Deformation-Aware Data-Driven Grasp Synthesis

Tran Nguyen Le ⓘ, *Graduate Student Member, IEEE*, Jens Lundell ⓘ, *Member, IEEE*, Fares J. Abu-Dakka ⓘ, and Ville Kyrki ⓘ, *Senior Member, IEEE*

Abstract—Grasp synthesis for 3-D deformable objects remains a little-explored topic, most works aiming to minimize deformations. However, deformations are not necessarily harmful—humans are, for example, able to exploit deformations to generate new potential grasps. How to achieve that on a robot is though an open question. This letter proposes an approach that uses object stiffness information in addition to depth images for synthesizing high-quality grasps. We achieve this by incorporating object stiffness as an additional input to a state-of-the-art deep grasp planning network. We also curate a new synthetic dataset of grasps on objects of varying stiffness using the Isaac Gym simulator for training the network. We experimentally validate and compare our proposed approach against the case where we do not incorporate object stiffness on a total of 2800 grasps in simulation and 560 grasps on a real Franka Emika Panda. The experimental results show significant improvement in grasp success rate using the proposed approach on a wide range of objects with varying shapes, sizes, and stiffnesses. Furthermore, we demonstrate that the approach can generate different grasping strategies for different stiffness values. Together, the results clearly show the value of incorporating stiffness information when grasping objects of varying stiffness. Code and video are available at: https://irobotics.aalto.fi/defggcnn/.

Index Terms—Deep Learning in grasping and manipulation, grasping.

I. INTRODUCTION

IN THE last decade, advancement in robotic grasping has enabled robots to automatically grasp a never-before-seen range of objects. However, most of the works on grasp synthesis still assume specific object properties such as uniform friction or rigidity. These assumptions do not hold for multi-material [2] or deformable objects and can lead to unsuccessful grasping in real-world scenarios.

Grasping non-rigid objects, on the other hand, is difficult because objects deform under interaction forces meaning that the 3-D contact locations also depend on the forces exerted on the object. Furthermore, the effect of the deformation varies across deformable objects and tasks. Because of these facts, grasps generated on rigid objects do not necessarily transfer well to deformable objects, as shown in Fig. 1(a). It is thus essential
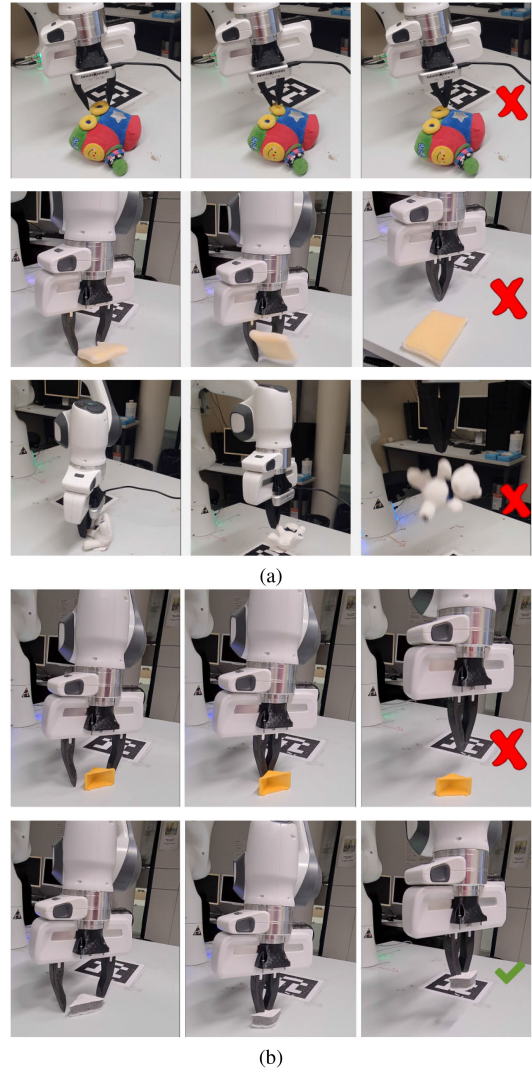
Fig. 1.　(a) Some grasps generated on rigid objects did not successfully hold the object during a dynamic task. (b) The robot executes the same grasp candidate on two objects with similar shapes but different stiffnesses.

to harness the deformation of objects when grasping. For some objects, such as grasping a water bottle, it might be important to minimize the object's deformation not to dislodge the liquid. While for other objects, such as the triangular-shaped object shown in Fig. 1(b), one can take advantage of the deformation to grasp them successfully.

To date, most of the existing works only focus on minimizing the object deformation [3]–[7]. Although there exist few works that take advantage of the deformation [8], [9], they mainly
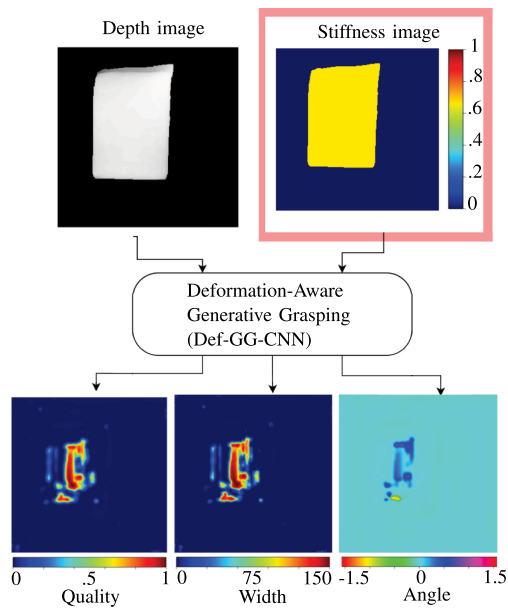
Fig. 2. The proposed pipeline where stiffness information (highlighted in red box) is incorporated.

focus on proposing control strategies given an initial grasp configuration. Thus, it is still an open question of how object stiffness affects the choice of grasp configuration and how to harness object deformation to generate better grasps.

