GP-net: Flexible Viewpoint Grasp Proposal

Anna Konrad, John McDonald and Rudi Villing

***Abstract*—We present the Grasp Proposal Network (GP-net), a Convolutional Neural Network model which can generate 6-DoF grasps from flexible viewpoints, e.g. as experienced by mobile manipulators. To train GP-net, we synthetically generate a dataset containing depth-images and ground-truth grasp information. In real-world experiments, we use the EGAD evaluation benchmark to evaluate GP-net against two commonly used algorithms, the Volumetric Grasping Network (VGN) and the Grasp Pose Detection package (GPD), on a PAL TIAGo mobile manipulator. In contrast to the state-of-the-art methods in robotic grasping, GP-net can be used for grasping objects from flexible, unknown viewpoints without the need to define the workspace and achieves a grasp success of 54.4% compared to 51.6% for VGN and 44.2% for GPD. We provide a ROS package along with our code and pre-trained models at https://aucoroboticsmu.github.io/GP-net/.**

***Index Terms*—grasping, robotics, neural networks, 6-DoF grasps, mobile manipulator, ROS**

I. Introduction

Manipulation is an ongoing and challenging problem in robotics due to its complexity and variability. Grasping objects is largely solved in industrial settings with known objects and fixed, foreknown poses of the object and robot. However, more complex, dynamic and unstructured environments are still an active field of research. Before the era of machine learning, analytical approaches were widely tested in simulation but had limited applicability in real-world scenarios due to noisy or partial data [1]. With the rise of computational power, researchers have switched to data-driven approaches, which are more robust to real-world conditions. To simplify the problem, these solutions initially focused on fixed overhead cameras and a reduced grasp space with 4 Degrees-of-Freedom (DoF) grasps [2]–[7], i.e. top-grasps which can vary their pose in 3D position and one rotational axis.

However, when grasping outside of a lab environment, the restriction to top-grasps limits the applicability of the algorithms, while 6-DoF grasps can enhance reachability for the robots. Furthermore, if the camera position is not fixed, camera pose changes must be handled. Finally, methods must be robust to sensor noise in real-world environments. Efforts are being made towards solutions for many of these problems, for example with 6-DoF grasp proposal algorithms [8]–[12], closed-loop grasping [5], [13], [14], or grasping transparent objects [15].

This publication has emanated from research conducted with the financial support of Science Foundation Ireland under Grant numbers 18/CRT/6049 and 16/RI/3399. For the purpose of Open Access, the author has applied a CC BY public copyright licence to any Author Accepted Manuscript version arising from this submission. The opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Science Foundation Ireland.

Anna Konrad is with the Hamilton Institute and the Department of Electronic Engineering, Maynooth University, Maynooth, Ireland. anna.konrad.2020@mumail.ie

John McDonald is with the Department of Computer Science, Maynooth University, Maynooth, Ireland. john.mcdonald@mu.ie

Rudi Villing is with the Department of Electronic Engineering, Maynooth University, Maynooth, Ireland. rudi.villing@mu.ie

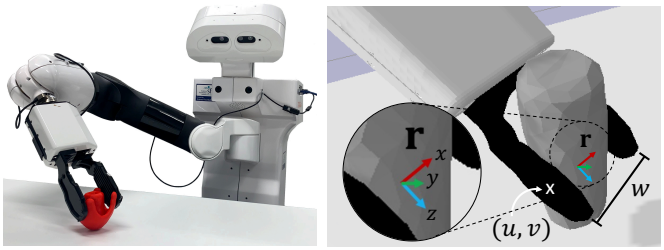

Fig. 1. a) A PAL TIAGo robot grasping an object. b) Visualisation of the contact-based grasp representation with visible grasp contact coordinates in the image (u, v), grasp orientation $\mathbf{r}$ and grasp width w.

Applying existing algorithms to mobile manipulators (robots that have a mobile base) often proves difficult. Both the robot and its camera can move, resulting in the need to operate from a variety of viewpoints with respect to the environment and objects to be grasped and manipulated. Furthermore, the environment tends to be dynamic and uncertain and, in particular, the robot's workspace is not known in advance as it is for fixed manipulators. While there exist algorithms that can be used to propose grasps on mobile robots, they typically require some sort of workspace definition to improve the grasp proposals [11] or to run the algorithm itself [8]. Other solutions require wrist-mounted depth cameras for visuomotor control [14] or high-end GPUs to run inference [9].

We present the grasp proposal network (GP-net), a model that can predict 6-DoF grasps on single objects. Unlike other approaches, GP-net operates directly on the input depth image without requiring a prior workspace definition and can thus handle flexible and unknown camera viewpoints. GP-net can be used to grasp novel objects from planar surfaces like tables or furniture units with mobile manipulators. In real-world experiments using the EGAD evaluation benchmark [16] with a PAL TIAGo manipulator, GP-net achieves a grasp success of 54.4%, performing slightly better than the Volumetric Grasping Network (VGN) with 51.6% and substantially outperforming the Grasp Pose Detection package (GPD) with 44.2%.

The contributions of this work can be summarised as follows:

- GP-net: a model to predict 6-DoF grasps for a parallel jaw gripper from flexible viewpoints without specifying the workspace or running table segmentation to filter grasp

poses.
- A ROS package to run GP-net to produce grasp proposals from a depth camera.
- A dataset to train GP-net or alternative network architectures. We make our code available so it can be used to train GP-net to adapt to different robots or grippers.

II. Background and Related Work

Grasping objects in real-world settings involves a lot of variation and uncertainty, including unknown object shapes, unknown poses of the robot and objects, and noisy sensor inputs. To solve this high-dimensional problem, early analytical approaches constrained and simplified the problem by making assumptions about the contacts, friction, and the geometric and physical model of the objects. Consequently, these approaches were usually only tested in simulation with accurate knowledge about the object, robot and environment and without the need to contend with noisy sensor measurements [1].

