Target-referenced Reactive Grasping for Dynamic Objects

Jirong Liu[1,2], Ruo Zhang[2], Hao-Shu Fang[2], Minghao Gou[2], Hongjie Fang[2],
Chenxi Wang[2,3], Sheng Xu[2], Hengxu Yan[2], Cewu Lu[1,2]†
[1]Shanghai Qi Zhi institute, [2]Shanghai Jiao Tong University, [3]Flexiv Robotics, LTD
jirong@sjtu.edu.cn, {ruozhang0608, fhaoshu}@gmail.com,
{gmh2015, galaxies, wcx1997, xs1020, hengxuyan, lucewu}@sjtu.edu.cn

Abstract

Reactive grasping, which enables the robot to successfully grasp dynamic moving objects, is of great interest in robotics. Current methods mainly focus on the temporal smoothness of the predicted grasp poses but few consider their semantic consistency. Consequently, the predicted grasps are not guaranteed to fall on the same part of the same object, especially in cluttered scenes. In this paper, we propose to solve reactive grasping in a target-referenced setting by tracking through generated grasp spaces. Given a targeted grasp pose on an object and detected grasp poses in a new observation, our method is composed of two stages: 1) discovering grasp pose correspondences through an attentional graph neural network and selecting the one with the highest similarity with respect to the target pose; 2) refining the selected grasp poses based on target and historical information. We evaluate our method on a large-scale benchmark GraspNet-1Billion. We also collect 30 scenes of dynamic objects for testing. The results suggest that our method outperforms other representative methods. Furthermore, our real robot experiments achieve an average success rate of over 80 percent. Code and demos are available at: https://graspnet.net/reactive.

1. Introduction

Reactive grasping is in great demand in the industry. For instance, in places where human-robot collaboration is heavily required like factories, stress on laborers will be significantly relieved if robots can receive tools from humans and complete the harder work for laborers. Such a vision is based on reactive grasping.

On the contrary to static environments, in reactive grasping, dynamic task setting poses new challenges for algorithm design. Previous research in this area mainly focuses on planning temporally smooth grasps [22, 42] to avoid wavy and jerky robot motion. Few of them pay attention to its semantic consistency. In short, given a targeted grasp at the first frame, we want the robot to grasp the same part of the object in the following frames. Additionally, it is not guaranteed that grasp predictions made by classical methods fall on the same object in cluttered scenes. Hence, most of their experiments are conducted on single-object scenes. Unlike previous works, this work is aimed at achieving temporally smooth and semantically consistent reactive grasping in clutter given a targeted grasp. We refer to such a task setting as target-referenced reactive grasping as shown in Fig.1. Note that despite robot handover is a major application scenario of reactive grasping, this work focuses on a more general task setting - dynamic object grasping.

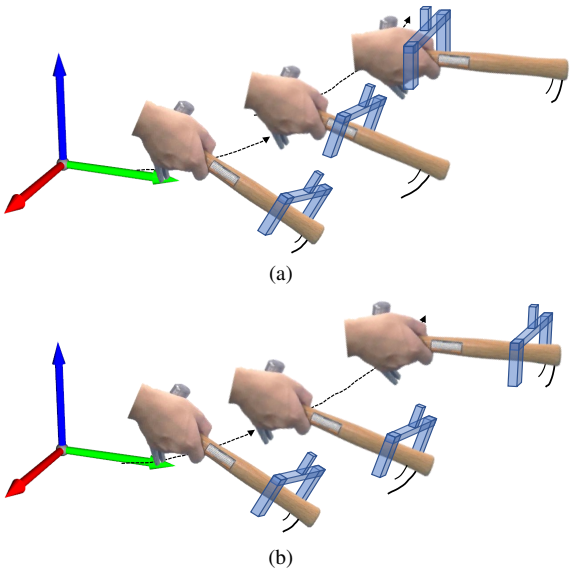

Figure 1. (a) Classic reactive grasping guarantees the smoothness of the grasp poses but cannot predicts grasps on the same part of the hammer. (b) Our target-referenced reactive grasping takes semantic consistency into consideration. The generated grasp poses across frames are illustrated with blue grippers.

† Cewu Lu is the corresponding author, a member of Qing Yuan Research Institute and MoE Key Lab of Artificial Intelligence, AI Institute, Shanghai Jiao Tong University, Shanghai, China

A naive idea to solve this task is to generate reference grasp poses for the initial scene and consecutively track the object's 6D pose. As the object moves, the initial grasp pose can be projected to a new coming frame based on the object's 6D pose. Although such an idea seems to be natural and valid, some bottleneck greatly degrades its viability. First of all, the solution to reactive grasping should be able to handle objects' motion in real-time, meaning that it requires fast inference speed and immediate response to continuous environmental changes. However, 6D pose tracking may be time-consuming due to commonly-used instance segmentation [11,40]. Second, 6D pose tracking usually requires objects' prior knowledge, such as CAD models [4,41], which is not always available in the real world as well or achieves only category-level generalization [37].

Different from tracking objects, we propose to track grasps by a two-stage policy instead. We also comply with the restriction that no prior knowledge of the objects is allowed. Given a target grasp on a partial-view point cloud, we first discover its corresponding grasp among future frame's detected grasp poses as coarse estimation. These gasp poses can be given by an off-the-shelf grasp detector. Inspired by recent progress in local feature matching, which often uses image descriptors like SIFT [17] to describe interesting regions of images, we view grasp poses and their corresponding features as geometric descriptors on a partial-view point cloud. Based on such an assumption, we can simply estimate correspondences between two grasp sets from two different observation frames by matching the associated grasp features. Note that in opposition to classical local feature matching, features of the entire scene are also incorporated to help achieve global awareness. Furthermore, consecutive matching along an observation sequence may lead to the accumulation of error, on top of the coarse estimation through correspondence matching, we further use a memory-augmented coarse-to-fine module which uses both target grasp features and historical grasp features to refine the grasp tracking results for better temporal smoothness and semantic consistency.

We conduct extensive experiments on two benchmarks to evaluate our method and demonstrate its effectiveness. The results show that our method outperforms two representative baseline methods. We also conduct real robot experiments on both single-object scenes and cluttered scenes. We report success rates of 81.25% for single-object scenes and 81.67% for multi-object scenes.