To address the aforementioned open issues, we propose to incorporate stiffness as an additional input to a state-of-the-art deep grasp planning pipeline as shown in Fig. 2. Our system generates a grasp candidate and corresponding grasp quality for every pixel given an input depth image and a *stiffness image*. The model outputs can be reprojected into 3-D space when combined with depth information, allowing a robot to execute a generated grasp in the real world.

We qualitatively evaluated the proposed grasp synthesis method on a Franka Emika Panda robot in simulation and the real world by comparing it to a method that ignores stiffness. In the simulator, we evaluated over 2800 grasps on a shake and twist task, measuring, respectively, the grasp's robustness to linear and angular disturbances. In the real world, we measured the grasp success rate of 560 grasps on 14 objects in three different scenarios: with stiffness information, assuming all objects are rigid, and assuming all objects are deformable. In both simulation and the real world, our proposed approach demonstrates an improvement in grasp success rate. Furthermore, the approach can generate different grasping strategies for different stiffness values.

In summary, the main contributions of this letter are:
- The first generative stiffness-aware deep grasping approach that adapts the grasp location depending on the object's stiffness.
- The first stiffness-dependent image-based grasping dataset consists of labeled top-down grasps on objects with varying stiffness.
- A thorough empirical evaluation of the proposed method presenting, both in simulation and on real hardware,

improvements in terms of grasp ranking and grasp success rate over a method that ignores stiffness.

II. RELATED WORK

To put our work in context, we next review three complementary viewpoints, grasping of deformable objects, data-driven grasp synthesis, and simulation of deformable object interactions.

A. Grasping Deformable Objects

Most recent research on deformable object manipulation has mainly focused on manipulating cloth items [10]–[12], ropes [13]–[15], and 3-D deformable object [16]. Grasping deformable objects remains a sparsely explored research area [17] with the majority of works focusing on formulating quality metrics to quantify the goodness of a grasp [3], [4], [18], [19] or proposing control strategies [6]–[9].

Most approaches for grasping deformable objects aim to minimize the deformation. For a particular grasp, the minimization can be performed on-line by employing a control strategy that regulates the force at each contact [6], [7]. To plan grasps, minimum deformation can be achieved by placing fingers at locations with maximal local stiffness, determined e.g. using simulation [3]. The deformation can also be integrated as an additional component of a wrench-space grasp quality metric [4].

In contrast to minimizing deformation, some works have proposed to utilize the object deformation, similar to this letter. Analytical grasp planning approaches following this line of study include bounded force closure [18] which guarantees force closure under a bounded external force, and deform closure [19] which generalizes form closure to deformable objects with frictionless contact. In the on-line case, finger displacements can be regulated in order to retain force closure, originally proposed for planar objects [8] and later extended to 3-D [9] by using Finite Element Method (FEM) to continuously model changes in shape and contact geometry during object lifting.

Although [8], [9] utilized object deformations for stable grasping, they focus on regulating either force or displacement of the fingers given an initial grasp configuration. This is in contrast to our work which focuses on the choice of the grasp configuration by taking advantage of the object deformation.

B. Data-Driven Grasp Synthesis

Rapid advances in deep learning research recently have changed the paradigm of robotic grasping from analytical methods to data-driven ones. The main reason for this paradigm shift is that data-driven methods have been proved to be able to generate grasps that typically achieve a high grasp success rate on a wide range of objects in just a matter of seconds, much faster compared to analytical methods [20]–[27]. For example, Mahler *et al.* [21] used a dataset consisting of millions of synthetic antipodal top-grasp to train a Grasp Quality Convolutional Neural Network (GQ-CNN) model that computes the probability of success of grasps from depth images. The GQ-CNN was further improved through the use of on-policy

data and a fully-convolutional network (FCN) structure called FC-GQ-CNN [23]. The FCN structure has recently been found to perform well in grasp synthesis [22], [23], [28]–[30], having the ability to generate dense, pixel-wise predictions for an input image efficiently.

Although the aforementioned works achieve impressive results on rigid objects, none of them explicitly investigate the usability on deformable objects especially 3-D deformable objects. In this work, we tackle this problem by incorporating object deformation into a state-of-the-art deep data-driven grasping planning pipeline.

C. Soft-Body Simulation

Data-driven grasping requires training data, which in this work we generate in simulation. Simulating dynamics of deformable objects relies heavily on their geometric representations, for instance, particle representation is a good choice for simulating the dynamics of fluids. Yin *et al.* [31] presents three primary deformable object modelling approaches, Mass-Spring System (MSS), Position-based Dynamics (PBD), and FEM, and their limitations. In this work, we decided to use FEM because it is often used to model 3-D objects such as food [32] and tissues [33] and, compared to other modeling approaches, offers a more physically accurate representation of a deformable object in a continuous domain at the expense of computational cost.

