

MultiGripperGrasp: A Dataset for Robotic Grasping from Parallel Jaw Grippers to Dexterous Hands

Luis Felipe Casas*, Ninad Khargonkar*, Balakrishnan Prabhakaran, Yu Xiang



Fig. 1: Dataset Visualization: Our dataset contains grasps from 11 different grippers on a diverse object set.

Abstract—We introduce a large-scale dataset named **Multi-GripperGrasp** for robotic grasping. Our dataset contains 30.4M grasps from 11 grippers for 345 objects. These grippers range from two-finger grippers to five-finger grippers, including a human hand. All grasps in the dataset are verified in the robot simulator Isaac Sim to classify them as successful and unsuccessful grasps. Additionally, the object fall-off time for each grasp is recorded as a grasp quality measurement. Furthermore, the grippers in our dataset are aligned according to the orientation and position of their palms, allowing us to transfer grasps from one gripper to another. The grasp transfer significantly increases the number of successful grasps for each gripper in the dataset. Our dataset is useful to study generalized grasp planning and grasp transfer across different grippers.¹

I. INTRODUCTION

Developing generalized and robust grasping methods for object manipulation is a core research topic in robotics. Autonomous robot manipulation requires grasping methods that can successfully grasp the desired object for any given task. An important problem in robotic grasping is grasp planning or grasp synthesis, where the focus is on generating the configuration of a robot gripper for grasping based on the perceived object information.

Traditional grasp planning methods use grasp quality measurements, such as task wrench space analysis [1], [2] or force closure analysis [3], [4] to analytically synthesize grasps. The main limitation of these approaches is that they require full-state information of the target object, which is

* Equal contribution. All authors are with Department of Computer Science, University of Texas at Dallas, Richardson, TX 75080, USA {luis.casasmurillo, ninadarun.khargonkar, bprabhakaran, yu.xiang}@utdallas.edu

¹Data, code and videos for the project are available at <https://irvlutd.github.io/MultiGripperGrasp>

usually represented by a 3D mesh for grasp planning. Consequently, these methods cannot synthesize grasps from partial observations, such as images or point clouds of objects. Therefore, learning-based grasp synthesis approaches [5]–[9] have attracted more attention in recent years. These methods use machine learning models, especially deep neural networks, to predict grasps from sensory input. By training on data with diverse objects, these methods can synthesize grasps of novel objects.

However, learning-based grasp planning requires large-scale datasets for training. Therefore, several efforts have been devoted to building grasping datasets for learning. For example, the ACRONYM dataset [10] used 8,872 3D objects and generated 17.7M parallel-jaw grasps of the Franka panda gripper based on physics simulation. The DexGraspNet dataset [11] used 5,355 3D objects to generate 1.32M grasps for the Shadow hand. We notice that most grasping datasets are constructed for one specific gripper. As a result, the models trained on these datasets cannot be easily transferred to different grippers. A large-scale grasping dataset with a diverse set of grippers can be useful to study generalized grasp planning among various grippers.

In this work, we introduce MultiGripperGrasp, a large-scale grasping dataset for studying generalized robotic grasping. Our dataset contains 30.4M grasps generated using 345 3D objects and 11 robotic grippers. There are 5 two-finger grippers, 3 three-finger grippers, 1 four-finger gripper, and 2 five-finger grippers in the dataset. One of the five-finger grippers is a human hand. Fig. 1 shows examples of grasps from our dataset. Compared to existing grasping datasets, our dataset has the following unique features. 1) It has the maximum number of grippers up to date. Only

TABLE I: Overview of representative grasping datasets. +ve and -ve indicate binary successful and failed grasps, respectively.

Dataset	Year	#Grippers	#Objects	#Grasps	Grasp Label	Synthesis Method	Code + Data
HO-3D [12]	2020	1 (Human hand)	10	78K	Only +ve	Human Demo	✓
EGAD [13]	2020	1 (2-finger)	2,331	233K	Only +ve	Evolutionary Algorithm	✓
DDG [6]	2020	1 (5-finger)	500	50K	Only +ve	GraspIt + modified Q1 [14]	✗
DexYCB [15]	2021	1 (Human hand)	20	582K	Only +ve	Human Demo	✓
Acronym [10]	2021	1 (2-finger)	8,872	17.7M	+ve & -ve	Flex [16]	✓
UniGrasp [17]	2020	12 (2 & 3finger)	1000	2M+	Only +ve	Contact Points Network + FastGrasp [18]	✓
DexGraspNet [11]	2023	1 (5-finger)	5,355	1.3M	Only +ve	Differentiable grasping	✓
Fast-Grasp'D [19]	2023	3 (3-5 finger)	2,350	1M	Only +ve	Differentiable grasping	✗
GenDexGrasp [8]	2023	5 (2-5 finger)	58	436K	Only +ve	Differentiable grasping	✓
MultiGripperGrasp (Ours)	2024	11 (2-5 finger & Human)	345	30.4M	Ranked	GraspIt + Isaac Sim [20]	✓

