

MANGO-Grasp: Mahalanobis Fields over Geometry-Oriented 3D Gaussians for Cross-Embodiment Dexterous Grasping

Heng Zhang^{1,2}, Kevin Yuchen Ma^{1,3}, Mike Zheng Shou³, Weisi Lin^{2*} and Yan Wu^{1*}

Abstract—Cross-embodiment dexterous grasping aims to synthesize stable grasps across heterogeneous multi-fingered hands with little or no embodiment-specific tuning. Existing interaction-centric methods achieve promising results, but their object representations often underrepresent local surface geometry, while their robot descriptors do not explicitly encode both robot morphology and kinematics. We propose MANGO-Grasp, an anisotropic interaction framework that represents objects as geometry-oriented 3D Gaussian primitives and robot hands as surface keypoints encoded into morpho-kinematic descriptors. The object primitives are adaptively allocated by geometric complexity and shaped as surface-aligned plates with outward normals, encoding local geometry. Mahalanobis fields over keypoint-primitive pairs serve as interaction prediction targets during training and as optimization guidance for grasp realization at inference. These fields rise sharply for displacement along the surface normal but only gently within the tangent plane, matching the directional structure of contact. Grasps are realized with one shared optimization formulation and hyperparameter setting across all embodiments. On the CMAP and MultiGripperGrasp benchmarks, MANGO-Grasp outperforms the strongest seen-hand baseline by up to 8.24 percentage points in simulation. It also transfers zero-shot to the unseen SharpaWave hand, improving over the strongest zero-shot baseline by up to 16.57 percentage points, and achieves 86% success in real-world experiments. The code and additional materials will be made available upon publication at <https://connor-zh.github.io/MANGO-Grasp/>.

I. INTRODUCTION

Dexterous grasping is a fundamental capability for robotic manipulation, yet synthesizing stable grasps remains challenging due to the high-dimensional kinematics, complex contact constraints, and morphology-dependent feasibility of multi-fingered hands. Existing pipelines often rely on hand-engineered, embodiment-specific optimization [1]–[3] or train dedicated models for individual hands [4]–[6]. As robotic hardware continues to diversify, such cost scales poorly, motivating *cross-embodiment dexterous grasping*: unified models that can generalize across different hands.

Recent progress in cross-embodiment grasping has moved from hand- or object-centric formulations toward interaction-centric modeling. Hand-centric methods directly regress

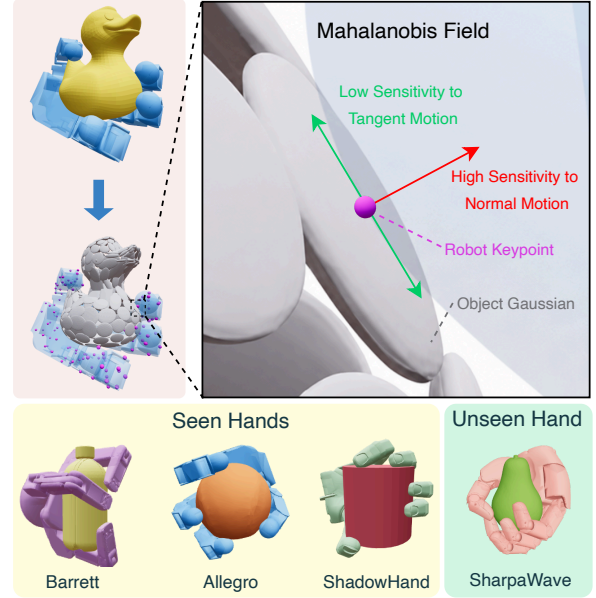


Fig. 1. Overview of the proposed anisotropic interaction formulation for cross-embodiment dexterous grasping. By representing objects with geometry-oriented 3D Gaussian primitives and hands with keypoints, we use Mahalanobis fields to model keypoint-primitive interaction with low sensitivity to surface tangential motion and high sensitivity to surface normal motion. By learning such transferable interaction fields, our framework generalizes across seen hands and transfers zero-shot to an unseen hand.

embodiment-specific configurations from object observations [7]–[10], but often struggle to transfer across hands with substantially different kinematics. Object-centric methods instead predict hand-agnostic contact targets [11]–[14], but leave a realization gap between object-side contacts and feasible hand configurations. Recent *interaction-centric* methods have achieved promising results by modeling hand-object spatial relations [15], [16], retaining embodiment awareness while capturing transferable contact structure. However, existing interaction-centric methods remain limited by how object geometry, robot morphology and kinematics, and hand-object interaction are represented.

On the object side, prior object representations commonly rely on uniformly sampled point clouds [15] or point-cloud patches [16]. Such representations allocate capacity independently of geometric complexity, which may under-represent contact-relevant high-curvature regions while over-sampling broad planar areas. Moreover, they do not explicitly encode local surface geometry. Consequently, hand-object compatibility is reduced to isotropic Euclidean proximity, although dexterous contact is direction-dependent: tangential sliding, and normal separation have distinct physical meanings.

On the robot side, existing representations either en-

* denotes the corresponding authors

¹Robotics & Autonomous Systems Division, Institute for Infocomm Research, Agency for Science, Technology and Research (A*STAR-I²R), Singapore wuy@i2r.a-star.edu.sg

²College of Computing and Data Science, Nanyang Technological University, Singapore HENG018@e.ntu.edu.sg, wslin@ntu.edu.sg

³Show Lab, National University of Singapore, Singapore yuchen_ma@u.nus.edu, mikeshou@nus.edu.sg

This work has been submitted to the IEEE for possible publication. Copyright may be transferred without notice, after which this version may no longer be accessible.

code URDF-derived link geometry and spatial pose [16], or learn dense configuration-invariant keypoint correspondences [15]. These designs improve embodiment awareness, but they do not explicitly couple morphological identity with configuration-dependent motion. Since grasp feasibility depends on both what the shape of a hand part is and how it can move, transferable robot descriptors should jointly encode morphology and kinematic behavior.

Motivated by these observations, we propose an *anisotropic interaction formulation* for cross-embodiment dexterous grasping, as illustrated in Fig. 1. On the object side, we adapt 3D Gaussian Splatting [17] into a geometry-oriented primitive generation pipeline that produces a fixed-size set of *surface-aligned, plate-like 3D Gaussian primitives* with assigned outward normals, preserving local tangent-normal geometry while adaptively allocating capacity according to local geometric complexity. On the robot side, we pretrain *morpho-kinematic* hand-keypoint descriptors by combining morphology-identity learning with a kinematic-awareness objective that captures cross-configuration motion.

Given these object and robot representations, we formulate robot–object interaction as Mahalanobis fields over hand keypoints and object Gaussian primitives. Unlike isotropic Euclidean proximity, these fields encode contact compatibility in a local anisotropic frame, permitting tangential variation within compatible surface patches while remaining sensitive to surface-normal deviation. Through the anisotropic scales of each primitive, broad planar primitives induce larger lateral tolerance, whereas compact primitives impose stricter localization. The predicted Mahalanobis fields serve as both interaction prediction targets and surface-aware guidance signals for grasp realization, which is performed using a shared optimization setup across embodiments.

We evaluate our method on CMAP [12] and MultiGripperGrasp [18] with three seen hands (ShadowHand, Allegro, and Barrett) and one unseen hand (SharpaWave). Our method achieves seen-hand simulation success rates of 97.59% and 89.47%. The same model transfers zero-shot to the unseen SharpaWave hand, obtaining 84.17% and 81.47% success on the two benchmarks, and reaches 86% success on the real hand without real-world fine-tuning. Ablation studies further validate the contribution of each proposed component.

