Efficient End-to-End Detection of 6-DoF Grasps for Robotic Bin Picking

Yushi Liu[1,2] Alexander Qualmann[1] Zehao Yu[2] Miroslav Gabriel[1]
Philipp Schillinger[1] Markus Spies[1] Ngo Anh Vien[1] Andreas Geiger[2]

***Abstract*— Bin picking is an important building block for many robotic systems, in logistics, production or in household use-cases. In recent years, machine learning methods for the prediction of 6-DoF grasps on diverse and unknown objects have shown promising progress. However, existing approaches only consider a single ground truth grasp orientation at a grasp location during training and therefore can only predict limited grasp orientations which leads to a reduced number of feasible grasps in bin picking with restricted reachability. In this paper, we propose a novel approach for learning dense and diverse 6-DoF grasps for parallel-jaw grippers in robotic bin picking. We introduce a parameterized grasp distribution model based on Power-Spherical distributions that enables a training based on all possible ground truth samples. Thereby, we also consider the grasp uncertainty enhancing the model's robustness to noisy inputs. As a result, given a single top-down view depth image, our model can generate diverse grasps with multiple collision-free grasp orientations. Experimental evaluations in simulation and on a real robotic bin picking setup demonstrate the model's ability to generalize across various object categories achieving an object clearing rate of around 90% in simulation and real-world experiments. We also outperform state of the art approaches. Moreover, the proposed approach exhibits its usability in real robot experiments without any refinement steps, even when only trained on a synthetic dataset, due to the probabilistic grasp distribution modeling.**

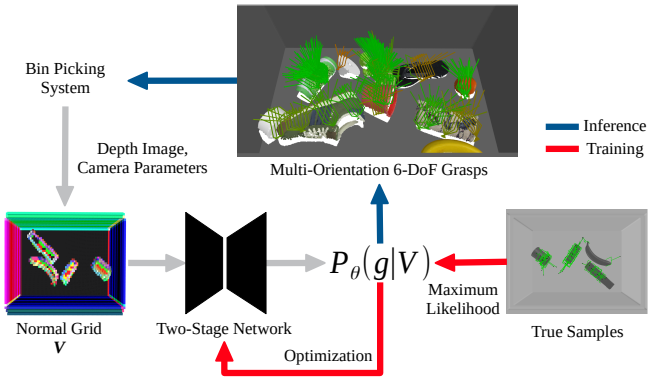

Fig. 1: Schematic overview on training and inference pipelines of the proposed 6-DoF grasp prediction model. The network is trained to predict the optimal parameter θ of the grasp distribution $P_\theta(g|V)$, maximizing the likelihood of ground truth grasp samples; During inference, the model infers grasps with multiple orientations including graspability and collision scores. Green forks indicate 6-DoF grasps.

I. INTRODUCTION

Grasp detection is an essential problem for the automation of pick-and-place tasks in industry and logistics, where robots are used to grasp and manipulate objects in unstructured environments. Recent research applies machine learning methods to enable model-free grasp detection for parallel-jaw grippers on diverse and previously unknown objects. Some approaches address the 4-DoF grasp problem [1], [2], primarily considering 3D position and gripper orientation about a gravity-aligned approach vector. While this simplifies the grasping problem, it requires assumptions on the gripper orientation, leading to a top-down grasp execution limitation. Other approaches propose end-to-end 6-DoF grasp detection learning based on a deep neural network architecture for feature extraction from 3D input data [3], [4], [5], [6], [7]. Even though these approaches directly output multiple complete gripper poses, they only predict one gripper orientation at a location an thus neglect other potential gripper orientations. Jeng. et. al. [8] propose an end-to-end network predicting confidence scores for various grasp orientations, along with refinement values for diverse gripper orientations per grasp point. However, this approach requires a specific fine-tuning process of grasp poses. Furthermore, current methods only consider a single ground truth grasp orientation at a grasp point during training which then limits the number of feasible grasps during inference and can reduce the grasp performance in bin picking with additional reachability restrictions. In this paper, we propose a novel approach for predicting dense parallel-jaw grasps in cluttered bin picking scenarios from single-view observations. Our method predicts multiple orientations per contact point based on grasp distribution modeling.

Our contributions are as follows: (1) A novel objective function based on a distribution modeling and integrating uncertainty learning for different grasp orientations at each contact point. The function enables learning with multiple diverse true grasp samples per contact point and enhances the method's robustness to noisy input. (2) A two-stage end-to-end trainable network for 6-DoF grasp prediction, which generates dense and collision-free grasps for a single-view depth image including multiple grasp orientations at each contact point. (3) Experimental evaluation in simulation and on a real robot bin picking setup (see Fig. 6). The results show that our method outperforms state of the art baseline methods and demonstrates the adaptability of the approach to real robot bin picking setups, even when only trained on a synthetic dataset, due to the integrated probabilistic grasp distribution modeling.

[1]Bosch Center for Artificial Intelligence, Renningen, Germany.
[2]University of Tuebingen, Tuebingen AI Center, Germany.

II. Related Work

Grasp detection is a widely researched topic, with notable surveys like [9] and [10]. In this section, we focus on the parts of this research most relevant to our work.

