One Hand to Rule Them All: Canonical Representations for Unified Dexterous Manipulation

Zhenyu Wei Yunchao Yao Mingyu Ding
University of North Carolina at Chapel Hill

Fig. 1: We introduce a canonical hand representation that unifies diverse dexterous hands into a shared parameter space and canonical URDF format, serving as a condition for cross-embodiment policy learning. It enables dexterous grasping and zero-shot generalization to novel hand morphologies, highlighting its potential for a wide range of dexterous manipulation tasks.

***Abstract*—Dexterous manipulation policies today largely assume fixed hand designs, severely restricting their generalization to new embodiments with varied kinematic and structural layouts. To overcome this limitation, we introduce a parameterized canonical representation that unifies a broad spectrum of dexterous hand architectures. It comprises a unified parameter space and a canonical URDF format, offering three key advantages. 1) The parameter space captures essential morphological and kinematic variations for effective conditioning in learning algorithms. 2) A structured latent manifold can be learned over our space, where interpolations between embodiments yield smooth and physically meaningful morphology transitions. 3) The canonical URDF standardizes the action space while preserving dynamic and functional properties of the original URDFs, enabling efficient and reliable cross-embodiment policy learning.**

We validate these advantages through extensive analysis and experiments, including grasp policy replay, VAE latent encoding, and cross-embodiment zero-shot transfer. Specifically, we train a VAE on the unified representation to obtain a compact, semantically rich latent embedding, and develop a grasping policy conditioned on the canonical representation that generalizes across dexterous hands. We demonstrate, through simulation and real-world tasks on unseen morphologies (*e.g.*, 81.9% zero-shot success rate on 3-finger LEAP Hand), that our framework unifies both the representational and action spaces of structurally diverse hands, providing a scalable foundation for cross-hand learning toward universal dexterous manipulation. Project Page: https://zhenyuwei2003.github.io/OHRA/

I. Introduction

Dexterous robotic hands with high degrees of freedom (DoF) and anthropomorphic designs offer advanced flexibility and control beyond the capabilities of simpler parallel or underactuated grippers [7, 4], enabling robots to interact with diverse objects in dynamic and unstructured environments. Recent advances in both learning-based and model-based approaches have driven impressive progress in object grasping [29, 13, 28, 41, 25, 22], dynamic in-hand manipulation [6, 5, 27, 2, 37], and tool use for goal-directed behaviors [1, 14, 12, 40, 19, 32, 9]. However, the specific DoF requirements and structural layouts vary substantially across tasks and robotic platforms. As a result, existing methods generally remain tailored to specific robotic hands, limiting their generalization and reusability across embodiments.

Variations in morphology, DoF, and kinematic layouts make policies trained on one hand design difficult to transfer to another [32]. This lack of interoperability increases the cost of data collection and prevents leveraging heterogeneous datasets across platforms. Recent studies have begun exploring cross-embodiment manipulation: $\mathcal{D}(\mathcal{R}, \mathcal{O})$ Grasp [29] transfers grasping skills across hands by modeling hand–object interactions, but remains task-specific; DexUMI [32] uses the human hand as a universal interface, yet its reliance on human-like kinematics and teleoperation setups limits scalability; and particle-based dynamics learning methods [10] model both hands and objects as particle systems, enabling transfer for deformable manipulation but struggling with structurally distinct hands and broader manipulation skills. Despite these efforts, a central challenge remains: *How to establish a unified representation and action space that policies can generalize across different hand embodiments and tasks?*

Fig. 2: Comparison of canonical and original URDFs across five dexterous hands with different finger numbers and handedness. For each hand (from left to right): canonical URDF, original URDF, overlay of initial poses, and overlay of grasp poses, showing close morphological and kinematic consistency between the canonical and original models.

Addressing this question requires solving two fundamental challenges: (i) obtaining a compact yet expressive representation of hand morphology that can serve as a conditioning input for learning-based models, and (ii) establishing a unified action space that allows policies to function seamlessly across hands with varying DoFs and kinematic structures. Although the URDF format encodes a complete specification of hand geometry and kinematics, its hierarchical and heterogeneous nature makes it ill-suited for direct use in neural networks for learning-based approaches. This motivates us to introduce a parameterized canonical representation, which encodes hand structure in a learning-friendly format while standardizing the action space across diverse dexterous hands.

Building on this canonical representation, we encode the geometric and kinematic properties of dexterous hands into a fixed set of parameters defined under a canonical URDF format. This format can approximate a wide range of robotic hand designs while maintaining a unified action space, where inactive joints are treated as dummy variables. Through an automated pipeline, existing hand URDFs are converted into this canonical format and corresponding parameter representation, yielding two unified spaces: (i) a parameter space that compactly captures hand morphology, and (ii) an action space that standardizes joint movements across embodiments. This unified formulation enables joint policy training across diverse hand embodiments, bridging their structural and dimensional discrepancies.

We validate our framework through three key experiments. First, training a variational autoencoder (VAE) on the canonical parameter space reveals a structured latent manifold for hand geometry and kinematics, where interpolation between different embodiments produces smooth, physically meaningful morphology transitions. Second, grasp policy replay and in-hand reorientation tasks show that the canonical URDF preserves the kinematic and functional characteristics of the original hands, achieving performance comparable to their native URDFs. Finally, conditioning on hand morphology representations, we train a cross-embodiment dexterous grasping policy that generalizes across diverse hand designs. We first demonstrate effective policy transfer across three distinct robotic hands, where the shared policy outperforms per-hand baselines. We further scale this approach by training on over one hundred LEAP Hand variants, enabling zero-shot generalization to unseen hand morphologies and robust grasping performance in both simulation and real-world experiments, without additional fine-tuning.

Our main contributions are summarized as follows:

1) We propose a canonical representation for dexterous hands that standardizes diverse morphologies and kinematic structures into a unified parameterized format, enabling consistent and learning-friendly structural encoding across hands.
2) Extensive experiments including morphology latent interpolation, grasp policy replay, in-hand reorientation, and unseen cross-hand grasping demonstrate that the canonical format not only faithfully preserves functional behavior but also provides a unified action space for zero-shot and effective policy transfer across embodiments.
3) For the first time, we establish an interpretable, interpolable, and scalable representation foundation that enables joint policy training for cross-embodiment dexterous manipulation, paving the way for unified large-scale and morphology-aware learning.

II. Related Works

Dexterous Manipulation. High-DoF robotic hands provide the articulation needed for fine-grained contact control and multi-stage manipulation, supporting tasks from stable grasping to dynamic in-hand reconfiguration and tool-mediated interactions [11, 42, 5, 29, 13, 26, 31, 14]. These capabilities have been extensively explored through analytic models and data-driven learning, producing strong performance on individual embodiment. In particular, reinforcement learning and imitation learning have yielded high-quality grasping and manipulation policies when confined to a fixed dexterous hand [38, 6, 39, 40]. However, because such policies are optimized for the specific kinematics, actuation, and workspace of a single hand, they become tightly coupled to that embodiment [33, 25]. This embodiment-specific specialization limits transfer to other hands and prevents the reuse of demonstrations across heterogeneous hardware, leaving progress fragmented across isolated hand designs rather than advancing toward methods that generalize to new embodiments.

Cross-Embodiment Policy Learning. Recent work has increasingly explored how to share manipulation abilities across robotic hands with distinct morphologies [29, 13, 32, 10, 30, 8, 3, 34, 16]. Much of this progress has centered on grasping. One line of work focuses on intermediate grasp representations that abstract away embodiment-specific kinematics. These methods leverage representations such as interaction-centric fields or contact patterns to enable grasp transfer across different robotic hands, but remain largely confined to grasp synthesis and do not naturally extend to sequential manipulation skills [29, 13]. Beyond grasping, some approaches target more general cross-embodiment behaviors through higher-level or embodiment-agnostic interfaces. Human-centric representations treat the human hand as a universal manipulation prior but typically assume human-like kinematics and require specialized hardware mappings [32]. Particle-based dynamics learning provides an alternative by representing hands and objects as particle systems, yet is mainly applicable to structurally similar hands and constrained manipulation tasks [10]. Despite these efforts, a unified cross-embodiment paradigm capable of supporting general manipulation across heterogeneous robotic hands is still lacking.

III. Canonical Hand Representation Design

Overview. To enable learning policies that generalize across dexterous hands with different morphologies, we propose a parameterized canonical hand representation, which serves as the foundation for cross-embodiment manipulation. The goal is to express diverse robotic hands within a unified structural and kinematic framework that can be efficiently processed by learning-based models.

We begin by motivating the need for a canonical representation (Sec. III-A), highlighting the limitations of existing URDF formats and the challenges of defining a consistent and learnable description for heterogeneous dexterous hands. We then present the design of the canonical URDF (Sec. III-B), which captures shared human-inspired kinematic structure while enforcing consistent coordinate conventions. Next, we define the canonical parameter set that encodes key morphological and kinematic properties in a compact and interpretable form (Sec. III-C). We further describe the automatic parsing process that converts arbitrary hand URDFs into this canonical parameterization and generates standardized URDF models (Sec. III-D). Finally, we establish a unified action space that aligns control dimensions across hands with different degrees of freedom (Sec. III-E), enabling a single policy to act consistently over diverse embodiments.

A. Motivation

To develop a task-agnostic policy that generalizes across dexterous hands with diverse embodiments, the learning framework must incorporate a representation of the current hand as part of the model input. Equally important is a consistent action space across hands, enabling a single policy to operate seamlessly over different morphologies.

A natural approach is to describe each hand using general 3D representations, such as point clouds or signed distance fields. While these capture detailed geometry, they only reflect static structure and ignore the kinematic and dynamic properties crucial for manipulation. In contrast, the Unified Robot Description Format (URDF) encodes comprehensive information about a hand's morphology and motion capabilities, including joint topology, link dimensions, and actuation limits. However, URDFs are typically handcrafted, vary across platforms, and are tree-structured and heterogeneous, making them difficult to use directly in learning-based pipelines.

TABLE I: Comparison of representative dexterous robotic hands and their kinematic configurations. F, A, and R denote flexion, abduction/adduction, and axial rotation. DoF values in parentheses include passive or mechanically coupled joints.

