DEXBEYOND: SCALING DEXTEROUS MANIPULATION THROUGH ITERATIVE HAND-OBJECT INTERACTION GENERATION

Anonymous authors
Paper under double-blind review

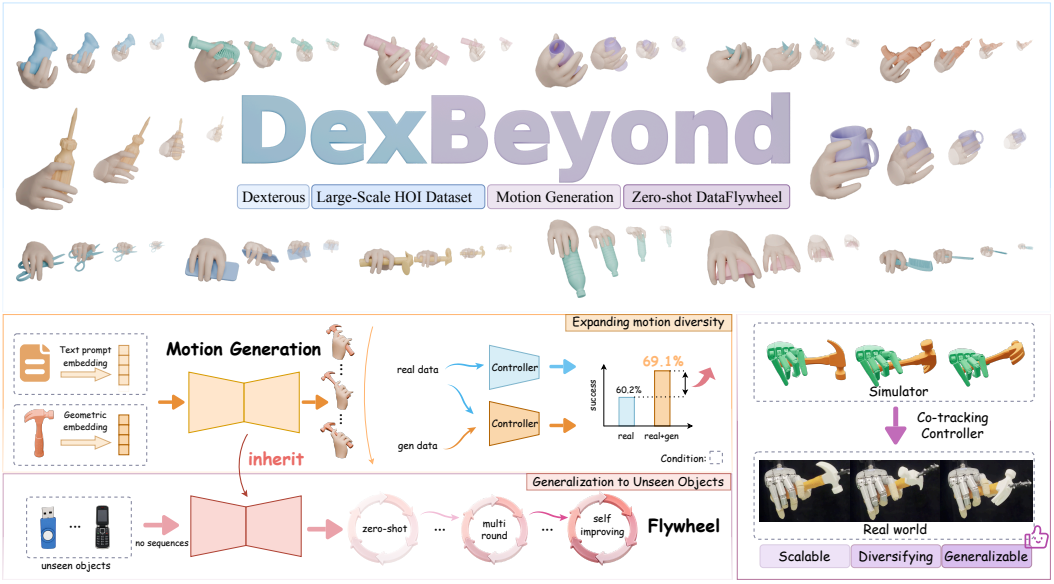

Figure 1: DexBeyond scales dexterous manipulation through a large-scale interaction dataset, generative expansion, and an iterative flywheel for unseen-object generalization.

ABSTRACT

Learning dexterous manipulation depends on continuous hand-object interaction data that capture fine-grained finger coordination. Many existing datasets emphasize grasping and placement, with limited coverage of both in-hand motion diversity and object variety. We present **DexBeyond**, a unified framework that combines a large-scale hand-object interaction dataset, a generative expansion model, and an iterative data flywheel to scale dexterous hand-object interaction data. We first construct **DexBeyond-D**, a large-scale hand-object interaction dataset comprising 20,891 sequences and 58.03 hours of valid hand-object interaction. It covers 219 object instances from 77 object categories and 30 manipulation behaviors, and provides synchronized 3D hand and object poses with the corresponding object meshes. Because exhaustive real-world capture is impractical, we develop **DexBeyond-G** to expand interaction diversity for captured objects and augment the dataset with generated motions, thereby improving downstream controller performance. Although DexBeyond-G can propose interactions for unseen objects by replacing its mesh condition, the resulting motions often exhibit penetration, lost contacts, and other physical inconsistencies. We therefore introduce **DexBeyond-FW**, an iterative, self-improving data flywheel that turns these proposals into feasible interaction data, enabling generalization to unseen objects without additional real-world collection. Experiments demonstrate that DexBeyond scales interaction synthesis across motions and objects, with downstream gains in simulation and the real world and progressive performance improvements on unseen objects.

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

1 INTRODUCTION

Dexterous manipulation (Rajeswaran et al., 2018; Chen et al., 2022; Pan et al., 2025; Hsieh et al., 2026; Yin et al., 2026) requires coordinated finger motion and continuous hand-object contact. Reorienting, regrasps, or repositioning tools within the hand requires coordinated changes in finger, object pose, and contact, often without releasing the object (Handa et al., 2023; Chen et al., 2023; Yin et al., 2023; Zhu et al., 2026b). Learning these behaviors calls for continuous trajectories that capture hand-object motion over time. Their value depends on both interaction quality and diversity.

The diversity has two dimensions: interaction diversity and object diversity. Interaction diversity spans different ways of manipulating an object, including in-hand reorientation, translation, and grasp transitions, whereas object diversity spans the geometries that these behaviors must accommodate. Capturing both at scale is costly because each new interaction-object pairing requires human demonstrations, motion capture, and processing (Brahmbhatt et al., 2020; Liang et al., 2026).

Existing hand-object datasets (Taheri et al., 2020; Chao et al., 2021; Liu et al., 2022; Zhan et al., 2024; Fan et al., 2023) provide valuable resources for grasping, placement, and task-level object interaction, including whole-body and bimanual activities. However, their collection objectives often prioritize particular tasks or activity contexts rather than jointly covering fine-grained, continuous in-hand interactions across diverse object geometries. More specialized in-hand datasets (Liang et al., 2026) capture detailed finger motion, but can remain limited in the diversity and scale of the interactions and objects they cover. Consequently, existing datasets do not yet provide sufficiently diverse and large-scale continuous hand-object interaction data for learning dexterous manipulation.

To address this gap, we introduce **DexBeyond**, a unified framework that combines a large-scale hand-object interaction dataset, a generative expansion model, and an iterative data flywheel to scale continuous, fine-grained in-hand interaction data, as shown in Fig. 1. We first construct **DexBeyond-D**, a large-scale hand-object interaction dataset comprising 58.03 valid hours across 77 object categories, 219 object instances, and 30 manipulation behaviors. It provides synchronized 3D hand and object poses together with the corresponding object meshes, covering continuous in-hand manipulation and object-specific functional actions across varied geometries.

Because even a large real-world collection cannot enumerate all interactions for every object, and further human demonstrations are costly, we develop **DexBeyond-G** to expand interaction and object coverage by generating hand-object sequences conditioned on object geometry and motion descriptions. The generated sequences can exhibit greater motion diversity than those in DexBeyond-D, and using them to augment the captured data improves downstream controller performance.

Replacing the conditioning mesh lets DexBeyond-G propose interactions for unseen objects, but without captured demonstrations for the target geometry, these proposals can exhibit penetration, lost contacts, and finger self-collisions. **DexBeyond-FW** addresses these issues through an iterative data flywheel. It generates candidate references, refines them geometrically, and validates their physical feasibility before training a controller and collecting rollouts. It then selects diverse successful rollouts to update the data pool and refine DexBeyond-G for the next round, generalizing interaction data to unseen objects without additional real-world demonstrations.

We evaluate captured and generated data along three dimensions: scale, physical plausibility, and interaction diversity. We also assess whether expanding interaction and object coverage improves downstream dexterous manipulation learning. Overall, our contributions are summarized as follows:

- We introduce DexBeyond-D, a large-scale hand-object interaction dataset of continuous in-hand interactions across diverse objects and behaviors.
- We develop DexBeyond-G, which generates interactions from object geometry and motion descriptions to expand motion diversity and improve downstream controller performance.
- We introduce DexBeyond-FW, an iterative flywheel that turns mesh-conditioned proposals into feasible robot interaction data and update DexBeyond-G using diverse rollouts on unseen objects.

2 RELATED WORK

Hand-Object Interaction Datasets. Existing datasets (Yang et al., 2022; Lu et al., 2025a; Fu et al., 2025) cover complementary aspects of hand-object interaction. DexYCB (Chao et al., 2021)

supports grasping and hand-object pose estimation, while ContactPose (Brahmbhatt et al., 2020) records grasps with paired contact and hand-object pose annotations. GRAB (Taheri et al., 2020) captures whole-body grasping with detailed hand motion and contact information, and HOI4D (Liu et al., 2022) provides egocentric task-level interactions across object categories. ARCTIC (Fan et al., 2023) focuses on bimanual manipulation of articulated objects, whereas OakInk2 (Zhan et al., 2024) organizes bimanual activities into affordances, primitive tasks, and complex tasks. Together, these datasets support the study of grasping, task-level object use, and bimanual manipulation.

Fine-grained finger motion poses additional requirements for data acquisition. HandX (Zhang et al., 2026b) extends hand-motion resources with new motion-capture sequences emphasizing finger articulation and bimanual coordination. DexCap (Wang et al., 2024) uses portable hand motion capture to collect human demonstrations for robot policy learning. More directly related to our scope, DexterCap (Liang et al., 2026) combines coded optical markers with automated reconstruction to build DexterHand, a dataset of fine-grained in-hand interactions with simple object primitives and a Rubik's Cube. Within these datasets, DexterHand is closest to our focus on in-hand interaction. DexBeyond-D builds on this focus by extending coverage to a broader range of object geometries, while providing synchronized 3D hand and object poses with the corresponding object meshes.

Data Generation for Dexterous Manipulation. Hand-object motion generation (Jiang et al., 2023; Li et al., 2024) extend a captured collection beyond its observed trajectories. Text2HOI (Cha et al., 2024) generates interactions from text and object geometry through contact prediction and hand refinement. DiffH2O (Christen et al., 2024) synthesizes one- or two-handed interactions from text and object geometry with separate grasping and manipulation stages. LatentHOI (Li et al., 2025) combines latent diffusion with a grasp variational autoencoder to synthesize interactions for unseen object geometries. OpenHOI (Zhang et al., 2026a) further considers open-world synthesis for novel objects under free-form instructions and long-horizon task decompositions. These methods study generation of human hand-object motion under semantic and geometric conditions.

