DexWild: Dexterous Human Interactions for In-the-Wild Robot Policies

Tony Tao*, Mohan Kumar Srirama*, Jason Jingzhou Liu, Kenneth Shaw, Deepak Pathak
Carnegie Mellon University

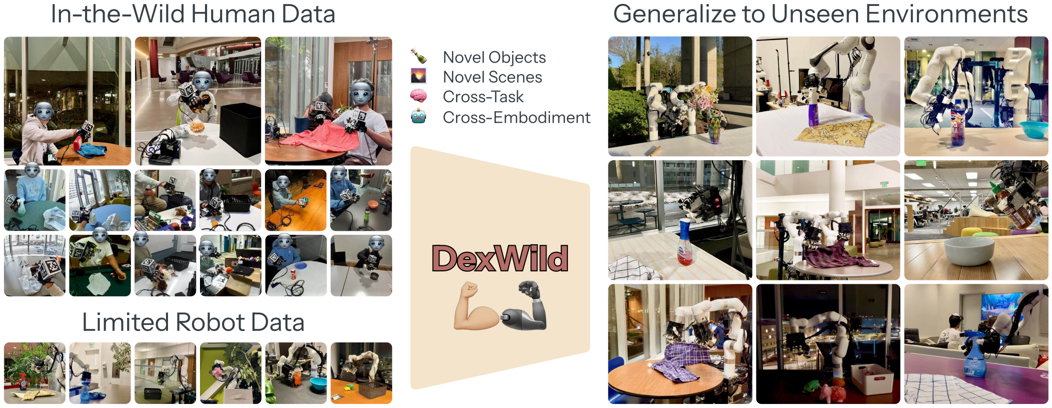


Figure 1: **DexWild** enables dexterous policies to generalize to new objects, scenes, and embodiments. This is achieved by leveraging large-scale, real-world human embodiment data collected in many scenes and co-trained with a smaller robot embodiment dataset for grounding.

Abstract: Large-scale, diverse robot datasets have emerged as a promising path toward enabling dexterous manipulation policies to generalize to novel environments, but acquiring such datasets presents many challenges. While teleoperation provides high-fidelity datasets, its high cost limits its scalability. Instead, what if people could use their own hands, just as they do in everyday life, to collect data? In DexWild, a diverse team of data collectors uses their hands to collect hours of interactions across a multitude of environments and objects. To record this data, we create DexWild-System, a low-cost, mobile, and easy-to-use device. The DexWild learning framework co-trains on both human and robot demonstrations, leading to improved performance compared to training on each dataset individually. This combination results in robust robot policies capable of generalizing to novel environments, tasks, and embodiments with minimal additional robot-specific data. Experimental results demonstrate that DexWild significantly improves performance, achieving a 68.5% success rate in unseen environments—nearly four times higher than policies trained with robot data only—and offering 5.8× better cross-embodiment generalization. Video results, codebases, and instructions at https://dexwild.github.io

Keywords: Imitation Learning, Foundation Models, Grasping & Manipulation

1 Introduction

Roboticists have long dreamed of creating robots that can perform tasks with the same dexterity and adaptability as humans. While there have been many breakthroughs in large language models (LLMs) [53, 51, 3] and vision language models (VLMs) [24, 48], the key to their success lies in harnessing vast datasets. Robotics faces a critical hurdle: large-scale, diverse robot datasets needed to train foundation models do not yet exist.

RSS 2025 Workshop on Whole-Body Control and Bimanual Manipulation (RSS2025WCBM).

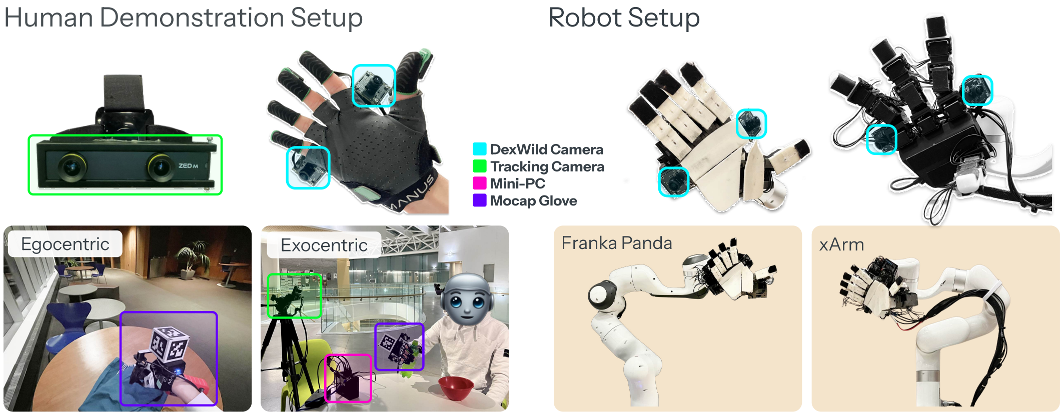


Figure 2: **Left:** DexWild efficiently capture high-fidelity data using an individual's own hands across various environments. **Right:** Robot hands are equipped with cameras aligned with the human cameras. We test DexWild on two distinct robot hands and robot arms.

In recent years, a key approach to collecting robot datasets has been through teleoperation, which provides high-precision, high-quality action data that a policy can directly train on. [8, 21, 54]. However, gathering data in diverse environments presents challenges such as physically relocating the robot to each new location and requiring multiple trained operators.

Another approach is to leverage internet-scale video data, which provide vast and diverse visual grounding in real-world environments [15, 10]. However, publicly available videos often lack the fine-grained accuracy needed to capture detailed hand states because vision-based body detection modules are noisy and unreliable. Additionally, these videos are not inherently structured with categorized episodes for task-specific learning, further complicating their application in robotics. [18, 1, 40]. While some data collection efforts exist with more accurate and structured data, [60, 2], they do not have enough environment diversity.

To overcome these barriers, some have explored collecting accurate in-the-wild human demonstrations by equipping users with a wearable gripper that directly maps their hand movements to robot actions [7]. However, this approach is cumbersome, ill-suited for natural, everyday interactions, and constrains the collected data to a specific embodiment. Other works [55] propose using dexterous hands and gloves, but they do not scale to in-the-wild environments.

In this paper, we present DexWild, a system that enables effective learning of robust dexterous manipulation policies through co-training on human and robot demonstrations. Our key contributions include:

1. **Scalable Data Collection System**: A novel human-embodiment DexWild-System that enables untrained operators to quickly collect 9,290 demonstrations across 93 diverse environments, achieving 4.6× speedup over conventional robot-based methods
2. **Efficient Co-training Framework**: An approach that optimally combines human and robot demonstrations, significantly improving policy generalization to achieve 68.5% success rate in novel environments, nearly four times higher than robot-only policies.
3. **Strong Cross Embodiment and Cross Task Performance**: Our data collection system combined with our co-training framework achieves of 5.8× improvement in cross-embodiment transfer over baselines and effective skill transfer across tasks.

2 DexWild

We introduce DexWild-System, a user-friendly, high-fidelity platform for efficiently gathering natural human hand demonstrations across diverse real-world settings. Compared to traditional teleoperation-based approaches, DexWild-System enables 4.6× faster data acquisition at scale.

Building on this system, we propose DexWild, an imitation learning framework that co-trains on large-scale DexWild-System human demonstrations alongside a small number of robot demonstrations. This approach combines the diversity and richness of human interactions with the grounding of the robot embodiment, enabling policies to robustly generalize across new objects, environments, and embodiments. Figure 1 displays our high level approach.

A scalable data collection system for dexterous robot learning must enable natural, efficient, and high-fidelity collection across diverse environments. To this end, we design DexWild-System: a portable, user-friendly system that captures human dexterous behavior with minimal setup and training. We aim to create an intuitive hardware interface that mirrors how humans naturally interact with the world.

DexWild-System is designed around three core objectives:

- **Portability:** Allow rapid, large-scale data collection across diverse environments without requiring complex calibration procedures.
- **High Fidelity:** Accurately capture fine-grained hand and environment interactions essential for training precise dexterous policies.
- **Embodiment-Agnostic:** Enable seamless retargeting from human demonstrations to a wide variety of robot hands.

Portability:

DexWild-System is lightweight, compact, and can be set up in minutes, making it suitable for untrained users in diverse real-world settings. As shown in Figure 2, it consists of three components: a tracking camera for wrist pose, a battery-powered mini-PC, and a custom pod with a motion-capture glove and palm-mounted cameras.

