Learning Any-View 6DoF Robotic Grasping in Cluttered Scenes via Neural Surface Rendering

Snehal Jauhri[1], Ishikaa Lunawat[2], and Georgia Chalvatzaki[1,3,4]
[1] Computer Science Dept., TU Darmstadt, Germany
[2] NIT Trichy, India [3] Hessian.AI, Darmstadt, Germany
[4] Center for Mind, Brain and Behavior, Uni. Marburg and JLU Giessen, Germany

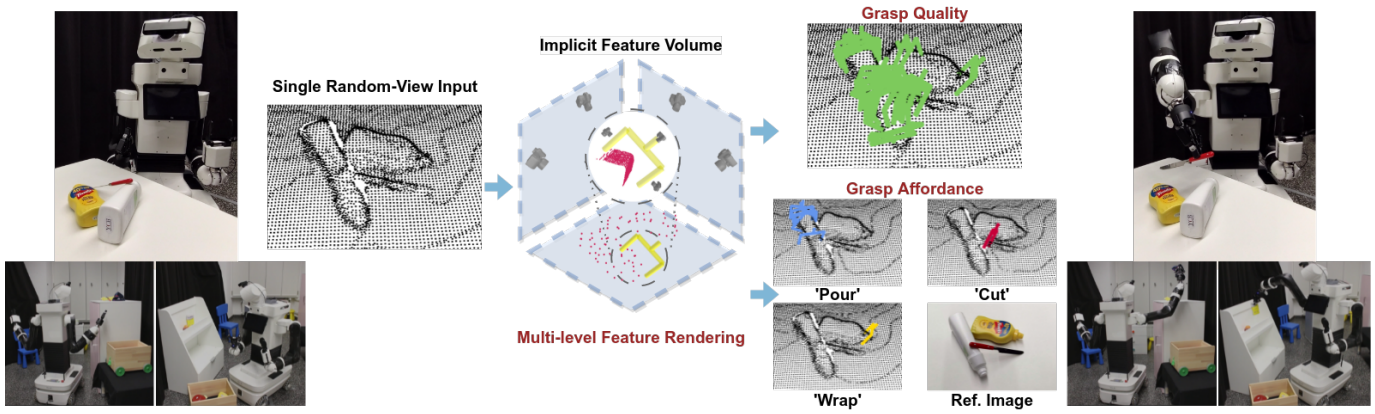


Fig. 1: **Grasping as rendering**. Our network, NeuGraspNet, uses a single random-view depth input, encodes the scene in an implicit feature volume, and uses multi-level rendering to select relevant features and predict grasping functions. As shown, NeuGraspNet generalizes to random-view mobile manipulation grasping scenarios. More at: sites.google.com/view/neugraspnet

***Abstract*—A significant challenge for real-world robotic manipulation is the effective 6DoF grasping of objects in cluttered scenes from any single viewpoint without needing additional scene exploration. This work re-interprets grasping as *rendering* and introduces NeuGraspNet, a novel method for 6DoF grasp detection that leverages advances in neural volumetric representations and surface rendering. We encode the interaction between a robot's end-effector and an object's surface by jointly learning to render the local object surface and learning grasping functions in a shared feature space. Our approach uses global (scene-level) features for grasp generation and local (grasp-level) neural surface features for grasp evaluation. This enables effective, fully implicit 6DoF grasp quality prediction, even in partially observed scenes. NeuGraspNet operates on random viewpoints, common in mobile manipulation scenarios, and outperforms existing implicit and semi-implicit grasping methods. We demonstrate the real-world applicability of the method with a mobile manipulator robot, grasping in open cluttered spaces.**

I. Introduction

Robotic manipulation is crucial for enabling various applications such as home-assistance, industrial automation etc. A key component for manipulation is the ability to grasp objects in unstructured, cluttered spaces under partial observability. This ability would enhance the efficiency, versatility, and autonomy of robots operating in everyday environments. Deep learning has been crucial in making advances in robotic grasping [37, 38, 44, 9] by training networks using simulation data and transferring to the real world. However, 6DoF grasping in the wild, i.e., grasping in the SE(3) space from *any viewpoint* remains a challenge [32, 47, 53]. Embodied AI agents, e.g., mobile manipulation robots [27, 25], are expected to perform manipulation tasks similar to humans; humans can leverage geometric information from limited views and mental object models to grasp objects without exploring the whole scene. Such grasping in open cluttered spaces requires that robots, given some spatial information, e.g., 3D pointcloud data, can reconstruct the scene, understand graspable areas of objects, and detect grasps likely to succeed. Moreover, robots should reason about the grasp's affordance [21], i.e., the subsequent task a grasp allows, while additionally avoiding collision with the surrounding environment.

6DoF grasping methods can be classified into methods that *explicitly generate* grasp poses [45, 61, 58] or *implicitly* classify the grasp quality of *any* grasp candidate in SE(3) using a discriminative model [59, 34, 26]. The ability to assess the quality of any grasp pose implicitly is essential to applications in which grasp candidates are pre-defined due to human demonstrations [67] or other affordance-based information [15, 33]. Moreover, explicit generative models are difficult to combine with additional constraints since the constraints can only be applied as a post-filtering step. In implicit methods, however, the distribution of grasp candidates can be chosen and constrained *before* querying the model for grasp quality. This ability is particularly useful in mobile manipulation tasks where constraints such as reachability of grasp poses are necessary.

Many existing grasping methods that use partial pointclouds either rely only on seen parts of a scene [48, 58] or accumulate more information from multiple views [5, 28, 20]. An approach to mitigate partial observability is to use neural scene representations [39, 52, 51] to learn scene completion in a continuous functional space. These representations are implicit in geometry, enabling querying arbitrary points in the scene. They also allow the learning of other geometric feature fields, making them an attractive solution for grasping [29, 17, 35, 56]. However, most neural scene representations still require multi-view information or overfit to specific objects/scenes.

This work investigates how to effectively leverage geometric and surface information about objects in *any scene* perceived from *any partial view* to detect high-fidelity 6DoF grasps. We propose a novel method, **NeuGraspNet**[1] for 6DoF grasp detection building on advances in neural surface rendering [49, 50]. We use a learned implicit scene representation to reconstruct and render the scene globally and effectively sample grasp candidates, even in occluded regions. Moreover, we argue that local geometric object features are essential for understanding the complementarity between the robot's end-effector and the object's surface for predicting grasp success. Thus, we treat *grasping as local neural surface rendering*. We learn shared local features that encode the response of an object part to a grasping pose, enabling fully implicit grasp quality evaluation in SE(3). To evince the benefit of NeuGraspNet, we show superior performance compared to representative implicit and semi-implicit baselines. We also demonstrate real-world applicability via sim-to-real transfer to a mobile manipulator robot grasping in open spaces.

In summary, our contributions are: (i) a new neural grasp quality network (NeuGraspNet) that operates on single viewpoints, *generalizes to any scene*, and is *fully implicit* in both SE(3) and scene/object geometry, (ii) a new structured method for sampling grasp candidates in occluded parts of the scene using global (scene-level) rendering, (iii) the re-interpretation of *grasping as surface rendering* for extracting local geometric features that capture the interaction between the robot's end-effector and the local object-geometry per grasp candidate.

II. Background & Related Work

A. Neural implicit scene representations

Implicit scene representations are useful since they encode the scene in a continuous feature space, allowing them to be queried at any resolution. Occupancy networks [39, 52] are neural representations for 3D reconstruction. Given a pointcloud or 3D voxel grid of the scene, they encode the scene in a feature space ψ using convolutional encoders which combine global and local scene information. Neural surface rendering approaches [49, 50, 62] employ occupancy or surface models [51, 62] to render realistic 3D object surfaces. While some learned geometric representations require explicit 3D supervision [39, 51, 52], advances in differentiable rendering [40] and surface model learning have also enabled reconstruction from multi-view 2D images only [49, 62]. While multi-view image-based methods such as NeRFs [40] are restricted to overfitting specific scenes, some initial success has been found towards generalization to arbitrary scenes [69].

[1]The acronym hints at the novel view of grasping as neural surface rendering, and a wordplay for the German word 'neu' that means new.

B. Grasp detection

A comprehensive review on deep grasp synthesis is introduced in [47]. We refer here to a subset of works covering explicit grasp generative models, implicit (discriminative) models, and grasp affordance detection.

Explicit grasp generation. Partial observability can greatly hinder explicit grasp pose generation [36, 1, 66, 42]. GraspNet [45] is a variational autoencoder generative model that learns a grasp distribution in SE(3), but additionally relies on an iterative grasp refinement process. ContactGraspNet[58] extends this approach by relying on a gripper contact point assumption in the input pointcloud. A similar keypoint-based representation that encodes contact points and the grasp center is proposed in [12, 11]. In [22, 61], the authors show complementarity between RGB and pointcloud data while predicting graspness over piled scenes. AnyGrasp [20] shows robust performance and also proposes a method for finding temporal associations among grasps in consecutive observations.

