

# Learning Dexterous Manipulation Using Contact Wrench Guidance From Human Demonstration

Xinghao Zhu<sup>\*‡</sup>, Zixi Liu<sup>\*</sup>, Shalin Jain<sup>\*</sup>, Chenran Li<sup>†</sup>, Milad Noori<sup>†</sup>, Huihua Zhao, John Welsh, Michael Andres Lin, Wei Liu, Tingwu Wang, Xingye Da, Zhengyi Luo, Vishal Kulkarni, Naema Bhatti, Yuke Zhu, Linxi Fan, Bowen Wen, Danfei Xu, Soha Pouya, Yan Chang<sup>‡</sup>

NVIDIA

<sup>\*</sup>Equal Contribution   <sup>†</sup>Core Contributor   <sup>‡</sup>Project Lead and Corresponding Author

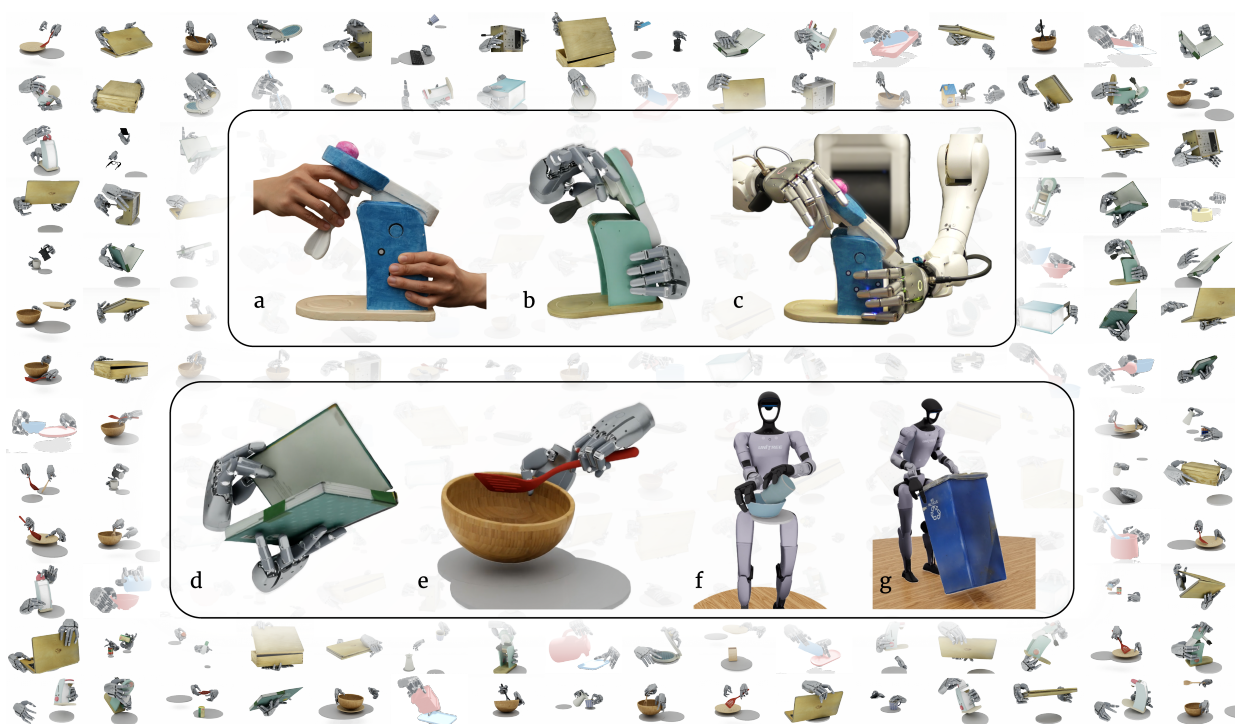


Figure 1: **CHORD** learns dexterous, contact-rich policies from human demonstration through contact wrench-space guided reinforcement learning. *Foreground:* (a) the framework takes reference hand-object trajectories from human demonstration and (b) learns a robot policy in simulation, which (c) transfers to real-robot execution. CHORD enables manipulation of (d) articulated objects, (e) rigid objects, and generalizes to (f, g) whole-body embodiments. *Background* shows CHORD on our large-scale benchmark of bimanual manipulation tasks.

**Abstract:** Dexterous robot manipulation can benefit from the abundance of human demonstrations, but transferring such demonstrations to robot policies remains challenging. We present Contact Wrench Guidance from Human Demonstration in Robotic Dexterous Manipulation (CHORD), a framework for long-horizon manipulation of rigid and articulated objects with reinforcement learning. The key idea is object-centric contact wrench space guidance: we represent human and robot motions by the forces and torques they can induce on the object, enabling similarity to be measured by the induced instantaneous motions. This guidance makes reinforcement learning more scalable for contact-rich dexterous manipulation. We further introduce a large-scale simulation benchmark with 4,739 bimanual dexterous manipulation tasks, constructed from motion-capture datasets and reconstructed in-house videos. Evaluated on 1,831 benchmark tasks, CHORD achieves an average success rate of 82.12%, demonstrating strong scalability. CHORD also generalizes to whole-body manipulation from hand-only and third-person demonstrations, achieving a 90.77% success rate, and the learned policies transfer to the real world in both open-loop and closed-loop settings. Videos and code are available at [https://nvidia-isaac.github.io/video\\_to\\_data/chord/](https://nvidia-isaac.github.io/video_to_data/chord/).

## 1. Introduction

Robotic dexterous hands offer a compelling platform for general-purpose manipulation because of their morphological similarity to human hands, enabling robots to exhibit human-level dexterity. Leveraging human demonstrations has shown promise in advancing dexterous manipulation by mitigating the exploration challenges of policy optimization [1–6] and supporting the learning of transferable human-robot representations [7–11]. However, existing methods that use human demonstrations for optimization often rely on brittle assumptions for how the demonstration should be transferred, while representation-learning approaches typically require aligned human-robot data for each task and object, limiting their scalability beyond curated settings. This motivates our work: *How can a robot learn transferable and scalable task knowledge from human demonstrations?*

A dexterous robot cannot simply replay human hand motion to reproduce human manipulation behavior [1, 6, 9]. Differences in morphology, kinematics, and hand geometries require the robot to use different motions and contacts to achieve the same functional effect on the object. Our key insight is that contact provides a natural bridge between human demonstrations and robot actions. Rather than matching 3D human contact locations [1–3], the robot should find contacts that induce object motions aligned with the human demonstration. To this end, we represent contacts in an object-centric wrench space, capturing the force-torque directions and object motions supported by each contact configuration [12]. This representation is shared across embodiments: humans and robots may differ in contact locations, contact counts, and hand morphology, but their contacts can still be compared in wrench space with respect to the object-level motions they are capable of inducing.

Concretely, we present **C**ontact **W**rench **G**uidance from **H**uman **D**emonstration in **R**obotic **D**exterous **M**anipulation (CHORD), a framework for learning dexterous, contact-rich manipulation from human demonstrations. Given a human reference, CHORD trains a robot policy to match the demonstrated object-state evolution, using contact wrench space rewards to encourage physically aligned contacts. Our method enables long-horizon RL across diverse rigid and articulated manipulation tasks without task-specific robot demonstrations. We further introduce a large-scale simulation benchmark of 4,739 bimanual dexterous manipulation tasks, built from open-source motion-capture datasets [13–16] and reconstructed in-house videos. Across 1,831 tasks, CHORD achieves an average success rate of 82.12%, outperforming prior baselines. The same formulation extends to whole-body manipulation from hand-only and third-person demonstrations, and transfers to a real bimanual dexterous robot in both open-loop and closed-loop settings.

Our contributions are: **(1)** CHORD, a wrench-space contact-guided framework for transferring human demonstrations to dexterous robot policies with reinforcement learning; **(2)** a benchmark of 4,739 long-horizon bimanual dexterous manipulation tasks spanning rigid, articulated, and multi-object interactions; and **(3)** validation of CHORD in large-scale simulation, achieving a 82.12% success rate on 1,831 tasks and 90.77% success in whole-body manipulation tasks, with successful deployment on real hardware.

## 2. Related Work

Learning from human demonstration is an increasingly prominent approach for enabling robots to acquire skills from existing experts [17], demonstrating success in learning whole-body control [18–20] and representation learning [7–9, 21]. However, dexterous manipulation with human demonstration remains difficult due to high-dimensional hand actions and non-smooth contact dynamics, where small motion changes can abruptly change object behavior [22–27]. Prior work uses reinforcement learning in simulation [28–30]. These methods rely on task-level imitation objectives such as object and hand-reference tracking, while recent approaches add contact guidance to improve exploration. ManipTrans [2] rewards contact forces near demonstrated hand-object interactions, whereas DexMachina [1] and SPIDER [3] use demonstrated *contact locations* together with curricula such as virtual object control. The virtual object controller (VOC) applies an auxiliary wrench to move the object along the reference trajectory during early training. This makes the manipulation problem smoother by reducing the need for the policy to immediately discover precise contact timing and contact forces. Instead of receiving a useful object-tracking reward only after successful contact, the policy observes a denser learning signal.

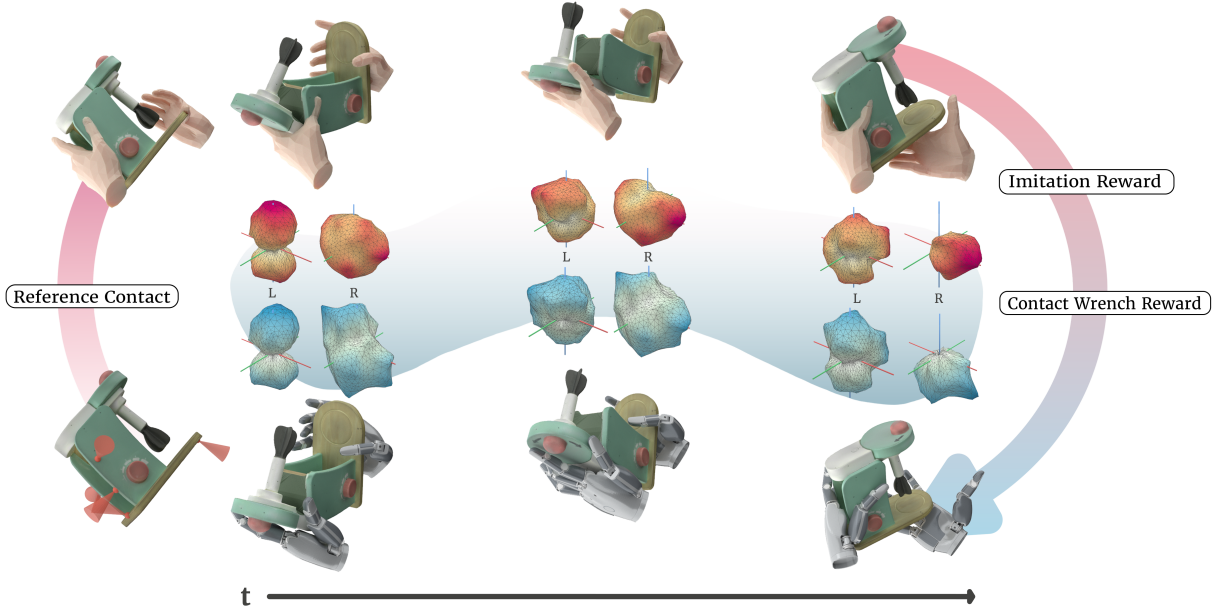


Figure 2: **CHORD** combines imitation and contact guidance rewards for RL training. *Left*: Contact wrench references extracted from the human demonstration, with the corresponding contact positions and friction cones (red) in the lower panel. *Top*: Human demonstration of a mixer-closing task. *Middle*: Evolution of per-hand contact wrenches, visualized with force manifolds, throughout the task, where red denotes the human contact wrench, and blue denotes the corresponding contact wrench generated by (*Bottom*) CHORD-trained policy.

