DemoDiffusion: One-Shot Human Imitation using pre-trained Diffusion Policy

Anonymous Authors

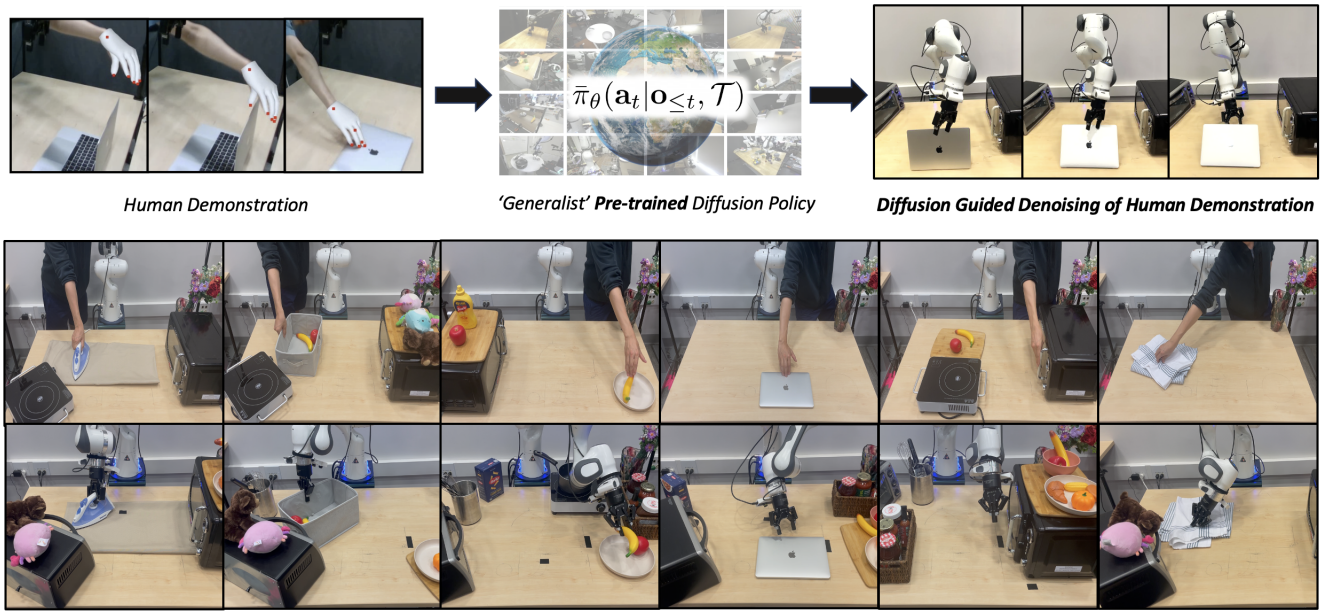

Fig. 1: **Overview of *DemoDiffusion*.** We show how *generalist* pre-trained diffusion policies can be used for following a generic human demonstration showing a manipulation task during deployment. Our real-world manipulation results encompass a wide diversity of manipulation tasks involving everyday objects.

***Abstract*— We propose *DemoDiffusion*, a simple method for enabling robots to perform manipulation tasks by imitating a single human demonstration, without requiring task-specific training or paired human-robot data. Our approach is based on two insights. First, the hand motion in a human demonstration provides a useful prior for the robot's end-effector trajectory, which we can convert into a rough open-loop robot motion trajectory via kinematic retargeting. Second, while this retargeted motion captures the overall structure of the task, it may not align well with plausible robot actions in-context. To address this, we leverage a pre-trained *generalist* diffusion policy to modify the trajectory, ensuring it both follows the human motion and remains within the distribution of plausible robot actions. Unlike approaches based on online reinforcement learning or paired human-robot data, our method enables robust adaptation to new tasks and scenes with minimal effort. In real-world experiments across 8 diverse manipulation tasks, DemoDiffusion achieves 83.8% average success rate, compared to 13.8% for the pre-trained policy and 52.5% for kinematic retargeting, succeeding even on tasks where the pre-trained *generalist* policy fails entirely. Project page: https://demodiffusion-anonymous.github.io/**

I. Introduction

How do we build robot manipulation systems that can be readily deployed in unstructured human environments? One possible answer is to learn 'generalist' policies that are capable of accomplishing any generic task (specified via some description *e.g.* language or image goal) in any environment. Indeed, there is a general optimism about this paradigm, reflected in several ongoing efforts to collect large-scale demonstration datasets to train such policies. While these efforts have produced unified policies capable of diverse tasks [1]–[5], these policies still struggle to perform meaningfully when deployed zero-shot to novel environments or asked to perform unseen tasks. Deployment thus often requires additional fine-tuning using task-and-scene-specific robot demonstrations [1], but this is a non-trivial overhead as collecting robot demonstrations in the real world can be time-consuming and beyond the expertise of an average user. In this work, we propose an alternate deployment mechanism – leveraging a pre-trained generalist policy to perform a task via imitating a single *human* demonstration.

We are of course not the first to consider this goal of allowing robots to imitate any given human demonstration. One common approach [6]–[9] is to instantiate this as a kinematic retargeting task and compute open-loop robot actions that maximize a manually defined similarity between the achieved robot end-effector configuration and observed human hand poses (*e.g.* matching locations of fingertips). However, the human-robot embodiment mismatch makes it difficult for the retargeted actions to achieve precisely the same effects as the human ones, and the open-loop execution further makes the approach brittle to noise and scene variations. Another line of work for human imitation attempts to learn robot policies through online reinforcement learning [10]–[14], where the human demonstration helps define reward functions. While this can overcome the embodiment gap, such test-time online RL requires hours of online interaction and resets, making it difficult to adopt for generic real-world manipulation tasks, especially in safety-critical scenarios. We seek an approach

that, like kinematic retargeting, can be deployed one-shot (*i.e.* without test-time training), but still benefits from learned priors for precise closed-loop interaction.

