Phantom: Training Robots Without Robots Using Only Human Videos

Marion Lepert[1], **Jiaying Fang**[1], **Jeannette Bohg**[1]
[1]Stanford University
https://phantom-human-videos.github.io/

Abstract: Training general-purpose robots requires learning from large and diverse data sources. Current approaches rely heavily on teleoperated demonstrations which are difficult to scale. We present a scalable framework for training manipulation policies directly from human video demonstrations, requiring no robot data. Our method converts human demonstrations into robot-compatible observation-action pairs using hand pose estimation and visual data editing. We inpaint the human arm and overlay a rendered robot to align the visual domains. This enables zero-shot deployment on real hardware without any fine-tuning. We demonstrate strong success rates—up to 92%—on a range of tasks including deformable object manipulation, multi-object sweeping, and insertion. Our approach generalizes to novel environments and supports closed-loop execution. By demonstrating that effective policies can be trained using only human videos, our method broadens the path to scalable robot learning.

Keywords: Robot Imitation Learning, Learning from human videos

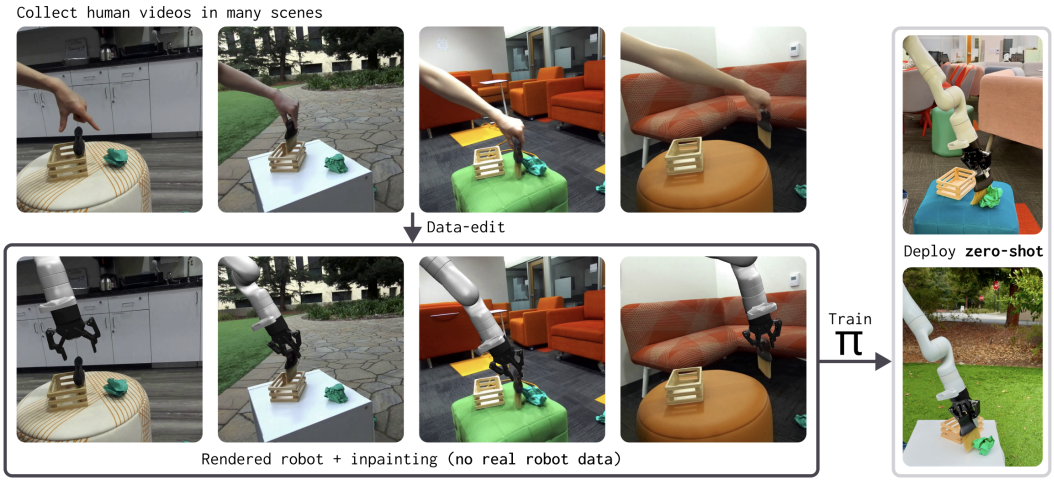

Figure 1: **Overview of Phantom.** Our method enables training robot policies without collecting any robot data. We first collect human video demonstrations in diverse environments and use inpainting to remove the human hand. A rendered robot is then inserted into the scene using the estimated hand pose. The resulting augmented dataset is used to train an imitation learning policy, which is deployed zero-shot on a real robot.

1 Introduction

Data scarcity remains a key challenge in advancing robotics research. While large-scale data collection efforts are gaining momentum, robotics datasets [1, 2, 3] remain orders of magnitude smaller than those used to train generalist vision and language models. These efforts are constrained by the slow and costly process of collecting data with robotics hardware. Moreover, increasing the quantity of data alone is not enough—*diversity* in data is equally critical for generalization [4]. Collecting diverse data across many environments with physical robots remains a formidable challenge.

9th Conference on Robot Learning (CoRL 2025), Seoul, Korea.

Human videos offer a compelling alternative: they are abundant, diverse, and rich in task information. However, leveraging them for robot learning poses two key challenges: (1) human videos lack explicit action labels, and (2) human appearance differs substantially from robots. Prior methods trying to leverage human videos typically rely on co-training with robot data [5, 6, 7, 8, 9] or reinforcement learning [10, 11, 12, 13]. Such approaches fail to extract sufficient learning signals from human data alone, necessitating robot data to bridge the gap. These strategies are fundamentally limited in scalability.

Our goal is to remove these bottlenecks by developing a framework that can scale with human videos alone. Our work focuses on the emerging data regime in which human videos become orders of magnitude more prevalent than robot demonstrations. While we do not demonstrate large-scale deployment in this work, we address the asymptotic setting where human data grows rapidly while robot data remains scarce. To that end, we propose an approach that requires only human video demonstrations to train a robot policy. Our method converts human videos into data-edited, *Phantom* robot demonstrations by extracting actions using a human hand pose estimator and replacing the human arm with a rendered robot. We then train a closed-loop imitation learning policy on these demonstrations and deploy it zero-shot on a real robot. Our method achieves high success rates across six tasks—including deformable object manipulation and generalization to novel environments—without requiring any robot data.

Our work draws inspiration from recent success in data-editing methods for robot-to-robot transfer [14, 15, 16]. These techniques rely on precise proprioception and action labels, and until now, have not been effectively adapted to the more challenging human-to-robot setting. Our method does exactly this, leveraging data-editing for human-to-robot transfer. That such a method works and generalizes across diverse environments underscores a striking and perhaps unexpected insight: human demonstrations alone, when subject to data editing and careful hand pose estimation, can be directly leveraged for training a robot policy. Our method requires no robot data, no manual annotations, no object models, and can be deployed on any robot capable of executing the task.

To summarize, our main contribution is demonstrating that data-editing-based cross-embodiment learning techniques are adaptable to human-to-robot transfer, which unlocks their utility for scaling robot learning through human video demonstrations.

2 Related Works

2.1 Learning from Human Videos

Many works have explored leveraging human videos to improve robot policies. A prominent line of research focuses on using diverse in-the-wild videos (e.g., YouTube) to improve generalization. Common strategies include pre-training visual representations [17, 18, 19, 20], learning reward functions [21, 10, 12, 13], building world models [22, 23, 24, 25, 26], and learning object motion priors [6, 27, 28]. These methods struggle to overcome the wide embodiment gap between humans and robots and rely extensively on robot data. Shi et al. [29] needs no robot data but only models post-grasp motion and requires manually collecting a goal image in the target scene.

Alternative approaches leverage curated human video demonstrations, which simplify the problem by ensuring that videos explicitly show task-relevant behaviors. While these videos must be manually collected, they are faster to gather relative to robot teleoperation data. Some works use human video demonstrations to learn motion priors [30, 8, 31, 32]. MimicPlay [8] trained a high-level planner using human videos alongside a plan-guided imitation learning policy trained with robot demonstrations. Other methods [33, 9] rely on paired human video and robot data to bridge the embodiment gap.

Object-centric approaches have also been explored as an alternative direction [34, 35, 36]. These methods typically estimate and track target object poses from video demonstrations, and learn a robot policy conditioned on the extracted object trajectories [37]. However, they require identifying objects of interest and estimating rigid-body transformations which makes it hard to apply

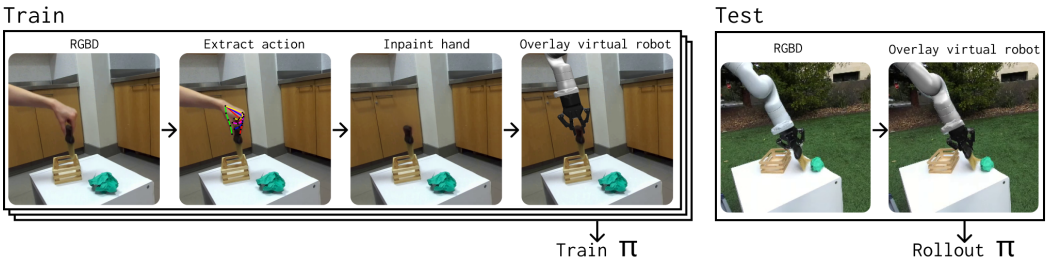

Figure 2: **Overview of our pipeline for learning robot policies from human videos.** During training, we first estimate the hand pose in each frame of a human video demonstration and convert it into a robot action. We then remove the human hand using inpainting and overlay a virtual robot in its place. The resulting augmented dataset is used to train an imitation learning policy, π. At test time, we overlay a virtual robot on real robot observations to ensure visual consistency, enabling direct deployment of the learned policy on a real robot.

them to scenarios with deformables, granular materials, or multiple objects. Flow-based methods [11, 38, 39, 40, 5] address some of these limitations by tracking trajectories of points instead of rigid body transformations. Wen et al. [39], Ren et al. [5] track points on the human embodiment, which provides information on the general direction the robot should move in, but because humans and robots move in different ways, these methods still require robot data to refine the motion. Conversely, Xu et al. [40] only tracks flow on the manipulated object, but relies on object detection and simulation environments to refine robot motions. [38] and [41] exclusively use human data but [38] is limited to open-loop execution and [41] requires manual annotations of semantically meaningful object points. In contrast, our method is closed-loop, not bottlenecked by an object detector or the need for manual annotation, and works equally well on rigid, deformable, and multiple objects. A summary comparison of our method with prior work across key dimensions is provided in Table 4 in the appendix.

