

SPIDER : Scalable Physics-Informed DEXterous Retargeting

Chaoyi Pan^{1,2}, Changhao Wang¹, Haozhi Qi¹, Julen Urain¹, Zixi Liu¹, Homanga Bharadhwaj¹, Akash Sharma¹, Tingfan Wu¹, Guanya Shi², Jitendra Malik¹, Francois Hogan¹

Human Data $\longrightarrow$ Physics-Based Retargeting $\longrightarrow$ Feasible Robot Data

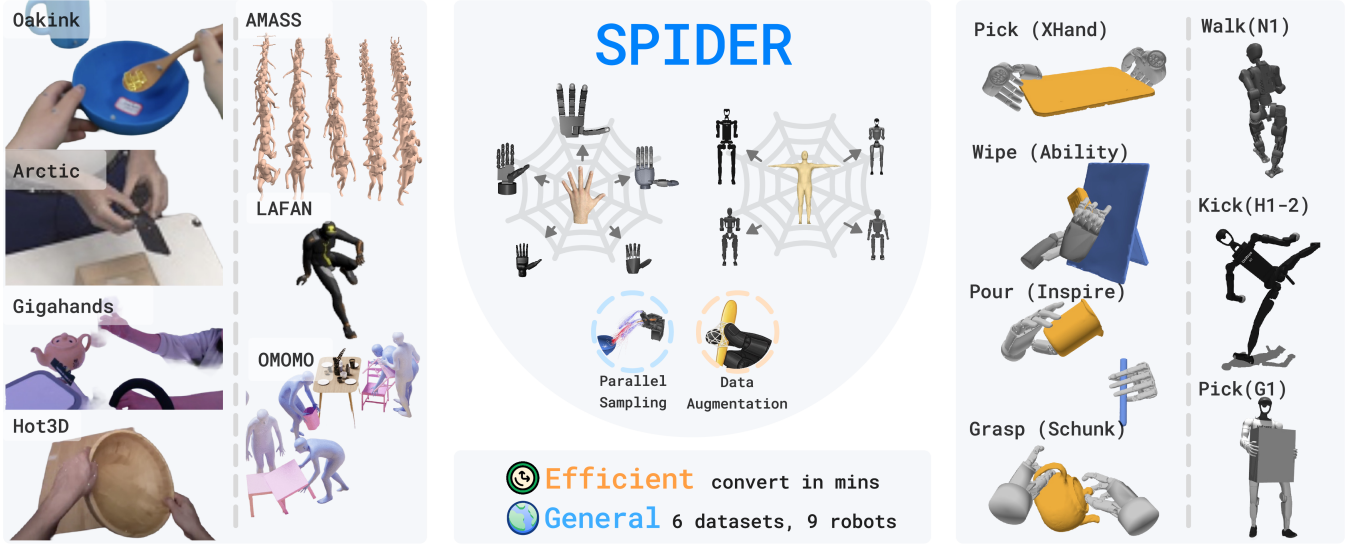


Fig. 1: **Overview of SPIDER method.** SPIDER is a general and scalable framework for cross-embodiment physics-informed retargeting from human demonstrations. It is the first universal physics-based retargeting method that supports both humanoid whole-body loco-manipulation and dexterous manipulation across 9 diverse robot embodiments, 6 datasets and 3 simulators. It can generate and augment dynamically feasible robot-object interaction trajectories at scale efficiently with a training-free sampling-based solver.

Abstract—Learning agile robotic policies requires large-scale demonstrations, but translating abundant human motion data to robots is bottlenecked by the embodiment gap and missing dynamic information. To bridge this gap, we propose **Scalable Physics-Informed DEXterous Retargeting (SPIDER)**, a physics-based retargeting framework to transform and augment kinematic-only human demonstrations into dynamically feasible robot trajectories at scale. Our key insight is that human demonstrations should provide global task structure and objective, while a sampling-based solver can be used to find the feasible solution given the physics constraints. As a general framework, SPIDER is an efficient physics-based retargeting method that can be applied to both humanoid whole-body loco-manipulation and dexterous manipulation across 9 humanoid/dexterous hand embodiments, 6 datasets and 3 simulators. By bypassing the need for policy optimization, it achieves state-of-the-art performance while being 10x faster than reinforcement learning (RL) baselines. Furthermore, SPIDER enables physics-based data augmentation—such as imposing external payloads, perturbations, and new contact patterns—to generate diverse data. Ultimately, the retargeted motion can be directly executed on physical robots using appropriate randomization, or serve as feasible reference motion for efficient RL policy learning.

I. INTRODUCTION

Whole-body dexterous manipulation has rapidly advanced by learning from massive internet-scale datasets [1] and human demonstrations [2]. Yet, acquiring generalizable whole-body manipulation across embodiments, from dexterous hands [3]–[5] to humanoid whole-body control [6], [7], remains prohibitively expensive due to hardware constraints, task complexity, and lack of large-scale embodiment-specific data [8], [9]. On the other hand, abundant human motion data is readily available, including large-scale motion-capture datasets [10], [11], internet-scale video collections [12], and VR-based interfaces [13]. Recent advances in computer vision have made it possible to reconstruct 3D human body [14] and hand motion [15], as well as object meshes [16] and trajectories [17], directly from videos. These developments create a unique opportunity to leverage human motion as a scalable source of demonstrations for learning humanoid and dexterous robot control. However, a fundamental challenge arises: the *embodiment gap* - the mismatch in morphology, dynamics, and actuation between humans and robots - which leads to infeasible motion during transfer. This motivates our central research question:

How can we efficiently and reliably transform human motion into feasible robot trajectories that

¹FAIR at Meta.

²Carnegie Mellon University.

Correspondence: chaoyip@andrew.cmu.edu. Work done at Meta.

respect dynamics and contact?

We formulate this as a physics-based retargeting problem [18]: given kinematic human demonstrations, generate robot motions that (a) align the robot’s poses with human poses, (b) follow physics constraints, and (c) preserve the task objectives of the demonstrated behavior (e.g. object manipulation). The problem arises from the following challenges: (a) *Flexibility and generalizability*: given diverse data sources and target embodiments, the framework should work for diverse tasks and human-object interactions. (b) *Scalability and efficiency*: the abundance of internet-scale human data requires approaches that are both computationally efficient and scalable to large datasets. Existing methods struggle to bridge these gaps: inverse-kinematics (IK) approaches [19], [20] are efficient but dynamically infeasible; reinforcement learning (RL) approaches [21], [22] are general but require expensive trajectory-specific policy optimization; teleoperation [3] is dynamically feasible but often labor-intensive and embodiment-specific.

To achieve scalable, general, and flexible physics-based cross-embodiment retargeting, we propose Scalable Physics-Informed DExterous Retargeting (**SPIDER**), a sampling-based approach with contact guidance. Our key insight is that human demonstrations provide high-level guidance in terms of robot motion and task specification, while physics simulator grounds trajectories to ensure dynamical feasibility and contact correctness through large-scale sampling. Our method lies in the middle of the trade-off between gradient-based optimization and reinforcement learning: it preserves the efficiency of optimization-based methods as a learning-free method, while achieving the generalizability of reinforcement learning as no gradient is required. The resulting framework is embodiment-agnostic and task-general: it can be applied to any robot-environment interaction as long as the scene can be simulated.

Contributions. our contributions are summarized as follows:

- We introduce **SPIDER**, a *general* and *scalable* physics-based retargeting framework that can be applied to both humanoid whole-body loco-manipulation and dexterous manipulation, which supports 9 diverse robot embodiments, 6 datasets and 3 simulators. It achieves state-of-the-art performance while being $10\times$ faster than RL-based methods.
- We demonstrate our method’s *flexibility* and *feasibility* to diverse policy learning and data generation applications, including more efficient RL policy learning, data augmentation for novel physics constraints, user intent preservation with contact guidance and robustifying trajectories for direct real-robot deployment.

II. RELATED WORK

A. Learning Manipulation from Human Data

Recent works learn robot skills from human videos by either mapping human poses directly to robot motions [23]–[26], or adapting human-centric policies via fine-tuning [27]–[29]. **SPIDER** complements these pipelines as a drop-in

human-to-robot transfer module, offering physically consistent, contact-aware mappings for dexterous and whole-body control.

B. Retargeting from Human Data

Motion retargeting converts human data into executable robot trajectories. Existing methods generally trade off between computational efficiency and physical fidelity. Kinematic methods [9], [19], [30] are highly efficient but ignore physics constraints, struggling with realism in contact-rich tasks [31]. Neural mapping approaches [3], [23] offer fast inference and outperform direct kinematics, but require extensive pretraining and often fail on out-of-distribution motions or novel embodiments. Optimization methods explicitly incorporate physics to ensure dynamical feasibility [18], but often rely on detailed contact priors [32], [33]. Sampling-based approaches [34], [35] address the non-convexity of this formulation. Similarly, RL produces physically feasible dexterous motions [21], [22], [36], [37], but typically requires per-trajectory training, limiting scalability to internet-scale data. **SPIDER** bridges this gap by leveraging the parallelism of sampling-based optimization to achieve both efficiency and physical fidelity.

C. Sampling-based Optimization for Robot Control

Sampling-based optimization (e.g., cross-entropy method [38], evolutionary algorithms [39], and Bayesian optimization [40]) efficiently solves non-convex, non-smooth control problems. While widely applied to navigation [41], legged locomotion [42], and dexterous manipulation [43], [44], these methods can suffer from trajectory sampling instability and solution ambiguity in contact-rich domains [45]. **SPIDER** mitigates these issues by integrating virtual contact guidance to preserve the intended contact sequence.

III. PHYSICS-BASED RETARGETING WITH SAMPLING

This section introduces our pipeline for converting human data to feasible robot trajectories, as illustrated in Fig. 2. We first formulate the physics-based retargeting problem in Sec. III-A, followed by its sampling-based solver design in Sec. III-B. Then, we introduce different data augmentation methods in Sec. III-C for downstream applications including trajectory robustification for sim-to-real transfer, data augmentation for novel physics constraints and user intent preservation with contact guidance.

A. Physics-based Retargeting Problem

We formulate physics-based retargeting as a constrained optimization problem where a robot control sequence $u_{0:T-1}$ (e.g., joint position or torque commands) is optimized to minimize the distance to the reference trajectory $x_{0:T}^{\text{ref}}$ and the control effort. As *input*, a kinematic reference state sequence $x_t^{\text{ref}}, \forall t \in \{0, \dots, T\}$ is first provided from human demonstrations. The state $x_t^{\text{ref}} = \{q_t^{\text{ref}}, v_t^{\text{ref}}\}$ is composed of reference generalized coordinates q_t^{ref} and generalized velocities v_t^{ref} . Specifically, for robots with n_u joints, the

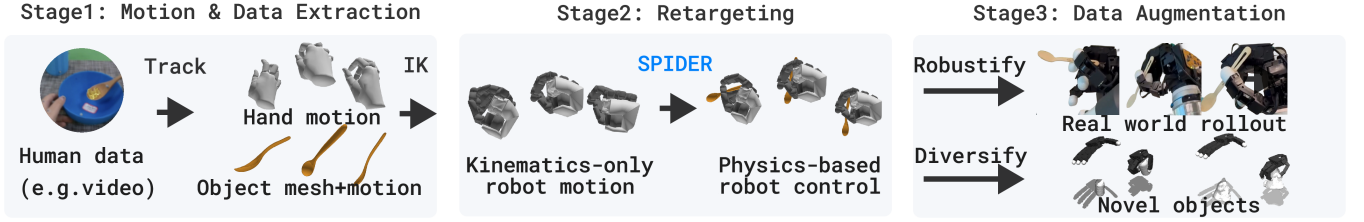


Fig. 2: **Overview of the SPIDER pipeline.** The pipeline takes reconstructed object meshes, reference robot motion, and object motion and converts them into a dynamically feasible robot trajectory with corrected contacts. The generated trajectory is further robustified and augmented before deployment or policy learning.

coordinates q_t^{ref} comprise the robot joint positions $q_t^{\text{ref, joint}} \in \mathbb{R}^{n_u}$, the robot base transformation $X_t^{\text{ref, base}} \in \text{SE}(3)$, and the object pose $X_t^{\text{ref, object}} \in \text{SE}(3)$. Physics-based retargeting is performed *offline*: the *output* is a dynamically feasible open-loop control sequence $u_{0:T-1}$ (e.g. joint position command used in this paper), valid under aligned initial conditions between the robot/object and the reference. Real-robot experiments demonstrate feasibility from such aligned starts. The optimization minimizes the distance to the reference trajectory and the control effort:

$$\begin{aligned} \min_{u_{0:T-1}} J(u_{0:T-1}) &= \min_{u_{0:T-1}} \|x_T \ominus x_T^{\text{ref}}\|_{Q_T}^2 + \\ &\quad \sum_{t=0}^{T-1} \left(\|x_{t+1} \ominus x_{t+1}^{\text{ref}}\|_{Q_t}^2 + \|u_t\|_{R_t}^2 \right) \quad (1a) \\ \text{s.t. } x_{t+1} &= f(x_t, u_t, t) \quad \forall t \in \{0, \dots, T-1\} \quad (1b) \\ x_{0:T} &\in \mathcal{X}, \quad u_{0:T-1} \in \mathcal{U} \quad (1c) \end{aligned}$$

where $x_{0:T}$ is the optimized feasible state, $x_{t+1} = f(x_t, u_t, t)$ is the state transition function, $\mathcal{X}$ is the state space, $\mathcal{U}$ is the control input space, and Q_t and R_t are the state and control input weighting matrices. Because the generalized coordinates contain poses on the $\text{SE}(3)$ manifold, standard subtraction is undefined. We use the operator $\ominus$ to denote the difference in generalized coordinates, which evaluates the error between variables in Euclidean space normally, while mapping differences between $\text{SE}(3)$ transformations to the local 6D tangent space using the logarithmic map. In practice, we use diagonal weighting matrices for Q_t and R_t , where $Q_t = \text{diag}(\{q_{\text{robot}}, q_{\text{object}}\})$ and $R_t = \text{diag}(\{r_{\text{robot}}, r_{\text{object}}\})$.

B. Sampling for Physics-based Retargeting

Due to the contact-rich nature of physics-based retargeting, the optimization problem in Eq. (1a) is highly non-convex and often discontinuous. Sampling-based optimization [46] provides a natural way to handle such landscapes, as it does not rely on smoothness or convexity assumptions. Intuitively, this resembles RL in that both rely on sampled trajectories from parallelized simulation to guide decision-making, but instead of updating a policy network, we directly use the samples to optimize the control sequence $U = u_{0:T-1}$. To this end, we adopt a sampling-based optimizer with an

Algorithm 1 Sampling for Physics-based Retargeting

```

1: Input:  $x_{0:T}^{\text{ref}}, U^0, H, N_W, \lambda, \beta_1, \beta_2, N, \epsilon_{\text{tol}}$ 
2:  $J_{\text{best}} \leftarrow \infty, i \leftarrow 0$ 
3: while  $i < N$  and not converged do
4:    $i \leftarrow i + 1$ ; Update  $\Sigma_h^i$  using Eq. (3)
5:   for  $j \in \{1, \dots, N_W\}$  do
6:     Sample  $[W]_j \sim \mathcal{N}(0, \Sigma_{0:H-1}^i)$ ;  $[U^i]_j \leftarrow U^i + [W]_j$ 
7:     Compute  $J([U^i]_j)$  by simulation rollout
      (Eq. (1a))
8:   end for
9:   Update  $U^{i+1}$  using weighted average from Eq. (2)
10:  if  $|J_{\text{best}} - \min_j J(U_j)| < \epsilon_{\text{tol}}$  then
11:    break  $\triangleright$  Early stop if improvement is small
12:  end if
13:   $J_{\text{best}} \leftarrow \min_j J(U_j)$ 
14: end while
15: Return  $U^*$ 

```

annealed sampling kernel [42], [47] as shown in Alg. 1.

$$U^{i+1} = U^i + \frac{\sum_{j=1}^{N_W} \exp\left(-\frac{J(U^i + [W]_j)}{\lambda}\right) [W]_j}{\sum_{j=1}^{N_W} \exp\left(-\frac{J(U^i + [W]_j)}{\lambda}\right)}, \quad (2)$$

$$\Sigma_h^i = \exp\left(-\frac{N-i}{\beta_1 N} - \frac{H-h}{\beta_2 H}\right) I, \quad (3)$$

where H is planning horizon, λ is the temperature parameter, U^i is the solution at iteration i , $i = 1, \dots, N$ is the iteration index with N being the total number of iterations, $[W]_j \sim \mathcal{N}(0, \Sigma_{0:H-1}^i)$, $j = 1, \dots, N_W$ are N_W sampled Gaussian noise sequences with scheduled covariance $\Sigma_h, h \in \{0, \dots, H-1\}$, $\beta_1, \beta_2 \in (0, 1)$ are annealing parameters controlling sampling covariance in line 6 of Alg. 1.

