GraspGen-X: Cross-Embodiment 6-DOF Diffusion-based Grasping

Beining Han[1,2] Yu-Wei Chao[1] Erwin Coumans[1] Clemens Eppner[1]
Jia Deng[2] Stan Birchfield[1] Adithyavairavan Murali[1]
[1] NVIDIA [2] Princeton University
graspgenx.github.io

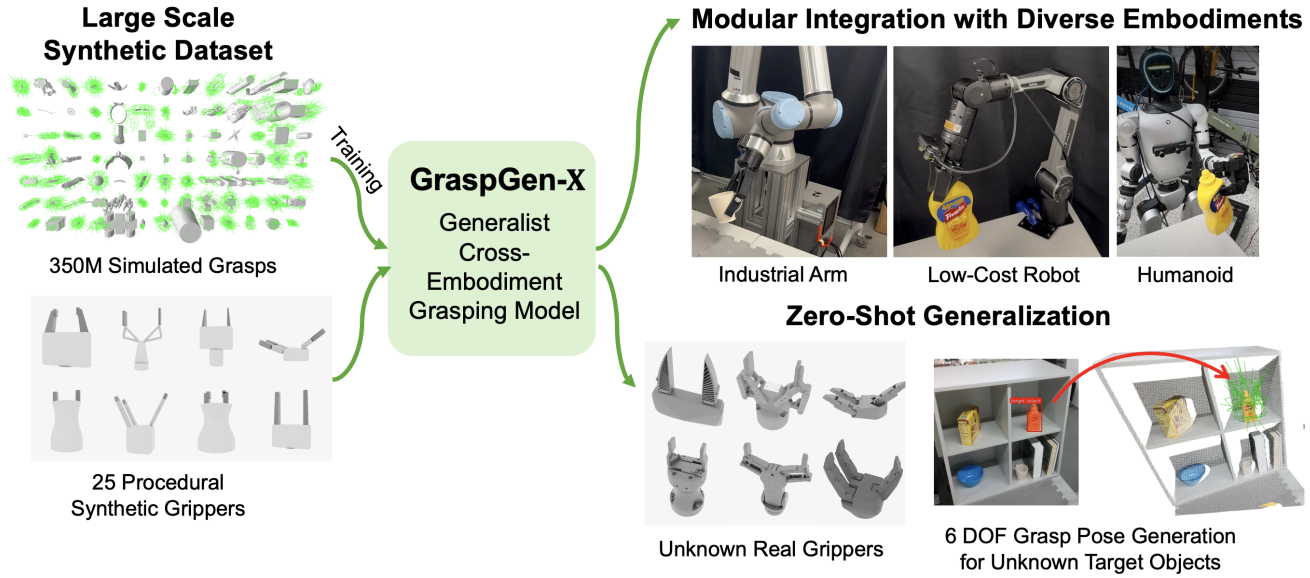

Figure 1. We introduce GraspGen-X, a cross-embodiment 6 DOF Grasping model trained with a large scale dataset of procedural grippers and 395 Million simulated grasps. We achieve zero-shot generalization to both unknown objects as well as grippers in the real world.

Abstract

We study cross-embodiment 6-DOF robot grasping. Unlike prior works, we require the model not only to generalize to novel objects / scenes but also to novel gripper morphologies and physical grasping processes. Our method extends diffusion model based generative 6-DOF grasping models to condition on the additional gripper's representation. We propose a swept-volume heuristic for encoding the gripper. We train our cross-embodiment model with procedural grippers and a large-scale dataset of 395 Million grasps. In simulation experiments, our model has the best zero-shot generalization to novel real-world grippers and objects over baseline methods. Our model also serves as a good initialization for fine-tuning to adapt to novel grippers. In ablations, we demonstrate the efficiency of our sweep-volume gripper representation and our procedural gripper training dataset. Last, we show zero-shot generalization to real-world novel grippers for 6-DOF grasping, surpassing baselines in cross-embodiment generalization.

1. Introduction

Grasping is a fundamental problem in robotics with widespread applications in industrial and household environments. In particular, the field of 6-DOF grasp generation has been significantly advanced by rapid progress in generative models [40, 43, 62], 3D perception [69], physics simulation [14], and vision language models (VLM) [11].

To broaden the utility of robotic grasping in downstream applications, it is crucial to develop a generalized pick-and-place (GPnP) system. Such a system should ideally demonstrate zero-shot performance or support in-context learning – not only for grasping novel objects across varied poses and environments, but also when deployed on a completely new robot embodiment. Current GPnP systems are typically modular, comprising interoperable components such

as grasp generation [15, 36, 57, 78], motion planning [55], occupancy mapping [37], and instance segmentation [50], all orchestrated by a high-level planner such as a VLM [11, 25, 28, 53, 79] or task planner [10, 33, 65]. Several modules of this GPnP system have already matured and can translate zero-shot when deployed on unseen manipulators. For example, 3D perception has matured significantly, with advances in both sensing hardware and learning-based stereo models [68], and instance segmentation has benefited greatly from foundation models such as SAM2 [50]. Motion generation tools, such as motion planners [55] and 3D reconstruction for occupancy mapping [37], are largely model-based rather than learned. As a result, adapting them to a new robot typically only requires a one-time effort to specify the robot's configuration. However, despite substantial progress in improving the generalization of 6-DOF grasp generation to unseen objects [40, 44], clutter [42], and tasks [41], existing methods still require retraining a new grasping model when the gripper changes. This makes grasp generation the least transferable component in cross-embodiment settings.

Recent work towards generalist robot models has explored cross-embodiment training [4, 13, 73]. For 6-DOF grasp generation, such strategy can also offer benefits. First, one can leverage a larger dataset, potentially sharing information between different embodiments to improve learned features. Second, it allows zero-shot adaptation to novel grippers and objects, reducing the need for hardware-specific training. This is especially important from the perspective of end-consumers of grasp pose generation, who may not have the time or compute resources. For example, [43] needs a week of an 8-GPU node to generate data and train a single-embodiment model. Many works have resorted to kinematic retargeting of grasp poses from a model trained on one embodiment and using it on a different one [25, 41]. For example, for pinch jaw grippers, this can be achieved with a simple translation offset along the grasp approach direction. We demonstrate quantitatively that grasp pose re-targeting, while simple to implement, does not yield the best performance in large-scale simulated evaluations.

In this work, we present GraspGen-X, a diffusion-based cross-embodiment 6-DOF grasp generation model that explicitly encodes a parameterized representation of the robot. Our model builds on GraspGen [43], a 6-DOF grasp generation framework composed of a diffusion-based grasp pose generator and a discriminator for ranking. We extend its design by conditioning both the generator and the discriminator on the gripper's representation. Here, we propose to parameterize the gripper by its Swept Volume, defined as the region traversed by the robot fingers during its grasping motion (Sec 3.1). With this heuristic, we demonstrate zero-shot generalization to novel grippers. Furthermore, in our training pipeline, we propose procedurally generating simulation grippers for grasp pose generation and training (Sec 3.2). While there are several high-quality real-world grippers that are commercially available, we find that they are not scalable to build the training dataset and the distribution is biased for learning cross-embodiment models. Our GraspGen-X is trained on a dataset of 350M sampled grasps generated with the ACRONYM pipeline [14] on 25 procedural grippers. To the best of our knowledge, this is the largest multi-embodiment dataset that has ever been used to train the grasping model.

In summary, our contributions are as follows. We present GraspGen-X, a cross-embodiment 6-DOF grasping model and demonstrate that it has a strong zero-shot generalization to novel real-world grippers, surpassing common methods such as gripper retargeting (Sec 5.1) and other baselines [15, 43]. Moreover, we find that GraspGen-X serves as a better initialization checkpoint for finetuning on a new target gripper, compared to training from scratch and finetuning from single-embodiment models (Sec 5.2). We also conduct an extensive study on other common choices of gripper's representations and on comparing training with procedural grippers v.s. real-world grippers. We show that our Swept Volume heuristic is an efficient representation for cross-embodiment grasping, and our procedural grippers are a better training distribution. Lastly, we will open source the model, code, and dataset of 395 Million grasps.