(Semi-)Implicit grasp prediction. Implicit grasping models can predict a quality value of *any* provided grasp candidate [59, 6]. PointNetGPD [34] learns pointcloud features for predicting the quality of grasp candidates sampled using the method from [23]. VGN [5] uses a scene TSDF to perform semi-implicit grasp detection (implicit in 3D position and explicit in orientation). Other semi-implicit methods have used coarse-to-fine predictions [28] or Hough voting [24]. EdgeGraspNet [26] follows ideas like the contact-based representation from [7] and proposes a point-and-edge-based grasp representation producing SOTA results. Implicit geometric representations [39, 52, 40] provide a valuable feature space for robot perception [30, 17]. GIGA [29] extends VGN by sharing implicit features between scene reconstruction and grasp prediction, improving the primary grasping task. In [65, 60], 6DoF grasping functions are learned as cost functions for use in trajectory optimization, while CenterGrasp [14] learns a semi-implicit grasp-object manifold.

Grasp affordances. It is important to consider the task or ability *afforded* by parts of objects [19, 55] when interacting with them. Several methods have extended object affordances to the affordance provided *by the grasp* to enable task-oriented manipulation [31]. TaskGrasp [46] follows [2] in using semantic knowledge graphs to generalize affordance prediction among similar objects. In [31, 3], human activity videos are used to predict visual affordances. Chen et al. [10] jointly learn visual affordances and grasping to guide grasp candidates toward requested affordances. Recently proposed open-vocabulary methods [56, 57] predict semantic affordances using language queries. These methods, however, use multi-view inputs and are restricted to overfitting to single scenes.

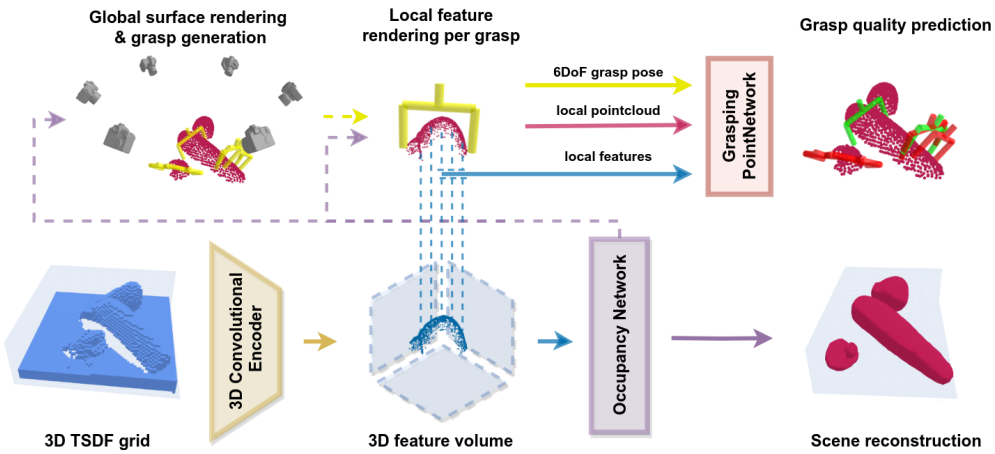


Fig. 2: **NeuGraspNet**: A single-view 3D Truncated Signed Distance Field (TSDF) grid is processed through a convolutional occupancy network to reconstruct the scene (cf. Section III-A). The occupancy network is used to perform global, scene-level rendering. The rendered scene is used for grasp candidate generation in SE(3) (cf. Section III-B). We re-interpret *grasping as rendering* of local surface points and query their features from the *shared* 3D feature volume. Local points, their features, and the 6DoF grasp pose are passed to a Grasping PointNetwork to predict per grasp quality (cf. Section III-C). NeuGraspNet effectively learns the interaction between the objects' geometry and the gripper to detect high-fidelity grasps.

Our approach. Our goal is to perform 6DoF grasping using *any-view* partial depth information. We also aim to *generalize* to any scene, unlike many neural geometric methods that fit to specific scenes/objects [56, 57]. Our novel method uses surface rendering at *multiple levels* to learn relevant grasping features. Our method is *fully implicit* in *both* scene geometry and the 6 grasp dimensions since we learn the complementarity between the grasp and the scene via feature rendering. The method is demonstrated via sim-to-real transfer to real, random-view mobile manipulation scenarios. We also extend the approach to grasp affordances and show promising initial grasp affordance prediction results toward task-oriented grasping.

III. Learning 6DoF Grasping via Neural Surface Rendering

In a cluttered scene, we are given a 3D depth pointcloud $\mathbf{x}$ captured from an arbitrary viewing angle. We consider a robot equipped with a two-fingered gripper and represent grasps using the 6D gripper pose $\mathbf{g} \in$ SE(3). Given the scene information $\mathbf{x}$, the likelihood of success of a grasp is interpreted as the grasp's quality $q \in [0, 1]$.

In this setting, our model, NeuGraspNet, learns an implicit function $f_\theta^{sg} : R^3 \rightarrow R$ that represents the *scene geometry* and a subsequent implicit function $h_\omega^q : R^6 \rightarrow R$ that evaluates the *quality of candidate grasp poses* in the scene (θ and ω being jointly trainable network parameters), leading to a *fully implicit* representation. Our full pipeline is depicted in Fig. 2. In the following, we describe our reconstruction and scene-level rendering approach. We then explain the candidate grasp generation method that samples grasps on the rendered scene. This is followed by our implicit grasp quality evaluation function that uses local grasp-level surface rendering to learn appropriate features relevant to each grasp.

A. Neural scene reconstruction

Our input pointcloud $\mathbf{x}$ only conveys partial information about the scene. To enable grasp generation in a scene-level feature space, we reconstruct the scene through an implicit scene geometry backbone, a convolutional occupancy network (ConvONet) [52]. The network first encodes the input pointcloud $\mathbf{x}$ into a feature space ψ. We can then implicitly query the occupancy probability of any 3D point $\mathbf{p} \in R^3$ by passing the corresponding feature vector $\psi(\mathbf{x}, \mathbf{p})$ through a fully-connected scene geometry decoder network f_θ^{sg}. The network is trained using ground truth simulated occupancy values $o(\mathbf{p}) \in \{0, 1\}$ of points uniformly sampled in the scenes. Formally, the reconstruction loss is a binary cross-entropy loss between the predicted occupancy probability $\hat{o}(\mathbf{p}) = f_\theta^{sg}(\psi(\mathbf{x}, \mathbf{p}))$ and the true occupancy $o(\mathbf{p})$:

$$\mathcal{L}_{occ} = \mathcal{L}_{CE}\left(\hat{o}(\mathbf{p}), o(\mathbf{p})\right) \tag{1}$$

Learning scene geometry as an auxiliary task for grasp quality prediction has been shown to be beneficial to grasp prediction [35, 29]. In this work, we adopt this intuitive idea and further highlight the importance of scene reconstruction not only as an auxiliary training task but as a core component of NeuGraspNet. Crucially, we use the learned implicit geometry to render the scene surface at a global level for grasp candidate generation (cf. Section III-B) and at a local level for feature extraction per 6D grasp candidate pose (cf. Section III-C).

B. Scene-level rendering & grasp candidate generation

Using the learned occupancy representation f_θ^{sg}, we can render the surface of the whole scene at inference time by ray-marching C "virtual" cameras placed in a circular path around the scene (Figure 3). We perform surface rendering using a

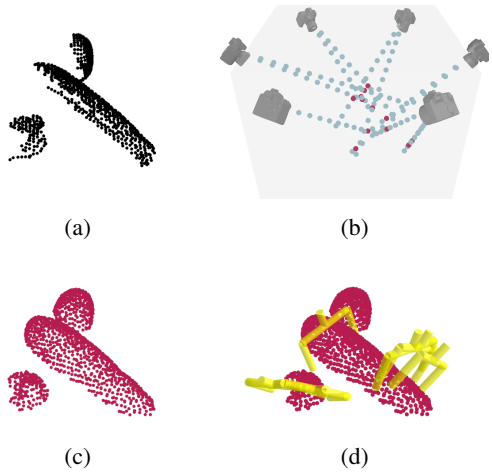


Fig. 3: Scene-level surface rendering: (a) an input single-view pointcloud; (b) surface rendering on the neural implicit geometry (grey volume) using 6 'virtual' cameras; (c) the reconstructed surface pointcloud; (d) sampled grasp candidates.