In contact-rich dexterous manipulation, the *feasible solution set* is typically narrow, so effective optimization requires a combination of *coarse search* to discover feasible contact modes and *fine refinement* to achieve stable contact. The annealed sampling covariance in Eq. (2) implements this exploration-exploitation trade-off: β_1 controls the rate across outer iterations (global-to-local search over optimization updates), and β_2 controls the rate along the prediction horizon (allocating more or less perturbation across timesteps). As the schedule progresses, the effective sampling radius transitions from broad exploration to targeted exploitation around promising trajectories. To further improve speed, we adopt an early stopping strategy [48], where optimization halts when improvements from further sampling become small.

C. Sampling for Physics-based Data Augmentation

Apart from generating feasible robot actions, another application of physics-based retargeting is the ability to systematically augment

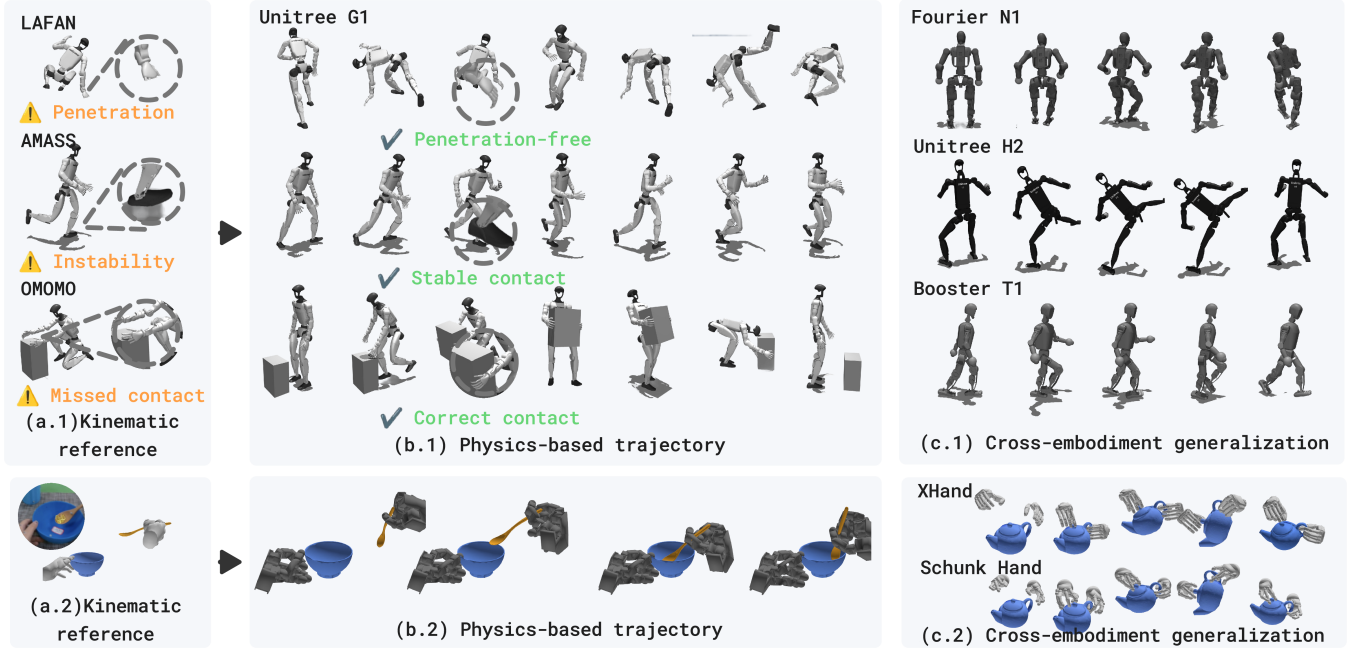


Fig. 3: **Sampling for Physics-based Retargeting on different robot embodiments.** This figure shows the open-loop rollout of the retargeted control on different robot embodiments. The retargeting process corrects artifacts such as foot penetration and slipping, enabling the robot to execute highly dynamic motions. The method can be applied to both dexterous manipulation and humanoid whole-body control tasks.

the retargeted data for trajectory robustness, data diversity and user intent preservation.

Trajectory Robustification. To bridge the gap from reconstructed demonstrations to real hardware, trajectories must be robust against unknown or misspecified dynamics (e.g., friction, contact compliance) and reconstruction noise (e.g., object mesh estimation error, pose estimation error) that could otherwise make a nominal plan infeasible. Our approach optimizes a control sequence with a pessimistic (min-max) objective over a bounded parameter set $\mathcal{D}$, similar to domain randomization (DR) method [49] but replacing expectation with worst-case cost to ensure universal feasibility:

$$J_{\text{rob}}(U) = \max_{d \in \mathcal{D}} J(U, d), \quad (4)$$

where $J(U, d)$ is the cost in Eq. (1a) under dynamics parameter d . In practice, $\mathcal{D}$ spans variations such as contact margin size, friction coefficients, and object mass. During optimization, each candidate sequence U_j is rolled out over a mini-batch $d_{1:K} \sim \mathcal{D}$ and evaluated by $\max_{k \leq K} J(U_j, d_k)$, following robust sampling-based control [50].

Geometric and Physics Augmentation. To generate diverse data, one human demonstration can be augmented into a diverse set of physically feasible actions which could be used for policy learning [51]. **SPIDER** supports generating diverse geometric variations of the objects while still grounding them with physics. For dexterous manipulation tasks, we perturb the initial object pose and scale to generate diverse interaction behaviors while the reference motion is kept the same. We can even directly replace the object mesh to generate new interaction behaviors. For instance, in Fig. 4, we augment the grasping motion of a coke to a cat toy simply by replacing the object mesh. Similarly, the same idea can be applied to human-object interaction for humanoid robot tasks. By modifying the object geometry to make it smaller and lighter, the robot can automatically adapt to the new object by lowering the body down and switching to single-hand pushing motion with smaller force. One can also change the terrain to stairs to transform a running

on flat ground motion into a running on stairs motion, which is challenging for kinematic-only retargeting because climbing stairs requires new contact patterns. A key advantage of physics-based retargeting is its force-awareness. By applying external forces to the robot when it is pulling a heavy object, the sampling-based solver can generate different behaviors responding to the force. For instance, if a large force (120N and 240N respectively in Fig. 4) is applied to pull the robot back, the generated trajectory will lean forward and move slower.

Human-preferred Contact Preservation. Due to the multi-modal nature of dexterous manipulation, the solution ambiguity could lead to unexpected contact behaviors with wrong reference motion as illustrated in Fig. 5. For gradient-based optimization methods, one can specify the contact mode in dynamics constraints to enforce the desired contact mode. Inspired by this idea, we apply a virtual constraint between intended contact points on the object and robot fingers, “sticking” the object to the target configuration in early stages and gradually relaxing this constraint in a curriculum-like fashion. For the k -th object-hand contact pair (e.g., handle-thumb), we define the current relative position as $p_{k,t}^{\text{robot,object}} = p_{k,t}^{\text{robot}} - p_{k,t}^{\text{object}}$ and the desired reference position as $p_{k,t}^{\text{robot,object,ref}} = p_{k,t}^{\text{robot,ref}} - p_{k,t}^{\text{object,ref}}$. The constraint is activated when the contact indicator $c_{k,t} = 1$, which occurs when the reference relative position is within the contact threshold: $c_{k,t} = 1(\|p_{k,t}^{\text{robot,object}} - p_{k,t}^{\text{robot,object,ref}}\|_2 \leq \epsilon_{\text{contact}})$. The constraint strength is controlled by a penalty parameter $\eta_i \rightarrow \infty$ when $i \rightarrow N$:

$$c_{k,t} \|p_{k,t}^{\text{robot,object}} - p_{k,t}^{\text{robot,object,ref}}\|_2^2 \leq \eta_i \quad (5)$$

The implementation details are described in Sec. III. Compared to previous methods [36], our formulation focuses on *relative* hand-object relationships instead of their absolute positions. The method requires annotations of the intended contact points and labels, which we detect automatically using the simulator’s collision detection on kinematic data.

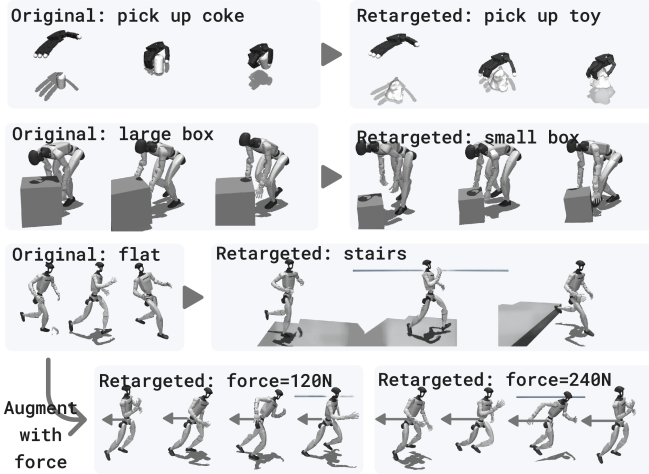


Fig. 4: **Geometric and physics augmentation.** We augment the retargeted data from a single demonstration into a diverse set of physically feasible actions. Here we demonstrate (a) generating motion with new object mesh for dexterous manipulation, (b) moving a lighter and smaller object for humanoid robot tasks, (c) adding stairs to the scene for humanoid running motion. (d) applying external forces to the robot when it is pulling a heavy object.

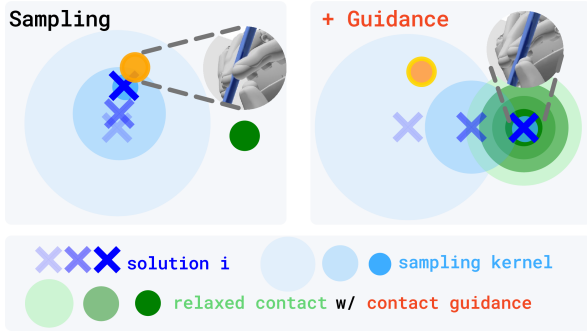


Fig. 5: **Human-preferred contact preservation.** Due to bad initial reference, the solution may converge to a different contact mode than the human demonstration. Contact guidance is applied to correct it with specified contact modes by making target contact mode easier to sample from. With virtual force applied between the object and the hand, the object is more likely to move towards the target contact region, guiding the sampling towards the desired contact mode.

IV. EXPERIMENTS

This section evaluates the generated motion quality by comparing with state-of-the-art retargeting baselines, testing with policy learning pipeline and deploying on physical robots. To provide a quantitative assessment of generated motion quality, we compare **SPIDER** with state-of-the-art retargeting baselines on dexterous manipulation in Sec. IV-A and humanoid whole-body control in Sec. IV-B. Lastly, we showcase the application of **SPIDER** with in-the-wild data and deploy it on physical robots in Sec. IV-C. Throughout, we adopt unified notation for tracking errors: e_t and e_r denote translation and rotation errors, e_j denotes joint angle error, e_{ft} denotes fingertip position error, and e_{arti} denotes articulation joint angle error for articulated objects. Full implementation details including datasets,

robot embodiments, simulator settings, and hyperparameters are provided in Sec. I.

A. Dexterous Manipulation Retargeting Results

Baselines. We compare against two RL-based physics retargeting methods, **ManipTrans** [21] and **DexMachina** [36], and a **Kinematic** retargeting baseline (fingertip-based IK, kinematically feasible but dynamically infeasible). Since RL-based methods train per-trajectory policies whereas **SPIDER** directly optimizes control sequences, this comparison serves as a performance reference. To isolate the effect of retargeting from confounds, each comparison aligns the simulator, dataset, hand, and reward setting with the corresponding baseline: we evaluate against **ManipTrans** on Oakink in Isaac Gym and against **DexMachina** on ARCTIC in Genesis, both using the 12-DoF Inspire hand, and additionally align the nominal base controller with **ManipTrans**’s universal base imitator; the kinematic comparison is run in MuJoCo. Full protocol details are in Sec. II.

Metrics. We report two metrics. **ADD-AUC** measures object tracking quality without committing to a single success threshold: we sample keypoints on the object mesh, transform them by both the achieved and the reference object poses at every frame, and take the area under the per-frame success-rate curve as the mean keypoint distance threshold sweeps from 0 to 10 cm (normalized so the metric lies in $[0, 100]\%$). Because it is computed on transformed mesh keypoints, it captures both translational and rotational error in a single number; the full definition is in Sec. I. We use **ADD-AUC** rather than a thresholded success rate because success is defined differently across baseline papers and neither has released a public evaluation protocol. **FPS** measures retargeting throughput, defined as the total number of retargeted frames divided by the total retargeting wall-clock time (higher is better).

Comparison Results. Table I aggregates all dexterous-hand comparisons. Against the RL-based baselines, **SPIDER** achieves comparable or better **ADD-AUC** while generating trajectories over an order of magnitude faster, since it optimizes control sequences with a parallel simulator instead of training a policy per trajectory. Against **Kinematic** retargeting, **SPIDER** more than doubles the **ADD-AUC**, and contact guidance (**SPIDER+C**, with contacts detected by MuJoCo’s contact detector) improves it further. The high variance is because results are aggregated across tasks of varying difficulty; detailed per-task, per-hand tracking performance is reported in Sec. II.

Physics-based Retargeting for Dexterous Grasping Policy Training. **SPIDER** is complementary to RL algorithms: it provides an already-feasible reference motion, making the tracking problem substantially easier. By removing physical artifacts such as hand-object penetration and the object drifting off the hand, it makes RL training both faster and better-converged. We validate this on a dexterous grasping task with a 7-DoF arm and a 22-DoF five-fingered hand, where the policy tracks the reference to pick up a cup. The reference is collected from in-the-wild data of a human pouring water from a bottle, reconstructed with the pipeline described in Sec. IV-C; due to occlusion, the kinematic motion sometimes exhibits the cup flying out of the hand, while physics grounding with **SPIDER** removes these artifacts and yields a feasible reference. As shown in Fig. 6, over 7 seeds with identical hyperparameters, the policy trained with **SPIDER** references completes the task (0.99 ± 0.01 task progress) while the kinematic baseline stalls (0.07 ± 0.11), converging within roughly 100M environment steps. Implementation details are provided in Sec. I-E.

B. Humanoid Whole-Body Control Retargeting Results

Baselines and Reference Methods. For humanoid whole-body loco-manipulation tasks, there is no existing physics-based retargeting method since the kinematic motion is directly used as the reference motion for RL training. Thus, we compare **SPIDER** with the state-of-the-art kinematics-based retargeting method general

	Method	Simulator	Dataset	Hand	AUC (%) $\uparrow$	FPS $\uparrow$
vs. RL	ManipTrans	Isaac Gym	Oakink	Inspire	19.4 ± 27.5	0.1
	SPIDER	Isaac Gym	Oakink	Inspire	22.3 ± 28.1	2.5
	DexMachina	Genesis	ARCTIC	Inspire	41.0 ± 25.3	0.05
	SPIDER	Genesis	ARCTIC	Inspire	43.6 ± 29.8	1.5
vs. kin.	Kinematic	MuJoCo	Oakink	mixed	11.0 ± 11.1	248.9
	SPIDER	MuJoCo Warp	Oakink	mixed	48.8 ± 24.9	6.1
	SPIDER+C	MuJoCo Warp	Oakink	mixed	56.9 ± 24.7	0.4
	Kinematic	MuJoCo	GigaHands	mixed	24.1 ± 22.3	244.2
	SPIDER	MuJoCo Warp	GigaHands	mixed	56.9 ± 20.6	5.9
	SPIDER+C	MuJoCo Warp	GigaHands	mixed	57.7 ± 23.9	0.5

TABLE I: **Dexterous manipulation retargeting comparison.** Each block aligns the simulator, dataset, and hand with the corresponding baseline for a fair comparison (see text). For ARCTIC, ADD-AUC is computed on the object base pose since the objects are articulated. The kinematic comparison is aggregated over five hands (Allegro, Ability, Inspire, Schunk, XHand; “mixed”), with the full per-hand breakdown in Sec. II. FPS is measured on a single NVIDIA L40S GPU; kinematic IK trades tracking quality for speed, while **SPIDER** attains far higher accuracy at interactive rates.