The VOC eases exploration; however, it can introduce local optima, leading to ineffective contact behavior. Specifically, the contact location-based reward [1, 3] encourages the robot to touch object regions similar to those touched by the human demonstrator. However, contact location alone does not determine the effect of a contact (Section 3.1). The same object region can produce different object motions depending on the contact surface normal and force direction. An alternative is to supervise contact in *wrench space*, the space of forces and torques that contacts can apply to the object [12, 25, 31, 32]. Contact wrench rewards in prior RL work [33–36], optimize for static stability, making them effective for grasping but too rigid for general robot manipulation. In contrast, we leverage the wrench space, for the first time, to measure the similarity between a human demonstration and a robot execution in terms of the *induced motion* from contact in the RL training. This allows contacts with different locations, contact counts, or hand morphologies to be compared by their object-level mechanical effects during RL training. Crucially, this metric accommodates transient, non-force-closure phases of long-horizon manipulation such as pushing, levering, and sliding, as well as the manipulation of articulated objects.

### 3. Method

Our goal is to learn long-horizon manipulation of rigid and articulated objects from human demonstrations. We consider objects with  $K$  rigid bodies, either as  $K$  separate rigid objects or as an articulated object with  $K$  parts. Each demonstration provides a reference trajectory  $\tau^{\text{ref}} = \{x_t^{\text{human}}, x_t^{\text{object}}\}_{t=1}^H$ , where  $x_t^{\text{human}}$  denotes 3D human hand keypoints and  $x_t^{\text{object}} = \{x_t^{\text{object},k}\}_{k=1}^K$  denotes object-part poses with  $x_t^{\text{object},k} \in \text{SE}(3)$  for each part (Figure 2). We begin by retargeting the hand motion by solving inverse kinematics from human keypoints to robot configurations  $x_t^{\text{robot}}$ . The policy  $\pi(a_t | o_t^{\text{robot}}, o_t^{\text{object}}; x_t^{\text{robot}}, x_t^{\text{object}})$  then predicts actions from current robot observations  $o_t^{\text{robot}}$ , object observations  $o_t^{\text{object}}$  and reference motion  $x_t^{\text{robot}}, x_t^{\text{object}}$ , with the objective that rollout object poses  $\{s_t^{\text{object}}\}_{t=1}^H$  track the reference trajectory  $\{x_t^{\text{object}}\}_{t=1}^H$ .

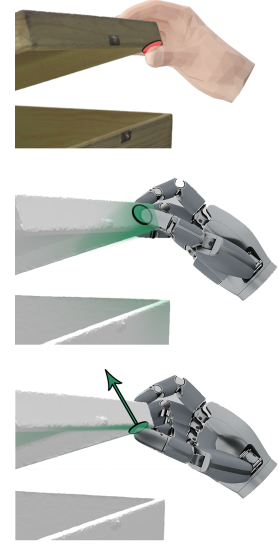
We present Contact Wrench Guidance from Human Demonstration in Robotic Dexterous Manipulation (CHORD), a framework for long-horizon, contact-rich manipulation of rigid, articulated objects, and multi-object interactions. We first formulate the reinforcement learning rewards (Section 3.1). We then describe how

CHORD enables efficient and robust learning from noisy human references, and how it extends to whole-body manipulation (Section 3.2). Finally, we introduce our dexterous manipulation benchmark (Section 3.3). More details on the MDP setup can be found in the Appendix.

### 3.1. RL with Wrench Space Contact Guidance

We train the policy with task-tracking, motion-imitation, and contact-guidance rewards,  $r = r_{\text{task}} + r_{\text{imit}} + r_{\text{contact}}$  (Figure 2) with the VOC [1]. The task reward encourages the rollout to follow the reference object motion, including both object-part poses and inter-part relations:  $r_{\text{task}} = \exp\left(-\frac{\sum_{k=1}^K \|x_t^{\text{object},k} \ominus s_t^{\text{object},k}\|_2^2}{\text{var}_{\text{obj}}}\right) + r_{\text{relative}}$ , where  $\ominus$  denotes the pose difference in SE(3). For multi-object interaction, tracking each object part independently is often insufficient for tasks that depend on precise object-object geometry, such as insertion, pouring, scooping, and tool use. We therefore add a relative reward,  $r_{\text{relative}} = m(t) \exp\left(-\frac{e_{\text{object}}}{\text{var}_{\text{rel}}}\right)$ , where  $m(t)$  activates the term only during object-object interaction phases, and  $e_{\text{object}}$  measures the inter-object pose error. The imitation reward regularizes the robot toward the retargeted human motion,  $r_{\text{imit}} = \exp\left(-\frac{\|x_t^{\text{robot}} - s_t^{\text{robot}}\|_2^2}{\text{var}_{\text{imit}}}\right)$ .  $\text{var}_{\text{obj}}, \text{var}_{\text{rel}}, \text{var}_{\text{imit}}$  are variances for the exponential kernel.

As discussed in Section 2, location-based contact rewards are often insufficient. In the right-side box-opening task, the top row shows the human demonstration, where the thumb contacts the underside of the lid. The middle-row heatmap visualizes the guidance score assigned to each candidate contact point by a location-based contact reward. Under this reward, the optimized contact is spatially close to the human contact but it is physically mismatched: the contact normal is nearly perpendicular to that of the demonstration. In contrast, the bottom-row heatmap shows the guidance using our proposed contact wrench-space reward, which encourages contacts capable of producing object motion consistent with the human demonstration.



At each time step, for each rigid part  $k$ , we extract  $c_{h,k}$  contact positions  $p_{h,k} = \{p_{h,k}^i\}_{i=1}^{c_{h,k}}$  and contact normals  $n_{h,k} = \{n_{h,k}^i\}_{i=1}^{c_{h,k}}$  from the human demonstration  $\tau^{\text{ref}}$  in the object frame [1]. For any feasible contact force  $f_{h,k}^j \in \mathbb{R}^3$  at contact  $i$ , the induced primitive wrench is defined as  $w_{h,k}^{i,j} = \begin{bmatrix} f_{h,k}^j \\ p_{h,k}^i \times f_{h,k}^j \end{bmatrix}^\top \in \mathbb{R}^6$ .

To characterize the contact wrenches for  $c_{h,k}$  contact points, we approximate the Coulomb friction cone at each contact with a polyhedral cone containing  $d$  unit-magnitude edge forces. The  $j$ -th edge force at contact  $i$  induces a primitive wrench  $w_{h,k}^{i,j}$ . We collect all primitive wrenches for the hand-part pair into the wrench matrix:

$$\mathcal{W}_{h,k} = \begin{bmatrix} w_{h,k}^{1,1} & \cdots & w_{h,k}^{1,d} & \cdots & w_{h,k}^{c_{h,k},1} & \cdots & w_{h,k}^{c_{h,k},d} \end{bmatrix} \in \mathbb{R}^{6 \times (c_{h,k}d)}. \quad (1)$$

The wrench matrix  $\mathcal{W}_{h,k}$  represents the force-torque directions that the human contacts can apply to object part  $k$ , and therefore captures their motion-inducing capability. However, directly comparing two wrench matrices, such as those from a human and a robot, is difficult because they may have different numbers of columns and arbitrary primitive-wrench orderings. We therefore propose to compare their induced wrench geometry using a support function. Given  $b$  pre-sampled unit directions  $\mathcal{B} \in \mathbb{R}^{6 \times b}$ , we define the support function of  $\mathcal{W}_{h,k}$  as

$$\sigma_{h,k} = \max_{\text{col}} (\mathcal{B}^\top \mathcal{W}_{h,k}) \in \mathbb{R}^b \quad (2)$$

where the maximum is taken independently across the columns of  $\mathcal{B}^\top \mathcal{W}_{h,k} \in \mathbb{R}^{b \times (dc_{h,k})}$ . We compute the robot support function  $\sigma_{r,k}$  in the same way.

We then define the contact wrench-space reward by comparing the robot support function  $\sigma_{r,k}$  against the human reference function  $\sigma_{h,k}$  with a relative tolerance  $\beta$ . For each object part  $k$ , the reward is defined as:

$$r_{\text{cws}}^k = \exp\left(-\frac{\|\max(0, (1-\beta)\sigma_{h,k} - \sigma_{r,k})\|_2^2}{v_{\text{cws}}} - \frac{\|\max(0, \sigma_{r,k} - (1+\beta)\sigma_{h,k})\|_2^2}{v_{\text{cws}}}\right), \quad (3)$$

where the maximum is applied element-wise. The first term rewards robot contacts whose support is larger than the lower tolerance bound, while the second term rewards support values that do not exceed the upper tolerance bound.  $v_{\text{cws}}$  is the variance for the exponential kernel.

Since the exponential kernel in Equation (3) always yields positive rewards, we additionally penalize contact mismatches. In particular, we penalize unintended contacts  $r_{\text{unintend}}^k$  when  $\sigma_{\text{h},k} = 0$  but  $\sigma_{\text{r},k} > 0$ , and missed contacts  $r_{\text{miss}}^k$  when  $\sigma_{\text{h},k} > 0$  but  $\sigma_{\text{r},k} = 0$ . This prevents the policy from creating extra contacts not present in the human demonstration, or fails to reproduce contacts that are required by the reference.

### 3.2. Efficient, Robust, and Generalizable Policy Learning

In addition to the contact reward in Section 3.1, we incorporate several mechanisms to improve training efficiency and robustness. First, we reset the simulator to arbitrary states along the reference trajectory and keep the VOC fully active for a short stabilization window, allowing the robot to recover contact before assistance is annealed. Second, we perturb object parts with wrenches sampled from the human contact wrench matrix  $\mathcal{W}_{\text{h},k}$ , producing task-relevant disturbances aligned with the demonstrated contact mechanics. We also use a residual action space with retargeted robot motion as the prior [2], and anneal the VOC through a curriculum [1].

When contact estimates are noisy due to hand-object misalignment, as in demonstrations reconstructed from RGB videos, directly matching human contact wrenches can be unreliable. In such scenarios we pivot from direct matching to a reduced contact wrench guidance objective that rewards the robot for generating positive support along each wrench-space basis direction:  $r_{\text{fc}}^k = \frac{1}{B} \sum_{b=1}^B \mathbb{1}[\sigma_{r,k,b} > \epsilon]$ , where  $\sigma_{r,k,b}$  is the robot support value for object part  $k$  along basis direction  $b$ , and  $\epsilon$  is a small threshold. When maximized, this is equivalent to force closure.

CHORD naturally extends to whole-body manipulation with either *hand-only* or *whole-body* references. For hand-only references, such as egocentric reconstructions, we train an inpainting module [37] to predict full-body motion from end-effector trajectories, then apply the same RL formulation with contact wrench-space rewards. For whole-body references, such as third-person reconstructions, full-body motion is available, but finger reconstruction is often inaccurate, causing noisy hand-object interactions [38]. We therefore train with the reduced force-closure objective  $r_{\text{fc}}^k$ . For additional details, please see Appendix E.

### 3.3. Dexterous Manipulation Benchmark with Human References

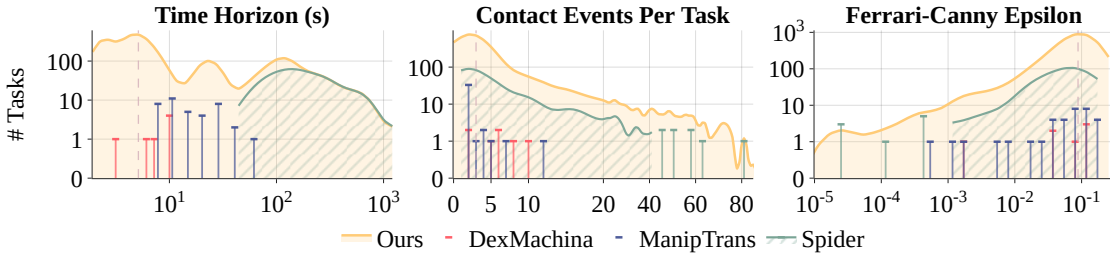


Figure 3: **Benchmark Distribution.** Our tasks span diverse, large-scale settings with long horizons and dense contact events, requiring dexterous contact strategies and coordinated manipulation.

