

# DexGraspNet: A Large-Scale Robotic Dexterous Grasp Dataset for General Objects Based on Simulation

Ruicheng Wang<sup>1\*</sup>, Jiali Zhang<sup>1\*</sup>, Jiayi Chen<sup>1,2</sup>, Yinchen Xu<sup>1,2</sup>, Puha Li<sup>2,3</sup>, Tengyu Liu<sup>2</sup>, He Wang<sup>1†</sup>

**Abstract**—Robotic dexterous grasping is the first step to enable human-like dexterous object manipulation and thus a crucial robotic technology. However, dexterous grasping is much more under-explored than object grasping with parallel grippers, partially due to the lack of a large-scale dataset. In this work, we present a large-scale robotic dexterous grasp dataset, DexGraspNet, generated by our proposed highly efficient synthesis method that can be generally applied to any dexterous hand. Our method leverages a deeply accelerated differentiable force closure estimator and thus can efficiently and robustly synthesize stable and diverse grasps on a large scale. We choose ShadowHand and generate 1.32 million grasps for 5355 objects, covering more than 133 object categories and containing more than 200 diverse grasps for each object instance, with all grasps having been validated by the Isaac Gym simulator. Compared to the previous dataset from Liu et al. generated by *GraspIt!*, our dataset has not only more objects and grasps, but also higher diversity and quality. Via performing cross-dataset experiments, we show that training several algorithms of dexterous grasp synthesis on our dataset significantly outperforms training on the previous one. To access our data and code, including code for human and Allegro grasp synthesis, please visit our project page: <https://pku-epic.github.io/DexGraspNet/>.

## I. INTRODUCTION

Robotic object grasping is an important technology for many robotic systems. Recent years have witnessed great success in developing vision-based grasping methods [1–6] and large-scale datasets for parallel-jaw grippers, *e.g.*, synthetic object-centric dataset, ACRONYM [7], and real-world dataset of grasping in clutter, GraspNet [3].

Although simple and effective for pick-and-place, parallel-jaw grippers show certain limitations in dexterous object manipulation, *e.g.*, using scissors, due to their low DoFs. On the contrary, multi-fingered robotic hands, *e.g.*, ShadowHand [8], are human-like, designed with very high DoFs (26 for ShadowHand), and can attain more diverse grasp types. Those dexterous hands can support many complex and diverse manipulations, *e.g.*, solving Rubik’s cube [11], and can be used in task-specific grasping [12].

Arguably, dexterous grasping is the first step to dexterous manipulation. However, dexterous grasping is highly under-explored, compared to parallel grasping. One major obstacle is the lack of large-scale robotic dexterous grasping datasets required by learning-based methods. Up to now, the only dataset is provided by Liu et al. [9] (Deep Differentiable Grasp, referred to as DDG), which contains only 6.9K grasps

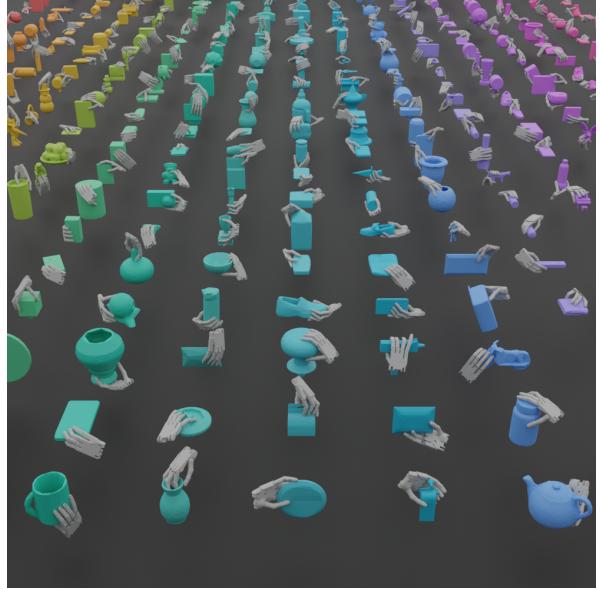


Fig. 1: A visualization of DexGraspNet. DexGraspNet contains 1.32M grasps of ShadowHand [8] on 5355 objects, which is two orders of magnitudes larger than the previous dataset from DDG [9]. It features diverse types of grasping that cannot be achieved using *GraspIt!* [10].

and 565 objects and is much smaller than the grasp datasets for parallel grippers, *e.g.*, GraspNet [3], ACRONYM [7]. Considering the high-DoF nature of the dexterous hand, dexterous grasping datasets need to be significantly larger and more diverse for the sake of generalization.

In this work, we propose DexGraspNet, a large-scale simulated dataset for robotic dexterous grasping. This dataset contains 1.32 million dexterous grasps for ShadowHand on 5355 objects, with more than 200 diverse grasps for each object instance. The objects are from more than 133 hand-scale object categories and collected from various synthetic and scanned object datasets. In addition to the scale, our dataset also features high diversity and high physical stability. All grasps have been examined by force closure and validated by Isaac Gym [13] physics simulator, enabling further tasks in both real-world and simulation environments.

