. Fidler. Nervenet: Learning structured policy with graph neural networks. In *International conference on learning representations*, 2018.

[70] A. Sanchez-Gonzalez, N. Heess, J. T. Springenberg, J. Merel, M. Riedmiller, R. Hadsell, and P. Battaglia. Graph networks as learnable physics engines for inference and control. In J. Dy and A. 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.

[71] D. Pathak, C. Lu, T. Darrell, P. Isola, and A. A. Efros. Learning to control self-assembling morphologies: a study of generalization via modularity. *Advances in Neural Information Processing Systems*, 32, 2019.

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

[73] W. Huang, I. Mordatch, and D. 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.

[74] V. Kurin, M. Igl, T. Rocktaschel, W. Boehmer, and S. Whiteson. My body is a cage: the role of morphology in graph- based incompatible control. In *Proceedings of the International Conference on Learning Representations*. OpenReview, 2021.

[75] H. Furuta, Y. Iwasawa, Y. Matsuo, and S. S. Gu. A system for morphology-task generalization via unified representation and behavior distillation. In *The Eleventh International Conference on Learning Representations*, 2023.

[76] Y. Zhou, S. Sonawani, M. Phielipp, S. Stepputtis, and H. Amor. Modularity through attention: Efficient training and transfer of language-conditioned policies for robot manipulation. In K. Liu, D. Kulic, and J. 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.

[77] P. Jian, E. Lee, Z. Bell, M. M. Zavlanos, and B. Chen. Policy stitching: Learning transferable robot policies. In *Conference on Robot Learning*, pages 3789–3808. PMLR, 2023.

[78] Z. Xiong, J. Beck, and S. Whiteson. Universal morphology control via contextual modulation. In *International Conference on Machine Learning*, pages 38286–38300. PMLR, 2023.

[79] A. Depierre, E. Dellandréa, and L. 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.

[80] D. Kalashnikov, A. Irpan, P. Pastor, J. Ibarz, A. Herzog, E. Jang, D. Quillen, E. Holly, M. Kalakrishnan, V. Vanhoucke, et al. Scalable deep reinforcement learning for vision-based robotic manipulation. In *Conference on robot learning*, pages 651–673. PMLR, 2018.

[81] S. Levine, P. Pastor, A. Krizhevsky, J. Ibarz, and D. 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.

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

[83] F. Ebert, Y. Yang, K. Schmeckpeper, B. Bucher, G. Georgakis, K. Daniilidis, C. Finn, and S. Levine. Bridge data: Boosting generalization of robotic skills with cross-domain datasets. In *Robotics: Science and Systems (RSS) XVIII*, 2022.

[84] H. R. Walke, K. Black, T. Z. Zhao, Q. Vuong, C. Zheng, P. Hansen-Estruch, A. W. He, V. Myers, M. J. Kim, M. Du, et al. Bridgedata v2: A dataset for robot learning at scale. In *Conference on Robot Learning*, pages 1723–1736. PMLR, 2023.

[85] S. Nair, A. Rajeswaran, V. Kumar, C. Finn, and A. Gupta. R3m: A universal visual representation for robot manipulation. In *CoRL*, 2022.

[86] T. Xiao, I. Radosavovic, T. Darrell, and J. Malik. Masked visual pre-training for motor control. *arXiv preprint arXiv:2203.06173*, 2022.

[87] Y. J. Ma, S. Sodhani, D. Jayaraman, O. Bastani, V. Kumar, and A. Zhang. Vip: Towards universal visual reward and representation via value-implicit pre-training. In *The Eleventh International Conference on Learning Representations*, 2023.

[88] S. Dasari, F. Ebert, S. Tian, S. Nair, B. Bucher, K. Schmeckpeper, S. Singh, S. Levine, and C. Finn. Robonet: Large-scale multi-robot learning. In *Conference on Robot Learning*, pages 885–897. PMLR, 2020.

[89] Y. Du, S. Yang, B. Dai, H. Dai, O. Nachum, J. Tenenbaum, D. Schuurmans, and P. Abbeel. Learning universal policies via text-guided video generation. *Advances in Neural Information Processing Systems*, 36, 2024.

[90] S. Yang, Y. Du, S. K. S. Ghasemipour, J. Tompson, L. P. Kaelbling, D. Schuurmans, and P. Abbeel. Learning interactive real-world simulators. In *The Twelfth International Conference on Learning Representations*, 2024.

[91] C. Lynch, A. Wahid, J. Tompson, T. Ding, J. Betker, R. Baruch, T. Armstrong, and P. Florence. Interactive language: Talking to robots in real time. *IEEE Robotics and Automation Letters*, 2023.

[92] M. Shridhar, L. Manuelli, and D. Fox. Cliport: What and where pathways for robotic manipulation. In *Conference on Robot Learning*, pages 894–906. PMLR, 2022.