To further support the research on learning from human demonstrations and evaluate the proposed algorithm at a larger scale, we collect and process common human manipulation datasets [13–16, 39–41] and import them into Isaac Lab for policy training [42]. In total, we curate 4,739 open-source tasks and convert them into simulatable, trainable environments. We also reconstruct additional demonstrations from in-house videos using hand-object reconstruction and tracking. The resulting benchmark covers bimanual manipulation of single and multiple rigid objects, as well as articulated objects (Figure 1). We evaluate task diversity against prior works [1–3] across three core metrics: time horizon length, the number of contact events per task, and grasp stability, measured by the Ferrari-Canny Epsilon-metric [32]. As shown in Figure 3, our library includes more tasks, longer horizons, and denser contact events. These tasks require dexterous contact strategies beyond simple grasping and coordinated use of both hands for successful manipulation. Additional

details, including metric definitions, the video reconstruction, and post-processing procedures, are provided in Appendix D.

## 4. Experiments

We conduct experiments to evaluate the scalability of CHORD on large-scale tasks and compare it against baseline methods (Section 4.1). We then analyze the proposed contact-wrench reward, showing its advantages over prior objectives (Section 4.2), its correlation with manipulation success (Section 4.3), and its utility in learning long-horizon dexterous tasks (Section 4.4). Next, we demonstrate the generalizability of CHORD to whole-body manipulation (Section 4.5) and real-world deployment (Section 4.6). Finally, we provide a pilot study that compares RL-based dexterous manipulation with teleoperation (Section 4.7).

### 4.1. Quantitative Analysis



Figure 4: **Contact Rich Tasks:** The learned policies handle complex object interactions including picking, placing, handing over, stirring a bowl, and using articulations across objects of varying geometry.

**Evaluation on Large Scale Tasks.** We evaluate CHORD on 1,831 tasks sampled from our benchmark Section 3.3, using the same hyperparameters across all tasks, including VOC gains, curriculum schedules, and reward weights. To the best of our knowledge, CHORD is the first RL-based dexterous manipulation method evaluated at this scale.

Qualitatively, Figure 4 shows that CHORD solves diverse contact-rich tasks, including single-hand and bimanual pick-and-place, handover, coordinated manipulation, articulated-object manipulation, and tool use. The learned behaviors complete the tasks while closely following the human demonstrations. With only simple keypoint-based inverse kinematics, CHORD remains robust to retargeting noise and hand-object interpenetration by using wrench-space guidance to avoid misaligned contact effects. Additional results are provided in the supplementary video and Appendix.

Quantitatively, we report the task completion ratio as the primary metric. A rollout succeeds if it completes the task without object-centric termination, defined by position error above 15cm or rotation error above  $40^\circ$ . Completion ratio is subsequently defined as the ratio of successful rollouts to total rollouts, and we consider a task success if the completion ratio is larger than 0.7. Figure 5 (left) shows consistently strong performance over 1,831 tasks across diverse object types and horizons. We find that stable training requires the contact wrench reward to converge before decaying the VOC; harder tasks with small contact regions or delicate interactions generally require longer optimization.

**Baselines Comparison.** We compare CHORD against state-of-the-art dexterous manipulation methods that learn from human demonstrations, including ManipTrans [2], DexMachina [1], and SPIDER [3]. For each baseline, we evaluate it on the method’s original task suite using its corresponding metrics with Sharpa hand. After excluding tasks that cannot be reliably simulated in Isaac Lab due to asset-quality issues, we evaluate our method using 8 tasks from ManipTrans (MT), 3 from SPIDER (SP), and all 7 from DexMachina

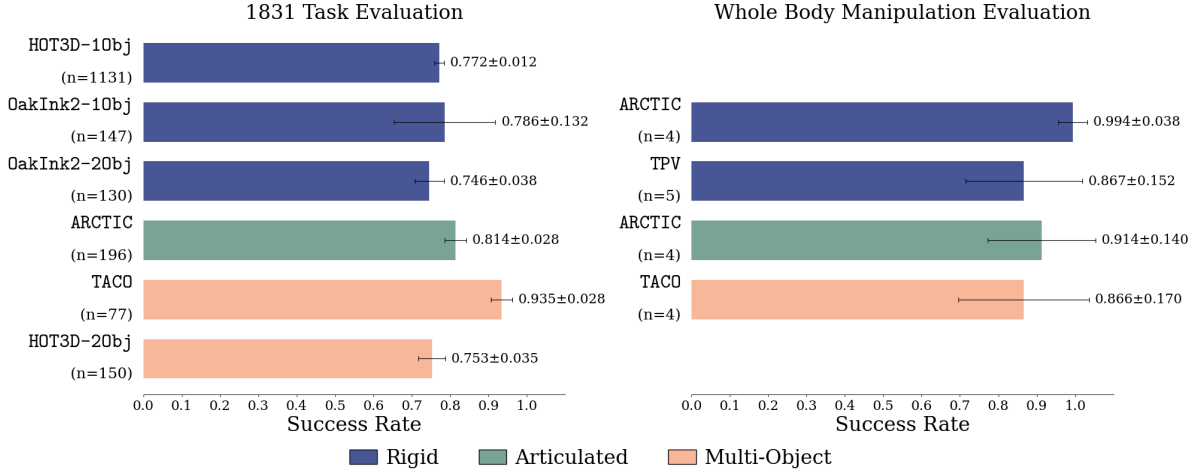


Figure 5: CHORD success rates on *left*: 1,831 manipulation tasks across four datasets, spanning both single-object and multi-object tasks involving rigid and articulated objects, and *right*: 17 of these subtasks, fully grounded on a humanoid robot equipped with articulated hands.

Task Suite	Metric	Ref. Method	Ref. Score	Our Score
DM	AUC	DexMachina	$0.232 \pm 0.214$	<b><math>0.687 \pm 0.358</math></b>
MT	MT-SR	ManipTrans	0.428	<b>0.639</b>
SP	SP-SR	Spider	$0.333 \pm 0.488$	<b><math>0.359 \pm 0.482</math></b>
Ours-1	AUC	DexMachina	$0.211 \pm 0.138$	<b><math>0.895 \pm 0.052</math></b>
Ours-1	SP-SR	Spider	$0.133 \pm 0.327$	<b><math>0.999 \pm 0.000</math></b>
Ours-2	SP-SR	Spider	$0.533 \pm 0.503$	<b><math>0.982 \pm 0.022</math></b>

Table 1: **Comparison with prior methods across task suites and metrics.** Scores are comparable within each row because baselines use different task suites and evaluation metrics.

(DM). We also select 9 tasks from our benchmark, covering single-object manipulation (Ours-1) and two-object interaction tasks (Ours-2). Since baselines have different evaluation metrics and protocols, we report each method using the metrics from the corresponding baseline: area under the object-tracking-error curve (AUC) from DexMachina [1], success rate from SPIDER (SP-SR) [3], and success rate from ManipTrans (MT-SR) [2]. For baseline reproduction details, see Appendix G.

Table 1 shows that CHORD achieves high success rates on both the baseline suites and our task suite, consistently matching or outperforming state-of-the-art methods under their original metrics. These results agree with the large-scale evaluation in Figure 5, showing that CHORD provides both broad manipulation capability and the precision needed for task completion. Moreover, prior methods are limited to subsets of rigid-object, articulated-object, or multi-object manipulation, whereas CHORD succeeds across all three categories. This broader coverage is important for building manipulation capabilities that are robust across task types. Additional robustness analysis and ablations are provided in Appendix A.

## 4.2. Examining Contact Wrench Guidance

We compare three reward formulations with decreasing levels of contact guidance: CHORD (contact wrench support reward), Position Only (contact position reward as in DexMachina), and No Contact (task and imitation rewards only). Position Only questions the need for contact wrench guidance, and No Contact questions the need for any contact guidance.

We select two representative sequences that are long-horizon and involve articulated object manipulation capabilities from ARCTIC [13]: **box grab**, **mixer use**. All variants share the same non-contact reward terms, including object pose tracking rewards, hand keypoint tracking reward, and hand joint position tracking

reward. The only difference between the compared variants is the form of contact supervision.

Sequence	CHORD (SR)	Position Only (SR)	No Contact (SR)
box grab	<b>0.702 ± 0.257</b>	0.334 ± 0.141	0.384 ± 0.206
mixer use	<b>0.894 ± 0.023</b>	0.624 ± 0.166	0.423 ± 0.273

Table 2: Performance comparison across different reward formulations. Values are reported as mean  $\pm$  standard deviation across 3 seeds, 4096 environments per seed. Best result per sequence is highlighted in bold.

As shown in Table 2, performance improves as richer contact guidance is introduced. CHORD achieves the highest performance, demonstrating the benefit of supervising manipulation in contact wrench space. Replacing wrench-based supervision with contact position rewards in Position Only leads to a noticeable degradation in performance, indicating that matching contact locations alone is insufficient for faithfully reproducing contact-rich manipulation behaviors. Finally, the No Contact baseline exhibits the weakest performance overall, highlighting the limitations of relying solely on kinematic tracking objectives for contact-rich manipulation tasks.

### 4.3. Correlation between Contact Wrench Reward and Task Success

To assess whether the contact-wrench-support (CWS) reward is predictive of downstream task performance, we aggregated 1,831 runs in our benchmark. For each run, we extracted the CWS reward and the task success (evaluation completion ratio) over 1,831 tasks. The achieved CWS reward was normalized on a per-sequence basis to render magnitudes comparable across datasets of differing contact density and trajectory length in Figure 6.

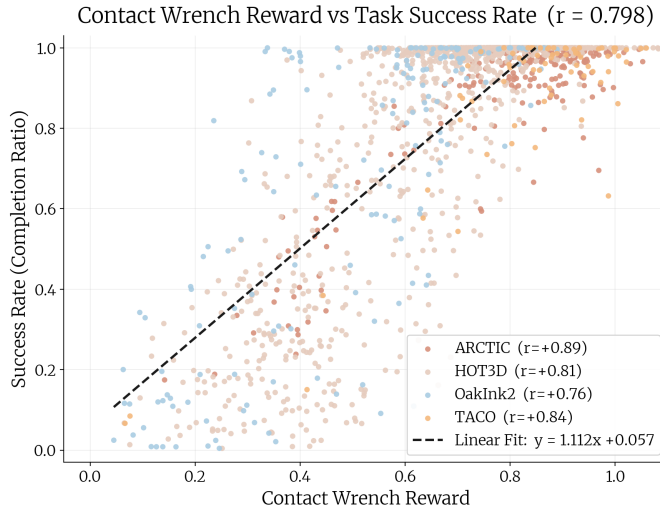


Figure 6: Correlation between contact wrench reward and task success rate.

Across all runs, we observe a strong positive association between the normalized CWS reward and task success (Pearson  $r \approx 0.80$ ), which holds consistently within every dataset ( $r = 0.76$ – $0.89$ ). The relationship is monotonic but saturating: success rises steeply with CWS reward and then plateaus near unity, so that an ordinary least squares line—reported here as a first-order summary—captures the overall trend (explaining roughly two-thirds of the variance) while necessarily understating the fit in the high-reward regime. These results indicate that the degree to which a policy satisfies the contact-wrench-support objective is a reliable correlate of its manipulation success, supporting CWS reward as a useful training signal and proxy metric.

