ZeroMimic: Distilling Robotic Manipulation Skills from Web Videos

Anonymous CVPR submission

Paper ID 10

Abstract

Many recent advances in robotic manipulation have come through imitation learning, yet these rely largely on mimicking a particularly hard-to-acquire form of demonstrations: those collected on the same robot in the same room with the same objects as the trained policy must handle at test time. In contrast, large pre-recorded human video datasets demonstrating manipulation skills in-the-wild already exist, which contain valuable information for robots. Is it possible to distill a repository of useful robotic skill policies out of such data without any additional requirements on robot-specific demonstrations or exploration? We present the first such system ZeroMimic, that generates immediately deployable image goal-conditioned skill policies for several common categories of manipulation tasks (opening, closing, pouring, pick&place, cutting, and stirring) each capable of acting upon diverse objects and across diverse unseen task setups. ZeroMimic is carefully designed to exploit recent advances in semantic and geometric visual understanding of human videos, together with modern grasp affordance detectors and imitation policy classes. After training ZeroMimic on the popular EpicKitchens dataset of ego-centric human videos, we evaluate its out-of-the-box performance in varied real-world and simulated kitchen settings with two different robot embodiments, demonstrating its impressive abilities to handle these varied tasks. To enable plug-and-play reuse of ZeroMimic policies on other task setups and robots, we release software and policy checkpoints of our skill policies.

1. Introduction

It is clear that animals and humans are able to observe third-person experiences to acquire functional sensorimotor skills, often "zero-shot" with limited or no need for additional practice. For example, one can learn to cook pasta, use a wood lathe, plant a garden, or tie a necktie, with reasonable proficiency by watching how-to video demonstrations on the web. While "imitation learning" has also been instrumental in many recent successes for *robotic* manipulation [1–4], these robots largely rely on a much stronger kind of demonstration — gathered by manually operating the very same robot in the same small set of scenarios (scenes, viewpoints, objects, lighting, background textures, and distractors) to perform the task of interest. This is an immediate stumbling block on the road to developing general-purpose robots: gathering robot- and scenario-specific demonstrations scales poorly.

Learning robot skills from in-the-wild human videos offers the enticing prospect that data would no longer be a bottleneck: videos of humans demonstrating varied manipulation tasks in diverse scenarios are already available on the web, it is easy to gather many more if needed, and further, the same videos could be re-used for many robots. However, there are serious challenges. Robots differ from humans in embodiments, action spaces, and hardware capabilities. Individual web videos often do not conveniently present all the details of how to perform a task (e.g. occlusions, out-of-frame objects and actions, or shaky moving cameras). Finally, the distribution of in-the-wild videos spans very large variations that may be hard to handle.

We present an approach, ZeroMimic, that systematically overcomes these challenges and distills in-the-wild egocentric videos from EpicKitchens [5] into a repository of off-the-shelf deployable image goal-conditioned robotic manipulation skill policies that transfer across scenarios. Briefly, we abstract the action spaces of humans and standard robot arms with two-fingered grippers to permit coarse action transfer, we exploit video activity understanding and pre-existing visuomotor robot primitives such as grasping to transfer the finer details of control, we exploit modern structure-from-motion systems to maintain 3D maps of noisy and shaky in-the-wild egocentric human videos, and demonstrate that large policy classes can digest the diversity of web video to learn useful behaviors. The resulting system empirically demonstrates zero-shot robotic manipulation capabilities to perform a wide range of skills with diverse objects. In summary, our contributions are:

1. We develop ZeroMimic, a system that distills robotic manipulation skills from web videos that can be deployed zero-shot in diverse everyday environments.

Figure 1: **ZeroMimic** distills robotic manipulation skills from egocentric web videos for zero-shot deployment across diverse real-world and simulated environments, a variety of objects, and different robot embodiments.

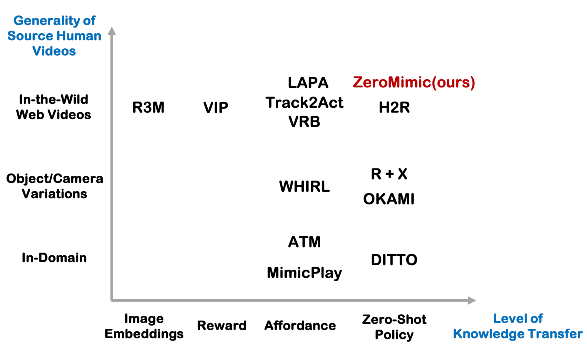


Figure 2: Representative related work organized by **Generality of Source Human Videos** and **Level of Knowledge Transfer**. ZeroMimic learns diverse zero-shot policies from in-the-wild web videos.

2. We evaluate ZeroMimic on 9 different skills and show that ZeroMimic achieves 71.0% out-of-the-box success rate in the real world, 73.8% success rate in simulation, can generalize to new objects unseen in our curated web video, and can be deployed across different robot embodiments.

3. Our ablation studies reveal important lessons of what is important in learning and executing robotic skills purely from in-the-wild human videos.

2. Related Work

Popular recent approaches [1–3] for enabling robot manipulation often rely on costly high-quality in-domain robot demonstrations. Therefore, recent works in robot learning have increasingly focused on leveraging unstructured or out-of-domain data. Some works have demonstrated the zero-shot capabilities of models trained on large robotics datasets [6–14], but the curation of such datasets incurs a significant cost. Some have exploited recent advances in VLMs, trained on "web" data without any connection to robotics, and directly elicit zero-shot robotic actions [15–21]. These policies are limited by the lack of physical understanding and slow inference speed of VLMs, as demonstrated by our experiments in Section 4.4. Human web videos [5, 22–26], due to their abundance, diversity, and rich information about interactions, emerge as a promising source of data for robotic skill acquisition.

Since generating robot policies from out-of-domain human videos directly is difficult, many works instead train representations [27–29] (e.g. R3M [27]), rewards [30–33] (e.g. VIP [31]), or affordances [34–60] (See Fig 2). Some works [34–45] (e.g. MimicPlay [34], WHIRL [36], and ATM [39]) explored learning afoordances from in-domain human videos. Recent works [46–60] (e.g. VRB [53], Track2Act [57], LAPA [60]) extended these approaches to learning from in-the-wild human videos. Since these visual representations, reward functions, and affordances are not explicitly actionable for robots, they still depend on in-domain robot data to learn manipulation policies.

Very limited prior work [61–66] such as DITTO [61], R+X [64] and OKAMI [65] has aimed to directly generate policies from human videos without any in-domain data. These methods typically require the distribution of human demonstrations to be similar enough to the test-time robot environment and assumes knowledge of ground truth camera and depth information, making them unsuitable for learning from diverse and unstructured web data. Some methods also rely on heuristic-based mappings from human hand poses to robot gripper actions during data collection [64] or have manually defined constraint formulations [66], limiting the range of demonstrations and tasks these methods can handle.

As Figure 2 shows, to our knowledge, the only prior work that attempts zero-shot policies from truly in-the-wild videos is H2R [67], which learns plausible 3D hand trajectories from egocentric in-the-wild EpicKitchens [5] videos and retarget them to robot end effector for zero-shot deployment in real-world settings. We too train policies on EpicKitchens data, but with critical pre-processing steps that ground the data in 3D and generate higher quality policies. Further, we design a robust system that combines learned pre-contact interaction affordances and learned post-contact action policies. As our experiments show, these method improvements translate to dramatic gains in the ability to generate functional out-of-the-box performance for manipulation skills in the real world.

3. Method

We focus on manipulation skills that permit decomposition into two phases: the **grasping phase** that consists of approaching and grasping an object of interest appropriately for the target task, and the **post-grasp phase** which consists of a rigid manipulation of the object while stably held in the gripper. This encompasses such diverse skills as pick&place, slide opening and closing, hinge opening and closing, pouring, cutting, and stirring. ZeroMimic pretrains components specific to these two phases, as described in Sec 3.1 and 3.2 before combining them, as in Figure 3. We focus on distilling human videos from EpicKitchens [5] into robotic skills. Pre-training on off-the-shelf human data naturally constrains our approach to be not tied to any specific robotic system design: we target static robot arms with 2-fingered grippers, observing the scene with an RGB-D camera from any egocentric-like vantage point of the robot workspace. See Appendix 6 for images and more details of our experiment setups.

