GraspFactory: A Large Object-Centric Grasping Dataset

Srinidhi Kalgundi Srinivas[†], **Yash Shukla**[§], **Adam Arnold**[†], **Sachin Chitta**[†]
[†]Autodesk Research
[§]Work done as an intern at Autodesk Research

Abstract: Robotic grasping is a crucial task in industrial automation, where robots are increasingly expected to handle a wide range of objects. However, a significant challenge arises when robot grasping models trained on limited datasets encounter novel objects. In real-world environments such as warehouses or manufacturing plants, the diversity of objects can be vast, and grasping models need to generalize to this diversity. Training large, generalizable robot-grasping models requires geometrically diverse datasets. In this paper, we introduce GraspFactory, a dataset containing over 109 million 6-DoF grasps collectively for the Franka Panda (with 14,690 objects) and Robotiq 2F-85 grippers (with 33,710 objects). GraspFactory is designed for training data-intensive models, and we demonstrate the generalization capabilities of one such model trained on a subset of GraspFactory in both simulated and real-world settings. The dataset and tools are made available for download at **graspfactory.github.io**.

Keywords: CAD, Grasp dataset, Learning

1 Introduction

Large datasets have been a major contributor to the success of AI models. The fields of Computer Vision and Natural Language Processing have seen tremendous progress due to the presence of internet-scale datasets like ImageNet [1] and Laion-5b [2]. Models such as Chat-GPT [3] and Dall-E[4] demonstrate strong generalization capabilities for tasks that were not explicitly represented in their training data, thanks to the use of diverse training datasets and large-scale transformer-based architectures. Similar efforts have been undertaken in robotics to collect large datasets, such as Open X-Embodiment [5] and DROID [6]. These datasets focus on end-to-end training of robots but there is still a need for task-specific datasets. Robot grasping is one such task, and a generalized grasping model remains elusive, in part due to the lack of geometrically diverse objects in existing datasets. In this work, we present an object-centric grasping dataset that offers greater geometric diversity compared to existing datasets.

Currently, object-centric grasping datasets [7, 8, 9] and scene-based grasping datasets [10, 11, 12] are mostly geared toward domestic robotics applications. These datasets have been used to train robot grasping models such as [13, 14, 15, 16]. The grasping datasets are generated using 3D CAD models[1] from 3D datasets such as Shapenet [17], YCB [18], Objaverse [19] and the Princeton Shape Benchmark [20]. These datasets, however, contain objects of low geometric diversity, as they contain only a small number of semantic classes [8]. Some of the recent advancements in 3D generative models, however, are fueled by larger 3D datasets like those presented in [21, 19, 17]. We leverage one such 3D dataset, *ABC-Dataset* [21], that contains *1M+* high quality geometric models.

We introduce GraspFactory, a large-scale dataset of 6-DoF parallel-jaw grasps generated in simulation. The dataset provides two-fingered grasps for the Franka Panda and Robotiq 2F-85 grippers. We utilize a scalable robotics simulation and synthetic data generation tool to annotate the objects with

[1]We use the term "CAD models" in this paper to specifically refer to triangular meshes.

6-DoF grasps. Further, we train an existing diffusion-based grasping model, SE(3)-DiffusionFields [14] on the Franka Panda subset, and evaluate the model's generalization capabilities on unseen objects. To the best of our knowledge, this is the largest object-centric grasping dataset containing 6-DoF, parallel-jaw grasps for geometrically diverse 3D data.

Our contributions are as follows:

- We present GraspFactory, a large-scale, object-centric dataset of 6-DoF parallel-jaw grasps with corresponding gripper widths, comprising over 109 million grasps in total. The dataset includes grasps for 33,710 objects randomly selected from the ABC dataset [21] for the Robotiq 2F-85 gripper, and 14,690 objects for the Franka Panda gripper, selected as a subset of the Robotiq object set. As part of ongoing work, we plan to extend the dataset with grasps for additional objects from [21].
- We train a diffusion-based grasp generative model [14] on the Franka Panda subset of GraspFactory, and demonstrate that training on geometrically diverse data improves generalization in both simulation and real-world experiments.

The rest of the paper is organized as follows: In Section 2, we review prior work. In Section 3, we present the method used for generating the dataset. Section 4 describes the experimental setup, both in simulation and real, and the results from training a model with GraspFactory.

2 Related Work

2.1 Existing datasets

Robot grasping datasets are generally collected through one of the following three methods: simulation, human annotation or human teleoperation. Datasets collected through simulation offer scalability, but they require highly accurate physics simulators to overcome the sim-to-real gap. Simulators built on physics engines such as PhysX [22] and Bullet [23] offer some level of physical realism. These simulators require CAD models of objects for scene generation. There are several datasets containing 3D CAD models [21, 17, 19, 24] that are currently used to train 3D generative models [25, 26], 3D segmentation and classification models [27], and normal estimation methods [21].

Prior work [7, 28, 8] uses physics simulators and 3D datasets to generate grasping datasets. Kappler et al. [29] show that physics simulation can be used to predict successful grasps. Eppner et al. [7] use ShapeNet [17], Mahler et al. [28] use 3D-Net [30] and the KIT object database [31] while Murali et al. [9] use the Objaverse dataset [19] to generate grasping datasets. However, the objects in these datasets are primarily used for 3D object recognition tasks containing only a small number of semantic classes, resulting in low geometric diversity within the datasets [8]. Morrison et al. [8] propose a method to use evolutionary algorithms to generate objects and grasps of varying complexities, but the generated objects are not representative of those encountered in the real-world.

There are also several datasets in the literature that contain grasps for images and point clouds of scenes with multiple objects. Jiang et al. [10] and Depierre et al. [11] propose datasets containing planar grasps in the image frame. Fang et al. [12] present a dataset that contains over 1.1 billion grasp annotations for cluttered, complex scenes. Zhang et al. [32] expand the Visual Manipulation Relationship Dataset [33] containing planar grasps for $15k+$ objects. Despite the large number of grasp annotations in them, the planar nature of the grasps in these datasets limits their utility for tasks like bin-picking, where the objects may not be presented on a plane. GraspFactory uses [21] to generate, to the best of our knowledge, the largest object-centric 6-DoF grasping dataset containing objects of varied geometries.

2.2 Grasp Sampling Methods

Given an object CAD model, one desired behavior of a grasp, is to maintain *force closure* in the presence of disturbing forces and moments while respecting velocity and kinematic constraints of the

manipulator [34]. A number of methods have been proposed in the literature to sample robust grasps from a CAD model. Gatrell [35] uses the information from CAD models such as polygons, edges and vertices to generate grasps using Extended Gaussian Images [36] that achieve force closure. Other grasp sampling methods using CAD models include uniform samplers [37], approach based samplers [38] and antipodal-based samplers [39, 28]. Zhu and Wang [40] and Han et al. [41] propose analytical approaches to test force-closure condition for the sampled grasps. Eppner et al. [7] present a two-fingered grasp dataset using objects from [17]. They evaluate the sampled grasps using the FleX [42] physics simulator. Similar to [7], we use a physics simulator to evaluate the robustness of sampled grasps under external wrenches. Our work uses the antipodal sampling method [43] to sample 6-DoF parallel-jaw grasps, and uses the Isaac Sim simulator [44] for evaluating grasp robustness.

