I Know Kung Fu: Synthetic Dexterous Hand Demonstration Collection via VR Teleoperation

Anonymous Author(s)
Affiliation
Address
email

Abstract

In robotic dexterous manipulation learning, limitations in the acquisition of large and scalable amounts of high-quality demonstration data have been a critical bottleneck. Although recent works show significant advances in increasing efficiency through the use of VR technology, human video, or synthetic demonstration generation techniques, these methods have limitations in accuracy and effectiveness. In this project, we propose a data collection pipeline that uses a VR-teleoperation system to track human hand motions in order to collect dexterous hand demonstrations in simulation. Our method utilizes a wrist motion tracker for hand orientation tracking, followed by a dexterous retargeting module to sync human movement with robot movement in real time. We will also implement demonstration augmentation, ultimately yielding multiple distinct, successful trajectories across different scenarios from a single human demonstration.

1 Introduction

Robotic dexterous manipulation refers to the ability of multi-fingered robotic hands to perform object-centered manipulation tasks. Unlike traditional robot manipulation with claw-like or two-fingered grippers, dexterous manipulation requires precise control of forces and motion similar to human movement.[1].

One fundamental problem in robotics is the sim-to-real gap — the misalignment between simulated training and real-world environments due to differences in physics modeling, collision detection, visual rendering, and more. While this gap can be mitigated by enlarging training datasets, dexterous manipulation data collection is difficult and expensive. Due to the complexity of multi-jointed dexterous hands that would require extensive training and meticulous reward configuration in RL methods, imitation learning from human demonstrations offers a promising alternative, allowing robots to directly mimic human movement [2, 3]. Previous demonstration collection methods like video extraction prove effective [4, 5, 6] but suffer from poor scalability and significant information loss during 2D to 3D transformation. Furthermore, many existing data collection pipelines lack cross-embodiment support and only target specific embodiments, which could drastically increase the costs of producing separate, large-scale datasets for different dexterous hand embodiments.

To address these issues, we propose a data collection and augmentation pipeline that produces physically plausible, scalable, and generalizable cross-embodiment demonstration data for dexterous manipulation imitation learning. Our pipeline is designed to use VR-teleoperation inspired by OpenVR [7] to collect data in simulation, followed by a data augmentation module inspired by DemoGen [8]. This captures comprehensive physical information while generating tens to hundreds of synthetic demonstrations from a single human example. By varying camera perspective, scene appearance, object models, and robot embodiments, we also effectively increase dataset collection

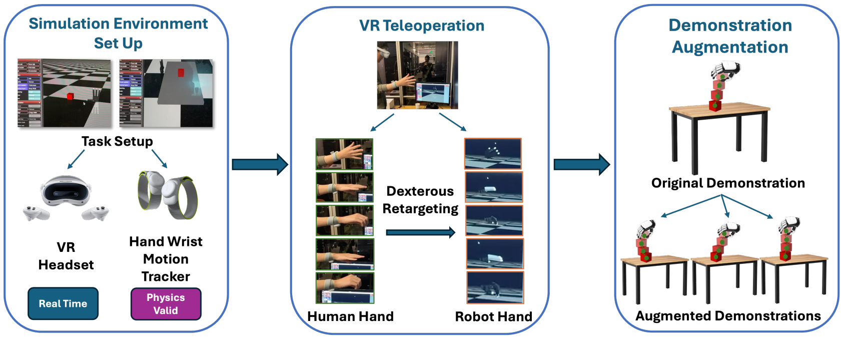

Figure 1: Overview of **I Know Kung Fu** pipeline.

scalability. Built on RoboVerse's [9] unified framework, this system provides cross-simulator and cross-embodiment support with retargeting capabilities. As a next step, we will fully implement DemoGen's demonstration augmentation and PWTF's [10] filtering modules with VLMs to identify successful trajectories.

2 Related Work

2.1 Learning from Human Demonstrations

Imitation learning from human demonstrations effectively trains dexterous hands to complete complex tasks, avoiding complex reward optimization and trial-and-error training in RL techniques. Traditional imitation learning through video-based demonstration extraction offers high scalability but loses crucial spatial and physical information during 2D to 3D transformation, producing physically implausible trajectories. Currently, teleoperation frameworks using gloves, joysticks, or other controllers, can capture precise trajectories. However, limited by specific teleoperation controllers and targeted embodiments, they are usually expensive, time-consuming, and hard to scale up.

VR-based teleoperation systems advance this by placing operators in simulation environments, capturing universal hand motions that are applicable to various embodiments. This eliminates real robot setups while ensuring physical plausibility and improving data collection scalability.