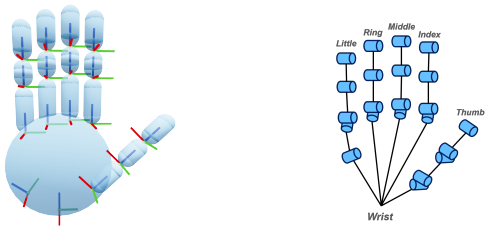
Name	DoF	Joint Types				
		Thumb	Index	Middle	Ring	Little
Shadow Hand	22	RAAFF	AFFF	AFFF	AFFF	FAFFF
Sharpa Wave	22	AFAFF	AFFF	AFFF	AFFF	FAFFF
WUJI Hand	20	FAFF	AFFF	AFFF	AFFF	AFFF
DexHand 021	12(19)	AFFF	AFFF	FFF	AFFF	AFFF
Orca Hand	16	FAFF	AFF	AFF	AFF	AFF
Gaia Hand	15	AFF	AFF	AFF	AFF	AFF
XHAND1	12	AFF	AFF	FF	FF	FF
Inspire	6(12)	AFFF	FF	FF	FF	FF
Allegro	16	ARFF	RFFF	RFFF	-	RFFF
LEAP Hand	16	ARFF	AFFF	AFFF	-	AFFF
Barrett	8	FF	AFF	AFF	-	-
Dex3	7	RFF	FF	FF	-	-

To overcome these limitations, we introduce a canonical URDF representation that unifies the structural and kinematic description of dexterous hands. Each hand is represented as a compact set of parameters capturing key geometric and motion attributes, from which a standard URDF can be automatically generated. This canonical form preserves both structural and functional consistency while providing a format suitable for neural network input. It enables policies to be conditioned on hand morphology and executed in a shared action space, laying the foundation for scalable cross-hand generalization.

B. Canonical URDF Design

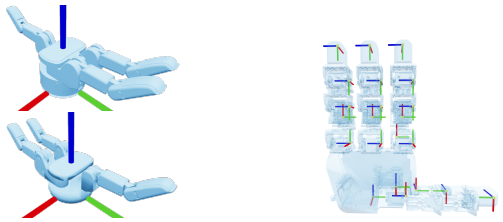
The canonical URDF unifies the structural representation of diverse dexterous hands by capturing shared kinematic principles in a standardized yet expressive format. The main challenge is selecting essential morphological parameters while remaining compatible with heterogeneous joint configurations and mechanical layouts. Preserving all native URDF details introduces numerous coordinate-dependent variables and inconsistent representations, while an overly compact model would constrain expressiveness. The canonical formulation therefore strikes a balance between representational richness, generality, and physical realizability.

1) Kinematic Analysis of Existing Dexterous Hands: To guide the canonical design, we analyzed a representative set of commercial and open-source dexterous hands, focusing on kinematic organization and degrees of freedom (DoF) (Table I). Despite mechanical differences, most designs share a human-inspired layout with recurring motion patterns. The thumb typically has two distal flexion joints and, when present, a separate abduction–adduction joint, while proximal thumb



(a) Mesh and frame visualization (b) Kinetic skeleton diagram

Fig. 3: Structure of the canonical URDF. A right-hand configuration is shown for clarity, but the representation is applicable to both left- and right-handed hands.



(a) Global frame inconsistency (b) Local frame ambiguity

Fig. 4: Coordinate frame inconsistencies in URDFs. (a) Global orientations vary across sources, (b) local joint frames use inconsistent axis definitions, leading to kinematic ambiguity.

joints exhibit greater variability due to mechanical design. Remaining fingers generally have an abduction–adduction proximal joint followed by a flexion chain dominating grasping, with some higher-DoF hands adding extra flexion for the little finger to approximate human-like coupled motion.

Guided by these observations, we define a canonical URDF supporting up to five fingers and 22 DoF (Fig. 3), capturing the shared human-like topology. To maintain geometric consistency across different link meshes, all links are represented as capsule primitives, which also reflect the cylindrical shape of most hands' collision meshes. This abstraction reduces geometric complexity while preserving the essential kinematic relationships required for dexterous manipulation.

2) Unified Coordinate Conventions: Dexterous hand URDFs vary in global and local coordinate frames. Even identical hardware can have different global axes (Fig. 4a) or incompatible local joint orientations (Fig. 4b), causing unnecessary transformations and ambiguity. Our canonical URDF enforces a unified convention aligned with human-hand kinematics (Fig. 3a): the palm normal along $+x$, the thumb (right hand) along $+y$, and other fingers along $+z$, with local axes x for abduction–adduction, y for flexion–extension, and z for axial rotation when applicable. This standardized convention ensures consistent, interpretable kinematics and enables parameterization that captures key morphological and kinematic variations for cross-hand policy learning.

C. Parameter Definition

Building on the canonical URDF design, we define a compact parameter set for dexterous hands that captures key morphological and kinematic information. The canonical representation comprises 82 parameters, while a more comprehensive URDF template with 173 parameters is provided to support specialized hand designs. Full details and design rationale are given in Appendix A. By consolidating geometric and kinematic attributes into this structured representation, the canonical model captures the most salient variations across embodiments and facilitates efficient learning and transfer among diverse dexterous hands.

D. URDF Parsing and Generation

To support a wide range of articulated hand designs, we develop an automatic framework for URDF parsing and generation. The framework extracts canonical parameters from arbitrary hand URDFs and reconstructs morphology and kinematics in the canonical space from minimal manual inputs, while also enabling full URDF generation from these parameters. Generation is implemented using the Jinja2 dynamic templating language [17], allowing conditional inclusion of elements and automatic adaptation to hands with varying numbers of fingers or links. Together, these capabilities provide consistent bidirectional conversion between diverse robot models and the unified representation (Sec. III-B). Full details of the parsing and generation procedures are provided in Appendix B.

E. Action Space Unification

With the canonical URDF, all dexterous hands share a fixed 22-DoF structure defining a unified control and observation space. Hands with fewer active DoF have the corresponding joints set inactive and the associated links removed, maintaining a consistent joint and link representation across embodiments. Using the joint-to-joint mapping from parsing, we enable bidirectional conversion between original and canonical joint vectors with consistent indexing and sign conventions. This unified formulation standardizes joint dimensionality and ordering, providing a coherent action space for scalable cross-hand policy learning and transfer.

IV. Canonical Representation Applications

We evaluate the proposed canonical representation through four complementary studies. First, we examine whether the parameter space introduced in Sec. III-C induces a continuous and interpretable latent manifold (Sec. IV-A) using a variational autoencoder trained on diverse sampled hand morphologies. Second, we assess the physical fidelity of the canonical URDF by comparing in-hand object reorientation performance against the original hand models (Sec. IV-B), verifying that the canonical formulation preserves key kinematic and control properties. Third, leveraging the unified structure and action representation, we train a single grasping policy transferable across multiple dexterous hands (Sec. IV-C), demonstrating scalability and cross-embodiment generalization. Finally, we investigate the zero-shot grasping capability enabled by hand conditioning using LEAP Hand and its variants (Sec. IV-D), assessing generalization to unseen hand morphologies.

A. Learning a Morphology Latent Space

To evaluate whether the canonical parameterization defines a compact and structured embedding, we train a variational

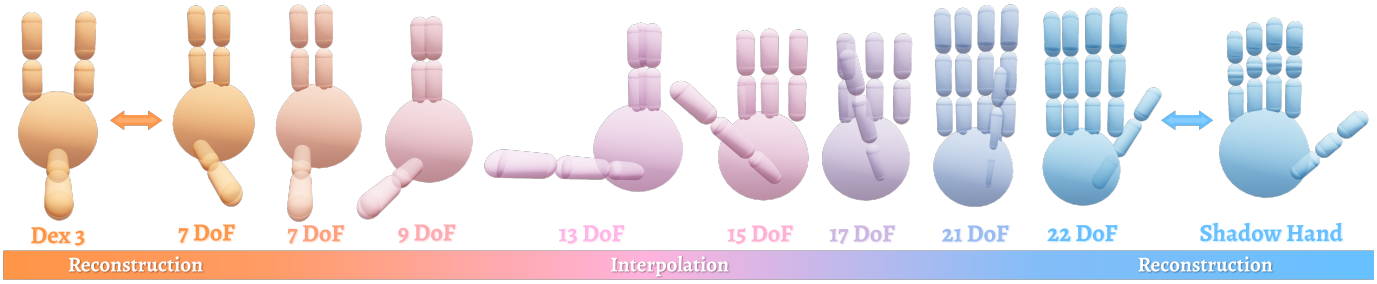


Fig. 5: Visualization of latent-space interpolation between two dexterous hands. Canonical URDFs are shown at the ends, with decoded reconstructions and interpolated morphologies in between, demonstrating smooth transitions in DoF, finger arrangement, and overall geometry.

autoencoder (VAE) to map hand morphology parameters into a 16-dimensional latent space. The latent representation captures essential geometric and kinematic variations across dexterous hands, forming a smooth manifold that supports interpolation and generalization.

A dataset of 65,536 synthetic hand configurations is generated by sampling each canonical parameter within physically feasible ranges. Continuous morphology parameters (palm/finger radii, link lengths) are sampled from bounded intervals, finger-origin translations preserve plausible proportions, joint axes are encoded as one-hot vectors over six canonical directions $(\pm x, \pm y, \pm z)$, and joint availability is encoded as binary indicators over the 22 canonical DoFs. The VAE reconstructs these parameters using type-specific losses with KL regularization. Full model architecture, training objectives, and hyperparameters are provided in Appendix C.

B. In-hand Reorientation

We evaluate the physical fidelity of the canonical representation using an in-hand object rotation task, where reinforcement learning agents are trained to rotate an object about a predefined axis. Separate policies are learned for the original dexterous hands and their canonical counterparts. Experiments are conducted in the IsaacGym simulator [15] using the Shadow Hand [20] and the LEAP Hand [23], following prior in-hand rotation setups [18, 23]. Observations include joint positions and velocities, PD controller targets, and the object's pose and velocities. The reward promotes high object angular velocity while penalizing object drift, excessive hand motion, and torque usage. Policies are trained with Proximal Policy Optimization (PPO) [21] using MLP-based agents. Full training details are provided in Appendix D.

C. Cross-Embodiment Dexterous Grasping

To evaluate the transferability of control across different robotic hands, we train a cross-embodiment grasp pose prediction model that operates entirely within the canonical representation. A grasp pose is defined by the wrist pose (T, R) relative to the object frame, parameterized by translation and rotation, together with the hand joint configuration θ.