Our approach builds on the insight that pre-trained diffusion policies can act as priors for robot action. Inspired by prior work in leveraging pre-trained diffusion models for image editing [15], we present *DemoDiffusion* – a formulation to utilize a diffusion policy trained on robot interaction data for synthesizing coherent robot actions from a human demonstration. Specifically, we first perform kinematic retargeting by extracting human hand poses from the demonstration and obtain an open-loop robot action trajectory. While this trajectory is typically suboptimal due to embodiment differences and lack of closed-loop feedback, it serves as an effective initialization that can be improved via a diffusion policy. We do so by injecting Gaussian noise and applying the pre-trained diffusion policy to iteratively denoise the trajectory conditioned on robot observations, yielding a refined, executable sequence of robot actions. *DemoDiffusion* thus enables the robot to use the pre-trained policy as a prior and adapt the human-derived trajectory to its own embodiment and environment in a closed-loop manner.

In summary, we propose *DemoDiffusion*: a framework for robotic manipulation that allows a robot to perform generic tasks in natural environments by following a human demonstration with guidance from a pre-trained diffusion policy. Importantly, our framework doesn't require any robot demonstration of the target task within the target environment, online interaction, or fine-tuning. Our experiments across simulation and real-world environments show that *DemoDiffusion* surpasses both the base policy and retargeted trajectory, achieving 83.8% success across 8 real-world tasks – even those where the pre-trained *generalist* policy fails entirely.

II. Method

We target the problem of one-shot visual imitation. Given a human demonstration $\mathbf{D}$ depicting a manipulation task with description $\mathscr{T}$, we want a robot manipulator to perform the same task. Unlike prior work that relies on test-time RL training [10] or paired human-robot datasets [16], we wish to enable such imitation 'one-shot' assuming access to a closed-loop diffusion policy $\bar{\pi}_\theta(\mathbf{a}_t|\mathbf{o}_{\leq t}, \mathscr{T})$ pre-trained on some broad robot interaction dataset $\mathscr{D}_{robot}$. Our assumption on the human demonstration $\mathbf{D}$ is also very minimal: it can either be an RGBD video or a multi-view video of a human doing a task, such that 3D hand poses per-timestep can be reliably extracted from it.

A. Overview

Our approach is based on two insights. The first is that the trajectory of the hand pose in the human demonstration $\mathbf{D}$ provides useful information of the approximate trajectory the robot end-effector should follow, and we can perform *kinematic retargeting* of the hand trajectory $\{\mathbf{h}_t\}_{t=0}^T$ to an open-loop robot end-effector trajectory $\{\hat{\mathbf{a}}_t\}_{t=0}^T$. The second insight is that the kinematically retargeted robot trajectory has the correct form of motion, but these actions may not be very precise in the distribution of plausible robot actions given the current observation. A diffusion policy models this likelihood, and we can use a pre-trained '*generalist*' diffusion policy $\bar{\pi}_\theta(\mathbf{a}_t|\mathbf{o}_{\leq t}, \mathscr{T})$ to refine the retargeted robot actions in a closed loop manner, thus inferring actions that, while still similar to the human demonstration, are more likely under the policy and better aligned with the robot embodiment (see Fig. 2). We describe both these steps in detail below.

B. Kinematic retargeting of Human Hand Trajectories to Robot End-Effector Poses

Given a human demonstration $\mathbf{D}$ of a manipulation task $\mathscr{T}$, our first step is to extract the 3D hand pose trajectory $\{\mathbf{h}_t\}_{t=0}^T$. Each hand pose $\mathbf{h}_t \in \mathbb{R}^{3\times J}$ corresponds to the 3D locations of J keypoints (e.g., wrist and fingertips), which we estimate using a pre-trained monocular hand pose estimator applied to each video frame [17]. These keypoints encode the motion of the human performing the task and form the basis for retargeting to a robotic end-effector.

To translate this human motion into a robot-executable trajectory, we define a simple geometric mapping function $f_{\text{retarget}} : \mathbb{R}^{3\times J} \to \mathbb{R}^6$ that converts the human hand pose $\mathbf{h}_t$ into a robot configuration $\hat{\mathbf{a}}_t = f_{\text{retarget}}(\mathbf{h}_t)$. The mapping aligns the wrist pose of the human to the robot end-effector pose. For a two-finger gripper, we use the distance between the thumb and the remaining fingers of the hand mesh to infer a binary robot grasp, and for a dexterous robotic hand we match the robot hand's fingertip positions to those of a human hand using inverse kinematics. We set the initial configuration of the robot as the kinematically retargeted initial configuration from the human demonstration.

This kinematic retargeting procedure yields a full trajectory $\{\hat{\mathbf{a}}_t\}_{t=0}^T$ in the robot's configuration space, which can be executed as an open-loop policy. However, due to differences in morphology and embodiment between humans and robots, the absence of environment feedback, and the inaccuracy of the hand estimation module, this trajectory often leads to suboptimal or unstable behavior during open-loop execution. We thus treat this as the basis for searching for a plausible robot action trajectory via denoising with a (pre-trained) closed-loop diffusion policy.