[93] A. Stone, T. Xiao, Y. Lu, K. Gopalakrishnan, K.-H. Lee, Q. Vuong, P. Wohlhart, S. Kirmani, B. Zitkovich, F. Xia, et al. Open-world object manipulation using pre-trained vision-language models. In *Conference on Robot Learning*, pages 3397–3417. PMLR, 2023.

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

[95] S. Reed, K. Zolna, E. Parisotto, S. G. Colmenarejo, A. Novikov, G. Barth-maron, M. Giménez, Y. Sulsky, J. Kay, J. T. Springenberg, T. Eccles, J. Bruce, A. Razavi, A. Edwards, N. Heess, Y. Chen, R. Hadsell, O. Vinyals, M. Bordbar, and N. de Freitas. A generalist agent. *Transactions on Machine Learning Research*, 2022. ISSN 2835-8856.

[96] I. Radosavovic, T. Xiao, S. James, P. Abbeel, J. Malik, and T. Darrell. Real-world robot learning with masked visual pre-training. In *Conference on Robot Learning*, 2022.

[97] H. Bharadhwaj, J. Vakil, M. Sharma, A. Gupta, S. Tulsiani, and V. Kumar. Roboagent: Generalization and efficiency in robot manipulation via semantic augmentations and action chunking. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4788–4795. IEEE, 2024.

[98] X. Chen, J. Djolonga, P. Padlewski, B. Mustafa, S. Changpinyo, J. Wu, C. R. Ruiz, S. Goodman, X. Wang, Y. Tay, S. Shakeri, M. Dehghani, D. Salz, M. Lucic, M. Tschannen, A. Nagrani, H. Hu, M. Joshi, B. Pang, C. Montgomery, P. Pietrzyk, M. Ritter, A. Piergiovanni, M. Minderer, F. Pavetic, A. Waters, G. Li, I. Alabdulmohsin, L. Beyer, J. Amelot, K. Lee, A. P. Steiner, Y. Li, D. Keysers, A. Arnab, Y. Xu, K. Rong, A. Kolesnikov, M. Seyedhosseini, A. Angelova, X. Zhai, N. Houlsby, and R. Soricut. Pali-x: On scaling up a multilingual vision and language model, 2023.

[99] D. Driess, F. Xia, M. S. Sajjadi, C. Lynch, A. Chowdhery, B. Ichter, A. Wahid, J. Tompson, Q. Vuong, T. Yu, et al. Palm-e: An embodied multimodal language model. In *International Conference on Machine Learning*, pages 8469–8488. PMLR, 2023.

[100] R. Girdhar, A. El-Nouby, Z. Liu, M. Singh, K. V. Alwala, A. Joulin, and I. Misra. Imagebind: One embedding space to bind them all. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15180–15190, 2023.

[101] B. Poole, A. Jain, J. T. Barron, and B. Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. In *The Eleventh International Conference on Learning Representations*, 2023.

[102] L. Melas-Kyriazi, I. Laina, C. Rupprecht, and A. Vedaldi. Realfusion: 360deg reconstruction of any object from a single image. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8446–8455, 2023.

[103] R. Liu, R. Wu, B. Van Hoorick, P. Tokmakov, S. Zakharov, and C. Vondrick. Zero-1-to-3: Zero-shot one image to 3d object. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9298–9309, 2023.

[104] A. Brohan, Y. Chebotar, C. Finn, K. Hausman, A. Herzog, D. Ho, J. Ibarz, A. Irpan, E. Jang, R. Julian, et al. Do as i can, not as i say: Grounding language in robotic affordances. In *Conference on robot learning*, pages 287–318. PMLR, 2023.

[105] W. Huang, F. Xia, T. Xiao, H. Chan, J. Liang, P. Florence, A. Zeng, J. Tompson, I. Mordatch, Y. Chebotar, et al. Inner monologue: Embodied reasoning through planning with language models. In *Conference on Robot Learning*, pages 1769–1782. PMLR, 2023.

[106] J. Liang, W. Huang, F. Xia, P. Xu, K. Hausman, B. Ichter, P. Florence, and A. Zeng. Code as policies: Language model programs for embodied control. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9493–9500. IEEE, 2023.

[107] Y. Tang, W. Yu, J. Tan, H. Zen, A. Faust, and T. Harada. Saytap: Language to quadrupedal locomotion. In *Conference on Robot Learning*, pages 3556–3570. PMLR, 2023.

[108] H. Wang, K. Kedia, J. Ren, R. Abdullah, A. Bhardwaj, A. Chao, K. Y. Chen, N. Chin, P. Dan, X. Fan, G. Gonzalez-Pumariega, A. Kompella, M. A. Pace, Y. Sharma, X. Sun, N. Sunkara, and S. Choudhury. MOSAIC: A modular system for assistive and interactive cooking. In *2nd Workshop on Mobile Manipulation and Embodied Intelligence at ICRA 2024*, 2024.