Another line of work (Yin et al., 2025) expands robot-space demonstrations in simulation. MimicGen (Mandlekar et al., 2023) adapts source demonstrations to new contexts, and DexMimicGen (Jiang et al., 2025) extends this strategy to coordinated bimanual dexterous manipulation. DexFlyWheel (Zhu et al., 2026a) combines imitation learning, residual reinforcement learning, policy rollouts, and augmentation in an iterative data-generation loop. DexBeyond unifies geometry-conditioned synthesis and iterative robot-data expansion: DexBeyond-G augments interactions for captured objects, while DexBeyond-FW turns proposals for unseen objects into feasible robot data.

3 METHOD

DexBeyond scales dexterous manipulation data through three components. DexBeyond-D (Sec. 3.1) provides real-world, continuous hand-object interaction trajectories and object meshes. DexBeyond-G (Sec. 3.2) learns from this collection to synthesize additional interactions for captured objects. DexBeyond-FW (Sec. 3.3) further converts mesh-conditioned proposals for objects without captured demonstrations into feasible robot interaction data.

3.1 DEXBEYOND-D: A LARGE SCALE DATASET OF HAND-OBJECT INTERACTION

Data Scale. DexBeyond-D, as shown in Fig. 2, contains **20,891 sequences** and **6.26 million frames**, corresponding to **58.03 hours** of valid hand-object interaction recorded at 30 fps.

Object Coverage. The dataset includes **77 object categories** and **219 object instances**. The objects span three groups: geometric primitives, everyday rigid objects, and articulated and multi-part objects. Geometric primitives cover basic shapes such as cubes, cylinders, spheres, and prisms. Everyday rigid objects are further organized by their primary use into tools, office-use, kitchen-use, and daily-use objects. The articulated and multi-part group includes objects whose components move relative to one another. The complete category breakdown is provided in Table. 6.

Manipulation Coverage. DexBeyond-D covers **30 manipulation behaviors**. In-hand reorientation and translation are collected for every object, while functional behaviors are selected according

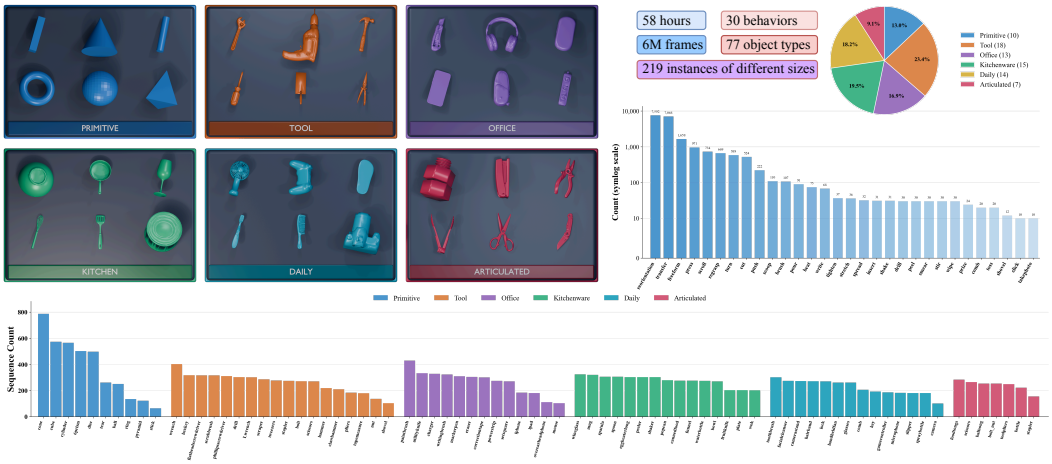

Figure 2: Overview of DexBeyond-D. Left: 3D object visualizations organized by group. Right and bottom: statistics of object counts, manipulation behaviors, and sequences per object.

to object use, such as cutting with a knife or pouring with a cup. We record these interactions under four wrist settings: palm-up, palm-sideways, palm-down, and free wrist motion. This design varies the direction of gravity relative to the hand while retaining object-specific manipulation patterns.

3D Reconstruction. We reconstruct the hand from these measurements using MANO (Romero et al., 2017). The resulting trajectories contain 3D hand poses and object poses. We denote the processed human hand-object dataset by $\mathcal{D}_{\mathrm{H}}$. Appendix A.1 details this reconstruction procedure.

3.2 DexBeyond-G: Hand-object interaction generation to expand diversity

Despite its scale, DexBeyond-D cannot exhaustively capture the valid ways each object can be manipulated. We therefore introduce **DexBeyond-G**, as shown on the left in Fig. 3, to expand interaction diversity by learning a conditional distribution over continuous interactions from $\mathcal{D}_{\mathrm{H}}$,

$$p_\phi(\mathbf{X} \mid \mathbf{c}_{\text{text}}, \mathbf{c}_{\text{shape}}), \tag{1}$$

where $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_T]$ contains the wrist, MANO joint, and object poses (Zhou et al., 2019). The conditions specify the desired object motion $\mathbf{c}_{\text{text}}$ and object geometry and scale $\mathbf{c}_{\text{shape}}$. We implement this distribution with a conditional denoising diffusion model (Ho et al., 2020): a Transformer decoder (Vaswani et al., 2017) maps a noisy motion and the condition tokens to the clean sequence, and iteratively denoises Gaussian noise at inference.

Condition Processing. To form $\mathbf{c}_{\text{text}}$, we verbalize discretized maximum object translation and rotation relative to the wrist in two temporal segments. We encode the resulting prompt with a frozen T5-base encoder (Raffel et al., 2020). Object geometry is represented by shape tokens from a frozen ReCon++ (Qi et al., 2023; 2024) point-cloud encoder together with a separate metric-scale token.

Training Objectives. We train DexBeyond-G with three complementary objectives,

$$\mathcal{L}_{\mathrm{G}} = \mathcal{L}_{\text{rec}} + \mathcal{L}_{\text{smooth}} + \mathcal{L}_{\text{geo}}, \tag{2}$$

where $\mathcal{L}_{\text{rec}}$ reconstructs the clean motion, while $\mathcal{L}_{\text{smooth}}$ regularizes hand-object dynamics.

The geometric term $\mathcal{L}_{\text{geo}} = \lambda_c \mathcal{L}_{\text{contact}} + \lambda_p \mathcal{L}_{\text{pen}}$, with hyperparameters λ_c and λ_p, preserves contacts and penalizes hand-object penetration. We query a metric signed-distance field (Tsang et al., 2019) from the predicted MANO surface in the object frame, which is positive outside the object. Let $\hat{d}_{t,v}$ and $d^*_{t,v}$ denote the predicted and captured signed distances of vertex v. We partition the palmar and lateral finger surfaces into 15 regions, let $\mathcal{V}_r$ denote the vertices in region r, and select up to the eight closest captured contact vertices in each region:

$$\mathcal{C}_{t,r} = \mathrm{TopK}_8\big(\{v \in \mathcal{V}_r : |d^*_{t,v}| \le \delta_g\}; |d^*_{t,v}|\big), \tag{3}$$

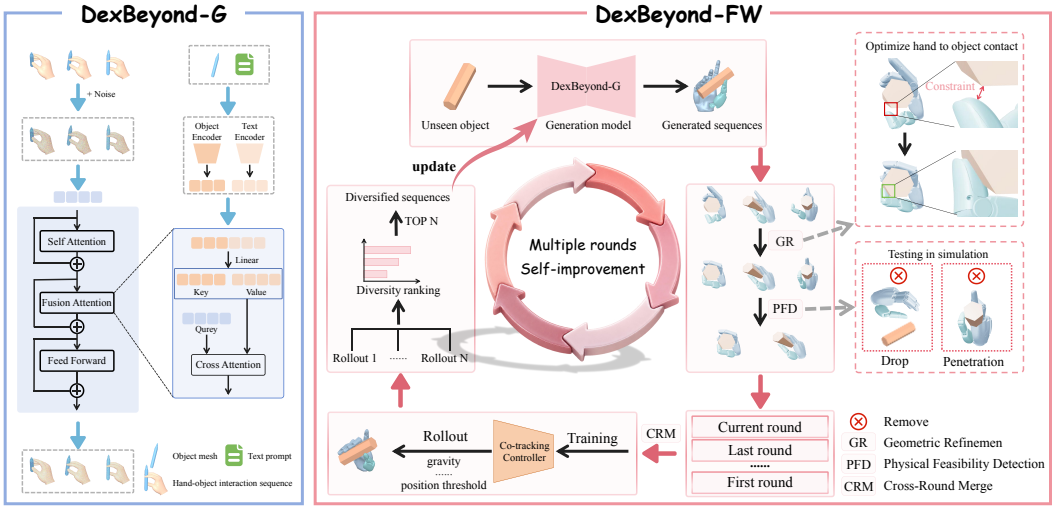

Figure 3: Overview of DexBeyond-G and DexBeyond-FW. Left: DexBeyond-G generates hand-object interactions from text and geometry conditions. Right: DexBeyond-FW refines and validates unseen-object proposals, trains a controller, and uses diverse rollouts to update the generator.