2. Related Works

2.1. Grasping in Static Scenes

Thanks to the advances in the field of 3D perception [2, 25, 26], 6-DoF static grasping which takes point cloud as input is gaining increasing attention in both the research community and industry. In general, there are two lines of work that have been explored. The first line adopts a sampling-evaluation manner [15,18,23,34]. Grasp candidates are sampled or generated on the point clouds and then their qualities are evaluated by neural networks. The second line processes the point cloud of the entire scene and predicts grasp poses across the scene [9,10,24,27,36] in an end-to-end manner. Compared to the first line, the end-to-end strategy achieves better balance in terms of the speed-accuracy tradeoff. Some researchers also propose to predict actions in continuous space [14,32,38]. Nevertheless, these methods can be adversely impacted by distribution shifts. Our method uses a pre-trained grasp pose detector from [9] which belongs to the second research line.

2.2. Reactive Grasping

Though reactive grasping remains much less explored, a small amount of literature has partially investigated this more difficult problem. Most of these papers rely on object motion tracking. [1] predicts object motion and adaptively plans the grasp for a reachable grasp candidate. However, the object's motion is limited to few prototypes and thus it cannot handle the case where the objects move unpredictably. [19] tracks the objects' poses, yet the grasp planning phase is limited by a set of fixed grasp trajectories. [12,20,28] requires prior knowledge of the objects which is not always available in the real-world.

Some attempts on reactive grasping without prior information have been conducted recently. [22] picks the nearest grasp poses across frames. [42] samples grasp candidates by adding disturbances to grasp poses from the previous frame and evaluates their qualities in the current frame's scene. These methods focused on the smoothness of the grasp pose sequence but did not impose semantic constraints. Their predicted grasp poses can switch between different objects. Thus they mainly demonstrate single object grasping, and [42] requires an extra human hand segmentation module to avoid grasping the human's hand during handover while our correspondence matching module mitigates such a problem since grasps on hands and grasps on objects have low correspondence scores. [8] proposed a temporal association module to alleviate this dilemma. However, it only considers correspondence across two frames which is prone to error accumulation. In this paper, we further adopt a refinement module to adjust the grasp pose according to the target and historical grasp poses.

Some other works adopt continuous action prediction [14, 32]. Even if it is not designed for dynamic environments, it can be executed on moving objects thanks to the closed-loop controller. These methods may suffer greatly from distribution shifts. Different from these methods, our method tracks grasp poses explicitly.

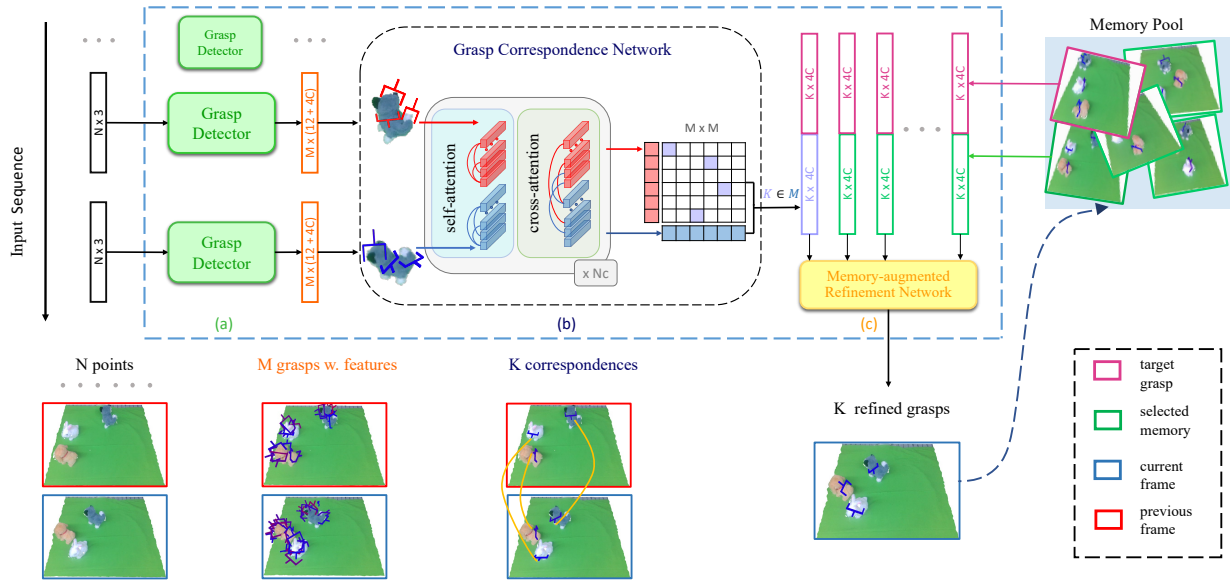

Figure 2. An overview of the proposed method for target-referenced reactive grasping. Given a target grasp and colored point clouds of a new coming frame with N points, our pipeline first detects M grasp candidates of size $M \times 12$ (3 for translation and 9 for rotation) across the scene. It then extracts grasp features of size $M \times 4C$ for each detected grasp by cropping and embedding the points along the grasping direction. Furthermore, grasp features aggregation is used to estimate corresponding grasps between the previous frame and the current frame. After filtering, the correspondence score matrix produces K valid pairs of grasp. The predicted corresponding grasps are then fed into a refinement network along with the target grasp and other frames selected from a memory pool to be updated toward the target. Contents in the memory pool are updated by the refined values as well.

2.3. Local Feature Matching in Computer Vision

Unlike global features which summarize the entire scene, local features distribute densely across the scene and describe interesting regions. Under slightly different camera views, local feature matching is aimed at recognizing distinctive regions and establishing associations in images or point clouds. The mainstream of local feature matching includes detector-based matching and detector-free matching. For detector-based matching, classical hand-crafted features such as SIFT [17] and ORB [29] are widely adopted. With the development of deep learning, learned local features [5, 6] achieve satisfying results as well. Different from detector-based methods that extract sparse local features, detector-free methods establish pixel-wise or point-wise dense features [3, 13, 16, 31, 33]. With the extracted local features, nearest neighbor search or learning-based approach [30, 39] are often used in the matching phase. In our task, we consider each grasp brings rich geometric and visual patterns of the grasped local patch and our grasp correspondence are built upon these local features.

