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

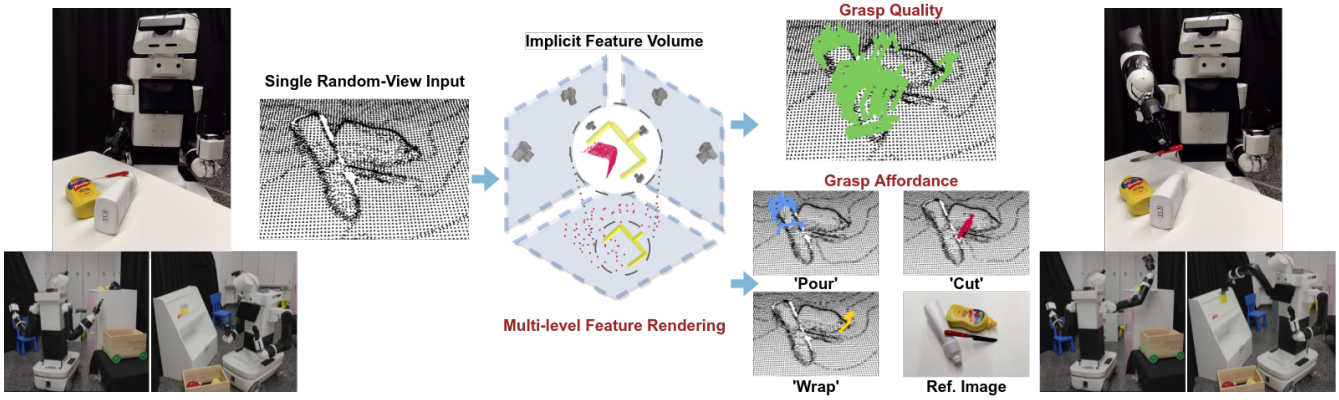

Fig. 1: **Grasping as rendering**. Our network uses just a single random-view depth input, encodes the scene in an implicit feature volume, and uses multi-level rendering to select relevant features and predict grasping functions. We generalize to random-view mobile manipulation grasping scenarios, as shown in the images. Project site: 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 real-world applicability by grasping with a mobile manipulator robot, 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.Deep learning has been crucial in making advances in robotic grasping [1], [2], [3], [4] 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 [5], [6], [7].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.

6DoF grasping methods can be classified into methods that *explicitly generate* grasp poses [8], [9], [10] or *implicitly* classify the grasp quality of *any* grasp candidate in SE(3) [11], [12], [13]. 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 [14] or other affordance-based information [15], [16]. 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 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 [17], [10] or accumulate more information from multiple views [18], [19], [20]. An approach to mitigate partial observability is to use neural scene representations [21], [22], [23] to learn scene completion in a continuous functional space. These

This work was funded by the German Research Foundation (DFG) Emmy Noether Programme (CH 2676/1-1). The authors also acknowledge computing time provided on the high-performance computer Lichtenberg at the NHR Centers NHR4CES at TU Darmstadt, funded by the Federal Ministry of Education and Research, and state governments participating on the basis of GWK resolutions for national high-performance computing at universities (www.nhr-verein.de/unsere-partner).

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 [24], [25], [26], [27]. 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 [28], [29]. 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.

II. Learning 6DoF Grasping via Neural Surface Rendering

In a cluttered scene with objects placed on a surface, we are given a 3D depth pointcloud $\mathbf{x}$ captured from an arbitrary viewing angle. Our robot is equipped with a two-fingered gripper and grasps are 6D gripper poses $\mathbf{g} \in \text{SE(3)}$. Given scene information $\mathbf{x}$, likelihood of grasp success is represented as quality $q \in [0, 1]$. In this setting, our model, NeuGraspNet, learns an implicit function $f_\theta^{sg} : \mathbb{R}^3 \rightarrow \mathbb{R}$ that represents the *scene geometry* and a subsequent implicit function $h_\omega^q : \mathbb{R}^6 \rightarrow \mathbb{R}$ that evaluates the *quality of candidate grasp poses* in the scene (θ and ω being trainable network parameters), leading to a *fully implicit* representation.

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) [22]. 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 $\mathbf{p} \in \mathbb{R}^3$ 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 [26], [24]. 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 (Section II-B) and at a local level for feature extraction per 6D grasp candidate (Section II-C).