2. Related Work

2.1. 6-DOF Grasping

6-DOF grasp generation refers to the problem of predicting SE(3) grasp poses for a robot gripper given an observation of an unknown object [44]. The pose will result in the gripper stably picking up the object when the gripper closes. It can be applied for several downstream applications: target-driven grasping in clutter [42, 57, 74], language-guided semantic manipulation [58] and can be orchestrated by a high-level task planner [10, 33] or VLM [11, 25, 28, 53, 79]. 6-DOF grasp generation is usually defined as a two-stage process: grasp sampling and grasp analysis [44]. Grasp sampling was initially implemented with heuristic samplers such as derivative-free optimization [27, 36], analytic antipodal sampling [30, 59] or pixel-wise prediction (i.e. pixels in the image depth input are grasp pose candidates) [15, 57, 78]. More recently, the progress in generative models has allowed us to learn samplers in the form of autoregressive models [60, 78], Variational Autoencoder (VAE) [40, 42], flow-matching [31] and diffusion models [3, 6, 17, 34, 62, 72]. Grasp analysis typically entails discriminator models to score sampled grasps [40, 42, 54, 70]. Recent methods such as GraspGen [43] propose a powerful combination of diffusion-based grasp sampling and discriminators trained through an in-

terleaved data flywheel. Our work is concerned with cross-embodiment models that can generalize not only to novel objects but also to novel grippers.

2.2. Cross-Embodiment Learning in Robotics

Cross-embodiment learning is a challenging transfer learning problem in robotics. Prior work has proposed many methods: learning an implicit encoding of robot [9, 19], conditioning on an explicit representation of robot properties [9], reward functions [80], modular policies [12, 24], model-based RL [23, 51] and test-time adaptation [77]. Action retargeting has been a popular approach, where actions from a source embodiment are mapped to a target embodiment, such as in tele-operation of dexterous hands [22, 47]. More recently, Vision-Language-Action (VLA) models such as RT-X [73] have proposed general-purpose architectures that can be trained by aggregating datasets across diverse robots [4, 5, 13, 73] and using methods like readout tokens [13] or masking [4] to account for variable action space. VLAs still do not achieve strong zero-shot performance in completely new robot hardware or tasks. We hypothesize that an explicit parameterization of the embodiment, such as our Swept Volume representation in Sec 3.1, is necessary to achieve zero-shot performance on unknown robots in grasping.

2.3. Multi-Embodiment Grasping

In the context of robot grasping, prior work has proposed both new datasets [7, 29] and algorithms [2, 18, 52, 75] for multi-gripper grasping problem. *Object-centric* methods have used to learn a Contact Map [29], contact points [2, 52] or dense point-wise contact correspondences between the object and robot [16, 67]. They then use inverse kinematics (IK) to generate joint configurations and align robot contact points. These approaches assume access to a complete 3D geometry of the object, hence they do not generalize to partial point cloud observations where reliable contact points are occluded. Several *gripper-aware* methods use various techniques to encode the robot: [64] uses a parameterized fingertip antipodal representation, UniGrasp [52] and $\mathcal{D}(\mathcal{R}, \mathcal{O})$ [67] uses a conditional Variational Autoencoder (cVAE), AdaGrasp [75] uses a Truncated Signed Distance Field (TSDF) representation, and [18] use a PointNet++ [46] backbone. All such methods are trained on just a handful of real-world grippers and hence struggle to generalize to new grippers. In contrast, we train on procedural synthetic grippers and propose a novel gripper representation for cross-embodiment grasping. Ours achieves strong zero-shot generalization to unknown real grippers.

3. GraspGen-X: Cross-Embodiment Grasping

In the 6-DOF cross-embodiment robot grasping problem, we are given the object/scene pointcloud, the gripper's

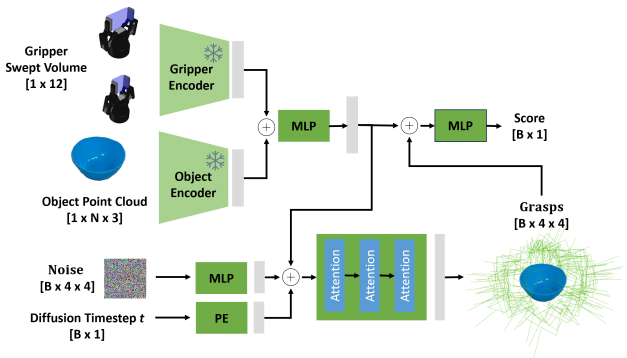

Figure 2. Architecture for our GraspGen-X model. The model is developed based on GraspGen [43]. We additionally condition on the gripper's representation for both the generator and the discriminator. The gripper embedding is computed from the Swept Volume heuristic (Sec 3.1).

URDF, and its gripper closing motion, i.e., the joint trajectory of the gripper from fully open to fully closed. The objective is to predict SE(3) grasp poses that can successfully grasp the object. Most importantly, our aim is to generalize to both novel grippers and objects (Fig 1).

Our model is developed from GraspGen [43], the SOTA diffusion-based grasp pose generator, which balances well between grasping accuracy and grasp pose coverage. GraspGen is composed of a diffusion model for SE(3) grasp pose generator and a discriminator to estimate the accuracy of the generated grasps. In GraspGen, the diffusion-based generator diffuses over the SE(3) space and is conditioned on the object embedding. The object embedding is encoded from object pointclouds with a PointTransformer [71] or PointNet++ [46]. The discriminator is trained with on-generator data of positive and negative grasp poses to predict the confidence of each grasp. It is conditioned on the same object embedding.

For our cross-embodiment model, we additionally condition on the embedding of gripper. Figure 2 shows the architecture of GraspGen-X , which is an extension of the GraspGen (Fig. 2 in [43]). The gripper embedding is encoded with a 3-layer MLP of the Swept Volume heuristic when the gripper is fully open and half open (Section 3.1).

3.1. Gripper Encoding with Swept Volume

Our work includes gripper categories of parallel grippers (e.g., Franka Panda Hand), 2-finger revolute grippers (e.g., Robotiq-2F85), and high-dof 3-finger grippers (e.g., Unitree G1 7-DOF Hand). To encode the gripper's morphology and closing motion into an efficient embedding for grasping models, we propose to use the Swept Volume of the gripper when it is fully open and half open.

Swept Volume is a heuristic to approximate the space that gripper fingers will sweep through during the closing process. As shown in Fig 3, we approximate the space with an axis-aligned cube. Thus, each Swept Volume consists of

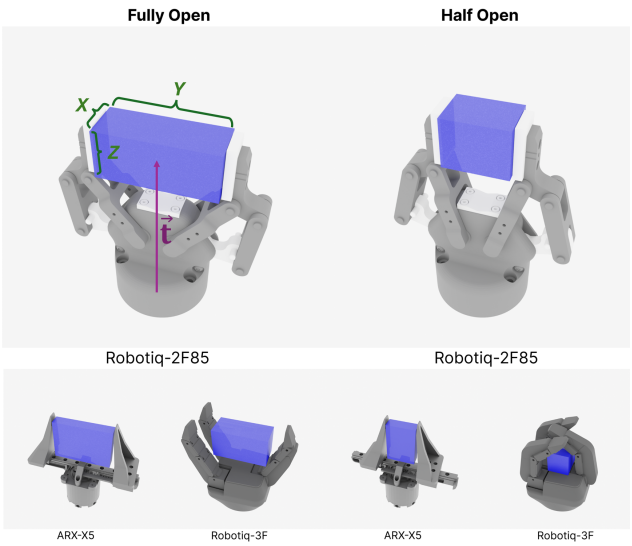

Figure 3. Illustration of the Swept Volume, i.e., cube dimensions (xyz) and cube center translation relative to gripper base ($\mathbf{t}$). We visualize the Swept Volume cube with the blue cube, when grippers are fully open (top-L) and half open (top-R). The Swept Volume varies between grippers (bottom).

the cube dimension (3-dim) and the translation of the center from the gripper's base frame (3-dim). We use the Swept Volume (6-dim) heuristic when the hand is fully open and when it is half-way through the closing process. The gripper's representation input is a 12-dim vector in total. We encode it with a simple 3-layer MLP to the 512-dim gripper embedding to both the geneartor and the discriminator.

