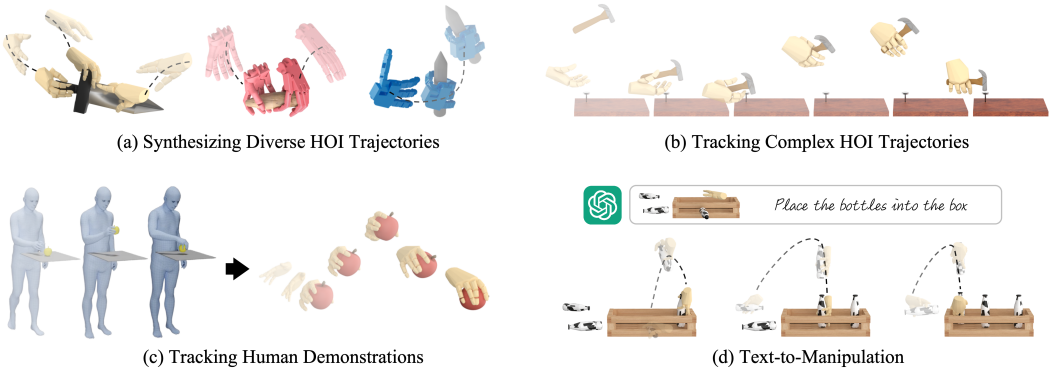
HOT: Learning Generalizable Hand-Object Tracking without Human Demonstration

Runyi Yu[1,*] **Xiaoyi Lin**[2,*] **Hok Wai Tsui**[1,*] **Yinhuai Wang**[1,*,†] **Zhijie Peng**[1]
Hui Zhang[4] **Qihan Zhao**[1] **Ke Fan**[5] **Miao Li**[2] **Jie Song**[6]
Jingbo Wang[3] **Qifeng Chen**[1,†] **Ping Tan**[1,†]
[1]HKUST [2]Wuhan University [3]Shanghai AI Laboratory
[4]ETH Zurich [5]SJTU [6]HKUST (GZ)
{ingrid.yu, yinhuai.wang}@connect.ust.hk
[*]Equal contribution [†]Corresponding authors
Project page: https://ingrid789.github.io/hot/

Abstract

We present a system for learning generalizable hand-object tracking controllers purely from synthetic data, without requiring any human demonstrations. Our approach makes two key contributions: (1) HOG, a highly efficient Hand-Object data generator that rapidly synthesizes diverse, large-scale, and uniformly distributed trajectories; and (2) HOT, a Hand-Object Tracker that bridges synthetic-to-physical transfer through reinforcement learning and interaction imitation learning. Our method extends to diverse object shapes and hand morphologies. HOT also enables cross-simulator transfer. Through extensive evaluations, we show that our approach enables dexterous hands to track challenging, long-horizon sequences including object re-arrangement and agile in-hand reorientation. These results represent a significant step toward scalable foundation controllers for manipulation that can learn entirely from simple, scalable synthetic data, breaking the data bottleneck that has long constrained progress in dexterous manipulation.



(a) Synthesizing Diverse HOI Trajectories (b) Tracking Complex HOI Trajectories

(c) Tracking Human Demonstrations (d) Text-to-Manipulation

Figure 1: Our system synthesizes and tracks hand-object trajectories for a variety of dexterous skills—such as grasping, placing, and in-hand reorientation—as well as their combinations. The approach generalizes across diverse hand and object morphologies and, notably, transfers flexibly to downstream applications including human motion tracking and text-guided task execution. We encourage readers to view our project page for visual results.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

1 Introduction

The ability to track and execute complex, dynamic hand-object interactions (HOI) with human-like dexterity is crucial for robots to perform varied dexterous manipulation tasks. Traditional teleoperation systems typically track only the robot or hand motion without considering object states, resulting in poor performance for highly dynamic, precision-demanding tasks [Cheng et al., 2024, Fu et al., 2024]. By incorporating object states into the tracking control loop, hand-object state tracking [Wang et al., 2023b, 2025, Xu et al., 2025, Yin et al., 2025, Liu et al., 2025] significantly enhances the agility and robustness of task execution.

However, training a generalizable hand-object tracking controller faces the fundamental challenge of acquiring sufficient HOI data. As summarized in Tab. 1, existing HOI data sources exhibit different trade-offs in hand generality, scalability, state-space coverage, and reliance on human demonstrations. Video-based estimation methods such as EgoGrasp [Fu et al., 2026] support scalable data acquisition, but provide limited state-space coverage. Motion-capture datasets such as GRAB [Taheri et al., 2020] and generative models such as Text2HOI [Cha et al., 2024] face additional limitations in scalability. Demonstration augmentation methods such as DexMimicGen [Jiang et al., 2024] improve scalability, while their coverage remains constrained by the source demonstrations. These representative approaches also remain tied to specific hand representations and depend on human demonstration data. HOG combines hand-agnostic synthesis, high scalability, and broad state-space coverage without requiring human demonstrations.

In this paper, we address this data challenge by exploring a different route: instead of improving how existing demonstration data are reused, we eliminate the need for human demonstrations in training altogether. Our main contribution is therefore systemic and conceptual: we show that a high-performance, generalizable hand-object tracker can be learned from fully synthetic hand-object data, enabling diverse zero-shot applications. Our approach is built on three key insights: (1) Dexterous manipulation involves continuous transitions between hand-object states, particularly between different grasp configurations. While capturing such continuous interactions is difficult, static states like grasp configurations can be synthesized in large quantities [Liu et al., 2021, Zhang et al., 2024a, Wang et al., 2023a]. We capitalize on this asymmetry by generating synthetic manipulation sequences through the transformation, interpolation, and concatenation of these static key frames into plausible pseudo-demonstrations. (2) Although synthetic HOI data may contain physical implausibilities, we find that a policy trained to imitate these HOI data through reinforcement learning can robustly track the corresponding interactions. (3) A generalizable HOI tracker can be trained effectively on sequences of meta-skills, rather than on long, complete tasks. For instance, instead of modeling "grasping a hammer and hammering a nail," we synthesize elemental skills like Grasp, Rotate, and Move. This decomposition supports training on short skills; we evaluate zero-shot composition on Grasp–Move–Place sequences, while more complex composites use dedicated training.

Based on these insights, we present HOG (Hand-Object Generator), a scalable manipulation data synthesis framework for generating diverse hand-object trajectories across various object shapes and hand morphologies, and HOT (Hand-Object Tracker), a tracking-based controller trained on these demonstrations, capable of tracking complex, highly dynamic, and long-horizon manipulation trajectories during inference. The key role of HOG is not merely to generate more data, but to produce data that are most conducive to scaling and generalization: supporting diverse objects, diverse hands, diverse skills, and a wide range of grasping and trajectory distributions. HOG generates synthetic meta-skill trajectories through a structured, two-stage process. First, given a rigid object and a dexterous hand model, it employs force-closure optimization and reinforcement learning [Wang et al., 2023a, Zhang et al., 2024a] to generate and refine a diverse set of stable grasp configurations. Subsequently, HOG anchors itself to these grasp poses to synthesize trajectories for fundamental meta-skills—such as Grasping, Moving, Placing, and Rotating—as well as their combinations. HOT is trained primarily on the meta-skill data synthesized by HOG, using a unified set of HOI imitation rewards. To enhance model capacity and learning efficiency, we introduce a two-stage teacher-student training framework, coupled with a data randomization curriculum that progressively increases the randomness of training samples—such as varying object sizes and initial poses—as the model converges, thereby improving overall robustness.

Our core benchmark compares DexTrack-style specialists, HOT teachers, and one unified seven-skill HOT student on the same HOG data and four objects. We use a separately trained Grasp/Move/Place student to evaluate object generalization after training on 15 objects and zero-shot composition

Table 1: Comparison of HOI data sources. Unlike existing pipelines that are often constrained by human demonstrations, specific hand morphologies, or narrow state-space coverage, HOG is hand-agnostic and offers high scalability and broad state-space coverage. This makes it an ideal framework for training highly generalizable hand-object trackers.

Data Source \ Property	Method	Hand-Agnostic	Scalability	State-Space Coverage	w/o Human Demo.
Video Estimation	EgoGrasp	✗	✓	✗	✗
Generative Model	Text2HOI	✗	✗	✗	✗
Dataset	GRAB	✗	✗	✗	✗
Data Augmentation	DexMimicGen	✗	✓	✗	✗
HOG (Ours)	-	✓	✓	✓	✓

on four-object Grasp–Move–Place trajectories. Beyond downstream tracking performance, we show that HOG occupies a favorable point in the data-design space compared with video-estimated trajectories such as EgoGrasp [Fu et al., 2026], motion-capture datasets, generative HOI models, and augmentation-based pipelines: it preserves physical structure while achieving substantially better scalability and state-space coverage. Notably, we find that the fast and intentionally noisy generation process of HOG is crucial for endowing HOT with robust interaction dynamics. With dedicated composite-task training, HOT also tracks long-horizon dynamic sequences, such as catching a thrown hammer, reorienting it in midair, and performing a hammering motion. Moreover, HOT shows effectiveness in tracking real human demonstrations [Taheri et al., 2020] and data from HOI generative models [Cha et al., 2024]. Notably, the system enables zero-shot transfer to downstream applications. When integrated with LLMs or VLMs [Achiam et al., 2023, Team et al., 2023], it follows language instructions to control both HOG-based data synthesis and HOT 's tracking for manipulation.

2 Related Work

Reinforcement Learning for Dexterous Manipulation. Recent years have witnessed significant progress in dexterous manipulation, particularly in grasping tasks [Chen et al., 2023b, Huang et al., 2023, Chen et al., 2023a, Yin et al., 2023, Andrychowicz et al., 2020, Fang et al., 2023, Allshire et al., 2022, Rajeswaran et al., 2018, She et al., 2022, 2024]. By leveraging advances in reinforcement learning [Schulman et al., 2017] and simulation technology [Makoviychuk et al., 2021, Todorov et al., 2012], current grasp models can handle objects of varying shapes [Zhang et al., 2024a, Xu et al., 2023, Wan et al., 2023]. However, these approaches typically rely on task-specific reward engineering, limiting their generalization to broader manipulation tasks [Chen et al., 2023b]. While similar paradigms have shown impressive results in specific dexterous tasks like object reorientation [Andrychowicz et al., 2020] and articulated object manipulation [Zhang et al., 2024b], they struggle to scale to general manipulation scenarios. Some methods attempt to enhance task diversity through a mixture of experts, but they still cannot circumvent the labor-intensive nature of task-specific reward engineering [Yang et al., 2020]. Recent studies [Wang et al., 2023b, 2025, Xu et al., 2025, Liu et al., 2025, Chen et al., 2024] propose a paradigm shift by introducing general imitation rewards to mimic robot-object state trajectories, enabling unified learning across different manipulation tasks. Notably, this paradigm demonstrates impressive robustness to data imperfections, substantially reducing requirements for demonstration quality and quantity. However, acquiring diverse manipulation data remains a significant challenge.

Demonstration Data for Dexterous Manipulation. Data collection remains a critical challenge for learning dexterous manipulation skills. Current approaches primarily fall into two categories: human demonstrations and synthetic data generation.

Human demonstrations are typically collected through teleoperation, device-based motion capture, and video-based estimation. While teleoperation provides accurate action data, it suffers from latency and perception barriers, making agile and precise manipulations challenging [Fu et al., 2024, He et al., 2024, Cheng et al., 2024]. Motion capture systems using specialized sensor gloves and camera arrays can capture detailed hand-object motion [Banerjee et al., 2024, Taheri et al., 2020, Fan et al., 2023], but incur substantial time and labor costs and remain tied to specific embodiments, objects, and recording setups. Recent advances in video-based pose estimation enable extraction of hand-object

motion from vast video datasets [Cao et al., 2021, Wen et al., 2023, Wang et al., 2023b]; however, these methods often struggle with occlusion, depth ambiguity, and unstable contact reconstruction, which limits their use as large-scale supervisory signals for precise HOI tracking.

Synthetic data generation has gained increasing attention recently [Mandlekar et al., 2023, Jiang et al., 2024, Garrett et al., 2024, Gao et al., 2024, Pumacay et al., 2024], with methods mainly focusing on transforming and recombining existing reference data [Mandlekar et al., 2023, Jiang et al., 2024, Garrett et al., 2024]. This improves combinatorial diversity, but it does not remove the underlying dependence on human demonstrations, and thus cannot escape the coverage ceiling imposed by the source corpus. HOI generative models are another promising direction, yet current systems still face difficulties in preserving physical validity, stable grasping, and broad state-space coverage for dexterous tracking supervision [Cha et al., 2024].

