Robotics: Science and Systems 2026
Sydney, Australia, July 13–17, 2026

DexGrasp-Zero: A Morphology-Aligned Policy for Zero-Shot Cross-Embodiment Dexterous Grasping

Yuliang Wu [1], **Yanhan Lin** [1], **WengKit Lao** [1], **Yuhao Lin** [1], **Yi-Lin Wei** [1],
Wei-Shi Zheng [1,2,3], **Ancong Wu** [1†]

[1] School of Computer Science and Engineering, Sun Yat-sen University, China
[2] Key Laboratory of Machine Intelligence and Advanced Computing, Ministry of Education, China
[3] Shenzhen Loop Area Institute

[†] Corresponding author

https://yliangwu.github.io/DexGrasp-Zero

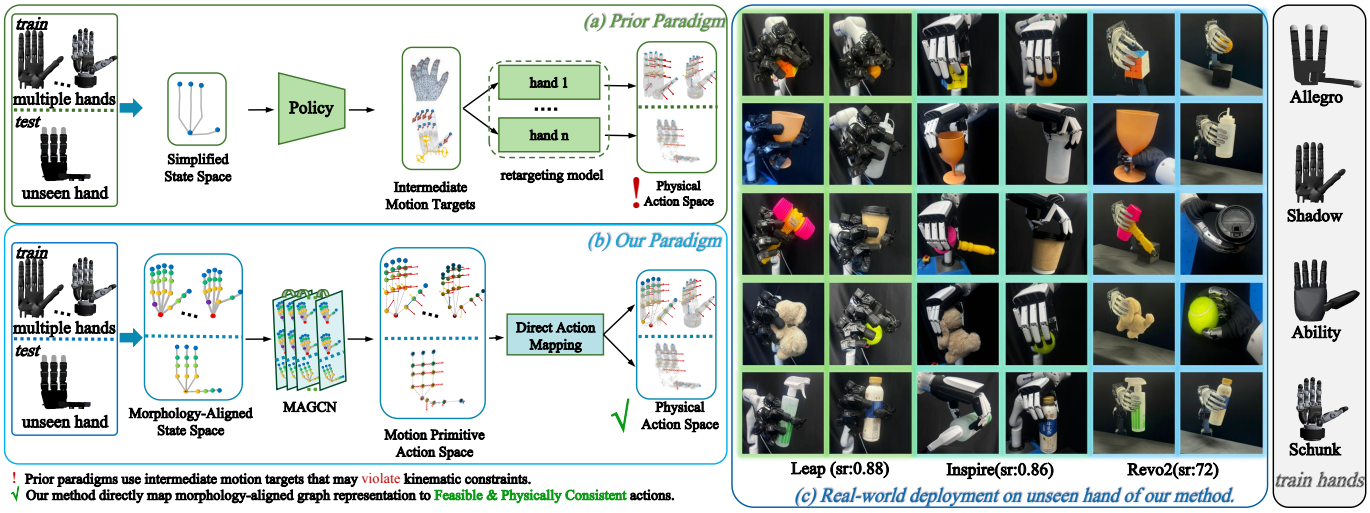

Fig. 1: **Paradigm comparison: prior approaches versus our method.** (a) Prior paradigm: Existing methods [35, 22] train on a simplified and lossy unified state space. They output intermediate motion targets that require hand-specific retargeting models to convert into physical joint commands. This adds complexity and can lead to kinematically infeasible actions. (b) Our paradigm: We learn a single universal policy end-to-end. The policy operates on a lossless morphology-aligned graph representation and outputs actions in a hand-agnostic motion-primitive space. Physical commands are generated directly through a fixed hand-specific mapping $\mathcal{M}_h$, removing the need for trainable retargeting modules. (c) Real-world deployment on unseen hands validate the effectiveness and zero-shot transfer capability of our approach.

***Abstract*—To meet the demands of increasingly diverse dexterous hand hardware, it is crucial to develop a policy that enables zero-shot cross-embodiment grasping without redundant re-learning. Cross-embodiment alignment is challenging due to heterogeneous hand kinematics and physical constraints. Existing approaches typically predict intermediate motion targets and retarget them to each embodiment, which may introduce errors and violate embodiment-specific limits, hindering transfer across diverse hands. To overcome these limitations, we propose *DexGrasp-Zero*, a policy that learns universal grasping skills from diverse embodiments, enabling zero-shot transfer to unseen hands. We first introduce a morphology-aligned graph representation that maps each hand's kinematic keypoints to anatomically grounded nodes and equips each node with tri-axial orthogonal motion primitives, enabling structural and semantic alignment across different morphologies. Relying on this graph-based representation, we design a *Morphology-Aligned Graph Convolutional Network* (MAGCN) to encode the graph for policy learning. MAGCN incorporates a *Physical Property Injection* mechanism that fuses hand-specific physical constraints into the graph features, enabling adaptive compensation for varying link lengths and actuation limits for precise and stable grasping. Our extensive simulation evaluations on the YCB dataset demonstrate that our policy, jointly trained on four heterogeneous hands (Allegro, Shadow, Schunk, Ability), achieves an 85% zero-shot success rate on unseen hardware (LEAP, Inspire), outperforming the state-of-the-art method by 59.5%. Real-world experiments further evaluate our policy on three robot platforms (LEAP, Inspire, Revo2), achieving an 82% average success rate on unseen objects.**

I. Introduction

In the past several years, industry and academia have rapidly advanced dexterous hand hardware [41, 31], and more morphologically heterogeneous hands are expected to emerge for dexterous manipulation. However, existing reinforcement learning (RL) policies are typically limited to the training morphology and fail to generalize across hand types [23, 36, 24, 38, 4, 7, 26, 12] , necessitating costly retraining and data collection for every new hand. Therefore, for practical application, we need a transferable policy that enables zero-shot cross-embodiment grasping without redundant relearning.

The fundamental obstacle in cross-embodiment learning is the morphological differences across hands, which make it difficult to align both the *input* and *output* spaces of a grasping policy. The observations and control commands of different dexterous hands are morphology-dependent: observations reflect the number and arrangement of joints and sensors, while control commands are shaped by degrees of freedom and physical properties such as joint limits and link geometry. This poses significant challenges for mapping the desired action targets across different hands.

Existing cross-embodiment dexterous grasping researches largely follows two routes: (1) open-loop grasp pose generation [30, 6, 37, 28, 25, 27], which lacks closed-loop feedback and is less robust to perturbations; and (2) RL-based policy transfer of intermediate action representations [35, 22] with retargeting. However, as illustrated in Fig. 1, the intermediate motion targets may violate kinematic or actuation constraints of the target hand, limiting zero-shot generalization.

One key observation is that, despite large embodiment variations, joints across different hands share underlying morphological and kinematic regularities. Moreover, the inter-joint relationships can be naturally represented as a graph structure. Motivated by this, we construct a cross-hand transferable policy based on graph neural network in a RL framework that directly outputs joint-level actions. This avoids explicit retargeting and thus overcoming the inherent limitations of previous methods.

To achieve this, we introduce **DexGrasp-Zero**, a universal grasping policy capable of zero-shot transfer to unseen hands. Specifically, we represent each hand as an anatomically grounded, morphology-aligned *graph* that matches the hand's kinematic structure, and define a hand-agnostic motion primitive space grounded in biomechanics [18] to align control semantics across morphologies. We then parameterize the policy with the **Morphology-Aligned Graph Convolutional Network** (MAGCN), which encodes the graph state for policy learning via graph convolution. To further ensure stable grasping, MAGCN incorporates URDF-derived embodiment physical properties and injects them into the graph representations, enabling the policy to adaptively compensate for varying link lengths and actuation limits.

We evaluate DexGrasp-Zero on six diverse dexterous hands. Our experiments show that a single policy, jointly trained on four hands, achieves a **85% zero-shot grasping success rate** on unseen hands (Leap, Inspire), outperforming prior methods by **59.5%**. We further evaluate our policy on three heterogeneous physical robots (Leap, Inspire, Revo2), achieving an average success rate of **82%** on 10 unseen objects.

Our contributions are summarized as follows:

1) We propose a morphology-aligned graph state representation and a hand-agnostic motion primitive space that align perception and control semantics across heterogeneous dexterous hands.
2) We design MAGCN, a GCN-based policy that injects URDF-derived physical properties into the learned graph features to better respect embodiment constraints.
3) We conduct extensive experiments in simulation and on three real-robot hands (Leap, Inspire, Revo2), validating the efficacy of our representation and framework for zero-shot cross-embodiment grasping on novel objects.