2.2 Data Editing for Cross-Embodiment Learning

While many works focus on human-to-robot transfer, robot-to-robot cross-embodiment learning is also gaining attention as large-scale robotics datasets increasingly incorporate diverse data sources. Vision-based policies face significant challenges with cross-embodiment learning due to distribution shifts caused by the varying appearances of different embodiments. To address this, several methods propose data-editing strategies to mitigate these shifts. RoviAug [15] uses inpainting during training to remove the source embodiment from images and overlays a virtual rendering of the target embodiment in the same pose. At test time, the policy is deployed directly on the target embodiment. Shadow [16] replaces both the source and target robots with composite segmentation masks at train and test time, ensuring a close match between the input data distributions. Other methods, such as EgoMimic [7] and AR2-D2 [42], adapt data-editing techniques for human-to-robot transfer. EgoMimic masks out each embodiment and overlays a red line along each arm, while AR2-D2 employs the same inpainting and virtual overlay strategy as [15]. However, both methods still rely on co-training with robot data to bridge the human-to-robot embodiment gap.

3 Approach

3.1 Problem Setup

We assume access to a dataset $\mathcal{D}_{\text{human}} = \{\tau_h^i\}_{i=1}^N$ of N human (h) video demonstrations τ_h^i of a manipulation task. Each demonstration consists of a sequence of images $\{I_{h,t}\}_{t=1}^T$ captured from a third-person viewpoint using an RGBD camera. The demonstration is performed using a pinch grasp with the thumb and index finger.

Our goal is to use only these human video demonstrations to train a closed-loop policy using imitation learning that can be deployed zero-shot on a target robot (r) for which no data has ever been collected. To do so, we use a data-editing strategy to convert our dataset $\mathcal{D}_{\text{human}}$ into $\mathcal{D}_{\text{robot}} = \{\tau_r^i\}_{i=1}^N$. Our objective is to convert each frame of a human demonstration into a corre-

sponding robot observation-action pair: $I_{h,t} \rightarrow (I_{r,t}, a_{r,t})$. The goal of data-editing is for each $I_{r,t}$ to be drawn from the same distribution as images at test time on the target robot. Then, we can simply train our imitation learning policy on $\mathcal{D}_{\text{robot}}$ and deploy it on test-time robot observations.

Each robot action $a_{r,t}$ consists of the position and orientation of the end-effector and the opening width of the gripper $a_{r,t} = (\mathbf{p}_t, \mathbf{R}_t, g_t)$, where $\mathbf{p}_t \in \mathbb{R}^3$ is the Cartesian position of the end-effector, $\mathbf{R}_t \in \mathbb{R}^6$ represents the orientation of the end-effector using a 6D continuous rotation representation, and $g_t \in [0, 1]$ is the normalized opening width of the gripper.

We assume the deployment setup camera extrinsics are known. While train and test scenes do not need to match, we assume the camera used for data collection is at a similar viewpoint to that used at test time. This constraint could be relaxed by collecting data across a broader range of viewpoints, as in [43], but such large-scale data collection is beyond the scope of this work.

3.2 Action Labeling of Human Videos

Since human videos lack explicit action information, we first address how to go from a frame of the human video $I_{h,t+1}$ to the corresponding robot action for the previous frame $a_{r,t} = (\mathbf{p}_t, \mathbf{R}_t, g_t)$. We start by estimating the human hand pose at each frame $I_{h,t}$ using HaMeR [44]. HaMeR predicts 21 keypoints, $\hat{\mathbf{X}}_t \in \mathbb{R}^{21\times 3}$, corresponding to anatomical landmarks, along with a dense set of 778 vertices, $\hat{\mathbf{V}}_t \in \mathbb{R}^{778\times 3}$, representing the hand mesh.

While HaMeR accurately captures hand shape, it struggles to estimate the absolute 3D pose due to its reliance on a monocular image. To refine this estimate, we incorporate depth data. We segment the hand in the RGB image using SAM2 [45], yielding a segmentation mask M_t. Using M_t and the corresponding depth image D_t, we extract a partial hand point cloud, $\mathbf{P}_t$. We then align the HaMeR-predicted mesh $\hat{\mathbf{V}}_t$ with $\mathbf{P}_t$ via Iterative Closest Point (ICP) registration, obtaining the optimal rigid transformation $\mathbf{T}_t \in SE(3)$ such that $\mathbf{P}_t \approx \mathbf{V}_t = \mathbf{T}_t\hat{\mathbf{V}}_t$ (see Fig. 3). Since $\hat{\mathbf{V}}_t$ and $\hat{\mathbf{X}}_t$ are internally consistent, we apply $\mathbf{T}_t$ to the predicted keypoints to refine their positions: $\mathbf{X}_t = \mathbf{T}_t\hat{\mathbf{X}}_t$.

HaMeR also struggles with keypoints that are occluded in the RGB image—an issue exacerbated during grasping. Since HaMeR models all hand joints as ball joints, it often predicts unrealistic finger configurations under occlusion. To address this, we constrain the last two joints of the thumb and index fingers to a single degree of freedom, limiting their movement to anatomically feasible ranges. This ensures more accurate finger pose estimation when occlusions occur.

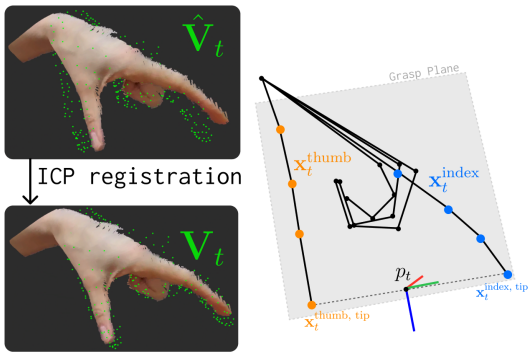

Figure 3: **Left**: To refine the HaMeR predicted mesh points $\hat{\mathbf{V}}_t$ shown in green, we use ICP registration to align them with the partial point cloud of the hand, $\mathbf{P}_t$ to obtain $\mathbf{V}_t$. **Right**: After aligning the HaMeR keypoints with the hand point cloud, we calculate the target position $\mathbf{p}_t$.

We use the refined keypoints $\mathbf{X}_t$ to define a target action for our policy, visualized in Fig. 3. The target position, $\mathbf{p}_t$, is set as the midpoint between the keypoints at the tips of the thumb, $\mathbf{x}_t^{\text{thumb, tip}}$, and index finger, $\mathbf{x}_t^{\text{index, tip}}$. For the target orientation, $\mathbf{R}_t$, we fit a plane through all the keypoints of the thumb $\mathbf{x}_t^{\text{thumb}}$ and index finger $\mathbf{x}_t^{\text{index}}$ and compute a principal axis by fitting a vector through the keypoints of the thumb. $\mathbf{R}_t$ is then defined using the normal of this plane and the fitted vector. The gripper opening g_t is computed as the distance between the keypoints corresponding to the fingertips of the thumb and index finger, $\mathbf{x}_t^{\text{thumb, tip}}$ and $\mathbf{x}_t^{\text{index, tip}}$. To mitigate slippage during grasping, we enforce a threshold, setting the bottom 20th percentile of predicted gripper distances in a single trajectory to fully closed.

