DVGG: Deep Variational Grasp Generation for Dextrous Manipulation

Wei Wei, *Student Member, IEEE*, Daheng Li, Peng Wang, *Member, IEEE*,
Yiming Li, *Student Member, IEEE*, Wanyi Li, Yongkang Luo, and Jun Zhong

***Abstract*—Grasping with anthropomorphic robotic hands involves much more hand-object interactions compared to parallel-jaw grippers. Modeling hand-object interactions is essential to the study of multi-finger hand dextrous manipulation. This work presents DVGG, an efficient grasp generation network that takes single-view observation as input and predicts high-quality grasp configurations for unknown objects. In general, our generative model consists of three components: 1) Point cloud completion for the target object based on the partial observation; 2) Diverse sets of grasps generation given the complete point cloud; 3) Iterative grasp pose refinement for physically plausible grasp optimization. To train our model, we build a large-scale grasping dataset that contains about 300 common object models with 1.5 M annotated grasps in simulation. Experiments in simulation show that our model can predict robust grasp poses with a wide variety and high success rate. Real robot platform experiments demonstrate that the model trained on our dataset performs well in the real world. Remarkably, our method achieves a grasp success rate of 70.7% for novel objects in the real robot platform, which is a significant improvement over the baseline methods.**

***Index Terms*—Deep learning in grasping and manipulation, multifingered hands, computer vision for automation, point cloud completion, iterative refinement.**

I. Introduction

GRASPING with parallel-jaw grippers has been well investigated and widely applied in robotic manipulation. However, the study of anthropomorphic robotic hand remains a challenge for the robotics community. Multi-finger grippers equipped with multiple actuated joints enable robots to perform more advanced operations such as grasping for tool-use.

Traditional analysis-based methods for multi-finger grasp generation rely on the assumption that object shapes and poses are known a priori [1]–[3] that they can synthesize grasp configurations with commonly used grasping metrics, *e.g.* force-closure [4] and ϵ-metric [5]. However, these approaches are not applicable to unseen objects. To generalize to unknown objects, most current works utilize shape completion module [6], [7], while these methods are time-consuming due to the huge search space for high-DOF grippers.

In recent years, learning-based approaches have made significant breakthroughs. Most learning-based grasping methods estimate grasp poses directly from raw sensor inputs [8]–[11]. Nevertheless, most of these approaches focus on studying parallel-jaw grippers with the majority of degrees of freedom concentrated in the wrist joint.

Although it is challenging to predict high-DOF grasp poses, researchers have proposed various solutions to this problem. [12] uses a neural network to predict pixel-wise heatmap for multi-finger placement but relies on a grasp planner to determine the final grasp pose. [13], [14] train evaluation models with grasps generated by a grasp sampler, which involves manually-engineered mappings from observation to grasp. [15], [16] propose to identify a one-to-one mapping from objects to grasp poses, which results in limited grasp postures. [17], [18] learn a grasp-success probability prediction model with a voxel-based 3D convolutional neural network, while the hand needs to approach the object with limited directions. [19] proposes a "GenerAL" framework with reinforcement learning for multi-fingered grasping in clutter, while encountering the sim-real gap problem.

Meanwhile, a large body of works concentrated on human grasp estimation [20]–[23] achieve promising results on daily objects [24]. These works focus on hand-object interactions based on contacts, which is intuitive for human grasping.

Inspired by this intuitive idea, we propose to generate diverse sets of robotics grasps based on contacts. As shown in Fig. 1, our method works in the following way: 1) Complete points are first reconstructed by the point completion module; 2) The variational grasp generator estimates diverse sets of coarse grasps given the complete object points; 3) Coarse grasps are further refined by the iterative refinement module.

We evaluate our approach on the YCB [24] and EGAD! [25] dataset. Experimental results show that our model can generate diverse sets of grasps in terms of approach direction and posture, and the refinement module helps to produce physically plausible grasps. Furthermore, our method shows significant improvement

Manuscript received September 9, 2021; accepted December 23, 2021. Date of publication January 5, 2022; date of current version January 18, 2022. This letter was recommended for publication by Associate Editor Gerhard Neumann and Editor Markus Vincze upon evaluation of the reviewers' comments. This work was supported in part by the National Natural Science Foundation of China under Grants 91748131, 62006229, and 61771471, and in part by the Strategic Priority Research Program of Chinese Academy of Science under Grant XDB32050106. *(Corresponding author: Peng Wang.)*

Wei Wei, Daheng Li, and Yiming Li are with the Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China, and also with the School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing 100049, China (e-mail: wei.wei2018@ia.ac.cn; lidaheng2020@ia.ac.cn; liyiming2019@ia.ac.cn).

Wanyi Li, Yongkang Luo, and Jun Zhong are with the Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China (e-mail: wanyi.li@ia.ac.cn; yongkang.luo@ia.ac.cn; jun.zhong@ia.ac.cn).

Peng Wang is with the Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China, with the CAS Center for Excellence in Brain Science and Intelligence Technology, Chinese Academy of Sciences, Shanghai 200031, China, and also with the Centre for Artificial Intelligence and Robotics, Hong Kong Institute of Science and Innovation, Chinese Academy of Sciences, Hong Kong 999077, Hong Kong (e-mail: peng_wang@ia.ac.cn).

Digital Object Identifier 10.1109/LRA.2022.3140424

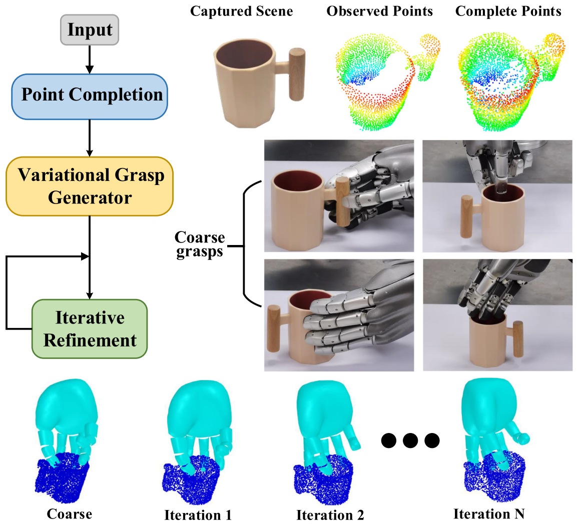

Fig. 1. Grasping a target object with dextrous grasp configurations. Our approach is able to generate diverse sets of grasps for unknown objects efficiently in a coarse to fine manner.

compared to baselines in terms of time efficiency and grasp success rate.

In summary, our primary contributions are: 1) A novel generative model for generating diverse sets of high-DOF hand grasps based on hand-object interactions; 2) A useful iterative refinement module for hand grasp refinement; 3) Significant improvement on grasp generation for both seen and unseen objects compared to the baseline methods.