With the surge of data-driven approaches, analytical methods have been surpassed by machine learning algorithms for grasp synthesis [1]. Initially, these methods typically used 4-DoF top-grasps, where the gripper approaches the object top-down and can only rotate around the approach axis. Furthermore, the initial techniques were usually discriminative [17], where grasp configurations are sampled and subsequently ranked according to a quality metric [2], [3], [18], [19]. While discriminative methods showed good grasping performance, algorithms directly proposing suitable grasp configurations in a generative manner can improve inference time significantly and enable grasping in real-time [4]–[8].

To make robotic grasping usable in less constrained environments, the algorithms must be able to propose 6-DoF grasp poses while coping with varying camera viewpoints and object poses. These requirements increase the search space for possible grasps and, therefore, the problem's difficulty significantly. While modern algorithms can find robust grasp poses in this increased search space, they often suffer from issues such as high latency [20], high computational requirements [9], or limitations to the sampling workspace [8].

Generally, comparing the performance of different methods proves difficult due to the different robots, grippers and objects being used. Usually, algorithms are tested on household objects without the opportunity to reproduce the experiments or compare performance [2], [8]–[11], [20]–[22]. More recently, benchmarks like EGAD [16] using 3D-printed objects have been proposed to generalise the testing of grasping algorithms and improve the comparability and reproducibility of experiments. For this reason, we use EGAD objects in our real-world experiments (see Section V-A).

One of the first methods that could be directly applied to find 6-DoF grasp poses based on point-clouds is the Grasp Pose Detection (GPD) package [11]. It identifies a region of interest, samples grasp proposals, encodes them in a multi-channel image, and finally uses a Convolutional Neural Network to predict the grasp quality. While it is widely used due to its availability in a ROS package, we find that it mainly suggests grasps on the table plane if the table segmentation fails. Further, the discriminative sampling nature of the algorithm leads to high latencies, taking an average of $14.7s$ to propose grasps in our experiments (see Section V-A).

Another family of approaches reconstructs the grasping surface into meshes [10], [20], point clouds [22] or signed distance functions [8], [21], [23] and uses the new representation to predict 6-DoF grasps. One of these approaches, the Volumetric Grasping Network (VGN) [8], exhibits promising results by predicting grasp proposals from a truncated signed distance function (TSDF). Once the TSDF is built, grasp proposals can be predicted in real-time within $10ms$, opening up possibilities for closed-loop control.

VGN achieves a grasp success of 80% using household objects in the original paper. However, the acquisition of the TSDF is achieved by integrating a stream of depth images acquired by a wrist-mounted depth camera, where the camera follows a pre-defined scan trajectory around a defined workspace prior to computing the grasp. When applying VGN to scenarios with an unknown camera-workspace transform, the approximate position of the object has to be estimated in advance to define the workspace around the object. Furthermore, if no wrist-mounted camera is available, the TSDF must be built from the incoming depth images of the head-mounted camera, which affects the quality of the TSDF.

Another recent method, Contact-GraspNet [9], proposes 6-DoF grasps from a single point cloud. The method achieves a grasp success of more than 90% in real-world experiments using a set of household objects. However, the model implementation requires a GPU with $\geq 8GB$ RAM for running inference. Such requirements make Contact-GraspNet unsuitable for usage on mobile robotic platforms that do not have access to high-end dedicated GPUs, as is the case for several currently available mobile manipulators [24], [25].

From the literature, it is clear that few methods can be used directly to identify 6-DoF grasps for unknown objects in unknown poses. In particular, no methods can be run on mobile manipulators without the need to run pre-processing, such as plane segmentation to filter grasp proposals or 3D object detection to identify the workspace, which can potentially lead to erroneous grasp proposals. We propose GP-net to address these issues and provide a ROS package that can be used to generate 6-DoF grasp proposals on mobile manipulators without any additional pre-processing.

III. Grasp Proposal Network

We consider the problem of proposing 6-DoF grasps for a parallel-jaw gripper and using an RGB-D camera. The environment consists of a single object $o \in \mathcal{O}$ placed in a stable resting pose on a planar surface. The camera rests in an unknown, versatile pose near the object facing the planar surface. The goal is to propose a diverse set of 6-DoF grasps $g \in \mathcal{G}$ for the object based on a depth image $\mathcal{I}$.

Similar to Sundermeyer et al. [9], we represent grasps based on the contact point between the gripper and the object. Each pixel (u, v) in a depth image $\mathcal{I}$ describes a potential grasp contact, i.e. the contact of a gripper plate during grasp execution. The full ground-truth grasp is defined as $g \in (u, v, q, \mathbf{r}, w)$,

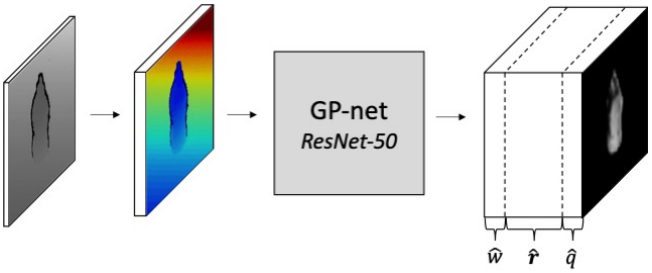

Fig. 2. Using a depth image as input, we apply a jet-colourscale, process the image with GP-net and output pixel-wise grasp proposal predictions with a quality $\hat{q}$, orientation $\hat{\mathbf{r}}$ and grasp width $\hat{w}$.

with the grasp quality q, the orientation of the grasp in camera coordinates $\mathbf{r}$ and the width of a grasp w. A visualisation of the grasp representation is depicted in Figure 1 (b).

We base the network architecture for GP-net on a ResNet-50 model pre-trained on ImageNet. A description of the pipeline and the output tensor of the model can be seen in Figure 2. Since the rendered depth images consist of one channel and the pre-trained ResNet-50 architecture uses 3 input channels, we apply a jet-colourscale to the depth image as described by Eiter et al. [26]. We set fixed normalisation boundaries of $0.4m$ and $1.4m$ to keep the relation to the real distance of a given scene. Note that points further away than $1.4m$ are outside the grasping range of the robot.