3.1. Human Affordance-Based Grasping

For this phase, we use human videos to learn to identify the appropriate region of the scene to seek to execute a grasp in, i.e., affordance prediction. Subsequent to this, given that human videos are of limited use in selecting the grasp itself due the vastly different embodiment of the robot's gripper and the human hand, we use an approach trained on robot data to identify suitable grasps for a 2-fingered gripper within that region, i.e. grasp selection.

For affordance prediction, we use VRB [53] to generate a 3D point of intended contact. VRB is pre-trained on EpicKitchens [5]. It generates pixel-space grasp locations, given an RGB image and a task description in natural language, e.g. "open drawer". Next, to select a grasp close to this chosen location, we use AnyGrasp [68], a widely used grasp generation model pre-trained on robot data for our 2-fingered robotic grippers. Once a grasp is chosen, we plan a linear end-effector motion through free space to execute it. See Figure 3 for examples of intermediate outputs after each stage of processing above, and the resulting grasp execution.

3.2. Human Movement-Based Post-Grasp Robot Policy

Once the robot has grasped the object, it must decide what 6D end-effector trajectory to execute to accomplish the task. ZeroMimic's post-grasp module is an imitation policy that distills this information from in-the-wild human videos. We first extract human wrist trajectories grounded in world 3D coordinates by reconstructing the hand pose and the egocentric camera, Given a skill, we take the corresponding subset of the data and train a skill model to predict 6D wrist trajectory.

Extracting Human Wrist Trajectories From Web Videos To curate diverse and large-scale human behavior, we use EpicKitchens [5], an in-the-wild egocentric vision dataset. It contains 20M frames in 100 hours of daily activities in the kitchen. To extract wrist trajectories from EpicKitchens, we run HaMeR [69], a state-of-the-art pre-trained hand-tracking model, to obtain 3D hand pose reconstruction. HaMeR outputs the locations and orientations of all hand joints relative to a canonical hand, along with camera parameters corresponding to a translation $t \in \mathbb{R}^3$. We use camera parameters inferred through the COLMAP [70] structure-from-motion algorithm, as provided in EPIC-Fields [71], to convert these pixel-coordinate-based hand pose outputs into world 3-D coordinates. We consider only the wrist joint, and the result is 6D wrist trajectories $\{h_t = (x_t, y_t, z_t, \alpha_t, \beta_t, \gamma_t)\}_{t=1}^{T}$ for a T-frame clip that is expressed in the world coordinate. See our website for videos of ZeroMimic's hand reconstructions.

Policy Training, Execution, and Implementation Details A major challenge for learning to predict trajectories from web videos is the highly multi-modal nature of human demonstrations – there are multiple ways to manipulate objects in a scene given the same image observation. To model this multi-modality, we use the recently popular action chunking transformer (ACT) [1] policy class to learn a generative model over action sequences. The input of our model is the current image I_t, goal image I_g, and the current wrist pose h_t, and our model outputs future wrist poses $\{h_i\}_{i=t+i}^{t+n+1}$, where n is the prediction chunk size. We use the last frame in the task-relevant clip as the goal image. See Figure 3 for an illustration. Since at test time, we perform robot experiments with a static camera, we relieve the burden

CVPR
#10

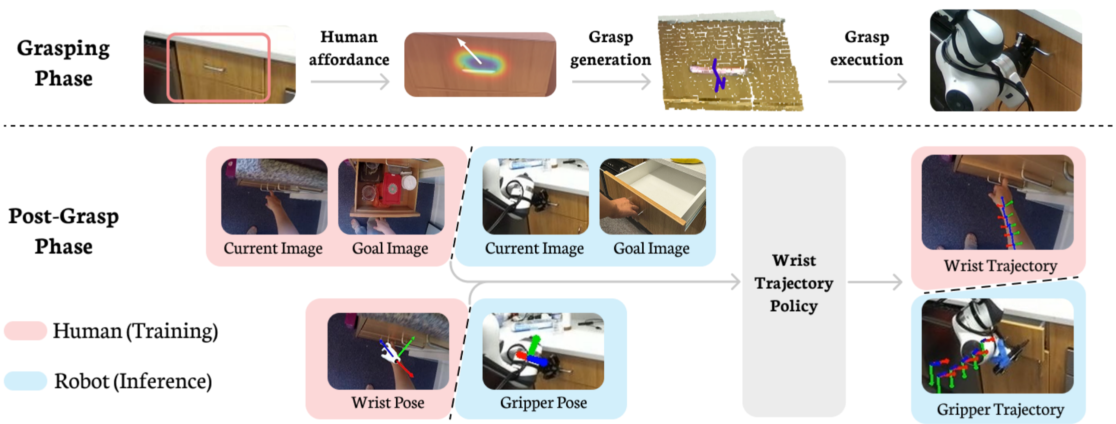
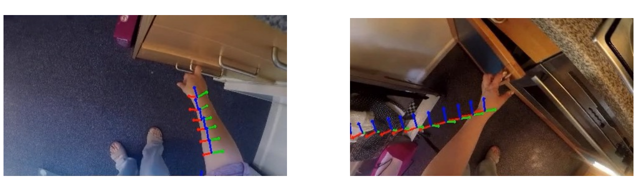


Figure 3: **ZeroMimic** is composed of the **grasping phase** and the **post-grasp phase**. The **grasping phase** (top) leverages human affordance-based grasping to execute a task-relevant grasp. The **post-grasp phase** (bottom) is an imitation policy trained on web videos to predict 6D wrist trajectories. We deploy this trained model directly on the robot.

of the model to predict camera parameters by transforming all current and future wrist poses into the current frame's camera coordinate using the camera extrinsics of each frame. See Figure 4 for qualitative visualization of generated wrist trajectories on unseen human videos. We train one model for each skill, obtaining a set of 9 skill policies. Our model predicts relative 6D wrist poses with a chunk size of $n = 10$, and we discuss the impact of these choices in Section 4.2. We train each skill policy for 1000 epochs, which takes approximately 18 hours on an NVIDIA RTX 3090 GPU.

Retargeting Human Wrist Policy to the Robot We deploy our trained post-grasp policies directly on the robot to generate 6D gripper trajectories (See Figure 3). We use a single image of a human achieving the desired outcome as the "goal image" for all trials of a task. In addition to the goal image, we provide the policy with the current RGB observation and the current gripper pose in the camera's frame. The model predicts 6D trajectories in the same camera frame, which is converted to the robot frame for execution. The robot executes all actions in a chunk before prompting the model for the next round of inference.



(a) Open drawer

(b) open cupboard

Figure 4: 6D wrist post-grasp policy outputs on unseen images. The red, green, and blue arrows denote the x, y, z coordinates of the wrist orientation in the camera frame.

4. Experiments

We evaluate ZeroMimic skill policies out-of-the-box on a diverse set of real-world and simulation objects with two different robot embodiments. See our website **zeromimic.github.io/anonymous.html** for videos. Our experiments aim to answer the following questions:

1. How important is each component of ZeroMimic to its eventual performance?
2. How well do ZeroMimic skill policies perform when deployed zero-shot to perform varied skills in diverse real-world and simulation environments?
3. How does ZeroMimic compare to other state-of-the-art zero-shot robotic system?
4. What are the failure modes and causes of ZeroMimic?

4.1. Experiment Setup

We use the text annotations of EpicKitchens [5] to curate human video data corresponding to each manipulation skill. Having trained the 9 ZeroMimic skill policies on in-the-wild human videos, we evaluate them on our robot in real-world and simulation environments. Our real-world evaluation spans 30 distinct scenarios across 18 object categories in 6 kitchen scenes. In simulation, we evaluate 4 skill policies, randomizing kitchen scenes across trials. Figure 5 show the results. See Appendix 7 for a detailed breakdown of skills, robots, object categories, scenarios, and success rates. None of the object instances or scenarios used in our experiments feature in our training data.