II. Related Work

Grasp Planners: Traditional grasp planners [1], [2], [4], [26] assumed that the environment is fully perceptible. Various algorithms [2]–[4] in this direction have been proposed to search for high-quality grasps based on certain grasp metrics [5], [27]. However, these methods are challenging for a number of factors: 1) Environments are not fully perceptible, especially for unstructured environments; 2) Time-consuming due to the large searching space for high-DOF grippers. Recently, learning-based grasp planners propose to predict grasps directly given the partial observation. Most works following this way [8], [9], [28], [29] focus on studying parallel-jaw grippers with different kinds of inputs, *e.g.* RGB-D images or point clouds. [9], [29] propose to conduct grasp pose detection as rectangle detection in 2D space, the grippers are limited to approaching the target object vertically. DexNet [28] collects numerous object models for GQ-CNN training and achieves state-of-the-art performance. [8], [30] propose to estimate 6 or 7 DOF grasp poses in cluttered scenes. However, for high-DOF anthropomorphic hands, in addition to the wrist pose, remaining hand joints play a more important role in human-like manipulation such as in-hand manipulation.

Learning for hand-object interactions: The research of hand-object interactions has been widely studied in the computer vision community [20]–[22]. Recently, several datasets are proposed for facilitating hand-object interactions research. [21], [31], [32] label human hand grasps with captured images or videos. [22], [33] propose to synthesize human hand grasps with GraspIt! [1], while the grasps may not look realistic in general. Most of these methods propose to predict affordance map for target objects, and predict grasps based on contacts and penetration jointly. However, only a few researchers pay attention to hand-object contacts in robotics grasping [34]–[37]. To our knowledge, [36] is the most similar work to ours, which refines sampled grasps with human-demonstrated contact map for functional grasping. In this work, we propose to generate grasps with implicitly hand-object contacts based on a generative model.

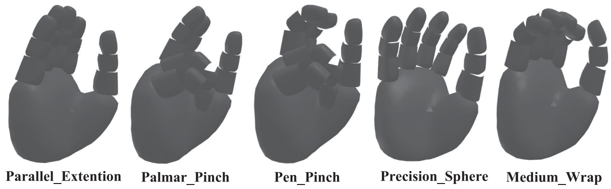

Fig. 2. Five taxonomy for annotating grasps in simulation.

III. Problem Statement

This work concentrates on planning dextrous high-DOF robotic hand grasping based on single view observation, which implies generating physically plausible and collisionless grasp configurations. More formally, our model $\mathcal{M}$ takes the observed point cloud $\mathcal{P}$ as input and predicts high quality grasps. Each grasp $\boldsymbol{g}$ is represented by a hand wrist pose $\mathbf{p}$ and a hand joint configuration $\boldsymbol{\theta}$, $i.e.$ $\boldsymbol{g} = \{\mathbf{p}, \boldsymbol{\theta}\}$. Hand wrist pose $\mathbf{p}$ is given in $\mathcal{S}E(3)$, including the translation $\mathbf{t} = [t_x, t_y, t_z]$ and orientation quaternion $\mathbf{q} = [q_w, q_x, q_y, q_z]$. Hand joint configuration $\boldsymbol{\theta}$ is denoted by the actual degree of freedom of the hand, $\boldsymbol{\theta} \in \mathbb{R}^{20}$ for HIT-DLR II Hand.

IV. Dataset Generation

In this section, we give a brief introduction for the procedure of building our grasp dataset in the physics simulator.

Objects: We collect 300 objects from the dataset released in the previous work [16]. Some of the objects are re-scaled to fit for the HIT-DLR II Hand. All objects share the same coefficient of friction (0.25) and density (1500 kg/m^3).

Grasp Annotation: We adopt the commonly used approach direction sampling scheme. It first samples a point on the surface, then the hand approaches the object and executes a grasp attempt with a predefined step interval. For each step interval, uniform in-plane rotations are sampled. Five taxonomy shown in Fig. 2 are selected for annotation [38].

Grasp Labelling in Simulation: We build our synthetic grasping dataset in the physics simulator MuJoCo [39]. The physics simulation consists of three steps: 1) Objects are fixed stationary at first and the hand is initialized with a pre-grasp state, then the hand approaches the object and executes a grasp attempt with a pre-defined grasp taxonomy until the simulator reaches a stable state. At this time, all fingers should contact the object or reach their maximum joint angles. 2) Then all fingers keep the grasping force while the gravity is present till the simulator reaches a stable state or the object falls from the hand. 3) Unstable grasps are filtered by shaking the hand and keep those grasps that consistently keep the object in hand.

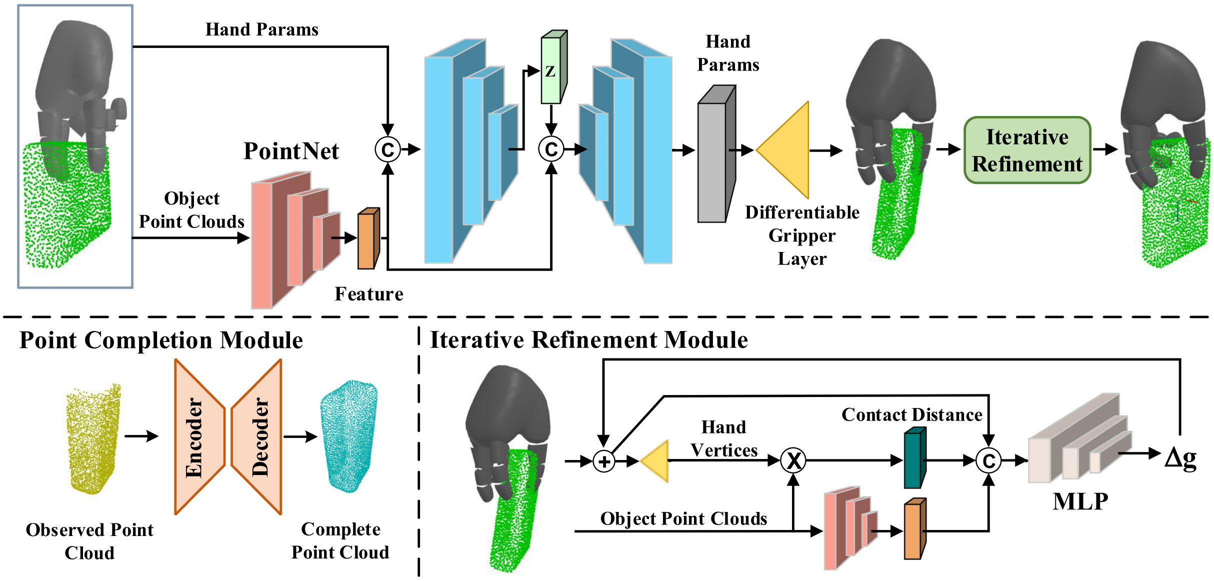

Fig. 3. Overview architecture of **DVGG** network for dextrous grasp pose generation and refinement. Point Completion Module for generating complete point clouds from partial object point clouds, shown in the left bottom. Iterative Refinement Module for grasp pose refinement, shown in the right bottom.

