2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

ICGNet: A Unified Approach for Instance-Centric Grasping

René Zurbrügg[1*], Yifan Liu[1], Francis Engelmann[1], Suryansh Kumar[2],
Marco Hutter[1], Vaishakh Patil[1], Fisher Yu[1]

***Abstract*— Accurate grasping is the key to several robotic tasks including assembly and household robotics. Executing a successful grasp in a cluttered environment requires multiple levels of scene understanding: First, the robot needs to analyze the geometric properties of individual objects to find feasible grasps. These grasps need to be compliant with the local object geometry. Second, for each proposed grasp, the robot needs to reason about the interactions with other objects in the scene. Finally, the robot must compute a collision-free grasp trajectory while taking into account the geometry of the target object. Most grasp detection algorithms directly predict grasp poses in a monolithic fashion, which does not capture the composability of the environment. In this paper, we introduce an end-to-end architecture for object-centric grasping. The method uses pointcloud data from a single arbitrary viewing direction as an input and generates an instance-centric representation for each partially observed object in the scene. This representation is further used for object reconstruction and grasp detection in cluttered table-top scenes. We show the effectiveness of the proposed method by extensively evaluating it against state-of-the-art methods on synthetic datasets, indicating superior performance for grasping and reconstruction. Additionally, we demonstrate real-world applicability by decluttering scenes with varying numbers of objects.**
Videos and Code icgraspnet.github.io.

I. Introduction

The ability of robots to perform accurate and collision-free grasp maneuvers holds the potential for a wide array of applications in embodied intelligence [1] such as assembly, pick and place, and packaging. Despite impressive progress [2]–[6], predicting grasps and their accurate pose from a single pointcloud remains a challenging task. A successful grasp prediction requires not only understanding an object's geometry, but also its physical properties including its mass, shape and friction. Beyond single object grasping this task is especially difficult in multi-object settings that exhibit clutter and strong occlusions limiting object visibility.

Current methods for grasping within cluttered environments primarily rely on pointcloud observations from a single viewpoint [4]–[9]. Generally, these methods directly operate on fully observed pointclouds to predict accurate, collision-free grasp poses, and achieve impressive results [6].

However, existing work processes pointclouds on a holistic scene level, without explicitly reasoning about individual object instances. Applying these methods for pick-and-place or target-driven grasping tasks, often requires additional post-processing or external components. For example, numerous methods rely on given (groundtruth) segmentation masks [10]–[13], object templates [14,15] and predict object shapes by iteratively filtering each instance in the pointcloud [10]. This introduces an extra level of complexity and often results in sub-optimal predictions, specifically under heavy occlusions.

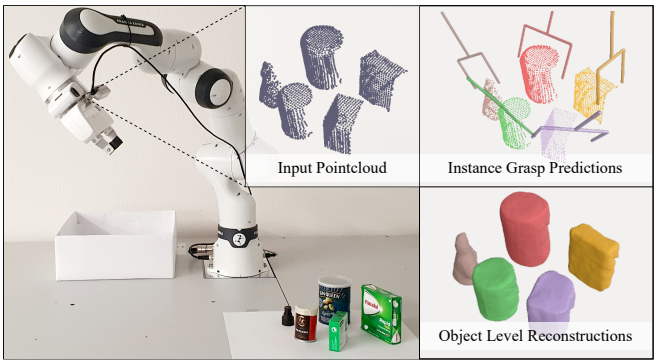


Fig. 1: Overview of our network predictions. Given a single view pointcloud we jointly predict instance segmentation masks, collision-free grasp predictions and reconstructions for each object.

Instead, our key contribution is to reason about grasping on an object level by explicitly modeling each individual instance which enables learning of shape priors and grasp affordances. Interestingly, as our method allows to predict object shapes and instances, it provides a clear interface for target-centric grasping and directly supports collision checks for manipulated instances which guarantees the collision-free removal and stable placements of unknown objects.

Specifically, we propose a unified architecture for instance centric grasp and shape prediction from single view pointclouds. The core idea of our method is to reason about an environment by extracting object embeddings on an instance level. To this end, we introduce a sparse feature volume, consisting of volumetric- and surface features at multiple scales. We then distill object-centric information into latent object embeddings through an iterative refinement process of masked cross- and self-attention. These features and object embeddings are used to model contact-based grasp affordances and object shapes as implicit fields. Additional object predictions such as semantics and pointwise instance assignments directly evolve from the refinement.

In experiments, significant improvements arise from our object-centric architecture, establishing a new state-of-the-art in the *packed* decluttering benchmark introduced in [4] while surpassing scene-centric task baselines [6,8]. These improvements are reflected in real-world experiments deployed on the Franka Research 3 robot. We additionally illustrate how instance-centric information can facilitate target-driven grasping (*i.e.*, "grasp instance number 1" or "grasp the bottle")

This research was partially supported by the ETH AI Center, ETH Zürich Career Seed Award and RobotX grant. The authors are with [1]ETH Zürich, [2]Visual Computing, School of PVFA, Texas A&M University. *Corresponding author zrene@ethz.ch.

and effectively prevent post-grasp object-object collisions.

