LEARNING ADAPTABLE DEXTEROUS SKILLS VIA HYBRID DEXTEROUS MIMICKING

Anonymous authors
Paper under double-blind review

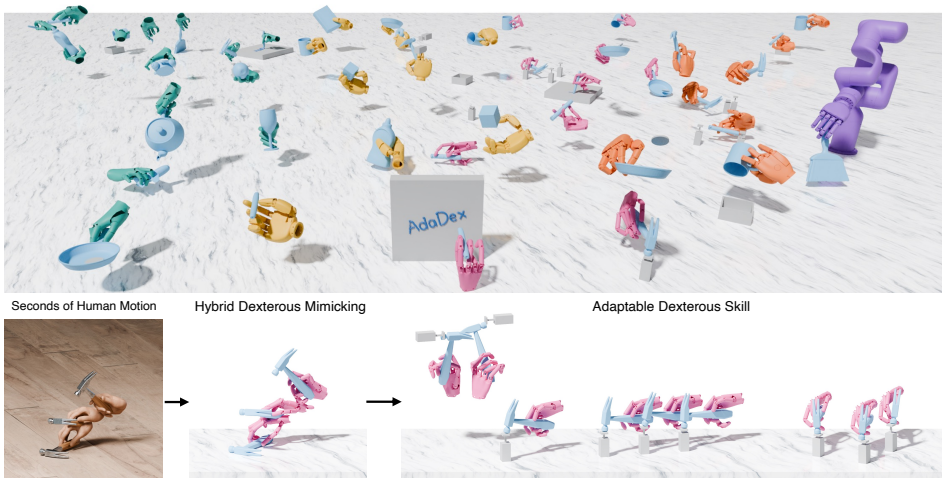

Figure 1: **ADADEX** learns diverse adaptable dexterous skills, such as functional grasping, object reposing, in-hand reorientation, and dynamic behaviors, from seconds of human motion.

ABSTRACT

Dexterous multi-finger hands hold promise to unlock diverse skills like functional grasps, in-hand reorientation, and dynamic handover that the human-designed world assumes, yet existing learning recipes force a trade-off. Goal-reaching reinforcement learning generalizes to arbitrary object poses but, lacking any human prior, converges to unnatural and unreliable finger strategies; motion tracking reproduces human-like behavior from a single clip but cannot deviate from it. We present **ADADEX**, a goal-conditioned RL framework that turns a single hand-object trajectory of a few seconds—from mocap data or HOI reconstruction from in-the-wild RGB(D) video—into an adaptable dexterous skill. At the core of **ADADEX** is **Hybrid Dexterous Mimicking**: a shared policy is trained on two populations of massively parallel environments: motion tracking environments trained with dense motion tracking objective, which instill the human prior from human motion; and generalization environments with noised initial states and goals and a sparse object-success reward, which force the skill to work far off the reference. From seconds of human motion per skill, the resulting policies can learn diverse adaptable dexterous skills, such as functional grasping, object reposing, in-hand reorientation with finger gaiting, and dynamic behaviors like throw-and-catch. Our method outperforms baselines in terms of generalization and functionality, and we demonstrate transfer across simulators and to the real world. Anonymous project page: `https://adadexanonymous.github.io/`.

1 INTRODUCTION

Our built environment is designed around the human hand: tools, handles, lids, and switches all presuppose a multi-fingered end-effector. Many of the skills this design affords, including func-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

tional tool use, in-hand reorientation, and dynamic behaviors such as throwing and catching, remain difficult for parallel-jaw grippers, which makes dexterous hands a natural embodiment for general-purpose manipulation. Despite this promise, learning dexterous skills is challenging. A multi-fingered hand must act in a high-dimensional continuous action space with 20+ degrees of freedom, and each skill must regulate contact forces, friction, and gravity across many simultaneous and often intermittent contacts under fast dynamics.

Two paradigms currently dominate learning-based dexterous manipulation, and they fail in complementary ways. Consider picking up a hammer lying on a table and striking a nail. *Goal-reaching reinforcement learning* (Kuang et al., 2026; Kedia et al., 2026) trains a single policy to move an object from any initial pose to any target pose, achieving broad coverage. Given target hammer poses, such a policy can bring the hammer to the nail. However, an object-centric reward specifies only *where* the object should end up, not *how* to get it there. Without explicit shaping, the policy could seize the hammer by its head or pinch it mid-shaft. These grasps might satisfy the goal but are not functional for hammering. Additionally, behaviors that require finger gaiting or regrasping, such as reorienting the hammer in hand from a lying-flat pickup to a striking grip with functional thumb bracing, are rarely discovered at all. As a result, the intricate motion the task demands remain largely out of reach. *Motion tracking* (Li et al., 2025; Mandi et al., 2025), in contrast, imitates a hand-object trajectory under dense supervision and thereby inherits the demonstrator's functional grasp, in-hand regrasp, and swing. The resulting policy, however, is bound to the reference. If the nail sits at a different location or angle, the policy has no mechanism to adapt, and it typically requires the dense reference as input at inference time. In short, tracking supplies the human prior and goal reaching supplies generalization, but neither alone yields a skill that is both human-like and adaptable.

To that end, we present **AdaDex**, which treats the two paradigms as complements rather than alternatives. At the core of **AdaDex** is **Hybrid Dexterous Mimicking**, built on the observation that motion tracking and goal reaching are the same goal-conditioned problem: both ask a policy $\pi_\theta(a \mid s, g)$ to bring the object and wrist to a target pose g, and differ only in where g comes from and how progress is rewarded. We therefore train one policy across two kinds of massively parallel environments: *Tracking Environments* (TrackEnv) initialize along a human hand-object reference $\hat{\tau}$, take goals from $\hat{\tau}$, and apply a dense tracking reward on object, wrist, fingers, and contacts, teaching the policy how a human grasps, gaits, and releases. *Generalization Environments* (GenEnv) perturb the initialization with large noise, offset the goal by a random $\mathrm{SE}(3)$ transformation that can be far from any frame of $\hat{\tau}$, and reward only sparse object success plus wrist guidance, with no finger terms. The two are mutually reinforcing. TrackEnv supplies the prior and the exploration: skills that sparse reward almost never discovers, such as throwing an object and catching it mid-flight, are learned under dense tracking and, because g is the only channel of task information, become directly addressable by novel goals. GenEnv supplies the generalization: it forces success from off-reference states toward off-reference goals so the skill can be adaptable to new scenarios and does not collapse into replaying one clip, while leaving the finger strategy to the human prior. Neither suffices alone; together they yield a skill that is both functional and adaptable. The recipe needs no discriminator, phase variable, or distillation stage, and the dense reference is not required at test time: at deployment the policy consumes only proprioception, object pose, object shape, and a goal pose, so it can be driven by any goal source, such as VLM Agent planner's output waypoints.

We evaluate **AdaDex** on functional grasping of diverse tools and household objects, in-hand reorientation requiring finger gaiting and contact switching, and dynamic behaviors like throw-and-catch. Every skill is learned from a single human motion trajectory of a few seconds. The learned policies preserve the grasp type, in-hand manipulation, and dynamic behaviors from human motion, while generalizing to unseen trajectories and randomized object goals far from the reference, and achieve various downstream tasks with VLM Agents under multiple embodiments. Additionally, we demonstrate that our recipe can be embedded in between real-to-sim and sim-to-real, where a few seconds of human motion from in-the-wild RGB(D) videos is converted into a simulation-ready hand-object reference via our advanced HOI reconstruction pipeline. And then the reference is broadened into an adaptable skill in simulation as described above, and the resulting simulation rollout is transferred to a real arm-hand robotic system with real-time dynamics. In summary, our contributions are:

- **Hybrid Dexterous Mimicking.** A goal-conditioned RL formulation in which dense reference tracking and sparse goal reaching are trained as parallel environment types under a single shared actor, so that the human prior from tracking and the coverage from goal reaching reinforce one another. The formulation requires no dense reference input at inference.

- **Adaptable dexterous skills.** Human-like functional grasping, in-hand reorientation with finger gaiting, and dynamic behaviors like throw-and-catch, each learned from seconds of human motion and generalized to novel states and goals. The resulting adaptable skills can achieve various downstream tasks under diverse embodiments with VLM Agent planners.
- **Learning from real-to-sim HOI motions.** A reconstruction pipeline that converts a clip of in-the-wild human video into a simulation-ready hand-object reference, outperforming existing HOI reconstruction baselines. This pipeline allows us to learn adaptable dexterous skills from human data and transfer their simulation rollouts across simulators and to the real world.

2 Related Work

2.1 Dexterous Skill Learning

Endowing robots with human-level, generalizable dexterity is a long-standing goal toward generalist agents that operate across diverse real-world scenarios. A large body of prior work formulates dexterous manipulation as contact-based optimization (Liu et al., 2020; 2021; Wang et al., 2022; Zhang et al., 2024; Yin & Abbeel, 2025), synthesizing static grasps and planning motions. These approaches are largely confined to grasping and, lacking closed-loop feedback, remain brittle under disturbances. An alternative line of work collects teleoperated demonstrations for behavior cloning (Wang et al., 2024a; Lin et al., 2025; Wen et al., 2025; Xu et al., 2025a). The resulting policies are bound to the distribution of in-domain demonstrations and generalize poorly; moreover, existing teleoperation interfaces (e.g., gloves and exoskeletons) struggle to capture behaviors that demand high dexterity or dynamic motion. More recently, reinforcement learning (RL) has emerged as a promising route to generalizable dexterity. Task-specific methods have been developed for dexterous grasping (Wan et al., 2023; Lum et al., 2024; Singh et al., 2025) and in-hand reorientation (Qi et al., 2023; Wang et al., 2024b; Liu et al., 2025b), while two more general paradigms have gained traction: motion tracking (Liu et al., 2025a; Li et al., 2025; Mandi et al., 2025; Xu et al., 2025b; Liang et al., 2026; Li et al., 2026; Feng et al., 2026) and generalized goal reaching (Kuang et al., 2026; Kedia et al., 2026). Each paradigm exhibits a complementary limitation: motion tracking faithfully reproduces a reference but cannot generalize beyond it, whereas goal reaching handles arbitrary goals but yields motions that lack human-like functionality. Our method unifies the two so that they reinforce each other, learning adaptable dexterous skills that span functional grasping, in-hand manipulation, and dynamic behaviors from only seconds of human motion.