V. Method

In this section, we present the deep variational grasp generation network (**DVGG**) for generating dextrous grasp poses. The overall pipeline is shown in Fig. 3. It consists of 3 submodules: Object Point Completion, Variational Grasp Generator, and Iterative Grasp Refinement.

A. *Object Point Completion*

Manipulation of anthropomorphic hands involves rich hand-object contacts. Most of the existing methods take the assumption that the target object model is known [20], [21], [36]. Some of the current works propose to utilize multi-view inputs for either object reconstruction [15] or feature aggregation [16], [23], while multi-camera platform setup imposes strict restrictions for real-world applications.

To this end, we propose to directly estimate the complete object model from single-view input inspired by [34]. In practice, we follow the method proposed by [40] and make two modifications during data synthesis to make it easy for real platforms application: 1) Position of partial observed point clouds are represented in camera coordinate system instead of the object coordinate system; 2) Origin is normalized to the center of the partial observed point clouds instead of the centroid of the objects. We synthesize about 1 M partial observed point clouds, and train the model from scratch. All objects come from [16]. Experimental results shown in Fig. 4 demonstrate that the model trained on our dataset performs well on real robotic platform.

B. *Variational Grasp Generator*

The Variational Grasp Generator is based on Conditional Variational Auto-Encoder (CVAE) [41] generative network. For training our grasp generator, we use ground-truth complete object point clouds as conditional information. In the training stage, both the encoder and decoder are used to learn the grasp generation task by optimizing the reconstruction errors with hand-object interactions constraint; At testing time, only the decoder is used. The network architecture is shown in Fig. 3. Both the encoder and decoder of the generator are composed of Multi-Layer Perceptrons (MLP).

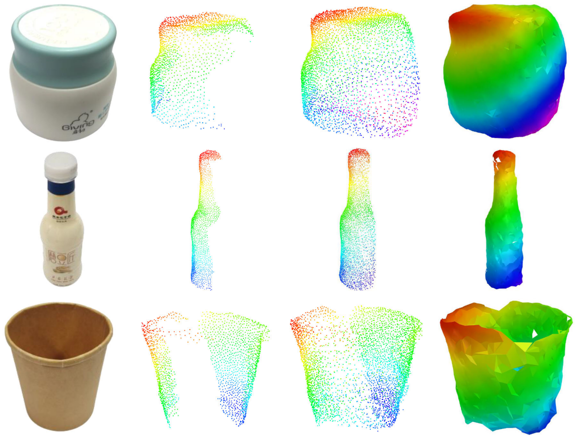

Fig. 4. Qualitative results of Point Completion Module on unseen objects. The first column shows the real-world object, the second and the third show the partial observed and predicted complete point clouds, the last column shows the reconstructed surface mesh.

During training stage, given the grasp configuration $\boldsymbol{g} \in \mathbb{R}^{27}$ and the object point cloud $\mathcal{P}^o \in \mathbb{R}^{N\times 3}$ as input, we utilize PointNet [42] to extract point cloud feature $\mathcal{F}^o$ and MLPs for hand configuration $\boldsymbol{g}$ feature extraction $\mathcal{F}^h$. These two features are then concatenated together as the input for the encoder. The encoder learn to map each pair of point cloud $\mathcal{P}^o$ and grasp $\boldsymbol{g}$ to a subspace in the latent space $\boldsymbol{z}$, where $P(\boldsymbol{z}) \sim \mathcal{N}(0, I)$.

Given the sampled latent code $\boldsymbol{z}$ and the extracted feature $\mathcal{F}^o$ as input to the decoder, the decoder learns to predict the hand parameters $\boldsymbol{g}$. Given $\boldsymbol{g}$ as input, the hand mesh is reconstructed by a differentiable hand layer. This layer is designed based on the forward kinematics of the hand. It takes the hand parameters as input and outputs the mesh $\boldsymbol{\mathcal{H}} = (\boldsymbol{V}, \boldsymbol{F})$, where $\boldsymbol{V}$ and $\boldsymbol{F}$ denote vertices and faces.

During testing stage, the encoder is removed, and a latent value $\boldsymbol{z}$ is randomly sampled from Normal Gaussian Distribution $\mathcal{N}(0, I)$. Then the latent code $\boldsymbol{z}$ and the extracted object feature $\mathcal{F}^o$ are concatenated together as input to the decoder. The decoder predicts hand parameters for the differentiable hand layer to reconstruct the hand model.

Given the above introduction for network architecture and workflow of the grasp generator, we then present the detailed objective function for training our network. If not specified, a hat on top refers to predicted variables.

KL-Divergence: We use a KL-divergence loss to regularize the distribution of latent code $\boldsymbol{z}$, enforcing the latent code distribution $Q(\boldsymbol{z}|\mathcal{P}^o, \boldsymbol{g})$ to be close to a standard Gaussian Distribution. This can be done by maximizing the KL-Divergence as follows:

$$\mathcal{L}_{\mathcal{KL}} = -\boldsymbol{KL}\left(Q(\boldsymbol{z}|\mathcal{P}^o, \boldsymbol{g})||\mathcal{N}(0, I)\right) \tag{1}$$

Reconstruction: We design the reconstruction objective based on the reconstructed hand mesh. It consists of the following terms: hand mesh vertices displacement and joint angles $\boldsymbol{\theta}$ error. We adopt the $\boldsymbol{L}_2$ loss for optimizing Euclidean distance of the vertices and the $\boldsymbol{L}_1$ loss for the joint angles. The losses are formulated as follows:

$$\begin{aligned}\mathcal{L}_{\mathcal{V}} &= \frac{1}{N}\sum_{i}^{N}||\hat{\boldsymbol{V}}_i - \boldsymbol{V}_i||_2^2\\ \mathcal{L}_{\boldsymbol{\theta}} &= |\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}|\\ \mathcal{L}_{\mathcal{R}} &= \lambda_{\mathcal{V}} \cdot \mathcal{L}_{\mathcal{V}} + \lambda_{\theta} \cdot \mathcal{L}_{\boldsymbol{\theta}}\end{aligned} \tag{2}$$

The reconstruction loss $\mathcal{L}_{\mathcal{R}}$ include two terms, $\mathcal{L}_{\mathcal{V}}$ for hand mesh vertices reconstruction and $\mathcal{L}_{\boldsymbol{\theta}}$ for hand joint angles prediction. Where N is the number of hand mesh vertices, $\boldsymbol{V}$ denotes the hand mesh vertices, $\boldsymbol{\theta}$ denote the hand joint angles. $\lambda_{\mathcal{V}}$ and λ_{θ} are constants for balancing the losses.

