On the Importance of Label Encoding and Uncertainty Estimation for Robotic Grasp Detection

Benedict Stephan, Dustin Aganian, Lars Hinneburg, Markus Eisenbach, Steffen Müller, Horst-Michael Gross

***Abstract*— Automated grasping of arbitrary objects is an essential skill for many applications such as smart manufacturing and human robot interaction. This makes grasp detection a vital skill for automated robotic systems. Recent work in model-free grasp detection uses point cloud data as input and typically outperforms the earlier work on RGB(D)-based methods. We show that RGB(D)-based methods are being underestimated due to suboptimal label encodings used for training. Using the evaluation pipeline of the GraspNet-1Billion dataset, we investigate different encodings and propose a novel encoding that significantly improves grasp detection on depth images. Additionally, we show shortcomings of the 2D rectangle grasps supplied by the GraspNet-1Billion dataset and propose a filtering scheme by which the ground truth labels can be improved significantly. Furthermore, we apply established methods for uncertainty estimation on our trained models since knowing when we can trust the model's decisions provides an advantage for real-world application. By doing so, we are the first to directly estimate uncertainties of detected grasps. We also investigate the applicability of the estimated aleatoric and epistemic uncertainties based on their theoretical properties. Additionally, we demonstrate the correlation between estimated uncertainties and grasp quality, thus improving selection of high quality grasp detections. By all these modifications, our approach using only depth images can compete with point-cloud-based approaches for grasp detection despite the lower degree of freedom for grasp poses in 2D image space.**

I. Introduction

For many applications, such as Industry 4.0 and human robot interaction, automated grasping is an essential skill [1], [2]. In most settings, exact shape information of the objects in undefined poses is often not available, as the acquisition of this information is hard to accomplish for the large variety of different objects. Therefore, estimating grasps based on model-based approaches is not feasible. Recent methods of model-free estimation of grasps using deep learning models may be more appropriate for these scenarios. Such methods rely on a dataset to learn grasp poses based on RGB(D) images or point clouds. In general, these datasets contain labeled grasp poses for parallel grippers only.

Recent work [4]–[6], focuses on point cloud inputs for estimating 6D grasp poses. This has the advantage of not restricting the possible orientation of the grasps to top down grasps, which was a downside of earlier work where the grasp poses were estimated as an oriented box in the image. Fang et al. [3] created the GraspNet-1Billion dataset and claimed in their experiments on this dataset that estimating grasp poses based on a point cloud is superior to estimation based on images.

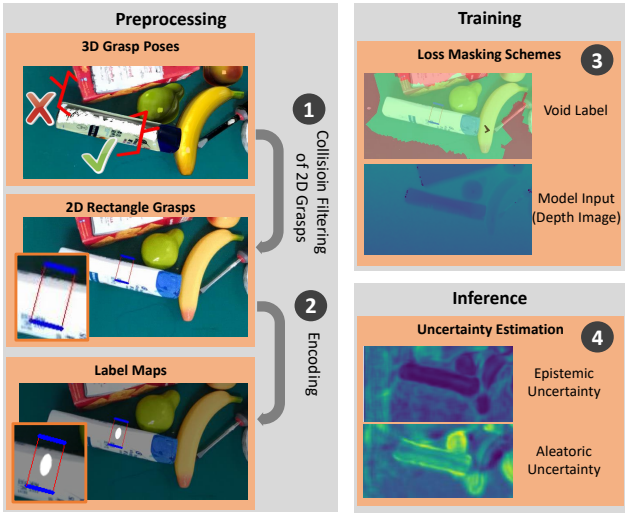


Fig. 1. Overview of the pipeline discussed in this work. First, we discuss a way to filter the 2D grasp labels provided by the GraspNet-1Billion dataset [3]. Second, we investigate ways to encode the 2D grasp labels into label maps needed for training our depth-based models. For training, we additionally propose to incorporate void labels for areas where no grasp labels are present. Last, we estimate two types of uncertainty for our model and investigate their plausibility and potential applications.

We show that image-based models, which can use color or depth data exclusively or in conjunction, were largely underestimated based on multiple reasons. As shown in Figure 1, we modify the usual pipeline in several ways in order to improve such image-based models: First, we show that the 2D rectangle grasp labels provided by the GraspNet-1Billion dataset contain grasps which are in collision when projecting them into 3D space, like it is done with the grasps detected by a trained model. Under the assumption a model would be able to perfectly replicate the provided labels this would put image-based models at a disadvantage in contrast to models trained on only valid labels. In Section III, we describe our filtering scheme for the supplied labels to overcome this drawback.

Since image-based models need to be trained to estimate a quality measure, an angle, and a gripper width for every pixel in the input image, it is necessary to convert the provided rectangular grasp labels to pixel-level maps that encode this information. We show that the common way of encoding grasp labels is insufficient for training well performing models. Thus, we propose an alternative encoding which

This work has received funding from the Carl Zeiss Foundation as part of the project E4SM under grant agreement no. P2017-01-005

All authors are with Neuroinformatics and Cognitive Robotics Lab, TU Ilmenau, 98693 Ilmenau, Germany benedict.stephan@tu-ilmenau.de

IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS), Kyoto, Japan, pp. 4781-88, IEEE 2022

enables the reconstruction of grasp poses close to the original grasp labels (see Section IV). By additionally introducing void regions in the image which are ignored during training, we direct the focus of the model to important regions, as further described in Section V.