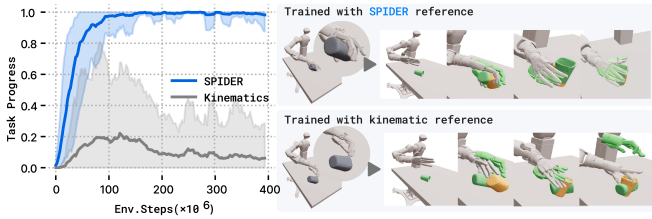


Fig. 6: **RL Policy Training for Dexterous Grasping.** Task progress during training with **SPIDER**-retargeted references versus kinematic-only references on the cup-grasping task, collected from in-the-wild human demonstration. Curves show mean over 7 seeds; shaded regions show min-max. The kinematic reference contains physical artifacts (e.g., the cup drifting out of the hand due to occlusion), so the policy trained on it fails to complete the task, while **SPIDER** references lead to fast and consistent convergence. For trajectory, we displayed both learned motion (yellow) and reference motion (green).

motion retargeting (**GMR**) [52], where we report the open-loop rollout motion tracking error. Since humanoid robots are unstable systems, a stabilizer policy is required to track the retargeted motion, where reference motion quality is crucial. To evaluate the importance of physics grounding, we train a policy with both the original kinematics-only motion and the physics-grounded motion from **SPIDER** and report their training curves given the same training settings.

Comparison Against Kinematics-Based Baselines. As summarized in Tab. II, **SPIDER** achieves lower open-loop tracking error thanks to physics grounding with minimal speed degradation, including significantly better object tracking on the OMOMO dataset.

Physics-based Retargeting for RL Policy Training. We compare a humanoid whole-body loco-manipulation policy, where the robot needs to move an object from one location to another with the adopted framework from [53]. We follow the standard practice of training PPO while removing the auxiliary contact reward for simplicity. Figure 7 shows the training progress of the original kinematics-only motion along with the retargeted motion with

Dataset	Method	Robot			Object		
		e_j (deg) $\downarrow$	e_t (cm) $\downarrow$	e_r (deg) $\downarrow$	e_t (cm) $\downarrow$	e_r (deg) $\downarrow$	FPS $\uparrow$
LAFAN1	GMR	1.08	2.01	2.40	–	–	35.2
	SPIDER	0.58	0.11	0.07	–	–	23.1
AMASS	GMR	6.2	4.1	18.7	–	–	37.2
	SPIDER	0.75	0.23	0.08	–	–	22.0
OMOMO	GMR	0.57	1.43	2.28	1.18	1.68	28.0
	SPIDER	0.83	0.20	0.17	0.18	0.06	19.6

TABLE II: **Tracking on Humanoid Datasets.** Robot columns report body tracking errors. Object columns report object pose tracking (OMOMO only). FPS: frames/sec.

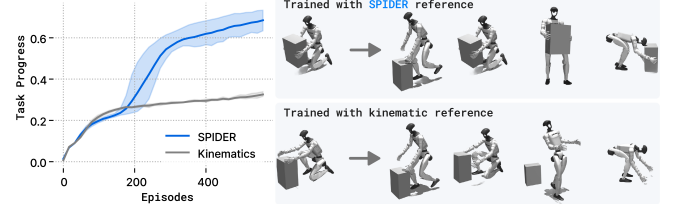


Fig. 7: **RL Policy Training on Humanoid Robots.** In reference motion, the robot will first pick up the box and then place it on the floor. Due to the motion capture error, the original reference motion missed the contact with the object. After retargeting with **SPIDER**, the contact is corrected and a feasible feedforward control is provided to assist the policy learning. Left: task progress of the original human motion, demonstrating how many percentage of the target motion is achieved. Right: resulted policy after training.

SPIDER. The original kinematics-only motion is offset from the object position, thus missing the contact. When training with the original kinematics-only motion, the robot fails to grasp the object and only achieves body tracking goal. On the other hand, the policy trained with **SPIDER** is not only converged faster, but also achieves better object tracking performance. This again highlights the importance of physics grounding for reference motion.

C. Deployment Results

Replay Generated Motion on Physical Robots. To demonstrate the dynamical feasibility of our retargeted trajectories, we deploy **SPIDER** on a physical system consisting of a 7-DoF Franka Emika Panda with an Allegro hand with the robustification method in Sec. III-C. The arm is controlled using operational space control (OSC), while the hand directly executes the optimized joint actions from our method. We evaluate trajectory feasibility on four dexterous manipulation tasks—rotating a light bulb, manipulating a small spoon, playing a guitar, and unplugging a charger—each demanding precise finger coordination and showcasing the practical applicability of our retargeting approach. Representative execution sequences are shown in Fig. 8. We note an inherent limitation of this open-loop execution: while the robustification and randomization in Sec. III-C make the trajectory tolerant to moderate perturbations and model mismatch, they cannot compensate for large deviations in the object’s initial position. If the object is placed significantly off from the reference initial pose, the open-loop rollout has no feedback mechanism to recover, and the execution fails; closing the loop with a learned tracking policy (e.g., Sec. I-D) is the natural remedy.

Using In-the-Wild Data. To demonstrate the flexibility of **SPIDER** for upstreaming data, we demonstrate the pipeline of converting single RGB camera video into executable robot trajectories. The pipeline consists of: (1) 3D mesh reconstruction with Trellis [16], (2) hand pose estimation with HAMER [15]

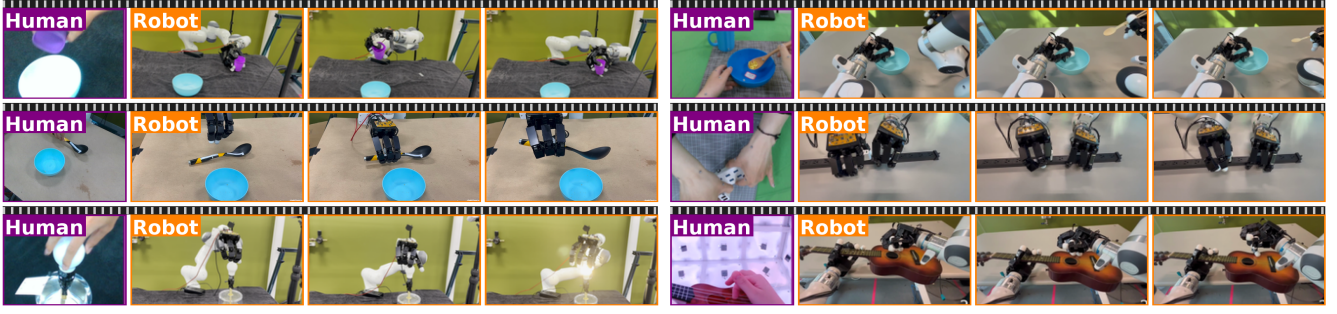


Fig. 8: **Open-loop rollout with robustification.** Left: showing single-arm tasks including pouring from a cup, picking a spoon, and rotating a light bulb. Right: showing dual-arm tasks including scooping a bowl, unplugging a charger, and playing a guitar. The retargeted trajectories are executed directly on the physical robot.

to obtain MANO parameters, and (3) object pose tracking with FoundationPose [17] for 6D trajectories. Due to the occlusion and limited resolution, the human and object motion are more noisy than the dataset, which requires more robust physics grounding. As shown in Fig. 9, we validate on single-hand manipulation tasks like pouring with a cup and installing a bolt. Real-world noise, artifacts, and penetrations are corrected through physics grounding in **SPIDER**, and the retargeted trajectories are directly executable on physical robots. One limitation is that, since it is an open-loop rollout, it can only handle a narrow range of initial states and disturbances. We leave the closed-loop control for future work.

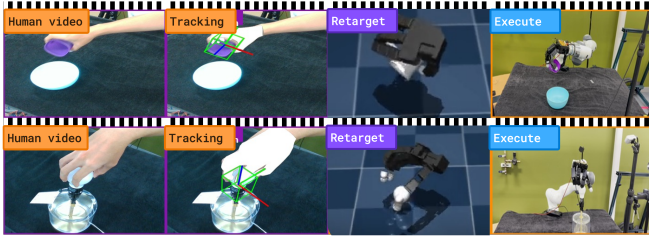


Fig. 9: **Retargeting from a Single RGB Camera.** Starting from a single RGB observation of a user manipulating an object, we first reconstruct the 3D scene with hand and object meshes, which often contain reconstruction artifacts and penetrations. Our method then retargets this noisy data into a physically plausible robot trajectory with corrected contacts. Finally, the retargeted trajectory is successfully executed on the physical robot, demonstrating the pipeline’s robustness to real-world data quality issues.

D. Limitations and Common Failure Cases

SPIDER has several failure modes. (1) **Perception errors**: inaccurate upstream reconstruction (HAMER, FoundationPose, Trellis) can introduce erroneous reference trajectories; large misalignments may cause the optimizer to converge to a semantically incorrect contact mode despite physical consistency. (2) **Severe occlusion**: heavy occlusion produces large temporal gaps in the reference, degrading retargeting quality; temporal consistency filtering is a promising mitigation. (3) **Open-loop sensitivity**: the generated trajectory is valid only near the aligned initial state—any initial pose mismatch leads to growing execution errors; closed-loop policy distillation is left as future work. (4) **Contact mode ambiguity**: without contact guidance, sampling may converge to an alternative valid grasp; **SPIDER+C** resolves most such cases at the cost of contact annotations.

V. CONCLUSION AND FUTURE WORK

This paper introduces **SPIDER**, a general, flexible and efficient physics-based retargeting framework that enables large-scale data transfer from human data. Our framework achieves competitive performance compared to state-of-the-art methods while being an order of magnitude faster. Despite its generality, **SPIDER**’s performance depends on the quality of reconstructed 3D human-object interaction data; noisy meshes and motion can yield degraded trajectories. Future work includes improving the robustness of the framework to noisy data; and distilling the framework output into a policy for closed-loop control.

REFERENCES

- [1] M. J. Kim, K. Pertsch, S. Karamcheti, T. Xiao, A. Balakrishna, S. Nair, R. Rafailov, E. Foster, G. Lam, P. Sanketi, Q. Vuong, T. Kollar, B. Burchfiel, R. Tedrake, D. Sadigh, S. Levine, P. Liang, and C. Finn, “OpenVLA: An Open-Source Vision-Language-Action Model,” in *CoRL*, 2024.
- [2] R. Team, “Rdt2: Enabling zero-shot cross-embodiment generalization by scaling up umi data,” 2025. [Online]. Available: <https://github.com/thu-ml/RDT2>
- [3] Z.-H. Yin, C. Wang, L. Pineda, F. Hogan, K. Bodduluri, A. Sharma, P. Lancaster, I. Prasad, M. Kalakrishnan, J. Malik, M. Lambeta, T. Wu, P. Abbeel, and M. Mukadam, “DexterityGen: Foundation Controller for Unprecedented Dexterity,” in *RSS*, 2025.
- [4] Y. Zhong, X. Huang, R. Li, C. Zhang, Z. Chen, T. Guan, F. Zeng, K. N. Lui, Y. Ye, Y. Liang, Y. Yang, and Y. Chen, “DexGraspVLA: A Vision-Language-Action Framework Towards General Dexterous Grasping,” *arXiv:2502.20900*, 2025.
- [5] H. Qi, B. Yi, S. Suresh, M. Lambeta, Y. Ma, R. Calandra, and J. Malik, “General in-hand object rotation with vision and touch,” in *CoRL*, 2023.
- [6] J. Li, X. Cheng, T. Huang, S. Yang, R.-Z. Qiu, and X. Wang, “AMO: Adaptive Motion Optimization for Hyper-Dexterous Humanoid Whole-Body Control,” in *RSS*, 2025.
- [7] Y. Ze, Z. Chen, J. P. Araújo, Z.-a. Cao, X. B. Peng, J. Wu, and C. K. Liu, “TWIST: Teleoperated Whole-Body Imitation System,” in *CoRL*, 2025.
- [8] H. Zhang, S. Hu, Z. Yuan, and H. Xu, “DOGlove: Dexterous Manipulation with a Low-Cost Open-Source Haptic Force Feedback Glove,” in *RSS*, 2025.
- [9] M. Xu, H. Zhang, Y. Hou, Z. Xu, L. Fan, M. Veloso, and S. Song, “DexUMI: Using Human Hand as the Universal Manipulation Interface for Dexterous Manipulation,” in *CoRL*, 2025.
- [10] X. Zhan, L. Yang, Y. Zhao, K. Mao, H. Xu, Z. Lin, K. Li, and C. Lu, “OAKINK2: A Dataset of Bimanual Hands-Object Manipulation in Complex Task Completion,” in *CVPR*, 2024.
- [11] N. Mahmood, N. Ghorbani, N. F. Troje, G. Pons-Moll, and M. J. Black, “AMASS: Archive of Motion Capture as Surface Shapes,” in *ICCV*, 2019.
- [12] J. Ren, P. Sundaresan, D. Sadigh, S. Choudhury, and J. Bohg, “Motion Tracks: A Unified Representation for Human-Robot Transfer in Few-Shot Imitation Learning,” *Computer Graphics and Interactive Techniques*, 2025.