C. Closed-Loop Denoising of Robot Actions with a Pre-Trained Diffusion Policy

A diffusion policy $\bar{\pi}_\theta(\mathbf{a}_t|\mathbf{o}_{\leq t}, \mathscr{T})$ [18], [19] pre-trained on diverse offline robot interaction data reliably models the distribution of plausible robot actions $\mathbf{a}_t$ given previous observations $\mathbf{o}_{\leq t}$. To predict actions $\mathbf{a}_t$, diffusion policies start with Gaussian noise $\tilde{\mathbf{a}}_t^{(S)} \sim \mathbf{N}(0, I)$ and iteratively denoise it at different diffusion steps s:

$$\tilde{\mathbf{a}}_t^{(s-1)} = \bar{\pi}_\theta\left(\tilde{\mathbf{a}}_t^{(s)}, \mathbf{o}_{\leq t}\right), \quad s = S, S-1 \ldots, 0 \tag{1}$$

We use this to modify the kinematically retargeted trajectory $\{\hat{\mathbf{a}}_t\}_{t=0}^T$ for obtaining robot actions that follow the high-level motion in the human trajectory and still lie within the distribution of plausible actions under $\bar{\pi}_\theta$. To do this, we

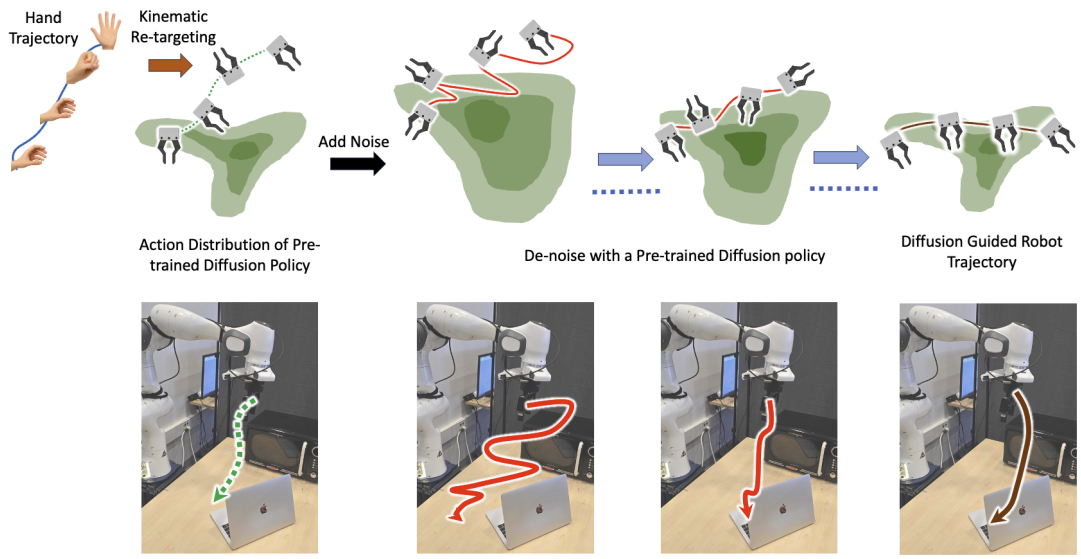

Fig. 2: **Retargeted human hand trajectory to closed-loop robot action sequence, for the task $\mathscr{T}$: *"shut down the laptop"***. The dotted line shows the trajectory of robot end-effector poses after kinematic retargeting. The olive contour plot depicts the distribution of trajectories from a pre-trained diffusion policy. Given a kinematic retargeting, we first perturb it with Gaussian noise and progressively remove the noise by simulating the reverse SDE with the diffusion policy. This process gradually projects a potentially unfeasible but approximately correct retargeting to the manifold of plausible robot actions that can perform real-world manipulation, in this case closing the laptop without missing the edge.

modify this typical reverse diffusion process so that instead of starting with pure noise $\tilde{\mathbf{a}}_t^{(S)}$ at step S, we start from an intermediate diffusion step s^* such that $0 < s^* < S$. We define $\{\tilde{\mathbf{a}}_t^{(s^*)}\}_{t=0}^T$ to be a noisy version of the kinematically retargeted trajectory as follows:

$$\tilde{\mathbf{a}}_t^{(s^*)} = \sqrt{\alpha_{s^*}}\hat{\mathbf{a}}_t + \sqrt{1-\alpha_{s^*}}\varepsilon_t, \quad \varepsilon_t \sim \mathbf{N}(0,I) \tag{2}$$

Here α corresponds to the diffusion schedule of the pre-trained policy. This procedure, inspired by SDEdit [15] which adopted a similar approach for image editing, relies on the assumption that $\{\hat{\mathbf{a}}_t\}_{t=0}^T$ is an approximate version of the ideal action trajectory that should be executed by the robot, and it potentially lies outside the distribution of feasible actions under the pre-trained diffusion policy $\bar{\pi}_\theta$. Intuitively, adding moderate noise to the retargeted trajectory and then denoising preserves the high-level structure of the human demonstration while allowing the policy to correct low-level infeasibilities from the embodiment gap. The diffusion policy $\bar{\pi}_\theta$ then performs iterative denoising steps, conditioned on the robot's observations $o_{\leq t}$, to refine this noisy trajectory into feasible robot actions, based on equation (2).

After s^* denoising steps, the final output $\mathbf{a}_t = \tilde{\mathbf{a}}_t^{(0)}$ is deployed on the robot. Importantly, this process is carried out in a closed-loop manner: the policy uses real-time observations from the cameras in the scene to iteratively improve its predictions, thereby compensating for embodiment mismatch and external perturbations (e.g., object slippage or occlusion). The key hyperparameter in this process for *DemoDiffusion* is the diffusion step s^* that trades off between the faithfulness to the demonstration and the likelihood under the robot policy – in the limit $s^* = S$ we recover a rollout from the base policy $\bar{\pi}_\theta$ and in the limit $s^* = 0$ we recover the kinematically retargeted trajectory $\{\hat{\mathbf{a}}_t\}_{t=0}^T$. While derived in the context of diffusion, note that this iterative refinement perspective also applies to flow-matching [20], which we show in real-world experiments.