2.2 Robot Learning from Human Motion

Human data constitutes the most scalable source of supervision for robot learning, and human motion in particular has been widely exploited. Motion tracking has become the dominant recipe for humanoid control and loco-manipulation (Liao et al., 2026; Luo et al., 2026; Xie et al., 2026), and the same idea has been extended to dexterous manipulation (Liu et al., 2025a; Xu et al., 2025b; Lum et al., 2025; Liang et al., 2026; Li et al., 2026; Feng et al., 2026). These methods, however, typically offer limited generalization and require a dense reference trajectory as the goal at inference time. We are inspired by recent work on hybrid imitation learning, including HIL (Wang et al., 2026b) and GfR (Wang et al., 2026c), which relax this requirement, enabling humanoid parkour from sparse goals while still generalizing beyond the reference. Building on this insight, our method couples motion tracking with goal reaching RL to learn dexterous skills that are simultaneously human-like and generalizable from seconds of human motion.

3 Method

Our goal is to turn a single human hand-object trajectory $\hat{\tau}$ of a few seconds into an adaptable dexterous skill that preserves the human's manipulation strategy yet remains adaptable from novel scenarios that the reference never visited. We first formalize the problem as goal-conditioned reinforcement learning (RL) (§3.1). We then present the core contribution, **Hybrid Dexterous Mimicking**, which trains one policy on two populations of parallel environments: *Tracking Environments* that supply human prior through dense tracking rewards (§3.2.1), and *Generalization Environments* that supply generalization through sparse goal-reaching rewards on noised goals (§3.2.2). Finally, we describe

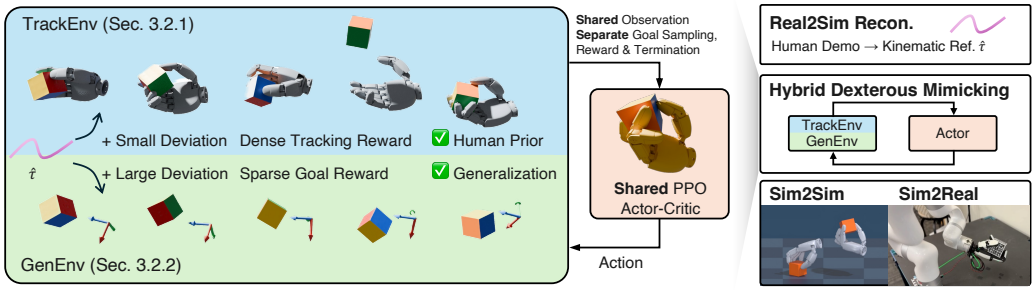

Figure 2: Left: **Hybrid Dexterous Mimicking** is a goal-conditioned RL formulation in which dense reference tracking and sparse goal reaching are trained as parallel environment types under a single shared actor. Right: our method can learn adaptable dexterous skills with a single human motion reconstructed from video and transfer to an unseen simulator (MuJoCo) and the real world.

how our method extends to real-to-sim and sim-to-real—how the reference is reconstructed from in-the-wild human videos (§B) and how the simulation rollout is transferred to hardware (§3.3).

3.1 Problem Formulation

Hand-object reference. We take a single hand-object interaction (HOI) reference

$$\hat{\tau} = \{\hat{x}_t^w,\ \hat{q}_t^h,\ \hat{x}_t^o\}_{t=1}^{T}, \tag{1}$$

where $\hat{x}_t^w \in \mathrm{SE}(3)$ is the human wrist pose, $\hat{q}_t^h$ are the MANO (Romero et al., 2022) hand parameters , $\hat{x}_t^o \in \mathrm{SE}(3)$ is the object pose, and T is the number of frames, typically a few seconds at 60 Hz. Together with a watertight object mesh O, $\hat{\tau}$ is all the human motion our method receives. The reference may come from a motion-capture dataset or be reconstructed from an in-the-wild RGB(D) video (§3.3 and §B); the remainder of the pipeline is agnostic to its origin.

Goal-conditioned MDP. We cast dexterous manipulation as a goal-conditioned Markov Decision Process (MDP) $\mathcal{M} = (\mathcal{S}, \mathcal{G}, \mathcal{A}, \mathcal{P}, r, \gamma)$ and learn a policy $\pi_\theta(a_t \mid s_t, g_t)$. The state

$$s_t = \big(\, q_t,\ x_t^w,\ \dot{x}_t^w,\ x_t^o,\ \mathrm{BPS}(O) \,\big) \tag{2}$$

consists of proprioception q_t, robot wrist pose $x_t^w \in \mathrm{SE}(3)$ and its linear and angular velocity $\dot{x}_t^w$, object pose $x_t^o \in \mathrm{SE}(3)$, and a fixed-size shape code given by Basis Point Set (BPS) distances to the object mesh (Prokudin et al., 2019). There is no privileged information in s_t. The goal

$$g_t = \big(\, g_t^w,\ g_t^o \,\big) \in \mathrm{SE}(3) \times \mathrm{SE}(3) \tag{3}$$

specifies a target wrist pose and target object pose. The action $a_t \in \mathbb{R}^{6+n}$ consists of a 6-DoF delta wrist command and a n-DoF absolute finger command, which are filtered and served as PD targets.

A defining property of the formulation is that **the dense reference $\hat{\tau}$ is not needed at test time**. It shapes initial states, goals, and rewards during training only. At test time the policy sees the state and a stream of goals, which can be different from what $\hat{\tau}$ gives. Therefore, the same interface serves any goal source: the reference itself, a video planner, or waypoints generated by VLM Agents, to accomplish downstream tasks, as shown in §4.3.

3.2 Hybrid Dexterous Mimicking

Pure motion tracking and pure goal-reaching RL sit at opposite ends of a trade-off. Tracking a single clip (Li et al., 2025; Mandi et al., 2025) yields human-like, dexterous motion but a policy that can only replay that clip. Goal-reaching with a sparse success reward, as in Dex4D (Kuang et al., 2026) and SimToolReal (Kedia et al., 2026), generalizes to arbitrary goals but has no prior on *how* to move and converges to unnatural, unreliable finger strategies, and it rarely discovers behaviors such as finger gaiting on its own. Hybrid Dexterous Mimicking, inspired by hybrid imitation for humanoid control (Wang et al., 2026b;c), obtains both by running the two settings as different *environments* that share one policy.

Concretely, a fraction p of the N parallel environments are *Tracking Environments* (TrackEnv) and the remaining $1-p$ are *Generalization Environments* (GenEnv). The two differ only in their initial-state distribution, their goal distribution, and their reward; the observation space, the action space, and the actor π_θ are identical. Because the goal g_t is the only channel through which task information reaches the actor, behaviors learned by tracking $\hat{\tau}$ in TrackEnvs become directly addressable by novel goals in GenEnvs. The joint objective is

$$J(\theta) = p \cdot \mathbb{E}_{\hat{\tau},\,\text{track}}\Big[\sum_t \gamma^t\, r_t^{\text{track}}\Big] + (1-p) \cdot \mathbb{E}_{\hat{\tau}\oplus\varepsilon,\,\text{gen}}\Big[\sum_t \gamma^t\, r_t^{\text{gen}}\Big], \tag{4}$$

with one single policy π_θ optimized with PPO (Schulman et al., 2017) on the union of rollouts.

Asymmetric critic. We give the critic privileged information during training (Pinto et al., 2018). Specifically, advantages are estimated with a single shared value network $V_\phi(s_t, g_t, s_t^{\text{priv}}, \hat{\tau})$ that receives, in addition to the actor's inputs, privileged simulator state s_t^{priv} (contact forces, object velocity, etc.) and the HOI reference $\hat{\tau}$. In GenEnvs the reference input is zeroed ($\hat{\tau} \equiv 0$) since there is no reference to track and the value must be inferred from the goal alone. The presence or absence of $\hat{\tau}$ therefore also lets one critic differentiate two environments without any explicit task flag.

3.2.1 Tracking Environments (TrackEnv)

Tracking environments teach the policy the human prior: how to approach the object, which grasp to use, how to gait fingers, and when to release.

Reference state initialization and goal sampling. Each episode samples a reference frame t_0 from $\{1, 2, \ldots, T\}$ and initializes hand joints, wrist pose, and object pose from $\hat{\tau}_{t_0}$ with small pose noise, so that every phase of the motion is learned in parallel rather than sequentially. The goal at time t is read directly from the reference with a short look-ahead k:

$$g_t = \big(\hat{x}^w_{t+k},\, \hat{x}^o_{t+k}\big). \tag{5}$$

After k steps, the goal will be sampled again with a new look-ahead.

Dense tracking reward. The reward is a weighted sum of exponentiated tracking errors plus a contact term and a regularizer:

$$r_t^{\text{track}} = \sum_i w_i \exp\big(-\|e_t^i\|^2/\sigma_i^2\big) + r_t^{\text{contact}} + r_t^{\text{reg}}, \tag{6}$$

with errors on object position $e_t^{o,\text{p}}$ and rotation $e_t^{o,\text{R}}$, wrist position $e_t^{w,\text{p}}$ and rotation $e_t^{w,\text{R}}$, and MANO link positions e_t^{link}.