a few grasping datasets contain multiple grippers in the literature. Fast-Grasp'D [19] and GenDexGrasp [8] used 3 and 5 grippers, respectively. Our dataset has more diverse grippers. 2) The grasps in our dataset are ranked according to the object fall-off time in Isaac Sim [20]. In order to verify whether a grasp is successful or not, we close the fingers of a gripper according to the grasp in Isaac Sim and check if the object will fall or not. The fall-off time of the grasped object is a good measurement of grasp quality. We recorded the fall-off time for each grasp in our dataset, and used it to rank the grasps. Ranking of grasps is useful for learning models that consider grasp quality in grasp synthesis. 3) The grippers in our dataset including a human hand, are aligned using a fixed grasp pose convention for the palm. This property enables us to transfer grasps from one gripper to another gripper. The grasp transfer significantly increases the number of successful grasps for each gripper in our dataset. Since grasp transfer is critical in skill sharing between robots and skill learning from human demonstration, our dataset can be used to study grasp transfer between different grippers.

In summary, our paper has the following contributions:

- We constructed a large-scale robotic grasping dataset that contains 30.4M grasps of 11 different grippers on 345 objects. These grippers range from two-finger grippers to five-finger grippers, including a human hand.
- The grasps in our dataset are verified in Isaac Sim, and ranked according to the object fall-off time. The grasp ranking can be used to learn models for grasp quality prediction.
- The grippers in our dataset are aligned, which can be utilized to study grasp transfer among different grippers.

II. RELATED WORK

A. Grasping Datasets

Learning-based grasp planning methods are crucially dependent on a labeled dataset of high-quality grasps for training. Table I compares our MultiGripperGrasp dataset with existing datasets for robotic grasping. Several previous datasets for parallel jaw grippers have been introduced for image-based grasp prediction [21], with extensions to suction grippers [22]. Recent datasets for parallel jaw grippers [10], [13] are much larger in scale in terms of the number of objects and grasps.

Most multifinger datasets focus on a single gripper such as the Barrett hand [23]–[25], the Shadow hand [6], [11] or

human hand [12], [15]. As a result, the models learned from these datasets cannot be easily transferred to other grippers. Only a few datasets [8], [11], [17], [19] provide grasps across multiple different grippers. Unigrasp [17] contains 12 different grippers in their dataset, but all have no more than 3 fingers, thereby making it insufficient for dexterous grasping. Several differentiable grasping methods [8], [11], [19] are also used to construct grasping datasets with multiple grippers. However, these datasets are limited to 3 or 5 grippers as shown in Table I.

B. Grasp Synthesis Approaches

Analytical Methods. Generating high-quality grasps synthetically has been studied for decades. Classical grasp synthesis methods [22], [23], [26] have proposed a variety of analytic solutions [14], [27] that reason about contact wrenches between a gripper and an object. These methods optimize a grasp quality metric [27] that measures the force closure property of a grasp and its ability to resist external forces and torques. Gradient-free methods, such as simulated annealing [26] are used to find grasp poses and joint angles that can achieve good grasp quality. Recent analytic methods focus on improving grasp synthesis for multi-fingered grippers. For example, [28] constrains the optimization over a lower-dimensional search space via the concept of eigen-grasps. [29] speeds up the optimization by introducing ellipsoids for friction cone approximations.

Differentiable Grasping. The other spectrum of grasp synthesis approaches leverages gradient-based techniques to optimize the search space using different notions of differentiable grasp quality metrics. Some reformulate the analytic force closure estimation in a differentiable setting, with [30] using a simplified version to increase the synthesis speed and [6] focusing on modified contact constraints between the gripper and the object. However, these methods are still slow to converge, which limits their application to the creation of large-scale datasets. Subsequently, [11] proposes alternatives for pose initialization and contact penetration energies. In a similar vein, [31] introduces a simulation protocol that can scale up to a higher number of contacts for contact-rich human hand manipulation and [19] extends it for robotic hands with a much faster grasp synthesis algorithm.