[109] Y.-J. Wang, B. Zhang, J. Chen, and K. Sreenath. Prompt a robot to walk with large language models. *arXiv preprint arXiv:2309.09969*, 2023.

[110] W. Yu, N. Gileadi, C. Fu, S. Kirmani, K.-H. Lee, M. G. Arenas, H.-T. L. Chiang, T. Erez, L. Hasenclever, J. Humplik, et al. Language to rewards for robotic skill synthesis. In *Conference on Robot Learning*, pages 374–404. PMLR, 2023.

[111] Y. J. Ma, W. Liang, G. Wang, D.-A. Huang, O. Bastani, D. Jayaraman, Y. Zhu, L. Fan, and A. Anandkumar. Eureka: Human-level reward design via coding large language models. In *The Twelfth International Conference on Learning Representations*, 2024.

[112] T. Xiao, H. Chan, P. Sermanet, A. Wahid, A. Brohan, K. Hausman, S. Levine, and J. Tompson. Skill acquisition by instruction augmentation on offline datasets. In *Workshop on Language and Robotics at CoRL 2022*, 2022.

[113] P. Katara, Z. Xian, and K. Fragkiadaki. Gen2sim: Scaling up robot learning in simulation with generative models. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6672–6679. IEEE, 2024.

[114] Y. Wang, Z. Xian, F. Chen, T.-H. Wang, Y. Wang, K. Fragkiadaki, Z. Erickson, D. Held, and C. Gan. Robogen: Towards unleashing infinite data for automated robot learning via generative simulation. In *Forty-first International Conference on Machine Learning*, 2024.

[115] T. Yu, T. Xiao, A. Stone, J. Tompson, A. Brohan, S. Wang, J. Singh, C. Tan, J. Peralta, B. Ichter, et al. Scaling robot learning with semantically imagined experience. *Robotics: Science and Systems*, 2023.

[116] Z. Chen, S. Kiami, A. Gupta, and V. Kumar. Genaug: Retargeting behaviors to unseen situations via generative augmentation. *Robotics: Science and Systems*, 2023.

[117] Z. Mandi, H. Bharadhwaj, V. Moens, S. Song, A. Rajeswaran, and V. Kumar. Cacti: A framework for scalable multi-task multi-scene visual imitation learning. In *CoRL 2022 Workshop on Pre-training Robot Learning*.

[118] K. Black, M. Nakamoto, P. Atreya, H. R. Walke, C. Finn, A. Kumar, and S. Levine. Zero-shot robotic manipulation with pre-trained image-editing diffusion models. In *The Twelfth International Conference on Learning Representations*, 2024.

[119] F. Sadeghi, A. Toshev, E. Jang, and S. Levine. Sim2real viewpoint invariant visual servoing by recurrent control. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4691–4699, 2018.

[120] M. Liu, X. Li, Z. Ling, Y. Li, and H. Su. Frame mining: a free lunch for learning robotic manipulation from 3d point clouds. In *Conference on Robot Learning*, pages 527–538. PMLR, 2023.

[121] A. Goyal, J. Xu, Y. Guo, V. Blukis, Y.-W. Chao, and D. Fox. Rvt: Robotic view transformer for 3d object manipulation. In *Conference on Robot Learning*, pages 694–710. PMLR, 2023.

[122] N. Hirose, D. Shah, A. Sridhar, and S. 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.

[123] K. Tahara and N. Hirose. Ex-dof: Expansion of action degree-of-freedom with virtual camera rotation for omnidirectional image. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 10382–10389. IEEE, 2022.

[124] Y. Seo, J. Kim, S. James, K. Lee, J. Shin, and P. Abbeel. Multi-view masked world models for visual robotic manipulation. In *International Conference on Machine Learning*, pages 30613–30632. PMLR, 2023.

[125] A. Zhou, M. J. Kim, L. Wang, P. Florence, and C. Finn. Nerf in the palm of your hand: Corrective augmentation for robotics via novel-view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17907–17917, 2023.

[126] X. Zhang, M. Chang, P. Kumar, and S. Gupta. Diffusion meets dagger: Supercharging eye-in-hand imitation learning. In *Proceedings of Robotics: Science and Systems*, Delft, Netherlands, 2024.

[127] J. H. Yang, D. Sadigh, and C. Finn. Polybot: Training one policy across robots while embracing variability. In *Conference on Robot Learning*, pages 2955–2974. PMLR, 2023.

[128] J. Yang, C. Glossop, A. Bhorkar, D. Shah, Q. Vuong, C. Finn, D. Sadigh, and S. Levine. Pushing the limits of cross-embodiment learning for manipulation and navigation. 2024.