The object reward term has the highest weight: they define where the object should go and task success. The hand reward term carries the human prior: approach direction, grasp type, finger gaiting, and release timing. The contact term r_t^{contact} rewards simulated fingertip contacts derived from the HOI reference based on fingertip-object distance. The regularizer r_t^{reg} penalizes action rate, torque, joint-limit violations, and table penetration.

Adaptive sampling and termination. Initial reference frames t_0 are grouped into B bins along the motion. We maintain an exponential moving average of the tracking success s_b of each bin and perform reference state initialization on bins with probability $P(b) \propto \sigma(s_b) + \epsilon$, where $\sigma(\cdot)$ is the success variance, so that both solved and hopeless bins receive little probability and the frontier receives most, facilitating learning harder initial states. Episodes terminate as soon as any tracked quantity (object pose, wrist pose, or MANO link position) leaves a tight band around $\hat{\tau}$, or when the object falls or leaves the workspace. Strict termination forces the policy to follow the reference closely to survive, which is what makes the tracking signal a faithful prior.

3.2.2 Generalization Environments (GenEnv)

Generalization environments stretch the skill off the reference. They pose the same problem as Dex4D (Kuang et al., 2026) and SimToolReal (Kedia et al., 2026), goal-reaching with object-centric reward, which alone produces unnatural finger strategies; here that problem is solved by an actor that is simultaneously constrained to the manifold of human-like behavior by TrackEnvs.

Noised initialization and goal sampling. Episodes also use reference state initialization as TrackEnvs, but $t_0 \sim \mathcal{U}\{1..T\}$ is sampled uniformly without adaptive sampling, with much larger noise on object and hand pose, so initial states lie far from $\hat{\tau}$. Goals are drawn by perturbing a k-step look-ahead reference frame in $\mathrm{SE}(3)$:

$$g_t = \big(\hat{x}^w_{t+k} \oplus \varepsilon,\ \hat{x}^o_{t+k} \oplus \varepsilon\big), \tag{7}$$

$$\varepsilon = (\delta_\mathrm{p}, \delta_\mathrm{R}), \qquad \delta_\mathrm{p} \sim \mathcal{U}(-\Delta_\mathrm{p}, \Delta_\mathrm{p})^3, \qquad \delta_\mathrm{R} = \mathrm{Exp}\big(\mathcal{U}(-\Delta_\mathrm{R}, \Delta_\mathrm{R})^3\big), \tag{8}$$

where $\oplus$ denotes transform composition in $\mathrm{SE}(3)$. The noise magnitudes $\Delta_\mathrm{p}, \Delta_\mathrm{R}$ can be large: the goal does not need to lie near any reference frame. Once the goal is reached, a new goal is sampled from perturbing a new look-ahead frame, so a single episode chains several transitions.

Sparse object-centric reward with dense wrist guidance. The GenEnv reward consists of sparse object success plus wrist guidance and regularizers:

$$r^\mathrm{gen}_t = \mathbb{1}\Big[\, \|p^o_t - g^{o,\mathrm{p}}_t\| < \delta'_\mathrm{p} \ \wedge\ \angle\big(R^o_t, g^{o,\mathrm{R}}_t\big) < \delta'_\mathrm{R} \,\Big] + r^\mathrm{wrist}_t + r^\mathrm{reg}_t. \tag{9}$$

The sparse object success indicator defines the task. The dense wrist-tracking term r^wrist_t, of the same exponentiated form as in Eq. 6, gives functional guidance toward g^w. There are no finger terms: how to get the object to the goal is left entirely to the prior learned in TrackEnvs. GenEnv episodes terminate when the object is too far away from the goal, when a goal is not reached within its time budget, or when the wrist stays too far from the object for too long. More details on reward shaping and termination are in §C.

Why the policy generalizes. The policy could generalize from the reference because we can apply **large noises** to states and goals. With only goal reaching objective, large goal noise $(\Delta_\mathrm{p}, \Delta_\mathrm{R})$ would drive a goal reaching policy toward whatever finger strategy the optimizer finds first, which can often be unnatural. However, in our Hybrid Dexterous Mimicking, TrackEnvs continually pull the shared actor back onto the manifold of human-like behaviors, while GenEnvs ask it to stretch that manifold to new states and goals. Therefore, the policy can generalize and deviate from the reference whenever the goal calls for it, but behaves with human-like grasps, gaits, and release.

3.3 Real-to-Sim HOI Reconstruction and Sim-to-Real Robot Execution

We embed our **Hybrid Dexterous Mimicking** recipe in between real-to-sim and sim-to-real, converting everyday videos of human manipulation to physically grounded robot execution.

Real2Sim: from video to HOI reference. When $\hat{\tau}$ is not available from a mocap dataset, we reconstruct it from a few seconds of RGB(D) video, recorded in the lab or in the wild (Fig. 7). We obtain a watertight object mesh O, per-frame object poses $\hat{x}^o_t$, and per-frame MANO hand parameters $\hat{q}^h_t$ and wrist poses $\hat{x}^w_t$. These independent estimates are then aligned and optimized post hoc, including depth-based hand motion alignment, gravity axis estimation, and table plane alignment. The output is a HOI reference that is consistent and smooth enough to be tracked in simulation. Mocap datasets plug into the pipeline directly at this output. Through a series of careful design choices, our reconstruction pipeline outperforms existing HOI reconstruction baselines (Paliwal et al., 2026; Wang et al., 2026a) in both reconstruction accuracy and runtime efficiency, as shown in Table 3 and Fig. 8. More details can be found in §B.

Sim: retargeting and Hybrid Dexterous Mimicking. The HOI reference is kinematically retargeted to specific hand embodiments to obtain initial wrist and finger configurations for reference state initialization (more details in §D). Training then proceeds as in §3.2 in Isaac Gym (Makoviychuk et al., 2021) with massive parallel environments, with TrackEnvs and GenEnvs co-existing.

Sim2Real: from simulation to physically grounded robot execution. In simulation, we adopt system identifications, domain randomizations, and sim2sim to facilitate sim2real transfer. Following Liu et al. (2025a); Li et al. (2025); Pan et al. (2025); Paliwal et al. (2026), in the real world we open-loop replay the simulation rollouts at original 1x speed (30Hz), generating smooth, human-speed and physically plausible real robot rollouts. More details are in §4.4.

Table 1: Motion tracking performance on (i) **seen** motions and (ii) **similar but unseen** motions. Best and second-best mean values for each metric within each setting are **bolded** and underlined.

Method	Progress (%) ↑	$\bar{e}^{o,\mathrm{p}}$ (cm) ↓	$\bar{e}^{o,\mathrm{R}}$ (°) ↓	$\bar{e}^{w,\mathrm{p}}$ (cm) ↓	$\bar{e}^{w,\mathrm{R}}$ (°) ↓
Setting (i): seen (trained) motions					
ManipTrans	35.20 ± 29.71	13.69 ± 16.11	41.61 ± 40.97	45.55 ± 65.28	55.22 ± 44.03
SimToolReal	9.72 ± 5.44	22.15 ± 18.32	53.56 ± 26.77	21.43 ± 9.79	65.30 ± 21.25
w/o TrackEnv	8.45 ± 4.98	34.10 ± 20.38	76.64 ± 31.72	3.65 ± 1.81	42.52 ± 25.23
w/o GenEnv	$\mathbf{97.47 \pm 8.40}$	$\mathbf{0.62 \pm 0.29}$	$\mathbf{2.24 \pm 0.79}$	$\mathbf{2.76 \pm 0.51}$	$\mathbf{8.16 \pm 4.73}$
AdaDex	96.74 ± 6.21	0.72 ± 0.32	4.16 ± 1.19	2.82 ± 0.61	11.45 ± 5.26
Setting (ii): similar but unseen motions					
ManipTrans	7.28 ± 6.57	37.95 ± 20.67	88.49 ± 26.81	256.50 ± 199.19	114.53 ± 21.40
SimToolReal	13.68 ± 17.88	19.80 ± 16.20	50.45 ± 25.85	24.34 ± 10.44	93.52 ± 33.78
w/o TrackEnv	9.62 ± 17.71	25.77 ± 16.92	63.15 ± 35.06	$\mathbf{14.24 \pm 8.11}$	79.26 ± 38.84
w/o GenEnv	31.75 ± 31.23	23.25 ± 22.82	52.13 ± 38.84	31.13 ± 23.28	87.91 ± 33.98
AdaDex	$\mathbf{80.02 \pm 26.01}$	$\mathbf{6.07 \pm 7.60}$	$\mathbf{15.83 \pm 12.93}$	14.32 ± 7.62	$\mathbf{72.68 \pm 44.90}$

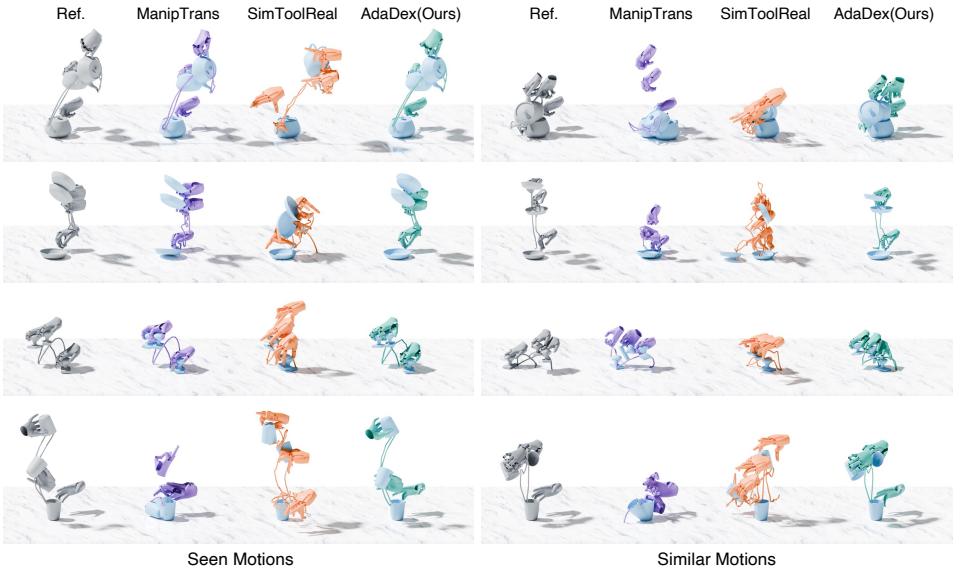

Figure 3: Qualitative comparisons. Trained on a single human motion, **AdaDex** can track seen and unseen motions naturally, while baselines struggle with generalization and functionality.