As illustrated in Fig 3, we determine the dimension and translation of Swept Volume in the following way. For 2-finger parallel grippers (e.g., Franka Panda Hand) and revolute grippers whose fingers are always parallel during the closing process (e.g., Robotiq-2F85), we estimate a Swept Volume to cover the space between the two fingers. For grippers whose fingers are not always parallel but will rotate w.r.t the gripper base (e.g., Robotiq-3F), we approximate the space that the finger will sweep through in the follow-up process with a cube.

3.2. Procedural Gripper Generation

Real-world grippers are limited in quantity and diverse in geometry and physical closing motion. In our work, we collect a total of 20 real-world grippers (Figure 10, Appendix) and evenly divide them into 10 training grippers and 10 test grippers. In Section 5.4, we find that training with limited real-world grippers has relatively poor performance, due to the distribution mismatch between training/test sets and the relatively scattered data points in gripper's space.

Consequently, we propose to train the cross-embodiment model with procedural grippers randomly generated from the procedural robot gripper generator. Our procedural gripper generator is implemented with Infinigen-Sim [26], which leverages Blender's geometry node to devise procedural mathematical rules to compose articulated objects under random configurations. Fig 4 shows examples of procedural grippers randomly generated with our generator.

We design a procedural generator class for each gripper category, i.e., parallel grippers, revolute 2-finger grippers, and 3-finger high-dof grippers. As our grippers are only used in simulation-based grasping data generation and in learning gripper's embedding, we do not need to model the fine details like screws and connectors in designing CAD models. Instead, we focus on the diversity of the overall size / morphology and on the diversity and realism of finger geometry, which will likely come into contact with the objects during the closing process. Additionally, our procedural generator also outputs the Swept Volume and other metadata needed for model training and data generation. Please refer to the appendix for more details of the generators.

To determine the distribution of the random configurations to our generator, we leverage the set of real-world training / test grippers. We tune the randomization range so that the Swept Volume heuristics cover the real-world counterpart's distribution. Fig 5 shows the distribution of Swept Volume dimensions of fully open grippers, comparing between 50 procedural grippers, 10 real training grippers, and 10 real test grippers. The distribution of the 10 real training grippers (Real-Train10) has some regions which are non-overlapping and out of distribution compared to the test set of grippers (Real-Test10). In contrast, our procedural gripper dataset (Proc-Train50) has a higher overlap in-distribution with the test set (Real-Test10), demonstrating the potential for superior generalization results.

4. GraspGen-X Dataset

To train the cross-embodiment model for zero-shot generalization, we have generated a large-scale grasping dataset. We follow the same antipodal sampling and simulation-based grasp labeling pipeline in ACRONYM [14], which is widely used in previous work [43, 57, 78]. For 3-finger high-dof grippers (e.g., Unitree G1 7-DOF Hand), we use the same antipodal sampler and find that it provides a reasonable grasp distribution for 3-finger gripper grasping.

In our GraspGen-X dataset, we use a subset of 3.5K training objects used in GraspGen [43] training and 453 objects as test objects. For all test objects, we ensure that each of 10 test grippers has at least 5 positive grasps to compute the coverage metric. To train the cross-embodiment model, we randomly generate 25 procedural grippers, with 10 parallel grippers, 10 2-finger revolute grippers, and 5 3-finger high-dof grippers. For each gripper and each object, we sample a maximal of 2K grasps and utilize Isaac-Sim [1] to simulate the grasp outcome. In total, we have sampled and evaluated 175M grasps for generator training, which takes

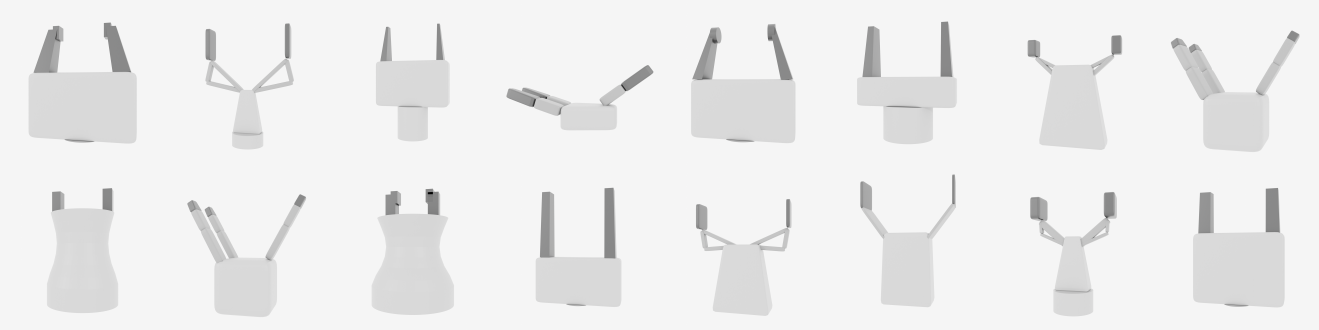

Figure 4. Examples of our procedural grippers. Each gripper is a random instance from our generator (Sec 3.2). Please refer to the appendix for generator details.

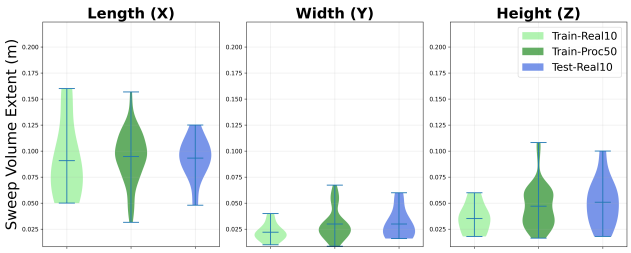

Figure 5. The distribution of Swept Volume cube dimensions along three axes, for training datasets (Train-Proc50 and Train-Real10) and a test dataset consisting of 10 novel real grippers (Test-Real10). All grippers are fully open.

approximately 8.7K GPU hours. The generator is trained with 8 A100 GPUs for 780K steps with a learning rate $1e$-5, which spans 80 hours.

To generate the on-generator data for discriminator training, we generate 2K grasps for each object and each gripper, and then we evaluate grasps with the same labeling pipeline, which have taken 5.2K GPU hours. We train the discriminator with 50% on-generator positive grasps and 50% on-generator negative grasps. The discriminator is trained with 8 A100 GPUs at a learning rate of $1e$-5 for 300K steps, which spans 76 hours. Please refer to the appendix for details of the GraspGen-X dataset and training.

5. Simulation Experiments

For simulation-based experiments, we follow the full pointcloud experiments and evaluation metrics in GraspGen[43]. For test grippers and test objects, we sample 5K grasps to curate the ground-truth grasp dataset with the ACRONYM pipeline. For each test object and gripper, the generator generates 2K grasps, validated with the discriminator. We use ranked grasps to produce the precision-recall (PR) curve. We report the final AUC value of the PR curve, averaged over all test objects and test grippers.

5.1. Zero-shot Evaluation

We compare GraspGen-X with the following baselines.