A. Data-driven 6-DoF Grasp Detection

Recent 6-DoF grasp detection techniques directly regress grasp poses using scene sensor data. Many methods treat each surface point as a grasp center and predict grasp parameters for every voxel center [5], [6], point [3], [4], or pixel [11]. This approach can be inefficient when the input space significantly exceeds the grasp translation space. Another common strategy [12], [13], [14], [15] uses multi-stage networks to simplify the 6-DoF grasp representation, with each stage predicting specific grasp parameters. These methods typically consider only the closest ground truth grasp during training, limiting the range of predicted orientations. Our method, in contrast, first identifies contact point candidates and then predicts grasp parameters based on these points. Additionally, we define a local grasp orientation distribution for each contact point. This allows our model to train on various grasp labels, producing multiple grasp orientations for each contact. While the work by Jeng et al.[8] is similar in that it also generates multiple orientations per grasp center using a coarse-to-fine approach, our method stands out by directly outputting accurate grasp poses for each contact point using local grasp distribution modeling, eliminating the need for refinement.

B. Scene Representation

Scene representation is crucial for efficient grasp prediction. The Truncated Signed Distance Function (TSDF) is a popular choice due to its ability to represent near-surface information. However, generating a comprehensive TSDF grid requires multi-view depth or RGB inputs [5], [7], [16], which is challenging in bin picking contexts. In contrast, TSDF grids from monocular view often miss essential geometric details, e.g. the contact surface. While auxiliary tasks such as shape completion or 3D reconstruction can improve predictions [6], we prioritize reconstructing spaces crucial for grasping, including the contact region, to boost training efficiency. As pointed out in [17], [18], [19], [20] that surface normals is of great importance for predicting grasp orientation, our model utilizes a normal grid where each voxel encoding a surface normal.

C. Grasp Pose Representation

Recent research favors the 6-DoF grasp representation due to its ability to offer greater grasping versatility. These methods primarily utilize points from the point cloud or voxel centers as contact candidates for the gripper's end-point position. [21] directly maps grasp poses in SE(3), employing grasp translation and a unit quaternion for rotation. Other methods [19], [5], [6], [22] adopt a similar representation to predict point-wise grasp configurations. Additionally, methods like [3], [12], [23] use Gram-Schmidt orthonormalization to define grasp poses with an approach and rotation vector. GDN [8] proposes learning multiple approach vectors for each feature volume. However, these methods predominantly represent only one grasp per contact point. In contrast, our approach constructs a parameterized grasp orientation distribution at each grasp contact point using a Power-Spherical distribution, enabling infinite ways to grasp objects at a single contact point.

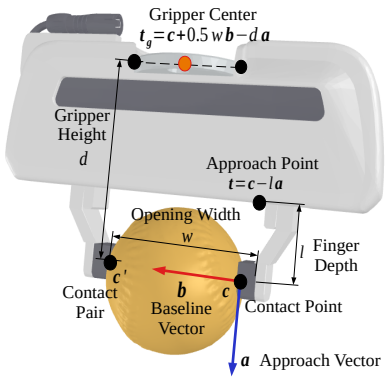

Fig. 2: Contact grasp.

III. Problem Statement

We consider the problem of 6-DoF grasp detection for random bin picking scenes with cluttered and unknown objects located inside a bin and captured by a single top-down view camera depth image.

A. Notations

3D Normal Grid: We define a normal grid, denoted as $\mathbf{V}$, as an N^3 voxel grid where each voxel that is close to the object surface stores the normal vector corresponding to the nearest surface point relative to the voxel's center. To create this 3D normal grid, we initially transform the raw depth image into a truncated signed distance field (TSDF). Subsequently, we extract surface normals for voxel centers located closely to the object surface. These extraction techniques are based on methods described in prior works such as [24], [25], [26].
Contact Grasp: We use the contact grasp representation, i.e. the Gram-Schmidt orthonormalization, for parallel-jaw grippers proposed in [3]. Each contact grasp g at a contact point $\mathbf{c} \in \mathbb{R}^3$ is represented as a 4-tuple:

$$g = (\mathbf{c}, \mathbf{b}, \mathbf{a}, w), \tag{1}$$

where $\mathbf{b} \in \mathbb{R}^3$ is the grasp baseline vector, $\mathbf{a} \in \mathbb{R}^3$ is the grasp approach vector, and $w \in \mathbb{R}^+$ is the grasp opening width of the parallel-jaw gripper (Fig. 2).
Approach Point: A reference point $\mathbf{t}$ of the gripper hand location (see Fig. 2). We select this point because it is close to the contact point and related to approach vector. We also observed that learning its distribution is beneficial for predicting collision-free approach directions.
Antipodal Quality: A metric q measuring grasp stability by calculating the *antipodality* [1], [19], [27] of the contact pair $(\mathbf{c_1}, \mathbf{c_2})$ of a parallel gripper, which is defined as

$$q(\mathbf{c_1}, \mathbf{c_2}) = |\cos(\mathbf{b}, \mathbf{n}(\mathbf{c_1})||\cos(\mathbf{b}, \mathbf{n}(\mathbf{c_2})|, \tag{2}$$

where $\mathbf{b}$ is the baseline vector, $\mathbf{n}(\cdot)$ computes the surface normal vectors at two contact points.

