Towards a Multi-Embodied Grasping Agent

Roman Freiberg, Alexander Qualmann, Ngo Anh Vien, and Gerhard Neumann

***Abstract*—Multi-embodiment grasping focuses on developing approaches that exhibit generalist behavior across diverse gripper designs. Existing methods often learn the kinematic structure of the robot implicitly and face challenges due to the difficulty of sourcing the required large-scale data. In this work, we present a data-efficient, flow-based, equivariant grasp synthesis architecture that can handle different gripper types with variable degrees of freedom and successfully exploit the underlying kinematic model, deducing all necessary information solely from the gripper and scene geometry. Unlike previous equivariant grasping methods, we translated all modules from the ground up to JAX and provide a model with batching capabilities over scenes, grippers, and grasps, resulting in smoother learning, improved performance and faster inference time. Our dataset encompasses grippers ranging from humanoid hands to parallel yaw grippers and includes 25,000 scenes and 20 million grasps.**

I. Introduction

Multi-embodiment grasping methods employ a single model to synthesize grasps for diverse gripper architectures with varying degrees of freedom (DoFs). Since these methods are not tied to one physical system, they adapt more effectively to a broad range of end effectors. To succeed in this setting, approaches must be robust to hyperparameter choices and hidden biases, enabling the integration of data from multiple sources [2], [3]. Recent work has combined diffusion models [4], [5] with architectures that leverage data symmetries through equivariance [6]–[9] in robotics. These techniques excel at capturing the multimodal nature of grasping data and currently define the state of the art in robotic grasping. In this paper, we extend these ideas to the multi-embodiment setting and introduce a single method capable of synthesizing grasps for grippers with variable DoFs. Our contributions are threefold. We first introduce an equivariant flow-based method that can synthesize grasps across a wide range of gripper types, including humanoid hands such as Allegro and ShadowHand, as well as parallel jaw grippers such as Panda and VX300, encompassing both rotary and prismatic joint types. We provide a complete implementation in JAX [10], which to our knowledge enables, for the first time in this setting, training and inference across multiple full-scene point clouds, grippers, and grasps in a single update step as a single batch. This capability results in smoother learning for both single-GPU and multi-GPU schemes, significantly faster inference, and lower memory consumption compared to prior approaches. In addition, we release the datasets containing grasps with joint information for five gripper types, together with their JAX-integrated kinematic models, spanning more than 25k scenes. We further provide the data-generation scripts that support both single-object and full-scene grasp generation, facilitating dataset expansion. Finally, we demonstrate that our approach achieves performance on part with the state of the art in both single- and multi-gripper grasp generation settings. Code and dataset are available as open source.[1]

R. Freiberg, A. Qualmann, and N. A. Vien are with Bosch Corporate Research, Renningen, Germany (e-mail of corresponding author: roman.freiberg@bosch.com. G. Neumann is with the Karlsruhe Institute of Technology, Karlsruhe, Germany.

II. Related Work

Grasp detection approaches synthesize grasp poses and configurations for objects and are typically categorized into offline methods, which generate grasps once, and online approaches, which refine grasps continuously during execution. Most work focuses on parallel-jaw grippers [6], [11]–[13], usually tied to a single gripper architecture and using large-scale grasp datasets such as ACRONYM [14] and GraspNet-1Billion [15].

Dexterous grasping datasets and algorithms target high-DoF gripper architectures. DexGraspNet 2.0 [16] learns a diffusion model to predict pre-grasp poses with joint configurations for the Leap gripper in cluttered scenes, and Fast-Grasp'D [17] generates multi-finger grasps via differentiable simulation. Dexterous grasping agents employ policies dedicated to high-DoF grippers such as the Leap and ShadowHand; notable examples include UniDexGrasp++ [18], Unified Generative Grasping [19], and AffordDexGrasp [20]. Furthermore, DexGraspVLA [21] demonstrates effective dexterous grasping using pretrained vision and language models, and DexGrasp Anything [22] uses a physics-informed diffusion process to generate dexterous grasps. MultiGripperGrasp [23], BoDex [24], and Get-a-Grip [25] contribute scalable training and evaluation resources for diverse hands, while GenDexGrasp [2] demonstrates object-centric features that transfer across grippers.

Cross-embodiment grasp representations define, for each gripper, a contact intent and treat inverse kinematics as a post-processing task. Prominent examples include Geometric Matching [3], which learns contact maps for each gripper type; $D(R, O)$ Grasp [26], which relates gripper and object point clouds, thereby defining contacts; and RobotFingerPrint [27], which enables transfer between hands through a unified gripper representation, as well as CEDex [28], which uses human-informed contacts to condition a variational autoencoder for diverse grippers. Furthermore, methods that learn gripper-specific encodings [1], [29]–[31] predict stable grasps by incorporating the gripper geometry. However, these approaches do not directly learn all DoFs of the gripper and either rely on fixed, predefined configurations [1], [29], [31] or reduce the space through eigengrasps [30]. MADRL [32] applies reinforcement learning to grasping across multiple gripper types but requires specific masking to adapt the model to variable DoFs.

[1]Code and dataset will be made publicly available soon

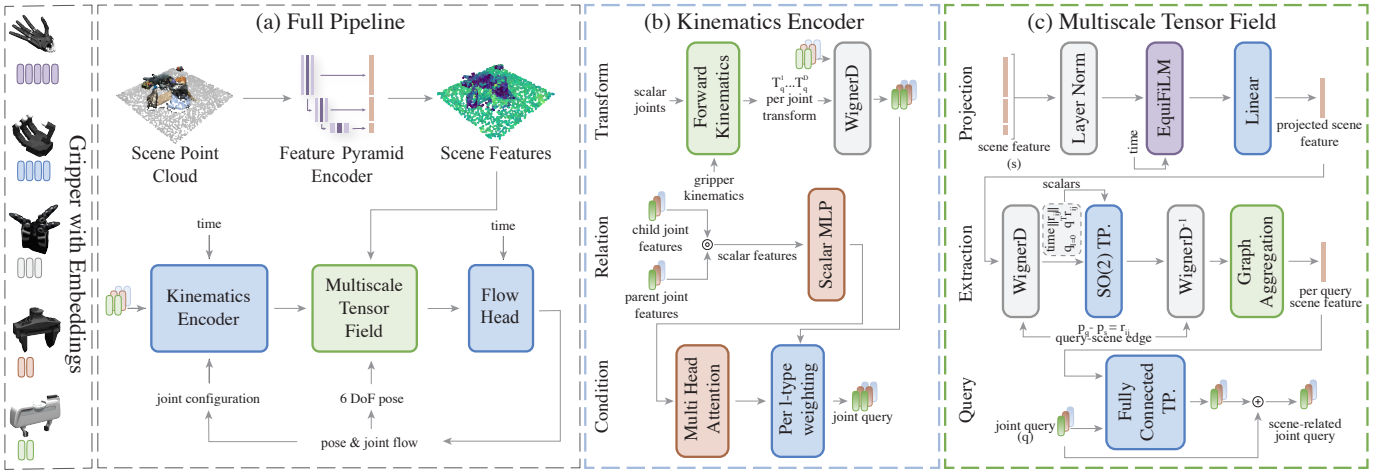


Fig. 1. **Method Overview**. (**Left**) Grippers are represented with per-joint equivariant embeddings. (**a**) **Full Pipeline.** A scene point cloud is encoded into a multi-scale equivariant feature pyramid. Time-conditioned joint features query this pyramid to extract pose and joint information. These scene-aware queries are then decoded to predict flow gradients, which generate the final pre-grasp configuration. (**b**) **Kinematics Encoder.** Joint values and kinematics are used to compute per-joint transformations, which are applied to the embeddings via Wigner-D matrices. Parent-child features interact through a dot product, conditioning the queries with per-ℓ-type weights. (**c**) **Multiscale Tensor Field.** Hierarchical scene features are time-conditioned using an equivariant FiLM layer [1] and projected to a lower dimension. Relative scene-query positions are encoded via a tensor product dependent on direction and length. The resulting aggregated features for each query are fused with the original joint query via a fully connected tensor product.