- **GraspGen Direct Transfer (DTR)**: We train a GraspGen model for the Franka gripper over the same set of 3.5K training objects and with 7M sampled grasps, and directly apply the model on test grippers.
- **GraspGen Retargeting (RTG)**: We use the GraspGen (DTR) model but retarget the predicted grasp poses. Specifically, we apply an offset to the grasp pose along the approach direction, based on the distance of the fingertip position between the test gripper and Franka. This process is commonly used for adapting a 6-DoF grasp pose to a new gripper [25, 41, 76].

Table 1 shows the zero-shot performance of 10 test grippers by category. GraspGen-DTR completely fails to adapt to grippers that fall in a different category, e.g., revolute 2-finger grippers. GraspGen-RTG achieves a relative improvement of over 200% compared to DTR, suggesting that the widely used heuristic is an efficient technique. However, it only considers the z-axis offset between different grippers' fingertips, and fails to consider the difference in the finger geometry and contact dynamics. Thus, there is still significant room for improvement, even in the parallel 2-finger gripper category.

In contrast, GraspGen-X achieves the SOTA performance in all categories, further improving over GraspGen-RTG by 25%. This suggests that it is more promising to learn an end-to-end cross-embodiment model rather than applying a simple pose correction technique to single-embodiment models. This is especially the case for high-dof 3-finger grippers where the relative improvement is nearly 40%. The gap between different grippers comes from a mixture of morphology difference, finger geometry difference, and the difference in the physical process of hand closing.

We also evaluated GraspGen-X on two *out-of-the-distribution* 5-finger grippers, that are not involved during the training. Surprisingly, our model can still achieve a relatively high performance of 0.404 on Surge Hand and 0.363 on Inspire Hand.

Additionally, Fig 6 visualizes the grasp poses predicted by ours GraspGen-X of a novel object with novel test grip-

Table 1. Zero-shot performance on novel test grippers and novel test objects. We report the average of 4 parallel grippers, 4 revolute 2-finger grippers, 2 high-dof 3-finger grippers, and all 10 grippers.

	Parallel 2F	Revolute 2F	High-dof 3F	All
GraspGen-DTR	0.215	0.033	0.136	0.126
GraspGen-RTG	0.365	0.379	0.503	0.398
GraspGen-X (**Ours**)	**0.502**	**0.413**	**0.699**	**0.506**

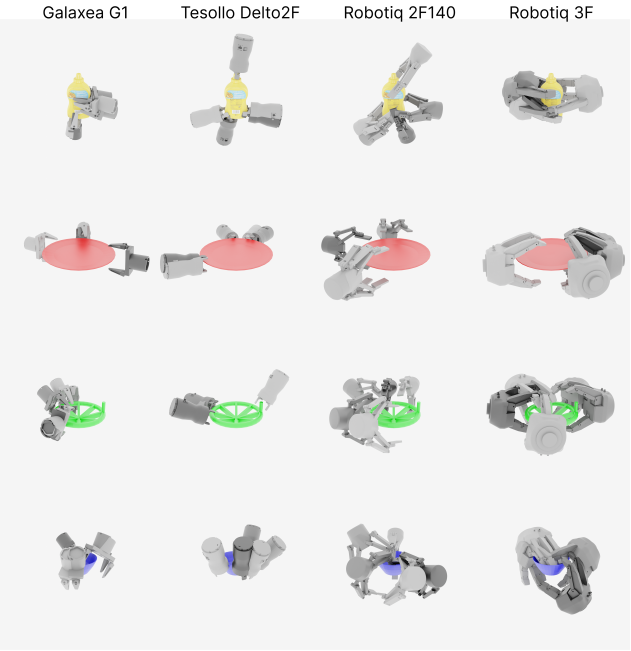

Figure 6. Visualization of generated grasp poses with GraspGen-X on Galaxea-G1, Tesollo-Delto2F, Robotiq-2F140, Robotiq3F grippers on 4 novel objects.

pers. Please refer to the appendix for more results.

5.2. Supervised Finetuning Adaptation

We show that our GraspGen-X is a better initialization for supervised finetuning (SFT) on novel grippers. We randomly select $1/5$ of all training objects to generate a small finetuning dataset of 140K sampled grasps following the same pipeline. For each gripper, this takes approximately 28 GPU hours of computation. We then finetune the generator on 4 A100 GPUs for 4 hours for each gripper separately. Here, we plot the learning curve of the generator's metrics on the test objects, averaged over all 10 test grippers. Similar to [43], we use metrics of grasp pose translation / rotation error and the recall rate of ground truth positive grasps. Please refer to the Appendix for the description of these metrics.

Fig 7 shows the learning curve of three metrics when training a GraspGen model from scratch (GraspGen-Scratch), finetuning the Franka Panda GraspGen in Sec 5.1 (GraspGen-Franka-SFT) and finetuning GraspGen-X in

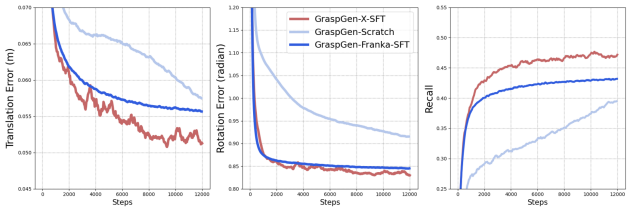

Figure 7. Learning curve of generator finetuning (Sec 5.2). We plot the metric of translation error, rotation error, and recall rate of the target gripper along the training. The curve is averaged over all 10 test grippers.

Sec 3 (GraspGen-X-SFT). For GraspGen-Franka-SFT and GraspGen-X-SFT, we use a learning rate of $1e$-6, which we find to be the most efficient. For GraspGen-Scratch, we use the same $1e$-5 learning rate as in Sec 3.

GraspGen-X-SFT shows the most efficient learning compared to GraspGen-Franka-SFT and GraspGen-Scratch, suggesting that the GraspGen-X model is a better initialization when adapting to a novel gripper. We believe that the ability to cross-embodiment GraspGen-X makes the model more adaptive to a novel gripper representation and a novel 6-dof grasping dataset.

5.3. Gripper Encoder Comparison

We compare our Swept Volume representation for the gripper encoding (Fig 3) with the following baseline encodings used in prior work.

- **AdaGrasp [75]:** We use a $64 \times 32 \times 64$ volumetric truncated sign distance field (TSDF), computed based on the gripper's mesh. It is first encoded with a 3D CNN and then projected with a 2D CNN to a vector embedding. We concatenate the embeddings when the gripper is fully open, half open, and fully closed.
- **UniGrasp [52]:** We first train a Pointnet-based VAE with grippers' pointcloud under random configurations. We use the 64-dim latent embedding of fully open and fully closed grippers. The input is projected to the 512-dim gripper embedding with MLP. For experiments with 10 real training grippers, the pointnet-based VAE is only trained with these 10 grippers. For experiments with procedural grippers, the pointnet-based VAE is trained with 100 randomly generated procedural grippers.
- **PointNet++ [46]:** Inspired by [17], we use PointNet++

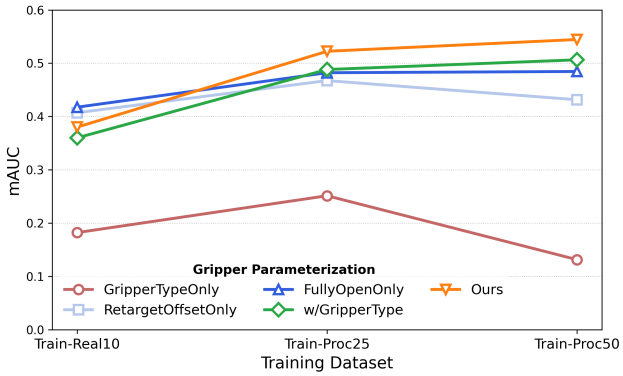

Figure 8. Ablation study on GraspGen-X gripper parameterization and gripper training sets (Sec 5.4), evaluated on 10 test grippers.

to encode the pointcloud sampled from the gripper's mesh surface. The embedding is the concatenation of encodings when the gripper is fully open and closed.