Since we are using image-based models whose output consist of maps, we can apply established methods for estimating uncertainties. Thus, in Section VI we examine the estimated uncertainties regarding plausibility and applicability in order to select detected grasps for actual execution on a robot that are most trustworthy.

Thus, key contributions of this paper are threefold:

1) We improve the labels of the GraspNet-1Billion datasets regarding their usability for 2D image-based models.
2) We analyze the impact of the encoding for 2D rectangle-based grasps and provide an encoding that significantly improves 2D image-based models to match the performance of point-cloud-based 6D grasp pose estimation approaches.
3) We compute epistemic and aleatoric uncertainties and demonstrate how they can be employed to further improve the quality of grasp pose selection.

Code for dataset generation and training models is available at `https://github.com/TUI-NICR/nicr-grasping`.

II. Related Work

Grasp Detection: Model-free grasp detection is applied to different types of input modalities. CNN-based methods such as [7], [8] detect planar grasps based on depth images. A planar grasp is often represented as an oriented rectangle in the image plane described by its quality, the angle, and the width the gripper should be opened for the grasp. Often, these parameters are estimated based on a pixel-level estimation resulting in the estimation of grasps for every pixel. To employ this in an application, a local maximum search over the estimated quality map is performed to find possible locations of grasps. The inclusion of RGB images for training was done in [9], [10], focusing grasps on foreground objects.

For model-free grasp detection, multiple datasets exist that can be used to train a model. The Cornell dataset [11] as well as the Jacquard dataset [12] contain RGB and depth images and multiple labeled grasps per image. The annotated labels are either feasible grasps or negative grasps, although they are not used directly for training. Evaluation on these datasets is commonly assessed by the Jaccard index, where a grasp is counted as true positive as long as it is close enough to an existing label. This metric, however, is not suitable for evaluating the actual performance of a model as discussed in [3], [12]. Therefore, Fang et al. [3] proposed a pipeline for evaluating grasps without the need for actual execution on hardware. In contrast to the evaluation pipeline proposed by Depierre et al. [12], this pipeline does not rely on simulation and allows for analytical evaluation of large numbers of grasp poses. This pipeline is discussed in more detail in Section III.

Other methods [4]–[6], [13] use point clouds instead of depth or RGB images as inputs. As integration of uncertainty estimation is more complex for these methods, and by reaching competitive results by means of using our encodings described in Section IV, in our work, we focus on image-based models.

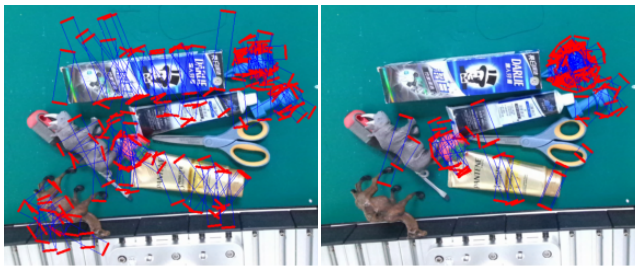
Uncertainties in Grasp Detection: The estimation of uncertainties for detected grasp poses has the potential of increasing the applicability of models in the real world as more information about the quality of estimated grasps could be gathered. Despite this huge potential, the application of known methods for uncertainty estimation of deep learning models, such as MC dropout [14] or density estimation [15], [16], was not yet evaluated in the literature. The only attempt towards this direction was done in [17]. Lundell et al. [17] used MC dropout for their 3D shape estimation for an object to be grasped. Then they used the uncertainty regarding the shape to select the most robust grasp poses. To the best of our knowledge, directly estimating the uncertainties of deep learning models for grasp detections has not yet been studied.

III. GraspNet evaluation

We use the evaluation pipeline of the GraspNet-1Billion dataset [3] in our experiments. The dataset contains color and depth images as well as point clouds recorded with a Kinect2 and a Realsense camera. As our experiments are independent of the camera used and can be applied to both parts of the dataset, we focus on the Kinect2 recordings. Additionally, we focus on depth-based methods and leave investigation of results on multimodal methods for future work.

The GraspNet-1Billion evaluation pipeline takes all estimated grasps and first applies a non-maximum suppression (NMS) in 3D space. After sorting the grasps by estimated quality for each of the top k proposed grasp poses remaining, the lowest friction coefficient μ_g for the grasp g to succeed is estimated based on the force closure metric [18]. Using these evaluated grasps, Fang et al. [3] define a metric AP_μ as the mean precision over the top $k \in [1, 50]$ proposed grasp poses. A positive grasp is defined as not being in collision and being successful at friction coefficient $\mu \in [0.2, 0.4, 0.6, 0.8, 1.0, 1.2]$. Collisions are thereby computed based on a fixed gripper model. Fang et al. [3] verified a correlation between their computed friction coefficients and the actual success rate of a grasp. Therefore, this score can be used as a substitute for actual execution.

As the original grasp poses for the GraspNet dataset were generated in Euclidean space, we first need to discuss the conversion procedure for generating 2D grasp labels. The authors of [3] first removed all 3D grasp labels whose grasp directions were not parallel to the camera axis. The remaining 3D grasp labels were then converted into 2D labels by projecting the positions of the gripper yaws into the image plane resulting in a rectangular grasp label. While this is a reasonable approach, this does not guarantee that the resulting grasp labels can be projected back to 3D space while still achieving the same quality measured by



Original labels Filtered labels

Fig. 2. Example images comparing rectangle grasp labels before and after filtering colliding labels in 3D space based on the evaluation pipeline. Each image shows 100 randomly sampled labels.