Our work instead focuses on how the training data distribution shapes generalization in HOI tracking. HOG is designed to directly control this distribution by synthesizing physically grounded and scalable trajectories without human demonstrations, overcoming the limited coverage of captured or augmented data and the weak physical reliability of current generative models.

3 Method

Our system consists of two parts: HOG (Hand-Object Generator), a scalable data generation framework, and HOT (Hand-Object Tracker), a tracking-based controller trained on synthetic data, capable of tracking complex, highly dynamic, and long-horizon manipulation trajectories during inference. In applications, our system follows a modular pipeline. First, a high-level directive—such as a text command processed by a VLM, a trajectory from an HOI generative model, or a captured human demonstration—is converted into sparse waypoints. These waypoints are then fed to HOG to generate a dense hand-object trajectory, which HOT executes in simulation. This workflow enables zero-shot adaptation to diverse downstream applications. Fig. 2 shows the overall system architecture.

3.1 Hand-Object Generator

The Hand-Object Generator (HOG) consists of two steps: first, synthesizing stable grasp configurations, and second, generating meta-skill demonstrations or their combinations based on these grasp configurations. Fig. 3 illustrates our data generation pipeline.

3.1.1 Synthesizing Grasp Pose Candidates

Static grasp configurations can be easily obtained from existing datasets [Wang et al., 2023a, Chao et al., 2021, Hasson et al., 2019, Taheri et al., 2020, Fan et al., 2023], grasp models [Xu et al., 2023, Wan et al., 2023, Zhang et al., 2024a, Miller and Allen, 2004], or through optimization [Wang et al., 2023a, Liu et al., 2021], thereby ensuring substantial scalability potential. In this study, we leverage an off-the-shelf grasp generator [Wang et al., 2023a] driven by differentiable force closure estimation [Liu et al., 2021] to generate diverse grasp configurations. Specifically, given a rigid object and a dexterous hand, we obtain a set of grasp configurations $\hat{\mathcal{G}} = \{\boldsymbol{g}_1, \boldsymbol{g}_2, ..., \boldsymbol{g}_M\}$. The grasp configuration is parameterized by the hand-object state

$$\boldsymbol{g} = \{\mathbf{T}^h, \mathbf{R}^h, \boldsymbol{\theta}, \mathbf{P}^h, \mathbf{T}^o, \mathbf{R}^o, \mathbf{P}^o\}. \tag{1}$$

For the MANO hand model [Romero et al., 2022], $\mathbf{T}^h \in \mathbb{R}^3$ and $\mathbf{R}^h \in SO(3)$ represent the global wrist pose, while $\boldsymbol{\theta} \in \mathbb{R}^{15\times3}$ describes the joint angles of the five fingers (each finger has three joints with three degrees of freedom). $\mathbf{P}^h \in \mathbb{R}^{15\times3}$ denotes the global positions of finger joints. Similarly, $\mathbf{T}^o$, $\mathbf{R}^o$, and $\mathbf{P}^o \in \mathbb{R}^{N\times3}$ represent the object's global pose and the global positions of N keypoints on the object surface.

3.1.2 Filtering and Refining Grasp Poses

After obtaining a large set of initial grasp pose candidates, we train a coach policy to filter and refine them into physically plausible and stable configurations for subsequent data synthesis. Specifically, the policy is trained in simulation by initializing the hand and object according to each candidate pose. The coach policy is optimized using a grasp reward that encourages successful grasping and adherence to the original candidate pose. To enhance robustness, we apply random force perturbations

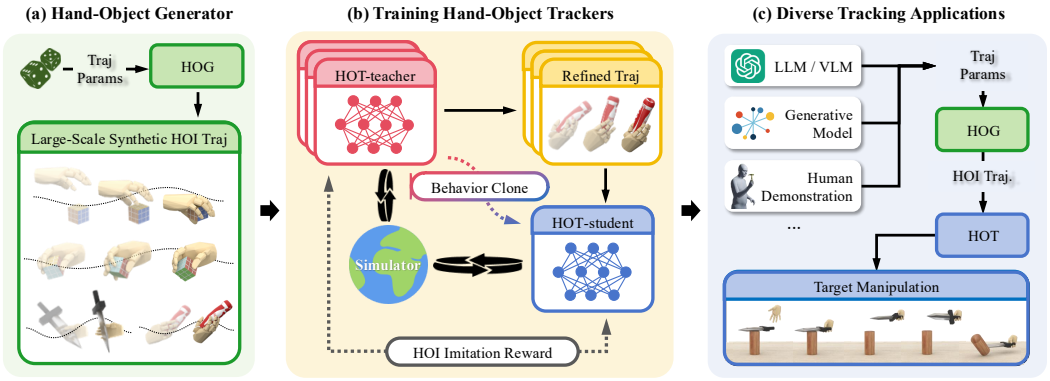


Figure 2: Our system learns generalizable hand-object tracking from synthetic data. **(a) HOG** synthesizes manipulation trajectories for meta-skills. **(b) HOT** is trained through a two-stage teacher-student framework using reinforcement learning with a unified HOI imitation reward, enabling robust tracking of the target HOI trajectories. **(c) At inference**, the system can accept high-level waypoints from language models, generative models, or human data, which HOG converts into trajectories for HOT to track, enabling diverse applications.

to the object during training. Additional implementation details of the coach policy follow the setup described in Sec. 3.2. Once the policy converges, we collect the successful grasp poses generated during its rollouts as the stable grasp pose set $\mathcal{G}$. These refined poses are significantly higher in quality, physical realism, and stability compared to the original candidates.

3.1.3 Synthesizing Meta-Skill Trajectories

A dexterous manipulation trajectory is represented as $\boldsymbol{\tau} = \{\boldsymbol{g}_t\}_{t=0}^{T}$, with each $\boldsymbol{g}_t$ parameterized as in Eq. 1. We synthesize pseudo-demonstrations by interpolating randomly generated hand-object states and grasp configurations. While these trajectories may be physically approximate, they serve as tracking targets in reinforcement imitation learning, enabling the policy to learn generalizable target-action mappings. For efficient synthesis and learning, we decompose manipulation into 7 meta-skills: Grasp, Place, Move, Catch, Throw, Rotate, and Regrasp. Below, we detail the synthesis of Grasp, Move, Place, and Rotate; see the Appendix for the others.

Grasp. The Grasp skill involves grasping objects placed on the ground. First, we determine the set of stable object poses for a given object through physics simulation by randomly dropping objects and recording their resting poses, denoted as $\mathcal{H} = \{\mathbf{h}_1, \mathbf{h}_2, ..., \mathbf{h}_K\}$, where each $\mathbf{h} = [\mathbf{T}^o, \mathbf{R}^o]$ represents a stable object pose. To generate a grasp trajectory, we sample a grasp configuration $\boldsymbol{g}$ from $\mathcal{G}$ and a stable object pose $\mathbf{h}$ from $\mathcal{H}$, then transform $\boldsymbol{g}$ to align the object pose with $\mathbf{h}$, obtaining $\boldsymbol{g}^{end}$. We reject and resample $\boldsymbol{g}^{end}$ if any hand keypoints penetrate the ground plane; otherwise, we keep it as the grasping frame. To generate the initial state $\boldsymbol{g}^{init}$, we compute the vector $\mathbf{v}$ from the object center to the first joint of the index finger, randomly sample the initial hand position within a cone centered along $\mathbf{v}$, and set all hand joint angles to zero while keeping other parameters identical to $\boldsymbol{g}^{end}$. Finally, we generate the continuous grasp trajectory by interpolating between $\boldsymbol{g}^{init}$ and $\boldsymbol{g}^{end}$.

Place. Place represents the inverse operation of Grasp: an object is placed stably on a surface and then released by the hand. We efficiently generate Place demonstrations by time-reversing the Grasp trajectories.

Move. Move involves moving an object through space while maintaining a fixed grasp configuration. To synthesize these demonstrations, we first randomly select a grasp configuration $\boldsymbol{g}$ from $\mathcal{G}$. We then generate random initial and final object poses within the operational space and create an object trajectory through interpolation. The wrist pose is subsequently aligned with the object trajectory at each timestep, maintaining the constant relative grasp configuration throughout the motion. Notably, we also generate Move trajectories starting from the final frame of Grasp demonstrations or ending at the initial frame of Place demonstrations to ensure smooth cross-skill transitions.

Rotate. Rotate demonstrates in-hand manipulation where the object is rotated while maintaining a fixed wrist pose. We provide details for generating general Rotate data in the appendix. However,

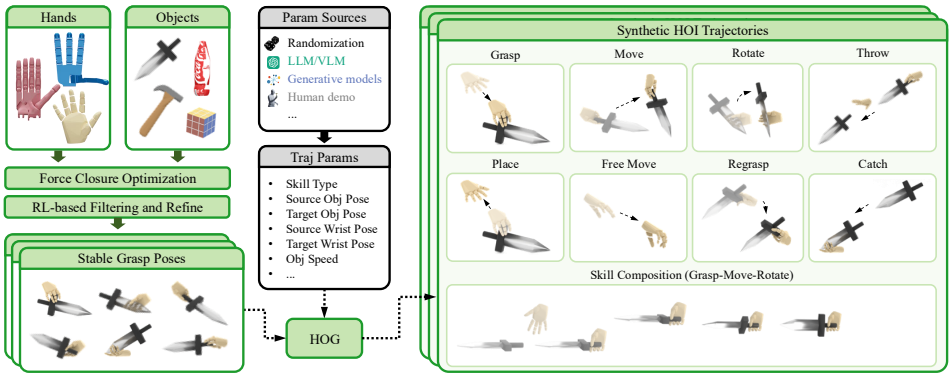


Figure 3: HOG synthesizes manipulation trajectories from grasp poses generated by force-closure optimization and refined by RL. It supports 8 meta-skills (7 HOI skills + Free Move) and multiple parameter sources, including randomization, LLM/VLM instructions, and human demonstrations, while naturally generalizing across diverse hands and objects.

due to its difficulty and lengthy training requirements, we propose a simplified version that balances practical value with complexity. First, we define a rotatable region for each object, such as the handle of a hammer, and an associated rotation axis. If a grasp configuration $\boldsymbol{g}^1$ from $\mathcal{G}$ contacts the object's rotatable region, we designate it as the first keyframe. We then rotate the object by a random angle around the defined axis to create the second keyframe, and continue this process for subsequent keyframes. To accommodate large state changes, we replicate each $\boldsymbol{g}_i$ for N_i frames, with N_i scaled by the magnitude of the state difference. This provides a sufficient learning buffer for the policy.

The resulting demonstration is:

$$\{\underbrace{\boldsymbol{g}_1, ..., \boldsymbol{g}_1}_{N_1}, \underbrace{\boldsymbol{g}_2, ..., \boldsymbol{g}_2}_{N_2}, ..., \underbrace{\boldsymbol{g}_k, ..., \boldsymbol{g}_k}_{N_k}\}, \tag{2}$$

3.2 Hand-Object Tracker

The final goal is to train a Hand-Object Tracker (HOT) that can not only track various meta-skill demonstrations but also track composite long-horizon sequences. We employ reinforcement learning in physical simulation with a unified imitation reward to learn from synthetic demonstrations generated using the procedure in Sec. 3.1. We condition the controller on target hand-object states, enabling the controller to accomplish tasks of varying duration and semantic complexity through tracking. Further, we improve the tracking performance through a two-stage teacher-student training framework.

3.2.1 HOT Structure

We employ a single policy that outputs controls for both the hand's native joints (e.g., MANO's 45 DoFs) and the wrist's 6-DoF pose, which is essential for dynamic tasks. To enhance generalization, we leverage the translational invariance of manipulation: the relative control strategy should be location-agnostic. Consequently, our policy is trained to output residual adjustments for the wrist, ensuring generalization across the workspace.

HOT is formulated as a policy parameterized by an MLP. The policy output is parameterized as a Gaussian distribution

$$\boldsymbol{a}_t \sim \mathcal{N}(\boldsymbol{\phi}_{\boldsymbol{\pi}}(\boldsymbol{h}_t), \boldsymbol{\Sigma}_{\boldsymbol{\pi}}), \tag{3}$$

where $\boldsymbol{\phi}_{\boldsymbol{\pi}}$ is a three-layer MLP that maps input $\boldsymbol{h}_t$ to action means. The variances $\boldsymbol{\Sigma}_{\boldsymbol{\pi}}$ are set to non-zero constants during training for exploration and to zero during testing for stability. The action $\boldsymbol{a}_t$ consists of a residual 6-DoF wrist pose $\boldsymbol{\delta}_t^{wrist}$ and target finger joint rotations $\boldsymbol{a}_t^{finger}$. For wrist pose control, we employ an incremental control scheme where the target 6-DoF wrist pose is updated as $\boldsymbol{a}_t^{wrist} = \boldsymbol{a}_{t-1}^{wrist} + \boldsymbol{\delta}_t^{wrist}$. For finger joints, we apply direct position control using $\boldsymbol{a}_t^{finger}$ as the target joint rotations. These targets are converted to torque commands through PD controllers.