Diffusion- and flow-based generative models learn a transport between an initial uninformative distribution and the target distribution approximated from samples. Work such as [33] learns full robot trajectories, while work in grasping typically focuses on target end-effector grasp poses on the $\mathrm{SE}(3)$ manifold [1], [4], [5], [34]–[37].

A surge of large-scale manipulation-skill datasets targets the multi-embodiment problem. RoboMind [38], DROID [39], and AgiBot World Colosseo [40] collect large-scale trajectories for arms and humanoids. All of these datasets rely on heavy task standardization and are incompatible with each other. RoboArena [41] addresses this gap with a distributed real-robot benchmark but, at the time of writing, lacks broad acceptance in the community.

Generalist agents leverage large-scale data and foundation models to produce policies that cover diverse tasks [42]–[46]. These methods often struggle with an agent–action representation appropriate for multi-embodiment settings. The work of [47] first attempts to learn a world model from images and extract a multi-embodiment policy via an embodiment-agnostic optical flow that is fine-tuned for a concrete embodiment. While these agents demonstrate impressive results, they often treat grasping as a downstream task.

Locomotion research mirrors the multi-embodiment trend. Work such as One Policy to Run Them All [48], Multi-Loco [49], and Towards Embodiment Scaling Laws in Robot Locomotion [50] demonstrates that increasing morphology diversity improves zero-shot transfer to unknown morphologies.

Zero-shot domain transfer methods bridge different data distributions through alignment of source and target domains, as presented in CycleGAN [51]. Notable examples in robotics include Mirage [52] and Shadow [53], which adapt between domains but are limited to a single physical gripper model, introducing additional biases.

Exploiting symmetries in data can improve robustness, data efficiency, and generalization. Work by Cohen and Welling [54] introduced group-equivariant CNNs for images. Subsequent methods extended these ideas to continuous groups [55]–[57], including the Euclidean $E(N)$ group, which has proven effective in robotics [5]–[7], [37] and imitation learning [8], [9], [34]. Often, equivariant layers are integrated into graph network architectures [5], [58], [59], producing features that capture local and global structure in robotic scenes.

Finally, modern depth sensors and foundation stereo models reduce hardware assumptions for point-cloud pipelines. Stereo-matching methods such as FoundationStereo [60] and the scalable S^2M^2 model [61] provide reliable depth estimation, reinforcing the feasibility of pure point-cloud models.

III. Preliminaries

A. Flows for Distribution Modeling in Robotics

Continuous Normalizing Flows (CNFs) provide a framework for learning complex probability distributions on smooth manifolds. For robotics applications, we are often interested in the manifold $\mathcal{M} = \mathrm{SE}(3) \times \mathbb{R}^D$. A state is given by $T = (R, \mathbf{p}, \mathbf{q})$ where, $R \in \mathrm{SO}(3)$ represents an orientation, $\mathbf{p} \in \mathbb{R}^3$ a position, and $\mathbf{q} \in \mathbb{R}^D$ a general configuration vector.

CNFs learn to transport samples from a simple prior distribution—typically uniform on compact manifolds like $\mathrm{SO}(3)$ and normal on open manifolds like $\mathbb{R}^3$ and $\mathbb{R}^D$—to a complex target data distribution $p(T)$. This transport is governed by time-dependent vector fields parameterized by a network with learnable weights θ

$$\begin{aligned}\boldsymbol{\omega}_\theta &: [0,1] \times \mathcal{M} \to \mathbb{R}^3,\\ \mathbf{v}_\theta &: [0,1] \times \mathcal{M} \to \mathbb{R}^3,\\ \dot{\mathbf{q}}_\theta &: [0,1] \times \mathcal{M} \to \mathbb{R}^D.\end{aligned}$$

These vector fields define velocities, with their outputs residing in the respective tangent spaces $T_R\mathrm{SO}(3)$, $T_{\mathbf{p}}\mathbb{R}^3$, and $T_{\mathbf{q}}\mathbb{R}^D$.

The canonical isomorphism for the tangent space of $\mathrm{SO}(3)$ is defined via a skew-symmetric matrix basis, as in [37]. Within the Flow Matching framework, these velocity fields are learned by minimizing the mean-squared error between the predicted flows and ground-truth flows constructed from pairs of samples drawn from the prior and target distributions.

B. Equivariant Representations

A function $f : X \to Y$ is equivariant if transforming its input is equivalent to transforming its output; formally, $f(D_X(r)x) = D_Y(r)f(x)$ for any group action $r \in \mathrm{SE}(3)$ with corresponding representations D_X and D_Y. The network layers in this work are designed to be $\mathrm{SO}(3)$-equivariant and translation-invariant. Features are represented as a direct sum of irreducible representations (irreps) of $\mathrm{SO}(3)$, indexed by type $\ell \in \mathbb{N}_0$: $\mathbf{f} = \bigoplus_{\ell=0}^{\ell_{\max}} \mathbf{f}^\ell$. Type $\ell = 0$ features are scalars, $\ell = 1$ are vectors, and higher types capture finer angular detail. A rotation $R \in \mathrm{SO}(3)$ acts on these features via the Wigner-D matrices, $D^\ell(R)$. The primary mechanism for equivariant interaction is the tensor product denoted as $(\otimes)$. For two features $\mathbf{x}^{\ell_1}$ and $\mathbf{y}^{\ell_2}$, the output of type ℓ is computed via Clebsch-Gordan coefficients (Wigner 3j-symbols)

$$(\mathbf{x} \otimes \mathbf{y})_{\ell m} = \sum_{m_1, m_2} \begin{pmatrix} \ell_1 & \ell_2 & \ell \\ m_1 & m_2 & m \end{pmatrix} x_{\ell_1 m_1} y_{\ell_2 m_2}. \quad (1)$$

We leverage two variants of the tensor product. The *fully connected tensor product* (FCTP) introduces a learnable weight for each valid path in Equation 1, enabling it to capture rich rotational correlations between features. When feature interactions are conditioned on a specific direction, we can use a more efficient $\mathrm{SO}(2)$*-equivariant tensor product*. This is achieved by rotating the local coordinate system to align with the direction vector, which simplifies the CG coefficients and renders the operation sparse [62]. This technique significantly reduces computational and memory costs and is empirically well-suited for encoding directional information. These operations are foundational to modern architectures like EquiformerV2 [63], which we utilize in our network. Geometric and other scalar information is incorporated by modulating the feature vectors. We define a directional modulation operation based on a relative vector $\mathbf{r}_{ji} = \mathbf{r}_j - \mathbf{r}_i$ and a set of arbitrary scalars s

$$\mathrm{dir\_mod}(\hat{\mathbf{r}}_{ji}, s) \;:=\; \bigoplus_{\ell=0}^{\ell_{\max}} \mathrm{MLP}_\ell(\mathrm{enc}(s))\; Y^\ell(\hat{\mathbf{r}}_{ji}), \quad (2)$$

where Y^ℓ are the real spherical harmonics. The function $\mathrm{enc}(\cdot)$ encodes the input scalars (e.g., distance, time) using suitable bases, and a multi-layer perceptron (MLP) then produces a set of weights that modulate the different irrep types ℓ. This is analogous to the gating mechanism in Allegro [59]. This modulation is critical to our approach, as it provides the mechanism to condition high-degree equivariant features on one-dimensional inputs such as joint angles.