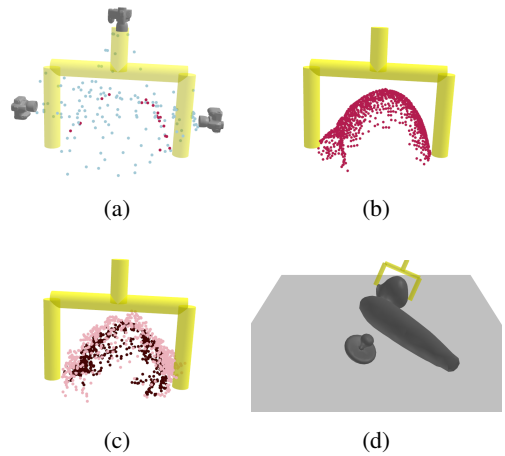


Fig. 4: Local surface rendering: (a) rendering the neural implicit geometry by ray-marching 3 'virtual' cameras at the three parts of the gripper (gripper used here only for visualization); (b) the neural rendered surface; (c) noisy ground-truth rendered surface used during training for local occupancy supervision (light pink points are unoccupied and dark red points are occupied); (d) ground-truth simulated scene.

root-finding approach similar to [49] and [50]. Formally, for every ray r emanating from each of the C virtual cameras, we evaluate the occupancy network $f_\theta^{sg}(.)$ at n equally spaced samples on the ray $\{\mathbf{p}_j^r\}_{j=1}^n$. The first point along a ray for which the occupancy changes from free space ($o(\mathbf{p}) < 0.5$) to occupied space ($o(\mathbf{p}) > 0.5$) is a surface point $\mathbf{p}_s$. To further refine the surface point estimation, we use an iterative secant search method along the ray (detailed in [49]). After obtaining the surface point-set from all virtual cameras, we merge and downsample the point-set to arrive at a reconstructed scene pointcloud $\{\mathbf{p}_i\}_{i=1}^M$ as shown in Fig. 3c. We note here that other reconstruction methods, such as running marching cubes over the occupancy volume, could also be used to obtain the surface points. However, since we are only interested in the surfaces of the objects that the robot can interact with, our ray-casting procedure, which focuses on the ray's first contact with the upper surfaces of objects, reconstructs surfaces at a high resolution with fewer queries to the implicit feature volume.

Our grasp detection is implicit, i.e., we can query the quality of any 6DoF grasp pose $\mathbf{g} \in$ SE(3). To generate suitable grasp proposals to discriminate upon during training, we use the sampling approach of Grasp Pose Generator (GPG) [23] due to its simple yet effective nature. GPG generates grasp candidates on the input pointcloud using the point surface normals and estimating the axis of curvature of a surface. For more details, we refer to [23]. We apply the GPG sampler on our *reconstructed* surface pointcloud. This provides two benefits over the partial pointcloud. First, we can sample more grasps, since we can use the *completed* scene information, i.e., sample grasps in occluded areas. Secondly, our candidates are less likely to be in collision with objects. This is because, in the partial pointcloud case, generated candidates can often intersect with objects because only parts of the object pointclouds are visible. A visualization of the grasps generated on our completed pointcloud can be seen in Fig. 3d.

C. Local surface rendering & learning grasping functions

Key to our approach is to obtain features that effectively capture the geometric interaction between any 6DoF grasp and the scene. To do so, we propose selecting the features from the scene's local surface points which are *rendered* by each grasp. We hypothesize that the local surface points and their *corresponding latent features* can effectively encode the geometric complementarity of object-surface and gripper to assess the quality of grasps. For each 6DoF grasp pose $\mathbf{g}$ in a scene, we use our implicit scene-geometry network f_θ^{sg} to render a local 3D surface point set $\{\mathbf{p}_i^{\mathbf{g}}\}_{i=1}^N$ corresponding to the grasp $\mathbf{g}$. Specifically, we place a virtual camera near each link of the gripper, i.e., near the two fingers and the base, such that the cameras point towards the inner hull of the gripper (Figure 4a). We thus obtain a dense local point-set and filter out spurious points too far away from the gripper links.

Grasp quality prediction. To predict the quality of a grasp $\mathbf{g}$, we use both the local surface point set of the grasp $\{\mathbf{p}_i^{\mathbf{g}}\}_{i=1}^N$ and the corresponding feature vectors in the *same feature space* $\psi(\mathbf{x})$ as the scene geometry network f_θ^{sg}. Thus, we jointly learn features ψ appropriate for both scene reconstruction and grasp quality prediction [29]. The N local surface points and their features are passed through an implicit grasp quality decoder network h_ω^q to predict the grasp success probability $\hat{q}(\mathbf{g}) = h_\omega^q\big(\{\mathbf{p}_i^{\mathbf{g}}, \psi(\mathbf{x}, \mathbf{p}_i^{\mathbf{g}})\}_{i=1}^N\big)$. The grasp quality decoder uses a permutation invariant point network architecture [54], trained using simulated ground truth labels $q(\mathbf{g}) \in \{0, 1\}$ of grasp success/failure and a binary cross-entropy loss

$$\mathcal{L}_{qual} = \mathcal{L}_{CE}\left(\hat{q}(\mathbf{g}), q(\mathbf{g})\right) \tag{2}$$

Local supervision using ground truth surfaces. Learning grasping functions based on surface features poses a challenge. Since we rely on local neural surface point rendering to pick appropriate point-wise features, the scene reconstruction needs to be accurate. A straightforward approach would be to train the grasp quality network h^q_ω in a subsequent step after the convergence of the scene reconstruction network f^{sg}_θ. However, we wish to train grasp quality and scene reconstruction *jointly* to ensure that the implicit geometric features ψ also capture information relevant for assessing grasp quality. Thus, we propose to use additional local surface supervision at training time. During the data generation in simulation, we obtain the ground truth surface points for each grasp: $\{\mathbf{p}^{\mathbf{g}}_{gt\ i}\}_{i=1}^N$. We then add noise to these points, as visualized in Fig 4c. These noisy surface points $\{\mathbf{p}^{\mathbf{g}}_{noisy\ i}\}_{i=1}^N$ serve two purposes. First, we can *train a robust grasp quality network* using these noisy surface points and their features ψ using the loss from Equation (2). Thus we only need to perform the local neural surface rendering at *inference time* while also ensuring regularization against imperfect neural surface renders. Second, we add additional supervision to the scene reconstruction using the occupancy values for these dense yet noisy surface points $o(\mathbf{p}_{noisy})$, refining the occupancy in scene parts that interact with the gripper during grasping (cf. Fig. 4). We use a loss function $\mathcal{L}_{local}$, similar to Equation (1), to additionally train *local object-part reconstruction* from these noisy surface points.

Grasp affordance prediction We also experiment with extending our approach to, additionally, implicitly predict grasp *affordances* towards task-oriented grasping. Using the local rendered features, we train another decoder network d^a_γ with parameters γ to predict the grasp affordance $\hat{a}(\mathbf{g})$. The affordances belong to a set of K object affordance classes from 3D AffordanceNet [19]. Given that a grasp can have more than one affordance, this becomes a multi-label classification problem for which we can use separate heads to predict the probability of each affordance class $\{a_k\}_{k=1}^K$. For this, we use another binary cross entropy loss between the predicted affordance $\hat{a}_k(\mathbf{g}) = d^{a_k}_\gamma\big(\{\mathbf{p}^{\mathbf{g}}_i, \psi(\mathbf{x}, \mathbf{p}^{\mathbf{g}}_i)\}_{i=1}^N\big)$ and the ground truth label $a_k(\mathbf{g}) \in \{0, 1\}$ for every class k. Moreover, since the multi-label dataset is imbalanced and can contain many more negative labels than positive ones, we add a dice loss as proposed in [19]. This loss ensures that the total error in the prediction of positive and negative labels in a batch is weighted equally. The overall affordance loss is then

$$\mathcal{L}_{aff} = \sum_{k=1}^{K}\big(\mathcal{L}_{CE}\left(\hat{a_k}(\mathbf{g}), a_k(\mathbf{g})\right) + \mathcal{L}_{DICE}\left(\hat{a_k}(\mathbf{g}), a_k(\mathbf{g})\right)\big).$$

D. Implementation

Our network uses a convolutional occupancy network encoder that encodes a scene Truncated Signed Distance Field (TSDF) (processed from the input pointcloud) into an implicit 3D feature space. For scene reconstruction, the decoder f^{sg}_θ is a ResNet-based fully-connected network (as in [52]). For grasp quality, the decoder h^q_ω uses a point network [54]. The overall loss during training includes the occupancy loss $\mathcal{L}_{occ}$, quality loss $\mathcal{L}_{qual}$ as well as the additional local occupancy loss $\mathcal{L}_{local}$, each weighted by factors w_i,

$$\mathcal{L}_{NeuGrasp} = w_{occ}\mathcal{L}_{occ} + w_{qual}\mathcal{L}_{qual} + w_{local}\mathcal{L}_{local}. \quad (3)$$

For affordance prediction, we also use the affordance loss to train separate heads for each affordance class:

$$\mathcal{L}_{NeuGraspAff} = \mathcal{L}_{NeuGrasp} + w_{aff}\mathcal{L}_{aff}. \quad (4)$$

Additional implementation details are provided in Appendix A. We also experiment with incorporating rotational SO(3) equivariance in the encoder [18] and using different point network decoder architectures such as DGCNN [63]. Experiments with these architectures are provided in Appendix C.