Our main contributions are as follows.

- We adapt a 3D Gaussian Splatting pipeline to convert object geometry into a fixed-size set of geometry-oriented 3D Gaussian primitives, preserving local surface geometry while adaptively allocating capacity according to geometric complexity.
- We propose a *morpho-kinematic* robot pretraining paradigm that learns hand-keypoint descriptors by capturing morphology identity and kinematic awareness.
- We formulate cross-embodiment robot–object interaction as *Mahalanobis fields* between hand keypoints and object Gaussian primitives, enabling anisotropic contact compatibility and surface-aware grasp realization.
- We validate the method in simulation across seen and unseen embodiments and on a real unseen SharpaWave

hand, demonstrating strong benchmark performance, zero-shot transfer, and direct hardware deployment.

II. RELATED WORK

We review prior work from three perspectives central to cross-embodiment dexterous grasping: robot encoding, object encoding, and interaction modeling.

Robot Encoding. Cross-embodiment dexterous grasping requires robot representations that transfer across heterogeneous hands while preserving morphology and kinematics. Object-centric methods encode the robot implicitly through hand-agnostic contacts or geometry embeddings [11]–[13], [19], but leave a gap between contact prediction and kinematic realization. Hand-centric methods expose URDF-derived graphs, link features, kinematic structures, or eigen-grasp bases [10], [16], [20], yet these inputs do not guarantee that the learned features capture morphology and kinematics. DRO [15] pretrains configuration-invariant keypoint descriptors through contrastive alignment, capturing morphology-level correspondence but not explicitly supervising kinematic behavior. We build on this paradigm by adding kinematic-awareness supervision, producing morpho-kinematic descriptors that preserve keypoint identity while remaining predictive of configuration-dependent mobility.

Object Encoding. Most cross-embodiment methods encode objects as point clouds [10], [11], [15], [19]–[21] or point-sampled patches and graphs [13], [14], [16]. Although general, these encodings allocate capacity mainly through sampling density and lack explicit surface geometry modeling, limiting their ability to capture contact-relevant geometry and distinguish tangential sliding, normal separation, and penetration. Motivated by 3D Gaussian Splatting [17] and surface-oriented variants [22], [23], we adapt localized anisotropic Gaussian primitives from visual reconstruction to dexterous contact modeling. Unlike these rendering-oriented representations, our method constructs a fixed-size set of geometry-oriented 3D Gaussian primitives whose adaptive density, local frames, outward normals, and anisotropic scales support contact-aware robot–object reasoning.

Interaction Modeling. Interaction-centric methods have recently improved cross-embodiment grasping by modeling spatial relations between hands and objects. DRO [15] represents interaction as a dense Euclidean distance matrix between robot keypoints and object points, while TRO [16] models transformations between hand links and object patches through graph diffusion. These formulations retain embodiment awareness and provide transferable contact structure, but their interaction metrics remain limited in directional expressiveness. Euclidean distances treat all directions equally, and patch-level transformations do not explicitly encode the local tangent-normal asymmetry of contact. We instead define interaction as Mahalanobis fields between robot keypoints and surface-aligned Gaussian plates, allowing contact compatibility to vary according to each primitive’s local frame and anisotropic scale.

III. METHOD

A. Problem Formulation

We address *cross-embodiment dexterous grasp synthesis*: given a target object and a robotic hand embodiment e from a set of heterogeneous candidates, produce a hand configuration $\mathbf{q}^*$ (joint angles together with a 6-DoF wrist pose) that yields a kinematically feasible and stable grasp.

Object Input. Each object mesh is processed into G 3D Gaussian primitives. We set $G = 256$ empirically to balance geometry coverage and downstream interaction matrix size.

$$\mathcal{G} = \{g_j\}_{j=1}^G, \quad g_j = (\boldsymbol{\mu}_j, \mathbf{R}_j, \boldsymbol{\sigma}_j, \mathbf{n}_j), \quad (1)$$

where $\boldsymbol{\mu}_j \in \mathbb{R}^3$ denotes the primitive center, $\mathbf{R}_j \in SO(3)$ its orientation, $\boldsymbol{\sigma}_j \in \mathbb{R}_+^3$ its anisotropic scale, and $\mathbf{n}_j \in \mathbb{S}^2$ its outward surface normal.

Robot Input. Following [15], we represent each embodiment e by N surface keypoints, obtained once via farthest point sampling over the link meshes at a canonical pose. We set $N = 256$ empirically to balance hand-surface resolution and the computational cost of downstream interaction matrix. Let $\mathcal{P}_e(\mathbf{q})$ denote the resulting keypoint set of embodiment e at configuration $\mathbf{q}$, obtained by mapping the canonical keypoints through analytical forward kinematics:

$$\mathcal{P}_e(\mathbf{q}) = \{\mathbf{p}_i(\mathbf{q})\}_{i=1}^N \subset \mathbb{R}^3, \quad (2)$$

where $\mathbf{p}_i$ is the i -th base-frame keypoint, corresponding to a fixed surface site across configurations.

Interaction Fields Generation and Grasp Realization.

We factor grasp synthesis into interaction fields generation and grasp realization. Given hand keypoints $\mathbf{P}_e^0 = \mathcal{P}_e(\mathbf{q}^0)$ from an initial configuration $\mathbf{q}^0$ and object primitives $\mathcal{G}$, a model f_θ generates target Mahalanobis fields, represented by a matrix $\hat{\mathbf{M}} \in \mathbb{R}_+^{N \times G}$. Each entry $\hat{M}_{ij}$ specifies the target Mahalanobis distance between keypoint i and primitive g_j at the target grasp. The final grasp is realized by solving:

$$\mathbf{q}^* = \arg \min_{\mathbf{q} \in \mathcal{Q}_e} \mathcal{L}_{\text{rec}}(\mathbf{q}; \hat{\mathbf{M}}, \mathcal{G}), \quad (3)$$

initialized from $\mathbf{q}^0$. Here, $\mathcal{Q}_e = \{\mathbf{q} \mid \underline{\mathbf{q}}_r \leq \mathbf{q}_r \leq \bar{\mathbf{q}}_r\}$ enforces revolute joint limits, and $\mathcal{L}_{\text{rec}}$ combines Mahalanobis field guidance with penetration and self-collision energies.

B. Method Overview

An overview of MANGO-Grasp is shown in Fig. 2. First, the object mesh is converted into a fixed-budget set of geometry-oriented Gaussians $\mathcal{G}$ (Sec. III-C). Second, a robot encoder is pretrained to extract Morpho-Kinematic Robot Descriptors $\mathbf{F}^R$ for the initial hand keypoints $\mathbf{P}_e^0$ (Sec. III-D). Third, the Interaction Fields Generator fuses $\mathbf{F}^R$ with Primitive-wise Object Features $\mathbf{F}^O$ encoded from $\mathcal{G}$, and generates the target Mahalanobis fields $\hat{\mathbf{M}}$ (Sec. III-E). Finally, grasp realization optimizes the hand configuration $\mathbf{q}$ from initial configuration $\mathbf{q}^0$ with the target Mahalanobis fields $\hat{\mathbf{M}}$, joint-limit constraints, penetration and self-collision energies to obtain final grasp $\mathbf{q}^*$ (Sec. III-F).