Real-world experiment setup All real-world experiments are performed in 3 different real kitchens on the UPenn campus and two different robots, a Franka Emika Panda

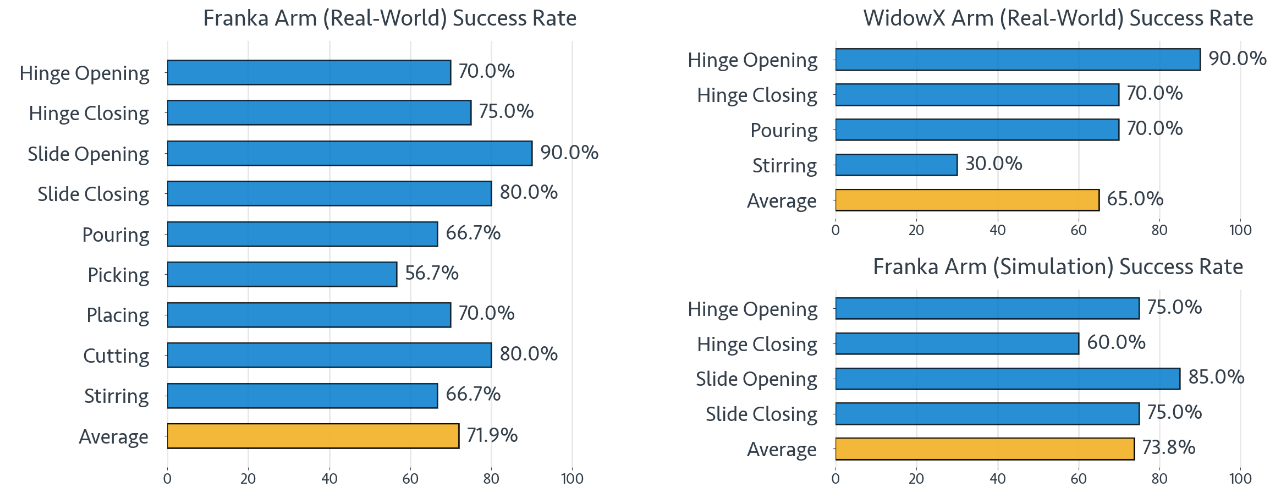


Figure 5: **ZeroMimic Zero-Shot Performance Overview**. ZeroMimic demonstrates strong generalization capabilities, achieving consistent success across diverse tasks, robot embodiments, and both real-world and simulated environments. The evaluation spans 34 distinct scenarios across 18 object categories in 7 kitchen scenes, highlighting the adaptability and robustness of the system. For a detailed breakdown of performance by skills, robots, object categories, and scenarios, refer to Appendix 7.

arm and a Trossen Robotics WidowX 250 S arm. Before our evaluations, we position the camera and the robot at a camera angle roughly similar to the relative camera angle of the human hand appearing in egocentric videos: camera at human height, and gripper within human arm's reach of the camera. We perform 10 trials with varying camera and robot positions to generally resemble a human's egocentric viewpoint. The success of all experiments is determined by their consistency with the goal provided by the human goal image. Visualizations of the trial positions are available on our website through an experiment time-lapse. See Appendix 6.1 for more details of our real-world experiment setup and images of our real kitchen scenes.

Simulation experiment setup We conduct our simulation experiments in RoboCasa [72]. We evaluate 4 ZeroMimic skill policies, each across 20 randomized kitchen trials. For each trial, we vary the camera and robot positions, background objects, and kitchen styles (e.g., textures, object placements). We select camera views most similar to a human egocentric perspective. See Appendix 6.2 for more details of our simulation experiment setup and images of our simulated kitchen scenes.

4.2. Contribution of Each System Component to ZeroMimic

We first validate the design of the ZeroMimic procedure by measuring the importance of each of its components on two real-world tasks with the Franka robot: *Open Drawer* and *Open Cupboard*. To do this, we construct ablated variants of ZeroMimic that either drop a component or replace it with simpler alternatives. More details about the setup of these variants can be found in Appendix 8.

Grasping Methods ZeroMimic employs the human interaction affordance provided VRB [53] to select which grasp produced by AnyGrasp [68] to execute. We compare our approach to two simpler alternatives: (1) selecting the best grasp directly using AnyGrasp's grasp score (**Ours w/o interaction affordance**), as done in [73], and (2) moving the end effector to the 2D contact point lifted to 3D with depth, and close the gripper (**Ours w/o grasp model**), as done in [53, 54]. The results in Table 1 indicate that **ours** is clearly the best method. **Ours w/o interaction affordance** fails by proposing grasps on irrelevant scene regions, while **Ours w/o grasp model** struggles due to incorrect gripper orientations and imprecise contact predictions.

Grasping Task	Ours	Ours w/o Affordance	Ours w/o Grasp Model
Drawer Handle	**8/10**	0/10	0/10
Cupboard Handle	**7/10**	4/10	6/10

Table 1: Success rates for different grasping methods.

Wrist Post-Grasp Policy After grasping the object, we deploy our 6D post-grasp policy to execute the task. H2R [67] also trains 6D wrist post-grasp policy on web videos, however the key difference is that it does not account for the impact of camera motion on the human hand motions detected in the video frames. We consider a strengthened version of H2R (**ours w/o SfM**) by simply removing camera extrinsics and intrinsics when processing our training data. Next, **VRB** is trained on web videos to produce post-contact trajectories only in terms of 2D pixel locations on the image, rather than 6-DOF wrist trajectories. To execute it on the robot, we sample a target end-point depth at random and interpolate the trajectory while fixing the gripper orientation.

We evaluate the task success rate of our model and two alternatives after a successful grasp, and the results in Table 2 show the superiority of our model, highlighting the importance of camera information from SfM and predicting dimensions beyond pixel coordinates. Both compared methods in this paragraph were designed as ablations of the post-contact wrist trajectory component of ZeroMimic; as such, they benefit from ZeroMimic's robust grasping phase. Without this, they would struggle still further: H2R cannot execute grasps in the original paper, and VRB does not provide grasp orientation even though it generates a contact point.

Task	Ours	Ours w/o SfM	VRB
Open drawer	**10/10**	4/10	2/10
Open cupboard	**10/10**	6/10	0/10

Table 2: Success rates for different post-grasp policies after a successful grasp.

Additionally, to understand critical factors for predicting wrist trajectories from web videos, we evaluate several design choices of the post-grasp module using teleoperated successful grasps. We find that ACT [1] and Diffusion Policy [2] policy architectures yield similar performance. Regarding action representation, relative actions in both translation and orientation significantly outperform absolute representations. Detailed results of these experiments are provided in Appendix 9.

4.3. ZeroMimic Zero-Shot Deployment Performance

Having established the robustness of ZeroMimic's system design above, we proceed to evaluate all 9 ZeroMimic skill policies zero-shot in varied real-world and simulated scenes with diverse objects and viewpoints. They achieve an impressive overall success rate of 71.9% in the real word with the Franka arm, 65.0% in the real world with the WidowX arm, and 73.8% in simulation. See Figure 5 for a breakdown of success rates by skills. These results indicate that ZeroMimic is capable of distilling a diverse set of unique skills from web videos. The results are best viewed in the videos presented on our website.

The slide closing/opening and hinge closing/opening skills require grasping the object handle and reasoning about the object articulation affordances. Articulated objects often have handles of different shapes, sizes, and orientations, which our grasping module needs to appropriately adjust to. Furthermore, slide and hinge skills require different movements with respect to the object's articulation axis: translation and rotation, respectively. Hinge skills in particular require the model to determine if an object should be manipulated clockwise or counterclockwise along the axis (e.g. the left and right door of a cupboard).

For the picking and placing skills, ZeroMimic needs to reason about the target object pose provided in the goal image. Picking has the elevated complexity of grasping the object first, resulting in worse performance than the placing skill.

ZeroMimic is also able to learn to use tools and perform pouring and cutting skills at a high level. Pouring requires reasoning about the target pour location and subsequently moving towards the location while rotating the object along the correct axis. Similarly, cutting requires reasoning about the cutting angle on the object given the initial knife pose and the target object pose. Afterwards, the robot needs to rotate the knife to align the edge and the object at the optimal angle and perform a swift downward motion. Interestingly, we observe that instead of always cutting straight down, which may result in an undesired cut on the object (e.g. slicing a vertically placed banana along its longer axis), our model is aware of the relative placements between the knife and the object and it learns to adjust its motion plan properly.

Stirring is arguably the hardest skill to learn since it requires a particular set of motions where the translational position remains roughly the same but the orientation continuously moves in the same direction. Also, there is not much information about the desired motions in the goal image. In evaluation, ZeroMimic can rotate a ladle and stir solid food objects as well as liquid in a container without excessive translational movements.