IV. Experiments

While evaluating our proposed approach, we aim to answer the following questions:

1) Can NeuGraspNet outperform representative implicit and non-implicit baseline 6DoF grasping methods?
2) What are the contributions of our rendering procedure at scene-level and at the local-level towards predicting accurate grasp quality?
3) Given that our method utilizes scene reconstruction, how well does the method *generalize* to unseen scenes with hard test objects where precise reconstruction is challenging?
4) How well does our approach extend to predicting grasp affordances toward task-oriented grasping?
5) How well does our method, trained in simulation, transfer to the real world?

A. Grasp Quality Prediction

Experimental setup. For training and evaluating NeuGraspNet, we first use the popular simulation benchmark from VGN [5], also used in [29, 26, 14], that uses the PyBullet simulator [16]. In the VGN setup, simulated objects are spawned in a random 'pile' or are 'packed' upright on a surface. The benchmark consists of 343 household objects split into 303 training and 40 testing objects. On average, five objects are spawned per scene. The only modification we do to make the task more realistic and challenging is to *randomize* the camera viewing angle when generating the scene TSDF or pointcloud. In addition to the VGN household object set, we also evaluate our method and baselines on test objects from the Evolved Grasping Analysis Dataset (EGAD) [43] *without re-training the networks*. EGAD contains objects of varying shape complexities, and testing on them helps determine the generalizability of our method, especially to situations where precise reconstruction is challenging.

Data generation and training. Since our network is implicit in all 6 grasp dimensions (unlike semi-implicit methods [5, 29]), we create a dataset with a larger number of grasps per scene to learn to discriminate grasps in SE(3). We execute grasp candidates sampled using the method from Section III-B

TABLE I: Comparative results of NeuGraspNet vs. baselines on Pile and Packed scenes [5] (5 seeds)

		Pile scenes						Packed scenes					
		Fixed Top View		Random View		Hard View		Fixed Top View		Random View		Hard View	
Method	Type	GSR (%)	DR (%)	GSR (%)	DR (%)	GSR (%)	DR (%)	GSR (%)	DR (%)	GSR (%)	DR (%)	GSR (%)	DR (%)
NeuGraspNet (ours)	Implicit	**86.51 ± 1.42**	**83.52 ± 2.24**	**85.05 ± 1.25**	**84.37 ± 1.52**	**73.95 ± 1.26**	**70.67 ± 1.69**	**97.65 ± 0.92**	**93.16 ± 1.48**	**92.49 ± 1.41**	**91.74 ± 1.24**	78.76 ± 1.89	82.80 ± 1.50
PointNetGPD [34]	Implicit	79.79 ± 2.28	77.81 ± 2.79	70.94 ± 3.12	68.88 ± 3.11	47.42 ± 3.40	36.02 ± 3.68	81.14 ± 2.52	86.04 ± 1.50	71.94 ± 1.38	76.23 ± 2.51	25.15 ± 2.61	14.18 ± 1.50
Scene-render+PNGPD	Implicit	79.64 ± 2.35	78.02 ± 1.37	71.85 ± 1.76	68.45 ± 3.74	52.72 ± 4.89	40.77 ± 4.10	80.16 ± 2.33	84.93 ± 2.40	73.23 ± 1.45	76.86 ± 2.11	39.10 ± 4.47	25.38 ± 2.61
VGN [5]	Semi-Implicit	77.44 ± 2.15	63.98 ± 5.03	78.48 ± 1.45	74.09 ± 1.16	68.46 ± 2.55	64.14 ± 4.37	83.42 ± 0.85	54.08 ± 5.35	78.11 ± 0.81	60.13 ± 1.96	70.27 ± 3.18	38.57 ± 3.40
EdgeGraspNet [26]	Implicit	80.25 ± 1.41	83.18 ± 1.43	78.76 ± 1.16	80.89 ± 2.65	68.11 ± 2.63	69.32 ± 3.79	85.09 ± 2.48	85.36 ± 2.74	86.06 ± 0.75	86.51 ± 0.92	**81.61 ± 1.41**	**85.11 ± 0.84**
GIGA [29]	Semi-Implicit	82.92 ± 2.08	73.58 ± 2.93	78.67 ± 1.86	75.99 ± 1.79	69.13 ± 4.43	64.83 ± 5.63	96.05 ± 0.20	76.81 ± 3.21	87.99 ± 0.84	75.64 ± 2.75	73.87 ± 1.57	68.52 ± 4.49

TABLE II: Ablation results of NeuGraspNet on Pile and Packed scenes [5] (5 seeds)

	Pile scenes						Packed scenes					
	Fixed Top View		Random View		Hard View		Fixed Top View		Random View		Hard View	
Method	GSR (%)	DR (%)	GSR (%)	DR (%)	GSR (%)	DR (%)	GSR (%)	DR (%)	GSR (%)	DR (%)	GSR (%)	DR (%)
NeuGraspNet (ours)	86.51 ± 1.42	83.52 ± 2.24	**85.05 ± 1.25**	**84.37 ± 1.52**	**73.95 ± 1.26**	**70.67 ± 1.69**	**97.65 ± 0.92**	**93.16 ± 1.48**	**92.49 ± 1.41**	**91.74 ± 1.24**	**78.76 ± 1.89**	**82.80 ± 1.50**
No-scene-render	85.79 ± 1.38	83.44 ± 2.40	83.57 ± 1.91	83.06 ± 1.50	69.71 ± 2.48	65.61 ± 3.37	97.18 ± 1.14	92.80 ± 1.75	89.83 ± 1.79	90.19 ± 1.57	56.43 ± 4.62	29.08 ± 3.10
No-local-render	79.83 ± 2.06	77.22 ± 2.72	77.04 ± 2.57	76.17 ± 2.51	63.51 ± 3.05	58.24 ± 3.14	96.31 ± 0.93	92.17 ± 1.43	89.81 ± 1.37	90.10 ± 0.70	73.86 ± 3.31	76.12 ± 2.16
No-shared-features	80.72 ± 1.96	78.15 ± 2.672	77.20 ± 2.36	78.01 ± 2.47	64.44 ± 2.96	59.34 ± 3.17	95.44 ± 0.96	90.26 ± 1.48	88.71 ± 1.43	89.11 ± 0.83	71.24 ± 3.51	69.06 ± 2.33
No-local-occ	**86.62 ± 1.75**	**83.74 ± 2.41**	84.37 ± 1.51	83.72 ± 0.89	72.08 ± 1.47	69.10 ± 2.13	96.53 ± 1.53	92.24 ± 1.37	90.62 ± 2.13	90.29 ± 1.56	78.14 ± 2.04	80.70 ± 1.73
No rendering	73.59 ± 1.58	72.92 ± 2.82	73.36 ± 0.84	72.79 ± 1.38	56.52 ± 1.11	50.30 ± 2.75	93.65 ± 0.76	91.46 ± 0.37	87.32 ± 1.89	88.34 ± 1.47	52.17 ± 4.12	26.88 ± 2.31

on simulated scenes and train both NeuGraspNet and the baselines on this dataset. We generate a dataset of 1.4 million grasps in 33,313 scenes for 'pile' scenes and 1.2 million grasps in 33,534 scenes for 'packed' scenes, balanced with both successful and unsuccessful grasps. We also sample 100,000 occupancy points per scene to train the scene reconstruction. More details about the dataset are provided in Appendix B.

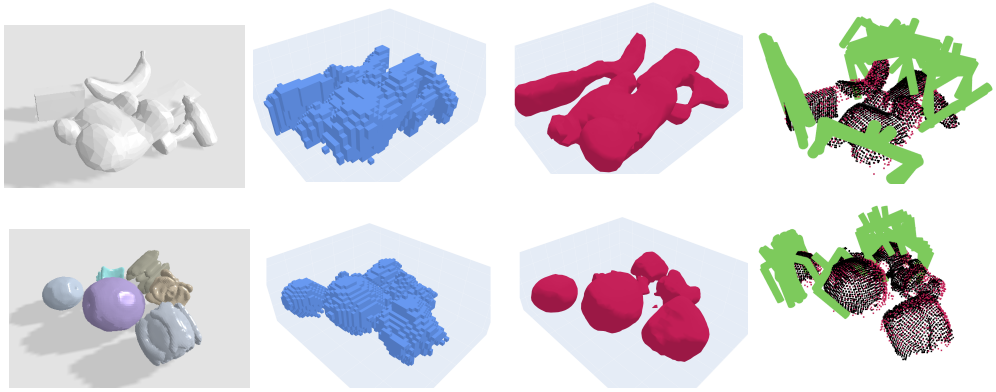
Baselines & metrics. We compare our method against strong baselines that use various representations for predicting grasp quality. Due to the scope of our work for fully implicit grasp detection, we emphasize comparison with implicit or semi-implicit methods. We compare against **PointNetGPD** [34], which operates directly on pointclouds, and experiment with adding our scene completion pipeline as preprocessing for PointNetGPD, termed **Scene-render+PNGPD**. We also compare against the current state-of-the-art **EdgeGraspNet** [26] that proposes a contact-edge-based grasp representation. Note that for EdgeGraspNet, we generate the dataset as per the sampling strategy in [26] as it requires grasp samples on edges.Additionally, we compare against the semi-implicit methods **VGN** [5] and **GIGA** [29], with the latter considering scene reconstruction as an auxiliary loss. We train VGN and GIGA with high-resolution TSDFs (as also done for NeuGraspNet), as it leads to better performance. We also run ablations of our model with different settings, e.g., with and without scene-level rendering, etc., to demonstrate the efficacy of our full model. We train all methods from scratch on our dataset with random-view (random elevation between 15 and 75 degrees) pointcloud or TSDF inputs for a fair comparison. Though our work focuses on implicit grasp quality prediction, for completeness, we also compare against two popular generative grasping networks, **AnyGrasp** [20] and **ContactGraspNet** [58]. For these, we use pre-trained networks and hence mainly test on the EGAD object setting where all the networks are evaluated without re-training. More details about training and evaluation are in Appendix B.