The observation of the HOT policy is

$$\boldsymbol{h}_t = \{\boldsymbol{s}_t, \hat{\boldsymbol{s}}_{t+1}^{goal}\} \tag{4}$$

where $\boldsymbol{s}_t = \{\boldsymbol{s}_t^{hand}, \boldsymbol{s}_t^{obj}\}$ is the environmental observation, consisting of the current hand state $\boldsymbol{s}_t^{hand}$ and object state $\boldsymbol{s}_t^{obj}$ in the simulation environment. The hand state $\boldsymbol{s}_t^{hand}$ encompasses the global wrist position $\boldsymbol{s}_t^{wp}$ and rotation $\boldsymbol{s}_t^{wr}$, finger joint angles $\boldsymbol{s}_t^{fr}$ and angular velocities $\boldsymbol{s}_t^{frv}$, finger joint positions $\boldsymbol{s}_t^{fp}$ and velocities $\boldsymbol{s}_t^{fpv}$, and fingertip force sensor readings $\boldsymbol{s}_t^{fs}$. The object state $\boldsymbol{s}_t^{obj}$ consists of the object position $\boldsymbol{s}_t^{op}$, rotation $\boldsymbol{s}_t^{or}$, velocity $\boldsymbol{s}_t^{opv}$, angular velocity $\boldsymbol{s}_t^{orv}$, and the positions of keypoints on the object surface $\boldsymbol{s}_t^{okp}$. $\hat{\boldsymbol{s}}_{t+1}^{goal}$ denotes the goal state for the subsequent time step, which encompasses multiple components:

$$\hat{\boldsymbol{s}}_{t+1}^{goal} = \{\hat{\boldsymbol{s}}_{t+1}^{wr}, \hat{\boldsymbol{s}}_{t+1}^{wp}, \hat{\boldsymbol{s}}_{t+1}^{wpv}, \hat{\boldsymbol{s}}_{t+1}^{fr}, \hat{\boldsymbol{s}}_{t+1}^{op}, \hat{\boldsymbol{s}}_{t+1}^{or}\} \tag{5}$$

Here, $\hat{\boldsymbol{s}}_{t+1}^{wpv}$ denotes the reference wrist linear velocity. The remaining symbols follow the notation above, and all goal state values are extracted from the next frame of reference data. Notably, all positional measurements (except for the global wrist position) are transformed into a local coordinate frame attached to the wrist.

3.2.2 Training the Hand-Object Tracker

As illustrated in Fig. 2(b), we train the Hand-Object Tracker (HOT) using a two-stage reinforcement learning framework guided by synthetic demonstration trajectories with unified HOI imitation rewards. Training starts by sampling a HOG-generated trajectory $\tau = \{\hat{\boldsymbol{s}}_t\}_{t=0}^{T}$ (Sec. 3.1) and a random starting index t_0. The hand-object system is initialized from $\hat{\boldsymbol{s}}_{t_0}$ with perturbations (Appendix D), and the policy tracks the subsequent reference states. At each timestep, the policy outputs actions (Eq. 3) that are executed in simulation, after which an imitation reward $r_t = f(\boldsymbol{s}_{t+1}, \hat{\boldsymbol{s}}_{t+1})$ (Sec. 3.2.3) is computed by comparing the current state with the reference. Network parameters $\boldsymbol{\phi_\pi}$ are optimized using PPO [Schulman et al., 2017].

Although a single policy is effective for individual skills, it faces scalability limitations when learning diverse skills or object categories. The primary challenge arises from physical implausibilities in synthetic data, which require extensive sampling to identify feasible solutions for each instance. As the dataset expands, sampling density per trajectory decreases, adversely affecting convergence. To overcome this bottleneck, we adopt a two-stage teacher-student distillation framework [He et al., 2024, Luo et al., 2023]. In the first stage, we train specialist teachers for each skill–object pair in the seven-skill benchmark, or object-specific teachers covering Grasp, Move, and Place in the multi-object experiment. In the second stage, one student learns the skills and objects covered by the corresponding teacher set, using the teacher policies for both online data refinement and action guidance through behavior cloning [Juravsky et al., 2024, Xu et al., 2025]. We train the seven-skill/four-object student and the three-skill/15-object student separately; each consolidates its own teacher set into one policy.

3.2.3 Unified Hand-Object Imitation Reward

We apply a unified imitation reward to learn manipulation from demonstrations. The reward combines complementary components multiplicatively:

$$r_t = r_t^{hand} * r_t^{wrist} * r_t^{object} * r_t^{interact} * r_t^{contact}, \tag{6}$$

where r_t^{hand} and r_t^{wrist} are the hand and wrist tracking rewards that encourage following demonstrated global hand motion trajectories, r_t^{object} captures global object motion, $r_t^{interact}$ highlights hand-object relative motion, and $r_t^{contact}$ encourages correct per-frame contact status. Given the degradation in our demonstration data, we deliberately exclude velocity terms from the imitation objectives to ensure robust learning. Details about the sub-rewards can be found in the appendix.

4 Experiments

4.1 Experimental Settings

Settings Experiments use Isaac Gym [Makoviychuk et al., 2021] and one RTX 4090 GPU, with simulation control at 120 Hz and policies at 60 Hz. HOT teachers train for 30,000 epochs and students for 20,000; DexTrack-style uses $3 \times 20{,}000$ epochs (Appendix F). The initial HOG data are shared, but compute is not matched.

Table 2: **Baseline comparison and reward ablations on meta-skills.** Reward ablation results are reported for the teacher policies. Bold and underlined values indicate the best and second-best distinct values in each column across all listed methods, respectively; ties share the same formatting. See Sec. 4.1 for metric definitions.

Method	*Grasp*				*Move*				*Place*				*Rotate*			
	SR (%) ↑	E_{op} ↓	E_{or} ↓	E_h ↓	SR (%) ↑	E_{op} ↓	E_{or} ↓	E_h ↓	SR (%) ↑	E_{op} ↓	E_{or} ↓	E_h ↓	SR (%) ↑	E_{op} ↓	E_{or} ↓	E_h ↓
Open-Loop	$34.2^{\pm 1.2}$	6.05	27.26	**2.50**	$82.0^{\pm 0.7}$	4.33	24.88	4.84	$91.5^{\pm 0.5}$	1.93	22.78	**1.99**	$0.0^{\pm 0.0}$	**1.31**	72.62	**0.99**
SkillMimic	$57.0^{\pm 1.5}$	6.15	21.99	5.69	$88.0^{\pm 0.6}$	4.56	17.46	5.70	$87.9^{\pm 0.8}$	2.42	14.92	4.61	$17.5^{\pm 1.0}$	8.41	86.51	36.36
DexGen-var	$4.8^{\pm 0.4}$	7.81	13.15	5.83	$78.7^{\pm 1.0}$	8.06	28.63	5.10	$79.0^{\pm 1.1}$	3.00	19.91	4.44	$0.0^{\pm 0.0}$	2.23	72.50	2.83
DexTrack-style	**$99.8^{\pm 0.1}$**	2.60	**9.85**	2.95	$97.9^{\pm 0.2}$	2.35	10.20	4.80	**$99.7^{\pm 0.2}$**	**0.95**	**8.75**	2.60	$34.5^{\pm 1.2}$	3.10	58.40	3.05
HOT w/o r^{wrist}	$77.93^{\pm 1.1}$	**2.07**	14.48	5.55	$93.8^{\pm 0.6}$	2.60	11.65	5.35	$55.61^{\pm 0.6}$	1.20	12.75	3.04	$42.7^{\pm 1.8}$	3.42	54.71	3.06
HOT w/o $r^{contact}$	$0.0^{\pm 0.0}$	10.80	24.60	12.40	$96.5^{\pm 0.6}$	2.60	11.30	5.10	$96.0^{\pm 0.5}$	1.10	9.70	2.85	$45.8^{\pm 1.4}$	2.95	52.80	3.25
HOT w/o $r^{interact}$	$72.3^{\pm 1.3}$	5.10	15.80	5.40	$98.2^{\pm 0.4}$	2.20	10.10	4.70	$96.1^{\pm 0.6}$	1.15	10.60	2.95	$5.8^{\pm 0.5}$	6.80	71.50	5.20
HOT (Teacher)	$86.0^{\pm 1.0}$	3.80	11.76	4.32	**$98.9^{\pm 0.4}$**	**2.01**	**9.83**	**4.50**	$94.3^{\pm 0.4}$	1.24	9.88	3.14	**$56.2^{\pm 0.8}$**	2.38	**44.96**	2.57
HOT (Student)	$79.7^{\pm 1.2}$	4.65	14.80	5.10	$97.7^{\pm 0.5}$	2.45	11.20	4.95	$97.3^{\pm 0.6}$	1.05	9.40	2.80	$45.4^{\pm 1.5}$	3.20	52.10	3.15

Method	*Throw*				*Catch*				*Regrasp*				**Average**			
	SR (%) ↑	E_{op} ↓	E_{or} ↓	E_h ↓	SR (%) ↑	E_{op} ↓	E_{or} ↓	E_h ↓	SR (%) ↑	E_{op} ↓	E_{or} ↓	E_h ↓	SR (%) ↑	E_{op} ↓	E_{or} ↓	E_h ↓
Open-Loop	$9.2^{\pm 0.6}$	20.20	91.64	**2.56**	$27.9^{\pm 1.4}$	13.04	89.30	**1.51**	$0.0^{\pm 0.0}$	21.89	71.84	**1.31**	35.0	9.82	57.19	**2.24**
SkillMimic	$0.4^{\pm 0.1}$	45.41	97.86	46.43	$64.8^{\pm 1.3}$	6.96	53.65	4.31	$0.0^{\pm 0.0}$	7.54	64.28	4.18	45.1	11.64	50.95	15.33
DexGen-var	$0.6^{\pm 0.2}$	41.19	90.00	44.72	$35.9^{\pm 1.6}$	12.13	85.61	4.35	$0.0^{\pm 0.0}$	10.47	58.29	4.92	28.4	12.13	52.58	10.31
DexTrack-style	$72.4^{\pm 1.5}$	19.80	47.50	10.20	**$83.6^{\pm 0.6}$**	3.55	46.20	2.85	$63.4^{\pm 1.0}$	2.45	22.80	3.65	78.8	4.97	29.10	4.30
HOT w/o r^{wrist}	$82.4^{\pm 1.0}$	18.25	42.93	9.62	$78.8^{\pm 0.5}$	4.61	55.21	3.57	$76.1^{\pm 1.4}$	2.03	16.44	3.45	72.5	4.88	29.74	4.81
HOT w/o $r^{contact}$	$0.0^{\pm 0.0}$	45.41	97.86	46.43	$30.7^{\pm 1.7}$	13.04	89.30	**1.51**	$72.1^{\pm 1.1}$	2.15	19.60	3.40	48.7	11.15	43.59	10.71
HOT w/o $r^{interact}$	$82.3^{\pm 1.2}$	17.60	42.80	9.15	$74.1^{\pm 1.0}$	5.20	52.30	3.65	$30.5^{\pm 1.4}$	5.60	39.20	5.70	65.6	6.24	34.61	5.25
HOT (Teacher)	**$91.9^{\pm 1.2}$**	**15.04**	**35.27**	8.34	**$83.6^{\pm 0.4}$**	**3.34**	**45.00**	2.67	**$81.9^{\pm 0.5}$**	**1.56**	**14.05**	2.90	**84.7**	**4.20**	**24.39**	4.06
HOT (Student)	$86.7^{\pm 1.4}$	16.90	40.50	9.25	$76.5^{\pm 0.9}$	4.80	51.60	3.40	$78.5^{\pm 1.1}$	1.95	17.30	3.25	80.3	5.00	28.13	4.56

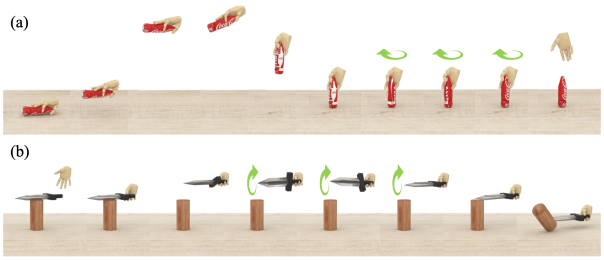


Figure 4: Long-horizon tracking with the HOT student.

