

# GraspNeRF: Multiview-based 6-DoF Grasp Detection for Transparent and Specular Objects Using Generalizable NeRF

Qiyu Dai<sup>1,2\*</sup>, Yan Zhu<sup>1\*</sup>, Yiran Geng<sup>1</sup>, Ciyu Ruan<sup>3</sup>, Jiazhao Zhang<sup>1,2</sup>, He Wang<sup>1,2†</sup>

**Abstract**—In this work, we tackle 6-DoF grasp detection for transparent and specular objects, which is an important yet challenging problem in vision-based robotic systems, due to the failure of depth cameras in sensing their geometry. We, for the first time, propose a multiview RGB-based 6-DoF grasp detection network, GraspNeRF, that leverages the generalizable neural radiance field (NeRF) to achieve material-agnostic object grasping in clutter. Compared to the existing NeRF-based 3-DoF grasp detection methods that rely on densely captured input images and time-consuming per-scene optimization, our system can perform zero-shot NeRF construction with sparse RGB inputs and reliably detect 6-DoF grasps, both in real-time. The proposed framework jointly learns generalizable NeRF and grasp detection in an end-to-end manner, optimizing the scene representation construction for the grasping. For training data, we generate a large-scale photorealistic domain-randomized synthetic dataset of grasping in cluttered tabletop scenes that enables direct transfer to the real world. Our extensive experiments in synthetic and real-world environments demonstrate that our method significantly outperforms all the baselines in all the experiments while remaining in real-time. Project page can be found at <https://pku-epic.github.io/GraspNeRF>.

## I. INTRODUCTION

Object grasping is a crucial technology for many robotic systems. Recent years have witnessed great progress in developing vision-based grasping methods [1], [2], [3], [4], [5], [6], [7]. These methods rely on single-view or multiview depth inputs and can reliably detect 6-DoF grasps from cluttered table-top scenes mainly composed of objects with diffuse material. Despite the great progress in diffuse objects, grasping transparent and specular objects, which are ubiquitous in our daily life and need to be handled by robotic systems, is still very challenging. Depth sensors struggle to sense those transparent and specular objects and usually generate wrong or even missing depths, thus further leading to the failure of grasping in those methods.

For grasping transparent and specular objects, recent works have been devoted to depth restoration and RGB-based grasping methods. Works, *e.g.*, ClearGrasp [8], TransCG [7], SwinDRNet [9], leverage RGBD inputs and learn to recover the missing depths and correct the wrong depths as a separate step prior to performing grasping. DexNeRF [10] provides a valuable alternative that only takes RGB inputs. It proposes to first construct a NeRF [11], the powerful implicit 3D representation that encodes scene geometry using an MLP,

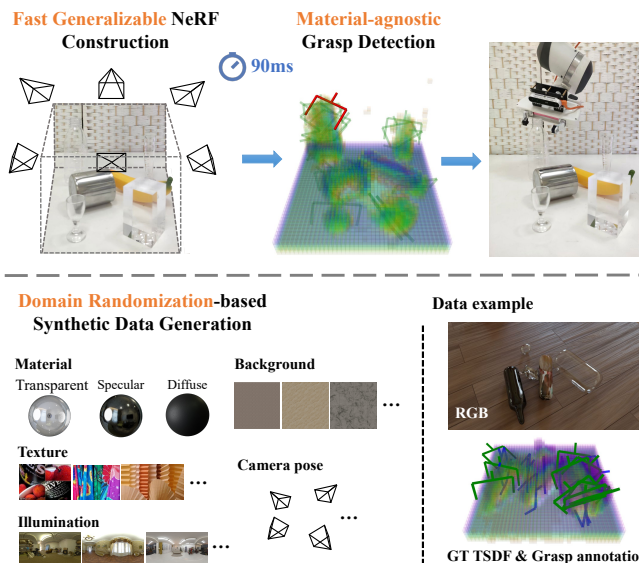


Fig. 1. Overview of the proposed GraspNeRF and the dataset. Our method takes sparse multiview RGB images as input, constructs a neural radiance field, and executes material-agnostic grasp detection within 90ms. We train the model on the proposed large-scale synthetic multiview grasping dataset generated by photorealistic rendering and domain randomization.

by capturing 49 RGB images from different viewpoints and then use volumetric rendering to render a top-view depth map that further passes to 3-DoF grasping module. Although being very novel, this method suffers from several important limitations. Its NeRF construction needs time-consuming per-scene training that takes at least hours, which deviates far from the desired real-time running. This gets even worse for multi-object sequential grasping, since the scenes can change after each grasping and thus the NeRF needs to be retrained. A concurrent work, EvoNeRF [12] proposes to use Instant-NGP [13] to speed up NeRF training and evolve the NeRF after each grasp. However, it still costs 7s for each update, relies on dense image inputs, and can only perform 3-DoF grasping from top-down views. Another big disadvantage of using vanilla NeRF is that the scene geometry extracted from NeRF is usually imperfect and can be very low-quality for scenes composed of transparent objects and textureless backgrounds, given that NeRF is only overfitted to their RGB colors. The wrong geometry can severely degrade the performance of the downstream pretrained grasping algorithms.