C. Geometry-Oriented 3D Gaussian Construction

We adapt 3D Gaussian Splatting [17], implemented with Nerfstudio [24], to convert an object mesh into a fixed-budget set of surface-aligned, plate-like 3D Gaussian primitives with assigned outward normals that encode geometry rather than appearance. The pipeline has four stages: geometry-driven densification allocates primitives according to local geometric complexity using mesh-derived signals; fixed-budget selection retains G primitives via a geometric retention score; surface alignment regularizes each primitive into a plate aligned with the local surface; and outward normal assignment assigns each primitive an outward-pointing normal.

1) *Geometry-Driven Densification:* Standard 3DGS optimizes T anisotropic primitives by photometric loss [17], using gradient-driven adaptive density control to clone or split primitives with large view-space positional gradients and prune low-opacity ones. However, under appearance-based supervision, these gradients may be dominated by texture rather than geometry. Primitives could be over-allocated to texturally rich but geometrically simple regions, while redundant floaters or overlapping splats may survive as long as their opacity remains above the pruning threshold.

We instead drive densification with mesh-rendered normal supervision. For each view, we encode the surface normals as RGB channels to form the target image C^{gt} , with $\hat{C}$ the corresponding rendering from the Gaussian primitives. This redirects the optimization from reconstructing appearance to reconstructing surface geometry: since normals are encoded as color, curved regions yield sharp color gradients, whereas planar regions stay nearly uniform. Since densification is triggered by accumulated image-space gradients, this supervision produces more, smaller primitives in geometrically complex regions and fewer, larger ones in simple regions.

To further strengthen densification in geometrically complex regions, we weight the standard 3DGS $\mathcal{L}_1$ photometric term by a curvature-aware map W from the mesh-rendered depth map D^{gt} :

$$W(u, v) = 1 + \text{Norm}(|\partial_u D^{\text{gt}}(u, v)| + |\partial_v D^{\text{gt}}(u, v)|), \quad (4)$$

where (u, v) indexes pixels, ∂_u, ∂_v are image-axis finite differences, and $\text{Norm}(\cdot)$ rescales the depth gradient to $[0, 1]$ over valid pixels. The unit offset retains supervision on smooth regions, while the additive term up-weights reconstruction errors at depth discontinuities and sharp transitions. With $\lambda_C = 0.8$ and $\lambda_S = 0.2$, the photometric loss is:

$$\mathcal{L}_{\text{photo}} = \lambda_C \|W \odot |\hat{C} - C^{\text{gt}}|\|_1 + \lambda_S \mathcal{L}_{\text{D-SSIM}}(\hat{C}, C^{\text{gt}}). \quad (5)$$

To remove redundant primitives, we add a soft opacity-sparsity term and optimize the objective at this stage:

$$\mathcal{L}_{\text{geo}} = \mathcal{L}_{\text{photo}} + \lambda_{\text{opa}} \mathcal{L}_{\text{opa}}, \quad \mathcal{L}_{\text{opa}} = \frac{1}{T} \sum_{i=1}^T o_i, \quad (6)$$

where o_i is the opacity of the i -th primitive and $\lambda_{\text{opa}} = 0.02$. Under this term, redundant floaters and overlapping primitives are driven toward transparency and removed by the standard opacity-pruning mechanism, whereas primitives

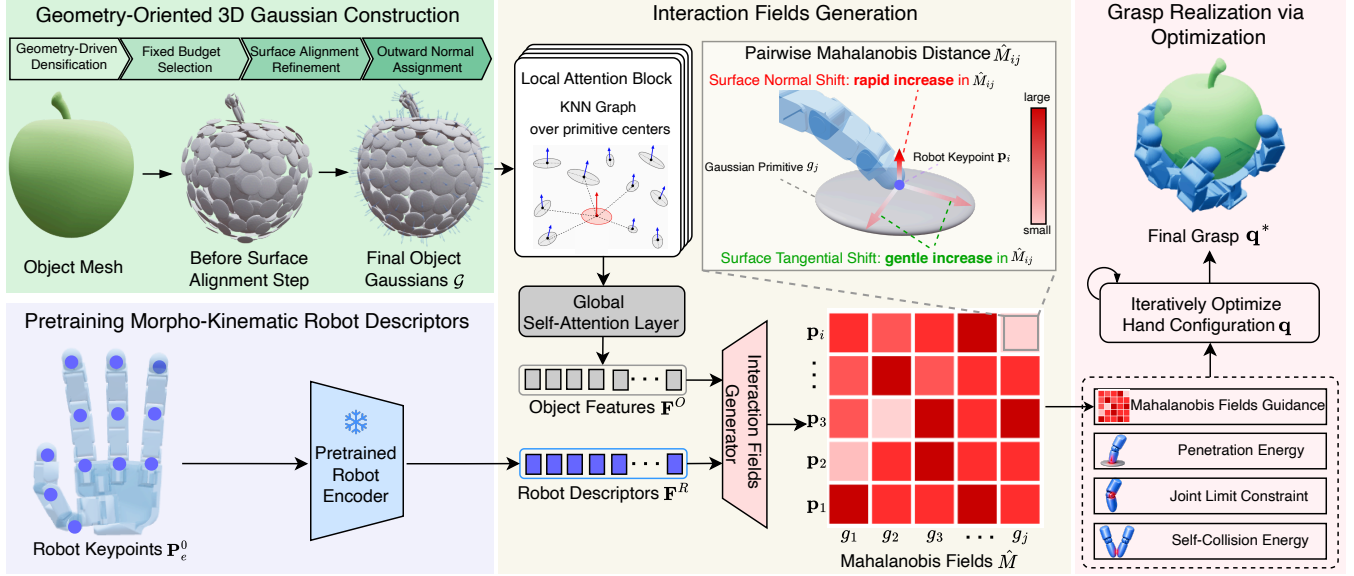


Fig. 2. Method overview. The object mesh is converted into a fixed-budget set of surface-aligned, plate-like 3D Gaussian primitives $\mathcal{G}$ with outward normals. The primitives are then encoded by a stack of local attention blocks, operating on a KNN graph built over primitive centers, followed by global self-attention layers, yielding Object Features $\mathbf{F}^O$. In parallel, initial robot keypoints $\mathbf{P}_e^0$ are encoded by a pretrained robot encoder into morpho-kinematic Robot Descriptors $\mathbf{F}^R$. The Interaction Fields Generator fuses both streams to generate the Mahalanobis fields $\hat{\mathbf{M}}$, which encode the interaction between each robot keypoint and each object primitive: the field value rises rapidly along the surface-normal direction and varies gently within the local tangent plane. The predicted fields then guide optimization from $\mathbf{q}^0$ to the final grasp $\mathbf{q}^*$ under joint-limit constraints, penetration energy, and self-collision energy.

essential for normal-view reconstruction retain sufficient opacity through $\mathcal{L}_{\text{photo}}$. $\mathcal{L}_{\text{opa}}$ is enabled only after a short warm-up period. Consequently, sharp and highly curved regions receive fine-scale Gaussian coverage, whereas smooth, approximately planar regions retain fewer non-redundant primitives that can cover broader surface patches.

2) *Fixed-Budget Selection*: To obtain a fixed-budget primitive set for the downstream model, we retain the top- G primitives according to $\text{score}(g_j) = o_j A_j (1 + \beta H_j)$, where A_j is the maximum projected area, and $H_j \in [0, 1]$ is the normalized discrete mean curvature of the nearest mesh vertex to μ_j . The term $o_j A_j$ favors primitives with substantial surface coverage, while H_j biases selection toward sharp geometric features. We set $\beta = 0.2$ to cap this curvature boost and prevent edge primitives from dominating flat regions.

