

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

Supplementary Material

8. Procedural Gripper Generator

We design procedural gripper generators with Infinigen Articulated [26]. Infinigen Articulated is the extension of Infinigen [48, 49], which supports the design of procedural articulated objects with the Blender Geometry Node. For all Infinigen objects, each generator consists of randomized mathematical functions, without using any existing data / meshes. Here, we design a generator for each category of parallel grippers, 2-finger revolute grippers, and 3-finger high-dof grippers.

2-Finger Parallel Gripper. Each parallel gripper consists of 3 links, i.e., gripper base, left finger, right finger, and 2 prismatic joints, i.e., the left finger joint and the right finger joint. The gripper base consists of a cube and a cylinder with a random value to control the ratio. We also randomize the base geometry in terms of its height, width, and depth. We create fingers as cubic objects and are modified based on the ratio of fingertip width to the finger bottom width, and its tilting ratio. In half of all samples, we add additional meshes of square, round, or triangle cylinders at the fingertip.

2-Finger Revolute Gripper. Each 2-finger revolute gripper consists of 5 links, i.e., gripper base, left mid finger, left top finger, right mid finger, right top finger, and 4 revolute joints, i.e., left mid finger joint, left top finger joints, right mid finger joint, and right top finger joint. The gripper base is randomized in dimension and in the ratio between the base top and base bottom. The mid finger links and the top finger links are cubic objects. For mid fingers, we randomly add an outer finger like in Robotiq-2F and OnRobot-RG grippers. For the top fingers, we randomly add round / square cylinders at fingertips. There are two modes for gripper closing motion. In the first mode, the right and left top finger links are always parallel to each other. Thus, the top finger joint and the mid finger joint rotate in a ratio of $1 : -1$. In the second mode, all top finger joints and the mid finger joints rotate in a ratio of $1 : 1$, where the gripper fingers will close like a pinch gripper.

3-Finger High-DOF Grippers. Each 3-finger high-dof gripper consists of the gripper base and 3 2-joint / 3-joint fingers. We create a cubic gripper base with randomized dimensions and wrist-to-palm ratio. We attach two fingers at the top of the palm and one finger at the center of the wrist. All fingers are stretching in x-axis when it is open. The total DOFs with 2-joint fingers is 6 and the total DOF with 3-joint fingers is 9. All finger links are cubic objects with random width, depth, and height. Moreover, we randomly change the orientation of the two fingers at the top of

the palm around the z-axis to mimic the potential side rotation that real 3-finger hands have (e.g., Robotiq-3F). For the gripper closing motion, all joints follow a linearly interpolated trajectory from the fully open state to the fully closed state.

Figure 9 shows examples of our generated grippers together with its closing motion. In addition to the gripper’s geometry, we also export the Swept Volume heuristic of the fully open to the half open state. The dimensions and translations from the base are automatically computed, given the configuration parameters of each gripper instance. We use COACD [66] to compute the convex decomposition of each visual mesh for the collision mesh in Isaac-sim simulation.

9. GraspGen-X Dataset

Our cross-embodiment dataset features the largest simulation dataset for 6-DOF multi-embodiment grasping. We will release the dataset in GraspGen-X training and evaluation. We categorize the dataset into GraspGen-X-Procedural and GraspGen-X-Real to distinguish the grippers. In Table 4, we summarize and compare with other 6-DOF grasping datasets. Ours GraspGen-X dataset is 8X larger than the second largest dataset. For the generator training dataset, it consists of $2000(\# \text{ of sample grasps}) \times 25(\# \text{ of procedural grippers}) \times 3500(\# \text{ of training objects}) \approx 175M$ sampled grasps and grasp labels evaluated with the ACRONYM pipeline [14] in Isaac-Sim simulation. Moreover, we also include a dataset for test objects of the grasps and labels sampled by $5000(\# \text{ of sample grasps}) \times 20(\# \text{ of real grippers}) \times 453(\# \text{ of test objects}) \approx 45M$. For the discriminator training dataset, we sampled on-generator grasps [43] with the generator and evaluated with the same labeling pipeline. The discriminator training dataset consists of another $2000(\# \text{ of sample grasps}) \times 25(\# \text{ of procedural grippers}) \times 3500(\# \text{ of training objects}) \approx 175M$ grasp samples and labels. In all, we have generated the largest multi-gripper 6-DOF grasping dataset. In total, we collect $350(=175+175)M$ grasps for training and our dataset contains a total of $395(=350+45)M$ grasps.