where $\delta_g = 4\,\mathrm{mm}$. Let $\mathcal{S}$ be the sampled frames, $\mathcal{V}$ the full MANO vertex set, $\mathcal{A} = \{(t,r) : t \in \mathcal{S}, |\mathcal{C}_{t,r}| > 0\}$ the active frame-region pairs, and ρ the Smooth-L_1 penalty; δ_c and δ_p are contact and penetration tolerances, and s is a distance scale. The two geometric losses are

$$\mathcal{L}_{\text{contact}} = \frac{1}{\max(1,|\mathcal{A}|)} \sum_{(t,r)\in\mathcal{A}} \frac{1}{|\mathcal{C}_{t,r}|} \sum_{v\in\mathcal{C}_{t,r}} \rho\left(\frac{[|\hat{d}_{t,v}| - \delta_c]_+}{s}\right), \tag{4}$$

$$\mathcal{L}_{\text{pen}} = \frac{1}{|\mathcal{S}||\mathcal{V}|} \sum_{t\in\mathcal{S}} \sum_{v\in\mathcal{V}} \rho\left(\frac{[-\hat{d}_{t,v} - \delta_p]_+}{s}\right), \tag{5}$$

where $[\cdot]_+ = \max(0,\cdot)$. The contact loss encourages vertices identified as contacts in the captured motion to remain within δ_c of the object surface, while the penetration loss penalizes any MANO vertex that penetrates the object by more than δ_p. Appendix A.2 provides implementation details.

3.3 DexBeyond-FW: Iterative Flywheel for Generalization to Unseen Objects

Replacing DexBeyond-G's mesh condition enables unseen-object proposals, but without demonstrations they may contain penetration and missing contacts. To address these issues, DexBeyond-FW, as shown on the right in Fig. 3, converts these proposals into feasible robot interaction data through an iterative generation–execution loop. It refines and filters the proposals, then uses the retained references to train a co-tracking controller and collect rollouts in simulation. Diverse successful rollouts are selected to update DexBeyond-G, which generates new proposals for the next round.

Unseen-Object Interaction Generation. We first retarget $\mathcal{D}_{\text{H}}$ to construct an initial robot-space dataset $\mathcal{D}_{\text{R}}^{(0)}$ by matching corresponding hand keypoints, then train a robot-space DexBeyond-G model $\mathcal{G}_{\text{R}}^{(0)}$ on it. At round r, we condition $\mathcal{G}_{\text{R}}^{(r)}$ on the object's mesh to generate candidate hand-object trajectories for unseen object. These candidates provide coarse hand-object trajectories.

Geometric Refinement and Physical Feasibility Detection. Geometric refinement (GR) adjusts robot joint configurations to improve the generated interaction while preserving the object trajectory. The objective penalizes hand-object penetration and inter-finger collisions while encouraging temporal smoothness, contact preservation, and proximity to the generated joint poses. Optimization proceeds in three stages: We first reduce severe penetration, then introduce contact and self-collision losses. Since these adjustments can make the initial contact candidates less suitable, we reselect them based on the refined hand pose and optimize with a stronger contact loss.

GR reduces penetration and improves contact, but does not guarantee physical feasibility. Physical Feasibility Detection (PFD) evaluates each frame by first checking whether hand-object penetration and inter-finger collisions exceed thresholds, then initializing the hand and object at the refined poses in Isaac Gym (Makoviychuk et al., 2021) and detecting object slipping or ejection. Frames that fail either check are removed, and sequences with fewer than a prescribed $N_{\min}$ valid frames are discarded, yielding physically feasible but temporally discontinuous hand-object references.

Cross-Round Merge and Rollout. To obtain continuous, physically feasible trajectories from these references, we use a co-tracking controller to execute them as discrete hand-object subgoals during simulation rollouts. At each round, cross-round data merge (CRM) combines refined hand-object trajectories from the current and previous rounds and ranks them separately by translational and rotational diversity. We select top-ranked trajectories from both rankings, ensuring a minimum number from each round, to form the policy-training set. The controller is initialized from the previous round's weights and trained on this set. During rollout, we vary gravity, tracking tolerances, and the number of consecutive steps required for subgoal success to further diversify the trajectories.

Diversity Ranking and Update. We first retain rollouts that complete at least $M = 5$ of subgoals. For a retained trajectory ξ, we quantify translational and rotational motion diversity by

$$d_{\text{trans}}(\xi) = \max_{i,j} \|\mathbf{p}_i^o - \mathbf{p}_j^o\|_2, \qquad d_{\text{rot}}(\xi) = \max_{i,j} d_{\text{SO}(3)}(\mathbf{R}_i^o, \mathbf{R}_j^o), \tag{6}$$

where $\mathbf{p}_i^o$ and $\mathbf{R}_i^o$ are the object position and orientation at frame i, and $d_{\text{SO}(3)}$ is the geodesic rotation distance. These scores measure the maximum object translation and rotation between frames of a rollout. For a nominal budget N, we select up to $N/2$ trajectories under each score and retain any trajectory selected by both rankings only once. Let $\mathcal{D}_{\text{R}}^{(r+1)}$ denote the diversity-ranked data selected above. We use these data to update the generator parameters, yielding $\mathcal{G}_{\text{R}}^{(r+1)}$ for the next round. Appendix A.3 provides implementation details for DexBeyond-FW.

4 Experiments

We evaluate DexBeyond in the order of its three components. We first assess DexBeyond-D in terms of scale, physical plausibility, interaction diversity, and downstream controller utility (Sec. 4.2). We then evaluate the ability of DexBeyond-G to expand motion diversity for captured objects and improve controller learning in simulation (Sec. 4.3) and the real world (Sec. 4.4). Finally, we evaluate DexBeyond-FW over four rounds on objects without captured demonstrations (Sec. 4.5).

4.1 Experimental Setup

Implementation Details. All experiments use NVIDIA RTX 5090 GPUs. We evaluate downstream policies on in-hand translation and reorientation, using an 80/20 sequence-level train/test split and 15 test initial states per sequence. Subgoals are spaced by a randomly sampled interval between 10 and 20 frames. DexBeyond-FW runs for four rounds; PFD retains only sequences with at least $N_{\min} = 80$ valid frames in round one and $N_{\min} = 120$ thereafter. For real-world validation, we deploy policies on SharpaWave, a 22-DoF five-finger anthropomorphic hand.

Comparison Datasets and Methods. We compare DexBeyond-D with OakInk2 (Zhan et al., 2024), HOI4D (Liu et al., 2022), ARCTIC (Fan et al., 2023), and DexterHand (Liang et al., 2026), which respectively target daily bimanual manipulation, egocentric 4D interaction, articulated-object manipulation, and fine-grained in-hand manipulation. For interaction generation, we additionally evaluate Text2HOI (Cha et al., 2024) and LatentHOI (Li et al., 2025) on DexBeyond-D.

Evaluation Metrics. We evaluate dataset quality along three dimensions: data scale, physical plausibility, and interaction diversity. Data scale is measured by valid interaction duration (Dur., h), source frame rate (FPS), object categories (Cat.), and instances (Inst.); physical plausibility by hand-object penetration (Pen., mm) and contact ratio (Cont., %). Over four-second windows, interaction diversity is measured by rotation span (RS, °), rotation cells (Cells), fine contact states (FS), state transitions (FST), persistent-contact transitions (PCT/s), and contact-motion transitions (CMT/s).

Table 1: Comparison with existing hand-object interaction datasets. DexBeyond-D stands out in data scale, physical plausibility (Physical), and interaction diversity, with broad object coverage and high-quality interactions. Arrows indicate whether higher or lower values are preferred.

Dataset	Data Scale				Physical		Interactive Diversity					
	Dur.	FPS	Cat.	Inst.	Pen.↓	Cont.↑	RS↑	Cells↑	FS↑	FST↑	PCT/s↑	CMT/s↑
OAKINK2	9.21	120	48	74	0.03	96.74	40.70	11.60	4.33	5.02	1.15	0.68
HOI4D	7.67	15	9	373	0.32	76.35	72.79	16.97	11.57	13.49	3.04	2.76
ARCTIC	2.02	30	11	11	0.07	75.72	109.18	23.47	15.66	18.11	3.98	2.89
DexterHand	1.24	20	6	8	0.08	97.92	110.15	22.55	25.09	29.39	7.30	3.22
DexBeyond-D (Ours)	58.03	30	77	219	0.06	99.85	104.29	27.16	22.45	28.37	7.05	3.88

Table 2: Simulation success rates for policies trained with real data and with real plus generated data. O1, O2, O3, O4, O5, O6, O7, O8, O9, and O10 denote the 6-prism, cylinder, facial cleanser, scrub brush, claw hammer, flat-head screwdriver, L-wrench, eraser, iPhone, and fruit knife, respectively.

Method	O1	O2	O3	O4	O5	O6	O7	O8	O9	O10
Real data	51.1%	86.7%	61.7%	42.2%	65.0%	58.3%	95.0%	40.0%	60.0%	42.2%
Real data + Gen data	65.6%	95.6%	70.0%	53.3%	68.3%	70.0%	100.0%	46.7%	75.0%	46.7%

RS captures the maximum relative rotation within a window, while Cells measures discretized rotational coverage. Appendix A.4 details the computation.