4 Experiments

4.1 Evaluation Protocols

Benchmark. We evaluate our method on reference motion clips over diverse objects and motions. For quantitative studies, we select 12 distinct objects from GRAB dataset (Taheri et al., 2020), with 3 distinct but similar motion clips for each object, where each object-specific policy is **only trained on one of the 3 motions**, and we use the other two unseen motions as the test set. We also capture 9 casual videos and reconstruct their HOI reference as in §3.3 and §B with a RealSense D455 camera, to compare tracking performance on real-to-sim reconstructed HOI motions. Quantitative experiments are conducted on a 22-DoF Arti-Mano floating hand with 6-DoF wrist control, and we also demonstrate transfer to other robot hands and arm+hand setting in §4.3 and §4.4.

Baselines. We compare our method with *motion tracking* baseline ManipTrans (Li et al., 2025) and *goal reaching* baseline SimToolReal (Kedia et al., 2026), to demonstrate our method's superiority on tracking accuracy and motion naturalness. For ours and ManipTrans, we train one policy per

Table 2: **Setting (iii): random goal reaching** comparison over the 12 GRAB objects.

Method	Progress (%) ↑	SR (%) ↑
ManipTrans	1.72 ± 2.27	0.00 ± 0.00
SimToolReal	$\underline{44.60 \pm 31.59}$	$\mathbf{31.71 \pm 32.56}$
w/o TrackEnv	4.34 ± 7.38	1.43 ± 3.58
w/o GenEnv	14.53 ± 9.22	0.52 ± 1.03
AdaDex	$\mathbf{51.80 \pm 21.80}$	$\underline{27.08 \pm 17.41}$

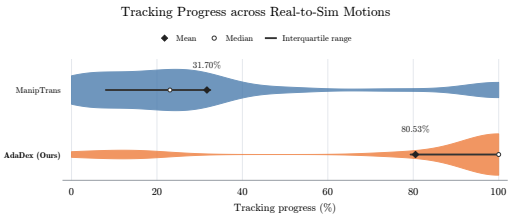

Figure 4: **Setting (iv):** comparison on tracking real-to-sim HOI motions.

motion clip, and for SimToolReal without explicit hand tracking, we finetune on their checkpoint to train object-specific policies for each object. We also include two ablation baselines *w/o TrackEnv* and *w/o GenEnv* to verify the effectiveness of both environments in our training.

Experimental Setup. We evaluate our method and baselines on four settings: (i) motion tracking on **seen** motions, which are the 12 motions the policies directly train on (except SimToolReal); (ii) motion tracking on **similar** motions which are *unseen* to the policies (2 unseen motions per object); and (iii) **random goal reaching** similar to SimToolReal (Kedia et al., 2026), where each trial attempts 10 consecutive procedural object goals (6–10 cm translation and 25–45° rotation; a goal is reached at $\leq$5 cm and $\leq$20°); (iv) motion tracking on the 9 **real-to-sim HOI references** from in-the-wild videos. Each motion is evaluated with 128 rollouts.

Metrics. For setting (i) and (ii) above, we use tracking progress as the main metric, which is the ratio of the first failure step to the full clip length, where failure is defined as $e^{o,\mathrm{p}} > 5\,\mathrm{cm} \vee e^{o,\mathrm{R}} > 20°$. Additionally, we report mean object and wrist pose error $\bar{e}^{o,\mathrm{p}}, \bar{e}^{o,\mathrm{R}}, \bar{e}^{w,\mathrm{p}}, \bar{e}^{w,\mathrm{R}}$ along the full trajectory (without termination). For setting (iii), we report average goal reaching progress (percentage of consecutive goals reached) and success rate (SR) of reaching all 10 goals. For setting (iv), we report tracking progress for real-to-sim references since there is no groundtruth label. We also loosen the failure as exceeding the threshold for 10 consecutive steps due to imperfect reconstructions.

4.2 Evaluation Results and Analysis

We report evaluation results for setting (i) & (ii) in Table 1, setting (iii) in Table 2, and setting (iv) in Fig. 4. As shown in Table 1, **AdaDex** outperforms both ManipTrans, SimToolReal, and *w/o TrackEnv* by a large margin. It's worth noting that although ours is slightly worse than *w/o GenEnv* (which only focuses on tracking/overfitting the seen motion) by less than 1% progress on trained motions, we perform much better on unseen motion tracking (80.02% vs. 31.75%), demonstrating GenEnv's contribution on generalizing beyond the reference.

In the SimToolReal-like random goal reaching setting (iii) in Table 2, on goal reaching progress, we even outperform SimToolReal which is explicitly trained on this objective ($+7.20\%$), and slightly worse in SR (-4.63%). From wrist tracking accuracy and qualitative comparisons in Fig. 3, it shows that SimToolReal struggles to grasp objects with diverse object geometries rather than the elongated tool objects they are pre-trained on, and exhibits unnatural grasp strategies without functionality, showing the effectiveness of our method which learns human priors from human motions.

We also show that our method is more capable than existing motion tracking method (Li et al., 2025) on tracking reconstructed real-to-sim HOI motions from in-the-wild videos in Fig. 4. Because reconstruction is imperfect and often has noise (e.g., finger and wrist drifting, object pose jittering), existing method cannot reliably track these motions, while our method can track these motions well due to better design choices such as sparse goal condition and adaptive sampling.

4.3 Downstream Tasks with VLM Agents

To demonstrate that our learned dexterous skills can be utilized in downstream tasks, we construct several task suites with several robot embodiments, as shown in Fig. 5. For each task, we use a high-level VLM Agent (Codex w/ GPT-6 Astra) for generating a set of waypoints $\{g_t = (g_t^w, g_t^o), t = 0, 1, 2, \dots\}$ for the skill. As shown in Fig. 5, our dexterous skills can



Figure 5: **ADADEX** learns diverse skills such as functional grasping, in-hand reorientation, dynamic throw-and-catch, and can achieve various downstream tasks with high dexterity. Our method also works on various robot hand embodiments and real arm-hand systems.

perform diverse downstream tasks like hammering, pouring, sweeping, writing, etc. These tasks further demonstrate our skills' adaptability and capability on dexterous functional grasping such as hammering, in-hand manipulation such as pen-spinning, and dynamic behaviors such as throw-and-catch. We envision a future where agentic robot systems shine, in which VLM-based agents generate high-level plans in low-frequency, where adaptable dexterous `\skills` control the robots to execute the plans in high frequency.

4.4 SIM-TO-SIM AND SIM-TO-REAL TRANSFER

To demonstrate transfer across simulators and to the real world, we first adopt a standard sim-to-sim transfer where we import policies trained in Isaac Gym (Makoviychuk et al., 2021) to MuJoCo (Todorov et al., 2012). In the real world, we tested on three GRAB motions (apple, hammer, cup) and two real-to-sim motions (sweep, AprilCube (Park & Agrawal, 2026)) to showcase dexterous functional grasping and in-hand manipulation. Following Li et al. (2025); Pan et al. (2025); Liang et al. (2026); Paliwal et al. (2026), we open-loop replay the **joint target** generated by the policy in **1x speed (30 Hz)** so that the real-time dynamics are preserved. As shown in Fig. 6, our method can perform functional grasping of a thin hammer, pouring of a cup, in-hand gait switching of a sweep, and in-hand reorientation of a cube, etc., showcasing real-world physical feasibility of our method.

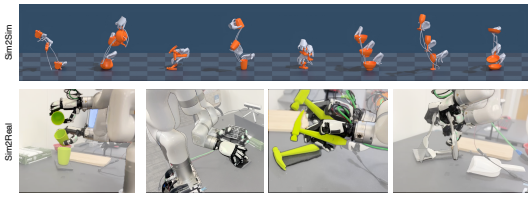

Figure 6: Closed-loop sim-to-sim transfer and open-loop sim-to-real rollout streamed in real time **(1x speed, 30 Hz)**, demonstrating the physical feasibility of the learned dexterous behaviors.

5 CONCLUSIONS

In this work we present **ADADEX**, a framework for learning adaptable dexterous skills from seconds of human motion. At the core of **ADADEX** is **Hybrid Dexterous Mimicking**: a goal-conditioned reinforcement learning framework in which a single policy is trained simultaneously in tracking environments, which supply dense supervision from the human reference, and generalization environments, which pose noised, off-reference goals under a sparse object reward. Extensive experiments in simulation and the real world verify the effectiveness of our framework, demonstrating dexterous functional grasping, in-hand reorientation, and dynamic behaviors. We further showcase our policies performing various downstream tasks with different embodiments, highlighting our recipe's generality and adaptability. Finally, we show transfer across simulators and to the real world.

