frontiers in Robotics and AI

ORIGINAL RESEARCH
published: 17 September 2020
doi: 10.3389/frobt.2020.00120

DGCM-Net: Dense Geometrical Correspondence Matching Network for Incremental Experience-Based Robotic Grasping

Timothy Patten†, Kiru Park† and Markus Vincze*

Vision for Robotics Laboratory, Automation and Control Institute, TU Wien, Vienna, Austria

This article presents a method for grasping novel objects by learning from experience. Successful attempts are remembered and then used to guide future grasps such that more reliable grasping is achieved over time. To transfer the learned experience to unseen objects, we introduce the dense geometric correspondence matching network (DGCM-Net). This applies metric learning to encode objects with similar geometry nearby in feature space. Retrieving relevant experience for an unseen object is thus a nearest neighbor search with the encoded feature maps. DGCM-Net also reconstructs 3D-3D correspondences using the view-dependent normalized object coordinate space to transform grasp configurations from retrieved samples to unseen objects. In comparison to baseline methods, our approach achieves an equivalent grasp success rate. However, the baselines are significantly improved when fusing the knowledge from experience with their grasp proposal strategy. Offline experiments with a grasping dataset highlight the capability to transfer grasps to new instances as well as to improve success rate over time from increasing experience. Lastly, by learning task-relevant grasps, our approach can prioritize grasp configurations that enable the functional use of objects.

Keywords: robotics, object grasping, incremental learning, dense correspondence matching, deep learning, metric learning, machine vision

OPEN ACCESS

Edited by:
Robert Krug,
Robert Bosch, Germany

Reviewed by:
M. Raheel Bhutta,
Sejong University, South Korea
Andras Kupcsik,
Bosch Center for Artificial Intelligence, Germany

***Correspondence:**
Timothy Patten
patten@acin.tuwien.ac.at

†*These authors have contributed equally to this work*

Specialty section:
This article was submitted to Robot and Machine Vision, a section of the journal Frontiers in Robotics and AI

Received: *18 December 2019*
Accepted: *31 July 2020*
Published: *17 September 2020*

Citation:
Patten T, Park K and Vincze M (2020) DGCM-Net: Dense Geometrical Correspondence Matching Network for Incremental Experience-Based Robotic Grasping. Front. Robot. AI 7:120. doi: 10.3389/frobt.2020.00120

1. INTRODUCTION

Grasping is an essential capability for robots in a large variety of fields, from warehouse operations to industrial assembly lines, applications in agriculture and many domestic service tasks. Grasping leads to the subsequent manipulation of objects, which is the most direct way for robots to interact with the world. Especially in human environments, where many man-made objects are designed to be handled by people, grasping with a robotic gripper or hand is necessary.

A popular approach for robot grasping is to exploit known objects (Klank et al., 2009; Srinivasa et al., 2010; Chitta et al., 2012a; Tremblay et al., 2018; Wang C. et al., 2019). However, this can only be applied to a given set of objects and thus does not generalize to new objects, which reduces the usability for real-world operation. It is possible to grasp unknown objects by learning classifiers, predictive or generative models (Saxena et al., 2008; Jiang et al., 2011; Fischinger et al., 2015; Lenz et al., 2015; Redmon and Angelova, 2015; Pinto and Gupta, 2016; Kumra and Kanan, 2017; Wang et al., 2017; Morrison et al., 2018) but large amounts of labeled data are required and this is time consuming when it is annotated by hand. The effort for annotating data is eased by restricting it to

2D grasp poses (i.e., assuming top-down grasps), but this limits the grasp configurations that are able to be applied. Training data is often generated without human annotation either offline using collections of object models (Mahler et al., 2016, 2017) or online by exploiting real-world robot trials. However, many hundreds or thousands of hours are required to generate a sufficient amount of data (Pinto and Gupta, 2016; Jang et al., 2017). Learning end-to-end strategies for grasping with reinforcement learning (Boularias et al., 2015; Kalashnikov et al., 2018; Levine et al., 2018; Zeng et al., 2018a) also suffers from substantial training time, with some work reporting training times in the order of months (Levine et al., 2018). While the burden of learning can be alleviated by leveraging physics-enabled simulation environments (e.g., James et al., 2017, 2019; Fang et al., 2018; Iqbal et al., 2019), this introduces the challenge of transferring from simulation to the real world.