C. Learning-based Grasping

Learning-based methods for grasping have recently found favor due to their capability of using partial observations for

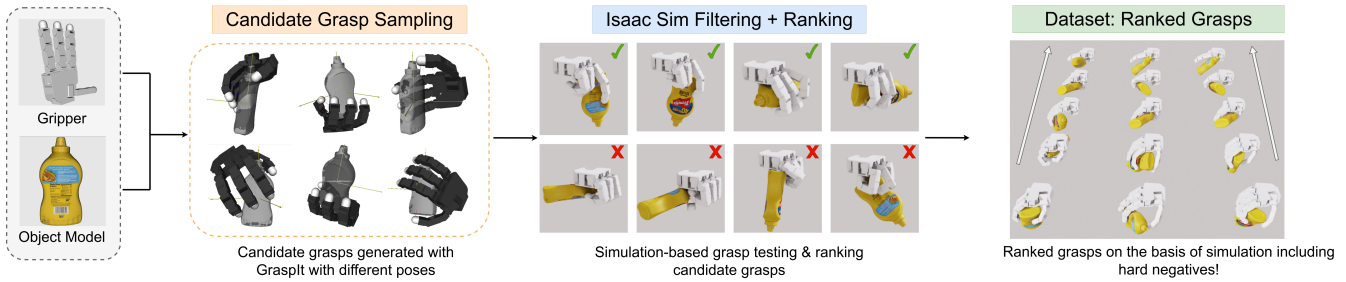


Fig. 2: Our dataset creation pipeline: Given a pair of an object and a gripper, GraspIt! is used to generate candidate grasps that are then filtered using Isaac Sim to produce ranked grasps.



Fig. 3: A sample set of objects used for grasp generation.

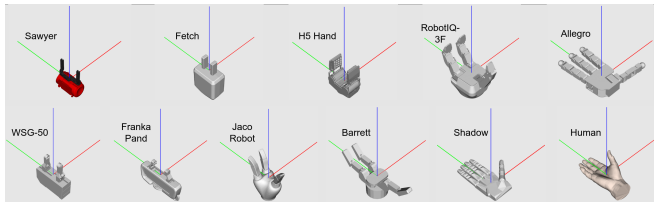


Fig. 4: Aligned grippers in our dataset with varying dexterity.

grasp planning, such as images or point clouds of objects. These methods learn direct regression mappings [21], [22], generative models [5], [25] or implicit functions [32], [33] conditioned on sensory input. In terms of input mode, some of the earlier work used RGB-D images [21], [22]. With advances in point-cloud representations [34], recent methods such as [5], [7], [33] leverage point clouds, making it easier to reason about contacts between grippers and objects.

In terms of gripper representation, most learning-based grasping methods have focused on parallel jaw grippers [5], [7], [21], [22] due to the widespread availability of large-scale datasets of parallel jaw grippers [10], [13]. Multifinger grasping methods usually employ generative models [24], [25], [35] to reason about gripper 6D pose and finger joint values simultaneously. At inference time, a large number of grasps are sampled, which are later refined to fit better to the object. Several methods have been proposed to generalize the grasp prediction between different grippers by using contact points as a bridge, where contact points can be represented by implicit distance functions [32], auto-regressive models [9] or hand-agnostic contact maps [8].

III. METHOD: DATASET GENERATION

The pipeline of our dataset creation is illustrated in Fig. 2. It starts with a pair of an object and a gripper. Then a set of initial grasps is generated using GraspIt! [26], followed by the filtering and ranking of these grasps in Isaac Sim [20]. We describe each component of the pipeline in detail.

A. Objects and Grippers

Object Set. We chose a subset of 329 objects from the GoogleScannedObjects dataset [36] as our core object set. These are 3D scanned objects in everyday settings. The subset is chosen according to the FewSOL [37] dataset for few-shot object recognition in robotic settings. Furthermore, we also included 16 objects from the YCB object set [38] due to the popularity of the YCB objects in robot manipulation. Some examples of objects in our dataset are shown in Fig. 3. The ground truth mass and friction coefficients of each object is unknown to us, consequently, we used a fixed density (100 kg/m^3) and friction coefficients (0.5).