IV. Problem Formulation

We consider an arbitrary gripper g with an internal joint configuration $\mathbf{q}_g \in \mathbb{R}^{D_g}$, where D_g denotes its number of degrees of freedom (DoFs), such as $D_g = 2$ for a parallel-jaw gripper or $D_g = 22$ for a dexterous hand. The gripper is modeled as a three-dimensional manifold $\mathcal{M}_g$ with known geometry. Its forward kinematics are described by the function

$$\mathrm{kin}_g \colon \mathbb{R}^{D_g} \times \{1, \ldots, D_g\} \to \mathrm{SE}(3),$$

which maps a joint configuration $(\mathbf{q}_g, i)$ to the transform $T_i = (R_i, \mathbf{p}_i)$ of the i-th joint relative to the gripper's base frame. A scene is composed of graspable objects and potential obstacles, observed as a point cloud $\mathbf{o}_s$ in a global coordinate frame. The objective is to learn the conditional distribution defined through

$$p(T_g, \mathbf{q}_g \mid \mathcal{M}_g, \mathbf{o}_s),$$

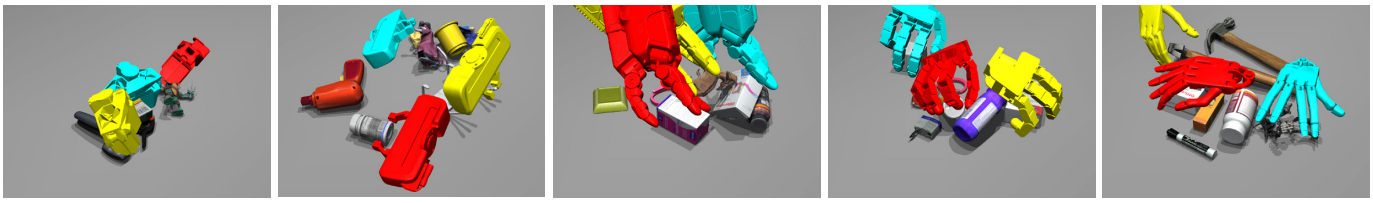
where the model leverages the known kinematics kin_g. Here, $T_g \in \mathrm{SE}(3)$ represents the desired pre-grasp pose of the gripper in the scene frame, and $\mathbf{q}_g \in \mathbb{R}^{D_g}$ specifies the corresponding pre-grasp joint configuration. A grasp, defined by the pair $(T_g, \mathbf{q}_g)$, is considered successful if, after positioning the gripper at this pre-grasp state, a subsequent pre-defined closing motion results in a stable configuration that ensures secure manipulation of an object.

V. Method

Our pipeline begins with a geometric scene encoder that converts the raw point-cloud observations into a feature map that represents the geometry. For each process time step, the kinematics encoder encodes the gripper configuration only through the transform information of the forward kinematics defining the joint queries. In the Multiscale Tensor Field the joint query extract grasp relevant scene information and relate the relative pose of the gripper and scene. In the Flow head, pose and joint gradients are decoded for the flow process. Figure 1 provides an overview of our pipeline. Our method is inspired by architectures such as [5], [6], [59] but is rebuilt from the ground up in the JAX [10] framework, with parallelization and memory efficiency as the main decision drivers. The resulting architecture allows batching over all grippers used, multiple scenes, and grasps in both single- and multi-GPU training schemes.

A. Geometric Scene Encoder

The scene geometry is represented by a point cloud $\{\mathbf{r}_i\}_{i=1}^N$, typically obtained from a dense RGB-D scan. Our method relies exclusively on geometric information from the point cloud. This design choice is motivated by the goal of robust sim-to-real transfer, as it circumvents the significant challenges of photorealistic rendering and domain randomization required to align simulated RGB data with real-world conditions. While color could be included as a straightforward scalar feature, our geometry-only approach ensures greater robustness out of the box. The raw point cloud is first pre-processed via voxel down-sampling (1 mm grid size) and then mapped to a fixed cardinality of 15,000 points using the Farthest-Point Sampling (FPS) algorithm. This fixed-size representation is necessary to accommodate JAX's [10] requirement for statically-sized arrays.



(a) VX300s (b) Franka (c) DEX-EE (d) Allegro (e) Shadow

Fig. 2. **Multi-Embodiment Grasp Synthesis Examples.** Renderings of three sampled pre-grasp configurations for five distinct grippers in cluttered scenes. Included grippers **(a)** ViperX 300s parallel gripper, **(b)** Franka Emika parallel gripper, **(c)** DEX-EE dexterous hand, **(d)** Allegro Hand, and **(e)** Shadow Hand.

Our encoder constructs a hierarchical representation of the scene through a feature pyramid composed of L stages. At the initial stage, we compute an equivariant feature embedding $\mathbf{f}_i^{(0)}$ for each point $\mathbf{r}_i$ by aggregating information from its local neighborhood $\mathcal{N}(i)$ as a weighted sum

$$\mathbf{f}_i^{(0)} = \sum_{j \in \mathcal{N}(i)} w_{ij} \ \text{dir\_enc}\big(\hat{\mathbf{r}}_{ji}, r_{ji}\big).$$

Here, dir_enc encodes the relative direction $\hat{\mathbf{r}}_{ji}$ and distance r_{ji} from a neighbor point $\mathbf{r}_j$ to the center point $\mathbf{r}_i$. The weights w_{ij} are distance-based attention scores, learned via the expression $w_{ij} = \text{softmax}_j(\text{MLP}(r_{ji}))$.

Each subsequent pooling stage $k \in \{1, \dots, L\}$ downsamples the feature point cloud $\{(\mathbf{r}_i^{(k-1)}, \mathbf{f}_i^{(k-1)})\}$ using FPS to produce a sparser set of points $\{\mathbf{r}_i^{(k)}\}$. The features for these new points are computed by aggregating features from the previous, denser layer using the sparse directional SO(2) tensor product operation in the EquiformerV2 modules. This process creates a multi-scale geometric representation $\{(\mathbf{r}_i^{(k)}, \mathbf{f}_i^{(k)})\}_{k=0..L}$. To expand the receptive field at each level, we apply multiple equivariant message-passing layers, where the neighborhood graphs are constructed using the K-Nearest Neighbors (K-NN) algorithm to ensure a fixed number of neighbors per point. This hierarchical structure enables efficient, point-wise feature querying by concatenating features across all stages for a given spatial location. The design is inspired by the Sonata model [64], which addresses the geometric shortcut problem present in U-Net-style architectures. By eliminating the decoder path, our approach significantly reduces memory utilization compared to similar equivariant models [5], [9], while preserving a rich, multi-scale geometric representation.

B. Geometric Gripper Encoding

The gripper encoder's objective is to produce a set of configuration-aware, equivariant query features that represent the gripper's state. A naive encoding of the scalar joint values $\mathbf{q}_g$ would force the model to learn the gripper's kinematic structure implicitly, hindering generalization. To avoid this, we adopt a more explicit, geometry-aware approach.