3. Problem Formulation

In this section, we first briefly introduce some notations and metrics for grasp pose and grasp distance in 3D space, followed by formulating the problem of target-referenced reactive grasping.

We define a grasp pose $\mathcal{G}$ as

$$\mathcal{G} = (\mathbf{R}\ \mathbf{t}), \tag{1}$$

where $\mathbf{R} \in \mathbb{R}^{3\times3}$ denotes the rotation of the grasp pose and $\mathbf{t} \in \mathbb{R}^{3\times1}$ denotes the translation. Consider two arbitrary grasps $\mathcal{G}_1$ and $\mathcal{G}_2$, a grasp distance measure is defined as

$$\begin{aligned} &\mathcal{D}(\mathcal{G}_1, \mathcal{G}_2) = \Delta\mathbf{R} + \Delta\mathbf{t}, \\ &\Delta\mathbf{R} = \arccos \frac{\mathrm{trace}(\mathbf{R}_1^\top \mathbf{R}_2) - 1}{2}, \\ &\Delta\mathbf{t} = ||\mathbf{t}_1 - \mathbf{t}_2||. \end{aligned} \tag{2}$$

Then we make a formal statement of target-referenced reactive grasping based on the background definition above. Given $\mathcal{G}^1$ at first frame as reference grasp, for any timestep $j > 1$, target-referenced reactive grasping is aimed at finding the grasp $\mathcal{G}^j$ that minimizes the grasp distance:

$$\mathcal{G}^{j*} = \arg\min_{\mathcal{G}^j} \mathcal{D}(\mathcal{G}^j, \mathbf{T}^{j1}\mathcal{G}^1), \tag{3}$$

where $\mathbf{T}^{j1} \in \mathbb{R}^{4\times4}$ means the ground-truth transformation matrix of the reference grasp from the initial frame to the frame at timestep j in the camera's coordinate system.

4. Method

In this section, we detail our grasp tracking pipeline, as illustrated in Fig.2. For a target grasp, we first detect grasp

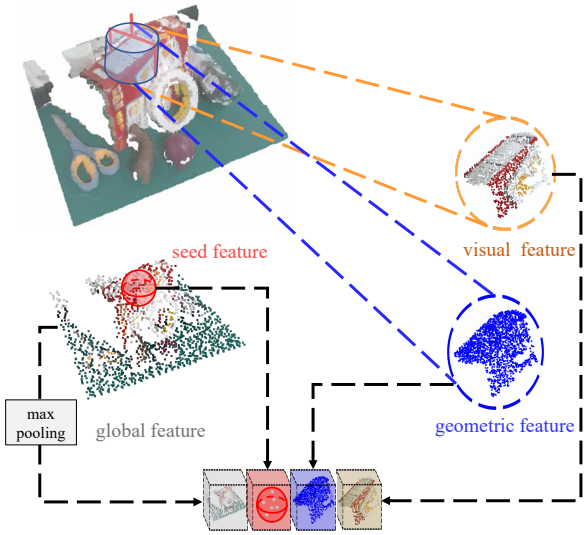

Figure 3. Visualization of grasp features representation.

candidates across the scene. Then we discover strong correspondences between the target grasp and grasp candidates by attentional grasp features aggregation. We finally use a memory-augmented refinement network to correct the predictions towards the target grasp.

4.1. Grasp Detector

For each incoming new frame, we generate a grasp set along with it. This work uses an off-the-shelf GraspNet baseline [9] for proposals of grasp candidates. GraspNet is a learned grasp pose detector that takes the partial point cloud of the scene $\mathcal{P}$ as input and outputs a grasp set $\mathbf{G}$ across the entire scene. In GraspNet, a point cloud $\mathcal{P}$ of size $N \times 3$ is downsampled and transformed to feature vectors of size $M \times C$ by the PointNet++ backbone network [26] where C denotes the channel dimension of the features and M is the number of points after farthest point sampling [7]. For each sampled point, GraspNet predicts a grasp $\mathcal{G}$ and its confidence score. The detected grasp set is then sent to the following steps for grasp tracking. While GraspNet performs well in this work, the following steps are not limited to any specific grasp detector.

4.2. Grasp Correspondence Network

We follow the formulation of grasp correspondence in [8]. A many-to-many correspondence matrix is learned during training, given two grasp sets in consecutive frames. During testing, we can choose the grasp pose with the highest correspondence score with the target pose in the previous frame as the tracked pose.

Grasp Features Representation First we illustrate how we represent grasp features. For M grasp poses in a scene, we follow [8] to extract seed features, geometric features and appearance features for them, each having a shape of $M \times C$. In addition, since it is quite common in the real world that multiple similar objects exist simultaneously in the scene, grasps on similar objects cannot be distinguished in such situations which results in correspondence switching between objects. Hence we further add global scene features of size $M \times C$ to provide information about the scene state. This global feature is extracted by max-pooling pointwise features from the backbone output, thus is the same for each of M grasp poses. Finally, we concatenate geometric features, visual features, seed point features and global scene features to construct grasp features of dimension $4C$ for each grasp, denoted by $\mathbf{x}$. We show the above process in Fig.3. We present more details of grasp features representation in the supplementary material.

Positional Information To provide positional information for grasps, we encode grasp poses by MLP layers of size (12, $2C$, $4C$). These feature vectors are then added to the raw grasp features $\mathbf{x}$, such that grasps can be treated in a similar way to words in a sentence.

$$\mathbf{x} \leftarrow \mathbf{x} + \mathbf{MLP}^{(1)}(p), \tag{4}$$

where p is the grasp pose of dimension 12, consisting of 9 parameters for the rotation matrix and 3 for translation. We use this feature as the final grasp features representation. Such a process makes our method position-dependent and benefits following the grasp features aggregation step.