Some robotic simulators that support FEM are PyBullet [34], SOFA [35], and NVIDIA's recent version of the Isaac Gym simulator [36], which supports soft body simulation through the NVIDIA Flex backend. Among the aforementioned simulators, Isaac Gym is chosen as it combines the advantages of the other two. Similar to SOFA, Isaac Gym includes a co-rotational linear model for precision modeling and simulating the object deformation under interaction. Like PyBullet, Isaac Gym provides capabilities to integrate robot-related functions, making it easier to build robotic applications. Huang *et al.* [17] also provides a grasping framework in Isaac Gym named DefGraspSim to automatically perform and evaluate grasp tests on an arbitrary target object. We use this framework in our work to generate training data and to test grasps.

III. PROBLEM FORMULATION

This work addresses the problem of generating antipodal grasps on unknown objects with different stiffnesses lying on a supporting surface. The goal is to calculate a grasp for each pixel in the depth image while taking into account object stiffness. More formally, we train a model $\mathcal{M}$ that takes as input a depth image $\mathbf{I_d}$ and a stiffness image $\mathbf{I_s}$, and produces a grasp map $\mathbf{G}$ that incorporates grasp quality and grasp parameters (orientation, gripper width) for grasps centered at each pixel in the input.

$$\mathcal{M} : (\mathbf{I_d}, \mathbf{I_s}) \mapsto \mathbf{G}.$$

To achieve this goal, we propose to use the Deep Neural Network (DNN) in Fig. 2 to map from depth and stiffness images to grasps $\mathbf{G}$ in the image, which we can easily transform to the real world using known coordinate transforms.

IV. METHOD

A. Network

Our solution is based on the Generative Grasping Convolutional Neural Network 2 (GG-CNN2) a fully-convolutional network proposed by Morrision *et al.* [30]. The detail architecture of the network is presented in [30]. This network is orders of magnitude smaller than other recent grasping networks, making it faster to train and evaluate, while still achieving state-of-the-art results in rigid object grasping. However, as shown in Fig. 2, we modify GG-CNN2 to take an additional stiffness image as input, enabling it to learn different grasps based on the object's stiffness. Specifically, we modify the GG-CNN2 convolution module to accommodate the extra input. We call this new network Deformation GG-CNN (*Def-GG-CNN*).

Similar to the GG-CNN2 network architecture, *Def-GG-CNN* also uses dilated convolutional layers to improve the performance in semantic segmentation. To enable *Def-GG-CNN* to learn stiffness-dependent grasping strategies, it has, alongside the depth image, an additional stiffness image input. The stiffness image represents Young's modulus of the object at each pixel. The output of the network is the grasp map $\mathbf{G}$ that represents a grasp quality, and gripper parameters (orientation, gripper width) for each pixel of the depth image. The proposed network is trained with supervised learning on a synthetic dataset with the L2 loss funtion $\mathcal{L}(G, \mathcal{M}_\theta(\mathbf{I_d}, \mathbf{I_s}))$. The synthetic training dataset further explained in Section V.

B. Grasp Map Representation

Each pixel in the grasp map $\mathbf{G}$ represents a top-down grasp, which is defined by the grasp center position, orientation, and width. We use the same representation of $\mathbf{G}$ as defined in [22]. As shown in Fig. 2, the grasp map $\mathbf{G}$ consist of three images: grasp quality $\mathbf{Q}$, orientation ϕ, and gripper width $\mathbf{W}$.

$\mathbf{Q}$ denotes the quality of a grasp centered at each pixel. The quality of a grasp is a scalar value between [0, 1], where the higher the value, the better the grasp. ϕ is the orientation image, representing the pixel-wise orientation of a grasp around the image normal. Because an antipodal grasp is symmetric beyond 180 degrees, we limit the orientation between $[-\pi/2, \pi/2]$ radians. Finally, $\mathbf{W}$ is the width image that describes the pixel-wise gripper width from [0, 150] pixels. We transform the pixel-dependent gripper width to real-world units using the measured depth and the camera parameters.

V. DATASET

To train *Def-GG-CNN*, we need a dataset consisting of depth, stiffness, quality, orientation, and width images. To date, there exists no such dataset, and, thus, we opted to curate our own synthetic dataset.

The pipeline of generating training data is visualized in Fig. 3. We first convert the triangular mesh of an object into tetrahedral mesh using fTetWild [37] and feed that tetrahedral mesh to the Isaac Gym simulator to enable its soft bodies simulation feature. The stiffness of an object can then be varied by adjusting the material parameters, *i.e.*, Young's modulus and Poisson's Ratio.
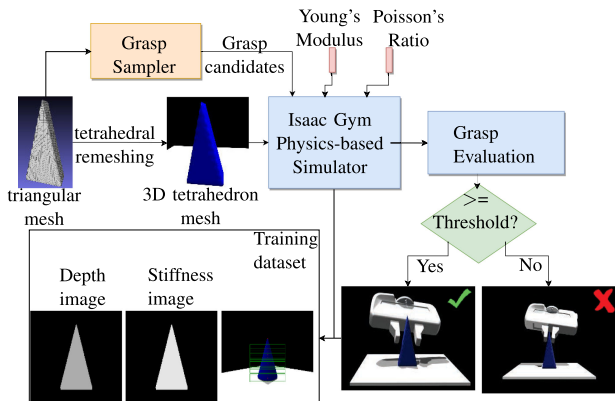
Fig. 3. The training data generation pipeline.

Then using the object triangular mesh, we sample grasp candidates which are later evaluated with Isaac Gym using proper quality metrics. Based on the performance of the grasps, we then label the grasps, convert them to the desired representation, and store them in the training dataset.