Having been trained exclusively on in-the-wild human videos, ZeroMimic demonstrates remarkable generalization when deployed across object instances, categories, scenes, and robot embodiments. Notably, it successfully executes tasks involving object categories unseen in the human training data, such as *Pour Salt from Spoon into Pan* and *Cut Cake*. ZeroMimic skill policies perform comparably on the WidowX and Franka arms for most tasks, except for the stirring skill, which is challenging due to the limited workspace of the smaller WidowX robot. Additionally, ZeroMimic exhibits robustness to the real-to-sim gap, with no significant performance differences observed between real-world and simulation experiments.

Task	ZeroMimic	ReKep [20]
Open Drawer	**8/10**	0/10
Close Drawer	**6/10**	**6/10**
Place Pasta Bag into Drawer	**8/10**	4/10
Pour Food from Bowl into Pan	**8/10**	0/10

Table 3: Success rates for different tasks using ZeroMimic and ReKep.

4.4. Comparison to Other Zero-Shot Robotic System

Concurrent work ReKep [20] optimizes keypoint-based constraints generated by vision-language models (VLMs) to achieve zero-shot robotic behavior. Similar to ZeroMimic, it does not require task-specific training or environment models. To compare ZeroMimic to ReKep, we perform real-world experiments on 4 tasks using the Franka robot in a kitchen environment (Figure 8a). The *Open Drawer* and *Close Drawer* tasks involve reasoning about the drawer's movement and articulation. The *Place Pasta Bag into Drawer* task requires spatial reasoning to understand the relationships between objects. The *Pour Food from Bowl into Pan* task demands reasoning about both object rotation and spatial relations.

Table 3 show the results. We observe that the failure cases of ReKep mostly stem from two issues: the vision module generates inaccurate keypoints or associates incorrect keypoints with target objects, and the VLM generates incorrect keypoint-based constraints due to its limited spatial reasoning capabilities. For more information about our implementation of ReKep and a detailed analysis of its failure cases, see Appendix 10.

4.5. ZeroMimic Failure Breakdown

We investigate the system errors by examining the intermediate outputs of various modules and manually recording the cause of failure for each unsuccessful trial and aggregating their likelihood. Out of 87 failure trials in our real-world experiments, 31.1% failed at the AnyGrasp stage, 24.1% failed at the VRB stage, and 44.8% failed at the post-grasp policy stage. We present failure analysis of each module below and several examples of these failures on our website.

AnyGrasp. AnyGrasp is sensitive to point cloud sensing failures. We use the "neural" mode of Zed depth cameras for more accurate and smooth depth estimates; however, performance still degrades with small, reflective objects or under poor lighting conditions (e.g., small shiny drawer handles). Occasionally, AnyGrasp also generates incorrect or unreachable grasps.

VRB. A common issue with VRB is its difficulty in predicting appropriate contact locations on large furniture (e.g. cabinets, refrigerators) and opened articulated objects. Additionally, since VRB internally relies on Grounded SAM [74] for language-based segmentation, segmentation errors can directly result in its failures.

Post-grasp policy. The post-grasp policy is sometimes sensitive to camera-robot relative positional configurations, especially if they deviate significantly from an egocentric perspective, since the policy models are trained on egocentric human data. Additionally, action reconstructions from human videos are inherently noisy, causing difficulties with fine-grained tasks such as pouring from a spoon or cutting small food items.

5. Conclusions & Limitations

We have presented ZeroMimic, a first step towards distilling zero-shot deployable a repertoire of robotic manipulation skill policies from purely in-the-wild human videos, each validated in real scenes with real objects. Presently, ZeroMimic exploits a simplified pre-grasp / post-grasp skill stricture, directly retargets human wrist movements to the robot without accounting for morphological differences, does not learn any in-hand manipulations, non-prehensile interactions, or gripper release, and does not handle tasks requiring two arms. Nevertheless, we have shown that it already suffices to learn many useful skills. ZeroMimic builds on the very best current models and hardware for grasp generation, interaction affordance prediction, depth sensing, and hand detection. We have shown that it is limited by their performance; as those models continue to improve, they will further increase the viability of our approach. Finally, we have trained ZeroMimic on a relatively modest 100 hrs of egocentric daily activity dataset, and expanding this to include larger datasets such as Ego-4D [25] and beyond could help to generate a more comprehensive and performant repository of web-distilled skill policies.

References

[1] T. Z. Zhao, V. Kumar, S. Levine, and C. Finn, "Learning fine-grained bimanual manipulation with low-cost hardware," 2023. 1, 2, 3, 6

[2] C. Chi, S. Feng, Y. Du, Z. Xu, E. Cousineau, B. Burchfiel, and S. Song, "Diffusion policy: Visuomotor policy learning via action diffusion," in *Proceedings of Robotics: Science and Systems (RSS)*, 2023. 6, 3

[3] Z. Fu, T. Z. Zhao, and C. Finn, "Mobile aloha: Learning bimanual mobile manipulation with low-cost whole-body teleoperation," in *Conference on Robot Learning (CoRL)*, 2024. 2

[4] T. Z. Zhao, J. Tompson, D. Driess, P. Florence, S. K. S. Ghasemipour, C. Finn, and A. Wahid, "ALOHA unleashed: A simple recipe for robot dexterity," in *8th Annual Conference on Robot Learning*, 2024. [Online]. Available: https://openreview.net/forum?id=gvdXE7ikHI 1

[5] D. Damen, H. Doughty, G. M. Farinella, S. Fidler, A. Furnari, E. Kazakos, D. Moltisanti, J. Munro, T. Perrett, W. Price, and M. Wray, "Scaling egocentric vision: The epic-kitchens dataset," in *European Conference on Computer Vision (ECCV)*, 2018. 1, 2, 3, 4

[6] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, J. Dabis, C. Finn, K. Gopalakrishnan, K. Hausman, A. Herzog, J. Hsu, J. Ibarz, B. Ichter, A. Irpan, T. Jackson, S. Jesmonth, N. Joshi, R. Julian, D. Kalashnikov, Y. Kuang, I. Leal, K.-H. Lee, S. Levine, Y. Lu *et al.*, "Rt-1: Robotics transformer for real-world control at scale," in *arXiv preprint arXiv:2212.06817*, 2022. 2

[7] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, X. Chen, K. Choromanski, T. Ding, D. Driess, A. Dubey, C. Finn, P. Florence, C. Fu, M. G. Arenas, K. Gopalakrishnan, K. Han, K. Hausman, A. Herzog, J. Hsu, B. Ichter, A. Irpan, N. Joshi, R. Julian, D. Kalashnikov, Y. Kuang, I. Leal, L. Lee, T.-W. E. Lee, S. Levine, Y. Lu *et al.*, "Rt-2: Vision-language-action models transfer web knowledge to robotic control," in *arXiv preprint arXiv:2307.15818*, 2023.

[8] O. X.-E. Collaboration, A. O'Neill, A. Rehman, A. Gupta, A. Maddukuri, A. Gupta, A. Padalkar, A. Lee, A. Pooley, A. Gupta, A. Mandlekar, A. Jain, A. Tung, A. Bewley, A. Herzog, A. Irpan, A. Khazatsky, A. Rai, A. Gupta, A. Wang, A. Kolobov, A. Singh *et al.*, "Open X-Embodiment: Robotic learning datasets and RT-X models," https://arxiv.org/abs/2310.08864, 2023.

[9] Octo Model Team, D. Ghosh, H. Walke, K. Pertsch, K. Black, O. Mees, S. Dasari, J. Hejna, C. Xu, J. Luo, T. Kreiman, Y. Tan, L. Y. Chen, P. Sanketi, Q. Vuong, T. Xiao, D. Sadigh, C. Finn, and S. Levine, "Octo: An open-source generalist robot policy," in *Proceedings of Robotics: Science and Systems*, Delft, Netherlands, 2024.

[10] H. Bharadhwaj, J. Vakil, M. Sharma, A. Gupta, S. Tulsiani, and V. Kumar, "Roboagent: Generalization and efficiency in robot manipulation via semantic augmentations and action chunking," 2023.

[11] S. Belkhale, T. Ding, T. Xiao, P. Sermanet, Q. Vuong, J. Tompson, Y. Chebotar, D. Dwibedi, and D. Sadigh, "Rt-h: Action hierarchies using language," in *https://arxiv.org/abs/2403.01823*, 2024.

[12] H. Etukuru, N. Naka, Z. Hu, S. Lee, J. Mehu, A. Edsinger, C. Paxton, S. Chintala, L. Pinto, and N. M. M. Shafiullah, "Robot utility models: General policies for zero-shot deployment in new environments," 2024.