2.3 Learning-based grasping

A number of deep learning-based methods have been proposed to estimate grasps from an object's CAD model and also directly from the scene point cloud. Newbury et al. [45] present a comprehensive survey on different methods and datasets used in the literature. Grasps are broadly classified into 4-DoF and 6-DoF, where 4-DoF planar grasp estimation methods involve determining the *x, y, z and* θ parameters, where, *x, y, z* are the 3D spatial position and θ is the rotation about the z-axis of the gripper and 6-DoF grasp estimation methods involve determining the full 6D pose of the gripper for a suitable grasp. Morrison et al. [46] proposed a convolution-based neural network for detecting suitable grasps from a depth image while also considering the width of the gripper as a parameter. Mousavian et al. [47] propose a Variational Auto-Encoder [48] based model and sample grasps through the latent space of the model. Sundermeyer et al. [15] propose an end-to-end network to sample grasps from a depth image of the scene. Barad et al. [16], Urain et al. [14], and Murali et al. [9] use diffusion models to generate 6-DoF grasps.

3 Approach: Generating the GraspFactory Dataset

Our approach to the GraspFactory dataset generation involves grasp sampling to generate candidate grasps, collision checks to filter out grasps where the gripper may be in collision with the object and physics based evaluation to determine whether the grasp can hold the object firmly. This section provides a detailed description of our approach.

3.1 Object CAD Models

We source the CAD models used in this study from the ABC dataset, which contains 1M+ diverse objects. We choose to work with 33,710 randomly selected objects from the ABC dataset for the Robotiq 2F-85 gripper and a subset of 14,690 of these objects for the Franka Panda gripper.

3.2 Grasp Sampling

The first step in our approach is to sample candidate grasps for all the chosen objects. Given the CAD model for an object, we utilize the antipodal sampling method to sample grasps. We ensure that the CAD models are watertight using [49]. We sample points on the mesh surface and compute their surface normals, denoted as $\hat{n}$. We then cast three rays within a cone aligned with the surface normal, with a vertex angle of $30°$, and identify the points on the CAD model that these rays intersect. We only consider the points whose surface normal is in the opposite direction of the ray origin's surface normal, as these points represent potential antipodal contact points. The gripper pose is determined by aligning the fingers' surface normals with the line connecting these antipodal points, and the z-axis of the gripper is aligned with four uniformly spaced vectors, each $90°$ apart, around this line.

Additionally, we decimate the object CAD model by a factor of 0.6 and repeat the antipodal grasp sampling process described above. A mesh decimation factor of 0.6 enables us to preserve the underlying shape of the mesh while also increasing the number of potential graspable surfaces. We

define a graspable surface on a mesh as a collection of triangles that come in contact with the finger grasping the object.

As shown in Fig. 3a, we perform collision checks between the gripper in sampled grasp poses and the object's CAD model using an internally developed robotics research software platform, eliminating grasps where finger geometry collides with the object.

We sample a total of 391.38 million non-colliding grasp candidates across 33,710 objects from the ABC dataset [21] using this approach.

3.3 Physics Based Grasp Evaluation

We evaluate the accuracy of each of the sampled grasps from Section 3.2 in the Isaac Sim simulator [44]. Due to computational limitations, we evaluate 2,000 grasps per object for Franka Panda robot equipped with a Franka hand as shown in Fig. 3b and 5,000 grasps per object for the Robotiq 2F-85 gripper as shown in Fig. 3c. The evaluated grasps are selected using Agglomerative Hierarchical Clustering [50] in the SE(3) space, which does not require a predefined number of clusters and can capture complex cluster structures. The distance d between two grasps g_1 and g_2 in the clustering process is defined as:

$$d = \|\mathbf{t}_1 - \mathbf{t}_2\| + \arccos\left(|q_1 \cdot q_2|\right) \tag{1}$$

where $\mathbf{t}_1$ and $\mathbf{t}_2$ are the translation components of g_1 and g_2, respectively, in $\mathbb{R}^3$. The orientations of g_1 and g_2 are represented as unit quaternions, q_1 and q_2 respectively.

We spawn 2,000 Franka Panda robots in the simulated environment, as shown Appendix B, to test each of the selected grasps for 14,690 objects. For each grasp, we spawn the object such that the grasp's z-axis is aligned with the world z-axis. We move the robot to the grasp pose and close the fingers around the object. Fig. 3b in Appendix B shows an example of a successfully grasped object. To ensure that the grasp is robust against external forces, we move the robot through a set of pre-defined poses, effectively testing whether the grasp can withstand perturbations. We record some extra information from the simulation runs and include it in the dataset for possible future use. Using the contact force information from the simulated environment, we record grasps that are in contact with the fingers during the entire simulation, and also record contact forces exerted on each spawned objects. We also record the duration of contact for the failed grasps.

Additionally, we spawn 5,000 Robotiq 2F-85 grippers in the simulated environment, as shown in Fig. 3c of Appendix B, to test each of the selected grasps for 33,710 objects. We move the gripper along the positive and negative world z-axis and rotate the gripper about the world z-axis to test the grasp robustness against external forces. We record a grasp to be successful if the object remains in between the fingers at the end of the simulation.

Successfully evaluated grasps are also referred to as *good* or *feasible* grasps in this paper.

Algorithm 1 summarizes our approach to generating the GraspFactory dataset.

3.4 Dataset Statistics

The GraspFactory dataset contains object-centric 6-DOF parallel grasps for the Franka Panda and Robotiq 2F-85 grippers. We include a list of grasp poses ${}_oT_g$ in the object coordinate frame and corresponding grasping width, and a list of indices of grasps that succeeded in physics simulation, g_w, where:

$${}_oT_g = \left\{ \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} R \in \text{SO}(3),\ t \in \mathbb{R}^3 \right\} \tag{2}$$

Grasping width is defined as the distance between the fingers of the gripper when grasping an object, measured in *mm*.

Algorithm 1: Grasp Sampling and Evaluation

Input: CAD Model M, Decimation Factor d
Output: Successful Grasp Poses $\{{}_oT_g\}$ with gripper widths $\{g_w\}$ in mm

Step 1: Grasp Sampling
foreach *point p on M* **do**
  Compute surface normal $\hat{n}$ at p
  Cast rays within 30° cone aligned with $\hat{n}$
  Find antipodal points (p_1, p_2) satisfying $\hat{n}_1 \cdot \hat{n}_2 = -1$
  Align fingers with points (p_1, p_2) and gripper z-axis spaced 90° apart
  if *collision-free* **then**
    Add $({}_oT_g, g_w)$ to list;
Decimate M by factor d and repeat sampling
Step 2: Grasp Clustering
Apply Agglomerative Hierarchical Clustering algorithm on sampled grasps in $SE(3)$ space
Select *N* representative grasps $\{{}_oT_g\}$ and their corresponding gripper widths $\{g_w\}$;
Step 3: Grasp Evaluation
foreach *grasp ${}_oT_g$ in $\{{}_oT_g\}$* **do**
  Spawn object in simulation such that z-axis of ${}_oT_g$ is aligned with the world z-axis
  Move robot to ${}_oT_g$ and close gripper
  Apply perturbations to test grasp robustness
  Record success and failures
Output: Successful grasps with poses, failed grasps with poses and widths

After physics based evaluation, GraspFactory contains 12.2 million feasible grasps for 14,690 objects for the Franka Panda gripper and 97.1 million feasible grasps for 33,710 objects for the Robotiq 2F-85 gripper, surpassing [7] in both the number of objects and grasps.

Given the size, diversity, and the real-world nature of this dataset, it is well-suited for training grasping models. We present the results of the point cloud-based SE(3)-DiffusionFields model [14] trained on Franka Panda subset of the GraspFactory dataset in Section 4.