III. Experiments

A. Real-World Manipulation

For the real-world experiments, we use a Franka Emika Panda arm equipped with a two-finger gripper from Robotiq. We use the publicly available pre-trained flow-matching policy $\bar{\pi}_\theta(\mathbf{a}_t|\mathbf{o}_{\leq t}, \mathscr{T})$ called Pi-0 released by Physical Intelligence [1], trained on the DROID dataset [21]. This policy takes as input a language task specification $\mathscr{T}$, and observations from two cameras in the scene, and outputs per-timestep joint velocity. This policy was trained on a large offline dataset of tele-operated robot demonstrations, and is a *generalist* policy that can perform a broad set of tasks. We do not fine-tune or adapt this policy in any way and directly use it for *DemoDiffusion*. We use an off-the-shelf monocular hand reconstruction model, Hamer [17] to obtain per-timestep hand mesh reconstructions from each of the external cameras given a human demonstration in the scene. Our experiments follow the protocol of a human first demonstrating an object manipulation in the scene, followed by robot execution after re-setting the scene.

Tasks. We perform evaluations for 8 different manipulation tasks, the language description and the human demonstration for which are described in Fig. 3. Each of these tasks requires a different manipulation strategy involving either prehensile or non-prehensile manipulation. We collect just a single human demonstration per task but for robot execution, we make organic variations in the location of

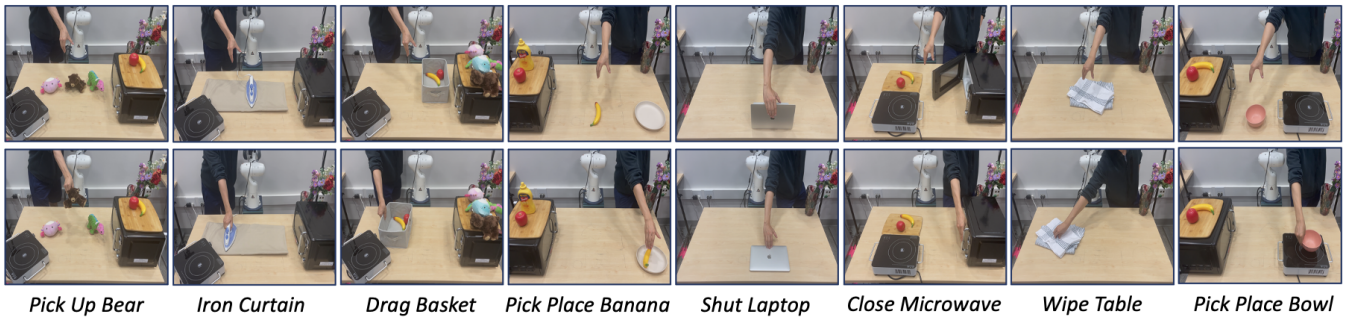

Fig. 3: **Real-World Manipulation Tasks.** Human demonstrations for the 8 evaluation tasks, shown as two frames per task. Tasks span prehensile and non-prehensile manipulation, including grasping, pushing, closing, wiping, and placing.

Task	Pi-0	Kinematic	***DemoDiffusion***
Shut Down Laptop	20	10	**60**
Close Microwave	10	80	**90**
Drag Basket	10	**80**	**80**
Wipe Table	50	0	**100**
Iron Curtain	20	70	**90**
Pick up Bear	0	40	**60**
Pick Place Bowl	0	70	**100**
Pick Place Banana	0	70	**90**
Average	13.8	52.5	**83.8**

TABLE I: **Quantitative Results for Real-World Manipulation.** Success rates (%) are over 10 trials per task.

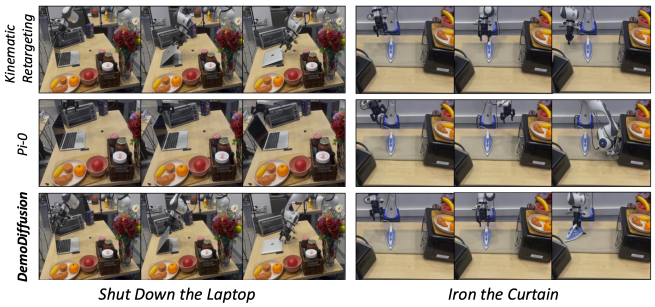

Fig. 4: **Qualitative comparisons for real-robot manipulation.** Rollout progressions (start, intermediate, final frame) for two tasks. Kinematic retargeting (top) produces plausible motion but loses contact before completing the task. Pi-0 (middle) performs general reaching but fails to manipulate the correct object. *DemoDiffusion* (bottom) reaches the object and maintains contact through task completion. See project page for full videos.

objects in the scene. Additionally, we add other objects in the scene as visual distractors, to see whether *DemoDiffusion* can generalize under such settings. All variations are consistent across the baselines and our method.

Baselines. For the real-world experiments we compare *DemoDiffusion* with two baselines. The first is open-loop kinematic retargeting obtained from the human demonstration. The second is directly deploying the Pi-0 model with the language instruction for the respective task. Since Pi-0 operates in the joint velocity space, for *DemoDiffusion* we first convert the kinematically retargeted trajectory to the joint velocity space with IK and then add noise to the trajectory, followed by denoising with Pi-0. We deploy *DemoDiffusion* with a heuristically selected noise level s^*/S depending on the task – preferring the kinematic trajectory with $s^*/S = 0.2$ when Pi-0 has a zero success rate and $s^*/S = 0.4$ otherwise. Note that in the limit of 0 noise added, we recover the first baseline, kinematic retargeting, and in the limit of maximum noise 1 we obtain the second baseline Pi-0.

Qualitative Visualizations. In Fig. 4, we show rollouts for two tasks across all the methods. We observe that Pi-0 can only perform general reaching behaviors: it either fails to follow the manipulation trajectory needed for the task or confuses the target object with a distractor. On the other hand, we observe that (open-loop) kinematic retargeting from the human demonstration has a plausible motion of the robot end-effector, but it completely misses grasping the iron, and loses contact with the laptop before fully closing it. Compared to these, *DemoDiffusion* reaches the respective object and also follows the approximate motion of the retargeted trajectory while remaining in contact with the object to solve the tasks.