As shown in Fig. 6, grasp generation follows a two-stage formulation inspired by Zhang et al. [41]. In the first stage,

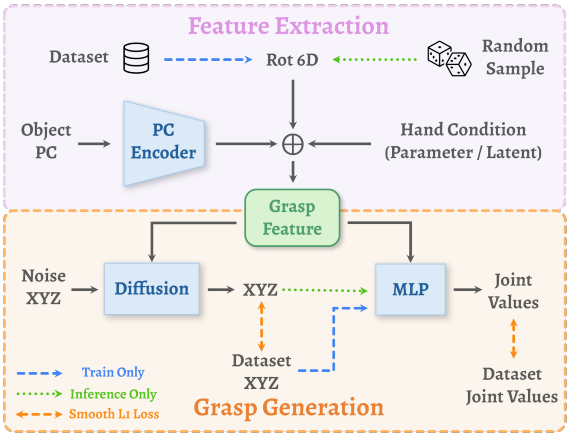


Fig. 6: Two-stage cross-embodiment grasp generation pipeline.

a diffusion-based conditional generator predicts the distribution of wrist translations around an object, conditioned on a grasp feature f_g and an explicitly provided wrist rotation R. Providing R separately decouples orientation from translation, allowing direct control over grasp direction and supporting diverse floating grasps. Training data spans multiple object orientations, enabling the model to generalize to arbitrary directions while supporting orientation-constrained sampling at test time. The second stage uses a lightweight MLP to predict the corresponding hand joint configuration θ conditioned on the sampled wrist pose (T, R) and f_g, implementing a deterministic mapping.

The grasp feature f_g integrates both object and hand morphology information. Object features are extracted from the input point cloud using the $\mathcal{D}(\mathcal{R}, \mathcal{O})$ point-based encoder, while hand features are obtained from the frozen VAE latent embedding introduced in Sec. IV-A. This allows the model to generalize grasp predictions across hands with different morphologies.

The training objective combines diffusion-based translation prediction with deterministic joint regression, both optimized using a Smooth-L_1 loss:

$$\mathcal{L} = \text{SmoothL1}(\hat{T}, T) + \text{SmoothL1}(\hat{\theta}, \theta). \quad (1)$$

D. LEAP Hand Zero-Shot Generalization

To further demonstrate the zero-shot generalization capability enabled by hand conditioning, and considering both

the limited availability of existing dexterous hand designs and the feasibility of real-world experiments, we adopt the open-source, modular LEAP Hand hardware. By systematically varying the presence of individual links for each finger, we construct $4^4 = 256$ LEAP Hand variants, denoted as `leap_xyzw`, where $x, y, z, w \in \{0, 1, 2, 3\}$ correspond to the number of links of the thumb, index, middle, and little fingers, respectively. For example, `leap_3333` corresponds to the original LEAP Hand design.

Under our canonical hand representation, different LEAP Hand variants can be generated by simply modifying the corresponding morphology parameters, allowing efficient and scalable instantiation of a large number of hand designs. Grasp data for each variant are generated using Lightning Grasp [36], where a dedicated configuration file is specified for each hand morphology. The generated grasps are further filtered using the data filtering strategy from $\mathcal{D}(\mathcal{R}, \mathcal{O})$ Grasp [29] to ensure physical plausibility. See Appendix F for more details.

To balance practical grasp feasibility and the evaluation of zero-shot generalization, we construct the grasping dataset using 66 LEAP Hand variants satisfying $x + y + z + w \geq 8$. The grasping policy is trained following the same procedure as in Sec. IV-C, with the hand condition directly specified by the canonical morphology parameters. Zero-shot generalization is then evaluated on LEAP Hand variants with $x+y+z+w < 8$, as well as additional variants whose grasp data are entirely excluded during training, allowing us to assess generalization to unseen hand morphologies without further fine-tuning.

V. Experiments

We evaluate the effectiveness of the proposed canonical URDF representation and the applications described in Sec. IV. Our experiments focus on four key aspects: (1) the quality and continuity of the learned morphology latent space, (2) the fidelity and dynamics preserved when motions are expressed using the canonical hand model, (3) the generalization of a single unified grasping policy across dexterous hands with diverse kinematic structures, and (4) the zero-shot capability enabled by hand conditioning. We further deploy the zero-shot grasping policy on physical hardware, demonstrating that models trained in simulation with the canonical model maintain strong performance in real-world grasping tasks.

A. VAE Latent Space Visualization

We first examine whether the variational autoencoder (VAE) introduced in Sec. IV-A learns a structured, continuous embedding of dexterous-hand morphology. To assess this, we visualize latent-space interpolations between two hands with distinct geometric and kinematic designs, such as a compact three-finger gripper and a high-DoF anthropomorphic hand.

Given latent features z_a and z_b for the two canonical encodings, we interpolate using $z(\alpha) = (1 - \alpha)z_a + \alpha z_b$ for $\alpha \in [0, 1]$, decode each feature into canonical URDF parameters, and generate the corresponding hand models. As shown in Figure 5, the interpolated structures evolve smoothly, with gradual variation in palm size, finger count and spacing, thumb placement, and DoF, indicating that the VAE captures a continuous morphological representation.

These results suggest that the latent space preserves intrinsic structural relationships among diverse hand designs, which is critical for downstream tasks such as morphology-conditioned grasping and cross-hand policy transfer.

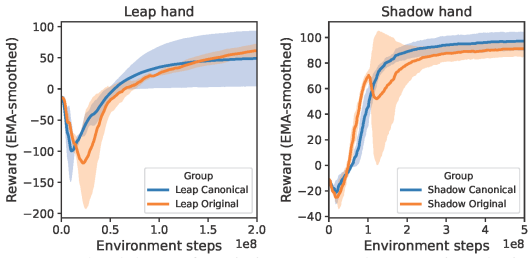


Fig. 7: Smoothed log of training reward over simulation steps during training. The solid lines mark the mean value, and the shaded area consists of the standard deviation.

TABLE II: Comparison of in-hand reorientation policies trained with the canonical URDF v.s. the original URDF.

Policy	Steps-to-Fall ↑	Cumulative Rotation ↑
Shadow (Original)	369.66	9.09
Shadow (Canonical)	390.62	10.92
LEAP (Original)	397.62	5.63
LEAP (Canonical)	326.98	6.31

B. Canonical Hand Fidelity

1) In-Hand Reorientation: We evaluate the physical fidelity of the canonical hand representation by comparing its performance to that of the original hand models in high-dynamic in-hand manipulation tasks, assessing whether essential kinematic and control properties are preserved.

Metrics. (1) *Steps-to-Fall*: the number of simulation steps the object remains stably grasped; and (2) *Cumulative Rotation*: the total rotation about the positive z-axis in radians, indicating how effectively the policy drives rotation. These metrics capture both stability and rotation, allowing direct comparison between canonical and original hands.

Results. Table II reports the average performance, showing that canonical representations achieve comparable Steps-to-Fall and Cumulative Rotation. Fig. 7 further indicates that the learning progress and convergence patterns are similar for both representations. These results demonstrate that the canonical parameterization preserves essential manipulation dynamics and can serve as a practical interface for downstream applications such as reinforcement learning.

2) Grasp Policy Transfer via Action Mapping: Using the bidirectional mappings introduced in Sec. III-E, we replay grasp policies between the canonical and original URDF spaces. The policy trained in canonical space (Ours, Sec. V-C) is mapped to the original URDF, while the policy trained on the original URDF ($\mathcal{D}(\mathcal{R}, \mathcal{O})$) is mapped to the canonical space. As shown in Table III, transfers in both directions achieve closely matched success rates, indicating that the canonical representation preserves the action semantics required for stable execution. The main discrepancy occurs with

TABLE III: Comparison of grasp success rates when transferring across canonical and original URDFs.

Method	Success Rate (%)		
	Allegro	Barrett	ShadowHand
Ours (Canonical)	84.20	88.10	62.90
Ours (Original)	71.60 (-12.60)	88.70 (+0.60)	62.60 (-0.30)
D(R,O) (Original)	92.30	87.30	83.00
D(R,O) (Canonical)	92.38 (+0.08)	87.34 (+0.04)	78.63 (-4.37)

TABLE IV: Grasp performance comparison.

Method	Success Rate (%) ↑			Time (sec.) ↓
	Allegro	Barrett	ShadowHand	
DFC	76.2	86.3	58.8	>1800
GenDexGrasp	51.0	67.0	54.2	19.71
D(R,O) Grasp	**92.3**	87.3	**83.0**	0.65
Ours	84.2	**88.1**	62.9	**0.13**

the Allegro Hand, due to the omission of its axial-rotation joint in the canonical URDF, creating a minor structural mismatch. Overall, these results demonstrate that the mapping is robust and that the canonical action space provides a coherent interface for executing policies across diverse hand designs.

C. Cross-Embodiment Grasping Performance

We build upon the filtered GenDexGrasp [13] dataset provided by $\mathcal{D}(\mathcal{R}, \mathcal{O})$ Grasp [29], which contains 24,764 valid grasps from three dexterous hands: Allegro, Barrett, and Shadow Hand. These hands vary substantially in geometry and kinematics, spanning 8–22 DoFs and three-, four-, and five-finger topologies. All grasps are converted into our canonical URDF, providing a consistent action space across embodiments and enabling evaluation of its generality and applicability across different hands.

We then train a single unified grasp-generation model on the canonical URDF introduced in Sec. IV-C, evaluating its performance on 10 unseen objects against state-of-the-art baselines and comparing unified training with embodiment-specific training. The evaluation metric is provided in Appendix E2.

Comparison with State-of-the-Art Methods. As shown in Table IV, our lightweight model achieves grasp success rates comparable to more complex pipelines, without requiring optimization-based refinement. Inference uses a 10-step DDIM sampler [24] and runs in only 0.13 s, making it the most efficient among the evaluated methods.

This comparison is intended to evaluate the canonical URDF as a downstream action space rather than to introduce a new grasping algorithm. The results show that even a simple model trained in this representation produces high-quality grasps across diverse hands, demonstrating that the canonical parameterization is expressive, coherent, and supports robust cross-hand grasp generation without reliance on hand-specific architectures or heavy engineering.

Unified Training v.s. Embodiment-Specific Training. The unified model consistently outperforms embodiment-specific models (Table V), indicating that the canonical URDF enables effective policy sharing across morphologies. By learning in a shared action space, hands with distinct kinematics benefit from each other's data, highlighting the representation's capacity to support cross-embodiment generalization.

TABLE V: Comparison of grasp success rates. "Specific" indicates that each embodiment is trained independently, while "Unified" denotes joint training across all embodiments.

Method	Success Rate (%)		
	Allegro	Barrett	ShadowHand
Specific	82.1	87.6	55.4
Unified	**84.2**	**88.1**	**62.9**

D. Zero-Shot Grasping Performance

We evaluate the zero-shot generalization of the canonical grasping policy using a set of 10 objects. Following the procedure described in Sec. IV-D, we generate and filter grasps for various LEAP Hand variants, limiting each hand–object pair to a maximum of 200 grasps and applying selection criteria on the hand variants. The resulting dataset contains 168,747 grasps. For zero-shot evaluation, the training data excludes grasps corresponding to the validated LEAP Hand variants.