3.5 Model Training

We train the point cloud-based model proposed in SE(3)-DiffusionFields using the Franka Panda Hand subset of GraspFactory and the ACRONYM datasets on two NVIDIA RTX-4090s with a batch size of 4 for 2,900 epochs over 19 days. We focus on the Franka Panda subset to align with the ACRONYM dataset, which also contains grasps for the Franka Panda gripper. SE(3)-DiffusionFields has been shown to outperform other grasp generative models in capturing and generating diverse grasps [14]. We use the same learning rate scheduler provided by [14]. We use 12,903 objects in the training set, 1,434 objects in the validation set, and 353 objects held out as part of the test set.

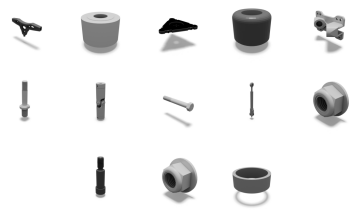
4 Experiments and Results

We evaluate the model's performance on a set of industrial objects of varying geometric complexities that were not part of the training data. We compare the model trained using GraspFactory to the same model trained using the ACRONYM dataset. We test the performance in both simulation and real-world settings. Simulation allows testing a larger set of grasps, predicted by the model, for their accuracy, while real-world settings test the physical feasibility and robustness of the grasps.

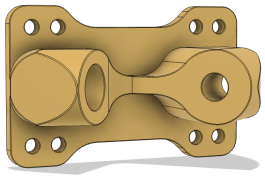
4.1 Simulated Experiments

We sample 100 grasps from the models trained on GraspFactory and ACRONYM for the objects shown in Fig. 1a. Since simulation is non-deterministic, there are minor differences in the absolute success rate numbers. To avoid any bias and to provide statistical consistency, we run the experiments with two random seeds. Qualitative evaluation shown in Appendix D demonstrates that the model trained on the ACRONYM dataset generates grasps that intersect with the object meshes, whereas the model trained on GraspFactory produces non-intersecting grasps, showing the effectiveness of our dataset for applications involving complex geometries.

Grasp success rate is evaluated using the same metrics to identify successful grasps as explained in Sec. 3.3 and define accuracy as the percentage of successful grasps in simulation. We show the success rate for 100 grasps sampled from the model trained on GraspFactory and ACRONYM datasets in Table 1. The model trained on GraspFactory outperforms the model trained on ACRONYM across all objects shown in Fig. 1a in simulation by a wide margin in most cases. In fact, the success rates are close only for objects like the *Strut*, whose constituent geometric primitives (cuboid and cylinder) are well represented in the ACRONYM dataset.



(a) **Top row**: Hanger, Hardcore Bearing, Top Plate, Wheel, Base
Middle row: Axle, Elbow Joint, Kingpin Bolt, Strut, Kingpin Nut
Bottom row: Shoulder Screw, Axle Nut, Hollow Cylinder



(b) Zoomed-in view of the Base, showing its complexity.

Figure 1: Examples of objects used in experiments and simulations.

Objects	Round 1		Round 2	
	GraspFactory (Ours)	ACRONYM	GraspFactory (Ours)	ACRONYM
Hanger	0.97	0.30	0.93	0.22
Hardcore Bearing	0.99	0.00	0.94	0.00
Top Plate	0.91	0.53	0.96	0.49
Wheel	1.00	0.00	1.00	0.00
Base	0.75	0.00	0.72	0.47
Axle	0.85	0.00	0.72	0.00
Elbow Joint	1.00	0.00	1.00	0.00
Kingpin Bolt	0.71	0.48	0.93	0.00
Strut	0.96	0.94	0.83	0.87
Kingpin Nut	0.84	0.00	0.85	0.00
Shoulder Screw	0.85	0.00	0.81	0.00
Axle Nut	0.99	0.00	0.96	0.00
Hollow Cylinder	0.97	0.40	0.91	0.55

Table 1: Simulation Results: Success rates for 100 grasps generated by the model trained on GraspFactory (Ours) and ACRONYM datasets in simulation.

The success rate in simulation is lower for *Base*, shown in Fig. 1b compared to the other objects, as the flat finger geometry results in unstable grasps near its center of mass. Incorporating finger geometry may improve the model's ability to predict stable grasps for a specific finger geometry. *Axle* and *Kingpin Bolt* have hexagonal heads, and we observe that some of the grasp poses generated by the model are located over the vertices of the hexagon. In simulation, these grasps fail during

grasp evaluation when we attempt to move the grasped object around (in the manner described in Section 3.3).

Train Dataset \ Test Dataset	GraspFactory	ACRONYM
GraspFactory (Ours)	0.75	0.59
ACRONYM	0.08	0.57

Table 2: Average success rates of 100 grasps for 353 and 95 test objects from GraspFactory (Ours) and ACRONYM respectively in simulation.

Results of the model trained on GraspFactory and ACRONYM on held-out objects from both the datasets are in Table 2.

4.2 Hardware Experiments

We evaluate the strengths of GraspFactory using physical experiments based on two metrics: *real-world feasibility* and *grasp robustness* using the hardware setup and perception pipeline outlined in Appendix E.[2] We use *real-world feasibility* to evaluate whether grasps generated by the model trained on GraspFactory can be used reliably to pick up an object without colliding with the object or the surrounding objects, such as the table. *Grasp robustness* measures the consistency of a grasp across repeated trials, highlighting its ability to maintain performance under minor uncertainty introduced by perception.

For each object, we sample 200 grasps from the model. Using the perception pipeline described in Appendix E, we obtain the pose of the object and grasps in the world-frame. Grasps that are in collision with the support surface (table) are then eliminated, resulting in a smaller non-colliding grasp set that we process further.

Parts	Random Pose 1			Random Pose 2			Random Pose 3		
	Num Eval	Num Success	Success Rate (%)	Num Eval	Num Success	Success Rate (%)	Num Eval	Num Success	Success Rate (%)
Strut	30	29	96.67	30	29	96.67	30	30	100.00
Elbow Joint	30	28	93.33	30	30	100.00	30	29	96.67
Wheel	30	30	100.00	30	29	96.67	30	30	100.00
Hanger	30	30	100.00	30	30	100.00	30	30	100.00
Base	30	24	80.00	30	29	96.67	30	27	90.00
Gear	30	30	100.00	30	26	86.67	30	29	96.67
Kingpin Bolt	30	28	93.33	30	30	100.00	30	29	96.67
Regrasp Fixture	30	30	100.00	30	29	96.67	30	28	93.33

Table 3: Hardware results with real objects: Evaluation of real-world feasibility of grasps sampled from the model trained on GraspFactory for three random poses across eight parts.

Part	Grasp 1	Grasp 2	Grasp 3	Grasp 4	Grasp 5	Average Success Rate (%)
Base	10	10	10	8	10	96.00
Hanger	10	10	10	10	10	100.00
Gear	10	10	10	10	10	100.00
Regrasp Fixture	10	10	10	10	10	100.00

Table 4: Hardware results with real objects: Evaluation results for grasp robustness for five random non-colliding grasps. **Num Trials=10**

Real-World Feasibility We evaluate real-world feasibility by randomly selecting 30 grasps per object from the non-colliding grasp set, resulting in a total of 720 grasps evaluated across eight objects for three random stable poses (shown in Fig. 10 in Appendix E). We consider a grasp to be successful if the gripper fingers do not collide with the object or the table, and the robot successfully picks up the object $100mm$ off the table and places it back.