An alternative approach is to transfer grasps for known objects to familiar objects. This makes the assumption that when a new object is similar to another object for which a grasp is known, then the new object is likely to be successfully grasped in a similar way (Bohg et al., 2014). Prior work on experience-based grasping build a database of sensory observations with associated grasp information such as a pose or contact points. The experience is accumulated by trial and error with a robot platform (Morales et al., 2004; Herzog et al., 2012; Detry and Piater, 2013), kinesthetically taught (Kroemer et al., 2012; Detry et al., 2013; Kopicki et al., 2016), or inferred by directly observing human behavior (Liu et al., 2019). Grasping an unseen object requires a strategy to map the current observation to the samples in the database and execute (or extrapolate from) the most similar experience. This is typically done using global shape (Morales et al., 2004; Bohg and Kragic, 2009; Kopicki et al., 2016), local descriptors (Liu et al., 2019), or object regions (Detry et al., 2012, 2013; Herzog et al., 2012; Kroemer et al., 2012; Detry and Piater, 2013). In contrast to end-to-end learning approaches, experience-based grasping has the potential to learn from very few exemplars. Only few methods have been demonstrated in an end-to-end pipeline with a robotic platform and currently they rely on hand-crafted features for retrieving similar experiences and for transferring grasps to new objects.

In this work we present a new method for incremental grasp learning from experience. The key to our approach is to apply dense geometrical correspondence matching. Familiar objects are identified through global geometric encoding and associated grasps are transferred through local correspondence matching. We introduce the dense geometrical correspondence matching network (DGCM-Net) that uses metric learning to encode the global geometry of objects in depth images such that similar geometries are represented nearby in feature space to allow accurate retrieval of experience. DGCM-Net additionally reconstructs dense geometrical correspondences between pairs of depth images using a variant of normalized object coordinate (NOC) values. These values are used to compute the rigid transformation between the local region around the grasp of a stored experience and the corresponding region on an object in a new scene. Precise 3D object models are not assumed, thus we define the view-dependent normalized object coordinates (VD-NOC) to extend NOC values to single views.

DGCM-Net is applied in an incremental grasp learning pipeline, in which a robot self-supervises grasp learning from its own experience. We show that a robot learns to repeatably grasp the same object after one or two successful experiences and also to grasp novel objects that have comparable geometry to a known experience. As an extension, we show that the predictions from DGCM-Net improve the performance of baseline grasping methods by combining their quality measures with our experience-based measure. The incremental learning pipeline is also flexible in that grasp success is not the only measure to constitute experience. Specific positions or configurations of grasps can be preferred and therefore used in future situations. In particular, semantic grasps, such as grasping the handle of a mug, are prioritized as they are more relevant for the subsequent manipulation of the object (Song et al., 2010; Dang and Allen, 2012; Antonova et al., 2018; Fang et al., 2018). As a result, task-oriented grasps are quickly learned, allowing a robot to perform meaningful actions with objects.

Studies with a dataset showcase the ability of the presented grasp prediction method to transfer between objects. In addition, our analysis confirms the intuition that the quality of grasp prediction improves with increasing experience. Real-world experiments with a mobile manipulator are performed to compare our grasping strategy against various other approaches. The experiments show that we achieve a comparable grasp success rate with the baselines and improve the baselines when integrating our predictions to achieve superior performance overall. Demonstrations of the full system show the continuous learning capability for completely novel objects from classes never before seen. Finally, the usability of our approach for semantic or task-oriented grasping is illustrated to grasp objects with handles.

In summary, this article makes the following contributions:

- The dense geometrical correspondence matching network to encode object geometry for nearest neighbor retrieval and to densely reconstruct 3D-3D correspondences in order to transfer grasps from stored experiences to unseen objects.
- An experience-based 6D grasp learning pipeline that incrementally grows a database of exemplars to guide grasp selection for the same object or novel unseen objects.
- Offline experiments with a new annotated dataset showing the capability of DGCM-Net to transfer grasps to unseen objects as well as to steadily improve over time with increasing accumulation of data.
- Online grasping experiments showing that the grasp success of our approach is competitive with common baselines and improves the baselines when combining their predictions with our experience-based grasp predictions.
- Demonstrations showing the extension of our method for semantic grasping by guiding grasp selection to the parts of objects that are relevant to the object's functional use.

The remainder of the paper is organized as follows. Section 2 discusses related work. In section 3, we present the dense

geometrical correspondence matching network and describe the incremental grasp learning pipeline. The results of the offline and robotic experiments are reported in sections 4, 5. Finally, section 6 concludes and discusses future work.

2. RELATED WORK

The significant amount of attention given to robotic grasping has resulted in a large number and high diversity of techniques. A common strategy uses known object instances, which are provided as CAD models or are captured by a modeling process (e.g., Prankl et al., 2015; Wang and Hauser, 2019). Given a known grasp configuration for an object in its local coordinate system, the task of grasping is simplified to estimating the pose of the object such that the grasp pose is transformed into the new scene. Traditional methods identify hand-crafted features to localize an object model within a scene (Klank et al., 2009; Srinivasa et al., 2010; Chitta et al., 2012a) but more recently advances for pose estimation have been made by the application of deep learning (Xiang et al., 2018; Li et al., 2019; Park et al., 2019b; Zakharov et al., 2019) and grasping pipelines achieve high success rate (Tremblay et al., 2018; Wang C. et al., 2019). The main limitation of this direction of research, however, is the closed-world assumption. The approach is restricted to only the objects for which a model is provided and thus cannot generalize to unknown objects.