Attentional Grasp Features Aggregation Thanks to the above-mentioned grasp features, it is now straightforward to reason about the visual, geometric and positional properties of grasps jointly. For a grasp $\mathcal{G}_i$, we first embed its associated grasp features using MLP layers and transform its feature size from $4C$ to $2C$ by MLP layers.

$$\mathbf{f}^{g_i} = \mathbf{MLP}^{(2)}(\mathbf{x}_i), \tag{5}$$

Apart from the features of a grasp itself, given a pair of grasp sets, it is intuitive that interacting with contextual grasps within grasp sets or across grasp sets is critical to reduce ambiguities and increase the distinctiveness of grasps as well. [8] adopts vanilla MLP blocks which cannot model long-range relationships among grasp poses. In this paper, we take advantage of the global receptive field of the attentional graph neural network (GNN) to model such dependencies. Following [30], self-attention and cross-attention [35] are adopted for context aggregation within grasp sets and across grasp sets. In particular, features are aggregated unidirectionally in self-attention and bidirectionally in cross-attention. Given two grasp sets $\mathbf{G}_1$ and $\mathbf{G}_2$

that each consists of M grasps, grasp features aggregation generates matching descriptors $\mathbf{f}^{m_1}, \mathbf{f}^{m_2} \in \mathbb{R}^{M \times 2C}$.

Here we outline the workflow of grasp features aggregation. In self-attention, we only process grasps within $\mathbf{G}_1$ or $\mathbf{G}_2$. Taking $\mathbf{G}_1$ as an example, grasp features $\mathbf{f}^{\mathbf{G}_1}$ of size $(M \times 2C)$ are first mapped to three elements which are named query, key and value in convention by learnable weights $\mathbf{W_Q}$, $\mathbf{W_K}$ and $\mathbf{W_V} \in \mathbb{R}^{2C \times 2C}$. The attention computation can then be written as $\mathbf{A} = (\mathbf{Q} \cdot \mathbf{K}^T)/\sqrt{C}$ where $\mathbf{A}$ denotes the attention matrix and has a size of $(M \times M)$ and $(\cdot)$ is the dot product operator. $\mathbf{A}$ actually measures the similarity between query and key. After that, the message propagation step can be defined as $\mathbf{M} = \text{softmax}(\mathbf{A}) \cdot \mathbf{V}$. $\mathbf{M}$ is later fed into MLP layers of size $(2C, 4C, 2C)$ to produce the output features which have the same size as $\mathbf{f}^{\mathbf{G}_1}$. Since $\mathbf{M}$ can be regarded as the sum of value vectors weighted by the similarity measure, therefore the output feature vectors represent correspondences among grasps in $\mathbf{G}_1$. In cross-attention, we consider grasp features from both $\mathbf{G}_1$ and $\mathbf{G}_2$. To produce the output feature vector, the only thing different is that key and value are no longer generated from $\mathbf{f}^{\mathbf{G}_1}$ but from $\mathbf{f}^{\mathbf{G}_2}$. Intuitively, in self-attention information flows only within $\mathbf{G}_1$ or $\mathbf{G}_2$ whereas in cross-attention information flows from $\mathbf{G}_2$ to $\mathbf{G}_1$ or vice versa. We consider the output of cross-attention layers as a representation of correspondence between $\mathbf{G}_1$ and $\mathbf{G}_2$. We interleave self-attention and cross-attention layers by N_c times and get the matching descriptors $\mathbf{f}^{m_1}, \mathbf{f}^{m_2}$ which are later used to compute correspondences within and across grasp sets.

Correspondence Estimation Given $\mathbf{f}^{m_1}, \mathbf{f}^{m_2}$, we can simply compute the correspondence matrix $\mathcal{S} \in \mathbb{R}^{M \times M}$. For each pair of grasps from these two sets, the correspondence score can be calculated by cosine similarity

$$\mathcal{S}(i \cdot j) = \frac{\mathbf{f}_i^{m_1} \cdot \mathbf{f}_j^{m_2}}{||\mathbf{f}_i^{m_1}||_2 \cdot ||\mathbf{f}_j^{m_2}||_2}. \tag{6}$$

Since cosine similarity is naturally normalized, it makes training more stable. Taking advantage of contrastive learning, during training a target grasp can be assigned the same class to any number of grasps in the other grasp set. This is important because some grasps are close in terms of translation and some are close in terms of rotation. During training, our method allows parallel prediction of multiple target grasps. We simply pick grasps that have the highest correspondence scores with respect to target grasps. Before entering the next stage, a grasp set is filtered to K grasps by removing upward grasps and grasps with low correspondence scores with respect to all other grasps. If the number of grasps is less than K, we repeat the first grasp. In real robot experiments, only one target grasp is used.

4.3. Memory-augmented Refinement Network

Since the correspondence is estimated between fixed grasp sets, the tracking performance greatly relies on the grasp detector which is not guaranteed to generate identical grasps across different frames. Also, consecutive correspondence matching may lead to an accumulation of errors. Therefore, to further improve the semantic consistency with the reference grasp pose, a refinement network is used to correct predictions made by correspondences towards the target grasp.

To be specific, we store the tracked grasp poses on past frames and their associated features in a memory pool. For each forward pass, the refinement network uses historical observations selected from the memory pool to refine the parameters of the selected grasp pose in the current frame. Here the historical frames from the memory pool are selected according to their time indexes. In practice, we select the reference pose in the first frame and apply uniform sampling for simplicity in the interval $(1, t-1]$ to select past frames, where t denotes the current time index.

For a L frame grasping sequence that consists of 1 frame of tracking target, 1 current frame, and $L-2$ frames selected from the memory pool, grasp features of size $L \times (4C)$ are first transformed to $L \times 2C$ by MLP layers. After that, we simply repeat the target frame and concatenate it with other frames, resulting in features of size $(L-1) \times (2C)$. Such that all the selected frames are conditioned on the tracking target. We then apply temporal encoding which is similar to positional encoding introduced in [35]. These features are then fed into a refinement network which outputs $(L-1)$ refined poses. Note that both historical grasp poses and the grasp pose in the current frame are updated. This refinement network has the same architecture as the grasp correspondence network, except that only self-attention layers are adopted. Moreover, the rotation is parameterized by the 6D representation introduced in [43]. Causal mask is used to avoid causality confusion as well.