Unlike traditional motion capture setups [60, 13, 4, 52] that require calibrated, fixed infrastructure, DexWild-System is calibration-free and works in any environment. This is made possible by using a relative state-action representation, allowing free placement of the tracking camera (e.g., egocentric or exocentric).

High Fidelity:

Despite its portability, DexWild-System captures rich, accurate data. We combine motion-capture gloves for precise hand tracking with ArUco-based wrist tracking to avoid failures common in SLAM-based approaches under occlusion or sparse features.

Stereo palm cameras provide high-quality, wide-FOV visual input with minimal motion blur. Their placement enables policies to rely solely on onboard views, removing dependence on static external cameras.

Embodiment-Agnostic:

To ensure compatibility with future robots, we align both observation and action spaces between humans and robots. The palm cameras are positioned to minimize hand visibility and focus on the environment, with mirrored placements between human and robot hands (Figure 3), enabling consistent visual representations across embodiments.

For actions, we follow prior work [17, 44] and retarget human fingertip motions to robot hand kinematics, enabling generalization across hardware platforms.

2.1 Training Data Modalities and Preprocessing

Generalization in dexterous manipulation requires both scale and embodiment grounding. With this goal, DexWild collects two complementary datasets: a large-scale human demonstration dataset D_H using DexWild-System and a smaller teleoperated robot dataset D_R. Human data is easy to collect in the wild and offers task diversity but lacks embodiment alignment. Robot data provides grounding in the robot's observation and action spaces but is limited in scale. We co-train policies using a

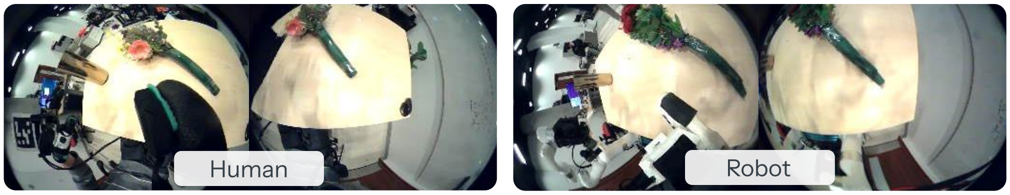
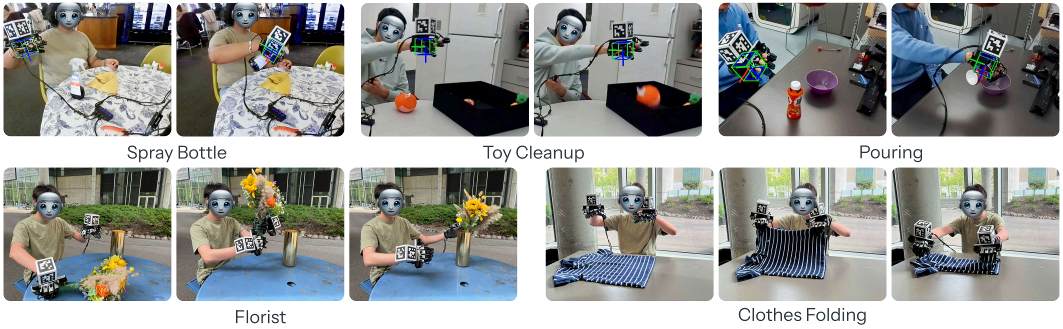


Figure 3: DexWild aligns the visual observations between humans and robots to bridge the embodiment gap. This incentivizes the model to learn a task-centric rather than embodiment-centric representation.


Spray Bottle

Toy Cleanup

Pouring

Florist

Clothes Folding

Figure 4: Using DexWild-System, humans can effortlessly collect accurate data with their own hands across a wide range of environments. This data is directly used to train any robot hand to perform dexterous manipulation in a human-like way in any environment. We validate this approach on five representative tasks.

fixed human-to-robot data ratio (w_h, w_r) to balance diversity and embodiment grounding for robust deployment.

At each iteration, we sample transitions x_h and x_r from D_H and D_R based on the co-training weights. Each transition x_i includes:

- **Observation** o_i: Two synchronized palm images (I_{pinky}, I_{thumb}) and a sequence of relative end-effector displacements $\Delta p_i, \Delta p_{i-\text{step}}, ..., \Delta p_{i-H}$ over horizon H.
- **Action** $a_{i:i+n-1}$: A chunk of n actions, where each a_i is a 26D vector—9D for relative end-effector pose (3D position + 6D orientation) and 17D for robot finger joints.

For bimanual tasks, observations and actions are duplicated, and the inter-hand pose is appended to facilitate coordination.

While our retargeting procedure brings human and robot trajectories into a shared action space, a few additional steps are necessary to make the human and robot datasets compatible for joint training:

- **Action Normalization**: Human and robot actions are normalized separately to correct distribution mismatches.
- **Demo Filtering**: A heuristic-based pipeline removes low-quality human demos, improving dataset quality without manual labeling.

For action space alignment, we build on insights from prior work [17, 44], optimizing robot hand kinematics to match the fingertip positions observed in human demonstrations. This method is general and can work for any robot hand embodiment.

2.2 Policy Training

Through the careful design of our hardware, observation, and action interfaces, we are able to train dexterous robot policies using a simple behavior cloning (BC) objective [31, 37, 36]. To effectively learn from our multimodal, diverse data, our training pipeline leverages large-scale pre-trained visual encoders and shows strong performance across different policy architectures.

Visual Encoder: Training on DexWild data exposes our policy to significant visual diversity—across scenes, objects, and lighting—requiring an encoder that generalizes well to such variability. To address this, we adopt a pre-trained Vision Transformer (ViT) backbone, which has shown superior performance over ResNet-based encoders on in-the-wild manipulation tasks [16, 23].

Policy Class: While several imitation learning architectures have been proposed recently [59, 6], we adopt a diffusion-based policy. Diffusion models are particularly well-suited for dexterous manipulation, as they can capture multi-modal action distributions more effectively than alternatives such as transformers. This capability becomes increasingly important in DexWild, where demonstrations are collected from multiple humans with diverse strategies, resulting in inherently multi-modal behaviors.

Concretely, the training procedure is outlined in Algorithm 1.

Algorithm 1 DexWild Imitation Learning Procedure

```
Require: Human dataset $\mathcal{D}_H$, Robot dataset $\mathcal{D}_R$, Co-training weights $\{\omega_h, \omega_r\}$
Initialize policy $\pi_\theta$ with ViT encoder $\phi_{\text{vit}}$
while not converged do
    Sample a batch of transitions $\{x_h\}, \{x_r\}$ from $\mathcal{D}_H, \mathcal{D}_R$ using weights $\{\omega_h, \omega_r\}$
    for each transition $x_i$ in the batch do
        Extract observation $o_i$
        Encode images: $Z_i = \phi_{\text{vit}}(o_i)$
        Extract ground truth action chunk $a_{i:i+n-1} = \{a_i, \dots, a_{i+n-1}\}$
        Sample noise scale $t \sim \mathcal{U}(1, T)$
        Add noise $\epsilon_t \sim \mathcal{N}(0, \sigma_t)$ to $a_{i:i+n-1}$
        Predict noise $\hat{\epsilon}_\theta = \pi_\theta(Z_i, a_{i:i+n-1} + \epsilon_t, t)$
        Compute diffusion loss $\mathcal{L}_\theta = \|\epsilon_t - \hat{\epsilon}_\theta\|_2^2$
    end for
    Update policy parameters $\theta$
end while
```

An important finding in our training framework is that tuning the human-to-robot data weighting significantly affects real-world performance. We discuss these effects in Section 4.1.

3 Experiments

Our experimental evaluation encompasses extensive real-world deployment across diverse environments and robots, utilizing both human demonstrations and robot teleoperation data. Below, we outline our data collection process, experimental setup, and evaluation tasks.

3.1 Scaling up Data Collection

Our hardware system was deployed to 10 untrained users to collect data across a wide range of real-world environments. The collectors themselves varied in hand sizes and demonstration styles, enabling us to learn from a wide distribution of environments and interactions.

We constructed two datasets through our collection efforts: D_H (human-collected data) and D_R (robot-collected data). The human dataset D_H comprises 9,290 demonstrations across five tasks. The robot dataset D_R includes 1,395 demonstrations. Robot data was collected using an xArm and LEAP hand V2 Advanced. Our training and test objects are detailed in Figure 8 and a detailed breakdown of task dataset sizes in Appendix 6.3

3.2 Evaluation Tasks

We evaluate our approach on five diverse manipulation tasks, each designed to assess specific aspects of dexterous manipulation: functional grasping, long-horizon planning, cross-task transfer, bimanual coordination, and deformable object manipulation. A task visualization is provided in Figure 4.