Quantitative Comparisons. In Table I, we compare *DemoDiffusion* and the baselines across all the real-world manipulation tasks. *DemoDiffusion* matches or outperforms both baselines across all 8 tasks, with the largest gains on tasks requiring precise contact – e.g., 60% vs. 10–20% on shutting the laptop and 100% vs. 0–50% on wiping the table. Notably, even for tasks where both baselines fail, *DemoDiffusion* succeeds by operating at an intermediate noise level s^* that combines the trajectory structure from the human demonstration with the policy's closed-loop refinement.

IV. Discussion

We presented *DemoDiffusion*, a simple approach to leverage *generalist* diffusion models for imitating a human demonstration. We envision that this can serve as a starting point for future efforts at human imitation across generic scenarios and perhaps also yield an improved exploration strategy for methods pursuing policy adaptation with online RL.

REFERENCES

[1] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. $\pi 0$: A vision-language-action flow model for general robot control. In *RSS*, 2025.

[2] Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In *CoRL*, 2023.

[3] Eric Jang, Alex Irpan, Mohi Khansari, Daniel Kappler, Frederik Ebert, Corey Lynch, Sergey Levine, and Chelsea Finn. Bc-z: Zero-shot task generalization with robotic imitation learning. In *CoRL*, 2022.

[4] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. Openvla: An open-source vision-language-action model. In *CoRL*, 2024.

[5] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, et al. Rt-1: Robotics transformer for real-world control at scale. In *RSS*, 2023.

[6] Jinhan Li, Yifeng Zhu, Yuqi Xie, Zhenyu Jiang, Mingyo Seo, Georgios Pavlakos, and Yuke Zhu. Okami: Teaching humanoid robots manipulation skills through single video imitation. In *CoRL*, 2024.

[7] Georgios Papagiannis, Norman Di Palo, Pietro Vitiello, and Edward Johns. R+ x: Retrieval and execution from everyday human videos. In *ICRA*, 2025.

[8] Chen Wang, Haochen Shi, Weizhuo Wang, Ruohan Zhang, Li Fei-Fei, and C Karen Liu. Dexcap: Scalable and portable mocap data collection system for dexterous manipulation. In *RSS*, 2024.

[9] Sirui Chen, Chen Wang, Kaden Nguyen, Li Fei-Fei, and C Karen Liu. Arcap: Collecting high-quality human demonstrations for robot learning with augmented reality feedback. In *ICRA*, 2025.

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

[11] Irmak Guzey, Yinlong Dai, Georgy Savva, Raunaq Bhirangi, and Lerrel Pinto. Bridging the human to robot dexterity gap through object-oriented rewards. In *ICRA*, 2025.

[12] Yuanpei Chen, Chen Wang, Yaodong Yang, and C Karen Liu. Object-centric dexterous manipulation from human motion data. In *CoRL*, 2024.

[13] Xueyi Liu, Jianibieke Adalibieke, Qianwei Han, Yuzhe Qin, and Li Yi. Dextrack: Towards generalizable neural tracking control for dexterous manipulation from human references. In *ICLR*, 2025.

[14] Xueyi Liu, Kangbo Lyu, Jieqiong Zhang, Tao Du, and Li Yi. Quasisim: Parameterized quasi-physical simulators for dexterous manipulations transfer. In *ECCV*, 2024.

[15] Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. Sdedit: Guided image synthesis and editing with stochastic differential equations. In *ICLR*, 2022.

[16] Vidhi Jain, Maria Attarian, Nikhil J Joshi, Ayzaan Wahid, Danny Driess, Quan Vuong, Pannag R Sanketi, Pierre Sermanet, Stefan Welker, Christine Chan, et al. Vid2robot: End-to-end video-conditioned policy learning with cross-attention transformers. In *RSS*, 2024.

[17] Georgios Pavlakos, Dandan Shan, Ilija Radosavovic, Angjoo Kanazawa, David Fouhey, and Jitendra Malik. Reconstructing hands in 3d with transformers. In *CVPR*, 2024.

[18] Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. In *IJRR*, 2023.

[19] Moritz Reuss, Maximilian Li, Xiaogang Jia, and Rudolf Lioutikov. Goal conditioned imitation learning using score-based diffusion policies. In *RSS*, 2023.

[20] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *ICLR*, 2023.

[21] Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, Siddharth Karamcheti, Soroush Nasiriany, Mohan Kumar Srirama, Lawrence Yunliang Chen, Kirsty Ellis, et al. Droid: A large-scale in-the-wild robot manipulation dataset. In *RSS*, 2024.

[22] Hui Zhang, Sammy Christen, Zicong Fan, Otmar Hilliges, and Jie Song. Graspxl: Generating grasping motions for diverse objects at scale. In *ECCV*, 2024.

[23] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015.

[24] Kaichun Mo, Shilin Zhu, Angel X Chang, Li Yi, Subarna Tripathi, Leonidas J Guibas, and Hao Su. Partnet: A large-scale benchmark for fine-grained and hierarchical part-level 3d object understanding. In *CVPR*, 2019.

[25] Yanjie Ze, Gu Zhang, Kangning Zhang, Chenyuan Hu, Muhan Wang, and Huazhe Xu. 3d diffusion policy: Generalizable visuomotor policy learning via simple 3d representations. In *RSS*, 2024.

[26] Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. Objaverse: A universe of annotated 3d objects. In *CVPR*, 2023.

[27] Marion Lepert, Jiaying Fang, and Jeannette Bohg. Phantom: Training robots without robots using only human videos. In *CoRL*, 2025.

APPENDIX

We evaluate our approach on dexterous grasping in simulation and across diverse real-world tasks comprising prehensile and non-prehensile table-top manipulation. We report additional results in here. Through experiments, we aim to understand the following research questions:

- Can *DemoDiffusion* outperform pure kinematic retargeting from the human demonstration?
- Do the human demonstrations allow *DemoDiffusion* to perform new tasks where the pre-trained diffusion policy fails?
- How to effectively tradeoff between faithfulness to the human demonstration and performing the task reliably, with varying noise level s^*/S?
- How robust is *DemoDiffusion* to noise in human demonstration, or different kinematic retargeting?