- [13] R. Hoque, P. Huang, D. J. Yoon, M. Sivapurapu, and J. Zhang, "EgoDex: Learning Dexterous Manipulation from Large-Scale Ego-centric Video," *arXiv:2505.11709*, 2025.
- [14] S. Goel, G. Pavlakos, J. Rajasegaran, A. Kanazawa, and J. Malik, "Humans in 4D: Reconstructing and Tracking Humans with Transformers," in *ICCV*, 2023.
- [15] G. Pavlakos, D. Shan, I. Radosavovic, A. Kanazawa, D. Fouhey, and J. Malik, "Reconstructing Hands in 3D with Transformers," in *CVPR*, 2024.
- [16] J. Xiang, Z. Lv, S. Xu, Y. Deng, R. Wang, B. Zhang, D. Chen, X. Tong, and J. Yang, "Structured 3D Latents for Scalable and Versatile 3D Generation," in *CVPR*, 2025.
- [17] B. Wen, W. Yang, J. Kautz, and S. Birchfield, "FoundationPose: Unified 6D Pose Estimation and Tracking of Novel Objects," in *CVPR*, 2024.
- [18] D. Reda, J. Won, Y. Ye, M. van de Panne, and A. Winkler, "Physics-based Motion Retargeting from Sparse Inputs," *arXiv:2307.01938*, 2023.
- [19] Y. Qin, H. Su, and X. Wang, "From One Hand to Multiple Hands: Imitation Learning for Dexterous Manipulation from Single-Camera Teleoperation," in *IROS*, 2022.
- [20] Z.-H. Yin, C. Wang, L. Pineda, K. Bodduluri, T. Wu, P. Abbeel, and M. Mukadam, "Geometric Retargeting: A Principled, Ultrafast Neural Hand Retargeting Algorithm," in *IROS*, 2025.
- [21] K. Li, P. Li, T. Liu, Y. Li, and S. Huang, "ManipTrans: Efficient Dexterous Bimanual Manipulation Transfer via Residual Learning," in *CVPR*, 2025.
- [22] T. G. W. Lum, O. Y. Lee, C. K. Liu, and J. Bohg, "Crossing the Human-Robot Embodiment Gap with Sim-to-Real RL using One Human Demonstration," in *CoRL*, 2025.
- [23] S. Park, H. Bharadhwaj, and S. Tulsiani, "DemoDiffusion: One-Shot Human Imitation using pre-trained Diffusion Policy," *arXiv:2506.20668*, 2025.
- [24] J. Li, Y. Zhu, Y. Xie, Z. Jiang, M. Seo, G. Pavlakos, and Y. Zhu, "OKAMI: Teaching Humanoid Robots Manipulation Skills through Single Video Imitation," in *CoRL*, 2024.
- [25] G. Papagiannis, N. D. Palo, P. Vitiello, and E. Johns, "R+X: Retrieval and Execution from Everyday Human Videos," in *ICRA*, 2025.
- [26] V. Liu, A. Adeniji, H. Zhan, S. Haldar, R. Bhirangi, P. Abbeel, and L. Pinto, "EgoZero: Robot Learning from Smart Glasses," *arXiv:2505.20290*, 2025.
- [27] C. Wang, L. Fan, J. Sun, R. Zhang, L. Fei-Fei, D. Xu, Y. Zhu, and A. Anandkumar, "MimicPlay: Long-Horizon Imitation Learning by Watching Human Play," in *CoRL*, 2023.
- [28] H. Bharadhwaj, R. Mottaghi, A. Gupta, and S. Tulsiani, "Track2act: Predicting point tracks from internet videos enables generalizable robot manipulation," in *ECCV*, 2024.
- [29] K. Shaw, S. Bahl, and D. Pathak, "VideoDex: Learning Dexterity from Internet Videos," in *CoRL*, 2022.
- [30] A. Wu, R. Wang, S. Chen, C. Eppner, and C. K. Liu, "One-Shot Transfer of Long-Horizon Extrinsic Manipulation Through Contact Retargeting," *arXiv:2404.07468*, 2024.
- [31] Y. Qin, W. Yang, B. Huang, K. V. Wyk, H. Su, X. Wang, Y.-W. Chao, and D. Fox, "AnyTeleop: A General Vision-Based Dexterous Robot Arm-Hand Teleoperation System," in *RSS*, 2023.
- [32] A. S. Lakshminpathy, J. K. Hodgins, and N. S. Pollard, "Kinematic Motion Retargeting for Contact-Rich Anthropomorphic Manipulations," *ACM Transactions on Graphics*, 2025.
- [33] S. Nakaoka, A. Nakazawa, F. Kanehiro, K. Kaneko, M. Morisawa, and K. Ikeuchi, "Task model of lower body motion for a biped humanoid robot to imitate human dances," in *IROS*, 2005.
- [34] L. Yang, H. J. T. Suh, T. Zhao, B. P. Graesdal, T. Kelestemur, J. Wang, T. Pang, and R. Tedrake, "Physics-Driven Data Generation for Contact-Rich Manipulation via Trajectory Optimization," in *RSS*, 2025.
- [35] Z. Si, J. E. Chen, M. E. Karagozler, A. Bronars, J. Hutchinson, T. Lampe, N. Gileadi, T. Howell, S. Saliceti, L. Barczyk, I. O. Correa, T. Erez, M. Shridhar, M. F. Martins, K. Bousmalis, N. Heess, F. Nori, and M. Bauza, "Exostart: Efficient learning for dexterous manipulation with sensorized exoskeleton demonstrations," 2025. [Online]. Available: <https://arxiv.org/abs/2506.11775>
- [36] Z. Mandi, Y. Hou, D. Fox, Y. Narang, A. Mandlekar, and S. Song, "DexMachina: Functional Retargeting for Bimanual Dexterous Manipulation," *arXiv:2505.24853*, 2025.
- [37] X. Liu, K. Lyu, J. Zhang, T. Du, and L. Yi, "QuasiSim: Parameterized Quasi-Physical Simulators for Dexterous Manipulations Transfer," in *ECCV*, 2024.
- [38] P.-T. De Boer, D. P. Kroese, S. Mannor, and R. Y. Rubinstein, "A Tutorial on the Cross-Entropy Method," *Annals of Operations Research*, 2005.
- [39] T. Salimans, J. Ho, X. Chen, S. Sidor, and I. Sutskever, "Evolution Strategies as a Scalable Alternative to Reinforcement Learning," *arXiv:1703.03864*, 2017.
- [40] P. I. Frazier, "A Tutorial on Bayesian Optimization," *arXiv:1807.02811*, 2018.
- [41] G. Williams, P. Drews, B. Goldfain, J. M. Rehg, and E. A. Theodorou, "Aggressive driving with model predictive path integral control," in *ICRA*, 2016.
- [42] H. Xue, C. Pan, Z. Yi, G. Qu, and G. Shi, "Full-Order Sampling-Based MPC for Torque-Level Locomotion Control via Diffusion-Style Annealing," in *ICRA*, 2025.
- [43] T. Howell, N. Gileadi, S. Tunyasuvunakool, K. Zakka, T. Erez, and Y. Tassa, "Predictive Sampling: Real-time Behaviour Synthesis with MuJoCo," *arXiv:2212.00541*, 2022.
- [44] A. H. Li, P. Culbertson, V. Kurtz, and A. D. Ames, "DROP: Dexterous Reorientation via Online Planning," in *ICRA*, 2025.
- [45] T. Kim, G. Park, K. Kwak, J. Bae, and W. Lee, "Smooth Model Predictive Path Integral Control without Smoothing," *RA-L*, 2022.
- [46] S. Mannor, R. Rubinstein, and Y. Gat, "The Cross Entropy Method for Fast Policy Search," in *ICML*, 2003.
- [47] C. Pan, Z. Yi, G. Shi, and G. Qu, "Model-Based Diffusion for Trajectory Optimization," in *NeurIPS*, 2024.
- [48] J. Kamat, J. Ortiz-Haro, M. Toussaint, F. T. Pokorny, and A. Orthey, "BITKOMO: Combining Sampling and Optimization for Fast Convergence in Optimal Motion Planning," in *IROS*, 2022.
- [49] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, "Domain Randomization for Transferring Deep Neural Networks from Simulation to the Real World," in *IROS*, 2017.
- [50] G. Williams, B. Goldfain, P. Drews, K. Saigol, J. Rehg, and E. Theodorou, "Robust Sampling Based Model Predictive Control with Sparse Objective Information," in *RSS*, 2018.
- [51] A. Mandlekar, S. Nasiriany, B. Wen, I. Akinola, Y. Narang, L. Fan, Y. Zhu, and D. Fox, "Mimicgen: A data generation system for scalable robot learning using human demonstrations," *arXiv:2310.17596*, 2023.
- [52] J. P. Araujo, Y. Ze, P. Xu, J. Wu, and C. K. Liu, "Retargeting matters: General motion retargeting for humanoid motion tracking," *arXiv:2510.02252*, 2025.
- [53] H. Weng, Y. Li, N. Sobanbabu, Z. Wang, Z. Luo, T. He, D. Ramanan, and G. Shi, "HDMI: Learning interactive humanoid whole-body control from human videos," *arXiv:2509.16757*, 2025.
- [54] R. Fu, D. Zhang, A. Jiang, W. Fu, A. Funk, D. Ritchie, and S. Sridhar, "GigaHands: A Massive Annotated Dataset of Bimanual Hand Activities," in *CVPR*, 2025.
- [55] J. Li, J. Wu, and C. K. Liu, "Object motion guided human motion synthesis," *ACM Transactions on Graphics*, 2023.
- [56] F. G. Harvey, M. Yurick, D. Nowrouzezahrai, and C. Pal, "Robust motion in-betweening," in *SIGGRAPH*, 2020.
- [57] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *1707.06347*, 2017.
- [58] Z. Fan, O. Taheri, D. Tzionas, M. Kocabas, M. Kaufmann, M. J. Black, and O. Hilliges, "ARCTIC: A Dataset for Dexterous Bimanual Hand-Object Manipulation," Apr. 2023.

APPENDIX I IMPLEMENTATION DETAILS

A. Experimental Setup

Dataset Selection. For dexterous manipulation tasks, we evaluate on 3 bimanual manipulation datasets: GigaHands [54], Oakink [10] and ARCTIC [55]. For humanoid tasks, we choose commonly used LAFAN1 [56], AMASS [11] for locomotion and OMOMO [55] dataset for human-robot interaction. To the best of our knowledge, **SPIDER is the first universal retargeting method that can handle both dexterous manipulation and humanoid whole-body control tasks at this scale.**

Robot Embodiments. For dexterous manipulation, we evaluate across 5 different robotic hands to demonstrate the generalizability of our method: Allegro, XHand, Inspire, Ability, and Schunk. On humanoid whole-body control tasks, we use Unitree G1, Unitree H1-2, Fourier N1 and Booster T1 as the target embodiment. These platforms exhibit significant variations in degrees of freedom, dimensions, and finger configurations (see Tab. III), showcasing our method’s cross-embodiment capabilities. Figure 10 shows some example trajectories generated by **SPIDER** on different hands.

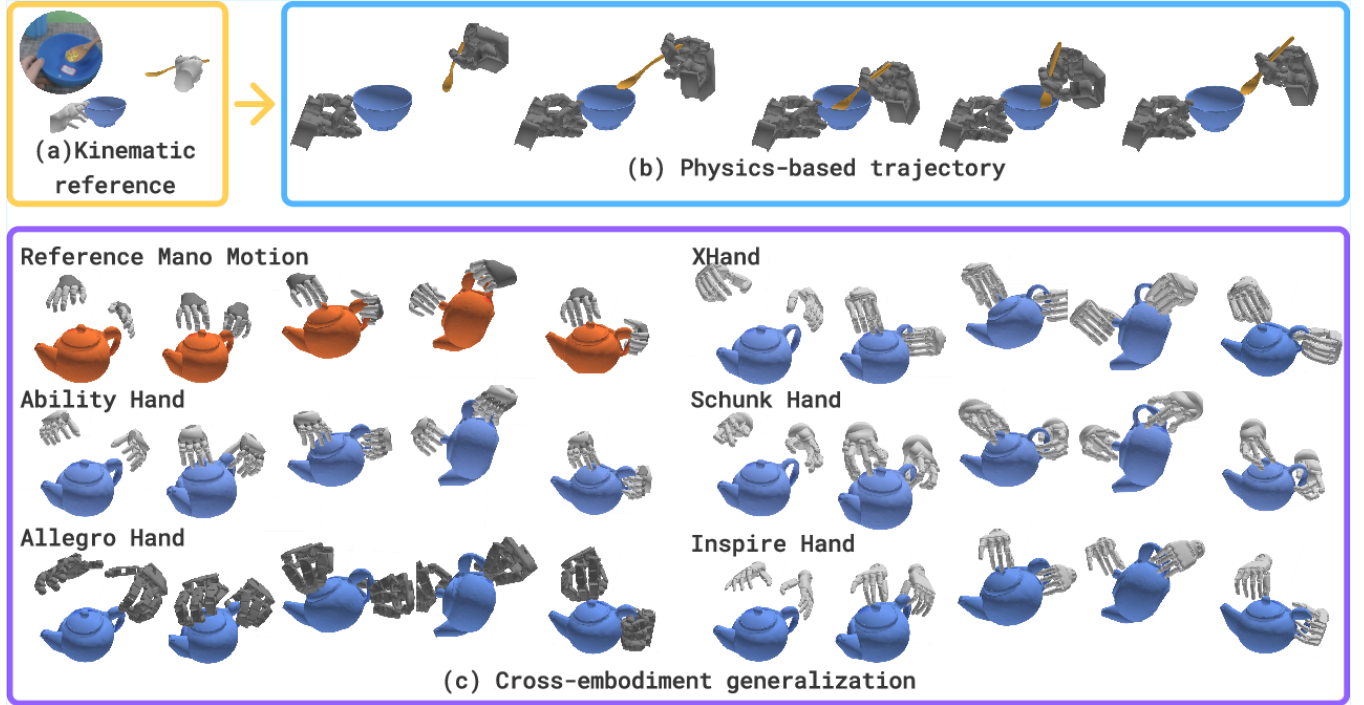


Fig. 10: **Retargeted dexterous manipulation trajectories on different hands.** Starting with an infeasible motion with penetration and infeasible contact, **SPIDER** grounds it with physics. Different hands exhibit different behaviors with the same sampling parameters.

Hand	DOF	Width (cm)	Length (cm)	Fingers
Allegro	16	12.0	26.0	4
XHAND	12	9.2	19.5	5
Inspire	12	8.8	15.6	5
Ability	12	7.5	14.6	5
Schunk SVH	20	9.2	24.2	5

TABLE III: **Specifications of robotic hands used in evaluation.** The significant variations in DoF, dimensions, and finger count demonstrate the cross-embodiment generalizability of our approach. We employ a simulated 12-DoF configuration [21] of the Inspire and Ability hands, removing the joint constraints compared to their real-world versions.

Evaluation Metrics. In our own ablation study where we validate the effect of our annealing and contact guidance design, we report tracking performance on the robot and object motion. For object tracking error, we compute per-step averaged rotation error $E_{\text{rot}} = \frac{1}{T} \sum_{t=1}^T \arccos(2\langle q_{\text{obj},t}, q_{\text{obj},t}^{\text{ref}} \rangle^2 - 1)$ and position error $E_{\text{pos}} = \frac{1}{T} \sum_{t=1}^T \|p_{\text{obj},t} - p_{\text{obj},t}^{\text{ref}}\|_2$, where T

Supported Robot	(a) Dexterous hand					(b) Humanoid			
									
Robot	Allegro	Schunk	Xhand	Inspire	Ability	Unitree H1-2	Unitree G1	Fourier N1	Booster T1
Length	26.0cm	24.2cm	19.5cm	15.6cm	14.6cm	1.8m	1.3m	1.3m	1.2m
Dof	16	20	12	12	12	26	29	23	23

Fig. 11: **Specifications of robots used in evaluation.** **SPIDER** supports both dexterous hands and humanoid robots. The significant variations in DoF, dimensions, and finger count demonstrate the cross-embodiment generalizability of our approach. We employ a simulated 12-DoF configuration [21] of the *Inspire* and *Ability* hands, removing the joint constraints compared to their real-world versions.

is the number of timesteps, $\langle \cdot, \cdot \rangle$ is the dot product, $q_{obj,t}$ and $p_{obj,t}$ are the object quaternion and position at timestep t , respectively. We define success only on the object motion since the goal is to move the object rather than to precisely track the robot hand. Task success is defined using: (1) the rotation error $E_{rot} < 0.5$ rad, and (2) the average per-step translation error $E_{pos} < 0.1$ m. Since a single success threshold is sensitive to its choice, we additionally report ADD-AUC10, following the ADD-AUC protocol [36]: we sample 512 keypoints on the object mesh surface, transform them by both the achieved and the reference object poses at every frame, and compute the mean per-keypoint L2 distance (the classic ADD metric for rigid bodies). The metric (denoted ADD-AUC, or ADD-AUC10 to make the 10 cm range explicit) is the area under the per-frame success-rate curve as the distance threshold sweeps from 0 to 10 cm, with the threshold axis normalized so the metric lies in $[0, 100]\%$ (higher is better). This threshold-free metric summarizes tracking quality without committing to a single cutoff and is sensitive to both translational and rotational errors through the mesh keypoints. For humanoid locomotion, the joint tracking error, pelvis position error and pelvis orientation error are computed.

When comparing with baselines, we use the same evaluation metrics in the original papers to make a fair comparison. Specifically, we follow the same evaluation setting as **ManipTrans** [21] and **DexMachina** [36] with their published evaluation protocol.

B. Data Preprocessing

We extract a 21D keypoint representation per hand using 3D fingertip positions and 6D wrist pose from MANO. An IK solver maps these keypoints to robot-specific joint positions by minimizing $\mathcal{L}_{IK} = \sum_{i=1}^n \|\mathbf{p}_i^{\text{robot}} - \mathbf{p}_i^{\text{human}}\|^2 + 0.1 \|\mathbf{R}_{\text{wrist}}^{\text{robot}} - \mathbf{R}_{\text{wrist}}^{\text{human}}\|_F^2$. Trajectories are resampled at 50 Hz and low-pass filtered at 10 Hz.

C. Hyperparameters and Setup

All simulations run at 100 Hz physics and 50 Hz control in MuJoCo Warp (most datasets) and Genesis (ARCTIC dataset, as in DexMachina [36]). Ablations and speed tests use RTX 4090 GPUs; dataset generation uses H100 GPUs. All hyperparameters are listed in Table IV.