Depth and stiffness input: We captured depth images of target objects with a virtual camera set to view the scene from top-down. To model variable object stiffness, four values of Young's modulus from $2 \cdot 10^4$ to $2 \cdot 10^9$ were used. The Young's modulus is normalized to $[0, 1]$ range and the corresponding stiffness value is assigned to every pixel in the stiffness image that the object occupies. To simplify the problem, we decided to keep the Poisson' ratio fixed at 0.3 as suggested in [17]. The dimension of the two images is 300×300.

Grasp candidates: Grasps are sampled with an antipodal grasp sampler to obtain approximately 200 grasp candidates for each target object. All grasps that collide with the mesh are filtered out, resulting in a final set of 25 to 40 collision-free grasps for each object.

Quality metrics: None of the standard grasp quality metrics, such as the Ferrari & Canny L_1 metric [38], are directly applicable for both rigid and deformable objects. As a quality metric, we use a *shake task* which measures how easily an object is displaced in hand under linear accelerations. A higher metric means a better grasp as it indicates that a grasp can withstand higher accelerations. We use this metric to label a grasp as a positive or negative grasp by checking if the linear acceleration it can withstand is above or below a threshold. Specifically, after successfully lifting the object for each grasp candidate, we linearly increase the acceleration of the grasp along 16 directions until the gripper loses contact with the objects or reaches the upper acceleration limit, which is set to 50 m/s^2. Then we compute the average acceleration over all directions, and if this value is higher than the threshold of 25 m/s^2, we label the grasp candidate as a positive grasp.

Ground-truth grasp map: To further simplify the data generation, we only use positively labeled grasps as ground-truth grasps to train the network. To generate the ground-truth grasps, we first transform all grasps to the image space. To do so, we represent the grasps as rectangles in the image as displayed in Fig. 4. Four parameters define the rectangles: grasp center, grasp
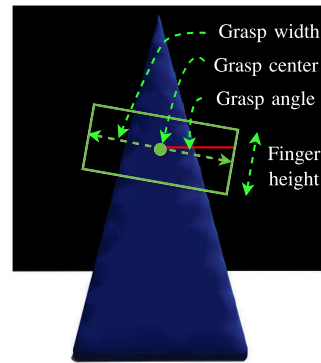

Fig. 4. A grasp is represented as a rectangle in 2D image plane.
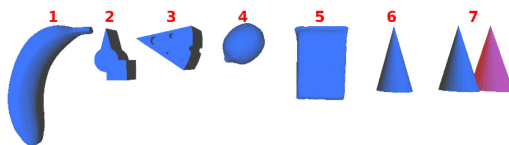

Fig. 5. The seven common objects used in the experiment. All objects are single-material except for object 7, where the stiffness of its red part can vary.

orientation, grasp width, and finger height. Finally, we use the rectangles as image masks to generate ground-truth grasp maps $\mathbf{G}$. Specifically, all pixels of the quality images $\mathbf{Q}$, angle images ϕ, and width images $\mathbf{W}$ within the rectangle are set to the values given from the *shake task*. In contrast, all pixels outside the rectangle are set to invalid.

Training dataset: As a training dataset, we generate and label grasps on 30 objects. The objects include 13 primitive objects provided in Isaac Gym, 5 objects from the YCB dataset [39], and 12 objects with adversarial geometry from the EGAD! dataset [40]. Because we set the stiffness for each object to four different values, the training set contains, in total, 120 objects. We use the Franka Emika Panda gripper model to execute grasps on objects in the simulator. To counteract the small size of the training set, we further augment the dataset with random crops, zooms, and rotations to create a set of 5400 depth and stiffness images with 27000 corresponding labeled grasp maps.

VI. EXPERIMENTS AND RESULTS

The experiments address the following three questions:
- Can *Def-GG-CNN* synthesize high-quality grasps for deformable objects and would they succeed in simulation?
- Is *Def-GG-CNN* robust against errors in stiffness input?
- Can *Def-GG-CNN*, trained purely on synthetic data, generate successful grasps in the real world?
- How does the stiffness information affect the grasp configuration?

A. Grasping in Simulation

We investigated the quality of synthesized grasps in the Isaac Gym simulation using the mentioned DefGraspSim framework on two sets of objects: 7 common objects shown in Fig. 5, and 28 adversarial objects from the recent EGAD! test dataset [40]. For

TABLE I
AVERAGE GRASP SUCCESS RATE (%) ON DIFFERENT STIFFNESSES FOR TWO DIFFERENT TASKS. THE HIGHER THE BETTER.

Stiffness (E)	Shake task				Twist task			
	Common		EGAD!		Common		EGAD!	
	With stiffness	No stiffness	With stiffness	No stiffness	With stiffness	No stiffness	With stiffness	No stiffness
$2 \cdot 10^4$	75.4	22.8	67	40.8	65.7	20	69.5	51.3
$2 \cdot 10^5$	88.5	38.5	77.4	68.6	74.2	40	80	65.2
$2 \cdot 10^6$	88.5	48.5	77.4	69.6	71.4	42.8	75	64.4
$2 \cdot 10^9$	51.4	46.2	51.3	51.3	40	31.4	41.7	41.7
All stiffnesses	**75.9**	40.3	**68**	57.6	**63.5**	34	**67.3**	55.6

each object and stiffness, we evaluated the five grasps with the highest quality on a *shake task* and *twist task*. While the *shake task* measures how easily an object slips out of the gripper under linear accelerations, the *twist task* measures that under angular accelerations. A grasp is successful if the object is in the gripper during the whole procedure, the grasp can withstand the linear acceleration limit of 25 m/s^2, and the angular acceleration limit of 500 rad/s^2. By doing this, we can quantify how the generated grasps behave under different disturbances.