[129] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4015–4026, 2023.

[130] L. Zhang, A. Rao, and M. Agrawala. Adding conditional control to text-to-image diffusion models, 2023.

[131] K. Sargent, Z. Li, T. Shah, C. Herrmann, H.-X. Yu, Y. Zhang, E. R. Chan, D. Lagun, L. Fei-Fei, D. Sun, et al. Zeronvs: Zero-shot 360-degree view synthesis from a single image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9420–9429, 2024.

[132] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(4):834–848, 2018. doi:10.1109/TPAMI.2017.2699184.

[133] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[134] P. Arbeláez, B. Hariharan, C. Gu, S. Gupta, L. Bourdev, and J. Malik. Semantic segmentation using regions and parts. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3378–3385. IEEE, 2012.

[135] M. S. Lee, W. Shin, and S. W. Han. Tracer: Extreme attention guided salient object tracing network (student abstract). In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pages 12993–12994, 2022.

[136] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

[137] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[138] C. Xu, H. Ling, S. Fidler, and O. Litany. 3difftection: 3d object detection with geometry-aware diffusion features. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10617–10627, 2024.

[139] P. Li, T. Liu, Y. Li, M. Han, H. Geng, S. Wang, Y. Zhu, S.-C. Zhu, and S. Huang. Ag2manip: Learning novel manipulation skills with agent-agnostic visual and action representations. *arXiv preprint arXiv:2404.17521*, 2024.

[140] Z. Li, C.-Z. Lu, J. Qin, C.-L. Guo, and M.-M. Cheng. Towards an end-to-end framework for flow-guided video inpainting. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[141] G. Qian, J. Mai, A. Hamdi, J. Ren, A. Siarohin, B. Li, H.-Y. Lee, I. Skorokhodov, P. Wonka, S. Tulyakov, et al. Magic123: One image to high-quality 3d object generation using both 2d and 3d diffusion priors. In *The Twelfth International Conference on Learning Representations*, 2024.

[142] C. Chi, S. Feng, Y. Du, Z. Xu, E. Cousineau, B. Burchfiel, and S. Song. Diffusion policy: Visuomotor policy learning via action diffusion. In *Learning agile robotic locomotion skills by imitating animals*, 2023.

[143] Y. Zhu, J. Wong, A. Mandlekar, R. Martín-Martín, A. Joshi, S. Nasiriany, and Y. Zhu. robosuite: A modular simulation framework and benchmark for robot learning. In *arXiv preprint arXiv:2009.12293*, 2020.

[144] L. Y. Chen, S. Adebola, and K. Goldberg. Berkeley UR5 demonstration dataset. https://sites.google.com/view/berkeley-ur5/home.

[145] Octo Model Team, D. Ghosh, H. Walke, K. Pertsch, K. Black, O. Mees, S. Dasari, J. Hejna, C. Xu, J. Luo, T. Kreiman, Y. Tan, D. Sadigh, C. Finn, and S. Levine. Octo: An open-source generalist robot policy. https://octo-models.github.io, 2023.

[146] B. Van Hoorick, R. Wu, E. Ozguroglu, K. Sargent, R. Liu, P. Tokmakov, A. Dave, C. Zheng, and C. Vondrick. Generative camera dolly: Extreme monocular dynamic novel view synthesis. 2024.

[147] A. Kodaira, C. Xu, T. Hazama, T. Yoshimoto, K. Ohno, S. Mitsuhori, S. Sugano, H. Cho, Z. Liu, and K. Keutzer. Streamdiffusion: A pipeline-level solution for real-time interactive generation. *arXiv preprint arXiv:2312.12491*, 2023.

[148] A. Mandlekar, D. Xu, J. Wong, S. Nasiriany, C. Wang, R. Kulkarni, L. Fei-Fei, S. Savarese, Y. Zhu, and R. Martín-Martín. What matters in learning from offline human demonstrations for robot manipulation. In *Conference on Robot Learning*, pages 1678–1690. PMLR, 2022.

# 7 Appendix

In this section, we provide additional implementation details of RoVi-Aug and our physical experiments.

## 7.1 Algorithm Pseudocode

In this section, we provide the pseudocode for Ro-Aug and Vi-Aug.