II. Related Work

Deep Grasp Synthesis. Recent advancements in robotic grasping have predominantly utilized deep learning techniques to detect robotic grasps directly from sensor data. Unlike more traditional heuristics [16], these methods have demonstrated superior generalization performance towards previously unseen objects as well as achieving successful grasps in cluttered scenes. While several methods have achieved high success rates in 4-degree-of-freedom (4 DoF) top-down grasping [2,17], their capabilities are often limited when dealing with cluttered scenes or task-dependent grasping scenarios [9]. This limitation has led to an increased research focus on 6 DoF grasp predictions, which aim to predict the full grasp pose from visual observations. We further classify these methods in sampling- and scene-based approaches. Sampling-based methods sample different grasp candidates and use a detector or diffusion [18,19] network to refine them further. GPD [3] and PointNetGPD [20] learn to detect grasp poses in cluttered scenes from raw pointclouds by first sampling feasible grasp predictions and then using a CNN or PointNet classifier to predict the quality of each grasp based on the enclosed points. 6-DoF Graspnet [9] extends the way that grasp proposals are generated to the full SE(3) space by leveraging a variational auto-encoder for singulated objects. Scene-based approaches directly predict feasible grasp poses for the whole scene in one forward pass. Contact based methods such as Edge Grasp Networks [6] or Contact GraspNet [5] directly predict the SE(3) pose, width, and grasp quality for each point in the pointcloud. VGN [4] and GIGA [8] predict grasps for each voxel in the reconstructed TSDF using 3D CNN.

Simultaneous Shape Reconstruction and Grasp Estimation. Recently, joint prediction of scene reconstructions and grasp poses have been studied in more detail [7,8,10,21] as both tasks are correlated and fundamental for environment interactions. Early works by Varley *et al.* [21] voxelized an observed pointcloud and utilized a 3D CNN as well as a marching cube algorithm for reconstruction. The reconstructed mesh is then used in combination with GraspIt! [22] to predict feasible grasp candidates. PointSDF [7] directly learns a signed distance field from the initial pointcloud and conditions a grasp classification on the latent representation of the occupancy decoder. ShellGrasp-Net [10] jointly learns the camera-ray intersections with singulated objects by predicting grasp-affordances as well as entry and exit-depth maps. GIGA [8] extends grasp-prediction and shape reconstruction to cluttered scenes. Their methods combine VGN [4] and Convolutional Occupancy Networks [23] to learn grasps and occupancy as continuous functions over 3D coordinates. They show that learning both tasks jointly can improve the grasp detection and reconstruction of graspable regions.

Target Driven Grasping. Prior work on grasp prediction typically deals with singulated objects [10,21,24] or predicts grasp affordances for the full scene without any notion of instances [4,8,20]. While these methods can be adapted for target-driven grasping by segmenting relevant objects and predicting grasp poses for them, the resulting grasps may not guarantee collision-free execution. To address this, Murali *et al.* [12] introduce a collision network that post-processes predicted grasps and discards or refines those that would cause collisions. [25] rely on accurate bounding boxes of the objects to post-process scene-based grasps, assuming that the fingertips of the gripper lie inside the bounding box of the target object. Sundermeyer *et al.* [5] propose contact-based grasp detectors that directly predict grasp poses for each observable point in the input pointcloud of a cluttered scene. This enables the selective grasping of objects based on the semantic class of the associated contact point. However, all these approaches rely on external segmentation modules to predict instance segmentation masks, adding complexity to the overall pipeline. Contrary to that, our work introduces a unified method for scene understanding which combines panoptic segmentation, reconstruction, and grasp detection.

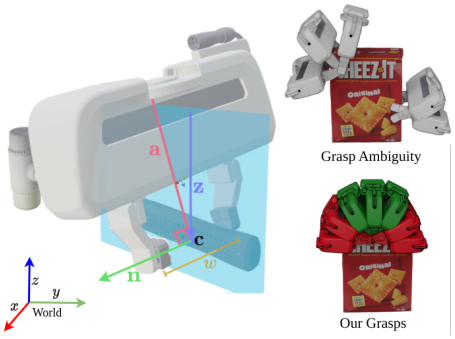


Fig. 2: Grasp Representation. *Left:* $\mathbf{c}$ is the contact point of the closed gripper and $\mathbf{n}$ is its estimated surface normal. $\mathbf{a}$ is the approach direction of the gripper. Given the gravity vector $\mathbf{z}$ and surface normal $\mathbf{n}$, $\mathbf{a}$ can be uniquely defined by the approach angle α. *Top Right:* Grasp ambiguity of different grasp representations. When dealing with a particular contact or gripper center, there can be multiple feasible approach direction resulting in a successful grasp. *Bottom Right:* For each contact point, our representation enables the prediction of grasp qualities for different gripper orientations perpendicular to the surface normal.

III. Method

Problem Formulation. Our setup consists of a robot arm with a parallel-jaw gripper operating within a planar tabletop workspace as shown in Fig. 1. The workspace contains multiple rigid objects placed on the tabletop. The objects are either placed randomly in an upright position (packed) or dumped from a box (pile) (Fig. 4). Prior to each object interaction, the scene is captured once using a depth camera from a static, randomized position following [6]. The captured depth image is converted into a pointcloud and fed into the proposed model. The model jointly reconstructs the full 3D shape of each object and predicts stable[1] grasps that do not collide with the other objects and the environment.