Full task specifications and scoring criteria for all tasks are provided in Appendix 6.2.

These tasks systematically evaluate DexWilds *functional grasping* capabilities, *generalization* across object types, *transferal* of skills across tasks, *coordination* between arms, and *adaptability* to deformable objects. Success requires the policy to adapt to varying object properties, environmental conditions, and task constraints.

3.3 Evaluation Environments

For robot experiments, we employed an xArm robot and Franka system, both equipped with either LEAP hand or LEAP hand V2 Advanced [38, 41]. Unless explicitly mentioned, xArm and LEAP hand V2 Advanced was used. We evaluate our approach across three scenarios:

1. In-Domain: Environments where robot training data was collected, testing with novel objects
2. In-the-Wild: Environments present in DexWild but absent from robot training data
3. In-the-Wild Extreme: Unseen environments absent from both datasets.

4 Analysis and Results

In our evaluations, we seek to investigate the following key questions:

1. How effectively does DexWild use human data to achieve strong in-the-wild performance?
2. Does DexWild enable policy transfer across tasks and robot embodiments?
3. Does policy performance scale with increasing amounts of DexWild-System data?

4.1 Zero Shot In the Wild Policies w/ DexWild

DexWild enables strong policy generalization in novel scenes. We evaluate policies in environments with increasing novelty to assess their generalization. As shown in Figure 5, policies trained exclusively on robot data perform well in in-domain settings (64.7% success rate) but degrade significantly in more challenging scenarios—in-the-wild (28.5%) and in-the-wild extreme (22.0%). This 36-point performance drop suggests that robot-only policies overfit to environment-specific features and fail to develop robust, transferable representations. In contrast, policies trained only on human data learn high-level object affordances and approach objects reliably, even in complex scenes. However, without robot-specific action grounding, they struggle to execute precise manipulation, resulting in poor performance across all scenarios (3.6% in-domain, 7.3% in-the-wild).

To combine the strengths of both modalities, we adopt a co-training strategy—jointly training on both robot and human data—a method validated in prior works [8, 49, 21, 20, 32]. This encourages the policy to learn task-relevant features rather than overfitting to specific embodiments or environments. We experiment with different **robot-to-human** data ratios (1:1 to 1:5) per training batch. Our empirical analysis reveals that a 1:2 ratio yields optimal performance across all scenarios:

1. In Domain: 79.8% vs. 64.7% (robot-only)
2. In-the-wild: 75.1% vs. 28.5% (robot-only)
3. In-the-wild Extreme: 62.7% vs. 22.0% (robot-only)

DexWild extends to complex bimanual coordination tasks. To evaluate whether DexWild generalizes beyond single-arm tasks, we test it on bimanual tasks that demand precise coordination between two hands. We compare co-trained policies (1:2 ratio) against robot-only policies in in-the-wild extreme settings. DexWild policies achieve a strong 68.1% average success rate, compared to just 13% for the robot-only baseline.

4.2 Robust Cross-Task and Cross-Embodiment Generalization

DexWild enables transfer of low-level skills across tasks. Many manipulation tasks share foundational motor skills—such as lifting, orienting, and rotating objects—which opens the door to skill reuse across related tasks. We evaluate this form of cross-task transfer using the *pouring* task, which shares many motion primitives with the *spray* task. Crucially, we use no robot data for pouring and instead combine human (DexWild-System) demonstrations of pouring with robot demonstrations from spraying. This setup enables **zero-shot generalization** to pouring in in-the-wild extreme

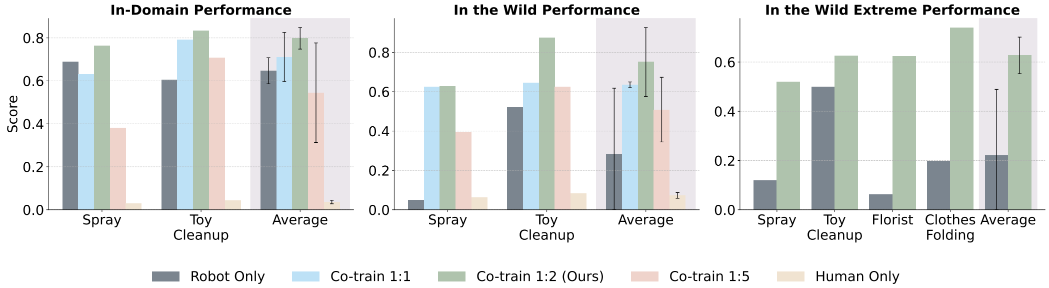


Figure 5: **How does co-training help with scaling up in the wild performance?** We evaluate our policy across three scenarios: (a) In-Domain scenes where robot training data was collected but with novel objects, (b) In-the-Wild scenes present in DexWild but not in robot data, and (c) In-the-Wild Extreme scenes absent from both datasets. Displayed ratio is Robot:Human.

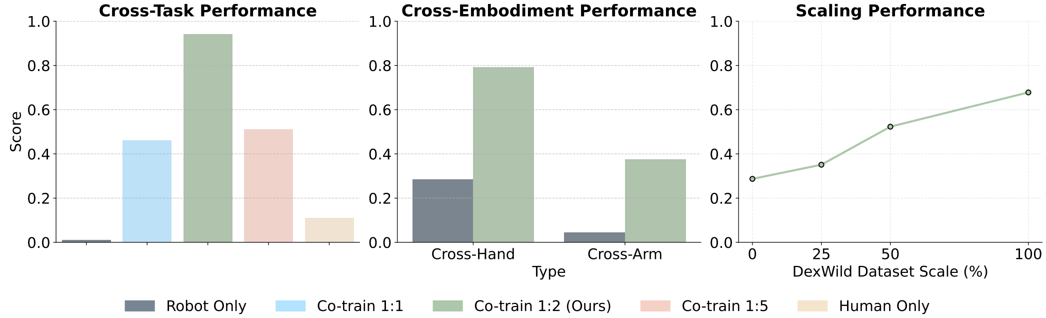


Figure 6: Left: **Cross-Task Performance** – Evaluating DexWild on the pour task using robot data exclusively from the spray task. Middle: **Cross-Embodiment Performance** – Testing DexWild policy on the Original LEAP hand and a Franka robot arm. Right: **Scaling Performance** – Demonstrating improved DexWild performance as dataset size increases. Displayed ratio is Robot:Human.

environments. Using a 1:2 robot-to-human co-training ratio, our policy achieves a **94% success rate**, far exceeding policies trained with only robot (0%) or only human data (11%).

DexWild enables transfer across robot embodiments. Since DexWild data is not tied to any specific embodiment, it naturally supports cross-platform transfer. This prolongs the value of our data, as collecting platform-specific data for every new robot is resource-intensive and impractical. We test two transfer scenarios in in-the-wild extreme scenes:

- **Cross-arm**: Transferring from an xArm to a Franka Panda arm. We achieve a 37.5% success rate, compared to 4.5% for the robot-only baseline—an **8.3× improvement**.
- **Cross-hand**: Transferring from the LEAP Hand V2 Advanced to the original LEAP Hand. We achieve 65.3% success versus 13.3% for the baseline, showing that DexWild generalizes not only across arms, but across dexterous hands as well.

These results, shown in Figure 6, demonstrate that DexWild is an efficient and general framework for dexterous policy learning on many robots.

4.3 Scalability of DexWild

Policy performance scales with dataset size. To understand how data scale impacts policy performance in the wild, we randomly sample subsets of the full human dataset at varying sizes and evaluate the resulting policies. We fix the size of the robot dataset. As shown in Figure 6, there is a clear positive correlation between dataset size and average task performance—rising from 28.7% at 20% dataset size to 67.8% with the full dataset, marking a 2.36× improvement. Interestingly, the learning curve is nonlinear, with especially steep gains in the 25–50% range, suggesting a critical threshold where the policy begins to reliably learn generalizable behaviors.

Importantly, performance continues to improve all the way to 100% data usage, indicating that the system has not yet plateaued. This suggests that even more capable policies could be learned with continued data collection.

