Deep Learning Reactive Robotic Grasping with a Versatile Vacuum Gripper

Hui Zhang, *Member, IEEE*, Jef Peeters, Eric Demeester, Karel Kellens

***Abstract*—In this paper, a 6-step approach is proposed to simulate the grasp and evaluate the grasp quality for a versatile vacuum gripper by tracking the deformation and force-torque wrench of the gripping pad. Over 100 K synthetic grasps are generated for neural network training. Furthermore, a Gripping Attention Convolutional Neural Network (GA-CNN) is developed to predict the grasp quality for real-world grasp, running by 15 Hz closed-loop control with the real-time robotic observation and force-torque feedback. Various experiments in both the simulation and physical grasps indicate that our GA-CNN can focus on the crucial region of the soft gripping pad to predict grasp qualities and perform a lower average error compared with a same-scale traditional CNN. Additionally, the complexity of grasping clutters is defined from Level 1 to Level 9. The proposed grasping method achieves an average success rate of 90.2% for static clutters at Level 1 to Level 8 and an average success rate of >80.0% for dynamic grasping at Level 1 to Level 7, which outperforms state-of-the-art grasping methods.**

***Index Terms*—Contact modeling, deep learning in robotics and automation, grasping, reactive and sensor-based planning.**

I. Introduction

ROBOTIC grasping and manipulation of unknown objects in an unstructured environment remain challenging due to the limitations in robotic perception and control, including perception disturbances, control errors, object deformations and environmental uncertainties.

Two primary research lines can be distinguished for the robotic grasping of unknown objects. One of them aims to develop dexterous grippers to fit complex objects. In the early stage [1], [2], multi-joint fingers were mounted on the dexterous gripper to increase the Degree of Freedom (DoF). A large amount of recent research indicated that novel grippers become more and more flexible with the help of under-actuated and adaptable soft gripping pads/fingers [3], [4].

Another line of research on the robotic grasping of unknown objects is to develop grasping methods based on various robotic perceptions. Robotic vision has been widely used to extract the primitive shapes of objects and detect feasible grasp regions [5]. Moreover, haptic and force-torque perceptions can be integrated into the grasping method to improve robotic performances [6], [7]. Most recent research revealed that neural networks trained on an extensive dataset can learn grasp principles and detect grasp poses for unknown objects [8], [9]. Many open-source datasets, including the ACRV Picking Benchmark (APB) [10] and Cornell Grasping [11], are available online to train neural networks for grasp planning, which typically contain millions of RGB-D images with manually-marked grasp regions, or grasp examples with corresponding grasp qualities.

These research lines are not contradictory. A robust hybrid grasping solution can be developed, combing a versatile gripper with a neural network. However, most open-source datasets are merely compatible with traditional parallel-jaw grippers and vacuum grippers. Unlike a traditional parallel-jaw gripper, it is not easy to collect grasp examples for a dexterous gripper by manual labeling [11] or grasp simulation [12], [13] with human-designed grasp principles due to the complex deformation of the dexterous gripper during grasping.

Fig. 1 presents the overview of the proposed hybrid grasping method. This paper extends our previous framework [14], and makes three contributions:

1) A novel 6-step method is proposed for the grasp simulation with a versatile vacuum gripper. The method evaluates grasp quality for a virtual grasp by tracking the deformation and force-torque wrench of the gripping pad under quasi-static conditions. The contact surface is tracked by a set of sub surfaces, instead of contact points in many existing methods. Over 100 K grasps are synthesized for neural network training.
2) A GA-CNN is designed to learn grasp principles for a versatile vacuum gripper, which is a 52-layer linear regression architecture with gripping attention modules. It is trained on a massive dataset with synthetic grasps to predict grasp quality focusing on the crucial region on the soft gripping pad, which performs a higher prediction accuracy than a traditional CNN with a similar architecture and the same number of parameters (3.23 M). Moreover, a closed-loop GA-CNN grasping method

Manuscript received August 11, 2022; accepted November 15, 2022. This work was supported by the China Scholarship Council under grant CSC201806090290. This paper was recommended for publication by Associate Editor J. Bohg and Editor W. Burgard upon evaluation of the reviewers' comments. *(Corresponding author: Hui Zhang.)*

Hui Zhang and Karel Kellens are with the ACRO Research Group, Department of Mechanical Engineering, Wetenschapspark 27, 3590 Diepenbeek, Belgium, also with the Flanders Make ROB Core Lab@KU Leuven, 3000 Leuven, Belgium (hui.zhang@kuleuven.be).

Jef Peeters is with the LCE Research Group, Department of Mechanical Engineering, Celestijnenlaan 300, 3001 Heverlee, Belgium, also with the Flanders Make VCCM Core Lab@KU Leuven, 3000 Leuven, Belgium.

Eric Demeester is with the ACRO Research Group, Department of Mechanical Engineering, Wetenschapspark 27, 3590 Diepenbeek, Belgium.

Code and video demo are available at https://github.com/huikul/Versatile Grasping.

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TRO.2022.3226148.

Digital Object Identifier 10.1109/TRO.2022.3226148.

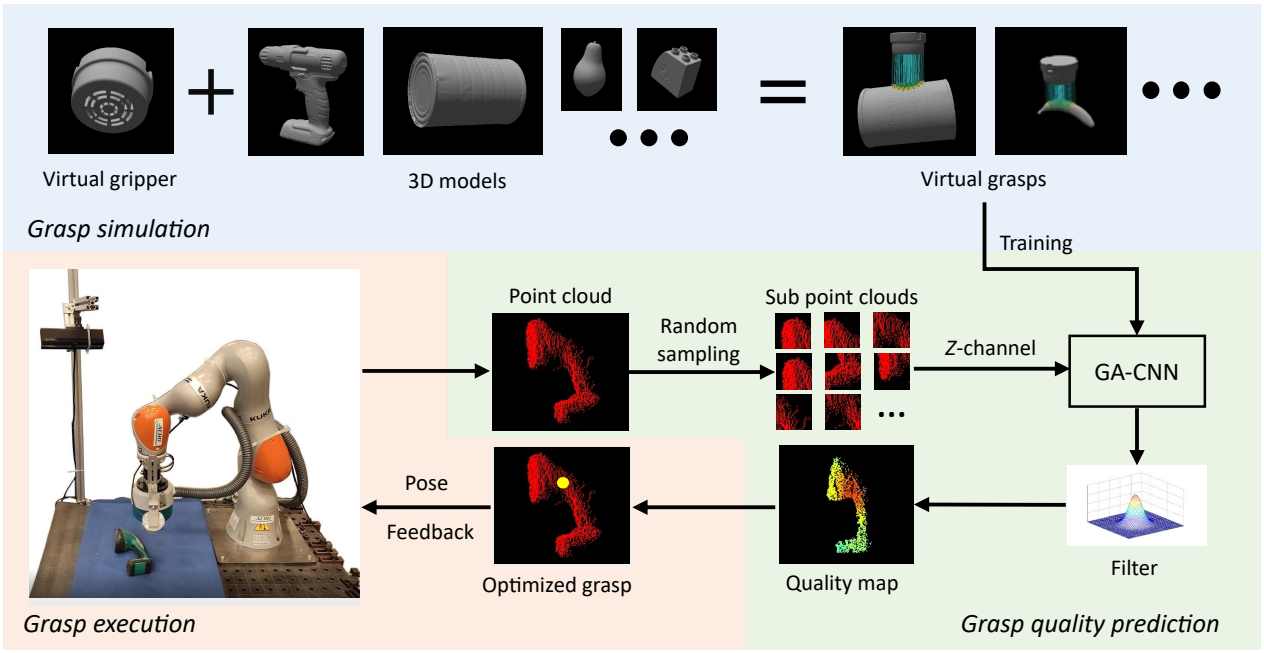

Fig. 1. Overview of the proposed reactive grasping method.

in SE(3) is developed based on the real-time robotic observation and force-torque wrench, which can grasp both static and moving objects.

3) The complexity of grasping clutters for vacuum grippers is defined from Level 1 to Level 9 according to objects' shapes and distributions. Benchmark experiments for the proposed grasping method and state-of-the-art methods are investigated with static scenes, dynamic scenes and multi-perspective robotic observations at Level 1 to Level 9.

Benefiting from the adaptability of both the 6-step grasp simulation and the GA-CNN, the combination of them could be used for the dataset collection and grasp learning of a novel soft gripper.

The remainder of this paper is organized as follows: Section II discusses the related work. In Section III, concepts and targets of the proposed method are described. Section IV introduces the grasp simulation in detail. The architecture and fine-tuning of the GA-CNN are presented in Section V. The implementation of the GA-CNN grasping method is presented in Section VI. Section VII demonstrates a series of practical experiments for the proposed method. Finally, Section VIII summarizes and concludes the performed work.

II. Related Work

A. Grasping of Unknown Objects

Grasping of unknown objects refers to the grasp detection and execution for objects without requiring their CAD models. The current grasping methods for unknown objects can be classified into three categories according to their basic frameworks: analytic methods, empirical methods and synthetic methods.

Analytic methods for unknown objects grasping typically extract the primitive shapes of objects and take them as simplified geometries [5], [15], like cubes, cylinders, cones, to detect feasible grasp poses. For example, Herzog *et al.* [16] proposed a shape-template-based matching algorithm for unknown objects grasping. Analytic methods assume that similarly shaped objects can be grasped in a similar way. However, the high-quality grasp for a primitive shape is not always the same as that for a real object. Hence, these methods cannot consistently execute successful grasps.

Empirical methods use deep neural networks to learn grasp principles. The neural networks for grasp pose detection are generally trained on abundant grasps with manual labels [17]. Reinforcement learning [18], [19] is an alternative approach to develop a neural network for grasping unknown objects. This approach collects grasps by physical robotic trials that detect failed/successful grasps with various sensors [20], [21], such as a depth camera, a force-torque sensor or a haptic sensor [22]. Nonetheless, a tedious collection of grasps is required. For instance, 700 robot hours were consumed to collect 50 K grasps in the research by Pinto *et al.* [18]. Levine *et al.* [23] ran two months of physical trials with 14 robots to collect 800 K grasps. The recent research by Dasari *et al.* [24] reported that video frames can be used to train a neural network for random picking. These studies concluded that neural networks trained on a plenty of grasps can detect grasp poses for unknown objects with >70.0% success rates.

Synthetic methods can be seen as an evolutionary version of empirical methods. Instead of collecting grasps by tediously human labeling or time-consuming robotic trials, synthetic methods generate datasets in grasp simulation. Typically, a virtual gripper is defined to grasp 3D models from different perspectives. Each grasp is recorded by a point cloud [9], [25] or a depth image [12], [26], with the corresponding grasp quality evaluated by human-designed metrics, such as force closure [27], friction closure [28] or Grasp Wrench

Space (GWS) [29]. Even though a neural network trained on synthetic grasps can predict the grasp quality for a real-world grasp, the grasp simulation remains challenging, especially for a dexterous/versatile gripper.

Furthermore, many of these methods above detect grasp poses merely based on robotic vision and execute physical grasping trials with open-loop control. The precise calibration and control of a robotic system are demanded. Once a grasp pose is determined, it cannot be pruned anymore. Therefore, open-loop control is unable to deal with moving objects and environments where the real-time calibration is difficult to be implemented. Both of them are typical cases for random picking on a conveyor belt.

B. Grasping with Dexterous Grippers

Dexterous grippers have been widely applied for the manipulation under shape uncertainties [30]. Unlike traditional parallel-jaw and vacuum grippers, dexterous grippers have more DoFs to fit an object and resist external disturbances. Early research on multi-finger grasping engaged in finding feasible contact points considering force closure [31] and kinematic constraints [32]. Bohg *et al.* [33] developed a grasp planning algorithm for multi-finger grippers, which detects the optimal grasp pose by matching real-world grasp scenes to pre-analyzed 3D models, and thus only works with known objects.

In recent years, under-actuated grippers and soft grippers have been attracting great research. The relevant studies involve both the devising and grasp planning of novel dexterous grippers. Due to the high flexibility of a dexterous gripper, the contact surface is often simplified into a set of contact points [34], [35], and the grasp quality is evaluated based on the contact stiffness, hand kinematics, dynamic frictions, etc., in a projected low-dimension space. However, the contact surface for a versatile gripper with soft gripping pads consists of several sub surfaces, instead of points. As a result, the simplified model with contact points only works well for grippers with rigid gripping pads/fingers. Some latest studies explored the contact modeling and simulation for deformable grippers by tracking nonplanar contact surface [36], [37]. They contributed more on grasp simulation, but relatively less on physical random picking in dense clutters.

In this paper, the authors address the grasp quality evaluation for a soft gripper by tracking the deformation of the contact surface with both the geometric and physical restrictions in grasp simulation. The contact surface is tracked by a set of sub surfaces, instead of contact points (Section IV). Besides, the proposed grasp evaluation method can be easily adjusted to fit other soft grippers. A deep neural network is investigated based on the grasp simulation to predict grasp quality for real-world random picking (Section V).

C. Robotic Grasping with Multi-Sensor Perceptions

A significant advantage of closed-loop grasping is the ability to adjust the robotic motion in a dynamic environment. Visual feedback is a popular solution to optimize the robotic motion towards a desired pose [38]. Morrison *et al.* [39] presented a reactive grasp pose detection approach for random picking, which built the closed-loop control with a wrist-mounted depth camera.