3) *Surface Alignment Refinement*: Inspired by [22], [23], we refine each retained primitive into a thin plate that adheres to the object surface and aligns with its local geometry. With the budget fixed at G primitives, densification and culling are disabled, so optimization is restricted entirely to the geometric parameters $\{\mu_j, \mathbf{R}_j, \sigma_j\}_{j=1}^G$ under the objective:

$$\mathcal{L}_{\text{refine}} = \mathcal{L}_{\text{photo}} + \lambda_s \mathcal{L}_{\text{surf}} + \lambda_n \mathcal{L}_{\text{norm}} + \lambda_p \mathcal{L}_{\text{plate}}, \quad (7)$$

where $\mathcal{L}_{\text{photo}}$ is retained to further drive existing primitives to translate and expand across unrepresented surface regions. The remaining three geometric regularizers act on each primitive against its closest sampled point $\mathbf{x}_j$ on the source mesh and the surface normal $\mathbf{n}_j^m$ at that location.

Surface attraction pulls each primitive center to its closest mesh point:

$$\mathcal{L}_{\text{surf}} = \sum_j \|\mu_j - \mathbf{x}_j\|^2. \quad (8)$$

Normal alignment forces the shortest principal axis $\mathbf{n}_j^s$ of the primitive to remain parallel to $\mathbf{n}_j^m$, ensuring the broad face of the plate becomes tangent to the surface:

$$\mathcal{L}_{\text{norm}} = \sum_j (1 - |\mathbf{n}_j^s \cdot \mathbf{n}_j^m|). \quad (9)$$

Plate regularization acts on the sorted primitive scales $s_j^{(1)} \leq s_j^{(2)} \leq s_j^{(3)}$ of σ_j through one-sided hinge penalties, $[x]_+ = \max(0, x)$:

$$\mathcal{L}_{\text{plate}} = \sum_j \left([s_j^{(1)} - \tau_h]_+ + [\tau_\ell - s_j^{(1)}]_+ + \sum_{k=2}^3 [\tau_f - s_j^{(k)}]_+ \right), \quad (10)$$

where the first two terms confine the primitive thickness $s_j^{(1)}$ to a narrow band $[\tau_\ell, \tau_h]$. The final term floors the in-plane axes at τ_f , encouraging the plates to stretch tangentially along the surface to broaden geometric coverage.

4) *Outward Normal Assignment*: The normal-alignment loss $\mathcal{L}_{\text{norm}}$ aligns the shortest principal axis $\mathbf{n}_j^s$ with the local surface normal but does not determine its sign. After refinement, we project each primitive center μ_j onto the source mesh and obtain the outward normal $\mathbf{n}_j^m$ of the closest surface element. We then set $\mathbf{n}_j = \text{sgn}(\mathbf{n}_j^s \cdot \mathbf{n}_j^m) \mathbf{n}_j^s$, to ensure that all primitive normals point outward.

D. Pretraining Morpho-Kinematic Robot Descriptors

We pretrain a robot encoder to produce morpho-kinematic descriptors for hand keypoints. For dexterous grasping, each descriptor should encode configuration-invariant morphological identity while remaining predictive of keypoint motion under changes in hand configuration. To this end, we combine a morphology-identity contrastive objective with a kinematic awareness objective, as illustrated in Fig. 3.

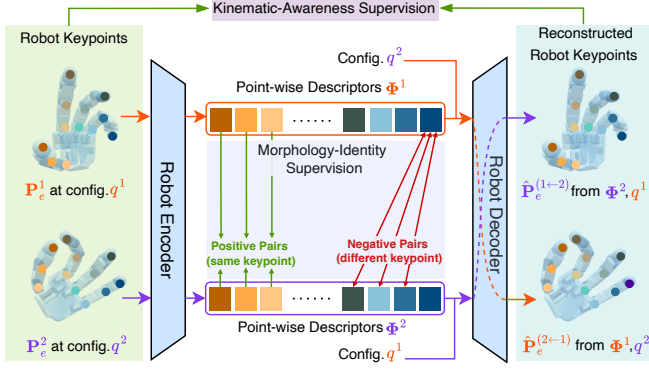


Fig. 3. Pretraining of morpho-kinematic robot descriptors. For two configurations $\mathbf{q}^1$ and $\mathbf{q}^2$ of the same embodiment, a robot encoder maps the corresponding keypoint sets $\mathbf{P}_e^1$ and $\mathbf{P}_e^2$ to point-wise descriptors Φ^1 and Φ^2 . The Morphology-Identity Supervision aligns descriptors of the same physical keypoints across configurations while separating those of different keypoints. Kinematic-Awareness Supervision uses a robot decoder to reconstruct $\hat{\mathbf{P}}_e^{(2 \leftarrow 1)}$ from Φ^1 and $\mathbf{q}^2$, and $\hat{\mathbf{P}}_e^{(1 \leftarrow 2)}$ from Φ^2 and $\mathbf{q}^1$, minimizing errors to the original keypoint sets $\mathbf{P}_e^2$ and $\mathbf{P}_e^1$, respectively.

Pretraining Setup. For each pretraining instance, we sample an embodiment e and two hand configurations $\mathbf{q}^1$ and $\mathbf{q}^2$. The corresponding keypoint sets $\mathbf{P}_e^1$ and $\mathbf{P}_e^2$ are obtained by forward kinematics. A DGCNN-based robot encoder [25] E_{robot} maps each keypoint set to point-wise descriptors: $\Phi^k = E_{\text{robot}}(\mathbf{P}_e^k) = \{\phi_i^k \in \mathbb{R}^{512}\}_{i=1}^N$, $k \in \{1, 2\}$.

Morphology-Identity Supervision. We follow the geometry-aware contrastive formulation of [15]. To preserve morphological identity, descriptors of the same physical keypoint should remain consistent across configurations, while descriptors of different keypoints should remain distinguishable. The descriptor similarity is defined as $S_{ij} = \frac{(\phi_i^1)^\top \phi_j^2}{\|\phi_i^1\|_2 \|\phi_j^2\|_2}$. Let $d_{ij} = \|\mathbf{p}_i^1 - \mathbf{p}_j^1\|_2$. The contrastive logit is defined as:

$$z_{ij} = \log\left(\frac{\tanh d_{ij}}{\max_k \tanh d_{ik}} + \delta_{ij}\right) + \frac{S_{ij}}{\tau}, \quad (11)$$

where δ_{ij} is the Kronecker delta and $\tau = 0.1$ is the temperature. Further, the morphology-identity loss is:

$$\mathcal{L}_{\text{morph}} = - \sum_i \log \frac{\exp(z_{ii})}{\sum_j \exp(z_{ij})}. \quad (12)$$