TABLE I

CHANGE IN QUALITY OF 2D LABELS THROUGH OUR FILTERING.

	original labels	filtered labels
AP	14.53	80.53
Num Grasps	879,071,408	136,225,781

their pipeline. In contrast to this approach, we filter the supplied rectangle grasps by projecting them into 3D space, evaluating them through the standard evaluation pipeline without applying NMS and finally filtering out all grasps which result in an empty grasp or a collision.

Table I shows the increase in quality over whole dataset by applying the evaluation pipeline to the projected ground truth 2D rectangle grasp labels before and after our filtering. This defines the upper limit of image-based models which are trained on these grasp labels. Additionally, it can be seen that the number of grasps is significantly lower (around 15%) after our filtering. As the GraspNet dataset contains a large amount of grasp labels, the number of remaining grasps is still sufficient as there are still far more grasp labels per sample than for other datasets such as Cornell [11]. Figure 2 shows an example of labeled grasps before and after our filtering approach. We can see that for some objects there are no grasp labels left. This issue will be addressed in Section V.

Now, having collision free grasp labels, we will deal with the subsequent processing of the now filtered ground truth grasp labels in the following Section.

IV. LABEL SHAPE WEIGHTING

In order to successfully train image-based models using grasp pose ground truth labels represented by oriented rectangles, it is necessary to choose an encoding to convert these labels to image maps in order to compute a pixel-wise loss. In the following, we address such encodings. First, we examine the shape of the labels and subsequently their weighting.

A. Encoding 2D-Grasp Shapes

Usually the inner third of the rectangle is used to draw the grasp parameters in the four label maps for position/quality, rotation (sine and cosine) and gripper width [7], [8], [19]. Afterwards, the entire label map is used for training using the ℓ^2 loss, thus always generating a loss for rotation and width, regardless of whether an object is present in an image area or not.

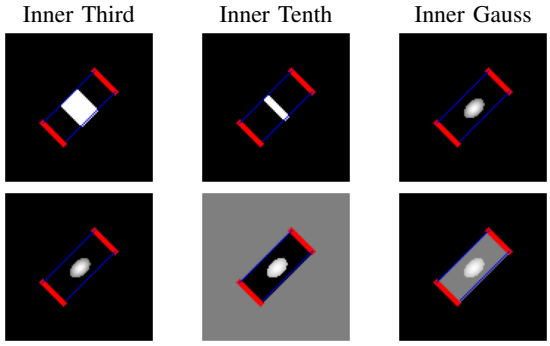


Fig. 3. Example of different grasp encodings. Top row shows different shapes. Bottom row shows different types of box ignore regions. White: positive label; Black: negative label; Grey: void; Red: gripper yaws; Blue: box enclosed by grippers.

Training an image-based model would, in the optimal case, learn exactly the encoding that is provided as a label. Therefore, we investigate the results of this inner third encoding, as well as other encodings, and thus show the theoretical upper limit when training image-based models with these encodings. To do so, we apply the GraspNet evaluation pipeline directly to the generated label maps of the different encodings. Additionally, we show the benefit for state-of-the-art 2D image-based methods when using these encodings.

For this analysis, we first consider three different types of shapes as presented in the upper part of Figure 3.

- First, the commonly found shape of using the inner third of the rectangle [7], [8], [19].
- Second, we use the inner tenth of the rectangle, as proposed by Fang et al. [3][1]
- Third, we examine our proposal for a shape, which is a two dimensional Gaussian centered in the grasp rectangle, called inner Gauss, which is explained below.

Inner Gauss: Drawing the Gaussian shape for an arbitrary grasp rectangle is done by using the convariance matrix $C \in \mathbb{R}^{2\times2}$. C can be computed using the diagonal matrix of eigenvalues $E \in \mathbb{R}^{2\times2}$ and the matrix of eigenvectors $V \in \mathbb{R}^{2\times2}$:

$$C = VEV^{-1}. \tag{1}$$

We use the length and width of the respective grasp rectangle as eigenvalues. The corresponding eigenvectors are given in image space. Using C and the grasp center in pixel coordiantes, the two-dimensional normal distribution allows us to compute the pixel-wise label values. Hereby, the maximum of a Gaussian will always be in the center of the grasp rectangle and, therefore, the original grasp position can be recovered through searching for local maxima. This is a major advantage of this encoding over the other two.

In Table II in the column "AP w/o Quality", we present the results for these three shapes when the ground truth labels are evaluated by the GraspNet evaluation pipeline. First, it

[1]The only reference of using the inner tenth in [3] can be found within an issue in the code repository for GraspNet:
https://github.com/graspnet/graspnetAPI/issues/30#issuecomment-1006422550

TABLE II

AP FOR GROUND TRUTH LABEL ENCODINGS TREATED AS PREDICTIONS.