To demonstrate the importance and the effect of stiffness input, we compared the stiffness-aware grasps against a GG-CNN2 baseline that only used depth images as input and no stiffness information. The baseline was also optimized on our synthetic dataset, but instead of having a varying Young's modulus we only used labeled grasps in the case of Young's modulus of $2 \cdot 10^9$ as ground truths. In total, we evaluated 1400 grasps per method.

Table I shows the simulation result on both test sets. We can see that the proposed approach that takes stiffness input into account achieves a higher grasp success rate across all object sets and disturbances. Interestingly, the network generated grasps that performed well on the *twist task* task despite being only trained on the quality metric from the *shake task*. The reason why the network was only trained on the quality metric of the *shake task* other than both *shake task* and *twist task* stems from the fact that the computation cost during data generation is extremely expensive.

Focusing on the *shake task* results, the average grasp success rate using our approach over all stiffnesses compared to the baseline is 35% higher on the Common objects and 11% higher on EGAD! objects. Moreover, the performance of the baseline approach deteriorates significantly when moving from a high to a low value of Young's modulus. For instance, on the Common test set, the relative performance drop for the baseline approach when changing the Young's modulus from $2 \cdot 10^6$ to $2 \cdot 10^5$ is 10%, and from $2 \cdot 10^6$ to $2 \cdot 10^4$ the drop is 26%. This decline is much higher compared to the 0% and 13% drop using our approach. Similar performance differences are also observed for the EGAD! test set. The primary reason the baseline approach witnesses a higher performance drop is because it generates the same grasps for a target object regardless of its stiffness. Although the generated grasps often picked the objects successfully, they usually slip away from the gripper during the shake or twist task. In contrast, the network that took the stiffness input into account learned to avoid areas with a high probability of
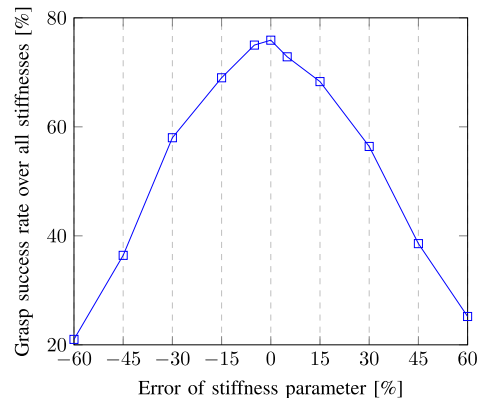

Fig. 6. Grasp success versus error in stiffness information. Grasp success deteriorates smoothly when the stiffness information is imprecise.

slippage, resulting in a higher grasp success rate. In addition, it is noteworthy that there is also some deterioration with the highest stiffness ($2 \cdot 10^9$) for both approaches. The primary reason is that some objects in both test set have very complex shape which are extremely hard to grasp when they are rigid. This observation strengthen the idea of taking advantage of object deformation to successfully grasp complex-shaped objects.

We also evaluated the models on the multi-material object 7 shown in Fig. 5, where the stiffness of the red part could differ from the blue part. The result showed that if we assumed object 7 is entirely rigid, the method could not generate any good grasps on it. However, if we assumed the red part was softer than the blue part, we could generate successful grasps that usually aimed for the softer area of the object. This simple example demonstrates the benefit of planning stiffness-aware grasps on irregular-shaped multi-material objects.

B. Sensitivity Analysis

To examine the robustness of the results of our approach in the presence of uncertainty, we conducted a sensitivity analysis where we introduce uncertainty to the input stiffness images by varying the stiffness parameter, *i.e.*, Young's modulus across an error range of [−60%, +60%]. We then evaluated the generated grasps on the Common test set under the *shake task*.

Fig. 6 shows the result of the sensitivity analysis. We can see that the grasp success rate decreases consistently with increasing error, but that small errors in the range of 5–15% have only

Fig. 7. The 14 individually numbered objects used in the real experiment. The objects represent a high variation in size, shape, and stiffness.

limited negative effect on the performance. This experiment indicates that our method is robust against some errors in expected stiffness, which suggests potential for real-world application even when stiffness is not precisely known.

C. Grasp Transfer to Physical Robot

To investigate how well the synthesized grasps perform in real world, we evaluated the grasp success rate on a Franka Emika Panda equipped with a parallel-jaw gripper. This allows us to study if grasps generated with the approach trained only on synthetic data transfer to real objects. 14 objects to grasp (Fig. 7) were chosen as they represent a high variation in size, shape, and stiffness.

We used an Intel RealSense D435 camera mounted to the robot's wrist to capture the RGB-D image. In addition to depth image, we also need to provide a stiffness image of the object. To do this, we segmented the object from the scene by subtracting the background and the table from the image and then assigning the same stiffness value to each pixel that the object occupied. We manually set the magnitude of stiffness for each object according to its perceived stiffness. The best grasp pose is computed using the proposed method. If there exist multiple best grasps, *i.e.*, multiple pixels that have the same value of the maximum quality, the first occurrence pixel is chosen as the best grasp.

The robot executed the best grasp by moving to a pre-grasp position approximately 25 mm above the grasp. Then, the robot moves linearly downwards until reaching the grasp pose or contact with the table is detected. From there, the robot closes its gripper, lifts the object, performs a predefined trajectory, and finally places an object at the goal position. A grasp is successful if the robot can pick the object and move it without dropping it. Otherwise, it is unsuccessful.