To mitigate these issues, we propose to leverage generalizable NeRF, *e.g.*, MVSNeRF [14], NeuRay [15], which train on many different scenes and learn to aggregate multiview observations and zero-shot construct NeRFs for novel scenes

<sup>1</sup>Center on Frontiers of Computing Studies, Peking University.

<sup>2</sup>Beijing Academy of Artificial Intelligence.

<sup>3</sup>College of Intelligence Science and Technology, National University of Defense Technology.

\*The first two authors contributed equally.

†Corresponding to hewang@pku.edu.cn.

without training. Moreover, it only requires sparse input views, which is more practical for real-world applications. Our framework first constructs a 3D grid-like workspace and obtains the grid features via aggregating multi-view image features. On these grid points, we then propose to predict truncated signed distance function (TSDF) to obtain a complete 3D geometric scene representation, which frees us from computing depths. Taking input this TSDF, our volumetric grasp detection network can then predict 6-DoF grasps at each grid point. Note that this TSDF can also be turned into density and used for volume rendering.

To train this learning-based framework, we generate a large-scale photorealistic synthetic dataset of grasping diverse objects with diffuse, transparent, or specular material in cluttered table-top scenes. This dataset is composed of 2.4 million images with 100K scenes, rendered by physics-based rendering and enhanced by diverse domain randomization. Trained on this extensive dataset, our network can directly generalize to novel real-world scenes with novel objects.

Our experiments further show that the proposed method outperforms all baselines that leverage either depth restoration or separate NeRF construction, by over 20% of grasp success rate, with 82.2% and 65.9% in packed and pile scenes, respectively on sequential transparent and specular object grasping in the real world. We also find an interesting phenomenon: although our grasping only cares about TSDF, enforcing NeRF rendering loss during training can bring performance improvements to grasping, especially on transparent and specular objects, indicating the synergies between view-dependent 2D rendering and grasping in 3D space.

To summarize, this proposed grasping algorithm has many remarkable advantages over the previous state-of-the-art works: it only takes sparse RGB inputs (we use 6) without depths; it is material-agnostic, generalizable, and thus directly works on novel scenes; it is end-to-end differentiable and directly optimized for grasping; it can perform 6-DoF grasping; and, finally, it runs at real-time with 11 FPS.

## II. RELATED WORK

### A. Transparent and Specular Object Grasping

Grasping transparent and specular objects is quite a challenging problem. Most depth-based methods fail due to incomplete depth maps. A direct way is to restore depths before detecting grasp. ClearGrasp[8] firstly estimates geometry information from a single RGB image and then refines depths via global optimization, which is used for 3-DoF transparent object grasping from top-down views. [7] proposes a CNN-based depth completion network, and [9] proposes a Swin Transformer-based RGBD fusion network, SwinDRNet, for depth restoration and the downstream 6-DoF grasping. Other studies focus on predicting grasp poses from RGB and depth. [16] learns an RGBD grasp detection model that is transferred from an existing depth-based model. [17] predicts grasp poses from the RGBD-based CNN and executes grasping using the proposed soft robotic hand. Those single-view-based grasping methods are difficult to handle object occlusions in cluttered scenes and cannot fully

utilize the 3D geometry information. Our method falls into the category which takes as input the multiview RGB images. GlassLoc [18] constructs the Depth Likelihood Volume (DLV) descriptor from the multiview light field observations, representing the transparent object clutter scenes. Ghost-Pose [19] conducts transparent object grasping by estimating the object 6D poses from the proposed model-free pose estimation approach based on multiview geometry. Departing from those methods, the proposed GraspNeRF represents the scene geometry as the generalizable NeRF, which is differentiable so that we are able to improve the grasping performance by jointly training with grasping and exploiting the synergies between both.

### B. Implicit Representations for Grasping

Recent works show a trend in leveraging implicit scene representation for robotic tasks. GIGA [3] proposes to leverage the synergies between geometry reconstruction and grasping prediction, which utilizes the convolutional occupancy network to construct the implicit scene representation. However, it takes a single-view depth image as input, which is hard to reconstruct transparent and specular objects. [20] renders depth maps from NeRF to provide dense correspondence for object descriptors applied to the manipulation task, which requires per-scene optimization and dense input. DexNeRF [10] is the first to utilize NeRF for grasping transparent objects, which avoids directly using the corrupted sensor depths by the proposed transparency-aware depth rendering. However, it is at the cost of capturing dense input images and hours of NeRF training for each grasp, which is impractical in real-world grasping. Furthermore, it only works on 3-DoF grasping, limiting its performance and applications. The concurrent work, EvoNeRF [12], speeds up the vanilla NeRF optimization using Instant-NGP [13], and MIRA [21], also utilizes Instant-NGP to synthesize a novel orthographic view, which is used to predict pixel-wise affordances for 6-DoF control. However, they still suffer from dense inputs and require training before each grasp, while the proposed GraspNeRF leverages generalizable NeRF that is free from per-scene optimization and only needs sparse input, which is suitable for real-world grasping tasks.