Formally, given a pointcloud $P_{scan} \in \mathbb{R}^{N_s \times 3}$ consisting of K different objects from C different classes, we predict

[1]Grasps which are able to steadily hold the object even under minor movement or perturbations.

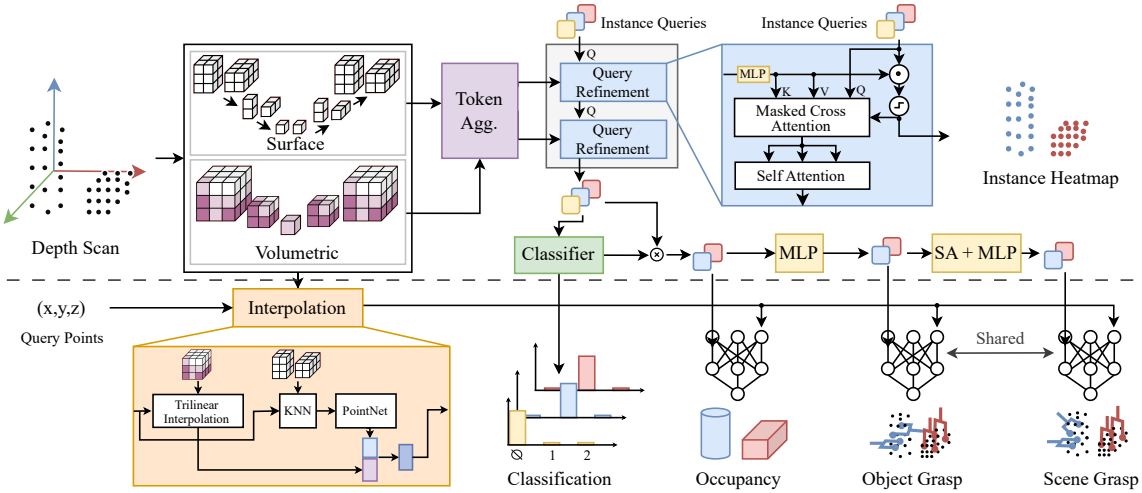


Fig. 3: Model Overview. Given an input pointcloud, we voxelize the pointcloud and extract volumetric and surface features at multiple scales using a sparse Minkowski- [26] and dense U-Net [27]. The surface features are enriched with volumetric information and treated as tokens with positional encodings based on voxel locations. Masked attention iteratively refines instance queries by cross-attending to extracted sparse tokens. This process allows each latent query to focus on a specific instance and to be classified as *"<semantic class>"* or *"no object"*. The refined queries condition the task-specific decoders to model the occupancy of each instance directly or to predict grasp affordance scores and gripper widths.

per-point instance labels $I \in \{1, \dots, K\}^{N_p}$ and semantic labels $S \in \{1, \dots, C\}^{N_p}$. To reconstruct the object shapes, we additionally predict occupancy values $o \in \{0, 1\}^{N_q}$ for a set of N_q query points $P_{query} \in \mathbb{R}^{N_q \times 3}$ for each instance. Finally, we predict grasp affordances (*i.e.* the probability of a grasp being successful) for N_α different approach directions $\mathcal{A} \in [0, 1]^{N_\alpha \times N_q}$ for each instance.

Grasp Representation. Existing grasp prediction methods typically rely on a single ground truth grasp for a given contact point [5] or gripper center position [4,10]. However, when a fixed contact point on an object is considered, there can be multiple equally "good" grasp orientations for removing the object from the scene (Fig. 2, *left*). In line with [6], we argue that the *distribution* of valid grasps given a query point provides more consistent supervision. This approach aims to reduce the number of positive predictions that are mistakenly treated as "wrong" while generating more diverse grasp proposals. Nonetheless, learning a continuous distribution over all potential grasp poses for each contact point is intractable. To address this, we enforce the approach direction $\mathbf{a}$ to be perpendicular to the surface normal, thereby restricting the approach vector to lie on a plane given a contact point[2]. Furthermore, we discretize the approach direction into a set of discrete angles and formulate the grasp prediction as a multi-class classification problem (Fig. 2, *bottom-right*). With this, our contact-based grasps take the following form:

$$\mathcal{G}_{\text{contact}} = (\mathbf{c}, \mathbf{n}, \mathbf{s}, w), \tag{1}$$

$\mathbf{c} \in \mathbb{R}^3, \mathbf{n} \in \mathbb{R}^3, \mathbf{s} \in [0, 1]^{N_\alpha}, w \in [0, w_{max}]$ with $\mathbf{c}$ being the contact point, $\mathbf{n}$ the surface normal, $\mathbf{s}$ the grasp affordance (= probability of successful grasp) values for each discretized approach direction and w_{max} being the maximal opening width of the gripper.

[2]While this limits the diversity of grasps, the assumption is often closely met in practice when requiring force closure of the resulting grasp.

The corresponding set of SE(3) poses for a given grasp affordance prediction g and gravity vector $\mathbf{z}$ is given by the tool-center point and computed as

$$\mathcal{G}_{SE3} = \{(\mathbf{t}_g, R_g^i)\}_{i=0}^{N_\alpha - 1} \text{ with } \mathbf{t}_g = \mathbf{c} + \frac{w_{max} - w}{2}\mathbf{n}, \tag{2}$$

$$R_g^i = R_y(\alpha_i) \cdot \begin{bmatrix} | & | & | \\ \mathbf{z} \times \mathbf{n} & \mathbf{n} & (\mathbf{z} \times \mathbf{n}) \times \mathbf{n} \\ | & | & | \end{bmatrix}, \tag{3}$$

where the approach angle α_i is within $\{-90°, \dots, 90°\}$ and R_y denotes to rotation matrix around the y axis.