AI USE STATEMENT

In this work, we used generative AI tools for polishing and rephrasing text in the manuscript and for designing the layout of figures. We have not used generative AI tools for generating or analyzing experimental results, and generative-AI-based data generation are not applicable to this work. We have reviewed all AI-assisted work. VLM-generated rephrasing and layout design are reviewed by at least 4 authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work studies learning dexterous manipulation skills from short human hand–object trajectories and does not raise concerns under the ICLR Code of Ethics. It involves no human-subject studies: the human motion we use comes from publicly available motion-capture datasets released under their respective licenses and from videos recorded by the authors themselves, which contain only hands and tabletop objects and no identifiable faces or personal information, so no IRB review was required. No sensitive, private, or harmful data was collected or used. Our method is applicable to general-purpose household and tabletop manipulation, and we do not foresee direct pathways to harmful applications beyond those common to robot learning research.

REPRODUCIBILITY STATEMENT

We have taken several steps to make this work reproducible. The problem formulation, observation and action spaces, and the hybrid TrackEnv/GenEnv design are specified in §3 and §C. The Real2Sim pipeline, including the off-the-shelf models used for mesh reconstruction and hand/object pose estimation and our alignment and filtering procedure, is described in §B. It relies only on public datasets and on RGB(D) videos captured with a low-cost consumer camera, and can therefore be reproduced without specialized capture equipment. We will release our code, trained policies, processed HOI references, and self-recorded videos upon acceptance.

REFERENCES

Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris Coll-Vinent, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, et al. Sam 3: Segment anything with concepts. In *International conference on learning representations*, volume 2026, pp. 138846–138923, 2026.

Xingyu Chen, Fu-Jen Chu, Pierre Gleize, Kevin J Liang, Alexander Sax, Hao Tang, Weiyao Wang, Michelle Guo, Thibaut Hardin, Xiang Li, et al. Sam 3d: 3dfy anything in images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7220–7232, 2026.

Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. Fast and accurate deep network learning by exponential linear units (elus). *arXiv preprint arXiv:1511.07289*, 2015.

Yunhai Feng, Natalie Leung, Jiaxuan Wang, Lujie Yang, Haozhi Qi, and Preston Culbertson. A minimalist retargeting-guided reinforcement learning recipe for dexterous manipulation. *arXiv preprint arXiv:2607.11874*, 2026.

Kushal Kedia, Tyler Ga Wei Lum, Jeannette Bohg, and C Karen Liu. Simtoolreal: An object-centric policy for zero-shot dexterous tool manipulation. *arXiv preprint arXiv:2602.16863*, 2026.

Yuxuan Kuang, Sungjae Park, Katerina Fragkiadaki, and Shubham Tulsiani. Dex4d: Task-agnostic point track policy for sim-to-real dexterous manipulation. *arXiv preprint arXiv:2602.15828*, 2026.

Kailin Li, Puhao Li, Tengyu Liu, Yuyang Li, and Siyuan Huang. Maniptrans: Efficient dexterous bimanual manipulation transfer via residual learning. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6991–7003. IEEE, 2025.

Puhao Li, Zeyuan Chen, Yingying Wu, Pengkun Wei, Yuyang Li, Tianyu Wang, Jiaxiao Shi, Mingrui Yu, Baoxiong Jia, Song-chun Zhu, et al. Towards human-level dexterous teleoperation. *arXiv preprint arXiv:2607.11481*, 2026.

Yutong Liang, Quanquan Peng, Ri-Zhao Qiu, and Xiaolong Wang. Contrack: Constrained hand motion tracking with adaptive trade-off control. *arXiv preprint arXiv:2606.03177*, 2026.

Qiayuan Liao, Takara E Truong, Xiaoyu Huang, Yuman Gao, Guy Tevet, Koushil Sreenath, and C Karen Liu. Beyondmimic: From motion tracking to versatile humanoid control via guided diffusion. *Science Robotics*, 11(117):eadx8924, 2026.

Toru Lin, Yu Zhang, Qiyang Li, Haozhi Qi, Brent Yi, Sergey Levine, and Jitendra Malik. Learning visuotactile skills with two multifingered hands. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5637–5643. IEEE, 2025.

Min Liu, Zherong Pan, Kai Xu, Kanishka Ganguly, and Dinesh Manocha. Deep differentiable grasp planner for high-dof grippers. *arXiv preprint arXiv:2002.01530*, 2020.

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*, 2025a.

Xueyi Liu, He Wang, and Li Yi. Dexndm: Closing the reality gap for dexterous in-hand rotation via joint-wise neural dynamics model. *arXiv preprint arXiv:2510.08556*, 2025b.

Tyler Ga Wei Lum, Martin Matak, Viktor Makoviychuk, Ankur Handa, Arthur Allshire, Tucker Hermans, Nathan D Ratliff, and Karl Van Wyk. Dextrah-g: Pixels-to-action dexterous arm-hand grasping with geometric fabrics. *arXiv preprint arXiv:2407.02274*, 2024.

Tyler Ga Wei Lum, Olivia Y Lee, C Karen Liu, and Jeannette Bohg. Crossing the human-robot embodiment gap with sim-to-real rl using one human demonstration. *arXiv preprint arXiv:2504.12609*, 2025.

Zhengyi Luo, Ye Yuan, Tingwu Wang, Chenran Li, Fernando Castañeda, Sirui Chen, Zi-Ang Cao, Jiefeng Li, David Minor, Qingwei Ben, et al. Sonic: Supersizing motion tracking for natural humanoid whole-body control. *Science Robotics*, 11(117):eaed4592, 2026.

Viktor Makoviychuk, Lukasz Wawrzyniak, Yunrong Guo, Michelle Lu, Kier Storey, Miles Macklin, David Hoeller, Nikita Rudin, Arthur Allshire, Ankur Handa, et al. Isaac gym: High performance gpu-based physics simulation for robot learning. *arXiv preprint arXiv:2108.10470*, 2021.

Zhao Mandi, Yifan Hou, Dieter Fox, Yashraj Narang, Ajay Mandlekar, and Shuran Song. Dexmachina: Functional retargeting for bimanual dexterous manipulation. *arXiv preprint arXiv:2505.24853*, 2025.

Bhawna Paliwal, Haritheja Etukuru, William Liang, Pieter Abbeel, Nur Muhammad Mahi Shafiullah, and Jitendra Malik. Do as i do: Dexterous manipulation data from everyday human videos. *arXiv preprint arXiv:2606.19333*, 2026.

Chaoyi Pan, Changhao Wang, Haozhi Qi, Zixi Liu, Homanga Bharadhwaj, Akash Sharma, Tingfan Wu, Guanya Shi, Jitendra Malik, and Francois Hogan. Spider: Scalable physics-informed dexterous retargeting. *arXiv preprint arXiv:2511.09484*, 2025.

Younghyo Park and Pulkit Agrawal. Aprilcube: 3d-printable fiducial targets for reliable 6-dof pose estimation, 2026. URL `https://github.com/younghyopark/aprilcube`.

Lerrel Pinto, Marcin Andrychowicz, Peter Welinder, Wojciech Zaremba, and Pieter Abbeel. Asymmetric actor critic for image-based robot learning. In *Robotics: Science and Systems (RSS)*, 2018.

Rolandos Alexandros Potamias, Jinglei Zhang, Jiankang Deng, and Stefanos Zafeiriou. Wilor: End-to-end 3d hand localization and reconstruction in-the-wild. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12242–12254. IEEE, 2025.

Sergey Prokudin, Christoph Lassner, and Javier Romero. Efficient learning on point clouds with basis point sets. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pp. 3072–3081. IEEE, 2019.

Haozhi Qi, Brent Yi, Sudharshan Suresh, Mike Lambeta, Yi Ma, Roberto Calandra, and Jitendra Malik. General in-hand object rotation with vision and touch. In *Conference on Robot Learning*, pp. 2549–2564. PMLR, 2023.

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.

Ritvik Singh, Karl Van Wyk, Pieter Abbeel, Jitendra Malik, Nathan Ratliff, and Ankur Handa. End-to-end rl improves dexterous grasping policies. *arXiv preprint arXiv:2509.16434*, 2025.

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*, pp. 581–600. Springer, 2020.

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*, pp. 5026–5033. IEEE, 2012.

Alexander Veicht, Paul-Edouard Sarlin, Philipp Lindenberger, and Marc Pollefeys. GeoCalib: Single-image Calibration with Geometric Optimization. In *ECCV*, 2024.

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*, pp. 3891–3902, 2023.

Chen Wang, Haochen Shi, Weizhuo Wang, Ruohan Zhang, Li Fei-Fei, and C Karen Liu. Dexcap: Scalable and portable mocap data collection system for dexterous manipulation. *arXiv preprint arXiv:2403.07788*, 2024a.

Gaotian Wang, Kejia Ren, Andrew Morgan, Yiting Chen, Howard H. Qian, Podshara Chanrungmaneekul, and Kaiyu Hang. Egoinfinity: A web-scale 4d hand-object interaction data engine for any-view robot retargeting and video-to-action robot learning. *arXiv preprint arXiv:2606.17385*, 2026a.

Jiashun Wang, Yifeng Jiang, Haotian Zhang, Chen Tessler, Davis Rempe, Jessica Hodgins, and Xue Bin Peng. Hil: Hybrid imitation learning for dynamic athletic control. *ACM Transactions on Graphics*, 2026b.