DexWild-System enables fast and scalable data collection. Given the observed benefits of scaling, we evaluate the data collection efficiency of DexWild-System via a comparative user study measuring demonstrations per hour. As shown in Figure 9, DexWild-System achieves an average collection rate of **201 demos/hour** across five representative tasks—nearly matching the rate of demonstrations collected using bare hands and **4.6× faster** than a traditional robot teleoperation system based on Gello [41, 56], which achieves just 43 demos/hour.

We identify three key limitations of Gello-based collection that our system overcomes:

1. **Lack of haptic feedback:** Operators cannot feel objects, making fine manipulation difficult for certain tasks.
2. **Scene reset:** Resetting the environment is cumbersome and often requires a second operator or pauses in data collection.
3. **Hardware setup overhead:** Robots are heavy and require time-consuming setup at each new location, whereas DexWild-System is portable and can be set up in minutes.

5 Conclusion and Limitations

We introduce DexWild, a scalable framework for learning dexterous manipulation policies that generalize to new tasks, environments, and robot embodiments. We present DexWild-System, a portable, human-centric data collection device that accelerates dataset creation (4.6× faster than conventional robot teleoperation). We propose a cotraining method that leverages large-scale human demonstrations with minimal robot data to achieve robust generalization—reaching a 68.5% success rate in completely unseen environments, nearly four times higher than methods using robot data alone. DexWild 's embodiment-agnostic design further enables strong cross-embodiment and cross-task transfer, reducing the need for robot-specific data.

Despite these strengths, several limitations motivate future research. Our approach still depends on a small amount of teleoperated robot data to bridge the gap between human and robot actions. Future work could explore improved retargeting or online adaptation to remove this need. Additionally, since human demonstrations rarely include errors, trained policies can struggle to recover from failures. Adding recovery examples or adaptive strategies could improve real-world robustness. Finally, our method uses only visual and kinematic data, limiting performance in contact-rich tasks. Incorporating tactile or haptic sensing could improve handling of delicate interactions.

In summary, DexWild is a step toward scalable, generalizable manipulation policies. Our results highlight the potential of large-scale human interaction data to enable dexterous, versatile robots in diverse real-world settings.

References

[1] Shikhar Bahl, Russell Mendonca, Lili Chen, Unnat Jain, and Deepak Pathak. Affordances from human videos as a versatile representation for robotics. 2023.

[2] Prithviraj Banerjee, Sindi Shkodrani, Pierre Moulon, Shreyas Hampali, Shangchen Han, Fan Zhang, Linguang Zhang, Jade Fountain, Edward Miller, Selen Basol, et al. Hot3d: Hand and object tracking in 3d from egocentric multi-view videos. *arXiv preprint arXiv:2411.19167*, 2024.

[3] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[4] Yu-Wei Chao, Wei Yang, Yu Xiang, Pavlo Molchanov, Ankur Handa, Jonathan Tremblay, Yashraj S Narang, Karl Van Wyk, Umar Iqbal, Stan Birchfield, et al. Dexycb: A benchmark for capturing hand grasping of objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9044–9053, 2021.

[5] Xuxin Cheng, Jialong Li, Shiqi Yang, Ge Yang, and Xiaolong Wang. Open-television: Teleoperation with immersive active visual feedback. *arXiv preprint arXiv:2407.01512*, 2024.

[6] Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. In *Proceedings of Robotics: Science and Systems (RSS)*, 2023.

[7] Cheng Chi, Zhenjia Xu, Chuer Pan, Eric Cousineau, Benjamin Burchfiel, Siyuan Feng, Russ Tedrake, and Shuran Song. Universal manipulation interface: In-the-wild robot teaching without in-the-wild robots. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2024.

[8] Open X-Embodiment Collaboration, Abby O'Neill, Abdul Rehman, Abhinav Gupta, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, Ajinkya Jain, Albert Tung, Alex Bewley, Alex Herzog, Alex Irpan, Alexander Khazatsky, Anant Rai, Anchit Gupta, Andrew Wang, Andrey Kolobov, Anikait Singh, Animesh Garg, Aniruddha Kembhavi, Annie Xie, Anthony Brohan, Antonin Raffin, Archit Sharma, Arefeh Yavary, Arhan Jain, Ashwin Balakrishna, Ayzaan Wahid, Ben Burgess-Limerick, Beomjoon Kim, Bernhard Schölkopf, Blake Wulfe, Brian Ichter, Cewu Lu, Charles Xu, Charlotte Le, Chelsea Finn, Chen Wang, Chenfeng Xu, Cheng Chi, Chenguang Huang, Christine Chan, Christopher Agia, Chuer Pan, Chuyuan Fu, Coline Devin, Danfei Xu, Daniel Morton, Danny Driess, Daphne Chen, Deepak Pathak, Dhruv Shah, Dieter Büchler, Dinesh Jayaraman, Dmitry Kalashnikov, Dorsa Sadigh, Edward Johns, Ethan Foster, Fangchen Liu, Federico Ceola, Fei Xia, Feiyu Zhao, Felipe Vieira Frujeri, Freek Stulp, Gaoyue Zhou, Gaurav S. Sukhatme, Gautam Salhotra, Ge Yan, Gilbert Feng, Giulio Schiavi, Glen Berseth, Gregory Kahn, Guangwen Yang, Guanzhi Wang, Hao Su, Hao-Shu Fang, Haochen Shi, Henghui Bao, Heni Ben Amor, Henrik I Christensen, Hiroki Furuta, Homanga Bharadhwaj, Homer Walke, Hongjie Fang, Huy Ha, Igor Mordatch, Ilija Radosavovic, Isabel Leal, Jacky Liang, Jad Abou-Chakra, Jaehyung Kim, Jaimyn Drake, Jan Peters, Jan Schneider, Jasmine Hsu, Jay Vakil, Jeannette Bohg, Jeffrey Bingham, Jeffrey Wu, Jensen Gao, Jiaheng Hu, Jiajun Wu, Jialin Wu, Jiankai Sun, Jianlan Luo, Jiayuan Gu, Jie Tan, Jihoon Oh, Jimmy Wu, Jingpei Lu, Jingyun Yang, Jitendra Malik, João Silvério, Joey Hejna, Jonathan Booher, Jonathan Tompson, Jonathan Yang, Jordi Salvador, Joseph J. Lim, Junhyek Han, Kaiyuan Wang, Kanishka Rao, Karl Pertsch, Karol Hausman, Keegan Go, Keerthana Gopalakrishnan, Ken Goldberg, Kendra Byrne, Kenneth Oslund, Kento Kawaharazuka, Kevin Black, Kevin Lin, Kevin Zhang, Kiana Ehsani, Kiran Lekkala, Kirsty Ellis, Krishan Rana, Krishnan Srinivasan, Kuan Fang, Kunal Pratap Singh, Kuo-Hao Zeng, Kyle Hatch, Kyle Hsu, Laurent Itti, Lawrence Yunliang Chen, Lerrel Pinto, Li Fei-Fei, Liam Tan, Linxi "Jim" Fan, Lionel Ott, Lisa Lee, Luca Weihs, Magnum Chen,

