Reproducing ViViDex: Learning Vision-based Dexterous Manipulation from Human Videos

Anderson Compalas

1 Introduction

Dexterous manipulation, the ability to grasp and reposition objects using multi-fingered hands, remains one of the most challenging problems in robotics. While humans accomplish such tasks effortlessly through visual perception, replicating this ability on robotic hands requires translating high-dimensional visual signals into precise joint-level control. The complexity of contact-rich manipulation and the difficulty of obtaining large-scale training data have historically limited progress in this area.

Imitation learning offers a promising path by transferring human demonstrations directly to robots. However, the standard approach of teleoperation requires substantial human effort and specialized hardware, making it difficult to scale. Prior work has explored extracting demonstrations from human videos to reduce this burden. DexMV [6], developed at UCSD, demonstrated that hand and object poses could be extracted from human videos and used to train dexterous manipulation policies, but required hundreds of demonstrations per task due to noisy single-camera pose estimation. Other methods, such as DexRepNet [5], rely on privileged sensor information not available from vision alone.

ViViDex [2] (ICRA 2025) addresses these limitations by introducing a framework that learns vision-based dexterous manipulation from a *single* human demonstration. By leveraging DexYCB's [1] high-quality multi-view pose estimation, ViViDex obtains clean reference trajectories and trains state-based policies via trajectory-guided PPO, subsequently distilling them into vision-based policies via Behaviour Cloning on rendered rollouts. The full framework is evaluated on the MuJoCo Adroit hand and the Allegro hand in SAPIEN with sim-to-real transfer across 3 different tasks; results show comparable or superior performance to DexMV using only one demonstration versus 20–100.

Project scope. We reproduce the first two stages of the ViViDex pipeline on only a single task: reference trajectory extraction and trajectory-guided state-based policy learning on the MuJoCo Adroit hand on the "relocate" task. We do not reproduce the Allegro hand, SAPIEN sim-to-real experiments, or the vision-based Behaviour Cloning stage. Our primary contribution is a from-scratch implementation of the retargeting pipeline, which ViViDex does not release, and an empirical investigation of how retargeting quality affects downstream PPO training.

Contributions. We make the following contributions:

- A complete re-implementation of the DexYCB-to-Adroit retargeting pipeline, including derivation of the camera-to-world coordinate transform from DexYCB calibration data.
- Two retargeting variants, naive (position-only NLopt) and chain (MANO-initialized), with qualitative comparison against ViViDex's reference trajectory.
- PPO training runs for all three trajectory variants, revealing that retargeting quality is the primary bottleneck for task success, and that our chain retargeting is the more promising direction for future work.
- Identification of a practical compute bottleneck: MuJoCo simulation is CPU-bound, making ViViDex's reported 2-hour A100 training time inapplicable to standard cloud GPU instances.

Code is available at:

github.com/acompalas/Reproducing-ViViDex-Learning-Vision-based-Dexterous-Manipulation-from-Human-Videos

2 Related Work

Dexterous manipulation from demonstrations. Learning dexterous manipulation from human demonstrations has been studied through teleoperation, motion capture, and video-based approaches. DexMV [6] introduced the use of human video for dexterous manipulation imitation, extracting poses via automated estimation and training policies with imitation learning algorithms DAPG, SOIL, and GAIL. The reliance on noisy single-camera estimation required hundreds of demonstrations. ViViDex [2] improves on this by using DexYCB's multi-view ground-truth annotations and a trajectory-guided PPO formulation to extract physically valid policies from a single video.

Hand pose estimation and datasets. The MANO model [7] provides a differentiable parametric hand representation that has become standard for 3D hand pose estimation. DexYCB [1] builds on MANO to provide a large-scale benchmark of human hand-object interactions with accurate multi-view annotations, making it well-suited as a demonstration source.

Trajectory-guided reinforcement learning. TCDM [3] introduced the use of object trajectory tracking as a reward for dexterous manipulation, decomposing the task into pregrasp and manipulation phases. ViViDex extends this decomposition to include hand pose guidance in the pregrasp phase, enabling the policy to learn from a single reference trajectory rather than requiring exemplar objects.