## III. METHOD

### A. Problem Statement and Method Overview

Given sparse multiview RGB observations of a cluttered tabletop scene composed of transparent, specular, or diffuse objects, the goal of the proposed robotic system is to detect the 6-DoF grasps and then execute grasp-to-remove operations for all the objects. Here we assume the camera intrinsics and extrinsics are known for the multiview images.

We then formulate the 6-DoF grasp detection as a learning problem that maps a set of images from  $N$  input views  $\{I_i\}_{i=1,\dots,N}$  to a set of 6-DoF grasps  $\{g_j | g_j = (t_j, R_j, w_j, q_j)\}$ , where, for each detected grasp  $g_j$ ,  $t_j \in \mathbb{R}^3$  is 3D position,  $R_j \in \text{SO}(3)$  is 3D rotation,  $w_j \in \mathbb{R}$  is opening width, and  $q_j \in \{0, 1\}$  is the quality.

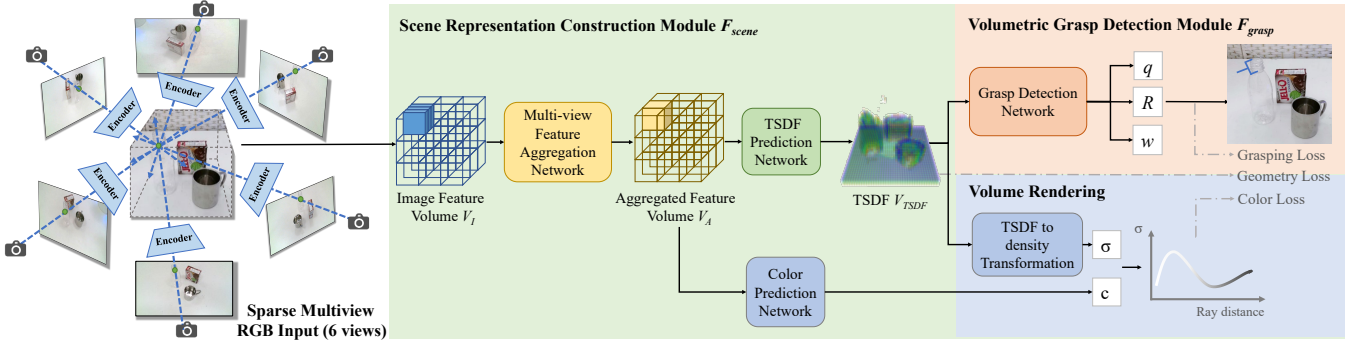


Fig. 2. The framework of our proposed method. The features of input sparse multiview color images are first extracted to construct the image feature volume  $V_I$  by querying the 2D image features corresponding to the 3D sample points. Then the implicit scene representation module maps  $V_I$  to the predicted TSDF  $V_{TSDF}$ , pixel color  $c$ , and volume density  $\sigma$ . Volumetric grasp detection module  $V_{TSDF}$  detects grasps with  $V_{TSDF}$  as input.

Our proposed framework is composed of a scene representation construction module  $F_{scene}$  and a volumetric grasp detection module  $F_{grasp}$ , as shown in Figure 2. In  $F_{scene}$ , we first extract and aggregate the multiview image features for two purposes: the extracted geometric features form a feature grid and will be passed to our TSDF prediction network to construct the TSDF of the scene, which encodes the scene geometry as well as the density of the underlying NeRF; at the same time, the features are used to predict color, which along with the density outputs enables the NeRF rendering. Taking the predicted TSDF as input, our volumetric grasp detection module  $F_{grasp}$  then learns to predict 6-DoF grasps at each voxel of the TSDF.

### B. Scene Representation Construction

For our grasping framework, one crucial demand is to wholly and accurately represent the scene, especially for transparent and specular objects, in a generalizable and grasping-oriented way. A vanilla NeRF needs scene-specific training and learns an MLP to overfit one scene without having an explicit representation. We instead propose to leverage generalizable NeRF that learns to aggregate observations from multiple views and forms a volumetric feature grid.

**Revisiting NeuRay.** The generalizable NeRF used in our scene representation construction module is inspired by NeuRay [15], an occlusion-aware generalizable NeRF. Taking as input a set of images  $\{I_i\}_{i=1,\dots,N}$ , NeuRay first utilizes a shared CNN image encoder, which is trained from scratch to extract the 2D image feature maps of  $C$  channels. To render a pixel from a novel view, it samples  $K$  points along the pixel ray, projects them to each input view, and then queries image features, which forms  $N$  ray features  $v_{I,i} \in \mathbb{R}^{K \times C}$ . To aggregate the feature from different views considering occlusion, NeuRay predicts a visibility weight  $w_i$  indicating whether a 3D point is visible in each input view. With this weight, the ray features  $\{v_{I,i}\}$  is aggregated to  $v_A = \sum_i w_i v_{I,i}$ . Then it passes  $v_A$  to a multi-head attention network for predicting the volume density of each sample point. Volume colors are also predicted via blending pixel colors from input views.

Note that the features in NeuRay are ray-based and do not form a volumetric representation of the whole scene, thus adopting NeuRay for grasping is non-trivial.