2.2 Data Collection via Teleoperation

Teleoperation enables positional data collection through remote robot control, often using equipment like gloves, keyboards, or gamepads. While such teleoperation systems are promising solutions for bridging sim-to-real gaps and accelerating the training process [2], they become extremely challenging for multi-fingered dexterous hands due to their complexity.

Instead, our work uses a VR teleoperation system inspired by OpenVR [7]. Previous works [11] have demonstrated VR teleoperation as a promising alternative to conducting imitation learning. Specifically, this system creates physically plausible training datasets by recording precise action information and object trajectories. For our pipeline, we leverage dexterous retargeting [12] to collect dexterous manipulation demonstrations by translating human hand motions to robot motion in real time through the VR headset.

2.3 Demonstration Augmentation

Demonstration augmentation reduces human effort in training manipulation and visuomotor policies. One common method of demonstration augmentation includes data generation through the use of LLMs. LLM-based planning methods can determine optimal trajectories but suffer from hallucinations, action inconsistencies, and a lack of low-level control commands like real physics, joint angles, and collision detection.

MimicGen [13] demonstrates another method by augmenting real, human demonstrations through spatial transformations, capturing precise human movements to improve learning speed and success rates. Despite these improvements, MimicGen requires expensive on-robot rollouts for visual observations, limiting scalability. DemoGen [8] advances it by using 3D point clouds and TAMP analysis instead of visual observation. Thus, its pipeline enables high-speed, cost-effective generation while decreasing tedious human labor. In our project, we will leverage DemoGen's advantages as a next step to augment our collected demonstrations and improve scalability.

3 Preliminaries

We target a scalable, physics-grounded collection of dexterous-hand demonstrations in simulation with a VR-teleoperation front–end. The system is orchestrated in **RoboVerse** with plug-in-able physics/rendering back-ends (e.g., IsaacLab/IsaacGym/Genesis/PyBullet/MuJoCo/SAPIEN), and interfaces to a **Pico 4 Ultra** HMD and an external **motion tracker** through the XR-Robotics SDK. The operator experiences an egocentric, head-coupled camera in simulation, while the simulator exposes time-synchronized states for the hand root, finger joints, and task objects.

Frames, units, and notation. We denote by $\mathcal{F}_{\text{pico}}$ the tracking frame (HMD/motion tracker) and by $\mathcal{F}_{\text{sim}}$ the simulator world frame. All linear quantities are in meters, angles in radians, and time in seconds. The hand root (palm/wrist) pose is $T_{\text{hand}}(t) \in SE(3)$, finger joint vector $q_{\text{finger}}(t) \in R^n$, and object pose $T_{\text{obj}}(t) \in SE(3)$ at time t. A fixed rigid transform $R_{\text{pico}\to\text{sim}} \in SO(3)$ maps tracker coordinates into the simulator world.

Head-coupled camera and logging. The VR camera extrinsics follow the operator's head pose to provide first-person depth and occlusion cues. At each step we log

$$\mathcal{L}(t) \;=\; \big\{T_{\text{hand}}(t),\, q_{\text{finger}}(t),\, T_{\text{obj}}(t),\, \Pi,\, \Xi(t),\, \tau(t)\big\},$$

where Π and $\Xi(t)$ are camera intrinsics/extrinsics and $\tau(t)$ is a monotonic timestamp. This schema enables exact replay and downstream training without additional instrumentation.

4 Methods

Our method comprises of three modules as shown in the Fig. 1 (i) hand wrist pose tracking in simulation, (ii) dexterous retargeting via finger motion capture, and (iii) physics-valid demonstration augmentation. Together they provide real-time VR teleoperation that yields contact-aware trajectories and a data engine for producing large, diverse, and physically plausible demonstrations suitable for imitation learning.

4.1 Hand Wrist Pose Tracking in Simulation

The external motion tracker provides a time-stamped 6-DoF signal $(\mathbf{p}_{\text{pico}}(t),\, \mathbf{q}_{\text{pico}}(t))$ in $\mathcal{F}_{\text{pico}}$. A fixed rotation $R_{\text{pico}\to\text{sim}} \in SO(3)$ (estimated once offline during calibration) maps tracker coordinates into the simulator world $\mathcal{F}_{\text{sim}}$. Let $\mathbf{q}_R$ be the unit quaternion associated with $R_{\text{pico}\to\text{sim}}$ and $\otimes$ the quaternion product. The hand-root pose applied in simulation is

$$\mathbf{p}_{\text{sim}}(t) \;=\; R_{\text{pico}\to\text{sim}}\,\mathbf{p}_{\text{pico}}(t), \qquad \mathbf{q}_{\text{sim}}(t) \;=\; \mathbf{q}_R \,\otimes\, \mathbf{q}_{\text{pico}}(t) \,\otimes\, \mathbf{q}_R^{-1}, \tag{1}$$

and we set $T_{\text{hand}}(t) := (\mathbf{p}_{\text{sim}}(t),\, \mathbf{q}_{\text{sim}}(t))$ each simulator step. To stabilize closed-loop control, we constrain the incremental motion by clamping translational and rotational updates to small bounds (e.g., $\|\Delta\mathbf{p}\| \le \epsilon_p$, $\angle(\Delta\mathbf{q}) \le \epsilon_q$). When tracker frames are dropped or stale, the system skips updates for that step rather than integrating erroneous samples.