[13] M. Kim, K. Pertsch, S. Karamcheti, T. Xiao, A. Balakrishna, S. Nair, R. Rafailov, E. Foster, G. Lam, P. Sanketi, Q. Vuong, T. Kollar, B. Burchfiel, R. Tedrake, D. Sadigh, S. Levine, P. Liang, and C. Finn, "Openvla: An open-source vision-language-action model," *arXiv preprint arXiv:2406.09246*, 2024.

[14] K. Black, N. Brown, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, L. Groom, K. Hausman, B. Ichter *et al.*, "π_0: A vision-language-action flow model for general robot control," *arXiv preprint arXiv:2410.24164*, 2024. 2

[15] J. Liang, W. Huang, F. Xia, P. Xu, K. Hausman, B. Ichter, P. Florence, and A. Zeng, "Code as policies: Language model programs for embodied control," in *arXiv preprint arXiv:2209.07753*, 2022. 2

[16] H. Huang, F. Lin, Y. Hu, S. Wang, and Y. Gao, "Copa: General robotic manipulation through spatial constraints of parts with foundation models," *arXiv preprint arXiv:2403.08248*, 2024.

[17] S. Nasiriany, F. Xia, W. Yu, T. Xiao, J. Liang, I. Dasgupta, A. Xie, D. Driess, A. Wahid, Z. Xu, Q. Vuong, T. Zhang, T.-W. E. Lee, K.-H. Lee, P. Xu, S. Kirmani, Y. Zhu, A. Zeng, K. Hausman, N. Heess, C. Finn, S. Levine, and B. Ichter, "Pivot: Iterative visual prompting elicits actionable knowledge for vlms," 2024.

[18] T. Kwon, N. Di Palo, and E. Johns, "Language models as zero-shot trajectory generators," *IEEE Robotics and Automation Letters*, 2024.

[19] H. Singh, R. J. Das, M. Han, P. Nakov, and I. Laptev, "Malmm: Multi-agent large language models for zero-shot robotics manipulation," *arXiv preprint arXiv:2411.17636*, 2024.

[20] W. Huang, C. Wang, Y. Li, R. Zhang, and L. Fei-Fei, "Rekep: Spatio-temporal reasoning of relational keypoint constraints for robotic manipulation," *arXiv preprint arXiv:2409.01652*, 2024. 6, 7, 3

[21] S. Patel, X. Yin, W. Huang, S. Garg, H. Nayyeri, L. Fei-Fei, S. Lazebnik, and Y. Li, "A real-to-sim-to-real approach to robotic manipulation with vlm-generated iterative keypoint rewards," *arXiv preprint arXiv:2502.08643*, 2025. 2

[22] R. Goyal, S. Ebrahimi Kahou, V. Michalski, J. Materzynska, S. Westphal, H. Kim, V. Haenel, I. Fruend, P. Yianilos, M. Mueller-Freitag *et al.*, "The" something something" video database for learning and evaluating visual common sense," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 5842–5850. 2

[23] D. Shan, J. Geng, M. Shu, and D. Fouhey, "Understanding human hands in contact at internet scale," in *CVPR*, 2020.

[24] A. Miech, D. Zhukov, J.-B. Alayrac, M. Tapaswi, I. Laptev, and J. Sivic, "HowTo100M: Learning a Text-Video Embedding by Watching Hundred Million Narrated Video Clips," in *ICCV*, 2019.

[25] K. Grauman, A. Westbury, E. Byrne, Z. Chavis, A. Furnari, R. Girdhar, J. Hamburger, H. Jiang, M. Liu, X. Liu *et al.*,

"Ego4d: Around the world in 3,000 hours of egocentric video," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 18 995–19 012. 7

[26] K. Grauman, A. Westbury, L. Torresani, K. Kitani, J. Malik, T. Afouras, K. Ashutosh, V. Baiyya, S. Bansal, B. Boote *et al.*, "Ego-exo4d: Understanding skilled human activity from first-and third-person perspectives," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 19 383–19 400. 2

[27] S. Nair, A. Rajeswaran, V. Kumar, C. Finn, and A. Gupta, "R3m: A universal visual representation for robot manipulation," 2022. 2

[28] Y. Ze, Y. Liu, R. Shi, J. Qin, Z. Yuan, J. Wang, and H. Xu, "H-index: Visual reinforcement learning with hand-informed representations for dexterous manipulation," 2023.

[29] M. K. Srirama, S. Dasari, S. Bahl, and A. Gupta, "Hrp: Human affordances for robotic pre-training," in *Proceedings of Robotics: Science and Systems*, Delft, Netherlands, 2024. 2

[30] 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. 2

[31] 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," 2023. 2

[32] Y. J. Ma, W. Liang, V. Som, V. Kumar, A. Zhang, O. Bastani, and D. Jayaraman, "Liv: Language-image representations and rewards for robotic control," 2023.

[33] T. Ayalew, X. Zhang, K. Y. Wu, T. Jiang, M. Maire, and M. R. Walter, "Progressor: A perceptually guided reward estimator with self-supervised online refinement," 2024. [Online]. Available: https://arxiv.org/abs/2411.17764 2

[34] 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. 2

[35] 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.

[36] S. Bahl, A. Gupta, and D. Pathak, "Human-to-robot imitation in the wild," 2022. 2

[37] S. Kareer, D. Patel, R. Punamiya, P. Mathur, S. Cheng, C. Wang, J. Hoffman, and D. Xu, "Egomimic: Scaling imitation learning via egocentric video," 2024. [Online]. Available: https://arxiv.org/abs/2410.24221

[38] S. Kumar, J. Zamora, N. Hansen, R. Jangir, and X. Wang, "Graph inverse reinforcement learning from diverse videos," *Conference on Robot Learning (CoRL)*, 2022.

[39] C. Wen, X. Lin, J. So, K. Chen, Q. Dou, Y. Gao, and P. Abbeel, "Any-point trajectory modeling for policy learning," 2023. 2

[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] C. Yuan, C. Wen, T. Zhang, and Y. Gao, "General flow as foundation affordance for scalable robot learning," *arXiv preprint arXiv:2401.11439*, 2024.

[42] 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.

[43] M. Levy, S. Haldar, L. Pinto, and A. Shirivastava, "P3-po: Prescriptive point priors for visuo-spatial generalization of robot policies," *arXiv preprint arXiv:2412.06784*, 2024.

[44] I. Guzey, Y. Dai, G. Savva, R. Bhirangi, and L. Pinto, "Bridging the human to robot dexterity gap through object-oriented rewards," *arXiv preprint arXiv:2410.23289*, 2024.

[45] S. Haldar and L. Pinto, "Point policy: Unifying observations and actions with key points for robot manipulation," *arXiv preprint arXiv:2502.20391*, 2025. 2

[46] M. Chang, A. Prakash, and S. Gupta, "Look ma, no hands! agent-environment factorization of egocentric videos," *Advances in Neural Information Processing Systems*, vol. 36, pp. 21 466–21 486, 2023. 2

[47] Y. Ye, X. Li, A. Gupta, S. De Mello, S. Birchfield, J. Song, S. Tulsiani, and S. Liu, "Affordance diffusion: Synthesizing hand-object interactions," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 22 479–22 489.

[48] Y. Ye, P. Hebbar, A. Gupta, and S. Tulsiani, "Diffusion-guided reconstruction of everyday hand-object interaction clips," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2023, pp. 19 717–19 728.

[49] Z. Zhu and D. Damen, "Get a grip: Reconstructing hand-object stable grasps in egocentric videos," *arXiv preprint arXiv:2312.15719*, 2023.

[50] Y. Kuang, J. Ye, H. Geng, J. Mao, C. Deng, L. Guibas, H. Wang, and Y. Wang, "Ram: Retrieval-based affordance transfer for generalizable zero-shot robotic manipulation," *arXiv preprint arXiv:2407.04689*, 2024.

[51] C. Bao, J. Xu, X. Wang, A. Gupta, and H. Bharadhwaj, "Handsonvlm: Vision-language models for hand-object interaction prediction," 2024.

[52] K. Pertsch, R. Desai, V. Kumar, F. Meier, J. J. Lim, D. Batra, and A. Rai, "Cross-domain transfer via semantic skill imitation," *6th Conference on Robot Learning*, 2022.