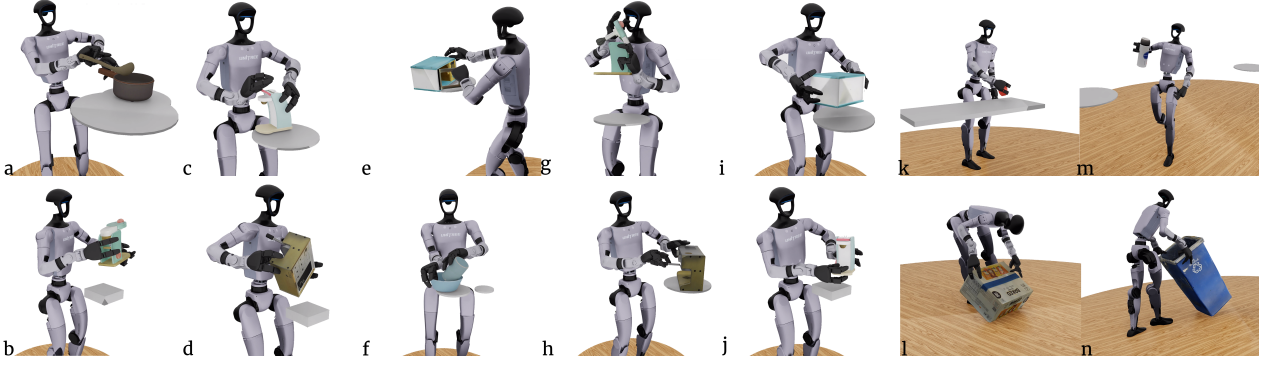


Figure 8: **Whole-body manipulation.** (a–j) Use hand-only references, where full-body motion is completed by an inpainting module, and RL training. (k–n) Use whole-body references, where the reduced force-closure objective is used during RL training.

#### 4.4. Diverse Capabilities Enable Long-Horizon Manipulation

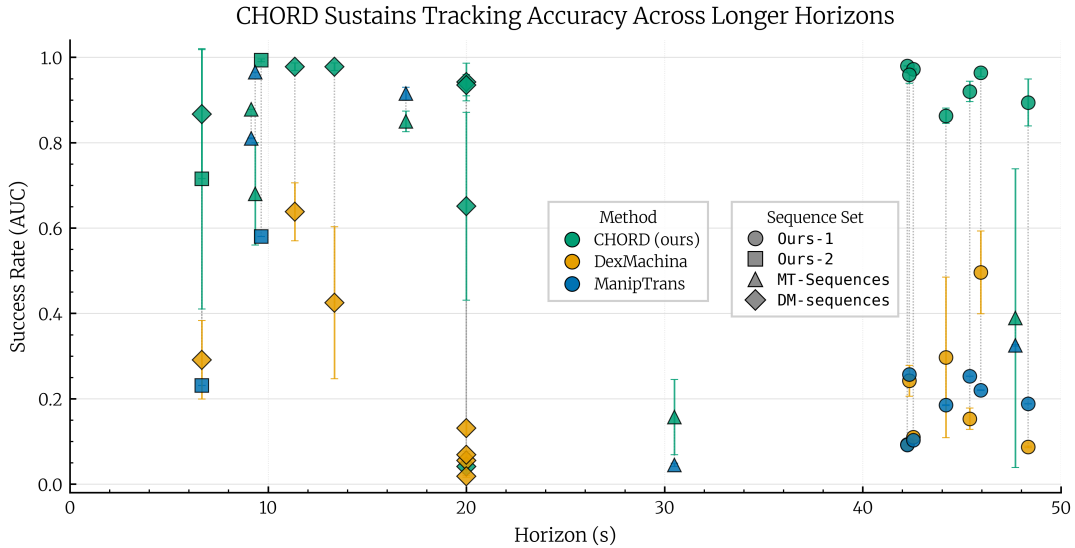


Figure 7: **CHORD sustains high tracking accuracy over long interaction horizons.** Per-sequence object-tracking performance measured by DexMachina ADD-AUC as a function of interaction horizon. Colors denote methods (CHORD, DexMachina, and ManipTrans), while marker shapes denote sequence sets (Ours-1, Ours-2, MT-Sequences, and DM-Sequences). Markers indicate the mean across up to five random seeds, with error bars showing one standard deviation. Faint dotted lines connect results obtained on the same sequence across methods.

Contact wrench serves as a unified abstraction that enables CHORD to perform a diverse range of dexterous manipulation tasks. This diversity is a key enabler of long-horizon task execution: by composing a broad repertoire of manipulation behaviors through a single mechanism, CHORD scales to task sequences spanning unprecedented temporal horizons. Figure 7 presents performance measured using the AUC metric introduced in DexMachina. The evaluated sequences cover a broad spectrum of interaction horizons, ranging from short manipulation episodes to extended task sequences lasting nearly a minute. Across this range, CHORD consistently achieves strong performance. For most sequences, it maintains near-saturated tracking accuracy (ADD-AUC  $\approx 0.85$ – $0.98$ ) even on the longest horizons (approximately 40–48 seconds), whereas baseline methods exhibit substantial degradation as the interaction horizon increases.

Category	$n$	CHORD (SR)	Position Only (SR)
Rigid	4	<b>0.994 <math>\pm</math> 0.008</b>	0.460 $\pm$ 0.487
Articulated	4	<b>0.914 <math>\pm</math> 0.107</b>	0.000 $\pm$ 0.000
Multi-object	4	<b>0.866 <math>\pm</math> 0.129</b>	0.192 $\pm$ 0.166
Overall	12	<b>0.925 <math>\pm</math> 0.104</b>	0.217 $\pm$ 0.333

Table 3: **Ablations on reward formulations.** Values are averaged across 4096 evaluation episodes and 4 seeds per sequence for each of the  $n$  sequences in each category. Best values (mean  $\pm$  std) are bolded.

#### 4.5. CHORD Extends to Whole-Body Manipulation

We evaluate the whole-body extension of CHORD described in Section 3.2 on representative tasks from both hand-only and whole-body references. The hand-only references include 12 tasks: 4 rigid-object manipulation, 4 articulated-object manipulation, and 4 multi-object manipulation tasks. The whole-body references are reconstructed from third-person-view (TPV) videos and include 5 loco-manipulation tasks (Figure 8). The results in Figure 5 (right) show that CHORD generalizes across embodiments, successfully transferring hand-only demonstrations to the G1 robot with Dex3 three-fingered hands under whole-body kinematic constraints. Wrench-space alignment enables the policy to learn effective contact behaviors despite the morphology difference between the human hand and the robot hand. For the TPV tasks [38], hand-object reconstruction noise makes the reference contact wrench unreliable, so we instead use the force-closure objective (Section 3.2), which achieves consistent success across all TPV tasks.

We ablate the role of our proposed contact wrench reward in learning whole body manipulation by comparing to an identical training recipe using contact position rewards (as used in DexMachina). As our reference contact is from a five finger human-hand and our target embodiment is a G1 humanoid with the Dex3 three fingered hand, this ablation further questions the ability of our contact wrench reward to support cross-embodiment transfer to different hand morphologies. As seen in Table 3, while contact position rewards suffer when transferring to a different embodiment, wrench space alignment guides learning towards contact behaviors that reproduce the induced object motion. We reason this is due to contact position rewards being tightly coupled with the demonstrated embodiment, whereas wrench space alignment guides learning independently of embodiment constraints.

#### 4.6. Evaluation on Real World

To evaluate real-world transfer, we deploy the learned policies on a Dexmate robot with two Sharpa dexterous hands. Robot and object poses are tracked using a motion-capture system. We test both open-loop action-chunk execution [2, 3] and closed-loop inference. As shown in Figure 9, the policies successfully manipulate both rigid and articulated objects with bimanual coordination, requiring accurate contact control. Details of the real-world setup are provided in the Appendix.

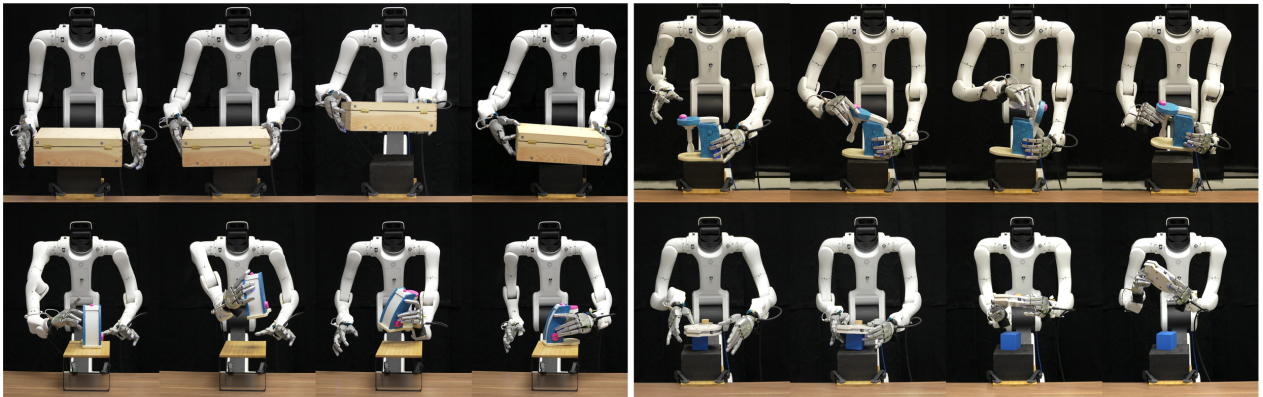


Figure 9: **Real-World.** The top-left shows closed-loop, and others show open-loop deployment.

#### 4.7. Comparing CHORD to Teleoperation

We conducted a small qualitative teleoperation pilot study to better understand the value of using RL for dexterous manipulation and how difficult it is for a human operator to reproduce our demonstration trajectories. In each trial, the operator first watched a demonstration video of the target task and then attempted to reproduce the same behavior using the teleoperation interface from [9]. We evaluated several contact-rich manipulation tasks, including box lifting, mixer use, and waffle-iron grasping.

The study revealed several consistent limitations of the teleoperation setup for contact-rich dexterous manipulation. Somewhat counterintuitively, the box-lifting task was especially difficult: successful execution required precise coordination among grasp formation, contact timing, and force application. Across tasks, we observed three primary sources of difficulty. First, the inverse kinematics module did not always preserve the operator’s intended finger configuration, particularly for wide hand apertures or poses near the robot hand’s kinematic limits. Second, because the operator received no haptic or contact feedback, the contact state had to be inferred solely from visual observation, which was often insufficient due to occlusions. Third, the teleoperation interface did not provide direct control over joint-level torques, limiting the operator’s ability to deliberately apply force through a specific finger or joint.

For the mixer and waffle-iron tasks, operators could often discover a strategy that produced an approximately correct object trajectory after extended practice. However, these executions still differed from the demonstrations in their underlying contact interactions. In particular, the resulting grasps were fragile and sensitive to small errors in contact location, timing, or applied force, often leading to unstable manipulation or task failure.

### 5. Conclusion and Discussion

We present CHORD, a framework for learning dexterous, contact-rich manipulation from human demonstrations using contact wrench-space guidance. We introduce a dexterous manipulation benchmark with 4,739 tasks spanning long-horizon, bimanual, rigid-object, and articulated-object manipulation. Evaluated on 1,831 tasks, CHORD achieves state-of-the-art performance with a 82.12% success rate. CHORD also adapts to whole-body manipulation with either hand-only or whole-body human demonstrations, achieving a 90.77% success rate. Policies trained with CHORD transfer to the real world in both open-loop and closed-loop settings.

### 6. Limitations

Despite the scalability and robustness of CHORD, several limitations remain. First, real-world deployment currently relies on state-based observations. Second, effective contact guidance requires relatively clean human demonstrations, as noisy contacts can degrade reward quality and will require a force-closure assumption. Third, object pose error is an imperfect evaluation metric: exact object placement may not be essential in some tasks, while a small pose error can lead to functional failure in others. Future work should address vision-based deployment, robustness to noisier demonstrations, and more task-aware success metrics.

### 7. Acknowledgments

We thank Chao-Yeh Chen, Lin Duan, Qingsi Wang, Xiufeng Xie, Yu-Hsiang Huang, Tiffany Chen, and David Chu for their help in standing up Dyn-HaMR. We are grateful to Ziqiang Huang, Jingrui Yu, and Chirag Majithia for their contributions to object reconstruction. We thank Daniel Zou, Jazmin Sanchez, and Hesam Rabeti for their support with third-person-view data collection, processing, and reconstruction, and Xianghui Xie and Stan Birchfield for their guidance on third-person-view reconstruction. We are also grateful to Edy Susanto Lim, Ehsan Hassani, Sam Wu, Jun Saito, Michael De Ruyter, Jiefeng Li, Ye Yuan, Umar Iqbal, and Simon Yuen for their advice on the human body model. We thank Jie Xu and Yashraj Narang for discussions on contact and simulation. Finally, we thank Spencer Huang and Rev Lebedian for their leadership and guidance.