3 Method

ViViDex [2] introduces a three-stage framework: (1) reference trajectory extraction from multi-view RGB-D data, (2) trajectory-guided state-based policy learning via PPO, and (3) vision-based policy learning via Behaviour Cloning. We describe each stage, with particular emphasis on Stage 1, which constitutes our primary implementation contribution.

3.1 Reference Trajectory Extraction

3.1.1 Multi-View Pose Estimation via DexYCB

The demonstration data is sourced from the DexYCB dataset [1], which captures human hand-object interactions using 8 synchronized RGB-D cameras. Human annotators label 2D joint keypoints for 21 hand joints in each camera view. DexYCB fits the MANO hand model [7] jointly across all 8 views to recover accurate 3D hand trajectories.

The MANO model [7] represents the hand as a parametric mesh with pose parameters $\theta \in \mathbb{R}^{48}$ and subject-specific shape parameters $\beta \in \mathbb{R}^{10}$, mapping (θ, β) to a mesh of 778 vertices and 21 joint positions. Per-frame annotations are stored in `labels_*.npz` files, one per camera per frame, containing:

- `joint_3d` $\in \mathbb{R}^{21\times 3}$ — 3D hand joint positions in camera space (the output of MANO forward kinematics, stored directly).
- `pose_y` $\in \mathbb{R}^{4\times 3\times 4}$ — object poses as $[R \mid t]$ matrices in camera space.
- `pose_m` $\in \mathbb{R}^{51}$ — MANO pose and translation parameters.

Camera intrinsics and extrinsics are provided in `extrinsics.yml` under the session calibration directory.

3.1.2 Camera-to-World Coordinate Transform

All DexYCB annotations are stored in camera space. The ViViDex simulator trains in a table-relative world frame. ViViDex does not release its retargeting code, so we derive the required transform by inspecting the coordinate conventions in `process_dataset.py` from the ViViDex codebase.

The extrinsics file provides, for each camera serial c, a flat 12-element tuple representing the 3×4 matrix $E_c = [R_c \mid t_c]$ expressing that camera's pose in the capture coordinate system. An additional `apriltag` entry provides the table frame matrix $E_{\text{table}} = [R_{\text{table}} \mid t_{\text{table}}]$.

We construct the camera-to-world transform as:

$$\mathbf{T}_{\text{full}} = \mathbf{T}_{\text{dir}}\, \mathbf{T}_{\text{ctr}}\, \bar{E}_{\text{table}}^{-1}\, \bar{E}_c \tag{1}$$

where $\bar{E} \in \mathbb{R}^{4\times4}$ denotes the homogeneous extension of E, and:

$$\mathbf{T}_{\text{ctr}} = \begin{bmatrix} I_3 & \mathbf{d} \\ \mathbf{0}^\top & 1 \end{bmatrix}, \quad \mathbf{d} = [-0.6,\ -0.3,\ 0]^\top \text{ m} \tag{2}$$

$$\mathbf{T}_{\text{dir}} = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3}$$

The centering matrix $\mathbf{T}_{\text{ctr}}$ translates the origin to the center of the table workspace, and $\mathbf{T}_{\text{dir}}$ negates the x- and y-axes to align with the MuJoCo world convention. To transform a point $\mathbf{p}_{\text{cam}} \in \mathbb{R}^3$ from camera space to world space:

$$\tilde{\mathbf{p}}_{\text{world}} = \mathbf{T}_{\text{full}} \begin{bmatrix} \mathbf{p}_{\text{cam}} \\ 1 \end{bmatrix} \tag{4}$$

Object poses are transformed analogously as 4×4 homogeneous matrices. We verified the correctness of this transform by comparing `object_translation` and `object_orientation` against ViViDex's reference NPZ, obtaining zero mean absolute error across all 72 frames.

3.1.3 Motion Retargeting

Because the MANO and Adroit hands differ in kinematic structure and morphology, the human hand trajectory cannot be directly applied to the robot. ViViDex adopts the retargeting formulation from DexMV [6].