We first train a full model on the complete dataset to assess cross-embodiment performance on seen hands. We then train three additional models, each excluding data from a specific LEAP Hand variant (`leap_3033`, `leap_3303`, and `leap_3330`) for zero-shot testing. As shown in Table VII, the models conditioned on hand morphology achieve performance on unseen hands comparable to that on seen hands, demonstrating strong zero-shot transfer capability across unobserved hand designs.

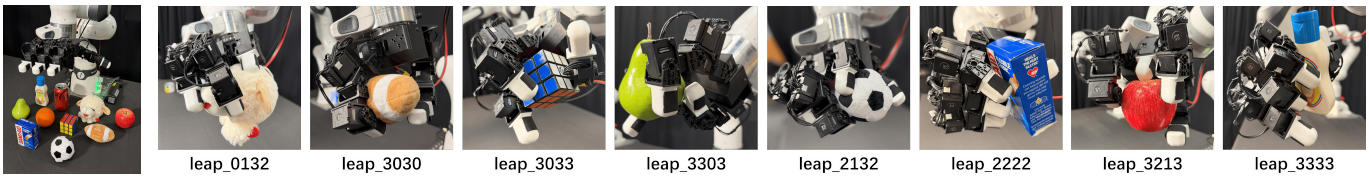
We further evaluate zero-shot generalization on a set of LEAP Hand variants with $x + y + z + w < 8$, which differ substantially from those in the training dataset, making the task more challenging. To isolate the effect of hand morphology on success rates, we generated and filtered grasp data specifically for these variants, training models only on their corresponding data. The resulting grasp data numbers and success rates are summarized in Table VIII.

As shown, the zero-shot models outperform the variant-specific models on `leap_0312`, `leap_2203`, and `leap_3103`. Performance is slightly lower on `leap_0303` and `leap_3030`, likely because these two variants have only two fingers, resembling a gripper, and their grasp patterns differ markedly from those of dexterous hands in the training set. In addition, these variants generated a large number of grasps (≈30k) in a single round, which benefits the variant-specific models. Nevertheless, the zero-shot models still demonstrate strong generalization to these challenging morphologies, highlighting the effectiveness of the canonical hand conditioning and unified action space.

To further validate the impact of hand conditioning, we evaluated the model using incorrect hand conditions on other LEAP Hand variants. As shown in Table IX, we tested the variant `leap_3033`. Overall, applying an incorrect hand condition substantially reduces grasp success rates. Using the original LEAP Hand parameters (`leap_3333`) in the "All

TABLE VI: Real-World grasp success rates.

Model	Success Rate										
	Apple	Band Aid	Coke	Cube	Football	Mayo	Orange	Pear	Sheep	Soccer	Average
`leap_3333` (trained)	8/10	7/10	9/10	7/10	10/10	6/10	8/10	9/10	10/10	9/10	83/100
`leap_3033` (trained)	8/10	8/10	2/10	6/10	9/10	6/10	7/10	9/10	10/10	10/10	75/100
`leap_3033` (zero-shot)	8/10	10/10	5/10	5/10	7/10	2/10	9/10	7/10	9/10	9/10	71/100
`leap_3303` (trained)	7/10	8/10	5/10	3/10	9/10	4/10	9/10	7/10	9/10	9/10	70/100
`leap_3303` (zero-shot)	9/10	6/10	4/10	5/10	9/10	5/10	8/10	6/10	9/10	10/10	71/100



(a) Object list (b) Zero-shot (left four) and trained (right four) grasping results with different LEAP Hand variants.

Fig. 8: Real-world grasping objects and results.

TABLE VII: Comparison of grasp success rates. Underlined values indicate zero-shot evaluation.

Model	Success Rate (%)		
	`leap_3033`	`leap_3303`	`leap_3330`
All Data	76.1	**85.4**	43.3
No `leap_3033` Data	<u>67.8</u>	83.4	31.5
No `leap_3303` Data	**81.5**	<u>81.9</u>	**46.9**
No `leap_3330` Data	74.7	81.6	<u>36.3</u>

TABLE VIII: Grasp success rates and dataset number for selected LEAP Hand variants.

Model	Success Rate (%)				
	0303	0312	2203	3030	3103
All Data	46.9	**13.3**	**65.1**	36.2	**46.6**
Specific Data	**75.1**	12.1	33.9	**55.4**	18.5
Data Num	37249	4368	2458	37217	2124

TABLE IX: Grasp results for `leap_3033` across hand conditions.

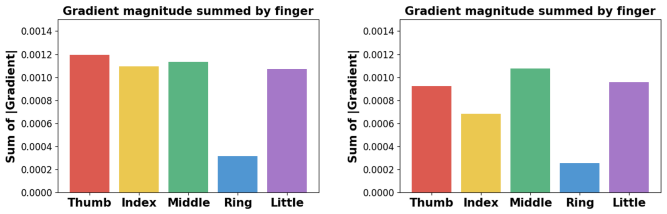
Condition	Success Rate (%)	
	All Data	Zero-Shot
`leap_3303`	**85.4**	**81.6**
`leap_3033`	33.9	12.8
`leap_3330`	20.5	2.4
`leap_3333`	85.1	71.9

Data" trained model results in only a minor drop, as the model can partially overfit to the included variant. In contrast, in the zero-shot setting, success rates drop significantly, highlighting the critical role of hand conditioning.

We also visualize the backward gradients with respect to the canonical parameters (Fig. 9). The gradient for the ring finger remains consistently low, as this finger is absent in the LEAP Hand. In variant `leap_3033`, where the index finger is removed, its gradient drops markedly, indicating that the model has learned to focus on the functional fingers essential for successful grasps.

E. Real-World Experiment

To validate the sim-to-real transfer and practical applicability of our approach, we deploy the LEAP Hand [23] grasping policies on a Franka Research 3 robotic arm. Object observations are captured using an Intel RealSense L515 depth camera. The evaluation is conducted on a set of 10 diverse objects, as illustrated in Fig. 8a, and across several LEAP Hand variants (Fig. 8b). Details of real-world experiments are in the Appendix.



(a) `leap_3333` (b) `leap_3033`

Fig. 9: Gradient magnitude visualization.

We evaluate both models trained on the canonical dataset and zero-shot models that have never seen the target hand variants. As summarized in Table VI, the trained models achieve high grasp success rates, demonstrating that the canonical hand representation preserves the essential dynamics and physical fidelity of the original hands, and that sim-to-real transfer is reliable. Importantly, the zero-shot models achieve success rates close to those of the trained models, highlighting strong generalization capabilities and the effectiveness of the hand condition in guiding grasping across unseen morphologies. The multi-panel illustrations in Fig. 8b further demonstrate that even for diverse and unusual LEAP Hand variants, the policy executes stable and robust grasps, highlighting the strength and effectiveness of the canonical hand conditioning.

VI. Conclusion and Discussion

We introduced a canonical representation that maps heterogeneous dexterous hands into a shared parameter and action space, enabling scalable cross-embodiment learning. Its continuous morphology parameterization enables explicit hand-aware conditioning, and the unified action space facilitates data and policy sharing across platforms. Together, these properties enable embodiment-agnostic manipulation policies that generalize in a zero-shot manner to unseen hands, establishing the canonical URDF as a practical foundation for scalable cross-embodiment manipulation. Beyond dexterous hand grasping, our insights can extend to broader robotic embodiments like humanoid robots and various manipulation tasks, potentially benefiting the embodied AI and robotics communities.

APPENDIX

A. Canonical URDF Parameter Definition

1) Base Canonical Parameterization: As introduced in Sec. III-C, we adopt a canonical parameterization of dexterous hands that abstracts a wide range of robotic hand designs into a compact and interpretable representation. Derived from a canonical URDF design, this parameterization captures the most salient morphological and kinematic variations across embodiments while removing redundant or simulator-specific details present in the original URDF files. In total, the canonical representation comprises 82 parameters, providing a unified description that can be directly used by learning-based models and supports cross-embodiment transfer.

Morphological Parameters. The first component of the parameter set describes the global geometry of the hand. The palm is modeled as a cylindrical primitive, and each finger link is modeled as a capsule. Hand morphology is parameterized by `palm_radius`, `finger_radius`, and `finger_lengths`.

To reduce dimensionality while preserving realistic proportions, we adopt two mild structural assumptions. First, all fingers share the same diameter, represented by a single `finger_radius` parameter. Second, palm thickness is assumed to scale proportionally with the finger radius and therefore does not require an independent parameter. These assumptions are consistent with common robotic hand designs and introduce minimal loss of generality.

Finger link lengths are encoded using a small set of shared parameters. Specifically, three parameters represent the flexion chain shared by all non-thumb fingers, reflecting the empirical regularity that the index, middle, ring, and little fingers exhibit highly correlated phalangeal proportions in both human anatomy and robotic implementations. The thumb, whose morphology differs more substantially across designs, is encoded using three additional length parameters. As a result, `finger_lengths` forms a six-dimensional vector in total.

This formulation preserves essential anthropomorphic structure while keeping the morphological parameter space compact and well-suited for learning.

Kinematic Parameters. The second component of the canonical representation defines the kinematic structure of the hand, including joint placement, joint axes, and joint limits.

Finger base locations on the palm are encoded by translational offsets `finger_xyz`. Based on the observation that non-thumb fingers are typically mounted perpendicular to the palm and approximately coplanar on the palm's yz-plane, we model only their translational origins and omit rotational offsets. This simplification substantially reduces the parameter count while remaining faithful to common mechanical layouts.

The thumb requires additional flexibility due to its diverse mounting configurations and more complex kinematic role. We therefore include an explicit orientation parameter `thumb_rpy`, which specifies the rotation of the thumb base relative to the palm frame. Following the kinematic analysis in Sec. III-B, only the proximal two thumb joints exhibit meaningful variability in joint axes across different hands. These are represented using `thumb_axes`, while all remaining joint axes are fixed to canonical directions.

Finally, joint feasibility is specified by `joint_lowers` and `joint_uppers`, which define the allowable motion ranges for each of the 22 canonical degrees of freedom. Together, these parameters capture the essential motion characteristics of dexterous hands while abstracting away unnecessary implementation details.

TABLE X: Canonical URDF parameter definition.

Param Name	Num	Param Meaning
palm_radius	1	radius of palm cylinder
finger_radius	1	radius of finger capsule
finger_lengths	6	thumb link lengths and finger link lengths
finger_xyz	15	knuckle origin translations of 5 fingers
little_extra_origin	6	joint origin (xyz+rpy) of little finger extra origin
thumb_rpy	3	knuckle origin rotation of thumb
thumb_axes	6	thumb axes of proximal two joints
joint_lowers	22	lower ranges of all joints
joint_uppers	22	upper ranges of all joints
Total	**82**	

TABLE XI: Extended canonical URDF parameter definition.