Gripper Set. The gripper set in our dataset contains a wide variety of grippers from parallel jaw grippers to dexterous hands. In total, there are 11 grippers with 5 parallel jaw grippers, 3 three-finger grippers, and 3 dexterous hands as shown in Fig. 4:

- 2-finger grippers: Fetch, Franka Panda, WSG50, Sawyer, H5 Hand
- 3-finger grippers: Barrett, Robotiq-3F, Jaco Robot
- 4-finger grippers: Allegro
- 5 finger grippers: Shadow, Human Hand

Methods using classical grasp quality metrics [6], [26] and contact-energy optimization require specifying contact regions on the gripper fingers and palm. For each gripper, we selected such regions using the graphical interface in GraspIt. To transfer grasps between different grippers, a common notion of gripper pose was used as illustrated by the axes in Fig. 4. The translation refers to the palm center of the gripper, and the orientation is with respect to a canonical pose of the gripper palm pointing in a fixed direction. Thus, any grasp pose from a gripper can be transferred to another by using this pose alignment. For collision checking of these grippers in Isaac Sim, we use a convex decomposition over the gripper meshes.



Fig. 5: Illustration of the grasp ranking process with Isaac Sim.

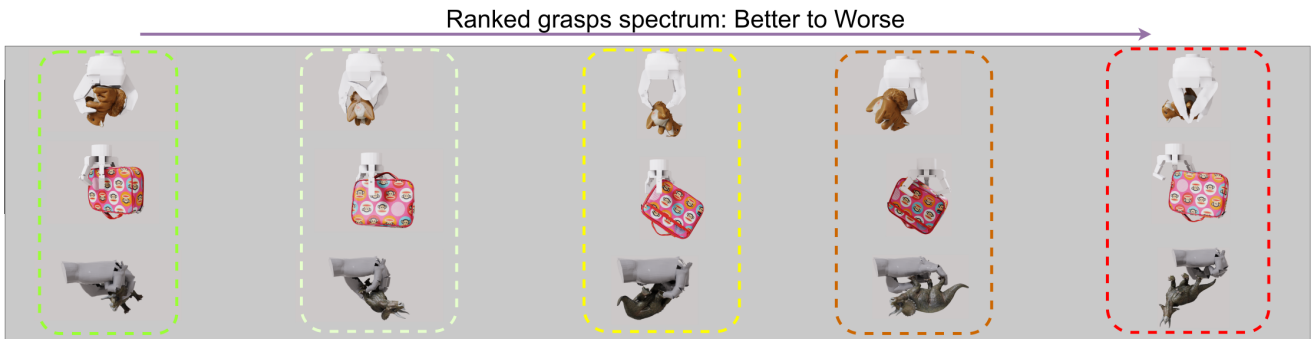


Fig. 6: Visualization of ranked grasp labels and negative examples illustrating the advantage of going beyond binary labels.

B. Grasp Generation

We use GraspIt! [26] to generate candidate grasps for each pair of an object and a gripper. GraspIt! fits our needs for the initial phase of grasp generation as it is easy to integrate novel grippers into its framework and is less memory intensive. An alternative could be to use a differentiable grasping method to generate the initial grasps. For each object-gripper pair, we generated around 8,000 candidate grasps with the caveat that not all of them will pass the simulation filtering check. Here, a contact-based energy formulation was chosen for the simulated annealing optimization used by GraspIt! with manually specified contact points on all grippers.

Specifically, GraspIt! proceeds by sampling a large number of 6-DOF poses for the gripper around the object as a starting configuration for optimizing a contact-based energy function. A queue of best few grasps is updated during the optimization until a maximum number of iterations are reached, after which the grasp is executed and analytical quality metrics are computed. Having a large number of grasp proposals was necessary to ensure grasp diversity.

C. Grasp Labeling

Motivation. To further evaluate the quality of the grasps generated by GraspIt!, we use a robotics simulator known as Isaac Sim [20] to verify these grasps, since GraspIt! only considers object and gripper geometry for grasp analysis. Crucially, we provide a ranked order for these grasps based on the simulation statistics instead of simply providing grasps that pass the simulation test as positive grasps. This allows some grasps in the dataset to be used as negative examples in learning-based methods that use grasp evaluators such

as 6D GraspNet [5]. These negative grasps are of higher quality than random sampled grasps, since they represent hard negative examples during training.

Procedure. The general idea of grasp labeling is to check if a grasped object will fall or not due to gravity in simulation as illustrated in Fig. 5. To test the grasps in Isaac Sim, the gripper is placed so that its palm normal is aligned with the direction of gravity, and the object is initialized according to its pose with respect to the gripper. This set up allowed us to reliably make gravity a adversarial force to every grasp. The joint positions of the gripper fingers are then initialized according to the given grasp. We designed a controller to gradually close the gripper fingers until contact with the object is established. Contact between the fingers and the object triggers the gravity force. Subsequently, physics is simulated over a fixed time interval or until the object falls. The maximum grasp test time was set to 3 seconds.