II. Related work

A. Learning-Based Dexterous Grasping

Dexterous grasping is a cornerstone of robotic manipulation. Prior work employs either generative models [9, 33, 13, 39, 32, 40] to sample diverse grasps or reinforcement learning (RL) [23, 24, 17, 36, 38, 4] for closed-loop policy optimization. However, both paradigms typically assume a fixed hand morphology: generative approaches are limited by the hand coverage in training data, and RL policies are often constrained by hand-specific state representations and network architectures, so neither generalizes across hand embodiments without retraining.

B. Cross-Embodied Dexterous Grasping

To enable cross-hand transfer under morphological heterogeneity, recent efforts often introduce intermediate representations that are less sensitive to hand-specific degrees of freedom and physical properties. Existing methods largely follow two routes: generating static grasping postures, or learning dynamic closed-loop grasping policies. Existing grasp synthesis methods can be broadly categorized as object-centric [20, 1, 11, 34, 5] or obj-hand interaction [30, 6], most of which generate static, kinematically feasible postures and struggle with real-time perturbations.

Another approach is to train closed-loop grasping strategies that execute complete grasping trajectories. A common design is to use a simplified intermediate action target (e.g., fingertip displacements [22] or MANO poses [35]) and then retarget it to each hand. The subsequent hand-specific retargeting step can produce infeasible joint commands under kinematic constraints. In contrast, our approach enables end-to-end policy learning that maps observations directly to physical joint commands, thereby avoiding kinematic infeasibility.

C. Graph Representation for Cross-Embodied Grasping

Graph neural networks provide a natural abstraction for dexterous hands, but prior graph-based approaches can suffer from misalignment in cross-hand representations, resulting in a

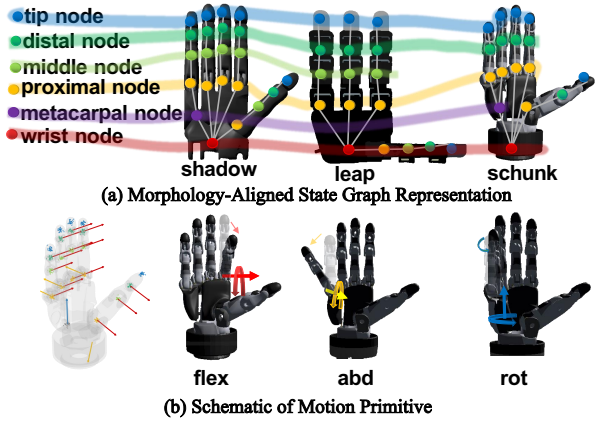

Fig. 2: Universal hand representation. (a) Morphology-Aligned State Graph Representation: nodes correspond to anatomical units, edges follow kinematic chains, yielding a hand-agnostic semantic graph structure. (b) Schematic of three motion primitives (Flexion, Abduction, Axial Rotation) on a Schunk hand, showing their physical motion effects at representative joints.

lossy unified interface. GeoMatch++ [29] uses kinematic links as graph nodes, which lacks anatomical semantics and does not explicitly encode topological connectivity. CrossDex [35] and She et al. [22] model only sparse keypoints without encoding topological relationships, leading to information loss. GET-Zero [16] builds a full kinematic graph but treats each physical joint as a node, splitting a single anatomical unit with multiple dofs into multiple nodes, which hinders semantic alignment across hands with different DoF distributions, and its evaluations are therefore limited to LEAP Hand variants.

In contrast, our method adopts an anatomically aligned and topology-aware graph representation that preserves task-relevant structure across heterogeneous hands. Together with hand-agnostic motion primitives, it aligns both state and action semantics across embodiments, enabling more direct and transferable policy learning.

III. DexGrasp-Zero

We propose **DexGrasp-Zero**, a morphology-aligned graph policy that transfers *one* grasping controller across diverse dexterous hands by explicitly extracting *shared* semantics from *hand-specific* mechanics. As shown in Fig. 3, we represent each hand with a morphology-aligned state graph, encode embodiment-specific physical priors from URDF, and fuse them via a morphology-aligned GCN to output a hand-agnostic motion-primitive space that is deterministically mapped to executable joint commands.

A. Problem Formulation for Zero-Shot Cross-Embodiment Grasping

We consider a unified cross-embodiment learning setting. Let $\mathcal{H}_{\text{train}}$ denote the set of training hands (embodiments) and $\mathcal{H}_{\text{test}}$ denote a disjoint set of unseen test hands. For each hand $h \in \mathcal{H}_{\text{train}} \cup \mathcal{H}_{\text{test}}$, we define a hand-conditioned MDP in which the agent observes a hand-specific state $\mathbf{s}_t^h$ and outputs an action at every time step t. Our goal is to learn a *single* shared policy π_θ on $\mathcal{H}_{\text{train}}$ and evaluate it *zero-shot* on $\mathcal{H}_{\text{test}}$ without any finetuning.

Crucially, π_θ outputs an intermediate action representation in a hand-agnostic action space, while a fixed, hand-specific mapping $\mathcal{M}_h$ deterministically converts it into an executable physical command for embodiment h. The cross-embodiment control loop is:

$$\boldsymbol{\alpha}_t^h \sim \pi_\theta\big(\cdot \mid \mathbf{s}_t^h\big), \qquad \boldsymbol{\alpha}_{\text{physical},t}^h = \mathcal{M}_h\big(\boldsymbol{\alpha}_t^h\big), \tag{1}$$

where $\boldsymbol{\alpha}_t^h$ denotes the hand-agnostic intermediate action and $\boldsymbol{\alpha}_{\text{physical},t}^h$ is the physical command executed on hand h. The learning objective is to maximize the expected discounted return across training hands:

$$\max_\theta \ \mathbb{E}_{h\sim\mathcal{H}_{\text{train}}}\left[\sum_{t=0}^{T} \gamma^t \, r\big(\mathbf{s}_t^h,\, \boldsymbol{\alpha}_{\text{physical},t}^h\big)\right], \tag{2}$$

with the requirement that the resulting policy generalizes zero-shot to unseen hands $h' \in \mathcal{H}_{\text{test}}$ at test time.

In the remainder of this section, we instantiate $\mathbf{s}_t^h$ and $\boldsymbol{\alpha}_t^h$ using a morphology-aligned graph representation and a universal motion-primitive space, and specify how $\mathcal{M}_h$ is constructed from the hand kinematics.

B. Morphology-Aligned State and Action Graph Representation

Prior cross-embodiment grasping systems often struggle with representation mismatch at both the state and action levels: state encodings are often not semantically aligned across morphologies, and actions are frequently output in a task-space 3D targets and require retargeting, which can introduce execution error and yield physically infeasible targets. These issues motivate us to design an morphology-aligned, information-preserving state space and a physically constrained action space that can be executed directly on each hand, avoiding retargeting.

1) Morphology-Aligned State Representation: Dexterous hands exhibit substantial variation in kinematic topology and DoF counts, hindering direct cross-embodiment policy transfer. Our key insight is that, despite these differences, hand kinematics share a set of anatomically meaningful functional units [15]. We therefore abstract any dexterous hand h into a morphology-aligned state graph $\mathcal{G}_h = (\mathcal{V}_h, \mathcal{E}_h)$, where $|\mathcal{V}_h| = N_h$ may vary across embodiments and each node falls into one of six semantic types: fingertip, distal, middle, proximal, metacarpal, or wrist. Edges $\mathcal{E}_h$ encode the kinematic relationships between these units, as shown in Fig. 2.

For each node v_i, we define a feature vector $\mathbf{x}_i^h \in \mathbb{R}^{d_{\text{node}}}$ capturing its dynamic state. Stacking all node features yields the **node-feature matrix**

$$\mathbf{X}_{\text{node}}^h = [\mathbf{x}_1^h, \ldots, \mathbf{x}_{N_h}^h]^\top \in \mathbb{R}^{N_h \times d_{\text{node}}}. \tag{3}$$

The hand's kinematic structure is encoded in an adjacency matrix $\mathbf{A}^h \in \{0,1\}^{N_h \times N_h}$, where $\mathbf{A}_{ij}^h = 1$ if node i is the parent of node j in the kinematic chain.