HaMeR predicts keypoints in the camera's reference frame, meaning that p_t and $\mathbf{R}_t$ are also expressed in this frame. We convert them into the robot's frame using the known camera extrinsics of our target setup to obtain the final robot action $a_{r,t}$.

3.3 Bridging the Visual Observation Gap

Human arms and hands appear visually distinct from robot arms and grippers. A vision-based policy trained solely on human demonstrations struggles to generalize to a robot embodiment. To address this, we adapt the data-editing scheme from Rovi-Aug [15] to the human-to-robot transfer setting to compute $I_{h,t} \rightarrow I_{r,t}$. The edited images are used to train an imitation learning policy, which is then deployed on the target robot.

Data-editing during training: Each frame in the training dataset contains an image of a human arm performing a task. To replace the human embodiment with a robot, we first segment out the pixels corresponding to the human arm using SAM2 [45], and then remove the segmented arm via inpainting using E2FGVI [46]. Next, we render a virtual model of the target robot with its end effector in the corresponding pose, obtained from Section 3.2 (i.e. its end effector pose at $(\mathbf{p}_t, \mathbf{R}_t, g_t)$). Given the known camera extrinsics, we synthesize an image of the robot from the appropriate viewpoint and overlay it onto the original image. To ensure realistic occlusions, we use depth data to determine which parts of the overlaid robot should be masked by objects in the environment. The final result is an image that closely resembles a real robot completing the task, as illustrated in Figure 2.

Data-editing at inference time: At inference time, each observation image contains a real robot arm. However, training images feature a rendered robot arm, which may have slight discrepancies in color and texture. To minimize domain shift, we overlay a virtual robot arm onto the real robot in each observation image, ensuring consistency between train and test distributions. An alternative approach, as proposed in Rovi-Aug, is to introduce color variations in the overlays during training to make the policy robust to these shifts. However, since this strategy has already been explored in prior work, we opt for the simpler inference-time overlay approach.

4 Results

We evaluate our method across a range of tasks that highlight the versatility of our method. To demonstrate that our method works across different robots, we present results on both a Franka and a Kinova robot. Policies are trained using Diffusion Policy [47], with OSC and IK controllers used for low-level control of the Franka and Kinova, respectively. Virtual robot renderings are generated using Mujoco [48].

4.1 Comparison of Data Editing Methods

To the best of our knowledge, no prior (non-concurrent) published work has trained closed-loop imitation learning policies using only human video demonstrations that can manipulate rigid and deformable objects, and groups of objects without requiring any manual annotations. After careful analysis of all candidate baselines (see detailed justification in Appendix 6.1), we focus our experimental analysis on determining the most effective data-editing strategy for human-to-robot policy transfer (see Figure 4):

Hand Inpaint (Phantom - Ours): We adapt the data-editing strategy from Rovi-Aug [15], which was developed for the simpler robot-to-robot setting, to the human-to-robot setting. During training, the human arm is segmented out and replaced with inpainting. An image of the target robot is synthesized using a virtual model from the appropriate viewpoint and overlaid onto the original image. At test time, a rendered robot arm is overlaid onto the real robot arm to minimize domain shift. See Section 3.3 for more details.

Hand Mask: We adapt the data-editing strategy from Shadow [16]. This method was also developed for the simpler robot-to-robot setting. During training, the human arm is masked out, and a virtual robot in the same pose is overlaid. While the original method overlays a black mask of the robot, we use an RGB image for a cleaner comparison with Hand Inpaint. At test time, a hand mask

generated by a trained diffusion model is applied, and a virtual robot is overlaid on the real robot. See Appendix 6.8 for more details.

Red Line: EgoMimic [7] proposes a data-editing approach for learning robot policies from egocentric human videos. While we do not directly compare with their full method, as it requires robot data, we evaluate their data-editing strategy. The human arm is masked out in black during training and overlaid with a red line along its length. At test time, the robot arm is similarly blacked out, with a red line overlaid in the same manner.

Vanilla: We also compare to a baseline that does not modify the train or test images in any way.

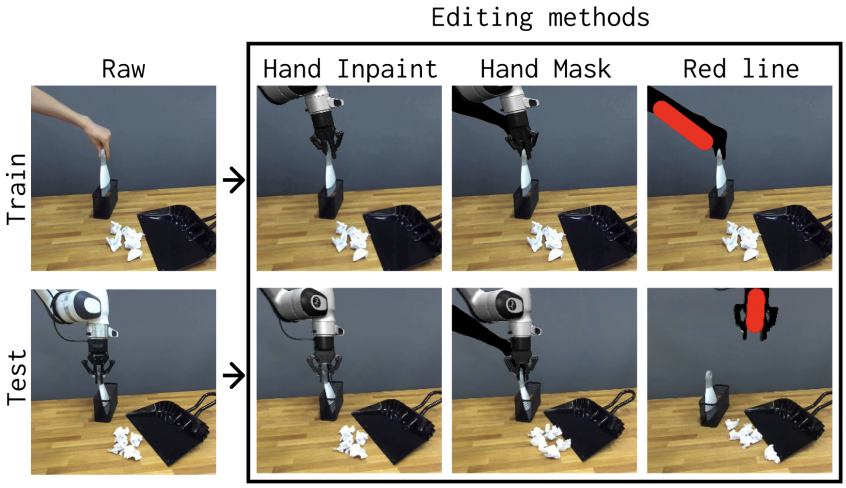

Figure 4: The three data-editing strategies we compare.

4.2 In-distribution Scene

We start by evaluating how well our method can transfer a policy trained exclusively on data-edited human video demonstrations collected in a single scene to a robot in the same scene. This evaluates how well our method bridges the physical and visual embodiment gap between human and robot without the added complexity of testing scene generalization. We evaluate our method on five tasks that highlight the diversity of skills our method can learn. For each task we collect between 250-350 human video demonstrations (see Appendix 6.5).

Hand Inpaint and Hand Mask achieve high success rates across all tasks. However, Hand Mask takes on average 73% longer to rollout due to having to run an additional diffusion model at test time to generate the hand masks. The Red Line data-editing strategy fails to complete any tasks, indicating that it does not adequately bridge the visual embodiment gap between humans and robots.

	Pick/ Place Book	Stack Cups	Tie Rope	Rotate Box	Grasp Brush	Sweep > 0	Sweep > 2	Sweep > 4
Hand Inpaint (Phantom)	0.92	0.72	0.64	0.72	0.88	0.80	0.72	0.40
Hand Mask	0.92	0.52	0.60	0.76	0.75	0.75	0.72	0.68
Red Line	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Vanilla	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Table 1: **In-distribution scene results**: Both Hand Inpaint and Hand Mask achieve high success rates across all tasks. The Red Line strategy fails to achieve success on any task, as does the Vanilla baseline. For the sweep task, we evaluate success at multiple levels of completion: Grasp Brush measures whether the robot successfully picks up the brush, while Sweep > 0, Sweep > 2, and Sweep > 4 indicate the number of pieces swept into the dustpan. 25 rollouts per evaluation.

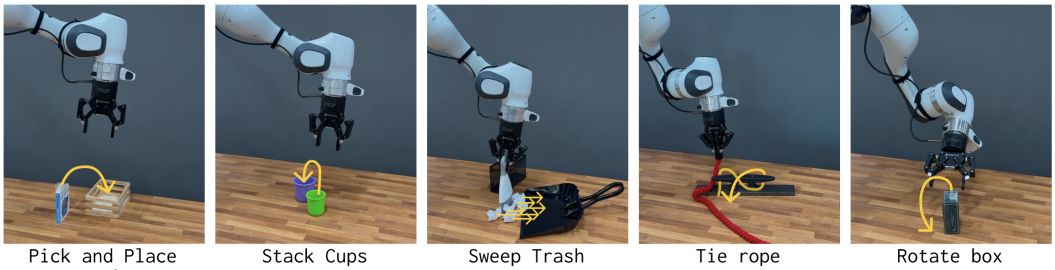

Figure 5: **Tasks evaluated in an in-distribution scene — Pick and Place Book**: The robot must pick up a book and place it inside a wooden container. **Stack Cups**: The robot must stack the green cup inside the purple cup. Precise alignment is critical, as the cups differ in diameter by only 1.5 cm. **Sweep Trash**: The robot must pick up a sweeper and sweep six pieces of trash into a dustpan. This task involves coordinated multi-object manipulation, requiring the robot to control the sweeper while simultaneously managing the movement of multiple loose objects. Additionally, the pieces of trash exhibit unpredictable dynamics, necessitating continuous adaptation based on real-time feedback. **Tie Rope**: The robot must tie a simplified cleat hitch, a sailing knot that follows a figure-eight ∞ pattern. This task is challenging due to the precise manipulation required of a highly deformable object. **Rotate Box**: The robot must rotate a box 90 degrees onto a new face in a controlled fashion (simply knocking it over is not valid).