Singularity: The proposed grasp representation results in a singularity when the surface normal and gravity vector coincide. If this is the case, ($|\mathbf{z} \cdot \mathbf{n}| > 0.98$), the x-axis of the grasp is chosen to align with the table surface pointing in the arbitrary x direction of the world frame.

Model Architecture. Fig. 3 shows our end-to-end instance aware grasp prediction model. It consists of two stages:
Encoder. Given a pointcloud in the world frame, our encoder network performs several key tasks. It extracts both sparse and dense features at multiple resolutions utilizing a sparse 3D-UNet architecture [26] for surface features and dense 3D-UNet architecture [27] for volumetric features. The sparse features are enriched with volumetric information through the proposed *token aggregation* module. To decompose the voxelized scene into individual instances, we apply multiple *instance query refinement* modules similar to Mask3D [28]. Further, we rely on a *classification head* to assign a class to each latent representation and filter out unmatched queries using a *no-object* class. All valid instance queries are directly used as latent embeddings for an occupancy network. Additionally, we employ a final Self Attention and MLP layer to exchange inter-object information between the queries and convert them to the affordance domain.
Decoder. Our decoder is designed as an implicit neural field and uses world coordinates (x, y, z) as input. It predicts

Method	G	R	S	#Params	Inference	Latency
VGN [4]	✓	–	–	0.3 M	3 ms	7 ms
GIGA [8]	✓	✓	–	0.6 M	26 ms	30 ms
GIGA-HR [8]	✓	✓	–	0.6 M	56 ms	66 ms
EdgeGraspNet [6]	✓	–	–	3.0 M	26 ms	34 ms
VN-EdgeGraspNet [6]	✓	–	–	1.7 M	264 ms	306 ms
ICGNet (Ours)	✓	✓	✓	3.2 M	137 ms	138 ms

TABLE I: Tasks, Model Size, and Runtime. We report the supported tasks, number of parameters, model inference times, and average latency (including grasp postprocessing) measured on a GeForce RTX 3080. Although, our network consists of slightly more parameters compared to [6], ICGNet (Ours) is the only architecture that is able to predict grasps (G), reconstruction (R) and semantics (S) while having a lower latency than VN-EdgeGraspNet [6].

occupancy p_{occ} and grasp affordances $g \in \mathcal{G}_{aff}$ for each instance. In addition, predictions such as classification and instance heatmaps are directly extracted from the encoder as shown in Fig. 3. These predictions are made for each instance query (class) and each point within the input pointcloud (instance id). To model each implicit decoder, we use a series of MLPs with residual connections following the approach proposed in [29]. Further, we concatenate each instance query with its positional encoding, which captures spatial information. Moreover, the queried coordinates (x, y, z) are enriched by incorporating the surface and volumetric features using our *interpolation* module. These enriched coordinates are then concatenated with the positional encodings of the original coordinates.

Token Aggregation. Ideally, our goal is to extract dense features at a high resolution and directly feed them into the instance query refinement module. However, this approach comes with significant drawbacks, notably increased memory consumption and computation cost, rendering it infeasible for most applications. On the other hand, sparse neural networks scale with the number of occupied voxels, making them suitable for scaling to large areas. This advantage arises as the number of occupied voxels typically grows slower than the total number of voxels. Here, we combine the best of both by enriching each occupied surface feature with a volumetric context that is extracted from the volumetric feature grid. To this end, each surface feature is concatenated with the feature of the nearest volumetric voxel, similar to PointNet [30]. This enables the volumetric feature backbone to operate on a larger resolution extracting context information to enrich the sparse features used in the query refinement.

Query Refinement. Given a set of K instance queries, we apply a series of masked cross-attention and self-attention to extract instance-centric information given the scene-level features extracted from the U-Net backbones. We adapt Mask3D [28] and add an MLP layer to further process the extracted scene features. We also add Fourier positional encodings [31] based on voxel positions and use farthest-point sampling to sample initial instance query positions.

Classifier. The classifier predicts a class label including a *non-object* class to address the varying number of instances in a scene. A small MLP followed by a softmax activation predicts a categorical distribution over the desired $C + 1$ classes. For our experiments, we manually annotate the points with class labels that correlate with the respective shape, consisting of six categories ("mug", "box", "can", "bottle", "cylindric", "ball" and "other"). Having the notion of semantic classes, allows to predict grasp candidates and reconstructions for individual objects as shown in Fig. 4.

Interpolation. Directly passing the input coordinates or the respective positional encoding to the occupancy network produces sub-optimal reconstructions and grasp predictions due to the limited scene information. Therefore, the input coordinates are concatenated with per-point features extracted from the sparse and dense feature grids. Volumetric features are extracted using trilinear interpolation. Interpolated features from the sparse surface volume stem from a K-Nearest Neighbour (KNN) search. Naively averaging the KNN leads to identical features for points on opposite sides of the surface. We therefore use a small PointNet for feature aggregations which relies on the KNN features and distance to the nearest neighbors. Both dense and sparse features are concatenated and fed through a MLP to extract the enriched embeddings for each coordinate.