To single out the effect stiffness input has on grasp performance for each object, we ran the experiments with three different cases where we: provide the correct objects stiffness, assume all objects are rigid, assume all objects are soft. Specifically, in the case of providing the correct objects stiffness, we manually set the magnitude of stiffness for each object according to its perceived stiffness. In other cases, we assume all objects are rigid or soft by setting the Young's modulus of objects to $2 \cdot 10^9$ and $2 \cdot 10^4$, respectively. In addition, we also ran the

experiments with the pre-trained model GG-CNN2 on this test set. For each object and stiffness, we randomly placed it ten times and evaluated the best grasp candidate. In total, this setup amounts to 560 grasps on 14 objects.

The experimental results are presented in Fig. 8. The results demonstrate that with the correct stiffness information, the grasp success rate of the proposed approach is 17% higher than if we either assume all objects are rigid or deformable. *Def-GG-CNN* also achieves a 10% higher grasp success rate than the baseline GG-CNN2. Together, these results indicates that grasps generated on rigid objects do not necessarily transfer successfully to deformable objects and vice versa.

For instance, if we assumed the deformable objects, such as objects 8, 9, 11, and 14, were rigid, many generated grasps were on specific parts of them, such as the arms or legs of the toy or wheel of the car. These grasps usually picked the object successfully but then, due to the deformation, dropped it when the robot started to accelerate. The same experiment on objects 6, 7 showed that grasps generated on the top or bottom of the object usually failed due to the elasticity of the object. If we assumed rigid objects, such as 1, 3, 4, 5, were soft, the network generated pinch grasps that failed due to collisions with the object. Some successful grasps are shown in Fig. 9 and some failed grasps are shown in Fig. 1(b).

An interesting observation was that given the correct stiffness information, the method generated different grasping strategies depending on the stiffness of the object, as shown in Fig. 10. Specifically, in the case of the soft sponge shown in Fig. 10 a, the proposed method predicted that the grasp quality is high across the whole object due to its deformation that enables pinch grasps. While in the case of the hard sponge shown in Fig. 10 b, the high-quality grasp tends to be generated at the center of the object, and the grasp width is as big as the object in order to successfully cage the object.

D. Discussion

All experimental results show the benefit of generating stiffness-aware grasps. By comparing the proposed approach to the case where the stiffness information is ignored, we see that the proposed approach achieves higher grasp success rates. The primary reason for the difference in performance is that the object stiffness facilitates learning where to generate grasps that minimize the slippage caused by the deformation. If the object's stiffness was ignored, the network generated the same grasps regardless of the object stiffness. Together, these result backs the claim made in [9] that grasps do not transfer well between rigid and deformable objects. Therefore, incorporating object stiffness in robotic grasping pipelines is beneficial when dealing with a wide range of unknown objects.

It is especially noteworthy that although *Def-GG-CNN* is trained purely on synthetic data it generalized well to real world scenes without any fine-tuning. We hypothesize that this level of generalization is due to the effect of data augmentation, where we apply random rotations, zooms and crops of the training dataset images to achieve camera viewpoint and size invariance.

Another interesting observation is that our approach generated grasps in different area on the object depending on its stiffness.
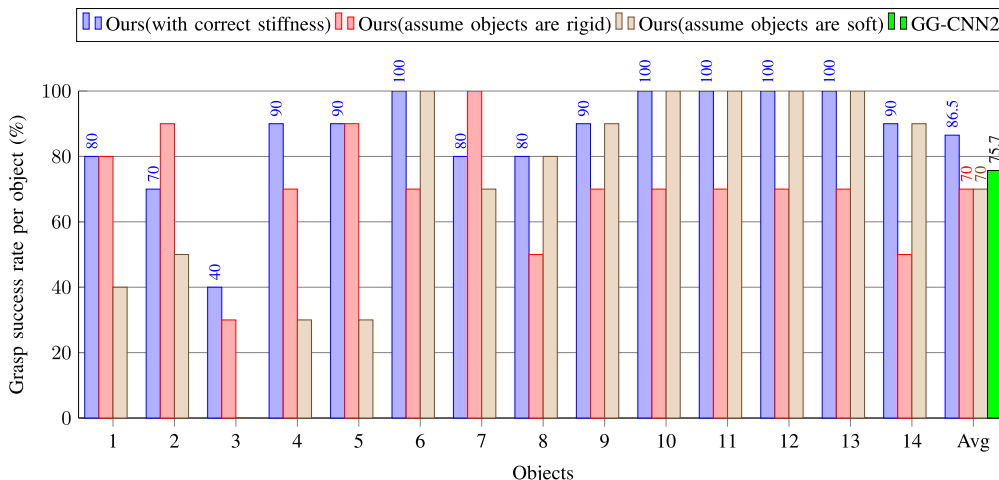
Fig. 8. Grasp success rate per object in three cases produced by our method. The last column shows the average success rate for all cases of our method and GG-CNN2.
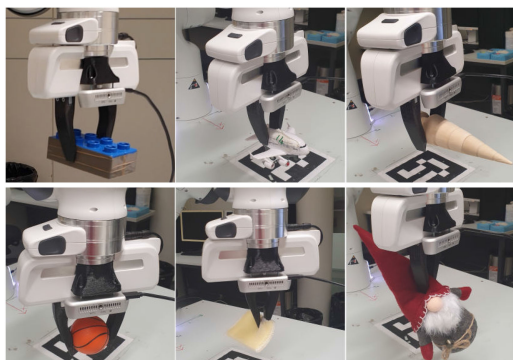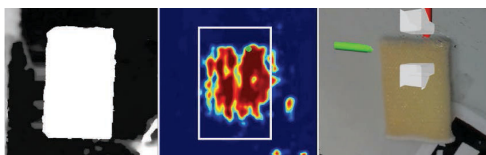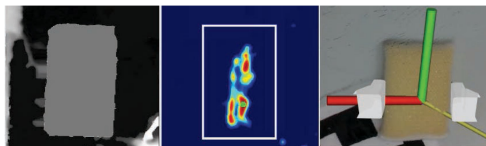

Fig. 9. Some successful grasps on the real robot.