Encoding Shape	AP w/o Quality	AP w/ Quality
Inner Third	3.18	6.09
Inner Tenth	4.43	21.85
Inner Gauss	20.15	55.53

shows that choosing inner third as encoding shape seems to perform worst. This is caused by the large position error of the recovered grasps through local maximum search, as a located maxmimum is unlikely to be located at the original center of the grasps. Furthermore, we confirm that the non-uniform label of the Gaussian encoding achieves the best results and seems to be particularly well suited for recovering the encoded grasp labels.

B. Weighting Labels according to their Quality

Since we computed the actual quality of the rectangle ground truth grasp labels in Section III, we can use this quality and weight the labels accordingly, rather than using a binary value as it is the case for the training with the 2D-labels in GraspNet [3] or other datasets such as the Cornell [11] or the Jaquard dataset [12]. Similar to [3], we define the score of a grasp g as $s(g) = 1.2 - \mu_g$ with μ_g as the computed friction coefficient for a grasp. When creating the label map for the position, the individual label shapes can now each be multiplied by the score for the associated grasp. For areas where grasp labels overlap, the grasp with the highest quality can now be used, also defining which parameters for the grasp angle and the gripper width are to be drawn into the label map. An evaluation of the resulting labels with quality is presented in Table II in the column "AP w/ Quality". Clearly, such a definition of the encoding is immensely beneficial, as grasps of lower quality can be sorted out by the NMS resulting in only the best grasp being evaluated. In contrast, using a binary label can result in lower quality grasps suppressing higher quality grasps. Likewise, in all subsequent experiments, we will only use labels that have been weighted with the quality for the training of image-based models, so that the models can be given the opportunity to differentiate in their quality estimation. Additionally, the models are less penalized if a bad label grasp is learned with a low estimated quality, as it would otherwise happen with binary labels.

Having dealt with the encoding of the grasp labels, in the next section we will examine how the remaining areas of the label maps should be addressed.

V. VOID MASKS AND IGNORE REGIONS

After encoding the rectangular grasp labels in the label maps, we will now examine how the areas in the label maps should be treated where grasp labels are not available.

A. Void Mask

First, we will examine the loss calculation for the rotation and width maps for the positions where grasp labels are not available. As explained at the beginning of Section IV, the loss is usually calculated for all positions on the label maps.

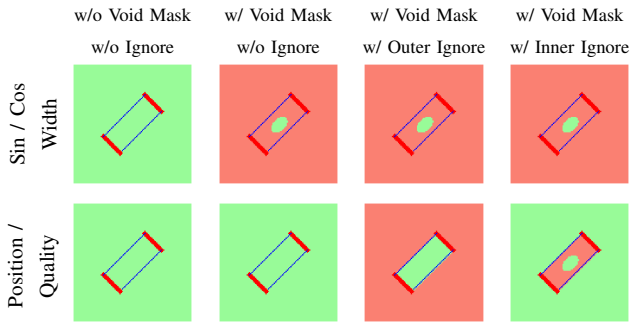


Fig. 4. Different masks applied during training. Masks shown in top row are applied to cosine, sine, and gripper width output, while masks shown in bottom row are applied to position output. Green: non-void; Red: void; Gripper yaws and box enclosed by grippers visualized as in Figure 3.

Thus, during training, a model is also forced to predict zero for rotation and width for positions where no objects and label are present, even though an output at such a position is not well defined.

We therefore propose that all positions in the rotation and width maps with no values greater than zero in the position/quality maps should be masked during training and thus not be included in the loss calculation. This should cause the model to concentrate more on the actual labels and therefore lead to better results. This approach is analogous to the void class in semantic image segmentation, where for unknown classes the model is also given a void label, which is masked during loss calculation. A visual representation for the comparison of the void masks between the standard from the state of the art and our proposal, can be obtained from the first two columns from Figure 4. Later in this section, we will use an comparative study to investigate the use of the void mask for the rotation and width maps.

B. Ignore Region

Next, we will consider the possibility of voiding the label on the position maps. Unlike rotation and width maps, position labels are defined for object-free surfaces. However, training of empty surfaces is not of interest for practical application. Since in practice, grasps are estimated directly in the regions containing interesting objects. For example, Ainetter et al. [20] apply image segmentation for this purpose, thus most of the captured scene, such as the empty table, is being omitted. Therefore, we believe, it would be advantageous for a model to focus on meaningful negative grasps located on objects, rather than focusing on trivial predictions such as on flat tables. Thus, to reduce the amount of unnecessary negative labels on the position map, we introduce an outer box ignore region located around the oriented rectangle grasp labels. In this ignore region, the negative position labels are replaced by void labels and are thus not taken into consideration during training. The underlying idea is that positions right next to working grasps are good examples of negative positions, since here a grasp should either collide with an object or miss. Furthermoe, in the filtered dataset (see Section III) some objects have no grasp labels, as all ground truth labels were invalid. Forcing