Param Name	Num	Param Meaning
palm_radius	1	radius of palm cylinder
finger_radii	5	radii of 5 fingers
finger_lengths	15	link lengths of 5 fingers
joint_origins	72	origins of 12 joints
joint_axes	36	axes of 12 joints
joint_lowers	22	lower ranges of all joints
joint_uppers	22	upper ranges of all joints
Total	**173**	

Modeling Assumptions. For clarity, we summarize the structural assumptions underlying the canonical parameterization. Joint indices i increase from the finger base toward the fingertip:

1) All fingers share the same capsule diameter; palm thickness is implicitly tied to this diameter.
2) All non-thumb fingers use identical link lengths.
3) All non-thumb fingers lie on the palm-aligned yz-plane.
4) For all fingers, joint1 and joint2 share the same origin.
5) For the thumb, joint3 and joint4 share the same origin.
6) Except for the thumb's joint1, all remaining joints have fixed local-frame orientations with $rpy = (0, 0, 0)$.
7) All joint axes are fixed except for the thumb's joint1 and joint2. Specifically, the thumb's joint3 and all non-thumb joint1 use the $+x$ axis, while all remaining joints use the $+y$ axis.

These assumptions capture the dominant structural regularities of dexterous hands while enabling a compact, standardized, and learning-friendly representation.

2) Extended Canonical Parameterization: While the base canonical parameterization is sufficiently expressive to model the vast majority of dexterous hand designs, certain embodiments exhibit structural deviations that cannot be captured exactly under the canonical assumptions. For example, in the Allegro Hand, the rotation axis of the non-thumb joint1 is aligned with the $+z$ direction rather than the canonical choice. In the LEAP Hand, the proximal two joints of each non-thumb finger are swapped in the kinematic tree, with the flexion joint located closer to the palm than the abduction/adduction joint. Although such designs can be approximated through reasonable canonical mappings, these cases may introduce small geometric or kinematic discrepancies.

To address these limitations, we additionally provide an extended parameterization that relaxes many of the canonical assumptions. As summarized in Table XI, this representation contains 173 parameters and can exactly encode all observed hand designs with substantially reduced approximation error. This extended formulation demonstrates the extensibility of our framework and allows the parameter set to be expanded when higher fidelity is required or when modeling unconventional embodiments.

Compared with the base canonical design, the extended parameterization increases both geometric and kinematic expressiveness. On the morphology side, it replaces the shared `finger_radius` and `finger_lengths` with per-finger specifications, `finger_radii` and expanded `finger_lengths`, enabling each finger to have its own radius and link-length configuration. This allows the representation to capture finer geometric variation and to accommodate future dexterous hand designs with more diverse proportions.

On the kinematic side, the extended representation introduces additional joint origins and rotation axes for twelve joints. These include the first three joints of the thumb and the little finger, as well as the first two joints of the remaining three fingers. This expansion enables a broader range of mounting configurations and kinematic couplings to be represented explicitly. Under the extended parameter set, the only remaining assumptions are that the two distal joints of each finger share fixed local-frame orientations ($rpy = (0, 0, 0)$) and use the flexion-aligned $+y$ axis. These minimal constraints preserve a consistent convention for distal articulation while maximizing compatibility with structurally diverse dexterous hands.

B. URDF Parsing and Generation Details

This section describes the automatic framework used for URDF parsing and generation, which enables bidirectional conversion between the original and the canonical robot URDFs. We detail both the parameter extraction procedure and the canonical URDF generation process.

1) Canonical Parameter Extraction: To obtain canonical parameters from an original robotic hand design, we parse the corresponding URDF and extract the geometric and kinematic information required by our representation. The parser requires only two minimal inputs: (i) a mapping between original URDF joints and their canonical counterparts, and (ii) the canonical palm root transform in the world frame. All remaining quantities are inferred automatically. Below, we summarize the extraction procedure for each parameter group.

Palm Geometry. The palm link is identified as the unique link that serves as the parent of multiple revolute joints. Its bounding box is computed from the associated mesh geometry, and the palm radius is estimated from the average in-plane dimensions of this bounding box. This provides a stable approximation of palm thickness and overall scale.

Finger Geometry. Finger radii are estimated by examining all links belonging to finger kinematic chains. For each link, we compute the minimum dimension of its mesh bounding box, and these values are averaged across all fingers to obtain a consistent capsule radius. Finger link lengths are derived from joint-to-joint distances along each kinematic chain. The translational components of joint origins specify the first two segment lengths, while the third link length is approximated by averaging adjacent segments when an explicit fingertip frame is not available.

Finger Base Positions. Finger base locations are computed by transforming the child link frame of each finger's base joint into the canonical palm coordinate system. The palm's pose is defined by `palm_origin` metadata, enabling consistent conversion from world coordinates to the palm frame.

Thumb Base Orientation. To determine `thumb_rpy`, we use the frames of the thumb's first and last existing joints. The vector from the thumb base to the thumb tip defines the local $+z$ direction, while the joint axis at the base provides the local $+y$ direction. Together, these constraints define a right-handed thumb coordinate frame, which is expressed in the palm frame and converted to Euler angles.

Thumb Joint Axes. The rotation axes of the first two thumb joints are read directly from the URDF joint specifications. To express them in the canonical thumb frame, we compute the relative rotation between the original URDF joint frames and the reordered thumb base frame.

Extra Little-Finger Orientation. For hand designs that include an additional abduction joint at the base of the little finger, we extract its joint frame from the URDF and reorder its axes using the same convention as the thumb (palm-up direction as $+x$, rotation axis as $+y$). This yields the `little_extra_origin` parameters in the extended representation.

Joint Limits. Joint lower and upper bounds are read directly from the URDF `limit` tags for each revolute joint. For joints that are absent due to structural differences across designs, zero-range limits are used as placeholders to maintain a consistent parameter dimensionality.

Overall, this procedure maps diverse URDF descriptions into the unified canonical parameter set by combining mesh-based geometry estimation, joint-origin analysis, coordinate frame reorientation, and direct extraction of URDF-specified joint axes and limits. In practice, the resulting parameters typically require only a brief manual inspection and, when necessary, minor adjustments to ensure consistency with the canonical conventions.

2) Canonical URDF Generation: Reconstruction of a full URDF from canonical parameters is implemented using a template-based generation module built with the Jinja2 dynamic templating language [17]. The URDF is defined as a parameterized text template whose placeholders are populated with canonical parameter values.

Conditional logic within the template allows dynamic inclusion or omission of elements. For example, a joint is instantiated only if its lower and upper limits differ, and optional fingers or links are generated only when corresponding parameters are present. This design enables automatic generation of valid and consistent URDFs for hands with varying numbers of fingers, links, and joint configurations, while strictly adhering to the canonical conventions.

Together, the parsing and generation components provide a consistent, bidirectional conversion pipeline between diverse robotic hand models and the unified canonical representation, supporting both analysis of existing designs and synthesis of new hand embodiments.

C. Morphology Latent Learning

1) Data Sampling: We first sample a global joint configuration for each synthetic hand, i.e., which fingers and which joints are present. Given this discrete topology, continuous geometric and frame parameters are sampled from uniform distributions over physically plausible ranges to preserve realistic hand proportions. Joint axes are drawn from six canonical directions $(\pm x, \pm y, \pm z)$ and encoded as one-hot vectors. Joint ranges are not supplied as continuous values; instead, each of the 22 canonical joints is represented by a binary indicator denoting its presence or absence.

Each sampled hand is serialized into a fixed-length vector by concatenating continuous geometry and frame parameters, joint-axis one-hot encodings, and joint-activation indicators. This unified representation provides the VAE with both continuous morphological variation and discrete structural information while keeping the sampling process simple and scalable.

2) Model Architecture: We use a standard variational autoencoder with MLP-based encoder and decoder networks. The encoder maps the input vector through three fully connected layers with hidden dimensions $[512, 256, 128]$, followed by BatchNorm and ReLU activations, and outputs the mean and log-variance of a 16-dimensional latent distribution. The decoder mirrors this architecture and reconstructs the hand parameters using multiple output heads: a continuous head for geometric and frame parameters, categorical heads for joint-axis prediction, and a sigmoid-activated head for joint-activation indicators.

3) Training Objective: Reconstruction losses are defined in a type-specific manner to reflect the heterogeneous nature of the parameters:

$$\begin{aligned} \mathcal{L}_{\text{cont}} &= \|\hat{q}_{\text{cont}} - q_{\text{cont}}\|_2^2 , \\ \mathcal{L}_{\text{axis}} &= \text{CrossEntropy}(\hat{q}_{\text{axis}}, q_{\text{axis}}), \\ \mathcal{L}_{\text{joint}} &= \text{BinaryCrossEntropy}(\sigma(\hat{q}_{\text{joint}}), q_{\text{joint}}) , \end{aligned} \tag{2}$$

where q_{cont} denotes continuous morphology parameters, q_{axis} the six-way joint-axis encodings, and q_{joint} the binary joint-activation indicators. The overall loss combines these terms with a KL-divergence regularizer:

$$\mathcal{L} = \mathcal{L}_{\text{cont}} + \mathcal{L}_{\text{axis}} + \mathcal{L}_{\text{joint}} + \beta\, \mathcal{L}_{\text{KL}}, \tag{3}$$

where $\beta = 0.01$ in all experiments.

Training is performed using the Adam optimizer with a learning rate of $1\text{e}{-4}$, $(\beta_1, \beta_2) = (0.95, 0.999)$, and a weight decay of $1\text{e}{-6}$.

TABLE XII: PPO training hyper-parameters.

Hyper-parameter	Value
Discount factor γ	0.99
GAE λ	0.95
Horizon length T	32
Sequence length (RNN)	4
Minibatch size	32768
PPO epochs per update	5
Learning rate	5×10^{-3}
PPO Clip range ϵ	0.2
KL threshold	0.02
Entropy coefficient	0.0
Critic loss coefficient	4
Max gradient norm	1.0
Reward scale	0.01
Bounds loss coefficient	1×10^{-4}

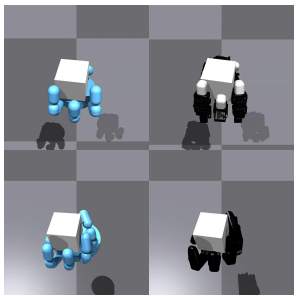


Fig. 10: Visualization of in-hand reorientation under the original and canonical URDFs. Top: LEAP Hand; bottom: Shadow Hand.

TABLE XIII: Observation states used for policy training.