Due to the high computational cost of training GraspGen-X with the full set of objects, we adopt a smaller scale training experiment with 453 test objects. Namely, we train and test on the same set of objects, while we still evaluate on the same 10 novel test grippers with the model trained on 25 procedural grippers.

Table 2. Comparison with baseline gripper encoding methods used in previous work (Sec 5.3). The table shows the mAUC of 453 test objects and 10 real test grippers.

AdaGrasp [75]	UniGrasp [52]	PointNet++ [17]	GraspGen-X
0.432	0.418	0.349	**0.528**

Table 2 shows the results of using other gripper encoders and inputs. Clearly, our Swept Volume heuristic shows a stronger zero-shot generalization performance on novel grippers, improving over the second best AdaGrasp (TSDF) by 25%. This suggests that our heuristic is an efficient representation of the grasping problem. Please refer to the appendix for more results on the gripper encoder comparison.

5.4. Ablation Study

In this section, we present ablations on our Swept Volume parameterization and procedural gripper dataset. We follow the same experiment setup as in Sec 5.3.

Procedural Gripper Training: To compare the performance with different training gripper distributions, we sample grasp poses and train the same model on test objects with the following set of grippers. For a fair comparison, we sample a total of 50K grasps for each object in all datasets.

- *10 Real Grippers (Train-Real10).* With the 10 exclusive real grippers as training grippers, we sample 5K grasps for each gripper on each object.
- *25 Procedural Grippers (Train-Proc25).* With the same 25 procedural grippers in Sec 3, we sample 2K grasps for each gripper on each object.
- *50 Procedural Grippers (Train-Proc50).* We randomly generate another 50 procedural grippers with the same generator, including 20 parallel grippers, 20 2-finger revolute grippers and 10 3-finger high-dof grippers. We then sample 1K grasps for each gripper on each object.

Compared with training with existing real grippers, procedural gripper training yields a significant improvement for all types of gripper encodings (Figure 8). We hypothesize that this comes from the fact that our procedural grippers provide a better coverage over test grippers, while training grippers and test grippers are largely non-overlapped and are sparsely distributed in the gripper space. Moreover, we observe that the use of more procedural grippers helps to improve performance, when using Swept Volume encoding, as well as TSDF and PointNet++ (Figure 11, Appendix). Here, we only use 25 procedural grippers in GraspGen-X due to limited computation resources.

Swept Volume Parameterization: We compare on our Swept Volume parameterization with other related heuristics. All inputs are encoded with a 3-layer MLP to the gripper embedding.

- *GraspGen-X-GripperTypeOnly:* We condition on the gripper type, i.e., parallel, 2-finger, and 3-finger high-dof, which is a 3-dim one-hot vector.
- *GraspGen-X-RetargetOffsetOnly:* We condition on the retargeting offset value, i.e., the gripper's z-axis distance between the base frame and the fingertip, a real value.
- *GraspGen-X-FullyOpenOnly:* We only condition the Swept Volume when the gripper is fully open, which is a 6-dim vector.
- *GraspGen-X-w/GripperType:* We use the concatenated gripper type one-hot 3-dim vector and the 12-dim Swept Volume vector (both fully open and half open) as input.

Figure 8 shows the results of all parameterizations in our ablation study. Ours Swept Volume (12-dim vector) achieves the best performance when training with procedural grippers. GripperTypeOnly and RetargeOffsetOnly fail to learn a comparable model, due to the oversimplified gripper information in these parameterizations. We also find that using FullyOpenOnly Swept Volume performs worse than with both fully open state and half open state. This is mainly because of 2-finger revolute grippers (Figure 12, Appendix). The finger of 2-finger revolute grippers such as Roboitq-2F140 and XArm Hand will move forward on the z-axis during gripper closure. Consequently, it is important to encode the information of the closing process with an additional half open Swept Volume. Interestingly, we find that conditioning on additional gripper type does not help Swept Volume. We hypothesize that when training a cross-embodiment model, it is important to share the information between different types of grippers. Thus, the additional gripper type input that separates the parameterization space degrades the performance.

6. Real Robot Experiment

GraspGen-X generalizes to the real world despite being only trained in simulation. Our real robot setups are shown in Fig 1. We demonstrate the model on two real grippers unseen by the model without any finetuning: a precise industrial manipulator of an UR10 robot equipped with a robotiq-2f-140 gripper (a 2-finger revolute gripper) and a low-cost Piper robot with its standard parallel gripper. Details of our real robot experimental setup is discussed in the Appendix.

6.1. Evaluation on an Industrial Manipulator

Baseline Comparisons: We evaluate in the context of zero-shot cross-embodiment grasping, where neither GraspGen-X nor baselines have been trained on this particular robot. We compare to two baselines: the GraspGen [43] model from Section 5.1 as well as AnyGrasp [15], a recent grasping in clutter model trained on real colored point clouds of tabletop objects. For GraspGen, we retarget the predicted grasp poses meant for the Franka gripper to the Robotiq-2F140 by applying an offset along the approach direction. For the pretrained AnyGrasp model, we were unable to get consistent grasp predictions without the following post-processing steps. First, we applied a translation offset in the camera's z-axis to match the original training dataset, which was collected at a randomized elevation/azimuth but a fixed camera depth. Second, we found better performance without applying non-maximum suppression, most likely since our motion planner [56] is proficient with batch targets. We evaluate on 12 isolated objects under 5 different poses without any clutter and on 5 objects under 3 different poses arranged on a cluttered shelf (an example is shown in the bottom left of Figure 1).

Discussion: As shown in Table 3, GraspGen-X achieved an overall success rate of 79.0 %, outperforming both GraspGen and AnyGrasp. GraspGen-X performed well across both environments, though it struggled in the more challenging cluttered shelf environment. Motion planning is more difficult in clutter as most grasps would be rejected due to to kinematic infeasibility and collisions. Apart from predicting precise grasps in these new environments, the grasp model also needs to generate grasps with high spatial coverage. This will increase the odds of having feasible grasps after a sequence of rejection sampling by the planner. Both GraspGen-X and GraspGen are object-centric models and hence they naturally generalized to more complicated shelf clutter with the help of SAM2. Since AnyGrasp is a scene-centric model trained only with data for tabletop clutter, it deteriorated in the out of distribution shelf scene.

6.2. Evaluation on a Low-Cost Robot

Low-Cost robots have the potential for democratizing robotics [20] but they come with an additional set of challenges regarding control error, actuator noise and calibration errors. We used the same GraspGen-X model checkpoint from the previous section on the AgileX's parallel gripper. Here, we used object pose estimation and complete models, demonstrating that our model also generalizes to complete point cloud input. GraspGen-X can grasp a YCB Mustard bottle and ArUCo cube with 100 % success rate averaged across 10 trials, demonstrating proficiency in cross-embodiment transfer.

Method	Isolated Objects	Clutter	Overall
GraspGen-X(Ours)	**85.7%**	**71.4%**	**79.0%**
GraspGen-RTG [43]	73.3%	57.1%	65.2%
AnyGrasp [15]	80.0%	42.9%	61.4%

Table 3. Grasp success rate on an industrial manipulator with Robotiq-2F140 gripper (Section 6.1).