4.3 Out-of-distribution Scenes

Next, we evaluate how well our method generalizes to new, unseen environments. We collect human video demonstrations of a sweeping task across diverse scenes (see Fig. 1 for examples). The robot must grasp the sweeper and sweep the green piece of trash off the surface. We collect 950 human video demonstrations across a wide range of indoor and outdoor scenes (see Appendix 6.5).

We assess generalization in three out-of-distribution (OOD) scenarios (see Fig 6) that were never seen during data collection. These consist of two OOD scenes — Outdoor Lawn and Indoor Lounge — and one OOD surface. Rollouts take place with dynamic background variations, including moving cars and passersby.

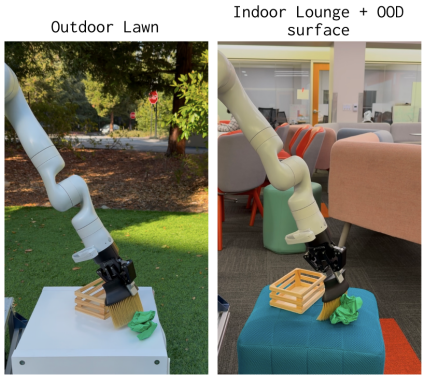

Figure 6: The out-of-distribution evaluation scenes used to evaluate the sweeping task on the Kinova robot.

Hand Inpaint achieves high success rates across all three OOD environments. Its best performance is in the indoor lounge, which aligns with expectations, as 80% of the training data was collected indoors. When evaluated on an unseen surface, performance drops by 20%, likely due to the limited diversity of training surfaces (four, see Appendix 6.5.)

Overall, Hand Inpaint and Hand Mask perform comparably across both in-distribution scenes and out-of-distribution scenes. Red Line and the Vanilla baseline fail on all tasks due to large visual mismatches between human and robot embodiments. While performance is comparable between Hand Inpaint and Hand Mask, Hand Inpaint is on average 73% faster to rollout, as it does not require hand mask generation with a diffusion model, and produces training images that more closely resemble real-world robot data, making it the more effective solution.

	Outdoor lawn	Indoor lounge	Indoor lounge + OOD surface
Hand Inpaint (Phantom)	0.72	0.84	0.64
Hand Mask	0.52	0.76	0.68

Table 2: **Out-of-distribution (OOD) scene results:** Success rates of policies trained on human video demos and tested in three unseen environments: Outdoor Lawn, Indoor Lounge, and Indoor Lounge with an OOD Surface. Hand Inpaint achieves the highest success rates across all settings. Hand Mask performs comparably but is worse in the outdoor lawn setting, perhaps due to weather variations during rollouts. 25 rollouts per eval.

4.4 Evaluating the Need for High-quality In-painting

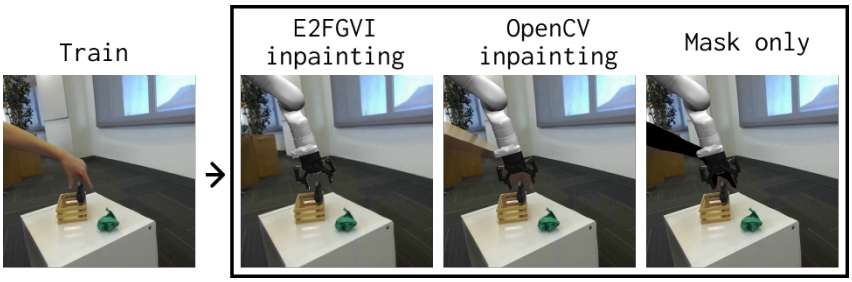

Figure 7: The three inpainting strategies we compare.

At train time, our preferred Hand Inpaint method relies on inpainting to remove the human arm from the human video and replace it with a realistic background. We test how important this inpainting is by comparing three variations (see Fig. 7) on the sweeping task in the unseen Indoor Lounge scene: (1) High-quality inpainting (E2FGVI [46]): A state-of-the-art video inpainting method, (2) Low-quality inpainting (OpenCV inpaint): The OpenCV inpainting function removes the arm but leaves visible artifacts, (3) No-inpainting (Mask Only): The human arm is simply masked out during training. Unlike Hand Mask, no hand mask is overlaid at test time.

High-quality inpainting with E2FGVI yields the best performance, achieving a 84% success rate. However, low-quality inpainting performs surprisingly well, with a 76% success rate. This suggests that there is enough variation in the low-quality inpainting at train time for the model to become agnostic to the artifacts. In contrast, using no-inpainting noticeably degrades performance. The mask-only approach results in a 24 percentage point performance drop. These results suggest that while high-quality inpainting is ideal, our method still performs well with primitive inpainting. Additionally, the high performance of the mask-only approach relative to the 0% success rate of the Red Line method in the easier in-distribution experiments implies that including an overlay of the robot at training time is essential.

	Indoor Lounge
E2FGVI inpaint	0.84
OpenCV inpaint	0.76
Mask only	0.60

Table 3: Comparison of in-painting methods

5 Conclusion

We present a method for training robot policies without collecting any robot data, using only human video demonstrations. Our approach successfully transfers policies across a diverse set of tasks, including deformable object manipulation and multiple objects manipulation. Furthermore, we demonstrate zero-shot deployment in novel scenes, showing that human video demonstrations and robot rollouts do not need to occur in the same environment. This flexibility makes our method highly scalable and accessible. By enabling anyone with an RGBD camera to collect meaningful training data anywhere, we lower the barrier to large-scale robot learning and broaden the potential for real-world deployment.

Lastly, a promising direction in robotics is training autoregressive generalist policies on large-scale robotics datasets [1, 17, 2]. Other interesting methods using human video demonstrations introduce complexities that are not amenable to such architectures [40, 8, 30, 37]. In contrast, our simple data-editing approach generates observation-action pairs of robots performing tasks, which can be easily integrated into datasets used to train these generalist policies — a promising avenue for future work.

6 Limitations

Our approach has several limitations. First, its performance is limited by the performance of existing hand pose estimators since it relies on them to obtain the target actions from a demonstration video. Because hand pose estimators currently still struggle with occlusions, our method does too. However, this also means that our method will get better as hand pose estimators improve. Second, our method only works when the robot can follow the same strategy as the human to complete the task. As a result, our policy may lead the robot to collide with the environment even though the human hand does not. Additionally, differences in the surface properties of a human fingertip and robot gripper may lead to different object motions. Third, we limit our demonstrations to pinch grasps because our robots are limited to using parallel jaw grippers - a limitation that is shared with virtually all large-scale robot data collection efforts [43, 1, 2]. Fourth, we only assess quasi-static tasks, as we do not address the latency mismatch between a human demonstration and a trained policy rolled out on real hardware.

Acknowledgments

This work was supported by the NSF through grant number #2327974 as well as Intrinsic. We thank Jimmy Wu for help with hardware, and Claire Chen, Priya Sundaresan, Juntao Ren, and Zi-ang Cao for their valuable help throughout the project.

References

[1] A. O. e. al. Open x-embodiment: Robotic learning datasets and rt-x models. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6892–6903, 2024. doi: 10.1109/ICRA57147.2024.10611477.

[2] K. Black, N. Brown, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, L. Groom, K. Hausman, B. Ichter, S. Jakubczak, T. Jones, L. Ke, S. Levine, A. Li-Bell, M. Mothukuri, S. Nair, K. Pertsch, L. X. Shi, J. Tanner, Q. Vuong, A. Walling, H. Wang, and U. Zhilinsky. π_0: A vision-language-action flow model for general robot control, 2024. URL https://arxiv.org/abs/2410.24164.