4.4. Supervision

The objective function used to train the above pipeline can be divided into two parts:

$$\mathcal{L} = \mathcal{L}_c + \beta \mathcal{L}_{R,t}, \tag{7}$$

where $\mathcal{L}_c$ denotes the loss for correspondence learning and $\mathcal{L}_{R,t}$ denotes the loss for grasp pose refinement. β is the weighting term which we set as 1 during training.

For $\mathcal{L}_c$, we follow [8] and adopt supervised contrastive loss. For any two grasp poses with positive correspondence label, we denote them as $\text{corr}(\mathcal{G}_i, \mathcal{G}_k) = 1$ or vice versa. Given two arbitrary grasp sets $\mathbf{G}_1$ and $\mathbf{G}_2$, we first locate the positive grasp set $\mathbf{P}(i) = \{\mathcal{G}_k^2 \in \mathbf{G}_2 | \text{corr}(\mathcal{G}_i, \mathcal{G}_k) = 1\}$ for any $\mathcal{G}_i \in \mathbf{G}_1$. This correspondence loss can then be

written as:

$$\mathcal{L}_c = \sum_{\mathcal{G}_i^1 \in \mathbf{G}_1} \frac{-1}{|\mathbf{P}(i)|} \sum_{\mathcal{G}_k^2 \in P(i)} \log \frac{\exp(\mathcal{S}(i \cdot k)/\tau)}{\sum_{\mathcal{G}_j^2 \in \mathbf{G}_2} \exp(\mathcal{S}(i \cdot j)/\tau)}, \tag{8}$$

where $|\mathbf{P}(i)|$ denotes the cardinality and τ is a scalar temperature parameter. This loss pulls together the positive grasp pairs in embedding space, whereas pushes apart negative grasp pairs meanwhile.

For grasp pose refinement, we adopt a transformation loss which consists two terms for optimizing rotation and translation respectively. Consider $\mathcal{J} = \{j_1, j_2, \cdots, j_n\}$ that stores arbitrary number of time index, for any time step $j \in \mathcal{J}$, given target grasp $\mathcal{G}^1$, we first compute its transformed grasp in j^{th} frame's coordinates, denoted by

$$\mathcal{G}^{j1} = (\mathbf{R}^{j1}\ \mathbf{t}^{j1}), \tag{9}$$

We then define the transformation loss as:

$$\begin{aligned} \mathcal{L}_{R,t} = & \frac{1}{|\mathcal{J}|} \sum_{j \in \mathcal{J}} \frac{1}{|\mathbf{G}_j|} \sum_{\mathcal{G}_i^j \in \mathbf{G}_j} ||(\mathbf{R}^*)^{-1}\mathbf{R}^{j1} - \mathbf{I}||_2 \\ & + \frac{1}{|\mathcal{J}|} \sum_{j \in \mathcal{J}} \frac{1}{|\mathbf{G}_j|} \sum_{\mathcal{G}_i^j \in \mathbf{G}_j} ||\mathbf{t}^{j1} - \mathbf{t}^*||_2, \end{aligned} \tag{10}$$

where $\mathbf{G}_j$ denotes grasp set at time step j.

5. Experiments

5.1. Dataset

The proposed pipeline is trained on GraspNet-1Billion dataset [9] which is a large-scale real-world dataset for grasp detection. It includes 89 objects, 190 scenes and 256 camera views for each scene. We follow [8] to generate grasp correspondence labels across viewpoints on the GraspNet-1Billion dataset and refer readers to [8] for more details. In addition, we further collect 30 scenes of moving objects for testing. For each frame, we record the RGBD images and objects' 6D poses. We manually annotate 10 grasps for the first frame of each scene as targets and further project the annotated grasps according to the 6D poses for the future frames. This test set is named Moving GraspNet.

5.2. Pre-processing and Augmentation

Before being fed into the network, the point cloud is first downsampled to 20000 points and the RGB values are normalized. We also remove points outside a pre-defined workspace. Furthermore, we adopt massive data augmentation to avoid over-fitting and enrich objects' motion. For each object in the scene, its point cloud is augmented on-the-fly by a random translation of $[-0.2, 0.2]$ meter in all XYZ directions and a random rotation of $[-30, 30]$ degrees around z-axis. Moreover the scene's point cloud is rotated by $[-30, 30]$ degrees around both x-axis and z-axis. Due to the space limitations, please refer to the supplementary materials for details about network implementation.

5.3. Performance of Grasp Tracking

Evaluation Metric We propose a new Multiple Grasp Pose Tracking (**MGTA**) metric which follows the idea of Multiple Object Tracking Accuracy (**MOTA**) [21] in multi-object tracking task, except that the similarity measure between two grasps is computed using Eq.(2). Details are given in supplementary materials. Apart from **MGTA**, we also report the mean translation error and rotation error to the tracking target of each sequence.

We follow the same evaluation procedure on GraspNet1-Billion and Moving GraspNet. For each scene, we pick 10 reference grasps in the first frame which are later used for grasp tracking. To be specific, these reference grasps are detected by a grasp detector on GraspNet1-billion but manually labeled on Moving GraspNet. For each remaining frame, we compute the ground truth grasps using objects' 6D poses and segmentation mask of the scene. The metrics are calculated using the transformed reference grasps and predictions associated with each frame.

Comparing with Representative Methods We compare our methods with two other baseline methods. A heuristic method is implemented as the first baseline. Given a grasp pose from the previous frame, its nearest neighbor detected in the current camera coordinates is picked. For the second method, we adopt Bundle Track [40] which is one representative method in the field of unseen object 6D pose tracking. It fits our task setting well since it requires no objects' CAD model.

As it is shown in Tab.1 and Tab.2, our method outperforms baseline methods on both GraspNet-1Billion and Moving GraspNet. It is found in experiments that, in many cases, grasps cannot fall on the same objects if we only pick the nearest neighbors. Also, it can be drawn from the results that tracking results are improved significantly after refinement, especially for rotation. Note that **MGTA** will be negative if there is a large number of false positives, false negatives, and ID switches.