[2]We use a UR10e robot and Robotiq 2F-85 gripper for testing on real-hardware due to an unanticipated lack of availability of our Franka Panda robot.

Part	Part
Axle (**5/5**)	Camera Mount B (**5/5**)
GPU Cooling Bracket (**5/5**)	Drone RPM Sensor Mount (**5/5**)
GPU Fan Bracket (**4/5**)	Drone Landing Gear Mount (**3/5**)
Camera Mount A (**5/5**)	Drone Support Structure (**5/5**)
Automotive Relay (**5/5**)	Drone Motor Mount (**5/5**)

Table 5: Results of real world experiments (number of successful grasps / number of grasps tested).

Table 3 shows the grasp success rate per object in each of the three selected poses.[3] We show that the model trained on GraspFactory produces grasps that can be executed in the real-world, demonstrating the real-world usefulness of the dataset.

We run our perception pipeline in an open-loop manner, meaning that we do not estimate the pose of the object when it is placed back on the support surface. We observe a minor change in the pose of the object (due to its shape) when the robot places the object back, resulting in subsequent picks to fail sometimes (without human intervention to restore the object to its original location). This is particularly pronounced for *Base* in *Pose 1* and *Gear* in *Pose 2* as shown in Table 3. Tested poses of objects are shown in Appendix E.

Grasp Robustness Grasp robustness is evaluated by selecting five grasps from the non-colliding grasp set for four objects. We perform 10 trials per grasp, where each trial involves picking up the object and placing it down, resulting in a total of 200 grasp evaluations. We use the same metrics as outlined in Section 4.2 for grasp success and note the average success rate across the five selected grasps in Table 4. Our results show that the tested grasps are fairly robust even with small perturbations in object pose.

We also evaluate five grasps per each of the 10 additional objects of varying geometric complexities shown in Appendix E's Fig. 8 and present our results in Table 5.

One of the challenges and limitations in our real-world experiments is differentiating errors due to pose estimation from those caused by grasp estimation. Any calibration error also affects our estimates of where the objects are, hence affecting the success of our chosen grasps as well. We note, however, that pose estimation is not the focus of our work presented here.

5 Conclusion

In this paper, we introduce GraspFactory, a large parallel-jaw grasp dataset containing 12.2 million feasible grasps for the Franka Panda gripper across 14,690 geometrically diverse objects and 97.1 million feasible grasps for the Robotiq 2F-85 gripper across 33,710 objects. The geometric diversity of the dataset addresses a critical gap in existing grasp datasets, which focus on objects with limited shape complexity or variety. Our results in simulation show that a model trained on GraspFactory significantly outperforms a model trained on existing datasets, such as ACRONYM, in terms of generalization to novel objects. Furthermore, we evaluated more than 900 grasps generated by the model trained on GraspFactory in real-world settings, demonstrating that our dataset enables models to generate grasps that can be used reliably in the real-world. GraspFactory contains information about grasping width for each grasp pose, which could be used to learn collision-free grasps in cluttered scenarios.

In the future, we plan to integrate finger geometry into training, enhancing both feasibility and robustness of generated grasps. We also aim to extend our dataset to include a larger number of objects from the ABC dataset. Subsequently, we aim to extend GraspFactory to include grasps for other end-effectors, such as suction-cup grippers. This extension would enhance the dataset's versatility, enabling researchers to develop and evaluate grasping algorithms applicable to a wider range of robotic systems and applications.

[3] Grasps generated by the ACRONYM-trained model were not evaluated, since qualitative inspection showed poor grasp quality.

References

[1] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115:211–252, 2015.

[2] C. Schuhmann, R. Beaumont, R. Vencu, C. Gordon, R. Wightman, M. Cherti, T. Coombes, A. Katta, C. Mullis, M. Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in Neural Information Processing Systems*, 35: 25278–25294, 2022.

[3] OpenAI. Chatgpt: Large language model. https://chat.openai.com/, 2023.

[4] OpenAI. Dall·e: Image generation model. https://openai.com/dall-e, 2023.

[5] O. X.-E. Collaboration, A. O'Neill, A. Rehman, A. Gupta, A. Maddukuri, A. Gupta, A. Padalkar, A. Lee, A. Pooley, and *et al.* Open X-Embodiment: Robotic Learning Datasets and RT-X Models. https://arxiv.org/abs/2310.08864, 2023.

[6] A. Khazatsky, K. Pertsch, S. Nair, A. Balakrishna, S. Dasari, S. Karamcheti, S. Nasiriany, M. K. Srirama, L. Y. Chen, K. Ellis, and *et al.* Droid: A large-scale in-the-wild robot manipulation dataset. *arXiv preprint arXiv:2403.12945*, 2024.

[7] C. Eppner, A. Mousavian, and D. Fox. Acronym: A large-scale grasp dataset based on simulation. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6222–6227. IEEE, 2021.

[8] D. Morrison, P. Corke, and J. Leitner. Egad! an evolved grasping analysis dataset for diversity and reproducibility in robotic manipulation. *IEEE Robotics and Automation Letters*, 5(3): 4368–4375, 2020.

[9] A. Murali, B. Sundaralingam, Y.-W. Chao, J. Yamada, W. Yuan, M. Carlson, F. Ramos, S. Birchfield, D. Fox, and C. Eppner. Graspgen: A diffusion-based framework for 6-dof grasping with on-generator training. *arXiv preprint arXiv:2507.13097*, 2025. URL https://arxiv.org/abs/2507.13097.

[10] Y. Jiang, S. Moseson, and A. Saxena. Efficient grasping from RGBD images: Learning using a new rectangle representation. In *2011 IEEE International Conference on Robotics and Automation*, pages 3304–3311, 2011. doi:10.1109/ICRA.2011.5980145.

[11] A. Depierre, E. Dellandréa, and L. Chen. Jacquard: A large scale dataset for robotic grasp detection. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3511–3516. IEEE, 2018.

[12] H.-S. Fang, C. Wang, M. Gou, and C. Lu. GraspNet-1Billion: A Large-Scale Benchmark for General Object Grasping. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11444–11453, 2020.

[13] U. Asif, J. Tang, and S. Harrer. GraspNet: An Efficient Convolutional Neural Network for Real-time Grasp Detection for Low-powered Devices. In *IJCAI*, volume 7, pages 4875–4882, 2018.

[14] 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)*, pages 5923–5930. IEEE, 2023.

[15] 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)*, pages 13438–13444. IEEE, 2021.

[16] 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.

[17] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015.

[18] B. Calli, A. Walsman, A. Singh, S. Srinivasa, P. Abbeel, and A. M. Dollar. Benchmarking in manipulation research: The ycb object and model set and benchmarking protocols. *arXiv preprint arXiv:1502.03143*, 2015.

[19] M. Deitke, D. Schwenk, J. Salvador, L. Weihs, O. Michel, E. VanderBilt, L. Schmidt, K. Ehsani, A. Kembhavi, and A. Farhadi. Objaverse: A universe of annotated 3d objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13142–13153, 2023.

[20] P. Shilane, P. Min, M. Kazhdan, and T. Funkhouser. The Princeton Shape Benchmark. In *Proceedings Shape Modeling Applications, 2004.*, pages 167–178, 2004. doi:10.1109/SMI.2004.1314504.

[21] S. Koch, A. Matveev, Z. Jiang, F. Williams, A. Artemov, E. Burnaev, M. Alexa, D. Zorin, and D. Panozzo. ABC: A Big CAD Model Dataset For Geometric Deep Learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[22] NVIDIA Corporation. PhysX SDK 4.1, 2019. URL https://developer.nvidia.com/physx-sdk. Accessed: 2025-01-06.