Downstream Controller Evaluation. We use TeleDexter (Li et al., 2026) as the downstream controller for all evaluations. We report the success rate (SR, %) over 20 consecutive subgoals. A rollout is successful only when all required subgoals satisfy the task-specific tracking criteria.

4.2 DexBeyond-D: Dataset Quality and Controller Utility

Comparison with Existing Datasets. Tab. 1 shows that, compared with DexterHand, the closest dataset in scope, DexBeyond-D provides $46.8\times$ more valid duration and covers $12.8\times$ more object categories. Its 58.03 hours of valid interaction also exceeds that of OakInk2, the second-largest dataset, by $6.3\times$. Despite this scale, DexBeyond-D achieves the highest contact ratio (99.85%) and the second-lowest penetration (0.056 mm). Its RS and rotation-cell scores characterize broad rotational coverage, and it attains the largest contact-motion transition rate. This combination of scale, reconstruction quality, and interaction diversity makes DexBeyond-D a strong foundation for hand-object interaction generation and dexterous manipulation learning.

Downstream Controller Utility. To assess the utility of DexBeyond-D for downstream controller, we randomly select ten objects, train a controller for each, and evaluate its SR. The baseline policies trained on DexBeyond-D reach an average SR above 60% across these objects, as reported in Tab. 2. This result demonstrates that DexBeyond-D provides sufficient coverage for learning continuous dexterous robot-hand manipulation policies.

4.3 DexBeyond-G: Motion Diversity and Controller Improvement

Interaction Diversity Expansion. To evaluate DexBeyond-G's ability to expand diversity, we construct an evaluation set of 1,850 real captured reference sequences (GT). We generate sequences using the same object and text conditions, then compare them with the captured references. As shown in Tab. 3, DexBeyond-G achieves higher scores on most interaction-diversity metrics than GT and other baselines, demonstrating its ability to expand diversity beyond the captured data.

Downstream Controller Improvement. We next compare policies trained with captured data alone and with captured data augmented by DexBeyond-G. The augmented controller improves SR on every evaluated object in Tab. 2, with the mean SR increasing from 60.2% to 69.1%. This im-

Table 3: Comparison and ablation of hand-object interaction generation methods on DexBeyond-D. DexBeyond-G broadens diversity beyond captured references. GT denotes real captured reference motions; "w/o" means without. Arrows indicate whether higher or lower values are preferred.

Method	Physical		Interactive Diversity					
	Pen.↓	Cont.↑	RS↑	Cells↑	FS↑	FST↑	PCT/s↑	CMT/s↑
GT	0.05	100.00	109.58	27.86	22.41	28.35	7.09	3.79
Text2HOI	0.06	71.21	77.41	15.45	2.62	3.34	0.37	0.20
LatentHOI	0.14	98.42	111.16	27.74	23.51	29.89	7.39	3.53
DexBeyond + Autoregressive	0.43	88.04	36.25	11.97	4.33	4.31	1.00	0.50
DexBeyond (Ours.)	0.11	92.51	154.84	33.04	26.12	31.45	7.28	4.98
+ w/o Geometric Loss	0.10	86.32	160.15	34.39	24.32	30.44	6.71	4.98
+ w/o Geometric Cond.	0.53	93.90	129.39	28.95	20.13	24.75	5.93	3.63

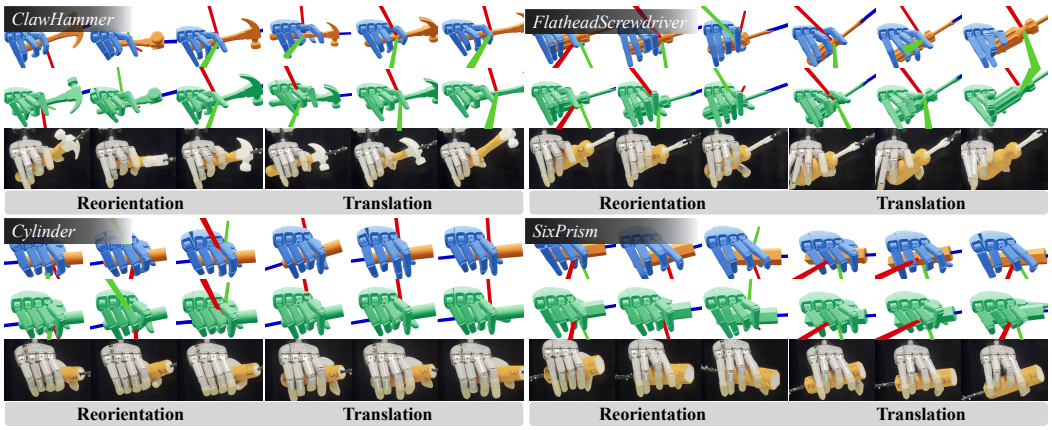

Figure 4: Controller execution in simulation (middle) and the real world (bottom), with reference subgoals at the top. Numbers distinguish rotations of symmetric objects. The objects' relative motions are reproduced in both settings.

provement shows that diversity expansion through DexBeyond-G translates into better downstream controller performance, rather than merely adding data without downstream utility.

4.4 Real-World Deployment

We select five representative objects for real-world validation on SharpaWave. Tab. 4 reports their deployment results. Augmenting the diversity with DexBeyond-G improves SR (mean SR from 42.1% to 54.4%). This result shows that the controller gains enabled by DexBeyond-G persist in the real world. Fig. 4 visualizes controller execution in simulation and the real world, showing that the controller follows the given subgoals in both settings.

Table 4: Real-world deployment SR, showing that controller gains from DexBeyond-G persist in the real world.

Method	O1	O2	O4	O5	O6
Real	35.2	72.0	33.4	40.6	29.2
Real + Gen	41.3	77.1	46.7	48.8	58.3

4.5 DexBeyond-FW: Generalization to Unseen Objects

We evaluate DexBeyond-FW on four randomly selected objects. To construct the training set, we exclude all demonstrations of the selected categories from DexBeyond-D and retarget the remaining motions to the robot hand, forming $\mathcal{D}_{\mathrm{R}}^{(0)}$ to train $\mathcal{G}_{\mathrm{R}}^{(0)}$. No captured demonstrations from the held-out categories are used throughout the entire process. The right panel of Fig. 6 shows the mean controller SR in each round, increasing from 8.3% in the first round to 40.7% in the fourth. These

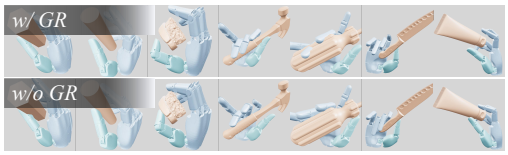

Figure 5: Generated interactions with geometric refinement (top) and without it (bottom).

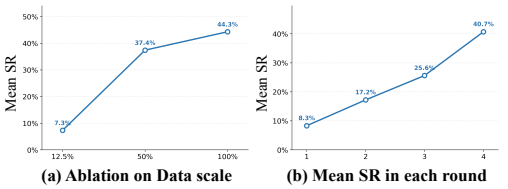

Figure 6: Ablation of DexBeyond-FW training-data scale (left) and number of rounds (right).

results show that DexBeyond-FW converts generated sequences into useful interaction data without collecting new demonstrations for the target objects, enabling generalization to unseen objects.

4.6 Ablation Studies

Ablation of DexBeyond-G Components. We ablate the generator architecture, geometric conditioning, and geometric loss of DexBeyond-G. As shown in Tab. 3, the diffusion model improves physical plausibility and all interaction diversity metrics over its autoregressive counterpart. Removing the geometric loss lowers contact quality and weakens contact-state diversity, indicating that reconstruction and smoothness alone do not sufficiently enforce hand-object structure. Removing geometric conditioning markedly increases penetration and degrades all diversity metrics, showing that object geometry is essential for generating interactions adapted to the target object.

Ablation of DexBeyond-FW Training Data. Due to cost, we conduct this ablation on three randomly selected objects. As shown in the first three rows of Tab. 5 and the left panel of Fig. 6, reducing DexBeyond-G's training set from 100% to 50% lowers SR (44.3% to 37.4%), while reducing it to 12.5% causes a sharp drop to 7.3%. Interaction diversity, averaged across rounds, also declines. These results suggest that broad data scale helps DexBeyond-G generalize across geometries.

Table 5: Ablation of DexBeyond-FW. DS denotes the fraction of object categories used to train DexBeyond-G; GR, PFD, and CRM denote geometric refinement, physical feasibility detection, and cross-round data merge.

Settings	SR↑	Cont.↑	RS↑	FST↑	PCT/s↑	CMT/s↑
Ours (100%)	44.3	93.5	143.61	33.66	8.16	5.17
DS (50%)	37.4	92.8	135.61	32.21	7.73	4.65
DS (12.5%)	7.3	90.7	122.76	23.03	5.52	2.99
w/o GR+PFD	19.2	91.4	135.47	32.82	7.70	4.34
w/o CRM	14.3	89.1	144.17	28.90	6.57	3.97

Ablation of DexBeyond-FW Components. The final two rows of Tab. 5 show that removing GR+PFD or CRM reduces SR from 44.3% to 19.2% or 14.3%, respectively. Without GR+PFD, interaction diversity remains relatively high, but unrefined and physically implausible samples can hinder effective controller learning. Without CRM, RS slightly increases, while FST, PCT/s, and CMT/s decrease, suggesting broad object motion but less varied hand-object contact patterns. This reflects overfitting to newly generated interactions when previous rounds are not merged. Fig. 5 compares interactions 'w/' and 'w/o' GR, illustrating how refinement improves hand-object contact.