5.4. Real Robot Experiments

In real robot experiments, we use Flexiv Rizon robot arm with an Intel Realsense L515 depth camera mounted on the end-effector. A Robotiq-85 parallel-jaw gripper is used with 3D-printed extended fingertips attached to it. The real robot experiments run on an NVIDIA 3090 GPU at around 10 fps. Please refer to the supplementary materials for details of the objects.

Method	Seen			Similar			Novel		
	MGTA↑	n (cm)↓	n (°)↓	MGTA↑	n (cm)↓	n (°)	MGTA↑	n (cm)↓	n (°)↓
Nearest	-0.81	15.2	95.98	-0.80	15.4	97.61	-0.83	16.6	96.90
Bundle Track [40]	0.74	1.52	15.80	0.72	1.47	16.37	0.72	1.93	15.41
Ours, w/o refinement	-0.32	2.31	53.24	-0.30	2.68	57.59	-0.39	2.93	58.49
Ours, M=512	0.80	1.58	10.89	0.80	1.74	10.62	0.75	2.10	15.03
Ours, M=1024	0.84	1.34	10.15	0.85	1.52	9.84	0.78	1.90	12.96
Ours, M=2048	0.85	1.27	9.82	0.86	1.35	9.50	0.79	1.84	11.48

Table 1. GraspNet-1Billion evaluation results on seen, similar and novel objects respectively. M denotes the number of grasp detected, its default value is 1024.

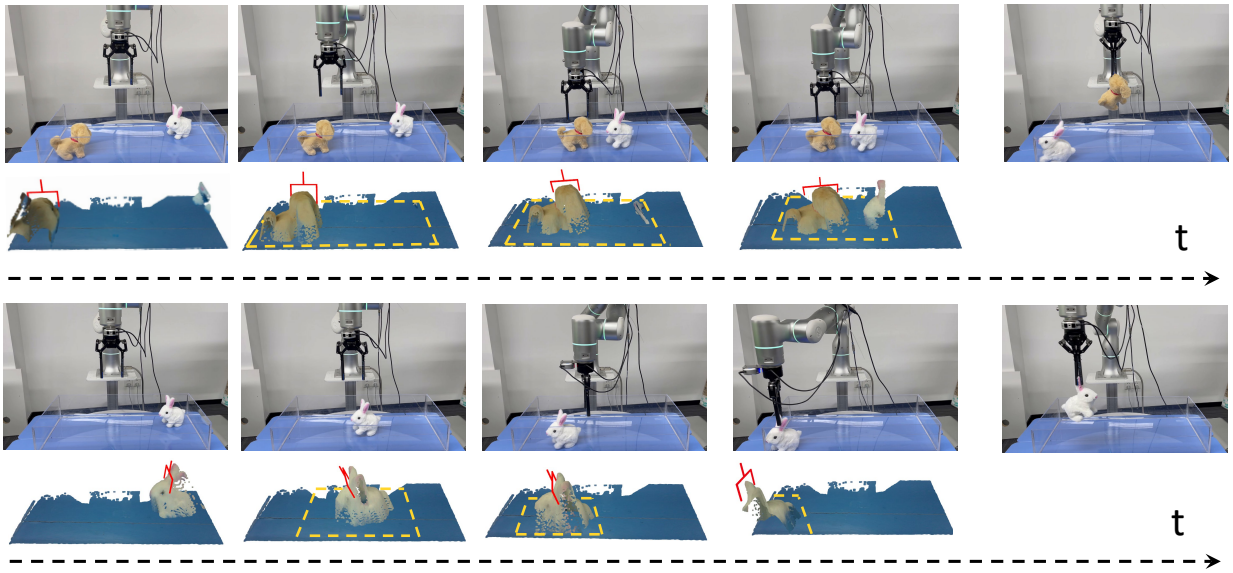

Figure 4. Illustration of tracking a toy dog and toy rabbit. We can see that the grasp poses are consistent, *i.e.,* on the head of the dog and the ear of the rabbit. Yellow bounding box represents to the view limit of the camera. No box means the camera can see the entire scene.

Method	MGTA↑	n (cm)↓	n (°)↓
Nearest	-0.47	8.09	67
Bundle Track	0.37	7.60	23.52
Ours, w/o refinement	-0.43	6.24	62.33
Ours, M=512	0.47	5.53	22.63
Ours, M=1024	0.52	4.60	20.37
Ours, M=2048	0.54	4.14	18.28

Table 2. Moving GraspNet evaluation results. All objects in Moving GraspNet are novel. M denotes the number of grasp detected, its default value is 1024.

Object ID	#Attempt	#Success	Success Rate
1	10	8	0.8
2	10	9	0.9
3	10	8	0.8
4	10	7	0.7
6	10	8	0.8
7	10	8	0.8
9	10	8	0.8
10	10	9	0.9
Total	80	65	0.8125

Table 3. Real robot experiments results on single object scenes.

Protocol For each frame, we receive RGBD images from the in-hand camera. We transform the point cloud to the camera coordinate system at ready state by the camera extrinsic for simplicity as well. Due to safety considerations, points outside a pre-defined workspace will be masked. After a grasp candidate generation phase, we select the grasp with the highest predicted score as a reference grasp and keep tracking it for each new frame. During the tracking phase, the robot moves to a pre-grasp pose which is 0.02 meters higher than the actual grasp pose. The robot executes the grasp once three conditions are met simultaneously: 1) the translation distance between the gripper center and the target grasp in XY-plane is lower than 0.01 meter (the XY-plane is parallel to the table); 2) the translation distance between the gripper center and the target grasp in 3D space is lower than 0.04 meters; 3) the rotation distance between the gripper and the target grasp in 3D space is lower than

Object ID	#Attempt	#Success	Success Rate
7,7,8,8	10	8	0.8
10,13,14	10	9	0.9
1,11,15,16,17,18	10	8	0.8
2,3,5	10	9	0.9
9,12	10	7	0.7
20,20	10	8	0.8
Total	60	49	0.8167

Table 4. Real robot experiments results on multi-object scenes.