GP-net outputs a 6-channel tensor with the same spatial resolution as the input image, $W \times H$. Each pixel in the output tensor represents a grasp whose visible contact point corresponds to that pixel, with the channels containing predictions of the width of the grasp $\hat{w}$, the grasp orientation $\hat{\mathbf{r}}$ in form of a quaternion, and the quality of the grasp $\hat{q}$. Similar to VGN [8], we normalise the quaternions to unit quaternions and apply a sigmoid function to the quality channel.

Loss function: Since creating ground-truth grasp proposals for training at each pixel is computationally infeasible, we generate sparse maps containing grasp information at visible grasp contacts for up to 100 pre-sampled grasps per object. The training data generation process is described in Section IV. We backpropagate the loss only through the output pixels at which we have ground-truth grasp information, similar to VGN [8]. We define our loss function as:

$$\mathcal{L} = \mathcal{L}_q + \mathbb{1}^{PosGrasp}(\alpha \mathcal{L}_{\mathbf{r}} + \beta \mathcal{L}_w) \tag{1}$$

with $\mathcal{L}_q$ being the binary cross-entropy loss between the ground truth and predicted binary quality values, q and $\hat{q}$, $\mathcal{L}_{\mathbf{r}} = 1 - |\mathbf{r} \cdot \hat{\mathbf{r}}|$ the distance metric between the target and predicted quaternions, $\mathbf{r}$ and $\hat{\mathbf{r}}$, as defined in [27] and $\mathcal{L}_w$ the L1 loss between the ground-truth and predicted grasp width, w and $\hat{w}$. Since unsuccessful grasps do not have valid configurations for the model to learn, we include the second part of the loss function only for ground-truth positive grasps indicated as $\mathbb{1}^{PosGrasp}$. We set $\alpha, \beta = 0.1$, which we experimentally found to give a good balance between $\mathcal{L}_q$, $\mathcal{L}_{\mathbf{r}}$ and $\mathcal{L}_w$.

In contrast to the loss function of VGN [8], we only allow one valid configuration for the orientation loss $\mathcal{L}_{\mathbf{r}}$. This change is rooted in the grasp contact representation, which anchors grasps to the visible grasp contact. Since the positive x-axis of the grasp coordinate frame points towards the second grasp contact (see Figure 1b), there remains only one valid orientation. This representation does not affect grasp variability since the parallel-jaw gripper is symmetric.

Using the model output for robotic grasping: To use the output of GP-net for grasping objects with a robot, it must be transformed into grasp proposals. As a first step, a maximum of $j = 10$ grasps are chosen from the output tensor using non-maximum suppression with a peak distance of 4 pixels and an acceptance threshold of $\gamma = 0.4$. Note that the acceptance threshold γ for applying non-maximum suppression defines the minimum predicted quality $\hat{q}$ that will result in a grasp proposal and is chosen based on our ablation studies in Section V-B. Once the image coordinates (u, v) for those grasp contacts are chosen, we re-project them into 3D camera coordinates using the camera intrinsics K and the depth value of (u, v) in the input depth image. Using the predicted quaternion $\hat{\mathbf{r}}$ and the predicted width $\hat{w}$ of each grasp, we translate the grasp contact to the tool centre point by moving them $0.5\hat{w}$ along the grasp x-axis between the gripper plates. As a last step, the grasp is transformed from the camera coordinate frame to the robot base coordinate frame.

Our ROS package provides this functionality by running inference with a trained model, selecting the best grasps and mapping them to the robot base transform. Further, we provide example code to use the package with a pre-trained GP-net model to plan grasps for a PAL TIAGo robot using a parallel jaw gripper. Using other gripper configurations, our code can generate an adjusted dataset, train a new model, and use it with our ROS package to produce grasp proposals.

IV. Training Dataset

To train GP-net, we render a synthetic depth-image-based dataset with sparse ground-truth grasp information. Similar to DexNet2.0 [2], we use the objects from the 3Dnet [28] and KIT [29] mesh datasets. When loading the 3D meshes from the mesh datasets, we re-scale them to fit into TIAGo's gripper, a parallel gripper manufactured by PAL robotics with an opening width of $8cm$ [24]. Re-scaling 3D meshes is commonly used to generate robotic grasping datasets. In contrast to the re-scaling methods in [2], [16], [30], we re-scale objects to a randomly chosen width, drawn uniformly from the range $6cm$ to $10cm$. In this way, our dataset includes objects which do not fit in our robotic gripper, which is a situation that will occur in real-world environments.

After re-scaling each object o, we calculate up to 25 stable resting poses $\mathcal{S}(o)$ and use the antipodal grasp sampler proposed in [2] to generate up to 100 parallel-jaw grasps $\mathcal{G}(o)$. Grasps are sampled by randomly choosing a surface point from the mesh and then sampling the grasp x-axis (see Figure 1b) based on the friction cone of the surface point. This procedure does not necessarily yield the best grasp for a grasp contact point. Our dataset should contain the best available grasps for a contact since we use it to train a network for proposing good grasps. Therefore, we modify the sampling procedure

to generate $k = 6$ potential grasps for a given contact. We calculate the robust force closure metric ϵ [31]–[33] for each of the sampled grasps at one contact point and keep the grasp with the highest ϵ. Using this method, we generate a total of $148,706$ ground-truth grasps for the object meshes in our dataset.

Since the robust force closure metric ϵ of a sampled ground-truth grasp depends on the grasp contacts, it is defined by the grasp x-axis and consistent for any grasp approach axis, i.e. the direction of the z-axis in Figure 1. However, collisions between the gripper, object and planar surface further influence the success of a grasp. We define the quality q of a grasp in our rendered dataset as

$$q(g) = \begin{cases} 1 & \epsilon \geq \delta \text{ and } coll\_free(g) \\ 0 & otherwise \end{cases} \tag{2}$$

with $\delta = 0.5$ being the robustness threshold and $coll\_free(g)$ indicating if a grasp is collision free, similar to DexNet2.0 [2].