5 Conclusions

We introduced DexBeyond, a framework for scaling dexterous manipulation data across interaction and object diversity, comprising three components: DexBeyond-D, DexBeyond-G, and DexBeyond-FW. DexBeyond-D provides 58.03 hours of continuous hand-object interaction across 77 object categories, 219 instances, and 30 behaviors. DexBeyond-G expands interactions for captured objects, improving downstream controller performance in simulation and the real world. While changing the geometry condition allows DexBeyond-G to propose interactions for unseen objects, these proposals can contain physical inconsistencies. DexBeyond-FW addresses this limitation by iteratively refining proposals, training controllers, and feeding diverse successful rollouts back to the generator. Experiments show that the framework combines large-scale, physically plausible, and diverse interaction data with improved control performance and progressive gains on unseen objects.

AI USE STATEMENT

In this work, we used generative AI tools to assist with translation and to provide feedback during research ideation, including discussions of research methodology and experimental design. We also used generative AI tools to draft parts of the manuscript and to improve the clarity and readability of the writing. We did not use generative AI tools to generate synthetic datasets, formulate or prove mathematical claims, or independently conduct experiments. All AI-assisted content was reviewed, verified, and revised by the authors. The authors take full responsibility for the final content of this work, including all text, claims, experimental results, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

Our data collection involves human participants performing hand-object interaction tasks in a motion-capture environment. All participants were informed of the purpose and procedure of the data collection and voluntarily participated in the study. The collected data are used solely for research on hand-object interaction and dexterous manipulation. To protect participant privacy, the dataset does not contain personally identifiable information. The dataset consists of processed hand-object interaction data, including 3D hand poses, object poses, and corresponding object meshes. Any subject-specific measurements used for hand reconstruction are used only for reconstructing the hand representation and are not intended for participant identification. We will release the dataset for research purposes with appropriatem documentation describing its intended use and limitations. The authors are responsible for ensuring that the data collection and release comply with the applicable institutional and legal requirements.

REFERENCES

Samarth Brahmbhatt, Chengcheng Tang, Christopher D Twigg, Charles C Kemp, and James Hays. Contactpose: A dataset of grasps with object contact and hand pose. In *European Conference on Computer Vision*, pp. 361–378. Springer, 2020.

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

Yu-Wei Chao, Wei Yang, Yu Xiang, Pavlo Molchanov, Ankur Handa, Jonathan Tremblay, Yashraj S Narang, Karl Van Wyk, Umar Iqbal, Stan Birchfield, et al. Dexycb: A benchmark for capturing hand grasping of objects. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9040–9049. IEEE, 2021.

Tao Chen, Jie Xu, and Pulkit Agrawal. A system for general in-hand object re-orientation. In *Conference on robot learning*, pp. 297–307. PMLR, 2022.

Tao Chen, Megha Tippur, Siyang Wu, Vikash Kumar, Edward Adelson, and Pulkit Agrawal. Visual dexterity: In-hand reorientation of novel and complex object shapes. *Science Robotics*, 8(84): eadc9244, 2023.

Sammy Christen, Shreyas Hampali, Fadime Sener, Edoardo Remelli, Tomas Hodan, Eric Sauser, Shugao Ma, and Bugra Tekin. Diffh2o: Diffusion-based synthesis of hand-object interactions from textual descriptions. In *SIGGRAPH Asia 2024 Conference Papers*, pp. 1–11, 2024.

Ke Fan, Shunlin Lu, Minyue Dai, Runyi Yu, Lixing Xiao, Zhiyang Dou, Junting Dong, Lizhuang Ma, and Jingbo Wang. Go to zero: Towards zero-shot motion generation with million-scale data. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 13336–13348. IEEE, 2025.

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

Rao Fu, Dingxi Zhang, Alex Jiang, Wanjia Fu, Austin Funk, Daniel Ritchie, and Srinath Sridhar. Gigahands: A massive annotated dataset of bimanual hand activities. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 17461–17474. IEEE, 2025.

Ankur Handa, Arthur Allshire, Viktor Makoviychuk, Aleksei Petrenko, Ritvik Singh, Jingzhou Liu, Denys Makoviichuk, Karl Van Wyk, Alexander Zhurkevich, Balakumar Sundaralingam, et al. Dextreme: Transfer of agile in-hand manipulation from simulation to reality. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5977–5984. IEEE, 2023.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Elvis Hsieh, Wen-Han Hsieh, Yen-Jen Wang, Toru Lin, Jitendra Malik, Koushil Sreenath, and Haozhi Qi. Learning dexterous manipulation skills from imperfect simulations. In *International Conference on Robotics and Automation (ICRA)*, 2026.

Biao Jiang, Xin Chen, Wen Liu, Jingyi Yu, Gang Yu, and Tao Chen. Motiongpt: Human motion as a foreign language. *Advances in Neural Information Processing Systems*, 36:20067–20079, 2023.

Zhenyu Jiang, Yuqi Xie, Kevin Lin, Zhenjia Xu, Weikang Wan, Ajay Mandlekar, Linxi Jim Fan, and Yuke Zhu. Dexmimicgen: Automated data generation for bimanual dexterous manipulation via imitation learning. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 16923–16930. IEEE, 2025.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015. URL https://arxiv.org/abs/1412.6980.

Kailin Li, Jingbo Wang, Lixin Yang, Cewu Lu, and Bo Dai. Semgrasp: Semantic grasp generation via language aligned discretization. In *European Conference on Computer Vision*, pp. 109–127. Springer, 2024.

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

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

Yutong Liang, Shiyi Xu, Yulong Zhang, Bowen Zhan, He Zhang, and Libin Liu. Dextercap: Affordable and automated capture of complex hand-object interactions. In *Computer Graphics Forum*, pp. e70330. Wiley Online Library, 2026.

Yunze Liu, Yun Liu, Che Jiang, Kangbo Lyu, Weikang Wan, Hao Shen, Boqiang Liang, Zhoujie Fu, He Wang, and Li Yi. Hoi4d: A 4d egocentric dataset for category-level human-object interaction. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20981–20990. IEEE, 2022.

Jiaxin Lu, Chun-Hao Paul Huang, Uttaran Bhattacharya, Qixing Huang, and Yi Zhou. Humoto: A 4d dataset of mocap human object interactions. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 10886–10897. IEEE, 2025a.

Shunlin Lu, Jingbo Wang, Zeyu Lu, Ling-Hao Chen, Wenxun Dai, Junting Dong, Zhiyang Dou, Bo Dai, and Ruimao Zhang. Scamo: Exploring the scaling law in autoregressive motion generation model. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 27872–27882. IEEE, 2025b.

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

Ajay Mandlekar, Soroush Nasiriany, Bowen Wen, Iretiayo Akinola, Yashraj Narang, Linxi Fan, Yuke Zhu, and Dieter Fox. Mimicgen: A data generation system for scalable robot learning using human demonstrations. In *7th Annual Conference on Robot Learning*, 2023.

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

Zekun Qi, Runpei Dong, Guofan Fan, Zheng Ge, Xiangyu Zhang, Kaisheng Ma, and Li Yi. Contrast with reconstruct: Contrastive 3d representation learning guided by generative pretraining. In *International Conference on Machine Learning*, pp. 28223–28243. PMLR, 2023.

Zekun Qi, Runpei Dong, Shaochen Zhang, Haoran Geng, Chunrui Han, Zheng Ge, Li Yi, and Kaisheng Ma. Shapellm: Universal 3d object understanding for embodied interaction. In *Computer Vision - ECCV 2024*, pp. 214–238, Berlin, Heidelberg, 2024. Springer-Verlag. ISBN 978-3-031-72774-0. doi: 10.1007/978-3-031-72775-7_13. URL https://doi.org/10.1007/978-3-031-72775-7_13.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67, 2020.

Aravind Rajeswaran, Vikash Kumar, Abhishek Gupta, Giulia Vezzani, John Schulman, Emanuel Todorov, and Sergey Levine. Learning complex dexterous manipulation with deep reinforcement learning and demonstrations. In *Proceedings of Robotics: Science and Systems*, Pittsburgh, Pennsylvania, June 2018. doi: 10.15607/RSS.2018.XIV.049.

Javier Romero, Dimitrios Tzionas, and Michael J. Black. Embodied hands: modeling and capturing hands and bodies together. *ACM Trans. Graph.*, 36(6), November 2017. ISSN 0730-0301. doi: 10.1145/3130800.3130883. URL https://doi.org/10.1145/3130800.3130883.

Omid Taheri, Nima Ghorbani, Michael J Black, and Dimitrios Tzionas. Grab: A dataset of whole-body human grasping of objects. In *European conference on computer vision*, pp. 581–600. Springer, 2020.