[23] E. Coumans. Bullet physics simulation. In *Special Interest Group on Computer Graphics and Interactive Techniques Conference, SIGGRAPH '15, Los Angeles, CA, USA, August 9-13, 2015, Courses*, page 7:1. ACM, 2015. doi:10.1145/2776880.2792704. URL https://doi.org/10.1145/2776880.2792704.

[24] K. D. Willis, Y. Pu, J. Luo, H. Chu, T. Du, J. G. Lambourne, A. Solar-Lezama, and W. Matusik. Fusion 360 gallery: A dataset and environment for programmatic cad construction from human design sequences. *ACM Transactions on Graphics (TOG)*, 40(4):1–24, 2021.

[25] K.-H. Hui, A. Sanghi, A. Rampini, K. R. Malekshan, Z. Liu, H. Shayani, and C.-W. Fu. Make-a-shape: a ten-million-scale 3d shape model. In *Forty-first International Conference on Machine Learning*, 2024.

[26] A. Sanghi, A. Khani, P. Reddy, A. Rampini, D. Cheung, K. R. Malekshan, K. Madan, and H. Shayani. Wavelet Latent Diffusion (Wala): Billion-Parameter 3D Generative Model with Compact Wavelet Encodings. *arXiv preprint arXiv:2411.08017*, 2024.

[27] C. R. Qi, H. Su, K. Mo, and L. J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017.

[28] J. Mahler, J. Liang, S. Niyaz, M. Laskey, R. Doan, X. Liu, J. A. Ojea, and K. 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.

[29] D. Kappler, J. Bohg, and S. Schaal. Leveraging big data for grasp planning. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4304–4311, 2015. doi:10.1109/ICRA.2015.7139793.

[30] W. Wohlkinger, A. Aldoma, R. B. Rusu, and M. Vincze. 3DNet: Large-scale object class recognition from CAD models. In *2012 IEEE International Conference on Robotics and Automation*, pages 5384–5391, 2012. doi:10.1109/ICRA.2012.6225116.

[31] A. Kasper, Z. Xue, and R. Dillmann. The KIT object models database: An object model database for object recognition, localization and manipulation in service robotics. *The International Journal of Robotics Research*, 31(8):927–934, 2012. doi:10.1177/0278364912445831. URL https://doi.org/10.1177/0278364912445831.

[32] H. Zhang, X. Lan, S. Bai, X. Zhou, Z. Tian, and N. Zheng. Roi-based robotic grasp detection for object overlapping scenes. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4768–4775. IEEE, 2019.

[33] H. Zhang, X. Lan, X. Zhou, Z. Tian, Y. Zhang, and N. Zheng. Visual Manipulation Relationship Network for Autonomous Robotics. In *2018 IEEE-RAS 18th International Conference on Humanoid Robots (Humanoids)*, pages 118–125, 2018. doi:10.1109/HUMANOIDS.2018.8625071.

[34] D. Prattichizzo and J. C. Trinkle. *Grasping*, pages 671–700. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008. ISBN 978-3-540-30301-5. doi:10.1007/978-3-540-30301-5_29. URL https://doi.org/10.1007/978-3-540-30301-5_29.

[35] L. Gatrell. CAD-based grasp synthesis utilizing polygons, edges and vertexes. In *Proceedings, 1989 International Conference on Robotics and Automation*, pages 184–189 vol.1, 1989. doi:10.1109/ROBOT.1989.99987.

[36] B. Horn. Extended Gaussian images. *Proceedings of the IEEE*, 72(12):1671–1686, 1984. doi:10.1109/PROC.1984.13073.

[37] S. M. LaValle. *Planning Algorithms*. Cambridge University Press, Cambridge, U.K., 2006. Available at http://planning.cs.uiuc.edu/.

[38] M. Veres, M. Moussa, and G. W. Taylor. An integrated simulator and dataset that combines grasping and vision for deep learning. *arXiv preprint arXiv:1702.02103*, 2017.

[39] A. t. Pas and R. Platt. Using geometry to detect grasps in 3d point clouds. *arXiv preprint arXiv:1501.03100*, 2015.

[40] X. Zhu and J. Wang. Synthesis of force-closure grasps on 3-D objects based on the Q distance. *IEEE Transactions on robotics and Automation*, 19(4):669–679, 2003.

[41] L. Han, J. C. Trinkle, and Z. X. Li. Grasp analysis as linear matrix inequality problems. *IEEE Transactions on Robotics and Automation*, 16(6):663–674, 2000.

[42] M. Macklin, M. Müller, N. Chentanez, and T.-Y. Kim. Unified particle physics for real-time applications. *ACM Trans. Graph.*, 33(4), July 2014. ISSN 0730-0301. doi:10.1145/2601097.2601152. URL https://doi.org/10.1145/2601097.2601152.

[43] C. Eppner, A. Mousavian, and D. Fox. A billion ways to grasp: An evaluation of grasp sampling schemes on a dense, physics-based grasp data set. In *The International Symposium of Robotics Research*, pages 890–905. Springer, 2019.

[44] N. Corporation. *NVIDIA Isaac Sim*, 2023. URL https://developer.nvidia.com/isaac-sim. Version 2023.1.

[45] R. Newbury, M. Gu, L. Chumbley, A. Mousavian, C. Eppner, J. Leitner, J. Bohg, A. Morales, T. Asfour, D. Kragic, et al. Deep learning approaches to grasp synthesis: A review. *IEEE Transactions on Robotics*, 39(5):3994–4015, 2023.

[46] D. Morrison, P. Corke, and J. Leitner. Closing the loop for robotic grasping: A real-time, generative grasp synthesis approach. *arXiv preprint arXiv:1804.05172*, 2018.

[47] A. Mousavian, C. Eppner, and D. Fox. 6-dof graspnet: Variational grasp generation for object manipulation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 2901–2910, 2019.

[48] D. P. Kingma. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[49] J. Huang, H. Su, and L. Guibas. Robust watertight manifold surface generation method for shapenet models. *arXiv preprint arXiv:1802.01698*, 2018.

[50] K. Hang, J. A. Stork, and D. Kragic. Hierarchical Fingertip Space for multi-fingered precision grasping. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1641–1648, 2014. doi:10.1109/IROS.2014.6942775.

[51] C. Goldfeder, M. Ciocarlie, H. Dang, and P. K. Allen. The Columbia grasp database. In *2009 IEEE International Conference on Robotics and Automation*, pages 1710–1716, 2009. doi:10.1109/ROBOT.2009.5152709.

[52] V. N. Nguyen, T. Groueix, G. Ponimatkin, V. Lepetit, and T. Hodan. Cnos: A strong baseline for cad-based novel object segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2134–2140, 2023.

[53] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár, and R. Girshick. Segment Anything. *arXiv:2304.02643*, 2023.

[54] X. Zhao, W. Ding, Y. An, Y. Du, T. Yu, M. Li, M. Tang, and J. Wang. Fast segment anything. *arXiv preprint arXiv:2306.12156*, 2023.

[55] M. Oquab, T. Darcet, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

[56] B. Wen, W. Yang, J. Kautz, and S. Birchfield. Foundationpose: Unified 6d pose estimation and tracking of novel objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17868–17879, 2024.

6 Appendix

We provide additional information about our dataset in Appendix A, B, and C. Appendix D and E show qualitative results and real-world experiment setup respectively.

A Objects in the dataset

Fig. 2 shows a random subset of objects in the dataset. We compare the GraspFactory dataset with prior work in the literature and highlight the comparison in Table 6.