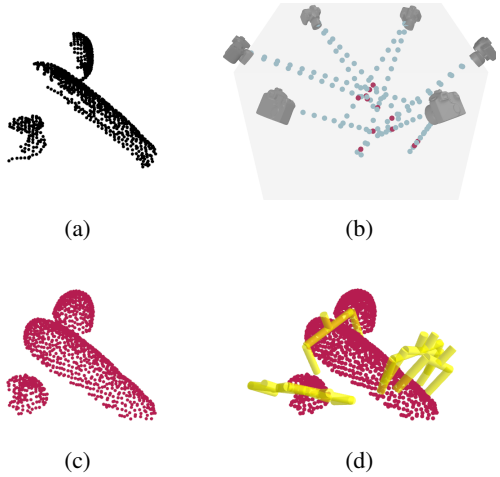

Fig. 2: Scene-level surface rendering: (a) an input single-view pointcloud; (b) surface rendering on the neural implicit geometry (grey volume) using 6 'virtual' cameras; (c) reconstructed surface pointcloud; (d) sampled grasp candidates.

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 2). We perform surface rendering using a root-finding approach similar to [28] and [29]. 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 $\left\{\mathbf{p}_j^r\right\}_{j=1}^n$. 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 [28]). After obtaining the surface point-set from all virtual cameras, we merge and downsample to arrive at a reconstructed scene pointcloud $\{\mathbf{p}_i\}_{i=1}^M$ (Fig. 2c). Our grasp detection is implicit, i.e., we can query the quality of any 6DoF grasp pose $\mathbf{g} \in \text{SE(3)}$. To generate suitable grasp proposals to discriminate upon during training, we use the sampling approach of Grasp Pose Generator (GPG) [30] due to its simple yet effective nature. GPG generates grasp candidates on the input pointcloud using the point surface

[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.

normals and estimating the axis of curvature of a surface. For more details, we refer to [30]. 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 is in Fig. 2d.

C. Local surface rendering & grasp quality prediction

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 3a). We thus obtain a dense local point-set and filter out 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 [24]. 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, trained using simulated ground truth labels $q(\mathbf{g}) \in \{0, 1\}$ of grasp success/failure & 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_ω^q in a subsequent step after the convergence of the scene reconstruction network f_θ^{sg}. 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: $\{\mathbf{p}_{gt\ i}^{\mathbf{g}}\}_{i=1}^N$. We then add noise to these points, as visualized in Fig 3c. These noisy surface points

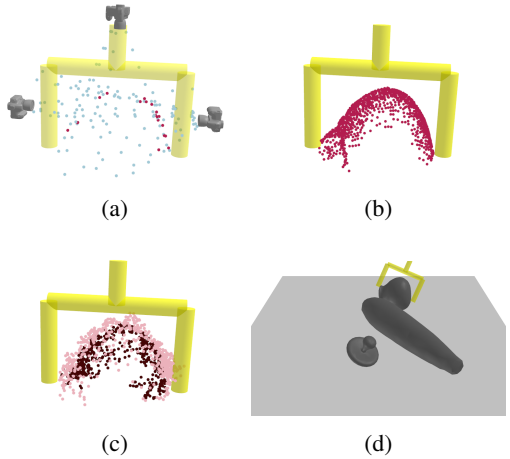

Fig. 3: Local surface rendering: (a) rendering the neural implicit geometry by ray-marching 3 'virtual' cameras at the three parts of the gripper (gripper used only for visualization); (b) the neural rendered surface; (c) noisy ground-truth rendered surface used during training for local occupancy supervision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

$\{\mathbf{p}_{noisy\ i}^{\mathbf{g}}\}_{i=1}^N$ 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. 3). We use a loss function $\mathcal{L}_{local}$, similar to Equation (1), to additionally train *local object-part reconstruction* from these noisy surface points.

D. Implementation

We use 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_θ^{sg} is a ResNet-based network (as in [22]). For grasp quality prediction, the decoder h_ω^q uses a point network [31].
The overall loss, weighted by factors w_i, is

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

III. Experiments

Experimental setup. For training and evaluating NeuGraspNet, we first use the popular simulation benchmark from VGN [18], also used in [24], [13], [32]. In this setup, random objects are spawned in a '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. A modification we do to make the task more realistic and challenging is to *randomize* the camera viewing angle when generating the scene TSDF.