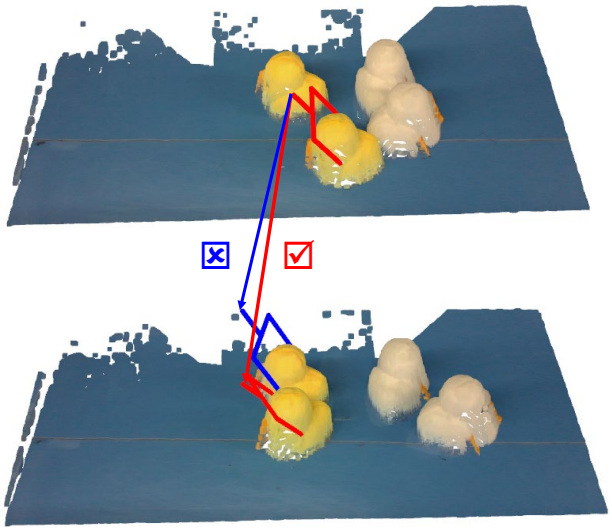

Figure 5. Correspondence switching between similar objects. Red arrow indicates correct correspondence and blue arrow indicates wrong correspondence.

15 degrees. The initial positions of the objects are chosen randomly.

Results As shown in Tab.3 and Tab.4, we execute 20 grasp attempts for each scene. In single-object scenes, we randomly sample 8 objects for testing and we achieve a success rate of 81 percent. In multi-object scenes, we only choose objects with similar visual and geometric patterns to test the temporal smoothness and semantic consistency of the predicted grasps. Results suggest that single-object scenes and multi-object scenes have nearly the same success rate which further proves the effectiveness of our method. We illustrate one case from real robot experiments in Fig.4.

5.5. Analysis

Effect of Global Features In Sec.4.2, we incorporate global features to represent grasp features. While local grasp features are sufficient to identify corresponding local structures, grasp trials may fail due to correspondence switches among objects with similar visual and geometric patterns as we can see in Fig.5. Such phenomenon may be due to lacking awareness of the scene state and we propose to solve it by global features as what we state in Sec.4.2. In order to analyze the actual effectiveness of global features, we further record how frequent correspondence switching happens with or without global features. We conduct these experiments on identical objects which are placed close to each other in the scenes. Tab.5 suggests that the number of correspondence switches decreased significantly with global features.

Object ID	Method	#Attempt	#SW
7,7,8,8	w/o global features	10	5
	Full model	10	1
1,17,17	w/o global features	10	4
	Full model	10	0

Table 5. Statistics of Correspondence Switching. SW represents correspondence switching.

Failure Analysis There are three major types of failure cases found in real robot experiments. First, some predicted grasp candidates from the detector do not have good qualities. Naturally, tracking such targets may lead to failed grasps. Second, it is hard to heuristically plan the timing to close the gripper. Due to some grasps may fall on the edges of objects, it is common that the objects may slip away immediately when the robot receives the command to close the gripper. Furthermore, since we mount the camera on the wrist, the point cloud may become fragmented due to view limitation and occlusion, especially when the objects fall behind the gripper. In such cases, the predictions can be inaccurate.

6. Conclusion

We present a novel solution to reactive grasping in a target-referenced setting. For a reference grasp, our method predicts temporally smooth and semantically consistent grasp poses in the future frames without any prior knowledge of instance. Experiments conducted on both single-object and multi-object scenes show that our method provides reliable grasp plans under various environments.

Acknowledgement This work was supported by the National Key R&D Program of China (No.2021ZD0110700), Shanghai Municipal Science and Technology Major Project (2021SHZDZX0102), Shanghai Qi Zhi Institute, and Shanghai Science and Technology Commission (21511101200).
Contributions: J. Liu and H-S. Fang initiated the project. J. Liu designed and trained the network and conducted the evaluation together with R. Zhang. J. Liu wrote the paper. H-S. Fang devised and mentored this project and edited the paper. M. Gou, H. Fang, and S. Xu constructed the testing dataset. C. Wang and H. Yan helped with the evaluation. C. Lu supervised the project and provided hardware support.

References

[1] Iretiayo Akinola, Jingxi Xu, Shuran Song, and Peter K. Allen. Dynamic grasping with reachability and motion awareness. *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 9422–9429, 2021. 2

[2] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4d spatio-temporal convnets: Minkowski convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3075–3084, 2019. 2

[3] Christopher B Choy, JunYoung Gwak, Silvio Savarese, and Manmohan Chandraker. Universal correspondence network. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. 3

[4] Xinke Deng, Arsalan Mousavian, Yu Xiang, Fei Xia, Timothy Bretl, and Dieter Fox. Poserbpf: A rao–blackwellized particle filter for 6-d object pose tracking. *IEEE Transactions on Robotics*, 37(5):1328–1342, 2021. 2

[5] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Toward geometric deep slam. *ArXiv*, abs/1707.07410, 2017. 3

[6] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 337–33712, 2018. 3

[7] Y. Eldar, M. Lindenbaum, M. Porat, and Y.Y. Zeevi. The farthest point strategy for progressive image sampling. *IEEE Transactions on Image Processing*, 6(9):1305–1315, 1997. 4

[8] Hao-Shu Fang, Chenxi Wang, Hongjie Fang, Minghao Gou, Jirong Liu, Hengxu Yan, Wenhai Liu, Yichen Xie, and Cewu Lu. Anygrasp: Robust and efficient grasp perception in spatial and temporal domains. *arXiv preprint arXiv:2212.08333*, 2022. 2, 4, 5, 6

[9] Hao-Shu Fang, Chenxi Wang, Minghao Gou, and Cewu Lu. Graspnet-1billion: A large-scale benchmark for general object grasping. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11444–11453, 2020. 2, 4, 6

[10] Minghao Gou, Hao-Shu Fang, Zhanda Zhu, Sheng Xu, Chenxi Wang, and Cewu Lu. Rgb matters: Learning 7-dof grasp poses on monocular rgbd images. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 13459–13466. IEEE, 2021. 2

[11] Yinlin Hu, Joachim Hugonot, Pascal Fua, and Mathieu Salzmann. Segmentation-driven 6d object pose estimation. In *CVPR*, 2019. 2

[12] Seungsu Kim, Ashwini Shukla, and Aude Billard. Catching objects in flight. *IEEE Transactions on Robotics*, 30(5):1049–1065, 2014. 2