Kinematic-Awareness Supervision. To make descriptors predictive of keypoint motion under changes in hand configuration, we reuse the same configuration pair and impose a cross-conditioned prediction task. Given descriptors extracted from a source configuration and the target hand configuration, robot decoder D_{fk} predicts the target keypoint locations: $\hat{\mathbf{P}}_e^{(2 \leftarrow 1)} = D_{\text{fk}}(\Phi^1, \mathbf{q}^2)$, $\hat{\mathbf{P}}_e^{(1 \leftarrow 2)} = D_{\text{fk}}(\Phi^2, \mathbf{q}^1)$. Since grasp stability is dominated by contact-relevant regions, we further weight the regression by an offline, embodiment-level contact prior. For each embodiment e , near-contact frequency statistics are accumulated over hand-mesh vertices from the training grasps and propagated to keypoints by nearest-neighbor association in the local link frame. Letting $c_i^e \in [0, 1]$ denote this prior for the i -th

keypoint of embodiment e , per-keypoint weights are:

$$w_i^e = \frac{c_i^e + \alpha}{\frac{1}{N} \sum_{j=1}^N (c_j^e + \alpha)}, \quad (13)$$

where $\alpha = 0.1$ floors near-zero weights. The contact-aware kinematic-awareness loss is then:

$$\mathcal{L}_{\text{kin}} = \sum_{i=1}^N w_i^e (\|\hat{\mathbf{p}}_i^{(1 \leftarrow 2)} - \mathbf{p}_i^{(1)}\|_2^2 + \|\hat{\mathbf{p}}_i^{(2 \leftarrow 1)} - \mathbf{p}_i^{(2)}\|_2^2). \quad (14)$$

Overall Objective. The two pretraining terms jointly learn morpho-kinematic descriptors, giving the final objective:

$$\mathcal{L}_{\text{pre}} = \mathcal{L}_{\text{morph}} + \mathcal{L}_{\text{kin}}. \quad (15)$$

E. Interaction Fields Generation

1) *Modeling Interaction with Mahalanobis Fields:* We define the interaction fields between the robot and the object as a pairwise Mahalanobis distance [26] matrix $\mathbf{M} \in \mathbb{R}^{N \times G}$ measured between robot keypoints and object primitives. For a keypoint $\mathbf{p}_i$ and primitive $g_j = (\boldsymbol{\mu}_j, \mathbf{R}_j, \boldsymbol{\sigma}_j)$, the field is

$$M_{ij} = \sqrt{(\mathbf{R}_j^\top (\mathbf{p}_i - \boldsymbol{\mu}_j))^\top \text{diag}(\boldsymbol{\sigma}_j^{-2}) (\mathbf{R}_j^\top (\mathbf{p}_i - \boldsymbol{\mu}_j))}. \quad (16)$$

Since our primitives are surface-aligned plates, with broad faces tangent to the surface and short axis along the surface normal, the resulting anisotropic sensitivity reflects the geometry of contact: tangential motion within a surface patch preserves contact, whereas motion away from the surface breaks it. Sensitivity also adapts automatically to local complexity, with broad plates on planar regions admitting lateral flexibility and small, dense plates on sharp regions demanding precision in every direction.

2) *Network Architecture:* Initialized robot keypoints $\mathbf{P}_e^0$ are encoded by the frozen pretrained robot encoder to obtain Keypoint-wise Robot Descriptors $\mathbf{F}^R \in \mathbb{R}^{N \times d}$. And the object branch embeds each Gaussian primitive into a feature vector and builds a KNN graph over primitive centers. Stacked local attention blocks refine these features using neighborhood edge features that encode relative position and scale between neighboring primitives. A global self-attention layer then mixes information across all primitives, injecting object-level context into each token, producing Primitive-wise Object Features $\mathbf{F}^O \in \mathbb{R}^{G \times d}$.

Given $\mathbf{F}^R$ and $\mathbf{F}^O$, the Interaction Fields Generator generates $\hat{\mathbf{M}}$. Following [15], it first contextualizes the two streams with bidirectional cross-attention, yielding $\tilde{\mathbf{F}}^R$ and $\tilde{\mathbf{F}}^O$. It then samples a latent code $\mathbf{z} \in \mathbb{R}^{d_z}$ from a CVAE [27], drawing from the posterior in training and from $\mathcal{N}(\mathbf{0}, \mathbf{I})$ at inference. Finally, for each pair (i, j) , it concatenates $\tilde{\mathbf{F}}_i^R$ and $\tilde{\mathbf{F}}_j^O$ with $\mathbf{z}$, and passes the result through a non-negative pairwise MLP to predict $\hat{M}_{ij}$, giving $\hat{\mathbf{M}} \in \mathbb{R}^{N \times G}$ over all $N \times G$ pairs.

3) *Training Objective:* We supervise the network against the ground truth $\mathbf{M}^*$, computed by evaluating Eq. (16) between the grasp-state hand keypoints and the corresponding object Gaussian primitives. Since the resulting targets are heavy-tailed: contact pairs cluster near zero while distant

pairs span much larger values, we apply a $\log(1+x)$ compression to both prediction and target. We also reuse the contact-aware keypoint weights w_i^e from Sec. III-D to concentrate supervision on grasp-relevant keypoints:

$$\mathcal{L}_{\text{reg}} = \frac{1}{NG} \sum_{i,j} w_i^e |\log(1 + \hat{M}_{ij}) - \log(1 + M_{ij}^*)|. \quad (17)$$

A KL term $\mathcal{L}_{\text{kl}}$ regularizes the CVAE posterior. Using $\lambda_{\text{reg}} = 10$ and $\lambda_{\text{kl}} = 0.01$, we optimize:

$$\mathcal{L} = \lambda_{\text{reg}} \mathcal{L}_{\text{reg}} + \lambda_{\text{kl}} \mathcal{L}_{\text{kl}}. \quad (18)$$

F. Grasp Realization via Optimization

We initialize optimization from an object-conditioned pre-grasp configuration $\mathbf{q}^0$. Specifically, we sample an object primitive g_j , place the wrist anchor at an offset from $\boldsymbol{\mu}_j$ along the outward normal $\mathbf{n}_j$, and orient the hand approach direction toward the object with a random roll around $\mathbf{n}_j$. The finger joints are initialized to an open configuration. Based on the predicted target Mahalanobis field $\hat{\mathbf{M}}$, we realize a stable, kinematically feasible grasp $\mathbf{q}^*$ by optimizing the hand configuration $\mathbf{q}$:

$$\min_{\mathbf{q}} \lambda_{\text{guide}} \mathcal{L}_{\text{guide}} + \lambda_{\text{pen}} \mathcal{L}_{\text{pen}} + \lambda_{\text{self}} \mathcal{L}_{\text{self}}, \quad (19)$$

and the revolute joints $\mathbf{q}_r$ are constrained to joint limit $[\underline{\mathbf{q}}_r, \bar{\mathbf{q}}_r]$. *Requiring no embodiment-specific retuning*, we use one weight setting $(\lambda_{\text{guide}}, \lambda_{\text{pen}}, \lambda_{\text{self}}) = (50, 60, 10)$ for all hands. The remaining three terms are defined as follows.

Mahalanobis Fields Guidance. $\mathcal{L}_{\text{guide}}$ guides the current Mahalanobis fields toward the prediction. The anisotropy of the fields translates into a guidance signal that pulls contact-seeking keypoints toward the surface while allowing tangential adjustment within compatible surface patches, constraining the grasp where contact matters and preserving flexibility elsewhere. $\log(1+x)$ compression is used to both fields for stability. The contact-aware keypoint weights w_i^e from Sec. III-D are reused to focus the guidance on grasp-relevant regions of the hand:

$$\mathcal{L}_{\text{guide}} = \frac{1}{NG} \sum_{i,j} w_i^e \cdot \psi_g(\log(1 + M_{ij}), \log(1 + \hat{M}_{ij})), \quad (20)$$

where $\psi_g(a, b) = [a - b]_+$ penalizes the current field only when it exceeds the prediction. This leaves the optimizer free to pull in tighter than predicted when the geometry permits, with over-closure handled by the penetration term $\mathcal{L}_{\text{pen}}$.