For each joint i, we introduce a unique, learnable, equivariant feature embedding, $\mathbf{z}_i$. Each embedding is composed of multiple irreducible representations (irreps), e.g., $\ell_0 \oplus \ell_1 \oplus \ell_2$. At a minimum, we use scalar ($\ell = 0$) and vector ($\ell = 1$) components. The gripper's forward kinematics, $\text{kin}_g(\mathbf{q}_g, i) = T_i = (R_i, \mathbf{p}_i)$, are used to place and orient these features

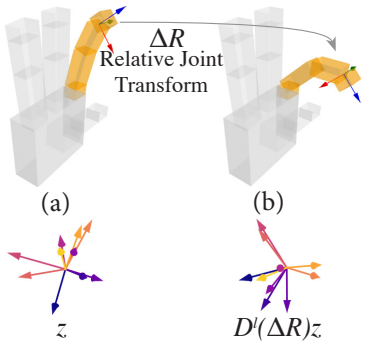


Fig. 3. **Equivariant Gripper Embeddings.** An initial gripper configuration **(a)** is represented by a learned feature embedding z. After a physical joint rotation ΔR, the gripper is in a new configuration **(b)**. Our method ensures the feature is correspondingly transformed via the Wigner-D matrix, $z' = \mathbf{D}^\ell(\Delta R)z$, keeping the representation consistent with the physical state.

in the gripper's base frame. Specifically, for each joint, the rotation R_i is applied to its corresponding embedding $\mathbf{z}_i$ via the Wigner-D matrices

$$\mathbf{z}_i' = \bigoplus_\ell D^\ell(R_i)\mathbf{z}_i^\ell.$$

This fundamental transformation, which ensures the feature representation covaries with the physical state, is visualized in Figure 3. This operation aligns the feature embedding with the joint's local orientation. To recover the scalar features of a revolute joint, we leverage the kinematic chain. We compute a channel-wise dot product between the rotated child embedding $\mathbf{z}_i'$ and its parent's rotated embedding, producing channel-wise scalars that are a function of the relative rotation between features. These scalars are then used in a lightweight attention mechanism to modulate the channels of $\mathbf{z}_i'$ with the dir_mod operator, thereby conditioning it on the whole gripper configuration. This scalar mechanism only requires the relative transform between parent and child, making it robust to the choice of coordinate systems for each gripper. For joints at the root of a kinematic chain, a static auxiliary embedding is used as an anchor for this computation, which is discarded afterwards. To incorporate information from prismatic joints, we apply an equivariant message-passing layer over the kinematic graph. The messages utilize the dir_mod operation on the relative positions $(\mathbf{p}_i - \mathbf{p}_j)$, which encodes the translational component of the joints. Finally, to represent the gripper's 6-DoF global pose, we introduce two additional learnable embeddings anchored at the gripper's origin. In the decoding phase, the $\ell = 1$ components of these two embeddings are used

to predict the rotational and translational velocity fields (the twist). Crucially, this representation circumvents the challenge of canonical coordinate frame selection, a problem present in prior works such as DiffusionEDFs [5] and [1]. Consequently, the local frames of different grippers are not required to be manually pre-aligned; their kinematic origins are sufficient. The learnable nature of the embeddings allows the model to implicitly establish a consistent internal frame and modulate the various frequency components of the output rotational flow. This process yields a total of $D_g + 2$ query features, each with an associated position, that explicitly encode the gripper's kinematic structure and configuration. This flexibility makes the equivariant design not just a feature for improving data efficiency, but a fundamental requirement for our multi-embodiment approach.

C. Gripper-Scene Relation

The Multiscale Tensor Field module is responsible for relating the gripper query features to the encoded scene geometry. This step provides the necessary context for predicting the gripper's pose relative to the scene and for adapting the finger placement. First, the set of $D_g + 2$ gripper query features, along with their associated positions, are transformed from the gripper's local frame into the global scene frame based on the current SE(3) pose estimate in the flow process.

We then construct a bipartite graph to establish neighborhoods. For each query, we identify its K-nearest neighbors among the scene points at the highest-resolution stage of the feature pyramid. Leveraging the many-to-one mapping established during the encoding phase, we then retrieve the corresponding features for these neighbors from all levels of the hierarchy. These hierarchical scene features are first projected to a lower-dimensional latent space and then conditioned on the flow-time t using an equivariant FiLM layer [1].

For each edge connecting a query to a scene, we apply the edge-direction-dependent $\mathrm{SO}(2)$ tensor product [62] on the scene feature. This operation is conditioned on a set of scalars derived from the flow-time, the relative distance, and the dot products of the query and scene features. The resulting edge features from a query's neighborhood are then aggregated into a single, scene-context feature using an attention mechanism. This aggregated scene feature is then fused with the original gripper query feature using a final FCTP. Note, the expensive FCTP operations are only applied per query thus limiting the computational and memory impact. This entire query-relation block can be stacked multiple times to build a more detailed representation. Finally, the scene-aware query features are transformed back into the gripper's local frame for decoding.

D. Flow Decoding

After the gripper queries have been enriched with scene context, we apply several additional equivariant message-passing layers over the gripper's kinematic graph. The final flow vector is then decoded by extracting specific components from the feature embeddings: the $\ell = 1$ components of the two pose tokens yield the rotational velocity $\boldsymbol{\omega}$ and the translational velocity $\mathbf{v}$, while the $\ell = 0$ components of the D_g joint tokens yield the joint-space velocities $\dot{\mathbf{q}}$. The model is trained by minimizing a weighted mean squared error between the predicted flow $(\boldsymbol{\omega}, \mathbf{v}, \dot{\mathbf{q}})$ and its ground-truth counterpart, following the flow-matching framework of Lim et al. [37].

VI. Experimental Results

TABLE I
Parallel Jaw Grasp success rate comparison

Method	Grasp Success Rate (%)
OrbitGrasp	**96.7**
Ours	94.8

Grasp success rate on the OrbitGrasp [6] benchmark. Models are trained on 2 500 scenes. While only competitive, our method only uses successful grasps during training, operates on full scenes and does not require scene normals.

We compare our method with the state-of-the-art grasping method OrbitGrasp [6] and in a multi-embodiment grasping settings [2]. It is worth noting that, compared with the previously mentioned methods, ours does not prioritize grasp success rate but rather reproducing the distribution of successful grasps. This distinction provides flexibility in adapting our method to the desired downstream task with minimal modifications.

A. Single Embodiment Parallel Jaw Grasping

First, we evaluate the feasibility of our approach on the benchmark presented in OrbitGrasp [6] for SE(3) grasp pose prediction of the Panda gripper in cluttered object scenes. For a direct comparison with OrbitGrasp [6], we reproduced its original dataset, which comprises 2,500 cluttered 'pile' scenes containing approximately 2.5 million successful and 3.5 million unsuccessful grasps. Adhering to the original evaluation protocol, we assess the grasp success rate in a simulated de-cluttering procedure across 100 scenes. The results are presented in Table I [3]. For the OrbitGrasp model, we select the variant with irreps up to $\ell = 2$, as this aligns with our model's architecture and has near negligible performance differences to significantly more computationally expensive versions [6].

The results show that while OrbitGrasp nearly saturates the success rate metric, our method remains competitive. Notably, our model achieves this performance despite being trained exclusively on successful grasps—utilizing less than half of the available grasp data—and without requiring scene normal or color information as input. Furthermore, there are two key methodological differences. First, our approach processes the full, high-resolution scene point cloud (15,000 points) at once, in contrast to OrbitGrasp which operates on multiple low-resolution scene patches of fewer than 1,000 points each. Second, the OrbitGrasp architecture is constrained to predicting per-point grasp scores along the object's surface normal. This inherent design choice limits its applicability to more general datasets, such as ours, where grasp poses can be freely positioned within the scene.