**Algorithm 1** Ro-Aug

**Input:** A sequence of source robot image observations $D_i^{\mathcal{S}} = \{o_1^{\mathcal{S}}, ..., o_{H_i}^{\mathcal{S}}\}$
**Output:** A sequence of cross-painted observations with synthesized target robot: $D_i^{\mathcal{S}\to\mathcal{T}} = \{o_1^{\mathcal{S}\to\mathcal{T}}, ..., o_{H_i}^{\mathcal{S}\to\mathcal{T}}\}$

**function** RO-AUG($D_j^{\mathcal{S}}$)
  **for** each image $o_j^{\mathcal{S}}$ in $D_i^{\mathcal{S}}$ **do**
    Segment the source robot out, resulting in the robot $r_j^{\mathcal{S}}$ and background $b_j^{\mathcal{S}}$ where $o_j^{\mathcal{S}} = r_j^{\mathcal{S}} \cup b_j^{\mathcal{S}}$
  **end for**
  **for** each robot image $r_j^{\mathcal{S}}$ **do**
    Apply the Robot-to-Robot generation model to get $r_j^{\mathcal{S}\to\mathcal{T}}$
  **end for**
  Apply video inpainting model E2FGV to the background video $\{b_1^{\mathcal{S}}, ..., b_{H_i}^{\mathcal{S}}\}$ to get $\{\tilde{b_1}^{\mathcal{S}}, ..., \tilde{b_{H_i}}^{\mathcal{S}}\}$
  **for** each robot image $r_j^{\mathcal{S}\to\mathcal{T}}$ **do**
    $o_j^{\mathcal{S}\to\mathcal{T}}$ = Overlay $r_j^{\mathcal{S}\to\mathcal{T}}$ onto $\tilde{b_j}^{\mathcal{S}}$ `# Combine the background and the generated target robot images`
  **end for**
  **return** $D_i^{\mathcal{S}\to\mathcal{T}} = \{o_1^{\mathcal{S}\to\mathcal{T}}, ..., o_{H_i}^{\mathcal{S}\to\mathcal{T}}\}$
**end function**

**Algorithm 2** Vi-Aug

**Input:** A sequence of robot image observations $D_i = \{o_1, ..., o_{H_i}\}$
**Output:** A sequence of viewpoint augmented images: $D_i^{\text{Vi-Aug}} = \{o_1^{R_1,T_1}, ..., o_{H_i}^{R_{H_i},T_{H_i}}\}$

**function** VI-AUG($D_i$)
  **for** each image $o_j$ in $D_i$ **do**
    Sample perturbations $(\tilde{R}_j, \tilde{T}_j) \in SE(3)$ from a box range
    Generate augmented images using ZeroNVS $f$: $o_j^{\tilde{R},\tilde{T}} = f\left(o_j; \tilde{R}, \tilde{T}\right)$
  **end for**
  **return** $D_i^{\text{Vi-Aug}} = \{o_1^{R_1,T_1}, ..., o_{H_i}^{R_{H_i},T_{H_i}}\}$
**end function**

## 7.2 Robot Augmentation

### 7.2.1 Training Data Generation

To train our robot segmentation and Robot-to-Robot generation models, we use the Robosuite simulator [143] to generate a large dataset of paired robot images with corresponding masks with randomly sampled robot poses and camera poses. The sampling procedure is as follows: The robot pose is specified by the end-effector pose. The translation component is sampled uniformly with $(x, y, z) \in [-0.25, 0.25] \times [-0.25, 0.25] \times [0.6, 1.3]$ (unit in meters). For the rotation component, we parameterize it as [inward, rightward, z_axis]. To bias the unit vector z_axis towards pointing downward, we parameterize it using spherical coordinate $\theta, \phi$ where $\theta$ (zenith angle) is sampled from a normal distribution $\mathcal{N}(\pi, \pi/3.5)$ and $\phi$ (azimuthal angle) is uniformly sampled between 0 and $2\pi$.

After sampling the robot pose, we randomly sample the camera pose with the following procedure: The position is sampled from a half hemisphere with radius $r \in \mathcal{N}(0.85, 0.2)$ and zenith angle $\theta \in \mathcal{N}(\pi/4, \pi/2.2)$, and azimuthal angle $\phi \in \text{Unif}[-\pi \cdot 3.7/4, \pi \cdot 3.7/4]$. The viewing direction is

towards the center of the hemisphere, which we offset as the gripper position. We also sample camera field of view between 40 and 70. Finally, we randomly perturb the camera pose with noises.

We randomly sample robot poses, and for each robot pose, we randomly sample 5 different camera poses. In addition to pure random sampling, we also add some camera poses and robot poses similar to those in the RT-X datasets and add perturbations. We obtain paired images between different robots and their segmentation mask from Robosuite, and we add random brightness augmentation with range $[-40, 40]$ to the source robot images to increase the robustness of the segmentation model and R2R model to real-world lighting. In this way, we obtain about 800k images for each of the 4 robot types: Franka, UR5, Sawyer, and Jaco. See Fig. 5 for some example images.