Note that synthesizing diverse high-quality dexterous grasps at scale is known to be very challenging. For dexterous grasping data synthesis, previous works, *e.g.*, DDG, mainly use *GraspIt!* [10], which lacks diversity in grasping poses due to its naive search strategy. A recent work [19] proposes a novel method to address this diversity issue. This work devises a differentiable energy term to approximate

<sup>1</sup>Peking University

<sup>2</sup>Beijing Institute for General Artificial Intelligence

<sup>3</sup>Tsinghua University

\*Equal contribution

†Corresponding author: [hewang@pku.edu.cn](mailto:hewang@pku.edu.cn)

TABLE I: Dexterous Grasp Dataset Comparison

Dataset	Hand	Observations	Sim./Real	Grasps	Obj.(Cat.)	Grasps per Obj.	Method
ObMan [14]	MANO	-	Sim.	27k	2772(8)	10	<i>GraspIt!</i>
HO3D [15]	MANO	RGBD	Real	77k	10	>7k	Estimation
DexYCB [16]	MANO	RGBD	Real	582K	20	>29k	Human annotation
ContactDB [17]	MANO	RGBD+thermal	Real	3750	50	75	Capture
ContactPose [18]	MANO	RGBD	Real	2306	25	92	Capture
DDGdata [9]	ShadowHand	-	Sim.	6.9k	565	>100	<i>GraspIt!</i>
DexGraspNet (Ours)	ShadowHand	-	Sim.	<b>1.32M</b>	<b>5355(133)</b>	>200	Optimization

force closure and then uses it to synthesize diverse and stable grasps via optimization. However, [19] suffers from low yield, slow convergence, and strict constraints on object meshes, making it infeasible for us to use for synthesizing a large-scale dataset.

To achieve our desired diversity, quality, and scale, we propose several critical improvements to [19], making it much more efficient and robust. First, we design a better hand pose initialization strategy and carefully select contact candidates to boost yield. For synthesizing 10000 valid grasps, we speed up from 400 GPU hours to 7 GPU hours. Second, we propose an alternative way to compute penetration energy and signed distances, which enables us to handle object meshes of much lower quality, and also highly simplifies their preprocessing procedures. Third, we introduce energy terms that punish self-penetration and out-of-limit joint angles to further improve grasp quality. Additionally, with simple modifications, the entire pipeline can be applied to other dexterous hands, such as MANO [20] and Allegro.

To verify the advantage of our dataset over the one from DDG, we train two dexterous grasping algorithms on our dataset and DDG. The cross-dataset experiments confirm that training on our dataset yields better grasping quality and higher diversity. Also, the great diversity of the hand grasps from our dataset leaves huge improvement space for future dexterous grasping algorithms.

## II. RELATED WORK

Researches in grasping can be broadly categorized by the types of end effectors involved. The most thoroughly studied ones are the suction cup and parallel jaw grippers, whose grasp pose can be defined by a 7D vector at most, including 3D for translation, 3D for rotation, and 1D for the width between the two fingers. Dexterous robotic hands with three or more fingers such as ShadowHand [8] and humanoid hands such as MANO [20] require more complex descriptors, sometimes up to 24DoF as in ShadowHand [8]. In this paper, we are dedicated to researches on the latter type. To bridge the gap between humanoid hands and robotic hands, numerous researches have shown the efficacy of retargeting humanoid hand poses to dexterous robotic hands [21–24].

### A. Analytical Grasping

Early researches in dexterous grasping focus on optimizing grasping poses to form force closure that can resist external forces and torques [25–28].

Due to the complexity of computing hand kinematics and testing force closure, many works were devoted to simplifying the search space [29–31]. As a result, these methods

were applicable to restricted settings and can only produce limited types of grasping poses. Another stream of work [32–34] looks for simplifying the optimization process with an auxiliary function. [19] proposed to use a differentiable estimator of the force closure metric to synthesize diverse grasping poses for arbitrary hands.

### B. Data-Driven Grasping

Recent works shift their focus to data-driven methods. Given an object, the most straightforward approach is to directly generate the pose vectors of the grasping hand [35–39]. A refinement step is usually implemented in these methods to remove inconsistencies such as penetration.

Other methods take an indirect approach that involves generating an intermediate representation first. Existing methods use contact points [40–42], contact maps [21, 22, 43–45], and occupancy fields [46] as the intermediate representations. The methods then obtain the grasping poses via optimization [40, 41, 44, 46], planning [43], RL policies [22, 42], or another generative model [45].

Compared to most analytical methods, data-driven methods show improved inference speed and diversity of generated grasping poses. However, the diversity is still limited by the training data.

### C. Dexterous Grasp Datasets

Dexterous grasping is impossibly difficult to annotate for its overwhelming degrees of freedom. Most existing works are trained on programmatically synthesized grasping poses [9, 14, 38, 47] using the *GraspIt!* [10] planner. The planner first searches the eigengrasp space for pregrasp poses that cross a threshold. Then, the planner squeezes all fingers in the selected pregrasp poses to construct a firm grasp. Since the initial search is performed in the low dimensional eigengrasp space, the resulting data follows a narrow distribution and cannot cover the full dexterity of multi-finger hands.

More recent works leverage the increasing capacity of computer vision to collect human hand poses when interacting with the object. HO3D [15, 48] computes the ground truth 3D hand pose for images from 2D hand keypoint annotations. The method resolves ambiguities by considering physics constraints in hand-object interactions and hand-hand interactions. DexYCB [16] and ContactPose [18] solve the 3D hand shape from multi-view RGBD camera recordings. Latest datasets [49–51] use optical motion capture systems to track hand and object shapes during interactions. While these

methods produce natural and smooth interaction demonstrations, the data is restricted to humanoid hand structures and daily hand poses.

In addition, ContactDB [17] and ContactPose [18] leverage IR cameras to collect contact maps on object surfaces.