(a) Pinch grasp on soft sponge $\mathbf{E} = 2 \cdot 10^5$.

(b) Cage grasp on rigid sponge $\mathbf{E} = 2 \cdot 10^9$.

Fig. 10. Stiffness input image, along with grasp quality map and best synthesized grasp candidate indicated by two white fingers. For stiffness input image, the darker the color the stiffer the object is. For the grasp quality map, red indicates higher quality, and the green point denotes the best grasp.

This behavior is shown on object 6 in Fig. 10. Specifically, the sponge with a low Young's modulus admits pinching behavior where the grasp press on the object and pinch, while the hard sponge only admits caging grasps. Our hypotheses regarding this may result from the fact that for soft objects the system provide good grasps across the objects to areas on those objects

that were not included in the training data. Similar behaviors were also reported in [41] where data was collected from 14 robots over the course of two months. However, it is noteworthy that our approach learned to produce the same behavior on a completely synthetic dataset with orders of magnitude less data. Furthermore, our proposed approach provides more meaningful insights regarding the relationship between object deformation and grasps.

VII. Conclusion and Future Work

Grasping deformable objects has not received as much attention as rigid object grasping due to complexity in the modeling and simulating the dynamic behavior of such objects. However, with the rapid development of physics-based simulators that support soft bodies, it is now possible to reliably simulate deformable objects. To leverage the capability of such simulators and to challenge the rigidity assumption that has dominated robotic grasping, we presented an approach to synthesize grasps on objects with varying stiffness by a deep neural network trained on purely synthetic data. The key idea in this work was to integrate the object stiffness property into the grasp planning pipeline to study the relationship between the object deformation and the generated grasps. To train the proposed network, we curated our own training dataset using the Isaac Gym simulator. We experimentally demonstrated the performance of the generated grasps in simulation and the real-world on a wide range of objects with varying sizes, shapes, and stiffness. The results show a clear improvement in grasp success rate when taking stiffness property into account. Furthermore, the proposed approach shows the ability to generate different grasp strategies depending on object stiffness. The generalizability to objects with non-uniform stiffness remains open although the method should be able to account for all variability captured in the training data.

The idea of exploiting deformations for grasping opens many interesting avenues. One of the limitation of our proposed method is that it employs FEM simulations for which the

computational cost makes our grasp evaluation procedure a central bottleneck. Investigating the possibility to devise analytical quality measures that would exploit the deformations is an important future work to tackle the mentioned problem. Another interesting future work would be to include more critical grasp parameters for deformable objects, which are currently not considered in this letter, such as grasp height, or closing width/ closing force into the proposed method or to expand the proposed method to incorporate more generic 6 DOF grasps.

ACKNOWLEDGMENT

REFERENCES

[1] T. Nguyen Le, J. Lundell, F. J. Abu-Dakka, and V. Kyrki, "Towards synthesizing grasps for 3-D deformable objects with physics-based simulation," in *Proc. DO-Sim Workshop Robot.: Sci. Syst.*, 2021.

[2] T. Nguyen Le, F. Verdoja, F. J. Abu-Dakka, and V. Kyrki, "Probabilistic surface friction estimation based on visual and haptic measurements," *IEEE Robot. Automat. Lett.*, vol. 6, no. 2, pp. 2838–2845, Apr. 2021.

[3] N. Alt, J. Xu, and S. Eckehard, "Grasp planning for thin-walled deformable objects," in *Proc. Robot. Hands, Grasping, Manipulation (ICRA Workshop)*, 2015.

[4] J. Xu, M. Danielczuk, J. Ichnowski, J. Mahler, E. Steinbach, and K. Goldberg, "Minimal work: A grasp quality metric for deformable hollow objects," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2020, pp. 1546–1552.

[5] Z. Pan, X. Gao, and D. Manocha, "Grasping fragile objects using a stress-minimization metric," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2020, pp. 517–523.

[6] A. Delgado, C. A. Jara, D. Mira, and F. Torres, "Tactile-based grasping strategy for deformable objects' manipulation and deformability estimation," in *Proc. 12th Int. Conf. Inform. Control Automat. Robot.*, vol. 02, 2015, pp. 369–374.

[7] T. Nguyen Le, J. Lundell, and V. Kyrki, "Safe grasping with a force controlled soft robotic hand," in *Proc. IEEE Int. Conf Syst., Man, Cybern.*, 2020, pp. 342–349.

[8] H. Lin, F. Guo, F. Wang, and Y.-B. Jia, "Picking up soft 3-D objects with two fingers," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2014, pp. 3656–3661.

[9] "Picking up a soft 3-D object by 'feeling' the grip," *Int. J. Robot. Res.*, vol. 34, no. 11, pp. 1361–1384, 2015.

[10] D. Tanaka, S. Arnold, and K. Yamazaki, "EMD Net: An encode-manipulate-decode network for cloth manipulation," *IEEE Robot. Automat. Lett.*, vol. 3, no. 3, pp. 1771–1778, Jul. 2018.