Table 3: Zero-shot tracking of long-horizon Grasp–Move–Place sequences with the distilled HOT student.

Dataset	SR (%) ↑	E_{op} ↓	E_{or} ↓	E_h ↓
Bottle	58.54	9.00	34.77	4.94
Box	84.21	5.69	34.45	4.69
Hammer	57.91	9.40	38.33	5.37
Sword	68.42	10.40	33.31	4.80
Average	**67.27**	**8.62**	**35.22**	**4.95**

Baseline methods We compare HOT against four baselines: Open-Loop, SkillMimic-style [Wang et al., 2025], DexGen-var [Yin et al., 2025], and DexTrack-style [Liu et al., 2025]. (1) Open-Loop is a controller that directly drives a PD controller using joint rotations from reference demonstrations. (2) SkillMimic uses an HOI imitation method that observes only the 6D object pose without shape information. (3) DexGen-var is our reimplementation of DexGen following its original design, inferring object states indirectly from PD controller errors without direct object observations. (4) DexTrack-style implements DexTrack's homotopy optimization within our codebase, refining reference trajectories over three rounds using successful policy rollouts. See details in Appendix F.

Metrics We measure object position error E_{op} (cm), object rotation error E_{or} (degrees), hand mean per-joint position error E_h (cm), and frame-wise success rate (SR). A frame is considered successful when both E_{op} and E_h fall below a skill-specific threshold: 10 cm for Grasp, Move, and Place, and 20 cm for Throw, Catch, Rotate, and Regrasp. Tab. 2 reports means across four objects (box, bottle, sword, and hammer) and five seeds, with standard deviations across seeds reported only for SR. Each seed uses 1,000 trials per object–skill pair.

4.2 Experimental Results

We evaluate seven skills with MANO on four objects: a box, a bottle, a sword, and a hammer. For each category, we generate about 20 grasp configurations, 1000 training demonstrations per skill, and 100 test demonstrations per skill, all at 60 fps. A dedicated HOT teacher is trained for each skill-object pair.

Table 4: Comparison of HOI data sources in terms of data quality and coverage. Penetration and jittering are reported in mm. GSR denotes the stable grasp rate. See Sec. 4.4 for details.

Source Type	Method	Data Quality			Data Coverage	
		Penetration ↓	Jittering ↓	GSR ↑	State-Space Coverage ↑	Grasp Diversity ↑
Video Estimation	EgoGrasp	4.52	2.36	14%	0.61	0.38
Generative Model	LatentHOI [Li et al., 2025b]	2.55	26.88	24%	0.08	0.23
Motion Capture	GRAB	4.49	0.03	28%	0.62	0.42
Data Augmentation	DexMimicGen	0.00	2.31	100%	0.02	0.08
HOG (Ours)	-	**1.28**	**1.43**	**100%**	**1.97**	**0.54**

Baseline configurations. In Tab. 2, DexTrack-style and HOT teachers use separate policies for each skill–object pair, while the HOT student is a single policy distilled across all seven skills and four objects. Additional implementation details are provided in Appendix F.

Performance. Quantitative results are summarized in Tab. 2. The Open-Loop baseline performs well in Place and Move tasks, as these skills either maintain a stable force-closure grasp (Move) or involve a simple object release onto a surface (Place). However, for skills such as Grasp—where synthetic references may contain physical inconsistencies (see Fig. 6)—the open-loop strategy often fails. In contrast, by using the same imperfect references within a reinforcement imitation learning framework, HOT yields high success rates and low tracking errors across most skills.

Both SkillMimic [Wang et al., 2025] and DexGen-var [Yin et al., 2025] perform substantially worse than HOT. DexGen-var struggles with precise object tracking due to its lack of direct object state observation. The low performance of SkillMimic further underscores the importance of our design choices—including reward shaping, observation localization, geometry perception, residual action, and adaptive sampling—in achieving robust HOI tracking performance.

DexTrack-style performs best on Grasp and Place, while HOT teachers achieve higher success rates on Rotate, Regrasp, and Throw, yielding a higher average SR (84.7% vs. 78.8%). The gains on Rotate and Regrasp suggest greater robustness to imperfect references generated from discrete keyframes, consistent with HOT's domain randomization curriculum.

The unified HOT student outperforms all four baselines in average SR (80.3%). Its advantages over DexTrack-style are most evident on Rotate, Throw, and Regrasp, suggesting that distillation preserves the teachers' ability to handle challenging contact transitions and imperfect references. Its modest gap to the specialist teachers (84.7%) reflects a practical trade-off between tracking performance and policy consolidation.

4.3 Ablation Study

We conduct ablation studies to evaluate the impact of key design choices. Using motion clips for four objects across seven meta-skills, we analyze reward design with 1000 training and 100 test clips per skill. As shown in Tab. 2, the contact reward is particularly important for grasping and dynamic manipulation: removing it reduces Grasp and Throw success rates to 0.0% and Catch success to 30.7%. Tab. 2 also shows that each term contributes noticeably, and the full reward yields the best overall results.

4.4 HOG as a Training Data Source

We compare HOG against four representative HOI data sources: video-estimated motion, generative HOI models, motion-capture data, and demonstration-driven data augmentation. Using two objects (bowl and cylinder), we evaluate each source along two axes: *data quality*, which measures physical usability for tracking, and *data coverage*, which measures diversity of object poses and grasp states.

The left half of Tab. 4 shows that each alternative source misses at least one key property. Video-estimated trajectories and GRAB capture human motion but exhibit substantial hand-object penetration and yield low grasp success rates after retargeting. Generative trajectories offer flexibility, yet exhibit large temporal jitter and low grasp stability. DexMimicGen maintains high trajectory quality, but only by reusing and augmenting demonstrations rather than expanding the reachable HOI state

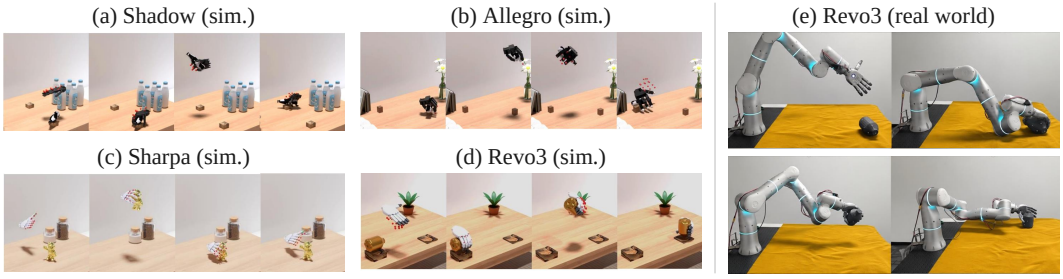


Figure 5: HOI tracking with different dexterous hands. Red dots denote hand joint tracking targets.

Table 5: **Tracking performance across hand morphologies** on three bottle manipulation skills.

Hand	*Grasp*				*Move*				*Place*				**Average**			
	SR (%) ↑	E_{op} ↓	E_{or} ↓	E_h ↓	SR (%) ↑	E_{op} ↓	E_{or} ↓	E_h ↓	SR (%) ↑	E_{op} ↓	E_{or} ↓	E_h ↓	SR (%) ↑	E_{op} ↓	E_{or} ↓	E_h ↓
Shadow	$92.6^{\pm 0.6}$	1.31	8.39	3.16	$95.8^{\pm 0.4}$	2.48	16.02	4.23	$86.3^{\pm 0.8}$	1.15	6.73	2.85	$91.6^{\pm 0.4}$	1.65	10.38	3.41
Allegro	$84.8^{\pm 1.0}$	2.01	19.50	6.60	$89.8^{\pm 0.8}$	3.44	23.78	6.43	$78.8^{\pm 1.2}$	1.66	13.72	5.23	$84.5^{\pm 0.7}$	2.37	19.00	6.09
Sharpa	$91.2^{\pm 0.7}$	1.48	10.72	3.54	$95.1^{\pm 0.5}$	2.62	17.35	4.51	$85.6^{\pm 0.9}$	1.24	8.16	3.12	$90.6^{\pm 0.5}$	1.78	12.08	3.72
Revo3	$89.5^{\pm 0.8}$	1.62	12.84	4.05	$93.7^{\pm 0.6}$	2.85	18.96	4.87	$83.9^{\pm 1.0}$	1.36	9.72	3.46	$89.0^{\pm 0.6}$	1.94	13.84	4.13
HOT Teacher	$94.4^{\pm 0.6}$	2.10	10.27	3.63	$99.1^{\pm 0.2}$	1.55	7.38	3.62	$98.2^{\pm 0.3}$	1.04	9.05	2.65	$97.3^{\pm 0.3}$	1.56	8.90	3.30

space. The right half of Tab. 4 complements this picture: HOG achieves the largest spatial coverage and the strongest grasp diversity. See Appendix M for details of the metrics.

4.5 Long-Horizon Tracking

HOG synthesizes diverse long-horizon sequences at scale by concatenating meta-skill trajectories. As shown in Tab. 3, the HOT student tracks Grasp–Move–Place sequences zero-shot. For more challenging compositions involving Rotate or Regrasp, we fine-tune the student on HOG-generated composite sequences for only 1,000 epochs. The qualitative results in Fig. 1(b) and Fig. 4 demonstrate tracking of Catch–Rotate–Move and Grasp–Move–Place–Rotate after this brief fine-tuning.

4.6 HOT Generalization

Object Generalization To evaluate object generalization, we distill object-specific teachers into student policies covering Grasp, Move, and Place, and compare training with 1, 5, and 15 objects. As shown in Fig. 8 and Tab. 11, increasing the diversity of training objects improves average tracking performance across test objects, demonstrating the benefit of multi-object training for generalization. Note that these students are trained separately from the HOT (student) evaluated in Sec. 4.2.

Motion Transfer We further test this student on HOG-retargeted and refined GRAB trajectories [Taheri et al., 2020] (Appendix G); Fig. 7 and Table 8 report transfer to these processed human-motion references.

Different hand morphologies. We evaluate the applicability of our framework to Shadow [Developers, 2005], Allegro [Robotics, 2013], Sharpa, and Revo3 by generating HOG trajectories and training a separate HOT policy for each hand. The quantitative results in Tab. 5 and qualitative examples in Fig. 5 show that the same data synthesis and training pipeline supports tracking across diverse hand morphologies.

5 Conclusion

We presented a system for generalizable dexterous manipulation that combines synthetic data generation with tracking-based control. Our approach consists of HOG, a scalable framework for synthesizing hand-object trajectories across diverse meta-skills, and HOT, a tracking policy trained via reinforcement imitation learning on such data. The resulting controller demonstrates effective generalization to unseen objects, long-horizon tasks, and even real-world human-object motion data. We further validate the framework across different hand morphologies and demonstrate cross-simulator transfer. By functioning effectively without human demonstrations while remaining compatible with them, our method offers a flexible, scalable, and powerful foundation for future work on general-purpose dexterous manipulation.

Acknowledgments and Disclosure of Funding

This work was supported by the Research Grants Council of HKSAR (Grant No. AoE/E-601/24-N), the Shenzhen Loop Area Institute (Grant No. 2026NSFP006), and the Hubei Provincial Science and Technology Plan Project (Grant No. 2025BEA002).

The authors declare no competing interests.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Arthur Allshire, Mayank MittaI, Varun Lodaya, Viktor Makoviychuk, Denys Makoviichuk, Felix Widmaier, Manuel Wüthrich, Stefan Bauer, Ankur Handa, and Animesh Garg. Transferring dexterous manipulation from gpu simulation to a remote real-world trifinger. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 11802–11809. IEEE, 2022.

OpenAI: Marcin Andrychowicz, Bowen Baker, Maciek Chociej, Rafal Jozefowicz, Bob McGrew, Jakub Pachocki, Arthur Petron, Matthias Plappert, Glenn Powell, Alex Ray, et al. Learning dexterous in-hand manipulation. *The International Journal of Robotics Research*, 39(1):3–20, 2020.

Prithviraj Banerjee, Sindi Shkodrani, Pierre Moulon, Shreyas Hampali, Fan Zhang, Jade Fountain, Edward Miller, Selen Basol, Richard Newcombe, Robert Wang, et al. Introducing hot3d: An egocentric dataset for 3d hand and object tracking. *arXiv preprint arXiv:2406.09598*, 2024.

Zhe Cao, Ilija Radosavovic, Angjoo Kanazawa, and Jitendra Malik. Reconstructing hand-object interactions in the wild. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12417–12426, 2021.