### III. DATASET GENERATION METHOD

#### A. Object Preparation

We collect object models from various datasets. ShapeNetCore and ShapeNetSem [52, 53] contain Computer-Aided-Design (CAD) models with category labels, from which we select 3980 objects in 133 categories. YCB [54], Big-BIRD [55], Grasp [56], KIT [57], and Google’s scanned Object Dataset [58] are scanned model repositories without category labels, from which we select 1375 objects. They are not labeled with categories in our dataset.

Since not all objects from these datasets are aligned to sizes in the real world, we choose to normalize all models into a unit sphere and then augment each object by uniformly scaling them with 5 fixed sizes. Then we remesh them into manifolds [59] to make closed figures. Finally, for simulation purposes, we create collision meshes for every object mesh through convex decomposition using CoACD [60].

#### B. Grasp Generation

1) *Assumptions and Notations*: To parameterize dexterous grasps, we use tuples  $g = (T, R, \theta)$ , where  $T \in \mathbb{R}^3$  and  $R \in SO(3)$  form the global hand pose, and  $\theta \in \mathbb{R}^d$  describes the  $d$  joint angles ( $d = 22$  for ShadowHand). Given the URDF of the dexterous hand, we can use  $g$  to compute the hand mesh  $H$  via forward kinematics. In the generation process, we need to consider the contact points. Following [19]’s idea that fewer contact candidates lead to faster convergence, we first manually select 140 contact candidates from the surface of the hand, and  $n$  contact points are randomly sampled among these candidates in each iteration of the optimization to calculate energy terms. We then augment our grasp tuple to  $g' = (T, R, \theta, x)$ , with  $x$  representing the  $n = 4$  contact points. Given object mesh  $O$ , the contact normal vectors  $c \in \mathbb{R}^{n \times 3}$  can be computed from  $x$ . Note that  $x$  is an intermediate variable of the grasp generation algorithm and is discarded after the algorithm finishes.

2) *Review of Differentiable Force Closure*: Our dexterous grasp generation method is mainly built upon the original work [19], in which they propose a novel differentiable force closure estimator as an energy term and use optimization to synthesize grasps. The proposed differentiable force closure term, which encourages a set of contact points to form force closure, can be expressed as

$$E_{fc} = \|Gc\|_2 \quad (1)$$

where

$$G = \begin{bmatrix} I_3 & \cdots & I_3 \\ [x_1]_\times & \cdots & [x_n]_\times \end{bmatrix}$$

$$[x_i]_\times = \begin{bmatrix} 0 & -x_i^{(z)} & x_i^{(y)} \\ x_i^{(z)} & 0 & -x_i^{(x)} \\ -x_i^{(y)} & x_i^{(x)} & 0 \end{bmatrix}$$

A pair of attraction and repulsion energy functions are introduced to ensure contact and prevent penetration:

$$E_{dis} = \sum_{i=1}^n d(x_n, O), E_{pen} = \sum_{v \in S(H)} [v \in O] d(v, O) \quad (2)$$

where  $S(H)$  is the surface point cloud of the hand mesh  $H$ ,  $d(\cdot, \cdot)$  is the point-to-mesh distance and  $[v \in O] = 1$  if point  $v$  is inside object mesh  $O$ . They also use an energy function  $E_{prior}$  to keep the hand in a natural state. The complete energy function is as follows:

$$E = E_{fc} + w_{dis}E_{dis} + w_{pen}E_{pen} + w_{prior}E_{prior} \quad (3)$$

They design a modified MALA optimization algorithm to minimize the energy  $E$  over the augmented grasp tuple  $g' = (T, R, \theta, x)$ . The algorithm takes an initial hand pose  $g'_0$ , which is randomly initialized. Then, in each iteration,  $T, R, \theta$  are updated according to Langevin dynamics, and contact points  $x$  are randomly re-sampled with a small probability. The update is accepted or rejected stochastically by the Metropolis-Hastings rule. The optimization ends after 10000 steps. For more details, please refer to [19].

3) *Our Method*: Though [19] takes a great step forward, it is still quite hard to obtain a large-scale grasp dataset directly using their method due to three reasons: 1. the algorithm suffers from a low success rate and slow convergence; 2. most object meshes we use have no thickness due to the poor quality of the object dataset, making it impossible to compute penetration energy; 3. due to random initialization, some generated grasping poses may look twisted. To overcome these issues, we propose several ways to improve the efficiency, effectiveness, and robustness of the original algorithm:

First, we propose an initialization strategy that can greatly improve the success rate and speed up the convergence. We find that their optimization algorithm’s success rate drops dramatically when the initial hand is closed. This motivates us to introduce two constraints to the initialization strategy: 1. the five fingers should be opened to form a space for grasping; 2. the palm should face the object.

More specifically, we manually choose a canonical hand pose  $\theta_{ref}$ , as shown in Figure 2, then jitter each joint angle within its limit using the truncated normal distribution. Then, on each object mesh, we first take its convex hull, then push every vertex of the hull away from the origin by  $0.2m$  to obtain the inflated convex hull. Next, we sample a random point  $p$  on the surface of the inflated convex hull, and compute the direction vector from  $p$  to its nearest point on the original object mesh, then jitter this direction vector within a cone and get  $\vec{n}$ . Finally, the hand is moved to  $p$ , and rotated to face the same direction as  $\vec{n}$ , then push away from the object mesh along  $\vec{n}$  by a random distance, and rotated around  $\vec{n}$  randomly.