Marion Lepert, Marius Memmel, Masayoshi Tomizuka, Masha Itkina, Mateo Guaman Castro, Max Spero, Maximilian Du, Michael Ahn, Michael C. Yip, Mingtong Zhang, Mingyu Ding, Minho Heo, Mohan Kumar Srirama, Mohit Sharma, Moo Jin Kim, Naoaki Kanazawa, Nicklas Hansen, Nicolas Heess, Nikhil J Joshi, Niko Suenderhauf, Ning Liu, Norman Di Palo, Nur Muhammad Mahi Shafiullah, Oier Mees, Oliver Kroemer, Osbert Bastani, Pannag R Sanketi, Patrick "Tree" Miller, Patrick Yin, Paul Wohlhart, Peng Xu, Peter David Fagan, Peter Mitrano, Pierre Sermanet, Pieter Abbeel, Priya Sundaresan, Qiuyu Chen, Quan Vuong, Rafael Rafailov, Ran Tian, Ria Doshi, Roberto Mart'in-Mart'in, Rohan Baijal, Rosario Scalise, Rose Hendrix, Roy Lin, Runjia Qian, Ruohan Zhang, Russell Mendonca, Rutav Shah, Ryan Hoque, Ryan Julian, Samuel Bustamante, Sean Kirmani, Sergey Levine, Shan Lin, Sherry Moore, Shikhar Bahl, Shivin Dass, Shubham Sonawani, Shubham Tulsiani, Shuran Song, Sichun Xu, Siddhant Haldar, Siddharth Karamcheti, Simeon Adebola, Simon Guist, Soroush Nasiriany, Stefan Schaal, Stefan Welker, Stephen Tian, Subramanian Ramamoorthy, Sudeep Dasari, Suneel Belkhale, Sungjae Park, Suraj Nair, Suvir Mirchandani, Takayuki Osa, Tanmay Gupta, Tatsuya Harada, Tatsuya Matsushima, Ted Xiao, Thomas Kollar, Tianhe Yu, Tianli Ding, Todor Davchev, Tony Z. Zhao, Travis Armstrong, Trevor Darrell, Trinity Chung, Vidhi Jain, Vikash Kumar, Vincent Vanhoucke, Wei Zhan, Wenxuan Zhou, Wolfram Burgard, Xi Chen, Xiangyu Chen, Xiaolong Wang, Xinghao Zhu, Xinyang Geng, Xiyuan Liu, Xu Liangwei, Xuanlin Li, Yansong Pang, Yao Lu, Yecheng Jason Ma, Yejin Kim, Yevgen Chebotar, Yifan Zhou, Yifeng Zhu, Yilin Wu, Ying Xu, Yixuan Wang, Yonatan Bisk, Yongqiang Dou, Yoonyoung Cho, Youngwoon Lee, Yuchen Cui, Yue Cao, Yueh-Hua Wu, Yujin Tang, Yuke Zhu, Yunchu Zhang, Yunfan Jiang, Yunshuang Li, Yunzhu Li, Yusuke Iwasawa, Yutaka Matsuo, Zehan Ma, Zhuo Xu, Zichen Jeff Cui, Zichen Zhang, Zipeng Fu, and Zipeng Lin. Open X-Embodiment: Robotic learning datasets and RT-X models, 2023.

[9] Juan Antonio Corrales, Francisco A Candelas, and Fernando Torres. Hybrid tracking of human operators using imu/uwb data fusion by a kalman filter. In *Proceedings of the 3rd ACM/IEEE international conference on Human robot interaction*, pages 193–200, 2008.

[10] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. Epic-kitchens: A large-scale dataset for recognizing, anticipating, and retrieving hand-object interactions. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 802–819, 2018.

[11] Sudeep Dasari, Mohan Kumar Srirama, Unnat Jain, and Abhinav Gupta. An unbiased look at datasets for visuo-motor pre-training. In *Conference on Robot Learning*. PMLR, 2023.

[12] Haritheja Etukuru, Norihito Naka, Zijin Hu, Seungjae Lee, Julian Mehu, Aaron Edsinger, Chris Paxton, Soumith Chintala, Lerrel Pinto, and Nur Muhammad Mahi Shafiullah. Robot utility models: General policies for zero-shot deployment in new environments, 2024.

[13] Zicong Fan, Omid Taheri, Dimitrios Tzionas, Muhammed Kocabas, Manuel Kaufmann, Michael J Black, and Otmar Hilliges. Arctic: A dataset for dexterous bimanual hand-object manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12943–12954, 2023.

[14] Authors from UC San Diego and MIT. Open-television: An open-source immersive teleoperation system with stereo visual feedback. *The Robot Report*, 2024.

[15] Kristen Grauman, Michael Ryoo, Aljoša Smolić, Minh Vo, and 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 (CVPR)*, pages 11743–11753, 2022.

[16] Huy Ha, Yihuai Gao, Zipeng Fu, Jie Tan, and Shuran Song. UMI on legs: Making manipulation policies mobile with manipulation-centric whole-body controllers. In *Proceedings of the 2024 Conference on Robot Learning*, 2024.

[17] Ankur Handa, Karl Van Wyk, Wei Yang, Jacky Liang, Yu-Wei Chao, Qian Wan, Stan Birchfield, Nathan Ratliff, and Dieter Fox. Dexpilot: Vision-based teleoperation of dexterous robotic hand-arm system. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9164–9170. IEEE, 2020.

[18] Yafei Hu, Quanting Xie, Vidhi Jain, Jonathan Francis, Jay Patrikar, Nikhil Keetha, Seungchan Kim, Yaqi Xie, Tianyi Zhang, Zhibo Zhao, et al. Toward general-purpose robots via foundation models: A survey and meta-analysis. *arXiv preprint arXiv:2312.08782*, 2023.

[19] Aadhithya Iyer, Zhuoran Peng, Yinlong Dai, Irmak Guzey, Siddhant Haldar, Soumith Chintala, and Lerrel Pinto. OPEN TEACH: A versatile teleoperation system for robotic manipulation. *arXiv preprint arXiv:2403.07870*, 2024.

[20] Simar Kareer, Dhruv Patel, Ryan Punamiya, Pranay Mathur, Shuo Cheng, Chen Wang, Judy Hoffman, and Danfei Xu. Egomimic: Scaling imitation learning via egocentric video, 2024.

[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. *arXiv preprint arXiv:2403.12945*, 2024.

[22] 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. *arXiv preprint arXiv:2406.09246*, 2024.

[23] Fanqi Lin, Yingdong Hu, Pingyue Sheng, Chuan Wen, Jiacheng You, and Yang Gao. Data scaling laws in imitation learning for robotic manipulation. In *Conference on Robot Learning (CoRL)*, 2024.

[24] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *NeurIPS*, 2023.

[25] Ajay Mandlekar, Danfei Xu, Josiah Wong, Soroush Nasiriany, Chen Wang, Rohun Kulkarni, Li Fei-Fei, Silvio Savarese, Yuke Zhu, and Roberto Martín-Martín. What matters in learning from offline human demonstrations for robot manipulation. *arXiv preprint arXiv:2108.03298*, 2021.

[26] Mayank Mittal, Calvin Yu, Qinxi Yu, Jingzhou Liu, Nikita Rudin, David Hoeller, Jia Lin Yuan, Ritvik Singh, Yunrong Guo, Hammad Mazhar, Ajay Mandlekar, Buck Babich, Gavriel State, Marco Hutter, and Animesh Garg. Orbit: A unified simulation framework for interactive robot learning environments. *IEEE Robotics and Automation Letters*, 8(6):3740–3747, June 2023. ISSN 2377-3774. doi:10.1109/lra.2023.3270034. URL `http://dx.doi.org/10.1109/LRA.2023.3270034`.

[27] Suraj Nair, Aravind Rajeswaran, Vikash Kumar, and Chelsea Finn. R3M: A universal visual representation for robot manipulation. *arXiv preprint arXiv:2203.12601*, 2022.

[28] Georgios Pavlakos, Dandan Shan, Ilija Radosavovic, Angjoo Kanazawa, David Fouhey, and Jitendra Malik. Hamer: Hand mesh recovery for the egoexo4d hand pose challenge.

[29] Georgios Pavlakos, Dandan Shan, Ilija Radosavovic, Angjoo Kanazawa, David Fouhey, and Jitendra Malik. Reconstructing hands in 3d with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[30] Alexandra Pfister, Alexandre M West, Shaw Bronner, and Jack Adam Noah. Comparative abilities of microsoft kinect and vicon 3d motion capture for gait analysis. *Journal of medical engineering & technology*, 38(5):274–280, 2014.

[31] Dean A Pomerleau. Alvinn: An autonomous land vehicle in a neural network. *Advances in neural information processing systems*, 1, 1988.

[32] Ri-Zhao Qiu, Shiqi Yang, Xuxin Cheng, Chaitanya Chawla, Jialong Li, Tairan He, Ge Yan, David J. Yoon, Ryan Hoque, Lars Paulsen, Ge Yang, Jian Zhang, Sha Yi, Guanya Shi, and Xiaolong Wang. Humanoid policy ˜ human policy. *arXiv preprint arXiv:2503.13441*, 2025.

[33] Ilija Radosavovic, Tete Xiao, Stephen James, Pieter Abbeel, Jitendra Malik, and Trevor Darrell. Real-world robot learning with masked visual pre-training. *CoRL*, 2022.

[34] Nathan D. Ratliff, Jan Issac, Daniel Kappler, Stan Birchfield, and Dieter Fox. Riemannian motion policies, 2018. URL `https://arxiv.org/abs/1801.02854`.

[35] Yu Rong, Takaaki Shiratori, and Hanbyul Joo. Frankmocap: A monocular 3d whole-body pose estimation system via regression and integration. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1749–1759, 2021.

[36] Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 627–635. JMLR Workshop and Conference Proceedings, 2011.

[37] Stefan Schaal. Is imitation learning the route to humanoid robots? *Trends in cognitive sciences*, 3(6):233–242, 1999.