[3] G. R. Team, S. Abeyruwan, J. Ainslie, J.-B. Alayrac, M. G. Arenas, T. Armstrong, A. Balakrishna, R. Baruch, M. Bauza, M. Blokzijl, S. Bohez, K. Bousmalis, A. Brohan, T. Buschmann, A. Byravan, S. Cabi, K. Caluwaerts, F. Casarini, O. Chang, J. E. Chen, X. Chen, H.-T. L. Chiang, K. Choromanski, D. D'Ambrosio, S. Dasari, T. Davchev, C. Devin, N. D. Palo, T. Ding, A. Dostmohamed, D. Driess, Y. Du, D. Dwibedi, M. Elabd, C. Fantacci, C. Fong, E. Frey, C. Fu, M. Giustina, K. Gopalakrishnan, L. Graesser, L. Hasenclever, N. Heess, B. Hernaez, A. Herzog, R. A. Hofer, J. Humplik, A. Iscen, M. G. Jacob, D. Jain, R. Julian, D. Kalashnikov, M. E. Karagozler, S. Karp, C. Kew, J. Kirkland, S. Kirmani, Y. Kuang, T. Lampe, A. Laurens, I. Leal, A. X. Lee, T.-W. E. Lee, J. Liang, Y. Lin, S. Maddineni, A. Majumdar, A. H. Michaely, R. Moreno, M. Neunert, F. Nori, C. Parada, E. Parisotto, P. Pastor, A. Pooley, K. Rao, K. Reymann, D. Sadigh, S. Saliceti, P. Sanketi, P. Sermanet, D. Shah, M. Sharma, K. Shea, C. Shu, V. Sindhwani, S. Singh, R. Soricut, J. T. Springenberg, R. Sterneck, R. Surdulescu, J. Tan, J. Tompson, V. Vanhoucke, J. Varley, G. Vesom, G. Vezzani, O. Vinyals, A. Wahid, S. Welker, P. Wohlhart, F. Xia, T. Xiao, A. Xie, J. Xie, P. Xu, S. Xu, Y. Xu, Z. Xu, Y. Yang, R. Yao, S. Yaroshenko, W. Yu, W. Yuan, J. Zhang, T. Zhang, A. Zhou, and Y. Zhou. Gemini robotics: Bringing ai into the physical world, 2025. URL https://arxiv.org/abs/2503.20020.

[4] J. Gao, A. Xie, T. Xiao, C. Finn, and D. Sadigh. Efficient data collection for robotic manipulation via compositional generalization. *arXiv preprint arXiv:2403.05110*, 2024.

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

[6] H. Bharadhwaj, R. Mottaghi, A. Gupta, and S. Tulsiani. Track2act: Predicting point tracks from internet videos enables generalizable robot manipulation. In *European Conference on Computer Vision*, pages 306–324. Springer, 2025.

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

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

[9] V. Jain, M. Attarian, N. J. Joshi, A. Wahid, D. Driess, Q. Vuong, P. R. Sanketi, P. Sermanet, S. Welker, C. Chan, et al. Vid2robot: End-to-end video-conditioned policy learning with cross-attention transformers. *arXiv preprint arXiv:2403.12943*, 2024.

[10] K. Pertsch, R. Desai, V. Kumar, F. Meier, J. J. Lim, D. Batra, and A. Rai. Cross-domain transfer via semantic skill imitation, 2022. URL https://arxiv.org/abs/2212.07407.

[11] 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.

[12] A. S. Chen, S. Nair, and C. Finn. Learning generalizable robotic reward functions from" in-the-wild" human videos. *arXiv preprint arXiv:2103.16817*, 2021.

[13] L. Shao, T. Migimatsu, Q. Zhang, K. Yang, and J. Bohg. Concept2robot: Learning manipulation concepts from instructions and human demonstrations. In *Proceedings of Robotics: Science and Systems (RSS)*, 2020.

[14] L. Y. Chen, K. Hari, K. Dharmarajan, C. Xu, Q. Vuong, and K. Goldberg. Mirage: Cross-embodiment zero-shot policy transfer with cross-painting. *arXiv preprint arXiv:2402.19249*, 2024.

[15] L. Y. Chen, C. Xu, K. Dharmarajan, M. Z. Irshad, R. Cheng, K. Keutzer, M. Tomizuka, Q. Vuong, and K. Goldberg. Rovi-aug: Robot and viewpoint augmentation for cross-embodiment robot learning. In *Conference on Robot Learning (CoRL)*, Munich, Germany, 2024.

[16] M. Lepert, R. Doshi, and J. Bohg. Shadow: Leveraging segmentation masks for cross-embodiment policy transfer. In *8th Annual Conference on Robot Learning*.

[17] M. J. Kim, K. Pertsch, S. Karamcheti, T. Xiao, A. Balakrishna, S. Nair, R. Rafailov, E. Foster, G. Lam, P. Sanketi, et al. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.

[18] 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*, pages 416–426. PMLR, 2023.

[19] A. Xie, L. Lee, T. Xiao, and C. Finn. Decomposing the generalization gap in imitation learning for visual robotic manipulation. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3153–3160. IEEE, 2024.

[20] S. Nair, A. Rajeswaran, V. Kumar, C. Finn, and A. Gupta. R3m: A universal visual representation for robot manipulation. *arXiv preprint arXiv:2203.12601*, 2022.

[21] 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. *arXiv preprint arXiv:2210.00030*, 2022.

[22] M. Yang, Y. Du, K. Ghasemipour, J. Tompson, D. Schuurmans, and P. Abbeel. Learning interactive real-world simulators. *arXiv preprint arXiv:2310.06114*, 1(2):6, 2023.

[23] Y. Du, M. Yang, B. Dai, H. Dai, O. Nachum, J. B. Tenenbaum, D. Schuurmans, and P. Abbeel. Learning universal policies via text-guided video generation. *arXiv e-prints*, pages arXiv–2302, 2023.

[24] J. Liang, R. Liu, E. Ozguroglu, S. Sudhakar, A. Dave, P. Tokmakov, S. Song, and C. Vondrick. Dreamitate: Real-world visuomotor policy learning via video generation. *arXiv preprint arXiv:2406.16862*, 2024.

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

[26] C.-L. Cheang, G. Chen, Y. Jing, T. Kong, H. Li, Y. Li, Y. Liu, H. Wu, J. Xu, Y. Yang, et al. Gr-2: A generative video-language-action model with web-scale knowledge for robot manipulation. *arXiv preprint arXiv:2410.06158*, 2024.

[27] S. Bahl, R. Mendonca, L. Chen, U. Jain, and D. Pathak. Affordances from human videos as a versatile representation for robotics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13778–13790, 2023.

[28] B. Wang, J. Zhang, S. Dong, I. Fang, and C. Feng. Vlm see, robot do: Human demo video to robot action plan via vision language model. *arXiv preprint arXiv:2410.08792*, 2024.

[29] J. Shi, Z. Zhao, T. Wang, I. Pedroza, A. Luo, J. Wang, J. Ma, and D. Jayaraman. Zeromimic: Distilling robotic manipulation skills from web videos, 2025. URL https://arxiv.org/abs/2503.23877.

[30] S. Bahl, A. Gupta, and D. Pathak. Human-to-robot imitation in the wild. *arXiv preprint arXiv:2207.09450*, 2022.

[31] 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.

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

[33] 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*, pages 991–1002. PMLR, 2022.

[34] C.-C. Hsu, B. Wen, J. Xu, Y. Narang, X. Wang, Y. Zhu, J. Biswas, and S. Birchfield. Spot: Se (3) pose trajectory diffusion for object-centric manipulation. *arXiv preprint arXiv:2411.00965*, 2024.

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

[36] A. Bahety, P. Mandikal, B. Abbatematteo, and R. Martín-Martín. Screwmimic: Bimanual imitation from human videos with screw space projection. *arXiv preprint arXiv:2405.03666*, 2024.

[37] Y. Zhu, A. Lim, P. Stone, and Y. Zhu. Vision-based manipulation from single human video with open-world object graphs. *arXiv preprint arXiv:2405.20321*, 2024.