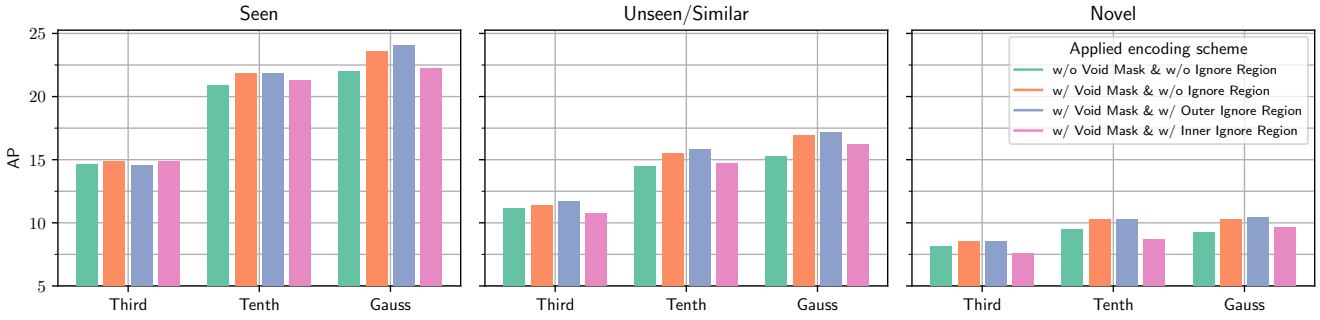


Fig. 5. Comparison of AP achieved on different test splits by GRConvNet models trained with different shape encodings and masking schemes.

TABLE III
OVERVIEW OF BEST RESULTS ON DIFFERENT MODELS WITH RESPECTIVE ENCODING AND MASKING SCHEMES.
WE ALSO REPORT RESULTS WHEN FILTERING COLLIDING GRASPS BEFORE EVALUATION.

Method	Encoding	Collision Filtered	Seen			Unseen / Similar			Novel		
			AP	$AP_{0.8}$	$AP_{0.4}$	AP	$AP_{0.8}$	$AP_{0.4}$	AP	$AP_{0.8}$	$AP_{0.4}$
Depth-based [3] (GGCNN [8])	Not Described		16.89	22.47	11.23	15.05	19.76	6.19	7.38	8.78	1.32
RGB(D)-based [3] (Chu et al. [10])	Not Described		17.59	24.67	12.74	17.36	21.64	8.86	8.04	9.34	1.76
Point-cloud-based [3]	6D Grasp Poses		29.88	36.19	19.31	27.84	33.19	16.62	11.51	12.92	3.56
GGCNN [8]	Gauss w/ Outer Ign		17.28	21.40	9.67	11.67	14.62	5.29	9.76	12.16	3.63
GGCNN [8]	Gauss w/ Outer Ign	✓	23.04	28.42	12.42	17.10	21.32	7.49	14.81	18.31	5.38
GRConvNet [7]	Gauss w/ Outer Ign		**23.18**	29.11	14.40	**16.12**	20.49	8.30	**11.15**	13.83	4.70
GRConvNet [7]	Gauss w/ Outer Ign	✓	**29.77**	37.20	17.82	**22.23**	28.12	11.00	**16.43**	20.27	6.72

the model to learn that grasps are not possible at these positions could degrade the model's ability to generalize to novel objects.

We investigate this outer box ignore region in more detail in the following experiments. We will also include an investigation of an inner box ignore region, which was proposed by Fang et al. [3][2]. The inner box ignore region is the opposite to the outer box ignore region. With the inner box ignore region all negative labels inside the oriented rectangle labels are replaced by void labels and the negative labels outside the rectangles remain negative labels. A visualization of the ignore regions is shown in Figure 3 and their respective void masks are shown in the right two columns of Figure 4.

C. Experimental Setup

We perform our comparative studies on the different masking schemes on the depth images of the Kinect camera of the GraspNet dataset [3]. As described in Section III, the general findings should be independent of the camera and also independent of whether RGB data are used in addition to depth data or not. As 2D image-based models, we trained two models: The first model is GGCNN [8], which had also served as a baseline in GraspNet. As second model, we trained the more powerful GRConvNet [7]. For each of the different combinations of shape, loss mask, and margin, we performed a search for all relevant hyperparameters, such as learning rate and learning rate scheduling, to select the best hyperparameters. Likewise, we have also repeated the training for each combination several times and present the AP for the best result in each case. As described in Section III, we determine and report the AP using the evaluation pipeline from GraspNet. We report the AP, $AP_{0.4}$, and $AP_{0.8}$ over the three test splits of seen, unseen (also called similar), and novel objects. Furthermore, as a sufficient amount of collisions of computed grasp poses can be efficiently estimated through the use of point clouds captured by a robot with nearly no computational overhead, we report computed metrics with and without filtering colliding grasps before applying the evaluation pipeline.

D. Comparative Studies

In the following comparative studies, we will examine our introduced encoding types, compare our best model with the state of the art, and evaluate it in more detail. Figure 5 provides a summary of all results, which we will discuss in detail below. For clarity, only the best GRConvNet models for the different combinations of shape, void mask, and ignore region are shown here. First, we can see general trends independent of the test split (seen, unseen/similar, novel)[3]. We start by comparing the three tested shapes for encoding grasp pose labels (as in Figure 3 top row). In this comparison, the results from Table II are also confirmed by our experiments. The proposed inner Gauss provides the best results. Next, we compare the void masks. When comparing the green and orange bars, the AP is always significantly better when non-relevant locations for rotation and grasp width are masked during loss estimation (orange bar). Next, we examine the results of the trainings with void mask, where no ignore region, outer box ignore region, or inner box ignore

[2]The only reference to them using the inner bock ignore region can be found within an issue to their published code for GraspNet:
https://github.com/graspnet/graspnetAPI/issues/30#issuecomment-1006422550

[3]These trends can also be seen for the different friction coefficients $AP_{0.4}$ and $AP_{0.8}$, which have also been omitted for the sake of clarity.

region was used (orange, blue and pink bars). Apparently, the outer box ignore region has an advantage over no ignore region with void mask, especially with inner Gauss encoding, whereas the inner box ignore region has no advantage. This also confirms our chain of reasoning above. In Section VI, we will discuss uncertainty estimation in more detail, and we will also show that the outer box ignore region provides a great advantage in this regard.