Penetration Energy. $\mathcal{L}_{\text{pen}}$ penalizes hand-object penetration using object Gaussian primitives. For each hand keypoint $\mathbf{p}_i$, we select its nearest primitive under the Mahalanobis metric, $j^*(i) = \arg \min_j M_{ij}$, and compute its signed normal offset $s_i = (\mathbf{p}_i - \boldsymbol{\mu}_{j^*(i)})^\top \mathbf{n}_{j^*(i)}$. The penetration loss penalizes the worst K violations:

$$\mathcal{L}_{\text{pen}} = \text{TopKSum}_{i,K}[\psi_p(s_i)], \quad (21)$$

where $\psi_p(s) = (m - s)_+$, and $\text{TopKSum}_{i,K}$ sums the largest K values over i . We set $m = -0.001$ and $K = 20$.

Self-Collision Energy. $\mathcal{L}_{\text{self}}$ penalizes close keypoint pairs on non-adjacent links:

$$\mathcal{L}_{\text{self}} = \text{TopKSum}_{i,K} \left[\max_{j: d_k(\ell_i, \ell_j) > 1} \psi_s(\|\mathbf{p}_i - \mathbf{p}_j\|_2) \right], \quad (22)$$

where, $\psi_s(r) = (\epsilon_s - r)_+$, d_k denotes graph distance in the hand kinematic tree, and $\text{TopKSum}_{i,K}$ sums the largest K values over i . We set $\epsilon_s = 0.01$ and $K = 20$.

IV. EXPERIMENTS

Experiments are designed to answer the following questions: **Q1)** How does MANGO-Grasp compare with existing cross-embodiment grasp synthesis baselines? **Q2)** Does it generalize zero-shot to unseen robotic hands? **Q3)** Does morpho-kinematic pretraining improve over morphology-identity-only pretraining? **Q4)** How does the proposed Mahalanobis Field compare with Euclidean fields for contact representation? **Q5)** How much does the 3D Gaussian representation improve over a point-cloud alternative, and **Q6)** does it outperform mesh-fitted anisotropic primitives? **Q7)** How does MANGO-Grasp perform in the real world?

A. Experimental Setting

Implementation Details. We implement the model in PyTorch. With 29.5M parameters, it is trained for 150 epochs on an NVIDIA RTX 5090 GPU, requiring approximately 7.4 hours. We use Adam optimizer with an initial learning rate of 10^{-5} , a learning rate scheduler and a batch size of 24.

Datasets. Two datasets are used for benchmark in this work. 1) We adopt the filtered CMAP dataset [12] with DRO [15]’s train/test split, which is also consistent with TRO [16]. The training split contains 14,011 grasps over 48 objects and three morphologically distinct hands (ShadowHand, Allegro, and Barrett), and the test split contains 10 novel objects. 2) To assess object-level generalization, we further evaluate on a randomly selected 40-object subset of MultiGripper-Grasp [18] that spans broader geometries and scales while remaining graspable by all four evaluated hands.

Simulation Setup. We evaluate grasp stability in Isaac Gym [28] using the PhysX engine. Following the simulation protocol in prior works [12], [15], each hand is initialized at the predicted pose with the target object present. External forces of magnitude $0.5m_o g$ where m_o is the object mass, are applied sequentially to the object along six orthogonal directions. A grasp is deemed successful if hand-object contact is preserved after the full perturbation sequence.

Baselines. We compare with three representative baselines. *DexGraspNet* [2] is an optimization-based pipeline that optimizes differentiable force closure [1] and kinematic feasibility; since its official implementation supports only ShadowHand, we extend it to Allegro and Barrett with careful hand-specific hyperparameter tuning, and report the best-performing results among multiple tuning settings. *DRO* [15] is a learning-based method that predicts dense distances between robot keypoints and object point clouds, followed by configuration optimization. *TRO* [16] is a leading open-source learning-based method that diffuses SE(3) transformations over hand-link and object-patch graphs, followed by

TABLE I

SIMULATION SUCCESS RATES (%) ON UNSEEN-OBJECT TEST SPLITS ACROSS THREE SEEN HANDS AND ZERO-SHOT SHARPAWAVE TRANSFER.

Method	CMAP Test Split					MultiGripperGrasp Test Split				
	Seen Hands				Unseen Hand	Seen Hands				Unseen Hand
	ShadowHand	Allegro	Barrett	Avg.	SharpaWave	ShadowHand	Allegro	Barrett	Avg.	SharpaWave
DexGraspNet [2]	69.23 $\pm$ 0.39	62.77 $\pm$ 0.41	70.63 $\pm$ 0.31	67.54 $\pm$ 0.06	—*	62.48 $\pm$ 0.71	59.38 $\pm$ 0.83	65.72 $\pm$ 0.61	62.53 $\pm$ 0.36	—*
DRO [15]	82.70 $\pm$ 0.36	92.10 $\pm$ 0.70	88.53 $\pm$ 0.82	87.78 $\pm$ 0.30	70.70 $\pm$ 0.91	70.45 $\pm$ 0.79	76.87 $\pm$ 0.57	78.88 $\pm$ 0.34	75.40 $\pm$ 0.56	64.90 $\pm$ 0.83
TRO [16]	95.23 $\pm$ 0.31	94.50 $\pm$ 0.45	92.90 $\pm$ 0.71	94.21 $\pm$ 0.19	39.33 $\pm$ 0.95	75.65 $\pm$ 0.58	80.81 $\pm$ 1.02	87.22 $\pm$ 0.35	81.23 $\pm$ 0.62	24.98 $\pm$ 0.76
MANGO-Grasp	96.50$\pm$0.45	98.47$\pm$0.05	97.80$\pm$0.08	97.59$\pm$0.12	84.17$\pm$0.70	85.97$\pm$0.51	93.12$\pm$0.33	89.33$\pm$0.28	89.47$\pm$0.13	81.47$\pm$0.78

*DexGraspNet is optimization-based and does not involve learned embodiment transfer, so it is excluded from the zero-shot unseen-hand comparison.

TABLE II

ABLATION STUDIES ON PROPOSED COMPONENTS WITH SIMULATION SUCCESS RATES (%) FOR SEEN HANDS AND UNSEEN SHARPAWAVE.

Variant	CMAP Test Split					MultiGripperGrasp Test Split				
	Seen Hands				Unseen Hand	Seen Hands				Unseen Hand
	ShadowHand	Allegro	Barrett	Avg.	SharpaWave	ShadowHand	Allegro	Barrett	Avg.	SharpaWave
MANGO-Grasp (full)	96.50$\pm$0.45	98.47$\pm$0.05	97.80$\pm$0.08	97.59$\pm$0.12	84.17$\pm$0.70	85.97$\pm$0.51	93.12$\pm$0.33	89.33$\pm$0.28	89.47$\pm$0.13	81.47$\pm$0.78
w/o Kinematic Awareness	87.03 $\pm$ 0.42	91.90 $\pm$ 0.37	88.50 $\pm$ 0.14	89.14 $\pm$ 0.20	77.53 $\pm$ 0.92	80.88 $\pm$ 0.49	89.28 $\pm$ 0.22	85.69 $\pm$ 0.79	85.28 $\pm$ 0.15	74.50 $\pm$ 0.75
Euclidean Fields	79.57 $\pm$ 0.38	84.97 $\pm$ 0.65	82.87 $\pm$ 0.17	82.47 $\pm$ 0.10	65.53 $\pm$ 0.54	72.49 $\pm$ 0.39	80.02 $\pm$ 0.64	76.35 $\pm$ 0.48	76.29 $\pm$ 0.38	61.50 $\pm$ 0.58
Point-Cloud Rep.	69.27 $\pm$ 0.66	75.23 $\pm$ 0.47	70.93 $\pm$ 0.12	71.81 $\pm$ 0.25	54.10 $\pm$ 0.78	64.43 $\pm$ 0.53	68.63 $\pm$ 0.43	66.41 $\pm$ 0.50	66.49 $\pm$ 0.44	51.63 $\pm$ 0.62
Mesh-Fitted Primitives	81.80 $\pm$ 0.67	87.23 $\pm$ 0.61	83.43 $\pm$ 0.33	84.16 $\pm$ 0.15	74.07 $\pm$ 0.66	76.99 $\pm$ 0.90	85.77 $\pm$ 0.34	80.62 $\pm$ 0.44	81.13 $\pm$ 0.47	67.90 $\pm$ 0.80