10. Simulation Experiments

In this section, we provide the details and additional results of simulation experiments.

10.1. Experiment Setup

Train / Test Real Grippers. Figure 10 shows the split of 20 grippers used in our experiments. We try to equally di-

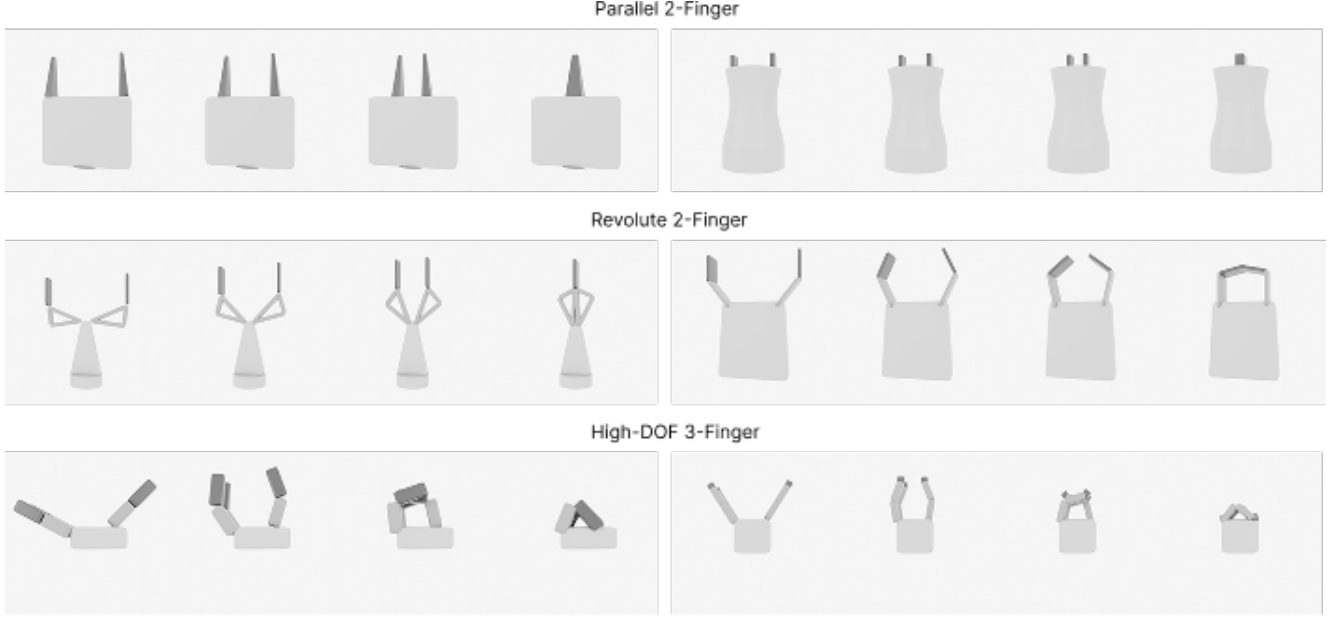


Figure 9. Examples of procedurally generated gripper of each category: parallel 2-finger, revolute 2-finger and high-dof 3-finger. Each row contains 2 randomly generated instances. We show 4 steps in gripper closing. The first one shows the state when the gripper is fully open and last one shows the state when the gripper is fully closed.