We aim to choose reproducible grasp orientations for a given scene and ground-truth grasp. To achieve this, we apply the following steps for selecting the grasp approach axis orientation for a given grasp: To check for collisions, we hinge-rotate the grasps in steps of $\Delta\omega = 15\,\text{deg}$ around the contact points and thereby the grasp x-axis (see Figure 1b). If we have collision-free grasps within those orientations, we set the ground-truth grasp orientation as the median collision-free grasp. If we have multiple collision-free regions, we choose the median collision-free grasp whose approach is most closely aligned with the principal ray of the camera, e.g. approaching from the front of the object rather than from behind the object. With these steps, we fully define each grasp orientation in our dataset in a reproducible way.

We render $n = 20$ images for each stable pose $s \in \mathcal{S}(o)$ of each object $o \in \mathcal{O}$ from camera poses selected uniformly at random as described in VGQ-CNN [34]. We then project the grasp contacts into the image plane and calculate the image coordinates of the visible contact (u, v). If multiple collision-free grasp contacts are visible at one pixel, we choose the grasp with the highest robust force closure value ϵ. In addition, we store the binary segmentation mask of the object and use the pixels not showing the object as ground-truth negative grasp contacts during training. The full dataset consists of $260,340$ images with an average of 88.1 grasps per image, totalling $22,944,376$ grasps.

V. Experiments

We train GP-net for 20 epochs using an Adam optimiser with a learning rate of $3e^{-4}$ and a batch size of 32. To reduce the sim-to-real-gap for GP-net, we simulate depth-camera noise on our depth images using the noise model described in [35], [36]. We find that compared to the Gaussian noise model suggested in DexNet2.0 [2], the added depth-camera noise model substantially improves the robustness of GP-net in real-world scenarios.

A. Results

We evaluate the performance of GP-net with simulation and real-world experiments using objects from the EGAD evaluation benchmark [16] not seen by the network during training. The simulation analysis is used to validate our model and conduct the ablation studies in Section V-B since it requires less time and fewer resources than tests with a real robot. For the real-world experiments, we use GP-net on a PAL TIAGo mobile manipulator to grasp the 49 3D printed objects from the EGAD evaluation benchmark [16]. We repeat the real-world experiments with VGN [8] and GPD [37] and compare them to the performance of GP-net.

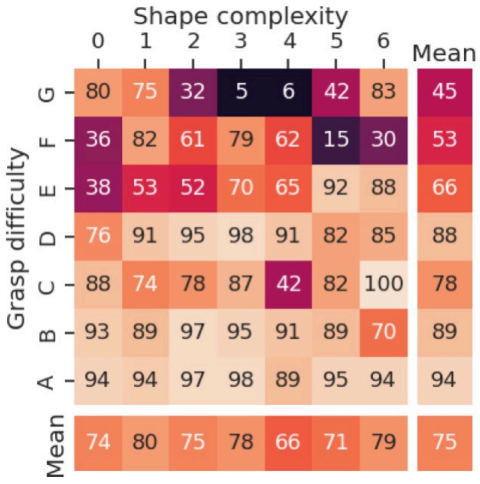

Fig. 3. Grasp success [%] in simulation using GP-net to grasp objects from EGAD with a PAL parallel jaw gripper.

Simulation experiments: We test the grasp success of the grasps proposed by GP-net in simulation using the pybullet [38] physics engine based on the simulation environment developed for VGN [8]. A parallel jaw gripper is simulated to approach the grasp pose linearly along the grasp z-axis for $0.15m$, close the gripper using a maximum force of $5N$, and lift the object. The grasp is labelled as successful if the gripper can lift the object $0.1m$ above the planar surface. For each trial, one of the 49 objects in the EGAD evaluation benchmark is placed in a random position within the workspace within a $30 \times 30cm^2$ workspace on a planar surface. Then, a depth image is rendered from a camera pose sampled uniformly at random from the spherical coordinates around the centre of the workspace with the sampling bounds used for training in Section IV. We run inference with GP-net on the depth image with added depth-camera noise and map the output tensors to grasp proposals as described in Section III.

We simulate the grasp with the highest predicted grasp quality $\hat{q}$ proposed by the model for one trial. Each experimental run comprises 100 such trials for each object from the EGAD [16] evaluation set. In the simulation, GP-net achieves a mean grasp success of 74.6% across all objects of the EGAD dataset. The grasp success on each object is depicted in Figure 3. The shape complexity and grasp difficulty of each object are defined in the EGAD evaluation set. Each object is named according to its shape complexity and grasp difficulty, e.g. object "A0" with grasp difficulty "A" and shape complexity "0". The grasp success decreases with increasing grasp difficulty, while it is consistent across the levels of shape