Junuk Cha, Jihyeon Kim, Jae Shin Yoon, and Seungryul Baek. Text2hoi: Text-guided 3d motion generation for hand-object interaction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1577–1585, 2024.

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

Yuanpei Chen, Yiran Geng, Fangwei Zhong, Jiaming Ji, Jiechuang Jiang, Zongqing Lu, Hao Dong, and Yaodong Yang. Bi-dexhands: Towards human-level bimanual dexterous manipulation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023a.

Yuanpei Chen, Chen Wang, Li Fei-Fei, and C Karen Liu. Sequential dexterity: Chaining dexterous policies for long-horizon manipulation. *arXiv preprint arXiv:2309.00987*, 2023b.

Yuanpei Chen, Chen Wang, Yaodong Yang, and C Karen Liu. Object-centric dexterous manipulation from human motion data. *arXiv preprint arXiv:2411.04005*, 2024.

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

Shadow Robot Developers. Shadow robot. *https://www.shadowrobot.com/dexterous-hand-series/*, 2005.

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

Hao-Shu Fang, Chenxi Wang, Hongjie Fang, Minghao Gou, Jirong Liu, Hengxu Yan, Wenhai Liu, Yichen Xie, and Cewu Lu. Anygrasp: Robust and efficient grasp perception in spatial and temporal domains. *IEEE Transactions on Robotics*, 2023.

Hongming Fu, Wenjia Wang, Xiaozhen Qiao, Rolandos Alexandros Potamias, Taku Komura, Shuo Yang, Zheng Liu, and Bo Zhao. Egograsp: World-space hand-object interaction estimation from egocentric videos. *arXiv preprint arXiv:2601.01050*, 2026.

Zipeng Fu, Qingqing Zhao, Qi Wu, Gordon Wetzstein, and Chelsea Finn. Humanplus: Humanoid shadowing and imitation from humans. *arXiv preprint arXiv:2406.10454*, 2024.

Jensen Gao, Annie Xie, Ted Xiao, Chelsea Finn, and Dorsa Sadigh. Efficient data collection for robotic manipulation via compositional generalization. *arXiv preprint arXiv:2403.05110*, 2024.

Caelan Garrett, Ajay Mandlekar, Bowen Wen, and Dieter Fox. Skillmimicgen: Automated demonstration generation for efficient skill learning and deployment. *arXiv preprint arXiv:2410.18907*, 2024.

Yana Hasson, Gul Varol, Dimitrios Tzionas, Igor Kalevatykh, Michael J Black, Ivan Laptev, and Cordelia Schmid. Learning joint reconstruction of hands and manipulated objects. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11807–11816, 2019.

Tairan He, Zhengyi Luo, Xialin He, Wenli Xiao, Chong Zhang, Weinan Zhang, Kris Kitani, Changliu Liu, and Guanya Shi. Omnih2o: Universal and dexterous human-to-humanoid whole-body teleoperation and learning. *arXiv preprint arXiv:2406.08858*, 2024.

Binghao Huang, Yuanpei Chen, Tianyu Wang, Yuzhe Qin, Yaodong Yang, Nikolay Atanasov, and Xiaolong Wang. Dynamic handover: Throw and catch with bimanual hands. *arXiv preprint arXiv:2309.05655*, 2023.

Zhenyu Jiang, Yuqi Xie, Kevin Lin, Zhenjia Xu, Weikang Wan, Ajay Mandlekar, Linxi Fan, and Yuke Zhu. Dexmimicgen: Automated data generation for bimanual dexterous manipulation via imitation learning. *arXiv preprint arXiv:2410.24185*, 2024.

Jordan Juravsky, Yunrong Guo, Sanja Fidler, and Xue Bin Peng. Superpadl: Scaling language-directed physics-based control with progressive supervised distillation. In *ACM SIGGRAPH 2024 Conference Papers*, pages 1–11, 2024.

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*, pages 6991–7003, 2025a.

Muchen Li, Sammy Christen, Chengde Wan, Yujun Cai, Renjie Liao, Leonid Sigal, and Shugao Ma. Latenthoi: On the generalizable hand object motion generation with latent hand diffusion. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17416–17425. IEEE, 2025b.

Tengyu Liu, Zeyu Liu, Ziyuan Jiao, Yixin Zhu, and Song-Chun Zhu. Synthesizing diverse and physically stable grasps with arbitrary hand structures using differentiable force closure estimator. *IEEE Robotics and Automation Letters*, 7(1):470–477, 2021.

Xueyi Liu, Jianibieke Adalibieke, Qianwei Han, Yuzhe Qin, and Li Yi. Dextrack: Towards generalizable neural tracking control for dexterous manipulation from human references. *arXiv preprint arXiv:2502.09614*, 2025.

Zhengyi Luo, Jinkun Cao, Kris Kitani, Weipeng Xu, et al. Perpetual humanoid control for real-time simulated avatars. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10895–10904, 2023.

Viktor Makoviychuk, Lukasz Wawrzyniak, Yunrong Guo, Michelle Lu, Kier Storey, Miles Macklin, David Hoeller, Nikita Rudin, Arthur Allshire, Ankur Handa, and Gavriel State. Isaac gym: High performance GPU based physics simulation for robot learning. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021. URL `https://openreview.net/forum?id=fgFBtYgJQX_`.

Ajay Mandlekar, Soroush Nasiriany, Bowen Wen, Iretiayo Akinola, Yashraj Narang, Linxi Fan, Yuke Zhu, and Dieter Fox. Mimicgen: A data generation system for scalable robot learning using human demonstrations. *arXiv preprint arXiv:2310.17596*, 2023.

Andrew T Miller and Peter K Allen. Graspit! a versatile simulator for robotic grasping. *IEEE Robotics & Automation Magazine*, 11(4):110–122, 2004.

Xue Bin Peng, Marcin Andrychowicz, Wojciech Zaremba, and Pieter Abbeel. Sim-to-real transfer of robotic control with dynamics randomization. In *2018 IEEE international conference on robotics and automation (ICRA)*, pages 3803–3810. IEEE, 2018.

Wilbert Pumacay, Ishika Singh, Jiafei Duan, Ranjay Krishna, Jesse Thomason, and Dieter Fox. The colosseum: A benchmark for evaluating generalization for robotic manipulation. *arXiv preprint arXiv:2402.08191*, 2024.

Aravind Rajeswaran, Vikash Kumar, Abhishek Gupta, Giulia Vezzani, John Schulman, Emanuel Todorov, and Sergey Levine. Learning complex dexterous manipulation with deep reinforcement learning and demonstrations. *Robotics: Science and Systems XIV*, 2018.

Wonik Robotics. Allegro hand. *https://www.allegrohand.com/*, 2013.

Javier Romero, Dimitrios Tzionas, and Michael J Black. Embodied hands: Modeling and capturing hands and bodies together. *arXiv preprint arXiv:2201.02610*, 2022.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Qijin She, Ruizhen Hu, Juzhan Xu, Min Liu, Kai Xu, and Hui Huang. Learning high-dof reaching-and-grasping via dynamic representation of gripper-object interaction. *arXiv preprint arXiv:2204.13998*, 2022.

Qijin She, Shishun Zhang, Yunfan Ye, Ruizhen Hu, and Kai Xu. Learning cross-hand policies of high-dof reaching and grasping. In *European Conference on Computer Vision*, pages 269–285. Springer, 2024.

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. URL `https://grab.is.tue.mpg.de`.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 23–30. IEEE, 2017.

Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pages 5026–5033. IEEE, 2012.

Weikang Wan, Haoran Geng, Yun Liu, Zikang Shan, Yaodong Yang, Li Yi, and He Wang. Unidexgrasp++: Improving dexterous grasping policy learning via geometry-aware curriculum and iterative generalist-specialist learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3891–3902, 2023.

Ruicheng Wang, Jialiang Zhang, Jiayi Chen, Yinzhen Xu, Puhao Li, Tengyu Liu, and He Wang. Dexgraspnet: A large-scale robotic dexterous grasp dataset for general objects based on simulation. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11359–11366. IEEE, 2023a.

Yinhuai Wang, Jing Lin, Ailing Zeng, Zhengyi Luo, Jian Zhang, and Lei Zhang. Physhoi: Physics-based imitation of dynamic human-object interaction. *arXiv preprint arXiv:2312.04393*, 2023b.

Yinhuai Wang, Qihan Zhao, Runyi Yu, Hok Wai Tsui, Ailing Zeng, Jing Lin, Zhengyi Luo, Jiwen Yu, Xiu Li, Qifeng Chen, et al. Skillmimic: Learning reusable basketball skills from demonstrations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

Bowen Wen, Jonathan Tremblay, Valts Blukis, Stephen Tyree, Thomas Müller, Alex Evans, Dieter Fox, Jan Kautz, and Stan Birchfield. Bundlesdf: Neural 6-dof tracking and 3d reconstruction of unknown objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 606–617, 2023.

Sirui Xu, Hung Yu Ling, Yu-Xiong Wang, and Liang-Yan Gui. Intermimic: Towards universal whole-body control for physics-based human-object interactions. *arXiv preprint arXiv:2502.20390*, 2025.

Yinzhen Xu, Weikang Wan, Jialiang Zhang, Haoran Liu, Zikang Shan, Hao Shen, Ruicheng Wang, Haoran Geng, Yijia Weng, Jiayi Chen, et al. Unidexgrasp: Universal robotic dexterous grasping via learning diverse proposal generation and goal-conditioned policy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4737–4746, 2023.

Ruihan Yang, Huazhe Xu, Yi Wu, and Xiaolong Wang. Multi-task reinforcement learning with soft modularization. *Advances in Neural Information Processing Systems*, 33:4767–4777, 2020.

Zhao-Heng Yin, Binghao Huang, Yuzhe Qin, Qifeng Chen, and Xiaolong Wang. Rotating without seeing: Towards in-hand dexterity through touch. *arXiv preprint arXiv:2303.10880*, 2023.

Zhao-Heng Yin, Changhao Wang, Luis Pineda, Francois Hogan, Krishna Bodduluri, Akash Sharma, Patrick Lancaster, Ishita Prasad, Mrinal Kalakrishnan, Jitendra Malik, et al. Dexteritygen: Foundation controller for unprecedented dexterity. *arXiv preprint arXiv:2502.04307*, 2025.

Runyi Yu, Yinhuai Wang, Qihan Zhao, Hok Wai Tsui, Jingbo Wang, Ping Tan, and Qifeng Chen. Skillmimic-v2: Learning robust and generalizable interaction skills from sparse and noisy demonstrations. *arXiv preprint arXiv:2505.02094*, 2025.

Hui Zhang, Sammy Christen, Zicong Fan, Otmar Hilliges, and Jie Song. Graspxl: Generating grasping motions for diverse objects at scale. In *European Conference on Computer Vision*, pages 386–403. Springer, 2024a.

Hui Zhang, Sammy Christen, Zicong Fan, Luocheng Zheng, Jemin Hwangbo, Jie Song, and Otmar Hilliges. Artigrasp: Physically plausible synthesis of bi-manual dexterous grasping and articulation. In *2024 International Conference on 3D Vision (3DV)*, pages 235–246. IEEE, 2024b.

A Additional Experiments

A.1 Impact of Data Volume and Model Design

We investigate the impact of three key design choices in our approach through comprehensive ablation studies. First, we examine critical HOT design elements, specifically the localization and residualization of input-output representations that simplify the learning distribution (HOT w/o LR). Second, we study the effectiveness of our data augmentation strategies (HOT w/o DA). Third, we analyze how model performance scales with training data quantity.

The experiments are conducted on three fundamental meta-skills—Grasp, Move, and Place—with each skill trained using 10, 100, or 1000 demonstrations. Tab. 6 presents the quantitative results of our ablation studies. The localization and residualization of the input-output representations of HOT prove crucial for generalization, allowing the model to achieve superior performance even with limited training data. This validates our hypothesis that simplifying the input-output distribution enhances learning efficiency. Data augmentation techniques consistently boost generalization performance across all skills, while increased training data leads to proportional performance improvements, demonstrating our method's ability to effectively utilize additional demonstrations.

Table 6: Ablation study of model design and data volume.