**Volumetric Feature Extraction.** To obtain grasping-oriented volumetric features, we first divide the 3D workspace into grids of size  $S$ . Each grid point  $x \in \mathbb{R}^{S^3}$  is projected to all of the  $N$  views ( $N=6$ ) and queries the corresponding feature, similar to NeuRay. This builds an image feature volume  $V_I \in \mathbb{R}^{S^3 \times N \times C}$ . Then, the image feature volume is aggregated among views using a multi-view feature aggregation network composed of MLPs, forming the aggregated feature volume  $V_A \in \mathbb{R}^{S^3 \times C}$ . Now, for rendering, we could do the same to turn  $V_A$  into color and density on the grid points. However, we argue that directly passing either  $V_A$  or volume density to grasp detection network can be suboptimal, due to a lack of accurate geometric information.

**Truncated Signed Distance Field Prediction.** To obtain a scene representation that encodes accurate geometric information, we are inspired by NeuS [22], which predicts SDF instead of volume density and link these two together. For grasp detection, we propose to predict TSDF from  $V_A$ . Specifically, we propose a TSDF prediction network that maps  $V_A$  to the truncated signed distance field  $V_{TSDF} \in \mathbb{R}^{S^3}$ . Inspired by NeuRay, for predicting the TSDF value at grid point  $(u, v, w)$ , we aggregate features from all  $S$  grid points that share the same  $u$  and  $v$  using a multi-head attention network and the predicted TSDF value can be supervised by a geometry loss (see Section III-E).

**Volume Rendering.** For volume rendering, we propose that this TSDF value can be tightly coupled with volume density  $\sigma$  as shown below:

$$\sigma = \max\left(\frac{-\frac{dS}{dx}(F_{TSDF}(x))}{S(F_{TSDF}(x))}, 0\right), \quad (1)$$

where  $S(\cdot)$  is the sigmoid function. In this way, since both grasp detection and volume rendering can leverage TSDF for their tasks, this enables the synergy between the two tasks. Note that the whole network can be learned in an end-to-end manner and directly optimized for grasping, whereas DexNeRF [10] needs staged training of NeRF and grasping.

### C. Volumetric Grasping Detection

The 3D CNN grasp detector  $F_{grasp}$  learns a mapping from the TSDF grid to the dense grasp candidate volume, inspired by [2]. We consider the center of each voxel within the

volume as the grasp position candidate. At given position  $t \in \mathbb{R}^3$ , the detector predicts three outputs from the separate decoder heads, including the grasping quality  $q$ , rotation  $R$ , and opening width  $w$ .

To select a proper grasp for execution, we discard the grasping position candidates far from the surface according to the predicted TSDF. Subsequently, we smooth the quality volume using Gaussian, filter out the grasp below the threshold and utilize the non-maximum suppression. We also choose the grasps that meet the gripper max opening width. Finally, we randomly select a grasp for robot execution from the remaining candidates to prevent repeated execution of an unsuccessful grasp due to a false positive.

#### D. Domain Randomization-based Synthetic Data Generation

Existing multiview grasping datasets like [1] do not consider transparent and specular objects, while the current multiview object datasets lack grasping annotations. In this work, we present a large-scale synthetic multiview RGB-based grasping dataset, which contains 100K scenes with objects ranging from diffuse, specular to transparent, 2.4 million RGB images, and 2 million 6-DoF grasp poses.

Our data generation pipeline is motivated by [9]. To construct the scenes, we transform an existing grasping dataset [3] that contains CAD models and annotates object poses, by changing their object materials to diffuse, specular or transparent. Then we render photorealistic multiview RGBs on Blender [23]. To bridge the sim2real gap, we leverage domain randomization, which randomizes object materials and textures, backgrounds, illuminations, and camera poses. After training on the synthetic dataset with sufficient variations, the network considers real data as a variation of training data in testing time, so as to generalize to real.

#### E. Network End-to-End Training

We adopt end-to-end training for jointly learning scene representation and grasping, encouraging the two tasks to benefit each other. For grasping, it is crucial to perceive and reconstruct the scene geometry; and accurate grasping predictions, in turn, require high-quality scene representations. The training objective consists of three parts.

**Grasping Loss.** For grasping learning, we supervise the grasp quality, orientation, and opening width as done in [2], which is formulated as:

$$L_{grasp} = L_q(\hat{q}, q) + q(L_w(\hat{w}, w) + L_o(\hat{R}, R)), \quad (2)$$

where  $L_q$  represents the binary cross-entropy loss between the predicted grasp quality label  $\hat{q} \in [0, 1]$  and the ground truth  $q \in \{0, 1\}$ .  $L_w$  is the  $L_2$  loss penalizing width error.  $L_o = 1 - |\hat{R} \cdot R|$  measures the quaternion distance. Note that because of the gripper’s symmetry, a grasp rotated by  $180^\circ$  around its wrist axis is deemed identical. Therefore, the orientation loss is determined by the closest distance from these two grasps to the ground truth. We also exclude the zero-quality grasp, encouraging the network to learn grasps with high confidence.