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

[39] 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.

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

[41] S. Haldar and L. Pinto. Point policy: Unifying observations and actions with key points for robot manipulation. *arXiv preprint arXiv:2502.20391*, 2025.

[42] J. Duan, Y. R. Wang, M. Shridhar, D. Fox, and R. Krishna. Ar2-d2: Training a robot without a robot. *arXiv preprint arXiv:2306.13818*, 2023.

[43] A. Khazatsky, K. Pertsch, S. Nair, A. Balakrishna, S. Dasari, S. Karamcheti, S. Nasiriany, M. K. Srirama, L. Y. Chen, K. Ellis, et al. Droid: A large-scale in-the-wild robot manipulation dataset. *arXiv preprint arXiv:2403.12945*, 2024.

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

[45] N. Ravi, V. Gabeur, Y.-T. Hu, R. Hu, C. Ryali, T. Ma, H. Khedr, R. Rädle, C. Rolland, L. Gustafson, E. Mintun, J. Pan, K. V. Alwala, N. Carion, C.-Y. Wu, R. Girshick, P. Dollár, and C. Feichtenhofer. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024. URL https://arxiv.org/abs/2408.00714.

[46] 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.

[47] C. Chi, Z. Xu, S. Feng, E. Cousineau, Y. Du, B. Burchfiel, R. Tedrake, and S. Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, page 02783649241273668, 2023.

[48] E. Todorov, T. Erez, and Y. Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5026–5033. IEEE, 2012. doi:10.1109/IROS.2012.6386109.

[49] K. Zakka, Y. Tassa, and MuJoCo Menagerie Contributors. MuJoCo Menagerie: A collection of high-quality simulation models for MuJoCo, 2022. URL http://github.com/google-deepmind/mujoco_menagerie.

[50] B. Lim, S. Son, H. Kim, S. Nah, and K. M. Lee. Enhanced deep residual networks for single image super-resolution. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, July 2017.

[51] N. Tumanyan, M. Geyer, S. Bagon, and T. Dekel. Plug-and-play diffusion features for text-driven image-to-image translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1921–1930, June 2023.

Appendix

6.1 Detailed justification for our baseline selection

In this section, we discuss the most relevant prior works and explain why we do not include direct comparisons to them in our experiments (with the exception of a partial comparison to EgoMimic). Table 4 summarizes our detailed comparison between Phantom and related works. Broadly, we identify two key reasons why existing methods are not suitable baselines for our approach:

1. **The robot data bottleneck**: Phantom is designed to explore how to build a highly scalable framework for learning from human videos. In contrast, many prior methods rely on co-training with teleoperated robot demonstrations (see Table. 4), which introduces a fundamental bottleneck to scalability. Collecting robot data is expensive, time-consuming, and difficult to scale across environments and tasks. These methods do not address the core challenge Phantom tackles: enabling true scalability by removing the need for robot data entirely.

   We acknowledge that large-scale efforts to collect robot demonstrations are ongoing, and such robot data will remain very useful for co-training with human data. However, many methods that claim to scale with human videos still require a substantial amount of robot data per task to work effectively (e.g., MotionTracks [5] requires at least 22% of its data to be teleoperated robot demonstrations; Track2Act [6] uses roughly 40 robot demos per task; EgoMimic [7] trains on datasets where, on average, 41% of the data comes from robot teleoperation). These methods fail in the target regime where human data is far more abundant than robot data ($\ll 1\%$ robot data).

   In contrast, Phantom is designed to function with zero robot data, and therefore is well-positioned to succeed even as human datasets scale rapidly and robot data remains limited. While we do not demonstrate large-scale deployment in this work, our experiments provide early evidence that Phantom can operate effectively under these future data conditions. Because existing methods rely on robot data, they are unable to address Phantom's central goal — true scalability through human videos — and are thus not suitable baselines for evaluating our approach.

2. **The object-centric bottleneck**: Many learning from human videos methods bridge the embodiment gap using an object centric approach. However, object tracking remains an ongoing, challenging research problem, especially for multi-object tracking, deformables, granular materials, and liquids. As a result, these methods fail on these types of objects. Phantom has no such constraints and has been shown to work with groups of objects and deformable objects. A fair comparison to Phantom should include works that are not specialized for one type of object.

EgoMimic [7] proposes a method to learn robot policies from human videos. EgoMimic and Phantom differ in two main ways:

- **Strategy to bridge visual embodiment gap**: EgoMimic attempts to bridge the visual embodiment gap using the Red Line overlay whereas Phantom uses a virtual robot overlay. Our experiments convincingly show that the virtual robot overlay works significantly better. EgoMimic only shows results for a much easier cross-embodiment setup where the robot arm is already closely aligned with the human arm, and it is possible that in this easier setup the EgoMimic data editing strategy would have more success than in our more challenging scenario.
- **Actions**: EgoMimic leverages human data only to improve predictions of the gripper's pose, and relies entirely on robot data to predict the width of the gripper. As a result, EgoMimic requires at least two hours of robot data collection before being able to cotrain with human data. A full comparison to EgoMimic would require adapting our Red Line overlay baseline to rely on robot data to determine the opening width of the gripper. Doing

so would defeat the purpose of evaluating how well our method transfers policies from only human videos to robots, since learning gripper opening widths from robot data does not achieve the goal of transferring policies exclusively from humans to robots.

MotionTracks [5] proposes a unified image-space action representation—2D motion tracks—for imitation learning from both human videos and robot demonstrations. However, all experiments include datasets with at least 22% teleoperated robot data. In contrast, our work focuses on building a highly scalable framework that learns exclusively from human videos, avoiding the teleoperation bottleneck entirely. By removing this bottleneck, Phantom tackles a harder problem with greater potential to scale.

Track2Act [6] learns 2D point tracks from large-scale web videos and lifts them to 3D object transformations to obtain robot actions, which are then refined by a residual closed-loop policy trained on teleoperated demonstrations. Despite leveraging 400,000 video clips, Track2Act still requires around 40 robot demos per task. This reliance on robot data remains a major bottleneck to scaling— the challenge Phantom aims to overcome. Phantom learns policies without any robot data, enabling scalable learning directly from human videos. Relying on co-training with robot data, as in Track2Act, does not address our core research objective.

Im2Flow2Act [40] uses object flow to transfer human video demonstrations to real-world robot manipulation without any real robot training data. It learns policies from simulated object flow, requiring a manually constructed simulation environment and task-specific heuristic actions. This involves substantial overhead, as humans must create simulation environments with relevant, similar objects and define heuristics for each task. While simulation-based transfer is a valuable direction, our work focuses on policy transfer without the manual effort of constructing simulation environments or hand-designing task heuristics.

R+X [38] automatically retrieves human videos that are relevant for a target task and executes a robot policy conditioned on keypoint trajectories from these human videos. However, R+X is only demonstrated to operate in an open-loop manner. In contrast, we enable closed-loop execution, which is critical for robustness—especially in tasks like Sweep Trash, where the robot must adapt to unpredictable object dynamics.

Orion [37] enables robots to learn manipulation tasks from a single human video in open-world settings using object-centric reasoning via Open-world Object Graphs (OOGs). It extracts keyframes and models object relations and motions to learn a policy. However, Orion relies on computing the desired SE(3) transformation of the target object to derive robot actions—a strategy that breaks down for deformable objects and multiple objects. In contrast, Phantom imposes no such constraint; for example, we demonstrate tying a figure-eight knot with a rope.

Vid2Robot [9] learns an end-to-end video-conditioned policy from paired videos of humans and robots performing the same task, enabling alignment across embodiments and environments. However, the need for such paired data significantly limits scalability, and therefore does not address our core research objective.

HOPMan [32] predicts future hand-object interaction plans from passive human videos using a diffusion model, and then translates these plans into robot actions via a policy trained on paired human-robot data. However, this approach required 3 days of data collection to obtain 600 paired demonstrations, plus an additional 1,000 robot-only demos to deploy across 16 skills. While promising, this heavy reliance on robot data remains a major bottleneck to scalability.