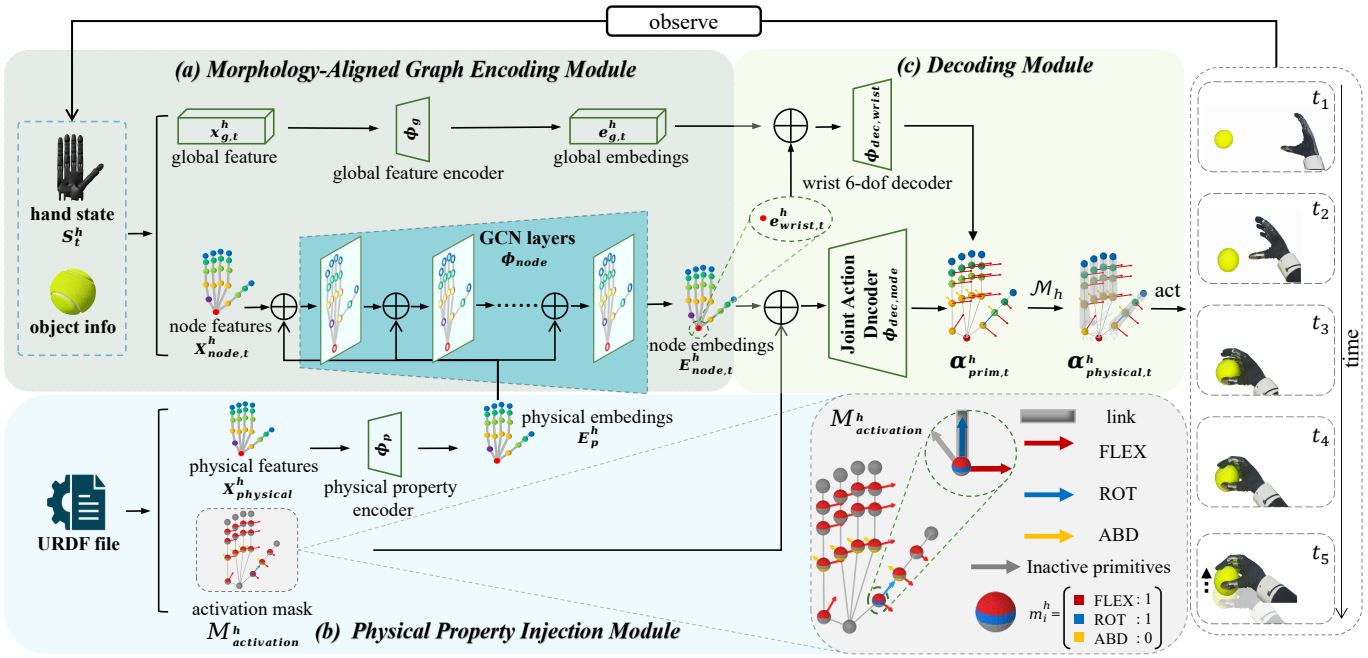

Fig. 3: **Architecture of DexGrasp-Zero.** At each time step t: (a) **Morphology-Aligned Graph Encoder** encodes hand-object state into node features $\mathbf{X}^h_{\text{node},t}$ and global feature $\mathbf{x}^h_{g,t}$ using a hand-specific graph (adjacency $\mathbf{A}^h$); a GCN with per-layer physical priors produces embeddings $\mathbf{E}^h_{\text{node},t}$ and $\mathbf{E}^h_{g,t}$. (b) **Physical Property Encoder** parses hand URDF to build a physical graph $\mathcal{G}^h_{\text{physical}}$ (joint limits, link lengths, etc.) and an activation mask $\mathbf{M}^h_{\text{activation}}$, encoded into $\mathbf{E}^h_{\text{p}}$ and fused into every GCN layer. (c) **Decoder** outputs motion primitives $\boldsymbol{\alpha}^h_{\text{prim}}$: wrist 6-DoF commands from $\mathbf{E}^h_{g,t}$ and wrist features, and joint actions from masked node embeddings; the latter are mapped via hand-specific $\mathcal{M}_h$ to executable joint commands $\alpha^h_{\text{physical},t}$.

While node features encode local articulation and contact, they lack explicit information about global hand–object relationships critical for coordinated grasping, such as wrist pose error relative to the target or object motion. To address this, we introduce a **global feature** $\mathbf{x}^h_g \in \mathbb{R}^{d_{\text{global}}}$ that provides task-level context. The full state representation is then:

$$\mathbf{s}^h = \left(\mathbf{X}^h_{\text{node}},\, \mathbf{A}^h,\, \mathbf{x}^h_g\right). \tag{4}$$

2) Hand-Agnostic Motion-Primitive Space: To achieve hand-agnostic control, we define a universal motion-primitive space $\boldsymbol{\alpha}^h_{\text{prim}}$ based on the morphology-aligned graph to align the control-semantic for each hand. It consists of two parts: (1) a 6-DoF wrist motion command, parameterized as $(\Delta\mathbf{p}^h_w, \Delta\boldsymbol{\theta}^h_w)$. Here and throughout the paper, the symbol Δ denotes an incremental change . Specifically, $\mathbf{p}_w \in \mathbb{R}^3$ represents the Cartesian position of the wrist, and $\boldsymbol{\theta}_w \in \mathbb{R}^3$ represents its orientation expressed in Euler angles. Thus, $\Delta\mathbf{p}^h_w$ and $\Delta\boldsymbol{\theta}^h_w$ are the desired translational and rotational displacements of the wrist, respectively.(2) node-level articulation commands defined over the hand-specific state graph $\mathcal{G}_h = (\mathcal{V}_h, \mathcal{E}_h)$. For each node $v_i \in \mathcal{V}_h$, we define three orthogonal motion primitives inspired by human hand biomechanics [18], as illustrated in Fig. 2b:

- **Flexion (FLEX)**: rotation that drives the child link toward the normal direction of the palm (i.e., bending inward toward the palm).
- **Abduction (ABD)**: in-plane spreading motion of the finger within the hand plane, moving the digit away from the middle finger axis;
- **Axial Rotation (ROT)**: torsional rotation of the link about its own longitudinal axis.

These primitives form a motion-primitive vector for node v_i:

$$\boldsymbol{\alpha}^h_i = [\Delta_{\text{flex}},\, \Delta_{\text{abd}},\, \Delta_{\text{rot}}]^\top \in \mathbb{R}^3. \tag{5}$$

For notational convenience, we denote the component of $\boldsymbol{\alpha}^h_i$ corresponding to primitive $p \in \{\text{FLEX}, \text{ABD}, \text{ROT}\}$ as $\alpha^h_{i,(p)}$. Stacking wrist 6-DoF commands and all nodes yields the full motion-primitive space $\boldsymbol{\alpha}^h_{\text{prim}}$ for hand h:

$$\boldsymbol{\alpha}^h_{\text{prim}} = [\Delta\mathbf{p}^{h\top}_w, \Delta\boldsymbol{\theta}^{h\top}_w, \boldsymbol{\alpha}^{h\top}_1, \ldots, \boldsymbol{\alpha}^{h\top}_{N_h}]^\top \in \mathbb{R}^{6+3N_h}. \tag{6}$$

This universal motion-primitive space achieves control-semantic alignment across heterogeneous hands.

3) Mapping to Executable Physical Commands: To execute the hand-agnostic motion-primitive space on a specific embodiment, we define a fixed, hand-specific mapping $\mathcal{M}_h$ that projects $\boldsymbol{\alpha}^h_{\text{prim}}$ onto the embodiment-specific physical joint command space. Specifically, $\mathcal{M}_h : \mathbb{R}^{6+3N_h} \to \mathbb{R}^{L_h}$ is a deterministic, sparse linear operator constructed from the hand's kinematic specification, which (1) selects valid node–primitive pairs for each physical DoF, (2) respects the ordering of joints in the robot's command interface, and (3) applies a

sign correction to align the motion-primitive direction with the physical joint axis. The physical joint displacement vector is then obtained as

$$\Delta \mathbf{q}^h = \mathcal{M}_h\big(\boldsymbol{\alpha}^h_{\text{prim}}\big). \tag{7}$$

In practice, $\mathcal{M}_h$ is implemented as an indexing rule; for the j-th joint ($j = 1, \ldots, L_h$), it satisfies

$$\Delta q^h_j = s^h_j \cdot \alpha^h_{n_j,(p_j)}, \tag{8}$$

where $n_j \in \{1, \ldots, N_h\}$ is the source node, p_j is the selected primitive, and $s_j \in \{-1, +1\}$ is the alignment sign.

We obtain $\mathcal{M}_h$ by applying unit joint excitations in simulation and matching the induced motion semantics to the corresponding motion primitives, which is easy to implement as an indexing rule (see Supplementary Sec. S1.2 and Fig. 8).

The complete physical action executed by the controller combines wrist motion and joint commands:

$$\boldsymbol{\alpha}^h_{\text{physical}} = \Big[\Delta \mathbf{p}_w^\top,\, \Delta \boldsymbol{\theta}_w^\top,\, \Delta \mathbf{q}^{h\top}\Big]^\top. \tag{9}$$

This hand-specific mapping bridges the universal motion-primitive space and the embodiment-specific command interface, enabling a single shared policy to transfer across hands by acting in a consistent action space.

C. DexGrasp-Zero Policy Design for Dexterous Grasping

Building on the above morphology-aligned state/action abstractions, we design the DexGrasp-Zero grasping policy by specifying (i) a grasping state graph (Fig. 3a), (ii) a URDF-derived physical-property graph (Fig. 3b), (iii) MAGCN with layer-wise physical fusion (Fig. 3a,c), and (iv) the reward.