The resulting initial hand pose  $(T_0, R_0, \theta_0)$  can be easily optimized to a grasp pose, thus raising the success rate. Moreover, for each object, if we sample enough initial hands, they can surround the object densely and evenly, so we can

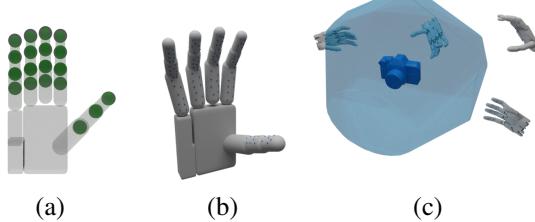


Fig. 2: (a) Green spheres with the radius of  $\delta = 1\text{cm}$  are manually selected to compute  $E_{\text{spen}}$ . (b) Contact candidates on the collision mesh in the canonical initial hand pose. (c) Initialization: 1. sample points on the object’s inflated convex hull (blue); 2. move the hands to the sampled points and jitter the translation, rotation, and joint angles.

generate diverse data. Also, we found that using our strategy, the final grasp poses look more natural.

Second, we propose an alternative way to compute penetration energy that can make the algorithm more robust to thin object meshes of low quality. In practice, the original algorithm will fail completely when the object mesh has no thickness because the penetration energy will always be zero. Therefore, instead of taking the point cloud from the hand, we take it from the object and compute each point’s distance to the hand mesh. We call this the reverse penetration energy. It does not require the object mesh to have any thickness at all, allowing us to process far more object CAD models.

Third, we modify  $E_{\text{prior}}$ , inspired by [45] to penalize self penetration and out-of-limit joint angles:

$$E_{\text{spen}} = \sum_{p \in P(H)} \sum_{q \in P(H)} [p \neq q] \max(\delta - d(p, q), 0) \quad (4)$$

$$E_{\text{joints}} = \sum_{i=1}^d (\max(\theta_i - \theta_i^{\max}, 0) + \max(\theta_i^{\min} - \theta_i, 0)) \quad (5)$$

These add to our final energy function:

$$E_{\text{fc}} + w_{\text{dis}} E_{\text{dis}} + w_{\text{pen}} E_{\text{pen}} + w_{\text{spen}} E_{\text{spen}} + w_{\text{joints}} E_{\text{joints}} \quad (6)$$

where  $w_{\text{dis}} = 100$ ,  $w_{\text{pen}} = 100$ ,  $w_{\text{spen}} = 10$ ,  $w_{\text{joints}} = 1$ .

Another minor difference between our implementation and [19] lies in the optimization algorithms. Because our initialization strategy has already raised the success rate of the algorithm to an acceptable level, we simplify MALA and use simple gradient descent to update  $T, R, \theta$  during each optimization step. In our settings, the optimization process converges in less than 6000 iterations, which reduces almost half of the original iterations.

Finally, we use a modified version of Kaolin [61] instead of DeepSDF [62] to compute point-to-mesh signed distances, which eliminates the need for pretraining category level DeepSDF networks, and significantly reduces the memory cost when optimizing grasps.

### C. Grasp Validation

To filter out those bad results after the optimization converges, we validate all of the grasps in a physical simulator

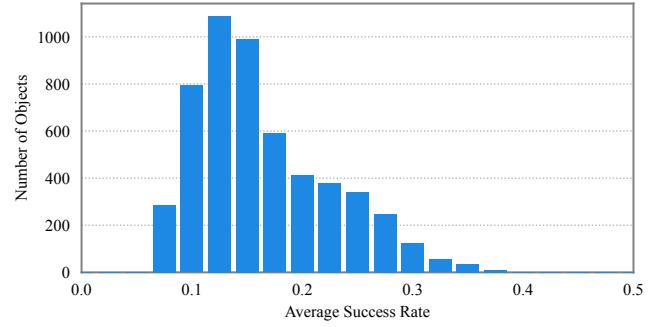


Fig. 3: Distribution of object numbers with respect to the average success rate for each object after final validation. We only keep successful grasps in our dataset.



Fig. 4: Visualization of grasps using different dexterous hands. From left to right: ShadowHand, MANO, Allegro.

Isaac Gym [13] with PhysX as the basic physics engine. We first initialize the gripper using the final grasp parameters. Then, in order to apply active forces on the object, we slightly move each contacting link of the gripper along the normal of its contact point, and set the moved pose as target positions for position control. Finally, gravity with a magnitude of  $9.8\text{m/s}^2$  is added to the scene. A grasp is considered successful if the gripper is still in contact with the object after 100 simulation steps under all 6 axis-aligned directions of gravity. The distribution of the object number with respect to the average success rate for each object is shown in Fig. 3. Moreover, if the max penetration depth exceeds  $0.1\text{cm}$ , we also consider the grasp as a failure. We only save those grasps who pass both the simulation validation and the penetration validation in our dataset.

### IV. DATASET ANALYSIS AND COMPARISON

With our improved pipeline, we generate more than 200 grasps per object, which sum up to 1.32 million grasps in total, forming the largest grasping dataset for ShadowHand. Some visual qualitative results are shown in Fig. 5. Additionally, this pipeline can be stably applied to other dexterous hands. Fig. 4 shows some synthesized results for human hand (MANO [20]) and Allegro along with ShadowHand.

Compared to the original algorithm [19], our improved pipeline achieves a significant speed-up. On NVIDIA A100 with 19.49 TFLOPS, our algorithm takes 74min to optimize 10000 grasps for 6000 steps, out of which about 18% are



Fig. 5: Visualization of the diverse grasps on the objects from DexGraspNet.