---

## References

- [1] Mandi Zhao, Yifan Hou, Dieter Fox, Yashraj Narang, Ajay Mandlekar, and Shuran Song. Dexmachina: Functional retargeting for bimanual dexterous manipulation. In *Proceedings of the Forty-Third International Conference on Machine Learning*, 2026.
- [2] Kailin Li, Puhao Li, Tengyu Liu, Yuyang Li, and Siyuan Huang. Maniptrans: Efficient dexterous bimanual manipulation transfer via residual learning. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025.
- [3] Chaoyi Pan, Changhao Wang, Haozhi Qi, Zixi Liu, Homanga Bharadhwaj, Akash Sharma, Tingfan Wu, Guanya Shi, Jitendra Malik, and Francois Robert Hogan. SPIDER: Scalable physics-informed dexterous retargeting. *arXiv preprint arXiv:2511.09484*, 2025.
- [4] Xueyi Liu, Kangbo Lyu, Jieqiong Zhang, Tao Du, and Li Yi. Parameterized quasi-physical simulators for dexterous manipulations transfer. In Aleš Leonardis, Elisa Ricci, Stefan Roth, Olga Russakovsky, Torsten Sattler, and Gül Varol, editors, *Computer Vision – ECCV 2024*, pages 164–182, Cham, 2025. Springer Nature Switzerland.
- [5] Xueyi Liu, Jianibieke Adalibieke, Qianwei Han, Yuzhe Qin, and Li Yi. Dextrack: Towards generalizable neural tracking control for dexterous manipulation from human references. In *The Thirteenth International Conference on Learning Representations*, 2025.
- [6] Shuqi Zhao, Xinghao Zhu, Yuxin Chen, Chenran Li, Yichen Xie, Xiang Zhang, Mingyu Ding, and Masayoshi Tomizuka. Dexh2r: Task-oriented dexterous manipulation from human to robots. *IEEE/ASME Transactions on Mechatronics*, pages 1–12, 2025.
- [7] Simar Kareer, Dhruv Patel, Ryan Punamiya, Pranay Mathur, Shuo Cheng, Chen Wang, Judy Hoffman, and Danfei Xu. Egomimic: Scaling imitation learning via egocentric video. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 13226–13233, 2025.
- [8] Ryan Punamiya, Dhruv Patel, Patcharapong Aphiwetsa, Pranav Kuppili, Lawrence Y. Zhu, Simar Kareer, Judy Hoffman, and Danfei Xu. Egobridge: Domain adaptation for generalizable imitation from egocentric human data. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2026.
- [9] Ruijie Zheng, Dantong Niu, Yuqi Xie, Jing Wang, Mengda Xu, Yunfan Jiang, Fernando Castañeda, Fengyuan Hu, You Liang Tan, Letian Fu, Trevor Darrell, Furong Huang, Yuke Zhu, Danfei Xu, and Linxi Fan. Egoscale: Scaling dexterous manipulation with diverse egocentric human data, 2026.
- [10] Bowen Wen, Wenzhao Lian, Kostas Bekris, and Stefan Schaal. You only demonstrate once: Category-level manipulation from single visual demonstration. *RSS*, 2022.
- [11] Cheng-Chun Hsu, Bowen Wen, Jie Xu, Yashraj Narang, Xiaolong Wang, Yuke Zhu, Joydeep Biswas, and Stan Birchfield. Spot: Se (3) pose trajectory diffusion for object-centric manipulation. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4853–4860. IEEE, 2025.
- [12] Richard M Murray, Zexiang Li, and S Shankar Sastry. *A mathematical introduction to robotic manipulation*. CRC press, 2017.
- [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 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.
- [14] Yun Liu, Haolin Yang, Xu Si, Ling Liu, Zipeng Li, Yuxiang Zhang, Yebin Liu, and Li Yi. Taco: Benchmarking generalizable bimanual tool-action-object understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21740–21751, 2024.
- [15] Prithviraj Banerjee, Sindi Shkodrani, Pierre Moulon, Shreyas Hampali, Shangchen Han, Fan Zhang, Linguang Zhang, Jade Fountain, Edward Miller, Selen Basol, Richard Newcombe, Robert Wang, Jakob Julian Engel, and Tomas Hodan. Hot3d: Hand and object tracking in 3d from egocentric multi-view videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7061–7071, June 2025.

- 
- [16] Xinyu Zhan, Lixin Yang, Yifei Zhao, Kangrui Mao, Hanlin Xu, Zenan Lin, Kailin Li, and Cewu Lu. Oakink2: A dataset of bimanual hands-object manipulation in complex task completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 445–456, June 2024.
- [17] Harish Ravichandar, Athanasios S Polydoros, Sonia Chernova, and Aude Billard. Recent advances in robot learning from demonstration. *Annual review of control, robotics, and autonomous systems*, 3(1):297–330, 2020.
- [18] Yanjie Ze, Zixuan Chen, João Pedro Araújo, Zi ang Cao, Xue Bin Peng, Jiajun Wu, and C. Karen Liu. Twist: Teleoperated whole-body imitation system. *arXiv preprint arXiv:2505.02833*, 2025.
- [19] Qiayuan Liao, Takara E. Truong, Xiaoyu Huang, Yuman Gao, Guy Tevet, Koushil Sreenath, and C. Karen Liu. Beyondmimic: From motion tracking to versatile humanoid control via guided diffusion, 2025.
- [20] Zhengyi Luo, Ye Yuan, Tingwu Wang, Chenran Li, Sirui Chen, Fernando Castañeda, Zi-Ang Cao, Jiefeng Li, David Minor, Qingwei Ben, Xingye Da, Runyu Ding, Cyrus Hogg, Lina Song, Edy Lim, Eugene Jeong, Tairan He, Haoru Xue, Wenli Xiao, Zi Wang, Simon Yuen, Jan Kautz, Yan Chang, Umar Iqbal, Linxi Fan, and Yuke Zhu. Sonic: Supersizing motion tracking for natural humanoid whole-body control. *arXiv preprint arXiv:2511.07820*, 2025.
- [21] Suraj Nair, Aravind Rajeswaran, Vikash Kumar, Chelsea Finn, and Abhinav Gupta. R3m: A universal visual representation for robot manipulation. *arXiv preprint arXiv:2203.12601*, 2022.
- [22] Allison M Okamura, Niels Smaby, and Mark 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. IEEE, 2000.
- [23] Rika Antonova, Jingyun Yang, Krishna Murthy Jatavallabhula, and Jeannette Bohg. Rethinking optimization with differentiable simulation from a global perspective. In *6th Annual Conference on Robot Learning*, 2022.
- [24] Tao Pang, H. J. Terry Suh, Lujie Yang, and Russ Tedrake. Global planning for contact-rich manipulation via local smoothing of quasi-dynamic contact models. *IEEE Transactions on Robotics*, 39(6):4691–4711, 2023.
- [25] Xinghao Zhu, Jinghan Ke, Zhixuan Xu, Zhixin Sun, Bizhe Bai, Jun Lv, Qingtao Liu, Yuwei Zeng, Qi Ye, Cewu Lu, Masayoshi Tomizuka, and Lin Shao. Diff-ldf: Contact-aware model-based learning from visual demonstration for robotic manipulation via differentiable physics-based simulation and rendering. In *7th Annual Conference on Robot Learning*, 2023.
- [26] H. J. Terry Suh, Tao Pang, Tong Zhao, and Russ Tedrake. Dexterous contact-rich manipulation via the contact trust region. *The International Journal of Robotics Research*, 0(0), 2026.
- [27] Hyung Ju Suh, Tao Pang, and Russ Tedrake. Bundled gradients through contact via randomized smoothing. *IEEE Robotics and Automation Letters*, 7:1–1, 04 2022.
- [28] 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, 2021.
- [29] Yuanpei Chen, Chen Wang, Yaodong Yang, and C Karen Liu. Object-centric dexterous manipulation from human motion data. *arXiv preprint arXiv:2411.04005*, 2024.
- [30] Yunhai Han, Zhenyang Chen, Kyle A Williams, and Harish Ravichandar. Learning prehensile dexterity by imitating and emulating state-only observations. *IEEE Robotics and Automation Letters*, 9(10):8266–8273, 2024.
- [31] Antonio Bicchi. On the closure properties of robotic grasping. *The International Journal of Robotics Research*, 14(4):319–334, 1995.
- [32] Carlo Ferrari and John Canny. Planning optimal grasps. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 2290–2295, 1992.
-

- 
- [33] Alexander Koenig, Zixi Liu, Lucas Janson, and Robert Howe. The role of tactile sensing in learning and deploying grasp refinement algorithms. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7766–7772. IEEE, 2022.
- [34] Haris Merzic, Miroslav Bogdanovic, Daniel Kappler, Ludovic Righetti, and Jeannette Bohg. Leveraging contact forces for learning to grasp. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 3615–3621. IEEE, 2019.
- [35] Andrew Melnik, Luka Lach, Matthias Plappert, Timo Korthals, Robert prestige Haschke, and Helge Ritter. Tactile sensing and deep reinforcement learning for in-hand manipulation tasks. In *IROS Workshop on Autonomous Object Manipulation*, 2019.
- [36] Andrew Melnik, Luka Lach, Matthias Plappert, Timo Korthals, Robert Haschke, and Helge Ritter. Using tactile sensing to improve the sample efficiency and performance of deep deterministic policy gradients for simulated in-hand manipulation tasks. *Frontiers in Robotics and AI*, 8:57, 2021.
- [37] Tingwu Wang, Olivier Dionne, Michael De Ruyter, David Minor, Davis Rempe, Kaifeng Zhao, Mathis Petrovich, Ye Yuan, Chenran Li, Zhengyi Luo, Brian Robison, Xavier Blackwell, Bernardo Antoniazzi, Xue Bin Peng, Yuke Zhu, and Simon Yuen. Motionbricks: Scalable real-time motions with modular latent generative model and smart primitives, 2026.
- [38] Xianghui Xie, Bowen Wen, Yan Chang, Hesam Rabeti, Jiefeng Li, Ye Yuan, Gerard Pons-Moll, and Stan Birchfield. Cari4d: Category agnostic 4d reconstruction of human-object interaction. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2026.
- [39] 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. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.
- [40] Omid Taheri, Nima Ghorbani, Michael J. Black, and Dimitrios Tzionas. GRAB: A dataset of whole-body human grasping of objects. In *European Conference on Computer Vision (ECCV)*, 2020.
- [41] Taein Kwon, Bugra Tekin, Jan Stühmer, Federica Bogo, and Marc Pollefeys. H2o: Two hands manipulating objects for first person interaction recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10138–10148, October 2021.
- [42] Mayank Mittal, Pascal Roth, James Tigue, Antoine Richard, Octi Zhang, Peter Du, Antonio Serrano-Muñoz, Xinjie Yao, René Zurbrügg, Nikita Rudin, Lukasz Wawrzyniak, Milad Rakhsha, Alain Denzler, Eric Heiden, Ales Borovicka, Ossama Ahmed, Iretoiayo Akinola, Abrar Anwar, Mark T. Carlson, Ji Yuan Feng, Animesh Garg, Renato Gasoto, Lionel Gulich, Yijie Guo, M. Gussert, Alex Hansen, Mihir Kulkarni, Chenran Li, Wei Liu, Viktor Makoviychuk, Grzegorz Malczyk, Hammad Mazhar, Masoud Moghani, Adithyavairavan Murali, Michael Noseworthy, Alexander Poddubny, Nathan Ratliff, Welf Rehberg, Clemens Schwarke, Ritvik Singh, James Latham Smith, Bingjie Tang, Ruchik Thaker, Matthew Trepte, Karl Van Wyk, Fangzhou Yu, Alex Millane, Vikram Ramasamy, Remo Steiner, Sangeeta Subramanian, Clemens Volk, CY Chen, Neel Jawale, Ashwin Varghese Kuruttukulam, Michael A. Lin, Ajay Mandlekar, Karsten Patzwaldt, John Welsh, Huihua Zhao, Fatima Anes, Jean-Francois Lafleche, Nicolas Moënne-Loccoz, Soowan Park, Rob Stepinski, Dirk Van Gelder, Chris Amevor, Jan Carius, Jumyung Chang, Anka He Chen, Pablo de Heras Ciechomski, Gilles Daviet, Mohammad Mohajerani, Julia von Muralt, Viktor Reutsky, Michael Sauter, Simon Schirm, Eric L. Shi, Pierre Terdiman, Kenny Vilella, Tobias Widmer, Gordon Yeoman, Tiffany Chen, Sergey Grizan, Cathy Li, Lotus Li, Connor Smith, Rafael Wiltz, Kostas Alexis, Yan Chang, David Chu, Linxi "Jim" Fan, Farbod Farshidian, Ankur Handa, Spencer Huang, Marco Hutter, Yashraj Narang, Soha Pouya, Shiwei Sheng, Yuke Zhu, Miles Macklin, Adam Moravanszky, Philipp Reist, Yunrong Guo, David Hoeller, and Gavriel State. Isaac lab: A gpu-accelerated simulation framework for multi-modal robot learning. *arXiv preprint arXiv:2511.04831*, 2025.
- [43] Javier Romero, Dimitrios Tzionas, and Michael J. Black. Embodied hands: Modeling and capturing hands and bodies together. *ACM Transactions on Graphics (TOG)*, 36(6), 2017.
-