The main focus of this work is to evaluate the representation capability of NeuGraspNet, which learns the interaction between scene/object geometry and the robot end-effector for detecting high-quality 6DoF grasps. Our benchmarking setup consists, therefore, of three settings. (I) **Fixed Top View**: In this setting, the camera is placed at a fixed viewing angle: a 60-degree elevation, and at a distance of twice the workspace width (width in this setup is 30cm), as in [29]. (II) **Random view**: This setup resembles the one of training; the viewing elevation is sampled between 15 and 75-degrees, and the distance of the camera is between 1.6 and 2.4 times the workspace width. This setting is harder than the one used in [26] that considers angles between 30 and 45 degrees. (III) **Hard view**: While we sample the camera's distance like the random view setting, we fix the viewing angle at 15 degrees. We argue that such views are realistic for robots like mobile manipulators that can move in the scene to grasp scattered objects with a limited field of view. Other methods that use fixed manipulators operating on tabletop surfaces have not considered such a setting. All these settings are tested on the 'pile' and 'packed' scenes of [5] and 'pile' scenes with EGAD [43] objects. To reconstruct the scene, we use N=6 virtual cameras in a circular trajectory around the scene. From our experiments, we note that using less than 4 virtual cameras is insufficient to reconstruct the surfaces well for the considered grasping scenes, while more than 6 virtual cameras do not provide any benefit. We use the same grasping metrics as [29, 5, 26], namely, the Grasp Success Rate (GSR), i.e., percentage of successful grasp executions w.r.t. the total attempts, and the Declutter Rate (DR), i.e., percentage of objects successfully removed to the number of total objects presented in the scenes. We perform 100 simulation rounds for 5 random seeds in all cases and report the average percentage and standard deviation per metric. As per the experiment setup of [5], a simulation round ends when (a) no objects remain, (b) no high-quality grasps are found, or (c) two consecutive failed grasp attempts have occurred.

1) Comparison with baselines on VGN [5] scenes: Observing Table I, in 'pile' scenes, which are more unstructured, NeuGraspNet outperforms the baselines in all settings. EdgeGraspNet and GIGA perform similarly in GSR, but EdgeGraspNet has a higher DR, indicating fewer consecu-

TABLE III: Grasping performance in simulated Pile scenes with objects from the EGAD [43] dataset. The networks are not re-trained but directly tested on EGAD objects of different complexities (5 seeds)

Method	Type	Fixed Top View		Random View		Hard View	
		GSR (%)	DR (%)	GSR (%)	DR (%)	GSR (%)	DR (%)
NeuGraspNet (ours)	Implicit	**79.76 ± 1.54**	**81.82 ± 1.45**	**75.24 ± 1.01**	**76.80 ± 1.29**	**67.21 ± 2.86**	**62.04 ± 4.94**
PointNetGPD [34]	Implicit	65.71 ± 3.20	64.73 ± 3.19	59.05 ± 2.50	56.19 ± 4.68	37.64 ± 1.16	25.99 ± 0.95
VGN [5]	Semi-Implicit	13.59 ± 1.16	11.71 ± 0.59	11.42 ± 0.54	9.98 ± 0.25	5.56 ± 0.56	4.72 ± 0.72
EdgeGraspNet [26]	Implicit	40.59 ± 2.38	31.99 ± 2.37	40.75 ± 1.97	31.81 ± 2.20	39.50 ± 1.91	30.03 ± 2.66
GIGA [29]	Semi-Implicit	27.49 ± 3.05	17.65 ± 2.42	25.53 ± 2.84	16.03 ± 2.52	17.10 ± 1.96	9.28 ± 1.42
ContactGraspNet [58]	Generative	48.98 ± 4.27	40.85 ± 2.96	42.25 ± 2.35	38.66 ± 2.48	30.46 ± 2.74	22.64 ± 1.28
AnyGrasp [20]	Generative	68.46 ± 2.50	68.06 ± 2.74	66.09 ± 1.01	66.83 ± 1.86	56.23 ± 0.97	47.43 ± 0.95



(a) Scene (b) Input TSDF (c) Reconstruction (d) Detected Grasps

Fig. 5: Example scene reconstructions & detected grasps for unseen test objects from the VGN [5] (top) and the EGAD [43] (bottom) datasets. We see that our network can sometimes create artifacts or is unable to reconstruct very fine details, especially for the hard EGAD objects. Nevertheless, even in these hard cases, our network is able to reconstruct the broad structure of the scene & objects which results in the detection of good grasps (d).

tive failures. In 'packed' scenes, we observe a similar high performance by NeuGraspNet. However, there are interesting remarks regarding the baselines. GIGA still performs well in fixed top view settings, where most of the scene is still observed. VGN performs close to GIGA in random and hard views. PointNetGPD performs reasonably in the fixed top view setting since it uses the GPG [23] grasp sampling strategy, which favors top grasps. However, it performs poorly in hard view settings since top surfaces of packed objects are rarely observed. Our version of PointNetGPD that uses scene completion, i.e. Scene-render+PNGPD, performs very similar to PointNetGPD in the fixed view cases, but shows improvements of up to 5-10% in hard views. EdgeGraspNet showcases good performance in 'packed' scenes, and with 'packed' hard views, it outperforms NeuGraspNet by $\sim$2.5% in GSR and DR. Note that we test EdgeGraspNet with their proposed edge grasp sampling strategy. In 'packed' scenes, this edge strategy favors side grasps, unlike GPG, which favors top grasps. This characteristic of GPG also hurts performance of NeuGraspNet in 'packed' hard-view settings. While we have shown gains due to our scene-level GPG approach (cf. Section III-B), an improved geometrically-informed sampling strategy could lead to further improvements. Nevertheless, we demonstrate that our local point and feature set successfully encodes the gripper-object complementarity and outperforms all baselines in 5 out of 6 settings.

2) Ablation study: Table II presents an ablation of different components of NeuGraspNet for evaluating the contribution of each component of the network. As the results show, a model without any rendering (global or local), shown at the bottom of the table, performs the worst. In this case the grasp sampler relies only on the input pointcloud, and the grasp quality network struggles to discriminate in the SE(3) space. When only removing global scene-level rendering, we achieve good performance with fixed & random views but see a significant drop with hard views. In hard views, much of the scene is unseen and the grasp sampler struggles to sample reasonable candidates. *Removing the local rendering at the grasp level from NeuGraspNet hurts performance the most* ('No-local-render' in table), underlying the importance of local features that allow learning the interaction of object surface and gripper. Not using a shared feature space for the reconstruction and grasp quality prediction ('No-shared features') also drops performance. We also observe worse performance without local occupancy supervision, especially for DR in 'packed' scenes, where objects are placed closely, and fine details around objects are needed. In this case, providing a refinement signal with local occupancy improves performance. Additional ablations are in Appendix C.