The Adroit hand is a 30-DoF robot with 25 rigid bodies: 6 global pose DOF (ARTx, ARTy, ARTz, ARRx, ARRy, ARRz), 2 wrist joints, and 22 finger joints across 5 fingers. Its kinematic structure is defined in an MJCF XML model with per-joint range limits. We target 6 keypoint positions corresponding to MANO joint indices $[0, 2, 6, 10, 14, 18]$ (palm, thumb middle, index middle, middle middle, ring middle, little middle) and Adroit bodies (`palm`, `thmiddle`, `ffmiddle`, `mfmiddle`, `rfmiddle`, `lfmiddle`).

Naive retargeting. We implement **naive retargeting** using `NaiveOptimizationRetargeting` from DexMV [6]. For each frame t, we solve:

$$\min_{q_r^t \in \mathcal{Q}} \sum_{j=1}^{6} \left\| \hat{x}_{rj}(q_r^t) - \psi_{hj}^t \right\|^2 + \alpha \left\| q_r^t - q_r^{t-1} \right\|^2 \tag{5}$$

where $q_r^t \in \mathbb{R}^{30}$ are Adroit joint angles at frame t, $\mathcal{Q}$ is the joint limit feasible set, $\hat{x}_{rj}(q_r^t)$ are target body positions via forward kinematics, ψ_{hj}^t are world-space human joint positions from Eq. 1, and α is a temporal smoothness weight. The problem is solved sequentially by NLopt, warm-starting each frame from q_r^{t-1}. Initialization uses joint limit midpoints $q_r^0 = \frac{1}{2}(q_{\min} + q_{\max})$.

Chain retargeting. **Chain retargeting** extends naive retargeting by incorporating finger orientation from MANO forward kinematics. The MANO model produces, in addition to joint positions, a sequence of $16 \times 4 \times 4$ global joint frame transforms $\{F_k^t\}_{k=1}^{16}$ per frame, where each $F_k^t \in SE(3)$ encodes the rotation and position of joint k in camera space. These frames are transformed to world space via:

$$\tilde{F}_k^t = \mathbf{T}_{\text{full}}\, F_k^t \tag{6}$$

The chain method uses $\{\tilde{F}_k^t\}$ to initialize finger joint angles on the Adroit hand before the NLopt solve, via `ChainMatchingPositionKinematicsRetargeting` from DexMV. Specifically, for each finger, the rotation component of $\tilde{F}_k^t$ is used to compute an initial estimate of the corresponding

Adroit proximal joint angle by solving for the joint rotation that best aligns the robot finger frame with the MANO frame. This initialization resolves a key failure mode of naive retargeting: without orientation guidance, the NLopt solver may converge to a local minimum where finger abduction joints are collapsed to their limits, producing a grasping posture where the hand approaches the object with the back of the hand rather than the palm as seen in Figure 2e

Algorithm 1 summarizes our full retargeting procedure.

Algorithm 1 Our Retargeting Pipeline (Naive and Chain variants)

Require: DexYCB labels $\{\texttt{joint\_3d}^t, \texttt{pose\_y}^t\}_{t=1}^{T}$, calibration $\mathbf{T}_{\text{full}}$, Adroit XML
Ensure: NPZ trajectory: `robot_qpos`, `robot_jpos`, `object_translation`, `object_orientation`

for $t = 1, \ldots, T$ **do**
  $\psi^t \leftarrow \mathbf{T}_{\text{full}} \cdot \texttt{joint\_3d}^t$ ▷ Transform hand joints to world space
  $(p^t_{\text{obj}}, \mathbf{q}^t_{\text{obj}}) \leftarrow \mathbf{T}_{\text{full}} \cdot \texttt{pose\_y}^t$ ▷ Transform object pose to world space
end for
$q^0 \leftarrow \frac{1}{2}(q_{\min} + q_{\max})$ ▷ Initialize at joint limit midpoints
for $t = 1, \ldots, T$ **do**
  if Chain variant **then**
    $q^t_{\text{init}} \leftarrow \texttt{FrameInit}(\tilde{F}^t, q^{t-1})$ ▷ Initialize fingers from MANO global frames
  else
    $q^t_{\text{init}} \leftarrow q^{t-1}$
  end if
  $q^t \leftarrow \arg\min_{q \in \mathcal{Q}} \sum_j \|\hat{x}_{rj}(q) - \psi^t_{hj}\|^2 + \alpha\|q - q^{t-1}\|^2$ ▷ NLopt solve
end for
Compute `robot_jpos` via FK on `robot_qpos`
Detect `pregrasp_step` as last frame before palm within threshold of object
Save NPZ with all arrays