[2]The results presented for our method are preliminary, as they were obtained from models that had not yet fully converged (both single and multi-embodiment models), and are therefore subject to change.

[3]Preliminary results of our model. The OrbitGrasp result is from the original publication [6]. Final evaluation results are subject to change

TABLE II
DATASET COMPOSITION BY GRIPPER TYPE

Gripper	Total Grasps
Franka	6,870,351
VX300s	6,238,898
DEX-EE	1,639,770
Allegro	2,116,060
Shadow	2,990,167

Available grasps per gripper over all scenes. For each gripper we generate 5 000 scenes and re-evaluate all stable gravity free grasps per object in scene.

B. Dataset Generation

Our evaluation requires a dataset composed of scenes containing grasps with explicit joint configurations for each gripper type. The data generation pipeline from MultiDiffusion [1] is unsuitable for our purposes, as it only provides grasps in fixed open and closed configurations and employs the Antipodal [15] method, which is not designed for generating diverse, high-DoF grasps. For our dataset, we follow a two-stage process. First, we generate stable grasps for each gripper-object pair in a gravity-free environment. Second, these candidate grasps are transferred to cluttered scenes, where they are filtered for collisions and their stability is re-evaluated via a lifting motion, similar to the protocol in [1].

For the Panda and VX300s parallel-jaw grippers, we employ the Antipodal [15] grasp generation method to collect 5,000 grasps per object. For the dexterous grippers (DEX-EE, Allegro, and Shadow Hand), we leverage a differentiable kinematics model implemented in JAX [10], identical to the one used in our main method. We define a set of potential contact points on each gripper's fingertips. For each object, we first sample 10 cm grasping regions by applying FPS to the object's surface points. Within each region, we randomly select target contact points on the object and corresponding target points on the gripper's fingers; object contact points are slightly offset from the surface to mitigate initial collisions. We then solve for the corresponding joint configurations via gradient descent by minimizing an objective function. The primary term is a positional consistency loss,

$$L_{\text{pos}} = \frac{1}{N}\sum_{i=1}^{N}\left\|\mathbf{p}^i_{\text{finger}} - \mathbf{p}^i_{\text{target}}\right\|^2,$$

where $\mathbf{p}^i_{\text{finger}}$ denotes the position of the i-th fingertip and $\mathbf{p}^i_{\text{target}}$ is its assigned target point. The secondary term encourages stable contacts by penalizing the angular deviation between surface normals, defined as

$$L_{\text{norm}} = \frac{1}{N}\sum_{i=1}^{N}(1 - \cos(\theta_i)),$$

where θ_i is the angle between the normals at the i-th contact pair. Although this optimization in some cases yields a stable grasp in only one to two percent of attempts, its key advantage is its generality; a single, parallelizable method can synthesize a large quantity of grasps in a single batch for any dexterous gripper. Grasp stability is validated in the MuJoCo [65] physics simulator. A candidate grasp is first checked for collisions. If collision-free, the gripper is closed, and a series of impulse forces are applied to the object. A grasp is deemed stable if the gripper maintains contact throughout all checks. This impulse-based procedure is significantly faster than the shaking-motion approach used in [1], while empirically showing more stable grasps. Finally, the validated stable grasps are placed back into the cluttered scenes, following the final data processing steps of [1]. Table II summarizes the total number of grasps generated for each gripper. Due to their geometric complexity and higher degrees of freedom, the dexterous grippers consequently have a lower final grasp density per scene in our dataset. In total, our generated dataset encompasses about 22 500 scenes with over 200 objects sourced from the Google Scanned Objects (GSO) [66] and YCB [67] dataset. Each scene contains one to seven objects and includes up to 1 500 grasps.

C. Multi-Embodiment Grasping

Our evaluation protocol is similar to that of [1]. For each gripper, we measure the grasp success rate across 10 held-out test scenes, executing 100 grasps sampled from our model in each scene. As a pre-processing step, we filter out any grasps that are in collision with the scene. However, grasps that result in gripper self-collisions are counted as failures, as such configurations would damage the physical hardware. Our optimized implementation can sample a full batch of 100 grasps for a dexterous gripper in under 10 seconds on a single consumer-grade GPU.

For the single-embodiment benchmark, we train a dedicated model for each gripper type on a single H200 GPU for 500 scene-epochs. We use a batch size of five scenes, from which we sample 128 grasps per batch. Note that since scenes can contain up to 1,500 valid grasps, each epoch consists of a different random subset. We employ several data augmentation techniques: randomizing the Farthest Point Sampling (FPS) initialization, applying random rotations around the z-axis, and weighting the loss to emphasize later time steps in the flow, which are critical for refining the final pose. As noted in [6], such rotational augmentations are beneficial even for equivariant models to mitigate the accumulation of minor numerical errors across layers.

For the multi-embodiment task, we train a single, unified model using a total batch size of 10 scenes distributed across all five gripper types, training for the same number of epochs. To improve conditioning, we incorporate a technique from Lim et al [37] by adding a sixth 'dummy' gripper with zero DoFs. This dummy gripper serves as an unconditional class, which helps steer the flow generation process more effectively for the specific, conditioned gripper types. During training, a fraction of the data from all real grippers is repurposed for this dummy class by replacing the true gripper label and nullifying the joint information. The results, presented in Table III, indicate a clear performance advantage for the generalist multi-embodiment model in four out of five gripper types.

VII. Limitations and Future Work

Addressing dexterous grasping is significantly more complex than the binary open-close problem statement presented

TABLE III
MULTI-EMBODIMENT GRASP SUCCESS RATES ACROSS VARIOUS GRIPPERS.

Method	Panda Franka (2 DoF)	ViperX 300s (2 DoF)	DEX-EE (12 DoF)	Allegro (16 DoF)	Shadow Hand (22 DoF)
Single-Embodiment	93.9	83.7	72.1	**73.6**	72.4
Multi-Embodiment	**97.0**	**88.3**	**74.5**	66.1	**79.6**

Grasp success percentage per gripper evaluated over 10 scenes testing 100 sampled grasps. Training data set for each gripper encompasses 5 000 scenes. Single-Embodiment models are trained on only gripper specific data, while Multi-Embodiment model is trained on all scenes combined.

in [1], necessitating a larger dataset while maintaining feasible computational costs. This led to several trade-offs in our data generation pipeline. The convex decomposition of objects required by MuJoCo [65] proved to be a bottleneck; certain objects yielded a disproportionately high number of components, making stability evaluation a computationally expensive endeavor, particularly for high-DoF grippers. To manage this, we set a hard cutoff, excluding any object that did not yield at least 5,000 stable grasps for at least one gripper. To mitigate the data generation costs and the dataset bias towards parallel-jaw grippers, we curated a representative subset of grippers. While this slightly reduces the complexity of individual scenes, it allowed us to increase the total scene count. Furthermore, our unified grasp generation strategy for dexterous grippers, which relies on a contact-matching objective, primarily produces fingertip grasps. Consequently, the diversity of grasp types (e.g., tripod, quadrupod) is dictated by the gripper's embodiment rather than a learned strategy. Addressing this limitation to generate a wider variety of grasp taxonomies is an important direction for future work.