Automatic difficulty boosting. In practice, **SPIDER** requires little per-task hyperparameter tuning because per-step failures are handled by an automatic boosting mechanism. When the optimizer fails to reach the error tolerance at a step, the system retries from the last checkpoint with a boosted budget: each boost level doubles the number of optimization iterations and scales the sampling noise by $\sqrt{2}$ (both relative to the original configuration), and the noise-decay rate is re-derived so that the final-iteration noise still matches the configured target. If local retries are exhausted, the system reverts to an earlier checkpoint and re-optimizes that segment with the boosted budget before continuing. This acts as an automatic per-step hyperparameter adaptation: easy segments are solved cheaply with the default budget, while hard segments (e.g., contact-rich transitions) automatically receive more samples and wider exploration, so the iteration count and noise scale do not need to be tuned for the hardest step in advance.

D. Retargeting for RL Policy Training

This section describes the training details for the RL policy training experiment in the main paper. We use the PPO algorithm [57] and port the implementation from [53].

Rewards. We use the following rewards:

- Body tracking reward: tracking humanoid pelvis, hand and legs body motion.
- Object tracking reward: tracking the object motion.
- Action rate penalty: penalize the action rate to avoid jittering.

TABLE IV: **SPIDER** hyperparameters for Oakink and GigaHands retargeting with MuJoCo Warp Simulator. Values that differ from the default are shown per dataset.

Parameter	Oakink	Oakink (w/ guidance)	GigaHands	GigaHands (w/ guidance)
<i>Simulation</i>				
Sim. timestep Δt_{sim}	0.01 s	0.005 s	0.01 s	0.005 s
Control timestep Δt_{ctrl}	0.4 s	0.05 s	0.4 s	0.2 s
Knot spacing Δt_{knot}	0.4 s	0.1 s	0.4 s	0.4 s
<i>MPC / CEM Optimization</i>				
Planning horizon H	2.0 s	0.4 s	1.6 s	1.6 s
Horizon steps	200	80	160	320
Parallel samples N_w	1024	1024	1024	1024
Max CEM iterations N	32	32	32	32
CEM temperature τ	1.0	1.0	1.0	1.0
Early stop threshold ϵ	0.01	0.0	0.01	0.0
Early stop patience	2	1	1	1
<i>Noise Schedule</i>				
Initial noise σ_{first}	0.5	1.0	0.5	1.0
Final noise (horizon end) σ_{last}	1.0	1.0	1.0	1.0
Noise decay β	0.931	0.963	0.931	0.963
Joint noise scale	0.03	0.02	0.03	0.15
Wrist pos noise scale	0.01	0.005	0.03	0.01
Wrist rot noise scale	0.01	0.01	0.01	0.01
<i>Reward Weights</i>				
Base pos $w_{\text{pos}}^{\text{base}}$	0.1	0.3	0.1	0.03
Base rot $w_{\text{rot}}^{\text{base}}$	0.03	0.3	0.03	0.03
Fingertip pos w_{pos}	1.0	1.0	1.0	1.0
Fingertip rot w_{rot}	0.3	1.0	0.3	1.0
<i>Contact Guidance (w/ guidance only)</i>				
Contact guidance	✗	✓	✗	✓
Guidance decay γ_g	—	0.8	—	0.5
Pos actuator gain	—	10.0	—	2.0
Rot actuator gain	—	0.3	—	0.3

We remove the contact reward, self-collision penalty as well as the auxiliary contact reward proposed in the paper to evaluate the reference motion retargeting quality.

Termination. We terminate the training when the average body tracking error is greater than 10 cm and the average object tracking error is greater than 10 cm.

E. Retargeting for Dexterous Grasping

This section describes the training details for the dexterous grasping policy experiment in the main paper. **SPIDER** is complementary to RL algorithms: it provides an already-feasible reference motion, so the policy only needs to track it rather than discover the behavior from scratch. The task uses a right arm (7-DoF) paired with a 22-DoF dexterous hand (5 fingers, 29 DoF total) grasping a cup on a table. The reference is collected from an in-the-wild human demonstration and physics-grounded with **SPIDER** (Sec. IV-C). The **SPIDER** nominal feedforward control is used both to initialize the policy targets and as the reference for the tracking reward; we do *not* use it as a residual control term, though such a variant is also possible.

Scene and simulation. The scene consists of the right arm mounted at a fixed base, the object (mesh loaded from the demonstration manifest) as a dynamic rigid body, and a table with its top surface at 0.75 m. The environment runs at 20 Hz control over a 120 Hz PhysX simulation (decimation 6; TGS solver with 16 position iterations, 2 mm contact offset), with episodes of 12 s (≈ 240 policy steps) and 4096 parallel environments. The arm uses per-joint PD gains identified from the physical system.

Action space (28-D). The policy outputs a 28-D vector in $[-1, 1]$ that is scaled by per-step limits and accumulated onto persistent wrist and finger targets (delta control). The 6-D wrist delta (position ± 1 cm per step, rotation $\pm 4.5^\circ$ per step as axis-angle) drives a nullspace differential-IK controller with damped least squares; the 22-D finger delta (± 0.2 rad per step) is sent to a joint-position controller clamped to joint limits.

Observation space (325-D). The observation concatenates: relative joint positions and velocities (29 + 29), wrist pose (3 + 6, using 6-D rotation), accumulated wrist and finger targets (3 + 6 + 22), object pose (3 + 6), the reference wrist pose, finger joints, and object pose at the current reference step (3 + 6 + 22 + 3 + 6), the goal object bounding box (24) and goal hand landmarks in the object frame (51), the current object bounding box (24) and hand landmarks (51), and the last action (28). We use an asymmetric actor-critic: the critic additionally receives privileged information (per-joint PD gains,

effort limits and armature, object mass and friction, object velocities, and the object–wrist relative pose). The policy network receives the current frame concatenated with a 4-step history buffer; actor and critic use (512, 256, 128) MLPs.

Rewards. We use dense tracking rewards of the form $\exp(-s \cdot e)$, where e is the tracking error and s a per-term scale, listed in Table V, together with a small action-magnitude penalty. The hand-landmark term tracks 17 landmarks on the hand expressed in the object frame, which couples the hand and object motion and is given the largest weight. We remove the lifting reward and the success bonus, keeping only the imitation tracking terms so that performance reflects the quality of the reference motion, mirroring the humanoid setting.

Term	Error e	Scale s	Weight
Object position	$\ \Delta p_{\text{obj}}\ $	30	0.065
Object rotation	geodesic angle	1.9	0.065
Wrist position	$\ \Delta p_{\text{wrist}}\ $	20	0.039
Wrist rotation	geodesic angle	1.9	0.039
Finger joints	mean $ \Delta q $	10	0.039
Hand landmarks	per-point distance (object frame)	60	0.279
Action penalty	mean(a_t^2)	–	–0.10

TABLE V: **Reward terms for the dexterous grasping policy.** Each tracking term is $\exp(-s \cdot e) \in (0, 1]$; weights are normalized. The lifting reward and success bonuses of the original environment are removed.

Termination and success. An episode terminates if the object leaves the workspace (height below 0.5 m or above 1.5 m in the environment frame), if the wrist drifts more than 1.2 m from the object, or if any arm joint exceeds its position limit; episodes truncate at 12 s. An episode is counted as successful when the object is lifted at least 4 cm above the table while the hand remains in contact (minimum hand-to-object distance below 2 cm).

Initial-state generation via sequential IK rollout. To turn the single **SPIDER**-retargeted demonstration into a diverse initial-state distribution, we augment it into thousands of randomized, reachability-validated placements. For each candidate placement, the object is randomly translated on the table and rotated in yaw (uniform over the table workspace, discarding placements outside the table bounds), and the demonstrated wrist trajectory—stored in an object-centered frame—is re-expressed in the robot frame at that placement. A GPU-batched damped-least-squares IK solver then rolls the trajectory forward waypoint-by-waypoint: the first waypoint is warm-started from the nearest configuration in a precomputed table of healthy arm configurations (with a larger iteration budget of 400 vs. 50 for subsequent waypoints), and every later waypoint is warm-started from the previous solution. A placement is accepted only if the arm can track the *entire* trajectory within tolerance (wrist position error < 3 cm, orientation error < 0.15 rad) while keeping the proximal joints within 0.5 rad of the healthy-configuration manifold (rejecting contorted elbow postures). Before the object is lifted, its pose follows the transformed reference; after lift-off, it is treated as rigidly attached to the wrist and its pose is recomputed from the achieved forward kinematics. Accepted placements—each carrying a complete arm, finger, and object trajectory—are further filtered by a physics rollout that verifies grasp stability before being used for training.

Reset. On reset, each environment samples a placement and a starting waypoint along its trajectory (biased toward early waypoints, with a fixed fraction starting from the beginning) and initializes the robot and object from the corresponding validated state; for early waypoints, the object position is additionally perturbed by ± 4 cm in the horizontal plane with $\pm 30^\circ$ yaw, and joint targets receive small noise. Episode start times are desynchronized across environments so that resets do not coincide.

PPO hyperparameters. We train with PPO [57]: adaptive learning rate initialized at 3×10^{-4} (KL-targeted at 0.016), discount $\gamma = 0.95$, GAE $\lambda = 0.9$, clip 0.2, horizon length 16, minibatch size 16384, 5 mini-epochs, gradient norm clipping at 1.0, and observation/value normalization. Each policy is trained on a single L40 GPU for roughly 3–4 hours.

APPENDIX II COMPARISON WITH BASELINES

A. Comparison Against Kinematics-Based Retargeting Baselines

Setup and Per-Hand Results. To ensure representative evaluation while maintaining computational feasibility, we conduct our ablation study on a collection of challenging manipulation tasks from established datasets: (1) for *Oakink* [10], we evaluate on 7 distinct bimanual two-object manipulation tasks that were previously benchmarked in **ManipTrans** [21]; and (2) for *GigaHands* [54], we utilize their released example dataset comprising 3 manipulation trajectories over five hands. Per-hand tracking results are presented in Tabs. VI and VII, where **SPIDER** wins in terms of object tracking precision, especially with contact guidance enabled. The aggregated comparison is summarized in the main paper (Tab. I).

We report per-trajectory tracking metrics for *Oakink* and *GigaHands* datasets in the following tables.

Robot		Kinematic	SPIDER	SPIDER+C
Allegro	e_t (cm) ↓	9.4 ± 6.8	7.7 ± 7.5	4.6 ± 2.8
	e_r (deg) ↓	44.2 ± 30.9	37.6 ± 33.1	19.9 ± 18.9
	AUC ↑	21.5	53.4	56.6
Ability	e_t (cm) ↓	76.3 ± 143.5	9.3 ± 9.1	10.9 ± 8.2
	e_r (deg) ↓	75.2 ± 26.5	44.4 ± 39.1	36.4 ± 32.7
	AUC ↑	8.2	43.7	57.9
Inspire	e_t (cm) ↓	196.6 ± 545.3	9.1 ± 8.1	11.7 ± 11.7
	e_r (deg) ↓	91.0 ± 31.2	41.0 ± 29.4	43.3 ± 29.2
	AUC ↑	7.7	46.4	63.4
Schunk	e_t (cm) ↓	17.8 ± 9.7	9.1 ± 8.7	7.6 ± 4.7
	e_r (deg) ↓	86.5 ± 24.7	33.1 ± 26.2	28.8 ± 19.6
	AUC ↑	14.6	46.7	52.4
XHand	e_t (cm) ↓	39.5 ± 21.4	7.1 ± 7.6	5.2 ± 3.3
	e_r (deg) ↓	108.4 ± 15.6	32.4 ± 29.7	24.1 ± 19.6
	AUC ↑	5.7	53.4	54.5

TABLE VI: Comparison with Kinematic retargeting on Oakink. Mean $\pm$ std over 5 seeds. AUC denotes ADD-AUC in percent (Sec. I).

B. ManipTrans Retargeting Results

To match the exact setting of **ManipTrans** and avoid discrepancies caused by different simulators, we directly port the training environment and evaluation protocol from **ManipTrans** on the Oakink dataset. The following are the details of the training and evaluation protocol when comparing **SPIDER** with **ManipTrans** RL baseline.

RL Baseline Training. We train one residual RL policy per trajectory using the **ManipTrans** framework on 16 randomly picked single-hand picking trajectories from the Oakink dataset. Each policy is trained for 100 epochs on a single trajectory with the Inspire dexterous hand with 12-DoF simulation model, as the one used in original **ManipTrans** paper. The action space consists of 18-dim residual actions (6 wrist force/torque + 12 joint DOF positions), applied on top of a pre-trained base imitator that provides nominal actions from current observations. We use an action smoothing factor of 0.4 as suggested in the baseline repository.

RL Evaluation. We evaluate each trained policy with 5 random seeds (0-4) to capture stochasticity. For each seed, we run 2048 rollouts across 256 parallel IsaacGym environments without early termination, deterministic initial states from the demo trajectory, and save the 256 longest rollouts to HDF5.

SPIDER Retargeting. We run **SPIDER** with 1024 parallel IsaacGym environments. **SPIDER** optimizes only the 18-dim residual actions; the base imitator provides nominal actions at each step. The action smoothing factor is 0.4, matching RL. For reward, we use the exact same reward function as the RL baseline, but removing the action penalty term for efficiency (since our action is smooth by interpolation). Each trajectory is run with 5 seeds. All hyperparameters are set as in Table XIII.

Evaluation Metrics. We compute four metrics against ground-truth demo data from reference motion, following the **ManipTrans** evaluation protocol. Evaluation metrics are:

- e_t : mean L_2 object translation error (cm)
- e_r : geodesic object rotation error (degrees), computed as the angle of the axis-angle representation of $R_{gt}R_{pred}^T$
- e_j : mean L_2 hand joint position error (cm), computed over all body positions including wrist
- e_{ft} : mean L_2 fingertip position error (cm), computed over 5 fingertip joints (thumb, index, middle, ring, pinky)

All metrics are averaged over the trajectory length, then reported as mean $\pm$ std across 5 seeds per trajectory.

Experiment Results. We randomly select 16 trajectories (8 left-hand, 8 right-hand) from the Oakink dataset. Full results are shown in Table XIV. Note that for some trajectories, **SPIDER**'s std is 0, this is because the object is not moving

Robot		Kinematic	SPIDER	SPIDER+C
Allegro	e_t (cm) ↓	23.9 ± 11.2	4.5 ± 1.0	8.7 ± 3.8
	e_r (deg) ↓	45.0 ± 22.5	9.1 ± 5.7	9.9 ± 2.9
	AUC ↑	41.4	58.7	83.1
Ability	e_t (cm) ↓	37.3 ± 30.8	7.9 ± 5.5	12.5 ± 4.8
	e_r (deg) ↓	67.7 ± 44.2	19.7 ± 10.1	17.8 ± 6.3
	AUC ↑	17.4	55.5	30.8
Inspire	e_t (cm) ↓	44.2 ± 27.3	4.5 ± 0.3	11.4 ± 7.6
	e_r (deg) ↓	71.3 ± 29.6	15.1 ± 12.3	14.8 ± 8.1
	AUC ↑	12.5	53.1	73.4
Schunk	e_t (cm) ↓	38.3 ± 17.5	11.1 ± 9.4	14.0 ± 2.5
	e_r (deg) ↓	59.2 ± 16.4	19.9 ± 20.6	17.1 ± 7.1
	AUC ↑	18.1	60.8	30.9
XHand	e_t (cm) ↓	46.4 ± 42.8	4.5 ± 1.1	8.1 ± 4.5
	e_r (deg) ↓	73.4 ± 45.6	14.3 ± 6.2	13.2 ± 8.1
	AUC ↑	31.1	56.6	64.0

TABLE VII: Comparison with Kinematic retargeting on GigaHands. Mean $\pm$ std over 5 seeds. AUC denotes ADD-AUC in percent (Sec. I).

and **SPIDER** will do perfect tracking of hand motion.

C. DexMachina Retargeting Results

We evaluate **SPIDER** on bimanual dexterous manipulation tasks from the ARCTIC dataset [58], using the **DexMachina** framework [36] and the Genesis physics engine. We use the inspire dexterous hand in a bimanual configuration (two hands manipulating articulated objects simultaneously). To ensure a fair comparison, **SPIDER** directly uses the same RL training environment and reward function as **DexMachina**, and all evaluation metrics are computed identically for both methods, while removing the object virtual joint curriculum from **DexMachina**.