[53] S. Bahl, R. Mendonca, L. Chen, U. Jain, and D. Pathak, "Affordances from human videos as a versatile representation for robotics," 2023. 2, 3, 5

[54] R. Mendonca, S. Bahl, and D. Pathak, "Structured world models from human videos," 2023. 5

[55] K. Shaw, S. Bahl, and D. Pathak, "Videodex: Learning dexterity from internet videos," in *Conference on Robot Learning*. PMLR, 2023, pp. 654–665.

[56] H. Bharadhwaj, A. Gupta, V. Kumar, and S. Tulsiani, "Towards generalizable zero-shot manipulation via translating human interaction plans," 2023.

[57] H. Bharadhwaj, R. Mottaghi, A. Gupta, and S. Tulsiani, "Track2act: Predicting point tracks from internet videos enables diverse zero-shot robot manipulation," 2024. 2

[58] 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.

[59] H. G. Singh, A. Loquercio, C. Sferrazza, J. Wu, H. Qi, P. Abbeel, and J. Malik, "Hand-object interaction pretraining from videos," 2024. [Online]. Available: https://arxiv.org/abs/2409.08273

[60] S. Ye, J. Jang, B. Jeon, S. Joo, J. Yang, B. Peng, A. Mandlekar, R. Tan, Y.-W. Chao, B. Y. Lin, L. Liden, K. Lee, J. Gao, L. Zettlemoyer, D. Fox, and M. Seo, "Latent action pretraining from videos," 2024. [Online]. Available: https://arxiv.org/abs/2410.11758 2

[61] N. Heppert, M. Argus, T. Welschehold, T. Brox, and A. Valada, "Ditto: Demonstration imitation by trajectory transformation," *arXiv preprint arXiv:2403.15203*, 2024. 2

[62] N. D. Palo and E. Johns, "Dinobot: Robot manipulation via retrieval and alignment with vision foundation models," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2024.

[63] 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.

[64] 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. 2, 3

[65] J. Li, Y. Zhu, Y. Xie, Z. Jiang, M. Seo, G. Pavlakos, and Y. Zhu, "Okami: Teaching humanoid robots manipulation skills through single video imitation," in *8th Annual Conference on Robot Learning (CoRL)*, 2024. 2

[66] G. Chen, M. Wang, T. Cui, Y. Mu, H. Lu, T. Zhou, Z. Peng, M. Hu, H. Li, L. Yuan *et al.*, "Vlmimic: Vision language models are visual imitation learner for fine-grained actions," *Advances in Neural Information Processing Systems*, vol. 37, pp. 77 860–77 887, 2024. 2, 3

[67] H. Bharadhwaj, A. Gupta, S. Tulsiani, and V. Kumar, "Zero-shot robot manipulation from passive human videos," 2023. 3, 5

[68] H.-S. Fang, C. Wang, H. Fang, M. Gou, J. Liu, H. Yan, W. Liu, Y. Xie, and C. Lu, "Anygrasp: Robust and efficient grasp perception in spatial and temporal domains," *IEEE Transactions on Robotics (T-RO)*, 2023. 3, 5

[69] G. Pavlakos, D. Shan, I. Radosavovic, A. Kanazawa, D. Fouhey, and J. Malik, "Reconstructing hands in 3d with transformers," 2023. 3

[70] J. L. Schönberger and J.-M. Frahm, "Structure-from-motion revisited," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 3

[71] V. Tschernezki, A. Darkhalil, Z. Zhu, D. Fouhey, I. Larina, D. Larlus, D. Damen, and A. Vedaldi, "EPIC Fields: Marrying 3D Geometry and Video Understanding," in *Proceedings of the Neural Information Processing Systems (NeurIPS)*, 2023. 3

[72] S. Nasiriany, A. Maddukuri, L. Zhang, A. Parikh, A. Lo, A. Joshi, A. Mandlekar, and Y. Zhu, "Robocasa: Large-scale simulation of everyday tasks for generalist robots," in *Robotics: Science and Systems (RSS)*, 2024. 5, 1

[73] P. Liu, Y. Orru, C. Paxton, N. M. M. Shafiullah, and L. Pinto, "Ok-robot: What really matters in integrating open-knowledge models for robotics," *arXiv preprint arXiv:2401.12202*, 2024. 5

[74] T. Ren, S. Liu, A. Zeng, J. Lin, K. Li, H. Cao, J. Chen, X. Huang, Y. Chen, F. Yan, Z. Zeng, H. Zhang, F. Li, J. Yang, H. Li, Q. Jiang, and L. Zhang, "Grounded sam: Assembling open-world models for diverse visual tasks," 2024. 7

[75] A. Khazatsky, K. Pertsch, S. Nair, A. Balakrishna, S. Dasari, S. Karamcheti, S. Nasiriany, M. K. Srirama, L. Y. Chen, K. Ellis, P. D. Fagan, J. Hejna, M. Itkina, M. Lepert, Y. J. Ma, P. T. Miller, J. Wu, S. Belkhale, S. Dass, H. Ha, A. Jain, A. Lee, Y. Lee, M. Memmel, S. Park, I. Radosavovic, K. Wang, A. Zhan, K. Black, C. Chi, K. B. Hatch, S. Lin, J. Lu, J. Mercat, A. Rehman, P. R. Sanketi, A. Sharma, C. Simpson, Q. Vuong, H. R. Walke, B. Wulfe, T. Xiao, J. H. Yang, A. Yavary, T. Z. Zhao, C. Agia, R. Baijal, M. G. Castro, D. Chen, Q. Chen, T. Chung, J. Drake, E. P. Foster, J. Gao, D. A. Herrera, M. Heo, K. Hsu, J. Hu, D. Jackson, C. Le, Y. Li, K. Lin, R. Lin, Z. Ma, A. Maddukuri, S. Mirchandani, D. Morton, T. Nguyen, A. O'Neill, R. Scalise, D. Seale, V. Son, S. Tian, E. Tran, A. E. Wang, Y. Wu, A. Xie, J. Yang, P. Yin, Y. Zhang, O. Bastani, G. Berseth, J. Bohg, K. Goldberg, A. Gupta, A. Gupta, D. Jayaraman, J. J. Lim, J. Malik, R. Martín-Martín, S. Ramamoorthy, D. Sadigh, S. Song, J. Wu, M. C. Yip, Y. Zhu, T. Kollar, S. Levine, and C. Finn, "Droid: A large-scale in-the-wild robot manipulation dataset," 2024. 1, 3

[76] Y. Rong, T. Shiratori, and H. Joo, "Frankmocap: A monocular 3d whole-body pose estimation system via regression and integration," in *IEEE International Conference on Computer Vision Workshops*, 2021. 3

[77] 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. [Online]. Available: https://arxiv.org/abs/2408.00714 4

CVPR
#10

ZeroMimic: Distilling Robotic Manipulation Skills from Web Videos

Supplementary Material

6. Experimental Setup Details

6.1. Real-World Experimental Setup Details

Figure 6: Our Franka hardware setup includes a 7-DOF Franka Emika Panda arm with a Robotiq 2-fingered gripper and a Zed 2 stereo camera mounted on the base.

Figure 7: Our WidowX hardware setup includes a 6-DOF Trossen Robotics WidowX 250 S arm attached to a table with a 2-fingered gripper and an Intel RealSense Depth Camera D435.

Our Franka experiments uses the hardware setup in Fig 6, which is similar to that used in prior works [75]. We use a Franka Emika Panda arm with a Robotiq two-finger gripper mounted on a mobile base, which we use to drag the robot across various scenes. We use a Zed 2 stereo camera mounted on the base to capture RGB and depth images.

(a) Levine Hall Kitchen (b) Grasp Lab Kitchen (c) Towne Hall Kitchen (d) Table Top 1 (e) Table Top 2 (f) Table Top 3

Figure 8: Real-world environments used in our experiments: (a-c) Various kitchen environments across different buildings, (d-f) Different tabletop setups. We perform our Franka experiments using setups (a-e), and our WidowX experiments using setup (f).

Our WidowX experiments uses the hardware setup in Fig 7. The WidowX arm is attached to a stationary table. An Intel RealSense Depth Camera D435 is mounted on a tripod beside the table to capture RGB and depth images.

Figure 8 shows our real-world experimental environments. We conducted experiments in three different kitchen environments (Figures 8a-8c) and three tabletop setups (Figures 8d-8f). For the Franka robot experiments, we used environments (a)-(e), moving the robot between different buildings. The WidowX robot experiments were conducted using the stationary tabletop setup shown in (f).