Actually, the vision-based reactive grasping is limited, as not every grasp status can be monitored via robotic vision. For instance, the in-hand deformation and slip for a soft object are difficult to be measured with a camera, and multi-sensor fusion is necessary for the grasping with a flexible gripper [40]. The fusion of robotic vision and 6-DoF gripper force-torque wrench is a proper solution to detect the grasp status and control the robotic motion in a dynamic scene [41]. Xiong *et al.* proposed a novel AMDL method, which first explored the force correlations among multiple fingers via tactile sensors and significantly improved the grasping state recognition of multi-finger grasping [42]. Besides, Huh *et al.* proposed a haptic sensing method to optimize the grasping performance of a multi-chamber vacuum gripper via a LSTM neural network [43].

In this paper, the authors present a reactive grasping method with a versatile vacuum gripper (Section VI). The real-time point cloud and gripper force-torque wrench are integrated into the closed-loop control to detect feasible grasp poses and monitor the grasp status. The real-time feedback enables the robot to grasp unknown objects in dynamic scenes.

D. Benchmarks for Robotic Grasping

Numerous benchmarks are available online for grasp simulation and physical tests. On the one hand, some researchers developed open-source simulators, for example, GraspIt! [44], OpenGRASP [45] and SynGrasp [46], for flexible grasp simulations based on physical principles. On the other hand, Calli *et al.* released the YCB grasp benchmarks [47], containing reconstructed 3D meshes and authentic RGB-D images sampled by physical depth cameras from different perspectives. The similar open-source datasets involve the KIT object database [48], Cornell Grasping, Dex-Net dataset [12] and APB. More recent work, like the Jacquard dataset [13], has collected millions of grasps with synthetic marks for parallel-jaw grippers, which can be directly used to train a neural network and liberate the follow-up researchers from tedious marking work [49].

Nevertheless, existing training datasets mainly aim to train neural networks for traditional parallel-jaw grippers, because it is challenging to manually mark graspable regions in 3D point clouds or depth images for a versatile gripper. Consequently, training a neural network on synthetic grasps for a versatile gripper is a practicable approach. The subsequent problem is that many released grasp simulators synthesize grasps by a simplified virtual gripper with several contact points, which is not precise enough for a soft gripper with deformable gripping pads/fingers. Also, the computational complexities of existing simulators are not favorable to generate a massive dataset.

In addition, many items from the benchmarks above do often not fit the investigated grippers in the aspects of sizes, weights, roughness, rigidity, etc. New items are usually needed for the physical tests with a novel gripper, causing an unfair comparison between different robotic grasping methods.

To address these issues, the authors propose a 6-step method to track the deformation and simulate the grasp for

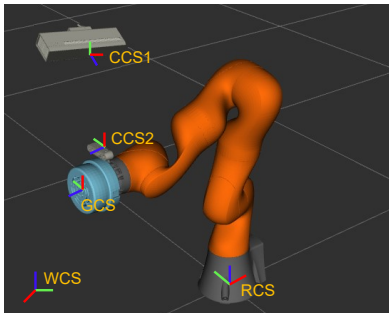

Fig. 2. Coordinate systems for grasp planning. Red, green and blue lines are the x, y, z axes of the used coordinate systems.

a soft gripper (Section IV). Furthermore, over one hundred 3D objects from the YCB, KIT and Dex-Net datasets are evaluated to define a metric for the complexity of the object's shape based on the normalized surface-area-to-volume ratio (*NSVR*) (Section VII-B). The complexity of picking clutters with vacuum grippers is defined from Level 1 to Level 9, considering the complexity and distribution of objects. New objects and clutters can be defined with similar principles.

III. Problem Statements

As shown in Fig. 1, given a 3D point cloud from a robotic observation, the grasp planning problem is to select a set of grasp candidates, evaluate their grasp qualities and find a robust grasp pose to pick up the object. In this paper, the problem can be described by the following processes: grasp simulation, grasp quality prediction and grasp execution.

A. Coordinate Systems

Four coordinate systems are built for the grasp simulation and physical grasp, named Camera Coordinate System (CCS), World Coordinate System (WCS), Robot Coordinate System (RCS) and Gripper Coordinate System (GCS). The details of them are shown in Fig. 2, and there are CCS1 and CCS2 listed for the static and wrist-mounted cameras, respectively.

B. Grasp Simulation

The primary task of the grasp simulation is to synthesize grasps using a virtual gripper and 3D meshes of objects. The grasp simulation digitizes the grasp quality by a numerical value relevant to the properties of the used gripper and 3D objects, and records the grasp with a point cloud.

In the grasp simulation, the 3D mesh of an object $\mathcal{O}$ is located with a random pose $\boldsymbol{P}_{\mathcal{O}}$ in SE(3) to simulate a randomly stacked object in the real world. A virtual versatile gripper $\mathcal{G}$ tries to grasp the object with a pose $\boldsymbol{P}_{\mathcal{G}}$ in SE(3). The grasp quality q is evaluated by human-designed principles, named $q = Q(\boldsymbol{P}_{\mathcal{O}}, \boldsymbol{P}_{\mathcal{G}}, S_{\mathcal{GO}})$, where $S_{\mathcal{GO}}$ is a set of parameters to describe grasp states, such as the coefficient of friction μ between the gripper and the object, physical features and limitations of the gripper and the object. A virtual camera $\mathcal{C}$ is deployed at the pose $\boldsymbol{P}_{\mathcal{C}}$ to render the grasp scene. Each grasp candidate $g(\mathcal{P}, q)$ is represented by the grasp observation $\mathcal{P}$ in GCS and the corresponding grasp quality q. An in-depth description of the proposed 6-step grasp simulation can be found in Section IV.

C. Grasp Quality Prediction

The goal of the grasp quality prediction $\hat{q} = Q_{\Theta}(\mathcal{P})$ is to learn grasp principles via the GA-CNN to predict the quality value $\hat{q}$ when the grasp observation $\mathcal{P}$ is given in a real-world grasping trial, where Θ defines the parameters of the proposed neural network. The GA-CNN is trained on millions of synthetic grasps $g(\mathcal{P}, q)$, which is described in Section V.

D. Grasp Execution

During physical grasping, the real-time 3D point clouds of grasp scenes are captured in CCS. A number of sub point clouds $\mathbb{P} = \{\mathcal{P}_1, \mathcal{P}_2, \cdots, \mathcal{P}_i, \cdots, \mathcal{P}_u\}$ are randomly sampled and transformed into GCS for the grasp quality prediction. The central pose of $\mathcal{P}_i$ is transformed into RCS to give a robotic pose $\boldsymbol{P}_{\mathcal{R}}(\mathcal{P}_i, \hat{q}_i, \boldsymbol{M}_{\mathcal{F}})$ in SE(3), according to the predicted grasp quality $\hat{q}_i$ and the real-time force-torque wrench of the gripper base $\boldsymbol{M}_{\mathcal{F}}$. The grasp execution is thoroughly presented in Section VI.

IV. Grasp Simulation

Grasp simulation plays a fundamental role in synthesizing a large-scale grasp dataset. In this paper, a 6-step grasp simulation is proposed to estimate the grasp quality for a versatile soft gripper. The grasp simulation is implemented with three assumptions to simplify the computational complexity:

1) All forces and torques are calculated based on Quasi-static physics with Coulomb friction.
2) Each target object has an airtight surface and a rigid body.
3) The weights of the target objects and the torques on the x, y axes are ignored.

The six steps in the simulation are summarized into:

1) Build a geometric model for the gripper.
2) Define geometric restrictions for the gripper.
3) Track the deformation of the virtual gripper.
4) Calculate GWS vectors for the contact surface.
5) Define physical restrictions for grasping.
6) Optimize the GWS vectors and estimate the grasp quality.

The following subsections IV-A to IV-G present the elaborate procedures of the grasp simulation considering the characteristics of the adopted gripper shown in Fig. 3 [50].

A. Geometric Model

The used versatile gripper $\mathcal{G}$ consists of a base frame and a soft gripping pad supported by an air-permeable cushion with a granulate filling. The functional principle of $\mathcal{G}$ is based on the gripping by vacuum and form closure, and the shape adaptation and solidification of the gripping cushion by airflow. The gripper cushion can freely deform or completely solid to fit a complex surface as presented in Fig. 3 (b).

Assuming that the gripping pad contacts an object with an airtight surface, part of the gripping pad will be located on

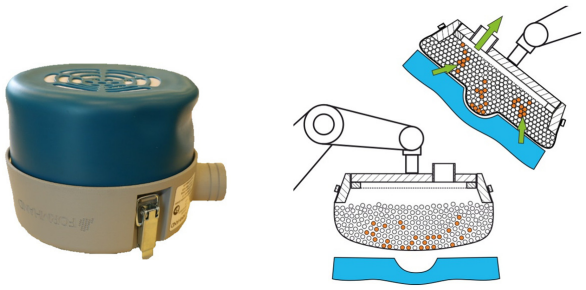

Fig. 3. Versatile vacuum gripper and its grasping demonstration [50]. (a) A FORMHAND high-adaptability vacuum gripper with a radius of 75 mm. (b) Soft cushion and gripping pad of the gripper.

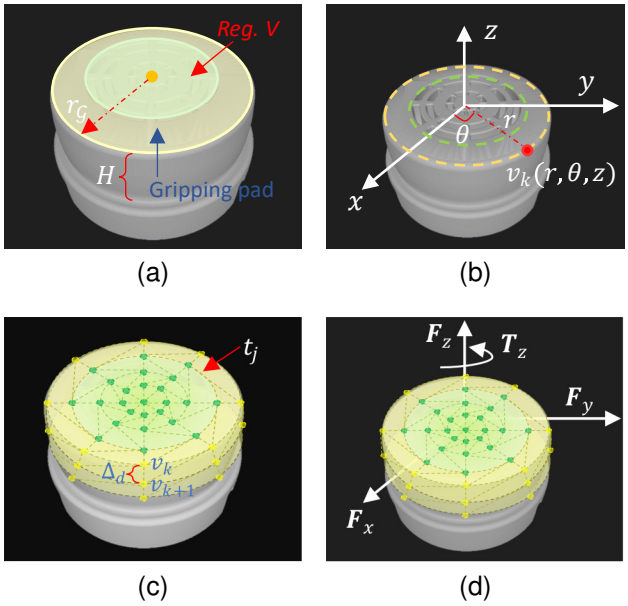

Fig. 4. Virtual gripper in the grasp simulation. (a) Soft gripping pad. (b) Vertices in the cylindrical system. (c) Triangles on the gripping pad. (d) A 4-DoF force-torque wrench. Note: 1) Δ_d restricts the euclidean distance for the neighbor points $\boldsymbol{v}_k$ and $\boldsymbol{v}_{k+1}$. 2) This figure simplifies the vertices and triangles on the gripping pad for visualization, and more triangles (about 600) are tracked in the practical grasp simulation.

the contact surface. A geometric model of the gripping pad is built in Fig. 4 to track the deformation, following the gripper base with a radius of $r_{\mathcal{G}}$ and a height of H in a cylindrical coordinate system. The orthogonal normal of the cylindrical coordinate system is defined as $\boldsymbol{n}_r$, $\boldsymbol{n}_\theta$ and $\boldsymbol{n}_z$ in (1). A set of points $\mathbb{V} = \{\boldsymbol{v}_1, \cdots, \boldsymbol{v}_k, \cdots, \boldsymbol{v}_m\}$ with $k = 1, 2, \cdots, m$ can be found on the gripping pad by the radial step Δ_r and the rotation step Δ_θ, wherein each point is defined as $\boldsymbol{v}_k(r, \theta, z)$. The whole gripping pad is simplified by a set of triangular planes $\mathbb{T} = \{t_1, \cdots, t_j, \cdots, t_n\}$, $\mathbb{T} \neq \emptyset$ with 3-neighbor vertices, as presented in Fig. 4 (c)-(d). Notably, the sets $\mathbb{V}$ and $\mathbb{T}$ do not always contain certain elements during grasping, as part of the gripping pad could not contact the object surface. The neighbor points with the same rotation angle are defined by $\boldsymbol{v}_k(r_k, \theta, z_k)$, $\boldsymbol{v}_{k+1}(r_{k+1}, \theta, z_{k+1})$ and $r_{k+1} > r_k$ in the following equations.

B. Geometric Restrictions

The gripping pad is deformable under the constraints of the gripper cushion. Given a contact surface, the projection of the gripping pad should always keep in a round shape on the plane $\prod(r, \theta, 0)$, as described in (2)-(3).

Moreover, the green region *Reg.V* on the gripping pad, as marked in Fig. 4 (a), is the most meaningful region to make an air-pressure differential p between the gripping pad and the atmosphere for grasping. The radius of *Reg.V* is a constant r_v on the 3D surface. Therefore, the cover area A^{cov} between the object and *Reg.V* is not larger than the area of *Reg.V* at the static status, which can be restricted by the neighbor points and surface area in (4)-(5).