**Color Loss.** To learn a generalizable radiance field, we randomly take 6 training views as input for each training scene and randomly pick another targeted view for rendering. We supervise the novel view synthesis using  $L_2$  loss:

$$L_{color} = \sum_k \|\hat{C}_k - C_k\|_2, \quad (3)$$

where  $\hat{C}_k$  and  $C_k$  denote the predicted and target color at pixel  $k$  respectively.

**Geometry Loss.** Since the ground-truth TSDF values of each scene can be easily obtained from our synthetic data generation pipeline, we supervise the explicit geometry using  $L_1$  loss along with the Eikonal regularization term [24]:

$$L_{geo} = \frac{1}{\|\chi\|} \sum_{p_i \in \chi} (\|\hat{V}_{TSDF}(p_i) - V_{TSDF}(p_i)\|_1 + (\|\nabla \hat{V}_{TSDF}(p_i)\|_2 - 1)^2), \quad (4)$$

where  $\hat{V}_{TSDF}(p_i)$  and  $V_{TSDF}(p_i)$  is the predicted and ground truth TSDF values respectively.  $p_i \in \chi$  denotes the 3D coordinate in the world frame.

The whole objective can be formulated as:

$$L = L_{grasp} + L_{color} + L_{geo}. \quad (5)$$

## IV. EXPERIMENTS

In this section, we evaluate the performance of our proposed method for grasping tasks through both simulation and real robot experiments. We also perform ablation studies on simulation data to analyze the impact of the main designs incorporated in our framework.

### A. Implementation Details

The size of the TSDF volume is  $40 \times 40 \times 40$ . We train our network for 300k steps, utilizing the Adam optimizer with a learning rate of  $1e-4$  and a weight decay of 0.01. We randomly sample 2048 rays as a batch during training. The methods are evaluated on an NVIDIA RTX 3090 GPU.

### B. Experiment Setup

**Grasping Environment Setup.** The Franka Emika Panda robot arm equipped with a parallel-jaw gripper performs grasping within a  $30 \times 30 \times 30 \text{ cm}^3$  tabletop workspace. A RealSense D415 RGBD camera (depth is not used by our method) is attached to the gripper’s wrist. To ensure adequate observation of objects, we uniformly sample 6 camera viewpoints on a hemisphere with a shared center in the workspace. All camera views point to the workspace center, represented in spherical coordinates as radius  $r = 0.5m$ , polar angle  $\theta = \frac{\pi}{6}$ , and azimuthal angle  $\varphi \in U(0, 2\pi)$ .

**Simulation Environment Setup.** We build our simulation environment on PyBullet [25] and Blender [23] for physically grasping simulation and multiview photorealistic image rendering, respectively. Note that synthetic depths obtained from the z-buffer cannot reflect the real depth missing and noise caused by specular and transparent materials. Thus we utilize the depth sensor simulator [9] to generate simulated depths with realistic sensor noise to serve as the baseline input.

**Object Set.** A total of 473 hand-scaled object meshes from [2] are used in simulation, with 417 for training, and 56 for testing. We randomize their textures and materials including transparent, specular, and diffuse. We also collect 20 household objects with various materials for real-world evaluations. In each scene, the objects are arranged as a *pile* with random poses or *packed* in an upright position.

### C. Baseline Methods

We compare our method with the following baselines:

- **VGN** [2]. A volumetric grasping detection network that takes the TSDF volume integrated from the multi-frame depths as input, and predicts the 6-DoF grasps.
- **SwinDR-VGN**. A baseline that utilizes the SOTA depth restoration network, SwinDRNet [9], to reduce depth errors from transparent and specular materials. It is designed to refine the TSDFs to benefit the VGN.
- **NeRF-VGN**. A baseline that uses the SOTA NeRF method, Instant-NGP [13], to learn the implicit scene representation and render depths from 49 camera viewpoints for TSDF integration. VGN then receives the TSDF and detects 6-DoF grasps. To make the approach perform appropriately, we give this baseline extensive viewpoints (49) compared with other baselines.

**Evaluation Metrics.** We measure the performance by 1) **Success Rate (SR)**: the ratio of the successful grasp number and the attempt number, and 2) **Declutter Rate (DR)**: the average percentage of removed objects in all rounds.

### D. Simulation Grasping Experiments

**Experimental Protocol.** In this subsection, we utilize extensive simulation data to evaluate all the methods. Our experiments include: 1) **Single object retrieval**: one object is placed at the center of the workspace. The robot arm performs grasps predicted by baselines, and a successful grasp indicates that the object is grasped out of the workspace. We evaluate with 12 test objects, 4 each of transparent, specular, or diffuse materials. Each object is tested three times, following the same protocol as EvoNeRF [12], and the grasping success rate is reported. 2) **Sequential clutter removal**: we conduct 200 round table clearing experiments on both pile and packed scenes, each containing 5 objects. Each round shares the same objects and object poses, with two material combinations: A) transparent and specular, and B) a mix of all diffuse, transparent, and specular materials. In each trial of one round, the robot arm grasps and removes one object until the workspace is cleared or grasp detection fails or two consecutive failures are reached. Results are reported for both material combinations.