Loss Formulation. The proposed model is trained in an end-to-end fashion. The training uses instance and semantic annotations as well as occupancy and grasp poses obtained from simulation. The implemented loss consists of binary cross-entropy, DICE and squared-error losses and is given as

$$\mathcal{L} = \mathcal{L}^{BCE}_{\texttt{inst}} + \mathcal{L}^{DICE}_{\texttt{inst}} + \mathcal{L}^{BCE}_{\texttt{sem}} + \mathcal{L}^{Grasp} + \mathcal{L}^{BCE}_{\texttt{occ}}, \tag{4}$$

with

$$\mathcal{L}^{Grasp} = \mathcal{L}^{BCE} + \|\cdot\|^2_2\,, \tag{5}$$

where the first three terms refer to the panoptic segmentation task and are calculated with respect to the input pointcloud $P_{\texttt{scan}}$. $\mathcal{L}^{Grasp}$ refers to the grasp predictions for each instance. This loss is calculated with respect to the sampled grasp coordinates which may differ from the actual observed pointcloud. We use the cross entropy loss to supervise the approach classification and the squared L2 loss for the gripper width. Finally, the reconstruction loss $\mathcal{L}^{BCE}_{\texttt{occ}}$ supervises the predicted occupancy for each instance and is computed with respect to an additional set of randomly sampled points within the scene. More information about the labels and dataset is provided in Sec. IV. Note that there is no ordering in the set of instances in a scene and we need to establish correspondence between the set of predicted and set of groundtruth instances during training. Following [28], we use Hungarian matching based on the instance segmentation[3] loss to find unique assignments and apply the same loss function to the predictions at multiple resolutions.

IV. Experiments

Training Details. Our model is trained on simulated data and makes use of zero-shot to transfer to a real Franka Research 3 arm from Franka Emika We train the proposed network for 60 epochs with an effective batch size of 8 using AdamW [32]

[3]We limit the matching to the segmentation task since calculating every possible assignment of the occupancy field is computationally extensive.

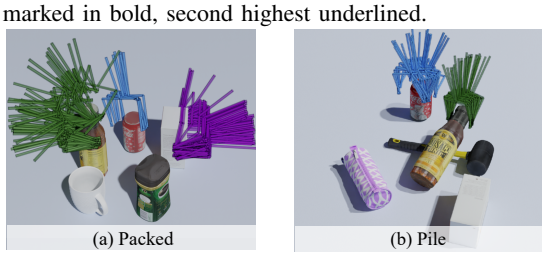
Method	Grasping				Reconstruction			
	Packed		Pile		Packed		Pile	
	GSR (%) ↑	DR (%) ↑	GSR (%) ↑	DR (%) ↑	C-L1 [mm] ↓	IoU (%) ↑	C-L1 [mm] ↓	IoU (%) ↑
PointNetGPD [20]	79.3 ± 1.8	82.5 ± 2.9	75.6 ± 2.3	77.0 ± 2.8	–	–	–	–
VGN [4]	80.2 ± 1.6	86.2 ± 2.0	64.9 ± 2.2	69.1 ± 3.2	–	–	–	–
GIGA† [8]	89.9 ± 1.7	87.6 ± 2.0	76.3 ± 2.4	80.9 ± 4.1	2.5 ± 1.2	82 ± 6.9	3.4 ± 1.4	71 ± 8.6
GIGA-HR† [8]	91.4 ± 1.5	88.5 ± 1.4	86.5 ± 1.2	80.8 ± 1.9	2.5 ± 1.2	82 ± 6.9	3.4 ± 1.4	71 ± 8.6
EdgeGraspNet* [6]	90.6 ± 0.9	93.9 ± 0.7	91.0 ± 2.0	93.7 ± 2.3	–	–	–	–
VN-EdgeGraspNet* [6]	90.4 ± 2.5	92.8 ± 1.0	91.9 ± 0.8	92.7 ± 1.2	–	–	–	–
ICGNet (Ours)	**97.7 ± 0.9**	**97.5 ± 0.3**	**92.0 ± 2.6**	**94.1 ± 1.4**	**2.3 ± 1.1**	**84 ± 6.1**	**2.9 ± 1.9**	**77 ± 9.9**

TABLE II: Comparison to State-of-the-Art on Synthetic data. Simulated results on the packed and piled scenes using the evaluation setup from [6]. We report grasp success rate (GSR), declutter rate (DR), and reconstruction performance, Chamfer L1 distance (C-L1), IoU. ICGNet denotes our approach that predicts grasp poses and occupancy for each individual instance. We retrain (†) GIGA on randomized viewpoints and 2M grasps for each environment separately and re-evaluate (*) the pre-trained checkpoints provided by [6] multiple times. The scores for VGN and PointnetGPD are taken from [6]. Highest number marked in bold, second highest underlined.

with a learning rate of $1e{-}3$ and a linear warmup, cosine annealing learning rate schedule [33]. Additionally, we make use of early stopping based on the F1 score of the grasp affordances calculated on the validation set. We find the F1 score to be a more robust performance metric as the instance-wise affordance scores are heavily imbalanced. The full training takes ∼45h on a Nvidia Titan RTX GPU.