Dataset	Year	#Grippers	#Objects	#Grasps	Grasp Label	Synthesis Method
HO-3D [21]	2020	1 (Human hand)	10	78K	Only +ve	Human Demo
EGAD [39]	2020	1 (2-finger)	2,331	233K	Only +ve	Evolutionary Algorithm
DDG [32]	2020	1 (5-finger)	500	50K	Only +ve	GraspIt + modified Q1
DexYCB [8]	2021	1 (Human hand)	20	582K	Only +ve	Human Demo
Acronym [14]	2021	1 (2-finger)	8,872	17.7M	+ve & -ve	Flex [35]
UniGrasp [52]	2020	12 (2 & 3finger)	1000	2M+	Only +ve	Contact Points Network + FastGrasp
DexGraspNet [63]	2023	1 (5-finger)	5,355	1.3M	Only +ve	Differentiable grasping
Fast-Grasp'D [61]	2023	3 (3-5 finger)	2,350	1M	Only +ve	Differentiable grasping
MultiGripperGrasp [7]	2024	11 (2-5 finger & Human)	345	30.4M	Ranked	GraspIt + Isaac Sim [45]
GraspGen [43]	2025	3 (2-finger & Suction)	8,515	53.1M	+ve & -ve	Sampling + Isaac Sim [45]
GraspGen-X-Procedural	2025	25 (3-finger, 2-finger procedural grippers)	3500	350M	+ve & -ve	Sampling + Isaac Sim [45]
GraspGen-X-Real	2025	20 (3-finger, 2-finger real-world grippers)	453	45M	+ve & -ve	Sampling + Isaac Sim [45]

Table 4. Comparison of GraspGen-X with existing grasping datasets. The table is adapted from [43].

vide the set based on morphology similarity. We follow the same simulation setup for parallel grippers and 2-finger revolute grippers in ACRONYM [14]. For 3-finger high-dof, we use PD controller to track a target joint trajectories. The target trajectory is a linear interpolation between the fully open state and fully closed state of each 3-finger gripper.

GraspGen-X Training. Our GraspGen-X extends GraspGen with the additional 512-dim gripper embedding, concatenated with the 512-dim object embedding. We use the PointNet++ [46] encoder for object embedding for training efficiency. For all other hyper-parameters, we follow the exact same setup in the GraspGen [43] codebase.

10.2. Zero-shot Evaluation

We use mAUC, i.e., the average AUC of the precision-coverage curve of all test grippers and test objects, as our primary metric. For each gripper and each object, we plot

the precision-coverage with the ranked 2k grasps iteratively.

Table 5 shows the mAUC of each test gripper. We find that GraspGen-X outperforms well in almost all grippers except OnRobot RG2, which falls significantly behind RTG. This suggests that there is still room for improvement in our end-to-end cross-embodiment model. We hypothesize that the OnRobot RG2’s morphology may not be well covered with our procedural grippers, but this can be mitigated by further training on more procedural grippers.

Nevertheless, our GraspGen-X still achieves the SOTA performance across each category. Figure 13 visualizes the 5 generated grasps with GraspGen-X for all test grippers.

10.3. Supervised Finetuning Adaptation

We adopt the same metrics in generator training as those used in GraspGen [43]. For a given gripper embodiment \mathcal{E} and object \mathcal{O} , we have a set of positive SE(3) grasp poses



Figure 10. The set of training (upper row) and test (bottom row) grippers used in our experiments. We create a balanced split of all 20 grippers.

Table 5. Zero-shot performance on novel test grippers and novel test objects. We report the mAUC of each gripper.

	GraspGen-DTR	GraspGen-RTG	GraspGen-X
ARX X5	0.270	0.532	0.620
Galaxea G1	0.319	0.537	0.663
Franka UMI	0.267	0.256	0.441
Tesollo Delto2F	0.002	0.134	0.285
Avg. Parallel 2F	0.215	0.365	0.502
Sake EZgripper	0.064	0.420	0.522
Robotiq 2F140	0.051	0.303	0.469
OnRobot RG2	0.002	0.241	0.136
XArm Hand	0.015	0.551	0.525
Avg. Revolute 2F	0.033	0.379	0.413
Unitree G1	0.269	0.662	0.818
Robotiq 3F	0.002	0.343	0.579
Avg. High-DOF 3F	0.136	0.503	0.699

$G_{\mathcal{E},\mathcal{O}}$ and the corresponding predictions from GraspGen-X $\hat{G}_{\mathcal{E},\mathcal{O}}$. For each grasp pose $\hat{g} \in \hat{G}_{\mathcal{E},\mathcal{O}}$, we find the corresponding nearest neighbor grasp g in the ground truth set $G_{\mathcal{E},\mathcal{O}}$ based on the cost function defined in [43]. To capture both the accuracy of these grasp predictions, we separately compute the translation error (L2 distance in m) and rotation error (the geodesic error in radians) between \hat{g} and g , where $g = (R, t)$ and $\hat{g} = (\hat{R}, \hat{t})$. Similarly, to measure how well the generator captures the ground truth distribution, recall refers to the ratio of grasps in $G_{\mathcal{E},\mathcal{O}}$ that have a nearest neighbour grasp in $\hat{G}_{\mathcal{E},\mathcal{O}}$ within a threshold of $t_{dist} = 1cm$.