1) Graph-based State Representation: To represent the state in a graph-compatible format, we construct a *state-graph topology* $\mathcal{G}_h = (\mathcal{V}_h, \mathcal{E}_h)$ for grasping observations. Our observation design follows RobustDexGrasp [38]. The time-varying observation is carried by node/global features (i.e., $\mathbf{X}^h_{\text{node}}$ and $\mathbf{x}^h_g$), while the topology is encoded by the adjacency $\mathbf{A}^h$. The state graph is represented by three components: node features, edge connectivity, and global features.

Node Features Matrix ($\mathbf{X}^h_{\text{node}}$): Given a hand h, each node $v_i \in \mathcal{V}_h$ is characterized by a feature vector $\mathbf{x}^h_i \in \mathbb{R}^{d_{\text{node}}}$ as:

$$\mathbf{x}^h_i = [\mathbf{d}^h_i,\, \boldsymbol{\theta}^h_i,\, \dot{\boldsymbol{\theta}}^h_i,\, c^h_i,\, f^h_i,\, \mathbf{m}^h_i,\, \mathbf{n}^h_i], \tag{10}$$

where $\mathbf{d}^h_i \in \mathbb{R}^3$ is the distance vector from the node to the closest point on the object surface, which is computed from the observed object point cloud by nearest-neighbor search. $\boldsymbol{\theta}^h_i,\, \dot{\boldsymbol{\theta}}^h_i \in \mathbb{R}^3$ are joint angles and velocities expressed in each the motion-primitive axis; $c^h_i \in \{0, 1\}$ indicates contact and $f^h_i \in \mathbb{R}_{\geq 0}$ is the contact force magnitude. $\mathbf{m}^h_i$, $\mathbf{n}^h_i$ are one-hot encodings of the finger semantic class (thumb, index, middle, ring, little, wrist) and node type (fingertip, distal, middle, proximal, metacarpal, wrist) respectively. Stacking all node features yields the node-feature matrix $\mathbf{X}^h_{\text{node}} = [\mathbf{x}^h_1, \ldots, \mathbf{x}^h_{N_h}]^\top \in \mathbb{R}^{N_h \times d_{\text{node}}}$.

Edge Connectivity ($\mathbf{A}^h$): We use untyped kinematic edges. The edge set $\mathcal{E}_h$ is represented by an adjacency matrix $\mathbf{A}^h \in \{0, 1\}^{N_h \times N_h}$, where $|\mathcal{V}_h| = N_h$ matches the hand graph $\mathcal{G}_h$ by construction. $\mathbf{A}^h_{ij} = 1$ indicates a kinematic connection between nodes v_i and v_j.

Global Features ($\mathbf{x}^h_g$): The global feature vector $\mathbf{x}^h_g \in \mathbb{R}^{d_{\text{global}}}$ summarizes object-centric and wrist-level signals:

$$\mathbf{x}^h_g = [\Delta \mathbf{p}^h_{\text{target}}, \mathbf{v}^h_{\text{wrist}}, \boldsymbol{\omega}^h_{\text{wrist}}, \mathbf{v}^h_{\text{obj}}, \boldsymbol{\omega}^h_{\text{obj}}], \tag{11}$$

where $\Delta \mathbf{p}_{\text{target}} \in \mathbb{R}^3$ is the displacement from the wrist frame to the object center, $(\mathbf{v}_{\text{wrist}}, \boldsymbol{\omega}_{\text{wrist}})$ are the wrist linear and angular velocities, and $(\mathbf{v}_{\text{obj}}, \boldsymbol{\omega}_{\text{obj}})$ are the object linear and angular velocities in the wrist frame.

2) Physical-Property Graph Construction: The state-graph observation $(\mathbf{X}^h_{\text{node}}, \mathbf{A}^h, x^h_g)$ captures grasping dynamics and geometry but does not encode embodiment-specific mechanical constraints. To expose these priors to the policy and improve cross-embodiment generalization, we parse the URDF files and build a physical-property graph $\mathcal{G}^h_{\text{physical}}$. Specifically, $\mathcal{G}^h_{\text{physical}} = (\mathcal{V}^h_{\text{physical}}, \mathcal{E}^h_{\text{physical}})$ is constructed to be topology-aligned with the grasping state graph: it shares the same semantic nodes and the same kinematic connectivity, i.e., $|\mathcal{V}^h_{\text{physical}}| = |\mathcal{V}_h| = N_h$ and $\mathcal{E}^h_{\text{physical}} = \mathcal{E}_h$. Each node $v_j \in \mathcal{V}^h_{\text{physical}}$ is characterized by feature vector $\mathbf{x}^{\text{physical},h}_j \in \mathbb{R}^{d_{\text{physical}}}$ that encodes static mechanical priors:

$$\mathbf{x}^{\text{physical},h}_j = [\boldsymbol{\ell}^h_j, \mathbf{a}^h_j, \mathbf{v}^h_j, \boldsymbol{\tau}^h_j, \mathbf{l}^h_j], \tag{12}$$

where $\boldsymbol{\ell}_j \in \mathbb{R}^6$ denotes normalized limits for the three motion primitive axes; $\mathbf{v}_j \in \mathbb{R}^6$ denotes the corresponding normalized velocity bounds in the same motion-primitive axis; $\mathbf{a}_j \in \mathbb{R}^9$ encodes the axis directions in 3D space for the three primitives. $\boldsymbol{\tau}_j \in \mathbb{R}^3$ contains per-axis damping coefficients and $\mathbf{l}_j \in \mathbb{R}^3$ is the link vector from the *parent node* to this node. Stacking all physical node features yields $\mathbf{X}^h_{\text{physical}}$:

$$\mathbf{X}^h_{\text{physical}} = [\mathbf{x}^{\text{physical},h}_1, \ldots, \mathbf{x}^{\text{physical},h}_{N_h}]^\top \in \mathbb{R}^{N_h \times d_{\text{physical}}}. \tag{13}$$

We encode node features $\mathbf{x}^{\text{physical},h}_j$ into a learnable physical embedding in Section III-C3.

Activation Mask $\mathbf{M}^h_{\text{activation}}$: Not all nodes support all motion-primitives due to mechanical constraints. To encode which primitives are physically realizable per node, we derive an activation mask aligned with the motion-primitive space in Section III-B2. For each node $v_i \in \mathcal{V}_h$, we define a vector

$$\mathbf{m}^h_i = [m^h_{i,\text{FLEX}},\, m^h_{i,\text{ABD}},\, m^h_{i,\text{ROT}}]^\top \in \{0, 1\}^3, \tag{14}$$

where $m^h_{i,p} = 1$ if primitive $p \in \{\text{FLEX}, \text{ABD}, \text{ROT}\}$ is physically realizable at node i for hand h, and $m^h_{i,p} = 0$ otherwise. Stacking all nodes gives

$$\mathbf{M}^h_{\text{activation}} = [\mathbf{m}^h_1, \ldots, \mathbf{m}^h_{N_h}]^\top \in \{0, 1\}^{N_h \times 3}. \tag{15}$$

Equivalently, $\mathbf{M}^h_{\text{activation}}$ can be derived from the indexing rule $\mathcal{M}_h$ by setting $m^h_{i,p} = 1$ if there exists a DoF index $j \in \{1, \ldots, L_h\}$ such that $\text{node}(j) = i$ and $\text{prim}(j) = p$. We use $\mathbf{M}^h_{\text{activation}}$ to (i) condition the decoder on which primitive axes are executable, and (ii) define an action-feasibility penalty term r_{pen} in the reward (Section III-C4) to discourage non-executable outputs on inactive primitive axes.