[38] Kenneth Shaw, Ananye Agarwal, and Deepak Pathak. Leap hand: Low-cost, efficient, and anthropomorphic hand for robot learning. *Robotics: Science and Systems (RSS)*, 2023.

[39] Kenneth Shaw, Shikhar Bahl, and Deepak Pathak. Videodex: Learning dexterity from internet videos. In Karen Liu, Dana Kulic, and Jeff Ichnowski, editors, *Proceedings of The 6th Conference on Robot Learning*, volume 205 of *Proceedings of Machine Learning Research*, pages 654–665. PMLR, 14–18 Dec 2023.

[40] Kenneth Shaw, Shikhar Bahl, and Deepak Pathak. Videodex: Learning dexterity from internet videos. In *Conference on Robot Learning*, pages 654–665. PMLR, 2023.

[41] Kenneth Shaw, Yulong Li, Jiahui Yang, Mohan Kumar Srirama, Ray Liu, Haoyu Xiong, Russell Mendonca, and Deepak Pathak. Bimanual dexterity for complex tasks. In *8th Annual Conference on Robot Learning*, 2024.

[42] Himanshu Gaurav Singh, Antonio Loquercio, Carmelo Sferrazza, Jane Wu, Haozhi Qi, Pieter Abbeel, and Jitendra Malik. Hand-object interaction pretraining from videos, 2024. URL `https://arxiv.org/abs/2409.08273`.

[43] Ritvik Singh, Arthur Allshire, Ankur Handa, Nathan Ratliff, and Karl Van Wyk. Dextrah-rgb: Visuomotor policies to grasp anything with dexterous hands. *arXiv preprint arXiv:2412.01791*, 2024.

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

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

[46] Shuran Song, Andy Zeng, Johnny Lee, and Thomas Funkhouser. Grasping in the wild: Learning 6dof closed-loop grasping from low-cost demonstrations. *Robotics and Automation Letters*, 2020.

[47] Mohan Kumar Srirama, Sudeep Dasari, Shikhar Bahl, and Abhinav Gupta. HRP: Human affordances for robotic pre-training. In *Proceedings of Robotics: Science and Systems*, Delft, Netherlands, 2024.

[48] Andreas Steiner, André Susano Pinto, Michael Tschannen, Daniel Keysers, Xiao Wang, Yonatan Bitton, Alexey Gritsenko, Matthias Minderer, Anthony Sherbondy, Shangbang Long, Siyang Qin, Reeve Ingle, Emanuele Bugliarello, Sahar Kazemzadeh, Thomas Mesnard, Ibrahim Alabdulmohsin, Lucas Beyer, and Xiaohua Zhai. PaliGemma 2: A Family of Versatile VLMs for Transfer. *arXiv preprint arXiv:2412.03555*, 2024.

[49] OM Team, D Ghosh, H Walke, K Pertsch, K Black, O Mees, S Dasari, J Hejna, C Xu, J Luo, et al. Octo: An open-source generalist robot policy. *Proceedings of Robotics: Science and Systems, Delft, Netherlands*, 2023.

[50] Yushuang Tian, Xiaoli Meng, Dapeng Tao, Dongquan Liu, and Chen Feng. Upper limb motion tracking with the integration of imu and kinect. *Neurocomputing*, 159:207–218, 2015.

[51] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[52] Valve Corporation. https://store.steampowered.com/steamvr. [Virtual reality platform].

[53] A Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.

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

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

[56] Philipp Wu, Yide Shentu, Zhongke Yi, Xingyu Lin, and Pieter Abbeel. Gello: A general, low-cost, and intuitive teleoperation framework for robot manipulators, 2023.

[57] Seonghyeon Ye, Joel Jang, Byeongguk Jeon, Sejune Joo, Jianwei Yang, Baolin Peng, Ajay Mandlekar, Reuben Tan, Yu-Wei Chao, Bill Yuchen Lin, et al. Latent action pretraining from videos. *arXiv preprint arXiv:2410.11758*, 2024.

[58] Han Zhang, Songbo Hu, Zhecheng Yuan, and Huazhe Xu. Doglove: Dexterous manipulation with a low-cost open-source haptic force feedback glove. *arXiv preprint arXiv:2502.07730*, 2025.

[59] Tony Z. Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. In *Proceedings of Robotics: Science and Systems (RSS)*, 2023.

[60] Christian Zimmermann, Duygu Ceylan, Jimei Yang, Bryan Russell, Max Argus, and Thomas Brox. Freihand: A dataset for markerless capture of hand pose and shape from single rgb images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 813–822, 2019.

6 Appendix

6.1 Related Works

Generalization for Imitation Learning
Learning generalizable policies for robot manipulation has seen rapid progress, driven largely by advances in visual representation learning and imitation learning from large-scale datasets. On the visual side, embodied representation learning has benefited from egocentric datasets such as Ego4D [15] and EPIC-KITCHENS [10], with recent methods [27, 11, 47, 39] leveraging these datasets to train scalable visual encoders. However, these approaches still require substantial downstream robot demonstrations to train control policies.

In parallel, robot-only demonstration datasets have grown significantly in scale and diversity [21, 8, 54], fueling research in behavior cloning and enabling generalist policy architectures [49, 8, 22]. While these policies show impressive performance across many tasks, they often struggle to generalize to unseen object categories, scene layouts, or environmental conditions [25]. This lack of robustness remains a key limitation of current systems.

Data Generation for Robot Manipulation
Overcoming the robot data bottleneck has become a central challenge in robot learning.

One approach leverages internet videos to extract action information. Several works, such as VideoDex [40] and HOP [42], utilize large scale human videos to learn an action prior through retargeting, which they use to bootstrap policy training. Others, such as LAPA [57], use unlabelled videos to generate latent action representations that can be used for downstream tasks. While these video-based schemes enjoy vast visual diversity, they typically fall short at capturing the precise, low-level motor commands needed for real-world manipulation.

Simulation enables rapid generation of action data at scale. However, creating diverse, realistic environments for many tasks and addressing the sim-to-real gap is challenging. Recent successes in transferring manipulation policies from simulation [43] have been confined to tabletop settings and lack the generalization needed for deployment in diverse environments.

Direct teleoperation on physical robots yields the highest fidelity, but scales poorly. Recent works have shown impressive dexterity and efficient learning in fixed scenarios [59, 56, 41, 19], yet collecting enough demonstrations to generalize across diverse scenes quickly becomes prohibitively expensive.

Recently, there has been a growing body of work that utilizes purpose-collected high quality human embodiment data without the tedious teleoperation. We discuss these approaches in the next section.

Human Action Tracking Systems
In order to acquire high-quality data from human motions, accurate hand and wrist tracking is of paramount importance. To bypass the complexities of hand pose estimation, several works equip users with handheld robot grippers [7, 12, 46]. While this approach simplifies retargeting, it constrains users to the specific morphology of the robot gripper, limiting the diversity of captured behavior. Moreover, many of these systems rely on SLAM-based wrist tracking, which can fail in feature-sparse environments or when occlusions occur [7, 23]—such as during drawer opening or tool use.

Other approaches aim to estimate both hand and wrist poses directly from visual input [29, 35, 5, 45, 28, 20, 32]. These methods are easy to deploy and require no instrumentation, but their performance degrades significantly under occlusion—an unavoidable situation in manipulation. Alternative strategies for wrist tracking, such as IMU-based [9, 50] and outside-in optical systems [30], come with their own limitations: IMUs are lightweight and portable but prone to drift, while optical systems are accurate yet require laborious calibration and controlled environments. DexWild leverages calibration-free Aruco tracking—significantly improving reliability and minimizing setup time as it requires a single monocular camera.

While vision-based methods often attempt to track both the wrist and fingers simultaneously, many recent systems decouple the two to improve accuracy. Kinematic exoskeleton gloves can provide high-fidelity joint measurements and even haptic feedback [58], but are bulky and uncomfortable for long-term use. Instead, DexWild, along with prior works [41, 55], adopts a lightweight glove-based solution that uses electromagnetic field (EMF) sensing to estimate fingertip positions. This allows for accurate, real-time hand tracking that is robust to occlusions and readily retargetable to a wide range of robot hands.

6.2 Detailed Task Description and Scoring Criteria:

We evaluate five dexterous manipulation tasks, each designed to assess different capabilities such as functional grasping, long-horizon planning, precision, bimanual coordination, and deformable object manipulation. Each task is scored according to a structured rubric based on discrete completion milestones.