To address the problem of grasping unknown objects, local geometry can serve as a strong cue. For example, fitting primitives and estimating grasps based on the geometrical structure of the primitives (Rusu et al., 2009) or fitting superquadratics and synthesizing grasp poses at the points of minimum curvature (Makhal et al., 2018) have been shown to work in certain cases. More often though, unknown object grasping is addressed by learning from data (Bohg et al., 2014). Along this line, methods predict the success of a proposed grasp by training a traditional classifier (Jiang et al., 2011; Fischinger et al., 2015) or deep neural network (Saxena et al., 2008; Lenz et al., 2015; Redmon and Angelova, 2015; Pinto and Gupta, 2016; Kumra and Kanan, 2017; Wang et al., 2017). Alternatively, grasp simulation or analytical grasp metrics are computed for objects in model databases to generate training data (Johns et al., 2016; Mahler et al., 2016, 2017; ten Pas et al., 2017; Cai et al., 2019; Liang et al., 2019; Mousavian et al., 2019). The task is then to learn a model that can predict the value of the grasp metric given a proposal and then select the grasp that is most likely to succeed. There is also work that avoids the sampling and scoring procedure by directly predicting a grasp pose with a quality measure (Morrison et al., 2018). The generative method has proven to be computationally superior and sufficiently fast to be integrated in a closed-loop system. While the work for unknown object grasping has made considerable achievements, they are limited by the diversity of the training data. Out of distribution objects may not receive accurate grasp quality predictions and may fail. Thus, it is necessary to continuously learn and add new examples to the training set. Unfortunately, the deep neural networks that are applied do not have the capacity to be updated online. Our approach, on the other hand, does not need to retrain for grasp prediction. By abstracting the learning component to correspondence matching, we simply add experience to a database and use the network to predict the closest matches for grasp transfer.

Another approach to grasping is to leverage real robot experience and learn end-to-end strategies. One direction is to employ reinforcement learning (Boularias et al., 2015; Kalashnikov et al., 2018; Levine et al., 2018; Zeng et al., 2018a). The advantage of an end-to-end approach is that complete grasping policies can be learned directly from visual input, which removes the need for a dedicated perception pipeline with an additional motion planner for execution. A disadvantage, however, is that the policies can only be applied to scenarios that are perceptually similar, and thus generalization to novel environments is limited. Unsupervised methods, such as Jang et al. (2018), better generalize to unseen scenarios and objects. They are more general to the task and less sensitive to the training scenes. These methods learn an embedding that can be used to retrieve manipulation policies for online execution. Despite these advances, the major drawback of both self- and unsupervised learning is that many attempts are needed for training. Our approach, in contrast, only needs a handful of experiences to reliably repeat past successes. Although physics simulation is now a popular alternative for training learning algorithms, the transfer from simulation to the real world requires additional attention (James et al., 2017, 2019; Fang et al., 2018; Iqbal et al., 2019).

Experience-based grasping is much more efficient than reinforcement learning methods since far fewer examples are needed to learn grasps. The common approach is to accumulate samples of past success or failure to guide the grasp selection in new scenarios, under the assumption that objects with similar shape (or appearance) can be grasped in a similar way. Some work define global shape descriptors and train a discriminative classifier to identify the similarity between object shapes to transfer grasps to familiar objects (Morales et al., 2004; Bohg and Kragic, 2009; Kopicki et al., 2016). Other work leverage local feature descriptors to identify the relevant local regions associated with contact points to transfer grasps between objects within the same class (Liu et al., 2019). Another approach is to analyze object regions and to maintain a library of prototypical grasps for recurring object parts. This is accomplished by measuring the similarity between regions on the surface of objects such as with height maps (Herzog et al., 2012) or by surface distributions or densities (Detry et al., 2012, 2013; Kroemer et al., 2012; Detry and Piater, 2013). A major assumption is that the observed parts are equivalent, which means grasp transfer is the application of a transformation from the prototype to the scene. They do not deal with the possibility of scale change or deformation. Such geometry variation would have to be stored as a new experience. Our approach deals with this challenge by aligning shapes through dense 3D-3D correspondences and thus also modifies the shape of the grasp to fit the new geometry. Another drawback of prior work is that they use hand-crafted features to encode shape information. We instead generate descriptive features through metric learning, which has been

shown to be powerful for similar geometric matching tasks (Zeng et al., 2017).