Stability Measure. Grasps are evaluated according to the time stamp when the object falls. When gravity is applied, a timer is started. Then we recorded the fall-off time for the object as a grasp quality measurement, with higher values for more stable grasps. The object fall-off time is recorded either when the object loses contact with the gripper fingers or when the maximum test time has expired. In the latter case, the grasp will be given the maximum test time as the fall-off time. Furthermore, the object is considered to have fallen when the current pose is 0.5 meters away from its initial position. If the gripper cannot make contact with the object, a negative value is assigned as the fall-off time, and the grasp is considered as a failed grasp. The gripper contacts with the object were calculated using the Rigid Body API from Isaac

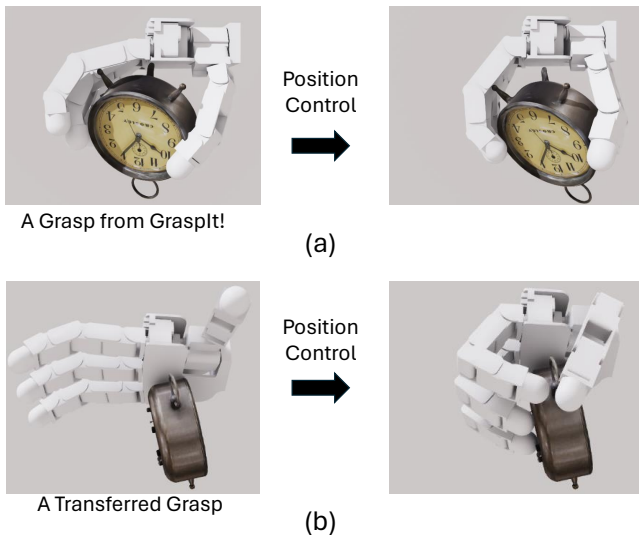


Fig. 7: Position control for closing fingers. (a) A grasp generated from GraspIt! (b) A transferred grasp

Sim [20]. Fig. 6 shows some ranked grasps according to their object fall-off time.

D. Controller Design

The grasps, when initialized within Isaac Sim, are in contact or close to contact with the objects. However, a gripper controller is needed to exert force on the object to perform the grasp. For parallel jaw grippers, the control can be performed by setting the two fingers to their close position. On the other hand, for multi-finger grippers some joints must remain static, e.g., lateral movement of the fingers. Given the large amount of gripper, object, and pose combinations within our dataset, we opted for a simple controller that could be easily used with all the grippers. For each gripper, we manually identify a subset of joints to move and use a position controller to reach the gripper closing configuration. The following versions of the controller were implemented:

Position Control for Grasp Ranking. For grasps generated from GraspIt!, We simply control the root joints of the fingers to reach their closing positions (see Fig. 7(a)). The closing positions are manually specified for each gripper.

Position Control for Grasp Transfer. To transfer a grasp from a source gripper to a target gripper, we first set the joint configuration of the target gripper to the open configuration. Then the controller closes all fingers until it comes into contact with the object. After that, the controller sets only the root joints of the fingers to reach their closing position (see Fig. 7(b)).

IV. EXPERIMENTS

To build our dataset, we conducted experiments to rank grasps generated from GraspIt! and transfer grasps from one gripper to another.

TABLE II: Success rates and failure rates against object fall-off times for different grippers in our dataset.

Gripper	#Fingers	Success Rate (%)				Failure Rate (%)	#Grasps (M)
		> 3s	> 2s	> 1s	> 0s		
Fetch	2	69.42	69.64	70.37	96.34	3.66	2.73
Franka	2	70.17	70.43	71.05	98.09	1.91	2.76
H5 Hand	2	1.18	1.27	1.47	83.9	16.4	3.38
Sawyer	2	48.15	48.60	49.77	87.31	12.69	2.49
WSG 50	2	61.29	61.51	62.24	96.86	3.14	2.75
Barrett	3	45.29	46.12	49.39	93.79	6.21	2.62
Jaco	3	52.20	52.76	54.3	98.96	1.04	2.76
Robotiq	3	76.09	77.03	79.03	99.94	0.06	2.72
Allegro	4	44.27	45.76	50.15	99.87	0.13	2.77
Shadow	5	4.37	4.67	6.72	96.66	3.34	2.78
Human	5	13.56	14.38	17.21	92.84	7.16	2.65

A. Grasp Ranking

For the 30.4 million grasps across 3,795 different object-gripper pairs, our simulation was able to record the object fall-off time for each grasp. Table II shows the amount of grasps simulated per gripper, the percentage of object fall-off times over a threshold per gripper and the failure rate of the grasps simulated, i.e. the percentage of grasps that were not able to make contact with the object. Based on the object fall-off time, the results showed that out of the 30.4 million grasps, 13.16M are successful grasps (the object did not fall within 3s), 0.17M lasted from 2s to 3s, 0.53M lasted from 1s to 2s, 14.97M fell shortly after the gripper touched the object (0s to 1s) and 1.59M are classified as failed grasps.