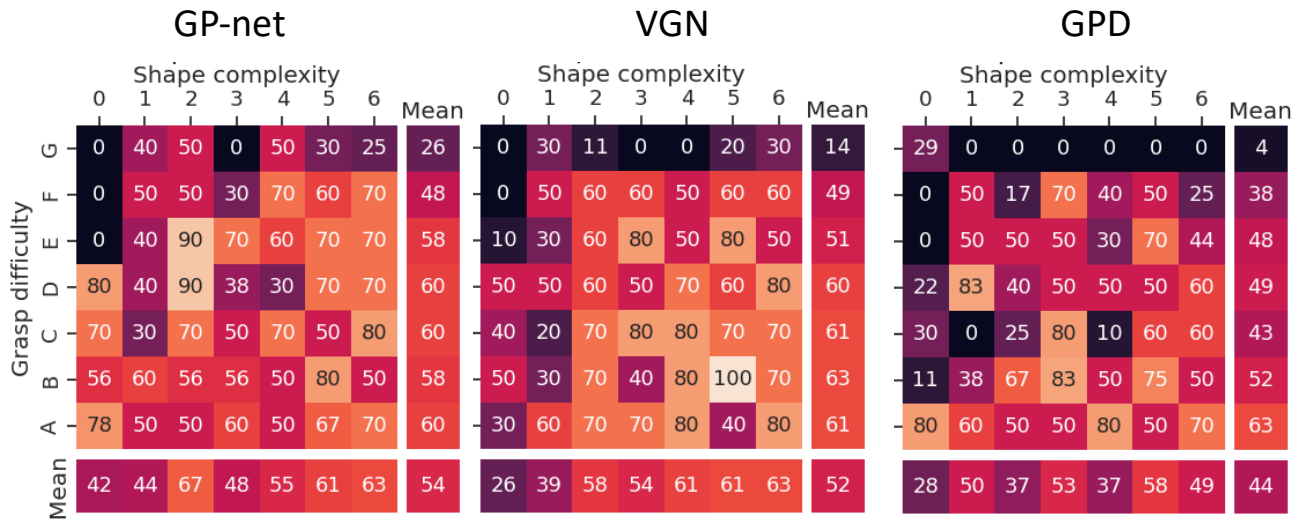

Fig. 4. Grasp success [%] out of 10 grasp attempts per object on the EGAD benchmark using GP-net, VGN and GPD for grasp proposal.

complexity.

Real-world experiments: We test the grasp success in real-world experiments with a PAL TIAGo mobile manipulator [24]. We use an Intel Realsense D435 depth camera mounted on TIAGo's head since it has a lower minimum depth distance, and we find it to give a better performance on thin features than the Orbbec Astra used in the TIAGo mobile manipulator by default. We average over ten consecutive depth image frames for noise reduction for all tested methods.

We use the same 49 evaluation objects from the EGAD dataset [16] as in our simulation analysis for the experiments. For each object, a total of 10 grasp trials are executed, resulting in 490 grasp attempts for each algorithm. We use two different initial robot poses for grasp-planning, each having a different head tilt and torso height and, thereby, a different camera viewpoint of the scene. The object is dropped within a $30 \times 30cm^2$ square on the table by a human operator before each grasp trial.

We choose VGN [8] and GPD [37] to compare to GP-net in real-world experiments, given their use as points of comparison by other researchers [21], [23]. To apply VGN to our experiments, we define the pose of the $30 \times 30 \times 30cm^3$ workspace for grasp sampling in front of the robot manually to sit on the table. Note that VGN was originally intended to be used with a wrist-mounted depth camera performing a grasp scan along a trajectory to build the TSDF. Since our work focuses on proposing grasps from a single viewpoint, we instead apply VGN by building the TSDF from a single, noise-reduced depth image.

To apply GPD to our experiments, we re-project a point cloud from a single, noise-reduced depth image and crop it to reduce the number of points and improve run-time. Further, the object points in the point cloud are indexed to indicate where grasps should be sampled. The indexing of the point cloud is achieved by fitting a plane to the point cloud and indexing points with a distance $\geq 0.005m$ to the table plane. While the approaches for setting the workspace for VGN and indexing the point cloud for GPD work for our experimental setting, more general applications would require more sophisticated workarounds to prevent erroneous grasp proposals. GP-net does not need any workspace definition and can be directly applied to the depth image.

	GP-net	VGN [8]	GPD [37]
Grasp success [%]	**54.4**	51.6	44.2
Inference time [s]	2.1	**1.2**	14.7
Grasp planning time [s]	**2.7**	4.9	19.1

TABLE I
GRASP SUCCESS, INFERENCE, AND GRASP PLANNING TIME FOR OUR REAL-WORLD EXPERIMENTS. INFERENCE REFERS TO THE TIME BETWEEN NETWORK INPUT AND OUTPUT, WHILE GRASP PLANNING ADDITIONALLY INCLUDES PRE-PROCESSING AND POST-PROCESSING.

We run the algorithms for all methods on an Intel i7-10750H CPU and NVIDIA RTX 2060 GPU. The results are shown in Figure 4. GP-net performs slightly better than VGN and substantially better than GPD with a grasp success of 54.4% compared to their 51.6% and 44.2%, respectively. We report each network's overall results and timings in Table I. In our experiments, GP-net takes 2.1s for proposing grasps on a given depth image, while VGN and GPD require 1.2s and 14.7s, respectively. These times do not include additional pre-processing steps like table segmentation and workspace definition, which are necessary for VGN and GPD. Due to this, for the overall grasp planning time, which includes pre-processing, post-processing and sending data between different ROS nodes, GP-net is faster than VGN and GPD with $2.7s$ compared to $4.9s$ and $19.1s$, respectively.

Note that experimental results are sensitive to the hardware and software used and thereby make it difficult to compare methods if not tested in the same setting. Further, real-world experiments are expected to result in a lower grasp success than our simulation results, which are simulated only with the end-effector. These performance differences are rooted in perception and actuation uncertainties, collision checking, and the performance of the path planning algorithms.

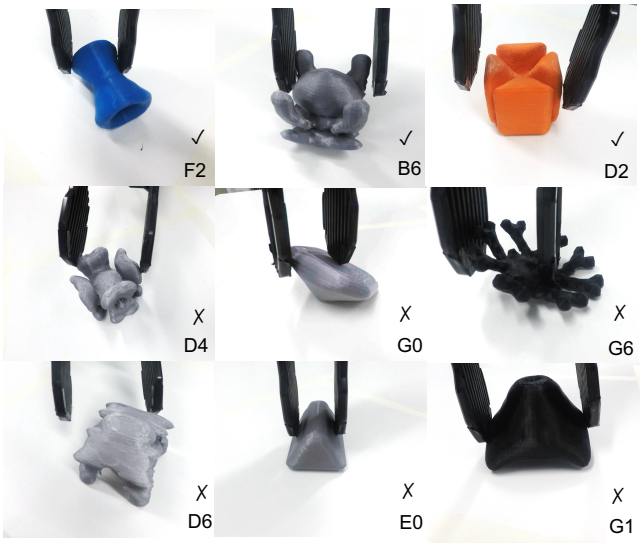

Fig. 5. Success and failure cases of using GP-net to propose grasps on objects of the EGAD evaluation set with a PAL TIAGo mobile manipulator. We label each figure according to the corresponding object in the dataset and indicate success with ✓and failures with X.