Because the VR sensor stream and the simulator may run at different rates, we align them using nearest-neighbor or linear interpolation in timestamp space. Denoting the simulator step times by $\{t_k\}$ and the tracker samples by $\{t_i\}$, we compute

$$(\mathbf{p}_{\text{pico}}(t_k),\, \mathbf{q}_{\text{pico}}(t_k)) \;\approx\; \text{Interp}\big(\{(\mathbf{p}_{\text{pico}}(t_i),\, \mathbf{q}_{\text{pico}}(t_i))\}_i,\, t_k\big),$$

followed by (1). A light exponential smoother can be applied on $\mathbf{p}_{\text{sim}}$ and $\mathbf{q}_{\text{sim}}$ to reduce micro-jitter while preserving responsiveness. The head pose from the HMD drives an egocentric, head-coupled camera in the simulator to provide the operator with consistent parallax and occlusion cues. In the final log $\mathcal{L}(t)$, we store the applied $T_{\text{hand}}(t)$ and the original time-synchronized tracker measurements for exact replay.

4.2 Dexterous Retargeting via Finger Motion Capture

Hand landmarks from the XR-Robotics SDK are normalized to a MediaPipe-style skeleton in $\mathcal{F}_{\text{pico}}$ and converted at each time step into dexterous joint targets $q_{\text{finger}}(t) \in R^n$ for the simulated hand. The retargeter is configured per embodiment through a specification that lists joint names, ranges, and per-joint scaling factors. In the vector retargeting mode, we first normalize bone lengths and landmark spreads to a canonical scale, then apply a linear mapping with joint-wise clamps:

$$q_{\text{finger}}(t) \;=\; \text{Clamp}\big(W\,\phi(\text{landmarks}(t)) + b,\; q_{\min},\; q_{\max}\big),$$

where $\phi(\cdot)$ encodes relative landmark directions and apertures, and (W, b) are fitted once from short calibration motions. For digits with coupled kinematics (e.g., distal/proximal phalanges), we enforce simple synergies so that closing motions remain contact-friendly and avoid hyperextension.

At each simulator step we send the pair $\big(T_{\text{hand}}(t),\, q_{\text{finger}}(t)\big)$ as the position command target. Because contact stabilization in dexterous manipulation is sensitive to small joint oscillations, we optionally add a first-order low-pass filter on $q_{\text{finger}}(t)$ and saturate velocity/acceleration changes to remain within the physical capabilities of the simulated hand. The operator receives immediate visual feedback through the head-coupled view, which empirically reduces overshoot near contact initiation. The log $\mathcal{L}(t)$ records $T_{\text{hand}}(t)$, $q_{\text{finger}}(t)$, the object pose $T_{\text{obj}}(t) \in SE(3)$, and camera parameters, all time-stamped for downstream learning and analysis.

5 Discussion and Future Work

Motivation and design rationale. We aim for a *scalable*, *generalizable* pathway to dexterous manipulation data. VR teleoperation captures human hand motion at fine spatiotemporal resolution and maps it, in real time, to simulator-native trajectories $\{T_{\text{hand}}(t),\, q_{\text{finger}}(t)\}$. To make seeds portable across embodiments, we also maintain a canonical human-hand parameterization and learn a retargeter per robot hand. Using **RoboVerse** separates task orchestration from physics/rendering and headset I/O, positioning the pipeline for *cross-VR* (different HMDs/trackers) and *cross-simulator* (MuJoCo/Isaac/SAPIEN) reuse without changing the logging schema. The resulting seeds are contact-rich, physics-valid, and ready for augmentation and policy learning.

Toward scalable augmentation. We will integrate a DemoGen-style inference module to expand seeds without extra teleoperation. A source trajectory is decomposed into *on-object skill* (contact) and*free-space* segments; the former are transformed coherently to preserve contact frames, while the latter are re-planned under new scene configurations so kinematics and collisions remain feasible. Visual observations are synthesized in a point-cloud modality by rearranging subjects via 3D editing, keeping action intent aligned to novel object configurations. We will sample *multiple* variants along object-pose resets, object instances, viewpoints, and small timing/pose disturbances, accepting only those that satisfy the seed's task predicates. The augmented set uses the same units and notation as $\{T_{\text{hand}}(t),\, q_{\text{finger}}(t),\, T_{\text{obj}}(t)\}$.