In addition, the radial step $\Delta_r > 0$ and the distance step $\Delta_d > 0$ constrain neighbor points to avoid unrealistic deformation, as listed in (6)-(7).

$$\boldsymbol{n}_r = (1,0,0), \quad \boldsymbol{n}_\theta = (0,1,0), \quad \boldsymbol{n}_z = (0,0,1) \tag{1}$$

$$0 \leq r \leq r_{\mathcal{G}}, 0 \leq \theta < 2\pi, -H \leq z \leq 0 \tag{2}$$

$$0 \leq \boldsymbol{v}_k \cdot \boldsymbol{n}_r \leq r_{\mathcal{G}} \tag{3}$$

$$\sum \|\boldsymbol{v}_{k+1}(r_{k+1}, \theta, z_{k+1}) - \boldsymbol{v}_k(r_k, \theta, z_k)\| = r_v, \quad \boldsymbol{v}_k, \boldsymbol{v}_{k+1} \in \textit{Reg.V} \tag{4}$$

$$A^{cov} = \sum A^{t_j} \leq A^{\textit{Reg.V}} = \pi r_v^2, t_j \in \textit{Reg.V} \tag{5}$$

$$0 \leq (\boldsymbol{v}_{k+1}(r_{k+1}, \theta, z_{k+1}) - \boldsymbol{v}_k(r_k, \theta, z_k)) \cdot \boldsymbol{n}_r \leq \Delta_r \tag{6}$$

$$\|\boldsymbol{v}_{k+1}(r_{k+1}, \theta, z_{k+1}) - \boldsymbol{v}_k(r_k, \theta, z_k)\| \leq \Delta_d \tag{7}$$

C. Deformation Tracking

Tracking the deformation for the gripping pad is the prerequisite to compute the force-torque wrench and estimate the grasp quality.

The soft gripping pad of $\mathcal{G}$ can freely deform to fit various non-flat contact surfaces. Given an object $\mathcal{O}$ with a random pose $\boldsymbol{P}_{\mathcal{O}}$ in SE(3), an upper surface point $\boldsymbol{c}(x, y, z)$ and a grasp direction $\boldsymbol{d}_{\mathcal{G}}$ are randomly selected to build a GCS and define a grasp pose $\boldsymbol{P}_{\mathcal{G}}$ in SE(3). When the gripper $\mathcal{G}$ contacts the object at the point $\boldsymbol{c}(x, y, z)$, the sets $\mathbb{V}$ and $\mathbb{T}$ are re-mapped into new coordinates to track the deformation of the gripping pad. The points set $\mathbb{V}$ is projected on the contact surface under the restrictions shown in (2)-(7), and only the points located on the contact surface are considered for further processing. The triangles in $\mathbb{T}$ are re-collected based on new 3-neighbor points. Fig. 5 exhibits three examples of deformation tacking for the gripping pad in the simulation and real world.

D. GWS Vector Calculation

Force-torque wrench has been widely investigated as a GWS vector to evaluate the grasp robustness in simulation. Generally, a force-torque wrench consists of the forces $\boldsymbol{F}_x$, $\boldsymbol{F}_y$, $\boldsymbol{F}_z$ and the torques $\boldsymbol{T}_x$, $\boldsymbol{T}_y$, $\boldsymbol{T}_z$ in the GCS. A simplified GWS vector of a grasp is defined as a 4D wrench $[\|\boldsymbol{F}_x\|, \|\boldsymbol{F}_y\|, \|\boldsymbol{F}_z\|, \|\boldsymbol{T}_z\|]^T$ in Fig. 4 (d), regarding the assumptions mentioned before Section IV-A. The GWS vector is analyzed by a set of sub force-torque wrenches

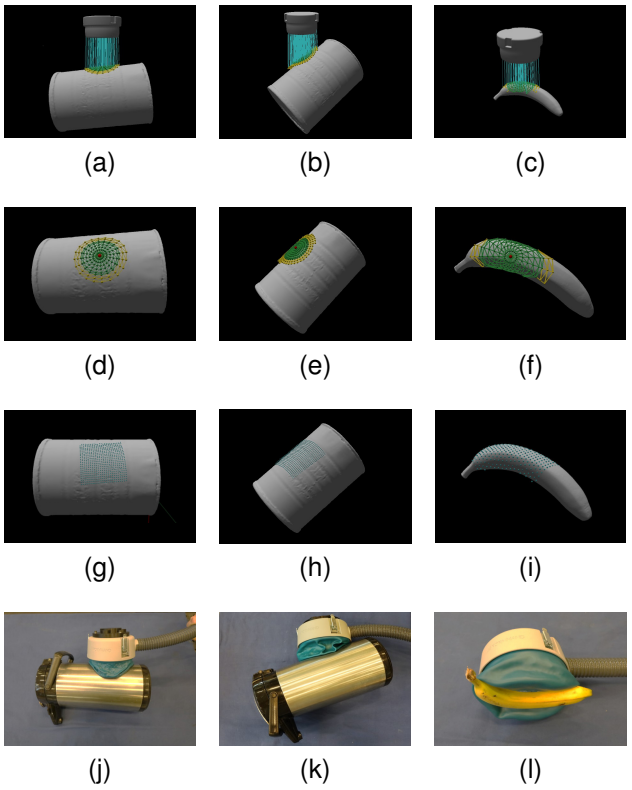

Fig. 5. Three grasps in the simulation. (a)-(c) Virtual grasps. (d)-(f) Tracking of the sets $\mathbb{V}$ and $\mathbb{T}$. (g)-(i) The corresponding point clouds. (j)-(l) The corresponding deformations with the similar objects in the real world. Note: 1) All grasps are visualized in GCS. 2) Green triangles are located on *Reg.V*, and yellow triangles are out of *Reg.V*. 3) This figure simplifies the tracking of vertices and triangles for visualization, and more triangles (around 600) are tracked in the practical grasp simulation.

on the triangular gripping pads in $\mathbb{T}$ as described in (8)-(9), where w_j is the weight value of a triangle t_j, and $[\|\boldsymbol{f}_x^{t_j}\|, \|\boldsymbol{f}_y^{t_j}\|, \|\boldsymbol{f}_z^{t_j}\|, \|\boldsymbol{\tau}_z^{t_j}\|]^T$ is the force-torque wrench of t_j.

As a result, the grasp matrix of a grasping trial is denoted as $G \in \mathbb{R}^{4\times n}$, wherein each column is the GWS vector $[\|\boldsymbol{f}_x^{t_j}\|, \|\boldsymbol{f}_y^{t_j}\|, \|\boldsymbol{f}_z^{t_j}\|, \|\boldsymbol{\tau}_z^{t_j}\|]^T$ for a sub triangular gripping pad t_j. The weight matrix is formulated as $W = [w_1, w_2, \cdots, w_j, \cdots, w_n]^T$.

$$\boldsymbol{F}_x = \sum_{j=1}^{n} w_j \boldsymbol{f}_x^{t_j}, \quad \boldsymbol{F}_y = \sum_{j=1}^{n} w_j \boldsymbol{f}_y^{t_j}, \quad \boldsymbol{F}_z = \sum_{j=1}^{n} w_j \boldsymbol{f}_z^{t_j} \tag{8}$$

$$\boldsymbol{T}_z = \sum_{j=1}^{n} w_j \boldsymbol{\tau}_z^{t_j} \tag{9}$$

E. Physical Restrictions

Physical restrictions mainly define the range of the force-torque wrench for the used gripper and ensure all forces and torques in the simulation to follow real-world physical principles. The force-torque wrench $[\|\boldsymbol{F}_x\|, \|\boldsymbol{F}_y\|, \|\boldsymbol{F}_z\|, \|\boldsymbol{T}_z\|]^T$ for a soft contact model is restricted by the elliptical equation [28]. Furthermore, the weight value w_j is also limited by the physical properties of a gripper. All formulas of the physical restrictions are listed in (22)-(28) in the Appendix.

F. Grasp Quality Estimation

In the ideal case, a perfect GWS vector in (10) is conducted when the gripper contacts a flat surface with the maximum value of p. The minimum Euclidean norm between the GWS vector of an actual grasping trial and that of the ideal grasping trial is a suitable reference to evaluate the grasp quality, named $min\|GW - \boldsymbol{\Lambda}\|$. A normalized grasp quality $q \subseteq (0.0, 1.0]$ is defined based on $min\|GW - \boldsymbol{\Lambda}\|$ as depicted in (11)-(12), where s is a constant to normalize to the GWS vector. The nature of $q = e^{-min\|s(GW-\boldsymbol{\Lambda})\|}$ is more sensitive for high-quality grasp candidates when the value of q nears 1.0 but less sensitive for low-quality grasp candidates, which is helpful for grasp predictions in the real world.

$$\boldsymbol{\Lambda} = \begin{bmatrix} \|\boldsymbol{F}_x\| \\ \|\boldsymbol{F}_y\| \\ max\|\boldsymbol{F}_z\| \\ \|\boldsymbol{T}_z\| \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ max(p) \cdot A^{\mathcal{G}} \\ 0 \end{bmatrix} \tag{10}$$

$$q = Q(\boldsymbol{P}_{\mathcal{O}}, \boldsymbol{P}_{\mathcal{G}}, S_{\mathcal{GO}}) = e^{-min\|s(GW-\boldsymbol{\Lambda})\|} \tag{11}$$

$$s = \frac{1}{A^{\mathcal{G}}} = \frac{1}{\pi r_{\mathcal{G}}^2} \tag{12}$$

Combing (8)-(12), the value of q is calculated by assigning appropriate weight values in the matrix W to minimize $\|GW - \boldsymbol{\Lambda}\|$. The minimization of $\|GW - \boldsymbol{\Lambda}\|$ is subjected to a series of physical restrictions in Section IV-E, which can be solved by Quadratic Programming (QP). The elaborate procedures of the QP are demonstrated in the Appendix.

G. Point Cloud Sampling

A sub point cloud $\mathcal{P}$ is demanded to record the grasp example when the grasp quality has been evaluated. As shown in Fig. 5 (g)-(i), a virtual camera $\mathcal{C}$ is deployed at the pose $\boldsymbol{P}_{\mathcal{C}}$ in WCS, then a set of virtual structured lights are projected towards the contact surface to render a point cloud in CCS. Finally, the point cloud $\mathcal{P}$ is transformed into GCS as the grasp observation. To improve the neural network learning efficiency [11], only a desampled 24×24 point cloud around the gripper is taken to record the grasp example, which is denoted as a 3×24×24 array with the 24×24 data from the x, y and z channels, separately.

H. Grasp Example Generation

In summary, the pseudo code to synthesize a grasp example is presented in Algorithm 1. Considering the average size of the 3D meshes, the virtual gripper is defined with $r_v = 20$ mm, $r_{\mathcal{G}} = 25$ mm, and $H = 30$ mm, which keeps the same shape but is smaller than a real gripper. The points set $\mathbb{V}$ is sampled by steps $\Delta_\theta = 0.045\pi$, $\Delta_r \leq 2$ mm, and $\Delta_d \leq 2$ mm. Both $\boldsymbol{P}_{\mathcal{O}}$ and $\boldsymbol{P}_{\mathcal{G}}$ are in SE(3) for each grasp. The grasp quality is evaluated by solving the QP within 100 iterations, and the sub point cloud is desampled into 24×24 points around the contract surface. Additionally, random noises with $\sigma_{\mathcal{C}} = 2$ mm

are implemented on sub point clouds to simulate the use of a physical depth camera. 35 high-resolution 3D meshes were selected from the YCB dataset, and their sizes were rescaled to simulate grasp scenarios with various objects. 1 K∼10 K valid grasps were randomly sampled for each 3D mesh, depending on the size of the mesh. Over 100 K grasps were synthesized, running 200 hours on the PC introduced in Section VII. When the gripper $\mathcal{G}$ cannot contact the object $\mathcal{O}$, a low-quality grasp will be synthesized with $q = e^{-min\|s(\mathbf{0}-\mathbf{\Lambda})\|} = e^{-1} = 0.368$ based on (11). Hence, the grasps with $q \leq 0.3$ are believed as extremely low-quality grasps and are not used for neural network training. Besides, all synthetic grasps are archived with an object-wise separation to ensure the objects used for the GA-CNN training are excluded from the testing dataset.

Algorithm 1 Synthesize a Grasp Example.

Assumptions:

A-1. Quasi-static physics with Coulomb friction.
A-2. The object has an airtight surface and a rigid body.
A-3. Ignore the weight of a target object and the torques on the x, y axes.

Input:

3D mesh of the object $\mathcal{O}$, virtual gripper $\mathcal{G}$, virtual camera $\mathcal{C}$.

Output:

Grasp quality q, point cloud $\mathcal{P}$.