3) MAGCN Model Design: DexGrasp-Zero is the overall shared policy π_θ introduced in Eq. (1). In our implementation, π_θ is parameterized by the *Morphology-Aligned Graph Convolutional Network* (MAGCN) with three modules shown in Fig. 3: Morphology-Aligned Graph Encoding, Physical Property Encoding, and Decoding. To keep the notation concise, we name the modules used in these modules as functions: a node-wise physical encoder ϕ_{p}, a global encoder ϕ_{g}, a node-embedding encoder ϕ_{node}, a node-action decoder $\phi_{\text{dec,node}}$, and a wrist-action decoder $\phi_{\text{dec,wrist}}$, as shown in Fig. 3.

a) Morphology-Aligned Graph Encoding Module: We first encode the grasping observation graph (Section III-C1) using a GCN backbone to capture kinematic topology and propagate local signals along the hand graph. Concretely, the MAGCN encoder produces (i) a global embedding $\mathbf{E}_{\text{g}}^h$ from global features $\mathbf{x}_g^h$, and (ii) a node-embedding matrix $\mathbf{E}_{\text{node}}^h$ from the node-feature matrix $\mathbf{X}_{\text{node}}^h$ and adjacency $\mathbf{A}^h$:

$$\mathbf{E}_{\text{g}}^h = \phi_{\text{g}}\big(\mathbf{x}_g^h\big), \qquad \mathbf{E}_{\text{node}}^h = \phi_{\text{node}}\big(\mathbf{X}_{\text{node}}^h,\, \mathbf{A}^h\big). \tag{16}$$

b) Physical Property Encoding Module: In order to inject embodiment-specific physical information (Section III-C2) and enable the policy to adaptively compensate for varying link lengths and actuation limits thereby ensuring precise and stable grasping, we encode the physical-property graph $\mathcal{G}_{\text{physical}}^h$ with a node-wise MLP ϕ_{p} to a learnable embedding. For each node $j = 1, \ldots, N_h$,

$$\mathbf{e}_j^{\text{p},h} = \phi_{\text{p}}\big(\mathbf{x}_j^{\text{physical},h}\big) \in \mathbb{R}^{d_p}, \tag{17}$$

and stacking yields the physical embedding matrix

$$\mathbf{E}_{\text{p}}^h = \big[\mathbf{e}_1^{\text{p},h}, \ldots, \mathbf{e}_{N_h}^{\text{p},h}\big]^\top \in \mathbb{R}^{N_h \times d_p}. \tag{18}$$

c) Layer-wise Physical Injection in MAGCN: With the physical information encoded in $\mathbf{E}_{\text{p}}^h$, we can inject these embodiment-specific priors into the state-graph encoder by fusing $\mathbf{E}_{\text{p}}^h$ at every GCN layer [10]. Specifically, we instantiate ϕ_{node} as L layers of GCN with layer-wise physical injection. Equivalently, the node encoder can be written as

$$\mathbf{E}_{\text{node}}^h = \phi_{\text{node}}\big(\mathbf{X}_{\text{node}}^h,\, \mathbf{A}^h,\, \mathbf{E}_{\text{p}}^h\big), \tag{19}$$

and is computed as follows. At each layer, we concatenate the current node representation with the physical prior and perform message passing with normalized adjacency:

$$\mathbf{H}^{h,(0)} = \mathbf{X}_{\text{node}}^h, \tag{20}$$

$$\mathbf{Z}^{h,(l)} = concat(\mathbf{H}^{h,(l-1)}, \mathbf{E}_{\text{p}}^h), \tag{21}$$

$$\mathbf{H}^{h,(l)} = \sigma\big(\text{Ln}(\hat{\mathbf{A}}^h\, \mathbf{Z}^{h,(l)} \mathbf{W}^{(l)})\big), \quad l = 1, \ldots, L, \tag{22}$$

$$\mathbf{E}_{\text{node}}^h = \mathbf{H}^{h,(L)}, \tag{23}$$

where Ln denotes LayerNorm and $\hat{\mathbf{A}}^h$ is the standard GCN normalized adjacency with self-loops: $\hat{\mathbf{A}}^h = (\tilde{\mathbf{D}}^h)^{-\frac{1}{2}} \tilde{\mathbf{A}}^h (\tilde{\mathbf{D}}^h)^{-\frac{1}{2}}$, $\tilde{\mathbf{A}}^h = \mathbf{A}^h + \mathbf{I}$, and $\tilde{\mathbf{D}}_{ii}^h = \sum_j \tilde{\mathbf{A}}_{ij}^h$.

d) Decoding Module: This module jointly parameterizes the motion-primitive space (Section III-B2) using two decoders (Fig. 3c). For node-level primitives, we condition on the activation mask by concatenating the node embedding with the corresponding mask row and decoding with $\phi_{\text{dec,node}}$:

$$\tilde{\mathbf{e}}_i^{\text{node},h} = \text{concat}(\mathbf{E}_{\text{node}}^h[i], \mathbf{M}_{\text{activation}}^h[i]), \tag{24}$$

$$\boldsymbol{\alpha}_i^h = \phi_{\text{dec,node}}\big(\tilde{\mathbf{e}}_i^{\text{node},h}\big) \in \mathbb{R}^3, \quad i = 1, \ldots, N_h. \tag{25}$$

For the wrist 6-DoF command, we decode from the global embedding $\mathbf{E}_{\text{g}}^h$ and the wrist node embedding $\mathbf{e}_{\text{wrist}}^h$:

$$(\Delta\mathbf{p}_w^h,\, \Delta\boldsymbol{\theta}_w^h) = \phi_{\text{dec,wrist}}\big(\text{concat}(\mathbf{E}_{\text{g}}^h, \mathbf{e}_{\text{wrist}}^h)\big), \tag{26}$$

where $\mathbf{e}_{\text{wrist}}^h$ denotes the embedding of the wrist node in $\mathbf{E}_{\text{node}}^h$. Finally, stacking wrist and node-level outputs yields $\boldsymbol{\alpha}_{\text{prim}}^h$ as defined in Eq. (6).

4) Reward design: Following RobustDexGrasp [38], we define a grasping reward r_{grasp} that encourages stable contact formation under geometric and physical constraints, and add an additional feasibility penalty r_{pen} tailored for our cross-embodiment motion-primitive space:

$$r = r_{\text{grasp}} + r_{\text{pen}}. \tag{27}$$

Grasping reward: $r_{\text{grasp}} = w_{\text{dis}}\, r_{\text{dis}} + w_{\text{contact}}\, r_{\text{contact}} + w_{\text{force}}\, r_{\text{force}} + w_{\text{reg}}\, r_{\text{reg}}$, where $r_{\text{dis}} = -\sum_{i=1}^{N_h} \|\mathbf{d}_i^h\|_2$ penalizes distances from hand nodes to the object surface; $r_{\text{contact}} = \sum_{i=1}^{N_h} c_i^h$ rewards contact formation; $r_{\text{force}} = -\sum_{i=1}^{N_h} \max(0, f_i^h - f_0)^2$ penalizes excessive contact forces above threshold f_0; $r_{\text{reg}} = -\|\Delta\mathbf{q}^h\|_2$ regularizes joint command increments.

Feasibility penalty: $r_{\text{pen}} = -w_{\text{pen}} \sum_{i=1}^{N_h} \|(\mathbf{1} - \mathbf{m}_i^h) \odot \boldsymbol{\alpha}_i^h\|_2^2$ suppresses outputs on inactive primitive axes.

All weights ($w_{\text{dis}}, w_{\text{contact}}, w_{\text{force}}, w_{\text{reg}}, w_{\text{pen}} > 0$) are shared across embodiments to ensure consistent learning objectives.

D. Sim-to-Real Transfer via Privileged Distillation

In real-world deployment, contact states and interaction forces are typically not directly observable, while they are readily available as privileged signals in simulation. We follow the teacher–student distillation strategy of RobustDexGrasp [38]: we first train a privileged *teacher* policy in simulation with access to contact/force-related observations, and then distill it into a *student* policy that operates without such privileged inputs. The student uses MAGCN as the backbone and is equipped with an LSTM to perform temporal state estimation, enabling it to implicitly recover missing information (e.g., contacts and forces) from observation histories and to execute the distilled grasping behavior in real scenes (see Supplementary Sec. S2.5.1 and Sec. S2.5.2).

IV. EXPERIMENTS

We evaluate DexGrasp-Zero in both simulation and real-world, including cross-hand transfer, single-hand transfer, and ablations.

TABLE I: Cross-embodiment training and zero-shot transfer results (success rate). Variants of our method: *w/o* $\mathcal{G}_{physical}$ *priors* removes hand-specific physical property encoding; *early fusion* concatenates physical features with node states at input (instead of layer-wise fusion in GCN); *w/o motion primitives* replaces the motion-primitive space with raw joint commands; *w/o* $M_{activation}\&r_{pen}$ disables the activation mask conditioning and the corresponding action-feasibility penalty in the reward; **full model** is our complete DexGrasp-Zero policy.