3) Comparison with baselines on EGAD [43]: Table III presents evaluations on piled EGAD [43] object scenes without re-training the networks. We note a very significant perfor-

TABLE IV: Grasping & affordance prediction performance in random-view scenes with PartNet objects [19] (5 seeds)

Method	GSR (%)	DR (%)	Grasp affordance prediction mAP (%)							
			Average	Handover	Cut	Stab	Lift	Wrap	Pour	Wear
NeuGraspAffNet (ours)	**71.78 ± 3.19**	**57.55 ± 3.79**	**64.21 ± 2.49**	**84.34 ± 2.60**	78.61 ± 5.89	78.58 ± 5.89	**35.68 ± 7.96**	**31.70 ± 7.85**	**75.68 ± 8.44**	**64.91 ± 4.63**
No-local-render (ours)	57.48 ± 1.30	39.78 ± 1.81	53.45 ± 3.58	82.86 ± 3.60	**79.05 ± 5.91**	**79.01 ± 5.89**	22.03 ± 8.44	21.71 ± 4.97	65.15 ± 5.96	24.35 ± 8.42
VGN+Aff[5]	16.11 ± 1.71	5.80 ± 0.87	28.73 ± 1.42	56.92 ± 1.18	19.06 ± 2.43	19.06 ± 2.43	21.10 ± 0.63	27.33 ± 2.22	13.23 ± 1.31	44.38 ± 2.06
GIGA+Aff [29]	53.72 ± 3.01	44.78 ± 2.65	42.64 ± 1.64	83.23 ± 0.87	29.26 ± 4.54	29.27 ± 4.54	34.39 ± 5.49	30.92 ± 4.56	34.71 ± 6.67	56.67 ± 5.32

TABLE V: Grasping performance in real-world scenes over 12 scene decluttering rounds

Method	Pile scenes (Random View)		Packed scenes (Random View)	
	GSR (%)	DR (%)	GSR (%)	DR (%)
NeuGraspNet (ours)	**83.63** (46/55)	**76.36** (46/60)	**87.72** (50/57)	**90.90** (50/55)
EdgeGraspNet [26]	64.81 (35/54)	58.33 (35/60)	70.58 (36/51)	65.45 (36/55)

mance drop in all methods due to the diversity in complexity of EGAD objects. However, NeuGraspNet sees the smallest drop, showing greater robustness than baselines. Example visualizations of reconstructions and detected grasps are provided in Figure 5. We note that even in difficult reconstruction scenarios, NeuGraspNet recovers the broad scene structure, leading to good grasps being detected. The performance of implicit and semi-implicit baselines is similar to the 'pile' scenes with VGN objects. The generative AnyGrasp performs better than most other baselines but still significantly lags behind NeuGraspNet in both GSR and DR while Contact-GraspNet performs poorly in hard views but comparably to EdgeGraspNet in fixed and random view settings.

B. Grasp Affordance Prediction

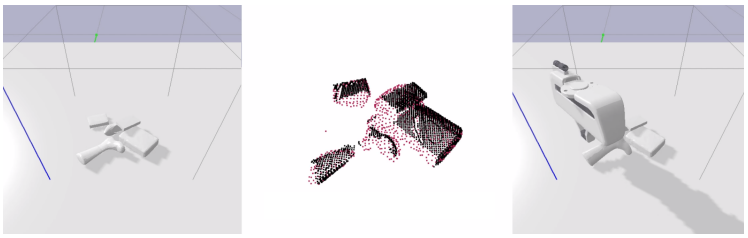
We evaluate the extension of our method to grasp affordance prediction in depth pointclouds and provide initial results.

Data generation and training. There exists a lack of a standard, reproducible benchmark for task-oriented grasping and grasp affordance prediction. Very few methods try to predict grasp affordances in 3D pointclouds while also operating in cluttered scenes. Thus, we generate a grasp affordance dataset using object and semantic affordances from 3DAffordanceNet [19]. We use the same simulation setup as above but use piled objects from [19], i.e., PartNet objects [41]. For simulated successful grasps of an object, we assign grasp affordances by checking grasp proximity to labeled affordance points from 3DAffordanceNet. Note that the dataset is multi-label, i.e., each grasp can have multiple assigned affordances. During data generation, we use a subset of object categories in 3DAffordanceNet that can be expected to be grasped, giving us 1,248 objects from categories 'Knife', 'Scissors', 'Mug', 'Bottle', 'Bag', 'Earphone', and 'Hat'. While this reduces our study to these categories, we note that significant diversity still exists within the objects to determine the extendability of our approach to task-oriented grasping. We end up with 7 affordance classes for grasps on our chosen object set, namely 'Handover', 'Cut', 'Stab', 'Lift', 'Wrap', 'Pour', and 'Wear'. We also remap affordances to be robot-centric, i.e., affordance labels denote actions that can be performed *after* grasping the object. Our dataset contains 2.6 million grasps in 37,696 scenes. We note that PartNet objects are significantly harder to grasp since many of them are too small or large in their original scale. We train our network, called NeuGraspAffNet, and baselines with binary affordance classification heads for each class, given the multi-label nature of the data.

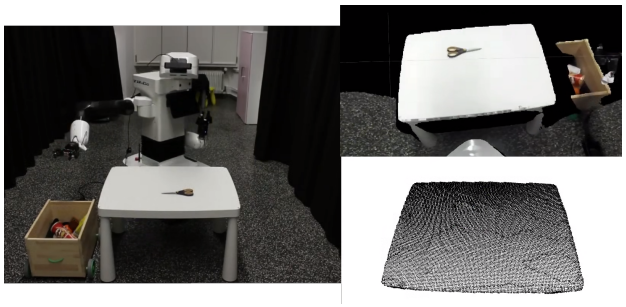
Metrics & comparison with baselines. We split the objects in the dataset into 1096 train and 152 test objects and evaluate grasping success and accurate affordance prediction on scenes with test objects (100 simulation rounds for 5 random seeds). We report the GSR and DR and also include the mean Average Precision (mAP) for affordance classification per class. We also report the average mAP over affordance classes, i.e., the macro-average. Since few methods exist that perform scene-level and shape-based grasp affordance prediction, we compare against modified versions of VGN and GIGA with additional affordance heads **GIGA+Aff** and **VGN+Aff**. For completeness, we also compare against our ablation with no-local-rendering to validate if lack of local rendering has the same effect as in the grasp quality evaluations. The results are listed in Table IV. Consistent with the previous results, NeuGraspAffNet significantly outperforms the baselines regarding success and declutter rates. It also performs better in terms of affordance prediction with an average mAP of 64.21%. Among the affordance classes, the 'Wrap' class is the most difficult to accurately predict, while the 'Handover' and 'Cut' classes are the easiest. An example visualization of grasp affordance prediction is provided in Appendix C in Figure 10. While we've shown the benefit of our approach compared to baselines, there still exists significant scope for improvement in task-oriented grasping for our random-view, scene-level setting. Using visual affordances from 2D RGB images, projecting them into 3D, and integrating them into our reconstruction pipeline is a promising direction for further research.

C. Real-world Evaluation

We perform real-world experiments with a mobile manipulator robot, TIAGo++, equipped with a head-mounted RGBD camera. The robot is placed in house-like scenarios with objects placed randomly in a 'pile' or 'packed' setting. We use 16 objects from the YCB dataset [8] for this experiment. For each setting, we perform 12 decluttering rounds and report the GSR and DR observed. Figure 1 shows example grasp executions in random-view mobile manipulation scenarios such as grasping from a shelf or grasping from the top of a cupboard. We also compare to the best-performing baseline EdgeGraspNet [26]. Table V lists the real-world experiment results. We observe a GSR and DR of 83.63 % and 76.36 %, respectively, for 'pile' scenes and a GSR and DR of 87.72 % and 90.90 %, respectively, for 'packed' scenes. We also demonstrate our affordance prediction capabilities by grasping based on



(a) In this example with input depth from a challenging view (black pointcloud in the middle image), the scene reconstruction (red pointcloud) is not sufficient to avoid grasp failure (right image).



(b) In this example, the thin scissor placed flat on the table is not perceived well by the depth camera mounted on the real robot head, leading to no grasps being detected.

Fig. 6: Example failure cases observed in (a) simulated and (b) real-world experiments.

TABLE VI: Grasp sampling time and inference time comparison between methods (in milliseconds)

Method	Grasp sampling	Grasp quality inference	Total time
NeuGraspNet (ours)	608 ms	257 ms	865 ms
EdgeGraspNet [26]	401 ms	30 ms	431 ms
No-scene-render (ours)	161 ms	257 ms	418 ms
ContactGraspNet [58]	n.a.	280 ms	280 ms
GIGA [29]	n.a.	225 ms	225 ms
AnyGrasp [20]	n.a.	207 ms	207 ms
PointNetGPD [34]	**170 ms**	**23 ms**	**193 ms**

user-queried affordances. Demonstrations are provided in the supplementary material. We note a drop in performance compared to Section IV-A due to the following failure cases. The execution of grasps on objects of relatively large width can sometimes fail due to imprecise perception or execution by the mobile robot, hurting the GSR. On the other hand, very thin objects are not always clearly perceived by the random-view depth camera since it is difficult to differentiate between the object and the table, hurting the DR. Nevertheless, with the use of better camera depth estimation techniques, we expect to get even better performance on real-world de-cluttering of scenes. Additional details and results are provided in Appendix C and the project website: sites.google.com/view/neugraspnet.

V. Conclusion and Limitations

We presented NeuGraspNet, a novel, fully implicit 6DoF-grasp prediction method that re-interprets robotic grasping as surface rendering. NeuGraspNet predicts high-fidelity 6DoF grasps from any random single viewpoint of a scene. Our method exploits a learned implicit geometric scene representation to perform global and local surface rendering. This enables effective grasp candidate generation (using global features) and grasp quality prediction (using local features from a shared feature space). We demonstrated the superior performance of NeuGraspNet in simulated experiments of varying difficulty. We also explored the method's applicability to task-oriented grasping by predicting grasp affordances. Finally, we exhibited the real-world applicability of NeuGraspNet in mobile manipulator grasping experiments.