The task scoring criteria are designed to quantify the performance of different robot tasks based on specific completion milestones. Each task has a set of defined actions with corresponding point values. Higher scores are assigned to more complex or functionally successful actions, while partial completions and failed attempts receive lower scores. This structured scoring system allows for consistent evaluation and comparison of task performance.

Spray Bottle
This task evaluates functional grasping and affordance understanding. The robot must grasp a spray bottle and orient it to spray over a target cloth.

- 0.00: Nothing
- 0.15: Tries functional grasp but fails
- 0.25: Grasp bottle
- 0.75: Grasp bottle, orient over cloth
- 0.75: Grasp bottle, use functional grasp
- 1.00: Grasp bottle, use functional grasp, orient over cloth

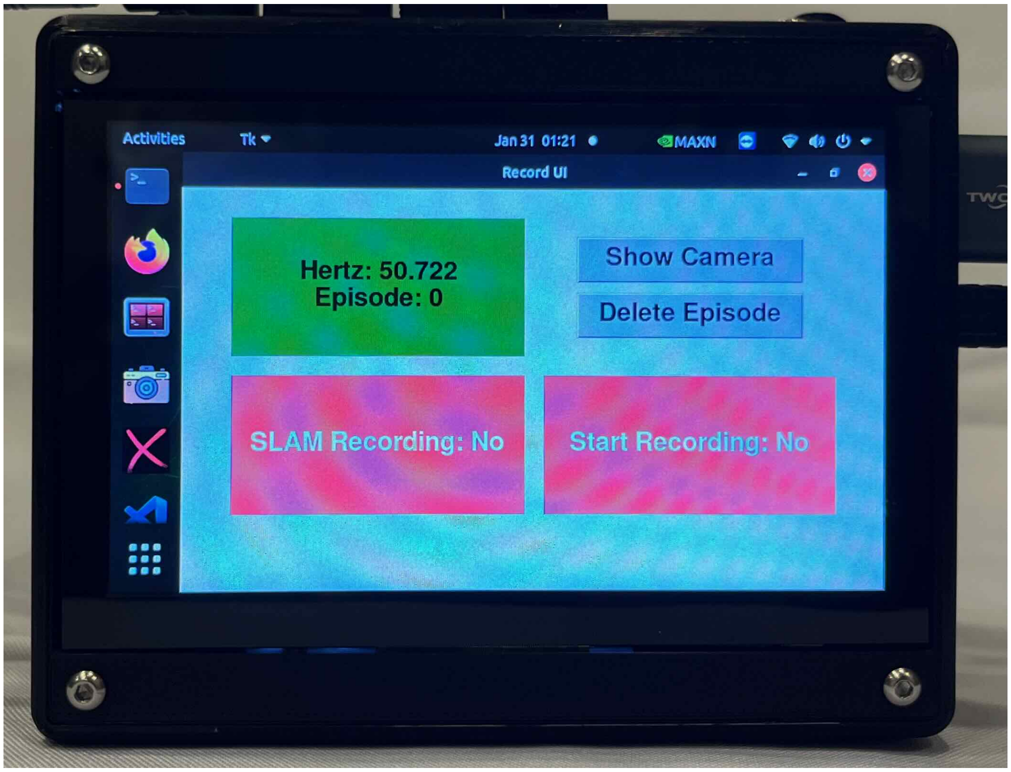
Toy Cleanup
This task tests long-horizon planning and generalization. The robot must collect scattered toys and deposit them in a designated bin.

- 0.00: Nothing
- 0.25: Tries for grasp but fails
- 0.50: Grasp object
- 1.00: Grasp object, drop into bin

Pouring
This task assesses precise motion control and transfer learning from the spray bottle task. The robot must pour liquid from a bottle into a container.

- 0.00: Nothing
- 0.15: Tries functional grasp but fails
- 0.25: Grasp bottle
- 0.75: Grasp bottle, pour into container
- 0.75: Grasp bottle, use functional grasp
- 1.00: Grasp bottle, use functional grasp, pour into container

Bimanual Florist
This task evaluates coordinated control of both hands. The robot must pick up a flower, hand it to the other arm, and insert it into a vase.



Hertz: 50.722
Episode: 0

SLAM Recording: No

Start Recording: No

Figure 7: DexWild-System features a simple and easy-to-use interface for deployment by untrained data collectors.

- 0.00: Nothing
- 0.15: Tries grasp but fails
- 0.25: Grasp the bouquet
- 0.75: Grasp the bouquet, handover
- 1.00: Grasp the bouquet, handover, insert into vase

Clothes Folding
This task tests manipulation of deformable objects using both hands. The robot must fold a clothing item placed on a surface.

- 0.00: Nothing
- 0.25: Tries grasp but fails
- 0.50: Grasp with one hand
- 0.75: Grasp with both hands
- 1.00: Grasp and fold

6.3 Dataset Sizes

Human Demonstrations: 3,000 demonstrations from 30 different environments for each of the *Spray Bottle* and *Toy Cleanup* tasks, 621 trajectories from 6 environments for the *Pour* task, 1,545 demonstrations from 15 environments for the *Florist* task, and 1,124 demonstrations from 12 environments for the *Clothes Folding* task.

Robot Demonstrations: The robot dataset D_R includes 1,395 demonstrations: 388 for *Spray Bottle*, 370 for *Toy Cleanup*, 111 for *Pour*, 236 for *Florist*, and 290 for *Clothes Folding* tasks.

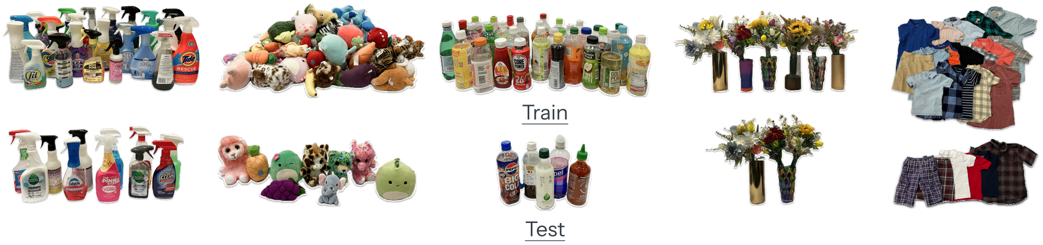


Figure 8: We collect data using a diverse set of objects across categories. *Spray Bottle Task* – 25 Train, 11 Test; *Toy Cleanup Task* – 64 Train, 9 Test; *Pour Task* – 35 Train, 5 Test; *Florist Task* - 6 Train, 2 Test; *Clothes Folding Task* - 17 Train, 6 Test.

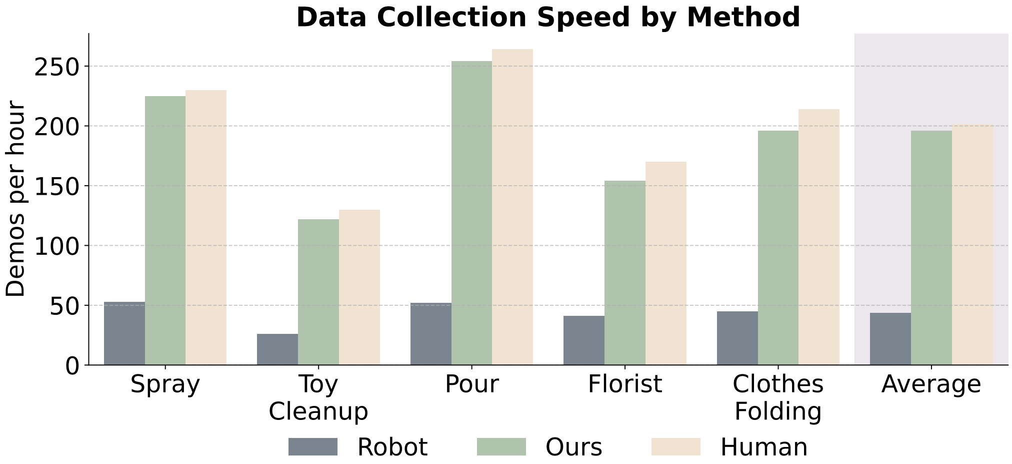


Figure 9: DexWild-System offers **4.6**× improvement over robot data collection speed and nearly matches the human bare hands data collection speed.