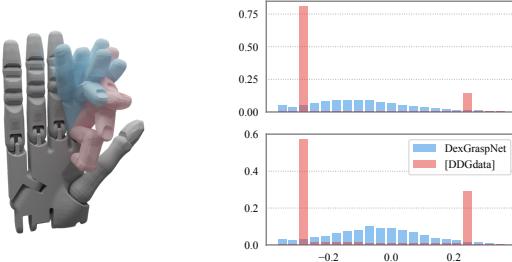


Fig. 6: Diversity comparison between DexGraspNet and DDGdata. Left: The poses of each finger of DDGdata (red) all collapse into a fan-shaped space, while fingers from our data (blue) can reach out in all directions. Right: Probability distribution of two joint angles over all grasp poses. Joint angles from DDGdata typically hover around the two limits, contributing to the dataset’s stiff and inflexible grasp mode.

considered valid under our settings. The original algorithm of [19] takes 37min on NVIDIA 3090 with 35.58 TFLOPS to optimize 512 grasps for 10000 steps, out of which about 3% are considered valid. It took us 950 GPU hours on A100 to generate 1.32 million valid grasps, which would have taken the original algorithm 50000 GPU hours. This speed-up is contributed by faster convergence, smaller memory (which leads to bigger batch size), and a higher success rate.

We demonstrate the quality of our dataset by comparing DexGraspNet with the dataset proposed in DDG [9] (in short DDGdata), a grasping dataset for ShadowHand generated by *GraspIt!* [10], on two aspects below.

First, we conclude that DexGraspNet is more diverse. As shown in Fig. 6, the planner in *GraspIt!* can only clench each finger in a fixed direction, so the root joints of every finger lose one DoF each. Moreover, the angles of many joints in DDGdata often collapse to their upper or lower limits due to their simple generation strategy. These phenomena contribute to a serious loss of diversity. In contrast, our optimization method can generate diverse grasps with much higher dexterity, as shown in Fig. 5. We further use the mean entropy to model the diversity quantitatively. To evaluate this metric, we first discretize each joint’s motion range into 100 bins, then use samples from each dataset to estimate a probability distribution, calculate the entropy of these distributions, and take the mean over all joints. Results are shown in Table II.

Second, we show that DexGraspNet is more stable by

TABLE II: Statistics of  $Q_1$  and Entropy

Dataset	100% $Q_1$ mean	best 10% $Q_1$ mean	$H$ mean
DDGdata	0.0712	0.2277	4.246
DexGraspNet	<b>0.1145</b>	<b>0.2533</b>	<b>5.962</b>

comparing the  $Q_1$  metric [63], which is intuitively the norm of the smallest wrench that can destabilize the grasp:

$$Q_1 = \text{inscribed sphere radius of ConvexHull}(\cup_i w_i) \quad (7)$$

where  $\{w_i\}$  are contact friction cone wrenches. We choose 1mm as the contact threshold, and allow at most one contact point for each link to save computational time. The results are shown in Table II. We can find that the average value of our dataset is significantly better than that of DDGdata. It is worth noting that our entire generation pipeline does not explicitly optimize these metrics.

## V. BENCHMARKS

We benchmark two methods of dexterous grasp synthesis, DDG [9] and GraspTTA [35], on our dataset, and compare them with the same methods trained on DDGdata [9].

### A. Benchmark Methods

DDG [9] designs a differentiable  $Q_1$  metric, which generalizes the standard  $Q_1$  metric to the case when the gripper is not in contact with objects. With this generalized  $Q_1$  metric, they are able to supervise the neural network to predict fine grasp end-to-end. Their network takes 5 depth images of the object as input and directly regresses 6D pose and joint angles of the ShadowHand. To ease learning, they divide the training process into two stages. In the first stage, they only use the loss of the grasp poses, and in the second stage, they fine-tune the network with differentiable  $Q_1$  loss and other losses to avoid penetration and pull the hand closer to the object. We follow their data pre-process pipeline to generate BVH representations and depth images of our objects, and train the network with official settings on each dataset.

Another work GraspTTA (short for Test-Time Adaptation) [35] proposes to synthesize high-quality grasps by ensuring the contact consistency between the hand and the object. They design two networks, one is a CVAE [64] to synthesize grasps, and the other is a contact net to predict contact regions of the object. During training, those two networks are trained separately. During testing, a grasp is synthesized in a two-stage process. First, the CVAE takes the object point cloud as condition, samples a latent code,

TABLE III: Benchmarks of the Grasp Quality

Method (Training Dataset)	Tested on DexGraspNet			Tested on DDGdata		
	success $\uparrow$	$Q_1 \uparrow$	pen $\downarrow$	success $\uparrow$	$Q_1 \uparrow$	pen $\downarrow$
DDG (DDGdata)	57.4	0.0493	0.353	56.4	0.0461	0.333
DDG (DexGraspNet)	<b>67.5</b>	<b>0.0582</b>	<b>0.173</b>	<b>75.9</b>	<b>0.0524</b>	<b>0.134</b>
GraspTTA (DDGdata)	17.1	0.0126	0.720	23.7	0.0265	0.666
GraspTTA (DexGraspNet)	<b>24.5</b>	<b>0.0271</b>	<b>0.678</b>	<b>39.3</b>	<b>0.0790</b>	<b>0.547</b>

and then decodes the hand 6D global pose and joint angles, which can be further transformed into the hand point cloud through forward kinematics. Second, the contact net takes both the object and the hand point cloud to predict a target contact map, and optimizes the hand parameters to minimize the difference between the current contact map and the target contact map. We re-implement GraspTTA on ShadowHand, process the data from DexGraspNet and DDGdata in the same way as in [35], and train the networks on each dataset for the same number of iterations.