We have encountered different failure cases when testing GP-net in our real-world experiments. An overview of different successful grasp attempts and failures is shown in Figure 5. One type of failure occurs when the gripper can not close fully due to obstructions by itself or the object, e.g. in the examples of objects "G0" and "G6". The object will lose contact with the gripper once lifted, and the grasp attempt will fail.

In other failure cases, the gripper pushes the object away when approaching due to perception and actuation uncertainties, as seen in the example of objects "D4" and "D6". This type of failure is hardware-dependent and not necessarily rooted in the grasp proposal algorithm. Further, it could be prevented when using closed-loop control, e.g., using an RGB camera mounted on the gripper and correcting the grasp approach. The grasps shown with objects "E0" and "G1" fail because the friction between the objects and the gripper is too low to compensate for the slanted surfaces that are being grasped. Note that grasping "E0" longitudinally is not an option, as this side of the object is longer than the gripper width, and therefore, "E0" fails to be grasped in most cases.

B. Ablation studies

We conduct a set of ablation studies to investigate the performance of GP-netfurther. We investigate how an alternative grasp representation defining the Tool Centre Point (TCP) performs compared to the contact-based grasp representation (see Figure 1b) used for GP-net. Further, we run simulation experiments and investigate how different acceptance thresholds γ for the non-maximum suppression influence the grasp success and the final number of grasp proposals.

Grasp representation: The grasp representation used in GP-net is contact-based, with each pixel representing a visible grasp contact of a potential grasp, see Section III. This grasp representation was first proposed in Contact-Graspnet [9], suggesting it facilitates the learning process by reducing dimensionality. However, the grasp representation in [9] was not compared to conventional, TCP-based grasp representations as used in most of the related work [2]–[8], [10]–[12].

While the contact-based grasp representation can potentially increase the model performance, using this grasp representation can lead to problems where no grasp contact is visible. For example, if a robot is supposed to grasp a book but is facing the spine of the book head-on. In this situation, the areas for potential grasp contacts on the left and right sides are not visible, and a contact-based method would not be able to produce valid grasp configurations. Here, a repositioning of the camera would be necessary.

For this reason, we compare the performance of a contact-based grasp representation to a TCP-based grasp representation. We modify GP-net to use a TCP-based grasp representation as $g \in (u, v, z, q, \mathbf{r}, w)$ with (u, v) being the image coordinates of the grasp centre and z being the distance between the grasp centre and the visible surface depth at (u, v). We train the model on our dataset as described in Section V-A and compare performance to GP-net using the simulation analysis. Due to the extra variable z in the grasp representation, the model has seven output channels, and we define the adapted loss function as:

$$\mathcal{L} = \mathcal{L}_q + \mathbb{1}^{PosGrasp}(\alpha\mathcal{L}_{\mathbf{r}} + \beta\mathcal{L}_w + \nu\mathcal{L}_z) \tag{3}$$

with $\mathcal{L}_z$ being the L1 loss between ground-truth z and predicted grasp distance $\hat{z}$. We set $\alpha, \nu = 0.1$ and $\beta = 0.01$ since the width is not necessary to predict the grasp position for the TCP-based grasp representation. After training the model for 20 epochs, we achieve a mean grasp success of 27.2% in simulation, substantially lower than GP-net's 74.6% with the contact-based grasp representation. We find that the orientation loss $\mathcal{L}_{\mathbf{r}}$ plateaus almost double that of GP-net's loss with 0.24 for the TCP-based grasp representation and 0.13 with the contact-based grasp representation.

We hypothesise that this difference is rooted in the reduction of ambiguities when grasps are anchored to their grasp contact points. Ground-truth positive grasps would have similar orientations for neighbouring grasp contacts on an object, while TCP-based grasps could approach the same TCP position from various angles. This is especially apparent when looking at cylindrical objects, where a TCP-based ground truth positive grasp could rotate the grasp axis to any orientation around the centre of the cylinder. In contrast, a contact-based ground truth positive grasp only permits one orientation such that the x-axis of the grasp passes through the centre of the cylinder (see Figure 1b).

Acceptance threshold γ: The acceptance threshold γ defines the minimum predicted grasp quality $\hat{q}$ of output grasp proposals for GP-net. Non-maximum suppression with γ as a cut-off point applied to the predicted grasp quality $\hat{q}$ yields the output grasp proposals. As such, γ balances the number of proposed grasps and the confidence the model has in those grasps. Ideally, γ should be set to propose as many grasps as possible while retaining good confidence in the quality of those grasps. To investigate how this trade-off affects GP-net, we apply a range of acceptance thresholds

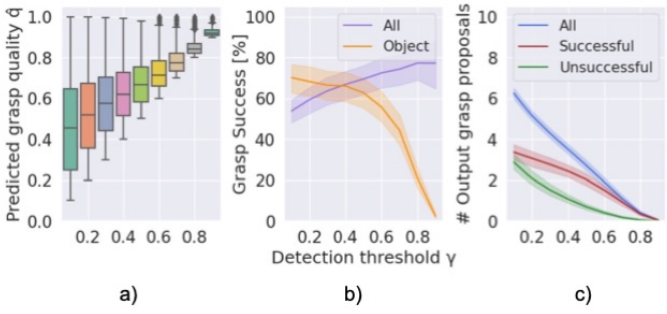

Fig. 6. Simulation results showing the influence of varying the acceptance threshold γ on a) the predicted quality $\hat{q}$ of grasp proposals, b) the grasp success and c) the number of output grasp proposals per object with a 95% confidence interval.

$\gamma = [0.1, 0.2, \ldots, 0.9]$ and investigate the influence on grasp performance in simulation.