Method	*10 Clips*			*100 Clips*			*1000 Clips*		
	HOT	*HOT w/o LR*	*HOT w/o DA*	*HOT*	*HOT w/o LR*	*HOT w/o DA*	*HOT*	*HOT w/o LR*	*HOT w/o DA*
Grasp	31.49%	19.41%	31.60%	73.24%	71.38%	60.22%	94.44%	64.97%	70.79%
Move	79.85%	62.04%	79.75%	97.67%	94.66%	90.89%	99.14%	95.18%	95.20%
Place	84.73%	28.12%	84.99%	100%	92.29%	100%	98.18%	100%	99.63%
Avg.	65.36%	36.52%	65.45%	90.30%	86.11%	83.70%	97.25%	86.72%	88.54%

A.2 Refining HOI Data Through Tracking

To quantitatively demonstrate HOT's ability to refine imperfect reference data through tracking, we evaluate the trained HOT on bottle Grasp sequences from the test set generated by HOG, using object-hand penetration as the key metric. As shown in Tab. 7, the original reference data contain noticeable penetration, while the trajectories produced by HOT exhibit significantly reduced penetration, validating its capacity to correct physically implausible artifacts. Figure 6 provides qualitative examples of refining imperfect synthetic demonstrations for sword grasp and bottle regrasp.

Table 7: Hand–object penetration before and after HOT tracking. Penetration depths are reported in mm; lower is better.

Trajectory	**Mean Depth** ↓	**Max Depth** ↓
HOG reference	2.3 ± 1.3	9.2 ± 7.1
HOT rollout	$\mathbf{1.2 \pm 0.8}$	$\mathbf{2.8 \pm 1.3}$

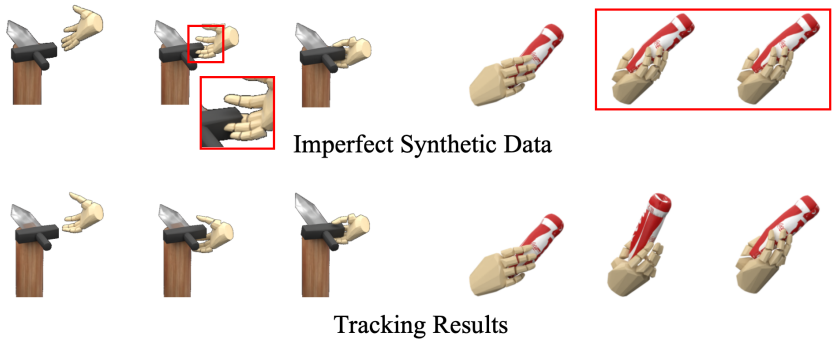


Figure 6: HOT is able to track and refine imperfect synthetic HOI demonstrations.

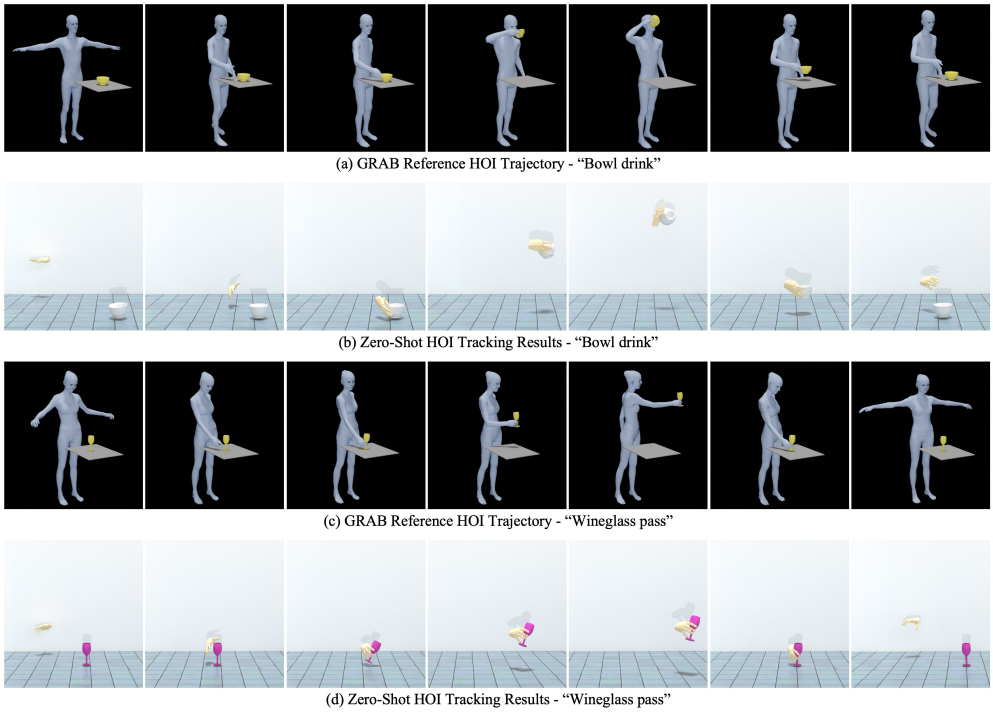


Figure 7: Zero-shot HOI tracking on GRAB data.

Table 8: HOI tracking performance on GRAB. $\mathrm{SR}_{\mathrm{grasp}}$ and $\mathrm{SR}_{\mathrm{traj}}$ report successful trials / total trials for stable grasping and full-trajectory tracking, respectively. Overall sums trial counts and averages errors.

Dataset	$\mathrm{SR}_{\mathrm{grasp}} \uparrow$	$\mathrm{SR}_{\mathrm{traj}} \uparrow$	$E_{op} \downarrow$	$E_{or} \downarrow$	$E_k \downarrow$
GRAB-Apple	3/3	3/3	1.64	46.43	4.82
GRAB-Bowl	3/3	3/3	2.59	49.22	5.03
GRAB-Cup	3/3	1/3	2.58	15.48	9.56
GRAB-Hammer	2/3	2/3	5.89	36.36	5.75
GRAB-Stanfordbunny	1/3	1/3	9.51	68.07	5.02
GRAB-Stapler	2/3	2/3	4.26	29.59	5.62
GRAB-Train	3/3	2/3	12.81	42.74	6.47
GRAB-Torus_large	1/2	0/2	24.77	89.07	11.71
GRAB-Waterbottle	3/4	3/4	4.38	19.40	4.33
GRAB-Wineglass	2/4	1/4	38.13	44.55	6.10
Overall	**23/31**	**18/31**	**10.66**	**44.09**	**6.44**

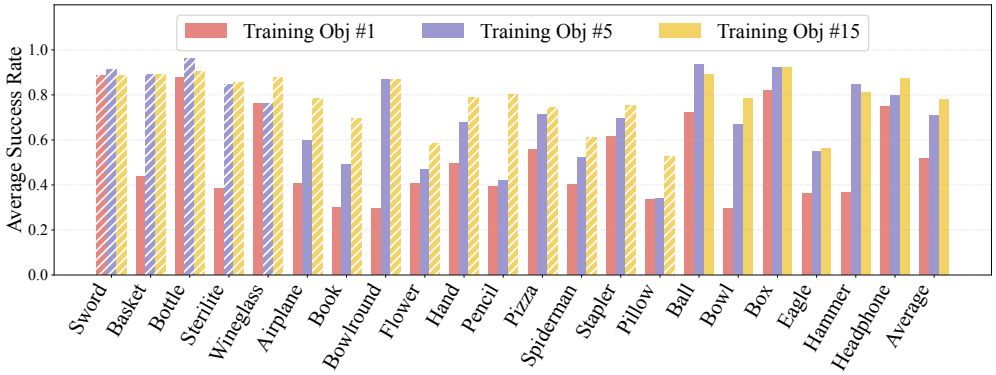


Figure 8: Evaluation of HOT's object generalization performance. The dashed bars denote objects present in the training set, while the undashed bars represent objects unseen during training.

B Detailed Sub-Rewards

The unified imitation reward combines complementary components multiplicatively:

$$r_t = r_t^{hand} r_t^{wrist} r_t^{object} r_t^{interact} r_t^{contact}, \tag{7}$$

Hand Tracking Reward:

$$r_t^{hand} = r_t^p r_t^r, \tag{8}$$

where r_t^p denotes the finger joint position reward and r_t^r represents the finger joint rotation reward.

Wrist Tracking Reward:

$$r_t^{wrist} = r_t^{wp} r_t^{wr}, \tag{9}$$

where r_t^{wp} denotes the wrist joint position reward and r_t^{wr} represents the wrist joint rotation reward.

Object Reward:

$$r_t^{object} = r_t^{op} r_t^{or}, \tag{10}$$

where r_t^{op} and r_t^{or} denote the global object position and rotation rewards, respectively.

Interaction Reward: $r_t^{interact}$ measures relative hand-object motion by comparing vectors from hand keypoints to object keypoints in the reference data with the corresponding vectors in the simulated environment. This consistency is essential for interaction stability.

Contact Reward: $r_t^{contact}$ is the contact reward guided by frame-wise binary labels of hand-object contact from reference data, which is crucial for reliable grasping and releasing operations.

Formulation of Sub-Rewards: Each subterm is defined as:

$$r_t^{\alpha} = \exp(-\lambda^{\alpha} e_{t+1}^{\alpha}), \tag{11}$$

where $\alpha \in \{p, r, wp, wr, op, or, interact, contact\}$, and e_{t+1}^{α} and λ^{α} denote the tracking error and reward weight, respectively. All reward weights are fixed during training, with $\lambda^{interact} = 20$. The values are listed in Table 12.

C Domain Randomization Curriculum

We apply several domain randomization techniques [Tobin et al., 2017, Peng et al., 2018, Yu et al., 2025] to enhance policy robustness. During training, we independently randomize three factors: (1) random perturbation forces applied to the object when Move data are sampled; (2) object physical properties, including geometry scale, density, restitution, and friction coefficients; and (3) perturbations to the initial hand-object state. We use a two-stage curriculum with fixed randomization settings within each stage, switching from weaker to stronger randomization at the stage boundary.

D Domain Randomization Settings

We employ domain randomization along three main axes: object scale, initial state, and a curriculum on observation/action noise.

Object scale randomization. During training, the object size is randomly perturbed to expose the policy to a range of geometries. At each environment reset, with probability 0.2 we uniformly sample a scale factor

$$s \sim \mathcal{U}(0.75, 1.5), \tag{12}$$

and uniformly scale the object by s. Otherwise, the object keeps its nominal size. This yields objects between 75% and 150% of the original scale across training episodes, resulting in improved generalization to geometric variations.

Random initial state. Beyond sampling a random reference frame from the motion dataset, we additionally inject controlled perturbations into the initial hand-object configuration to further enlarge the distribution of starting states. After selecting a trajectory and a time index, we perturb both the object pose and the hand configuration using zero-mean bounded noise in a phase-dependent manner. For the object, let $\mathbf{p}^o$ and R^o denote the sampled position and orientation. We add a stochastic positional offset

$$\mathbf{p}^o \leftarrow \mathbf{p}^o + \boldsymbol{\epsilon}_p, \tag{13}$$

where the magnitude of $\boldsymbol{\epsilon}_p$ is larger for non-contact frames, encouraging wider spatial variability, and smaller for contact frames to preserve grasp stability. Rotational perturbations follow the same principle: non-contact frames receive full 3D angular noise up to a bounded angle, whereas contact frames use restricted in-plane perturbations to avoid breaking established grasps. For the hand, let $\mathbf{q}$ and $\dot{\mathbf{q}}$ denote joint positions and velocities. Finger-joint angles are perturbed as

$$\mathbf{q} \leftarrow \mathbf{q} + \boldsymbol{\epsilon}_q, \tag{14}$$

where $\boldsymbol{\epsilon}_q$ is sampled within a bounded angular interval to encourage variability in finger postures, and joint velocities receive mild perturbations to model natural initialization variability. For tasks requiring substantial wrist reconfiguration, we also apply small perturbations to the wrist's 6-DoF parameters, providing diverse but physically coherent wrist poses. All perturbations are applied stochastically and independently per environment, ensuring a broad yet physically valid distribution of initial states. Combined with random reference state sampling, this strategy enhances robustness to variations in contact configuration, object placement, and hand posture at the start of each episode.

Randomization curriculum. We use a two-stage curriculum with fixed settings within each stage. In the first stage, object-scale randomization is disabled, and initial-state perturbations use a low probability and small magnitude. In the second stage, object-scale randomization is enabled, and both the probability and magnitude of initial-state perturbations switch to higher fixed values.

E Adaptive Sampling

To enhance performance on challenging examples, we dynamically adjust sampling probabilities according to sample difficulty [Yu et al., 2025]. Specifically, for a trajectory $\tau_i = \{\hat{\boldsymbol{s}}_1, ..., \hat{\boldsymbol{s}}_T\}$, we calculate its mean reward $\bar{r}_i = \frac{1}{T}\sum_{t=1}^{T} r_t$ as a metric of learning difficulty, where r_t is defined in Eq. 7. Based on this difficulty assessment, we assign a sampling probability p_i for trajectory τ_i using:

$$p_i = \frac{e^{-\lambda_s \bar{r}_i}}{\sum_{j=0}^{N-1} e^{-\lambda_s \bar{r}_j}} \tag{15}$$

where N denotes the dataset size, and $\lambda_s \in [0, \infty)$ controls the balance between uniform sampling ($\lambda_s = 0$) and difficulty-prioritized sampling ($\lambda_s > 0$).