Grasp Configuration: An output representation of our network, representing a set of grasps with different approach directions at a contact point $\mathbf{c}$ (Fig. 3). We denote it as C and formulate it as

$$C = (\mathbf{c}, \mathbf{b}, \{\mathbf{a}_i\}_{i=1}^{n_r}, \{\sigma_i\}_{i=1}^{n_r}, w, q). \tag{3}$$

The approach vectors $\{\mathbf{a}_i\}_{i=1}^{n_r}$ are derived from the $\mathbf{b}$. Each $\mathbf{a_i}$ is obtained by rotating an orthogonal vector $\mathbf{b}_\perp$ around $\mathbf{b}$ by a rotation angles γ_i. The vector $\mathbf{b} \perp$ is obtained using the Gram-Schmidt process. The angles $\gamma_i{}_{i=1}^{n_r}$ are calculated by dividing an angle range into n_r equal steps. This formulation allows each approach vector to be expressed as a linear transformation of $\mathbf{b}$, as shown in the equation:

$$\mathbf{a}_i = \mathbf{R}_{\gamma_i}\mathbf{b}_\perp, \mathbf{b}_\perp = \mathbf{R}_\perp \mathbf{b}, \tag{4}$$

where $\mathbf{R}_{\gamma_i}$ is the rotation matrix of γ_i and $\mathbf{R}_\perp$ is a constant rotation matrix.

Each approach vector $\mathbf{a}_i$ is labeled by a binary collision-score σ_i, which indicates if the grasp $(\mathbf{c}, \mathbf{b}, \mathbf{a}_i, w)$ is collision-free. The antipodal quality q measures the *antipodality* (Eq. 2) of the contact pair $(\mathbf{c}, \mathbf{c} + \mathbf{b}w)$.

B. Supervised Learning

The goal is to train a network, which takes a 3D normal grid $\mathbf{V}$ as input and infers a set of grasp configurations. We use each grasp configuration together with an uncertainty κ to parameterize the grasp distribution for each scene. Given the true grasp samples, the network is trained to predict optimal parameters that maximize the likelihood (Figure 1). We decompose the learning problem into learning the following two functions:

$$f_1 : \mathbf{V}_i \rightarrow p_\mathbf{c} \qquad f_2 : \mathbf{c} \rightarrow (\mathbf{b}, \kappa, \{\sigma_i\}_{i=1}^{n_r}, w, q) \tag{5}$$

The first function f_1 approximates the contact point distribution, which maps each voxel $\mathbf{V}_i$ of the input grid to the probability that the voxel contains a contact point. The second function f_2 then infers the antipodal quality q and the grasp parameters given the contact points sampled from f_1.

IV. METHODS

A. Grasp Distribution Modeling

We introduce a parameterized grasp distribution model to consider multiple grasp orientations and their orientation uncertainty at a contact point by representing the grasp distribution as a joint distribution over the grasp parameters $\mathbf{b}, \mathbf{a}, \mathbf{c}, \mathbf{t}, w$ conditioned on the input normal grid $\mathbf{V}$:

$$P(\mathbf{b}, \mathbf{a}, \mathbf{c}, \mathbf{t}, w|\mathbf{V}). \tag{6}$$

Here, we incorporate the approach point locations denoted as $\mathbf{t}$ into our model, as they indicate the gripper hand's position relative to an approach direction, which is critical for assessing whether a grasp may lead to collisions. The grasp distribution can then be factorized into a contact and approach point distribution $P(\mathbf{c}|\mathbf{V})$ and the local grasp distribution $P(\mathbf{b}, \mathbf{a}, \mathbf{t}, w|\mathbf{c}, \mathbf{V})$.

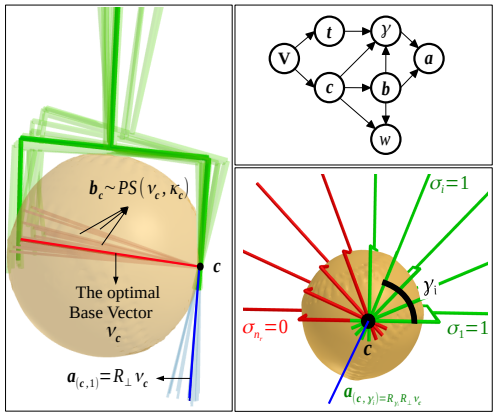

Fig. 3: A baseline vector distribution (left), modeled as a PS distribution $PS(\nu_\mathbf{c}, \kappa_\mathbf{c})$; An example of the grasp configuration (bottom right), where red and green forks represent collision and collision-free grasps, respectively, at a contact point; The graphical model (top right) illustrating the conditional dependency between variables.

1) Contact Point Distribution: We represent each bin picking scene as a 3D label grid L where each voxel belongs to one of the three non-overlapping sub-spaces: (1) contact space of the grasp contact point locations; (2) gripper space of the collision-free gripper poses represented by the approach point locations; (3) empty space which is the complement of the contact space and the gripper space. Accordingly, each voxel is classified as contact voxel $l_\mathbf{c}$, approach point voxels $l_\mathbf{t}$, or empty voxel $l_\mathbf{e}$. The contact point distribution is then approximated by the prediction of contact voxels.