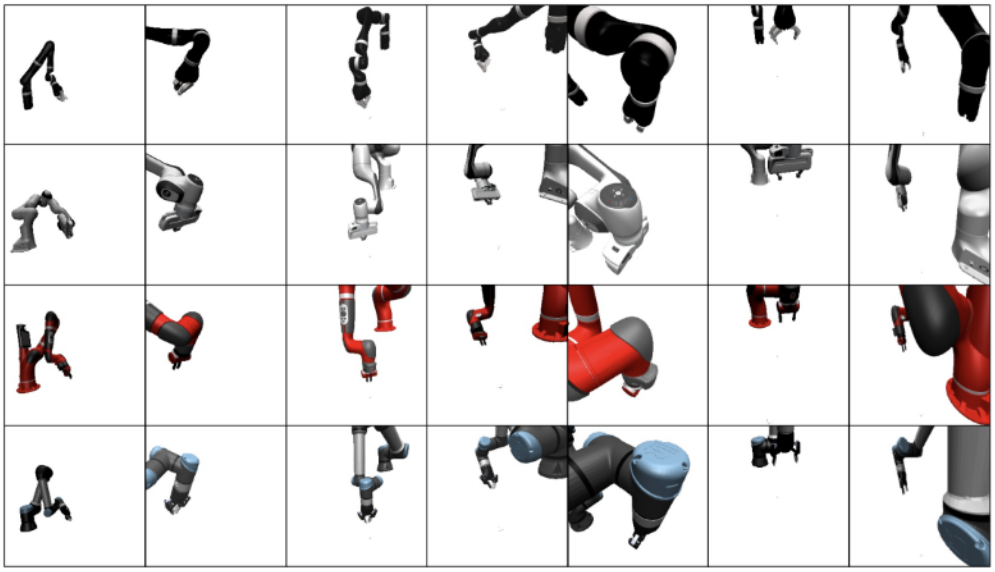

Figure 5: **Example of paired images for training the R2R model.** We use Robosuite [143] to generate pairs of Jaco, Franka, Sawyer, and UR5 at the same pose.

To create the dataset for training the segmentation model, we paste the generated robot image onto backgrounds from ImageNet [137]. See Fig. 6 for some example images.

Figure 6: Example of pasted images on ImageNet used for training the segmentation model.

### 7.2.2 Model Training Details

Regarding the robot segmentation model, we fine-tune SAM with LoRA with 4 A6000 GPU for 1.5 hours. In particular, we leverage mixed-precision (8-bit and 16-bit) and the torch.compile feature to accelerate training. The model is trained with a mini-batch size of 64, a learning rate of 1e-5, and a LoRA rank of 4.

Regarding the Robot-to-Robot generation model, we finetune Stable Diffusion with ControlNet on 1 A100 GPU for 36 hours on 800K paired images. We use a learning rate of 1e-4 and a batch size of 512. During inference, we leverage the Stream Batch proposed by Kodaira et al. [147] to batchify the generation phase, making the generation phase achieve around 3.2 FPS.