Translation Error (m):

$$\mathcal{E}_{trans} = \|t - \hat{t}\|_2 \quad (1)$$

Rotation Error (radians):

$$\mathcal{E}_{rot} = \arccos\left(\frac{\text{tr}(R^\top \hat{R}) - 1}{2}\right) \quad (2)$$

Recall:

$$\text{Recall} = \frac{\left| \left\{ g \in G_{\mathcal{E},\mathcal{O}} \mid \exists \hat{g} \in \hat{G}_{\mathcal{E},\mathcal{O}} \text{ s.t. } \|t - \hat{t}\|_2 < 0.01 \right\} \right|}{|G_{\mathcal{E},\mathcal{O}}|} \quad (3)$$

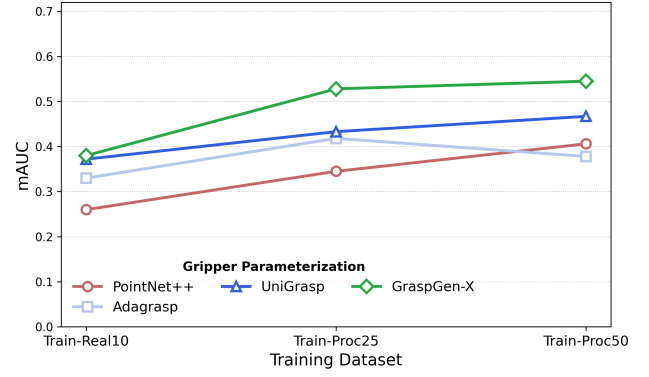


Figure 11. Results comparing GraspGen-X and baselines’ gripper encoding representations. We repeat the experiments in Sec 5.3 on different sets of grippers.

10.4. Gripper Encoding Representation

We follow the experiment in Figure 8 to compare different gripper encoding representations used in previous work. The results show that GraspGen-X gripper encoding performs better than baselines in different sets of training grippers. Moreover, training with 50 procedural grippers improves training with 25 procedural grippers also in the PointNet++ [17] and AdaGrasp [75] (TSDF) representations.

10.5. Ablation Study

Figure 12 shows the results of ablation in the parameterization of the gripper and the distribution of the training grippers. We plot the results of average AUC over grippers of each category, including parallel 2-finger grippers, revolute 2-finger grippers, and high-dof 3-finger grippers.

Comparing FullyOpenOnly (6-dim) and ours (12-dim), we find that the gap primarily comes from the revolute 2-finger grippers. Many grippers in this category, e.g., Robotiq-2F140 and OnRobot-RG2, will rotate the fingers so that the fingertip will move forward along the z-axis during closing. Consequently, it is important to capture the information of the closing motion when computing the grasp pose. However, FullyOpenOnly does not contain this infor-

mation in its parameterization while ours Swept Volume of fully open and half open states provides.