Method	**Variant**	**Training Hands**				**Unseen Hands**		**Average**	
		Allegro	Shadow	Ability	Schunk	LEAP	Inspire	Seen	Unseen
CrossDex [35]	per-object	0.81	0.85	**0.90**	0.90	0.34	0.44	0.865	0.39
CrossDex [35]	multi-object	0.39	0.69	0.42	0.60	0.19	0.34	0.525	0.265
GET-Zero [16]	multi-object	0.83	0.89	0.84	0.90	0.62	0.32	0.865	0.47
DexGrasp-Zero (Ours)	*w/o motion primitives*	0.52	0.51	0.64	0.59	0.39	0.29	0.565	0.34
	early fusion	0.42	0.47	0.59	0.54	0.46	0.34	0.505	0.40
	w/o $\mathcal{G}_{physical}$ *priors*	0.91	0.89	**0.90**	0.84	0.82	0.79	0.885	0.805
	w/o $M_{activation}\&r_{pen}$	**0.92**	0.91	**0.90**	0.81	0.50	0.76	0.885	0.63
	full model	**0.92**	**0.95**	**0.90**	**0.91**	**0.93**	**0.82**	**0.92**	**0.85**

A. Experimental Setup

1) Simulation: We evaluate DexGrasp-Zero on six dexterous robot hands. Following the cross-embodiment protocol used by CrossDex [35], we train a single policy on four seen hands (Allegro, Shadow, Ability, and Schunk) and evaluate zero-shot transfer on two unseen hands (LEAP[21] and Inspire) in the RaiSim simulator [8].

a) Dataset and Benchmark:

CrossDex/YCB benchmark. To benchmark cross-hand generalization, we adopt the 45-object YCB split from CrossDex [35] where all objects are used for both training and testing. This protocol keeps the object set fixed so performance differences primarily reflect embodiment variation rather than changes in objects. We compare against CrossDex[35] under (i) its original *per-object* setting (one policy trained per object, averaged across objects) and (ii) a *multi-object* adaptation (a single policy jointly trained on all objects).

GraspXL benchmark. To study single-hand training and cross-hand transfer under a standard single-hand benchmark, we follow the protocol of GraspXL[36]: the training set consists of 26 ShapeNet [3] objects and 32 PartNet [14] objects, and evaluation is performed on 48 PartNet test objects. We compare against GraspXL in the single-hand setting.

b) Evaluation Protocol and Metric: In all simulation experiments, we evaluate with 25 grasp trials per object; the object is spawned at a fixed initial position while the hand starts from a randomized initial pose. We report the success rate (SR): the object is lifted up by 0.5 m and held for 2 s without dropping.

2) Real-World Evaluation: We deploy distilled policy (Sec. III-D) on 3 robot platforms: (i) Kinova arm with LEAP hand, (ii) Kinova arm with Inspire hand, and (iii) Piper arm with Revo2 hand (Fig. 4). We evaluate on 10 unseen objects with 5 random poses per object (50 trials per platform). A grasp is successful if the object is lifted to 30 cm and held for 5 s.

B. Implementation Details

Simulation training. We train all policies with PPO [19] and on a NVIDIA RTX 3090 GPU for 6000 rounds. The MAGCN backbone uses a 10-layer GCN. We instantiate 3 parallel simulation environment for each (hand, object) pair. The simulator runs at a 20 Hz. Each episode has 120 exploration steps, followed by 30 steps with an additional lift signal. We use the following reward coefficients: $w_{\text{dis}} = 0.3, w_{\text{contact}} = 1.0, w_{\text{force}} = 0.5, w_{\text{reg}} = 1.5, w_{\text{pen}} = 0.3$.

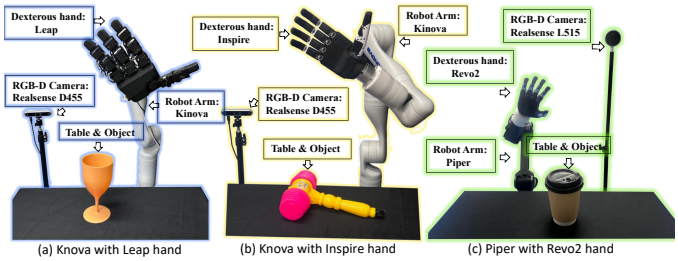

Fig. 4: Hardware setup. We evaluate our method on three robot platforms: (a)Kinova arm with LEAP hand, (b)Kinova arm with Inspire hand, and (c)Piper arm with Revo2 hand.

Real-world deployment details. Inference runs on a single NVIDIA RTX 3090 GPU at 20 Hz. Perception is provided by a single RGB-D camera. We standardize the initial robot configuration across platforms: the thumb starts with maximum opening distance. The arm starts 25 cm to the left of the object with the palm facing the object. During execution, we run the grasp controller for 130 steps and then send a lift command to raise the object.

C. Cross-Hand Zero-Shot Transfer from Multi-Hand Training

To evaluate cross-embodiment generalization, we train a single policy jointly on four hands (Allegro, Shadow, Ability, Schunk) on the 45-object YCB split and evaluate on both the training and unseen hands (LEAP and Inspire) without fine-tuning. We compare against CrossDex [35], a state-of-the-art method with open code for zero-shot cross-embodiment dexterous grasping. We additionally reproduce GET-Zero [16] under the same CrossDex protocol. Following GET-Zero, we construct a kinematic graph where each physical DoF is represented as a node, and each node outputs a single action scalar that directly controls the corresponding DoF (with the original axis-aligned mapping to physical joint commands for cross-hand evaluation).

Fig. 5: Simulated grasps of training hands on 5 diverse objects.

As shown in Table I, DexGrasp-Zero achieves strong performance across both seen and unseen hand embodiments. In particular, our full model reaches **92.0%** average success on the four training hands and **85.0%** on two unseen hands. Compared with the CrossDex multi-object baseline (52.5% seen/26.5% unseen), we substantially improve zero-shot transfer on unseen hands (26.5% $\rightarrow$ 85%), driven by our morphology-aligned graph design, physical-property injection, and action-space alignment.

As a point of comparison, GET-Zero achieves strong performance on the *training hands* (0.865 average), likely because its Transformer-based multi-hand training can learn hand-specific patterns with high discrimination on the seen embodiments. However, its success rate drops sharply on unseen hands (0.47 average), suggesting limited transfer when the action semantics are not explicitly unified across morphologies. In contrast, DexGrasp-Zero maintains high unseen-hand performance (0.85 average) by explicitly modeling a unified action space, which substantially improves zero-shot transfer across morphologically distinct hands.

D. Cross-Hand Zero-Shot Transfer from Single-Hand Training

Here we ask a more practical question: if we train DexGrasp-Zero on a *single* hand, can the resulting policy still transfer well to other embodiments, and how competitive is it compared with methods tailored for single-hand training?

Using the GraspXL benchmark, we train DexGrasp-Zero on one embodiment at a time and evaluate the resulting policy on all hands without any adaptation. We report success rates in Table II together with the GraspXL, a method design for single-hand training and can not transfer to new hand.

Overall, single-hand training with DexGrasp-Zero also achieves strong zero-shot transfer. Policies trained on Shadow or Schunk reach **0.94 on unseen Inspire**, surpassing the GraspXL specialist (0.91). Training on Schunk also yields **0.90 on LEAP** and exceeds GraspXL on Schunk itself (0.92 vs. 0.90), demonstrating effective cross-embodiment generalization. But this single-hand transfer performance is not uniformly stable, it depends heavily on morphological and dynamic similarities between the source and target hands.

TABLE II: Single-hand training transfer on PartNet (success rate): rows show the training embodiment for DexGrasp-Zero and columns show the test embodiment. Entries marked with † indicate *in-domain* evaluation.

Method	Train Hand	Allegro	Shadow	Ability	Schunk	LEAP	Inspire
GraspXL [36]	–	0.94	0.93	0.91	0.90	0.95	0.91
DexGrasp-Zero (Ours)	Allegro	$0.93^\dagger$	0.83	0.80	0.69	0.48	0.77
	Shadow	0.55	$0.98^\dagger$	0.80	0.82	0.77	0.94
	Ability	0.60	0.60	$0.86^\dagger$	0.86	0.88	0.90
	Schunk	0.61	0.52	0.83	$0.92^\dagger$	0.90	0.94
	LEAP	0.50	0.68	0.65	0.69	$0.90^\dagger$	0.69
	Inspire	0.61	0.83	0.87	0.88	0.88	$0.96^\dagger$

E. Ablation Studies

We conduct ablation studies to quantify the contributions of four key design choices: (1)conditioning the policy on URDF-derived *physical-property* priors via $\mathcal{G}_{\text{physical}}$, (2)the proposed layer-wise injection of task features with physical priors (vs. early fusion), (3)the hand-agnostic motion-primitive space, and (4)the activation mask $M_{\text{activation}}$ conditioning with the corresponding action-feasibility penalty r_{pen}. Result is summarized in Table I, removing any component degrades performance. Specifically:

- **Without motion primitives**, performance degrades markedly: the average success rate decreases from 92.0% to 56.5% on seen hands and from 85.0% to 34.0% on unseen hands, confirming that the motion-primitive provides essential inductive bias that facilitates structural and semantic alignment across different morphologies.
- **With early fusion** ($\mathcal{G}_{\text{task}} \oplus \mathcal{G}_{\text{physical}}$): replacing layer-wise injection with a one-shot feature concatenation performs substantially worse (from 92.5% to 50.0% on seen hands) and exhibits *highly unstable* learning dynamics, suggesting that injecting physical priors at every layer is important for stable optimization.
- **Without $\mathcal{G}_{\text{physical}}$ priors**, performance drops from 92.0% to 88.5% on seen hands and from 85.0% to 80.5% on unseen hands. This indicates that URDF-derived physical constraints provide useful embodiment conditioning, especially for zero-shot transfer.
- **Without $M_{\text{activation}}$ and r_{pen}**, performance drops from 85.0% to 63.0% on unseen hands (while remaining relatively high on seen hands: 92.0% to 88.5%). This suggests that explicitly conditioning on executable primitive axes and penalizing infeasible outputs is crucial for preventing invalid actions that do not transfer across embodiments.