Simulation Environment and Dataset. We leverage the simulation setup and the object dataset introduced in VGN [4]. This dataset consists of 303 train and 40 test objects from various sources [34]–[36]. The experimental setup involves a free-floating Franka Emika Gripper and we sample grasps and occupancy values in 30 cm^3 sized tabletop workspace. Contrary to [4,8], which randomly samples the gripper centers close to the surface and stores the resulting best gripper orientation as a ground truth label, we adapt a different strategy. We select an observed point and corresponding surface normal, then execute the grasp from different approach angles as outlined in Sec. III. Specifically, we sample twelve different approach angles for each contact point and the final gripper width as well as the grasped object are recorded. Additionally, we sample the occupancy values for each scene. A total of 200'000 occupancy values are sampled of which 70% are uniformly sampled and 30% are sampled closer to the object surfaces to allow more accurate reconstruction. We store the occupancy value for each object yielding a total of 200'000 $\times$ k binary labels for each scene. We further make use of the *packed* and *piled* splits introduced by [4] and sample a total of 1M and 2M grasps from each split, resulting in 3M grasps from 15'000 scenes of which 13'500 and 1'500 are used for training and validation respectively.

Observations. For each scene, a single depth-image is captured and the unprojected pointcloud in the world frame is used as the input to our network. We randomize the depth camera pose to be uniformly located at the spherical coordinates $(r \in [0.48, 0.72], \theta \in [0, \frac{\pi}{4}], \phi \in [0, 2\pi])$ looking at the center of the workspace. To allow for better sim-to-real transfer, we further randomly rotate the pointcloud, add noise sampled from $\mathcal{N}(0, 0.002)$ and apply 3D elastic deformation [37]. In addition, the scanned pointcloud is divided into regions of 2 cm, which are randomly erased with probability $p = 0.2$. The tabletop is removed based on the points height.



(a) Packed (b) Pile

Fig. 4: Our grasp predictions on simulated, unseen test objects from [39]. Predicted grasps for "bottle", "can" and "box" in the packed *(left)* and pile setup *(right)*. More qualitative examples at can be found on the project page

Grasp Selection and Reconstruction. For each scene, 32 instance queries are spawned and refined using our query refinement and the observed pointcloud. The pointcloud is denoised using statistical outlier filters. It is then down-sampled using a voxel size of 2 mm before computing the grasp affordances for each point. Surface normals for each contact point are estimated using covariance analysis from Open3D [38]. We then execute the grasp associated with the highest affordance score which is collision-free, given our implicit shape encodings of the scene and height of the table. To evaluate the scene reconstruction quality, we compute the occupancy field for the whole scene by combining instance level predictions $p^{scene}_{\text{occ}}(x) = \max_{\mathcal{I} \in Inst} p_{\text{occ}}(x|\mathcal{I})$ and use adaptive marching cubes algorithm [29].

Simulation Results. We evaluate our method using the evaluation process from [6] for the *pile* and *packed* scenes. For the *grasping* task, we report the mean and standard deviation of over 4 different runs, each consisting of 100 different scenes. We report the Grasp success rate (GSR, percentage of successful grasps) and declutter rate (DR, percentage of removed objects after the task has finished) For the *reconstruction* task, we randomly sample 100 different *pile* and *packed* scenes using the objects from the test split. The Chamfer L1 and IoU scores are calculated following [29]. Scores are shown in Tab. II where ICGNet exhibits superior or equal performance compared to other methods in both grasp success and reconstruction, achieving the highest GSR, DR, IoU and lowest Chamfer L1 distance. Our method significantly outperforms the baseline methods on the *packed* dataset, highlighting the advantage of our grasp representation and instance priors. We additionally compare the model size and performance scores in I, showing that our model is competitive

Method	Packed		Pile	
	GSR ↑	DR↑	GSR (%) ↑	DR (%) ↑
GIGA-HR[1]	73%	59%	83%	71%
VN-EdgeGraspNet[2]	84%	76%	81%	71%
ICGNet (Ours)	**90%**	**88%**	**90%**	**83%**

TABLE III: Clutter Removal Experiment on Real-World Data. We report the grasp success rate (GSR) as well as declutter rate (DR) evaluated in the real world setting.

in size and inference speed while combining multiple tasks.

Challenging Viewpoints. We further evaluate our method on the challenging task of grasp detection from top-down viewpoints. To this end, we modify the evaluation algorithm from [6] to spawn the camera facing downwards at a random angle $\theta \in [0, 2°]$. This setup is extremely challenging for contact-based grasping since almost no graspable regions are observed and the surface normals are harder to estimate, resulting in significantly lower declutter rates of (85% and 72%) on the packed and pile environments respectively. To overcome this limitation of contact based grasping, we leverage our multi-task architecture and resample, unobserved points from the (implicit) object surfaces if no feasible grasp is found. We then use these new surface points for grasp prediction, allowing us to improve the declutter rate to **90**%(*+6%*) and **94**% (*+31%*) respectively.