A further limitation stems from an architectural trade-off made to simplify the model. In contrast to MultiDiffusion [1], our current approach does not include an explicit geometric representation of the gripper's body during grasp synthesis. This simplification leads to an increased rate of collisions between the gripper and the scene. While we currently filter these invalid grasps in a post-processing step, efficiently re-integrating the full gripper geometry into the model is a key priority for future work.

A prerequisite for our method is a JAX [10] implementation of the gripper's kinematic model. While we implemented these manually, this process could be largely automated by developing a script to parse standard URDF files, facilitating the integration of new gripper types. Our geometric joint representation can naturally accommodate under-actuation via masking during the feature readout; however, we leave the implementation of this capability for future work. As a result, joint constraints are currently enforced softly, learned implicitly from the training data. Experiments with partial point cloud observations were omitted due to increased engineering efforts with variable sized point clouds in JAX [10].

We optimized each module using JAX's vector mapping capabilities to enable parallelization across scenes, grippers, and grasps. As a result, our method can generate a batch of 100 grasps for a dexterous gripper in under 10 seconds on a consumer-grade GPU (<12GB VRAM), a task previously infeasible with comparable Torch-based implementations [5]. Although equivariant methods have a reputation for being memory-intensive, some of our optimized modules are already highly efficient. For instance, the Kinematic Encoder processes a batch of 128 configurations over all five grippers i.e. 640 grasps configurations total, in under 10 ms for one training iteration. This level of performance suggests that such geometric encoders are becoming viable for latency-sensitive tasks beyond grasping. While our full model is not yet ready for real-time online applications, future work could leverage low-level equivariant operations implemented in CUDA e.g. cuEquivariance to bridge this gap. Such efforts are well-justified, as the promise of these methods extends beyond mere efficiency; a strong argument for their use in manipulation is their native capacity to represent vector-based information, such as friction forces and surface normals, while seamlessly integrating scalar features from color or foundation models.

During development, it became evident that learning a universal kinematic encoder for zero-shot grasp generation was infeasible with a dataset of only five grippers. We therefore adopted a simpler approach, using fixed, learnable embeddings for each joint to reduce compute compared to a full encoder learning per class fixed features. We hypothesize that with a sufficiently large and diverse dataset of gripper embodiments, this encoder could be learned, enabling true zero-shot grasp synthesis for unseen grippers [50].

VIII. CONCLUSION

In this work, we present an equivariant, multi-embodiment grasp synthesis method. To our knowledge, this is the first equivariant approach capable of generating grasps for diverse grippers using their full degrees of freedom directly from full-scene point clouds, without relying on object pose estimation. A key feature of our architecture is its ability to sample and process batches simultaneously across multiple scenes and gripper types, a significant step towards scalable robotic learning for equivariant methods. We demonstrate the effectiveness of using equivariant representations to encode joint kinematics, which provides a powerful geometric bias that generalizes across both prismatic and revolute joints. Our experiments empirically validate that this method achieves performance competitive with state-of-the-art techniques for single-embodiment grasp synthesis. To facilitate future research, we release our memory- and compute-efficient JAX [10]implementation, which includes the kinematic models for five distinct grippers and the complete datasets used in our evaluation.

REFERENCES

[1] R. Freiberg, A. Qualmann, N. A. Vien, and G. Neumann, "Diffusion for multi-embodiment grasping," *IEEE Robotics and Automation Letters*, 2025.

[2] P. Li, T. Liu, Y. Li, Y. Geng, Y. Zhu, Y. Yang, and S. Huang, "Gendexgrasp: Generalizable dexterous grasping," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 8068–8074.

[3] M. Attarian, M. A. Asif, J. Liu, R. Hari, A. Garg, I. Gilitschenski, and J. Tompson, "Geometry matching for multi-embodiment grasping," in *Conference on Robot Learning*. PMLR, 2023, pp. 1242–1256.

[4] J. Urain, N. Funk, J. Peters, and G. Chalvatzaki, "Se (3)-diffusionfields: Learning smooth cost functions for joint grasp and motion optimization through diffusion," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 5923–5930.

[5] H. Ryu, J. Kim, H. An, J. Chang, J. Seo, T. Kim, Y. Kim, C. Hwang, J. Choi, and R. Horowitz, "Diffusion-edfs: Bi-equivariant denoising generative modeling on se (3) for visual robotic manipulation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 18 007–18 018.

[6] B. Hu, X. Zhu, D. Wang, Z. Dong, H. Huang, C. Wang, R. Walters, and R. Platt, "Orbitgrasp: Se (3)-equivariant grasp learning," in *8th Annual Conference on Robot Learning*, 2024.

[7] C. Gao, Z. Xue, S. Deng, T. Liang, S. Yang, L. Shao, and H. Xu, "RiEMann: Near real-time SE(3)-equivariant robot manipulation without point cloud segmentation," in *8th Annual Conference on Robot Learning*, 2024.

[8] J. Yang, Z.-a. Cao, C. Deng, R. Antonova, S. Song, and J. Bohg, "Equibot: Sim(3)-equivariant diffusion policy for generalizable and data efficient learning," in *8th Annual Conference on Robot Learning*, 2024.

[9] X. Zhu, F. Wang, R. Walters, and J. Shi, "SE(3)-equivariant diffusion policy in spherical fourier space," in *Forty-second International Conference on Machine Learning*, 2025. [Online]. Available: https://openreview.net/forum?id=U5nRMOs8Ed

[10] J. Bradbury, R. Frostig, P. Hawkins, M. J. Johnson, C. Leary, D. Maclaurin, G. Necula, A. Paszke, J. VanderPlas, S. Wanderman-Milne, and Q. Zhang, "JAX: composable transformations of Python+NumPy programs," 2018. [Online]. Available: http://github.com/jax-ml/jax

[11] M. Sundermeyer, A. Mousavian, R. Triebel, and D. Fox, "Contact-graspnet: Efficient 6-dof grasp generation in cluttered scenes," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 13 438–13 444.

[12] M. Breyer, J. J. Chung, L. Ott, S. Roland, and N. Juan, "Volumetric grasping network: Real-time 6 dof grasp detection in clutter," in *Conference on Robot Learning*, 2020.

[13] Z. Jiang, Y. Zhu, M. Svetlik, K. Fang, and Y. Zhu, "Synergies between affordance and geometry: 6-dof grasp detection via implicit representations," *Robotics: science and systems*, 2021.

[14] C. Eppner, A. Mousavian, and D. Fox, "ACRONYM: A large-scale grasp dataset based on simulation," in *2021 IEEE Int. Conf. on Robotics and Automation, ICRA*, 2020.

[15] H.-S. Fang, C. Wang, M. Gou, and C. Lu, "Graspnet-1billion: A large-scale benchmark for general object grasping," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 11 441–11 450.

[16] J. Zhang, H. Liu, D. Li, X. Yu, H. Geng, Y. Ding, J. Chen, and H. Wang, "Dexgraspnet 2.0: Learning generative dexterous grasping in large-scale synthetic cluttered scenes," in *8th Annual Conference on Robot Learning*, 2024.

[17] 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 *ICRA*, 2023.

[18] 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," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 3891–3902.

[19] J. Lu, H. Kang, H. Li, B. Liu, Y. Yang, Q. Huang, and G. Hua, "Ugg: Unified generative grasping," in *European Conference on Computer Vision*. Springer, 2024, pp. 414–433.