6.2. Simulation Experimental Setup Details

Figure 9 shows our simulation setup in RoboCasa [72]. We use a Franka Emika Panda arm with a two-finger gripper in a simulated kitchen layout. We perform 20 trials for each task, varying the camera position, randomizing the robot's position, altering the background object instances and their positions, and selecting a random kitchen style from the 12 available options. Each kitchen style features unique textures, distractor objects, and fixture attributes, such as cabinet and drawer handle types. Figure 10 are example

Real-World Results

Skill	Robot	Object Category	Scenario	Success Rate (%)
Hinge Opening	Franka	Cupboard	Levine Hall Kitchen	6/10
	Franka	Cupboard	Table Top 1	6/10
	Franka	Cupboard	Table Top 2	8/10
	Franka	Fridge	GRASP Lab Kitchen	8/10
	WidowX	Cupboard	Table Top 3	9/10
Hinge Closing	Franka	Cupboard	Levine Hall Kitchen	4/10
	Franka	Cupboard	Table Top 1	8/10
	Franka	Cupboard	Table Top 2	10/10
	Franka	Fridge	GRASP Lab Kitchen	8/10
	WidowX	Cupboard	Table Top 3	7/10
Slide Opening	Franka	Drawer	Levine Hall Kitchen	8/10
	Franka	Drawer	Towne Hall Kitchen	10/10
Slide Closing	Franka	Drawer	Levine Hall Kitchen	6/10
	Franka	Drawer	Towne Hall Kitchen	10/10
Pouring	Franka	Water from Bowl into Sink	Levine Hall Kitchen	8/10
	Franka	Food from Bowl into Pan	Levine Hall Kitchen	8/10
	Franka	Salt from Spoon into Pan	Levine Hall Kitchen	4/10
	WidowX	Water from Cup into Pot	Table Top 3	7/10
Picking	Franka	Can	Levine Hall Kitchen	7/10
	Franka	Banana	Levine Hall Kitchenn	4/10
	Franka	Marker	Table Top 1	6/10
Placing	Franka	Spoon	Levine Hall Kitchen	10/10
	Franka	Pasta Bag into Drawer	Levine Hall Kitchen	4/10
Cutting	Franka	Tofu	Levine Hall Kitchen	8/10
	Franka	Banana	Levine Hall Kitchen	8/10
	Franka	Cake	Levine Hall Kitchen	8/10
Stirring	Franka	Food in Pan	Levine Hall Kitchen	6/10
	Franka	Pasta in Water	Levine Hall Kitchen	8/10
	Franka	Water in Pan	Table Top 1	6/10
	WidowX	Food in Pot	Table Top 3	3/10
9 Skills	**2 Robots**	**18 Categories**	**30 Total Instances**	**71.0%**

Simulation Results

Skill	Robot	Object Category	Scenario	Success Rate (%)
Hinge Opening	Franka	Cupboard	Simulated Kitchen	15/20
Hinge Closing	Franka	Cupboard	Simulated Kitchen	12/20
Slide Opening	Franka	Drawer	Simulated Kitchen	17/20
Slide Closing	Franka	Drawer	Simulated Kitchen	15/20
4 Skills	**1 Robots**	**2 Categories**	**4 Total Instances**	**73.8%**

Table 4: Summary of skills, robots, object categories, scenarios, and success rates for real-world and simulation results.

images of different kitchen scene styles. The success of a trial is evaluated based on the specific success conditions defined for each task provided by RoboCasa.

7. Detailed Breakdown of ZeroMimic Zero-Shot Deployment Performances

Table 4 provides a comprehensive overview of ZeroMimic's performance across both real-world and simulation environments. The results are categorized by skills, robots, object categories, and scenarios, offering insight into the system's versatility and adaptability.

In the real-world evaluation, we assessed 9 skills across 2 robots and 18 object categories, spanning 30 distinct scenarios. These evaluations resulted in an overall success rate of 71.0%. Additionally, we evaluated 4 skills in a controlled simulated kitchen environment using one robot across two object categories, totaling four distinct scenarios. This simulation study achieved an overall success rate of 73.8%.

Figure 9: Our RoboCasa simulation setup includes a 7-DOF Franka Emika Panda arm with a 2-fingered gripper.

Figure 10: RoboCasa environment images showcasing different setups and configurations. Each image corresponds to a different kitchen environment style.

8. Ablation Experiment Details

In Section 4.2, the *Open Drawer* task is performed with the "slide opening" skill policy, and the *Open Cupboard* task is performed with the "hinge opening" skill policy.

8.1. Grasping Methods Ablation Details

Ours w/o grasp model is an ablated variant of ZeroMimic where the end effector is moved to the 2D contact point proposed by VRB [53] lifted to 3D with depth, and the gripper is then closed. In this experiment, since VRB does not output orientation, we use the gripper's initial orientation for the grasp pose.

8.2. Wrist Post-Grasp Policy Ablation

In our strengthened version of H2R (**ours w/o SfM**), we remove camera extrinsics and intrinsics when processing our training data. As a result, the 3D location of the wrist is only represented by its pixel coordinate and hand size, the output of depth-ambiguous monocular hand reconstruction methods [69, 76].

VRB produces post-contact trajectories only in terms of 2D pixel locations on the image. To execute it on the robot, we convert VRB's 2D outputs to 6D using the following procedure: we sample a target end-point depth at random and interpolate the waypoints while fixing the gripper orientation as the initial post-grasp gripper orientation throughout the trajectory.

9. Additional Post-Grasp Module Ablation Experiments

To gain insight into what is critical to learning to predict wrist trajectories from web videos, we teleoperate the robot to obtain a successful grasp and then evaluate a number of alternative post-grasp trajectory generation options and present our findings in this section.

Imitation Policy Architecture We compare ACT [1] and Diffusion Policy [2], two popular imitation learning policy classes, for training our post-grasp policy on EpicKitchens. As illustrated in Table 5, they perform similarly when evaluated in the real world with the Franka robot. ACT performs slightly better on skills that mostly require gripper translation, while Diffusion Policy is marginally better at more rotation-heavy tasks. For consistency, we use ACT for all of our other experiments and ablations.

Method	Open drawer	Open cupboard	Pour water
ACT	**10/10**	**8/10**	7/10
DiffPo	8/10	**8/10**	**9/10**

Table 5: Success rates for different post-grasp policies after a successful grasp.

Relative vs. Absolute Action Representation For both the translation (**T**) dimensions and the orientation (**O**) dimensions, we compare training an ACT model with absolute and relative action representations, resulting in four variants: absT+absO, absT+relO, relT+absO, and relT+relO. Evaluating on the real-world "pour water" task with the Franka arm, their respective success rates are 1/10, 3/10, 2/10, and 7/10, indicating that relT+relO performs significantly better than other variants. We hypothesize that the orientation distribution shift from the human hand to the gripper as well as discontinuity in orientation space from $-\pi$ to π makes it harder for the model to learn meaningful absolute orientation representation.

10. ReKep Baseline Details

10.1. ReKep Implementation Details

We adapted the publicly released simulation code of ReKep [20] for OmniGibson to integrate with our real-world Franka arm setup [75]. To evaluate ReKep as a zero-shot system without human intervention, we use its ”Auto” mode, which automatically generates keypoints and constraints, instead of the ”Annotation” mode, which requires manual annotation for both.

As part of the adaptation, we rewrote the environment module, including the robot controller and keypoint registration components. To ensure optimal performance and to use ReKep's steelman version as a competitive baseline, we rely on teleoperated grasping for its grasping module, effectively minimizing grasp failures. In the perception module, we replace the ground-truth masks provided by the simulator with those generated by the Segment Anything Model 2 (SAM2) [77]. These masks are filtered using area upper and lower bounds to ensure accuracy. Additionally, we modify ReKep's original k-means and mean-shift clustering algorithms to refine the generated keypoints, providing the VLM with cleaner input data for generating keypoint constraints. Lastly, we replace the simulator's ground-truth depth data with depth data from a ZED 2 Stereo Camera. We use its neural depth mode and apply band filtering to improve the accuracy and reliability of depth values.

10.2. ReKep Failure Cases

We present specific examples and a detailed analysis of ReKep's failure cases across three tasks with low success rates: *Open Drawer*, *Place Pasta Bag into Drawer*, and *Pour Food from Bowl into Pan*.