The results can be seen in Figure 6. Increasing γ increases the predicted grasp quality $\hat{q}$ in Figure 6a) and decreases the number of output grasp proposals in Figure 6c). The number of unsuccessful grasp proposals decreases non-linearly when increasing γ. We show both the grasp success of all proposed grasps and the grasp success for each object in Figure 6b). Note that the grasp success for each object uses only the output grasp proposal with the highest predicted quality $\hat{q}$ and reports an unsuccessful grasp if there is no grasp proposal, as explained in Section V-A. The two curves show the trade-off between not predicting any grasp and thereby failing to pick up the object, i.e. a low grasp success for the object, and the success of all grasps proposals, i.e. a lower grasp success on all grasp proposals if many unsuccessful grasps are proposed.

To balance the number of grasps proposed with the success of those grasps, we set $\gamma = 0.4$ for all of our other experiments.

VI. Discussion and Conclusion

In this paper, we presented GP-net, a model to propose 6-DoF grasps based on depth images from flexible and unknown viewpoints. In contrast to widely used algorithms for robotic grasping like GPD [11] and VGN [8], the viewpoint flexibility enables GP-net to be used without the need to predefine a workspace or filter grasps to produce good grasp proposals. Hence, GP-net can be used on mobile robots without additional pre-processing steps like plane-fitting, table segmentation or 3D object detection. To train GP-net, we created a synthetic dataset based on more than 1400 object meshes with ground-truth grasp information.

We evaluated GP-net using a PAL TIAGo mobile manipulator in real-world experiments and compared its performance to GPD and VGN on the EGAD evaluation benchmark [16]. GP-net achieved a grasp success of 54.4%, slightly better than VGN's 51.6% and substantially higher than GPD's 44.2% in our setup while not requiring any of the pre-processing steps for workspace definition and grasp filtering necessary for VGN and GPD.

The contact-based grasp representation utilised in GP-net can not represent grasps without a visible grasp contact. This limitation is not existent in TCP-based grasp representations. To weigh the benefits of a contact-based grasp representation against these limitations, we compare it to a TCP-based grasp representation in our ablation studies in Section V-B. We find a significant performance advantage of a model trained with the contact-based representation.

When using a mobile manipulator in real-world scenarios, the robot is able to move the camera and view objects from different perspectives. In this process, previously invisible grasp contacts can be made visible. Under these circumstances, we think the benefits of a contact-based representation outweigh its limitations.

Similar to most of the commonly used methods [8]–[10], a limitation of GP-net is the gripper-dependence of our dataset and hence the model. Since the quality of grasps depends on the grasp being collision-free, the design of the gripper is used implicitly during dataset generation when checking for collisions, see Eq. 2. The resulting model learns the implicit gripper design, and hence, the performance with alternative gripper configurations cannot be guaranteed.

VGN and GPD work in cluttered environments, while GP-net is trained for single objects. In this work, we proved the method of our approach to propose 6-DoF grasps from versatile, unknown viewpoints without workspace definition. In future work, we plan to extend GP-net to cluttered scenes, which has been shown to work for image-based grasp quality prediction algorithms [39]. Further, we aim to base this extension of GP-net to be used in more diverse and realistic scenarios with various furniture units, e.g. desks, sideboards or shelves.

References

[1] J. Bohg, A. Morales, T. Asfour, and D. Kragic, "Data-Driven Grasp Synthesis—A Survey," *IEEE Transactions on Robotics*, vol. 30, no. 2, pp. 289–309, 4 2014.

[2] 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 *Robotics: Science and Systems (RSS)*, 2017.

[3] J. Zhang, M. Li, Y. Feng, and C. Yang, "Robotic grasp detection based on image processing and random forest," *Multimedia Tools and Applications*, vol. 79, 4 2020.

[4] V. Satish, J. Mahler, and K. Goldberg, "On-Policy Dataset Synthesis for Learning Robot Grasping Policies Using Fully Convolutional Deep Networks," *IEEE Robotics and Automation Letters*, 2019.

[5] D. Morrison, P. Corke, and J. Leitner, "Learning robust, real-time, reactive robotic grasping," *The International Journal of Robotics Research*, vol. 39, no. 2-3, pp. 183–201, 2019. [Online]. Available: https://doi.org/10.1177/0278364919859066

[6] J. Redmon and A. Angelova, "Real-time grasp detection using convolutional neural networks," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 1316–1322.

[7] S. Kumra, S. Joshi, and F. Sahin, "Antipodal Robotic Grasping using Generative Residual Convolutional Neural Network," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 9626–9633.

[8] M. Breyer, J. J. Chung, L. Ott, S. Roland, and N. Juan, "Volumetric Grasping Network: Real-time 6 DOF Grasp Detection in Clutter," in *Conference on Robot Learning*, 2020.

[9] M. Sundermeyer, A. Mousavian, R. Triebel, and D. Fox, "Contact-GraspNet: Efficient 6-DoF Grasp Generation in Cluttered Scenes," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 13 438–13 444.

[10] J. Lundell, F. Verdoja, and V. Kyrki, "Beyond Top-Grasps Through Scene Completion," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 545–551.

[11] A. ten Pas, M. Gualtieri, K. Saenko, and R. Platt, "Grasp Pose Detection in Point Clouds," *The International Journal of Robotics Research*, vol. 36, no. 13-14, pp. 1455–1473, 2017. [Online]. Available: https://doi.org/10.1177/0278364917735594

[12] L. Berscheid, C. Friedrich, and T. Kröger, "Robot Learning of 6 DoF Grasping using Model-based Adaptive Primitives," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 4474–4480.

[13] 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," *The International Journal of Robotics Research*, vol. 37, no. 4-5, pp. 421–436, 2018. [Online]. Available: https://doi.org/10.1177/0278364917710318

[14] U. Viereck, A. Pas, K. Saenko, and R. Platt, "Learning a visuomotor controller for real world robotic grasping using simulated depth images," in *Conference on Robot Learning*, 2017, pp. 291–300.