RL Baseline Training. We use the **DexMachina** RL policy as our baseline. **DexMachina** trains one RL policy per ARCTIC clip using the Genesis physics engine with the inspire hand in bimanual configuration. The action space consists of 36 dimensions in hybrid mode: for each hand, 6 wrist force/torque dimensions (3 position + 3 rotation) and 12 finger joint position targets. The RL policy is deterministic at evaluation time; since the initial state is fixed, the RL policy produces identical rollouts across random seeds.

RL Evaluation. We evaluate the PPO policy once per task with 5 seeds and report the mean and std of the metrics.

Reward Setting. To ensure **SPIDER** optimizes the same objective as the RL baseline, we delegate reward computation to **DexMachina**’s reward function, but removing the action penalty term for efficiency (since our action is smooth by interpolation).

Curriculum Learning. In **SPIDER** retargeting, we *disable curriculum learning* in **DexMachina**, so the object joints are always free (passive) throughout optimization. For **DexMachina**, we use the default curriculum learning settings where object joint stiffness and damping are annealed from high values (providing a guiding force) to zero across training iterations.

SPIDER Retargeting. We run **SPIDER** with 1024 parallel Genesis environments. **SPIDER** optimizes the full 36-dim action space (both hands’ wrist and finger actions) using sampling-based MPC. Each task is run with 5 random seeds. All hyperparameters are listed in Table XV.

Evaluation Metrics. We evaluate object tracking performance using three metrics computed at each simulation timestep in original **DexMachina** paper [36], comparing the simulated object state against the ground-truth reference trajectory from the ARCTIC dataset:

- d_{pos} : L2 position distance (m) between simulated and reference object positions, $d_{\text{pos}} = \|\mathbf{p} - \mathbf{p}^*\|_2$.
- d_{rot} : geodesic rotation distance (rad) on $\text{SO}(3)$ between simulated and reference orientations, computed as $d_{\text{rot}} = 2 \arcsin(\min(\|\text{Im}(\mathbf{q} \otimes \bar{\mathbf{q}}^*)\|_2, 1))$, where $\mathbf{q}$ and $\mathbf{q}^*$ are unit quaternions in (w, x, y, z) convention and $\bar{\mathbf{q}}^*$ denotes the conjugate.
- d_{arti} : articulation distance (rad), the absolute difference in the object’s articulated joint angle, $d_{\text{arti}} = |a - a^*|$.

All metrics are averaged over the trajectory length, then reported as mean $\pm$ std across 5 seeds per task for **SPIDER**. For the RL baseline, we report the single deterministic evaluation, as it produces identical rollouts across seeds (std = 0). In the original **DexMachina** paper, they also report the success rate by computing Average Distance Area Under Curve (ADD-AUC) for each object part separately. However, since this metric is not open source, we directly report the raw metrics for comparison.

Experiment Results. We evaluate on 7 bimanual dexterous manipulation tasks from the ARCTIC dataset (box, ketchup, ketchup-u02, laptop, mixer, notebook, waffleiron), each defined by a 200-frame clip at 50 Hz. Each task is named in the format of <object-name>-<start-frame>-<end-frame>, similar to the original **DexMachina** paper. Full results are shown in Table XVI.

SPIDER wins on 6/7 tasks for d_{pos} , 5/7 for d_{rot} , and 5/7 for d_{arti} . On average, **SPIDER** achieves lower position error (0.421 vs. 0.517 m) and rotation error (0.959 vs. 1.099 rad), while the RL baseline achieves slightly lower articulation error (0.622 vs. 0.643 rad). Notably, **SPIDER** achieves these results without any task-specific training, using only test-time optimization with the same reward function as the RL baseline.

APPENDIX III VIRTUAL CONTACT GUIDANCE DETAILS

Solution Ambiguity. For a retargeting method to be valid, it is also important to preserve the human-preferred contact in the demonstration. Take the stir stick manipulation in Fig. 12 as an example, the robot can hold the stick either between the thumb and index finger or between the index and middle finger; both achieve the task, but the former is more natural and aligns with the human demonstration. However, due to the non-convex cost landscape of Eq. (1a) there may be multiple solutions that achieve similar object motion with different contact behaviors. As a result, a sampling-based optimizer may converge to alternative contact modes, leading to implausible contact trajectories in a demonstration even when the object trajectory is reproduced.

Virtual Contact Guidance. To guide sampling towards the desired contact mode, we introduce *virtual contact guidance*.

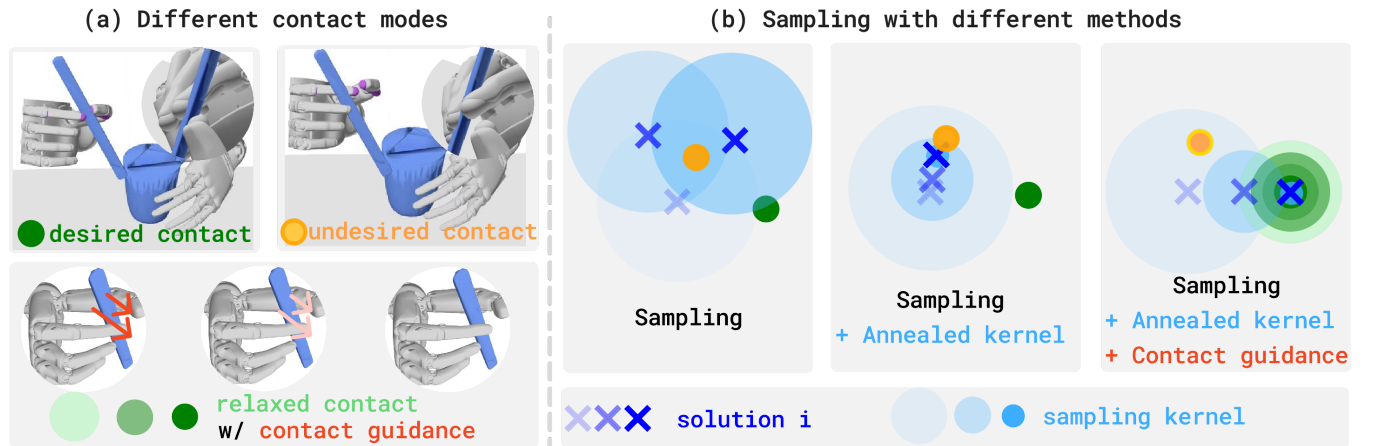


Fig. 12: **Contact mode mismatch in sampling and virtual guidance method to correct it.** (a) Given the same task, the robot can hold the object in different contact modes while still finishing the task. However, the contact mode from human is preferred. (b) Given different sampling methods when seeking a feasible motion with correct contact: *Standard sampling* (left): uses a fixed search radius, leading to high variance. The resulting solution fails to converge well. *Annealed sampling* (middle): gradually shrinks the search radius, starting coarse and narrowing down to a finer solution, but may drift toward a feasible solution with wrong contact. *Annealed sampling with virtual contact guidance* (right): expands the feasible region by adding virtual guidance near target contacts. This enlarges the feasible region to a relaxed feasible set, steering sampling away from undesired feasible solutions and towards the intended contact sequence.

This guidance factor expands the feasible solution set and biases the optimization process towards the target configuration. Unlike a soft contact reward or cost - which may still fail when the desired mode is harder to sample - virtual contact guidance explicitly *enlarges* the basin of attraction to make it easier to sample from, as illustrated in Fig. 12. Our method applies a virtual constraint between intended contact points on the object and robot fingers, “sticking” the object to the

target configuration in early stages and gradually relaxing this constraint in a curriculum-like fashion. Concretely, we impose a virtual constraint that maintains the relative position between contact pairs. For the k -th object-hand contact pair (e.g., handle-thumb), we define the current relative position as ${}^{\text{robot}}p_{k,t}^{\text{object}} = p_{k,t}^{\text{robot}} - p_{k,t}^{\text{object}}$ and the desired reference position as ${}^{\text{robot}}p_{k,t}^{\text{object,ref}} = p_{k,t}^{\text{robot,ref}} - p_{k,t}^{\text{object,ref}}$. The constraint is activated when the contact indicator $c_{k,t} = 1$, which occurs when the reference relative position is within the contact threshold: $c_{k,t} = \mathbf{1}(\|{}^{\text{robot}}p_{k,t}^{\text{object,ref}}\|_2 \leq \epsilon_{\text{contact}})$. The constraint strength is controlled by a penalty parameter $\eta_i \rightarrow \infty$ when $i \rightarrow N$:

$$c_{k,t} \|{}^{\text{robot}}p_{k,t}^{\text{object}} - {}^{\text{robot}}p_{k,t}^{\text{object,ref}}\|_2^2 \leq \eta_i \quad (6)$$

This strategy connects to prior ideas: virtual object constraint [36] $\|p_{k,t}^{\text{object}} - p_{k,t}^{\text{object,ref}}\|_2^2 \leq \eta$, which expands feasibility around absolute object states in RL, and contact cost [32], biasing optimization towards desired hand-object *relative* states. Our formulation reduces sampling complexity while preserving the intended contact sequence by maintaining *relative* hand-object relationships.

Robustness against imperfect reference contact. It is important to note that, in order to maintain robustness against imperfect demonstrations with noisy or unstable contact, virtual constraints should be selectively relaxed rather than enforced indiscriminately. To achieve this, we apply a contact filter that detects unstable interactions: if a contact duration is shorter than $t_{c,\min}$ or if the contact point drifts more than $d_{c,\max}$ during that period, the contact is classified as unstable and the corresponding virtual constraint is disabled. This implementation ensures that only reliable contacts contribute to guidance, thereby preventing noisy demonstrations from biasing the optimization process.

Dataset	Task	Metric	Kinematic	SPIDER	SPIDER+C
oakink	lift_board	e_{ft} (cm)	1.6 ± 0.0	2.8 ± 0.4	4.4 ± 0.9
		e_j (rad)	0.06 ± 0.00	0.03 ± 0.00	0.04 ± 0.00
		e_t (cm)	3.8 ± 0.0	1.9 ± 0.2	1.8 ± 0.2
		e_r (deg)	12.5 ± 0.0	7.6 ± 1.1	4.4 ± 1.1
	pick_spoon_bowl	e_{ft} (cm)	0.9 ± 0.0	2.4 ± 0.6	3.8 ± 1.9
		e_j (rad)	0.05 ± 0.00	0.03 ± 0.00	0.08 ± 0.09
		e_t (cm)	6.1 ± 0.0	4.5 ± 0.8	7.2 ± 1.4
		e_r (deg)	56.3 ± 0.0	20.9 ± 2.0	50.7 ± 26.4
	pour_tube	e_{ft} (cm)	0.9 ± 0.0	0.7 ± 0.2	5.7 ± 0.2
		e_j (rad)	0.06 ± 0.00	0.20 ± 0.15	0.04 ± 0.00
		e_t (cm)	23.3 ± 0.0	23.6 ± 2.5	5.7 ± 0.4
		e_r (deg)	109.0 ± 0.0	110.2 ± 14.2	20.6 ± 2.5
	stir_beaker	e_{ft} (cm)	1.3 ± 0.0	1.9 ± 0.5	2.9 ± 0.2
		e_j (rad)	0.06 ± 0.00	0.02 ± 0.00	0.02 ± 0.00
		e_t (cm)	15.4 ± 0.0	13.5 ± 1.4	7.3 ± 3.1
		e_r (deg)	46.9 ± 0.0	53.1 ± 11.7	24.7 ± 5.2
	uncap_alcohol_burner	e_{ft} (cm)	1.1 ± 0.0	1.4 ± 0.2	2.6 ± 0.5
		e_j (rad)	0.06 ± 0.00	0.01 ± 0.00	0.02 ± 0.00
		e_t (cm)	3.3 ± 0.0	2.6 ± 0.2	3.1 ± 0.4
		e_r (deg)	22.6 ± 0.0	24.4 ± 0.9	9.9 ± 2.5
	unplug	e_{ft} (cm)	1.0 ± 0.0	1.2 ± 0.1	2.7 ± 0.3
		e_j (rad)	0.06 ± 0.00	0.02 ± 0.00	0.06 ± 0.06
		e_t (cm)	6.4 ± 0.0	3.8 ± 1.6	6.0 ± 1.5
		e_r (deg)	14.9 ± 0.0	22.4 ± 4.4	25.4 ± 12.2
	wipe_board	e_{ft} (cm)	0.9 ± 0.0	2.1 ± 0.3	3.0 ± 0.2
		e_j (rad)	0.05 ± 0.00	0.02 ± 0.00	0.05 ± 0.00
		e_t (cm)	7.5 ± 0.0	3.7 ± 0.6	0.9 ± 0.1
		e_r (deg)	47.0 ± 0.0	24.3 ± 4.6	3.3 ± 0.4
	Avg.	e_{ft} (cm)	1.1 ± 0.2	1.8 ± 0.7	3.6 ± 1.0
		e_j (rad)	0.06 ± 0.00	0.05 ± 0.06	0.04 ± 0.02
		e_t (cm)	9.4 ± 6.8	7.7 ± 7.4	4.6 ± 2.4
		e_r (deg)	44.2 ± 30.9	37.6 ± 32.2	19.9 ± 15.2
gigahand	p36-tea	e_{ft} (cm)	1.7 ± 0.4	4.9 ± 0.5	7.0 ± 1.5
		e_j (rad)	0.29 ± 0.17	0.12 ± 0.13	0.11 ± 0.01
		e_t (cm)	11.1 ± 0.0	3.6 ± 1.1	4.4 ± 2.9
		e_r (deg)	72.9 ± 0.0	16.1 ± 4.5	7.9 ± 2.6
	p44-dog	e_{ft} (cm)	4.0 ± 0.1	5.8 ± 0.6	2.0 ± 0.0
		e_j (rad)	0.20 ± 0.09	0.10 ± 0.08	0.05 ± 0.00
		e_t (cm)	37.7 ± 0.0	4.7 ± 0.7	11.3 ± 0.8
		e_r (deg)	40.0 ± 0.0	4.6 ± 0.4	13.1 ± 0.8
	p52-instrument	e_{ft} (cm)	3.1 ± 0.0	4.2 ± 0.1	4.5 ± 0.3
		e_j (rad)	0.08 ± 0.00	0.07 ± 0.00	0.13 ± 0.12
		e_t (cm)	22.6 ± 0.0	5.2 ± 0.3	10.4 ± 2.3
		e_r (deg)	17.6 ± 0.0	6.7 ± 0.1	8.7 ± 1.5
	Avg.	e_{ft} (cm)	3.0 ± 0.9	5.0 ± 0.6	4.5 ± 2.0
		e_j (rad)	0.19 ± 0.09	0.10 ± 0.02	0.10 ± 0.04
		e_t (cm)	23.8 ± 10.9	4.5 ± 0.7	8.7 ± 3.1
		e_r (deg)	43.5 ± 22.7	9.1 ± 5.0	9.9 ± 2.3

TABLE VIII: Per-trajectory tracking metrics for *allegro*. Mean $\pm$ std across 5 seeds.