3.2 Trajectory-Guided State-Based Policy Learning

3.2.1 Simulation Environment

The retargeted trajectory is kinematically plausible but not physically valid, the optimization does not enforce contact constraints. ViViDex trains a physically valid policy in MuJoCo simulation using PPO, guided by the reference trajectory. The observation is a 346-dimensional state vector concatenating joint positions, joint velocities, fingertip positions, and goal information derived from upcoming frames of the reference trajectory.

3.2.2 Trajectory-Guided Reward

ViViDex introduces a two-phase reward inspired by TCDM [3].

Pregrasp phase.

$$r_{\text{pre}} = -\sum_{j=1}^{6} \left\| \hat{x}_{rj}(q^t_r) - x^{*,t}_{rj} \right\|^2 \tag{7}$$

where $x^{*,t}_{rj}$ are the reference fingertip positions from `robot_jpos` at frame t.

Manipulation phase.

$$r_{\text{manip}} = -\lambda_{\text{obj}} \left\| p^t_{\text{obj}} - p^{*,t}_{\text{obj}} \right\|^2 + \lambda_{\text{lift}} \cdot \mathbf{1}[\Delta z > \tau_{\text{lift}}] \tag{8}$$

where $\lambda_{\text{obj}} = 10.0$, $\lambda_{\text{lift}} = 2.5$, $\tau_{\text{lift}} = 0.02\,\text{m}$. A curriculum scheduler advances training through the stages as success rates improve.

3.3 Unified Visual Policy Learning

The trained state-based policy generates physically valid rollouts, which are used to train a vision-based Behaviour Cloning policy taking proprioceptive state concatenated with 3D point clouds as input. This stage is outside our scope.