DITTO [35] is a two-stage framework that extracts object-centric trajectories by segmenting and tracking objects offline, then re-detects and re-localizes these objects at deployment to warp the demonstration trajectory accordingly. It executes the warped trajectory using an off-the-shelf grasp planner. However, DITTO relies on estimating the desired SE(3) motion of the target object, which fails for deformable objects and multiple objects—a limitation Phantom avoids entirely. Moreover, DITTO does not report quantitative results for its real-world experiments.

MimicPlay [8] is a hierarchical imitation learning framework that learns high-level task plans from unlabeled human play videos and uses them to guide low-level robot control trained on teleoperated robot demonstrations. EgoMimic [7] has already been shown to outperform MimicPlay, and we compare to EgoMimic to the extent possible. Additionally, MimicPlay is trained on a dataset in which at least 75% of the data consists of robot demonstrations, as measured by data collection time. MimicPlay's heavy reliance on robot data makes it unsuitable for comparison.

AR2-D2 [42] is a system for collecting robot manipulation demonstrations without requiring a physical robot, using an iOS app that overlays a virtual robot onto real-world scenes and tracks the user's hand to generate robot-aligned trajectories. However, this method requires that videos be collected with an iPhone/iPad and that the human video demonstrator manually annotate keypoint poses for the robot to follow live during data collection. This makes collecting human video demonstrations significantly slower and constrains the type of motions that can be demonstrated, as evidenced by the very limited tasks shown in the paper: pick (without placing), press, and push.

WHIRL [30] learns policies by extracting interaction priors from third-person human videos and refining them through real-world robot interactions using sampling-based policy optimization. However, its reliance on physical robot trials limits scalability. In contrast, Phantom directly produces high-quality actions from human videos without requiring any robot interaction beforehand.

Learning by Watching [11] uses an image-to-image translation network to map a single human video into the robot domain, extracts keypoints from the translated images, and builds a reward function to train a policy in simulation. However, this method is only demonstrated to work in a single scene in simulation, while our work focuses on real-world deployment.

	No Robot Data	Deformable Objects	Closed-loop
EgoMimic [7]	✗	✓	✓
MotionTracks [5]	✗	✓	✓
Track2Act [6]	✗	✗	✓
Im2Flow2Act [40]	✗*	✓	✓
R+x [38]	✓	✓	✗
ORION [37]	✓	✗	✓
Vid2Robot [9]	✗	✓	✓
HOPMan [32]	✗	✓	✓
DITTO [35]	✓	✗	✓
Mimicplay [8]	✗	✓	✓
AR2-D2 [42]	✗	✓	✓
WHIRL [30]	✗	✓	✓
Learning by watching [11]	✗*	✗	✓
Ours	✓	✓	✓

Table 4: Comparison between our method and other related works. **No Robot Data**: the method does not require robot data in policy training. ✗* indicates that the method relies on simulation data which is limited by the need to create simulation environments that are representative of real world interactions. **Deformable Objects**: the method is demonstrated to work on deformable objects. **Closed-loop**: the method is closed-loop.

6.2 Phantom is robot agnostic

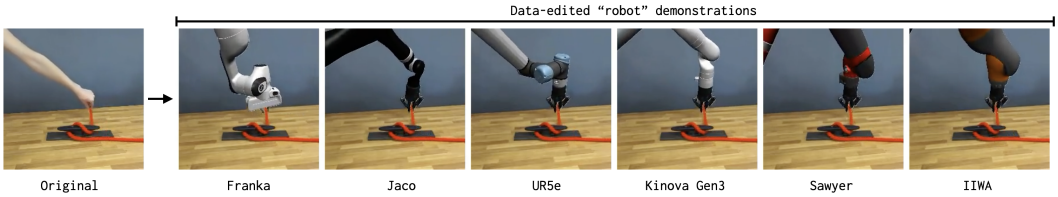

Figure 8: Our method is robot agnostic. Each human video can be converted into a robot demonstration for any robot capable of completing the task.

6.3 Evaluating the Benefits of Co-training with Diverse Human Data

While we have already shown in previous experiments that our policy can be deployed zero-shot without any robot data, we also investigate the benefits of co-training with robot data given that there already exists considerable amounts of robot data. To do this, we collect 100 teleoperated demonstrations in a single scene on the Kinova robot using an Oculus controller for the sweeping task. In each observation image from these demonstrations, we overlay a virtual Kinova on the real robot to align the visual distributions between our datasets. Next we co-train our robot data with our larger scale human video dataset consisting of 950 demonstrations in many different scenes. We see that while the robot-only policy performs well in-distribution, its success rate drops to zero in a new scene. Co-training with human videos from diverse scenes, however, increases the performance to 80%.

	In-distribution Scene	Out-of-distribution Scene
Robot	0.88	0.0
Robot + Human	—	0.80

Table 5: **Co-training with diverse human data**. Evaluating the benefits of co-training with diverse human data. 25 rollouts per evaluation.

6.4 Comparing Human vs. Robot Data

While our approach significantly reduces the cost of scaling data collection across diverse environments, it introduces a tradeoff: human video demonstrations provide scalability at the expense of some precision, due to uncertainty in hand pose estimation from RGBD videos. To investigate how much precision is lost, we compare policies trained on teleoperated robot demonstrations (collected using an Oculus controller) against those trained on human video demonstrations for the Kinova sweeping task. All data is collected and evaluated in the same scene.

As shown in Table 6, a policy trained on 50 teleoperated demonstrations achieves a 52% success rate, compared to 44% for 50 human demonstrations—a modest drop that is not statistically significant ($p = 0.778$, where significance is defined as $p < 0.05$). At 100 demonstrations, the gap widens (88% vs. 64%), but still falls short of statistical significance ($p = 0.095$). Scaling to 300 human demonstrations improves success to 84%. These results suggest that while individual human demonstrations may be less precise, their ease of collection allows for scaling that effectively closes the performance gap.

# demos	Robot only	Human only	p-value (Fisher)
50	0.52	0.44	0.778
100	0.88	0.64	0.095
300	—	0.84	—

Table 6: **Performance of policies trained on robot vs. human demonstrations.** Each configuration was evaluated with 25 rollouts. P-values (Fisher's exact test) assess whether success rates differ significantly ($p < 0.05$) between robot and human data. While human only policies exhibit a drop in performance, no statistically significant differences were found. The robot-only condition for 300 demos was not trained due to the significant time requirement and because it would not meaningfully add to this analysis.

Importantly, this single scene comparison understates the strength of our approach. **The central value of human video data is its scalability across environments.** Collecting robot data in many scenes requires moving physical hardware—an expensive and time-consuming process—whereas human video data can be gathered quickly and at scale with minimal overhead. A full comparison should also compare the time required to collect diverse human data across many scenes and the time required to collect equivalent robot data across those same scenes. However, we do not perform this experiment because robot data collection in many scenes is prohibitively slow.

6.5 Data collection Details

The details of the human video demonstration datasets collected for all tasks are outlined in Table 7.

	Task	Number of demos	Max Horizon
Panda tasks	Zebra	313	228
	Stack	268	148
	Sweep	279	407
	Cleat	307	516
	Soap	283	175
Kinova task	Sweep	950	168

Table 7: Human video datasets. **Number of demos**: number of human video demonstrations used in training, **Max Horizon**: the maximum number of steps in the human video demonstration dataset.

Figure 9: Left: surfaces used during human video data collection for the kinova sweep task. Right: Unseen surface used to evaluate policy.

6.6 Policy Training Details

Table 8 describes the hyperparameters used to train diffusion policy. All experiments done on the Kinova robot use the same set of hyperparameters. This includes the diverse scene experiments,

inpainting quality experiments, and robot vs. human video data comparison experiments. We used the same set of hyperparameters for all data-editing methods across each task. All tasks in the paper use the DDIM scheduler with 100 training steps and 10 inference steps.

Robot	Task	To	Ta	ImgRes	Batch	Lr	Aug
Panda	Zebra	2	8	240	200	1e-4	Yes
	Stack	2	8	240	200	1e-4	Yes
	Sweep	2	8	240	200	1e-4	Yes
	Cleat	2	8	240	200	1e-4	Yes*
	Soap	2	8	240	200	1e-4	Yes
Kinova	Sweep	2	8	240	256	1e-4	Yes

Table 8: Hyperparameters for Diffusion policy training. **To**: observation horizon, **Ta**: action horizon. **Aug**: Image augmentations (RandomCrop, RandomRotation, ColorJitter) used during training. Only ColorJitter was used for the Cleat task to avoid cropping out grasps of the rope that frequently occur on the edge of the image.