Real World Experiments. We validate our method and compare it against GIGA [8] and VN-EdgeGraspNet [6] on different real-world declutter tasks. The experimental setup is depicted in Fig. 5. A RealSense D415 RGB-D camera is attached to the gripper on a Franka Research 3 Arm. We use a 30×30 cm workspace to allow for comparison with [8].[4] We use a total of 17 different test objects. Of these, 3-6 objects are randomly placed on a 30×30 cm workspace. Before each experiment, we collect a top-down image of all objects, which are placed on a grid to ensure the repeatability and a fair comparison of all methods. Before each grasp trial, the arm moves to the designated image acquisition position (see Fig. 5) and captures depth image, which is post-processed and fed into the network. We use MoveIt! [40] to plan and execute collision-free grasp maneuvers. For collision checking, we rely on the predicted scene reconstructions or use the collision mesh from the tsdf volume.Given a set of predicted grasp poses, we follow the procedure of [6] and filter the grasps for a 0.9 confidence level and execute the pose with the highest z component that is kinematically feasible and collision-free. If no grasp is found, we lower the confidence level to 0.8 and 0.7. The current task is terminated if two consecutive grasp failures occur or if no grasps are found for five sequential observations. Grasp success rate (GSR) and declutter rate (DR) for different clutter categories are reported in Tab. III.

We observe that the reported success and declutter rates follow the same trends but are lower than what is usually reported in [6,8] (grasp success rates of 90% and almost perfect declutter rate). We attribute this to our hardware setup (a panda gripper that has a bigger collision shape and smaller gripper width than the robotiq gripper used in [8]) and the object set, which is quite challenging as some materials are slightly reflecting. Additionally, our test objects often require the gripper to be fully open and some objects might not be re-grasped once tipped over due to their height or aspect ratio, prohibiting the scene from being fully decluttered. Our results show that our network successfully transfers to the real world and achieves higher success and declutter rates compared to [6,8]. Additionally, we observe an increased amount of object-object collisions for [6,8] when moving to the drop-off location, as their method does not directly support collision checks with the grasped objects. Our method allows for instance-aware reconstructions of the scenes and can therefore anticipate the attached object geometry and often finds a collision-free path to the drop-off location.

[4]Note that our method is capable of inferring grasps on larger scenes due to the combination of dense and sparse voxelization.

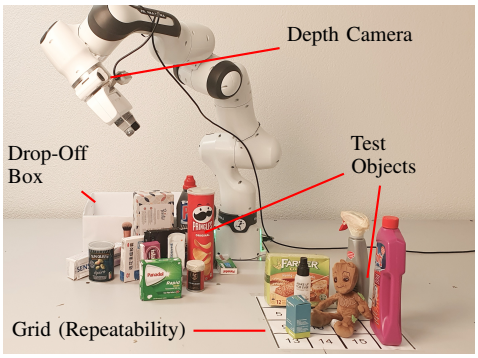


Fig. 5: Real world experimental setup. We use 17 different objects of which 3-6 are placed on a $30\,\text{cm}^3$ workspace.

Limitations. Although our architecture demonstrates remarkable performance, we have identified certain limitations. In line with [28], we observe that our network occasionally over-segments or combines instances even when they are not in contact. Most of these errors can be corrected by post-processing the network predictions, which comes at the cost of higher latency. Additionally, we observe that the reconstructions can be further improved by learning occupancy for the full scene instead of for each instance independently. We plan to address these limitations in future work by utilizing dilated attention to to improve spatial coverage of the query and include color into the pointcloud features. Additionally, we plan to train our network on datasets with diverse objects in order to learn robust object priors.

V. Conclusion

We introduce ICGNet, a unified architecture for target-driven grasping in cluttered environments that allow for instance centric, target-driven grasping and object reconstruction from a single view pointcloud. We evaluate our network performance both in simulation and challenging real-world scenes. Our results show, that the proposed method outperforms the current state of the art for grasping in cluttered environments and can be successfully transferred to the real world. A clear direction for future work includes learning more robust instance priors on more extensive and diverse datasets and including language embeddings for target-driven grasping.

REFERENCES

[1] A. Cangelosi, J. Bongard, M. H. Fischer, and S. Nolfi, "Embodied intelligence," *Springer handbook of computational intelligence*, pp. 697–714, 2015.

[2] 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.

[3] M. Gualtieri, A. ten Pas, K. Saenko, and R. W. Platt, "High precision grasp pose detection in dense clutter," *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 598–605, 2016.

[4] M. Breyer, J. J. Chung, L. Ott, R. Y. Siegwart, and J. I. Nieto, "Volumetric grasping network: Real-time 6 dof grasp detection in clutter," in *Conference on Robot Learning (CoRL)*, 2021.

[5] M. Sundermeyer, A. Mousavian, R. Triebel, and D. Fox, "Contact-graspnet: Efficient 6-dof grasp generation in cluttered scenes," *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 13 438–13 444, 2021.

[6] H. Huang, D. Wang, X. Zhu, R. Walters, and R. Platt, "Edge grasp network: A graph-based se (3)-invariant approach to grasp detection," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 3882–3888.

[7] M. Van der Merwe, Q. Lu, B. Sundaralingam, M. Matak, and T. Hermans, "Learning continuous 3d reconstructions for geometrically aware grasping," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 11 516–11 522.

[8] 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 (RSS)*, 2021.