2) Local Grasp Distribution: We further factorize the local grasp distribution into three distributions based on the conditional dependency between the grasp parameters as shown in Fig. 3. The factorization is formulated as

$$\begin{aligned} P(\mathbf{b}, \mathbf{a}, \mathbf{t}, w|\mathbf{c}) &= \int_\gamma P(\mathbf{b}, \mathbf{a}, \mathbf{t}, w, \gamma|\mathbf{c}) d\gamma \\ &= P(w|\mathbf{b}, \mathbf{c}) P(\mathbf{b}|\mathbf{c}) \int_\gamma P(\mathbf{a}, \mathbf{t}, \gamma|\mathbf{b}, w, \mathbf{c}) d\gamma \\ &= \underbrace{P(w|\mathbf{b}, \mathbf{c})}_{\text{opening width}} \underbrace{P(\mathbf{b}|\mathbf{c})}_{\text{baseline}} \underbrace{\int_\gamma P(\mathbf{a}|\mathbf{b}, \gamma) P(\mathbf{t}|\mathbf{b}, \mathbf{c}, \gamma) P(\gamma|\mathbf{b}, \mathbf{c}) d\gamma}_{\text{approach}} \end{aligned} \tag{7}$$

Here, we removed $\mathbf{V}$ just for a simplification.

3) Baseline Vector Distribution: For a given contact point, there are multiple baseline vectors that represent a feasible grasp as shown in Fig. 3. We model the distribution of baseline vectors at a contact point using the the Power-Spherical (PS) distribution [28] parameterized by the optimal baseline vector $\nu_\mathbf{c}$ and the concentration parameter $\kappa_\mathbf{c}$, formulated as

$$P(\mathbf{b}|\mathbf{c}) = P_{PS}(\mathbf{b}|\nu_\mathbf{c}, \kappa_\mathbf{c}), \tag{8}$$

where $P_{PS}(\cdot|\nu_\mathbf{c}, \kappa_\mathbf{c})$ is the probabilistic density function (PDF) of the PS distribution $PS(\nu_\mathbf{c}, \kappa_\mathbf{c})$. The optimal base-

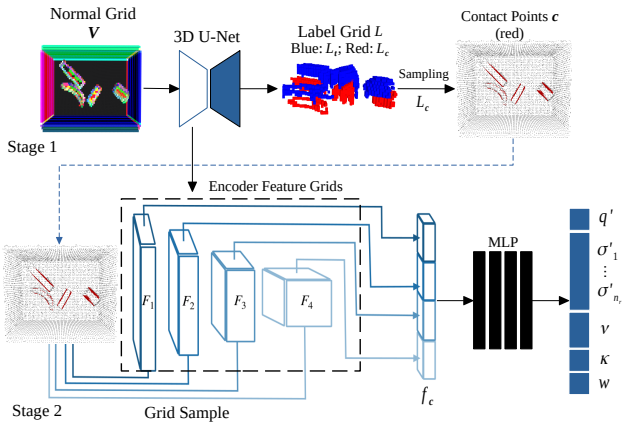

Fig. 4: The two-stage network architecture.

line vector refers to the one that represents the most stable contact point pair with the highest antipodal grasp quality.

4) Approach Vector Distribution: Since the PS distribution is closed under linear transformation [28], based on Eq. 4, approach vectors with respect to a rotation γ follows the PS distribution $PS(\mathbf{R}_\gamma \mathbf{R}_\perp \nu_\mathbf{c}, \kappa_\mathbf{c})$. Thus, we model the approach vector distribution as a mixture PS distribution based on Eq. 7 by defining the component and weight functions as

$$P(\mathbf{a}|\mathbf{b}, \gamma) := P_{PS}(\mathbf{a}|\mathbf{R}_\gamma \mathbf{R}_\perp \nu_\mathbf{c}, \kappa_\mathbf{c}), \tag{9}$$

$$P(\mathbf{t}|\mathbf{b}, \mathbf{c}, \gamma, \mathbf{V}) P(\gamma|\mathbf{b}, \mathbf{c}) := \sigma(\gamma). \tag{10}$$

We assume that the weight function $\sigma(\gamma)$ is a binary piecewise constant function, indicating whether an approach direction $\mathbf{a}_\gamma$ results in collisions. Computing the continuous integral of this mixture model during our training is intractable. Hence, we approximate the function using n_r binary values, corresponding to collision-scores $\{\sigma_i\}_{i=1}^{n_r}$ of n_r rotation angles (Eq. 3). We can then formulate the approach vector distribution at a contact point $\mathbf{c}$ as

$$P_\mathbf{c}(\mathbf{a}) = \frac{1}{\sum_{i=1}^{n_r} \sigma_i} \sum_{i=1}^{n_r} \sigma_i P_{PS}(\mathbf{a}|\mathbf{R}_{\gamma_i} \mathbf{R}_\perp \nu_\mathbf{c}, \kappa_\mathbf{c}). \tag{11}$$

B. Two-stage Network

Based on the grasp distribution modeling, we propose a two-stage neural network shown in Fig. 4.