Clement Fuji Tsang, Maria Shugrina, Jean-Francois Lafleche, Or Perel, Charles Loop, Towaki Takikawa, Vismay Modi, Alexander Zook, Jiehan Wang, Wenzheng Chen, Tianchang Shen, Jun Gao, Krishna Murthy Jatavallabhula, Edward Smith, Artem Rozantsev, Sanja Fidler, Gavriel State, Jason Gorski, Tommy Xiang, Jianing Li, Michael Li, and Rev Lebaredian. Kaolin: A PyTorch Library for Accelerating 3D Deep Learning Research, 2019. URL https://github.com/NVIDIAGameWorks/kaolin.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Chen Wang, Haochen Shi, Weizhuo Wang, Ruohan Zhang, Li Fei-Fei, and Karen Liu. DexCap: Scalable and Portable Mocap Data Collection System for Dexterous Manipulation. In *Proceedings of Robotics: Science and Systems*, Delft, Netherlands, July 2024. doi: 10.15607/RSS.2024.XX.043.

Lixin Yang, Kailin Li, Xinyu Zhan, Fei Wu, Anran Xu, Liu Liu, and Cewu Lu. Oakink: A large-scale knowledge repository for understanding hand-object interaction. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20921–20930. IEEE, 2022.

Zhao-Heng Yin, Binghao Huang, Yuzhe Qin, Qifeng Chen, and Xiaolong Wang. Rotating without Seeing: Towards In-hand Dexterity through Touch. In *Proceedings of Robotics: Science and Systems*, Daegu, Republic of Korea, July 2023. doi: 10.15607/RSS.2023.XIX.036.

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

Zhao-Heng Yin, Guanya Shi, Pieter Abbeel, and C Karen Liu. Mana: Dexterous manipulation of articulated tools. *arXiv preprint arXiv:2606.13677*, 2026.

Xinyu Zhan, Lixin Yang, Yifei Zhao, Kangrui Mao, Hanlin Xu, Zenan Lin, Kailin Li, and Cewu Lu. Oakink2: A dataset of bimanual hands-object manipulation in complex task completion. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 445–456. IEEE, 2024.

Zhenhao Zhang, Ye Shi, Lingxiao Yang, Suting Ni, Qi Ye, and Jingya Wang. Openhoi: Open-world hand-object interaction synthesis with multimodal large language model. *Advances in Neural Information Processing Systems*, 38:166582–166612, 2026a.

Zimu Zhang, Yucheng Zhang, Xiyan Xu, Ziyin Wang, Sirui Xu, Kai Zhou, Bing Zhou, Chuan Guo, Jian Wang, Yu-Xiong Wang, and Liang-Yan Gui. Handx: Scaling bimanual motion and interaction generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2274–2284, June 2026b.

Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the continuity of rotation representations in neural networks. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5738–5746. IEEE, 2019.

Kefei Zhu, Fengshuo Bai, YuanHao Xiang, Yishuai Cai, Xinglin Chen, Ruochong Li, Xingtao Wang, Hao Dong, Yaodong Yang, Xiaopeng Fan, et al. Dexflywheel: A scalable and self-improving data generation framework for dexterous manipulation. *Advances in Neural Information Processing Systems*, 38:4236–4265, 2026a.

Xinghao Zhu, Zixi Liu, Shalin Jain, Chenran Li, Milad Noori, Michael Andres Lin, Huihua Zhao, John Welsh, Mrinal Verghese, Wei Liu, et al. Learning dexterous manipulation using contact wrench guidance from human demonstration. *arXiv preprint arXiv:2607.00033*, 2026b.

A APPENDIX

A.1 DEXBEYOND-D DETAILS

This appendix provides the acquisition and reconstruction details omitted from the main text.

Object Categories. Table 6 lists the object categories in the three groups described in Section 3.1. Fig. 8 shows more examples of interactive data.

Motion Capture Setup. We use an optical motion-capture workspace with 18 NOKOV cameras (Fig. 7(a)). A marker-based capture glove and reusable object-tracking mount track hand and object motion (Fig. 7(b–e)). The glove carries 24 reflective markers across the wrist, the metacarpophalangeal (MCP), proximal interphalangeal (PIP), and distal interphalangeal (DIP) joints, and the fingertips, providing dense observations of finger position during manipulation.

Reusable Object Tracking Mount. To accommodate objects of varied shapes and sizes, we use a reusable tracking mount instead of attaching markers directly to each object, which can cause inconsistent marker layouts and calibration errors. The mount comprises a **base mount**, a **snap-fit connector**, and a **tracking stem** (Fig. 7(c–e)); four reflective markers are rigidly attached to the stem, and the base is fixed to a predefined surface of the object. After calibrating the rigid transformation from the tracking stem to the object coordinate frame, we recover the object pose from the tracked marker assembly. The mount also keeps markers away from hand-object contact regions, reducing occlusion, particularly for small objects manipulated inside the hand.

Collection Protocol. All manipulation sequences are collected using the right hand. During collection, the operator maintains contact with the object and performs the predefined manipulation behavior under the corresponding collection setting. Each recorded sequence lasts approximately 10 seconds, yielding trajectories of the hand markers and object tracking markers for subsequent reconstruction and post-processing. The processed motion sequences are released at 30 fps.

Motion Reconstruction and Post-Processing. We flag hand-marker samples whose wrist-local coordinates exceed 1 m in absolute value and repair each marker trajectory independently. Interior gaps are linearly interpolated, while leading and trailing gaps are filled with the nearest valid sample; tracks with no valid samples are left unchanged. Object poses are copied unchanged.

Subject-Specific MANO Shape Calibration. We reconstruct the hand mesh using the MANO model (Romero et al., 2017). For each operator, we obtain an initial shape by interactively adjusting MANO's 10 shape coefficients to match its five finger-chain lengths to those measured from a reference motion-capture sequence; manually specified chain-length targets can be used when no reference sequence is available. We then refine the shared shape coefficients through alternating per-frame pose fitting and shape optimization against mapped motion-capture markers. For each operator, we randomly sample 1,132 manipulation sequences and uniformly select four frames from each sequence, resulting in 4,528 frames per operator. We optimize the shape using Adam (Kingma & Ba, 2015) with joint limits, then fix it for all sequences from that operator.

MANO Pose Fitting. The reflective markers are attached to the surface of the capture glove, while the motion-capture system provides only their 3D positions. We therefore recover the MANO pose by fitting the corresponding points on the MANO surface to the observed marker positions.

Let $\mathbf{p}_{t,i} \in \mathbb{R}^3$ denote the observed 3D position of the i-th mocap marker at frame t, and let $\hat{\mathbf{p}}_{t,i}$ denote the corresponding point on the MANO surface. The reconstructed hand at frame t is parameterized by the articulated hand pose $\boldsymbol{\theta}_t$, global wrist rotation $\mathbf{R}_t^w$, global wrist translation $\mathbf{t}_t^w$, and the fixed subject-specific shape $\boldsymbol{\beta}$. We establish a fixed correspondence between each mocap marker and a location on the MANO surface. Anatomical joint limits are additionally imposed during fitting to prevent implausible configurations and excessive joint rotations.

We adopt a two-stage coarse-to-fine fitting strategy. In the first stage, we optimize the global wrist rotation, then jointly refine it with the hand pose while keeping the wrist translation fixed. In the second stage, we first refine the wrist rotation and translation from wrist-anchor markers with the

Table 6: Object groups and corresponding object categories in DexBeyond-D. Everyday rigid objects are further organized by their primary use.

Object group	Subgroup	Categories
Geometric primitives	Primitives	sixprism, ball, cone, cube, cylinder, disc, pyramid, ring, stick, tear
Everyday rigid objects	Tools	Lwrench, bolt, clawhammer, drill, flatheadscrewdriver, hammer, hexkey, nut, phillipsscrewdriver, pliers, scraper, scrubbrush, shovel, tapemeasure, tweezers, wrench, scissors, stapler
	Office-use	charger, correctiontape, eraser, ipad, iphone, markerpen, mouse, powerstrip, setsquare, paintbrush, utilityknife, writingbrush, overearheadphone
	Kitchen-use	bowl, cannedfood, eggbeaterlong, fruitknife, funnel, mug, peeler, plate, popcan, shaker, spatula, spoon, waterbottle, wineglass, wok
	Daily-use	camera, camerastand, comb, facialcleanser, gamecontroller, glasses, hairband, handheldfan, key, lock, microphone, slipper, spraybottle, toothbrush
Articulated	Articulated	balisong, bolt_nut, bottle, foodtongs, scissors, stapler, toolpliers



(a) Optical motion-capture workspace (b) Capture glove (c) Base mount (d) Snap-fit connector (e) Tracking stem

Figure 7: Overview of the motion capture system.

hand pose fixed, then refine the hand pose with the wrist transform fixed. We use a robust marker fitting loss, defined at frame t as

$$\mathcal{L}_{\text{marker}}^{\text{D},t} = \frac{1}{3N_{\text{m}}} \sum_{i=1}^{N_{\text{m}}} \sum_{k=1}^{3} \rho_h(\hat{p}_{t,i,k} - p_{t,i,k}), \tag{7}$$

where $N_{\text{m}} = 24$ is the number of hand markers, k indexes the spatial coordinates, and ρ_h is the Huber loss:

$$\rho_h(e) = \begin{cases} \frac{1}{2}e^2, & |e| \le h, \\ h\left(|e| - \frac{1}{2}h\right), & |e| > h. \end{cases} \tag{8}$$