Hand-Object Contact: A physically plausible grasp should hold the object tightly. Intuitively, we propose to model reasonable grasp based on contacts in two folds: *Which part of the object that the fingers should be in contact with?* And *Which part of finger should grasp?* Specifically, given the ground-truth hand-object grasp, both object affordance points $\mathcal{O}^c$ and hand grasp vertices $\mathcal{H}^c$ can be derived from the distance between object points and hand vertices, as shown in Fig. 5(a). We denote the object point subset that is close enough to the hand vertices as $\mathcal{O}^c$, and the contact points on hand that is close enough to the object points as $\mathcal{H}^c$. We formulate the contact losses as follows:

$$\begin{aligned}\mathcal{L}_{\mathcal{O}} &= \frac{1}{|\mathcal{O}^c|}\sum_{\boldsymbol{p}\in\mathcal{O}^c}\left(\boldsymbol{f}(\boldsymbol{p}|\hat{\boldsymbol{V}}) - \boldsymbol{f}(\boldsymbol{p}|\boldsymbol{V})\right)\\ \mathcal{L}_{\mathcal{H}} &= \frac{1}{|\mathcal{H}^c|}\sum_{\boldsymbol{v}\in\mathcal{H}^c}\left(\boldsymbol{f}(\hat{\boldsymbol{v}}|\mathcal{P}^o) - \boldsymbol{f}(\boldsymbol{v}|\mathcal{P}^o)\right)\\ \mathcal{L}_{\mathcal{C}} &= \lambda_{\mathcal{O}} \cdot \mathcal{L}_{\mathcal{O}} + \lambda_{\mathcal{H}} \cdot \mathcal{L}_{\mathcal{H}}\end{aligned} \tag{3}$$

Where $f(\cdot|\cdot)$ refers to Signed Distance Field (SDF), the magnitude of a point represents the distance to the surface boundary and the sign indicates whether the region is inside (-) or outside (+), $e.g.$ $\boldsymbol{f}(\boldsymbol{p}|\boldsymbol{V})$ outputs the signed distance for point $\boldsymbol{p}$ to hand vertices set $\boldsymbol{V}$. The hand-object contact loss $\mathcal{L}_{\mathcal{C}}$ includes two terms, $\mathcal{L}_{\mathcal{O}}$ for object affordance loss and $\mathcal{L}_{\mathcal{H}}$ hand contact loss. The loss $\mathcal{L}_{\mathcal{O}}$ encourages the object affordance map to be consistent with the ground-truth affordance map, and the loss $\mathcal{L}_{\mathcal{H}}$ penalizes the difference between the predicted hand contact region and the ground-truth. $\lambda_{\mathcal{O}}$ and $\lambda_{\mathcal{H}}$ are constants for balance.

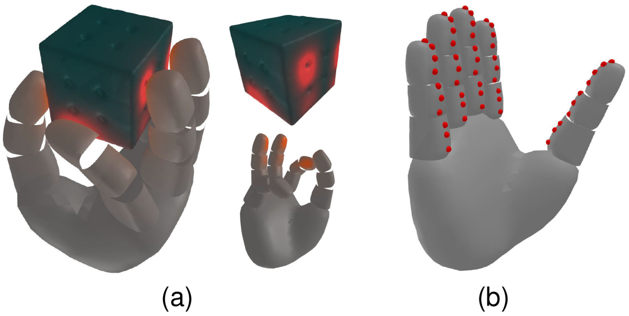

Fig. 5. (a) An example of the hand-object contact map, the red part shows the affordance map of the object, the orange part shows the contact region on the hand. (b) Simplified mesh of the HIT-DLR II Hand model and the sampled potential grasp points (red).

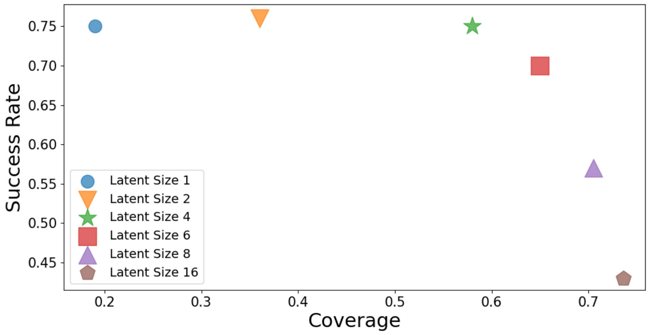

Fig. 6. Scatter chart shown the influence of number of dimensions.

Interpenetration: In order to generate realistic hand grasp, we need to take consideration of physical constraints, $i.e.$ interpenetration between hand gripper and the target object. To alleviate hand-object intersection, we formulate interpenetration loss as follows:

$$\mathcal{L}_{\mathcal{P}} = \sum_{\boldsymbol{p}\in\mathcal{P}^o}\min\left(-\langle\boldsymbol{1}, f(\boldsymbol{p}|\boldsymbol{V})\rangle, 0\right) \tag{4}$$

Where $\boldsymbol{1}$ is a 2D one-vector, and $\langle\cdot,\cdot\rangle$ denotes a dot product. The interpenetration loss $\mathcal{L}_{\mathcal{P}}$ actually penalize the negative sum of signed distances of the object point to the hand mesh.

Finally, the overall loss are summarized as follows:

$$\mathcal{L}_{grasp} = \lambda_{\mathcal{KL}} \cdot \mathcal{L}_{\mathcal{KL}} + \mathcal{L}_{\mathcal{R}} + \mathcal{L}_{\mathcal{C}} + \lambda_{\mathcal{P}} \cdot \mathcal{L}_{\mathcal{P}} \tag{5}$$

where $\lambda_{\mathcal{KL}}$ and $\lambda_{\mathcal{P}}$ are constants to balance the losses.

C. Iterative Grasp Refinement

Although our model can predict realistic grasps for most of the objects in our dataset, two failure cases remains, as shown in Fig. 8: 1) Penetration with thin-walled objects, such as bowls and mugs; 2) Hand-object contact is not tight enough, causing grasp failures. Considering the above drawbacks when operating on the real robotic platform, we further refine the grasp quality

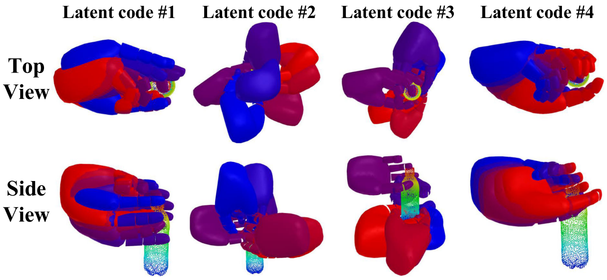

Fig. 7. Qualitative results for the latent codes.