In the first stage, a 3D Residual U-Net [29], [30] infers the label grid $\mathbf{L}$ given a 3D normal grid. Then, we sample n contact points from contact voxels. Each voxel of $\mathbf{L}$ stores the three confidence scores $[p_\mathbf{c}, p_\mathbf{t}, p_\mathbf{e}]$ for the classes of contact voxel $l_\mathbf{c}$, approach voxel $l_\mathbf{t}$ and empty voxel $l_\mathbf{e}$. We denote the grid storing $p_\mathbf{c}$ as $L_\mathbf{c}$ and the grid storing $p_\mathbf{t}$ as $L_\mathbf{t}$.

In the second stage, according to n contact point locations, we first grid sample and concatenate encoder features from each features grid F_i, which are intermediate outputs of U-Net encoder layers, forming n multi-resolution encoder feature vectors $\mathbf{f}_\mathbf{c}$. Each feature vector is then passed into a Multi-Layer Perceptron (MLP) generating the optimal baseline vector $\nu_\mathbf{c}$, inverse concentration $\kappa'_\mathbf{c}$, opening width $w_\mathbf{c}$, as well as the prior antipodal quality $q'_\mathbf{c}$ and n_r prior collision-scores $\sigma'_{1,\mathbf{c}}, \ldots, \sigma'_{n_r,\mathbf{c}}$ for each contact point. We refer them as "prior" quality/score because they are conditioned on the predicted label grid. We compute the quality $q_\mathbf{c}$ by multiplying $q'_\mathbf{c}$ with the confidence score of the respective contact point $\mathbf{c}$, formulated as

$$q_\mathbf{c} = I(L_\mathbf{c}, \mathbf{c}) q'_\mathbf{c}, \tag{12}$$

where I denotes the trilinear interpolation function. Similarly, a collision-score is calculated by multiplying its prior approach direction to the confidence score of the predicted approach points $\mathbf{t}_{(i,\mathbf{c})}$, formulated as

$$\sigma_{(i,\mathbf{c})} = I(L_\mathbf{t}, \mathbf{t}_{(i,\mathbf{c})})) \sigma'_{(i,\mathbf{c})}, \tag{13}$$

which aligns with the weight function Eq. 10. To obtain the concentration parameter κ, we derive it from κ' using the equation:

$$\kappa = \min(\max(\frac{\kappa_0}{\kappa' + \epsilon}, \kappa_0), 4\kappa_0) \tag{14}$$

where κ_0 is a tunable scale hyperparameter, and ϵ is used to avoid division by zero. In our training, we set $\kappa_0 = 25$.

1) Loss Functions: The loss function $\mathcal{L}$ is a weighted sum of a *label loss* $\mathcal{L}_l$ (the first stage) and a *grasp loss* $\mathcal{L}_g$ (the second stage), formulated as

$$\mathcal{L} = \phi_l \mathcal{L}_l + \phi_g \mathcal{L}_g, \tag{15}$$

where $\mathcal{L}_l$ is the focal loss [31] for the label grid classification and ϕ_l and ϕ_g are weights of two losses, respectively. The grasp loss is represented as

$$\mathcal{L}_g = \eta_b \mathcal{L}_b + \eta_a \mathcal{L}_a + \eta_w \mathcal{L}_w + \eta_q \mathcal{L}_q, \tag{16}$$

where $\mathcal{L}_b, \mathcal{L}_a, \mathcal{L}_w, \mathcal{L}_q$ are losses from the predictions of baseline vectors, approach vectors, opening widths and grasp qualities at the sampled contact points, respectively and $\eta_b, \eta_a, \eta_w, \eta_q$ are corresponding weights.

Baseline vector and approach vector loss functions are their negative log-likelihood functions. The opening width loss $\mathcal{L}_w$ is minimum distance between predicted contact point pair to its true contact point pair. For a better training convergence, we use the estimated baseline vector $\hat{\nu}_\mathbf{c}$ to compute predicted approach vectors and contact pairs, which is an average over baseline vectors of neighbour grasp labels. Moreover, $\mathcal{L}_q$ is the L_1 loss function.

C. Training & Inference

We employ a 64^3 normal grid input with a voxel size of approximately $9\,\mathrm{mm}$. Our neural network is trained with a batch size of 1, a learning rate initialized at 10^{-3} and decayed exponentially to 10^{-5} at the training's end. The MLP in the second stage starts training after 100 iterations of U-Net training. During training, we randomly sample a contact point within each voxel designated as a contact region. For inference, each contact region voxel is evenly subdivided into 8 sub-voxels. For each contact point, We generate 18 approach vectors. We define the rotation angle range as $[\frac{\pi}{2}, \frac{3\pi}{2}]$ to ensure that all approach vectors point

downward. To compute the loss, we assign weights of 10, 5, and 0.1 to the label grid classes (contact space, gripper space, and empty space) for computing the label loss. In the grasp loss, we set the weights for $\mathcal{L}_b$, $\mathcal{L}_a$, $\mathcal{L}_w$, and $\mathcal{L}_q$ as 1, 0.01, 0.1, 10, and 0.1, respectively. During training, we label each predicted contact point by searching for a maximum of 16 true contact points within a $3\,\mathrm{mm}$ radius. We compute $\mathcal{L}_b$, $\mathcal{L}_a$, and $\mathcal{L}_w$ only for contact points that have at least one true neighbor grasp sample.