### B. Experiments and Results

We report the following metrics for evaluation. 1) **Simulation success rate(%) in Isaac Gym.** We adopt an easier criterion (the criterion in Sec. III-C is too strict for the baselines): a grasp is considered valid if it can withstand at least one of the six gravity directions and has a maximal penetration less than 5mm. 2) **Mean  $Q_1$**  [63], which is introduced in Sec. IV. Since these methods cannot guarantee exact contact, we relax the contact threshold to 1cm. Particularly, if the penetration depth is greater than 5mm, the  $Q_1$  metric is not well defined, so we manually set  $Q_1$  of these results to 0. 3) **Maximal penetration depth(cm).** This is defined as the maximal penetration depth from the object point cloud to hand meshes.

The main results are presented in Table III. Comparing models trained on DexGraspNet with models trained on DDGdata, we observe that no matter which baseline, test set, or metric we use, the former always scores higher. We thus conclude that learning-based grasping methods achieve higher performance when they are trained under our dataset. Table III also shows that the output of DDG has higher quality than GraspTTA most of the time. More specifically, GraspTTA suffers severely from penetration.

Apart from grasp quality, we use joint angle entropy (the same as in Section IV) to evaluate the diversity of the grasps generated by the two methods. Table IV shows the joint angle entropy of models trained on DexGraspNet always have higher means and lower standard deviations than models trained on DDGdata, which means DexGraspNet improves the diversity of grasping methods. We also find that GraspTTA has higher diversity than DDG, partly due to the test time optimization used in GraspTTA that can generate many variations. To compare the joint angle entropy of models trained on DexGraspNet and the original joint angle entropy of DexGraspNet, we further find that: 1) DDG’s entropy is lower than DexGraspNet’s entropy, meaning DDG cannot fully recover DexGraspNet’s diversity; 2) although GraspTTA yields an entropy higher than DexGraspNet, this

TABLE IV: Benchmarks of the Grasp Diversity

Method (Training Dataset)	DexGraspNet		DDGdata	
	$H$ mean	$H$ std	$H$ mean	$H$ std
DDG (DDGdata)	4.958	2.653	3.709	1.942
DDG (DexGraspNet)	<b>5.683</b>	<b>1.993</b>	<b>4.272</b>	<b>1.287</b>
GraspTTA (DDGdata)	5.952	0.934	5.837	1.047
GraspTTA (DexGraspNet)	<b>6.111</b>	<b>0.569</b>	<b>5.947</b>	<b>0.528</b>

does not necessarily mean that GraspTTA learns diverse grasping, given its success rate is very low. We interpret this as the trade-offs that DDG and GraspTTA individually make, given that stability and diversity are contradictory to some degree. More importantly, this status quo shows that none of the existing grasping methods can fully learn the highly diverse grasp poses of DexGraspNet while keeping a reasonable success rate at the same time.

### VI. LIMITATIONS

By comparing grasps in our dataset with the taxonomy from [65], we notice that our dataset cannot cover every grasping type described. Since the optimization step tends to pull every candidate point closer to the object, the final grasps are always contact-rich, or power grasps. Therefore, precision grasps hardly appear, which represents the dexterity of multi-finger robotic hands. Additionally, our method lacks semantic guidance, which makes it hard to generate functional grasps, *e.g.* picking up the mug by its handle. Precision grasps and functional grasps remain important issues for us to explore.

### VII. CONCLUSIONS

In this paper, we present a large-scale synthetic dexterous grasping dataset, DexGraspNet, synthesized via our proposed deeply-accelerated optimization-based method. This dataset has a much larger scale, better grasp quality, and higher diversity than previous datasets. Trained on DexGraspNet, previous grasp synthesis methods can achieve consistent improvements in both quality and diversity. However, none of the existing methods can perform well on both metrics. Compared to grasping using parallel grippers, we argue that dexterous grasping has a larger room for research. We release DexGraspNet and hope that its scale, quality, and diversity can help future methods tackle the task of dexterous grasping, and exploit more potential of dexterous grippers.

### VIII. ACKNOWLEDGEMENTS

This work is supported in part by the National Key R&D Program of China (2022ZD0114900) and the Beijing Municipal Science & Technology Commission (Z221100003422004).