[11] B. Jia, Z. Pan, Z. Hu, J. Pan, and D. Manocha, "Cloth manipulation using random-forest-based imitation learning," *IEEE Robot. Automat. Lett.*, vol. 4, no. 2, pp. 2086–2093, Apr. 2019.

[12] R. Jangir, G. Alenyà, and C. Torras, "Dynamic cloth manipulation with deep reinforcement learning," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2020, pp. 4630–4636.

[13] J. Zhu, B. Navarro, R. Passama, P. Fraisse, A. Crosnier, and A. Cherubini, "Robotic manipulation planning for shaping deformable linear objects withenvironmental contacts," *IEEE Robot. Automat. Lett.*, vol. 5, no. 1, pp. 16–23, Jan. 2020.

[14] M. Yan, Y. Zhu, N. Jin, and J. Bohg, "Self-supervised learning of state estimation for manipulating deformable linear objects," *IEEE Robot. Automat. Lett.*, vol. 5, no. 2, pp. 2372–2379, Apr. 2020.

[15] T. Tang, C. Wang, and M. Tomizuka, "A framework for manipulating deformable linear objects by coherent point drift," *IEEE Robot. Automat. Lett.*, vol. 3, no. 4, pp. 3426–3433, Oct. 2018.

[16] Z. Hu, T. Han, P. Sun, J. Pan, and D. Manocha, "3-D deformable object manipulation using deep neural networks," *IEEE Robot. Automat. Lett.*, vol. 4, no. 4, pp. 4255–4261, Oct. 2019.

[17] I. Huang *et al.*, "DefGraspSim: Simulation-based grasping of 3-D deformable objects," in *Proc. Robot.: Sci. Syst. Workshop Deformable Object Simul.*, 2021.

[18] H. Wakamatsu, S. Hirai, and K. Iwata, "Static analysis of deformable object grasping based on bounded force closure," in *Proc. IEEE Int. Conf. Robot. Automat.*, 1996, pp. 3324–3329.

[19] K. Gopalakrishnan and K. Goldberg, "D-space and deform closure grasps of deformable parts," *Int. J. Robot. Res.*, vol. 24, no. 11, pp. 899–910, 2005.

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

[21] 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.

[22] D. Morrison, P. Corke, and J. Leitner, "Closing the loop for robotic grasping: A real-time, generative grasp synthesis approach," in *Proc. Robot.: Sci. Syst.*, 2018.

[23] V. Satish, J. Mahler, and K. Goldberg, "On-policy dataset synthesis for learning robot grasping policies using fully convolutional deep networks," *IEEE Robot. Automat. Lett.*, vol. 4, no. 2, pp. 1357–1364, Apr. 2019.

[24] S. Song, A. Zeng, J. Lee, and T. Funkhouser, "Grasping in the wild: Learning 6DoF closed-loop grasping from low-cost demonstrations," *IEEE Robot. Automat. Lett.*, vol. 5, no. 3, pp. 4978–4985, Jul. 2020.

[25] 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.

[26] 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.

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

[28] A. Zeng *et al.*, "Robotic pick-and-place of novel objects in clutter with multi-affordance grasping and cross-domain image matching," *Int. J. Robot. Res.*, Aug. 2019, doi: 10.1177/0278364919868017.

[29] E. Johns, S. Leutenegger, and A. J. Davison, "Deep learning a grasp function for grasping under gripper pose uncertainty," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2016, pp. 4461–4468.

[30] 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, 2020.

[31] H. Yin, A. Varava, and D. Kragic, "Modeling, learning, perception, and control methods for deformable object manipulation," *Sci. Robot.*, vol. 6, no. 54, 2021.

[32] E. Heiden, M. Macklin, Y. S. Narang, D. Fox, A. Garg, and F. Ramos, "DiSECt: A differentiable simulation engine for autonomous robotic cutting," in *Proc. Robot.: Sci. Syst.*, 2021.

[33] I. Leizea, A. Mendizabal, H. Alvarez, I. Aguinaga, D. Borro, and E. Sanchez, "Real-time visual tracking of deformable objects in robot-assisted surgery," *IEEE Comput. Graph. Appl.*, vol. 37, no. 1, pp. 56–68, Jan./Feb. 2017.

[34] E. Coumans and Y. Bai, "Pybullet, A python module for physics simulation for games, robotics and machine learning," 2016–2021. [Online]. Available: http://pybullet.org

[35] F. Faure *et al.*, "SOFA: A multi-model framework for interactive physical simulation," in *Soft Tissue Biomech. Modelling for Comp. Assisted Surgery*. Berlin, Heidelberg, Germany: Springer, 2012 pp. 283–321.

[36] J. Liang, V. Makoviychuk, A. Handa, N. Chentanez, M. Macklin, and D. Fox, "GPU-accelerated robotic simulation for distributed reinforcement learning," in *Proc. Conf. Robot Learn.*, 2018, pp. 270–282.

[37] Y. Hu, T. Schneider, B. Wang, D. Zorin, and D. Panozzo, "Fast tetrahedral meshing in the wild," *ACM Trans. Graph.*, vol. 39, no. 4, Jul. 2020, Art. no. 117.

[38] C. Ferrari and J. Canny, "Planning optimal grasps," in *Proc. IEEE Int. Conf. Robot. Automat.*, vol. 3, 1992, pp. 2290–2295.

[39] 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. Automat. Mag.*, vol. 22, no. 3, pp. 36–52, Sep. 2015.

[40] 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.

[41] 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, 2018.