D. Data Generation

We create a synthetic 6-DoF grasp dataset to train our model, which contains diverse bin picking scenes along with corresponding dense grasp labels. We generate a total of 1000 training scenes, each accommodating 1 to 20 objects. On average, each graspable object within these scenes is annotated with 470 grasps. The simulator used for dataset generation is PyBullet [32]. Scenes are simulated by randomly dropping objects into the bin in a simulation environment. We select 200 objects from the YCB [33], Google Scanned Objects (GSO) [34] and ABC [35] datasets, splitting them randomly into training (149 objects) and testing (51 objects) sets. Objects are also randomly scaled during scene generation. Two different bin sizes are used for dataset creation. For each scene we render a top-down view depth images with a resolution of 640x480. We add noise to the rendered images in simulation using the same additive noise model applied in [6]. Afterwards, we generate grasp labels for each scene following the pipeline proposed in [20]. The difference is that we kept all valid grasps that are collision-free and have a high antipodal quality (above 0.5) for each scene.

V. EXPERIMENTS

We conducted simulated and real-robot experiments to evaluate the performance of our grasp prediction model.

A. Simulated Experiments

Simulated experiments were executed on hardware comprising an NVIDIA Corporation GP107GLM and an Intel(R) Core(TM) i7-8850H CPU. A standard Franka parallel-jaw gripper was employed. To generate diverse test scenarios, we followed the training scene generation pipeline. We created 300 test scenes, evenly distributed among $easy$, $medium$ and $challenging$ scenarios, accommodating up to 5, 15, and 35 objects, respectively. $Easy$ scenes featured objects with simple structures (box, ball), whereas $medium$ and $challenging$ scenarios contain all objects from the test set. Additionally, we add noise to $challenging$ depth images, replicating the conditions of generating training images. The dimensions of the bin model in each test scene were $600 \times 400 \times 280\,\mathrm{mm}^3$. Performance was evaluated based on a PyBullet [32] simulation using the following metrics averaged over 100 scenes in each scenario:

- *Success Rate (SR)*: The ratio of successful grasps.
- *Clearing Rate (CR)*: The ratio of objects removed.

In each round, a grasp is executed if its antipodal quality and collision-score are greater than 0.5, and no collision at its pre-grasp pose. We test grasp configurations in a order of high to low antipodal quality. For each grasp configuration, we test first 8 out of 18 approach directions according to their collision-scores until find an executable grasp. A round ends when the bin is empty, no grasp is detected, or three consecutive failed attempts occur.

1) Baselines: We compare our approach with the three baseline methods:

- GPD [17]: a two-stage 6-DoF grasp detection algorithm that generates a large set of grasp candidates sampled based on an input point cloud and evaluates each of them using a neural function.
- VGN [5]: a 3D convolutional neural network takes a TSDF grid as input and outputs grasps parameters at each voxel centers.
- GIGA [6]: a two-stage network that jointly detects 6-DoF grasp poses and reconstructs the 3D scene given a single-view TSDF grid.

GIGA and VGN were retrained on our dataset. To this end, we label each voxel containing at least one grasp center with the grasp orientation having the highest antipodal quality. We use the same top-down view depth images to test these baseline methods. We additionally compare the performance to two ablated versions of our methods: "wo κ + 1app and "w κ + 1app". For the first one, we trained our network to learn the optimal baseline vector by maximizing cosine similarity between each predicted baseline vector and the average vector over its neighbor ground truth baseline vectors, *without* considering the uncertainty κ. The second one is trained *with* the uncertainty learning. However, during experiments, we only return the optimal approach direction for each contact point.

2) Results: Our experimental results (Table I) across 100 test scenes in three scenarios demonstrate our method's robustness in managing cluttered and noisy input. Even in the most challenging scenario, it achieves 70% success rate and 83% clearing rate, outperforming the baseline methods. Even though "w κ + 1app" only executes a single approach direction, it still generates more accurate and denser grasps than GIGA and VGN (see Fig. 5). Additionally, we compare between models trained with and without uncertainty learning for the baseline vector prediction, specifically "w κ + 1app" and "wo κ + 1app". Our observations indicate a significant performance gap between the two, particularly in transitioning from easy to challenging scenarios. This suggests that uncertainty learning is crucial for improving grasp prediction accuracy and making the network more robust in noisy and cluttered input situations. In Fig. 5, we also present an example of the uncertainty prediction, which we can see that the model predict low uncertainty values (yellow points) to contact regions that are less cluttered and more easier to approach. Lastly, we also observe the multi-approaches grasp prediction (8app) effectively handles collisions that occur when the gripper approaching the target object without prior motion planning. Importantly, this capability was not explicitly trained but emerges in practice. This is also reflected by the higher SR and CR when employing

Fig. 5: Predicted grasps from GIGA (left), ours (middle), and our uncertainty prediction (right) on a *medium* test scene.

multi-approach grasps in medium and challenging scenarios, in comparison to using single-approach grasps. In terms of runtime, our model has an inference time of approximately 236 ms on our machine, while GIGA requires about 233 ms, and VGN about 136 ms. The runtime for GPD varies with difficulty levels, taking around 373 ms, 496 ms, and 638 ms in three different scenarios.