Observation	Description
$\tilde{\mathbf{q}} \in \mathbb{R}^{n_{\text{dof}}}$	Normalized hand joint positions
$\mathbf{q}^{\text{tar}} \in \mathbb{R}^{n_{\text{dof}}}$	Normalized target joint (action)
$\mathbf{p}_{\text{cube}} \in \mathbb{R}^3$	Cube position
$\mathbf{r}_{\text{cube}} \in \mathbb{R}^3$	Cube orientation Euler angles.
$\dot{\mathbf{q}} \in \mathbb{R}^{n_{\text{dof}}}$	Hand joint velocities.
$\mathbf{v}_{\text{cube}} \in \mathbb{R}^3$	Cube linear velocity.
$\boldsymbol{\omega}_{\text{cube}} \in \mathbb{R}^3$	Cube angular velocity.
$\boldsymbol{\phi} \in \mathbb{R}^2$	Phase variables (periodic task encoding).

D. In-hand Reorientation

1) RL Network Architecture: The reinforcement learning policy is implemented using a multi-layer perceptron (MLP) with hidden layer dimensions of [512, 256, 128]. To capture temporal dependencies across consecutive observations, a single gated recurrent unit (GRU) layer with a hidden size of 256 is applied before the MLP backbone. The GRU-processed observation is then concatenated with the original observation and passed as input to the MLP. The ELU activation function is used throughout the network.

2) PPO Optimization Details: We train our in-hand rotation policies using Proximal Policy Optimization (PPO) [21]. The reinforcement learning hyperparameters are summarized in Table XII, and the observation inputs for the agents are listed in Table XIII. n_{dof} is the degree of freedom of the robot hand, and the parameters are the same for both the original robot hand and the canonical robot hands. We train the LEAP

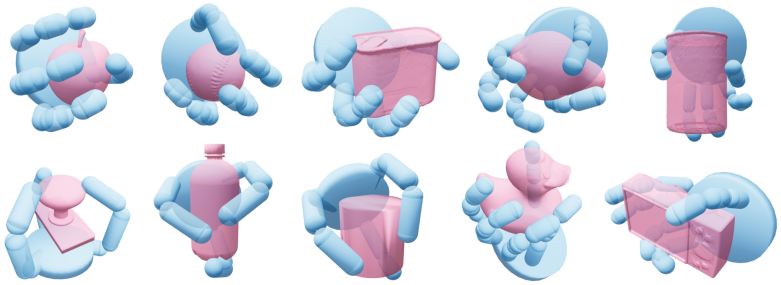


Fig. 11: Grasp visualizations of the canonical URDF for the Allegro, Barrett, and Shadow Hand. All grasps are generated using the same cross-embodiment policy.

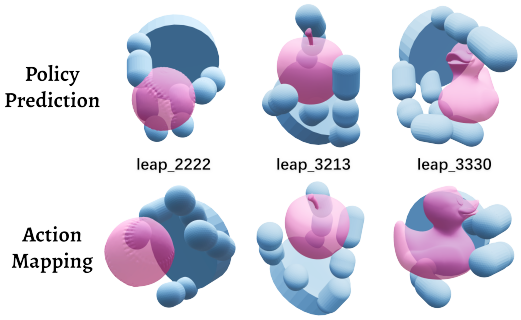


Fig. 12: Grasp visualization comparison between direct inference on a target hand and mapping actions from `leap_3333`.

Hand policies for 400 gradient iterations (approximately 200M environment steps) and the Shadow Hand policy for 1,000 policy update iterations (approximately 500M environment steps). The reward function consists of the following reward and penalty components:

1) $r_{\text{rot}} = \text{clip}(\omega_z, \omega_{\min}, \omega_{\max})$, where ω_z is the z-axis angular velocity of the cube, and the value is clipped between $\omega_{\min}$ and $\omega_{\max}$.
2) $p_{\text{pose}} = \|q - q^0\|_2^2$, where $q \in \mathbb{R}^{n_{\text{dof}}}$ is the current joint positions, and $q^0 \in \mathbb{R}^{n_{\text{dof}}}$is the initial joint positions at the beginning of the episode.
3) $p_\tau = \|\tau\|_2^2$, where $\tau \in \mathbb{R}^{n_{\text{dof}}}$ is the applied torques of the robot joints.
4) $p_{\text{work}} = (\tau^\top \dot{q})^2$ penalizes excessive force applied on the cube by the fingers.
5) $p_{\text{v}} = \|\mathbf{v}\|_1$, where $\mathbf{v} \in \mathbb{R}^3$ is the linear velocity of the cube.
6) The cube fell penalty is:

$$r_{\text{fallen}} = \begin{cases} 1, & \text{if } z_{\text{cube}} < z_{\text{threshold}}, \\ 0, & \text{otherwise.} \end{cases} \tag{4}$$

where z_{cube} is the z coordinate of the cube, and $z_{\text{threshold}}$ is height threshold to keep the cube from falling.

The final reward is the aggregation of the individual reward terms, weighted by each term's scaling terms:

$$r_{\text{base}} = s_{\text{rot}} r_{\text{rot}} - s_{\text{v}}\, p_{\text{v}} - s_{\text{pose}}\, p_{\text{pose}} - s_\tau\, p_\tau - s_{\text{work}}\, p_{\text{work}}.$$

3) Generate initial grasp configuration: We construct our in-hand rotation task following previous works on in-hand rotation [23, 18]. At the start of each episode, the hand is initialized in a stable grasp pose of a cube. To increase the diversity of initial configurations, we first generate a canonical stable grasp for each robot hand (e.g., LEAP Hand, Shadow Hand). We then introduce random perturbations to the joint angles to create varied but feasible grasp poses. For the LEAP Hand, joint angles are perturbed by uniformly sampled noise from $(-0.25, 0.25)$ radians, while for the ShadowHand, we apply uniform noise from $(-0.1, 0.1)$ radians. The perturbed configuration is then simulated for 50 steps without control inputs. During the rollout, small random forces are applied to the cube to perturb it further. At the end of each rollout, we validate the grasp using the following criteria: the fingertip–cube distance is below a threshold, at least two fingers are in contact with the cube, and the cube height exceeds a threshold relative to the palm center.

We generate 10,000 valid grasp configurations per robot hand, which serve as initial states. During training, environments randomly sample from these precomputed grasp poses at reset. For evaluation, we also randomly sample from the same set to ensure consistency across experiments.

E. Cross-Embodiment Dexterous Grasping

1) Hyperparameters: We use an MLP diffusion model with two hidden layers of sizes 512 and 256, and a diffusion-step embedding dimension of 64. The diffusion process is trained with 1000 timesteps using the "sample" prediction formulation. Optimization is performed with Adam using an initial learning rate of $1\text{e}{-3}$ and a cosine annealing schedule that decays the learning rate to $1\text{e}{-7}$ over the full training horizon.

2) Evaluation Metric: We follow the evaluation procedure of $\mathcal{D}(\mathcal{R}, \mathcal{O})$ Grasp [29], adapted to our canonical URDF controller. Grasp success is assessed using a force-closure–based criterion in Isaac Gym. For each predicted grasp, we execute the controller on the canonical hand and then sequentially apply external forces along the six orthogonal directions for 1 second each. A grasp is deemed successful if the resulting object displacement remains below 2 cm after all perturbations.

F. LEAP Hand Zero-Shot Generalization

1) Extended Canonical URDF for LEAP Hand: To reduce the geometric discrepancy between the canonical URDF and the original LEAP Hand URDF under different configurations, we adopt an extended canonical URDF design in this experiment. In particular, we use the extended parameter set described in Appendix A2 and adjust the placement of the abduction/adduction joints for non-thumb fingers by relocating them to the second link joint. This modification more faithfully reflects the distinctive kinematic structure of the LEAP Hand while remaining fully compatible with our canonical parameterization framework. When conditioning the policy, we directly use the original LEAP Hand morphology parameters as the hand condition.

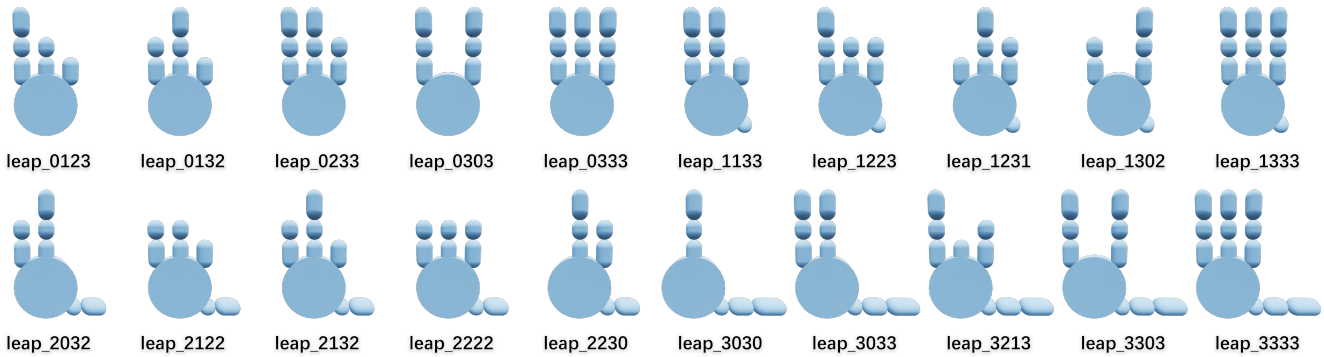


Fig. 13: Visualization of canonical LEAP Hand variants.