For further evaluation of our achieved results, in Table III in addition to our best results for the depth-image-based GGCNN and GRConvNet models, we present the results published in [3] for 2D-image-based models. Likewise, we also present the results when a collision filtering is performed prior to the NMS in the evaluation pipeline, as it is done in real-world applications preceding the grasp execution. The differences in performance regarding collision filtering show that some of the grasps are predicted in collision. When these are filtered, many good grasps remain. Comparing our best model with the models from [3], we notice that our model not only outperforms their depth-based model but also outperforms their RGB(D)-based model. Furthermore, comparing our model with the point-cloud-based model also shows that we achieve comparable results in the seen test set and even clearly outperform it in the most difficult test set with novel objects. This shows the great advantage that our encoding with inner Gauss and outer box ignore regions brings for training 2D image-based models.

In [3], grasps are estimated mainly in 3D space, where a larger input and output space allows for much more diverse grasps to be estimated than when estimating on 2D input data, as in our approach. Due to this fact, the choice to use the top 50 grasps per scene in the AP calculation might be too large for the evaluation of 2D grasp estimation. To investigate this further, we looked at how many grasps remain on average per scene after collision filtering and NMS. For our best models, on average about 20 grasps remain regardless of the test split. To show how good these 20 grasps actually are, in Figure 6 we present the collision filtered evaluation of our best model for different top k, with $k \in (10, 20, 30, 40, 50)$. Here, an evaluation of the actual estimated grasps can be examined rather than an assessment that largely penalizes the score in case of missing grasps when applying the top 50 AP metric. We can see that the curves actually improve the smaller the k is. Based on this observation, we conclude that the quality estimation of the model actually correlates with the real quality of the grasps.

After successfully achieving improved grasp detection results by incorporating void areas in conjunction with our improved grasp label encoding, we will now describe our experiments on uncertainty estimation.

VI. Uncertainty Estimation

For estimating uncertainties, we focus on the best models from Section V. Therefore, we use GRConvNet [7] as a base with our Gaussian-based margin encoding as labels unless noted otherwise. To be able to estimate uncertainties, we first need to adapt the model.

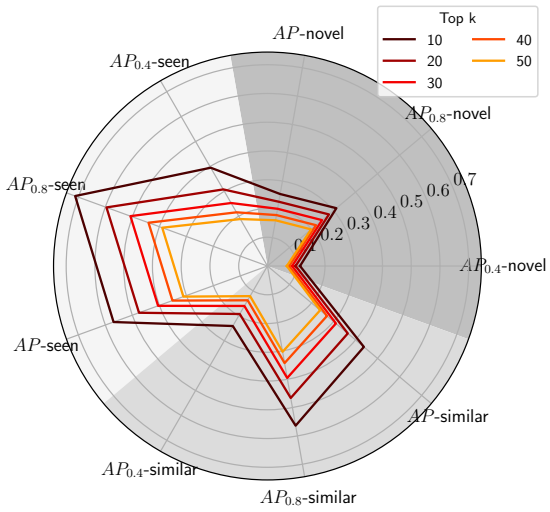


Fig. 6. AP for different top k for our best model shown in Table III. Values reported are results when filtering colliding grasps before evaluation.

Epistemic uncertainty is commonly estimated using MC dropout [14]. This method approximates the distribution over the weights by sampling the model with different dropout masks during inference. Therefore, we added dropout with $p = 0.2$ before all convolution layers except the first and last ones. Additionally, we need to choose the number of samples to draw during inference to estimate the uncertainty. We choose a sample size of 50 as this is the value used most commonly [14]. It is important to note that the number of samples drawn has an impact on the inference speed of the model during application. Furthermore, the applicability of methods to optimize inference speed, such as parallel computation of multiple samples, depends on available hardware resources and model size.

Aleatoric uncertainty can be estimated as a function of the input. Thus, we can train a model for this task using the Gaussian-based negative log likelihood [15], [16]. We could add additional heads to the original model and estimate uncertainties and actual grasp detection in tandem. As this tends to result in a slight loss of quality ($\sim 2\%$ for AP in our experiments), we froze the model used for grasp detection and trained an additional model with the same architecture for uncertainty estimation. Since these models can be computed in parallel and are identical in complexity, this does not reduce inference speed.

A. Quantitative evaluation

Similar to [16], we first evaluate the estimated uncertainties based on their theoretical properties and visual plausibility. The usage of grasp encodings with margin allows us to verify the plausibility of epistemic uncertainty, since large parts of the image are labeled as void, and therefore have no influence on the model during training. When training a model while ignoring the parts in the input image where no grasp is labeled, such as the empty table, this type of input basically becomes unknown to the model. Therefore, these inputs should generate higher epistemic uncertainty than areas containing known objects.