Jiashun Wang, M Eva Mungai, He Li, Jean Pierre Sleiman, Jessica Hodgins, and Farbod Farshidian. Generalizing from references using a multi-task reference and goal-driven rl framework. *arXiv preprint arXiv:2602.20375*, 2026c.

Jun Wang, Ying Yuan, Haichuan Che, Haozhi Qi, Yi Ma, Jitendra Malik, and Xiaolong Wang. Lessons from learning to spin" pens". *arXiv preprint arXiv:2407.18902*, 2024b.

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. *arXiv preprint arXiv:2210.02697*, 2022.

Ruicheng Wang, Sicheng Xu, Yue Dong, Yu Deng, Jianfeng Xiang, Zelong Lv, Guangzhong Sun, Xin Tong, and Jiaolong Yang. Moge-2: Accurate monocular geometry with metric scale and sharp details. *Advances in Neural Information Processing Systems*, 38:35928–35959, 2026d.

Shibo Wang, Haonan He, Maria Parelli, Christoph Gebhardt, Zicong Fan, and Jie Song. Magichoi: Leveraging 3d priors for accurate hand-object reconstruction from short monocular video clips. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 5957–5968, October 2025a.

Zihan Wang, Jiashun Wang, Jeff Tan, Yiwen Zhao, Jessica Hodgins, Shubham Tulsiani, and Deva Ramanan. Crisp: Contact-guided real2sim from monocular video with planar scene primitives. *arXiv preprint arXiv:2512.14696*, 2025b.

Bowen Wen, Wei Yang, Jan Kautz, and Stan Birchfield. Foundationpose: Unified 6d pose estimation and tracking of novel objects. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 17868–17879. IEEE, 2024.

Bowen Wen, Shaurya Dewan, and Stan Birchfield. Fast-foundationstereo: Real-time zero-shot stereo matching. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 7513–7524, 2026.

Ruoshi Wen, Guangzeng Chen, Zhongren Cui, Min Du, Yang Gou, Zhigang Han, Liqun Huang, Mingyu Lei, Yunfei Li, Zhuohang Li, et al. Gr-dexter technical report. *arXiv preprint arXiv:2512.24210*, 2025.

Tianyi Xie, Haotian Zhang, Jinhyung Park, Zi Wang, Bowen Wen, Jiefeng Li, Xueting Li, Qingwei Ben, Haoyang Weng, Yufei Ye, et al. Grail: Generating humanoid loco-manipulation from 3d assets and video priors. *arXiv preprint arXiv:2606.05160*, 2026.

Mengda Xu, Han Zhang, Yifan Hou, Zhenjia Xu, Linxi Fan, Manuela Veloso, and Shuran Song. Dexumi: Using human hand as the universal manipulation interface for dexterous manipulation. *arXiv preprint arXiv:2505.21864*, 2025a.

Sirui Xu, Yu-Wei Chao, Liuyu Bian, Arsalan Mousavian, Yu-Xiong Wang, Liangyan Gui, and Wei Yang. Dexplore: Scalable neural control for dexterous manipulation from reference scoped exploration. In *Conference on Robot Learning*, pp. 2184–2199. PMLR, 2025b.

Wenhao Yan and Jie Chu. FoundationPose++, March 2025. URL `https://github.com/teal024/FoundationPose-plus-plus`.

Lujie Yang, Xiaoyu Huang, Zhen Wu, Angjoo Kanazawa, Pieter Abbeel, Carmelo Sferrazza, C Karen Liu, Rocky Duan, and Guanya Shi. Omniretarget: Interaction-preserving data generation for humanoid whole-body loco-manipulation and scene interaction. *arXiv preprint arXiv:2509.26633*, 2025.

Zhao-Heng Yin and Pieter Abbeel. Lightning grasp: High performance procedural grasp synthesis with contact fields. *arXiv preprint arXiv:2511.07418*, 2025.

Zhengdi Yu, Stefanos Zafeiriou, and Tolga Birdal. Dyn-hamr: Recovering 4d interacting hand motion from a dynamic camera. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 27716–27726. IEEE, 2025.

Jialiang Zhang, Haoran Liu, Danshi Li, XinQiang Yu, Haoran Geng, Yufei Ding, Jiayi Chen, and He Wang. Dexgraspnet 2.0: Learning generative dexterous grasping in large-scale synthetic cluttered scenes. In *8th Annual Conference on Robot Learning*, 2024.

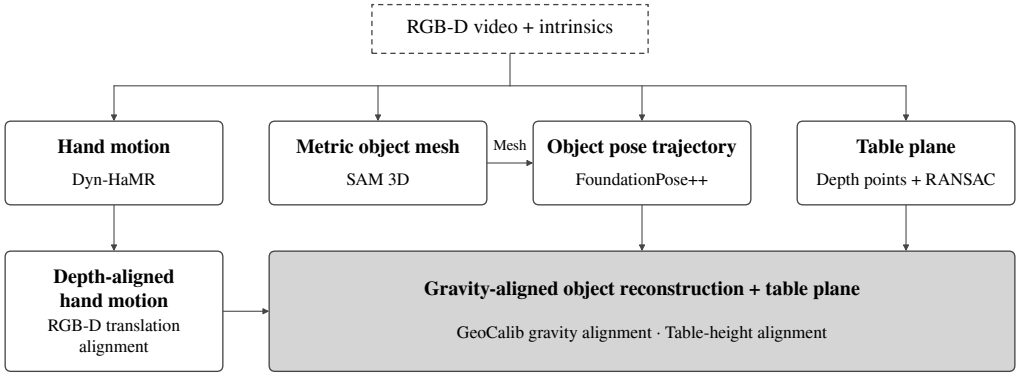

Figure 7: **Overview of our RGB-D hand-object reconstruction pipeline.** Given an RGB-D video and camera intrinsics, we recover hand motion with Dyn-HaMR, reconstruct a metric object mesh with SAM 3D, and track its pose with FoundationPose++. A table plane is estimated from depth points using RANSAC. We align hand translations to the observed depth, then apply GeoCalib-based gravity alignment and table-height alignment to express the hand, object, and table in a shared world coordinate frame.

A Videos and Visualizations

Videos and visualization results can be found in our anonymous project page: `https://adadexanonymous.github.io/`.

B Real2Sim: HOI reconstruction

Real2Sim is an important problem and would boost the data generation for simulation training and sim2real (Wang et al., 2025b). Below we elaborate our real2sim HOI reconstruction pipeline.

B.1 Reconstruction Pipeline

We use RealSense D455 stereo camera for video capture. Our reconstruction pipeline takes a static-camera RGB-D sequence $\{(I_t, D_t, K_t)\}_{t=0}^{T-1}$ as input, where $I_t \in \mathbb{R}^{H\times W\times 3}$ is an RGB frame, $D_t \in \mathbb{R}^{H\times W}$ is its pixel-aligned metric depth map in meters, and $K_t \in \mathbb{R}^{3\times 3}$ is the camera intrinsic matrix. For captured videos, depth D_t can come from RealSense SDK, Fast-FoundationStereo (Wen et al., 2026) or monocular depth estimation (Wang et al., 2026d). The pipeline outputs a metric 4D hand-object reconstruction, consisting of a canonical object mesh $\mathcal{M}_O$, world-frame object poses $\{T_{O,t}^W\}_{t=0}^{T-1}$, and world-frame MANO hand tracks $\{(\boldsymbol{\beta}, \boldsymbol{\theta}_t, R_{b,t}, \mathbf{t}_{b,t})\}_{t=0}^{T-1}$. We assume the manipulated object is rigid. Figure 7 illustrates our RGB-D hand-object reconstruction pipeline.

Object Mesh Reconstruction and Pose Tracking. We use SAM 3D (Chen et al., 2026) to reconstruct a metric object mesh from the first frame. Following SimToolReal (Kedia et al., 2026), we replace SAM 3D's internal monocular depth prediction with the observed RGB-D geometry. Given the reconstructed mesh and object masks, FoundationPose++ (Wen et al., 2024; Yan & Chu, 2025) tracks the object's 6-DoF pose across the sequence.

Hand Reconstruction and RGB-D Depth Alignment. We initialize hand motion with WiLoR (Potamias et al., 2025) MANO estimates and the known camera intrinsics, and use only the Stage 1 output of Dyn-HaMR (Yu et al., 2025) before our RGB-D depth alignment. For each frame, we obtain a hand mask from SAM 3 (Carion et al., 2026) using both the text prompt "hand" and the corresponding WiLoR detection box. We remove pixels in the object mask to avoid hand–object mask overlap.

We then hold the predicted hand pose and scale fixed and refine only the hand-root translation. Specifically, we perform $K = 3$ refinement steps, moving the hand root along its camera ray to align the rendered hand-surface depth with the observed depth measurements. At each iteration $k \in \{0, \dots, K-1\}$, we re-render the hand and construct $\Omega_t^{(k)}$, the set of valid inlier pixels in the intersection of the observed hand mask and the rendered hand silhouette. We then compute

$$\mathbf{d}_t^{(k)} = \frac{\mathbf{t}_t^{(k)}}{\left\|\mathbf{t}_t^{(k)}\right\|_2}, \qquad d_{t,z}^{(k)} = \left[\mathbf{d}_t^{(k)}\right]_z = \frac{t_{t,z}^{(k)}}{\left\|\mathbf{t}_t^{(k)}\right\|_2}, \tag{10}$$

$$r_t^{(k)} = \underset{u \in \Omega_t^{(k)}}{\text{median}} \left[D_t(u) - \hat{D}_t^{(k)}(u) \right], \tag{11}$$

$$\alpha_t^{(k)} = \text{clip}\left(\frac{r_t^{(k)}}{d_{t,z}^{(k)}}, -\delta, \delta \right), \tag{12}$$

$$\mathbf{t}_t^{(k+1)} = \mathbf{t}_t^{(k)} + \alpha_t^{(k)} \mathbf{d}_t^{(k)}. \tag{13}$$

Here, $\mathbf{t}_t^{(k)}$ denotes the hand-root translation in camera coordinates at frame t and iteration k, while $\mathbf{d}_t^{(k)}$ is the corresponding unit viewing direction and $d_{t,z}^{(k)}$ is its z-component. D_t and $\hat{D}_t^{(k)}$ denote the observed and rendered depth maps, respectively, and δ specifies the maximum translation update per iteration (set to 10 cm). The median residual provides robustness to inaccurate masks, occlusions, and depth outliers in our RGB-D setting.