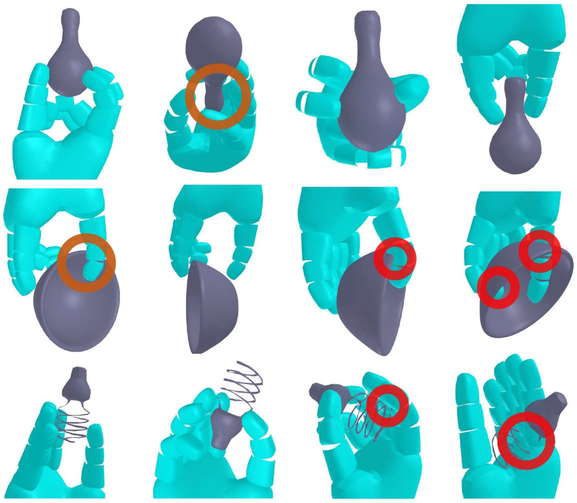

Fig. 8. Qualitative results of the variational grasp generator. The red circles mark the parts with collisions and the brown circles mark the parts with no tight contacts.

to avoid collisions. To achieve this, we propose to improve the grasp quality by penalizing hand-object penetration and optimizing the grasp contact energy in our refinement module. Specifically, given the predicted grasp pose of the grasp sampler and complete point cloud, the grasp refinement module takes as input the predicted grasp g and hand-object contact distance, then predicts the residual grasp Δg transformation. In this way, the refinement module can work in an iterative manner:

$$\begin{aligned}
\mathcal{L}_{CE} &= \min \sum_{\boldsymbol{p} \in \mathcal{H}^g} \mathbb{I}(f(\boldsymbol{p}|\boldsymbol{V})) \\
\mathbb{I}\left(f(\cdot|\cdot)\right) &= \begin{cases} f(\cdot|\cdot) & \text{if } f(\cdot|\cdot) < \mathcal{T} \\ T, & \text{otherwise.} \end{cases} \\
\mathcal{L}_{\mathcal{D}} &= \mathcal{D}(g, g^*) \\
\mathcal{L}_{refine} &= \lambda_{CE} \cdot \mathcal{L}_{CE} + \lambda_{\mathcal{P}} \cdot \mathcal{L}_{\mathcal{P}} + \lambda_{\mathcal{D}} \cdot \mathcal{L}_{\mathcal{D}}
\end{aligned} \tag{6}$$

The refinement loss $\mathcal{L}_{refine}$ includes three terms: $\mathcal{L}_{CE}$ for optimizing the grasp distance contact energy by attracting potential grasp points $p \in \mathcal{H}^g$ to be close to the target object, which helps to produce wrap-around grasps. $\mathcal{L}_{\mathcal{P}}$ for penalizing hand-object interpenetration and $\mathcal{L}_{\mathcal{D}}$ regularize the refined grasp g^* should be close to the input grasp g. $\mathbb{I}$ denotes the Indicator function for judging whether the hand grasp points are close enough, $\mathcal{T}$ is the distance threshold, we set it to 2 centimeters. We define $D(g, g^*)$ as the distance measurement function, where g is the predicted hand grasp of the sampler, g^* is the output grasp configuration of the refinement module. We manually label $N = 50$ potential grasp contact points $\mathcal{H}^g$ on the gripper surface, as shown in Fig. 5(b).

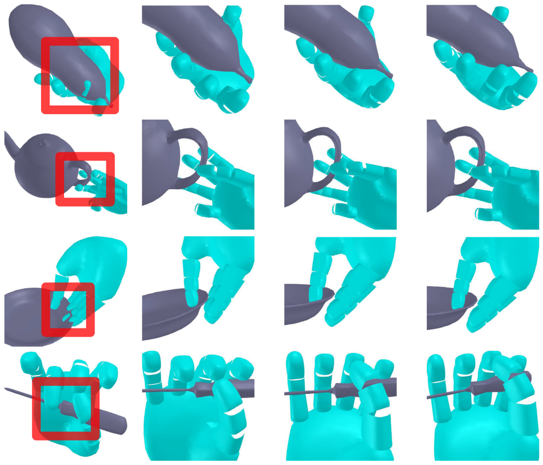

Fig. 9. Cases of refinement.The first column shows coarse grasps, the right columns show grasps after 1, 2 and 3 times refinement.

TABLE I
HYPER-PARAMETERS SETTING

parameters	$\lambda_{\mathcal{KL}}$	λ_{θ}	λ_{v}	$\lambda_{\mathcal{O}}$	$\lambda_{\mathcal{H}}$	$\lambda_{\mathcal{P}}$	$\lambda_{\mathcal{D}}$	λ_{CE}
value	0.1	0.5	30	30	30	1	1	20

TABLE II
EFFECT OF ITERATIVE REFINEMENT. ↑: HIGHER THE BETTER; ↓: LOWER THE BETTER

	Penetration		Success Rate ↑
	Depth (cm) ↓	Volume (cm^3) ↓	
DVGG (w/o refine)	0.53	5.15	64.6%
DVGG (+1 refine)	0.46	4.26	72.7%
DVGG (+2 refine)	**0.41**	3.87	74.9%
DVGG (+3 refine)	**0.41**	**3.80**	**75.1%**

TABLE III
ABLATION STUDY ON VARIOUS LOSS FUNCTIONS

Loss removed		none	$L_{\mathcal{C}}$	$L_{\mathcal{P}}$	L_{CE}	$L_{\mathcal{D}}$
Success Rate (%) ↑		72.7	40.5	10.2	70.0	65.4
Penetration	depth (cm) ↓	0.46	0.15	3.62	0.42	0.52
	Volume (cm^3) ↓	4.26	0.92	80.25	4.06	5.60

Qualitative results shown in Fig. 9 show our refinement algorithm can deal with inaccurate grasp postures and refine them to the pose with fewer collisions and higher quality.

VI. EXPERIMENTS

We evaluate our model both in simulation and on a real robot platform consisting of a UR5 robotic arm equipped with a HIT-DLR II Hand.