Method	Easy		Medium		Challenging	
	SR (%)	CR (%)	SR (%)	CR (%)	SR (%)	CR (%)
GPD [17]	45.0	36.6	33.5	11.1	20.7	4.1
VGN [5]	58.4	51.9	48.4	27.0	39.8	12.0
GIGA [6]	64.1	59.1	57.8	37.9	54.7	32.6
Ours (wo κ + 1app)	71.0	93.6	59.5	73.5	49.9	43.8
Ours (w κ + 1app)	81.0	96.6	66.0	82.5	62.8	72.3
Ours (w κ + 8app)	**87.4**	**97.8**	**75.8**	**91.5**	**70.5**	**83.0**

TABLE I: Averaged success rate (SR) and clearing rate (CR) over 100 test scenes in *easy*, *medium* and *challenging* scenarios. Bold text indicates the optimal performance.

B. Real Robot Experiments

1) Experiment Setup: In the real robot experiments, we used a 7-axis Franka Emika Panda robot equipped a Franka parallel-jaw gripper with custom silicon fingers (see Fig. 6). A RealSense D415 RGBD camera with a resolution of 1920×1080 is statically mounted above a source bin with an distance of around 0.6 m for top-down view image capturing of the bin picking scene. The test object portfolio consists of 10 diverse objects including boxes, tubes, industrial objects, textiles, a USB cable and a screw driver. In each round, we place all test objects in the source bin an perform a manual mixing to generate random object poses within the bin. Additionally, we remove objects close to the bin walls that would not be graspable due to collisions between the used gripper and the bin and drop them manually from a random position above the bin. We use the same model weights and network configuration as in the simulated experiments. During the experiments, we determine the final grasp among all grasp predictions by selecting the grasp configuration with the highest grasp quality score. Then we select the grasp with the lowest collision score within the previously selected grasp configuration. In case of collisions between gripper and bin or objects during a grasp attempt, we first manually stop the robot and then consider the next highest predicted collision-score for the selected grasp configuration. With this setting, we run 10 rounds in total. A round stops if we had 3 consecutive failure grasps.

Fig. 6: Experimental setup with robotic arm, parallel-jaw gripper, overhead RGBD camera and bins (left), diverse object portfolio (middle), successful grasp (top right), failure grasp (middle right), collision grasp (bottom right).

2) Results: The robot experiments result in a clearing rate of 90% and a success rate of 77% with a total number of 116 grasp attempts and 90 successful grasps. We also encountered 4 double grasps where two objects were grasped at once during a grasp attempt. In this case, the grasp counts as a failed attempt and we dropped the objects in the source bin from a random position above the bin. The results show a similar performance as in the simulated experiments without any adjustments of the trained model which indicates the robustness to noisy image input data due to the grasp distribution modeling.

VI. CONCLUSIONS

In this work, we considered the problem of 6-DoF grasp detection for cluttered and unknown objects located inside a bin. We presented a novel approach to predict dense and collision-free grasps with multiple orientations per grasp point based on two-stage network architecture. The approach includes a new method for grasp distribution modeling based on Power Spherical distributions that accounts for grasp orienation uncertainty and enables a training based on multiple orientations per grasp contact. Hence, the network can be trained with more potential ground truth grasps and allows to predict more feasible grasps which is essential when grapsing with limited reachability inside a bin. Using only a single-view depth image we can predict dense and collision-free grasps on cluttered object scenes including multiple grasp orientations at grasp point. We evaluated our approach with simulated and real robot experiments on diverse object types resulting in a clearing rate of around 90%. We also showed that we outperform state of the art approaches for 6-DoF grasp prediction in bin picking tasks. Future work could consider diverse gripper designs in the dataset generation or the network architecture to enable the usage for different robot setups. Moreover, the approach might be combined with additional robot pushing skills or advanced grasp selection methods to further increase the object clearing performance in challenging bin picking scenarios.

REFERENCES

[1] 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," 2017.

[2] D. Morrison, P. Corke, and J. Leitner, "Closing the Loop for Robotic Grasping: A Real-time, Generative Grasp Synthesis Approach," in *Proc. of Robotics: Science and Systems (RSS)*, 2018.

[3] 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)*. IEEE, 2021, pp. 13 438–13 444.

[4] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," *Advances in neural information processing systems*, vol. 30, 2017.

[5] 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.

[6] 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.

[7] J. Cai, J. Cen, H. Wang, and M. Y. Wang, "Real-time collision-free grasp pose detection with geometry-aware refinement using high-resolution volume," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 1888–1895, 2022.

[8] K.-Y. Jeng, Y.-C. Liu, Z. Y. Liu, J.-W. Wang, Y.-L. Chang, H.-T. Su, and W. Hsu, "Gdn: A coarse-to-fine (c2f) representation for end-to-end 6-dof grasp detection," in *Conference on Robot Learning*. PMLR, 2021, pp. 220–231.

[9] H. Tian, K. Song, S. Li, S. Ma, J. Xu, and Y. Yan, "Data-driven robotic visual grasping detection for unknown objects: A problem-oriented review," *Expert Systems with Applications*, vol. 211, p. 118624, 2023.