F Implementation Details of Baseline Methods

Because public implementations of SkillMimic [Wang et al., 2025] and DexGen [Yin et al., 2025] do not directly support tracking with the MANO hand model, we construct comparable variants by modifying the reward, input representation, and output format of HOT following the design choices of these methods, as summarized in Tab. 9. These SkillMimic-style and DexGen-var variants use the same HOG data and training configurations as the specialist HOT policies in Table 2.

DexTrack-style baseline. For the unified benchmark, we adapt the action representation and homotopy optimization of DexTrack [Liu et al., 2025] within our codebase. The baseline uses the same observations and rewards as HOT, with residual actions for all joints and the same initial HOG trajectories. Data augmentation and reweighting are disabled. Training proceeds through three stages of 20,000 epochs each. Whenever a reference trajectory is successfully tracked, its simulated rollout replaces the corresponding HOG reference for the next stage. We train separate policies for each skill and object, as for the HOT teachers. This adaptation is labeled DexTrack-style rather than an evaluation of the original implementation. Its total training budget is 60,000 epochs, compared with 30,000 epochs for each HOT teacher; the data-controlled comparison is not compute-matched.

Table 9: Implementation details of the compared tracking baselines.

Feature	HOT	SkillMimic	DexGen-var	DexTrack-style
Observation	HOT default	No object keypoints	No object information	HOT default
Reward	HOT default	HOT default	Object pose & finger DoF only	HOT default
Action	Direct finger & residual wrist	Direct action	Direct action	Residual actions for all joints
Data augmentation	Yes	No	No	No
Reweighting	Yes	No	No	No
Homotopy optimization	No	No	No	3 rounds

G HOG for Other Meta-Skills

Free Move. Free Move is an auxiliary non-contact skill for teaching the hand to navigate freely in space. It is not included in the seven interaction-skill evaluations in the main seven-skill benchmark, which focus on hand-object interaction. To generate Free Move data, we randomly select two poses in the operational space as the wrist's initial and final poses. We then randomly generate two sets of finger configurations to form complete initial and final hand states. The trajectory is obtained through interpolation between these states.

Catch. Unlike Grasp, the Catch skill involves catching moving objects. To synthesize these demonstrations, we first randomly select a grasp configuration $\boldsymbol{g}$ from $\mathcal{G}$ and generate a parabolic trajectory for the object $[\mathbf{T}^o(t), \mathbf{R}^o(t)] = \text{Parabola}(t), t \in [0, T]$. We then select the last frame as the grasp moment and transform $\boldsymbol{g}$ to align with the object pose $[\mathbf{T}^o(T), \mathbf{R}^o(T)]$, yielding $\boldsymbol{g}^{end}$. The initial hand position and orientation are set to be identical to those in $\boldsymbol{g}^{end}$, while the initial joint angles are set to zero. The hand trajectory is then generated via interpolation between the initial and end hand states. The complete Catch demonstration consists of both the generated hand trajectory and the previously computed parabolic object trajectory. Demonstrations with significant hand-object collisions are filtered out.

Throw. Throw demonstrations, which involve throwing objects with a specified velocity, are generated by time-reversing Catch trajectories. The parabolic motion of a thrown object remains physically reasonable due to the symmetric structure of the trajectory.

General Rotate. In addition to the simplified Rotate data generation method described in the main text, we also propose a more general approach for generating rotation data. This method produces more challenging data that require longer training times. Specifically, we first select a grasp configuration $\boldsymbol{g}^1$ from $\mathcal{G}$ as the initial state and remove it from $\mathcal{G}$. We then iteratively find the closest grasp configuration in the remaining set: for each $\boldsymbol{g}^i$, we search for its nearest grasp configuration $\boldsymbol{g}^{i+1}$ and remove $\boldsymbol{g}^{i+1}$ from $\mathcal{G}$, continuing this process for k steps. We transform all grasp configurations to align their wrist poses with that of $\boldsymbol{g}^1$. To provide sufficient buffer time to learn such large transitions, we replicate each $\boldsymbol{g}^i$ for N frames.

Regrasp. We define Regrasp as a skill that differs from Rotate in that it maintains a fixed object pose while transitioning between different hand configurations. The generation methodology mirrors that of Rotate, with the key distinction being that the grasp configurations are aligned to the same object pose rather than the same wrist pose.

Retargeting GRAB for Evaluation. We validated the system's ability to track real-world HOI trajectories using sequences from the GRAB dataset [Taheri et al., 2020]. These were segmented into individual grasp-move-place episodes. An initial hand trajectory was obtained by retargeting based on the ManipTrans [Li et al., 2025a] method, providing preliminary wrist and joint configurations. We further employed HOG to synthesize an improved grasp pose by fixing the object and wrist rotations from the initial grasp frame and optimizing the hand configuration toward higher grasp quality and physical plausibility while preserving kinematic similarity. The original finger and object data in



Figure 9: Tracking a Regrasp trajectory.

each trajectory were replaced with this refined pose, yielding a corrected GRAB dataset that serves as a high-quality benchmark for evaluation.

H LLM-based Trajectory Planning

We employ an LLM to generate semantically annotated keyframes and subsequently apply keyframe-guided interpolation in both position and orientation. This pipeline ensures that the generated trajectories maintain both semantic coherence and dynamic feasibility, making them directly applicable to our tracking policy for practical manipulation tasks.

LLM Prompt and Keypoint Specification. Given a high-level manipulation command (e.g., "set a lying bottle upright"), we construct a structured prompt that includes the task description, the initial wrist and object poses in a fixed robot base frame, the coordinate system definition, rotation conventions in XYZ Euler angles and quaternions, and smoothness constraints on positional and rotational changes. The prompt also provides a full desktop configuration listing all objects together with their feasible candidate grasp configurations, each associated with a valid wrist pose. The LLM is explicitly required to *select one* candidate grasp and align the generated grasp keypoint to that wrist pose.

The LLM outputs a JSON object containing keypoints and metadata. Each keypoint specifies a unique index, wrist position $\mathbf{p}_i$, orientation q_i, and an action label. Keypoints with *grasp* and *release* actions must be designated, with their indices explicitly provided in top-level fields. These frames define interpolation segments, while start/end frames are derived from sequence boundaries. Other keypoints serve as internal anchors.

Keyframe-Based Piecewise Bezier Interpolation. The four special frame types naturally partition the LLM keypoints into three semantic segments: start $\rightarrow$ grasp, grasp $\rightarrow$ release, and release $\rightarrow$ end. Within each segment, all keypoints (including intermediate approach/transport points) are used as interpolation anchors, while different segments are interpolated independently. For each segment, we construct a cubic Bezier curve for the wrist position using the segment endpoints as Bezier endpoints and generating two inner control points from finite differences of neighboring anchors, so that the curve tangents align with the discrete motion directions proposed by the LLM. We then sample a fixed number of trajectory points per LLM keypoint and concatenate all segments into a dense position trajectory, removing duplicated boundary samples. This piecewise scheme ensures smooth motion inside each phase while strictly preserving the grasp and release boundaries (e.g., if the LLM predicts indices 3 and 8 as grasp and release, we only interpolate within [1,3], [3,8], and [8,9]).

Orientation Interpolation. Wrist orientations are interpolated independently of positions. We first convert all keyframe orientations to quaternions, enforce sign continuity along the sequence to avoid double-cover flipping, and then apply spherical linear interpolation (SLERP) within each segment using the same temporal parameterization as for the position Bezier curves. Segments with negligible rotation are treated as static segments, for which the orientation is kept fixed throughout the segment.

The trajectories generated by both the LLM and Text2HOI [Cha et al., 2024] are subsequently processed by HOG, which replaces the original finger degrees of freedom with synthesized grasp poses.

Table 10: Additional four-skill distillation experiment with its own student, distinct from the seven-skill and 15-object students.

Skill	Method	Train				Test			
		SR (%) ↑	E_{op} (cm) ↓	E_{or} (°) ↓	E_h (cm) ↓	SR (%) ↑	E_{op} (cm) ↓	E_{or} (°) ↓	E_h (cm) ↓
Grasp	Teacher	98.19	1.03	6.20	3.74	73.89	3.21	22.93	5.66
	Naive Multi-Task	66.01	4.57	16.61	7.82	68.38	5.32	17.39	7.59
	Distilled Student	97.88	0.86	4.85	3.06	79.72	2.50	20.76	4.81
Move	Teacher	99.29	2.31	10.57	4.12	99.14	2.50	10.85	5.16
	Naive Multi-Task	81.40	4.57	16.61	7.82	89.45	3.17	11.76	5.56
	Distilled Student	98.25	1.97	10.45	4.43	97.71	2.27	10.68	3.95
Place	Teacher	98.41	1.04	6.02	3.04	98.13	1.09	8.42	2.85
	Naive Multi-Task	82.63	3.49	29.02	5.77	84.88	3.47	27.83	5.84
	Distilled Student	97.29	1.15	7.39	3.03	97.32	1.04	8.67	2.66
Rotate	Teacher	97.30	0.98	7.75	1.71	56.23	2.38	44.96	2.57
	Naive Multi-Task	2.42	2.27	27.67	4.75	0.39	3.00	27.97	5.06
	Distilled Student	83.76	1.49	11.23	2.28	45.44	2.42	56.61	2.71

I Policy Distillation

Distinct policy settings. The main seven-skill benchmark uses one student distilled across Grasp, Move, Place, Throw, Catch, Rotate, and Regrasp on four objects. The object-generalization experiment uses a separately trained student distilled from object-specific teachers on Grasp, Move, and Place across 15 objects; this is the student used for the 67.27% frame-wise SR on unseen Grasp–Move–Place compositions. The four-skill experiment in Table 10 is an additional distillation ablation, with its own student. These settings do not share a single student checkpoint.

To address skill interference in multi-task and multi-object learning, we implement a progressive knowledge distillation framework that transfers expertise from specialized teacher policies to a unified student policy. Inspired by previous work [Luo et al., 2023, Xu et al., 2025], we employ a four-stage training strategy: **Stage I** (0–500 epochs) uses the DAgger algorithm for behavioral cloning, where actions are sampled entirely from teacher policies and the student policy is updated using an expert supervision loss (ELoss). **Stage II** (500–5000 epochs) introduces hybrid decision-making through expert-mode masking, with the teacher action sampling probability linearly decaying from 100% to 0% to enable a smooth transition to autonomous policy execution. **Stage III** (5000–7000 epochs) focuses on value function optimization, progressively incorporating the policy gradient loss when explained variance (EV) consistently exceeds 0.6 for three consecutive evaluation windows. **Stage IV** (7000+ epochs) combines policy gradient, value function, boundary constraint, and residual expert supervision losses, with dynamic weights determined by training progress and critic performance.

To evaluate the effectiveness of policy distillation, we compare the performance of teacher policies, the distilled student policy, and a naively trained multi-task policy in an additional four-skill tracking experiment covering Grasp, Move, Place, and Rotate. As shown in Tab. 10, the distilled student policy mitigates skill interference commonly observed in multi-task learning, raising Rotate test SR from 0.39% to 45.44% (a gain of 45.05 percentage points). Moreover, the distillation process reduces the required training cycles by approximately 45% while achieving higher final performance compared to naive multi-task training.

To evaluate how distillation improves generalization across objects, we conduct experiments on the Grasp, Move, and Place skill data under several training settings: a student policy distilled without RL from 15 objects; a student policy using both distillation and RL with 15 objects; a Distill&RL policy trained on 5 objects; and a policy trained on a single object (Sword). As shown in Tab. 11, the Distill&RL approach yields clear gains over distillation alone. Moreover, performance improves consistently as the number of training objects increases, confirming the benefit of multi-object training for generalization.

Table 11: Multi-object distillation on Grasp, Move, and Place. These students are trained separately from the seven-skill student.