A. Implementation Details

We sample 2048 surface points for each object during training the variational grasp generator, and these points are sampled using the Farthest Point Sampling (FPS) method. During the

TABLE IV
COMPARISON WITH GRASPIT! IN SIMULATION. PC DENOTES PREDICTED COMPLETE OBJECT POINTS AS INPUT, PO FOR PARTIAL OBSERVED POINTS, GT FOR GROUND-TRUTH OBJECT POINTS. * DENOTES ADJUSTMENT WITH ITERATIVE REFINEMENT

		YCB						EGAD!			
		GraspIt!			DVGG			GraspIt!		DVGG	
Input		PC	GT	GT*	PO	PC*	GT*	PC	GT	PC*	GT*
ϵ-metric ↑		0.028	0.029	0.034	0.01	0.071	0.063	0.043	0.044	0.075	0.081
Penetration	depth (cm) ↓	0.44	0.25	0.028	1.95	0.65	0.41	0.31	0.20	0.053	0.37
	volume (cm^3) ↓	2.68	0.75	0.87	25.3	4.18	3.84	1.21	0.52	3.90	3.23
Sampling Time (sec.) ↓		36.6	37.8	37.8	0.16	0.16	0.16	37.6	37.5	0.16	0.16
Success Rate (%) ↑		51.3	50.1	57.2	27.0	72.4	74.9	72.3	71.0	75.2	82.2

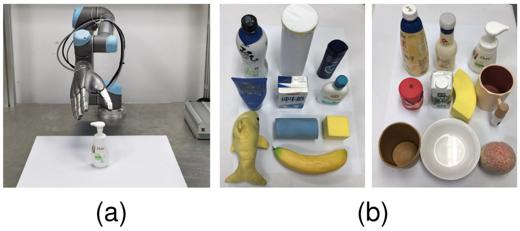

Fig. 10. Real setting of our robotic grasping experiments. (a) Robotic grasping experiment setup with HIT-DLR II Hand and UR5 robotic arm. (b) Objects used in our robotic experiments. Left part shows similar objects, right part shows novel objects.

TABLE V
RESULTS OF ROBOTIC PLATFORM EXPERIMENTS

	Similar objects SR (Avg)↑	Novel objects SR (Avg)↑
GraspIt! [1]	48.7%	46.0%
PointNetGPD [43]	52.3%	48.3%
DVGG (w/o refine)	67.3%	57.3%
DVGG (+3 refine)	**74.7%**	**70.7%**

inference stage, 2048 points are sampled among the complete points predicted by the point completion module. The variational grasp generator and refinement network are trained for 250 and 100 epochs respectively with the learning rate set to 0.002 at start and divided by 10 when the validation error plateaus. Batch size is 512. We train our model on a RTX-3090 GPU. The dimension of the latent space is set to 4 for real robot platform evaluation. Other hyper-parameters for training our network are listed in Table I

B. Evaluate Metric

We use three quantitative metrics to evaluate our approach consistently with previous literature [11], [33].

Penetration for measurement of the penetration between the hand mesh and the target object. It consists of two terms: penetration **depth** and **volume**. We follow the implementation used in [20]. If the hand and the object collide, the penetration depth is the maximum of the distances from hand mesh vertices to the object surface.

Success Rate (SR) is used to measure the stability and quality of the generated grasps, which is commonly used in grasping tasks.

Coverage Rate is utilized to measure the diversity of the generated grasps and measures how well the generated grasps cover the space of positive grasps G^*. We follow the same setting in [11], that only the distance in the translation of the grasps is used for evaluating whether a grasp is covered or not. Grasp g no further than 2 cm away from any of the grasp $\hat{g} \in G$ will be considered as a hit.

Time Cost is utilized to measure the time efficiency.

C. Simulation Experiments

We carry out most of our experiments in the physics simulator [39], since we can access complete object models which helps to evaluate the grasp sampler and the iterative refinement module. In addition, the gripper can move freely in simulation environments that we do not take motion-planning for robot arm in simulation. 58 objects from YCB dataset (seen) and 48 objects from EGAD! (unseen) are selected for comprehensive evaluation.

Why is VAE needed: Firstly, our method generates, on average, a grasp in around 0.16 seconds compared to the 35+ seconds required by GraspIt!, as shown in Table IV. VAE is 228 times faster. Secondly, given the completed object model, sample a single grasp candidate using approach-based sampling method is inefficient, diverse grasps can be obtained by choosing different approach vectors, while graspable approach direction can be sparse for certain objects *e.g.* mug and bowl. To encode the space of successful grasps efficiently, we use the VAE network.

Dimensionality of Latent Space: The dimensionality of the latent space is a key hyper-parameter influencing the quality of the generated grasps when training VAE. In general, a high-dimensional latent space helps to improve the capacity for reconstruction, but it can also lead to increasing the possibility of over-fitting. To study the influence of dimensionality of the latent space, we show a quantitative analysis with the increasing number of dimensions. Specifically, 1000 grasps are sampled for calculating the Success Rate and the Coverage Rate in Fig. 6. As shown in the figure, a dimensionality of four achieves the most appropriate balance between the Success Rate and the Coverage Rate. We choose this setting for all experiments afterwards. In Fig. 8, qualitative results show that our model can generate diverse sets of grasp for a same object geometry. In Fig. 7, qualitative results demonstrate what happens to the grasps when interpolating latent codes. For each dimension of the latent code, we sample 7 instances uniformly and normalize the color from blue to red, all other dimensions are set to 0. The top row shows the top view, the bottom row shows the side view. Some dimensions of the latent space encoding produce smooth interpolations between samples. For example, the first column seems to encode smooth translation in the z-axis, the fourth column seems to encode smooth rotation in the z-axis, the

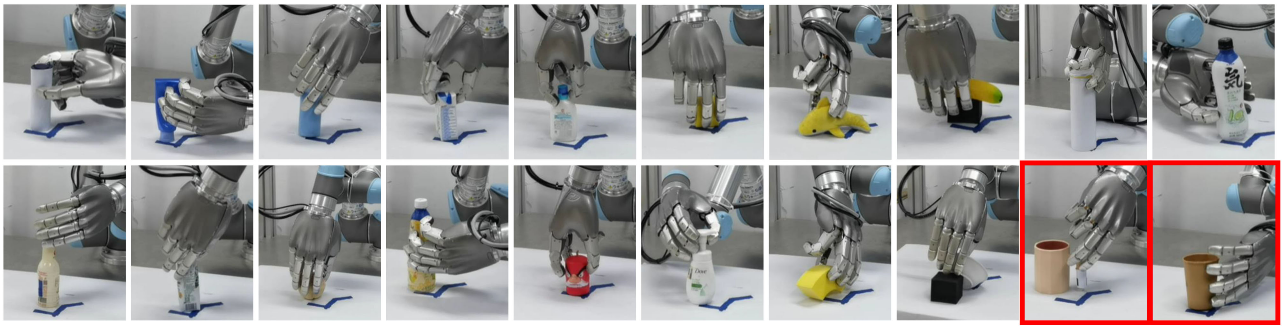

Fig. 11. Qualitative grasps generated by DVGG on 20 real objects. The upper row shows grasps for the similar objects, the bottom row shows grasps for the novel objects. Failure cases are shown in red boxes.

second and third column both may encode approach direction and grasp position jointly.