7. Conclusion

Our work studies cross-embodiment 6-DOF grasping. Namely, we require the model to predict 6-DOF grasp poses not only for novel objects but also for novel gripper morphology and closing motion. In particular, our work consider grippers of 2-finger parallel grippers, 2-finger revolute grippers, and high-dof 3-finger grippers. Towards this problem, we propose GraspGen-X, a diffusion-based cross-embodiment 6-dof grasping model, extending GraspGen [43] for cross-embodiment generalization. In GraspGen-X, we use the Swept Volume heuristic to represent the gripper in the model, i.e., a 12-dim vector approximating the space that fingers will sweep during gripping. We train GraspGen-X with procedural grippers and on a large-scale dataset of 350M sampled grasps, the largest synthetic dataset for 6-DOF multi-embodiment grasping. Experiments suggest that our end-to-end GraspGen-X model achieves the best performance over baseline techniques for cross-embodiment generalization. Moreover, our model serves as a good initialization for finetuning of the target novel gripper. Our ablation studies show that our gripper representation is more efficient compared to other common representations, and our procedural grippers provide a more adequate training gripper distribution. Real robot experiments show that our model, trained with only synthetic data, generalizes well to real novel grippers in 6-DOF grasping, surpassing baselines in novel objects, novel environments, and a novel embodiment.

Future work can emphasize training large models with more diverse grippers and objects. Moreover, it is necessary to extend our model to handle more complex grippers, e.g., training with 5-finger dexterous hands. Due to computation restrictions, we were only able to train GraspGen-X on 3.5K objects and 350M grasps, while we believe that further scaling of this training will yield additional performance gains.

References

[1] Isaac sim - robotics simulation and synthetic data _ nvidia developer. https://www.einscan.com/. Accessed: 2024-03-07. 4

[2] Maria Attarian, Muhammad Adil Asif, Jingzhou Liu, Ruthrash Hari, Animesh Garg, Igor Gilitschenski, and Jonathan Tompson. Geometry matching for multi-embodiment grasping. In *Proceedings of the 7th Conference on Robot Learning (CoRL 2023)*, 2023. 3

[3] Kuldeep Barad, Andrej Orsula, ANntoine Richard, Jan Dentler, Miguel Olivares-Mendez, and Carol Martinez. Graspldm: Generative 6-dof grasp synthesis using latent diffusion models. In *IEEE Access*, 2024. 2

[4] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolò Fusai, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Lucy Xiaoyang Shi, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. : A vision-language-action flow model for general robot control. arXiv preprint arXiv:2410.24164v1, 2024. 2, 3

[5] Konstantinos Bousmalis, Giulia Vezzani, Dushyant Rao, Coline Devin, Alex X. Lee, Maria Bauza, Todor Davchev, Yuxiang Zhou, Agrim Gupta, Akhil Raju, Antoine Laurens, Claudio Fantacci, Valentin Dalibard, Martina Zambelli, Murilo Martins, Rugile Pevceviciute, Michiel Blokzijl, Misha Denil, Nathan Batchelor, Thomas Lampe, Emilio Parisotto, Konrad Żołna, Scott E. Reed, Sergio Gómez Colmenarejo, Jon Scholz, Abbas Abdolmaleki, Oliver Groth, Jean-Baptiste Regli, Oleg Sushkov, Thomas Rothörl, José Enrique Chen, Yusuf Aytar, Dave Barker, Joy Ortiz, Martin A. Riedmiller, Jost Tobias Springenberg, Raia Hadsell, Francesco Nori, and Nicolas Heess. Robocat: A self-improving generalist agent for robotic manipulation. *Transactions on Machine Learning Research*, 12, 2023. 3

[6] Joao Carvalho, An T. Le, Philipp Jahr, Qiao Sun, Julen Urain, Dorothea Koert, and Jan Peters. Grasp diffusion network: Learning grasp generators from partial point clouds with diffusion models in so(3)xr3, 2024. 2

[7] Luis Felipe Casas, Ninad Khargonkar, Balakrishnan Prabhakaran, and Yu Xiang. Multigrippergrasp: A dataset for robotic grasping from parallel jaw grippers to dexterous hands. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2024)*, pages 2978–2984, 2024. 3, 14

[8] Yu-Wei Chao, Wei Yang, Yu Xiang, Pavlo Molchanov, Ankur Handa, Jonathan Tremblay, Yashraj S Narang, Karl Van Wyk, Umar Iqbal, Stan Birchfield, et al. Dexycb: A benchmark for capturing hand grasping of objects. In *CVPR*, 2021. 14

[9] Tao Chen, Adithyavairavan Murali, and Abhinav Gupta. Hardware conditioned policies for multi-robot transfer learning. In *Advances in Neural Information Processing Systems 31 (NeurIPS 2018)*, pages 9355–9366, 2018. 3

[10] Murtaza Dalal, Min Liu, Chen Chen, Deepak Pathak, Jian Zhang, and Ruslan Salakhutdinov. Local policies enable zero-shot long-horizon manipulation. In *ICRA*, 2025. 2

[11] Abhay Deshpande, Yuquan Deng, Arijit Ray, Jordi Salvador, Winson Han, Jiafei Duan, Kuo-Hao Zeng, Yuke Zhu, Ranjay Krishna, and Rose Hendrix. Graspmolmo: Generalizable task-oriented grasping via large-scale synthetic data generation. In *CoRL*, 2025. 1, 2

[12] Coline Devin, Abhinav Gupta, Trevor Darrell, Pieter Abbeel, and Sergey Levine. Learning modular neural network policies for multi-task and multi-robot transfer. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA 2017)*, pages 2169–2176, 2017. 3

[13] Ria Doshi, Homer Walke, Oier Mees, Sudeep Dasari, and Sergey Levine. Scaling cross-embodied learning: One policy for manipulation, navigation, locomotion and aviation. In *Proceedings of the Conference on Robot Learning (CoRL 2024)*, 2024. 2, 3

[14] Clemens Eppner, Arsalan Mousavian, and Dieter Fox. ACRONYM: A large-scale grasp dataset based on simulation. In *Under Review at ICRA 2021*, 2020. 1, 2, 4, 13, 14

[15] Hao-Shu Fang, Chenxi Wang, Hongjie Fang, Minghao Gou, Jirong Liu, Hengxu Yan, Wenhai Liu, Yichen Xie, and Cewu Lu. Anygrasp: Robust and efficient grasp perception in spatial and temporal domains. *Transactions on Robotics*, 2023. 2, 8

[16] Xin Fei, Zhenyu Wang, Jiayu Luo, Chongkai Gao, Zhehao Cai, and Lin Shao. T(r,o) grasp: Efficient graph diffusion of robot-object spatial transformation for cross-embodiment dexterous grasping. arXiv preprint arXiv:2510.12724, 2025. 3

[17] Roman Freiberg, Alexander Qualmann, Ngo Anh Vien, and Gerhard Neumann. Diffusion for multi-embodiment grasping. In *Robotics and Automation Letters*, 2024. 2, 6, 7, 15

[18] Roman Freiberg, Alexander Qualmann, Ngo Anh Vien, and Gerhard Neumann. Diffusion for multi-embodiment grasping. *IEEE Robotics and Automation Letters*, PP(99):1–8, 2025. 3

[19] Abhinav Gupta, Adithyavairavan Murali, Dhiraj Gandhi, and Lerrel Pinto. Robot learning in homes: Improving generalization and reducing dataset bias. In *Advances in Neural Information Processing Systems 31 (NeurIPS 2018)*, 2018. 3

[20] Abhinav Gupta, Adithyavairavan Murali, Dhiraj Gandhi, and Lerrel Pinto. Robot learning in homes: Improving generalization and reducing dataset bias. *NeurIPS*, 2018. 8

[21] Shreyas Hampali, Mahdi Rad, Markus Oberweger, and Vincent Lepetit. Honnotate: A method for 3d annotation of hand and object poses. In *CVPR*, 2020. 14

[22] Ankur Handa, Karl Van Wyk, Wei Yang, Jacky Liang, Yu-Wei Chao, Qian Wan, Stan Birchfield, Nathan Ratliff, and Dieter Fox. Dexpilot: Vision-based teleoperation of dexterous robotic hand-arm system. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA 2020)*, pages 9164–9170, 2020. 3