Steps:

S-1.01: $\boldsymbol{P}_{\mathcal{O}}$ = RandomSet($\mathcal{O}$) in SE(3).
S-1.02: Build a geometric model for $\mathcal{G}$ in GCS by parameters $r_{\mathcal{G}}$, $\mathbb{V}$, $\mathbb{T}$...
S-1.03: Set geometric restrictions by r, θ, z, Δ_r, Δ_θ...
S-1.04: $\boldsymbol{c}$ = RandomPoint($\mathcal{O}$).
S-1.05: $\boldsymbol{d}_{\mathcal{G}}$ = RandomSet($\boldsymbol{c}, \mathcal{G}$).
S-1.06: $\boldsymbol{P}_{\mathcal{G}}$ = RandomSet($\boldsymbol{c}, \boldsymbol{d}_{\mathcal{G}}$) in SE(3).
for $k \leftarrow 1$ to m **do**:
S-1.07: Track $\mathbb{V} \leftarrow \{\boldsymbol{v}_1, \cdots, \boldsymbol{v}_k, \cdots, \boldsymbol{v}_m\}$.
end for
S-1.08: Update $\mathbb{T} = \{t_1, \cdots, t_j, \cdots, t_n\} \leftarrow \mathbb{V}$.
for $j \leftarrow 1$ to n **do**:
S-1.09: Compute $\boldsymbol{f}_x^{t_j}$, $\boldsymbol{f}_y^{t_j}$, $\boldsymbol{f}_z^{t_j}$ and $\boldsymbol{\tau}_z^{t_j}$.
end for
S-1.10: $G \leftarrow \{\boldsymbol{f}_x^{t_j}, \cdots, \boldsymbol{f}_y^{t_j}, \cdots, \boldsymbol{f}_z^{t_j}, \cdots, \boldsymbol{\tau}_z^{t_j}, \cdots\}$.
S-1.11: Set physical restrictions for μ, p, $\boldsymbol{F}_x$, $\boldsymbol{F}_y$, $\boldsymbol{F}_z$, $\boldsymbol{T}_z$...
S-1.12: Solve $q = e^{-min\|s(GW-\mathbf{\Lambda})\|}$ by QP.
S-1.13: $\boldsymbol{P}_{\mathcal{C}}$ = Set($\mathcal{C}$) in WCS.
S-1.14: Render $\mathcal{P}$ in CCS.
S-1.15: $\mathcal{P}$ = Desample(Disturb(Transform($\mathcal{P}$))).

V. GA-CNN

A. Architecture of GA-CNN

The GA-CNN can be considered as a function $\hat{q} = Q_\Theta(\mathcal{P})$ to learn grasp principles and to replace the function $q = Q(\boldsymbol{P}_{\mathcal{O}}, \boldsymbol{P}_{\mathcal{G}}, S_{\mathcal{GO}})$ for the quality prediction when a sub point cloud $\mathcal{P}$ is given during a physical grasping trial.

Unlike many grasp detection networks consisting of traditional shallow-layer CNNs [9], [25], the GA-CNN is defined by a deep CNN architecture illustrated in Fig. 6, which is a pyramid topology containing a 2-layer CNN and 4 Convolutional Block Attention Modules (CBAMs) [51] with the classic ResNet [52]. GA-CNN takes in a 24×24 array from the z-channel data of a point cloud and provides a quantitative value $\hat{q} > 0$ as output. The CBAMs in the GA-CNN are expected to concentrate more on *Reg.V* in Fig. 4 when evaluating the grasp quality.

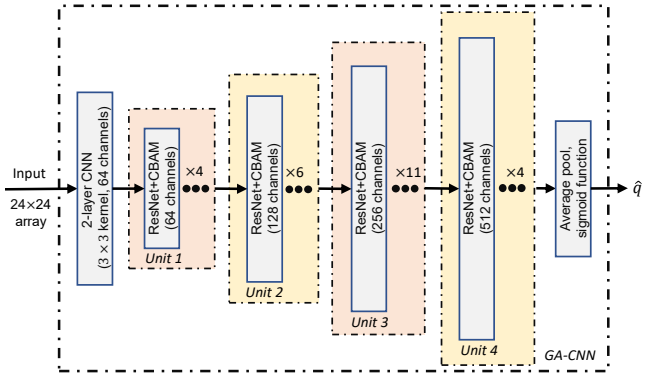

Fig. 6. The architecture of the proposed GA-CNN.

B. Training of GA-CNN

95 K synthetic grasps were selected to train the GA-CNN, and extra 5 K synthetic grasps were used for tests. Assuming a batch size of b is defined during the training, the Mean Squared Error (MSE) loss function is defined in (13) as the criterion to optimize the parameters using an Adam optimizer [53] with an initial learning rate of 0.000005.

$$L(q, \hat{q}) = \frac{1}{b}\sum_{i=1}^{b}\left(q_i - Q_\Theta(\mathcal{P}_i)\right)^2 \tag{13}$$

The PC introduced in Section VII was used for the GA-CNN training. More than 50 similar networks were trained to find the optimal architecture for the GA-CNN regarding both the prediction error and computational complexity. For each grasp example in the testing dataset, the prediction error of the GA-CNN is denoted as $|\hat{q} - q|_{abs}$. Fig. 7 presents the average prediction error and execution time affected by the jointly varying channels and depths of CBAMs in the GA-CNN. In detail, the depth of the GA-CNN varies from 18 layers to 102 layers with four different combinations of the CBAM channels.

With the increase of the depth and channels, the computational complexity of GA-CNNs consistently grows, but their prediction errors do not constantly decrease. Especially when a GA-CNN has more than 80 layers or 32-64-128-256 channels, the prediction error often enlarges due to overfitting. Consequently, the 52-layer GA-CNN with 64-128-256-512 channels keeps satisfying performance in the aspects of both the average error and computational complexity. The final GA-CNN is constructed by a 2-layer CNN with a 3×3 kernel, and 4 CBAMs with 64, 128, 256 and 512 channels respectively (Fig. 6), containing 3.23 M parameters. It reports an average

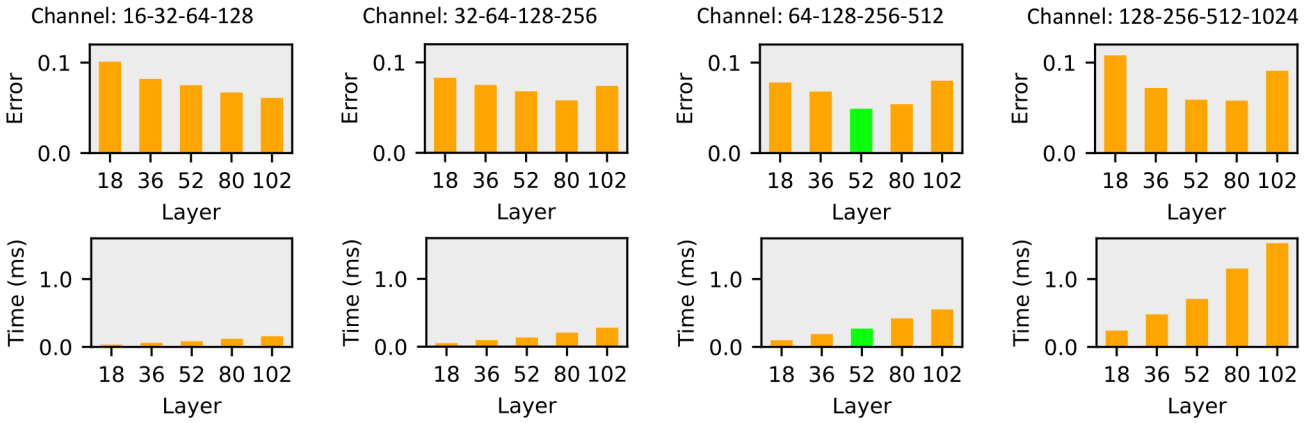

Fig. 7. Performance of the GA-CNN with different channels and depths of CBAMs. Note: Change the depths of CBAMs to get GA-CNNs with different layers, and green bars mark the selected architecture.

error of 0.049 and spends 0.27 ms for each synthetic grasp in the test dataset.

VI. Closed-loop Grasping

A closed-loop grasping method outperforms an open-loop one owing to its ability to utilize real-time robotic perceptions and adjust grasp strategies. The proposed reactive grasping method integrates the real-time point cloud of robotic observation and the 6-DoF force-torque wrench of the gripper base to develop the closed-loop control, as shown in Fig. 1.

The role of the real-time force-torque has been explained in our published work [14]. It is mainly utilized to monitor the grasp status, like collision and grasp success, optimize the robotic motion, and minimize the moments on the gripper base during grasping.

Different from the previous work [14], the real-time point cloud $\mathbb{P}$ is evaluated at each timestep in the closed-loop control. Besides, the proposed reactive grasping method is able to detect feasible grasp poses in SE(3). As mentioned in Section III-D, a number of sub point clouds $\mathbb{P} = \{\mathcal{P}_1, \mathcal{P}_2, \cdots, \mathcal{P}_i, \cdots, \mathcal{P}_u\}$ are randomly sampled and transformed into GCS for the grasp quality prediction. The grasp direction of $\mathcal{P}_i$ is calculated based on its average surface normal.

Furthermore, grasping objects from a dense clutter requires robust strategies to overcome visual occlusions. Multi-viewpoint grasping based on visual fusion improved the success rates for parallel-jaw grippers by 10.0% in previous research [39], [54]. Accordingly, multi-perspective grasping is developed for the proposed grasping method. The role of force-torque feedback keeps the same in both single-perspective and multi-perspective grasping, and the difference exists in using visual information with hybrid strategies for multi-perspective grasping. Specifically, the proposed multi-perspective grasping method merges more than one point cloud from multiple viewpoints, for example, the static and wrist-mounted cameras in Fig. 2, to acquire a global point cloud and detect a globally optimized grasp pose before grasping. Afterward, the robot adjusts the gripper pose towards the globally optimized grasp pose. Then the wrist-mounted camera keeps activated to observe the clutter in a local scope, optimize grasp pose and prune the grasp motion during grasping. The local scope is typically set with a rectangle region double the size of the applied gripper pad. A multi-viewpoint point cloud ensures that the robot can overcome visual occlusions and adapt the poses of the gripper and wrist-mounted camera before grasping. The use of a multi-perspective point cloud prevents the GA-CNN from missing a feasible grasp region that is invisible in a single-viewpoint point cloud before the robot adapts the poses of the gripper and wrist-mounted camera. The single-viewpoint point clouds during grasping ensure the proposed grasping method runs with 15 Hz real-time speed when the robot approaches a target object.

VII. Experiments

This section describes extensive experiments to evaluate the performance of the proposed grasping method both in simulation and on a physical robot. A computer running Ubuntu 20.04 OS was used in the experiments, which consists of a multi-kernel 3.5 GHz Intel Core i9-9920X CPU, 64 GB of dynamic system memory (DRAM), and two Nvidia GeForce RTX 2080Ti graphics cards.

A. Experiments on Simulation

This subsection describes experiments with synthetic grasps, aiming to verify the robustness of the GA-CNN and visualize the gripping attention of the GA-CNN in grasp quality prediction. The virtual gripper is defined by $r_v = 20$ mm, $r_\mathcal{G} = 25$ mm, and $H = 30$ mm in the subsequent tests.

1) Prediction Error: First, the robustness of the GA-CNN was validated by synthetic point clouds with different noise levels. As mentioned in Section IV-H, the grasps with $0.3 < q \leq 1.0$ were used for the GA-CNN training. Fig. 8 shows the grasp qualities of over 2,400 synthetic point clouds with random noises $\sigma_\mathcal{P} = 2$ mm, 4 mm and 8 mm, respectively. In this figure, the grasp qualities are ranked by ascending order based on their standard values q, then the predicted values $\hat{q}$ under different noise levels are separately fitted using polynomial regression. The coefficient of determination R^2

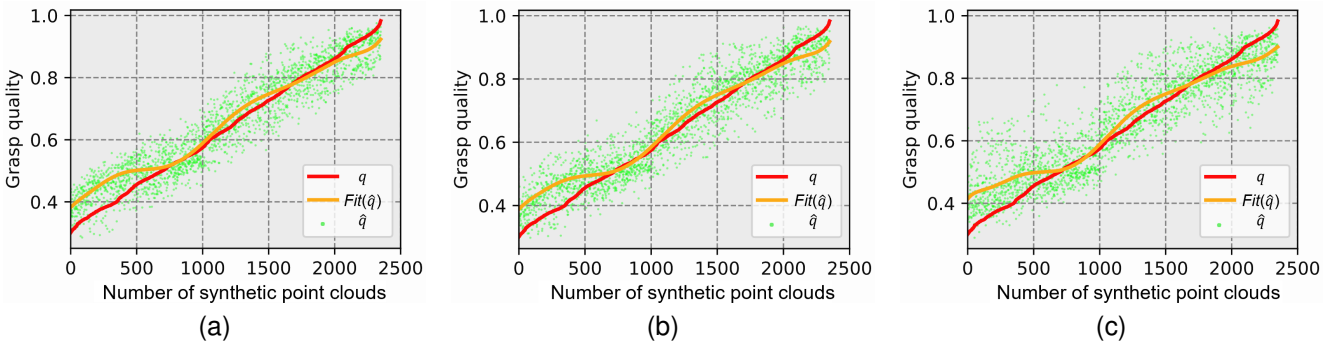

Fig. 8. Grasp quality predictions for synthetic point clouds under different noise levels. (a) $\sigma_{\mathcal{P}}$ =2 mm, R^2 =0.904. (b) $\sigma_{\mathcal{P}}$ = 4 mm, R^2 =0.893. (c) $\sigma_{\mathcal{P}}$ = 8 mm, R^2 =0.812.

was computed to indicate the divergence of $\hat{q}$ from q. Each fitted curve is monotonically increasing when $\sigma_{\mathcal{P}}$ is a constant, revealing that the GA-CNN predicts grasp qualities with low errors for synthetic point clouds. The GA-CNN achieves an average prediction error of ≤ 0.053 when $0.3 < q \leq 1.0$ and $\sigma_{\mathcal{P}} \leq 4$ mm. Thousands of tests in the simulation indicated that the grasp candidates with $0 < q \leq 0.3$ were not robust to conduct a successful grasp. These grasp candidates often contained large non-flat regions that the gripping pad cannot touch, which will never be selected as the final grasp pose in a physical grasping trial.