Effect of Iterative Refinement: To demonstrate the effectiveness of our proposed refinement module, we compare the performance of Success Rate and penetration and for both the grasp generator and refinement module on YCB dataset. As shown in Table II, the successful rate of the grasp configurations after refinement has 12.4% and 16.0% improvement respectively over the coarse grasps generated by the grasp sampler with one and two times refinements, and improvement gets saturated with higher iterations. The experimental results also demonstrate that the iterative refinement model helps in alleviating interpenetration between the hand model and the target objects. Fig. 9 shows four qualitative cases during the iterative refinement process.

Learned Grasp Sampler Vs. GraspIt! To illustrate the efficiency and quality of the generated grasp of our model, we follow the similar setting in [34] that we sample 360 grasp candidates on average in GraspIt! [1] within 75000 steps. Most of the grasp generated by GraspIt! are of low quality, only the top 20 grasp are used for evaluation. As for our method, we randomly sample 360 grasps for computing the sampling time and 20 random grasps of them to compute other metrics. Experimental results listed in Table IV show that our method outperforms baseline by a large margin in terms of grasp quality and Success Rate for both the seen and unseen objects. The results also demonstrate that: 1) the model trained with complete object point clouds outperforms the model trained with partial observed point clouds on the YCB dataset 2) the iterative refinement module can improve the grasps generated by GraspIt! as well.

Effect of Various Loss: To study the impact of various loss functions, we conduct an ablation study on these loss terms. As shown in Table III, we train our model with one of these loss terms removed, and only one time refinement is applied. We show the performance of Success Rate and Penetration on the YCB dataset. As expected, the network trained without penetration loss $L_{\mathcal{P}}$ achieves the lowest performance, since the hand model often intersects the object. The model trained without hand object contact loss $L_{\mathcal{C}}$ achieves the second lowest performance in terms of Success Rate, while it intersects the object slightly. We believe this is due to the lack of contact loss, which results in most grasps being far from the object. The model trained without regularization loss $L_{\mathcal{D}}$ performs slightly better than coarse grasps generated by the variational grasp generator. The network trained with no contact energy loss produces the highest Success Rate, since the contact energy loss is designed to generate tighter hand-object contact, which can not help to optimize unreasonable grasp, *e.g.* grasps far away from objects and grasps with collisions.

D. Robotic Experiments

We validate the effectiveness and reliability of our method in HIT-DLR II Hand with a UR5 robot arm. We capture the point cloud with the Ensenso N35 camera. Objects are presented to the robot individually on top of the foam pad, as shown in Fig. 10(a). We keep the following setting in actual robot platform experiments: 1) The camera captures the scene from the backside at a 60-degree viewpoint; 2) Objects are randomly placed within a 25x25 cm square area with stable poses and their point clouds segmented by subtracting the background; 3) 10 similar and 10 novel objects are selected to evaluate the generalization ability of the proposed network, as shown in Fig. 10(b); 4) We use the point completion method [40] described above; 5) Grasps cause collision with the ground will be filtered out. 6) We employ MoveIt for path planning of the UR5 robot arm. 7) We employ joint position control with extra +10 degrees for each flexion joint.

We compare **DVGG** to GraspIt! and PointNetGPD [43]. GraspIt! requires triangle-mesh based object model, we follow the poisson surface reconstruction algorithm to complete the target surface. Since PointNetGPD is proposed to classify parallel-jaw grasps, we take the following modifications: 1) We use GraspIt! to sample grasps. 2) Given the sampled grasps, we crop the point clouds within fixed radius to train the classification model proposed by [2].

The experimental procedure is as follows: 1) We generate 15 grasps per object for each algorithm; 2) Only physically reachable grasps will be executed. For GraspIt!, we adopt the widely used ϵ-metric to select grasps. As for PointnetGPD and our method, the first physical reachable grasp is executed. 3) A grasp will be classified as a successful grasp if the robot hand can grasp the target object and lift it to the predefined position without dropping it during translation.

As shown in Table V, our method outperforms baseline methods by a large margin, which demonstrates the superiority of our method. Qualitative results shown in Fig. 11 demonstrate that DVGG is able to generate diverse sets of grasps in terms of posture and grasping area.

In fact, DVGG also produced some failed grasps. Two examples are shown in Fig. 11. The main reason causing grasp failure is that unstable grasps require large frictional forces to lift the target object. As shown in the second example from the right in the bottom row, the failure grasp is caused by the low friction between the hand and the target object. Another reason for grasp failure is collision between the hand and the target object. The failure shown in the rightmost image of the bottom row is caused by displacement of the object due to slight collision during hand approaching.

VII. Conclusions

In this paper, we propose DVGG for generating diverse sets of grasps for high-DOF anthropomorphic robotic hand. Our method focuses on hand-object interaction constrain, which helps to estimate physical plausible grasps. Meanwhile, we build a large-scale synthetic grasping dataset with 300 objects with various shape. Experiments show that our model trained on the synthetic dataset performs well in real-world scenarios and outperforms the baseline by a large margin.

References

[1] A. T. Miller and P. K. Allen, "Graspit! a versatile simulator for robotic grasping," *IEEE Robot. Automat. Mag.*, vol. 11, no. 4, pp. 110–122, Dec. 2004.

[2] M. Ciocarlie, C. Goldfeder, and P. Allen, "Dexterous grasping via eigengrasps: A low-dimensional approach to a high-complexity problem" in *Proc. Robot.: Sci. Syst. Manipulation Workshop-Sens. Adapting Real World.*, Princeton, New Jersey, USA: Citeseer, 2007.

[3] H. Dai, A. Majumdar, and R. Tedrake, "Synthesis and optimization of force closure grasps via sequential semidefinite programming," in *Robotics Research*. Cham: Springer, 2018, pp. 285–305.

[4] V. Nguyen, "Constructing force-closure grasps," *Int. J. Robot. Res.*, vol. 7, no. 3, pp. 3–16, 1988.

[5] C. Ferrari and J. F. Canny, "Planning optimal grasps," in *Proc. IEEE Int. Conf. Robot. Automat.*, 1992, vol. 3, no. 4, pp. 6.

[6] J. Varley, C. DeChant, A. Richardson, J. Ruales, and P. Allen, "Shape completion enabled robotic grasping," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2017, pp. 2442–2447.