Hand	# Grasps	Hand	# Grasps	Hand	# Grasps	Hand	# Grasps	Hand	# Grasps	Hand	# Grasps	Hand	# Grasps	Hand	# Grasps
leap_0000	0	leap_0001	0	leap_0002	0	leap_0003	0	leap_0010	0	leap_0011	0	leap_0012	0	leap_0013	0
leap_0020	0	leap_0021	2	leap_0022	39	leap_0023	404	leap_0030	0	leap_0031	1191	leap_0032	284	leap_0033	20464
leap_0100	0	leap_0101	0	leap_0102	11	leap_0103	249	leap_0110	0	leap_0111	0	leap_0112	3	leap_0113	10
leap_0120	0	leap_0121	0	leap_0122	44	leap_0123	65	leap_0130	454	leap_0131	467	leap_0132	18	leap_0133	2702
leap_0200	0	leap_0201	215	leap_0202	1896	leap_0203	7382	leap_0210	0	leap_0211	0	leap_0212	347	leap_0213	705
leap_0220	166	leap_0221	43	leap_0222	1792	leap_0223	3898	leap_0230	10278	leap_0231	982	leap_0232	979	leap_0233	11995
leap_0300	0	leap_0301	3705	leap_0302	20457	leap_0303	37249	leap_0310	606	leap_0311	884	leap_0312	4368	leap_0313	2140
leap_0320	5914	leap_0321	1567	leap_0322	10221	leap_0323	25894	leap_0330	54709	leap_0331	4020	leap_0332	5264	leap_0333	34966
leap_1000	0	leap_1001	0	leap_1002	0	leap_1003	1	leap_1010	0	leap_1011	0	leap_1012	0	leap_1013	0
leap_1020	0	leap_1021	0	leap_1022	0	leap_1023	84	leap_1030	1	leap_1031	485	leap_1032	5551	leap_1033	1030
leap_1100	0	leap_1101	0	leap_1102	0	leap_1103	30	leap_1110	0	leap_1111	0	leap_1112	3	leap_1113	4
leap_1120	0	leap_1121	40	leap_1122	0	leap_1123	41	leap_1130	13	leap_1131	17	leap_1132	227	leap_1133	500
leap_1200	0	leap_1201	13	leap_1202	293	leap_1203	314	leap_1210	0	leap_1211	14	leap_1212	29	leap_1213	94
leap_1220	0	leap_1221	14	leap_1222	126	leap_1223	125	leap_1230	231	leap_1231	707	leap_1232	1465	leap_1233	967
leap_1300	0	leap_1301	113	leap_1302	447	leap_1303	497	leap_1310	5	leap_1311	11	leap_1312	312	leap_1313	146
leap_1320	66	leap_1321	111	leap_1322	1688	leap_1323	1502	leap_1330	228	leap_1331	403	leap_1332	1846	leap_1333	6506
leap_2000	0	leap_2001	165	leap_2002	36	leap_2003	1285	leap_2010	5	leap_2011	3	leap_2012	22	leap_2013	241
leap_2020	1711	leap_2021	6	leap_2022	175	leap_2023	475	leap_2030	5131	leap_2031	80	leap_2032	1023	leap_2033	3234
leap_2100	0	leap_2101	187	leap_2102	48	leap_2103	80	leap_2110	3	leap_2111	0	leap_2112	8	leap_2113	5
leap_2120	231	leap_2121	84	leap_2122	34	leap_2123	18	leap_2130	43	leap_2131	66	leap_2132	110	leap_2133	112
leap_2200	995	leap_2201	133	leap_2202	632	leap_2203	2458	leap_2210	144	leap_2211	37	leap_2212	85	leap_2213	476
leap_2220	1018	leap_2221	109	leap_2222	206	leap_2223	718	leap_2230	2001	leap_2231	43	leap_2232	418	leap_2233	1472
leap_2300	3272	leap_2301	674	leap_2302	1971	leap_2303	4735	leap_2310	743	leap_2311	154	leap_2312	343	leap_2313	776
leap_2320	2772	leap_2321	118	leap_2322	593	leap_2323	1323	leap_2330	3217	leap_2331	110	leap_2332	925	leap_2333	1776
leap_3000	0	leap_3001	5142	leap_3002	606	leap_3003	6091	leap_3010	8017	leap_3011	1004	leap_3012	235	leap_3013	1475
leap_3020	4071	leap_3021	6700	leap_3022	318	leap_3023	8631	leap_3030	37217	leap_3031	9516	leap_3032	445	leap_3033	29140
leap_3100	2646	leap_3101	987	leap_3102	234	leap_3103	2124	leap_3110	1035	leap_3111	787	leap_3112	77	leap_3113	641
leap_3120	1412	leap_3121	1238	leap_3122	83	leap_3123	1528	leap_3130	3950	leap_3131	854	leap_3132	83	leap_3133	2918
leap_3200	2674	leap_3201	4457	leap_3202	2219	leap_3203	2838	leap_3210	12478	leap_3211	546	leap_3212	514	leap_3213	615
leap_3220	1645	leap_3221	3041	leap_3222	758	leap_3223	1764	leap_3230	2902	leap_3231	4046	leap_3232	1223	leap_3233	4281
leap_3300	28972	leap_3301	11097	leap_3302	10051	leap_3303	31407	leap_3310	19666	leap_3311	1268	leap_3312	3049	leap_3313	3481
leap_3320	9589	leap_3321	3540	leap_3322	2965	leap_3323	12469	leap_3330	34795	leap_3331	5090	leap_3332	2691	leap_3333	13309

TABLE XIV: Number of valid grasps generated for different LEAP Hand variants after filtering.

While it is possible to further expand the parameter set to more precisely capture all aspects of the LEAP Hand design, doing so would introduce hand-specific parameters that are not shared by most dexterous hands. Since the abduction/adduction joint placement is the primary structural deviation of the LEAP Hand from the canonical design, we do not incorporate this variation into the general canonical URDF. Instead, we apply this extension only in this experiment to minimize the sim-to-real gap and ensure a fair evaluation of zero-shot generalization.

Importantly, this adjustment does not alter the underlying learning framework and illustrates the extensibility of our paradigm: hand-specific kinematic features can be incorporated through targeted extensions without compromising the generality of the canonical representation.

2) LEAP Hand Variant Generation: Thanks to the semantic structure of the canonical parameters, different LEAP Hand morphologies can be generated programmatically by a simple scripting procedure that sets the corresponding link and joint parameters to zero. Using the extended canonical URDF template, we batch-generate LEAP Hand variants with different numbers of links for each finger. Visualizations of representative variants used in our experiments are shown in Fig. 13, using Viser [35].

3) Grasp Data Generation: Grasp data for each LEAP Hand variant is generated using Lightning Grasp [36], a recent analytical grasp synthesis method. For each hand morphology, we specify only the fingertip links and active joints in the configuration file, enabling efficient batch generation across all variants. We generate grasp candidates over four independent rounds to improve coverage.

All generated grasps are subsequently filtered in Isaac Gym using the same physical validity criteria as in the main experiments. Statistics of the resulting grasp datasets, including the number of valid grasps per hand variant, are summarized in Table XIV.

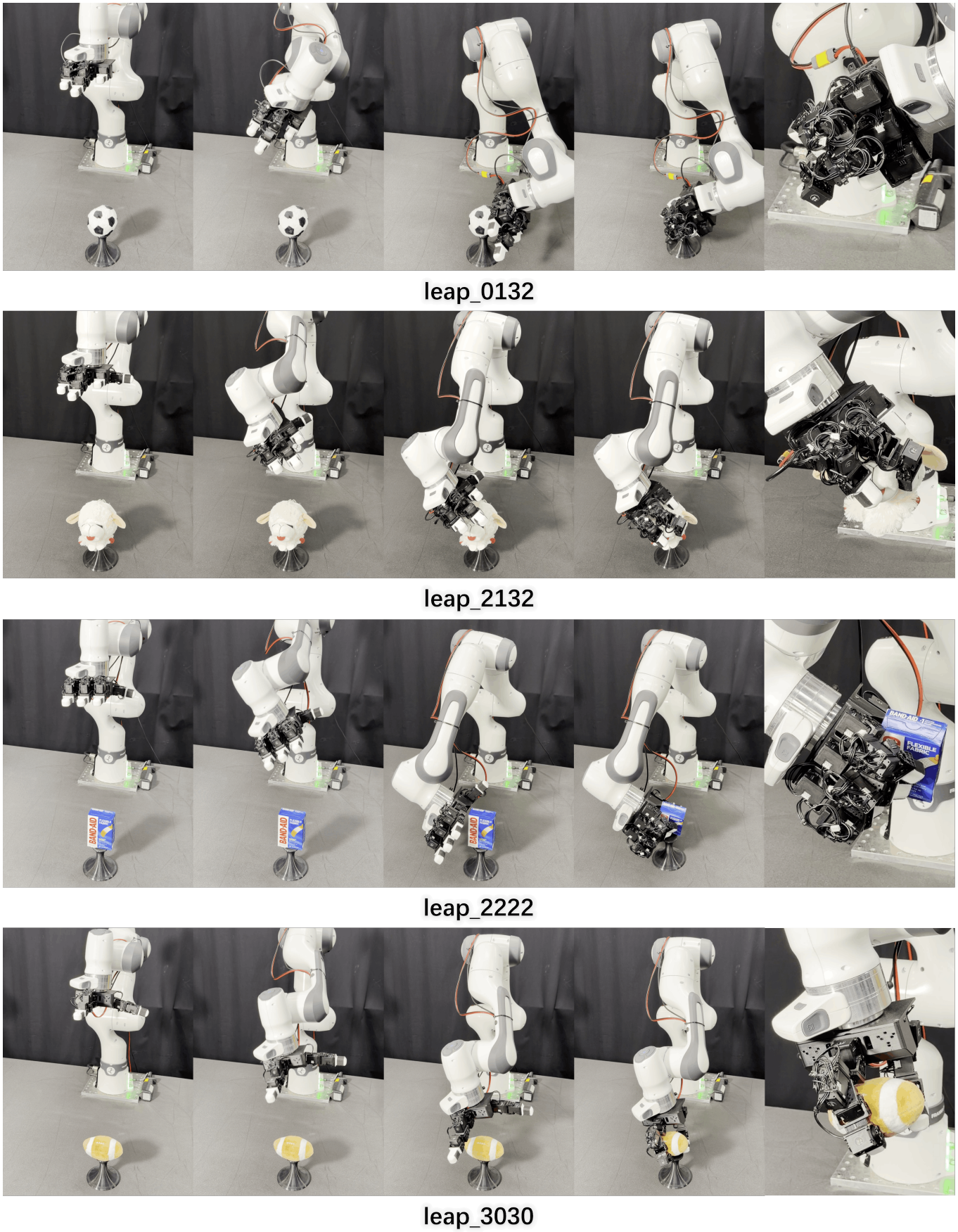


Fig. 14: Visualization of real-world experiment (I).