TABLE I: Comparative results of NeuGraspNet vs. baselines on Pile and Packed scenes [18] (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 [12]	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
VGN [18]	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 [13]	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 [24]	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 [18] (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

Data generation and training. Since our network is implicit in all 6 grasp dimensions (unlike semi-implicit methods [18], [24]), 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 II-B 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.

Baselines & metrics. 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** [12], which operates directly on pointclouds, and the current state-of-the-art **EdgeGraspNet** [13] that proposes a contact-edge-based grasp representation. Note that for EdgeGraspNet, we generate the dataset as per the sampling strategy in [13] as it requires grasp samples on edges. Additionally, we compare against the semi-implicit methods **VGN** [18] and **GIGA** [24], with the latter considering scene reconstruction as an auxiliary loss.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) inputs for a fair comparison.

We use the same metrics as [24], [18], [13], 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.

1) Comparison with baselines on VGN [18] 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 consecutive 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 fixed top view settings since it uses the GPG [30] grasp sampler, which favors top grasps, but performs poorly in hard view settings.

2) Ablation study: Table II presents an ablation of different components of NeuGraspNet. 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 the global scene-level rendering, we see a significant drop in 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') and not using local occupancy supervision, also drops performance.

A. Real-world Evaluation

We perform real-world experiments with a mobile manipulator robot, TIAGo++, equipped with a head-mounted RGBD camera. In house-like scenarios with YCB objects [33], 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. Video demonstrations are provided at https://sites.google.com/view/neugraspnet.

IV. Conclusion

We presented NeuGraspNet, a novel, fully implicit 6DoF-grasp prediction method that re-interprets robotic grasping as surface rendering. 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). Finally, we exhibited the real-world applicability of NeuGraspNet in mobile manipulator grasping experiments.

REFERENCES

[1] J. Mahler, J. Liang, S. Niyaz, M. Laskey, R. Doan, X. Liu, J. A. Ojea, and K. Goldberg, "Dex-net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics," *arXiv preprint arXiv:1703.09312*, 2017.

[2] J. Mahler, M. Matl, X. Liu, A. Li, D. Gealy, and K. 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)*. IEEE, 2018, pp. 5620–5627.

[3] 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.

[4] G. Chalvatzaki, N. Gkanatsios, P. Maragos, and J. Peters, "Orientation attentive robotic grasp synthesis with augmented grasp map representation," *arXiv preprint arXiv:2006.05123*, 2020.

[5] O. Kroemer, S. Niekum, and G. Konidaris, "A review of robot learning for manipulation: Challenges, representations, and algorithms," *The Journal of Machine Learning Research*, vol. 22, no. 1, pp. 1395–1476, 2021.

[6] R. Newbury, M. Gu, L. Chumbley, A. Mousavian, C. Eppner, J. Leitner, J. Bohg, A. Morales, T. Asfour, D. Kragic *et al.*, "Deep learning approaches to grasp synthesis: A review," *arXiv preprint arXiv:2207.02556*, 2022.

[7] R. Platt, "Grasp learning: Models, methods, and performance," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 6, no. 1, pp. 363–389, 2023. [Online]. Available: https://doi.org/10.1146/annurev-control-062122-025215

[8] A. Mousavian, C. Eppner, and D. Fox, "6-dof graspnet: Variational grasp generation for object manipulation," in *ICCV*. IEEE, 2019, pp. 2901–2910.

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

[10] M. Sundermeyer, A. Mousavian, R. Triebel, and D. Fox, "Contact-graspnet: Efficient 6-dof grasp generation in cluttered scenes," in *ICRA*. IEEE, 2021, pp. 13 438–13 444.

[11] A. ten Pas, M. Gualtieri, K. Saenko, and R. P. Jr., "Grasp pose detection in point clouds," *Int. J. Robotics Res.*, vol. 36, no. 13-14, pp. 1455–1473, 2017.

[12] 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 *ICRA*. IEEE, 2019, pp. 3629–3635.

[13] H. Huang, D. Wang, X. Zhu, R. Walters, and R. Platt, "Edge grasp network: A graph-based se(3)-invariant approach to grasp detection," *ICRA*, 2023.