**Results and Analysis.** For single object retrieval, as presented in table I, our method achieves an average success rate of 86.1% over 36 trials, outperforming all the baselines. Furthermore, our method exhibits superior performance for sequential clutter removal, as shown in Table II, where we evaluate the methods on various objects with transparent and specular as well as mixed materials. Specifically, our method

significantly improves the performance of transparent and specular object grasping compared to other methods.

For TSDF-based methods, the captured multiview depths contain sensor noises or missing values, especially on specular or transparent surfaces. This presents a challenge for maintaining multiview geometry consistency required by the TSDF, resulting in lower geometry quality that impacts the performance of VGN. SwinDR-VGN restores the depths before TSDF integration, thereby enhancing the TSDF quality to some extent. However, it suffers from over-smooth surfaces, causing objects to appear larger and potentially erasing smaller-sized objects in the TSDF, thus degrading the grasping performance.

For multiview RGB-based methods, NeRF-VGN requires 49 views and a long optimization time to overfit one scene, making it unsuitable for sequential clutter removal. Additionally, it exhibits a low grasping success rate in single object retrieval. We also observe that NeRF-VGN fails to perform adequately in textureless backgrounds, as illustrated in Figure 3, because vanilla NeRFs struggle to learn the geometry consistency of corresponding points in multiple views from poorly textured surfaces.

By comparison, our method, trained on a domain randomization-enhanced dataset that includes randomized backgrounds, learns a background prior during training, enabling it to generalize to both textureless and textured tabletop during inference. Figure 3 demonstrates our superiority over competing methods for geometry reconstruction.

TABLE I  
SUCCESS RATES (%) OF SINGLE OBJECT RETRIEVAL IN SIMULATION

	Transparent	Specular	Diffuse	Overall
VGN	25.0	41.7	91.7	52.8
SwinDR-VGN	58.3	66.7	91.7	72.2
NeRF-VGN	25.0	41.7	50.0	38.9
Ours	<b>75.0</b>	<b>91.7</b>	<b>91.7</b>	<b>86.1</b>

TABLE II  
RESULTS OF SEQUENTIAL CLUTTER REMOVAL IN SIMULATION  
EACH CELL: TRANSPARENT&SPECULAR (MIXED)

	Pile		Packed	
	SR (%)	DR (%)	SR (%)	DR (%)
VGN	40.6(48.9)	26.0(33.7)	56.0(61.8)	57.3(64.0)
SwinDR-VGN	57.9(61.7)	40.6(42.1)	43.8(54.8)	36.0(49.2)
Ours	<b>67.3(67.0)</b>	<b>46.2(45.8)</b>	<b>82.3(83.8)</b>	<b>77.7(74.3)</b>

### E. Real Robot Experiments

To evaluate the real-world performance of baselines, we further conduct real robot experiments.

**Experimental Protocol.** We follow the protocol of simulation experiments. For single object retrieval, we select 6 objects, including 2 each of transparent, specular, and diffuse, and place them in the same location with the same pose. The grasping iteration is repeated three times. For sequential clutter removal, we conduct 15 rounds of iteration on both pile and packed scenes, reporting results on transparent and specular object scenes, as well as mixed-material scenes. Each scene includes 4-6 randomly selected

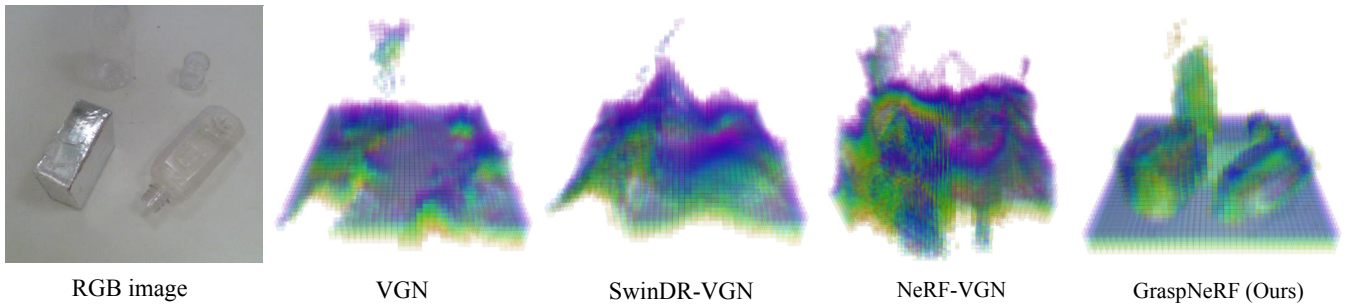


Fig. 3. Comparison of TSDFs constructed by different methods. The geometry from VGN, directly fused from raw depth observations, is incomplete. Even restored by SwinDRNet, the geometry is still over-smooth and hard for grasp detection. NeRF-VGN also fails due to its inaccurate rendered depths under texture-less background. Only our method yields a good TSDF for specular and transparent objects, thanks to learning on diverse data.

objects, and we capture RGB images from 6 views, as in the simulation environment. The network receives the multiview input and outputs 6-DoF grasp candidates, where the robot arm executes the top grasp. We consider a grasp successful if the object is removed from the workspace. The round ends when the grasp fails twice or all objects are cleared.