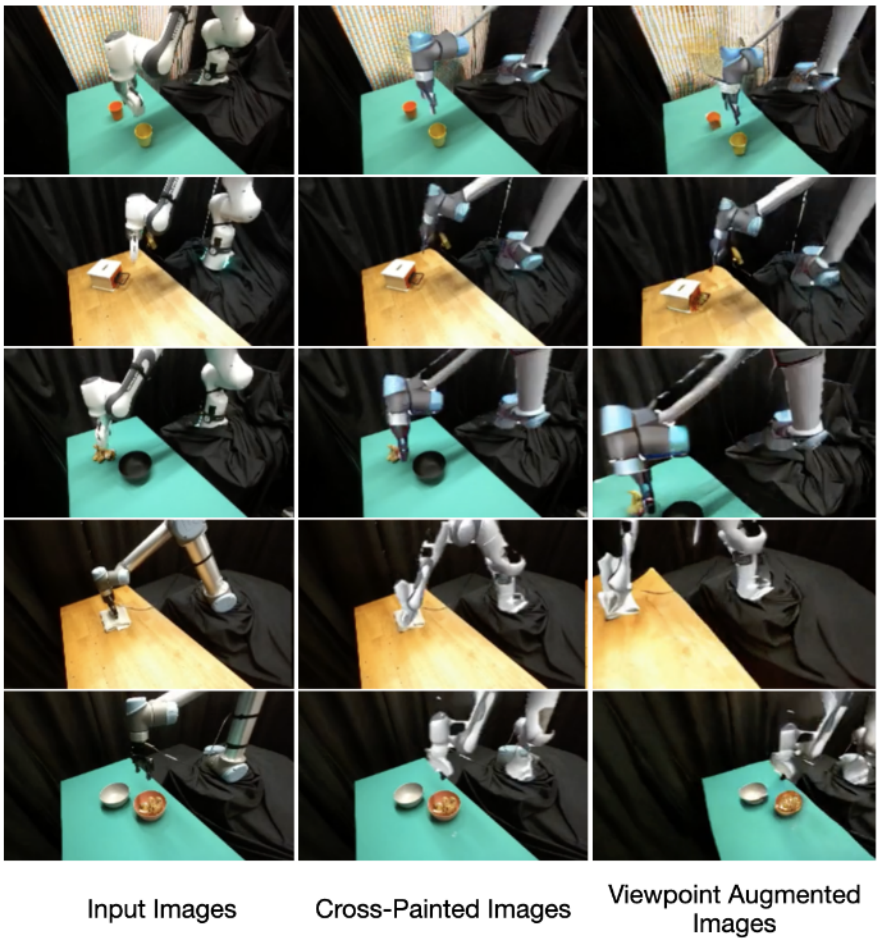


Figure 7: **Example of RoVi-Aug results.** We show some example results of RoVi-Aug applied to the training images of the 5 tasks.

We use ZeroNVS and the video inpainting model E2FGVI off-the-shelf without finetuning.

#### 7.2.3 Computation Time for Data Augmentation

The advantage of RoVi-Aug over Mirage is that the primary of the compution is performed offline, not during execution time. Moreover, each model in RoVi-Aug's pipeline can be parallelized to process batchified video frames efficiently. We measured the throughputs of each module: Robot segmentation model achieves 4.1FPS, Robot-to-Robot achieves 3.2FPS, and the video inpainting model achieves 4.6FPS. On a single A100 GPU, it takes about 4-5 hours to perform Ro-Aug on a dataset of 200 trajectories. Similarly, the throughput for ZeroNVS inference is 1.3FPS, translating to 4.2 hours of viewpoint augmentation time on a dataset of 200 trajectories.

#### 7.2.4 Example Augmented Images

In Fig. 7, we show some example results of RoVi-Aug applied to the training images of the 5 tasks. The left column is the original images; the middle column is the cross-painted images using the robot augmentation pipeline; the right column shows the view augmented images applied on top of the robot augmented images. The black regions in the generated robot are due to incomplete segmentation mask (missing some regions in the generated robot) when pasting the generated robot to the original image. We can see that in general, RoVi-Aug generates diverse view angles of the target robot performing the task of interest.

#### 7.2.5 Generation Artifacts

We observe a few different types of artifacts: 1) illumination difference, 2) inaccurate object segmentation, 3) temporal inconsistency, and 4) inaccurate robot-to-robot generation.

For 1), since there are almost always differences in the lighting conditions between the simulated images that are used to train the R2R diffusion model and that of the test robots which are unknown a priori, we perform random brightness augmentation to the generated robot scenes in the augmentation pipeline. As shown in Table 1, we find this mitigation strategy is generally effective.

For 2), the robot segmentation model may sometimes under-segment or over-segment, particularly when the source robot is occluded or interacting with objects. As the R2R diffusion model is not trained on source robot images with objects in the gripper or with a partially segmented robot, the generated target robot can have large artifacts including distortion or hallucination due to out-of-distribution inputs.

For 3), due to the stochastic nature of diffusion models and possible multiple inverse kinematics solutions for putting the end effector of the target robot at the position of the source robot with different joint angles, the generated images may not be consistent across time. We did not observe this as a big problem potentially due to two reasons: (1) The Diffusion Policy does not use a long history so temporally inconsistent artifacts may not have a large effect; (2) The stochasticity of the generated images has an effect of randomization, which may help the policy be more robust to visual artifacts. Future work could also use a video diffusion model [13] to perform robot generation based on the entire robot trajectory to improve robot pose consistency.

For 4), even though our robot-to-robot diffusion model is trained on a large number of paired robot data, the generated images may still contain visible artifacts. For example, due to the ambiguity of inferring the field of view parameter from an image, the generated robot arm may be too thin or too thick. The generated gripper may also have artifacts or its position or orientation may not completely align with the source robot.

Due to these artifacts, we observe in Table 1 that Mirage achieves better performance than Ro-Aug on tasks that require more precision, such as cup stacking. This is because Mirage has the benefit of using a URDF with precise camera calibration to put the gripper at the exact location desired. On the other hand, artifacts in the R2R Generation model mean that the gripper of the target robot may not have the exact same pose as the original robot. However, as we show in Table 2, the ability of RoVi-Aug to perform finetuning can bring the performance higher than Mirage.

### 7.3 Physical Experiment Details

We provide more details on the physical experiment setups described in Section 5.2.

For the Franka-UR5 transfer experiments, we study 5 tasks: (1) Open a drawer, (2) Pick up a toy tiger from the table and put it into a bowl (Place Tiger), (3) Stack cups, (4) Sweep cloth from right to left, and (5) Transport a toy tiger between two bowls. For each task, the initial position of the robot gripper is randomized. For (1), the position and orientation of the drawer on the table is randomized, and the goal for the robot gripper is to go into the handle, pull it out, and leave the drawer. For (2), the positions of the tiger and the drawer are randomized. For (3), the positions of both cups are randomized. For (4), the initial position of the cloth is randomized in the right region of the table, and the robot needs to push it to the left region of the table, a distance of about 0.5 m. For (5), there are 2 bowls (red and grey) whose positions are randomized, and the toy tiger is always in the red bowl initially. The robot needs to grasp it and drop it into the grey bowl. Among them, stacking cup requires high precision and is most difficult, and sweeping cloth is the easiest.