[23] Edward S. Hu, Kun Huang, Oleh Rybkin, and Dinesh Jayaraman. Know thyself: Transferable visual control policies through robot-awareness. In *Proceedings of the International Conference on Learning Representations (ICLR 2022)*, 2022. 3

[24] Wenlong Huang, Igor Mordatch, and Deepak Pathak. One policy to control them all: Shared modular policies for agent-agnostic control. In *Proceedings of the International Conference on Machine Learning (ICML 2020)*, pages 4455–4464, 2020. 3

[25] Wenlong Huang, Chen Wang, Yunzhu Li, Ruohan Zhang, and Li Fei-Fei. Rekep: Spatio-temporal reasoning of relational keypoint constraints for robotic manipulation. In *CoRL*, 2024. 2, 5

[26] Abhishek Joshi, Beining Han, Jack Nugent, Max Gonzalez Saez-Diez, Yiming Zuo, Jonathan Liu, Hongyu Wen, Stamatis Alexandropoulos, Karhan Kayan, Anna Calveri, Tao Sun, Gaowen Liu, Yi Shao, Alexander Raistrick, and Jia Deng. Procedural generation of articulated simulation-ready assets, 2025. 4, 13

[27] Sergey Levine, Peter Pastor, Alex Krizhevsky, and Deirdre Quillen. Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection, 2016. 2

[28] Jinming Li, Yichen Zhu, Zhibin Tang, Junjie Wen, Minjie Zhu, Xiaoyu Liu, Chengmeng Li, Ran Cheng, Yaxin Peng, Yan Peng, and Feifei Feng. Coa-vla: Improving vision-language-action models via visual-textual chain-of-affordance. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV 2025)*, 2025. arXiv preprint arXiv:2412.20451. 2

[29] Puhao Li, Tengyu Liu, Yuyang Li, Yiran Geng, Yixin Zhu, Yaodong Yang, and Siyuan Huang. Gendexgrasp: Generalizable dexterous grasping. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA 2023)*, 2023. 3

[30] Hongzhuo Liang, Xiaojian Ma, Shuang Li, Michael Gorner, Song Tang, Bin Fang, Fuchun Sun, and Jianwei Zhang. Pointnetgpd: Detecting grasp configurations from point sets. In *ICRA*, 2019. 2

[31] Byeongdo Lim, Jongmin Kim, Jihwan Kim, Yonghyeon Lee, and Frank C Park. Equigraspflow: Se(3)-equivariant 6-dof grasp pose generative flows. In *CoRL*, 2024. 2

[32] Min Liu, Zherong Pan, Kai Xu, Kanishka Ganguly, and Dinesh Manocha. Deep differentiable grasp planner for high-dof grippers. *arXiv preprint arXiv:2002.01530*, 2020. 14

[33] Peiqi Liu, Yaswanth Orru, Chris Paxton, Nur Muhammad Mahi Shafiullah, and Lerrel Pinto. Ok-robot: What really matters in integrating open-knowledge models for robotics. *preprint arXiv:2401.12202*, 2024. 2

[34] Tyler Ga Wei Lum, Albert H. Li, Preston Culbertson, Krishnan Srinivasan, Aaron Ames, Mac Schwager, and Jeannette Bohg. Get a grip: Multi-finger grasp evaluation at scale enables robust sim-to-real transfer. In *CoRL*, 2024. 2

[35] Miles Macklin, Matthias Müller, Nuttapong Chentanez, and Tae-Yong Kim. Unified particle physics for real-time applications. *ACM Transactions on Graphics (TOG)*, 33(4):1–12, 2014. 14

[36] Jeffrey Mahler, Jacky Liang, Sherdil Niyaz, Michael Laskey, Richard Doan, Xinyu Liu, Juan Aparicio Ojea, and Ken Goldberg. Dex-net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics. *arXiv preprint arXiv:1703.09312*, 2017. 2

[37] Alexander Millane, Helen Oleynikova, Emilie Wirbel, Remo Steiner, Vikram Ramasamy, David Tingdahl, and Roland Siegwart. nvblox: Gpu-accelerated incremental signed distance field mapping. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2698–2705, 2024. 2

[38] Alexander Millane, Helen Oleynikova, Emilie Wirbel, Remo Steiner, Vikram Ramasamy, David Tingdahl, and Roland Siegwart. nvblox: Gpu-accelerated incremental signed distance field mapping, 2024. 16

[39] Douglas Morrison, Peter Corke, and Jürgen Leitner. Egad! an evolved grasping analysis dataset for diversity and reproducibility in robotic manipulation. *IEEE Robotics and Automation Letters*, 5(3):4368–4375, 2020. 14

[40] Arsalan Mousavian, Clemens Eppner, and Dieter Fox. 6-dof graspnet: Variational grasp generation for object manipulation. In *ICCV*, 2019. 1, 2

[41] Adithyavairavan Murali, Weiyu Liu, Kenneth Marino, Sonia Chernova, and Abhinav Gupta. Same object, different grasps: Data and semantic knowledge for task-oriented grasping. In *CoRL*, 2020. 2, 5

[42] Adithyavairavan Murali, Arsalan Mousavian, Clemens Eppner, Chris Paxton, and Dieter Fox. 6-dof grasping for target-driven object manipulation in clutter. In *ICRA*, 2020. 2

[43] Adithyavairavan Murali, Balakumar Sundaralingam, Yu-Wei Chao, Jun Yamada, Wentao Yuan, Mark Carlson, Fabio Ramos, Stan Birchfield, Dieter Fox, and Clemens Eppner. Graspgen: A diffusion-based framework for 6-dof grasping with on-generator training. *arXiv preprint arXiv:2507.13097*, 2025. 1, 2, 3, 4, 5, 6, 8, 13, 14, 15

[44] Rhys Newbury, Morris Gu, Lachlan Chumbley, Arsalan Mousavian, Clemens Eppner, Jürgen Leitner, Jeannette Bohg, Antonio Morales, Tamim Asfour, Danica Kragic, et al. Deep learning approaches to grasp synthesis: A review. *IEEE Transactions on Robotics*, 2023. 2

[45] NVIDIA. Nvidia isaac sim, 2023. 14

[46] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017. 3, 6, 14

[47] Yuzhe Qin, Wei Yang, Binghao Huang, Karl Van Wyk, Hao Su, Xiaolong Wang, Yu-Wei Chao, and Dieter Fox. Anyteleop: A general vision-based dexterous robot arm-hand teleoperation system. arXiv preprint arXiv:2307.04577v1, 2023. 3

[48] Alexander Raistrick, Lahav Lipson, Zeyu Ma, Lingjie Mei, Mingzhe Wang, Yiming Zuo, Karhan Kayan, Hongyu Wen, Beining Han, Yihan Wang, et al. Infinite photorealistic worlds using procedural generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12630–12641, 2023. 13

[49] Alexander Raistrick, Mei Lingjie, Kaan Kayan Karhan, Yan David, Zuo Yiming, Han Beining, Wen Hongyu, Parakh Meenal, Stamatis Alexandropoulos, Lipson Lahav, Ma Zeyu, and Deng Jia. Infinite indoors: Photorealistic indoor scenes using procedural generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024. 13

[50] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollár, and Christoph Feichtenhofer. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024. 2, 16

[51] Gaurav Salhotra, I-Chun Arthur Liu, and Gaurav S. Sukhatme. Learning robot manipulation from cross-morphology demonstration. In *Proceedings of the 7th Annual Conference on Robot Learning (CoRL 2023)*, 2023. 3

[52] Lin Shao, Fabio Ferreira, Mikael Jorda, Varun Nambiar, Jianlan Luo, Eugen Solowjow, Juan Aparicio Ojea, Oussama Khatib, and Jeannette Bohg. Unigrasp: Learning a unified model to grasp with multifingered robotic hands. *IEEE Robotics and Automation Letters*, 5(2):2286–2293, 2020. 3, 6, 7, 14