All virtual robot overlays are generated using models from Mujoco Menagerie [49].

6.7 Detailed Task Descriptions

The variation in the placement of objects for each task is visualized in Fig. 10.

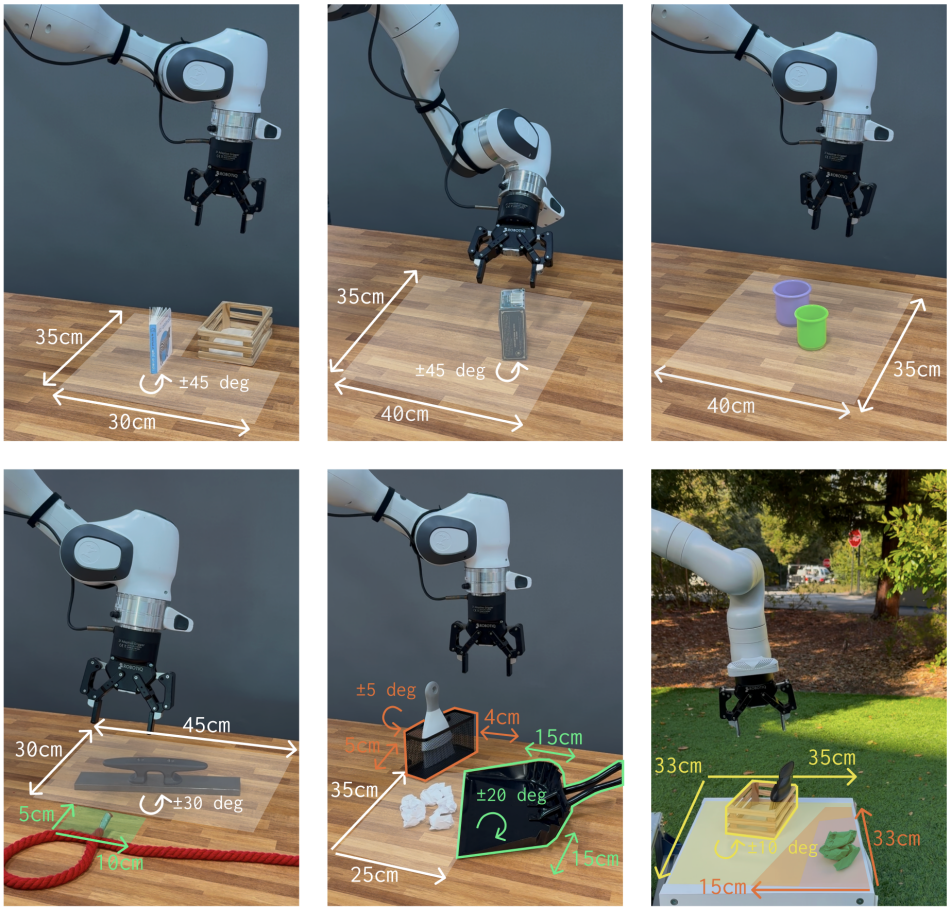

Figure 10: Variation in object placement during evaluations of each task.

Pick and Place Book: The robot must pick up a book and place it inside a wooden container. The book's initial position is randomly sampled within the outlined 30 cm × 35 cm rectangular region. Additionally, its orientation can vary by ±45 degrees.

Rotate Box: The robot must rotate a box 90 degrees onto a new face in a controlled fashion (simply knocking it over is not valid). The box's initial position is randomly sampled within the outlined 40 cm x 35 cm rectangular region.

Stack Cups: The robot must stack the green cup inside the purple cup. Precise alignment is critical, as the cups differ in diameter by only 1.5 cm. The cups' initial positions are randomly sampled within the outlined 40 cm x 35 cm rectangular region.

Tie Rope: The robot must tie a simplified cleat hitch, a sailing knot that follows a figure-eight ∞ pattern. This task is challenging due to the precise manipulation required of a highly deformable object. The position of the cleat is randomly sampled within the outlined 30cm x 45cm white rectangular region and rotated by ±30 degrees.

Franka Sweep Trash: The robot must pick up the sweeper and sweep six pieces of trash into a dustpan. This task involves coordinated multi-object manipulation, requiring the robot to control the sweeper while simultaneously managing the movement of multiple loose objects. Additionally, the pieces of trash exhibit unpredictable dynamics, necessitating continuous adaptation based on real-time feedback. The position of the six pieces of trash is randomly sampled within the 25cm x 35cm rectangular region outlined in white. The position of the sweeper is randomly varied by 5cm x 4cm laterally and rotated by ±5 degrees. The position of the dustpan is randomly varied by 15cm x 15cm laterally and rotated by ±20 degrees.

Kinova Sweep Trash: The robot must pick up the sweeper and sweep the green piece of trash off the table. The position of the sweeper is randomly sampled within the 33cm x 35cm rectangular region outlined in yellow. The position of the green piece of trash is randomly sampled within the 15cm x 33cm region outlined in orange.

6.8 Hand Mask Data Editing Method

We describe in detail how we adapt the data-editing strategy from Shadow [16] to the human-to-robot setting.

Data-editing during training: We segment out the pixels corresponding to the hand and set them to black. We extract the hand pose using the strategy described in Section 3.2 and overlay an RGB rendering of the target robot in this pose using the known camera extrinsics.

Data-editing at inference time: To ensure that the train and test time images match closely, we overlay a black segmentation mask of the human arm and hand in the same pose as the robot. However, unlike in Shadow where the authors transferred policies between two robots, we do not have access to a realistic virtual model of a human arm and hand. Instead we train a diffusion model to predict the segmentation mask of the hand given a 6-DOF pose. This diffusion model is trained from scratch using the hand masks and corresponding target poses from our training dataset. At test time, we overlay the segmentation mask predicted by the diffusion model onto our image, and feed this edited image into our policy.

6.8.1 Training the Hand Mask Diffusion Model

We train the diffusion model for generating hand masks using the hyperparameters in Table 9. Due to high compute requirements, we generate 64×64 images and upscale them to the desired resolution using a super-resolution model [50]. To improve temporal consistency, the Hand Mask Diffusion Model generates hand masks for both time t and $t+1$ given a robot pose at time t. Additionally, we apply attention injection [51] at test time, using the model's output at time t as the reference image for time $t+1$.

Forward Diffusion Timesteps	Sampling Steps	ImgRes	Batch	Lr
1000	50	64	32	1e-4

Table 9: Hyperparameters used to train the Hand Mask diffusion model. **Forward Diffusion Timesteps**: the number of forward process steps at train time. **Sampling Steps**: the number of sampling steps used during inference.

6.8.2 Data Augmentation

Because the angle of the human arm can vary for the same 6-DOF pose, we need to make our policy agnostic to this variation. In addition, we find that the diffusion model we use to render the human segmentation mask does not output temporally consistent angles of the arm across frames. To address this problem, we augment our training data to include a second segmentation mask of the arm that is randomly shifted by a few pixels at each timestep. Importantly, the segmentation mask of the robot is not shifted, ensuring that the model can rely on the segmentation mask of the robot to localize the embodiment without relying on the hand mask.

We evaluate the impact of this augmentation on the Pick and Place Book and Stack Cups tasks. As shown in Table 10, incorporating the augmented hand mask achieves comparable performance in the easier Pick and Place Book task, but increases the success rate for the harder Stack Cups task by 12 percentage points. This suggests that the augmentation helps mitigate inconsistencies in the human arm mask generation, leading to more robust policy learning.

	Pick/ Place Book	Stack Cups
Hand Mask	0.92	0.52
Hand Mask (no data aug)	0.88	0.40

Table 10: **Effect of data augmentation on the Hand Mask strategy.** Adding random shifts to the hand mask during training improves performance in Stack Cups, where success increases from 40% to 52%. This suggests that the augmentation helps the policy generalize despite inconsistencies in the human arm segmentation. 25 rollouts per evaluation.