Our method, however, comes with certain limitations. Firstly, as noted in section IV.C the network struggles to accurately perceive depth and reconstruct very thin objects. An example of this is provided in Figure 6. Moreover, NeuGraspNet requires a computational overhead due to the multiple surface rendering processes, reported in Table VI. Nevertheless, we posit that advances in neural rendering can make such methods increasingly accessible, even with onboard computing. Another limitation is that our model presumes a particular instantiation of gripper. While we have tested with two different grippers, the Franka Panda gripper and the Robotiq 2F-85, the model can struggle with a new gripper of a very different size. This problem could be mitigated by providing the gripper model or size as an input to the network instead, an addition that we wish to explore in future work. A further limitation that affects most grasp detection approaches is the open-loop nature of execution, without consideration of sudden scene changes. Thus, coupling grasp detection (perception) and manipulation (action) dynamically is part of our plans. Finally, while we show promising initial results with respect to shape-based grasp affordances, the reliability and diversity of predicted affordances can be further improved. Projecting pre-trained 2D RGB features, using recent approaches such as generalizable feature fields [68] into our pipeline is a promising direction toward any-scene task-oriented grasping. Moreover, the evaluation of downstream manipulation tasks after the grasp has succeeded is also an important metric to consider in our framework, one that we wish to explore further in future work.

Acknowledgements

This research received funding from the European Union's Horizon program under grant agreement no. 101120823, project MANiBOT and the German Research Foundation (DFG) Emmy Noether Programme (CH 2676/1-1). We also gratefully acknowledge the computing time provided to us on the high-performance computer Lichtenberg at the NHR Centers NHR4CES at TU Darmstadt, Germany. This is funded by the Federal Ministry of Education and Research, and the state governments participating based on the resolutions of the GWK for national high-performance computing at universities (www.nhr-verein.de/unsere-partner).

References

[1] Antonio Alliegro, Martin Rudorfer, Fabio Frattin, Aleš Leonardis, and Tatiana Tommasi. End-to-end learning to grasp via sampling from object point clouds. *IEEE Robotics and Automation Letters*, 7(4):9865–9872, 2022.

[2] Paola Ardón, Eric Pairet, Ronald PA Petrick, Subramanian Ramamoorthy, and Katrin S Lohan. Learning grasp affordance reasoning through semantic relations. *IEEE Robotics and Automation Letters*, 4(4):4571–4578, 2019.

[3] Shikhar Bahl, Russell Mendonca, Lili Chen, Unnat Jain, and Deepak Pathak. Affordances from human videos as a versatile representation for robotics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13778–13790, 2023.

[4] Alexandre Boulch and Renaud Marlet. Poco: Point convolution for surface reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6302–6314, 2022.

[5] Michel Breyer, Jen Jen Chung, Lionel Ott, Roland Siegwart, and Juan I. Nieto. Volumetric grasping network: Real-time 6 DOF grasp detection in clutter. In *CoRL*, volume 155 of *Proceedings of Machine Learning Research*, pages 1602–1611. PMLR, 2020.

[6] Junhao Cai, Jun Cen, Haokun Wang, and Michael Yu Wang. Real-time collision-free grasp pose detection with geometry-aware refinement using high-resolution volume. *IEEE Robotics and Automation Letters*, 7(2): 1888–1895, 2022.

[7] Junhao Cai, Jingcheng Su, Zida Zhou, Hui Cheng, Qifeng Chen, and Michael Y Wang. Volumetric-based contact point detection for 7-dof grasping. In *6th Annual Conference on Robot Learning*, 2022.

[8] Berk Calli, Aaron Walsman, Arjun Singh, Siddhartha Srinivasa, Pieter Abbeel, and Aaron M Dollar. Benchmarking in manipulation research: The ycb object and model set and benchmarking protocols. *arXiv preprint arXiv:1502.03143*, 2015.

[9] Georgia Chalvatzaki, Nikolaos Gkanatsios, Petros Maragos, and Jan Peters. Orientation attentive robotic grasp synthesis with augmented grasp map representation. *arXiv preprint arXiv:2006.05123*, 2020.

[10] Wenkai Chen, Hongzhuo Liang, Zhaopeng Chen, Fuchun Sun, and Jianwei Zhang. Learning 6-dof task-oriented grasp detection via implicit estimation and visual affordance. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 762–769, 2022. doi: 10.1109/IROS47612.2022.9981900.

[11] Yiye Chen, Yunzhi Lin, and Patricio Vela. Keypoint-graspnet: Keypoint-based 6-dof grasp generation from the monocular rgb-d input. *arXiv preprint arXiv:2209.08752*, 2022.

[12] Yiye Chen, Ruinian Xu, Yunzhi Lin, and Patricio A Vela. Kgnv2: Separating scale and pose prediction for keypoint-based 6-dof grasp pose synthesis on rgb-d input. *arXiv preprint arXiv:2303.05617*, 2023.

[13] Yunlu Chen, Basura Fernando, Hakan Bilen, Matthias Nießner, and Efstratios Gavves. 3d equivariant graph implicit functions. In *ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part III*, pages 485–502. Springer, 2022.

[14] Eugenio Chisari, Nick Heppert, Tim Welschehold, Wolfram Burgard, and Abhinav Valada. Centergrasp: Object-aware implicit representation learning for simultaneous shape reconstruction and 6-dof grasp estimation. *arXiv preprint arXiv:2312.08240*, 2023.

[15] Sammy Christen, Muhammed Kocabas, Emre Aksan, Jemin Hwangbo, Jie Song, and Otmar Hilliges. D-grasp: Physically plausible dynamic grasp synthesis for hand-object interactions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 20577–20586, June 2022.

[16] Erwin Coumans and Yunfei Bai. Pybullet, a python module for physics simulation for games, robotics and machine learning. http://pybullet.org, 2016–2021.

[17] Qiyu Dai, Yan Zhu, Yiran Geng, Ciyu Ruan, Jiazhao Zhang, and He Wang. Graspnerf: Multiview-based 6-dof grasp detection for transparent and specular objects using generalizable nerf. *arXiv preprint arXiv:2210.06575*, 2022.

[18] Congyue Deng, Or Litany, Yueqi Duan, Adrien Poulenard, Andrea Tagliasacchi, and Leonidas J. Guibas. Vector neurons: A general framework for so(3)-equivariant networks. In *ICCV*, pages 12180–12189. IEEE, 2021.

[19] Shengheng Deng, Xun Xu, Chaozheng Wu, Ke Chen, and Kui Jia. 3d affordancenet: A benchmark for visual object affordance understanding. In *proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1778–1787, 2021.

[20] Hao-Shu Fang, Chenxi Wang, Hongjie Fang, Minghao Gou, Jirong Liu, Hengxu Yan, Wenhai Liu, Yichen Xie, and Cewu Lu. Anygrasp: Robust and efficient grasp perception in spatial and temporal domains. *arXiv preprint arXiv:2212.08333*, 2022.

[21] James J Gibson. The theory of affordances. *Hilldale, USA*, 1(2):67–82, 1977.

[22] Minghao Gou, Hao-Shu Fang, Zhanda Zhu, Sheng Xu, Chenxi Wang, and Cewu Lu. Rgb matters: Learning 7-dof grasp poses on monocular rgbd images. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 13459–13466. IEEE, 2021.

[23] Marcus Gualtieri, Andreas ten Pas, Kate Saenko, and Robert Platt Jr. High precision grasp pose detection in dense clutter. In *IROS*, pages 598–605. IEEE, 2016.

[24] Dinh-Cuong Hoang, Johannes A Stork, and Todor Stoyanov. Context-aware grasp generation in cluttered scenes. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 1492–1498. IEEE, 2022.

[25] Jiaheng Hu, Peter Stone, and Roberto Martín-Martín. Causal policy gradient for whole-body mobile manipulation. *arXiv preprint arXiv:2305.04866*, 2023.

[26] Haojie Huang, Dian Wang, Xupeng Zhu, Robin Walters,

and Robert Platt. Edge grasp network: A graph-based se(3)-invariant approach to grasp detection. *ICRA*, 2023.

[27] Snehal Jauhri, Jan Peters, and Georgia Chalvatzaki. Robot learning of mobile manipulation with reachability behavior priors. *IEEE Robotics and Automation Letters*, 7(3):8399–8406, 2022.

[28] Kuang-Yu Jeng, Yueh-Cheng Liu, Zhe Yu Liu, Jen-Wei Wang, Ya-Liang Chang, Hung-Ting Su, and Winston Hsu. Gdn: A coarse-to-fine (c2f) representation for end-to-end 6-dof grasp detection. In *Conference on Robot Learning*, pages 220–231. PMLR, 2021.

[29] Zhenyu Jiang, Yifeng Zhu, Maxwell Svetlik, Kuan Fang, and Yuke Zhu. Synergies between affordance and geometry: 6-dof grasp detection via implicit representations. In *Robotics: Science and Systems*, 2021.