The robustness and applicability of the GA-CNN are further evaluated on a physical robot in Section VII-B.

2) Grasp Quality Visualization: Given a synthetic grasp scenario with a randomly-posed object and a pre-defined grasp pose, the gripping attention of the GA-CNN was compared with a traditional CNN having a similar architecture and the same number of parameters. Fig. 9 (a)-(c) show three synthetic grasp scenarios with the grasp centers $\boldsymbol{c}$, including two adversarial items from the Dex-Net dataset in Fig. 9 (b)-(c). Fig. 9 (d)-(f) and Fig. 9 (g)-(i) respectively present the gripping attention of the CNN and GA-CNN via the Class Activation Map (CAM) [55], wherein the regions with warm color own larger weight values and more substantial attention. The CAMs of the GA-CNN in Fig. 9 (g)-(i) focus on *Reg.V* more than those of the CNN in Fig. 9 (d)-(f). Significantly, *Reg.V* can deform to fit non-flat contact surfaces S_{con} that cannot entirely match the gripping pad, like the grasp scenes in Fig. 9 (b)–(c). In such cases, the *Reg.V* on a CAM is no longer a round region as in Fig. 9 (g), and thus the grasp quality prediction becomes challenging for the CNN, as demonstrated in Fig. 9 (e)-(f). In contrast, the CAMs of the GA-CNN still pay more attention to the *Reg.V* located on S_{con}. Furthermore, the predicted values $\hat{q}$ via the GA-CNN are more accurate, since CBAMs can find the crucial gripping region and extract its features for the grasp quality prediction. The feature refinement of CBAMs eventually leads the GA-CNN to utilize features better than the CNN.

The performance of the GA-CNN was further investigated with the grasp scenario in Fig. 9 (c), assuming the grasp direction is fixed. Fig. 10 shows the 2D quality maps of the target object, where each 2D quality map was generated by the

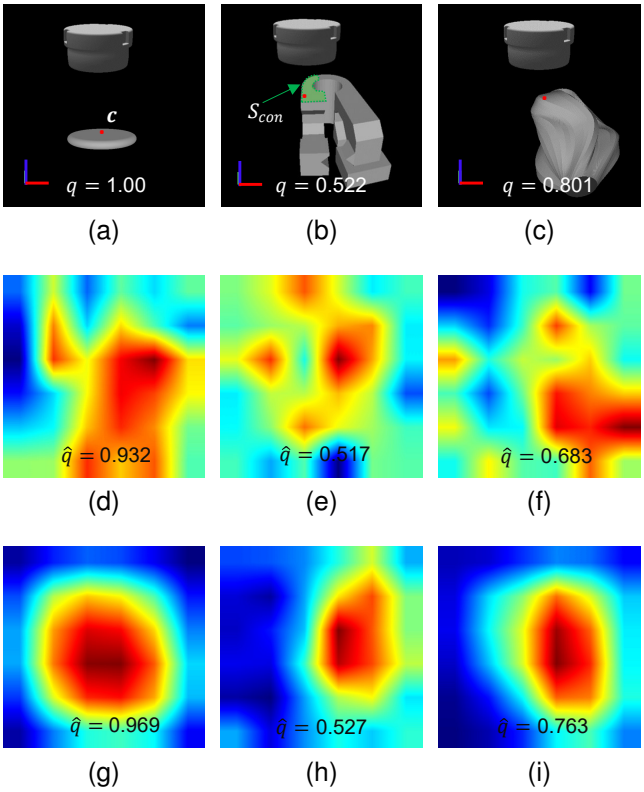

Fig. 9. Gripping attention of three synthetic grasp scenarios. (a) Grasping a disk with a radius of 25 mm. (b) Grasping a complex object with a dimension of 110 mm×75 mm×54 mm. (c) Grasping a complex object with a dimension of 105 mm×80 mm×73 mm. (d)-(f) CAMs of the CNN. (g)-(i) CAMs of the GA-CNN.

pixels $q(x, y)$ or $\hat{q}(x, y)$ on the object surface. A 2D Gaussian filter with a 5×5 kernel was applied to smooth the 2D quality maps in Fig. 10.

The quality map $\mathbb{Q}$ in Fig. 10 (a) acquired from the grasp simulation (Section IV) is believed as a baseline and compared with the quality maps predicted via the CNN and GA-CNN. In Fig. 10 (a), the high-quality region is often located on the convex surface of the target object, where an airtight contact can be conducted. Fig. 10 (b) visualizes the predicted quality map using the CNN, named $\hat{\mathbb{Q}}_{\text{CNN}}$. Briefly, inaccurate prediction is reported on the region of $x \subseteq [-2 \text{ cm}, 2 \text{ cm}] \cap y \subseteq$

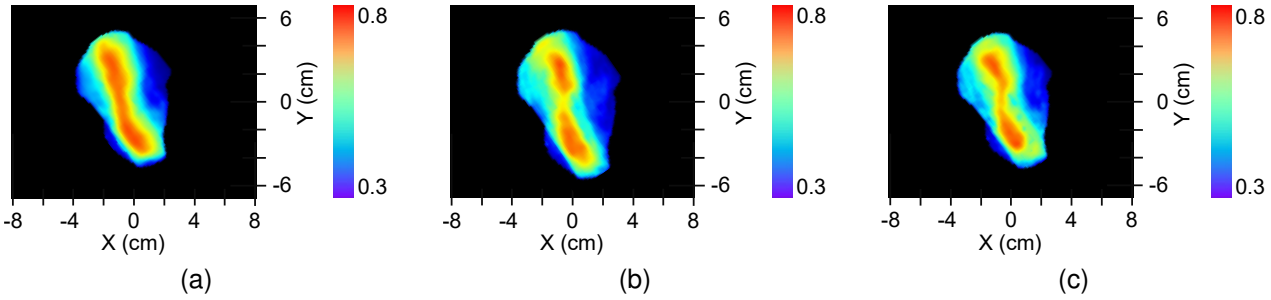

Fig. 10. 2D Grasp quality maps for the synthetic grasp in Fig. 9 (c). (a) Baseline 2D quality map $\mathbb{Q}$ acquired from the grasp simulation. (b) 2D quality map based on the CNN, named $\hat{\mathbb{Q}}_{\text{CNN}}$. (c) 2D quality map based on the GA-CNN, named $\hat{\mathbb{Q}}_{\text{GA-CNN}}$.

$[-1\text{ cm}, 1\text{ cm}]$ in $\hat{\mathbb{Q}}_{\text{CNN}}$, and the average prediction error of $\hat{\mathbb{Q}}_{\text{CNN}}$ is 0.065 compared with $\mathbb{Q}$, which is calculated by $|\hat{q}-q|_{abs}$ for all foreground pixels in $\mathbb{Q}$ and $\hat{\mathbb{Q}}_{\text{CNN}}$. In contrast, the predicted quality map of the GA-CNN, named $\hat{\mathbb{Q}}_{\text{GA-CNN}}$, performs better and is more similar to the baseline map $\mathbb{Q}$, although it often overestimates low-quality regions, such as the regions of $x \subseteq [-4\text{ cm}, -2\text{ cm}] \cap y \subseteq [0\text{ cm}, 2\text{ cm}]$ and $x \subseteq [1\text{ cm}, 3\text{ cm}] \cap y \subseteq [2\text{ cm}, 4\text{ cm}]$ in $\hat{\mathbb{Q}}_{\text{GA-CNN}}$. Finally, $\hat{\mathbb{Q}}_{\text{GA-CNN}}$ reports an average error of 0.041 based on the same metric $|\hat{q}-q|_{abs}$.

Notably, the grasping trials in both Fig. 9 and Fig. 10 were conducted with a top-down grasping assumption in SE(2) for randomly-posed objects, which are insufficient to indicate the GA-CNN is competent for the real-world grasp quality estimation in SE(3). Hence, the GA-CNN grasping method has been thoroughly assessed in the subsequent real-world experiments in Section VII-B.

B. Experiments on Physical Robotic Grasping

This subsection presents the performance of the GA-CNN with physical setups for static grasping, dynamic grasping and multi-perspective grasping. The robotic system is depicted in Fig. 11. It is composed of a 6-DoF robot (KUKA LBR IIWA 14 R820), a versatile vacuum gripper with a radius of 75 mm (FORMHAND FH-R150) that is much larger than the virtual gripper in simulation, a static camera (Microsoft Kinect Version 2), a wrist-mounted camera (Intel RealSense L515) and a PC, linked via ROS nodes [56]. Notably, the force-torque wrench in the closed-loop control is projected from the joint torque sensors on the KUKA IIWA, and necessary configurations and calibration are needed for the robot regarding the weights of the used gripper, wrist-mounted camera and flange.

1) Clutter Levels: Despite the fact that abundant benchmark objects are available online for robotic grasping tests, many items from the benchmarks are not always suitable for the investigated grippers in the aspects of sizes, weights and rigidity. A fair criterion is needed to benchmark clutters in physical grasping. In this paper, the complexities of grasp scenes are defined with nine-level metrics for the random picking with vacuum grippers.

In this research, the complexity of the object's shape is defined based on the *NSVR* in (14), where A_{sur} and V_{obj} are the surface area and volume of the object. Definitely, it is

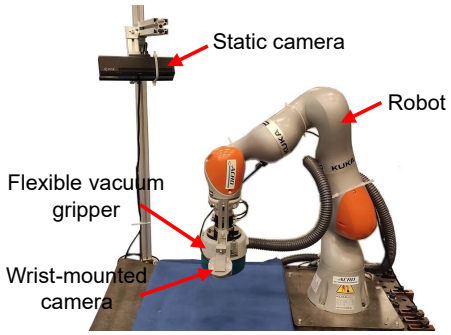

Fig. 11. Robotic setup for the experiments.

a property of the object and not related to any gripper. The *NSVR* keeps the same when an object is rescaled.

$$NSVR = \frac{\sqrt{A_{sur}}}{\sqrt[3]{V_{obj}}} \tag{14}$$

Fig. 12 lists some objects and their *NSVR*s from the Dex-Net [12], KIT [48] and YCB [47] grasping databases. The *NSVR*s of more than one hundred 3D meshes from these databases were calculated, and the complexities of objects can be divided into three categories based on *NSVR*s: Basic ($0 < NSVR < 2.6$), Typical ($2.6 \leq NSVR < 3.5$), and Complex ($3.5 \leq NSVR$). With these criteria, any real-world object can be classified based on the *NSVR* of its reconstructed 3D model as exhibited in Fig. 13 (a)-(c).

Moreover, the distribution of objects in a grasp scene can be classified into three levels: isolated, multiple and stacked, as shown in Fig. 13 (d)-(f). In a grasp scene with isolated objects, the objects are manually deployed with random poses on the table in SE(2), and the minimum gap between each object and its neighbors is not smaller than 5 cm. Similarly, a grasp scene with multiple objects is manually deployed on the table in SE(2), where the maximum gap between each object and its neighbors is within 1 cm. As a comparison, stacked objects in a grasp scene stay with random poses in SE(3) and touch each other, which can be deployed following the method in the existing work [9].

Therefore, the complexities of grasping clutters are divided into nine levels in Table I. Fig. 13 (d)-(f) illustrate part of the items for the subsequent physical experiments, including 5 adversarial objects from the Dex-Net dataset. In the subsequent

TABLE I
THE COMPLEXITY LEVELS OF GRASPING CLUTTERS.

Level \ Dist. / Comp.	Isolated	Multiple	Stacked
Basic	1	2	3
Typical	4	5	6
Complex	7	8	9

Note: 1) Comp. is the abbreviation of "Complexity." 2) Dist. is the abbreviation of "Distribution."

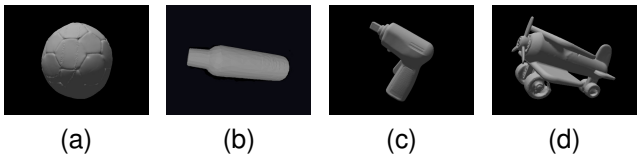

Fig. 12. Several 3D models from benchmark datasets and their *NSVR*s. (a) *NSVR* = 2.22. (b) *NSVR* = 2.42. (c) *NSVR* = 2.63. (d) *NSVR* = 3.87.