4 Experiments

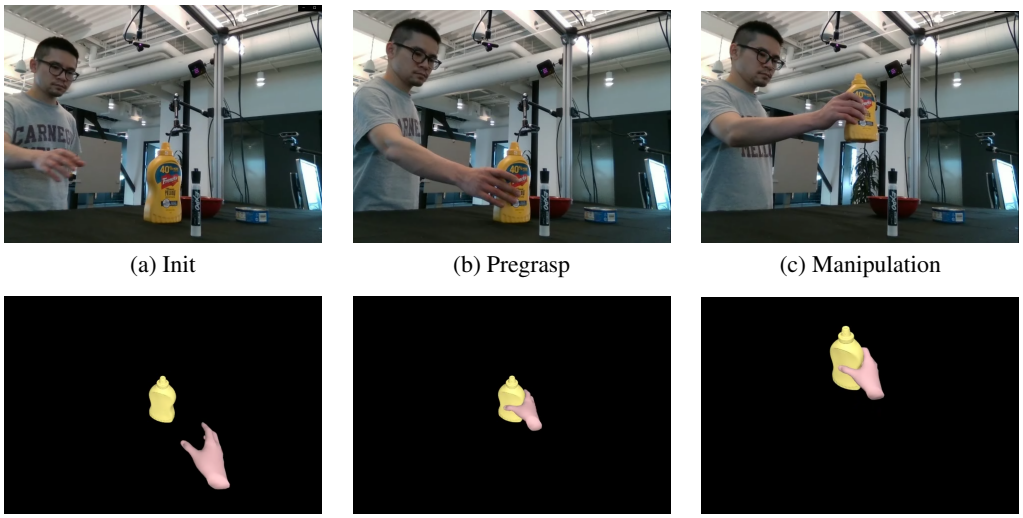
4.1 Experimental Setup

We evaluate the relocate task on a single mustard bottle demonstration from DexYCB (subject 01, sequence `20200709_143211`, $T = 72$ frames). We train three variants: **Baseline** (ViViDex's reference NPZ), **Chain** (our chain retargeting), and **Naive** (our naive retargeting). All runs use PPO with $n_{\text{envs}} = 32$, $n_{\text{steps}} = 4096$, batch size 256, 5 epochs per update, $\gamma = 0.95$, $\lambda_{\text{GAE}} = 0.95$, learning rate 10^{-5}, entropy coefficient 10^{-3}, and a two-layer MLP policy with hidden dimensions $[256, 128]$.

Compute constraints. MuJoCo physics simulation is CPU-bound; GPU utilization remained below 1% throughout training. ViViDex reports 2 hours per video on an A100, implying a cluster with $\sim$32 dedicated CPU cores. Our Colab instances provided 12 CPU cores at 2.20 GHz (Intel Xeon L4 instances), requiring 24–48 hours per session. Due to Colab's 24-hour session limit, each run was trained in 50M-step increments across multiple resumed sessions, reaching approximately 1.5×10^8 total steps. Training on ViViDex's trajectory did not show major improvement in object and hand success until around 90M timesteps during the second resumed session, illustrating the effect of the session limit constraint.

4.2 Reference Trajectory Extraction

Figure 1 shows the original DexYCB demonstration and our MANO hand reconstruction. The coordinate transform in Eq. 1 achieves exact agreement with ViViDex's reference NPZ on object translation and orientation.



(a) Init (b) Pregrasp (c) Manipulation

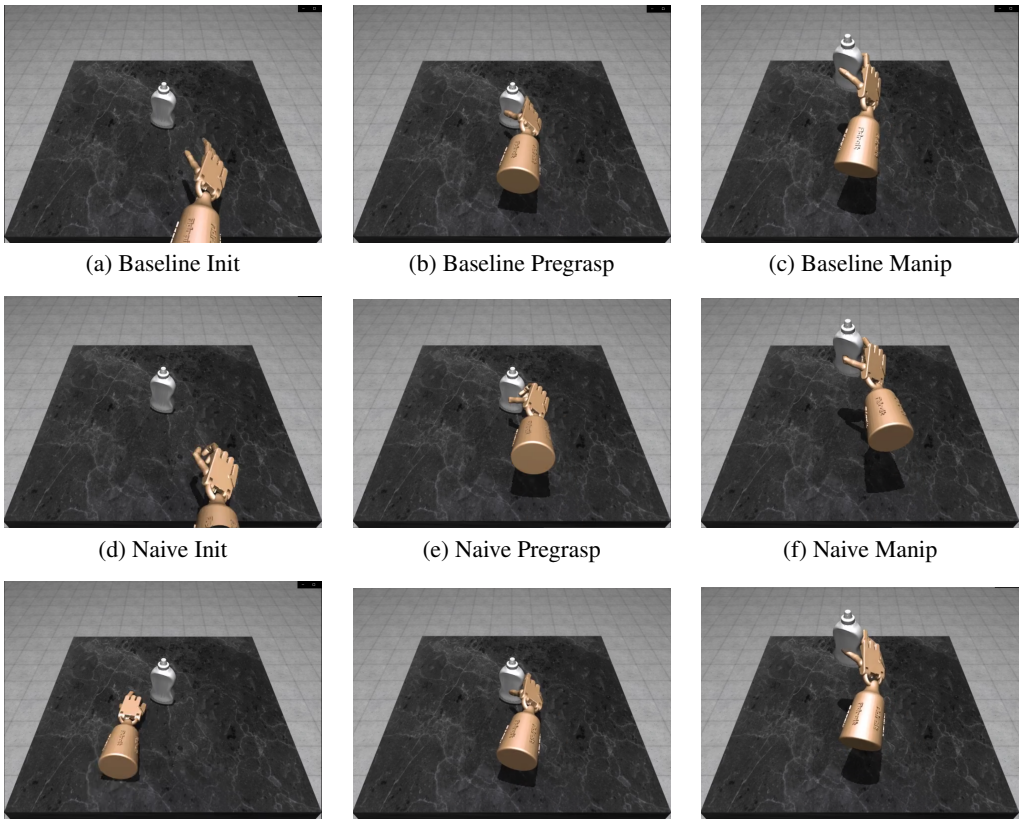
(d) MANO Init (e) MANO Pregrasp (f) MANO Manipulation