From Table II, we can see that the Robotiq gripper achieves the highest success rate. We observed low success rates for three grippers in our dataset: the H5 Hand, the Shadow Hand, and the human hand. For the H5 hand, GraspIt! had difficulties in planning good grasps due to the presence of unsupported mimic joints. Therefore, the grasps through simulated annealing were sub-optimal. In the case of Shadow hand and human hand, we observed that our controller is sub-optimal for many grasps when considering highly articulated grippers with complex joints. We observed that the generated joint values for the thumbs made the optimal control non-constant. In some cases, the joints may be aligned in a way that our controller creates a stable grasp, while in others it applies adversarial forces to the grasp. The question of how to design a better controller for multi-finger hands remains a challenging task.

B. Grasp Transfer

Our simulation was able to evaluate the object fall-off time for a large amount of generated grasps. The successful grasps of one gripper can represent successful grasps in others and increase the overall amount of successful grasps in the dataset. To test this hypothesis, we implemented the grasp transfer of successful grasps from one gripper to others and evaluated the transferred grasps using Isaac Sim.

We utilize the alignment between grippers as described in Sec. III-A to transfer grasps. We first transform a source gripper pose to its aligned pose, and then transform the aligned pose to the target gripper. Fig. 8 illustrates an example for grasp transfer. To execute the new grasp, the target gripper is



Fig. 8: An example for grasp transfer: original GraspIt! grasp (left), transferred grasps (right).

TABLE III: Confusion matrix of grasp transfer. Success rates of grasps transferred from a source gripper (row) to a target gripper (column) are presented. The last column represents the number of unique grasps transferred from the source gripper, and the last row is the number of successful grasps across all the transferred grasps for the target gripper.

Source \ Target	Fetch	Franka	H5 Hand	Sawyer	WSG 50	Barret	Jaco	Robotiq	Allegro	Shadow	Human	#Source Grasps (K)
Fetch	1	0.91	0.84	0.86	0.45	0.71	0.85	0.77	0.32	0.38	0.25	147
Franka	0.84	1	0.71	0.80	0.38	0.64	0.72	0.68	0.31	0.37	0.22	145
H5 Hand	0.05	0.02	1	0.03	0.04	0.45	0.11	0.50	0.16	0.07	0.03	24
Sawyer	0.88	0.76	0.57	1	0.15	0.57	0.71	0.65	0.29	0.30	0.15	117
WSG 50	0.46	0.39	0.36	0.51	1	0.56	0.57	0.63	0.23	0.25	0.12	152
Barrett	0.60	0.49	0.48	0.60	0.41	1	0.59	0.77	0.23	0.29	0.19	149
Jaco	0.76	0.65	0.64	0.70	0.31	0.70	1	0.74	0.30	0.35	0.23	140
Robotiq	0.42	0.34	0.35	0.39	0.31	0.58	0.38	1	0.19	0.23	0.13	167
Allegro	0.18	0.12	0.05	0.26	0.16	0.47	0.30	0.51	1	0.42	0.13	158
Shadow	0.18	0.11	0.06	0.26	0.08	0.42	0.38	0.56	0.51	1	0.10	67
Human	0.21	0.23	0.14	0.27	0.07	0.48	0.51	0.62	0.37	0.43	1	141
#Success Target Grasps (K)	778	706	622	789	489	873	836	987	518	508	358	

controlled using the position controller described in Sec. III-D. The joint values at the moment of contact between the gripper and object are recorded, and the object fall-off time is simulated.

Table III shows the confusion matrix of grasp transfer in our experiment. To limit the runtime, we sampled 500 successful grasps per object-gripper pair, where a successful grasp is a grasp with an object fall-off time greater than three seconds. Many successful grasps of one gripper were successful grasps in other grippers when transferred. For example, for the Franka Panda gripper, 561K transferred grasps were evaluated as successful by the simulation. For the H5 Hand, where the original grasps generated from GraspIt! are poor, we were able to generate 598K new successful grasps from other grippers compared to the 24K original successful grasps. The table also shows that some grasps from similar grippers were not successful. Gripper reach and morphology are not taken into account when transferring grasps, thus some grasps may be bad when tested in grippers with different finger shape and size.