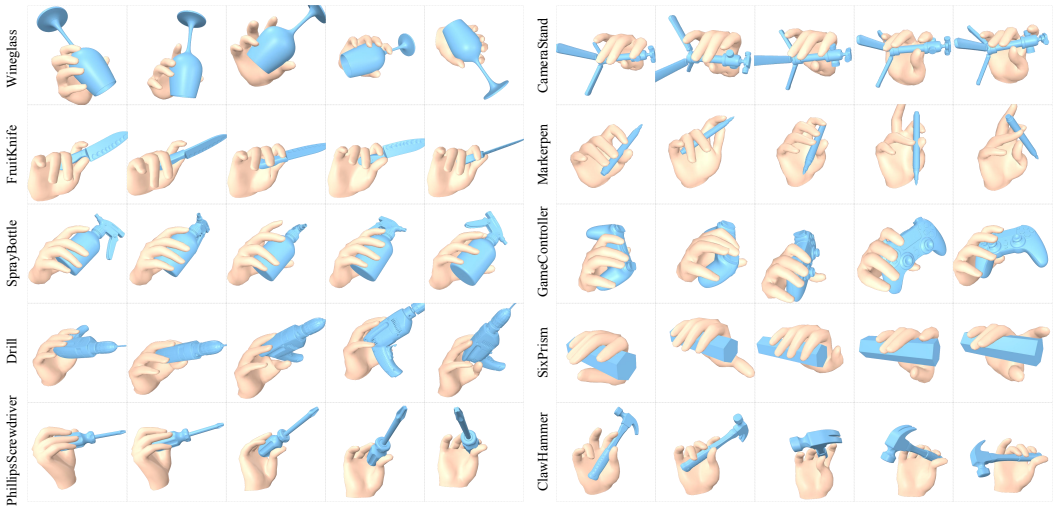

Figure 8: Visualization of more interactive data in DexByond-D.

Marker coordinates are expressed in meters, and we set $h = 0.01$ m. During wrist refinement, the marker term is restricted to the wrist anchors and uses normalized marker weights together with wrist rotation and translation priors.

Contact-Aware Post-Optimization. Marker-based fitting preserves the captured hand motion but does not explicitly model the geometric relationship between the hand and object. Consequently, the reconstructed hand may exhibit small penetrations into the object or unrealistic gaps around contact regions. We therefore introduce a contact-aware post-optimization stage to refine the reconstructed trajectories. Let $\boldsymbol{\theta}_t^0$ denote the articulated MANO hand pose obtained from the preceding fitting stage and $\boldsymbol{\theta}_t$ denote the refined pose. The fitted global wrist rotation and translation remain fixed during this stage; j indexes articulated-pose coordinates. The overall optimization objective is

$$\mathcal{L}_{\text{post}}^{\text{D}} = 10\mathcal{L}_{\text{marker}}^{\text{D}} + \mathcal{L}_{\text{pose}}^{\text{D}} + 50\mathcal{L}_{\text{pen}}^{\text{D}} + 5\mathcal{L}_{\text{contact}}^{\text{D}} + \mathcal{L}_{\text{smooth}}^{\text{D}}. \tag{9}$$

The marker loss preserves consistency with the original motion-capture observations:

$$\mathcal{L}_{\text{marker}}^{\text{D}} = \frac{1}{T}\sum_{t=1}^{T}\mathcal{L}_{\text{marker}}^{\text{D},t}, \tag{10}$$

where T denotes the sequence length. To prevent the optimized motion from deviating excessively from the initially fitted trajectory, we introduce a pose-preservation loss:

$$\mathcal{L}_{\text{pose}}^{\text{D}} = \text{mean}_{t,j}\left(\theta_{t,j} - \theta_{t,j}^0\right)^2. \tag{11}$$

For hand-object penetration, let $\mathcal{V}_t$ denote the set of MANO vertices at frame t, and let $d(\mathbf{v}, \mathcal{O}_t)$ denote the signed distance from a hand vertex $\mathbf{v}$ to the object surface $\mathcal{O}_t$. We define the signed distance to be positive outside the object and negative inside the object. The penetration loss is defined as

$$\mathcal{L}_{\text{pen}}^{\text{D}} = \frac{1}{\sum_{t=1}^{T}|\mathcal{V}_t|}\sum_{t=1}^{T}\sum_{\mathbf{v}\in\mathcal{V}_t}\max\left(0, -d(\mathbf{v}, \mathcal{O}_t)\right). \tag{12}$$

For contact refinement, we select candidate vertices once from the MANO contact regions, as shown in Fig. 9 (b). Let $\mathcal{C}_t$ contain the 20% of hand vertices with the smallest signed distances among those



(a) Visualization of sharpa contact regions in four views

(a) Visualization of mano contact regions in four views

Figure 9: Visualization of the contact regions between Sharpa and Mano.

satisfying $d(\mathbf{v}, \mathcal{O}_t) < 0.03$ m. Thus, the candidate set can include vertices inside the object as well as nearby exterior vertices. The contact loss penalizes candidate vertices that lie farther than a 1 mm margin outside the object:

$$\mathcal{L}^{\mathrm{D}}_{\mathrm{contact}} = \frac{1}{\sum_{t=1}^{T} |\mathcal{C}_t|} \sum_{t=1}^{T} \sum_{\mathbf{v} \in \mathcal{C}_t} \max(0, d(\mathbf{v}, \mathcal{O}_t) - m)\,, \tag{13}$$

where $m = 0.001$ m. If no candidate vertices are selected, the contact loss is set to zero. Finally, temporal smoothness is encouraged by penalizing pose changes between adjacent frames:

$$\mathcal{L}^{\mathrm{D}}_{\mathrm{smooth}} = \mathrm{mean}_{t,j} \left(\theta_{t,j} - \theta_{t-1,j}\right)^2, \tag{14}$$

The resulting reconstruction pipeline converts raw marker trajectories into temporally coherent MANO motion while preserving the captured movement and improving hand-object geometric consistency. We denote the resulting processed human hand-object trajectory dataset by $\mathcal{D}_{\mathrm{H}}$.

A.2 Additional Details of DexBeyond-G

Motion Representation and Diffusion Architecture. Each frame $\mathbf{x}_t \in \mathbb{R}^{108}$ contains the wrist translation relative to the initial wrist frame (3D), wrist rotation in the continuous 6D representation, rotations of 15 MANO joints in 6D (90D), and object translation (3D) and rotation (6D) in the current wrist frame. Given a clean sequence $\mathbf{X}_0$, the forward diffusion process samples

$$\mathbf{X}_k = \sqrt{\bar{\alpha}_k}\mathbf{X}_0 + \sqrt{1-\bar{\alpha}_k}\boldsymbol{\epsilon}, \qquad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \tag{15}$$

and the model predicts $\hat{\mathbf{X}}_0 = f_\phi(\mathbf{X}_k, k, \mathbf{c})$, where $\mathbf{c} = (\mathbf{c}_{\mathrm{text}}, \mathbf{c}_{\mathrm{shape}})$. We project each noisy frame to a temporal token and use a Transformer decoder with cross-attention to the condition tokens. The model has 12 Transformer layers, a hidden dimension of 512, four attention heads, 1,000 diffusion steps, and a cosine noise schedule.

Training Windows and Motion Descriptions. We randomly crop 200-frame windows, corresponding to approximately 6.67 seconds at 30 fps. Shorter sequences are padded by repeating their final valid frame, and padded frames are excluded from all training losses. To construct the motion description, we select the start, middle, and end keyframes and divide a window into two segments. For segment i spanning $[\tau_i, \tau_{i+1}]$, the maximum relative translation and rotation are

$$a_i^{\mathrm{trans}} = \max_{t,u \in [\tau_i, \tau_{i+1}]} \left\| \mathbf{p}_t^{o|w} - \mathbf{p}_u^{o|w} \right\|_2, \quad a_i^{\mathrm{rot}} = \max_{t,u \in [\tau_i, \tau_{i+1}]} \cos^{-1}\left(\frac{\mathrm{tr}\left(\mathbf{R}_t^{o|w} (\mathbf{R}_u^{o|w})^\top\right) - 1}{2} \right). \tag{16}$$

We discretize these magnitudes into motion-intensity levels and verbalize the two segments in temporal order. The middle boundary is randomly perturbed during training and fixed during validation.

For the geometry condition, we sample 1,024 points from the object mesh, center the point cloud, and normalize it to the unit sphere. A frozen ReCon++ (Qi et al., 2024) encoder maps the point cloud to 16 shape tokens. Since normalization removes metric size, an MLP maps the three-dimensional object extent in meters to a separate scale token. Another MLP maps the 10-dimensional MANO shape vector to the hand-shape condition.