**Results and Analysis.** Table III reports the success rates of real-world single object grasping, where GraspNeRF obtains 88.9% of 18 trials and beats competing baselines. For sequential clutter removal, GraspNeRF achieves the highest grasping success rate and declutter rate in various materials in both pile and packed settings, as shown in Table IV. Compared to NeRF-VGN (49 views), GraspNeRF only requires 6 views. Moreover, our method runs inference at 11 FPS without any optimization, which is significantly faster than NeRF-VGN with per-scene optimization. Importantly, our method successfully generalizes to new categories, as evidenced by the results for the two transparent objects (Glass gourd and small wine glass). These results demonstrate the effectiveness and robustness of our method in the real world.

Nonetheless, we do note some failure cases. For example, when grasping upright cylindrical objects in packed scenes, such as a hard glass bottle, the gripper might slip off the bottle due to insufficient friction.

TABLE III

SUCCESS RATES (%) OF REAL-WORLD SINGLE OBJECT RETRIEVAL

	Transparent	Specular	Diffuse	Overall
VGN	33.3	66.6	100.0	66.6
SwinDR-VGN	50.0	50.0	83.3	61.1
NeRF-VGN	33.3	50.0	50.0	44.4
Ours	<b>83.3</b>	<b>83.3</b>	<b>100.0</b>	<b>88.9</b>

TABLE IV

REAL-WORLD SEQUENTIAL CLUTTER REMOVAL RESULTS  
EACH CELL: TRANSPARENT&SPECULAR(MIXED)

	Pile		Packed	
	SR (%)	DR (%)	SR (%)	DR (%)
VGN	47.6(56.7)	32.4(44.7)	57.1(65.4)	49.3(59.7)
SwinDR-VGN	44.6(57.3)	29.7(39.5)	64.3(71.2)	53.7(66.3)
Ours	<b>65.9(72.4)</b>	<b>51.2(57.3)</b>	<b>82.2(81.6)</b>	<b>71.6(77.4)</b>

### F. Ablation Studies

To analyze the design of GraspNeRF, we conduct ablation studies on various configurations. Experiments are conducted

on the challenging pile scene in simulation under the sequential decluttering setting, which can be found in Table V.

We compare our end-to-end training strategy with two other schemes: 1)  $F_{scene}$  and  $F_{grasp}$  are trained separately, where  $F_{grasp}$  is trained using perfect depth-integrated TSDF, and 2) train  $F_{scene}$  first, then fix  $F_{scene}$  to train  $F_{grasp}$ . The big performance gap (especially between ours and 1)) demonstrates the benefits brought by end-to-end training, which minimizes the impact of the TSDF errors on grasping.

We then analyze the effect of predicting TSDF on grasping detection. We let  $F_{grasp}$  directly take the aggregated feature volume  $V_A$  in  $F_{scene}$  and thus remove the geometry supervision. Compared to TSDF input, the performance significantly degrades, demonstrating that supervised-learned geometric representation is essential for volumetric grasping learning.

To further analyze the synergies between neural radiance field and grasping, the volume rendering branch of  $F_{scene}$  and its color loss are removed. We observe that the performance drops, especially on transparent and specular object declutter rate, indicating that image supervision helps grasp detection on transparent and specular objects.

TABLE V

ABLATION STUDIES ON SEQUENTIAL PILE OBJECT REMOVAL  
EACH CELL: TRANSPARENT&SPECULAR(MIXED)

	SR (%)	DR (%)
Train $F_{scene}$ & $F_{grasp}$ separately	45.0(47.5)	32.5(33.9)
Fix $F_{scene}$ , train $F_{grasp}$	65.0(64.0)	41.1(40.2)
w/o TSDF	57.8(55.9)	37.5(36.0)
w/o volume rendering	66.0(65.9)	42.1(44.0)
Full model	<b>67.3(67.0)</b>	<b>46.2(45.8)</b>

## V. CONCLUSIONS

In this work, we propose GraspNeRF, a multiview RGB-based 6-DoF grasp detection network for transparent and specular objects. Our method extends the generalizable NeRF to grasping, enabling sparse input and direct real-time inference without optimization. We propose the joint learning of scene representation and grasping to exploit the synergy between them. We also present a large-scale synthetic multiview grasping dataset, leveraging domain randomization to bridge the sim-to-real gap. Experiments demonstrate our superiority over competing methods. One limitation lies in the volume grid, whose limited resolution may be the bottleneck to the performance of geometry reconstruction and grasping. We leave further improvements to future works.