V. LIMITATIONS

First, our dataset is limited by the number of objects in the dataset. Scaling our dataset to larger object sets

along with cluttered scenes would be better for downstream tasks, and such aspects can be continually improved upon. Second, most of the datasets in Table I including ours do not consider affordance-driven grasps. Rather than relying solely on object geometry and physics simulation, the dataset can be expanded to include semantic grasps. Finally, our position controller for grasping is sub-optimal. More sophisticated controller design for grippers can also help to obtain better ranking metrics, such as using a force controller to adjust for object disturbances.

VI. CONCLUSION

In this work, we present MultiGripperGrasp, a large-scale grasping dataset with 11 different grippers across the spectrum of two to five finger grippers. Moreover, we go beyond binary labels of success or failure for grasps and include a ranked order based on grasping performance in a physics simulation along with aligning the gripper poses for grasp transfer. Incorporating different classes of grippers also brings out interesting observations in our experiments, especially with respect to leveraging grasp transfer as a simple baseline for augmenting a gripper with lower performing grasps. With MultiGripperGrasp, we hope to help research on generalized grasping [8], [9] and allow the learning

of cross-functional skills between highly different gripper morphologies.

ACKNOWLEDGEMENT

This work was supported by the Sony Research Award Program.

REFERENCES

- [1] C. Ferrari and J. F. Canny, "Planning optimal grasps." in *ICRA*, vol. 3, no. 4, 1992, p. 6.
- [2] C. Borst, M. Fischer, and G. Hirzinger, "Grasp planning: How to choose a suitable task wrench space," in *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004*, vol. 1. IEEE, 2004, pp. 319–325.
- [3] V.-D. Nguyen, "Constructing force-closure grasps," *The International Journal of Robotics Research*, vol. 7, no. 3, pp. 3–16, 1988.
- [4] D. Berenson and S. S. Srinivasa, "Grasp synthesis in cluttered environments for dexterous hands," in *Humanoids 2008-8th IEEE-RAS International Conference on Humanoid Robots*. IEEE, 2008, pp. 189–196.
- [5] 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*, 2019, pp. 2901–2910.
- [6] M. Liu, Z. Pan, K. Xu, K. Ganguly, and D. Manocha, "Deep differentiable grasp planner for high-dof grippers," *arXiv preprint arXiv:2002.01530*, 2020.
- [7] M. Sundermeyer, A. Mousavian, R. Triebel, and D. Fox, "Contact-graspnet: Efficient 6-dof grasp generation in cluttered scenes," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 13 438–13 444.
- [8] P. Li, T. Liu, Y. Li, Y. Geng, Y. Zhu, Y. Yang, and S. Huang, "Gendex-grasp: Generalizable dexterous grasping," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 8068–8074.
- [9] M. Attarian, M. A. Asif, J. Liu, R. Hari, A. Garg, I. Gilitschenski, and J. Tompson, "Geometry matching for multi-embodiment grasping," in *Conference on Robot Learning*. PMLR, 2023.
- [10] 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)*. IEEE, 2021, pp. 6222–6227.
- [11] R. Wang, J. Zhang, J. Chen, Y. Xu, P. Li, T. Liu, and H. Wang, "Dexgraspnet: A large-scale robotic dexterous grasp dataset for general objects based on simulation," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 11 359–11 366.
- [12] S. Hampali, M. Rad, M. Oberweger, and V. Lepetit, "Honnotate: A method for 3d annotation of hand and object poses," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 3196–3206.
- [13] 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*, vol. 5, no. 3, pp. 4368–4375, 2020.
- [14] C. Ferrari, J. F. Canny, *et al.*, "Planning optimal grasps." in *International Conference on Robotics and Automation (ICRA)*, vol. 3, no. 4. IEEE, 1992, p. 6.
- [15] Y.-W. Chao, W. Yang, Y. Xiang, P. Molchanov, A. Handa, J. Tremblay, Y. S. Narang, K. Van Wyk, U. Iqbal, S. Birchfield, *et al.*, "Dexycyb: A benchmark for capturing hand grasping of objects," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 9044–9053.
- [16] M. Macklin, M. Müller, N. Chentanez, and T.-Y. Kim, "Unified particle physics for real-time applications," *ACM Transactions on Graphics (TOG)*, vol. 33, no. 4, pp. 1–12, 2014.