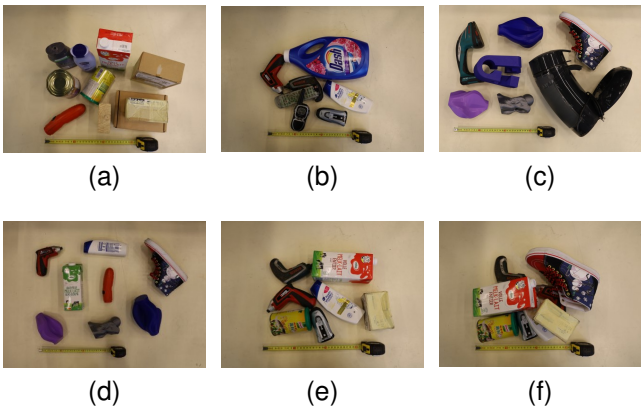

Fig. 13. Examples of objects' complexities and distributions. (a) A set of basic objects. (b) A set of typical objects. (c) A set of complex objects. (d) A set of isolated objects. (e) Multiple objects on the table in SE(2). (f) A set of stacked objects.

physical experiments, every clutter consisted of 10 randomly distributed objects, and five clutters were deployed for the test at each level (50 objects in total).

2) Static Grasping: The real-world grasping experiments start with the pre-analysis of grasp scenes with isolated objects. For instance, Fig. 14 (a) briefly demonstrates a grasping trial with the GA-CNN grasping method in SE(3). 5 K grasp candidates were randomly selected from the point cloud to evaluate the grasp robustness. The corresponding grasp directions estimated by the average surface normal and 3D quality map predicted via the GA-CNN are shown in Fig. 14 (b)-(c). A similar grasping trial is separately presented in Fig. 14 (d)-(f). As a conclusion, the high-quality grasp regions on the quality maps almost correspond with relatively wide, flat surfaces on the real-world objects in Fig. 14 (a) and (d), which are consistent with human intuition and experience.

To evaluate the performance of the GA-CNN for random picking in static scenes, benchmark grasping trials were formulated on static clutters with the following six baseline methods: 1) a grasp pose on the clutter's center in SE(3), 2) a minor-revised Dex-Net 3.0 in SE(3) [57], 3) the CNN

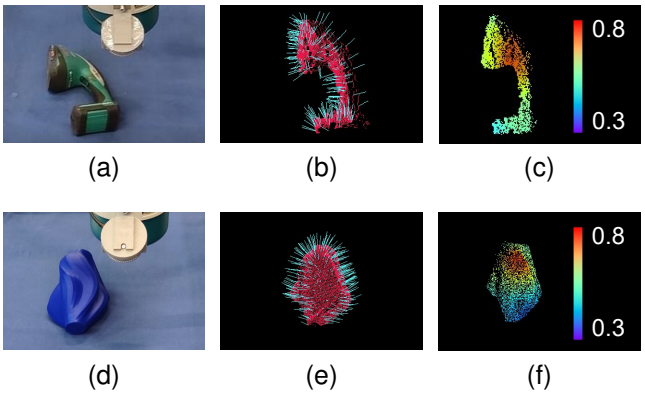

Fig. 14. Two physical grasping trials in SE(3). (a), (d) Target objects. (b), (e) Part of grasp directions estimated by the average surface normal. (c), (f) 3D grasp quality maps predicted via the GA-CNN.

grasping method in SE(2) with a top-down grasp direction, as mentioned in Section VII-A, 4) the CNN grasping method in SE(3), 5) the GA-CNN grasping method in SE(2) with a top-down grasp direction, and 6) the GA-CNN grasping method in SE(3). Only the static camera in Fig. 11 was used for the static grasping. Besides, a collision-avoidance algorithm [14] was integrated into the benchmark grasping methods to avoid potential collisions on the robotic moving path.

Table II presents the success rates of the six grasping methods above. Apparently, most grasping methods achieve high success rates at Level 1 to Level 4, proving that the versatile vacuum gripper can fit objects well with the soft gripping pad, and the grasping methods are not very important at those levels. The traditional grasping method based on clutter's center reports low success rates for the clutters at Level 5 to Level 9. Because a feasible grasp pose for a complex object or dense clutter is not always located at its geometric center. The minor-revised Dex-Net 3.0 shows low success rates for the objects in clutters, especially at Levels 6, 8 and 9. Also, the minor-revised Dex-Net 3.0 is not good at grasping relatively small objects. Because the Dex-Net 3.0 is developed for the typical use cases of basic vacuum grippers and trained on a dataset containing plenty of grasp scenarios with relatively large objects. Given a small and complex object, the Dex-Net 3.0 could not find a feasible grasp region that fits the soft gripping pad of the adopted gripper and covers the crucial region *Reg.V*. It is concluded that the Dex-Net 3.0 is a robust grasping method for a basic vacuum gripper, but it is not competent for the grasp planning with a versatile vacuum gripper. The traditional CNN trained on our synthetic dataset performs an average success rate of 93.5% at Level 1-Level 4. Nonetheless, the performance shows a relevant decrease at Level 5 to Level 9, revealing that the traditional CNN has difficulties to learn the grasp principles for a versatile vacuum gripper. As a comparison, the GA-CNN can learn the grasp principles of the adopted gripper and detect feasible grasp poses that fit the soft gripping pad and wrap *Reg.V* as much as possible for various objects. The GA-CNN grasping in SE(3) consistently works better than others and achieves an average success rate of 90.2% at Level 1 to Level 8 for the grasping

TABLE II
RESULTS OF THE BENCHMARK EXPERIMENTS FOR STATIC GRASPING.

S.R. Level / Method	1	2	3	4	5	6	7	8	9
Center	100.0	64.1	46.3	83.3	-	-	59.5	-	-
Dex-Net 3.0	100.0	90.9	80.6	84.7	67.6	51.0	66.7	48.1	-
CNN, SE(2)	100.0	92.6	82.0	92.6	72.5	60.2	73.5	59.5	48.5
CNN, SE(3)	100.0	96.2	83.3	94.3	73.5	61.7	76.9	61.7	49.0
GA-CNN, SE(2)	100.0	100.0	94.3	96.2	84.7	71.4	90.9	74.6	63.3
GA-CNN, SE(3)	100.0	100.0	96.2	96.2	86.2	73.5	92.6	76.9	64.1

Note: 1) *S.R.* is the abbreviation of "Success Rate." 2) Results marked with "-" mean that the corresponding success rates are <33.3%.

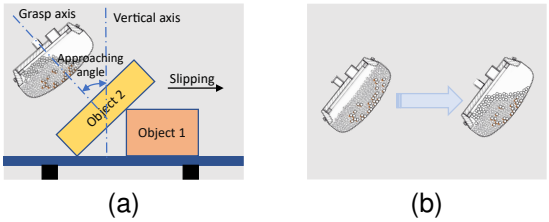

Fig. 15. Limitations of the grasping in SE(3). (a) Slipping of the target objects. (b) Deformation of the granulate filling in the gripper cushion.

in SE(3), which is significantly higher than other methods.

In addition, the grasping methods in SE(3) with the CNN and GA-CNN do not considerably outperform the grasping in SE(2), attributed to the limitations of the used setup. On the one hand, the approaching angle of the gripper in Fig. 15 (a) is often limited by the ranges of the robotic joints, friction on the table, and so on. A large approaching angle could result in the slipping of target objects. Also, the granulate filling inside the gripper cushion will move due to the gravity and be accumulated at the bottom of the cushion before grasping, and thus the soft gripping pad and flexible cushion cannot fit the contact surface well, as shown in Fig. 15 (b). On the other hand, if the approaching angle of the gripper is too small, the advantage of a grasp pose in SE(3) will not be apparent, compared with a top-down grasp pose in SE(2).

3) Dynamic Grasping: Grasping moving objects is a typical application for a robot. Closed-loop control is demanded to monitor the grasp status for dynamic grasping. Morrison *et al.* [39] measured the offset for a moving clutter by the grids on the table. However, it is not easy to keep the same moving speed and offset for clutters in several tests. In this research, the clutters stayed with static poses and the moving objects were simulated by the random disturbance of robotic motion. The GA-CNN method in SE(2) was implemented for dynamic grasping to simplify the experiments, as Table II has concluded that the GA-CNN method in SE(3) cannot significantly improve the grasping performance. Given a desired robotic pose $\boldsymbol{P}_{\mathcal{R}} = [x, y, z, \alpha, \beta, \gamma]^T$ to grasp an object, an random offset $\boldsymbol{P}_{\sigma} = [x_{\sigma}, y_{\sigma}, 0, 0, 0, 0]^T$ was added to $\boldsymbol{P}_{\mathcal{R}}$ at each timestep in the closed-loop control to simulate the moving objects, as formulated in (15).

$$\boldsymbol{P}'_{\mathcal{R}} = \boldsymbol{P}_{\mathcal{R}} + \boldsymbol{P}_{\sigma}, \quad ||\boldsymbol{P}_{\sigma}|| = \sqrt{x_{\sigma}^2 + y_{\sigma}^2} \tag{15}$$

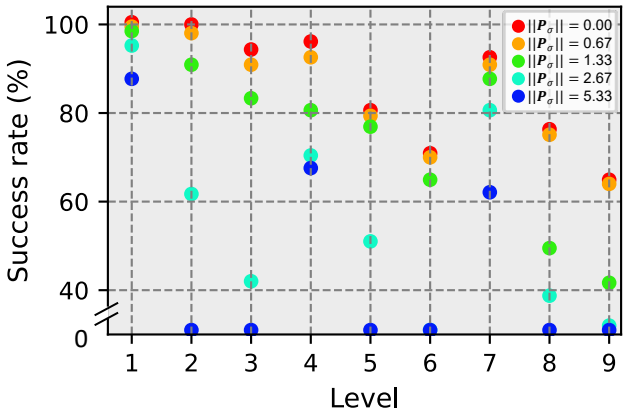

Fig. 16. Success rates of the GA-CNN grasping method for dynamic grasping in SE(2).

The dynamic grasping was conducted at Level 1 to Level 9 with a robotic speed of $v_{\mathcal{R}} = 250$ mm/s, and $||\boldsymbol{P}_{\sigma}|| =$ 0.00 mm, 0.67 mm, 1.33 mm, 2.67 mm and 5.33 mm at each control loop, and only the wrist-mounted camera in Fig. 11 was activated for the subsequent tests. As illustrated in Fig. 16, the grasp success rates do not dramatically reduce compared with the static grasping at the same levels when $||\boldsymbol{P}_{\sigma}|| \leq 1.33$ mm, especially for the clutters at Level 1 to Level 7. Hence, the 15 Hz closed-loop control enables the GA-CNN grasping method to track and pick up moving objects. However, the GA-CNN grasping method performs lower and lower success rates with the increase of $||\boldsymbol{P}_{\sigma}||$ and hardly works when $||\boldsymbol{P}_{\sigma}|| \geq$ 5.33 mm.

4) Multi-Perspective Grasping: This subsection conducts static grasping of the GA-CNN method with multi-perspective robotic observations, taking full advantage of the static and wrist-mounted cameras, as mentioned in Section VI.

Fig. 17 compares the success rates of the GA-CNN grasping method in SE(3) for static objects in single-perspective and multi-perspective robotic observations. The multi-perspective observation remarkably improves the GA-CNN performance for the random picking at Levels 5, 6, 8 and 9, by 4.7%, 9.8%, 3.7% and 4.4% respectively. In other words, a multi-viewpoint observation becomes necessary with the increasing complexity of grasping clutters. Because the point cloud acquired from multiple perspectives contains surface features in both the top and side of a clutter, which cannot be acquired merely from a single viewpoint. Hence, the GA-CNN grasping method can

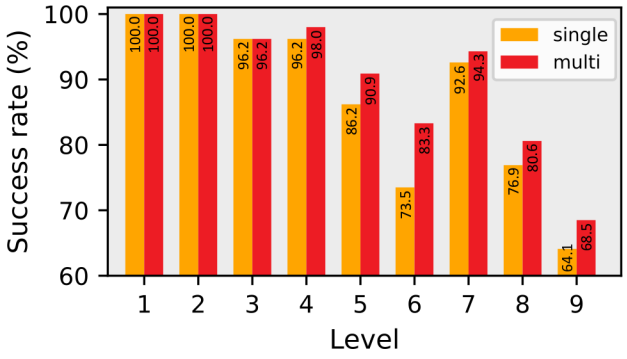

Fig. 17. Success rates of the GA-CNN grasping in SE(3) for static objects in single-perspective and multi-perspective robotic observations. Note: The coordinates of the y axis start from 60.0% to visualize the difference in the success rates.

detect a better grasp candidate based on more features from a multi-perspective point cloud.

C. Failed Trials and Limitations

There are four typical failure grasp cases about the proposed grasping method.

First, Fig. 18 (a) reports a failed grasp for a transparent object, since the structured light of the depth camera is unable to be reflected on a transparent surface, and the robot fails to detect the object.

Second, Fig. 18 (b) shows a failed grasp for two neighbor objects with the same height. The flat boundary cannot be detected due to the limited resolution of the depth camera. Implementing a segmentation algorithm before the grasp quality prediction is a potential solution for this problem. However, the segmentation [58] and tracking [59] of unknown objects in a clutter are still complicated.

Moreover, the proposed grasping method has mediocre performance for the random picking of slim objects, as illustrated in Fig. 18 (c)-(d). Specifically, while the GA-CNN grasping method detects the feasible grasp regions for both grasping trials, a failed grasp is still reported in Fig. 18 (d) when two slim objects are near-distributed and the neighbor object inhibits the soft gripping pad from fitting the target object. The grasp success is affected by the dimensions of objects and gripper.