Figure 2: A subset of the objects in the GraspFactory Dataset showing Geometric Variability.

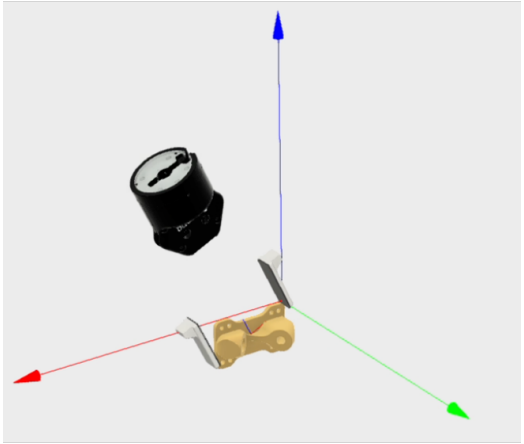
Dataset	Planar/6D	Labels	Number of Objects	Candidate Grasps	Good Grasps/ Evaluated Grasps	Object-Centric Grasps
Cornell [10]	Planar	Manual	240	8k	NA	✗
Jacquard [11]	Planar	Sim	11k	1.1M	NA	✗
VMRD + Grasps [32]	Planar	Manual	∼15k	100k	NA	✗
Columbia [51]	6D	Analytical	7256	238k	NA	✓
Dex-Net [28]	6D	Analytical	1500	6.7M	NA	✗
6-DoF GraspNet [13]	6D	Sim	206	7.07M	NA	✗
GraspNet [12]	6D	Analytical	88	1.1B	NA	✗
EGAD [8]	6D	Analytical	2331	233k	NA	✓
ACRONYM [7]	6D	Sim	8872	17.7M	10.5M/17.7M	✓
GraspGen [9]	6D	Sim	8515	53.1M	NA	✓
GraspFactory (Ours)	**6D**	**Sim**	**14,690**	**227.22M**	**12.2M/29.38M**	✓
GraspFactory - Robotiq 2F-85 (Ours)	**6D**	**Sim**	**33,710**	**391.38M**	**97.1M/164.16M**	✓

Table 6: Summary of various grasp datasets, highlighting labeling methods, number of objects and grasps.
NA - no data available, ✓- dataset contains object centric grasps,
✗ - dataset contains grasps for a scene (images and point clouds).

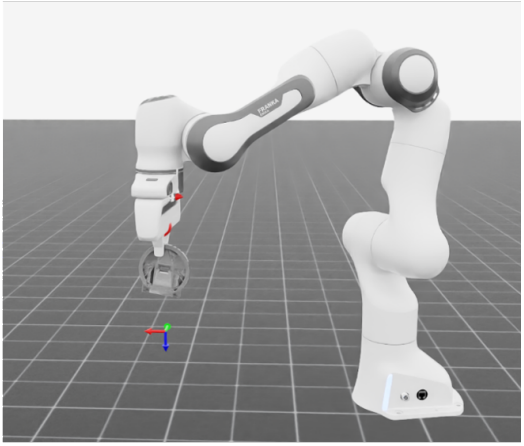
B Data Generation

We use an internally developed robotics research platform to perform collision check between the gripper fingers and the sampled objects as shown in Fig. 3a. Simulated Robotiq 2F-85 is shown in Fig. 3c and Franka Panda Hand in Isaac Sim to test physical feasibility is shown in Fig. 3d.

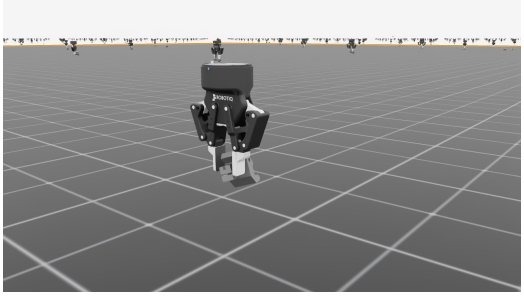
The evaluation pipeline for ABC-Grasp dataset generation in simulation assumes uniform physical properties, including mass and the coefficient of friction, across all objects in the dataset to ensure



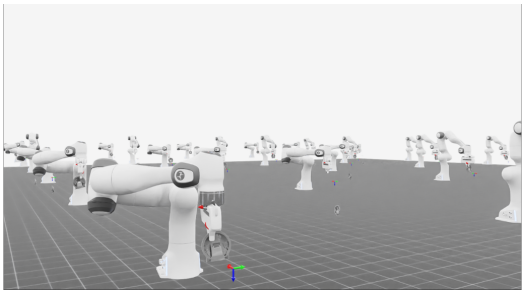
(a) Collision check between an object and the gripper fingers using our internally developed robotics research software platform.



(b) Single robot executing a sampled grasp.



(c) Isaac Sim evaluation environment with Robotiq 2F-85.



(d) Isaac Sim evaluation environment with Franka Panda Robot with Panda Hand.

Figure 3: Simulation environments for evaluating collision checks, sampled grasp execution, and physics-based evaluation.

computational feasibility. We note that variations in mass and friction may influence grasp stability and robustness for objects with slippery or uneven surfaces. Additionally, we treat all objects in the dataset as rigid.

C Data Quality

A plot of the location of successful grasps, shown in Fig. 4 shows that our method covers the entire space around the objects.

Fig. 5 presents metrics such as the *number of triangles*, *number of vertices*, and *edge length statistics* for the GraspFactory, ACRONYM, Dex-Net, and EGAD datasets. The graphs demonstrate that GraspFactory exhibits a wider spread compared to both ACRONYM and Dex-Net. While EGAD shows a more uniform distribution than GraspFactory, GraspFactory contains approximately seven times more objects, and its objects better align with those encountered in the real world compared to the EGAD dataset.

These metrics were chosen to highlight geometric diversity as they are directly related to the structural complexity of the meshes, serving as quantifiable indicators of geometric diversity. They are also computationally efficient to calculate and provide an immediate sense of the detail in a CAD model.

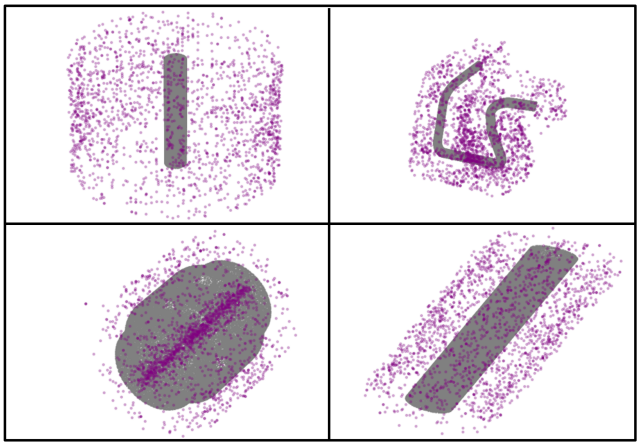


Figure 4: Qualitative analysis of grasp coverage for four randomly selected objects from the dataset. Purple points represent gripper positions around the objects.