Figure 1: Top: DexYCB RGB demonstration. Bottom: our MANO hand mesh reconstruction with object, in the simulator world frame. The hand approaches (a,d), reaches pregrasp (b,e), and lifts the bottle (c,f).

4.3 Motion Retargeting

Figure 2 shows the kinematic replay of all three retargeted trajectories on the Adroit hand. Physics are not simulated; these visualizations confirm the quality of the reference joint angles only. The

baseline trajectory produces a natural palm-facing grasp. Naive retargeting fails to achieve sufficient finger abduction, the fingers do not spread wide enough to encircle the bottle, and the hand tends to approach with the dorsal side. Chain retargeting initializes with the hand oriented flat due to the MANO global frame alignment, but finger spread is substantially improved compared to naive.



(a) Baseline Init (b) Baseline Pregrasp (c) Baseline Manip

(d) Naive Init (e) Naive Pregrasp (f) Naive Manip

(g) Chain Init (h) Chain Pregrasp (i) Chain Manip

Figure 2: Kinematic replay of retargeted Adroit trajectories (no physics). Baseline (top) shows a natural grasping configuration. Naive (middle) exhibits insufficient finger abduction at pregrasp. Chain (bottom) shows improved finger spread from MANO global frame initialization.

4.4 PPO Training Results

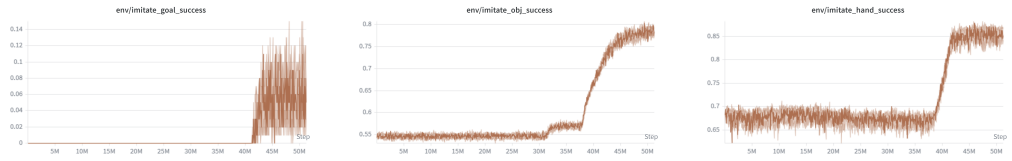
Table 1 reports quantitative metrics for all three trained policies at the end of training. Figure 3 shows the baseline training curves, and Figure 4 shows representative rollout frames.

Table 1: PPO training results on the Adroit hand mustard bottle relocate task ($n_{\text{envs}} = 32$, 12 CPU cores, $\sim 1.5 \times 10^8$ steps).

Method	Goal Success	Object Success	Hand Success	Grad. Updates
Baseline (ViViDex ref.)	**0.190**	**0.812**	**0.887**	197,745
Chain (ours)	0.000	0.547	0.841	198,695
Naive (ours)	0.000	0.519	0.180	222,145

4.5 Analysis

Retargeting quality determines task success. The baseline trajectory achieves 19% goal success and 81% object success, while neither of our reproduced trajectories achieves any goal success. The baseline training curve (Figure 3) reveals that significant improvement only occurs after approximately



(a) Goal Success (b) Object Success (c) Hand Success

Figure 3: Baseline training curves (ViViDex reference trajectory). All three metrics remain flat until approximately 40M steps (90M when added to previous checkpoint training steps), at which point hand success jumps from 0.68 to 0.85, object success rises sharply from 0.55 to 0.80, and goal success first appears near 42M steps (92M). This phase transition corresponds to the curriculum advancing to Stage 2 and the policy discovering a stable grasping strategy.

80M steps, when a sharp phase transition brings hand success from 0.68 to 0.85 and object success from 0.55 to 0.80. This suggests the curriculum stage transition is a critical threshold, and that our reproduced trajectories may simply require more training to cross it, particularly chain retargeting, which already achieves 0.84 hand success and 0.55 object success, comparable to baseline values before the phase transition.