[7] J. Lundell, F. Verdoja, and V. Kyrki, "Robust grasp planning over uncertain shape completions," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2019, pp. 1526–1532.

[8] W. Wei *et al.*, "GPR: Grasp pose refinement network for cluttered scenes," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2021, pp. 4295–4302.

[9] S. Kumra and C. Kanan, "Robotic grasp detection using deep convolutional neural networks," in *Proc. IEEE Int. Conf. Intell. Robots Syst.*, 2017, pp. 769–776.

[10] 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 *Proc. IEEE Int. Conf. Robot. Automat.*, 2021, pp. 13459–13466.

[11] A. Mousavian, C. Eppner, and D. Fox, "6-DoF graspNet: Variational grasp generation for object manipulation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 2901–2910.

[12] J. Varley, J. Weisz, J. Weiss, and P. Allen, "Generating multi-fingered robotic grasps via deep learning," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2015, pp. 4415–4420.

[13] U. R. Aktas, C. Zhao, M. Kopicki, A. Leonardis, and J. L. Wyatt, "Deep dexterous grasping of novel objects from a single view," 2019, *arXiv:1908.04293*.

[14] M. S. Kopicki, D. Belter, and J. L. Wyatt, "Learning better generative models for dexterous, single-view grasping of novel objects," *Int. J. Robot. Res.*, vol. 38, no. 10–11, pp. 1246–1267, 2019.

[15] M. Liu, Z. Pan, K. Xu, K. Ganguly, and D. Manocha, "Generating grasp poses for a high-dof gripper using neural networks," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2019, pp. 1518–1525.

[16] M. Liu, Z. Pan, K. Xu, K. Ganguly, and D. Manocha, "Deep differentiable grasp planner for high-DoF grippers," in *Proc. Robot.: Sci. Syst. XVI*, 2020, P. 66. [Online]. Available: http://www.roboticsproceedings.org/rss16/p066.html

[17] Q. Lu, M. Van der Merwe, B. Sundaralingam, and T. Hermans, "Multifingered grasp planning via inference in deep neural networks: Outperforming sampling by learning differentiable models," *IEEE Robot. Automat. Mag.*, vol. 27, no. 2, pp. 55–65, Jun. 2020.

[18] Q. Lu, M. Van der Merwe, and T. Hermans, "Multi-fingered active grasp learning," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2020, pp. 8415–8422.

[19] B. Wu *et al.*, "Generative attention learning: A "general" framework for high-performance multi-fingered grasping in clutter," *Auton. Robots*, vol. 44, no. 6, pp. 971–990, 2020.

[20] H. Jiang, S. Liu, J. Wang, and X. Wang, "Hand-object contact consistency reasoning for human grasps generation," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2021, pp. 11107–11116.

[21] O. Taheri, N. Ghorbani, M. J. Black, and D. Tzionas, "GRAB: A dataset of whole-body human grasping of objects," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 581–600.

[22] E. Corona, A. Pumarola, G. Alenya, F. Moreno-Noguer, and G. Rogez, "Ganhand: Predicting human grasp affordances in multi-object scenes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 5031–5041.

[23] 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 *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 361–378.

[24] B. Calli, A. Walsman, A. Singh, S. Srinivasa, P. Abbeel, and A. M. Dollar, "The YCB object and Model set: Towards common benchmarks for manipulation research," in *Proc. IEEE Int. Conf. Adv. Robot.*, 2015, pp. 510–517.

[25] D. Morrison, P. Corke, and J. Leitner, "EGAD! an evolved grasping analysis dataset for diversity and reproducibility in robotic manipulation," *IEEE Robot. Automat. Lett.*, vol. 5, no. 3, pp. 4368–4375, Jul. 2020.

[26] C. Borst, M. Fischer, and G. Hirzinger, "A fast and robust grasp planner for arbitrary 3D objects," in *Proc. IEEE Int. Conf. Robot. Automat.*, vol. 3, 1999, pp. 1890–1896.

[27] Y. Zheng, "An efficient algorithm for a grasp quality measure," *IEEE Trans. Robot.*, vol. 29, no. 2, pp. 579–585, Apr. 2013.

[28] J. Mahler *et al.*, "DEX-Net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics," in *Proc. Robot.: Sci. Syst.*, 2017, pp. 5620–5627.

[29] J. Redmon and A. Angelova, "Real-time grasp detection using convolutional neural networks," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2015, pp. 1316–1322.

[30] H.-S. Fang, C. Wang, M. Gou, and C. Lu, "GraspNet-1Billion: A large-scale benchmark for general object grasping," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 11441–11450.

[31] S. Hampali, M. Rad, M. Oberweger, and V. Lepetit, "Honnotate: A method for 3D annotation of hand and object poses," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 3196–3206.

[32] G. Garcia-Hernando, S. Yuan, S. Baek, and T.-K. Kim, "First-person hand action benchmark with rgb-d videos and 3D hand pose annotations," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 409–419.

[33] Y. Hasson *et al.*, "Learning joint reconstruction of hands and manipulated objects," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 11799–11808.

[34] J. Lundell *et al.*, "Multi-finGAN: Generative coarse-to-fine sampling of multi-finger grasps," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2021, pp. 4495–4501.

[35] J. Lundell, F. Verdoja, and V. Kyrki, "DDGC: Generative deep dexterous grasping in clutter," *IEEE Robot. Automat. Lett.*, vol. 6, no. 4, pp. 6899–6906, 2021.

[36] S. Brahmbhatt, A. Handa, J. Hays, and D. Fox, "Contactgrasp: Functional multi-finger grasp synthesis from contact," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2019, pp. 2386–2393.

[37] L. Shao *et al.*, "Unigrasp: Learning a unified model to grasp with multifingered robotic hands," *IEEE Robot. Automat. Lett.*, vol. 5, no. 2, pp. 2286–2293, Apr. 2020.

[38] T. Feix, J. Romero, H.-B. Schmiedmayer, A. M. Dollar, and D. Kragic, "The grasp taxonomy of human grasp types," *IEEE Trans. Hum.- Mach. Syst.*, vol. 46, no. 1, pp. 66–77, Feb. 2016.

[39] E. Todorov, T. Erez, and Y. Tassa, "Mujoco: A physics engine for model-based control," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2012, pp. 5026–5033.

[40] L. Pan *et al.*, "Variational relational point completion network," in *IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 8520–8529.

[41] K. Sohn, H. Lee, and X. Yan, "Learning structured output representation using deep conditional generative models," *Adv. Neural Inf. Process. Syst.*, vol. 28, pp. 3483–3491, 2015.

[42] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 77–85.

[43] H. Liang *et al.*, "PointNetGPD: Detecting grasp configurations from point sets," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2019, pp. 3629–3635.