Figure 7 shows example predictions for models trained with no ignore region and outer box ignore region. We can see that the model trained with outer box ignore region and thus had regions without objects labeled as void generates high epistemic uncertainty in the empty area around the object whereas the model trained with no ignore region does not. This shows that unknown areas in the input can be detected by means of epistemic uncertainty. For real-world applications the knowledge of the distribution of epistemic uncertainty over the image is an important tool. Especially, if no application-specific dataset can be collected or generated, and therefore all inputs are potentially novel for the model, it is important to know where we can trust the models' decisions and where it cannot make trustworthy decisions regarding robust grasp poses.

When using estimated uncertainties in an application, it is important to know how the uncertainties correlate with the actual quality of the estimated grasps. If they are not correlated, they are not applicable in a meaningful way. Figure 8 shows the uncertainties versus the grasp quality score computed by the GraspNet evaluation. We can see a negative correlation of epistemic uncertainty with the grasp quality score. The lower the epistemic uncertainty is, the higher the quality score. Aleatoric uncertainty is positively correlated with quality. This means high aleatoric uncertainty is associated with high grasp quality. This is plausible, since contradictions in the ground truth lead to high aleatoric uncertainty estimates. For grasping, this is in particular the case when grasps with maximum quality are available for some objects in a subset of the data and no valid grasps are labeled in another subset of the data, e.g., due to filtering out grasp labels as in our proposed approach. In this case, maximum quality labels and negative labels contradict, leading to high aleatoric uncertainty. However, selecting grasps with high aleatoric uncertainty would mean, selecting grasps based on contradictory knowledge. While we do not propose to use aleatoric uncertainty due to this reason, it could be useful when dealing with the described contradictions during training.

Furthermore, we can observe similar correlation of epistemic and aleatoric uncertainty with grasp quality over different test splits. The only exception is the correlation of aleatoric uncertainties and grasp quality in the novel split. As the aleatoric uncertainty is estimated by a network as a function of the input, the novel split presents unknown data

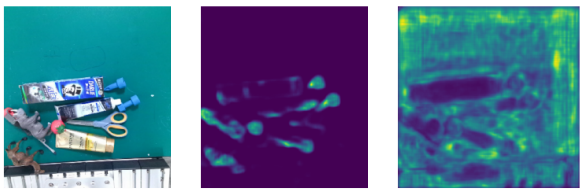


Fig. 7. Example prediction of epistemic uncertainty for models trained with no ignore regions and with outer box ignore regions respectively. Dark color means low uncertainty while bright colors represent high uncertainty.

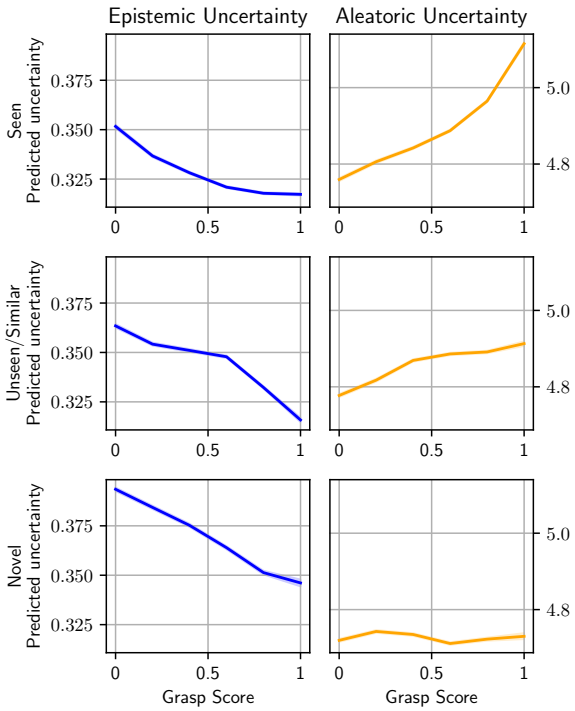


Fig. 8. Mean uncertainty vs. grasp score. Rows show results on different test splits. Correlation of epistemic uncertainty with the grasp quality score is visible across test splits while aleatoric uncertainty is only informative over known objects.

and therefore the network has to extrapolate, which is known to be a source of error for neural networks.

B. Qualitative evaluation

As the investigation of correlation showed that aleatoric uncertainties are not suitable for selecting high quality grasps, we focus on epistemic uncertainty in the following. To evaluate the usefulness of the estimated epistemic uncertainties, we investigate the change in quality of the estimated grasps when filtering the estimated grasps based on the estimated uncertainties. We apply a threshold to the estimated uncertainties and keep only the grasps with lower epistemic uncertainty than the threshold. The evaluation of the filtered grasps with the GraspNet evaluation pipeline is not representative as the number of remaining grasps after the NMS is to small to investigate the impact of the filtering. Therefore, we investigate the correlation of uncertainty and quality score of all estimated grasps, meaning without NMS applied. As in practice a threshold for the estimated quality is applied, we report results for a minimum estimated quality of $q \in [0.0, 0.1, 0.2]$.

Figure 9 shows the mean score of all grasps versus the percentage of remaining grasps after filtering by epistemic uncertainty. We varied the threshold for uncertainty with a stepsize of 0.05. As it is not practical to apply a filter such that there are samples with no grasps remaining, we only report results as long as every sample in the dataset has at least one grasp remaining after filtering.