Object	Distillation Only (15 Objects)	Distillation + RL (15 Objects)	Distillation + RL (5 Objects)	Sword Teacher (1 Object)
	Grasp/Move/Place SR ↑	Grasp/Move/Place SR ↑	Grasp/Move/Place SR ↑	Grasp/Move/Place SR ↑
Airplane	39% / 60% / 50%	**62%** / **84%** / **89%**	59% / 66% / 54%	44% / 46% / 32%
Ball	60% / 91% / 90%	79% / 88% / **100%**	**96%** / 85% / **100%**	46% / **99%** / 72%
Basket	56% / 91% / **99%**	76% / **98%** / 93%	**87%** / 82% / 98%	38% / 57% / 36%
Book	14% / 59% / 41%	**51%** / **83%** / **74%**	33% / 53% / 61%	15% / 45% / 30%
Bottle	65% / 94% / 96%	78% / 95% / 98%	**89%** / **100%** / **100%**	75% / 95% / 93%
Bowl	38% / 78% / 74%	**76%** / **79%** / **80%**	62% / 76% / 62%	22% / 56% / 11%
Bowlround	64% / 88% / 84%	**86%** / 89% / **85%**	84% / **91%** / **85%**	38% / 43% / 8%
Box	55% / 97% / 95%	**78%** / 98% / **100%**	77% / **100%** / **100%**	57% / **100%** / 89%
Eagle	28% / **67%** / 42%	45% / 51% / **73%**	**46%** / **67%** / 52%	28% / 53% / 27%
Flower	22% / 62% / 53%	**44%** / **65%** / **66%**	35% / 56% / 49%	26% / 58% / 38%
Hammer	20% / 88% / 20%	67% / 85% / 91%	**70%** / **90%** / **94%**	17% / 87% / 6%
Hand	37% / 81% / 73%	**66%** / **83%** / **87%**	62% / 74% / 67%	34% / 64% / 51%
Headphones	53% / 93% / 89%	**71%** / **97%** / **94%**	68% / 87% / 84%	64% / 86% / 75%
Pencil	37% / 67% / 79%	**67%** / **94%** / **80%**	37% / 58% / 31%	23% / 50% / 45%
Pillow	11% / 52% / 44%	**30%** / **61%** / **67%**	14% / 45% / 43%	9% / 42% / 50%
Pizza	**48%** / 70% / 75%	44% / **100%** / 79%	46% / 74% / **94%**	**48%** / 54% / 65%
Spidermanfigure	31% / 69% / 45%	**59%** / 53% / **71%**	40% / **70%** / 46%	29% / 61% / 30%
Stapler	37% / 78% / **80%**	**62%** / **86%** / 78%	60% / 72% / 76%	41% / 72% / 71%
Sterilite	46% / 91% / 84%	73% / 88% / **95%**	**79%** / **94%** / 81%	35% / 65% / 15%
Sword	71% / 96% / 94%	68% / **100%** / 98%	76% / 98% / **100%**	**79%** / 99% / 88%
Wineglass	65% / 74% / 82%	**90%** / **89%** / **84%**	80% / 71% / 78%	70% / 80% / 79%
Average	43% / 78% / 71%	**65%** / **84%** / **85%**	62% / 77% / 74%	40% / 67% / 48%



J Hyperparameters

We summarize the simulation parameters, training hyperparameters, reward coefficients, and data augmentation settings in Tab. 12.

Table 12: Hyperparameters for policy training, imitation rewards, data augmentation, and adaptive sampling. For entries with two values, the first applies to Regrasp and the second to all other skills. Joint and object disturbances are specified by their amplitudes.

Training

Parameter	Value
Action variance $\Sigma_{\boldsymbol{\pi}}$	0.055
Samples per update	65,536
Policy/value minibatch size	16,384
Discount γ	0.99
Adam learning rate	2×10^{-5}
GAE(λ)	0.95
TD(λ)	0.95
PPO clip threshold	0.2
Episode length	60

Reward Coefficients

Term	Weight
Position λ^{p}	$200/20$
Rotation λ^{r}	$200/20$
Object position λ^{op}	50
Object rotation λ^{or}	50
Interaction $\lambda^{interact}$	20
Contact $\lambda^{contact}$	5
Wrist position λ^{wp}	20
Wrist rotation λ^{wr}	20

Augmentation & Sampling

Parameter	Value
Initial disturbance probability	$0.5/0.3$
Joint position disturbance	$\pi/8$
Joint velocity disturbance	0.1
Object velocity disturbance	0.02
Object rotation disturbance	0.02
Reweighting coefficient λ^{s}	10

K Following Text Instructions

Our framework supports manipulation from natural language instructions through multiple complementary approaches. The typical pipeline uses a Vision-Language Model (VLM) for high-level semantic planning. The VLM takes the following inputs: the object's initial 6D pose, a set of feasible wrist poses corresponding to stable grasp configurations, the initial wrist pose, the target object position, an image of the current environment, and a natural-language task description. It outputs a structured plan containing (1) sparse 6D waypoints for the wrist at key states (start, grasp, release, end), (2) a selected grasp pose, and (3) binary flags indicating grasp and release events. We also explore an alternative that uses Text2HOI [Cha et al., 2024] to generate wrist trajectories directly from language. In both cases, we apply trajectory smoothing and Bezier interpolation between keyframes to ensure smooth motions. The resulting wrist path is combined with the chosen grasp pose using the procedure in Sec. 3.1.3 of the main text to form a complete hand-object demonstration. This is

then tracked by our distilled HOT student, demonstrating end-to-end execution of language-guided manipulation. Fig. 1 in the main text shows representative results.

L Unified Seven-Skill Benchmark

Protocol and policy scope. The unified benchmark uses the MANO hand and four objects: a box, a bottle, a sword, and a hammer. For each skill and object, DexTrack-style and HOT teacher results come from separately trained specialist policies. The seven-skill HOT student is one policy shared across all seven skills and four objects. Each object–skill setting is evaluated using five random seeds with 1,000 trials per seed. Table 2 reports the mean and standard deviation from these evaluations, with results averaged over the four objects. The reported variability concerns evaluation seeds, not independent training runs.

Reward trade-offs. Contact supervision is especially important for Grasp, Throw, and Catch, while interaction supervision has a large effect on Rotate and Regrasp. Removing either term can improve Place SR, so these ablations support skill-dependent benefits rather than uniform gains. These are teacher-policy ablations, not student ablations.

M Evaluation of HOI Data Quality and Coverage

M.1 Evaluation Data

We evaluate data quality and coverage on two object categories: bowls and cylinders. Each method contributes 20 hand-object interaction trajectories. EgoGrasp trajectories are reconstructed from captured footage; LatentHOI [Li et al., 2025b] provides 10 generated trajectories per object category; GRAB trajectories are selected from the original dataset; and DexMimicGen and HOG trajectories are randomly sampled from their respective outputs.

M.2 Data Quality Metrics

Penetration. Penetration measures the average depth of intersection between the hand and object meshes, reported in millimeters. Lower values indicate less interpenetration.

Jittering. Jittering measures high-frequency pose oscillations, reported in millimeters. Lower values indicate smoother motion.

Grasp Success Rate (GSR). GSR is the percentage of simulated hold tests completed without dropping the object:

$$\mathrm{GSR} = \frac{N_{\text{successful holds}}}{N_{\text{hold tests}}} \times 100\%. \tag{16}$$

M.3 Data Coverage Metrics

Spatial Coverage. The State-Space Coverage column in Table 4 measures spatial coverage using the volume of the three-dimensional axis-aligned bounding box enclosing all wrist positions across the evaluated trajectories. Let $\mathcal{P}$ denote these wrist positions. The coverage is

$$C_{\text{spatial}} = \prod_{a \in \{x,y,z\}} \left(\max_{\boldsymbol{p} \in \mathcal{P}} p_a - \min_{\boldsymbol{p} \in \mathcal{P}} p_a \right). \tag{17}$$

Larger values indicate that the trajectories explore a wider workspace.

Grasp Diversity. Grasp diversity is the average k-nearest-neighbor distance between grasp descriptors expressed in the object coordinate frame. Each descriptor includes the grasp translation, rotation, and hand joint configuration. For descriptors $\{\boldsymbol{g}_i\}_{i=1}^{N}$, the score is

$$D_{\text{grasp}} = \frac{1}{Nk} \sum_{i=1}^{N} \sum_{j \in \mathcal{N}_k(i)} d(\boldsymbol{g}_i, \boldsymbol{g}_j), \tag{18}$$

where $\mathcal{N}_k(i)$ contains the k nearest descriptors other than $\boldsymbol{g}_i$, and d denotes the descriptor distance. Larger scores indicate more diverse grasps.

N Limitations and Discussion

While our method presents a promising path to addressing the data bottleneck in dexterous manipulation, several limitations remain. First, although HOT provides a general-purpose low-level control capability for trajectory tracking, and we have demonstrated its integration with high-level planners such as LLMs [Achiam et al., 2023] and Text2HOI [Cha et al., 2024], the current high-level planners operate in an open-loop manner. They do not yet support the kind of long-horizon, closed-loop task execution provided by an LLM-based agent. For example, when an object is dropped, the system lacks the capability to autonomously regenerate a recovery trajectory. Achieving such robustness requires both more adaptive tracking policies and more reactive, long-horizon planners.

Second, the unified seven-skill student is evaluated on four objects and loses performance relative to specialist teachers on six of the seven skills. Our 15-object student is a separately trained model covering only Grasp, Move, and Place. Thus, the two experiments do not establish one policy that jointly scales to all seven skills and a large object set. The 67.27% frame-wise SR applies only to zero-shot Grasp–Move–Place composition on four objects; the more complex long-horizon examples use separately trained composite-task policies. Skills such as Regrasp, while tractable for individual objects (see Fig. 9), become increasingly difficult to learn as object diversity grows. Scaling to more complex skills and broader object categories may require large-scale, learning-based refinement of HOG data and hierarchical distillation strategies.

Furthermore, real-world deployment faces the sim-to-real gap, particularly for the highly dynamic motions that our method excels at in simulation. This demands hardware with high control frequency and force-sensing capabilities—features that remain limited in current robotic hands.

Overall, this work demonstrates the feasibility of a general-purpose dexterous controller learned from synthetic data, yet several open challenges remain for further scaling and real-world application.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction summarize the contributions on synthetic trajectory generation, tracking, and policy distillation, and these claims are supported by experimental evaluations and ablations.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The appendix discusses open-loop high-level planning, skill interference in multi-task and multi-object learning, and sim-to-real challenges.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [N/A]

Justification: This paper is primarily empirical and focuses on robotic control algorithms and data generation; it does not include new theoretical proofs.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: The manuscript and the public code repository provide implementation details, synthetic motion data, pretrained teacher checkpoints, environment setup, and instructions for data generation, HOT training, evaluation, and policy distillation. However, the release documentation does not yet provide a complete experiment-by-experiment reproduction guide for all reported baseline comparisons and ablations.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in

some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: Code, synthetic motion data, and pretrained teacher checkpoints are publicly available through `https://github.com/wyhuai/Hand-Object-Tracking`, including a link to the full motion dataset. The repository provides installation and data preparation instructions, data generation code, and commands for HOT training, evaluation, and distillation; it documents the supported workflows, while a complete reproduction guide for all baseline comparisons and ablations is not yet provided.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The main paper specifies the evaluation protocol, and the appendix provides domain randomization settings, baseline implementation details, policy distillation details, and hyperparameter configurations needed to understand the results.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

Justification: The core seven-skill comparison and teacher reward ablations (Table 2) report means and standard deviations from five evaluation seeds, with 1,000 trials per object–skill setting and seed. These measure evaluation variability, not variability across independently trained policies. Other experiments do not systematically report uncertainty, and we do not claim formal statistical significance.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [No]

   Justification: We specify the main hardware platform and training schedules, but the paper does not report detailed memory usage, wall-clock time, or total compute estimates for all experiments.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research was conducted in compliance with the NeurIPS Code of Ethics using simulation, synthetic data, and publicly available benchmark data.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.

- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [No]

    Justification: The paper focuses on technical contributions to simulated dexterous manipulation and does not include a dedicated discussion of both positive and negative societal impacts.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: This research focuses on physical simulation for robot control and does not pose risks associated with high-stakes generative models or sensitive data.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [No]

Justification: We cite the sources of the existing datasets, simulation tools, and baseline methods, and our released code repository includes an Apache-2.0 license. However, the manuscript does not comprehensively list the licenses and terms of use for the third-party assets used in this work.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: The public repository provides a README and a data generation guide documenting the released code, synthetic motion data, and pretrained teacher checkpoints, including installation, data organization and generation, training, evaluation, and distillation. The code repository includes an Apache-2.0 license, and methodological limitations are discussed in Appendix N.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: This research did not conduct new crowdsourcing experiments or human-subject studies; existing public human-motion data are used only as a benchmark.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: No new human-subject study, crowdsourcing study, or sensitive personal data collection was conducted in this research.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [Yes]

    Justification: LLMs/VLMs are used for high-level planning, grasp selection, and keyframe generation, as described in Appendix H and the *Following Text Instructions* appendix. These are components of the manipulation pipeline, rather than tools used only for manuscript preparation.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.