[53] Junyao Shi, Rujia Yang, Kaitian Chao, Selina Bingqing Wan, Yifei Shao, Jiahui Lei, Jianing Qian, Long Le, Pratik Chaudhari, Kostas Daniilidis, Chuan Wen, and Dinesh Jayaraman. Maestro: Orchestrating robotics modules with vision-language models for zero-shot generalist robots. *arXiv preprint arXiv:2511.00917*, 2025. 2

[54] Pinhao Song, Pengteng Li, and Renaud Detry. Implicit grasp diffusion: Bridging the gap between dense prediction and sampling-based grasping. In *CoRL*, 2024. 2

[55] Balakumar Sundaralingam, Siva Kumar Sastry Hari, Adam Fishman, Caelan Garrett, Karl Van Wyk, Valts Blukis, Alexander Millane, Helen Oleynikova, Ankur Handa, Fabio Ramos, et al. Curobo: Parallelized collision-free robot motion generation. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 8112–8119. IEEE, 2023. 2

[56] Balakumar Sundaralingam, Siva Kumar Sastry Hari, Adam Fishman, Caelan Garrett, Karl Van Wyk, Valts Blukis, Alexander Millane, Helen Oleynikova, Ankur Handa, Fabio Ramos, Nathan Ratliff, and Dieter Fox. curobo: Parallelized collision-free minimum-jerk robot motion generation. In *ICRA*, 2023. 8, 16

[57] Martin Sundermeyer, Arsalan Mousavian, Rudolph Triebel, and Dieter Fox. Contact-graspnet: Efficient 6-dof grasp generation in cluttered scenes. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 13438–13444. IEEE, 2021. 2, 4

[58] Chao Tang, Dehao Huang, Wenqi Ge, Weiyu Liu, and Hong Zhang. Graspgpt: Leveraging semantic knowledge from a large language model for task-oriented grasping. *RAL*, 2023. 2

[59] Andreas ten Pas, Marcus Gualtieri, Kate Saenko, and Robert Platt. Grasp pose detection in point clouds. *The International Journal of Robotics Research*, 36(13–14):1455–1473, 2017. 2

[60] Josh Tobin, Lukas Biewald, Rocky Duan, Marcin Andrychowicz, Ankur Handa, Vikash Kumar, Bob McGrew, Alex Ray, Jonas Schneider, Peter Welinder, Wojciech Zaremba, and Pieter Abbeel. Domain randomization and generative models for robotic grasping. In *IROS*, 2018. 2

[61] Dylan Turpin, Tao Zhong, Shutong Zhang, Guanglei Zhu, Jingzhou Liu, Ritvik Singh, Eric Heiden, Miles Macklin, Stavros Tsogkas, Sven Dickinson, et al. Fast-grasp'd: Dexterous multi-finger grasp generation through differentiable simulation. *arXiv:2306.08132*, 2023. 14

[62] Julen Urain, Niklas Funk, Jan Peters, and Georgia Chalvatzaki. Se(3)-diffusionfields: Learning smooth cost functions for joint grasp and motion optimization through diffusion. *ICRA*, 2023. 1, 2

[63] Ruicheng Wang, Jialiang Zhang, Jiayi Chen, Yinzhen Xu, Puhao Li, Tengyu Liu, and He Wang. Dexgraspnet: A large-scale robotic dexterous grasp dataset for general objects based on simulation. In *ICRA*, 2023. 14

[64] Xianli Wang and Qingsong Xu. Transferring grasping across grippers: Learning–optimization hybrid framework for generalized planar grasp generation. *IEEE Transactions on Robotics*, 2024. 3

[65] Zi Wang, Caelan Reed Garrett, Leslie Pack Kaelbling, and Tomas Lozano-Pérez. Learning compositional models of robot skills for task and motion planning. *The International Journal of Robotics Research*, 2020. 2

[66] Xinyue Wei, Minghua Liu, Zhan Ling, and Hao Su. Approximate convex decomposition for 3d meshes with collision-aware concavity and tree search. *ACM Transactions on Graphics (TOG)*, 41(4):1–18, 2022. 13

[67] Zhenyu Wei, Zhixuan Xu, Jingxiang Guo, Yiwen Hou, Chongkai Gao, Zhehao Cai, Jiayu Luo, and Lin Shao. D(r,o) grasp: A unified representation of robot and object interaction for cross-embodiment dexterous grasping. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA 2025)*, 2025. 3

[68] Bowen Wen, Wei Yang, Jan Kautz, and Stan Birchfield. Foundationpose: Unified 6d pose estimation and tracking of novel objects. *arXiv preprint arXiv:2312.08344*, 2023. 2, 16

[69] Bowen Wen, Matthew Trepte, Joseph Aribido, Jan Kautz, Orazio Gallo, and Stan Birchfield. Foundationstereo: Zero-shot stereo matching. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 5249–5260, 2025. 1, 16

[70] Zehang Weng, Haofei Lu, Danica Kragic, and Jens Lundell. Dexdiffuser: Generating dexterous grasps with diffusion models. *Robotics and Automation Letters*, 2024. 2

[71] Xiaoyang Wu, Li Jiang, Peng-Shuai Wang, Zhijian Liu, Xihui Liu, Yu Qiao, Wanli Ouyang, Tong He, and Hengshuang Zhao. Point transformer v3: Simpler, faster, stronger. In *CVPR*, 2024. 3

[72] Yueh-Hua Wu, Jiashun Wang, and Xiaolong Wang. Learning generalizable dexterous manipulation from human grasp affordance. In *CoRL*, 2023. 2

[73] Open X-Team. Open x-embodiment: Robotic learning datasets and rt-x models. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA 2024)*, pages 6892–6903, 2024. 2, 3

[74] Pengwei Xie, Siang Chen, Wei Tang, Dingchang Hu, Wenming Yang, and Guijin Wang. Rethinking 6-dof grasp detection: A flexible framework for high-quality grasping. *Pattern Recognition*, 2025. 2

[75] Zhenjia Xu, Beichun Qi, Shubham Agrawal, and Shuran Song. Adagrasp: Learning an adaptive gripper-aware grasping policy. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4620–4626. IEEE, 2021. 3, 6, 7, 15

[76] Jun Yamada, Adithyavairavan Murali, Ajay Mandlekar, Clemens Eppner, Ingmar Posner, and Balakumar Sundaralingam. Grasp-mpc: Closed-loop visual grasping via value-guided model predictive control. arXiv preprint arXiv:2509.06201v1, 2025. 5

[77] Wenhao Yu, Jie Tan, C. Karen Liu, and Greg Turk. Preparing for the unknown: Learning a universal policy with online system identification. In *Proceedings of the Robotics: Science and Systems (RSS 2017)*, 2017. 3

[78] Wentao Yuan, Adithyavairavan Murali, and Arsalan Mousavian. M2t2: Multi-task masked transformer for object-centric pick and place. In *7th Annual Conference on Robot Learning*. https://openreview. net/forum? id= 6zGpfOBImD, 2023. 2, 4

[79] Wentao Yuan, Jiafei Duan, Valts Blukis, Wilbert Pumacay, Ranjay Krishna, Adithyavairavan Murali, Arsalan Mousavian, and Dieter Fox. Robopoint: A vision-language model for spatial affordance prediction for robotics. In *Proceedings of the 8th Conference on Robot Learning (CoRL 2024)*, pages 4005–4020. PMLR, 2025. 2

[80] Kevin Zakka, Andy Zeng, Pete Florence, Jonathan Tompson, Jeannette Bohg, and Debidatta Dwibedi. Xirl: Cross-embodiment inverse reinforcement learning. In *Proceedings of the Conference on Robot Learning (CoRL 2021)*, pages 537–546, 2021. 3