Dataset	Task	Metric	Kinematic	SPIDER	SPIDER+C
oakink	lift_board	e_{fl} (cm)	6.5 ± 0.0	3.7 ± 1.1	2.0 ± 0.4
		e_j (rad)	0.19 ± 0.07	0.04 ± 0.00	0.02 ± 0.00
		e_t (cm)	18.0 ± 0.0	7.8 ± 2.0	5.5 ± 2.6
		e_r (deg)	19.3 ± 0.0	6.6 ± 2.3	4.9 ± 1.8
	pick_spoon_bowl	e_{fl} (cm)	1.3 ± 0.0	3.3 ± 0.6	7.0 ± 0.3
		e_j (rad)	0.13 ± 0.07	0.02 ± 0.00	0.12 ± 0.00
		e_t (cm)	12.4 ± 0.0	6.5 ± 1.8	9.0 ± 0.3
		e_r (deg)	53.7 ± 0.0	64.3 ± 12.4	41.1 ± 12.3
	pour_tube	e_{fl} (cm)	1.8 ± 0.4	2.2 ± 0.4	8.6 ± 1.3
		e_j (rad)	0.16 ± 0.10	0.06 ± 0.09	0.01 ± 0.00
		e_t (cm)	30.9 ± 0.0	30.0 ± 2.3	26.6 ± 1.0
		e_r (deg)	93.2 ± 0.0	126.7 ± 7.2	92.6 ± 8.7
	stir_beaker	e_{fl} (cm)	3.5 ± 0.1	3.5 ± 0.3	8.4 ± 0.3
		e_j (rad)	0.09 ± 0.05	0.05 ± 0.04	0.08 ± 0.00
		e_t (cm)	14.2 ± 0.0	8.3 ± 3.2	19.5 ± 0.7
		e_r (deg)	72.6 ± 0.0	38.8 ± 6.1	74.2 ± 5.7
	uncap_alcohol_burner	e_{fl} (cm)	3.5 ± 0.4	3.8 ± 1.6	4.0 ± 0.8
		e_j (rad)	0.26 ± 0.15	0.07 ± 0.09	0.05 ± 0.00
		e_t (cm)	10.1 ± 0.0	5.1 ± 2.4	4.0 ± 2.0
		e_r (deg)	75.6 ± 0.0	43.5 ± 11.7	12.4 ± 2.9
	unplug	e_{fl} (cm)	3.2 ± 0.1	2.2 ± 0.5	2.6 ± 0.4
		e_j (rad)	0.09 ± 0.05	0.05 ± 0.05	0.01 ± 0.00
		e_t (cm)	422.5 ± 0.0	1.6 ± 0.3	5.4 ± 2.0
		e_r (deg)	98.9 ± 0.0	12.6 ± 0.8	19.9 ± 7.3
	wipe_board	e_{fl} (cm)	3.0 ± 0.1	3.3 ± 0.3	4.3 ± 0.4
		e_j (rad)	0.15 ± 0.08	0.07 ± 0.07	0.01 ± 0.00
		e_t (cm)	16.6 ± 0.0	4.9 ± 0.2	6.2 ± 1.5
		e_r (deg)	104.0 ± 0.0	22.4 ± 0.6	9.8 ± 1.0
	Avg.	e_{fl} (cm)	3.3 ± 1.6	3.1 ± 0.6	5.3 ± 2.5
		e_j (rad)	0.15 ± 0.05	0.05 ± 0.02	0.04 ± 0.04
		e_t (cm)	74.9 ± 142.0	9.2 ± 8.8	10.9 ± 8.0
		e_r (deg)	73.9 ± 27.5	45.0 ± 38.0	36.4 ± 32.0
gigahand	p36-tea	e_{fl} (cm)	2.0 ± 0.2	4.3 ± 0.4	7.2 ± 1.6
		e_j (rad)	0.11 ± 0.04	0.08 ± 0.03	0.11 ± 0.02
		e_t (cm)	10.8 ± 0.0	4.2 ± 1.1	11.3 ± 5.2
		e_r (deg)	49.0 ± 0.0	21.3 ± 0.6	21.7 ± 2.7
	p44-dog	e_{fl} (cm)	5.8 ± 0.0	5.5 ± 0.1	3.9 ± 0.0
		e_j (rad)	0.15 ± 0.04	0.11 ± 0.04	0.09 ± 0.00
		e_t (cm)	24.7 ± 0.0	4.7 ± 0.2	10.9 ± 0.5
		e_r (deg)	30.3 ± 0.0	6.7 ± 0.4	9.4 ± 0.4
	p52-instrument	e_{fl} (cm)	6.7 ± 0.0	3.5 ± 0.1	3.7 ± 0.3
		e_j (rad)	0.08 ± 0.00	0.10 ± 0.00	0.14 ± 0.02
		e_t (cm)	84.2 ± 0.0	14.8 ± 4.0	15.4 ± 5.5
		e_r (deg)	135.1 ± 0.0	31.0 ± 1.9	22.2 ± 2.3
	Avg.	e_{fl} (cm)	4.8 ± 2.0	4.4 ± 0.8	4.9 ± 1.6
		e_j (rad)	0.11 ± 0.03	0.10 ± 0.01	0.11 ± 0.02
		e_t (cm)	39.9 ± 31.8	7.9 ± 4.9	12.5 ± 2.0
		e_r (deg)	71.5 ± 45.6	19.7 ± 10.0	17.8 ± 5.9

TABLE IX: Per-trajectory tracking metrics for ability. Mean $\pm$ std across 5 seeds.

Dataset	Task	Metric	Kinematic	SPIDER	SPIDER+C
oakink	lift_board	e_{ft} (cm)	7.1 ± 0.0	1.8 ± 0.2	2.4 ± 0.7
		e_j (rad)	0.05 ± 0.00	0.03 ± 0.00	0.04 ± 0.00
		e_t (cm)	66.1 ± 0.0	8.1 ± 0.2	2.6 ± 1.9
		e_r (deg)	21.7 ± 0.0	6.3 ± 0.8	8.3 ± 1.4
	pick_spoon_bowl	e_{ft} (cm)	6.5 ± 0.0	1.1 ± 0.3	2.0 ± 0.3
		e_j (rad)	0.14 ± 0.05	0.05 ± 0.06	0.02 ± 0.01
		e_t (cm)	184.1 ± 0.0	9.4 ± 2.5	12.8 ± 1.7
		e_r (deg)	87.0 ± 0.0	44.5 ± 8.5	52.6 ± 12.6
	pour_tube	e_{ft} (cm)	7.0 ± 0.0	0.4 ± 0.1	3.7 ± 1.4
		e_j (rad)	0.05 ± 0.00	0.01 ± 0.00	0.02 ± 0.01
		e_t (cm)	149.4 ± 0.0	27.0 ± 1.5	31.7 ± 17.8
		e_r (deg)	124.3 ± 0.0	85.2 ± 8.2	88.3 ± 17.7
	stir_beaker	e_{ft} (cm)	6.9 ± 0.0	2.2 ± 0.2	3.9 ± 0.5
		e_j (rad)	0.05 ± 0.00	0.03 ± 0.00	0.02 ± 0.00
		e_t (cm)	169.2 ± 0.0	4.5 ± 1.4	17.5 ± 1.8
		e_r (deg)	106.7 ± 0.0	32.4 ± 10.7	65.4 ± 9.2
	uncap_alcohol_burner	e_{ft} (cm)	7.0 ± 0.0	1.5 ± 0.2	3.4 ± 0.5
		e_j (rad)	0.05 ± 0.00	0.04 ± 0.00	0.07 ± 0.00
		e_t (cm)	23.9 ± 0.0	3.6 ± 1.0	4.6 ± 0.6
		e_r (deg)	82.2 ± 0.0	30.1 ± 5.8	36.8 ± 28.1
	unplug	e_{ft} (cm)	9.2 ± 0.0	2.5 ± 0.7	2.7 ± 0.9
		e_j (rad)	0.22 ± 0.07	0.07 ± 0.09	0.03 ± 0.00
		e_t (cm)	731.4 ± 1313.0	2.8 ± 1.2	7.1 ± 1.0
		e_r (deg)	109.1 ± 1.1	13.6 ± 6.5	24.7 ± 10.5
	wipe_board	e_{ft} (cm)	7.0 ± 0.2	1.7 ± 0.3	3.7 ± 0.3
		e_j (rad)	0.19 ± 0.07	0.14 ± 0.09	0.03 ± 0.00
		e_t (cm)	51.9 ± 18.7	5.9 ± 3.7	5.9 ± 0.8
		e_r (deg)	106.1 ± 4.7	70.3 ± 21.9	27.0 ± 4.6
	Avg.	e_{ft} (cm)	7.2 ± 0.8	1.6 ± 0.6	3.1 ± 0.7
		e_j (rad)	0.11 ± 0.07	0.05 ± 0.04	0.03 ± 0.02
		e_t (cm)	196.6 ± 225.8	8.8 ± 7.7	11.7 ± 9.4
		e_r (deg)	91.0 ± 31.2	40.3 ± 26.6	43.3 ± 25.3
gigahand	p36-tea	e_{ft} (cm)	8.2 ± 0.0	4.7 ± 0.2	6.5 ± 0.6
		e_j (rad)	0.07 ± 0.00	0.07 ± 0.00	0.13 ± 0.01
		e_t (cm)	11.5 ± 0.0	4.9 ± 0.3	4.2 ± 1.1
		e_r (deg)	108.4 ± 0.0	32.4 ± 1.6	13.7 ± 1.3
	p44-dog	e_{ft} (cm)	6.3 ± 2.4	3.8 ± 0.1	3.6 ± 0.0
		e_j (rad)	0.12 ± 0.09	0.07 ± 0.00	0.06 ± 0.00
		e_t (cm)	65.0 ± 24.4	4.3 ± 0.2	10.5 ± 0.3
		e_r (deg)	60.0 ± 21.4	5.9 ± 0.2	8.4 ± 0.2
	p52-instrument	e_{ft} (cm)	6.5 ± 0.0	4.1 ± 0.1	1.7 ± 0.3
		e_j (rad)	0.07 ± 0.00	0.06 ± 0.00	0.10 ± 0.01
		e_t (cm)	55.9 ± 0.0	4.3 ± 0.2	19.6 ± 7.3
		e_r (deg)	45.4 ± 0.0	7.1 ± 0.5	22.4 ± 9.8
	Avg.	e_{ft} (cm)	7.0 ± 0.9	4.2 ± 0.4	3.9 ± 2.0
		e_j (rad)	0.09 ± 0.02	0.06 ± 0.00	0.10 ± 0.03
		e_t (cm)	44.2 ± 23.4	4.5 ± 0.3	11.4 ± 6.3
		e_r (deg)	71.3 ± 26.9	15.1 ± 12.2	14.8 ± 5.8

TABLE X: Per-trajectory tracking metrics for inspire. Mean $\pm$ std across 5 seeds.

Dataset	Task	Metric	Kinematic	SPIDER	SPIDER+C
oakink	lift_board	e_{ft} (cm)	5.2 ± 0.0	3.1 ± 0.4	3.0 ± 0.3
		e_j (rad)	0.28 ± 0.00	0.03 ± 0.00	0.02 ± 0.00
		e_t (cm)	17.8 ± 0.0	9.2 ± 3.6	6.6 ± 8.6
		e_r (deg)	54.6 ± 0.0	10.7 ± 3.7	9.9 ± 11.4
	pick_spoon_bowl	e_{ft} (cm)	4.1 ± 0.0	3.3 ± 0.3	4.2 ± 0.4
		e_j (rad)	0.06 ± 0.00	0.02 ± 0.00	0.03 ± 0.00
		e_t (cm)	9.2 ± 0.0	4.9 ± 0.4	6.6 ± 1.2
		e_r (deg)	98.2 ± 0.0	43.3 ± 11.1	54.0 ± 11.8
	pour_tube	e_{ft} (cm)	5.1 ± 0.0	1.7 ± 0.5	5.9 ± 0.3
		e_j (rad)	0.28 ± 0.00	0.02 ± 0.00	0.03 ± 0.00
		e_t (cm)	35.8 ± 0.0	28.8 ± 0.6	14.6 ± 0.4
		e_r (deg)	103.4 ± 0.0	82.5 ± 10.5	55.6 ± 5.9
	stir_beaker	e_{ft} (cm)	4.1 ± 0.0	2.8 ± 0.2	3.9 ± 0.2
		e_j (rad)	0.15 ± 0.00	0.02 ± 0.00	0.02 ± 0.00
		e_t (cm)	21.2 ± 0.0	10.6 ± 1.7	9.9 ± 0.5
		e_r (deg)	112.1 ± 0.0	48.3 ± 8.2	35.4 ± 6.1
	uncap_alcohol_burner	e_{ft} (cm)	3.3 ± 0.0	1.4 ± 0.1	2.1 ± 0.2
		e_j (rad)	0.05 ± 0.00	0.02 ± 0.00	0.01 ± 0.00
		e_t (cm)	22.3 ± 0.0	3.0 ± 0.8	4.5 ± 0.9
		e_r (deg)	106.4 ± 0.0	5.8 ± 0.4	16.9 ± 9.1
	unplug	e_{ft} (cm)	8.0 ± 1.9	2.2 ± 0.0	3.9 ± 0.2
		e_j (rad)	0.28 ± 0.11	0.02 ± 0.00	0.02 ± 0.00
		e_t (cm)	12.3 ± 7.4	1.9 ± 0.1	3.8 ± 0.2
		e_r (deg)	58.8 ± 22.5	12.5 ± 0.7	11.5 ± 5.2
	wipe_board	e_{ft} (cm)	4.1 ± 0.1	3.4 ± 0.1	4.3 ± 0.1
		e_j (rad)	0.13 ± 0.04	0.03 ± 0.00	0.02 ± 0.00
		e_t (cm)	6.0 ± 0.9	5.1 ± 0.3	8.2 ± 1.0
		e_r (deg)	71.8 ± 16.8	28.3 ± 1.8	23.6 ± 9.3
	Avg.	e_{ft} (cm)	4.8 ± 1.4	2.6 ± 0.7	3.9 ± 1.1
		e_j (rad)	0.18 ± 0.10	0.02 ± 0.00	0.02 ± 0.01
		e_t (cm)	17.8 ± 9.3	9.1 ± 8.5	7.8 ± 3.4
		e_r (deg)	86.5 ± 22.3	33.1 ± 25.3	29.6 ± 17.8
gigahand	p36-tea	e_{ft} (cm)	4.4 ± 0.0	4.8 ± 0.7	4.0 ± 0.2
		e_j (rad)	0.07 ± 0.00	0.05 ± 0.00	0.08 ± 0.00
		e_t (cm)	15.2 ± 0.0	7.1 ± 9.1	14.1 ± 0.9
		e_r (deg)	80.7 ± 0.0	36.1 ± 28.7	26.9 ± 1.2
	p44-dog	e_{ft} (cm)	4.8 ± 0.1	5.5 ± 0.3	4.3 ± 0.0
		e_j (rad)	0.12 ± 0.10	0.05 ± 0.00	0.05 ± 0.00
		e_t (cm)	45.3 ± 8.5	4.5 ± 0.2	11.0 ± 0.3
		e_r (deg)	42.0 ± 5.8	6.6 ± 0.3	10.2 ± 0.2
	p52-instrument	e_{ft} (cm)	4.9 ± 0.0	5.9 ± 0.6	4.1 ± 0.1
		e_j (rad)	0.07 ± 0.00	0.04 ± 0.00	0.07 ± 0.00
		e_t (cm)	54.5 ± 0.0	21.8 ± 3.1	16.8 ± 1.1
		e_r (deg)	54.9 ± 0.0	16.9 ± 1.9	14.3 ± 0.8
	Avg.	e_{ft} (cm)	4.7 ± 0.2	5.4 ± 0.5	4.1 ± 0.1
		e_j (rad)	0.09 ± 0.03	0.05 ± 0.00	0.07 ± 0.01
		e_t (cm)	38.3 ± 16.8	11.1 ± 7.6	14.0 ± 2.4
		e_r (deg)	59.2 ± 16.1	19.9 ± 12.2	17.1 ± 7.1

TABLE XI: Per-trajectory tracking metrics for schunk. Mean $\pm$ std across 5 seeds.