Additionally, the closed-loop GA-CNN grasping merely runs within 15 Hz on the used PC due to the deep architecture of the GA-CNN. The decrease in grasping performance can be seen if objects move too fast. This is the reason why the success rates are pretty low for the dynamic grasping with $||\boldsymbol{P}_\sigma|| \geq 2.67$ mm presented in Fig. 16.

In consequence, the applicability of the proposed closed-loop grasping method is also related to physical setups and target objects. Essential adjustments are needed to fit different use cases. For instance, an appropriate gripper's size should be decided regarding the objects' sizes in physical grasping trials.

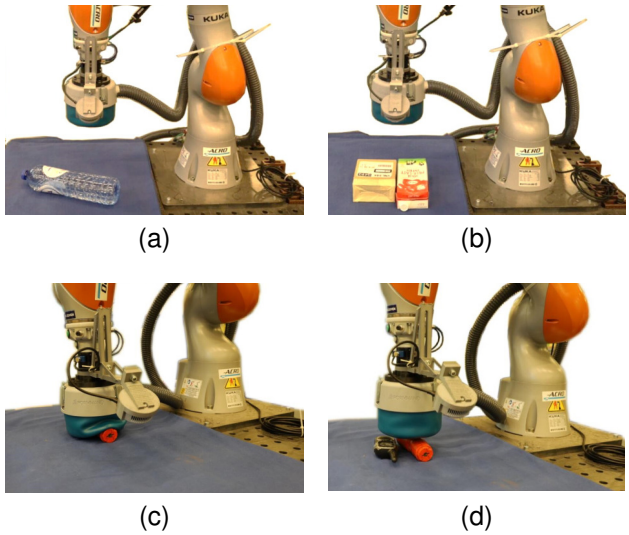

Fig. 18. Some failed grasping trials. (a) Grasping trial for a transparent object. (b) Grasping trial for two flat objects with a similar height. (c) Successful grasping trial for a single and slim object. (d) Failed grasping trial for two near-distributed and slim objects.

VIII. Conclusion

In this paper, the authors present a flexible grasping method for the random picking of unknown objects. A 6-step approach is proposed to simulate the grasp and evaluate the grasp quality for a versatile vacuum gripper, which tracks the deformation and force-torque wrench of the gripping pad and computes the grasp quality under quasi-static conditions. Over 100 K synthetic grasps are generated for neural network training. A 52-layer GA-CNN is proposed to learn the grasp principles for the versatile gripper, combing the CBAMs and traditional ResNets. The proposed closed-loop GA-CNN grasping method is a 15 Hz reactive grasping approach and detects grasp poses in SE(3), which reports satisfying performance in both the simulation and physical grasping trials. Additionally, failure modes of the GA-CNN grasping method are discussed.

Future research is scheduled in two aspects. First, a more versatile grasping simulator will be developed to model the object's weight in the grasping wrench. The novel simulator will record the force-torque feedback for each virtual grasp. A neural network can be trained on the sequence of force-torque feedback to detect the grasp status and optimize the grasping performance, which could be based on an architecture of RNN, LSTM [43] or Transformer. The main challenge exists in the weight estimation, regarding a real-world object and its 3D mesh. Second, the metric of objects' distribution in a grasp scene will be further improved for benchmark tests. For example, a random grasp scene with isolated, multiple or stacked objects can be synthesized in a simulator [60] with the reconstructed 3D meshes of real-world objects, and then a similar real-world grasp scene can be deployed with the same objects from the synthetic grasp scene. This method succeeds on some datasets of 3D models but remains challenging on real-world data.

ACKNOWLEDGMENTS

The authors would like to give special thanks to Prof. Aiguo Song, Prof. Lifeng Zhu and Dr. Qiang Chen at Southeast University (Nanjing, China), and Dr. Jie Hu at China University of Geosciences (Wuhan, China) for the suggestions on the grasp simulation.

APPENDIX
QP FOR GRASP QUALITY ESTIMATION

To calculate the force-torque wrench in Fig. 4 (d), a set of orthogonal normal is formulated in (16). Let $p > 0$ be the air-pressure differential between the gripping pad and the atmosphere, and $\mu > 0$ be the coefficient of friction between the gripping pad and the target object. In this paper, the coefficient of friction is set with a value of $\mu = 0.5$, which is a typical constant for the contact of a rubber gripping pad and unknown objects [12], [25], [57]. Nevertheless, how to estimate the air-pressure differential p in the grasp simulation remains challenging. The value of p in the real world depends on the operating curve of the fan type, the geometry of the tubing, the power of pump and so on, which is a computationally expensive problem in the simulation involving Computational Fluid Dynamics (CFD). Hence, the value of p is estimated by the monotonically increasing function in (17).

Given a triangle t_j, $\boldsymbol{r}_z^{t_j}$ is the moment arm for the geometric center of t_j towards the z axis, $(n_x^{t_j}, n_y^{t_j}, n_z^{t_j})$ is the surface normal of t_j, and A^{t_j} is the area of t_j. Then the sub force-torque wrench is computed by (18)-(21).

$$\boldsymbol{n}_x = (1,0,0), \quad \boldsymbol{n}_y = (0,1,0), \quad \boldsymbol{n}_z = (0,0,1) \tag{16}$$

$$p = (\frac{\sum A^{t_j}}{A^{Reg.V}})^2, t_j \in Reg.V, p \subseteq [0,1] \tag{17}$$

$$\boldsymbol{f}_x^{t_j} = (pA^{t_j} n_x^{t_j} + \mu p A^{t_j} n_x^{t_j}) \boldsymbol{n}_x \tag{18}$$

$$\boldsymbol{f}_y^{t_j} = (pA^{t_j} n_y^{t_j} + \mu p A^{t_j} n_y^{t_j}) \boldsymbol{n}_y \tag{19}$$

$$\boldsymbol{f}_z^{t_j} = (pA^{t_j} n_z^{t_j}) \boldsymbol{n}_z \tag{20}$$

$$\boldsymbol{\tau}_z^{t_j} = \boldsymbol{r}_z^{t_j} \times (\boldsymbol{f}_x^{t_j} + \boldsymbol{f}_y^{t_j}) \tag{21}$$

$$\boldsymbol{F}_v = pA^{\mathcal{G}} \boldsymbol{n}_z = \sum_{j=1}^{n} pA^{t_j} \boldsymbol{n}_z \tag{22}$$

$$|\boldsymbol{F}_x \cdot \boldsymbol{n}_x| = |\sum_{j=1}^{n} w_j \boldsymbol{f}_x^{t_j} \boldsymbol{n}_x| \leq \frac{\sqrt{3}}{3} \mu \boldsymbol{F}_v \cdot \boldsymbol{n}_z \tag{23}$$

$$|\boldsymbol{F}_y \cdot \boldsymbol{n}_y| = |\sum_{j=1}^{n} w_j \boldsymbol{f}_y^{t_j} \boldsymbol{n}_y| \leq \frac{\sqrt{3}}{3} \mu \boldsymbol{F}_v \cdot \boldsymbol{n}_z \tag{24}$$

$$0 \leq \boldsymbol{F}_z \cdot \boldsymbol{n}_z = \sum_{j=1}^{n} w_j \boldsymbol{f}_z^{t_j} \boldsymbol{n}_z \leq \boldsymbol{F}_v \cdot \boldsymbol{n}_z \tag{25}$$

$$|\boldsymbol{T}_z \cdot \boldsymbol{n}_z| = |\sum_{j=1}^{n} w_j \boldsymbol{\tau}_z^{t_j} \boldsymbol{n}_z| \leq \frac{\sqrt{3}}{3} \mu r_{\mathcal{G}} \boldsymbol{F}_v \cdot \boldsymbol{n}_z \tag{26}$$

$$\sum_{j=1}^{n} w_j = n \tag{27}$$

$$1 - \sigma_W \leq w_j \leq 1 + \sigma_W, \quad \sigma_W = 0.1 \tag{28}$$

$$LW \leq l, \quad JW = n \tag{29}$$

A force-torque wrench for a soft contact model is restricted by the elliptical equation [28]. Therefore, the limitations of $[\|\boldsymbol{F}_x\|, \|\boldsymbol{F}_y\|, \|\boldsymbol{F}_z\|, \|\boldsymbol{T}_z\|]^T$ can be calculated by (22)-(26), where $\boldsymbol{F}_v$ is the vacuum force on the whole gripping pad.

Furthermore, the weight value w_j of each sub gripping pad is also limited by the physical properties of the used gripper. Specifically, the air-pressure differential p on the contact surface is generated from a unique air-flow tube on the gripper base, and thus differential of weight values w_j cannot be too large, which can be briefly restricted in (27)-(28).

Given a weight matrix $W = [w_1, w_2, \cdots, w_j, \cdots, w_n]^T$, all restrictions in (22)-(28) can be combined and converted into a formula in (29), wherein the restriction matrices $L \in \mathbb{R}^{(8+n)\times n}$ and $l \in \mathbb{R}^{(8+n)\times 1}$ are converted from (22)-(26) and (28), and $J \in \mathbb{R}^{1\times n}$ is an all-ones matrix to reformulate the constraint in (27). In other words, the physical restrictions of $[\|\boldsymbol{F}_x\|, \|\boldsymbol{F}_y\|, \|\boldsymbol{F}_z\|, \|\boldsymbol{T}_z\|]^T$ are converted into the limitations of W.

According to (11)-(12) in Section IV-F, it is derived that $q = e^{-min\|s(GW-\boldsymbol{\Lambda})\|} \propto -min\|GW - \boldsymbol{\Lambda}\| \propto -min\|GW - \boldsymbol{\Lambda}\|^2 \propto -min(0.5W^T GW - \boldsymbol{\Lambda}^T GW)$. Therefore, the grasp quality estimation in this paper can be seen as the minimization of $0.5W^T GW - \boldsymbol{\Lambda}^T GW$ subjected to the conditions in (29), which is solved by QP.

REFERENCES

[1] D. Reznik and V. Lumelsky, "Multi-finger "hugging": a robust approach to sensor-based grasp planning," in *Proc. IEEE Int. Conf. Robot. Autom.*, vol. 1, May 1994, pp. 754–759.

[2] J. D. Crisman, C. Kanojia, and I. Zeid, "Graspar: a flexible, easily controllable robotic hand," *IEEE Robot. Autom. Mag.*, vol. 3, no. 2, pp. 32–38, Jun. 1996.

[3] M. Zhu, Y. Mori, T. Wakayama, A. Wada, and S. Kawamura, "A fully multi-material three-dimensional printed soft gripper with variable stiffness for robust grasping," *Soft Robot.*, vol. 6, no. 4, pp. 507–519, Aug. 2019.

[4] Q. Hu, E. Dong, and D. Sun, "Soft gripper design based on the integration of flat dry adhesive, soft actuator, and microspine," *IEEE Trans. Robot.*, vol. 37, no. 4, pp. 1065–1080, Aug. 2021.

[5] A. T. Miller, S. Knoop, H. I. Christensen, and P. K. Allen, "Automatic grasp planning using shape primitives," in *Proc. IEEE Int. Conf. Robot. Autom.*, vol. 2, Sep. 2003, pp. 1824–1829.

[6] L. Jiang, K. Low, J. Costa, R. J. Black, and Y.-L. Park, "Fiber optically sensorized multi-fingered robotic hand," in *Proc. IEEE Int. Conf. Robot. Autom.*, Sep. 2015, pp. 1763–1768.

[7] S. Jadhav, M. R. A. Majit, B. Shih, J. P. Schulze, and M. T. Tolley, "Variable stiffness devices using fiber jamming for application in soft robotics and wearable haptics," *Soft Robot.*, Feb. 2021.

[8] J. Mahler, M. Matl, V. Satish, M. Danielczuk, B. DeRose, S. McKinley, and K. Goldberg, "Learning ambidextrous robot grasping policies," *Sci. Robot.*, vol. 4, no. 26, Jan. 2019.

[9] A. ten Pas, M. Gualtieri, K. Saenko, and R. Platt, "Grasp pose detection in point clouds," *Int. J. Robot. Res.*, vol. 36, no. 13-14, pp. 1455–1473, Oct. 2017.

[10] J. Leitner, A. W. Tow, N. Sunderhauf, J. E. Dean, J. W. Durham, M. Cooper, M. Eich, C. Lehnert, R. Mangels, C. McCool, P. T. Kujala, L. Nicholson, T. Pham, J. Sergeant, L. Wu, F. Zhang, B. Upcroft, and P. Corke, "The ACRV picking benchmark: a robotic shelf picking benchmark to foster reproducible research," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2017, pp. 4705–4712.

[11] I. Lenz, H. Lee, and A. Saxena, "Deep learning for detecting robotic grasps," *Int. J. Robot. Res.*, vol. 34, no. 4-5, pp. 705–724, Mar. 2015.

[12] J. Mahler, J. Liang, S. Niyaz, M. Laskey, R. Doan, X. Liu, J. A. Ojea, and K. Goldberg, "Dex-net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics," in *Robot.: Scien. Sys.*, Jul. 2017.