For all splits, we can see that filtering by epistemic uncertainty in general increases the mean quality of the remaining grasps. The improvement in quality score is greatest for the

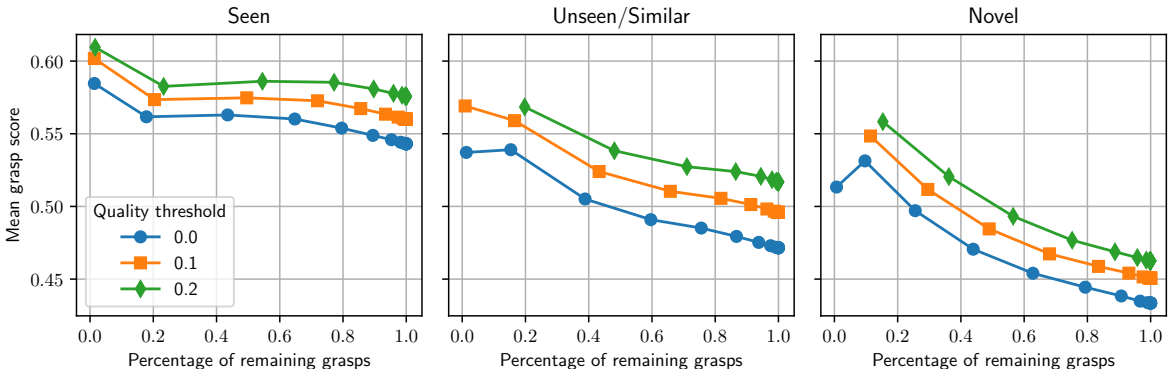


Fig. 9. Results of filtering estimated grasps based on epistemic uncertainty. Results are only reported as long as every sample has at least one grasp remaining. The figure shows results for different thresholds over estimated quality. While being low for seen objects, the improvement when filtering out uncertain grasps is visible the more unknown the data becomes to the model.

novel split. This implies that the epistemic uncertainty is informative even on objects the model has not seen during training. For the application of a trained model in a specific scenario, this can be an advantage.

By estimating uncertainties and showing their potential for real-world applications, we further improved our depth-based models over the point-cloud-based approach of [3].

VII. Conclusion

In this work, we showed through improved label filtering and label generation that depth-based models still have potential for grasp detection that was overlooked in previous work, as we achieved comparable results to point-cloud-based methods. We would like to point out that this performance is achieved on depth inputs only. In future work, it could be further improved by also considering the RGB inputs available. Additionally, we have taken the first step towards applying uncertainty estimation to our grasp detections in order to incorporate the potential that these uncertainties hold for real-world application. By showing the immediate applicability and the plausibility of the estimated epistemic uncertainties, we pave the way for future studies regarding this topic. Even though we did not use aleatoric uncertainty directly, its correlation with the grasp scores was confirmed. Further work on utilizing these type of uncertainty provides potential for improving contradicting labels, which are mostly unavoidable when estimating grasp poses, as there are a multitude of different grasps which can be applied to a single position on an object.

References

[1] M. Eisenbach, D. Aganian, M. Koehler, B. Stephan, Ch. Schroeter, and H.-M. Gross, "Visual scene understanding for enabling situation-aware cobots," in *IEEE Int. Conf. on Automation Science and Engineering (CASE)*, p. 2 pages, IEEE, 2021.

[2] C. Pohl, K. Hitzler, R. Grimm, A. Zea, U. D. Hanebeck, and T. Asfour, "Affordance-based grasping and manipulation in real world applications," *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 9569–9576, 2020.

[3] 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*, pp. 11444–11453, 2020.

[4] 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.

[5] J. Varley, J. Weisz, J. Weiss, and P. Allen, "Generating multi-fingered robotic grasps via deep learning," in *2015 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pp. 4415–4420, IEEE, 2015.

[6] 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)*, pp. 3629–3635, IEEE, 2019.

[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)*, pp. 9626–9633, IEEE, 2020.

[8] 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.

[9] 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)*, pp. 13459–13466, IEEE, 2021.

[10] F.-J. Chu, R. Xu, and P. A. Vela, "Real-world multiobject, multigrasp detection," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3355–3362, 2018.

[11] 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.

[12] A. Depierre, E. Dellandréa, and L. Chen, "Jacquard: A large scale dataset for robotic grasp detection," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3511–3516, IEEE, 2018.

[13] A. Mousavian, C. Eppner, and D. Fox, "6-dof graspnet: Variational grasp generation for object manipulation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2901–2910, 2019.

[14] Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," in *international conference on machine learning*, pp. 1050–1059, PMLR, 2016.

[15] C. M. Bishop, "Mixture density networks," 1994.

[16] A. Kendall and Y. Gal, "What uncertainties do we need in bayesian deep learning for computer vision?," *Advances in neural information processing systems*, vol. 30, 2017.

[17] J. Lundell, F. Verdoja, and V. Kyrki, "Robust grasp planning over uncertain shape completions," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1526–1532, IEEE, 2019.

[18] V.-D. Nguyen, "Constructing force-closure grasps," *The International Journal of Robotics Research*, vol. 7, no. 3, pp. 3–16, 1988.

[19] 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, 2020.

[20] S. Ainetter and F. Fraundorfer, "End-to-end trainable deep neural network for robotic grasp detection and semantic segmentation from rgb," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 13452–13458, IEEE, 2021.