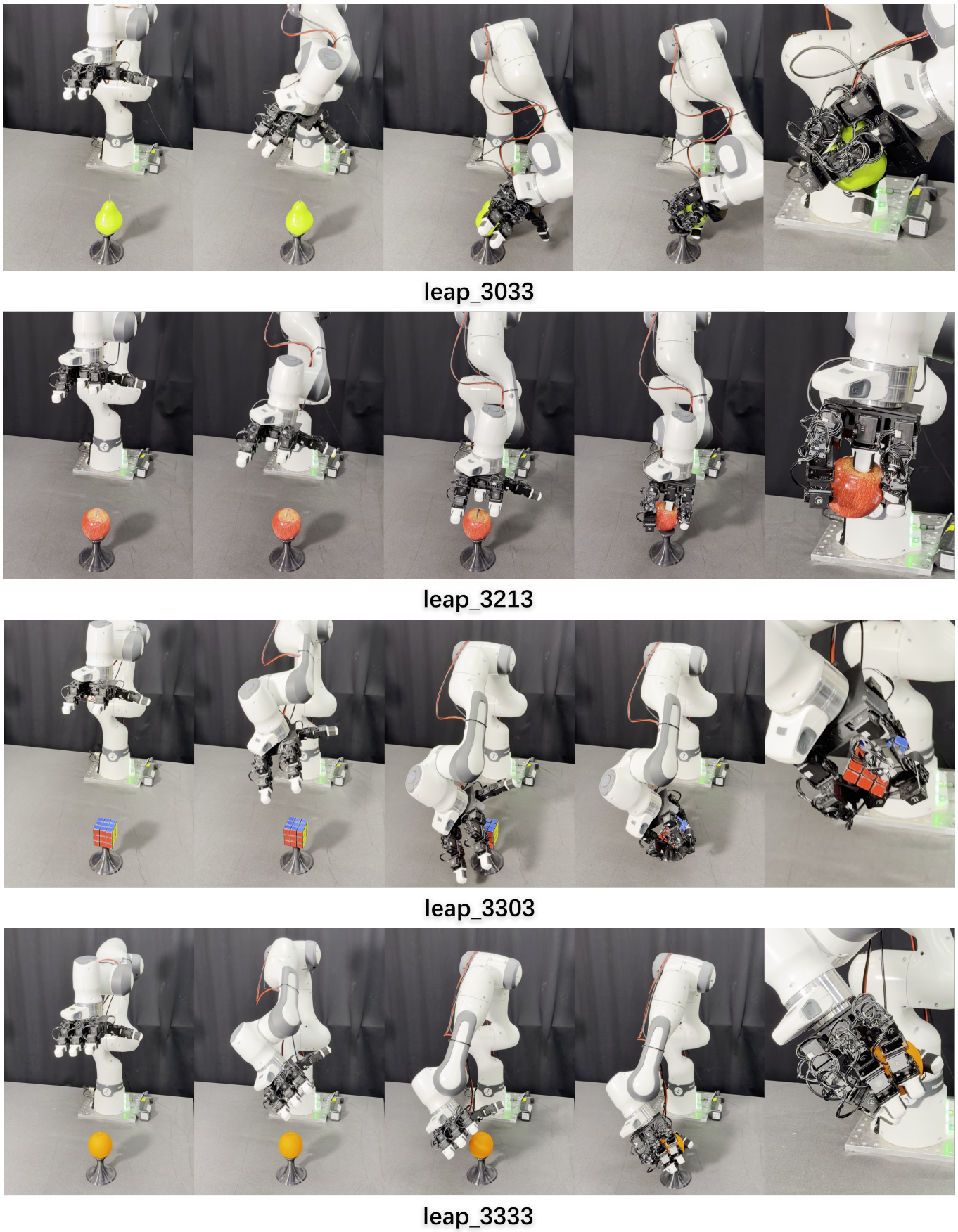


Fig. 15: Visualization of real-world experiment (II).

References

[1] Ananye Agarwal, Shagun Uppal, Kenneth Shaw, and Deepak Pathak. Dexterous functional grasping. *arXiv preprint arXiv:2312.02975*, 2023.

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

[3] Erik Bauer, Elvis Nava, and Robert K Katzschmann. Latent action diffusion for cross-embodiment manipulation. *arXiv preprint arXiv:2506.14608*, 2025.

[4] A. Bicchi. Hands for dexterous manipulation and robust grasping: a difficult road toward simplicity. *IEEE Transactions on Robotics and Automation*, 16(6):652–662, 2000. doi: 10.1109/70.897777.

[5] Tao Chen, Jie Xu, and Pulkit Agrawal. A system for general in-hand object re-orientation. In *Conference on Robot Learning*, pages 297–307. PMLR, 2022.

[6] 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. doi: 10.1126/scirobotics.adc9244. URL https://www.science.org/doi/abs/10.1126/scirobotics.adc9244.

[7] M.R. Cutkosky. On grasp choice, grasp models, and the design of hands for manufacturing tasks. *IEEE Transactions on Robotics and Automation*, 5(3):269–279, 1989. doi: 10.1109/70.34763.

[8] Xin Fei, Zhixuan Xu, Huaicong Fang, Tianrui Zhang, and Lin Shao. T(r, o) grasp: Efficient graph diffusion of robot-object spatial transformation for cross-embodiment dexterous grasping. *arXiv preprint arXiv:2510.12724*, 2025.

[9] Jingxiang Guo, Jiayu Luo, Zhenyu Wei, Yiwen Hou, Zhixuan Xu, Xiaoyi Lin, Chongkai Gao, and Lin Shao. Telepreview: A user-friendly teleoperation system with virtual arm assistance for enhanced effectiveness. *arXiv preprint arXiv:2412.13548*, 2024.

[10] Zihao He, Bo Ai, Tongzhou Mu, Yulin Liu, Weikang Wan, Jiawei Fu, Yilun Du, Henrik I Christensen, and Hao Su. Scaling cross-embodiment world models for dexterous manipulation. *arXiv preprint arXiv:2511.01177*, 2025.

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

[12] Kun Lei, Huanyu Li, Dongjie Yu, Zhenyu Wei, Lingxiao Guo, Zhennan Jiang, Ziyu Wang, Shiyu Liang, and Huazhe Xu. Rl-100: Performant robotic manipulation with real-world reinforcement learning. *arXiv preprint arXiv:2510.14830*, 2025.

[13] Puhao Li, Tengyu Liu, Yuyang Li, Yixin Zhu, Yaodong Yang, and Siyuan Huang. Gendexgrasp: Generalizable dexterous grasping. *arXiv preprint arXiv:2210.00722*, 2022.

[14] Zhixuan Liang, Yao Mu, Yixiao Wang, Tianxing Chen, Wenqi Shao, Wei Zhan, Masayoshi Tomizuka, Ping Luo, and Mingyu Ding. Dexhanddiff: Interaction-aware diffusion planning for adaptive dexterous manipulation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 1745–1755, 2025.

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

[16] Austin Patel and Shuran Song. Get-zero: Graph embodiment transformer for zero-shot embodiment generalization. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 14262–14269. IEEE, 2025.

[17] Pallets Projects. Jinja documentation. https://jinja.palletsprojects.com/en/stable/, 2022.

[18] Haozhi Qi, Ashish Kumar, Roberto Calandra, Yi Ma, and Jitendra Malik. In-hand object rotation via rapid motor adaptation. In *Conference on Robot Learning*, pages 1722–1732. PMLR, 2023.

[19] 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. *arXiv preprint arXiv:1709.10087*, 2017.

[20] Shadow Robot. Dexterous hand series. https://shadowrobot.com/dexterous-hand-series/.

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

[22] Lin Shao, Fabio Ferreira, Mikael Jorda, Varun Nambiar, Jianlan Luo, Eugen Solowjow, Juan Aparicio Ojea, Oussama Khatib, and Jeannette Bohg. Unigrasp: Learning a unified model to grasp with multifingered robotic hands. *IEEE Robotics and Automation Letters*, 5(2):2286–2293, 2020.

[23] Kenneth Shaw, Ananye Agarwal, and Deepak Pathak. Leap hand: Low-cost, efficient, and anthropomorphic hand for robot learning. *arXiv preprint arXiv:2309.06440*, 2023.

[24] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

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

[26] Jun Wang, Yuzhe Qin, Kaiming Kuang, Yigit Korkmaz,

Akhilan Gurumoorthy, Hao Su, and Xiaolong Wang. Cyberdemo: Augmenting simulated human demonstration for real-world dexterous manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17952–17963, 2024.

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

[28] Ruicheng Wang, Jialiang Zhang, Jiayi Chen, Yinzhen Xu, Puhao Li, Tengyu Liu, and He Wang. Dexgraspnet: A large-scale robotic dexterous grasp dataset for general objects based on simulation. *arXiv preprint arXiv:2210.02697*, 2022.

[29] Zhenyu Wei, Zhixuan Xu, Jingxiang Guo, Yiwen Hou, Chongkai Gao, Zhehao Cai, Jiayu Luo, and Lin Shao. $\mathcal{D}(\mathcal{R}, \mathcal{O})$ grasp: A unified representation of robot and object interaction for cross-embodiment dexterous grasping. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4982–4988, 2025. doi: 10.1109/ICRA55743.2025.11127754.

[30] Zhiyuan Wu, Rolandos Alexandros Potamias, Xuyang Zhang, Zhongqun Zhang, Jiankang Deng, and Shan Luo. Cedex: Cross-embodiment dexterous grasp generation at scale from human-like contact representations. *arXiv preprint arXiv:2509.24661*, 2025.

[31] Lixin Xu, Zixuan Liu, Zhewei Gui, Jingxiang Guo, Zeyu Jiang, Zhixuan Xu, Chongkai Gao, and Lin Shao. Dexsingrasp: Learning a unified policy for dexterous object singulation and grasping in cluttered environments. *arXiv preprint arXiv:2504.04516*, 2025.

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

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

[34] Zhixuan Xu, Chongkai Gao, Zixuan Liu, Gang Yang, Chenrui Tie, Haozhuo Zheng, Haoyu Zhou, Weikun Peng, Debang Wang, Tianrun Hu, et al. Manifoundation model for general-purpose robotic manipulation of contact synthesis with arbitrary objects and robots. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 10905–10912. IEEE, 2024.

[35] Brent Yi, Chung Min Kim, Justin Kerr, Gina Wu, Rebecca Feng, Anthony Zhang, Jonas Kulhanek, Hongsuk Choi, Yi Ma, Matthew Tancik, et al. Viser: Imperative, web-based 3d visualization in python. *arXiv preprint arXiv:2507.22885*, 2025.

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

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

[38] Ying Yuan, Haichuan Che, Yuzhe Qin, Binghao Huang, Zhao-Heng Yin, Kang-Won Lee, Yi Wu, Soo-Chul Lim, and Xiaolong Wang. Robot synesthesia: In-hand manipulation with visuotactile sensing. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6558–6565. IEEE, 2024.

[39] Zhecheng Yuan, Tianming Wei, Shuiqi Cheng, Gu Zhang, Yuanpei Chen, and Huazhe Xu. Learning to manipulate anywhere: A visual generalizable framework for reinforcement learning. *arXiv preprint arXiv:2407.15815*, 2024.

[40] Yanjie Ze, Gu Zhang, Kangning Zhang, Chenyuan Hu, Muhan Wang, and Huazhe Xu. 3d diffusion policy: Generalizable visuomotor policy learning via simple 3d representations. *arXiv preprint arXiv:2403.03954*, 2024.

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

[42] Yuanhang Zhang, Tianhai Liang, Zhenyang Chen, Yanjie Ze, and Huazhe Xu. Catch it! learning to catch in flight with mobile dexterous hands. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 14385–14391. IEEE, 2025.