[13] Donghoon Lee, Onur C. Hamsici, Steven Feng, Prachee Sharma, and Thorsten Gernoth. Deeppro: Deep partial point cloud registration of objects. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5663–5672, 2021. 3

[14] Sergey Levine, Peter Pastor, Alex Krizhevsky, Julian Ibarz, and Deirdre Quillen. Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection. *The International journal of robotics research*, 37(4-5):421–436, 2018. 2

[15] Hongzhuo Liang, Xiaojian Ma, Shuang Li, Michael Görner, Song Tang, Bin Fang, Fuchun Sun, and Jianwei Zhang. Pointnetgpd: Detecting grasp configurations from point sets. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 3629–3635. IEEE, 2019. 2

[16] Ce Liu, Jenny Yuen, and Antonio Torralba. Sift flow: Dense correspondence across scenes and its applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(5):978–994, 2011. 3

[17] David Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60:91–, 11 2004. 2, 3

[18] Jeffrey Mahler, Matthew Matl, Vishal Satish, Michael Danielczuk, Bill DeRose, Stephen McKinley, and Ken Goldberg. Learning ambidextrous robot grasping policies. *Science Robotics*, 4(26), 2019. 2

[19] Naresh Marturi, Marek Kopicki, Alireza Rastegarpanah, Vijaykumar Rajasekaran, Maxime Adjigble, Rustam Stolkin, Aleš Leonardis, and Yasemin Bekiroglu. Dynamic grasp and trajectory planning for moving objects. *Autonomous Robots*, 43(5):1241–1256, 2019. 2

[20] Arjun Menon, Benjamin Cohen, and Maxim Likhachev. Motion planning for smooth pickup of moving objects. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 453–460. IEEE, 2014. 2

[21] Anton Milan, Laura Leal-Taixé, Ian D. Reid, Stefan Roth, and Konrad Schindler. Mot16: A benchmark for multi-object tracking. *ArXiv*, abs/1603.00831, 2016. 6

[22] Douglas Morrison, Juxi Leitner, and Peter Corke. Closing the loop for robotic grasping: A real-time, generative grasp synthesis approach. In *Robotics: Science and Systems*, 2018. 1, 2

[23] Arsalan Mousavian, Clemens Eppner, and Dieter Fox. 6-dof graspnet: Variational grasp generation for object manipulation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2901–2910, 2019. 2

[24] Peiyuan Ni, Wenguang Zhang, Xiaoxiao Zhu, and Qixin Cao. Pointnet++ grasping: Learning an end-to-end spatial grasp generation algorithm from sparse point clouds. *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3619–3625, 2020. 2

[25] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas 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*, pages 652–660, 2017. 2

[26] Charles R Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017. 2, 4

[27] Yuzhe Qin, Rui Chen, Hao Zhu, Meng Song, Jing Xu, and Hao Su. S4g: Amodal single-view single-shot se (3) grasp

detection in cluttered scenes. In *Conference on robot learning*, pages 53–65. PMLR, 2020. 2

[28] Patrick Rosenberger, Akansel Cosgun, Rhys Newbury, Jun Kwan, Valerio Ortenzi, Peter Corke, and Manfred Grafinger. Object-independent human-to-robot handovers using real time robotic vision. *IEEE Robotics and Automation Letters*, 6(1):17–23, 2021. 2

[29] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: An efficient alternative to sift or surf. In *2011 International Conference on Computer Vision*, pages 2564–2571, 2011. 3

[30] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4938–4947, 2020. 3, 4

[31] Tanner Schmidt, Richard Newcombe, and Dieter Fox. Self-supervised visual descriptor learning for dense correspondence. *IEEE Robotics and Automation Letters*, 2(2):420–427, 2017. 3

[32] Shuran Song, Andy Zeng, Johnny Lee, and Thomas Funkhouser. Grasping in the wild: Learning 6dof closed-loop grasping from low-cost demonstrations. *IEEE Robotics and Automation Letters*, 5(3):4978–4985, 2020. 2

[33] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. LoFTR: Detector-free local feature matching with transformers. *CVPR*, 2021. 3

[34] Andreas ten Pas, Marcus Gualtieri, Kate Saenko, and Robert Platt. Grasp pose detection in point clouds. *The International Journal of Robotics Research (IJRR)*, 36(13-14):1455–1473, 2017. 2

[35] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. 4, 5

[36] Chenxi Wang, Hao-Shu Fang, Minghao Gou, Hongjie Fang, Jin Gao, and Cewu Lu. Graspness discovery in clutters for fast and accurate grasp detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15964–15973, 2021. 2

[37] Chen Wang, Roberto Martín-Martín, Danfei Xu, Jun Lv, Cewu Lu, Li Fei-Fei, Silvio Savarese, and Yuke Zhu. 6-pack: Category-level 6d pose tracker with anchor-based keypoints. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 10059–10066, 2020. 2

[38] Lirui Wang, Yu Xiang, Wei Yang, Arsalan Mousavian, and Dieter Fox. Goal-auxiliary actor-critic for 6d robotic grasping with point clouds. In *Conference on Robot Learning*, pages 70–80. PMLR, 2022. 2

[39] Yue Wang and Justin M. Solomon. Deep closest point: Learning representations for point cloud registration. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019. 3

[40] Bowen Wen and Kostas Bekris. Bundletrack: 6d pose tracking for novel objects without instance or category-level 3d models. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 8067–8074. IEEE, 2021. 2, 6, 7

[41] Bowen Wen, Chaitanya Mitash, Baozhang Ren, and Kostas E. Bekris. se(3)-tracknet: Data-driven 6d pose tracking by calibrating image residuals in synthetic domains. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 10367–10373, 2020. 2

[42] Wei Yang, Chris Paxton, Arsalan Mousavian, Yu-Wei Chao, Maya Cakmak, and Dieter Fox. Reactive human-to-robot handovers of arbitrary objects. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3118–3124. IEEE, 2021. 1, 2

[43] Yi Zhou, Connelly Barnes, Lu Jingwan, Yang Jimei, and Li Hao. On the continuity of rotation representations in neural networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 5