- [17] L. Shao, F. Ferreira, M. Jorda, V. Nambiar, J. Luo, E. Solowjow, J. A. Ojea, O. Khatib, and J. Bohg, "Unigrasp: Learning a unified model to grasp with multifingered robotic hands," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 2286–2293, 2020.
- [18] F. T. Pokorny and D. Kragic, "Classical grasp quality evaluation: New algorithms and theory," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2013, pp. 3493–3500.
- [19] D. Turpin, T. Zhong, S. Zhang, G. Zhu, J. Liu, R. Singh, E. Heiden, M. Macklin, S. Tsogkas, S. Dickinson, *et al.*, "Fast-grasp'd: Dexterous multi-finger grasp generation through differentiable simulation," *arXiv preprint arXiv:2306.08132*, 2023.
- [20] NVIDIA, "Nvidia isaac sim: Robotics simulation and synthetic data," 2023. [Online]. Available: <https://developer.nvidia.com/isaac-sim>
- [21] I. Lenz, H. Lee, and A. Saxena, "Deep learning for detecting robotic grasps," *The International Journal of Robotics Research*, vol. 34, no. 4-5, pp. 705–724, 2015.
- [22] 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.
- [23] C. Goldfeder, M. Ciocarlie, H. Dang, and P. K. Allen, "The columbia grasp database," in *2009 IEEE international conference on robotics and automation*. IEEE, 2009, pp. 1710–1716.
- [24] J. Lundell, E. Corona, T. N. Le, F. Verdoja, P. Weinzaepfel, G. Rogez, F. Moreno-Noguer, and V. Kyrki, "Multi-fingan: Generative coarse-to-fine sampling of multi-finger grasps," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 4495–4501.
- [25] J. Lundell, F. Verdoja, and V. Kyrki, "Ddgc: Generative deep dexterous grasping in clutter," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 6899–6906, 2021.
- [26] A. T. Miller and P. K. Allen, "Graspi! a versatile simulator for robotic grasping," *IEEE Robotics & Automation Magazine*, vol. 11, no. 4, pp. 110–122, 2004.
- [27] E. Rimon and J. Burdick, "On force and form closure for multiple finger grasps," in *Proceedings of IEEE International Conference on Robotics and Automation*, vol. 2. IEEE, 1996, pp. 1795–1800.
- [28] M. Ciocarlie, C. Goldfeder, and P. Allen, "Dexterous grasping via eigengrasps: A low-dimensional approach to a high-complexity problem," in *Robotics: Science and systems manipulation workshop-sensing and adapting to the real world*, 2007.
- [29] T. Tsuji, K. Harada, and K. Kaneko, "Easy and fast evaluation of grasp stability by using ellipsoidal approximation of friction cone," in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2009, pp. 1830–1837.
- [30] T. Liu, Z. Liu, Z. Jiao, Y. Zhu, and S.-C. Zhu, "Synthesizing diverse and physically stable grasps with arbitrary hand structures using differentiable force closure estimator," *IEEE Robotics and Automation Letters*, vol. 7, no. 1, pp. 470–477, 2021.
- [31] D. Turpin, L. Wang, E. Heiden, Y.-C. Chen, M. Macklin, S. Tsogkas, S. Dickinson, and A. Garg, "Grasp'd: Differentiable contact-rich grasp synthesis for multi-fingered hands," in *European Conference on Computer Vision*. Springer, 2022, pp. 201–221.
- [32] N. Khargonkar, N. Song, Z. Xu, B. Prabhakaran, and Y. Xiang, "Neuralgrasps: Learning implicit representations for grasps of multiple robotic hands," in *Conference on Robot Learning*. PMLR, 2022.
- [33] T. Weng, D. Held, F. Meier, and M. Mukadam, "Neural grasp distance fields for robot manipulation," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 1814–1821.
- [34] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," *Advances in neural information processing systems*, vol. 30, 2017.
- [35] W. Wei, D. Li, P. Wang, Y. Li, W. Li, Y. Luo, and J. Zhong, "Dvvg: Deep variational grasp generation for dextrous manipulation," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 1659–1666, 2022.
- [36] L. Downs, A. Francis, N. Koenig, B. Kinman, R. Hickman, K. Reymann, T. B. McHugh, and V. Vanhoucke, "Google scanned objects: A high-quality dataset of 3d scanned household items," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 2553–2560.
- [37] J. J. P. Y.-W. Chao, and Y. Xiang, "Fewsol: A dataset for few-shot object learning in robotic environments," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 9140–9146.
- [38] B. Calli, A. Singh, A. Walsman, S. Srinivasa, P. Abbeel, and A. M. Dollar, "The ycb object and model set: Towards common benchmarks for manipulation research," in *2015 international conference on advanced robotics (ICAR)*. IEEE, 2015, pp. 510–517.