A. Dexterous Grasping in Simulation

To verify our intuition, we first consider a simulation environment where the target task is restricted to picking up a generic object with a 16-DOF four-fingered Allegro hand. Specifically, we train a dexterous grasping policy across a small set of generic objects in simulation, and test our method on human hand grasping trajectories on a different set of objects. This serves as the pre-trained diffusion model for this experiment.

For training the robot policy, we collect a total of 985 grasping trajectories of Allegro hand over 58 training objects [22] (randomly sampled 26 and 32 objects from ShapeNet [23] and PartNet [24], respectively), resulting in 22.2% success rate. We use a variant of 3D Diffusion Policy [25], which takes 3D point tracks of an object instead of point clouds. At test time, we provide human grasping trajectories on a subset of Objaverse dataset [26], which are unseen during training. The test set contains 1220 objects, with 1 human grasping trajectory per object, provided by GraspXL [22]. Note that there is no error from hand estimation or the initial configuration of the object in this setup. We use ground-truth human hand 3D keypoints, and the object is located at the same location as the human demonstration.

Group	Robot Policy	Kinematic	***DemoDiffusion***
Small	25.4	2.6	**31.8**
Medium	26.6	0.7	**31.6**
Large	27.4	1.5	**29.6**
Average	26.5	1.6	**31.0**

TABLE II: **Simulated Dexterous Grasping Results.** The numbers show the average of success rates over 3 seeds. We use $s^*/S = 0.2$, where the robot policy uses $S = 1000$. *DemoDiffusion* consistently outperforms both baselines, across all groups.

Results. As in GraspXL [22], we group the objects based on their size, resulting in small, medium, and large objects. We compare *DemoDiffusion* with two baselines. The first is open-loop kinematic retargeting obtained from the human demonstration. The second is directly deploying the robot policy we trained.

Table II shows the main results. Our method outperforms baselines, with the largest gains on small objects. We hypothesize this is because larger objects are easier to grasp, supported by base robot policy performance improving for larger objects. This experiment shows that even in a highly controlled setup where the robot environment is initialized identically to that of human demonstration, kinematic retargeting cannot solve the task, while slightly refining with a generalist grasping policy significantly improves the performance. Using kinematic retargeting to initiate the denoising process also helps the denoising process itself, outperforming the robot policy. Additionally, *DemoDiffusion* achieves higher inference speed, by the factor of S/s^*.

Ablation Studies. We analyze the influence of s^* on *DemoDiffusion* in Fig. 5. Overall, we observe a consistent trend of performance gain from $s^*/S = 1$ to $s^*/S = 0.2$, showing that *DemoDiffusion* is robust to the choice of hyperparameter.

B. Real-World Manipulation

Workspace and Hardware Setup. Fig. 6 describes our real-robot manipulation setup. The scene has 5 cameras including 4 Realsense external cameras and 1 Zed-Mini wrist camera mounted on the robot gripper. We calibrate the four external cameras and use them for triangulating the 3D hand pose so that we can obtain the position and orientation of a human hand in the world coordinates, with origin at the base of the robot. The pre-trained policy Pi-0 requires just two cameras (marked in purple) and the policy does not use any calibration information. While Pi-0 uses Zed 2 camera for external view, we use Realsense camera due to hardware availability, which introduces additional difficulty.

Task Setup Description. The overview of scenes for each task at test time is shown in Fig. 7. During data collection, we mark the target object location with a tape. During evaluation, we randomize the target object location within the tape, which is consistent across all methods. Overall, kinematic retargeting may fail due to small misalignment of object placement, inaccurate 3D hand keypoints, or camera extrinsics, etc. Pi-0 may fail if it cannot identify the object to interact with and manipulate other objects in the scene. It may also reach the target object but fail to manipulate in the desired way. Note that previous approaches [6]–[9] use kinematic retargeting to transfer human demonstration to the robot.

- *Pick and Place the Banana.* The task is to pick up the toy banana and place on the plate on the right side of the table. The episode is considered successful if the banana is successfully placed on the plate.
- *Drag the Basket.* The task is to drag the basket to the right side of the table. The human demonstration grasps the edge of the basket and pulls it to the right. The episode is considered successful if the robot grasps the basket/puts the gripper inside the basket and moves it to any side of the table.

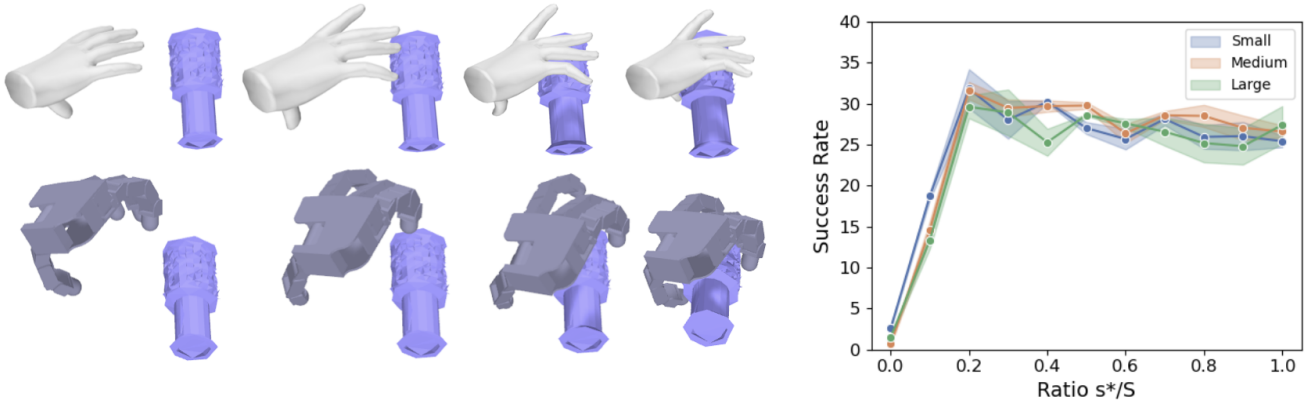

Fig. 5: **Dexterous Grasping Results in Simulation.** (Left) A human demonstration and *DemoDiffusion* rollout for dexterous grasping. (Right) Success rates (mean $\pm$ std over 3 seeds) as a function of diffusion step s^*. Here, $s^*/S = 0$ corresponds to kinematic retargeting and $s^*/S = 1$ corresponds to the robot policy.