[20] Y.-L. Wei, M. Lin, Y. Lin, J.-J. Jiang, X.-M. Wu, L.-A. Zeng, and W.-S. Zheng, "Afforddexgrasp: Open-set language-guided dexterous grasp with generalizable-instructive affordance," *arXiv preprint arXiv:2503.07360*, 2025.

[21] Y. Zhong, X. Huang, R. Li, C. Zhang, Z. Chen, T. Guan, F. Zeng, K. N. Lui, Y. Ye, Y. Liang *et al.*, "Dexgraspvla: A vision-language-action framework towards general dexterous grasping," *arXiv preprint arXiv:2502.20900*, 2025.

[22] Y. Zhong, Q. Jiang, J. Yu, and Y. Ma, "Dexgrasp anything: Towards universal robotic dexterous grasping with physics awareness," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 22 584–22 594.

[23] 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 *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2024, pp. 2978–2984.

[24] J. Chen, Y. Ke, and H. Wang, "Bodex: Scalable and efficient robotic dexterous grasp synthesis using bilevel optimization," *arXiv preprint arXiv:2412.16490*, 2024.

[25] T. G. W. Lum, A. H. Li, P. Culbertson, K. Srinivasan, A. Ames, M. Schwager, and J. Bohg, "Get a grip: Multi-finger grasp evaluation at scale enables robust sim-to-real transfer," in *8th Annual Conference on Robot Learning*, 2024.

[26] Z. Wei, Z. Xu, J. Guo, Y. Hou, C. Gao, Z. Cai, J. Luo, and L. Shao, "D (r, o) grasp: A unified representation of robot and object interaction for cross-embodiment dexterous grasping," *CoRR*, 2024.

[27] N. Khargonkar, L. F. Casas, , B. Prabhakaran, and Y. Xiang, "Robotfingerprint: Unified gripper coordinate space for multi-gripper grasp synthesis," *arXiv preprint arXiv:2409.14519*, 2024.

[28] 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," *arXiv preprint arXiv:2509.24661*, 2025.

[29] Z. Xu, B. Qi, S. Agrawal, and S. Song, "Adagrasp: Learning an adaptive gripper-aware grasping policy," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 4620–4626.

[30] H. Zhang, K. Y. Ma, M. Z. Shou, W. Lin, and Y. Wu, "Cross-embodiment dexterous hand articulation generation via morphology-aware learning," *arXiv preprint arXiv:2510.06068*, 2025.

[31] H.-S. Fang, H. Yan, Z. Tang, H. Fang, C. Wang, and C. Lu, "Anydexgrasp: General dexterous grasping for different hands with human-level learning efficiency," *arXiv preprint arXiv:2502.16420*, 2025.

[32] M. Bonyani, M. Soleymani, and C. Wang, "Multi-agent deep reinforcement learning for variable-finger dexterous grasping through multi-stream embedding fusion," in *ICRA 2025 Workshop"Handy Moves: Dexterity in Multi-Fingered Hands"Paper Submission*, 2025.

[33] M. Janner, Y. Du, J. Tenenbaum, and S. Levine, "Planning with diffusion for flexible behavior synthesis," in *International Conference on Machine Learning*, 2022.

[34] N. Funk, J. Urain, J. Carvalho, V. Prasad, G. Chalvatzaki, and J. Peters, "Actionflow: Equivariant, accurate, and efficient policies with spatially symmetric flow matching," 2024.

[35] K. R. Barad, A. Orsula, A. Richard, J. Dentler, M. Olivares-Mendez, and C. Martinez, "Graspldm: Generative 6-dof grasp synthesis using latent diffusion models," *IEEE Access*, 2024.

[36] K. Chen, E. Lim, L. Kelvin, Y. Chen, and H. Soh, "Don't Start From Scratch: Behavioral Refinement via Interpolant-based Policy Diffusion," in *Proceedings of Robotics: Science and Systems*, Delft, Netherlands, July 2024.

[37] B. Lim, J. Kim, J. Kim, Y. Lee, and F. C. Park, "Equigraspflow: Se(3)-equivariant 6-dof grasp pose generative flows," in *8th Annual Conference on Robot Learning*, 2024.

[38] K. Wu, C. Hou, J. Liu, Z. Che, X. Ju, Z. Yang, M. Li, Y. Zhao, Z. Xu, G. Yang *et al.*, "Robomind: Benchmark on multi-embodiment intelligence normative data for robot manipulation," *arXiv preprint arXiv:2412.13877*, 2024.

[39] A. Khazatsky, K. Pertsch, S. Nair, A. Balakrishna, S. Dasari, S. Karamcheti, S. Nasiriany, M. K. Srirama, L. Y. Chen, K. Ellis *et al.*, "Droid: A large-scale in-the-wild robot manipulation dataset," in *RSS 2024 Workshop: Data Generation for Robotics*, 2024.

[40] Q. Bu, J. Cai, L. Chen, X. Cui, Y. Ding, S. Feng, S. Gao, X. He, X. Huang, S. Jiang *et al.*, "Agibot world colosseo: A large-scale manipulation platform for scalable and intelligent embodied systems," *arXiv preprint arXiv:2503.06669*, 2025.

[41] P. Atreya, K. Pertsch, T. Lee, M. J. Kim, A. Jain, A. Kuramshin, C. Eppner, C. Neary, E. Hu, F. Ramos, J. Tremblay, K. Arora, K. Ellis, L. Macesanu, M. Leonard, M. Cho, O. Aslan, S. Dass, J. Wang, X. Yuan, X. Yang, A. Gupta, D. Jayaraman, G. Berseth, K. Daniilidis, R. Martin-Martin, Y. Lee, P. Liang, C. Finn, and S. Levine, "Roboarena: Distributed real-world evaluation of generalist robot policies," *arXiv preprint*, 2025.

[42] M. Kim, K. Pertsch, S. Karamcheti, T. Xiao, A. Balakrishna, S. Nair, R. Rafailov, E. Foster, G. Lam, P. Sanketi, Q. Vuong, T. Kollar, B. Burchfiel, R. Tedrake, D. Sadigh, S. Levine, P. Liang, and C. Finn, "Openvla: An open-source vision-language-action model," *arXiv preprint arXiv:2406.09246*, 2024.

[43] Octo Model Team, D. Ghosh, H. Walke, K. Pertsch, K. Black, O. Mees, S. Dasari, J. Hejna, C. Xu, J. Luo, T. Kreiman, Y. Tan, P. Sanketi, Q. Vuong, T. Xiao, D. Sadigh, C. Finn, and S. Levine, "Octo: An open-source generalist robot policy," in *Proceedings of Robotics: Science and Systems*, Delft, Netherlands, 2024.