[14] J. Ye, J. Wang, B. Huang, Y. Qin, and X. Wang, "Learning continuous grasping function with a dexterous hand from human demonstrations," *IEEE Robotics and Automation Letters*, vol. 8, no. 5, pp. 2882–2889, 2023.

[15] S. Christen, M. Kocabas, E. Aksan, J. Hwangbo, J. Song, and O. 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)*, June 2022, pp. 20 577–20 586.

[16] H. Li, X. Lin, Y. Zhou, X. Li, Y. Huo, J. Chen, and Q. Ye, "Contact2grasp: 3d grasp synthesis via hand-object contact constraint," 2023.

[17] P. Ni, W. Zhang, X. Zhu, and Q. 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)*. IEEE, 2020, pp. 3619–3625.

[18] M. Breyer, J. J. Chung, L. Ott, R. Siegwart, and J. I. Nieto, "Volumetric grasping network: Real-time 6 DOF grasp detection in clutter," in *CoRL*, ser. Proceedings of Machine Learning Research, vol. 155. PMLR, 2020, pp. 1602–1611.

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

[20] H.-S. Fang, C. Wang, H. Fang, M. Gou, J. Liu, H. Yan, W. Liu, Y. Xie, and C. Lu, "Anygrasp: Robust and efficient grasp perception in spatial and temporal domains," *arXiv preprint arXiv:2212.08333*, 2022.

[21] L. M. Mescheder, M. Oechsle, M. Niemeyer, S. Nowozin, and A. Geiger, "Occupancy networks: Learning 3d reconstruction in function space," in *CVPR*. Computer Vision Foundation / IEEE, 2019, pp. 4460–4470.

[22] S. Peng, M. Niemeyer, L. M. Mescheder, M. Pollefeys, and A. Geiger, "Convolutional occupancy networks," in *ECCV*, vol. 12348, 2020, pp. 523–540.

[23] J. J. Park, P. Florence, J. Straub, R. A. Newcombe, and S. Lovegrove, "Deepsdf: Learning continuous signed distance functions for shape representation," in *CVPR*. Computer Vision Foundation / IEEE, 2019, pp. 165–174.

[24] Z. Jiang, Y. Zhu, M. Svetlik, K. Fang, and Y. Zhu, "Synergies between affordance and geometry: 6-dof grasp detection via implicit representations," in *Robotics: Science and Systems*, 2021.

[25] Q. Dai, Y. Zhu, Y. Geng, C. Ruan, J. Zhang, and H. Wang, "Graspnerf: Multiview-based 6-dof grasp detection for transparent and specular objects using generalizable nerf," *arXiv preprint arXiv:2210.06575*, 2022.

[26] J. Lundell, F. Verdoja, and V. Kyrki, "Ddgc: Generative deep dexterous grasping in clutter," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 6899–6906, 2021.

[27] S. Sharma, A. Rashid, C. M. Kim, J. Kerr, L. Y. Chen, A. Kanazawa, and K. Goldberg, "Language embedded radiance fields for zero-shot task-oriented grasping," in *7th Annual Conference on Robot Learning*, 2023. [Online]. Available: https://openreview.net/forum?id=k-Fg8JDQmc

[28] M. Niemeyer, L. Mescheder, M. Oechsle, and A. Geiger, "Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 3504–3515.

[29] M. Oechsle, S. Peng, and A. Geiger, "UNISURF: unifying neural implicit surfaces and radiance fields for multi-view reconstruction," in *ICCV*. IEEE, 2021, pp. 5569–5579.

[30] M. Gualtieri, A. ten Pas, K. Saenko, and R. P. Jr., "High precision grasp pose detection in dense clutter," in *IROS*. IEEE, 2016, pp. 598–605.

[31] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *CVPR*. IEEE Computer Society, 2017, pp. 77–85.

[32] E. Chisari, N. Heppert, T. Welschehold, W. Burgard, and A. Valada, "Centergrasp: Object-aware implicit representation learning for simultaneous shape reconstruction and 6-dof grasp estimation," *arXiv preprint arXiv:2312.08240*, 2023.

[33] B. Calli, A. Walsman, A. Singh, S. Srinivasa, P. Abbeel, and A. M. Dollar, "Benchmarking in manipulation research: The ycb object and model set and benchmarking protocols," *arXiv preprint arXiv:1502.03143*, 2015.