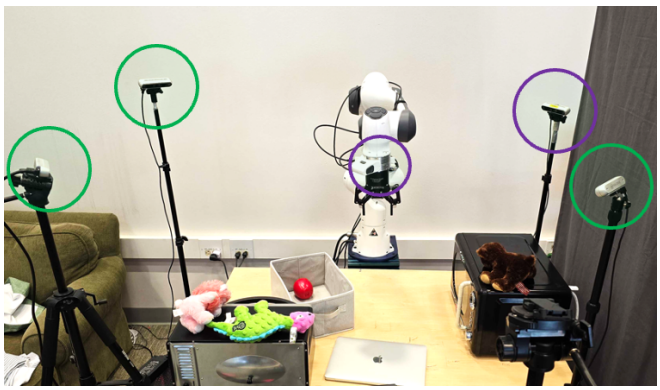

Fig. 6: **Workspace with 5 Cameras.** We use the four external cameras for triangulation to obtain the global pose of the hand mesh from a human demonstration. The pre-trained policy uses the two cameras marked in purple.

- *Pick Up the Teddy Bear.* The task is to pick up the teddy bear located at the center of the table. The human demonstration grasps the neck of the bear and picks it up. The episode is considered successful if the robot picks up the bear by the end of the episode.
- *Pick and Place the Bowl.* The task is to pick up the bowl and place it on the stove on the right side of the table. The episode is considered successful if the bowl is successfully placed on the stove.
- *Iron Curtain.* The task is to grasp the ironing machine located at the center of the table and slide it to the right side of the curtain. The episode is considered successful if the robot grasps/pushes the ironing machine to any part of the curtain.
- *Shut Down the Laptop.* The task is to close the laptop located in front of the robot. The episode is considered successful if the angle of the laptop is smaller than 10 degrees at the end of the episode.
- *Close the Microwave.* The task is to close the microwave located at the corner of the table. The episode is considered successful if the handle part touches the body part of the microwave at the end of the episode.
- *Wipe the Table.* The task is to pinch/poke the given tissue and wipe the table. The episode is considered successful if the robot pinches/pokes the tissue and performs one stroke to any side of the table without missing it.

Text Prompt. We used the following text conditions for Pi-0 [1], which are also used for *DemoDiffusion*.

- **Pick and Place the Banana**: *pick and place the banana on the plate*
- **Drag the Basket**: *drag the basket to the right side*
- **Pick Up the Teddy Bear**: *pick up the teddy bear*
- **Pick and Place the Bowl**: *pick and place the bowl on the stove*
- **Iron the Curtain**: *move the ironing machine to the right side*
- **Wipe the Table**: *wipe the table*
- **Close the Microwave**: *close the microwave*
- **Shut Down the Laptop**: *close the laptop*

Inference Details. At each timestep t, we index the human demonstration at same timestep t, and compute the kinematically retargeted actions. As Pi-0 predicts a chunk of actions with length $H = 10$, we also use the action chunk computed from human demonstration from timestep t to $t + H - 1$. The episode length is set to be the same as the length(number of timesteps) of human demonstrations across all methods.

Kinematic Retargeting Implementation Details. A human demonstration $\mathbf{D}$ of a manipulation task $\mathscr{T}$ consists of the 3D hand pose trajectory $\{\mathbf{h}_t\}_{t=0}^{T}$. Each hand pose $\mathbf{h}_t \in \mathbb{R}^{3\times J}$, which corresponds to the 3D locations of J keypoints (e.g., wrist and fingertips). To kinematically retarget these 3D keypoints to a robotic end-effector, we use the human's wrist position A, thumb tip position B, and the average of other fingertips C. Specifically, $\vec{AB}$, along with the cross product

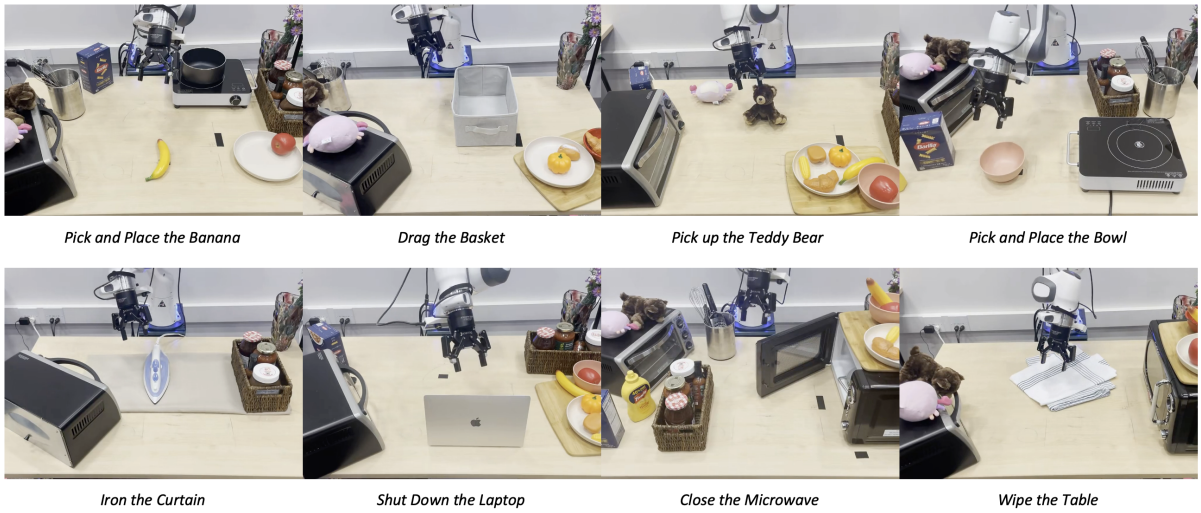

Fig. 7: **Real-World Manipulation Tasks at Test Time.** Each image corresponds to an example of the scene for each task at test time. To test our method in a natural setup, we add other objects in the scene as visual distractors. For each rollout, we use the same set of distractors across methods.