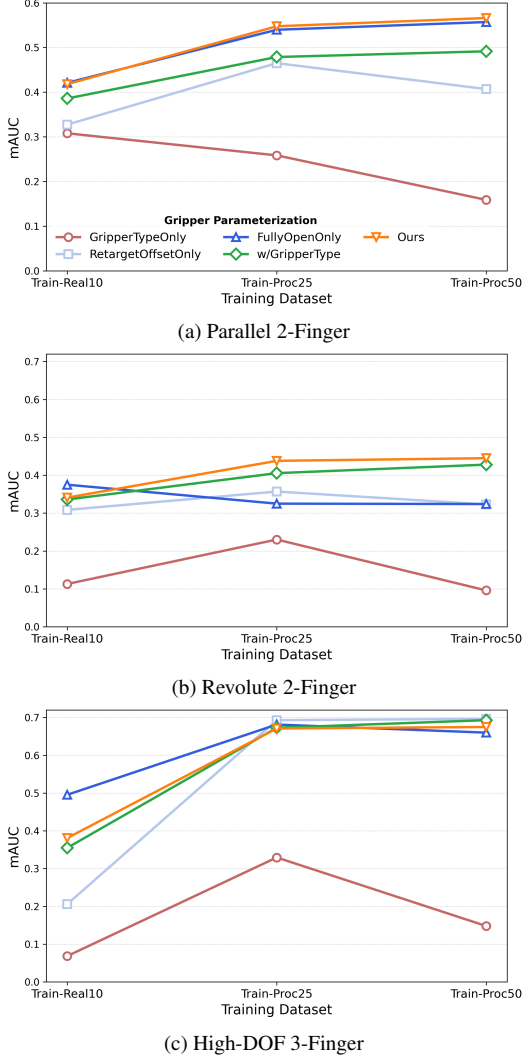


Figure 12. Ablation on gripper parameterization and gripper training dataset. We show the results of each category: (a) parallel 2-finger gripper, (b) revolute 2-finger gripper, (c) high-dof 3-finger gripper.

11. Real Robot Experiment

Industrial Manipulator: Our UR10 setup consists of a single extrinsically calibrated RealSense D435 RGB-D camera overlooking the scene. We use SAM2 [50] running on a 6000 Ada GPU for segmentation, as well as FoundationStereo [69] for depth estimation. The input to GraspGen-X is a partial pointcloud segmented on the target object. cuRobo [56] and NVBlox [38] running on a Jetson is used for collision-free motion planning. All models returned a set of predicted grasps and confidence scores. We use the top-100 grasps as pose targets for the motion plan-

ner, which filters out grasps that are in collision or do not have an IK solution.

Low-Cost Arm: We use AgileX Piper robot with its parallel 2-finger hand. We mount a ZED2 camera on the arm’s end-effector. In this experiment, we assume access to the object model including a YCB Mustard bottle and a cube. We use FoundationPose [68] to estimate an accurate 6D object pose. The input to GraspGen-X is the complete pointcloud under the 6D pose. The target grasp pose is then transformed into the base frame of the robot. We then compute an IK solution in Pybullet and use a linearly interpolated trajectory to command the arm to the target joint configuration to execute the grasp.

Humanoid Manipulator: Additionally, we also validate GraspGen-X on a Unitree G1 3-finger 7-DOF hand and a stereo-camera mounted on the chest of G1 robot (Figure 1). Similarly to the low-cost arm experiment, we infer the grasp poses with the full object pointcloud and generate the grasping motion with a linearly interpolated trajectory from the initial joint configurations to the target configuration, which is computed via IK.

We test GraspGen-X with the YCB mustard bottle with randomly placed stable poses. We observe that it achieves 100% success rate out of 5 trials.

12. Discussion

Non-graspable Objects. Figure 15 visualizes the model’s prediction of a non-graspable object. We choose a non graspable cube (12cm edge) with two Parallel045 2F grippers (Gallaxea G1 and Tesollo-DG2F) with a maximal opening width smaller than the cube size. The figure shows the top-ranked grasps of GraspGen-X. For all such grasps, the discriminator has a low confidence score.

Antipodal Sampling of 3-Finger Grippers. 3-finger grippers are actuated to follow a predefined finger closure trajectory with PD control (Figure 14) to grip the object. We assume that the thumb finger is stretched in the $+x$ direction and the other two fingers are stretched in the $-x$ direction. For antipodal sampling, we align $+x$ at the tip of the thumb with the normal of the contact point sampled on the object. Then, distance offset and pose orientation are randomized to search for positive grasps. We acknowledge that the distribution of end grasps is more limited in variation than the more general dexterous hand grasping problem, which is future work.