[10] K. Kleeberger, R. Bormann, W. Kraus, and M. F. Huber, "A survey on learning-based robotic grasping," *Current Robotics Reports*, vol. 1, pp. 239–249, 2020.

[11] M. Gou, H.-S. Fang, Z. Zhu, S. Xu, C. Wang, and C. Lu, "Rgb matters: Learning 7-dof grasp poses on monocular rgbd images," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 13 459–13 466.

[12] H.-S. Fang, C. Wang, M. Gou, and C. Lu, "Graspnet-1billion: A large-scale benchmark for general object grasping," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 11 444–11 453.

[13] C. Wang, H.-S. Fang, M. Gou, H. Fang, J. Gao, and C. Lu, "Graspness discovery in clutters for fast and accurate grasp detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 15 964–15 973.

[14] B. Zhao, H. Zhang, X. Lan, H. Wang, Z. Tian, and N. Zheng, "Regnet: Region-based grasp network for end-to-end grasp detection in point clouds," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 13 474–13 480.

[15] W. Wei, Y. Luo, F. Li, G. Xu, J. Zhong, W. Li, and P. Wang, "Gpr: Grasp pose refinement network for cluttered scenes," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 4295–4302.

[16] Q. Dai, Y. Zhu, Y. Geng, C. Ruan, J. Zhang, and H. Wang, "Graspnerf: Multiview-based 6-dof grasp detection for transparent and specular objects using generalizable nerf," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 1757–1763.

[17] 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.

[18] M. A. Riedlinger, M. Völk, K. Kleeberger, M. U. Khalid, and R. Bormann, "Model-free grasp learning framework based on physical simulation," in *International Symposium on Robotics (ISR). Munich, Germany*, 2020.

[19] J. Cai, J. Su, Z. Zhou, H. Cheng, Q. Chen, and M. Y. Wang, "Volumetric-based contact point detection for 7-dof grasping," in *6th Annual Conference on Robot Learning*, 2022. [Online]. Available: https://openreview.net/forum?id=SrSCqW4dq9

[20] C. Eppner, A. Mousavian, and D. Fox, "Acronym: A large-scale grasp dataset based on simulation," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 6222–6227.

[21] A. Mousavian, C. Eppner, and D. Fox, "6-dof graspnet: Variational grasp generation for object manipulation," in *International Conference on Computer Vision (ICCV)*, 2019.

[22] H. Liang, X. Ma, S. Li, M. Görner, S. Tang, B. Fang, F. Sun, and J. Zhang, "Pointnetgpd: Detecting grasp configurations from point sets," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 3629–3635.

[23] H.-S. Fang, C. Wang, H. Fang, M. Gou, J. Liu, H. Yan, W. Liu, Y. Xie, and C. Lu, "Anygrasp: Robust and efficient grasp perception in spatial and temporal domains," *IEEE Transactions on Robotics*, 2023.

[24] B. Curless and M. Levoy, "A volumetric method for building complex models from range images," in *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, 1996, pp. 303–312.

[25] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, S. Hodges, and A. Fitzgibbon, "Kinectfusion: Real-time dense surface mapping and tracking," in *2011 10th IEEE international symposium on mixed and augmented reality*. IEEE, 2011, pp. 127–136.

[26] Q.-Y. Zhou, J. Park, and V. Koltun, "Open3d: A modern library for 3d data processing," *arXiv preprint arXiv:1801.09847*, 2018.

[27] R. Newbury, M. Gu, L. Chumbley, A. Mousavian, C. Eppner, J. Leitner, J. Bohg, A. Morales, T. Asfour, D. Kragic, *et al.*, "Deep learning approaches to grasp synthesis: A review," *IEEE Transactions on Robotics*, 2023.

[28] N. De Cao and W. Aziz, "The power spherical distrbution," *Proceedings of the 37th International Conference on Machine Learning, INNF+*, 2020.

[29] K. Lee, J. Zung, P. Li, V. Jain, and H. S. Seung, "Superhuman accuracy on the snemi3d connectomics challenge," *arXiv preprint arXiv:1706.00120*, 2017.

[30] A. Wolny, L. Cerrone, A. Vijayan, R. Tofanelli, A. V. Barro, M. Louveaux, C. Wenzl, S. Strauss, D. Wilson-Sánchez, R. Lymbouridou, *et al.*, "Accurate and versatile 3d segmentation of plant tissues at cellular resolution," *Elife*, vol. 9, p. e57613, 2020.

[31] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2980–2988.

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

[33] B. Calli, A. Singh, A. Walsman, S. Srinivasa, P. Abbeel, and A. M. Dollar, "The ycb object and model set: Towards common benchmarks for manipulation research," in *2015 international conference on advanced robotics (ICAR)*. IEEE, 2015, pp. 510–517.

[34] L. Downs, A. Francis, N. Koenig, B. Kinman, R. Hickman, K. Reymann, T. B. McHugh, and V. Vanhoucke, "Google scanned objects: A high-quality dataset of 3d scanned household items," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 2553–2560.

[35] S. Koch, A. Matveev, Z. Jiang, F. Williams, A. Artemov, E. Burnaev, M. Alexa, D. Zorin, and D. Panozzo, "Abc: A big cad model dataset for geometric deep learning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 9601–9611.