For the OXE dataset experiments, the 2 tasks from the Berkeley UR5 datasets (Transport Tiger, Sweep Cloth) are the same as (4) and (5) above. For the 2 tasks from the Jaco Play datasets, the "Pick Cup" task requires the robot to pick up a cup that is randomly initialized on the table, and the "Bowl in Oven" task requires the robot to pick up a bowl and put it into a toaster oven.

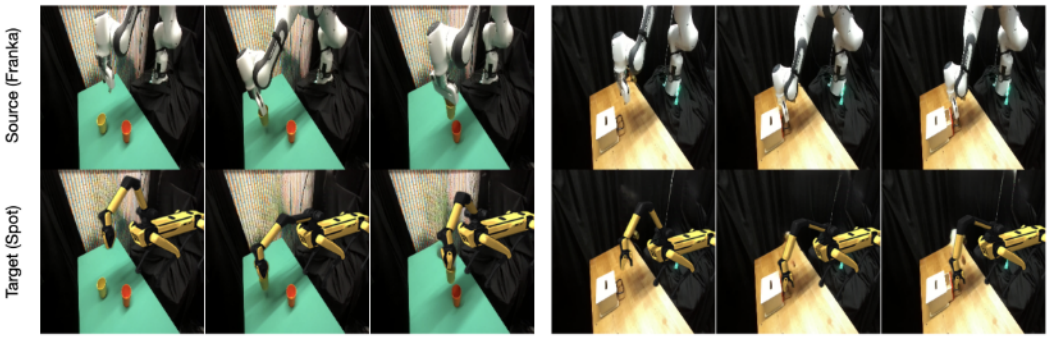


Figure 8: Example of robot augmentation from Franka to a Boston Dynamics Spot.

#### 7.3.1 Policy Learning Details

We use the codebase from DROID [5] as our Diffusion Policy implementation, which is an open-source version integrated with Robomimic [148]. Similar to them, We use downsample camera observations at a resolution of $128 \times 128$ and the robot proprioception as input, and produce absolute robot end-effector translation, rotation, and gripper actions. And as with DROID and the original Diffusion Policy implementation, we train the diffusion policy to generate 16-step action sequences, and during rollouts, step 8 actions open loop before re-running policy inference. Compared to DROID, we use a ResNet-18 visual encoder instead of a pre-trained ResNet-50 for faster training, and we do not condition the policy with language input since we train a separate policy for each task (or 2 tasks for Table 5).

For few-shot finetuning experiments, we did not freeze any part of the diffusion policy and simply continued training on the target robot dataset (5/10 demonstrations) for only 100 epochs (about 20 minutes) to prevent the policy from overfitting to the target data too much.

#### 7.3.2 Failure Modes

We describe the common failure modes of RoVi-Aug and baselines here. For the 3 pick and place tasks ("Place Tiger," "Stack Cup," and "Transport Tiger"), failure cases are usually missed grasp or inaccurate placing. For "Open Drawer," failure cases are typically gripper missing the drawer handle. For "Sweep Cloth," failure cases include inaccurate reaching and gripper being too high or leaving the table too early during the trajectory. For baselines, failure modes also include the robot getting confused and simply hovering over the objects without performing the task.

### 7.4 Model and Computation Details

For Ro-Aug, our segmentation model is a 636M-parameter SAM model with 35.6M-parameter LoRA layers; the video inpainting model E2FGVI is a 41.8M parameter model that we use off-the-shelf; the Robot-to-Robot (R2R) Generation model is a 1B-parameter Stable Diffusion model with around 350M-parameter ControlNet. For Vi-Aug, ZeroNVS is a 1B model that we use off-the-shelf. For policy learning, we use Diffusion Policy with a ResNet18 encoder and 1D-UNet with 80M parameters in total.

### 7.5 Example of Cross-Painting with a Mobile Robot

Generalization from arms mounted on a stationary base to mobile robots is much more challenging. In this section, we try an experiment using images of the Franka arm and apply robot augmentation to replace the Franka with a Boston Dynamics Spot to illustrate some examples with cross-painting to a mobile robot. While we do not have the hardware to perform physical experiments, the cross-painted images look somewhat realistic (see Figure 8), so it may be possible that the cross-painted Franka dataset could jumpstart the training for Spot. There are additional challenges associated with mobile

manipulation, such as coordination between base and arm movements and less accurate arm control, which we will leave as future work.