## REFERENCES

- [1] H.-S. Fang, C. Wang, M. Gou, and C. Lu, "Graspnet-1billion: A large-scale benchmark for general object grasping," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 11 444–11 453.
- [2] M. Breyer, J. J. Chung, L. Ott, R. Siegwart, and J. Nieto, "Volumetric grasping network: Real-time 6 dof grasp detection in clutter," in *Conference on Robot Learning*. PMLR, 2021, pp. 1602–1611.
- [3] Z. Jiang, Y. Zhu, M. Svetlik, K. Fang, and Y. Zhu, "Synergies between affordance and geometry: 6-dof grasp detection via implicit representations," *Robotics: science and systems*, 2021.
- [4] M. Sundermeyer, A. Mousavian, R. Triebel, and D. Fox, "Contact-graspnet: Efficient 6-dof grasp generation in cluttered scenes," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021.
- [5] M. Gou, H.-S. Fang, Z. Zhu, S. Xu, C. Wang, and C. Lu, "Rgb matters: Learning 7-dof grasp poses on monocular rgbd images," in *Proceedings of the International Conference on Robotics and Automation (ICRA)*, 2021.
- [6] 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 (ICCV)*, October 2021, pp. 15 964–15 973.
- [7] H. Fang, H.-S. Fang, S. Xu, and C. Lu, "Transcg: A large-scale real-world dataset for transparent object depth completion and a grasping baseline," *IEEE Robotics and Automation Letters*, pp. 1–8, 2022.
- [8] S. Sajjan, M. Moore, M. Pan, G. Nagaraja, J. Lee, A. Zeng, and S. Song, "Clear grasp: 3d shape estimation of transparent objects for manipulation," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 3634–3642.
- [9] Q. Dai, J. Zhang, Q. Li, T. Wu, H. Dong, Z. Liu, P. Tan, and H. Wang, "Domain randomization-enhanced depth simulation and restoration for perceiving and grasping specular and transparent objects," in *European Conference on Computer Vision (ECCV)*, 2022.
- [10] J. Ichnowski, Y. Avigal, J. Kerr, and K. Goldberg, "Dex-nerf: Using a neural radiance field to grasp transparent objects," in *Conference on Robot Learning*. PMLR, 2022, pp. 526–536.
- [11] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "Nerf: Representing scenes as neural radiance fields for view synthesis," in *ECCV*, 2020.
- [12] J. Kerr, L. Fu, H. Huang, J. Ichnowski, M. Tancik, Y. Avigal, A. Kanazawa, and K. Goldberg, "Evo-nerf: Evolving nerf for sequential robot grasping," in *6th Annual Conference on Robot Learning*, 2022. [Online]. Available: <https://openreview.net/forum?id=Bxr45keYrf>
- [13] T. Müller, A. Evans, C. Schied, and A. Keller, "Instant neural graphics primitives with a multiresolution hash encoding," *ACM Trans. Graph.*, vol. 41, no. 4, pp. 102:1–102:15, Jul. 2022. [Online]. Available: <https://doi.org/10.1145/3528223.3530127>
- [14] A. Chen, Z. Xu, F. Zhao, X. Zhang, F. Xiang, J. Yu, and H. Su, "Mvs-nerf: Fast generalizable radiance field reconstruction from multi-view stereo," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 14 124–14 133.
- [15] Y. Liu, S. Peng, L. Liu, Q. Wang, P. Wang, C. Theobalt, X. Zhou, and W. Wang, "Neural rays for occlusion-aware image-based rendering," in *CVPR*, 2022.
- [16] T. Weng, A. Pallankize, Y. Tang, O. Kroemer, and D. Held, "Multi-modal transfer learning for grasping transparent and specular objects," *IEEE Robotics and Automation Letters*, vol. 5, no. 3, pp. 3791–3798, 2020.
- [17] H. Cao, J. Huang, Y. Li, J. Zhou, and Y. Liu, "Fuzzy-depth objects grasping based on fsg algorithm and a soft robotic hand," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 3948–3954.
- [18] Z. Zhou, T. Pan, S. Wu, H. Chang, and O. C. Jenkins, "Glass-loc: plenoptic grasp pose detection in transparent clutter," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 4776–4783.
- [19] J. Chang, M. Kim, S. Kang, H. Han, S. Hong, K. Jang, and S. Kang, "Ghostpose\*: Multi-view pose estimation of transparent objects for robot hand grasping," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 5749–5755.
- [20] L. Yen-Chen, P. Florence, J. T. Barron, T.-Y. Lin, A. Rodriguez, and P. Isola, "NeRF-Supervision: Learning dense object descriptors from neural radiance fields," in *IEEE Conference on Robotics and Automation (ICRA)*, 2022.
- [21] Y.-C. Lin, P. Florence, A. Zeng, J. T. Barron, Y. Du, W.-C. Ma, A. Simeonov, A. R. Garcia, and P. Isola, "MIRA: Mental imagery for robotic affordances," in *6th Annual Conference on Robot Learning*, 2022. [Online]. Available: <https://openreview.net/forum?id=AmPeAFzU3a4>
- [22] P. Wang, L. Liu, Y. Liu, C. Theobalt, T. Komura, and W. Wang, "Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction," *NeurIPS*, 2021.
- [23] B. O. Community, *Blender - a 3D modelling and rendering package*, Blender Foundation, Stichting Blender Foundation, Amsterdam, 2018. [Online]. Available: <http://www.blender.org>
- [24] A. Gropp, L. Yariv, N. Haim, M. Atzmon, and Y. Lipman, "Implicit geometric regularization for learning shapes," in *International Conference on Machine Learning*. PMLR, 2020, pp. 3789–3799.
- [25] E. Coumans and Y. Bai, "Pybullet, a python module for physics simulation for games, robotics and machine learning," <http://pybullet.org>, 2016–2021.