6.4 Training and Test Objects

Please see Figure 8 for breakdown of train and test objects

6.5 Data Collection Procedure

To deploy DexWild-System with untrained data collectors, we provide a one-page instruction sheet outlining the task, object setup, and system startup/shutdown. DexWild-System includes three core components: a wrist-tracking camera, a battery-powered mini-PC for onboard data capture, and a custom sensor pod with a motion-capture glove and palm-mounted cameras. At a new site, users simply wear the mocap glove and power on the mini-PC with a provided power bank. For egocentric tracking, a headstrap holds the tracking camera; for exocentric tracking, we provide a collapsible tripod. Once booted, users launch our custom desktop app and control recording via a Bluetooth clicker or foot pedal. The UI (Fig. 7) shows sensor status, SLAM recording, and data capture indicators, along with buttons to view the tracking camera feed and delete the last episode. Collectors gather 100 episodes per location. After the day is finished, we upload the data to our remote machine for processing.

6.6 Data Collection Speed

Please see Figure 9 for comparison of data collection speed of different methods

6.7 Downstream Data Processing

Each episode is stored in its own folder, with subfolders organizing individual actions and observations. SVO recordings from the Zed Mini camera—used for SLAM and wrist pose tracking—are saved separately, with each file covering five episodes. To begin data processing, we use the Zed SDK to decode these SVO files, reconstruct the camera's motion, and perform ArUco cube tracking and wrist pose estimation using both the left image and stereo depth data. We then apply a filtering pipeline to assess tracking quality; episodes are discarded if the wrist pose cannot be reliably tracked for more than 75% of the duration. Next, we compute the action distribution and clip outliers outside the 2nd and 97th percentiles. We smooth the trajectories using interpolation and Gaussian filtering to ensure fluid motion. Hand motions are then retargeted using inverse kinematics in PyBullet, following the method in [41]. The entire pipeline is parallelized using Ray for efficiency.

6.8 Behavior Cloning Policy Architecture and Training Hyper-Parameters

Our behavior cloning policy takes as input RGB images and relative state history. We obtain tokens for the image observation via a ViT and tokens for relative states via linear layers. The weights of ViT is initialized from the Soup 1M model from [11]. We decide to include relative states as we found it greatly increases the robustness of the policy, and enables smoother motions. In particular, for bimanual tasks, we find that including the interhand pose (pose of left hand relative to right hand) greatly increases success rate in tasks like Florist We implement both Action Chunking Transformer [59] and Diffusion U-Net [6] as policy classes, which output a sequence of actions. The network outputs actions which consists of relative end effector actions and absolute hand joint angles.

We list the hyper-paramaters that we used for policy training using behavior-cloning in this Table 5

6.9 Low Level Motion Control

For optimal smoothness of our policies and safety, we employ a Riemannian Motion Policy (RMP) [34] implemented in Isaac Lab [26], where the RMP dynamically generates joint-space targets given end effector targets. RMP also has the added benefit of incorporating real-time collision avoidance, preventing self-collision between the arms and a set table height. Although our policies does not rely on RMP to prevent collisions, the peace of mind is appreciated.

6.10 Comparing Policy Classes

Does DexWild work with different behavior cloning policy classes? Table 1 compares the performance of ACT and Diffusion—across both the In-the-Wild and In-the-Wild Extreme settings. Each policy is evaluated in a robot-only setting and a co-trained (1:2) setting using the DexWild dataset. Notably, Diffusion policies benefit more from DexWild co-training, achieving the highest scores in all tasks, including substantial improvements on the Pour task where the policy must generalize across tasks. These results suggest that DexWild co-training enables stronger generalization, especially when paired with expressive policy architectures like Diffusion.

6.11 Cross Hand Extended Results

Does DexWild generalize across different robot hands? Table 2 reports LEAP Hand performance under both *In the Wild* and *In the Wild Extreme* conditions. In every case, DexWild co-training substantially outperforms the robot-only baseline. These results highlight the effectiveness of DexWild in cross embodiment generalization even when using a completely different robot hand.

6.12 Scaling Extended Results

Does DexWild improve as more DexWild data is added? Table 3 shows steady gains as we scale from 0% to 100% of the DexWild dataset. Performance increases steadily with more human demonstrations, with a notable jump between 25% and 50% of the dataset. These results demonstrate that DexWild enables scalable learning, where even comparably smaller data scales yields substantial gains, and additional data continues to enhance generalization

6.13 Cotraining Extended Results

How does DexWild react to different cotraining ratios? Table 4 groups all three raw metrics: (a) In-Domain, (b) In-the-Wild, and (c) In-the-Wild Extreme. All evaluations were run on xArm + LEAP Hand V2 Advanced.

Task	Policy Class	In the Wild		In the Wild Extreme	
		Robot Only	1:2	Robot Only	1:2
Spray	ACT	0.000	**0.680**	0.115	**0.395**
	Diffusion	0.050	**0.628**	0.120	**0.520**
Toy Cleanup	ACT	0.458	**0.583**	0.125	**0.458**
	Diffusion	0.521	**0.875**	0.500	**0.625**
Pour (Cross Task)	ACT	0.025	**0.508**	0.000	**0.350**
	Diffusion	0.000	**0.958**	0.000	**0.917**

Table 1: DexWild Performance on Different Policy Classes

	In the Wild		In the Wild Extreme	
Task	Robot Only	1:2	Robot Only	1:2
Spray	0.305	**0.805**	0.150	**0.600**
Toy Cleanup	0.500	**0.656**	0.250	**0.542**
Pour (Cross Task)	0.050	**0.917**	0.000	**0.817**

Table 2: LEAP Hand Performance on In-the-Wild and In-the-Wild Extreme Tasks. Ratio is Robot:Human

Scale	0%	25%	50%	100%
Spray	0.060	0.260	0.605	0.565
Toy Cleanup	0.514	0.442	0.440	0.792
Average	**0.287**	**0.351**	**0.523**	**0.678**
Std	**0.321**	**0.129**	**0.116**	**0.160**

Table 3: Performance Scaling with DexWild Dataset Size

Task	**Robot**	**1:1**	**1:2**	**1:5**	**Human**
Spray	0.690	0.630	0.763	0.381	0.030
Toy Cleanup	0.604	0.792	0.833	0.708	0.042
Average	**0.647**	**0.711**	**0.798**	**0.545**	**0.036**
Std	**0.061**	**0.114**	**0.050**	**0.232**	**0.008**

(a) In Distribution Task Performance

Task	**Robot**	**1:1**	**1:2**	**1:5**	**Human**
Spray	0.050	0.625	0.628	0.393	0.063
Toy Cleanup	0.521	0.646	0.875	0.625	0.083
Average	**0.285**	**0.635**	**0.751**	**0.509**	**0.073**
Std	**0.333**	**0.015**	**0.175**	**0.164**	**0.015**

(b) In-the-Wild Task Performance

Task	**Robot**	**1:2**
Spray	0.120	0.520
Toy Cleanup	0.500	0.625
Bimanual Florist	0.063	0.623
Bimanual Clothes Folding	0.198	0.740
Average	**0.220**	**0.627**
Std	**0.195**	**0.090**

(c) In-the-Wild Extreme Task Performance

Table 4: Performance Across Cotrain Ratios for Varying Deployment Conditions. Ratio is Robot:Human

Hyperparameter	**Value**
Training Configuration	
Optimizer	AdamW
Base Learning Rate	3e-4
Optimizer Momentum	$\beta_1, \beta_2 = 0.95, 0.999$
Learning Rate Schedule	Cosine (diffusers)
Warmup Steps	2000
Total Steps	70000
Batch Size	256
Environment Frequency	30 Hz
Observation Settings	
Proprioception Horizon	1 (Spray, Toy, Pour) 3 (Florist, Clothes)
Image Horizon	1 (all tasks)
Observation Resolution	224×224
Observation Dim	9 (Spray, Toy, Pour) 27 (Florist, Clothes)
Action Dimension	26 (Spray, Toy, Pour) 52 (Florist, Clothes)
Action Chunk Size	48
Action Chunking Transformer	
# Encoder Layers	4
# Decoder Layers	6
# MHSA Heads	8
Feed-Forward Dim	3200
Hidden Dim (Token Dim)	768
Dropout	0.1
Feature Norm	LayerNorm
Diffusion U-Net Policy	
Train Diffusion Steps	100
Eval Diffusion Steps	16
Down Channels	[256, 512, 1024]
Kernel Size	3
Groups (GN)	8
Dropout	0.1
Feature Norm	None

Table 5: Full training and architecture settings used across our experiments.