- [44] Ruicheng Wang, Sicheng Xu, Cassie Dai, Jianfeng Xiang, Yu Deng, Xin Tong, and Jiaolong Yang. MoGe: Unlocking accurate monocular geometry estimation for open-domain images with optimal training supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.
- [45] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, and Lei Zhang. Grounding DINO: Marrying DINO with grounded pre-training for open-set object detection. In *European Conference on Computer Vision (ECCV)*, 2024.
- [46] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollár, and Christoph Feichtenhofer. SAM 2: Segment anything in images and videos. In *International Conference on Learning Representations (ICLR)*, 2025.
- [47] SAM 3D Team. SAM 3d: 3dfy anything in images. *arXiv preprint arXiv:2511.16624*, 2025.
- [48] Bowen Wen, Wei Yang, Jan Kautz, and Stan Birchfield. FoundationPose: Unified 6d pose estimation and tracking of novel objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.
- [49] Jiahui Huang, Qunjie Zhou, Hesam Rabeti, Aleksandr Korovko, Huan Ling, Xuanchi Ren, Tianchang Shen, Jun Gao, Dmitry Slepichev, Chen-Hsuan Lin, Jiawei Ren, Kevin Xie, Joydeep Biswas, Laura Leal-Taixé, and Sanja Fidler. ViPE: Video pose engine for 3d geometric perception. *arXiv preprint arXiv:2508.10934*, 2025.
- [50] Zhengdi Yu, Stefanos Zafeiriou, and Tolga Birdal. DynHAMR: Recovering 4d interacting hand motion from a dynamic camera. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 27716–27726, 2025.
- [51] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d Gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics (TOG)*, 42(4), 2023.
- [52] Zachary Teed and Jia Deng. DROID-SLAM: Deep visual SLAM for monocular, stereo, and RGB-D cameras. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

# Appendix

[https://nvidia-isaac.github.io/video\\_to\\_data/chord/](https://nvidia-isaac.github.io/video_to_data/chord/)

This chapter supplements our manuscript, *Learning Dexterous Manipulation Using Contact Wrench Guidance From Human Demonstration*. It provides additional details on:

- Ablation studies on noise in demonstrations (Section A.1)
- Reinforcement learning implementation details (Section B)
- Reconstruction and learning from human video (Section C)
- Benchmark details and statistics (Section D)
- Whole-body motion planning and control (Section E)
- Hardware experiment setup (Section F)
- Baseline implementation details (Section G)

## A. Ablation Studies

### A.1. Robustness on Human Demonstration Noise

To evaluate the robustness of CHORD, we conduct an ablation study in which increasing levels of noise are injected into the demonstration data. The goal is to understand how sensitive CHORD is to imperfections commonly encountered in reconstructed demonstrations. We sample random walk noise as follows,

Scale	wrist pos. (mm)	wrist ori. (deg)	contact pos. (mm)	contact normal (deg)
1	10	5	5	5
2	20	10	10	10
3	30	15	15	15
4	40	20	20	20
5	50	25	25	25
6	60	30	30	30

Table 4: Noise bounds per scale.

We consider three settings:

- **Noisy Hand Pose + Contact Wrench Reward (Hand Noise Only)**: noise is injected into the hand pose, while contact guidance remains noise-free. Contact supervision is provided using the contact-wrench-space reward. This setting simulates human demonstrations captured through video-based motion reconstruction, where contact information may still be obtained from tactile gloves.
- **Noisy Hand Pose + Noisy Contact + Contact Wrench Reward (Hand+Contact Noise)**: noise is injected into both the hand pose and contact guidance, including contact positions and surface normals, thereby perturbing the wrench supports. Contact supervision is provided using the contact-wrench reward. This setting simulates human demonstrations captured purely from video-based reconstruction, without additional sensors.
- **Noisy Hand Pose + Noisy Contact + Force-Closure Reward (Hand+Contact Noise w/ FC)**: noise is injected into both the hand pose and contact guidance, including contact positions and surface normals. However, training is guided by a force-closure reward rather than the contact-wrench reward. This setting evaluates whether wrench-space supervision is more robust to noisy demonstrations than force-closure-based objectives.

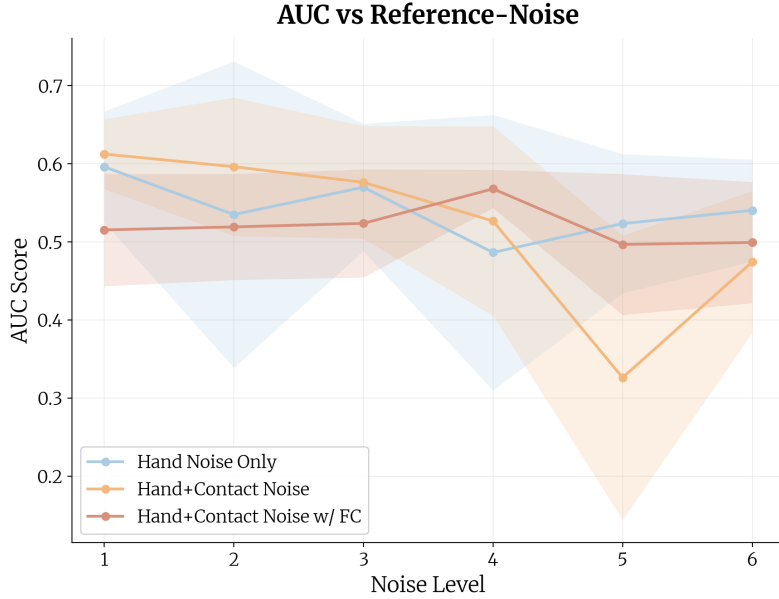


Figure 10: Learning performance vs demonstration noise level.

Figure 10 compares task performance as the magnitude of injected noise increases. CHORD remains robust to moderate hand-pose noise, with gradual performance degradation, suggesting that the method can tolerate realistic levels of state estimation and tracking error.

When noise is additionally introduced into contact guidance, performance decreases more noticeably, reflecting the increased difficulty of recovering accurate contact interactions from corrupted demonstration data. At very high noise levels, force-closure guidance can outperform noisy wrench-space guidance, since heavily corrupted wrench supports may provide misleading supervision. However, we also observe that relying on force-closure guidance reduces dexterity, as it prioritizes stable grasps over faithful imitation of human manipulation behavior.

## B. Implementation Details of Reinforcement Learning

**Observation Space.** The policy observation is formed by concatenating proprioceptive state, object state, task command, action history, and contact information. Specifically, the observation includes the following terms:

- **Wrist state:** right and left wrist positions, orientations, and body-frame velocities.
- **Finger state:** right and left finger joint positions and velocities, with joint positions scaled by joint limits.
- **Object state:** flattened object/body positions and orientations for all objects in the scene.
- **Command:** target deltas for wrist poses, finger joints, and object poses.
- **Action history:** the previous raw policy action, together with processed right- and left-hand actions.
- **Contact observation:** contact positions and force directions transformed into the wrist frame, with inactive contacts zeroed out.

Although noise configurations are defined for several observation terms, observation corruption is disabled in our experiments during large-scale experiments, but enabled in real-world closed-loop policy training.

**Action Space.** The policy controls residual actions for both hands. For each hand, the action consists of a 3D wrist position residual, a 3D wrist orientation residual, and finger joint residuals. The action scales are set to  $0.05m$  for wrist position,  $0.15rad$  for wrist orientation, and  $0.15rad$  for finger joints. We apply an exponential moving average with factor 0.3 to smooth the actions. The resulting targets are executed through a wrist force/torque tracking controller and finger joint position targets.

For object control, the scene setup additionally creates one virtual object-control term for each rigid or articulated object. These terms have zero policy action dimension and therefore are not directly controlled by the policy. Instead, they apply curriculum-scaled PD wrenches or efforts internally.

**Termination and Reset.** Episodes terminate when any of the following conditions is satisfied:

- **Time-out:** the timestep counter reaches the retargeted motion horizon.
- **Wrist deviation:** either wrist deviates too far from the commanded wrist trajectory.
- **Object deviation:** any object exceeds the allowed position or orientation error threshold.

Each episode starts from a random frame along the retargeted motion. At reset, the environment restores the object root pose and velocity, resets articulated object joints when present, restores the right and left wrist root poses and velocities, and initializes the right and left finger joints to a partially open version of the reference pose using a random interpolation factor in  $[0, 0.7]$ . The per-environment virtual object-control scale is reset to 1.0, followed by a 20-step warm-up period before applying the curriculum target. Finally, the action terms clear raw actions, previous actions, wrist forces and torques, and finger targets.

**Relative Task Reward.** For two-object manipulation sequences, tracking each object independently is not always sufficient to capture the task-relevant geometry between objects. Many interactions, such as inserting, pouring, scooping, or tool use, depend primarily on the relative pose between the manipulated objects rather than their absolute poses in the world. We therefore introduce a relative object pose reward,  $r_{\text{relative}}$ , that encourages the policy to reproduce the demonstrated spatial relationship between objects, defined as an exponential tracking kernel over both translational and rotational error between the two objects:

$$r_{\text{relative}}(t) = m(t) \exp\left(-\frac{e_p(t) \cdot e_R(t)}{\text{var}_{\text{rel}}}\right)$$

where  $\text{var}_{\text{rel}}$  controls the tolerance of the translational and rotational components, respectively. The relative translation and rotation errors are defined as  $e_p(t) = \|p_{1|0}(t) - \bar{p}_{1|0}(t)\|_2$ ,  $e_R(t) = d_{\text{geo}}(R_{1|0}(t), \bar{R}_{1|0}(t))$ , where  $d_{\text{geo}}$  is the geodesic rotation distance. The binary mask  $m(t)$  activates the reward only during demonstrated object-object interaction phases. Importantly, this gating depends solely on the demonstration inter-object distance, rather than the policy’s current object distance. This design prevents the policy from trivially avoiding the reward by keeping objects far apart, while still avoiding unnecessary relative-pose constraints during approach phases in which the demonstrated objects are not yet interacting. For single-object sequences, the reward is zero. For multi-object sequences, the demonstrated relative poses and inter-object distances are precomputed once from the reference trajectory, and the reward is evaluated at every simulation step using the current object root poses. The multiplicative formulation ensures that high reward is achieved only when both the relative translation and relative orientation match the demonstration, making the reward directly aligned with preserving task-relevant object-object geometry.