## REFERENCES

- [1] M. Breyer, J. J. Chung, L. Ott, R. Siegwart, and J. Nieto, “Volumetric grasping network: Real-time 6 dof grasp detection in clutter,” *arXiv preprint arXiv:2101.01132*, 2021.
- [2] M. Sundermeyer, A. Mousavian, R. Triebel, and D. Fox, “Contact-grasnet: 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.
- [3] H.-S. Fang, C. Wang, M. Gou, and C. Lu, “Grasnet-1billion: A large-scale benchmark for general object grasping,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 11 444–11 453.
- [4] M. Gou, H.-S. Fang, Z. Zhu, S. Xu, C. Wang, and C. Lu, “Rgb matters: Learning 7-dof grasp poses on monocular rgbd images,” in *Proceedings of the International Conference on Robotics and Automation (ICRA)*, 2021.
- [5] C. Wang, H.-S. Fang, M. Gou, H. Fang, J. Gao, and C. Lu, “Graspness discovery in clutter for fast and accurate grasp detection,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2021, pp. 15 964–15 973.
- [6] H. Fang, H.-S. Fang, S. Xu, and C. Lu, “Transeg: A large-scale real-world dataset for transparent object depth completion and a grasping baseline,” *IEEE Robotics and Automation Letters*, pp. 1–8, 2022.
- [7] C. Eppner, A. Mousavian, and D. Fox, “ACRONYM: A large-scale grasp dataset based on simulation,” in *2021 IEEE Int. Conf. on Robotics and Automation, ICRA*, 2020.
- [8] “ShadowRobot,” URL <https://www.shadowrobot.com/dexterous-hand-series/>, 2005.
- [9] 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.
- [10] A. T. Miller and P. K. Allen, “Graspit! a versatile simulator for robotic grasping,” *IEEE Robotics & Automation Magazine*, 2004.
- [11] I. Akkaya, M. Andrychowicz, M. Chociej, M. Litwin, B. McGrew, A. Petron, A. Paine, M. Plappert, G. Powell, R. Ribas, *et al.*, “Solving rubik’s cube with a robot hand,” *arXiv preprint arXiv:1910.07113*, 2019.
- [12] M. Kokic, J. A. Stork, J. A. Haustein, and D. Kragic, “Affordance detection for task-specific grasping using deep learning,” in *2017 IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids)*. IEEE, 2017, pp. 91–98.
- [13] J. Liang, V. Makoviychuk, A. Handa, N. Chentanez, M. Macklin, and D. Fox, “Gpu-accelerated robotic simulation for distributed reinforcement learning,” in *Conference on Robot Learning*. PMLR, 2018, pp. 270–282.
- [14] Y. Hasson, G. Varol, D. Tzionas, I. Kalevatykh, M. J. Black, I. Laptev, and C. Schmid, “Learning joint reconstruction of hands and manipulated objects,” in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.
- [15] S. Hampali, M. Rad, M. Oberweger, and V. Lepetit, “Honnoteate: A method for 3d annotation of hand and object poses,” in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.
- [16] 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.*, “Dexycb: 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.
- [17] S. Brahmbhatt, C. Ham, C. C. Kemp, and J. Hays, “Contactdb: Analyzing and predicting grasp contact via thermal imaging,” in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.
- [18] S. Brahmbhatt, C. Tang, C. D. Twigg, C. C. Kemp, and J. Hays, “Contactpose: A dataset of grasps with object contact and hand pose,” in *European Conference on Computer Vision (ECCV)*, 2020.
- [19] 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 (RA-L)*, 2021.
- [20] J. Romero, D. Tzionas, and M. J. Black, “Embodied hands: Modeling and capturing hands and bodies together,” *ACM Transactions on Graphics (TOG)*, 2017.
- [21] S. Brahmbhatt, A. Handa, J. Hays, and D. Fox, “Contactgrasp: Functional multi-finger grasp synthesis from contact,” in *International Conference on Intelligent Robots and Systems (IROS)*, 2019.
- [22] P. Mandikal and K. Grauman, “Learning dexterous grasping with object-centric visual affordances,” in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 6169–6176.
- [23] Y. Qin, H. Su, and X. Wang, “From one hand to multiple hands: Imitation learning for dexterous manipulation from single-camera teleoperation,” *arXiv preprint arXiv:2204.12490*, 2022.
- [24] J. Ye, J. Wang, B. Huang, Y. Qin, and X. Wang, “Learning continuous grasping function with a dexterous hand from human demonstrations,” *arXiv preprint arXiv:2207.05053*, 2022.
- [25] A. Rodriguez, M. T. Mason, and S. Ferry, “From caging to grasping,” *International Journal of Robotics Research (IJRR)*, vol. 31, no. 7, pp. 886–900, 2012.
- [26] D. Prattichizzo, M. Malvezzi, M. Gabiccini, and A. Bicchi, “On the manipulability ellipsoids of underactuated robotic hands with compliance,” *Robotics and Autonomous Systems*, vol. 60, no. 3, pp. 337–346, 2012.
- [27] C. Rosales, R. Suárez, M. Gabiccini, and A. Bicchi, “On the synthesis of feasible and prehensile robotic grasps,” in *International Conference on Robotics and Automation (ICRA)*, 2012.
- [28] R. M. Murray, *A mathematical introduction to robotic manipulation*. CRC press, 2017.
- [29] J. Ponce, S. Sullivan, J.-D. Boissonnat, and J.-P. Merlet, “On characterizing and computing three-and four-finger force-closure grasps of polyhedral objects,” in *International Conference on Robotics and Automation (ICRA)*, 1993.
- [30] J. Ponce, S. Sullivan, A. Sudsang, J.-D. Boissonnat, and J.-P. Merlet, “On computing four-finger equilibrium and force-closure grasps of polyhedral objects,” *International Journal of Robotics Research (IJRR)*, vol. 16, no. 1, pp. 11–35, 1997.
- [31] J.-W. Li, H. Liu, and H.-G. Cai, “On computing three-finger force-closure grasps of 2-d and 3-d objects,” *IEEE Transactions on Robotics and Automation*, vol. 19, no. 1, pp. 155–161, 2003.
- [32] Y. Zheng and C.-M. Chew, “Distance between a point and a convex cone in  $n$ -dimensional space: Computation and applications,” *Transactions on Robotics (T-RO)*, vol. 25, no. 6, pp. 1397–1412, 2009.
- [33] H. Dai, A. Majumdar, and R. Tedrake, “Synthesis and optimization of force closure grasps via sequential semidefinite programming,” in *Robotics Research*. Springer, 2018, pp. 285–305.
- [34] Y.-H. Liu, “Qualitative test and force optimization of 3-d frictional form-closure grasps using linear programming,” *IEEE Transactions on Robotics and Automation*, vol. 15, no. 1, pp. 163–173, 1999.
- [35] H. Jiang, S. Liu, J. Wang, and X. Wang, “Hand-object contact consistency reasoning for human grasps generation,” in *International Conference on Computer Vision (ICCV)*, 2021.
- [36] E. Corona, A. Pumarola, G. Alenya, F. Moreno-Noguer, and G. Rogez, “Ganhand: Predicting human grasp affordances in multi-object scenes,” in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.
- [37] 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 *International Conference on Robotics and Automation (ICRA)*, 2021.
- [38] J. Lundell, F. Verdoja, and V. Kyrki, “Ddgc: Generative deep dexterous grasping in clutter,” *IEEE Robotics and Automation Letters (RA-L)*, 2021.
- [39] L. Yang, X. Zhan, K. Li, W. Xu, J. Li, and C. Lu, “Cpf: Learning a contact potential field to model the hand-object interaction,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 11 097–11 106.
- [40] 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 (RA-L)*, 2020.
- [41] A. Wu, M. Guo, and C. K. Liu, “Learning diverse and physically feasible dexterous grasps with generative model and bilevel optimization,” *arXiv preprint arXiv:2207.00195*, 2022.
- [42] K. Li, N. Baron, X. Zhang, and N. Rojas, “Efficientgrasp: A unified data-efficient learning to grasp method for multi-fingered robot hands,” *IEEE Robotics and Automation Letters (RA-L)*, vol. 7, no. 4, pp. 8619–8626, 2022.
- [43] J. Varley, J. Weisz, J. Weiss, and P. Allen, “Generating multi-fingered robotic grasps via deep learning,” in *2015 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 2015, pp. 4415–4420.