[15] J. Ichnowski*, Y. Avigal*, J. Kerr, and K. Goldberg, "Dex-NeRF: Using a Neural Radiance field to Grasp Transparent Objects," in *Conference on Robot Learning (CoRL)*, 2020.

[16] D. Morrison, P. Corke, and J. Leitner, "EGAD! An Evolved Grasping Analysis Dataset for Diversity and Reproducibility in Robotic Manipulation," *IEEE Robotics and Automation Letters*, vol. 5, no. 3, pp. 4368–4375, 2020.

[17] K. Kleeberger, R. Bormann, W. Kraus, and M. F. Huber, "A survey on learning-based robotic grasping," *Current Robotics Reports*, pp. 1–11, 2020.

[18] I. Lenz, H. Lee, and A. Saxena, "Deep learning for detecting robotic grasps," *The International Journal of Robotics Research*, vol. 34, no. 4-5, pp. 705–724, 2015. [Online]. Available: https://doi.org/10.1177/0278364914549607

[19] U. Asif, J. Tang, and S. Harrer, "EnsembleNet: Improving Grasp Detection using an Ensemble of Convolutional Neural Networks." in *BMVC*, 2018, p. 10.

[20] J. Varley, C. DeChant, A. Richardson, J. Ruales, and P. Allen, "Shape completion enabled robotic grasping," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 2442–2447.

[21] Z. Jiang, Y. Zhu, M. Svetlik, K. Fang, and Y. Zhu, "Synergies Between Affordance and Geometry: 6-DoF Grasp Detection via Implicit Representations," *Robotics: science and systems*, 2021.

[22] W. Chen, H. Liang, Z. Chen, F. Sun, and J. Zhang, "TransSC: Transformer-based Shape Completion for Grasp Evaluation," 2021.

[23] J. Cai, J. Su, Z. Zhou, H. Cheng, Q. Chen, and M. Y. Wang, "Volumetric-based Contact Point Detection for 7-DoF Grasping," in *Proceedings of The 6th Conference on Robot Learning*. PMLR, 3 2023, pp. 824–834. [Online]. Available: https://proceedings.mlr.press/v205/cai23a.html

[24] P. A. L. Robotics, "TIAGo technical specifications," 2021. [Online]. Available: https://pal-robotics.com/wp-content/uploads/2021/07/Datasheet-complete_TIAGo-2021.pdf

[25] T. Yamamoto, T. Nishino, H. Kajima, M. Ohta, and K. Ikeda, "Human Support Robot (HSR)," in *ACM SIGGRAPH 2018 Emerging Technologies*, ser. SIGGRAPH '18. New York, NY, USA: Association for Computing Machinery, 2018. [Online]. Available: https://doi.org/10.1145/3214907.3233972

[26] A. Eitel, J. T. Springenberg, L. Spinello, M. Riedmiller, and W. Burgard, "Multimodal deep learning for robust RGB-D object recognition," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015, pp. 681–687.

[27] J. J. Kuffner, "Effective sampling and distance metrics for 3D rigid body path planning," in *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04. 2004*, vol. 4, 2004, pp. 3993–3998.

[28] W. Wohlkinger, A. Aldoma, R. B. Rusu, and M. Vincze, "3DNet: Large-scale object class recognition from CAD models," in *2012 IEEE International Conference on Robotics and Automation*, 2012, pp. 5384–5391.

[29] A. Kasper, Z. Xue, and R. Dillmann, "The KIT object models database: An object model database for object recognition, localization and manipulation in service robotics," *The International Journal of Robotics Research*, vol. 31, no. 8, pp. 927–934, 2012. [Online]. Available: https://doi.org/10.1177/0278364912445831

[30] C. Eppner, A. Mousavian, and D. Fox, "ACRONYM: A Large-Scale Grasp Dataset Based on Simulation," in *2021 IEEE Int. Conf. on Robotics and Automation, ICRA*, 2020.

[31] D. Seita, F. T. Pokorny, J. Mahler, D. Kragic, M. Franklin, J. Canny, and K. Goldberg, "Large-scale supervised learning of the grasp robustness of surface patch pairs," in *2016 IEEE International Conference on Simulation, Modeling, and Programming for Autonomous Robots (SIMPAR)*, 2016, pp. 216–223.

[32] J. Mahler, F. T. Pokorny, B. Hou, M. Roderick, M. Laskey, M. Aubry, K. Kohlhoff, T. Kröger, J. Kuffner, and K. Goldberg, "Dex-net 1.0: A cloud-based network of 3d objects for robust grasp planning using a multi-armed bandit model with correlated rewards," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2016, pp. 1957–1964.

[33] J. Weisz and P. K. Allen, "Pose error robust grasping from contact wrench space metrics," in *2012 IEEE International Conference on Robotics and Automation*, 2012, pp. 557–562.

[34] A. Konrad, J. McDonald, and R. Villing, "VGQ-CNN: Moving Beyond Fixed Cameras and Top-Grasps for Grasp Quality Prediction," *Proceedings of the International Joint Conference on Neural Networks*, vol. 2022-July, 2022.

[35] A. Handa, T. Whelan, J. McDonald, and A. J. Davison, "A benchmark for RGB-D visual odometry, 3D reconstruction and SLAM," *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 1524–1531, 9 2014.

[36] J. T. Barron and J. Malik, "Intrinsic Scene Properties from a Single RGB-D Image," *CVPR*, 2013.

[37] A. Ten Pas and R. Platt, "Using geometry to detect grasp poses in 3d point clouds," in *Robotics Research*. Springer, 2018, pp. 307–324.

[38] E. Coumans and Y. Bai, "Pybullet, a python module for physics simulation for games, robotics and machine learning," 2016.

[39] J. Mahler and K. Goldberg, "Learning Deep Policies for Robot Bin Picking by Simulating Robust Grasping Sequences," in *Proceedings of the 1st Annual Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, S. Levine, V. Vanhoucke, and K. Goldberg, Eds., vol. 78. PMLR, 4 2017, pp. 515–524.