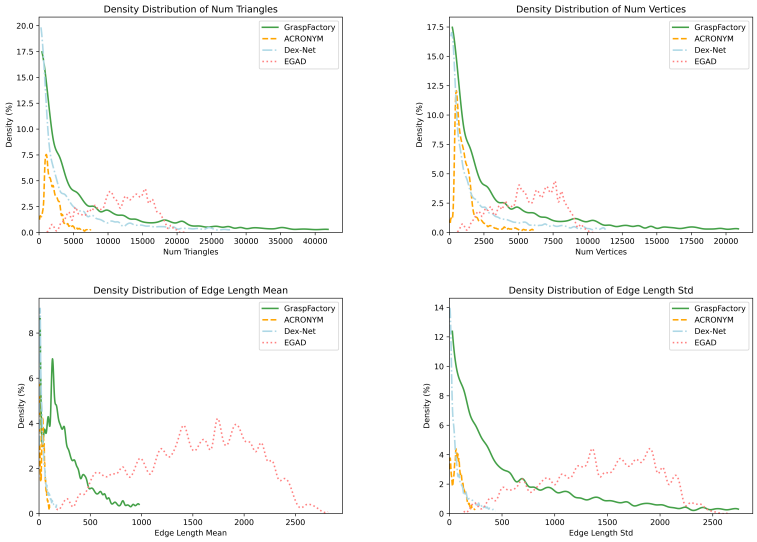


Figure 5: Density distribution curves for the number of triangles, number of vertices and edge lengths' mean and standard deviation for GraspFactory (Ours), ACRONYM [7], Dex-Net [28], EGAD [8] showing a larger variance in GraspFactory.
EGAD shows more uniform spread in Edge Length metrics, but contains approximately seven times fewer objects and is less representative of real-world objects.

D Qualitative results

We show the qualitative results of the model trained on GraspFactory and ACRONYM datasets in Fig. 6. The model trained on ACRONYM produces grasps that intersect with the objects, whereas, the model trained on our GraspFactory dataset produces grasps uniformly around the objects.

E Real-World Experiments

We perform physical experiments on 18 real-world objects shown in Fig. 8. The workcell setup for the experiment consists of a Zivid 2+ M60 camera mounted on a UR-10 robot, and grasping is

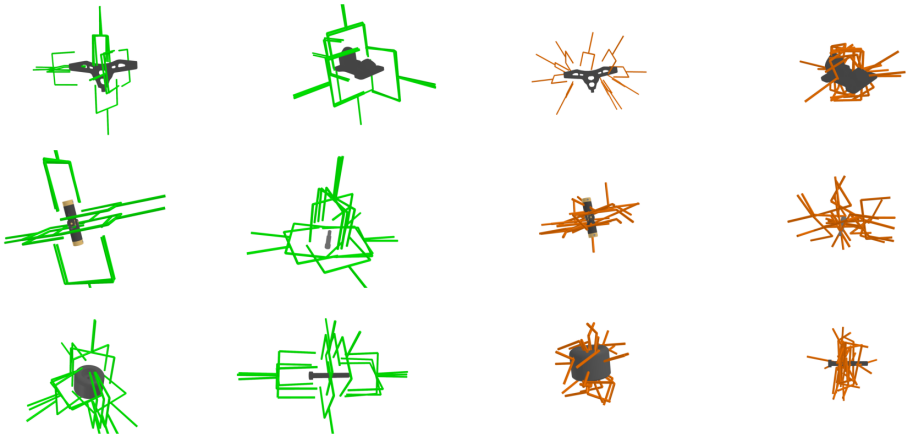


Figure 6: Qualitative comparison of grasps generated for unseen objects by the model trained on GraspFactory (ours, left two columns in green) and the model trained on ACRONYM (right two columns in orange).

performed by a UR-10e robot, equipped with a Robotiq 2F-85 two-fingered gripper [4] as shown in Fig. 7. We note that the model was trained on grasps that were validated using a Franka Panda robot with Franka hand in simulation. This gripper has a finger width of $18mm$, while our real-world evaluation is performed using a Robotiq 2F-85 two-fingered gripper whose finger width is $22mm$.

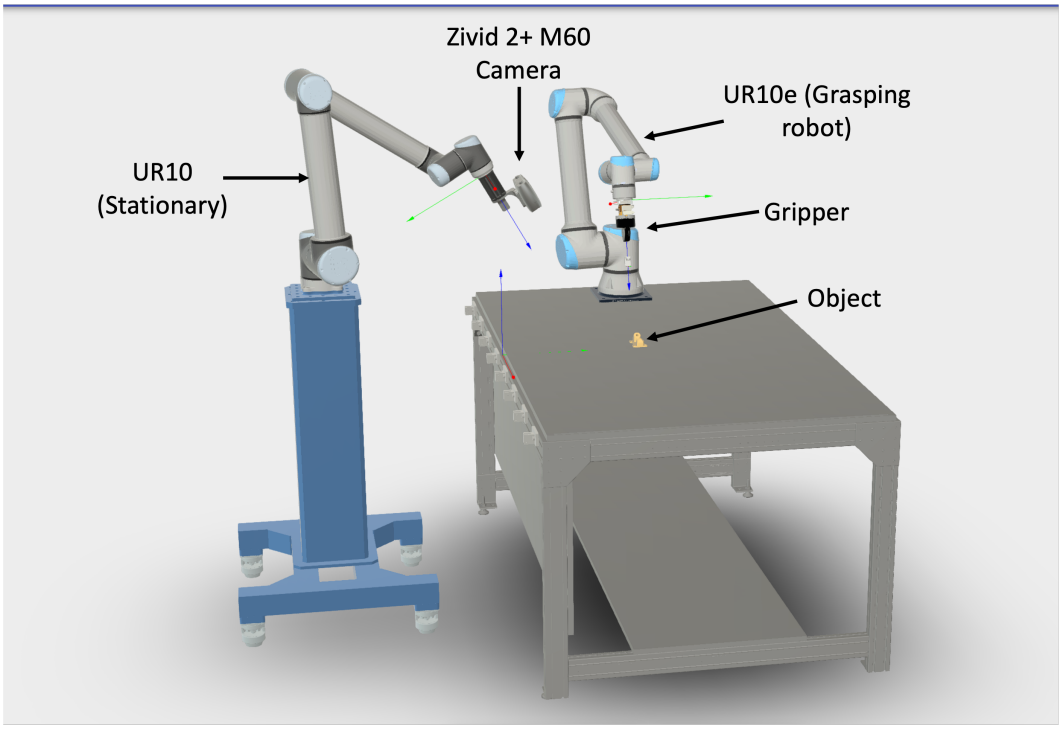


Figure 7: Workcell setup for real-world experiments. We use a UR-10e robot equipped with a Robotiq 2F-85 gripper for grasping. Zivid 2+ M60 camera is mounted on a UR-10 robot.

In each experiment, we first place individual objects in front of the robot and implement a perception based pipeline, as shown in Fig. 9, to locate the object with respect to the robot. Our perception

[4] Due to an unanticipated lack of availability of our Franka Panda robot, we chose to use a UR10e robot and Robotiq 2F-85 gripper for testing on real-hardware.