Table Plane Estimation and Gravity Alignment. We recover the static table support plane from the metric depth of the anchor frame. The user selects a rectangular table region through our annotation interface, within which valid depth pixels are back-projected to camera-space 3D points. We robustly fit a plane to these points using RANSAC and allow the result to be visually inspected through an inlier overlay. Finally, we express the reconstructed hand-object trajectories in a gravity-aligned coordinate frame. Using the known camera intrinsics, we apply GeoCalib (Veicht et al., 2024) to estimate the vertical direction in camera coordinates from the first RGB frame. We construct the minimum-angle rotation that aligns this direction with the world $+z$ axis and apply it consistently to all hand poses, object poses, and the estimated table-plane normal.

B.2 Results

B.2.1 Evaluation Protocol

Evaluation Setup. We evaluate on DexYCB and HO3D dataset. For DexYCB, we evaluate on the `s0_test` subset, using only camera sequence `836212060125`, downsampling the sequences to 10 FPS. For HO3D, we evaluate on the official evaluation set, using `HO3D_v3` version, downsampling to 6 FPS.

Our method and Do-As-I-Do require the user to specify the target object; Do-As-I-Do additionally requires a reference frame to be selected. We therefore evaluate all methods under a common initialization protocol: the target object is specified by its ground-truth mask in frame 0, subsequent masks are propagated using SAM 2, and frame 0 is used as the object reconstruction reference. (For EgoInfinity, this replaces its automatic object identification and reconstruction-frame selection.) To ensure sufficient visibility for first-frame object reconstruction and subsequent tracking, we retain only sequences where at least 70% of the target object's rendered silhouette is visible in the first frame. This retains only 135 sequences in DexYCB `s0_test`, while all sequences are kept in HO3D.

For Do-As-I-Do, we use the middle frame of each sampled clip as the preferred alignment reference frame and, if it is invalid, select the nearest valid frame. This avoids fixing the reference to frame 0, where the hand is often absent.

RGB-D Adaptation of Baselines. We adapt both Do-As-I-Do (Paliwal et al., 2026) and EgoInfinity (Wang et al., 2026a) to the RGB-D setting by replacing their monocular geometry estimates with

Table 3: Hand reconstruction results on DexYCB and HO3D.

Method	DexYCB				HO3D				Runtime (s/frame) ↓
	MPJPE (mm) ↓	CD_h (cm) ↓	Seq. Success (%) ↑	Frame Success (%) ↑	MPJPE (mm) ↓	CD_h (cm) ↓	Seq. Success (%) ↑	Frame Success (%) ↑	
EgoInfinity (RGB-D)	167.81	32.24	**100.00**	97.64	81.33	15.62	**100.00**	91.73	27.06
Do-As-I-Do (RGB-D)	42.11	19.11	72.59	73.41	40.88	4.77	92.31	92.03	76.04
Ours	**25.03**	**7.64**	**100.00**	**99.53**	**27.01**	**2.89**	**100.00**	**99.73**	**0.97**

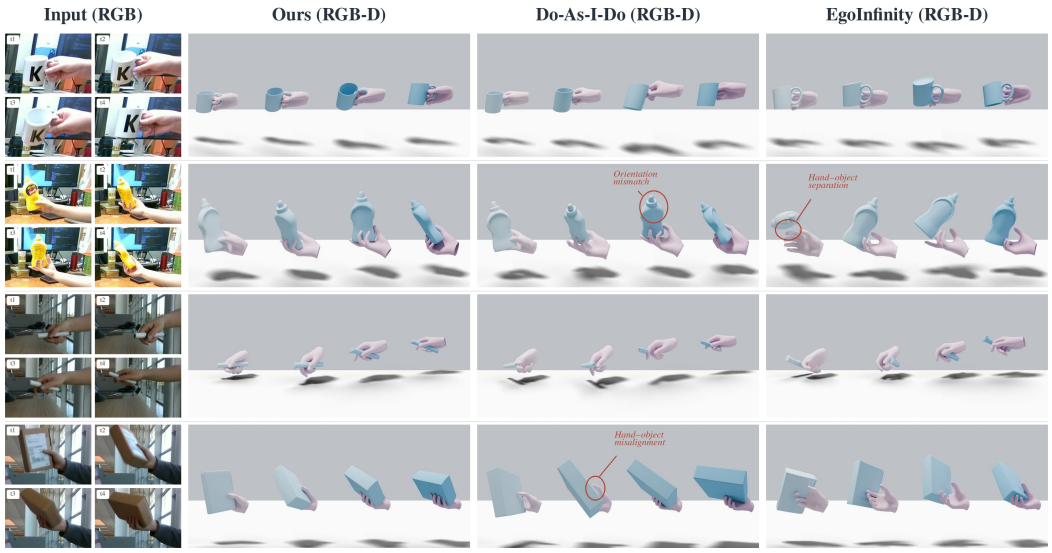

Figure 8: **Qualitative comparison of hand–object reconstruction.** Each row shows four time steps from an interaction, with RGB inputs arranged in reading order and reconstructions ordered from left to right. Do-As-I-Do and EgoInfinity are adapted to use the provided depth maps and calibrated camera intrinsics. Our reconstructions more closely preserve the object orientations and grasp configurations observed in the input videos.

the depth and camera intrinsics provided by dataset. The supplied geometry is used whenever their methods require depth or intrinsics estimation, throughout their depth-dependent reconstruction, tracking, and alignment stages, while retaining each method's underlying estimation and alignment procedures.

Metrics. We evaluate frames with valid ground-truth hand annotations. A frame is successfully evaluated if the method also produces a valid hand–object reconstruction. We report MPJPE and wrist-relative object Chamfer distance (CD_h) as frame-weighted means over these successfully evaluated frames.

MPJPE is the mean Euclidean distance between the 21 predicted and ground-truth hand joints, including the wrist, in camera coordinates. Following MagicHOI (Wang et al., 2025a), we compute CD_h between the predicted and ground-truth object meshes after expressing each relative to its corresponding wrist. We use non-squared bidirectional Chamfer distance with up to 3,000 sampled vertices per mesh. MPJPE is reported in millimeters (mm), while CD_h is reported in centimeters (cm).

Sequence-level success is the percentage of evaluation sequences for which a method produces a valid final reconstruction: Do-As-I-Do may fail to produce a final result when it cannot find a valid reference frame for hand–object alignment. Frame-level success is the percentage of ground-truth-valid frames with a valid predicted reconstruction, taking frames from failed sequences into account. A frame may fail when, for example, the hand is not detected or the method produces no valid reconstruction for that frame. We additionally report runtime measured by the average seconds needed to reconstruct a single frame.

Quantitative Results. Table 3 reports results on DexYCB and HO3D. Our method achieves the lowest MPJPE and CD_h on both datasets: 25.03 mm and 7.64 cm on DexYCB, and 27.01 mm and

2.89 cm on HO3D. It also achieves the highest frame-level success rates, at 99.53% and 99.73%, respectively. In addition, our method is the most efficient, taking the average least time (0.97 s compared to 27.06 s and 76.04 s) to reconstruct a frame.

Qualitative Results. Qualitative examples in Fig. 8 show that, even with depth observations supplied to both baselines, our method better preserves the object orientations and hand–object spatial relationships observed in the input videos. The highlighted examples illustrate orientation mismatches, hand-object separation and local hand–object misalignment in the adapted baselines.

C Implementation Details

C.1 State and Action Space

As mentioned in 3, our state s_t includes proprioception roprioception q_t, robot wrist pose $x_t^w \in \mathrm{SE}(3)$ and its linear and angular velocity $\dot{x}_t^w$, object pose $x_t^o \in \mathrm{SE}(3)$, and a fixed-size shape code given by Basis Point Set (BPS) distances to the object mesh (Prokudin et al., 2019). There is no privileged information in s_t.

For critic observation, besides the state above, it also has privileged information including hand joint velocities $\dot{q}_t$, object velocity, fingertip contact forces, object COM, and object weight. For TrackEnvs, the critic also gets the HOI reference $\hat{\tau}$, which is zero masked out in GenEnvs.

Regarding action space, for the floating Arti-Mano setting, the action a_t consists of 6-DoF delta wrist command and 22-DoF absolute finger commmand. We first apply a exponential moving average (EMA) filter with smoothing factor $\alpha = 0.4$ to the raw action, and then scale the action to joint ranges. For the arm, the scaled action is interpreted as delta wrist pose Δx_t^w with respect to current wrist pose w_t^w and converted into 6D force/torque by a PD controller, which is then applied to the floating hand base. For the finger, the scaled action is the absolute finger joint targets for the PD controller.

For physical xArm6+XHand training, instead of outputing 6-DoF delta wrist command, the first 6-DoF of a_t is the delta joint angles for the 6 xArm6 joints, which is filtered and scaled to generate PD targets for the xArm6 PD controller. The hand action remains the same.