[13] A. Depierre, E. Dellandrea, and L. Chen, "Jacquard: a large scale dataset for robotic grasp detection," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, Oct. 2018, pp. 3511–3516.

[14] H. Zhang, J. Peeters, E. Demeester, and K. Kellens, "A CNN-based grasp planning method for random picking of unknown objects with a vacuum gripper," *J. Intell. Robot. Syst.*, vol. 103, no. 64, pp. 1–19, Nov. 2021.

[15] N. Yamanobe and K. Nagata, "Grasp planning for everyday objects based on primitive shape representation for parallel jaw grippers," in *Proc. IEEE Int. Conf. Robot. Biomim.*, Dec. 2010, pp. 1565–1570.

[16] A. Herzog, P. Pastor, M. Kalakrishnan, L. Righetti, J. Bohg, T. Asfour, and S. Schaal, "Learning of grasp selection based on shape-templates," *Auton. Robots*, vol. 36, no. 1-2, pp. 51–65, Sep. 2013.

[17] Y. Jiang, S. Moseson, and A. Saxena, "Efficient grasping from RGBD images: learning using a new rectangle representation," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2011, pp. 3304–3311.

[18] L. Pinto and A. Gupta, "Supersizing self-supervision: learning to grasp from 50K tries and 700 robot hours," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2016, pp. 3406–3413.

[19] D. Kalashnikov, A. Irpan, P. Pastor, J. Ibarz, A. Herzog, E. Jang, D. Quillen, E. Holly, M. Kalakrishnan, V. Vanhoucke, and S. Levine, "Scalable deep reinforcement learning for vision-based robotic manipulation," in *Proc. Conf. Robot Learn.*, vol. 87, Oct. 2018, pp. 651–673.

[20] S. Levine, A. Kumar, G. Tucker, and J. Fu, "Offline reinforcement learning: tutorial, review, and perspectives on open problems," 2020, *arXiv:2005.01643*. [Online]. Available: https://arxiv.org/abs/2005.01643.

[21] A. Singh, L. Yang, K. Hartikainen, C. Finn, and S. Levine, "End-to-end robotic reinforcement learning without reward engineering," in *Robot.: Sci. Sys.*, Jun. 2019.

[22] B. Wu, I. Akinola, J. Varley, and P. K. Allen, "Mat: multi-fingered adaptive tactile grasping via deep reinforcement learning," in *Proc. Conf. Robot Learn.*, vol. 100, Oct. 2020, pp. 142–161.

[23] S. Levine, P. Pastor, A. Krizhevsky, J. Ibarz, and D. Quillen, "Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection," *Int. J. Robot. Res.*, vol. 37, no. 4-5, pp. 421–436, Jun. 2017.

[24] S. Dasari, F. Ebert, S. Tian, S. Nair, B. Bucher, K. Schmeckpeper, S. Singh, S. Levine, and C. Finn, "RoboNet: large-scale multi-robot learning," in *Proc. Conf. Robot Learn.*, vol. 100, Oct. 2020, pp. 885–897.

[25] H. Liang, X. Ma, S. Li, M. Gorner, S. Tang, B. Fang, F. Sun, and J. Zhang, "PointNetGPD: detecting grasp configurations from point sets," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2019, pp. 3629–3635.

[26] T. Patten, K. Park, and M. Vincze, "DGCM-Net: dense geometrical correspondence matching network for incremental experience-based robotic grasping," *Front. Robot. AI*, vol. 7, no. 120, Sep. 2020.

[27] D. Prattichizzo and J. C. Trinkle, "Grasping," in *Springer Handb. Robot.* Cham, Germany: Springer, 2016, pp. 955–988.

[28] I. Kao, K. M. Lynch, and J. W. Burdick, "Contact modeling and manipulation," in *Springer Handb. Robot.* Cham, Germany: Springer, 2016, pp. 931–954.

[29] S. El-Khoury and A. Sahbani, "On computing robust n-finger force-closure grasps of 3D objects," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2009, pp. 2480–2486.

[30] M. Li, K. Hang, D. Kragic, and A. Billard, "Dexterous grasping under shape uncertainty," *Robot. Auton. Syst.*, vol. 75, pp. 352–364, Jan. 2016.

[31] B. Mirtich and J. Canny, "Easily computable optimum grasps in 2-D and 3-D," in *Proc. IEEE Int. Conf. Robot. Autom.*, vol. 1, May 1994, pp. 739–747.

[32] L. Jaillet and J. M. Porta, "Path planning under kinematic constraints by rapidly exploring manifolds," *IEEE Trans. Robot.*, vol. 29, no. 1, pp. 105–117, Feb. 2013.

[33] J. Bohg, M. Johnson-Roberson, B. Leon, J. Felip, X. Gratal, N. Bergstrom, D. Kragic, and A. Morales, "Mind the gap-robotic grasping under incomplete observation," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2011, pp. 686–693.

[34] D. Berenson and S. S. Srinivasa, "Grasp synthesis in cluttered environments for dexterous hands," in *Proc. IEEE–RAS Int. Conf. Humanoind Robots*, Dec. 2008, pp. 189–196.

[35] A. Firouzeh and J. Paik, "Grasp mode and compliance control of an underactuated origami gripper using adjustable stiffness joints," *IEEE/ASME Trans. Mechatronics*, vol. 22, no. 5, pp. 2165–2173, Oct. 2017.

[36] V. Wall, G. Zöller, and O. Brock, "A method for sensorizing soft actuators and its application to the RBO hand 2," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2017, pp. 4965–4970.

[37] J. Xu, T. Aykut, D. Ma, and E. Steinbach, "6DLS: Modeling nonplanar frictional surface contacts for grasping using 6-D limit surfaces," *IEEE Trans. Robot.*, vol. 37, no. 6, pp. 2099–2116, Dec. 2021.

[38] F. Chaumette and S. Hutchinson, "Visual servo control. I.: basic approaches," *IEEE Robot. Autom. Mag.*, vol. 13, no. 4, pp. 82–90, Dec. 2006.

[39] D. Morrison, P. Corke, and J. Leitner, "Learning robust, real-time, reactive robotic grasping," *Int. J. Robot. Res.*, vol. 39, no. 2-3, pp. 183–201, Jun. 2020.

[40] D. Guo, F. Sun, B. Fang, C. Yang, and N. Xi, "Robotic grasping using visual and tactile sensing," *Inf. Sci.*, vol. 417, pp. 274–286, Nov. 2017.

[41] Y. Zhang, Z. Kan, Y. Yang, Y. A. Tse, and M. Y. Wang, "Effective estimation of contact force and torque for vision-based tactile sensors with helmholtz–hodge decomposition," *IEEE Robot. Autom. Lett.*, vol. 4, no. 4, pp. 4094–4101, Oct. 2019.

[42] P. Xiong, X. Tong, A. Song, and P. X. Liu, "Robotic multifinger grasping state recognition based on adaptive multikernel dictionary learning," *IEEE Trans. Instrum. Meas.*, vol. 71, pp. 1–14, May 2022.

[43] T. M. Huh, K. Sanders, M. Danielczuk, M. Li, Y. Chen, K. Goldberg, and H. S. Stuart, "A multi-chamber smart suction cup for adaptive gripping and haptic exploration," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, Dec. 2021, pp. 1786–1793.

[44] A. T. Miller and P. K. Allen, "GraspIt!" *IEEE Robot. Autom. Mag.*, vol. 11, no. 4, pp. 110–122, Dec. 2004.

[45] B. León, S. Ulbrich, R. Diankov, G. Puche, M. Przybylski, A. Morales, T. Asfour, S. Moisio, J. Bohg, J. Kuffner, and R. Dillmann, "OpenGRASP: a toolkit for robot grasping simulation," in *Proc. IEEE Int. Conf. Simul. Model. Program. Auton. Robots*, Nov. 2010, pp. 109–120.

[46] M. Malvezzi, G. Gioioso, G. Salvietti, D. Prattichizzo, and A. Bicchi, "SynGrasp: a MATLAB toolbox for grasp analysis of human and robotic hands," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2013, pp. 1088–1093.

[47] B. Calli, A. Walsman, A. Singh, S. Srinivasa, P. Abbeel, and A. M. Dollar, "Benchmarking in manipulation research: using the Yale-CMU-Berkeley object and model set," *IEEE Robot. Autom. Mag.*, vol. 22, no. 3, pp. 36–52, Sep. 2015.

[48] 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," *Int. J. Robot. Res.*, vol. 31, no. 8, pp. 927–934, May 2012.

[49] S. Kumra, S. Joshi, and F. Sahin, "Antipodal robotic grasping using generative residual convolutional neural network," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, Oct. 2020, pp. 9626–9633.

[50] Formhand, "FH-R150: Allrounder greifkissen," *Formhand.de*, https://www.formhand.de/produkte/FH-R150 [Accessed Feb. 04, 2022].

[51] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "CBAM: convolutional block attention module," in *Proc. Eur. Conf. Comput. Vis.*, Sep. 2018, pp. 3–19.

[52] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 770–778.

[53] D. P. Kingma and J. Ba, "Adam: a method for stochastic optimization," in *Proc. Int. Conf. Learn. Representions*, May 2015.

[54] A. Zeng, K.-T. Yu, S. Song, D. Suo, E. Walker, A. Rodriguez, and J. Xiao, "Multi-view self-supervised deep learning for 6D pose estimation in the Amazon Picking Challenge," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2017, pp. 1386–1383.

[55] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 2921–2929.

[56] C. Hennersperger, B. Fuerst, S. Virga, O. Zettinig, B. Frisch, T. Neff, and N. Navab, "Towards MRI-based autonomous robotic US acquisitions: a first feasibility study," *IEEE Trans. Med. Imag.*, vol. 36, no. 2, pp. 538–548, Feb. 2017.

[57] J. Mahler, M. Matl, X. Liu, A. Li, D. Gealy, and K. Goldberg, "Dex-Net 3.0: computing robust vacuum suction grasp targets in point clouds using a new analytic model and deep learning," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2018, pp. 5620–5627.

[58] R. Hu, P. Dollár, K. He, T. Darrell, and R. Girshick, "Learning to segment every thing," in *Proc. IEEE Conf. Comput. Vis. Patttern Recognit.*, 2018, pp. 4233–4241.

[59] M. Tuscher, J. Hörz, D. Driess, and M. Toussaint, "Deep 6-dof tracking of unknown objects for reactive grasping," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2021, pp. 14 185–14 191.

[60] K. Greff, F. Belletti, L. Beyer, C. Doersch, Y. Du, D. Duckworth, D. J. Fleet, D. Gnanapragasam, F. Golemo, C. Herrmann *et al.*, "Kubric: A scalable dataset generator," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2022, pp. 3749–3761.

Hui Zhang received the B.S. degree in electronic information engineering from the China University of Geosciences, Wuhan, China, in 2015, and the M.Sc. degree in instrument science and technology from Southeast University, Nanjing, China, in 2018. He is currently working toward the Ph.D. degree, titled "Random Bin Picking of Unknown Objects", with KU Leuven, Leuven, Belgium.

Since 2021 he have been a Member of the Flanders Make ROB core lab @ KU Leuven, Belgium. His research interests include robotic vision, contact modeling, grasp simulation, deep learning, haptic sensing, random picking of unknown objects, and their applications.

Jef Peeters obtained his Engineering degree in industrial design in 2008 from the University College Howest, Kortrijk, Belgium. He received a Master degree in Industrial Management with the option of product and production management in 2010 and defended his Ph.D. titled "Demanufacturing strategies for electronic products" in 2016 from the KU Leuven, Leuven, Belgium.

He is currently an Assistant Professor and Co-chair of the Lifecycle Engineering (LCE) research group at the Department of Mechanical Engineering of the KU Leuven. He is a member of SIM^2 - KU Leuven Institute for Sustainable Metals and Minerals.

Since 2022, he becomes a member of the Flanders Make VCCM Core Lab @ KU Leuven, 3000 Leuven, Belgium. His research focuses on (eco)design and rework, reuse, repair, remanufacturing and demanufacturing for the development of innovative products and processes for a circular economy.

Eric Demeester received the Ph.D. degree in engineering science from the Department of Mechanical Engineering, KU Leuven, Leuven, Belgium, in 2007.

Since 2013, he has been cochairing ACRO Research Group, KU Leuven, which focuses on automation, computer vision, and robotics. From 2013 to 2018, he was an Assistant Professor with KU Leuven. Since 2018 he has been an Associate Professor, with KU Leuven. His research interests include human-robot collaboration, robot motion planning, and probabilistic state estimation for industrial applications.

Karel Kellens obtained his Ph.D. degree in Engineering Science at KU Leuven, Belgium, in 2013.

Since 2018 he has been an Assistant Professor and Cochair of the Automation, Computer Vision & Robotics (ACRO) Research Group, Department of Mechanical Engineering, KU Leuven, where he is appointed as an Associate Professor "Flexible Handling and Assembly Technology." Since 2021 he has been ROB Core Lab Manager of Flanders Make @ KU Leuven, Leuven, Belgium. His research interests include on flexible handling, robotic product manipulation and adaptive (dis-)assembly systems.