Chain outperforms naive due to finger abduction. Chain retargeting substantially outperforms naive in hand success (0.84 vs. 0.18), confirming that MANO global frame initialization resolves the finger abduction failure mode. However, neither variant achieves goal success, indicating that the remaining quality gap from ViViDex's reference trajectory is the binding constraint.

CPU throughput is the binding compute constraint. GPU utilization remained below 1% throughout all training runs. The binding constraint is CPU clock speed for MuJoCo simulation. ViViDex's 2-hour A100 training time is misleading for reproduction purposes, as it implicitly assumes $\sim$32 dedicated CPU cores running at high clock speeds on a compute cluster.

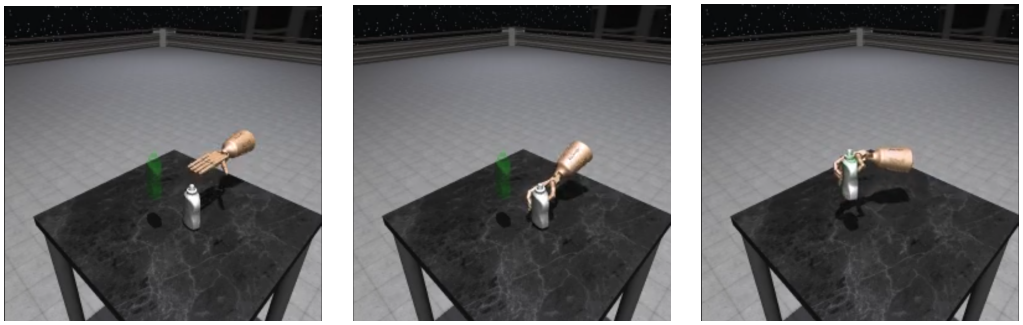
5 Conclusion

We have reproduced the first two stages of the ViViDex pipeline from scratch: reference trajectory extraction from DexYCB and trajectory-guided PPO training on the MuJoCo Adroit hand. Our primary contribution is a complete re-implementation of the retargeting pipeline, which ViViDex does not release, including derivation of the camera-to-world coordinate transform and two retargeting variants.

Our results confirm that retargeting quality is the critical bottleneck for task success. The baseline trajectory (ViViDex's undisclosed reference) achieves 19% goal success, while our reproduced trajectories do not. However, our chain retargeting achieves comparable pre-transition metrics to the baseline (0.84 hand success, 0.55 object success vs. 0.68 and 0.55 for baseline at 80M steps), suggesting that extended training may close the gap.

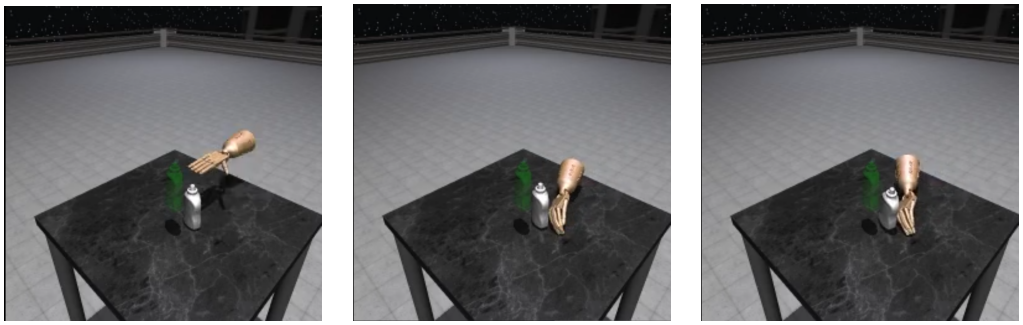
Limitations. We were unable to train to full convergence due to Colab session limits (50M steps per session, 24–48 hours each). We did not reproduce the vision-based Behaviour Cloning stage, the Allegro hand experiments, or comparisons against DexMV's SOIL/GAIL/DAPG methods. Our chain retargeting diverges from ViViDex's undisclosed pipeline in ways that cannot be fully characterized without access to their code.