These results demonstrate that physical-property conditioning, a stable injection strategy, motion primitives, and feasibility-aware action conditioning are all important for achieving strong zero-shot generalization across diverse hand embodiments.

F. Real-World Deployment

Under the unified experimental setup described in Section IV-A2, we evaluate our policy on three robot platforms.

We evaluate real-world transfer performance under the unified setup by comparing our cross-hand policy with an

TABLE III: Real-world zero-shot grasping success rate on unseen dexterous hands.

Method	LEAP	Inspire	Revo2	Average
Intra-hand (Oracle)	0.90	0.90	0.78	0.86
Cross-hand w/o $\mathcal{G}_{\text{physical}}$	0.84	0.80	0.62	0.75
Cross-hand (Ours)	0.88	0.86	0.72	0.82

intra-hand oracle and an ablation without $\mathcal{G}_{\text{physical}}$ (Table III). This ablation is chosen for real-world testing because physical priors are essential for embodiment realism.

The results in Table III reveal two key insights. First, our cross-hand policy achieves performance close to the intra-hand oracle (e.g., 90% vs. 88% on LEAP), demonstrating that training on diverse hands enables strong zero-shot transfer without sacrificing much capability. Second, removing URDF-derived physical-property priors causes a consistent performance drop (e.g., from 0.88 to 0.84 on LEAP and from 0.72 to 0.62 on Revo2), confirming that explicit physical constraints encoded by $\mathcal{G}_{\text{physical}}$ are essential for effective cross-embodiment generalization and for adaptively compensating varying link lengths and actuation limits. Revo2 shows the lowest successrate, likely due to its smaller size, which makes large objects harder to stably grasp.

V. Conclusion

We presented **DexGrasp-Zero** for zero-shot cross-embodiment dexterous grasping. Our main contributions are a morphology-aligned graph representation with motion primitives for structural and control-semantic alignment across morphologies and MAGCN, a GCN policy network with Physical Property Injection that incorporates URDF-derived constraints for stable grasping. Experiments show strong transfer to unseen hands (85% in simulation; 82% on real robots), validating its potential as a universal paradigm for cross-embodiment dexterous manipulation.

Future work. While our current formulation is designed for anthropomorphic dexterous hands and already supports heterogeneous actuation mechanisms, extending the same paradigm to non-anthropomorphic end-effectors (e.g., parallel-jaw grippers) and soft/continuum manipulators remains an open problem. We believe this will require new embodiment-aware abstractions and primitives beyond the current joint-based formulation. Another direction is to improve representations for fine-grained finger operations: message passing in deep GCNs can smooth local signals, and future work could explore stronger positional encoding and topology-aware features to better preserve subtle, contact-critical distinctions.

Acknowledgments

This work was supported partially by NSFC(92470202), Guangdong NSF Project (No.2023B1515040025), Guangdong Key Research and Development Program (No. 2024B0101040004, No. 2025B0909020002).

References

[1] Maria Attarian, Muhammad Adil Asif, Jingzhou Liu, Ruthrash Hari, Animesh Garg, Igor Gilitschenski, and Jonathan Tompson. Geometry matching for multi-embodiment grasping. In *Conference on Robot Learning (CORL)*, pages 1242–1256, 2023. URL https://arxiv.org/pdf/2312.03864.

[2] Berk Calli, Arjun Singh, Aaron Walsman, Siddhartha Srinivasa, Pieter Abbeel, and Aaron M. Dollar. The YCB object and Model set: Towards common benchmarks for manipulation research. In *International Conference on Advanced Robotics (ICAR)*, pages 510–517, 2015. doi: 10.1109/ICAR.2015.7251504. URL https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7251504.

[3] Angel X. Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. ShapeNet: An Information-Rich 3D Model Repository, 2015. URL https://arxiv.org/abs/1512.03012.

[4] Zeyuan Chen, Qiyang Yan, Yuanpei Chen, Tianhao Wu, Jiyao Zhang, Zihan Ding, Jinzhou Li, Yaodong Yang, and Hao Dong. ClutterDexGrasp: A Sim-to-Real System for General Dexterous Grasping in Cluttered Scenes. In *Conference on Robot Learning (CORL)*, 2025. URL https://arxiv.org/pdf/2506.14317v1.

[5] Hao-Shu Fang, Hengxu Yan, Zhenyu Tang, Hongjie Fang, Chenxi Wang, and Cewu Lu. AnyDexGrasp: General Dexterous Grasping for Different Hands with Human-level Learning Efficiency. In *ICLR Workshop: Towards Robots with Human-Level Abilities*, 2025. URL https://arxiv.org/abs/2502.16420.

[6] Xin Fei, Zhixuan Xu, Huaicong Fang, Tianrui Zhang, and Lin Shao. $\mathcal{T}(\mathcal{R}, \mathcal{O})$ Grasp: Efficient Graph Diffusion of Robot-Object Spatial Transformation for Cross-Embodiment Dexterous Grasping. *arXiv preprint arXiv:2510.12724*, 2025. URL https://arxiv.org/pdf/2510.12724.

[7] Linyi Huang, Hui Zhang, Zijian Wu, Sammy Christen, and Jie Song. FunGrasp: Functional Grasping for Diverse Dexterous Hands. *IEEE Robotics and Automation Letters (RAL)*, 10(6):6175–6182, 2025. doi: 10.1109/LRA.2025.3561573. URL https://doi.org/10.1109/LRA.2025.3561573.

[8] J. Hwangbo and D. Kang. Raisim: A High-Performance Physics Engine for Robotics. https://raisim.com/, 2023. Accessed: 2025-03-15.

[9] Hanwen Jiang, Shaowei Liu, Jiashun Wang, and Xiaolong Wang. Hand-object contact consistency reasoning for human grasps generation. In *Proceedings of the IEEE/CVF international conference on computer vision (ICCV)*, pages 11107–11116, 2021. URL https://openaccess.thecvf.com/content/ICCV2021/papers/Jiang_Hand-Object_Contact_Consistency_Reasoning_for_Human_Grasps_Generation_ICCV_2021_paper.pdf.

[10] Thomas N. Kipf and Max Welling. Semi-Supervised Classification with Graph Convolutional Networks, 2017. URL https://arxiv.org/abs/1609.02907.

[11] Puhao Li, Tengyu Liu, Yuyang Li, Yiran Geng, Yixin

Zhu, Yaodong Yang, and Siyuan Huang. GenDexGrasp: Generalizable Dexterous Grasping. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 8068–8074, 2023. doi: 10.1109/ICRA48891.2023.10160667. URL https://doi.org/10.1109/ICRA48891.2023.10160667.

[12] Yuhao Lin, Yi-Lin Wei, Haoran Liao, Mu Lin, Chengyi Xing, Hao Li, Dandan Zhang, Mark Cutkosky, and Wei-Shi Zheng. Typetele: Releasing dexterity in teleoperation by dexterous manipulation types. *arXiv preprint arXiv:2507.01857*, 2025.

[13] Jiaxin Lu, Hao Kang, Haoxiang Li, Bo Liu, Yiding Yang, Qixing Huang, and Gang Hua. Ugg: Unified generative grasping. In *European Conference on Computer Vision (ECCV)*, pages 414–433. Springer, 2024. URL https://www.ecva.net/papers/eccv_2024/papers_ECCV/papers/08453.pdf.

[14] Kaichun Mo, Shilin Zhu, Angel X. Chang, L. Yi, Subarna Tripathi, Leonidas J. Guibas, and Hao Su. PartNet: A Large-Scale Benchmark for Fine-Grained and Hierarchical Part-Level 3D Object Understanding. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 909–918, 2018. URL https://openaccess.thecvf.com/content_CVPR_2019/papers/Mo_PartNet_A_Large-Scale_Benchmark_for_Fine-Grained_and_Hierarchical_Part-Level_3D_CVPR_2019_paper.pdf.

[15] Donald A. Neumann. *Kinesiology of the Musculoskeletal System : Foundations for Rehabilitation*. Elsevier Health Sciences, 2009.