Experimental plan. (i) *Cross-VR reproducibility:* collect with different headsets/trackers; report fingertip error and DTW of contact sequences, plus user-time → usable-demo throughput. (ii) *Cross-simulator portability:* replay identical seeds across engines; report terminal object error $\|T^{\text{final}}_{\text{obj}} - T^{\text{goal}}_{\text{obj}}\|$, contact-event alignment, and success gaps. (iii) *Augmentation scaling:* with the Demonstration Augmentation module enabled, plot success versus reset coverage and dataset size; ablate pose/instance/viewpoint/disturbance sampling; compare BC/Diffusion policies trained on one seed vs. seeds+augmented.

Limitations and open problems. VR introduces noise, latency, and frame drops that can cause micro-oscillations in $q_{\text{finger}}(t)$; smoothing and rate limits help but may not eliminate artifacts. MANO-to-robot retargeting can distort contact geometry for disparate kinematics; per-embodiment calibration and contact-aware synergies are needed. Cross-simulator replay may shift contact timing; we will quantify and, if necessary, constrain tasks to regimes where engines agree. Augmentation can bias resets toward easy regions if acceptance is too strict; coverage heatmaps and tuned sampling/validation will counteract mode collapse. Despite these caveats, VR-in-sim seeds plus DemoGen-style expansion offer a principled route to scalable, generalizable dexterous datasets.

References

[1] A.M. Okamura, N. Smaby, and M.R. Cutkosky. An overview of dexterous manipulation. In *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*, volume 1, pages 255–262 vol.1, 2000.

[2] Gaofeng Li, Ruize Wang, Peisen Xu, Qi Ye, and Jiming Chen. The developments and challenges towards dexterous and embodied robotic manipulation: A survey, 2025.

[3] Shan An, Ziyu Meng, Chao Tang, Yuning Zhou, Tengyu Liu, Fangqiang Ding, Shufang Zhang, Yao Mu, Ran Song, Wei Zhang, Zeng-Guang Hou, and Hong Zhang. Dexterous manipulation through imitation learning: A survey, 2025.

[4] Georgios Pavlakos, Dandan Shan, Ilija Radosavovic, Angjoo Kanazawa, David Fouhey, and Jitendra Malik. Reconstructing hands in 3d with transformers, 2023.

[5] 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.

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

[7] Abraham George, Alison Bartsch, and Amir Barati Farimani. Openvr: Teleoperation for manipulation, 2023.

[8] Zhengrong Xue, Shuying Deng, Zhenyang Chen, Yixuan Wang, Zhecheng Yuan, and Huazhe Xu. Demogen: Synthetic demonstration generation for data-efficient visuomotor policy learning. *arXiv preprint arXiv:2502.16932*, 2025.

[9] Haoran Geng, Feishi Wang, Songlin Wei, Yuyang Li, Bangjun Wang, Boshi An, Charlie Tianyue Cheng, Haozhe Lou, Peihao Li, Yen-Jen Wang, Yutong Liang, Dylan Goetting, Chaoyi Xu, Haozhe Chen, Yuxi Qian, Yiran Geng, Jiageng Mao, Weikang Wan, Mingtong Zhang, Jiangran Lyu, Siheng Zhao, Jiazhao Zhang, Jialiang Zhang, Chengyang Zhao, Haoran Lu, Yufei Ding, Ran Gong, Yuran Wang, Yuxuan Kuang, Ruihai Wu, Baoxiong Jia, Carlo Sferrazza, Hao Dong, Siyuan Huang, Yue Wang, Jitendra Malik, and Pieter Abbeel. Roboverse: Towards a unified platform, dataset and benchmark for scalable and generalizable robot learning, 2025.

[10] Chuanruo Ning, Kuan Fang, and Wei-Chiu Ma. Prompting with the future: Open-world model predictive control with interactive digital twins, 2025.

[11] Tianhao Zhang, Zoe McCarthy, Owen Jow, Dennis Lee, Xi Chen, Ken Goldberg, and Pieter Abbeel. Deep imitation learning for complex manipulation tasks from virtual reality teleoperation, 2018.

[12] Yuzhe Qin, Wei Yang, Binghao Huang, Karl Van Wyk, Hao Su, Xiaolong Wang, Yu-Wei Chao, and Dieter Fox. Anyteleop: A general vision-based dexterous robot arm-hand teleoperation system. In *Robotics: Science and Systems*, 2023.

[13] Ajay Mandlekar, Soroush Nasiriany, Bowen Wen, Iretiayo Akinola, Yashraj Narang, Linxi Fan, Yuke Zhu, and Dieter Fox. Mimicgen: A data generation system for scalable robot learning using human demonstrations. In *7th Annual Conference on Robot Learning*, 2023.