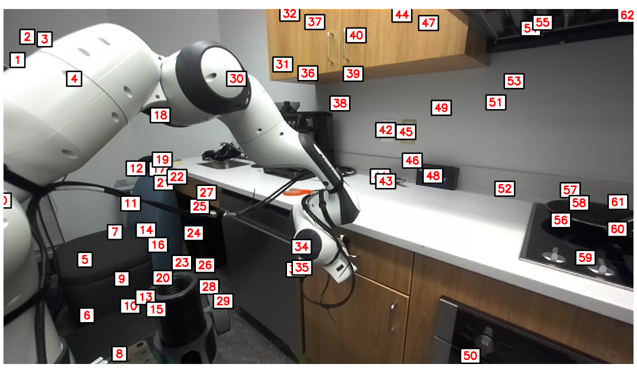


Figure 11: Keypoints proposed by ReKep for the *Open Drawer* task.

```
1  def path_constraint1(end_effector, keypoints):
2      """The robot must still be grasping the drawer
3      handle (keypoint 35)."""
4      handle_position = keypoints[35]
5      cost = np.linalg.norm(end_effector -
        handle_position)
6      return cost
7
8  def subgoal_constraint1(end_effector, keypoints):
9      """The drawer handle (keypoint 35) should be
        displaced
10     outward by 10cm along the x-axis."""
11     handle_position = keypoints[35]
12     offsetted_position = handle_position + np.array
        ([-0.1, 0, 0])
13     cost = np.linalg.norm(handle_position -
        offsetted_position)
14     return cost
```

Code Snippet 1: The constraints generated by ReKep for the *Open Drawer* task instruct the end effector to move leftward in the camera frame (Line 12). However, this direction deviates from the drawer's actual outward articulation axis.

Open Drawer Failures in the *Open Drawer* task arise because the VLM struggles with identifying the drawer's articulation axis in the camera frame, causing the gripper to become stuck. Figure 11 illustrates the keypoints proposed by ReKep, while Code Snippet 1 presents the corresponding constraints generated by ReKep. These constraints direct the end effector to move 10 cm along the negative x-axis (leftward) in the camera frame. However, the actual outward articulation axis of the drawer corresponds to $-x$ (left), $+y$ (down), and $-z$ (towards the screen) in the camera frame. As a result, the actions generated by ReKep's constraints cause the gripper to become stuck, despite its attempts to move.

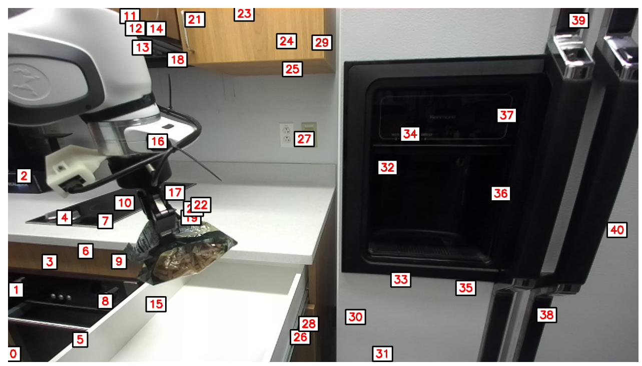


Figure 12: Keypoints proposed by ReKep for the *Place Pasta Bag into Drawer* task.

```
def path_constraint1(end_effector, keypoints):
    """
    Ensure the robot is still grasping the pasta bag
     during the movement.
    The cost is the Euclidean distance between the
     end-effector and the pasta bag's keypoint (
     keypoint 22).
    """
    pasta_bag_keypoint = keypoints[22]
    cost = np.linalg.norm(end_effector -
     pasta_bag_keypoint)
    return cost

def subgoal_constraint1(end_effector, keypoints):
    """
    Ensure the pasta bag is inside the drawer.
    The cost is the Euclidean distance between the
     pasta bag's keypoint (keypoint 22)
    and the drawer's keypoint (keypoint 6).
    """
    pasta_bag_keypoint = keypoints[22]
    drawer_keypoint = keypoints[6]
    cost = np.linalg.norm(pasta_bag_keypoint -
     drawer_keypoint)
    return cost
```

Code Snippet 2: For the *Place Pasta Bag into Drawer* task, ReKep generates constraints based on incorrectly identified keypoints. Specifically, it misclassifies keypoint 22, a background keypoint, as the pasta keypoint, and keypoint 6, another background keypoint, as the drawer keypoint. See Lines 16-17 and Figure 12 for the misclassified keypoints.

Place Pasta Bag into Drawer Figure 12 illustrates the keypoints proposed by ReKep, while Code Snippet 2 presents the corresponding constraints. The keypoint proposal reveals that ReKep's vision module struggles to generate a reliable keypoint on the inside of an empty drawer. Additionally, ReKep projects 3D keypoints onto 2D images, which can result in spatially close keypoints overlapping and cause errors in the VLM's keypoint selection. For example, it identifies a keypoint near the edge of the pasta bag but slightly outside its actual boundary as belonging to the bag. This misplacement leads to the keypoint's depth value being incorrectly interpreted as the larger background depth value. Additionally, it sometimes associates nearby background keypoints with the drawer. By generating constraints based on these misidentified keypoints, ReKep produces ineffective movement instructions for the end effector, ultimately resulting in task failure.

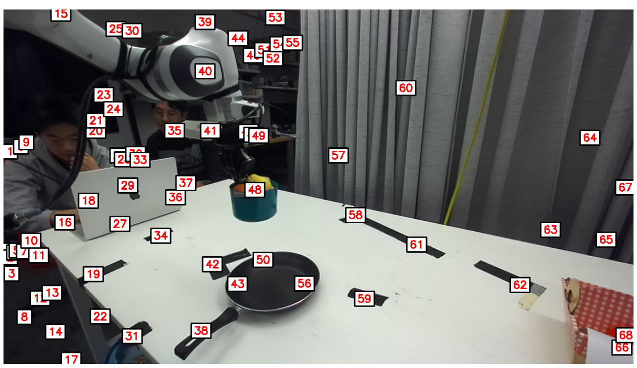


Figure 13: Keypoints proposed by ReKep for the *Pour Food from Bowl into Pan* task.

```
1  def path_constraint1(end_effector, keypoints):
2      """
3      Ensure the robot continues to hold the bowl
        during the pouring process.
4      This can be achieved by keeping the end-effector
        aligned with the bowl's keypoint (e.g., keypoint
         48).
5      """
6      cost = np.linalg.norm(end_effector - keypoints
        [48])
7      return cost
8
9  def subgoal_constraint1(end_effector, keypoints):
10     """
11     Ensure the bowl is tilted to pour the object into
         the pot.
12     This can be achieved by ensuring the vector
        formed by two keypoints on the bowl (e.g.,
        keypoints 48 and 49)
13     is at a specific angle with respect to the z-axis.
14     """
15     bowl_vector = keypoints[49] - keypoints[48]
16     z_axis = np.array([0, 0, 1])
17     angle = np.arccos(np.dot(bowl_vector, z_axis) / (
        np.linalg.norm(bowl_vector) * np.linalg.norm(
        z_axis)))
18     desired_angle = np.pi / 4  # Tilt the bowl by 45
        degrees
19     cost = np.abs(angle - desired_angle)
20     cost = np.linalg.norm(bowl_vector)
21     return cost
```

Code Snippet 3: The constraints generated by ReKep for the *Pour Food from Bowl into Pan* task ensure that the bowl is tilted at an angle of 45° with respect to the z-axis to facilitate pouring (Line 18). However, this angle is insufficient to effectively pour the food out of the bowl.

Pour Food from Bowl into Pan Figure 13 illustrates the keypoints proposed by ReKep, while Code Snippet 3 presents the corresponding constraints. In the pouring task, while ReKep correctly establishes a rotation constraint, it underestimates the numerical value of the required rotation. As a result, the bowl is only slightly tilted at 45°, failing to achieve the intended pouring motion to empty the bowl. While ReKep demonstrates the Pour Tea task in its paper, the prompt used for the VLM in the publicly released implementation includes helpful guidance on constraint construction for this task, such as suggesting that "the teapot must remain upright to avoid spilling". This additional guidance may have enhanced ReKep's performance on the task. While pouring tea requires only a slight tilt, pouring food from a bowl into a pan demands a significantly larger tilt, something the VLM fails to reason about effectively.