Temporal Training Objectives. Let $\mathbf{h}_t$ and $\mathbf{o}_t$ denote the hand-related 99 dimensions and object-related 9 dimensions of $\mathbf{x}_t$, respectively, and let m_t be the valid-frame mask. The reconstruction objective is the masked mean squared error between the predicted and clean motion. The temporal objective is

$$\begin{aligned} \mathcal{L}_{\text{smooth}} &= \lambda_v \mathcal{L}_{\text{vel}} + \lambda_a \mathcal{L}_{\text{acc}} + \lambda_v^o \mathcal{L}_{\text{obj-vel}} + \lambda_a^o \mathcal{L}_{\text{obj-acc}}, \\ \mathcal{L}_{\text{vel}} &= \text{MSE}_m\left(\Delta\hat{\mathbf{h}}_t, \Delta\mathbf{h}_t\right), \qquad \mathcal{L}_{\text{acc}} = \text{MSE}_m\left(\Delta^2\hat{\mathbf{h}}_t, \mathbf{0}\right), \\ \mathcal{L}_{\text{obj-vel}} &= \text{MSE}_m(\Delta\hat{\mathbf{o}}_t, \Delta\mathbf{o}_t), \qquad \mathcal{L}_{\text{obj-acc}} = \text{MSE}_m\left(\Delta^2\hat{\mathbf{o}}_t, \mathbf{0}\right). \end{aligned} \tag{17}$$

We use $\lambda_v = \lambda_v^o = 0.1$ and $\lambda_a = \lambda_a^o = 0.05$. Thus, the velocity terms preserve the temporal changes of the captured sequence, while the acceleration terms suppress high-frequency variation in the generated hand and object trajectories.

Geometric-Loss Implementation. For the geometric losses in Equations 4 and 5, we set the contact tolerance to $\delta_c = 0.9\,\text{mm}$, the penetration tolerance to $\delta_p = 1\,\text{mm}$, and the distance scale to $s = 10\,\text{mm}$. We evaluate both objectives on 10 stratified frames from each sequence and on a subset of 16 sequences in each batch. We use $\lambda_c = 2 \times 10^{-5}$ and $\lambda_p = 10^{-4}$. Both losses are linearly warmed up during the first 30,000 optimization steps and further weighted at diffusion step k by $[1 - 2k/(K-1)]_+$, which disables geometric supervision in the noisier half of the schedule.

Autoregressive Variant. For the architecture comparison in Section 4.3, we also implement a discrete autoregressive generator (Fan et al., 2025; Lu et al., 2025b). A temporal convolutional autoencoder with a finite scalar quantization bottleneck compresses each 200-frame sequence into 50 discrete codes from a vocabulary of 4,096 entries. Its decoder reconstructs the continuous 108D motion. After training the tokenizer, we freeze it and train an eight-layer causal Transformer to model

$$p_\phi(\mathbf{z} \mid \mathbf{c}) = \prod_{\ell=1}^{L} p_\phi(z_\ell \mid z_{<\ell}, \mathbf{c}), \tag{18}$$

where the text, geometry, scale, and MANO-shape embeddings form the condition prefix. The Transformer is trained with teacher forcing and next-code cross-entropy. To apply geometric supervision, we convert its logits to finite-scalar-quantization values with a straight-through argmax estimator, decode them to continuous motion, and evaluate the contact and penetration losses every four optimization steps.

Frame-Conditioned Variants. We additionally implement auxiliary variants conditioned on the first frame or on both the first and last frames. Each supplied frame is projected to a typed condition token. In the diffusion model, known frames are clamped to their prescribed values during forward noising and every reverse step, providing hard boundary constraints. In the autoregressive model, frame tokens are appended to the condition prefix and provide soft guidance. These variants are not used in the main experiments.

A.3 Additional Details of DexBeyond-FW

Geometric Refinement. For each generated trajectory, we optimize the robot joint configurations while keeping the wrist and object trajectories fixed. Unlike the reconstruction of captured human motion, this stage does not use a marker loss because the generated trajectories already lie in robot space. The objective combines hand-to-object penetration, temporal smoothness, deviation from the generated joint poses, contact preservation, and inter-finger self-collision. We measure self-collision with collision spheres attached to different fingers. For a sphere pair (i, j) with centers $\mathbf{s}_i, \mathbf{s}_j$ and radii r_i, r_j, the penalized overlap is

$$[\alpha(r_i + r_j) - \|\mathbf{s}_i - \mathbf{s}_j\|_2 - \delta]_+, \tag{19}$$

where α scales the sphere radii and δ is the allowed margin. The self-collision loss sums these overlaps across inter-finger pairs. When resolving an overlap, gradients are stopped through the higher-priority finger so that the lower-priority finger is adjusted, using the order thumb, index, middle, ring, and little finger.

The first optimization stage prioritizes hand-object penetration while regularizing temporal smoothness and deviation from the generated joint poses; fingers without substantial penetration are anchored more strongly. The second stage introduces contact and self-collision losses. Contact candidates are selected at the start of this stage and held fixed during optimization. The third stage reselects candidates from the updated hand pose and optimizes again with stronger contact recovery.

Physical Feasibility Detection. We evaluate each refined frame independently in Isaac Gym. The geometric checks measure the maximum penetration of sampled hand points into the object, the average of the deepest 32 sampled points per finger averaged across fingers, the summed inter-finger sphere overlap, and both the maximum and top-32 mean penetration of sampled object-surface points into the hand. The current object-specific configurations limit these quantities to 5–$7\,\mathrm{mm}$, 2–$3\,\mathrm{mm}$, 2–$3\,\mathrm{mm}$, and 5–$7\,\mathrm{mm}$ for each of the two object-to-hand measures, respectively.

For the physical check, we initialize the hand and object at the refined poses, reset their velocities, and simulate 10 or 20 physics steps, depending on the object configuration, while holding the hand at the reference configuration. A frame is rejected if the final object position differs from its initial position by more than $5\,\mathrm{cm}$. A displaced object is classified as ejected if, within the first three simulation steps, its speed exceeds $1\,\mathrm{m/s}$, its upward velocity exceeds $0.3\,\mathrm{m/s}$, or its displacement exceeds $3\,\mathrm{cm}$; otherwise, it is classified as slipped.

Cross-Round Merge and Controller Rollout. At round $r > 1$, Cross-Round Merge gathers the feasible interactions from all rounds available up to r. It rejects trajectories that are too short, computes their maximum pairwise object translation and rotation, and ranks the remaining trajectories separately by the two measures. The policy-training set is the deduplicated union of the two ranked subsets. A minimum quota per round can supplement the global rankings so that earlier or later rounds are not removed entirely.

Each rollout starts from a random frame of a feasible reference and tracks hand-object subgoals spaced by 10–20 frames. A subgoal is completed when the object pose and fingertips remain within their tracking tolerances for the required consecutive steps; the total number completed provides the initial filter before diversity ranking.

A.4 Evaluation Metric Details

Data Scale and Aggregation. Valid duration sums valid source-frame intervals, taking the temporal union when views, hands, or converted clips overlap. FPS reports the source sampling rate, computed as the reciprocal of the median timestamp interval. For metric computation, we subsample OakInk2 from its native 120 Hz rate to 30 Hz by taking every fourth frame. This limits the effect of high-rate frame-to-frame jitter on rotational statistics. Cat. and Inst. count distinct object categories and instances. We compute temporal metrics over complete four-second windows whose start times are one second apart. Window scores are averaged within each sequence, then averaged equally across sequences with valid scores.

Physical Plausibility. Let $d_{t,v}$ be the signed distance from MANO vertex v to the object surface, positive outside. A frame is included in penetration evaluation if any vertex has $d_{t,v} < -0.1\,\mathrm{mm}$. For each sequence, Pen. is the mean raw penetration depth $[-d_{t,v}]_+$ over all MANO vertices on the included frames, including vertices with zero depth. It is zero if no frame qualifies. The 0.1 mm tolerance selects frames; it is not subtracted from the measured depth.

For contact, each MANO vertex is assigned to the region with its largest skinning weight, giving one palm region and three regions per finger. We estimate vertex areas by equally distributing each adjacent triangle's area among its vertices. A vertex is near the object when $-5\,\mathrm{mm} \le d_{t,v} \le 2\,\mathrm{mm}$, and a region is active when near-object vertices cover at least 2% of its area. We debounce region-state changes with a 0.1 s persistence rule over the full sequence before windowing. Cont. is the time-weighted percentage of each window with at least one active region.

Interaction Diversity. For a four-second window with T frames, let $A_t = R_{w,t}^{\top} R_{o,t}$ be the object orientation relative to the wrist. Rotation span (RS) is the maximum pairwise geodesic angle among

these orientations:

$$\mathrm{RS} = \frac{180}{\pi} \max_{1 \le i,j \le T} d_{\mathrm{SO}(3)}(A_i, A_j), \qquad d_{\mathrm{SO}(3)}(A_i, A_j) = \cos^{-1}\left(\frac{\mathrm{tr}(A_i^\top A_j) - 1}{2}\right), \tag{20}$$

RS is reported in degrees and captures the window's rotational extent without accumulating frame-to-frame increments. For Cells, we express each orientation relative to the first frame as $A_t A_1^\top$ and assign it to the codebook entry with the largest absolute quaternion inner product. The codebook contains the identity and rotations about 42 fixed Fibonacci-sphere axes, with 10° angular-shell intervals. Cells counts the distinct entries visited, complementing RS with discretized rotational coverage. For translation metrics, we transform the object's surface-area centroid into the wrist frame at each timestamp.

The 16 binary contact-region labels at each frame form a fine contact state. FS counts distinct states per window, including the no-contact state; FST counts state-change events, not distinct transition types. A transition is persistent when at least one finger region is active in both neighboring states; palm-only contact does not qualify. PCT/s is the number of persistent transitions per four-second window, divided by four seconds. CMT/s counts the subset for which the object translates more than $5\,\mathrm{mm}$ or rotates more than 10° relative to the wrist between representative midpoints of the neighboring stable-state runs, also divided by four seconds. Events are assigned to windows by their transition timestamps.