## C. CHORD Validation from Human Video

To validate CHORD end-to-end from raw egocentric human video, we select a brush manipulation task. We reconstruct the hand-object interaction with a pipeline that recovers, from a single monocular RGB video and a text prompt naming the object, a per-frame MANO [43] hand pose and a textured object mesh with its 6-DoF pose, expressed in a shared, roughly-metric space. This trajectory is the input we pass to CHORD for retargeting and grounding into simulation.

Monocular depth from MoGe [44] is the metric anchor for the pipeline: object scale, hand scale and translation, and the camera trajectory are all aligned to the same depth, placing the two streams in a common frame with a consistent hand-object displacement. Each stream is otherwise estimated independently, until a final Gaussian-splat refinement jointly optimizes and temporally smooths all outputs.

**Object.** We detect the object on a reference frame with an open-vocabulary detector (Grounding DINO [45]) prompted by the text label, segment it across the video (SAM2 [46]), and reconstruct a textured mesh from

the reference frame (SAM3D [47]). As the mesh is recovered only up to scale, we estimate its metric size by registering it against the MoGe depth, then track its 6-DoF pose across the video (FoundationPose [48]).

**Hand.** We recover per-frame MANO parameters with a monocular hand tracker (ViPE [49] + Dyn-HaMR [50]) and align them to metric depth: we render the hand and shift it along its viewing ray so that its rendered depth matches MoGe, recovering a per-frame translation and a global hand scale. The hand and object then share the same depth anchor.

**Refinement.** Finally, we fit a Gaussian-splat scene [51] to the video, with Gaussians anchored to the object and hand meshes (and to a background point cloud), and the camera is initialized from monocular SLAM (DROID-SLAM [52]). Optimizing photometric, segmentation, and depth losses under temporal-smoothness penalties jointly refine and smooth the object pose and scale, hand pose and scale, and camera poses that constitute the final reconstruction.

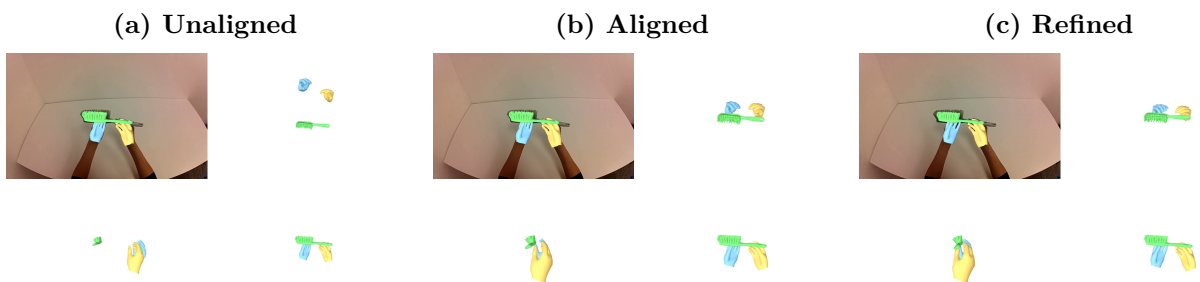


Figure 11: Hand-object reconstruction across three stages of processing. Each panel shows the source camera view (top-left) and three world-frame views. **(a)** Unaligned: raw Dyn-HaMR output before any correction. **(b)** Aligned: after depth alignment with MoGe monocular depth estimates. **(c)** Refined: after Gaussian splat-based joint optimization.

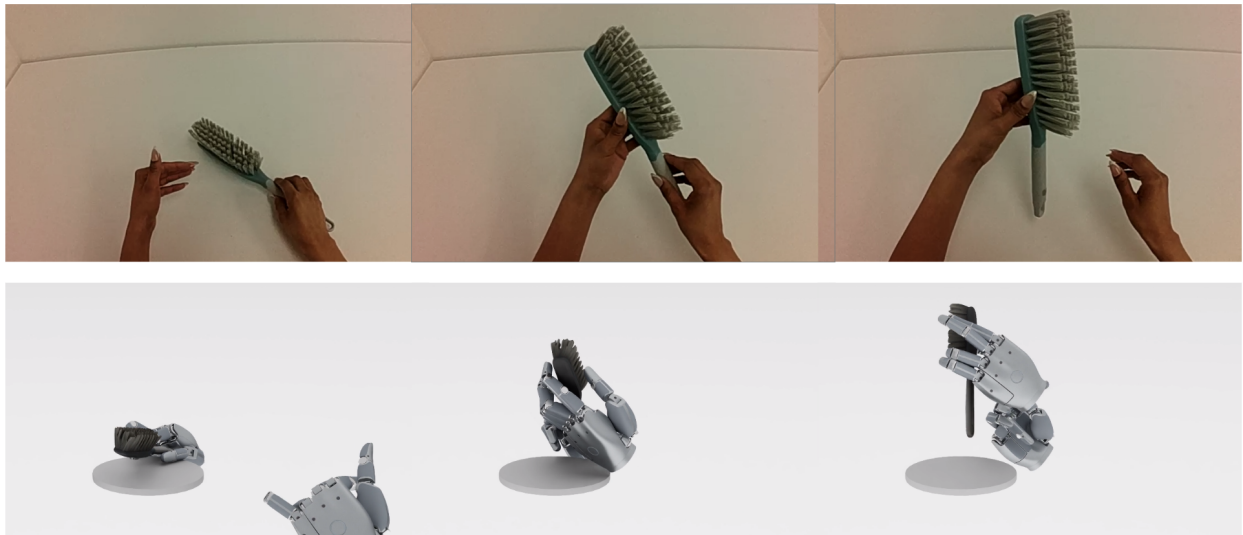


Figure 12: Hand-object reconstruction result after the full reconstruction pipeline.

Figure 12 visualizes the demonstration video and resulting grounded behavior. The top row shows representative frames from the original human demonstration, while the bottom row shows the corresponding grounded robot execution in simulation. Despite substantial embodiment differences between the human demonstrator and robot, CHORD successfully transfers the demonstrated interaction and produces physically plausible manipulation behavior.

To quantitatively evaluate grounding quality, we report 0.647 AUC demonstrating that the proposed pipeline can transform noisy human video observations into executable robot trajectories. These results validate the complete system, including human motion reconstruction, contact extraction, contact-aware optimization, and embodiment grounding, highlighting the practicality of CHORD for real-world human-to-robot behavior transfer.

## D. Dexterous Manipulation Benchmark Details

**Post-process.** Our task library converts human hand-object interaction data into Sharpa-Wave robot trajectories. It ingests per-frame MANO hand fits and rigid-object 6-DoF trajectories from seven motion-capture datasets: ARCTIC, TACO, HOT3D, OakInk2, DexYCB, GRAB, and H2O. Each dataset-specific loader parses the source’s native format and exports every sequence into a unified representation containing the wrist pose, 21 MANO joint positions, per-link hand-object contacts, and the object’s full 6-DoF pose.

For HOT3D, the long egocentric recordings are first segmented into atomic hand-object interaction clips before retargeting. Contact is detected when the minimum fingertip-to-object distance falls below 2 cm. Brief contact breaks of up to 15 frames are bridged, and each resulting contact window is split according to the dominant contacted object using a per-frame majority vote. This process expands roughly 294 raw recordings into approximately 4,045 atomic clips.

Each sequence is then retargeted to the robot using differential inverse kinematics solved with a quadratic-programming optimizer. The per-frame objective tracks the human wrist position and orientation, as well as fingertip residuals, while respecting the robot’s joint-position and joint-velocity limits. Optimization runs for up to 200 iterations per frame and terminates early when the residual change falls below  $1e-6$ .

**Quality Check.** After retargeting, every sequence undergoes two complementary quality checks to determine whether it is accepted into the training set or rejected. The penetration check models the robot hand as a set of capsules and measures both hand-object overlap and hand-hand overlap on each frame. Hand-object overlap is computed against the convex hull of the object, while hand-hand overlap uses analytic segment-to-segment distances between capsule pairs. A sequence is rejected if either penetration measure exceeds 2 cm in any frame. Highly concave objects, defined as having a convex-hull-to-mesh volume ratio above 3, are excluded from the hand-object penetration check because their convex hulls can falsely classify the hand as being inside open objects such as mugs or AR glasses.

The replay check validates each sequence in the Isaac Lab simulator by replaying the trajectory with tracking actions for 300 steps. This verifies that the scene initializes correctly and that the simulator can execute the trajectory without failure. A sequence is marked successful only after the rendered video is fully written to disk. As a result, any simulator issue, including CUDA asserts, NaN observations, physics instabilities, joint-limit violations, Omniverse shutdown deadlocks, or video-encoding failures, prevents the success marker from being recorded.

**Dataset Statistics.** To characterize the contact behavior in our task library with human motions, we report the following metrics, computed from the precomputed contact wrench matrix  $\mathcal{W}_{h,k}$  and its support values  $h_{h,k}$  along 512 basis directions.

- **Contact events:** number of contact intervals per hand. Together with the *segment duration* distribution, it distinguishes long contact events from frequent short touches.
- **Ferrari-Canny epsilon**  $\varepsilon = \min h_{h,k}$ : distance from the origin to the nearest face of the convex wrench hull in wrench matrix;  $\varepsilon > 0$  certifies force closure and measures disturbance resistance.

## E. Extending CHORD to Whole Body Manipulation

We propose a generic pipeline for extending CHORD to whole-body manipulation, dependent on the human demonstration data available. Our pipeline leverages a complete or synthetically generated whole body reference trajectory and learns a residual following the CHORD recipe to perform whole body manipulation.

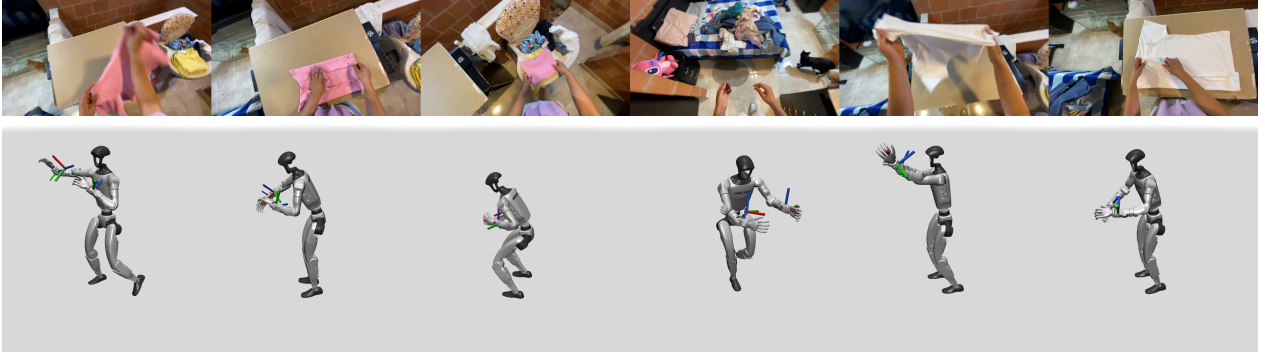


Figure 13: Planned whole body motion for end effector trajectories extracted from egocentric video.

We discuss our approach for generating a whole body reference trajectory given a hand-only reference (ex. obtained from an egocentric video) in Sec. E.1. We then discuss how we apply CHORD using a whole body reference trajectory in Sec. E.2.

### E.1. Planning Whole Body Motion from End Effector Trajectories

To leverage hand-only references for learning whole-body manipulation, we first retarget the reconstructed MANO hand to the target robotic dexterous hand. Given the retargeted end effector trajectories, we seek a way to generate a corresponding whole-body motion that can be tracked by a whole-body controller [20]. To this end, we train an end-effector to a whole-body inpainting module, utilizing the architecture proposed in [37], largely reusing their proposed notation for clarity.