[16] Austin Patel and Shuran Song. GET-Zero: Graph Embodiment Transformer for Zero-Shot Embodiment Generalization. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 14262–14269, 2025. doi: 10.1109/ICRA55743.2025.11127922. URL https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=11127922.

[17] Dmytro Pavlichenko and Sven Behnke. Dexterous Pre-Grasp Manipulation for Human-Like Functional Categorical Grasping: Deep Reinforcement Learning and Grasp Representations. *IEEE Transactions on Automation Science and Engineering (TASE)*, 23:2231–2244, 2026. doi: 10.1109/TASE.2025.3541768. URL https://doi.org/10.1109/TASE.2025.3541768.

[18] Marco Santello, Martha Flanders, and John F. Soechting. Postural Hand Synergies for Tool Use. *The Journal of Neuroscience*, 18:10105 – 10115, 1998. URL https://www.jneurosci.org/content/18/23/10105.

[19] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal Policy Optimization Algorithms, 2017. URL https://arxiv.org/pdf/1707.06347.

[20] 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 (RAL)*, 5 (2):2286–2293, 2020. doi: 10.1109/LRA.2020.2969946. URL https://doi.org/10.1109/LRA.2020.2969946.

[21] Kenneth Shaw, Ananye Agarwal, and Deepak Pathak. LEAP Hand: Low-Cost, Efficient, and Anthropomorphic Hand for Robot Learning. *Robotics: Science and Systems (RSS)*, 2023. doi: 10.15607/RSS.2023.XIX.089. URL https://www.roboticsproceedings.org/rss19/p089.pdf.

[22] Qijin She, Shishun Zhang, Yunfan Ye, Ruizhen Hu, and Kai Xu. Learning Cross-Hand Policies of High-DOF Reaching and Grasping. In *European Conference on Computer Vision (ECCV)*, 2024. URL https://www.ecva.net/papers/eccv_2024/papers_ECCV/papers/04377.pdf.

[23] 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 (ICCV)*, pages 3891–3902, October 2023. URL https://openaccess.thecvf.com/content/ICCV2023/papers/Wan_UniDexGrasp_Improving_Dexterous_Grasping_Policy_Learning_via_Geometry-Aware_Curriculum_and_ICCV_2023_paper.pdf.

[24] Wenbo Wang, Fangyun Wei, Lei Zhou, Xi Chen, Lin Luo, Xiaohan Yi, Yizhong Zhang, Yaobo Liang, Chang Xu, Yan Lu, et al. Unigrasptransformer: Simplified policy distillation for scalable dexterous robotic grasping. In *Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR)*, pages 12199–12208, 2025. URL https://openaccess.thecvf.com/content/CVPR2025/papers/Wang_UniGraspTransformer_Simplified_Policy_Distillation_for_Scalable_Dexterous_Robotic_Grasping_CVPR_2025_paper.pdf.

[25] Yi-Lin Wei, Jian-Jian Jiang, Chengyi Xing, Xian-Tuo Tan, Xiao-Ming Wu, Hao Li, Mark Cutkosky, and Wei-Shi Zheng. Grasp as you say: Language-guided dexterous grasp generation. *Advances in Neural Information Processing Systems*, 37:46881–46907, 2024.

[26] Yi-Lin Wei, Haoran Liao, Yuhao Lin, Pengyue Wang, Zhizhao Liang, Guiliang Liu, and Wei-Shi Zheng. Cyclemanip: Enabling cyclic task manipulation via effective historical perception and understanding. *arXiv preprint arXiv:2512.01022*, 2025.

[27] Yi-Lin Wei, Mu Lin, Yuhao Lin, Jian-Jian Jiang, Xiao-Ming Wu, Ling-An Zeng, and Wei-Shi Zheng. Afforddexgrasp: Open-set language-guided dexterous grasp with generalizable-instructive affordance. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11818–11828, 2025.

[28] Yi-Lin Wei, Zhexi Luo, Yuhao Lin, Mu Lin, Zhizhao Liang, Shuoyu Chen, and Wei-Shi Zheng. Omnidexgrasp: Generalizable dexterous grasping via foundation model and force feedback. *arXiv preprint arXiv:2510.23119*, 2025.

[29] Yunze Wei, Maria Attarian, and Igor Gilitschenski. GeoMatch++: Morphology Conditioned Geometry Matching

for Multi-Embodiment Grasping, 2024. URL https://arxiv.org/abs/2412.18998.

[30] 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 *IEEE International Conference on Robotics and Automation (ICRA)*, pages 4982–4988, 2025. doi: 10.1109/ICRA55743.2025.11127754. URL https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=11127754.

[31] Zhengyang Kris Weng. BiDexHand: Design and Evaluation of an Open-Source 16-DoF Biomimetic Dexterous Hand, 2025. URL https://arxiv.org/abs/2504.14712.

[32] Guo-Hao Xu, Yi-Lin Wei, Dian Zheng, Xiao-Ming Wu, and Wei-Shi Zheng. Dexterous Grasp Transformer. In *Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR)*, pages 17933–17942, 2024. doi: 10.1109/CVPR52733.2024.01698. URL https://doi.org/10.1109/CVPR52733.2024.01698.

[33] Yinzhen Xu, Weikang Wan, Jialiang Zhang, Haoran Liu, Zikang Shan, Hao Shen, Ruicheng Wang, Haoran Geng, Yijia Weng, Jiayi Chen, Tengyu Liu, Li Yi, and He Wang. UniDexGrasp: Universal Robotic Dexterous Grasping via Learning Diverse Proposal Generation and Goal-Conditioned Policy. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4737–4746, 2023. URL https://openaccess.thecvf.com/content/CVPR2023/papers/Xu_UniDexGrasp_Universal_Robotic_Dexterous_Grasping_via_Learning_Diverse_Proposal_Generation_CVPR_2023_paper.pdf.

[34] Zhixuan Xu, Chongkai Gao, Zixuan Liu, Gang Yang, Chenrui Tie, Haozhuo Zheng, Haoyu Zhou, Weikun Peng, Debang Wang, Tianrun Hu, Tianyi Chen, Zhouliang Yu, and Lin Shao. ManiFoundation Model for General-Purpose Robotic Manipulation of Contact Synthesis with Arbitrary Objects and Robots. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 10905–10912, 2024. doi: 10.1109/IROS58592.2024.10801782. URL https://doi.org/10.1109/IROS58592.2024.10801782.

[35] Haoqi Yuan, Bohan Zhou, Yuhui Fu, and Zongqing Lu. Cross-Embodiment Dexterous Grasping with Reinforcement Learning. In *International Conference on Learning Representations (ICLR)*, volume 2025, pages 81413–81434, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/ca8c6f28d8ba1e732e3f217ab05c4ec0-Paper-Conference.pdf.

[36] Hui Zhang, Sammy Christen, Zicong Fan, Otmar Hilliges, and Jie Song. GraspXL: Generating Grasping Motions for Diverse Objects at Scale. In *European Conference on Computer Vision (ECCV)*, 2024. URL https://www.ecva.net/papers/eccv_2024/papers_ECCV/papers/03801.pdf.

[37] Hui Zhang, Sammy Christen, Zicong Fan, Luocheng Zheng, Jemin Hwangbo, Jie Song, and Otmar Hilliges. ArtiGrasp: Physically Plausible Synthesis of Bi-Manual Dexterous Grasping and Articulation. In *International Conference on 3D Vision (3DV)*, pages 235–246, 2024. doi: 10.1109/3DV62453.2024.00016. URL https://doi.org/10.1109/3DV62453.2024.00016.

[38] Hui Zhang, Zijian Wu, Linyi Huang, Sammy Christen, and Jie Song. RobustDexGrasp: Robust Dexterous Grasping of General Objects. In *Conference on Robot Learning (CoRL)*, 2025. URL https://arxiv.org/pdf/2504.05287.

[39] 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 *Conference on Robot Learning (CORL)*, 2024. URL https://arxiv.org/pdf/2410.23004.

[40] Yiming Zhong, Qi Jiang, Jingyi Yu, and Yuexin Ma. Dexgrasp anything: Towards universal robotic dexterous grasping with physics awareness. In *Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR)*, pages 22584–22594, 2025. URL https://openaccess.thecvf.com/content/CVPR2025/papers/Zhong_DexGrasp_Anything_Towards_Universal_Robotic_Dexterous_Grasping_with_Physics_Awareness_CVPR_2025_paper.pdf.

[41] Zelong Zhou, Wenrui Chen, Zeyun Hu, Qiang Diao, Qixin Gao, and Yaonan Wang. Design of an Adaptive Modular Anthropomorphic Dexterous Hand for Human-like Manipulation, 2025. URL https://arxiv.org/abs/2511.22100.