inverse-kinematics-based joint realization. For fair comparison, we retrain *DRO* and *TRO* using their official implementations and report the reproduced results.

B. Main Results

To address **Q1**, we benchmark all methods on the two test splits in Sec. IV-A. For each method, we generate 100 candidate grasps for each test object and report the mean simulation success rate together with its standard deviation across three independent runs.

Table I shows that MANGO-Grasp achieves the highest seen-hand average on both unseen-object benchmarks, reaching 97.59% on CMAP and 89.47% on MultiGripperGrasp. Among the baselines, TRO achieves the best seen-hand average on both benchmarks. MANGO-Grasp further improves this average by 3.38 and 8.24 percentage points (pp) on CMAP and MultiGripperGrasp, respectively. The gains are consistent across ShadowHand, Allegro, and Barrett, indicating robust grasp synthesis across seen embodiments.

MultiGripperGrasp is more challenging due to broader object geometry and scale variation. From CMAP to MultiGripperGrasp, DRO and TRO drop by 12.38 and 12.98 pp, respectively, while MANGO-Grasp drops by only 8.12 pp. This smaller degradation suggests stronger object-level generalization. DexGraspNet yields more modest results than the learning-based methods despite extensive per-hand hyperparameter tuning, reflecting the sensitivity of optimization-based methods to embodiment-specific tuning.

C. Zero-Shot Generalization to Unseen Hand

To address **Q2**, we assess zero-shot embodiment-level generalization by testing the learning-based methods on the unseen SharpaWave hand with complex morphology and kinematics (5 fingers with 22 fully actuated joints). SharpaWave is excluded from pretraining and training. Since no contact prior is available for the unseen SharpaWave, we set $w_i^e = 1$ for all keypoints during grasp realization. As shown in Table I, MANGO-Grasp outperforms the strongest

zero-shot baseline DRO by 13.47 pp on CMAP and 16.57 pp on MultiGripperGrasp, respectively. TRO performs strongly on seen hands but degrades on SharpaWave, consistent with its reported limitation that zero-shot generalization is limited to hands with similar morphology rather than distinct unseen embodiments. This suggests that MANGO-Grasp learns a more transferable interaction representation which is robust to embodiment shifts.

D. Ablation Studies

To answer **Q3–Q6**, we ablate the three core components: kinematic-awareness supervision in robot pretraining, Mahalanobis fields and 3D Gaussians. Table II reports each variant’s results on the seen hands and unseen SharpaWave. **Effect of Kinematic-Awareness Encoding (Q3).** We remove the kinematic-awareness supervision from robot pretraining, leaving only the morphology-identity supervision. This reduces the seen-hand average by 8.45 and 4.19 pp on CMAP and MultiGripperGrasp, and decreases zero-shot SharpaWave performance by 6.64 and 6.97 pp. These results highlight the importance of encoding both morphology and kinematics in robot descriptors.

Mahalanobis Fields vs. Euclidean Fields (Q4). We replace the Mahalanobis prediction target with Euclidean Fields between robot keypoints and Gaussian primitive centers, and realize grasps from the Euclidean Fields. This reduces keypoint–primitive compatibility to isotropic proximity and ignores the directional weighting encoded in each primitive’s covariance. On seen hands, the average drops by 15.12 pp on CMAP and 13.18 pp on MultiGripperGrasp; unseen SharpaWave drops by 18.64 and 19.97 pp. These results show that Mahalanobis Fields provide a more effective surface-aware signal than Euclidean Fields.

3D Gaussians vs. Point Clouds (Q5). We replace the 3D Gaussians with point clouds. Unlike Gaussians, point clouds offer no explicit surface geometry encoding and lack adaptive density. They also carry no covariance, so the interaction fields have to reduce to Euclidean fields. This variant yields

seen-hand average drops of 25.78 and 22.98 pp on CMAP and MultiGripperGrasp, and zero-shot SharpaWave drops of 30.07 and 29.84 pp, respectively. The result reflects two roles of the 3D Gaussian: it describes object geometry more faithfully than discrete points, and its covariance gives the interaction field the surface-aware, anisotropic support that a Euclidean field over isolated points cannot reproduce.

3D Gaussians vs. Mesh-Fitted Primitives (Q6). We replace the 3D Gaussians with anisotropic primitives directly built from the mesh using curvature-adaptive sampling and quadric fitting. This variant yields 13.43 and 8.34 pp drops on seen hands for CMAP and MultiGripperGrasp, respectively, and 10.10 and 13.57 pp drops on zero-shot SharpaWave, confirming that the improvement is not merely due to oriented anisotropic surface primitives, but also to the proposed 3DGS-based construction.

E. Real-World Experiments

To answer **Q7**, we deploy our pipeline on a real KUKA-mounted SharpaWave hand under the same zero-shot setting as Sec. IV-C. Object poses are estimated using FoundationPose [29] with an Intel RealSense D435 camera. We evaluate 10 unseen objects with 10 grasp attempts per object, and report the per-object results in Table III. Our pipeline achieves a real-world success rate of **86%**, demonstrating effective sim-to-real transfer to an unseen hand morphology and novel objects. The remaining failures are mainly associated with residual object-pose estimation errors and open-loop execution of the predicted final hand configuration, where the finger-closing profile is not explicitly controlled. These effects are more pronounced for the apple, whose high-curvature geometry and low-friction 3D-printed surface reduce the contact margin, leaving grasps more prone to slip.

TABLE III

REAL-WORLD EXPERIMENT RESULTS WITH THE UNSEEN SHARPAWAVE HAND.

Object	Apple	Tomato Soup Can	Rugby Ball	Spam Can	Ketchup Bottle
Success	6/10	10/10	9/10	8/10	8/10
Object	Toothpaste Box	Chips Can	Coke Can	Salt Box	Scrub Sponge
Success	10/10	8/10	8/10	9/10	10/10

V. CONCLUSION

We presented MANGO-Grasp, an anisotropic interaction framework for cross-embodiment dexterous grasp synthesis. MANGO-Grasp represents objects as geometry-oriented 3D Gaussian primitives and encodes hand keypoints into pretrained morpho-kinematic descriptors. It models the interaction between robot keypoints and object primitives as Mahalanobis fields that capture surface tangent-normal anisotropy. The predicted fields guide grasp realization with one shared optimization setup across hands, without hand-specific retuning. Experiments demonstrate strong seen-hand performance, zero-shot transfer to an unseen SharpaWave hand, and effective real-world deployment. The current pipeline relies on mesh-based 3D Gaussian construction and does open-loop grasp execution. Future work will explore mesh-free object representations and closed-loop execution.