Figure 8: Objects used for real-world experiments

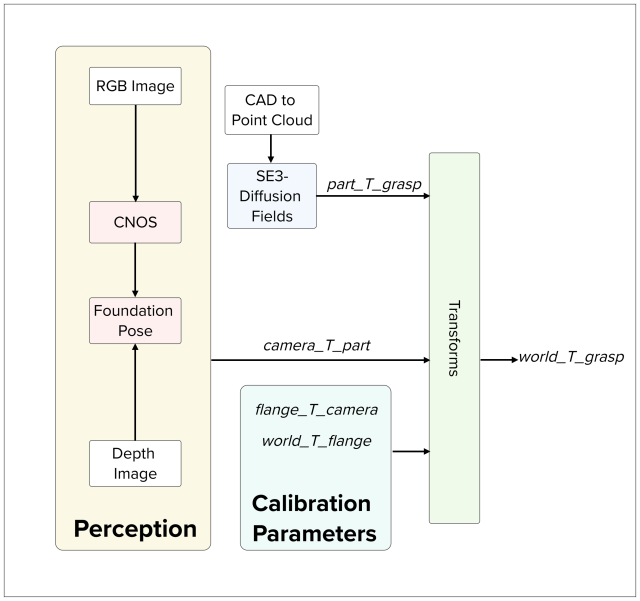


Figure 9: Pipeline for real-world experiments

pipeline builds on CNOS presented by Nguyen et al. [52], a method that utilizes segmentation proposals generated from the captured RGB image using the Segment Anything Model [53] or Fast Segment Anything model (FastSAM) [54] to localize an object of interest in the scene. CNOS matches the DINOv2 cls [55] tokens of the proposed segmentation regions against tokens of object templates that are pre-rendered using their CAD models. We use this localization information to segment the point cloud of the object captured by the camera.

With the object localized in the scene, we use the model-based setup of FoundationPose [56], which uses the CAD model and the segmented point cloud of the object (which we obtain from CNOS) to estimate the object's 6-DoF pose in the camera frame, denoted by ${}_cT_o$. We then use the camera extrinsics and robot calibration parameters to transform the computed grasps into the world-coordinate frame, as described by the equation below:

$$ {}_wT_g = {}_wT_r \cdot {}_rT_f \cdot {}_fT_c \cdot {}_cT_o \cdot {}_oT_g \tag{3} $$

where, T is a transformation matrix defined as shown in Eq. 2, w is the world frame, r is the robot frame, f is the robot flange frame, c is the camera frame, o is the part or the object frame.

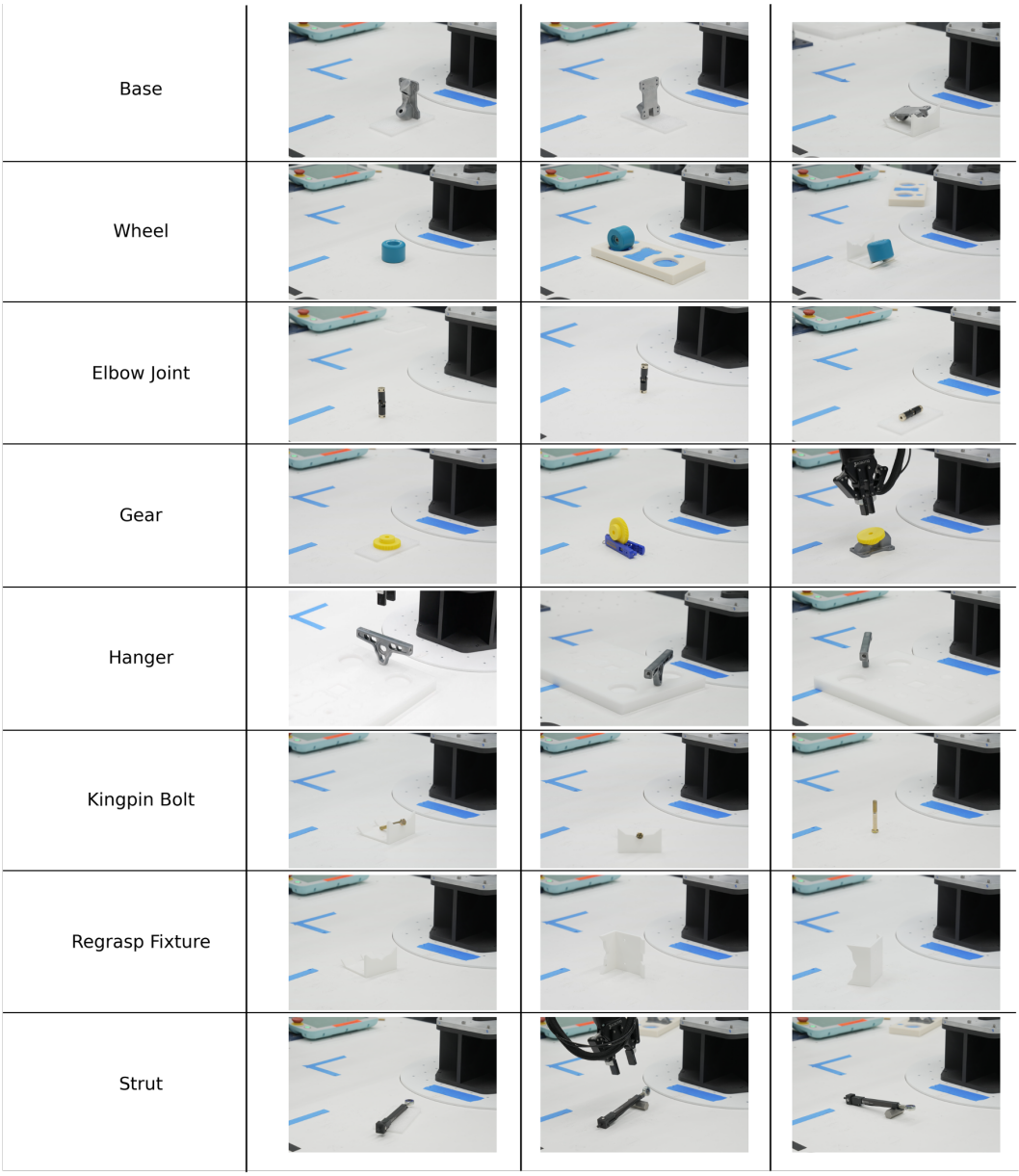


Figure 10: Real world experiment with eight objects, each in three random stable poses.

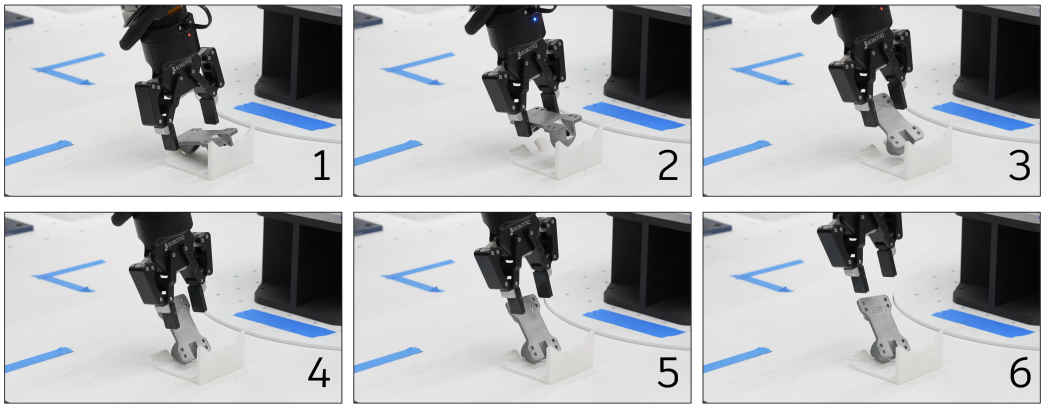


Figure 11: Sequence of frames depicting the *Base* slipping out of the fingers. Frame sequence numbers are embedded in the images.

We choose 18 diverse set of objects, shown in Fig. 8 to evaluate the model trained on our dataset in real-world settings.

Fig. 11 shows that the flat finger geometry leads to unstable grasps near the center of mass of the *Base*, a behavior also observed in our simulation experiments. Since the model we train does not account for finger geometry, incorporating this factor could help ensure the generation of only stable grasps.