[9] A. Mousavian, C. Eppner, and D. Fox, "6-dof graspnet: Variational grasp generation for object manipulation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[10] N. Chavan-Dafle, S. Popovych, S. Agrawal, D. D. Lee, and V. Isler, "Simultaneous object reconstruction and grasp prediction using a camera-centric object shell representation," *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1396–1403, 2021.

[11] Y. Li, T. Kong, R. Chu, Y. Li, P. Wang, and L. Li, "Simultaneous semantic and collision learning for 6-dof grasp pose estimation," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 3571–3578.

[12] A. Murali, A. Mousavian, C. Eppner, C. Paxton, and D. Fox, "6-dof grasping for target-driven object manipulation in clutter," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 6232–6238.

[13] Y. Yang, H. Liang, and C. Choi, "A deep learning approach to grasping the invisible," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 2232–2239, 2020.

[14] M. Danielczuk, A. Kurenkov, A. Balakrishna, M. Matl, D. Wang, R. Martín-Martín, A. Garg, S. Savarese, and K. Goldberg, "Mechanical search: Multi-step retrieval of a target object occluded by clutter," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 1614–1621.

[15] X. Lou, Y. Yang, and C. Choi, "Collision-aware target-driven object grasping in constrained environments," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 6364–6370.

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

[17] P. Raj, A. Kumar, V. Sanap, T. Sandhan, and L. Behera, "Towards object agnostic and robust 4-dof table-top grasping," in *2022 IEEE 18th International Conference on Automation Science and Engineering (CASE)*, 2022, pp. 963–970.

[18] T. Weng, D. Held, F. Meier, and M. Mukadam, "Neural grasp distance fields for robot manipulation," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 1814–1821.

[19] J. Urain, N. Funk, J. Peters, and G. Chalvatzaki, "Se (3)-diffusionfields: Learning smooth cost functions for joint grasp and motion optimization through diffusion," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 5923–5930.

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

[21] J. Varley, C. DeChant, A. Richardson, J. Ruales, and P. Allen, "Shape completion enabled robotic grasping," in *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 2017, pp. 2442–2447.

[22] A. T. Miller and P. K. Allen, "Graspit! a versatile simulator for robotic grasping," *IEEE Robotics & Automation Magazine*, vol. 11, no. 4, pp. 110–122, 2004.

[23] S. Peng, M. Niemeyer, L. Mescheder, M. Pollefeys, and A. Geiger, "Convolutional occupancy networks," in *European Conference on Computer Vision (ECCV)*, 2020.

[24] 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.

[25] M. Breyer, L. Ott, R. Siegwart, and J. J. Chung, "Closed-loop next-best-view planning for target-driven grasping," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 1411–1416.

[26] C. Choy, J. Gwak, and S. Savarese, "4d spatio-temporal convnets: Minkowski convolutional neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 3075–3084.

[27] Ö. Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox, and O. Ronneberger, "3d u-net: learning dense volumetric segmentation from sparse annotation," in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2016: 19th International Conference, Athens, Greece, October 17-21, 2016, Proceedings, Part II 19*. Springer, 2016, pp. 424–432.

[28] J. Schult, F. Engelmann, A. Hermans, O. Litany, S. Tang, and B. Leibe, "Mask3d: Mask transformer for 3d semantic instance segmentation," in *International Conference on Robotics and Automation (ICRA)*, 2023.

[29] L. M. Mescheder, M. Oechsle, M. Niemeyer, S. Nowozin, and A. Geiger, "Occupancy networks: Learning 3d reconstruction in function space," *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4455–4465, 2018.

[30] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, 2017, pp. 652–660.

[31] M. Tancik, P. Srinivasan, B. Mildenhall, S. Fridovich-Keil, N. Raghavan, U. Singhal, R. Ramamoorthi, J. Barron, and R. Ng, "Fourier features let networks learn high frequency functions in low dimensional domains," *Advances in Neural Information Processing Systems*, vol. 33, pp. 7537–7547, 2020.

[32] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," *arXiv preprint arXiv:1711.05101*, 2017.

[33] I. Loshchilov and F. Hutter, "Sgdr: Stochastic gradient descent with warm restarts," *arXiv preprint arXiv:1608.03983*, 2016.

[34] B. Calli, A. Singh, J. Bruce, A. Walsman, K. Konolige, S. Srinivasa, P. Abbeel, and A. M. Dollar, "Yale-cmu-berkeley dataset for robotic manipulation research," *The International Journal of Robotics Research (T-RO)*, vol. 36, no. 3, pp. 261–268, 2017.

[35] A. Kasper, Z. Xue, and R. Dillmann, "The kit object models database: An object model database for object recognition, localization and manipulation in service robotics," *The International Journal of Robotics Research (T-RO)*, vol. 31, no. 8, pp. 927–934, 2012.

[36] A. Singh, J. Sha, K. S. Narayan, T. Achim, and P. Abbeel, "Bigbird: A large-scale 3d database of object instances," *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 509–516, 2014.

[37] G. van Tulder. (2021) elasticdeform: Elastic deformations for N-dimensional images. Zenodo. [Online]. Available: https://doi.org/10.5281/zenodo.4569691

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

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

[40] D. Coleman, I. Sucan, S. Chitta, and N. Correll, "Reducing the barrier to entry of complex robotic software: a moveit! case study," *Journal of Software Engineering for Robotics*, 2014.