Future directions. The most promising direction is extended chain retargeting training beyond 1.5×10^8 steps, given its strong hand success. Additionally, MJX (MuJoCo's GPU-accelerated backend) could enable true 32-environment parallelism on GPU, potentially reproducing ViViDex's training speed on cloud hardware. Finally, deep learning based retargeting in latent space can yield promising results as an alternative to the retargeting pipeline used in this reproduction. [4]

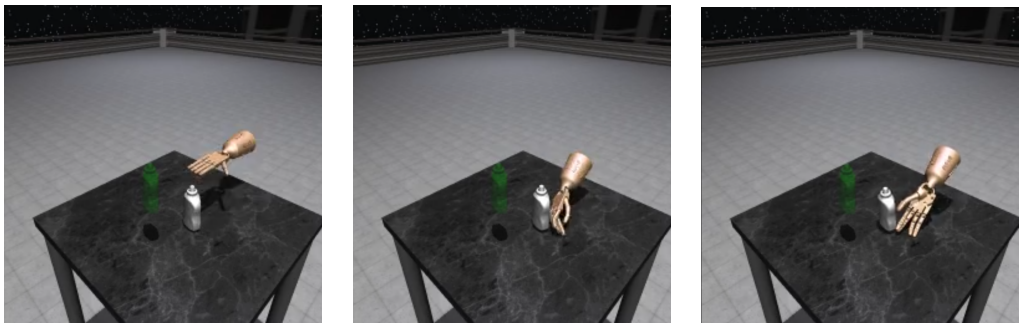


(a) Pretrained Init (b) Pretrained Pregrasp (c) Pretrained Manip

(d) Baseline Init (e) Baseline Pregrasp (f) Baseline Manip



(g) Chain Init (h) Chain Pregrasp (i) Chain Manip



(j) Naive Init (k) Naive Pregrasp (l) Naive Manip

Figure 4: PPO rollout frames. The green marker is the target object position; the dark circle marks the initial position. Top: ViViDex pretrained checkpoint, successful lift and relocation. Second: our baseline policy, partial success, achieves contact and some lift. Third/fourth: chain and naive policies, neither achieves lift; chain makes more consistent contact than naive.

References

[1] Yu-Wei Chao, Wei Yang, Yu Xiang, Pavlo Molchanov, Ankur Handa, Jonathan Tremblay, Yashraj S. Narang, Karl Van Wyk, Umar Iqbal, Stan Birchfield, Jan Kautz, and Dieter Fox. Dexycb: A benchmark for capturing hand grasping of objects, 2021.

[2] Zerui Chen, Shizhe Chen, Etienne Arlaud, Ivan Laptev, and Cordelia Schmid. Vividex: Learning vision-based dexterous manipulation from human videos, 2025.

[3] Sudeep Dasari, Abhinav Gupta, and Vikash Kumar. Learning dexterous manipulation from exemplar object trajectories and pre-grasps, 2023.

[4] Kailin Li, Puhao Li, Tengyu Liu, Yuyang Li, and Siyuan Huang. Maniptrans: Efficient dexterous bimanual manipulation transfer via residual learning, 2025.

[5] Qingtao Liu, Yu Cui, Qi Ye, Zhengnan Sun, Haoming Li, Gaofeng Li, Lin Shao, and Jiming Chen. Dexrepnet: Learning dexterous robotic grasping network with geometric and spatial hand-object representations, 2023.

[6] Yuzhe Qin, Yueh-Hua Wu, Shaowei Liu, Hanwen Jiang, Ruihan Yang, Yang Fu, and Xiaolong Wang. Dexmv: Imitation learning for dexterous manipulation from human videos, 2022.

[7] Javier Romero, Dimitrios Tzionas, and Michael J. Black. Embodied hands: modeling and capturing hands and bodies together. *ACM Transactions on Graphics*, 36(6):1–17, November 2017.

Appendix: Teaching Evaluation Submission

The following screenshot confirms completion of the SP26 ECE 228 course teaching evaluation prior to the June 6, 2026 deadline.