- [44] 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,” 2022.
- [45] T. Zhu, R. Wu, X. Lin, and Y. Sun, “Toward human-like grasp: Dexterous grasping via semantic representation of object-hand,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 15 741–15 751.
- [46] K. Karunaratnakul, J. Yang, Y. Zhang, M. Black, S. Tang, and K. Muandet, “Grasping field: Learning implicit representations for human grasps,” in *International Conference on 3D Vision (3DV)*, 2020.
- [47] 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.
- [48] S. Hampali, S. D. Sarkar, M. Rad, and V. Lepetit, “Keypoint transformer: Solving joint identification in challenging hands and object interactions for accurate 3d pose estimation,” in *CVPR*, 2022.
- [49] O. Taheri, N. Ghorbani, M. J. Black, and D. Tzionas, “Grab: A dataset of whole-body human grasping of objects,” in *European Conference on Computer Vision (ECCV)*, 2020.
- [50] O. Taheri, V. Choutas, M. J. Black, and D. Tzionas, “Goal: Generating 4d whole-body motion for hand-object grasping,” *arXiv preprint arXiv:2112.11454*, 2021.
- [51] Z. Fan, O. Taheri, D. Tzionas, M. Kocabas, M. Kauffmann, M. J. Black, and O. Hilliges, “Articulated objects in free-form hand interaction,” *arXiv preprint arXiv:2204.13662*, 2022.
- [52] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, J. Xiao, L. Yi, and F. Yu, “ShapeNet: An Information-Rich 3D Model Repository,” Stanford University — Princeton University — Toyota Technological Institute at Chicago, Tech. Rep. *arXiv:1512.03012 [cs.GR]*, 2015.
- [53] M. Savva, A. X. Chang, and P. Hanrahan, “Semantically-Enriched 3D Models for Common-sense Knowledge,” *CVPR 2015 Workshop on Functionality, Physics, Intentionality and Causality*, 2015.
- [54] B. Calli, A. Singh, J. Bruce, A. Walsman, K. Konolige, S. Srinivas, P. Abbeel, and A. M. Dollar, “Yale-cmu-berkeley dataset for robotic manipulation research,” *The International Journal of Robotics Research*, vol. 36, no. 3, pp. 261–268, 2017.
- [55] A. Singh, J. Sha, K. S. Narayan, T. Achim, and P. Abbeel, “Bigbird: A large-scale 3d database of object instances,” in *2014 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2014, pp. 509–516.
- [56] D. Kappler, J. Bohg, and S. Schaal, “Leveraging big data for grasp planning,” in *2015 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2015, pp. 4304–4311.
- [57] 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*, vol. 31, no. 8, pp. 927–934, 2012.
- [58] 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,” *arXiv preprint arXiv:2204.11918*, 2022.
- [59] J. Huang, Y. Zhou, and L. Guibas, “Manifoldplus: A robust and scalable watertight manifold surface generation method for triangle soups,” *arXiv preprint arXiv:2005.11621*, 2020.
- [60] X. Wei, M. Liu, Z. Ling, and H. Su, “Approximate convex decomposition for 3d meshes with collision-aware concavity and tree search,” *arXiv preprint arXiv:2205.02961*, 2022.
- [61] K. M. Jatavallabhula, E. Smith, J.-F. Lafleche, C. F. Tsang, A. Rozantsev, W. Chen, T. Xiang, R. Lebaredian, and S. Fidler, “Kaolin: A pytorch library for accelerating 3d deep learning research,” *arXiv preprint arXiv:1911.05063*, 2019.
- [62] J. J. Park, P. Florence, J. Straub, R. Newcombe, and S. Lovegrove, “Deepsdf: Learning continuous signed distance functions for shape representation,” in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.
- [63] C. Ferrari and J. F. Canny, “Planning optimal grasps,” in *ICRA*, vol. 3, no. 4, 1992, p. 6.
- [64] K. Sohn, H. Lee, and X. Yan, “Learning structured output representation using deep conditional generative models,” *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 28, 2015.
- [65] T. Feix, J. Romero, H.-B. Schmidtmayer, A. M. Dollar, and D. Kragic, “The grasp taxonomy of human grasp types,” *IEEE Transactions on Human-Machine Systems*, vol. 46, no. 1, pp. 66–77, 2015.