[44] Q. Vuong, S. Levine, H. R. Walke, K. Pertsch, A. Singh, R. Doshi, C. Xu, J. Luo, L. Tan, D. Shah, C. Finn, M. Du, M. J. Kim, A. Khazatsky, J. H. Yang, T. Z. Zhao, K. Goldberg, R. Hoque, L. Y. Chen, S. Adebola, G. S. Sukhatme, G. Salhotra, S. Dass, L. Pinto, Z. J. Cui, S. Haldar, A. Rai, N. M. M. Shafiullah, Y. Zhu, Y. Zhu, S. Nasiriany, S. Song, C. Chi, C. Pan, W. Burgard, O. Mees, C. Huang, D. Pathak, S. Bahl, R. Mendonca, G. Zhou, M. K. Srirama, S. Dasari, C. Lu, H.-S. Fang, H. Fang, H. I. Christensen, M. Tomizuka, W. Zhan, M. Ding, C. Xu, X. Zhu, R. Tian, Y. Lee, D. Sadigh, Y. Cui, S. Belkhale, P. Sundaresan, T. Darrell, J. Malik, I. Radosavovic, J. Bohg, K. Srinivasan, X. Wang, N. Hansen, Y.-H. Wu, G. Yan, H. Su, J. Gu, X. Li, N. Suenderhauf, K. Rana, B. Burgess-Limerick, F. Ceola, K. Kawaharazuka, N. Kanazawa, T. Matsushima, Y. Matsuo, Y. Iwasawa, H. Furuta, J. Oh, T. Harada, T. Osa, Y. Tang, O. Kroemer, M. Sharma, K. L. Zhang, B. Kim, Y. Cho, J. Han, J. Kim, J. J. Lim, E. Johns, N. D. Palo, F. Stulp, A. Raffin, S. Bustamante, J. Silvério, A. Padalkar, J. Peters, B. Schölkopf, D. Büchler, J. Schneider, S. Guist, J. Wu, S. Tian, H. Shi, Y. Li, Y. Wang, M. Zhang, H. B. Amor, Y. Zhou, K. Majd, L. Ott, G. Schiavi, R. Martín-Martín, R. Shah, Y. Bisk, J. T. Bingham, T. Yu, V. Jain, T. Xiao, K. Hausman, C. Chan, A. Herzog, Z. Xu, S. Kirmani, V. Vanhoucke, R. Julian, L. Lee, T. Ding, Y. Chebotar, J. Tan, J. Liang, I. Mordatch, K. Rao, Y. Lu, K. Gopalakrishnan, S. Welker, N. J. Joshi, C. M. Devin, A. Irpan, S. Moore, A. Wahid, J. Wu, X. Chen, P. Wohlhart, A. Bewley, W. Zhou, I. Leal, D. Kalashnikov, P. R. Sanketi, C. Fu, Y. Xu, S. Xu, brian ichter, J. Hsu, P. Xu, A. Brohan, P. Sermanet, N. Heess, M. Ahn, R. Rafailov, A. Pooley, K. Byrne, T. Davchev, K. Oslund, S. Schaal, A. Jain, K. Go, F. Xia, J. Tompson, T. Armstrong, and D. Driess, "Open x-embodiment: Robotic learning datasets and RT-x models," in *Towards Generalist Robots: Learning Paradigms for Scalable Skill Acquisition @ CoRL2023*, 2023.

[45] J. Yang, C. Glossop, A. Bhorkar, D. Shah, Q. Vuong, C. Finn, D. Sadigh, and S. Levine, "Pushing the limits of cross-embodiment learning for manipulation and navigation," in *Proceedings of Robotics: Science and Systems*, Delft, Netherlands, 07 2024.

[46] K. Black, M. Y. Galliker, and S. Levine, "Real-time execution of action chunking flow policies," *arXiv preprint arXiv:2506.07339*, 2025.

[47] Y. Wang, M. Verghese, and J. Schneider, "Latent policy steering with embodiment-agnostic pretrained world models," *arXiv preprint arXiv:2507.13340*, 2025.

[48] N. Bohlinger, G. Czechmanowski, M. P. Krupka, P. Kicki, K. Walas, J. Peters, and D. Tateo, "One policy to run them all: an end-to-end learning approach to multi-embodiment locomotion," in *8th Annual Conference on Robot Learning*, 2025.

[49] S. Yang, Z. Fu, Z. Cao, G. Junde, P. Wensing, W. Zhang, and H. Chen, "Multi-loco: Unifying multi-embodiment legged locomotion via reinforcement learning augmented diffusion," *arXiv preprint arXiv:2506.11470*, 2025.

[50] B. Ai, L. Dai, N. Bohlinger, D. Li, T. Mu, Z. Wu, K. Fay, H. I. Christensen, J. Peters, and H. Su, "Towards embodiment scaling laws in robot locomotion," *arXiv preprint arXiv:2505.05753*, 2025.

[51] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networkss," in *Computer Vision (ICCV), 2017 IEEE International Conference on*, 2017.

[52] C. Lawrence, H. Kush, D. Karthik, X. Chenfeng, V. Quan, and G. Ken, "Mirage: Cross-embodiment zero-shot policy transfer with cross-painting," in *Robotics: Science and Systems*, 2024.

[53] M. Lepert, R. Doshi, and J. Bohg, "Shadow: Leveraging segmentation masks for cross-embodiment policy transfer," *arXiv preprint arXiv:2503.00774*, 2025.

[54] T. Cohen and M. Welling, "Group equivariant convolutional networks," in *International conference on machine learning*. PMLR, 2016, pp. 2990–2999.

[55] D. E. Worrall, S. J. Garbin, D. Turmukhambetov, and G. J. Brostow, "Harmonic networks: Deep translation and rotation equivariance," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 5028–5037.

[56] E. J. Bekkers, "B-spline cnns on lie groups," in *International Conference on Learning Representations*, 2020.

[57] M. Weiler, P. Forré, E. Verlinde, and M. Welling, "Coordinate independent convolutional networks–isometry and gauge equivariant convolutions on riemannian manifolds," *arXiv preprint arXiv:2106.06020*, 2021.

[58] G. Li, M. Müller, A. Thabet, and B. Ghanem, "Deepgcns: Can gcns go as deep as cnns?" in *The IEEE International Conference on Computer Vision (ICCV)*, 2019.

[59] A. Musaelian, S. Batzner, A. Johansson, L. Sun, C. J. Owen, M. Kornbluth, and B. Kozinsky, "Learning local equivariant representations for large-scale atomistic dynamics," *Nature Communications*, vol. 14, no. 1, p. 579, 2023.

[60] B. Wen, M. Trepte, J. Aribido, J. Kautz, O. Gallo, and S. Birchfield, "Foundationstereo: Zero-shot stereo matching," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 5249–5260.

[61] J. Min, Y. Jeon, J. Kim, and M. Choi, "$s2m^2$: Scalable stereo matching model for reliable depth estimation," *arXiv preprint arXiv:2507.13229*, 2025.

[62] S. Passaro and C. L. Zitnick, "Reducing so (3) convolutions to so (2) for efficient equivariant gnns," in *International conference on machine learning*. PMLR, 2023, pp. 27 420–27 438.

[63] Y.-L. Liao, B. Wood, A. Das*, and T. Smidt*, "EquiformerV2: Improved Equivariant Transformer for Scaling to Higher-Degree Representations," in *International Conference on Learning Representations (ICLR)*, 2024. [Online]. Available: https://openreview.net/forum?id=mCOBKZmrzD

[64] X. Wu, D. DeTone, D. Frost, T. Shen, C. Xie, N. Yang, J. Engel, R. Newcombe, H. Zhao, and J. Straub, "Sonata: Self-supervised learning of reliable point representations," in *CVPR*, 2025.

[65] E. Todorov, T. Erez, and Y. Tassa, "Mujoco: A physics engine for model-based control," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2012, pp. 5026–5033.

[66] L. Downs, A. Francis, N. Koenig, B. Kinman, R. Hickman, K. Reymann, T. B. McHugh, and V. Vanhoucke, "Google scanned objects: A high-quality dataset of 3d scanned household items," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 2553–2560.

[67] B. Calli, A. Singh, A. Walsman, S. Srinivasa, P. Abbeel, and A. M. Dollar, "The ycb object and model set: Towards common benchmarks for manipulation research," in *2015 international conference on advanced robotics (ICAR)*. IEEE, 2015, pp. 510–517.