[30] Ninad Khargonkar, Neil Song, Zesheng Xu, Balakrishnan Prabhakaran, and Yu Xiang. Neuralgrasps: Learning implicit representations for grasps of multiple robotic hands. In *Conference on Robot Learning*, pages 516–526. PMLR, 2023.

[31] Mia Kokic, Danica Kragic, and Jeannette Bohg. Learning task-oriented grasping from human activity datasets. *IEEE Robotics and Automation Letters*, 5(2):3352–3359, 2020.

[32] Oliver Kroemer, Scott Niekum, and George Konidaris. A review of robot learning for manipulation: Challenges, representations, and algorithms. *The Journal of Machine Learning Research*, 22(1):1395–1476, 2021.

[33] Haoming Li, Xinzhuo Lin, Yang Zhou, Xiang Li, Yuchi Huo, Jiming Chen, and Qi Ye. Contact2grasp: 3d grasp synthesis via hand-object contact constraint, 2023.

[34] Hongzhuo Liang, Xiaojian Ma, Shuang Li, Michael Görner, Song Tang, Bin Fang, Fuchun Sun, and Jianwei Zhang. Pointnetgpd: Detecting grasp configurations from point sets. In *ICRA*, pages 3629–3635. IEEE, 2019.

[35] Jens Lundell, Francesco Verdoja, and Ville Kyrki. Ddgc: Generative deep dexterous grasping in clutter. *IEEE Robotics and Automation Letters*, 6(4):6899–6906, 2021. doi: 10.1109/LRA.2021.3096239.

[36] Jens Lundell, Francesco Verdoja, Tran Nguyen Le, Arsalan Mousavian, Dieter Fox, and Ville Kyrki. Constrained generative sampling of 6-dof grasps. *arXiv preprint arXiv:2302.10745*, 2023.

[37] Jeffrey Mahler, Jacky Liang, Sherdil Niyaz, Michael Laskey, Richard Doan, Xinyu Liu, Juan Aparicio Ojea, and Ken Goldberg. Dex-net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics. *arXiv preprint arXiv:1703.09312*, 2017.

[38] Jeffrey Mahler, Matthew Matl, Xinyu Liu, Albert Li, David Gealy, and Ken Goldberg. Dex-net 3.0: Computing robust vacuum suction grasp targets in point clouds using a new analytic model and deep learning. In *2018 IEEE International Conference on robotics and automation (ICRA)*, pages 5620–5627. IEEE, 2018.

[39] Lars M. Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *CVPR*, pages 4460–4470. Computer Vision Foundation / IEEE, 2019.

[40] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, volume 12346, pages 405–421, 2020.

[41] Kaichun Mo, Shilin Zhu, Angel X. Chang, Li Yi, Subarna Tripathi, Leonidas J. Guibas, and Hao Su. PartNet: A large-scale benchmark for fine-grained and hierarchical part-level 3D object understanding. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[42] Seyed S Mohammadi, Nuno F Duarte, Dimitris Dimou, Yiming Wang, Matteo Taiana, Pietro Morerio, Atabak Dehban, Plinio Moreno, Alexandre Bernardino, Alessio Del Bue, et al. 3dsgrasp: 3d shape-completion for robotic grasp. *arXiv preprint arXiv:2301.00866*, 2023.

[43] 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*, 5(3):4368–4375, 2020.

[44] Douglas Morrison, Peter Corke, and Jürgen Leitner. Learning robust, real-time, reactive robotic grasping. *The International journal of robotics research*, 39(2-3):183–201, 2020.

[45] Arsalan Mousavian, Clemens Eppner, and Dieter Fox. 6-dof graspnet: Variational grasp generation for object manipulation. In *ICCV*, pages 2901–2910. IEEE, 2019.

[46] Adithyavairavan Murali, Weiyu Liu, Kenneth Marino, Sonia Chernova, and Abhinav Gupta. Same object, different grasps: Data and semantic knowledge for task-oriented grasping. In *CoRL*, 2020.

[47] Rhys Newbury, Morris Gu, Lachlan Chumbley, Arsalan Mousavian, Clemens Eppner, Jürgen Leitner, Jeannette Bohg, Antonio Morales, Tamim Asfour, Danica Kragic, et al. Deep learning approaches to grasp synthesis: A review. *arXiv preprint arXiv:2207.02556*, 2022.

[48] Peiyuan Ni, Wenguang Zhang, Xiaoxiao Zhu, and Qixin Cao. Pointnet++ grasping: Learning an end-to-end spatial grasp generation algorithm from sparse point clouds. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3619–3625. IEEE, 2020.

[49] Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger. Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3504–3515, 2020.

[50] Michael Oechsle, Songyou Peng, and Andreas Geiger. UNISURF: unifying neural implicit surfaces and radiance fields for multi-view reconstruction. In *ICCV*, pages 5569–5579. IEEE, 2021.

[51] Jeong Joon Park, Peter Florence, Julian Straub, Richard A. Newcombe, and Steven Lovegrove. Deepsdf:

Learning continuous signed distance functions for shape representation. In *CVPR*, pages 165–174. Computer Vision Foundation / IEEE, 2019.

[52] Songyou Peng, Michael Niemeyer, Lars M. Mescheder, Marc Pollefeys, and Andreas Geiger. Convolutional occupancy networks. In *ECCV*, volume 12348, pages 523–540, 2020.

[53] Robert Platt. Grasp learning: Models, methods, and performance. *Annual Review of Control, Robotics, and Autonomous Systems*, 6(1):363–389, 2023. doi: 10.1146/annurev-control-062122-025215. URL https://doi.org/10.1146/annurev-control-062122-025215.

[54] Charles Ruizhongtai Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *CVPR*, pages 77–85. IEEE Computer Society, 2017.

[55] Zengyi Qin, Kuan Fang, Yuke Zhu, Li Fei-Fei, and Silvio Savarese. Keto: Learning keypoint representations for tool manipulation, 2019.

[56] Satvik Sharma, Adam Rashid, Chung Min Kim, Justin Kerr, Lawrence Yunliang Chen, Angjoo Kanazawa, and Ken Goldberg. Language embedded radiance fields for zero-shot task-oriented grasping. In *7th Annual Conference on Robot Learning*, 2023. URL https://openreview.net/forum?id=k-Fg8JDQmc.

[57] William Shen, Ge Yang, Alan Yu, Jansen Wong, Leslie Pack Kaelbling, and Phillip Isola. Distilled feature fields enable few-shot language-guided manipulation. In *7th Annual Conference on Robot Learning*, 2023.

[58] Martin Sundermeyer, Arsalan Mousavian, Rudolph Triebel, and Dieter Fox. Contact-graspnet: Efficient 6-dof grasp generation in cluttered scenes. In *ICRA*, pages 13438–13444. IEEE, 2021.

[59] Andreas ten Pas, Marcus Gualtieri, Kate Saenko, and Robert Platt Jr. Grasp pose detection in point clouds. *Int. J. Robotics Res.*, 36(13-14):1455–1473, 2017.

[60] Julen Urain, Niklas Funk, Georgia Chalvatzaki, and Jan Peters. Se(3)-diffusionfields: Learning cost functions for joint grasp and motion optimization through diffusion. *arXiv preprint arXiv:2209.03855*, 2022.

[61] Chenxi Wang, Hao-Shu Fang, Minghao Gou, Hongjie Fang, Jin Gao, and Cewu Lu. Graspness discovery in clutters for fast and accurate grasp detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15964–15973, 2021.

[62] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. In *NeurIPS*, pages 27171–27183, 2021.

[63] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E. Sarma, Michael M. Bronstein, and Justin M. Solomon. Dynamic graph CNN for learning on point clouds. *ACM Trans. Graph.*, 38(5):146:1–146:12, 2019.

[64] Zhen Wang, Shijie Zhou, Jeong Joon Park, Despoina Paschalidou, Suya You, Gordon Wetzstein, Leonidas Guibas, and Achuta Kadambi. Alto: Alternating latent topologies for implicit 3d reconstruction. *CVPR*, 2023.

[65] Thomas Weng, David Held, Franziska Meier, and Mustafa Mukadam. Neural grasp distance fields for robot manipulation. *arXiv preprint arXiv:2211.02647*, 2022.

[66] Chaozheng Wu, Jian Chen, Qiaoyu Cao, Jianchi Zhang, Yunxin Tai, Lin Sun, and Kui Jia. Grasp proposal networks: An end-to-end solution for visual learning of robotic grasps. *Advances in Neural Information Processing Systems*, 33:13174–13184, 2020.

[67] Jianglong Ye, Jiashun Wang, Binghao Huang, Yuzhe Qin, and Xiaolong Wang. Learning continuous grasping function with a dexterous hand from human demonstrations. *IEEE Robotics and Automation Letters*, 8(5):2882–2889, 2023. doi: 10.1109/LRA.2023.3261745.

[68] Jianglong Ye, Naiyan Wang, and Xiaolong Wang. Featurenerf: Learning generalizable nerfs by distilling foundation models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8962–8973, 2023.

[69] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelNeRF: Neural radiance fields from one or few images. In *CVPR*, 2021.