**Constraint Specification.** The keyframe constraints  $\mathcal{T} = \{\mathcal{T}_1, \mathcal{T}_2, \mathcal{T}_3\}$  provide sparse, optional anchors on the local root, global root, and pose respectively. Our specialisation fixes  $\mathcal{T}_1, \mathcal{T}_2$  to be empty (i.e. masked at every temporal slot) and restricts  $\mathcal{T}_3$  to the wrist body subset, populated at *every* frame. Let,  $\mathcal{H} \subset \mathbb{B}$  index the EE body slots ( $\mathcal{H} = \{L, R\}$  for the left and right wrist), and let

$$\mathcal{T}^{\text{EE}} = \{ (p_h^{*,t}, q_h^{*,t}) \mid t \in [1, T], h \in \mathcal{H} \} \quad (4)$$

be the dense EE constraint set, where  $p_h^{*,t} \in \mathbb{R}^3$  is the target wrist position in the global root-relative frame and  $q_h^{*,t}$  is its rotation in 6D.

**Training Data and Constraint Sampling.** We train the tokenizer, root module, and pose module on the retargeted BONES Seed dataset. Motion segments are sampled following [37]. For each segment we extract the ground-truth EE conditioning,

$$\mathcal{T}_{\text{gt}}^{\text{EE}} = \{ (p_h^t, q_h^t) \mid t \in [1, T], h \in \mathcal{H} \} \quad (5)$$

and with probability  $\rho_{\text{drop}}$  each  $(h, t)$  entry is replaced by a learnable mask embedding.

**Tokenizer Training.** The tokenizer is trained as described in [37] with the addition of an auxiliary end effector loss.

**Root Module Training.** Given the keyframe constraints  $\mathcal{T}_{\text{gt}}^{\text{EE}}$ , the root module is trained to predict:

1. ( $\mathcal{F}_1$ ): predict the in-betweening frame count  $T_2$  as described in [37], optimized with cross entropy loss on binned frame counts.
2. ( $\mathcal{F}_2$ ): predict the global root trajectory conditioned on  $T_2$ , as described in [37], optimized with smooth- $\ell_1$  loss on the ground truth root trajectory.

**Pose Module Training.** Given the keyframe constraints  $\mathcal{T}_{\text{gt}}^{\text{EE}}$ , we transform them to be root relative given the ground truth root trajectory and then train the pose module to predict the discrete tokenizer codebook indices corresponding to the ground truth whole body motion, optimized with cross entropy loss.

**Inference Loop.** At inference, the planner consumes (i) a per-frame EE target trajectory  $\mathcal{T}^{\text{EE}}$  of total length  $T_{\text{seq}}$  and (ii)  $T_{\text{ctx}} = 4$  context frames seeding the autoregressive rollout. It produces the full-body state sequence  $\{r_g^t, r_l^t, p^t, q^t, v^t, c^t\}_{t=1}^{T_{\text{seq}}}$ . Generation proceeds chunk by chunk in MotionBricks tokens. The autoregressive seed for chunk  $j$  is provided by the last  $T_{\text{ctx}}$  frames of chunk  $j-1$ . For chunk 0, the seed is supplied as a nominal pose that is not otherwise available. In practice, we infer the planner produces overlapping chunks and apply blending to the decoded motions.

## E.2. Applying CHORD

Given a whole body reference, we learn a residual on top of a pretrained whole body controller [20] to adjust the controller’s actions to induce the desired object trajectory by aligning the robot’s contact wrench to the human demonstrated contact wrench. Our implementation details largely follow Sec. B with notable differences as follows,

**Observation Space.** We make the following additions to the observation space,

- **Joint state:** whole body joint positions and velocities.
- **Command:**
  - **Anchor:** target deltas for the position and orientation of the anchor body from the current frame to N state reference frames.
  - **End Effectors:** target deltas for the position and orientation of the end effector bodies from the current state to N future reference frames.
  - **Joint Position:** target deltas for the joint position from the current state to N future reference frames.

We remove **contact observation** from the whole body training observation space.

**Action Space.** The policy controls residual actions for the whole body joint positions. The residual actions are applied to the output of a whole body controller tracking the original whole body reference with an action scale of 0.5 for non-finger joints and 0.15 for finger joints. These adapted actions are then used as joint position targets.

**Termination and Reset.** We add the following terminations:

- **Anchor Position Error:** the anchor deviates too far from the commanded anchor position.
- **Anchor Orientation Error:** the anchor deviates too far from the commanded anchor orientation.

To reset the trajectory at arbitrary points, we leverage the stability of the whole body controller where the environment restores the robot state and object state at a randomly sampled frame in the trajectory and holds the commanded reference at this frame for N frames. During this period, selected joints are "opened" using an offset factor applied to the nominal joint positions in the reference frame and the per-environment virtual object control is held at scale 1.0 for  $M < N$  frames before decaying to the training curriculum scale. In our experiments, we use an offset factor of 0.5 on the shoulder yaw joints of the G1 to pull the arms back from the manipulated object(s) and allow the policy to learn to recover towards the object(s).

**Additional Rewards** We add the following rewards,

- **Anchor Position Reward:** rewards tracking the reference anchor position with a Gaussian error kernel.
- **Anchor Orientation Reward:** rewards tracking the reference anchor orientation with a Gaussian error kernel.

## F. Hardware Setup

**Hardware Platform.** Our real-world experiments are conducted on a Dexmate robot equipped with two Sharpa dexterous hands. The deployment controller actuates the left and right Dexmate arms, the torso, and both Sharpa hands. Each arm has seven actuated joints, the torso has three actuated joints, and each Sharpa hand provides 22 actuated joints, giving a 61-dimensional commanded joint vector during deployment. The robot head and mobile base are not used by the policy controller in these experiments. The Dexmate torso is initialized and held at a fixed crouched configuration,  $q_{\text{torso}} = [0.78, 1.57, 0.44]rad$ . The low-level bridge publishes robot state at 100 Hz and sends joint commands with a zero-order hold at up to 500 Hz. Object and robot poses are measured with a six-camera Vicon motion-capture system. Adhesive reflective markers are attached to the robot and manipulated objects to define trackable rigid bodies.

### Vicon Setup and Calibration Pipeline

The Vicon system streams rigid-body poses to the deployment process through a ZMQ publisher with 100 Hz. Each Vicon frame contains a timestamped set of subjects, where each subject pose is represented by a 3D position and a unit quaternion in  $wxyz$  convention. The deployment process subscribes to this stream, keeps the latest frame, and rejects stale frames older than 0.25s. We use the Vicon-tracked robot segment to align the Vicon world frame with the robot kinematic model. The tracked robot segment is associated with the `torso_13` frame in the Dexmate URDF. Given the current robot joint state, forward kinematics computes the transform from the robot base to this tracked frame. Combining this kinematic transform with the Vicon measurement yields the robot base pose in the Vicon world frame. Before policy execution, we calibrate the frame alignment using the object pose. We denote the Vicon world frame by  $V$ , the robot base frame by  $B$ , the policy environment frame by  $E$ , and the object frame by  $O$ . The policy predicts wrist targets in  $E$ , while the real robot controller and inverse kinematics operate in  $B$ . Therefore, we first solve the environment-to-base transform  $T_E^B$ . Let  $T_O^B$  be the live object pose expressed in the robot base frame, and let  $T_O^E$  be the corresponding object pose in the recorded motion-reference trajectory used by the policy. Since both poses describe the same physical object configuration at initialization, they satisfy  $T_O^B = T_E^B T_O^E$ , giving

$$T_E^B = T_O^B (T_O^E)^{-1}, \quad (6)$$

After this one-time startup calibration,  $T_E^B$  is held fixed and used to map policy wrist targets from the policy frame into the robot base frame before inverse kinematics.

The policy observation, however, also needs live wrist and object poses expressed in  $E$ , while Vicon measures those poses in  $V$ . To avoid routing every live Vicon pose through the robot base frame, we compute a second fixed transform  $T_V^E$  at the same calibration instant. Let  $T_O^V$  be the live object pose measured directly by Vicon. Because  $T_O^E$  and  $T_O^V$  correspond to the same initialized object pose, we set

$$T_V^E = T_O^E (T_O^V)^{-1}. \quad (7)$$

At runtime, any live Vicon pose  $T_X^V$  is converted to the policy frame as  $T_X^E = T_V^E T_X^V$ , where  $X$  may be the object or either wrist frame.

### Policy Deployment Workflow

The deployed policy is exported as an ONNX model and executed with ONNX Runtime inside a real-time inference loop. The loop runs at the trajectory rate, 20 Hz in our experiments, while the lower-level hardware bridges hold the most recent command at a higher frequency. The inference loop uses a finite-state machine with HOLD, PLAY, SAFETY, and DONE states. In HOLD, the robot maintains the calibrated initial pose. Pressing the operator start key resets the policy action accumulator from the current robot state and enters PLAY. If Vicon tracking becomes stale or the robot marker body violates the drift threshold, the system enters SAFETY, freezes the current joint state, and requires an operator reset before continuing.

## G. Baseline Reproduction Implementation Details

To produce the results reported in Table 1, we extended the prior methods DexMachina, ManipTrans, and Spider with Sharpa hands and leveraged their original code base. In this section, we describe our implementation efforts to faithfully reproduce these methods on the new embodiment. Notably, both

DexMachina and ManipTrans claim to be embodiment-agnostic and capable of supporting arbitrary robot embodiments.

### G.1. DexMachina

We first follow the dexterous hand asset processing instructions provided by the authors of DexMachina to add the Sharpa Hand. This includes running the provided urdf inspection script, adding 6 DOF joints to the wrists, and creating the retargeting configs mapping the robot fingers to MANO keypoints. We then specify collision groups using the provided utilities and add a corresponding configuration python file for the Sharpa Hand. We tune the controller gains following the procedure described in the documentation and also keep an alternate configuration copying the gains used in our IsaacLab CHORD training. Finally, we run retargeting on the provided sequences as well as our own and visually inspect each retargeting in simulation kinematic playback to verify the hand trajectory, object trajectory, and contact markers are all consistent with the MANO demonstration.

After asset verification and retargeting, we use the released DexMachina RL training recipe as implemented but with the Sharpa Hand asset and previously retargeted sequences. We run 4 variants of training with 5 seeds each for all sequences. We train (1) using the controller gains tuned in DexMachina with Hybrid Action space (2) using the controller gains used in CHORD IsaacLab with Hybrid Action Space (3) using the controller gains tuned in DexMachina with Residual Action Space (4) using the controller gains in CHORD IsaacLab with Residual Action Space. We report the best performing variant (1).

### G.2. Maniptrans

We reproduced and verified ManipTrans in two phases. First, we ran the original ManipTrans pipeline end-to-end on the Inspire hand using the authors' released assets. ManipTrans consists of MANO-to-dexhand IK retargeting, per-side imitator policies, per-primitive residual policies, and rollout evaluation. On an OakInk-v2 validation subset of 77 primitive tasks drawn from the DexManipNet validation list, our reproduction achieves 64.1% single-hand and 29.7% bimanual success, matching the single-hand-over-bimanual trend reported in the original paper and confirming that our harness faithfully reproduces the method.

Second, we ported Sharpa into the same pipeline to isolate the effect of embodiment. We trained new Sharpa imitators from scratch on the same 77-task set, then trained and evaluated residual policies on the subset reported in Table 1. Porting Sharpa required additional embodiment-specific processing, especially separate left- and right-hand calibration of the hand-to-MANO alignment rotation. After this calibration, Sharpa reached imitator tracking success comparable to Inspire, showing that ManipTrans can train on Sharpa within its original short-primitive regime. We then used this imitator as prior to train the manipulation policies.

### G.3. Spider

We first reproduced the experiments from the original Spider codebase to verify that our results were consistent with those reported in the paper. We then integrated the Sharpa Hands model into the baseline pipeline. To generate robot motions, we used CHORD IK solver and also evaluated several alternative IK solutions with extensive parameter tuning, but none produced consistently satisfactory results. We manually inspected the generated IK motions to ensure that they were visually plausible and functionally feasible for the target tasks. Finally, we ran Spider on the selected tasks using five random seeds per task and recorded the resulting errors for analysis.