Sweep Volume Representation. Our swept-volume representation omits some aspects of the gripper, such as its contact friction and kinematics. For example, in 3-finger grippers, it fails to reflect the asymmetric contact pattern of the $+x$ thumb finger and two x fingers, which may be crucial to grasp small objects. Despite the loss of information, the representation is an efficient representation for cross-embodiment 6-DOF grasping compared to prior methods

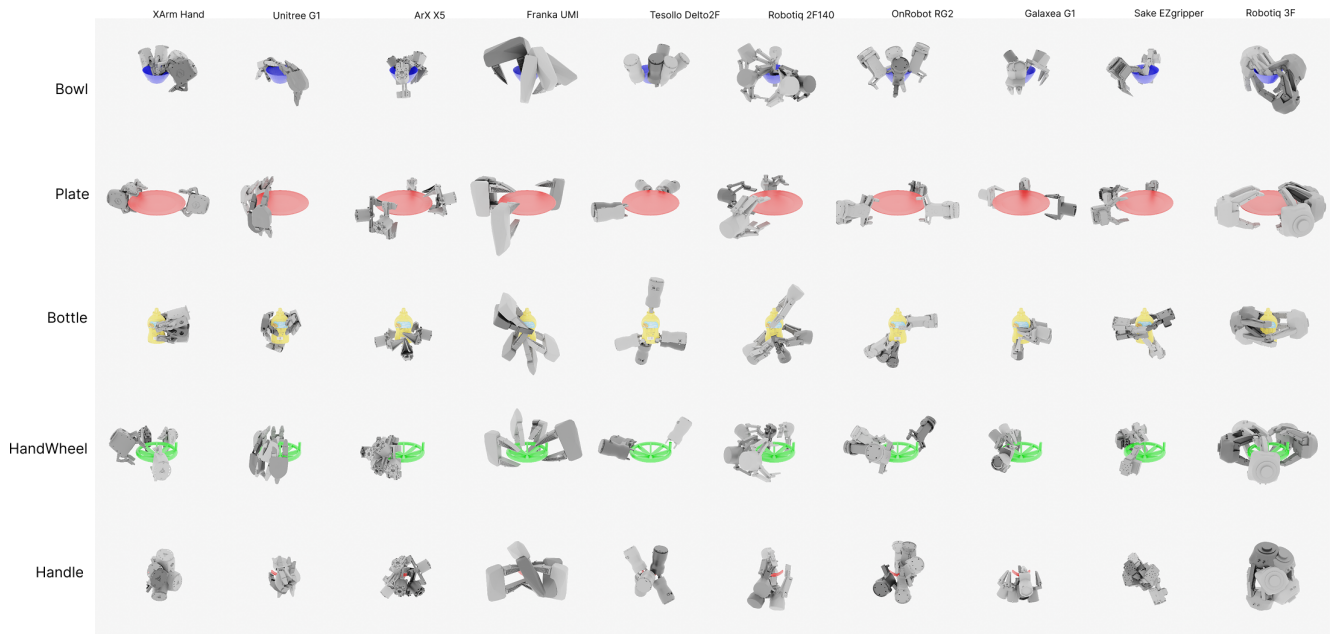


Figure 13. Visualization of GraspGen-X generated grasps with on all 10 test grippers and on 5 novel objects.

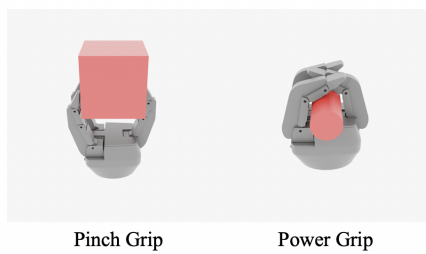


Figure 14. Visualization of sampled antipodal grasps of a 3-finger gripper. It can be either power grip or precision grip.

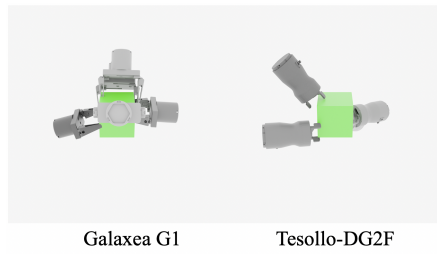


Figure 15. Visualization of GraspGen-X generated grasps with on all 10 test grippers and on 5 novel objects.

and it works well on 3-finger and even 5-finger grippers.