C.2 Network Architecture and Training Details

We train the policy using PPO (Schulman et al., 2017) with asymmetric actor-critic. We initialize both actor and critic networks as MLP with hidden layers $[1024, 1024, 512]$ with ELU activations (Clevert et al., 2015). The policy action is sampled from a Gaussian distribution with learnable action std.

Our training is divided to two stages for a curriculum purpose. In stage I all parallel environments are TrackEnvs to give a good human prior for the policy, where in stage II we initialize from stage I actor checkpoint, and train under 50/50 TrackEnv and GenEnv to provide the policy with stage/goal coverage and generalization.

The policy is trained using Isaac Gym (Makoviychuk et al., 2021) with 8,192 parallel environments per GPU. Each stage is trained till convergence, with a 20k-iteration hard cap. It takes around 1-2 days for each stage to converge on a single GeForce RTX 4090 GPU. More training details and hyperparameters are reported in Table 4.

C.3 Reward Shaping

We report our reward shaping for TrackEnv and GenEnv in Table 5.

C.4 Curriculum and Termination

For curriculum design, except the GenEnv ramping up curriculum mentioned in §C.2, we also adopt a gravity and friction curriculum similar to Li et al. (2025), where at stage I early training stage,

Table 4: Network details and training hyperparameters.

Hyperparameter	**Stage I**	**Stage II**
TrackEnv / GenEnv mixture	$1.0 / 0.0$	$0.5 / 0.5$
Actor–critic MLP	$[1024, 1024, 512]$, ELU; separate AC networks	
Action log-std at stage start	-0.5 ($\sigma = e^{-0.5} \approx 0.607$)	restored from Stage I actor
Learning rate (LR)	1×10^{-4}	2×10^{-5}
LR warmup	10 epochs: $10^{-6} \rightarrow 10^{-4}$	10 epochs: $10^{-6} \rightarrow 2 \times 10^{-5}$
KL threshold	0.008	
PPO clip ϵ	0.2	
Entropy coefficient	1×10^{-3}	1×10^{-4}
Discount γ / GAE λ	$0.99 / 0.95$	
Rollout horizon	32 simulation steps	
Minibatch size	8192	
PPO mini-epochs	5	
Critic coefficient	4	
Gradient-norm limit	1.0	
Bounds-loss coefficient	1×10^{-4}	
Input / value normalization	Enabled	

Table 5: Reward shaping for tracking and generalization environments.

Reward term	**TrackEnv**	**GenEnv**
Wrist pose	$e^{-40d_w} + 6e^{-\theta_w}$	$2e^{-40d_w} + e^{-3\theta_w}$
Object pose	$10e^{-80d_o} + 5e^{-3\theta_o}$	0
Finger pose	Weighted exponential matching	0
Velocity matching	Weighted exponential matching	0
Low power	$e^{-10P_h} + e^{-2P_w}$	0
Action smoothness	$-\operatorname{mean}[(a_t - a_{t-1})^2]$	0
Goal reached (once per goal)	0	20
Final-goal bonus	0	50

the gravity is small and friction is high to make tracking easier. As training progresses, the tracking difficulty gradually goes up and gravity and friction values go back to normal values.

For early termination, for TrackEnvs, it's a standard tracking-based termination design which forces the policy to keep close to the reference. We also have a termination curriculum where termination thresholds are divided by a scaling factor 0.7, which gradually goes to 1.0 through time to make termination tighter (Li et al., 2025). For GenEnvs, episodes are early terminated when no goal reached for 30 consecutive steps (0.5 s), object position more than 50 cm from sampled goal (after the initial 8-step grace period), or wrist more than 30 cm from object for 3 s.

D Kinematic Motion Retargeting

Inspired by Yang et al. (2025); Feng et al. (2026), we adopt a interaction mesh-based kinematic retargeting method to turn the MANO (Romero et al., 2022) parameters into embodiment-specific joint positions and wrist pose. Note that the retargeted joint positions are only used for reference state initialization. The tracking reward, termination, etc., all depend on raw MANO links from the HOI reference $\hat{\tau}$, where each embodiment maintains a one-to-one MANO-to-robot mapping based on its kinematic structure.

E Benchmark Motions

Our benchmark motions are reported in Table 6. All motions are in 60 Hz.

F BASELINE DETAILS

In this work, we compare our method with representative baselines from motion tracking and general goal reaching methodologies. We compare with ManipTrans (Li et al., 2025) for motion tracking baseline and SimToolReal (Kedia et al., 2026) for goal reaching baseline. We also compare with two ablation baselines *w/o TrackEnv* and *w/o GenEnv* to show necessity and synergy between the two sets of environments.

For ManipTrans, we utilize their pretrained first-stage imitator checkpoint and learn a residual policy for each trajectory, aligned with our observation and action space. For SimToolReal, we use their original Kuka+Sharpa embodiment, and finetune their pretrained checkpoint to train object-specific policies for each object. Note that due to different interfaces of the baselines and our method, for tracking **seen** and **similar** motions, we only give object goal poses for SimToolReal, while for goal reaching, we give ManipTrans hand references from contact frames where candidate frames are restricted to frames at/after the handoff where at least two fingertips are within 4 cm of the object. For each trajectory, the candidate with the lowest deterministic contact cost is selected. Therefore, the random goal gets a plausible, source-recorded grasp configuration that is transported with the object.

G DOWNSTREAM TASK SPECIFICATIONS

In total, we have five downstream tasks: *Hammer*, *MugPour*, *Marker*, *Sweep*, and *PanToss*. Their task specifications are:

- *Hammer*: grasp the hammer on the table and hammer two nails of distinct locations.
- *MugPour*: grasp the mug by the handle, and pour the cube inside to a container on the table.
- *Marker*: in-hand spin the marker, and draw **ADADEX** on a whiteboard.
- *Sweep*: use the sweep to sweep cubes on the table to a dustbin.
- *PanToss*: grasp the handle of the pan, and toss the egg around.

We also demonstrate other robot hands (e.g., Sharpa Hand, XHand, Wuji Hand v1) to show that our recipe can be transferred to real robot embodiments and make them do actual work rather than animation purposes only.

H LIMITATIONS AND FUTURE WORK

Our method currently learns one policy per object and per reference trajectory. Scaling to a single policy that covers many objects and motions is a natural next step, and would also enable skill composition across tasks. In addition, we represent object state as a 6-DoF pose plus a BPS shape code, chosen as a minimal recipe. Adopting a point track representation as in Dex4D (Kuang et al., 2026) would unlock manipulating objects with more complex geometry, such as articulated or deformable objects.

Table 6: Benchmark motions: 12 distinct GRAB objects with one seconds-long motion each, 24 unseen GRAB motions (2 for each object), and 9 real-to-sim motions. Frames refer to the frame length of the trimmed motions for training; for unseen GRAB motions, they refer to the evaluated reference movement windows. All motions are in 60 Hz.

Object	Motion name	Frames	Time (s)
Setting (i): GRAB seen (trained) motions			
Airplane	s1_airplane_fly_1	463	7.72
Alarm clock	s1_alarmclock_pass_1	279	4.65
Apple	s1_apple_eat_1	490	8.17
Bowl	s1_bowl_pass_1	208	3.47
Cup	s1_cup_pour_1	387	6.45
Frying pan	s1_fryingpan_cook_2	351	5.85
Hammer	s1_hammer_use_1	353	5.88
Hand	s1_hand_inspect_1	423	7.05
Stamp	s1_stamp_stamp_1	367	6.12
Stapler	s1_stapler_staple_2	346	5.77
Teapot	s1_teapot_pour_1	393	6.55
Wineglass	s1_wineglass_drink_2	520	8.67
Seen GRAB subtotal		**4,580**	**76.33**
Setting (ii): GRAB unseen motions			
Airplane	s6_airplane_fly_1	200	3.33
Airplane	s2_airplane_fly_1	270	4.50
Alarm clock	s4_alarmclock_pass_1	124	2.07
Alarm clock	s3_alarmclock_pass_1	133	2.22
Apple	s2_apple_eat_1	196	3.27
Apple	s3_apple_eat_1	147	2.45
Bowl	s6_bowl_drink_1	143	2.38
Bowl	s5_bowl_pass_1	157	2.62
Cup	s2_cup_drink_1	191	3.18
Cup	s3_cup_pour_1	144	2.40
Frying pan	s4_fryingpan_cook_2	226	3.77
Frying pan	s4_fryingpan_cook_1	253	4.22
Hammer	s2_hammer_use_1	233	3.88
Hammer	s2_hammer_use_2	289	4.82
Hand	s2_hand_inspect_1	181	3.02
Hand	s5_hand_inspect_1	234	3.90
Stamp	s2_stamp_stamp_1	228	3.80
Stamp	s4_stamp_stamp_1	153	2.55
Stapler	s4_stapler_staple_1	74	1.23
Stapler	s6_stapler_staple_1_Retake	220	3.67
Teapot	s6_teapot_pour_1	350	5.83
Teapot	s4_teapot_pour_2	213	3.55
Wineglass	s3_wineglass_toast_1	178	2.97
Wineglass	s5_wineglass_toast_1	122	2.03
Unseen GRAB subtotal		**4,659**	**77.65**
Setting (iv): Real-to-sim motions			
Bottle	–	880	14.67
Box	–	626	10.43
Fork	–	193	3.22
Marker 0	–	161	2.68
Marker 1	–	200	3.33
Mug 0	–	450	7.50
Mug 1	–	297	4.95
Pen	–	144	2.40
Sweep	–	265	4.42
Real-to-sim subtotal		**3,216**	**53.60**
Total: 45 motions		**12,455**	**207.58**