TABLE III: Hyperparameters in Real-world.

Hyperparameter	**Value**
Open Loop Horizon	8
Predict Action Horizon	10
Denoising Steps at Inference (Pi-0)	20

of $\vec{AB}$ and $\vec{AC}$ define the orientation of the end effector, and the human's wrist position is used as the position of the end effector. As Pi-0 uses joint velocity as action space, the end effector pose is further converted into joint velocity using inverse kinematics. For the gripper, we use the norm of $\vec{BC}$ as its width. To stabilize grasping, we set the gripper action to be fully closed if the norm of $\vec{BC}$ is smaller than 80% of the gripper's maximum width, similar to [27].

Hyperparameters. To show our method's applicability across diverse setups, we do not change the speed of the given human demonstration when performing kinematic retargeting. At inference, we use the following hyperparameters for Pi-0 and *DemoDiffusion*, with only difference on the number of denoising steps, as *DemoDiffusion* starts denoising from intermediate steps. Specifically, we use the following hyperparameters. Depending on the noise level s^*/S, *DemoDiffusion* uses $20 \times s^*/S$ denoising steps.

Results with Different Hyperparameters. We present the results of *DemoDiffusion* using two different hyperparameter settings, $s^*/S = 0.2$ and $s^*/S = 0.4$. The results are summarized in Table IV. Overall, we observe a consistent trend based on the performance of the base diffusion policy (*Pi-0*):

- When the base policy *Pi-0* achieves a non-zero success rate, $s^*/S = 0.4$ generally yields better performance. This is observed in tasks such as *shut laptop*, *close microwave*, *drag basket*, *wipe table*, and *iron curtain*.
- Conversely, when *Pi-0* has a zero success rate, $s^*/S = 0.2$ tends to perform better. This is the case for tasks including *pick up bear*, *pick place banana*, and *pick place bowl*. We observe that while $s^*/S = 0.4$ shows adaptive end effector motion, it tends to avoid activating the gripper at the correct timestep.

Fig. 8: **Zero-Shot Generalization to New Objects.** We evaluate *DemoDiffusion*'s generalization capabilities to new objects on 3 tasks - drag the basket, wipe the table, pick and place the bowl. While the objects have different colors and shapes from those of objects in the human demonstration, *DemoDiffusion* adjusts its behavior to these new objects. Please refer to the website for detailed visualizations.

TABLE IV: **Additional Quantitative Results for Real-World Manipulation.**

Task	**Pi-0**	**Kinematic Retargeting**	***DemoDiffusion*** (0.2)	***DemoDiffusion*** (0.4)
Shut Laptop	20	10	40	**60**
Close Microwave	10	80	80	**90**
Drag Basket	10	**80**	70	**80**
Wipe Table	50	0	60	**100**
Iron Curtain	20	70	70	**90**
Pick Up Bear	0	40	**60**	40
Pick Place Banana	0	70	**90**	20
Pick Place Bowl	0	70	**100**	0
Average	13.8	52.5	**71.3**	60.0

These findings suggest that a more effective base policy enables *DemoDiffusion* to benefit from a higher noise level, as it can leverage the base policy's strengths more effectively. In contrast, when the base policy is less reliable, a lower

noise level tends to show better performance. We believe this can serve as a guideline for choosing the noise level s^*/S for *DemoDiffusion*.

Zero-Shot Generalization to New Objects. Lastly, we investigate whether *DemoDiffusion* can achieve zero-shot generalization to new objects within the same category. We use the same human demonstration and text prompt we previously used while evaluating on a new object within the same category. Results are shown in Fig. 8. Although we only provide human demonstration on a different object, we observe that *DemoDiffusion* can adapt its behavior to a new object.

Noisy Human Demonstrations. To evaluate the robustness of *DemoDiffusion* to imperfect inputs, we injected noise into the tracked human trajectories and evaluated *DemoDiffusion*. Specifically, we place the object at the same location as the human demonstration, and randomly shift the tracked hand 3D keypoints by 5cm. The results in Table V indicate that *DemoDiffusion* maintains high-quality performance despite the presence of tracking noise, thereby reducing the reliance on highly accurate 3D annotations in human demonstration.

Different Kinematic Retargeting. Lastly, we further assessed the sensitivity of our method to the choice of retargeting by evaluating an alternative approach ('Thumb+Index'). Specifically, we use the distance between thumb and index finger [27], instead of all fingers, to infer the grasp. Table V shows the results. While this alternative retargeting itself shows degraded performance compared to the previous retargeting, *DemoDiffusion* consistently improves task performance despite suboptimal retargeting.

	Shut Laptop	Pick Place Banana
DemoDiffusion	60	90
+5cm noise	40	70
with Thumb+Index	30	80
Thumb+Index	0	30

TABLE V: **Robustness of *DemoDiffusion*.** We ablate *DemoDiffusion* under noisy 3D keypoints (2nd row) and with different kinematic retargeting (3rd row). For noisy 3D keypoints, we randomly shift the tracked hand 3D keypoints by 5cm. For different kinematic retargeting, we use the distance between the thumb tip and the index fingertip instead of all fingers.