Dataset	Task	Metric	Kinematic	SPIDER	SPIDER+C
oakink	lift_board	e_{ft} (cm)	6.4 ± 0.1	2.4 ± 0.1	2.4 ± 0.2
		e_j (rad)	0.08 ± 0.06	0.03 ± 0.00	0.04 ± 0.00
		e_t (cm)	74.6 ± 18.3	1.9 ± 0.5	1.5 ± 0.1
		e_r (deg)	74.3 ± 8.0	9.4 ± 0.4	4.3 ± 0.7
	pick_spoon_bowl	e_{ft} (cm)	6.1 ± 0.4	2.8 ± 0.4	4.7 ± 1.0
		e_j (rad)	0.08 ± 0.07	0.02 ± 0.00	0.01 ± 0.00
		e_t (cm)	29.0 ± 0.8	5.4 ± 0.2	7.5 ± 1.3
		e_r (deg)	109.1 ± 5.0	21.6 ± 2.8	33.2 ± 15.4
	pour_tube	e_{ft} (cm)	6.0 ± 0.0	1.3 ± 0.7	5.2 ± 3.9
		e_j (rad)	0.05 ± 0.00	0.12 ± 0.12	0.08 ± 0.14
		e_t (cm)	51.1 ± 0.0	22.7 ± 8.5	10.5 ± 3.4
		e_r (deg)	112.5 ± 0.0	95.0 ± 27.6	58.6 ± 9.7
	stir_beaker	e_{ft} (cm)	6.8 ± 0.0	2.7 ± 0.5	5.0 ± 0.3
		e_j (rad)	0.05 ± 0.00	0.05 ± 0.05	0.03 ± 0.00
		e_t (cm)	48.2 ± 0.0	9.8 ± 2.4	7.0 ± 0.9
		e_r (deg)	105.2 ± 0.0	45.0 ± 7.3	21.7 ± 0.5
	uncap_alcohol_burner	e_{ft} (cm)	5.9 ± 0.0	1.6 ± 0.2	2.4 ± 0.2
		e_j (rad)	0.06 ± 0.00	0.04 ± 0.00	0.02 ± 0.00
		e_t (cm)	8.4 ± 0.0	5.1 ± 1.5	3.4 ± 0.3
		e_r (deg)	119.2 ± 0.0	22.1 ± 4.3	10.8 ± 1.6
	unplug	e_{ft} (cm)	6.4 ± 0.0	1.4 ± 0.1	3.8 ± 0.3
		e_j (rad)	0.05 ± 0.00	0.02 ± 0.00	0.03 ± 0.00
		e_t (cm)	43.7 ± 0.0	1.3 ± 0.4	3.9 ± 0.7
		e_r (deg)	125.5 ± 0.0	12.4 ± 1.1	32.0 ± 12.3
	wipe_board	e_{ft} (cm)	7.1 ± 0.0	2.5 ± 0.1	3.1 ± 0.2
		e_j (rad)	0.05 ± 0.00	0.03 ± 0.00	0.02 ± 0.00
		e_t (cm)	21.5 ± 0.0	3.9 ± 0.3	2.6 ± 1.2
		e_r (deg)	113.0 ± 0.0	21.5 ± 0.2	8.5 ± 6.6
	Avg.	e_{ft} (cm)	6.4 ± 0.4	2.1 ± 0.6	3.8 ± 1.1
		e_j (rad)	0.06 ± 0.01	0.04 ± 0.03	0.03 ± 0.02
		e_t (cm)	39.5 ± 20.2	7.1 ± 6.8	5.2 ± 3.0
		e_r (deg)	108.4 ± 15.2	32.4 ± 27.6	24.1 ± 17.5
gigahand	p36-tea	e_{ft} (cm)	6.9 ± 0.0	5.8 ± 0.5	4.2 ± 1.0
		e_j (rad)	0.07 ± 0.00	0.07 ± 0.00	0.12 ± 0.03
		e_t (cm)	30.6 ± 0.0	3.6 ± 0.4	8.3 ± 4.2
		e_r (deg)	114.2 ± 0.0	19.4 ± 1.1	23.3 ± 4.4
	p44-dog	e_{ft} (cm)	5.2 ± 1.2	5.7 ± 0.3	4.1 ± 0.0
		e_j (rad)	0.16 ± 0.16	0.07 ± 0.00	0.07 ± 0.00
		e_t (cm)	97.8 ± 36.5	4.8 ± 0.4	12.5 ± 0.8
		e_r (deg)	90.0 ± 31.8	5.9 ± 0.4	11.7 ± 0.9
	p52-instrument	e_{ft} (cm)	1.7 ± 0.0	4.3 ± 1.2	2.9 ± 0.2
		e_j (rad)	0.05 ± 0.00	0.07 ± 0.01	0.06 ± 0.00
		e_t (cm)	10.7 ± 0.0	5.1 ± 1.3	3.4 ± 0.7
		e_r (deg)	16.0 ± 0.0	17.5 ± 2.7	4.7 ± 0.8
	Avg.	e_{ft} (cm)	4.6 ± 2.2	5.3 ± 0.7	3.7 ± 0.6
		e_j (rad)	0.10 ± 0.04	0.07 ± 0.00	0.09 ± 0.03
		e_t (cm)	46.4 ± 37.3	4.5 ± 0.7	8.1 ± 3.7
		e_r (deg)	73.4 ± 41.8	14.3 ± 6.0	13.2 ± 7.7

TABLE XII: Per-trajectory tracking metrics for xhand. Mean $\pm$ std across 5 seeds.

TABLE XIII: SPIDER hyperparameters for the ManipTrans (IsaacGym) experiments.

Parameter	Symbol	Value
<i>Simulation</i>		
Simulator	–	IsaacGym (GPU PhysX)
Simulation timestep	Δt_{sim}	1/120 s
Control timestep	Δt_{ctrl}	0.2 s
Knot point spacing	Δt_{knot}	0.2 s
Action dimensions	n_u	18
<i>Sampling-based Optimization</i>		
Planning horizon	H	0.6 s (36 sim steps)
Parallel samples	N_w	1024
Max sampling iterations	N	32
Early stopping threshold	ϵ	0.001
Early stopping patience	–	4 iterations
<i>Noise Schedule</i>		
Initial noise scale	σ_{first}	0.5
Final noise scale (horizon end)	σ_{last}	1.0
Noise decay per iteration	β	0.93
Wrist pos/rot noise scale	–	0.0
Joint noise scale	–	1.0
<i>Reward</i>		
Terminal reward scale	α_T	10.0
Sampling temperature	τ	1.0
<i>ManipTrans-specific</i>		
Actions moving average	γ	0.4

TABLE XIV: Multi-seed comparison between **SPIDER** and **ManipTrans** RL baseline on 16 trajectories from Oakink dataset. Each method is evaluated with 5 seeds per trajectory. Metrics: e_t = object translation error (cm), e_r = object rotation error (deg), e_j = hand joint position error (cm), e_{ft} = fingertip position error (cm). Values reported as mean $\pm$ std.

Data ID	Side	Method	e_t (cm) $\downarrow$	e_r (deg) $\downarrow$	e_j (cm) $\downarrow$	e_{ft} (cm) $\downarrow$
01857@10	right	SPIDER	34.81 $\pm$ 0.30	101.76 $\pm$ 1.12	4.73 $\pm$ 0.01	4.10 $\pm$ 0.01
		ManipTrans	34.50 $\pm$ 3.34	112.67 $\pm$ 19.26	25.98 $\pm$ 19.13	25.59 $\pm$ 18.77
031b7@0	right	SPIDER	40.74 $\pm$ 0.74	47.17 $\pm$ 22.58	5.03 $\pm$ 0.01	3.95 $\pm$ 0.01
		ManipTrans	40.17 $\pm$ 0.07	35.20 $\pm$ 3.39	101.47 $\pm$ 2.08	98.49 $\pm$ 2.02
03ac9@0	right	SPIDER	44.27 $\pm$ 0.00	80.72 $\pm$ 0.00	4.80 $\pm$ 0.00	3.94 $\pm$ 0.00
		ManipTrans	44.31 $\pm$ 0.00	80.54 $\pm$ 0.00	293.28 $\pm$ 20.40	294.40 $\pm$ 20.62
07301@0	right	SPIDER	25.96 $\pm$ 0.02	27.95 $\pm$ 0.67	5.09 $\pm$ 0.01	4.05 $\pm$ 0.00
		ManipTrans	13.40 $\pm$ 4.65	137.87 $\pm$ 2.71	10.20 $\pm$ 1.02	10.03 $\pm$ 1.07
07bb1@1	right	SPIDER	31.09 $\pm$ 15.33	29.26 $\pm$ 3.27	4.86 $\pm$ 0.06	4.03 $\pm$ 0.07
		ManipTrans	34.43 $\pm$ 2.16	57.47 $\pm$ 40.98	11.12 $\pm$ 2.91	10.04 $\pm$ 3.18
0b3a0@6	left	SPIDER	26.88 $\pm$ 0.01	40.46 $\pm$ 0.16	4.79 $\pm$ 0.00	3.96 $\pm$ 0.00
		ManipTrans	25.64 $\pm$ 0.38	73.22 $\pm$ 10.33	8.05 $\pm$ 0.05	7.63 $\pm$ 0.05
0f4a7@1	left	SPIDER	5.99 $\pm$ 2.61	56.57 $\pm$ 10.51	4.57 $\pm$ 0.07	3.94 $\pm$ 0.09
		ManipTrans	21.33 $\pm$ 14.55	57.51 $\pm$ 12.53	9.93 $\pm$ 2.07	9.68 $\pm$ 2.29
0f900@0	left	SPIDER	20.56 $\pm$ 10.22	52.75 $\pm$ 31.28	4.45 $\pm$ 0.01	3.89 $\pm$ 0.04
		ManipTrans	11.66 $\pm$ 10.62	58.05 $\pm$ 41.20	7.19 $\pm$ 2.73	6.34 $\pm$ 3.06
155cb@0	right	SPIDER	45.45 $\pm$ 0.00	51.92 $\pm$ 0.00	4.89 $\pm$ 0.02	3.98 $\pm$ 0.01
		ManipTrans	45.41 $\pm$ 0.00	51.81 $\pm$ 0.00	54.05 $\pm$ 1.19	52.83 $\pm$ 1.17
235ac@0	right	SPIDER	22.06 $\pm$ 0.40	59.61 $\pm$ 2.24	4.94 $\pm$ 0.00	3.91 $\pm$ 0.00
		ManipTrans	17.04 $\pm$ 3.14	70.61 $\pm$ 44.86	21.54 $\pm$ 7.76	23.50 $\pm$ 6.88
2ddc8@2	left	SPIDER	14.00 $\pm$ 0.00	85.78 $\pm$ 0.00	4.60 $\pm$ 0.01	3.96 $\pm$ 0.01
		ManipTrans	14.03 $\pm$ 0.00	85.77 $\pm$ 0.00	6.40 $\pm$ 0.03	5.36 $\pm$ 0.02
380c7@0	left	SPIDER	11.36 $\pm$ 4.52	23.57 $\pm$ 12.31	5.82 $\pm$ 0.57	5.87 $\pm$ 0.81
		ManipTrans	16.97 $\pm$ 0.17	15.78 $\pm$ 8.39	19.06 $\pm$ 0.81	20.73 $\pm$ 0.51
39e5e@5	left	SPIDER	12.64 $\pm$ 0.01	33.06 $\pm$ 0.30	4.73 $\pm$ 0.02	4.01 $\pm$ 0.02
		ManipTrans	18.61 $\pm$ 4.90	99.13 $\pm$ 26.68	19.24 $\pm$ 2.30	18.77 $\pm$ 2.90
49793@0	right	SPIDER	21.04 $\pm$ 9.41	49.52 $\pm$ 3.41	4.64 $\pm$ 0.12	4.03 $\pm$ 0.18
		ManipTrans	23.99 $\pm$ 3.82	122.98 $\pm$ 20.92	1618.07 $\pm$ 215.12	1617.23 $\pm$ 214.88
70ca2@0	left	SPIDER	6.25 $\pm$ 4.36	85.93 $\pm$ 13.46	4.55 $\pm$ 0.02	3.99 $\pm$ 0.04
		ManipTrans	3.93 $\pm$ 0.12	50.81 $\pm$ 5.08	11.91 $\pm$ 3.74	10.67 $\pm$ 3.93
81a95@5	left	SPIDER	6.77 $\pm$ 5.14	44.97 $\pm$ 21.95	4.90 $\pm$ 0.13	3.71 $\pm$ 0.04
		ManipTrans	3.13 $\pm$ 2.85	38.57 $\pm$ 25.20	6.61 $\pm$ 2.34	4.88 $\pm$ 2.33
Overall	–	SPIDER	23.12 $\pm$ 12.91	54.44 $\pm$ 22.43	4.84 $\pm$ 0.31	4.08 $\pm$ 0.47
		ManipTrans	23.03 $\pm$ 12.93	71.75 $\pm$ 32.30	139.01 $\pm$ 388.16	138.51 $\pm$ 388.12

TABLE XV: **SPIDER** hyperparameters for the **DexMachina** (Genesis) experiments.

Parameter	Symbol	Value
<i>Simulation</i>		
Simulator	–	Genesis (GPU)
Simulation timestep	Δt_{sim}	0.02 s (50 Hz)
Control timestep	Δt_{ctrl}	0.2 s (5 Hz)
Knot point spacing	Δt_{knot}	0.2 s
Action dimensions	n_u	36
<i>Sampling-based Optimization</i>		
Planning horizon	H	0.4 s (20 sim steps)
Parallel samples	N_w	1024
Max sampling iterations	N	32
Early stopping threshold	ϵ	0.01
Early stopping patience	–	4 iterations
<i>Noise Schedule</i>		
Initial noise scale	σ_{first}	0.5
Final noise scale (horizon end)	σ_{last}	1.0
Noise decay per iteration	β	0.87
Wrist pos/rot noise scale	–	1.0
Joint noise scale	–	1.0
<i>Reward</i>		
Reward mode	–	RL training reward
Terminal reward scale	α_T	10.0
Sampling temperature	τ	1.0
<i>DexMachina-specific</i>		
Imitation reward weight	w_{imi}	0.3
Contact reward weight	w_{con}	3.0
BC reward weight	w_{bc}	0.3
Object joint stiffness	k_p	0.0 (no curriculum)
Object joint damping	k_v	0.0 (no curriculum)

TABLE XVI: Multi-seed comparison between **SPIDER** and **DexMachina** RL baseline on 7 bimanual tasks from the ARCTIC dataset. Both methods are evaluated with 5 seeds per task. Metrics: d_{pos} = object position distance (m), d_{rot} = object rotation distance (rad), d_{arti} = articulation distance (rad). Values reported as mean $\pm$ std. Best per task in **bold**.

Task	Method	d_{pos} (m) $\downarrow$	d_{rot} (rad) $\downarrow$	d_{arti} (rad) $\downarrow$
box-30-230	SPIDER	0.0172 $\pm$ 0.0069	0.0988 $\pm$ 0.0198	0.0929 $\pm$ 0.0801
	DexMachina	0.0906 $\pm$ 0.0373	0.3709 $\pm$ 0.1155	0.2897 $\pm$ 0.0522
ketchup-30-230	SPIDER	0.8917 $\pm$ 0.1311	1.7862 $\pm$ 0.1723	0.7836 $\pm$ 0.0850
	DexMachina	1.1865 $\pm$ 0.3021	2.0616 $\pm$ 0.1768	0.8057 $\pm$ 0.0806
ketchup-30-230-s01-u02	SPIDER	0.7573 $\pm$ 0.3397	1.5418 $\pm$ 0.3500	1.1717 $\pm$ 0.0227
	DexMachina	1.1569 $\pm$ 0.2231	2.0379 $\pm$ 0.2021	1.1863 $\pm$ 0.0851
laptop-30-230	SPIDER	0.0427 $\pm$ 0.0171	0.2155 $\pm$ 0.0726	1.0548 $\pm$ 0.5011
	DexMachina	0.0573 $\pm$ 0.0840	0.1980 $\pm$ 0.2344	0.4094 $\pm$ 0.3296
mixer-30-230	SPIDER	0.0176 $\pm$ 0.0098	0.1203 $\pm$ 0.0360	0.1440 $\pm$ 0.0118
	DexMachina	0.1211 $\pm$ 0.1563	0.7360 $\pm$ 0.2814	0.3654 $\pm$ 0.1230
notebook-30-230	SPIDER	0.9968 $\pm$ 0.0245	1.9491 $\pm$ 0.2248	0.7024 $\pm$ 0.0007
	DexMachina	0.4969 $\pm$ 0.0682	1.1062 $\pm$ 0.2850	0.7468 $\pm$ 0.1131
waffleiron-30-230	SPIDER	0.2224 $\pm$ 0.1585	1.0033 $\pm$ 0.2985	0.5528 $\pm$ 0.3142
	DexMachina	0.5113 $\pm$ 0.0682	1.1791 $\pm$ 0.0903	0.5480 $\pm$ 0.0753
Overall	SPIDER	0.4208 $\pm$ 0.4096	0.9593 $\pm$ 0.7562	0.6432 $\pm$ 0.3840
	DexMachina	0.5172 $\pm$ 0.4485	1.0985 $\pm$ 0.6854	0.6216 $\pm$ 0.2915