REFERENCES

- [1] T. Liu, Z. Liu, Z. Jiao, Y. Zhu, and S.-C. Zhu, "Synthesizing diverse and physically stable grasps with arbitrary hand structures using differentiable force closure estimator," *IEEE Robot. Autom. Lett.*, vol. 7, no. 1, pp. 470–477, 2021.
- [2] R. Wang, J. Zhang, J. Chen, Y. Xu, P. Li, T. Liu, and H. Wang, "Dexgraspnet: A large-scale robotic dexterous grasp dataset for general objects based on simulation," *arXiv preprint arXiv:2210.02697*, 2022.
- [3] D. Turpin, T. Zhong, S. Zhang, G. Zhu, E. Heiden, M. Macklin, S. Tsogkas, S. Dickinson, and A. Garg, "Fast-grasp'd: Dexterous multi-finger grasp generation through differentiable simulation," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2023, pp. 8082–8089.
- [4] G.-H. Xu, Y.-L. Wei, D. Zheng, X.-M. Wu, and W.-S. Zheng, "Dexterous grasp transformer," in *CVPR*, 2024.
- [5] Z. Weng, H. Lu, D. Kragic, and J. Lundell, "Dexdiffuser: Generating dexterous grasps with diffusion models," 2024. [Online]. Available: <https://arxiv.org/abs/2402.02989>
- [6] Y. Zhong, Q. Jiang, J. Yu, and Y. Ma, "Dexgrasp anything: Towards universal robotic dexterous grasping with physics awareness," 2025. [Online]. Available: <https://arxiv.org/abs/2503.08257>
- [7] Y. Xu, W. Wan, J. Zhang, H. Liu, Z. Shan, H. Shen, R. Wang, H. Geng, Y. Weng, J. Chen, *et al.*, "Unidexgrasp: Universal robotic dexterous grasping via learning diverse proposal generation and goal-conditioned policy," in *CVPR*, 2023.
- [8] W. Wan, H. Geng, Y. Liu, Z. Shan, Y. Yang, L. Yi, and H. Wang, "Unidexgrasp++: Improving dexterous grasping policy learning via geometry-aware curriculum and iterative generalist-specialist learning," *arXiv preprint arXiv:2304.00464*, 2023.
- [9] W. Wang, F. Wei, L. Zhou, X. Chen, L. Luo, X. Yi, Y. Zhang, Y. Liang, C. Xu, Y. Lu, *et al.*, "Unigraspttransformer: Simplified policy distillation for scalable dexterous robotic grasping," in *CVPR*, 2025.
- [10] H. Zhang, K. Y. Ma, M. Z. Shou, W. Lin, and Y. Wu, "Machagrasp: Morphology-aware cross-embodiment dexterous hand articulation generation for grasping," 2026. [Online]. Available: <https://arxiv.org/abs/2510.06068>
- [11] L. Shao, F. Ferreira, M. Jorda, V. Nambiar, J. Luo, E. Solowjow, J. A. Ojea, O. Khatib, and J. Bohg, "Unigrasp: Learning a unified model to grasp with multifingered robotic hands," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 2286–2293, 2020.
- [12] P. Li, T. Liu, Y. Li, Y. Geng, Y. Zhu, Y. Yang, and S. Huang, "Gendexgrasp: Generalizable dexterous grasping," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*. IEEE, 2023, pp. 8068–8074.
- [13] M. Attarian, M. A. Asif, J. Liu, R. Hari, A. Garg, I. Gilitschenski, and J. Tompson, "Geometry matching for multi-embodiment grasping," in *CoRL*, 2023.
- [14] Y. Wei, M. Attarian, and I. Gilitschenski, "Geomatch++: Morphology conditioned geometry matching for multi-embodiment grasping," *arXiv preprint arXiv:2412.18998*, 2024.
- [15] Z. Wei, Z. Xu, J. Guo, Y. Hou, C. Gao, Z. Cai, J. Luo, and L. Shao, " $\mathcal{D}(\mathcal{R}, \mathcal{O})$ grasp: A unified representation of robot and object interaction for cross-embodiment dexterous grasping," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2025, pp. 4982–4988.
- [16] X. Fei, Z. Xu, H. Fang, T. Zhang, and L. Shao, "T(r,o) grasp: Efficient graph diffusion of robot-object spatial transformation for cross-embodiment dexterous grasping," 2025. [Online]. Available: <https://arxiv.org/abs/2510.12724>
- [17] B. Kerbl, G. Kopanas, T. Leimkühler, G. Drettakis, *et al.*, "3d gaussian splatting for real-time radiance field rendering," *ACM Trans. Graph.*, vol. 42, no. 4, pp. 139–1, 2023.
- [18] L. F. Casas, N. Khargonkar, B. Prabhakaran, and Y. Xiang, "Multi-grippergrasp: A dataset for robotic grasping from parallel jaw grippers to dexterous hands," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, 2024.
- [19] Z. Wu, R. A. Potamias, X. Zhang, Z. Zhang, J. Deng, and S. Luo, "Cedex: Cross-embodiment dexterous grasp generation at scale from human-like contact representations," 2025. [Online]. Available: <https://arxiv.org/abs/2509.24661>
- [20] Z. Wu, X. Zhang, Z. Chen, J. Deng, R. A. Potamias, and S. Luo, "Unimorphgrasp: Diffusion model with morphology-awareness for cross-embodiment dexterous grasp generation," 2026. [Online]. Available: <https://arxiv.org/abs/2602.00915>
- [21] H. Yuan, B. Zhou, Y. Fu, and Z. Lu, "Cross-embodiment dexterous grasping with reinforcement learning," in *ICLR*, 2025.

- [22] A. Guédon and V. Lepetit, “Sugar: Surface-aligned gaussian splatting for efficient 3d mesh reconstruction and high-quality mesh rendering,” 2023. [Online]. Available: <https://arxiv.org/abs/2311.12775>
- [23] B. Huang, Z. Yu, A. Chen, A. Geiger, and S. Gao, “2d gaussian splatting for geometrically accurate radiance fields,” in *Proc. ACM SIGGRAPH Conf. Papers*, 2024, pp. 1–11.
- [24] M. Tancik *et al.*, “Nerfstudio: A modular framework for neural radiance field development,” in *Proc. ACM SIGGRAPH Conf.*, 2023.
- [25] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, “Dynamic graph cnn for learning on point clouds,” *ACM Trans. Graph.*, vol. 38, no. 5, pp. 1–12, 2019.
- [26] P. MAHALANOBIS, “On the generalised distance in statistics,” in *Proc. Nat. Inst. Sci.(India)*, vol. 12, 1936, pp. 49–55.
- [27] K. Sohn, H. Lee, and X. Yan, “Learning structured output representation using deep conditional generative models,” *Adv. Neural Inf. Process. Syst.*, vol. 28, 2015.
- [28] J. Liang, V. Makovychuk, A. Handa, N. Chentanez, M. Macklin, and D. Fox, “Gpu-accelerated robotic simulation for distributed reinforcement learning,” in *CoRL*, 2018.
- [29] B. Wen, W. Yang, J. Kautz, and S. Birchfield, “FoundationPose: Unified 6d pose estimation and tracking of novel objects,” in *CVPR*, 2024.