Our approach for grasping relies on first finding the nearest observation in a database and second predicting dense geometric correspondences to transform a successful grasp pose to a new observation. Retrieving similar samples has been addressed using learned feature descriptors from RGB-D images (Wohlhart and Lepetit, 2015; Balntas et al., 2017; Park et al., 2019a). These employ the triplet loss to train a network to produce smaller feature distances for pairs of images with similar viewpoints while producing larger feature distances for pairs of images with different viewpoints or those that contain different object classes. However, retrieving a similar viewpoint is insufficient when samples do not cover the entire object pose or when target objects are not constrained to a fixed set of a classes. This motivates our method that predicts geometric correspondences between images to match local areas regardless of different scales or detailed shapes. A simple approach is to encode and match local feature descriptors that represent local shapes in 3D point clouds or depth images (Zeng et al., 2017; Zeng et al., 2018b). Leveraging this idea, determining pixel-wise correspondences has been demonstrated in a pipeline to predict local correspondences or key-points while considering the global contexts of objects (Florence et al., 2018; Manuelli et al., 2019). The task of this line of research is to find the corresponding points between an input and a known reference object, therefore, they are not applicable when the reference image has to be selected from various objects and viewpoints based only on global shape.

In order to make this extension, we employ the normalized object coordinate space that has been used to estimate the 6D pose of instances (Li et al., 2019; Park et al., 2019b) and classes (Wang H. et al., 2019). Since NOC values represent coordinate values in the object's local frame and correspondences between the object model and the scene, predicting NOC values is sufficient for computing the transformation between local points from one observation to another. However, it is difficult to define NOC values without knowing the full 3D shape of an object or a common representation for a class. For our work, it is necessary to predict dense correspondences between pairs of images that have similar geometry. To that end, we represent NOC values in the reference frame of the camera viewpoint instead of the object. This alteration to the NOC representation is referred to as the view-dependent normalized object coordinate space. The prediction of VD-NOC values is used to compute the transformation of local areas that are relevant for grasps in order to transform stored grasp poses to the object in the input image.

3. METHODS

This section describes our methodology for incremental experience-based grasp learning. We begin with an overview of the framework. We then describe the dense geometrical correspondence matching network for retrieving experience samples and for generating dense 3D-3D correspondences. Lastly, we outline how grasps are transferred between local regions using the predicted correspondences.

3.1. Incremental Grasp Learning Framework

The main components of the incremental experience-based grasp learning framework are shown in **Figure 1**. The input is a depth image $D_i \in \mathbb{R}^{W \times H}$ and a segmentation mask $M_i \in \mathbb{R}^{W \times H}$ that has entries 1 for pixels belonging to the target object and 0 otherwise. The goal of the framework is to generate a pose for the gripper that will result in a successful grasp. This is represented as a rigid transformation $T \in SE(3)$ of the gripper pose in the camera coordinate frame.

The first step is to match the target object to samples stored in an experience database $\mathcal{E}$. Matching is done using the global geometric encoding from DGCM-Net, where the feature map f_i of the input image is compared to the feature maps of the database samples. Feature maps are the output of a geometry encoder that takes as input a surface normal image derived from the initial depth image. The set of samples with high matching score are used to propose a candidate grasp. For each database match in $\mathcal{E}$, the output of the VD-NOC encoder c_e and the geometry feature encoding f_e as well as f_i from the input are passed to the decoder of DGCM-Net to reconstruct the VD-NOC values $V_i \in \mathbb{R}^{W \times H \times 3}$. This represents a dense mapping between the pixels of the sample and the input and thus a transformation of the points in the 3D coordinates can be computed. Each experience has an associated grasp pose, therefore, the transformation between the images is applied to transform the experience grasp to the target object. Sensitivity to the difference in geometry between the input and sample is reduced by confining the alignment to the region around the grasp pose. The region of interest (ROI) on the sample $\mathcal{R}_e$ is determined from the overlap of the gripper with the 3D coordinates of the segmented object in the depth image. The corresponding ROI on the target object $\mathcal{R}_i$ is derived through the matches between the VD-NOC values. The ROIs are aligned by finding the optimal rotation and translation. The outcome is a proposal for a full 6D grasp pose for the target object.

Incremental learning operates by executing a selected grasp proposal and updating the database online with a new exemplar if the grasp is successful. Specifically, the depth image, surface normal image, VD-NOC values, ROI, and transformation of the grasp pose are stored. Unsuccessful grasp attempts do not provide any information for replicating past experience, therefore, no data is stored for failed grasps. As more successful experience is accumulated, the likelihood of finding a nearby match for a new input increases. The method is not restricted to only finding samples of exact object instances, but can match to new or unseen objects if they have geometry resembling those from experience.

3.2. Dense Geometrical Correspondence Matching

3.2.1. View-Dependent Normalized Object Coordinate Space

Predicting dense correspondences between two depth images (i.e., the depth image of the object to grasp and an experience in the database) is done by predicting a variant of NOC values. The

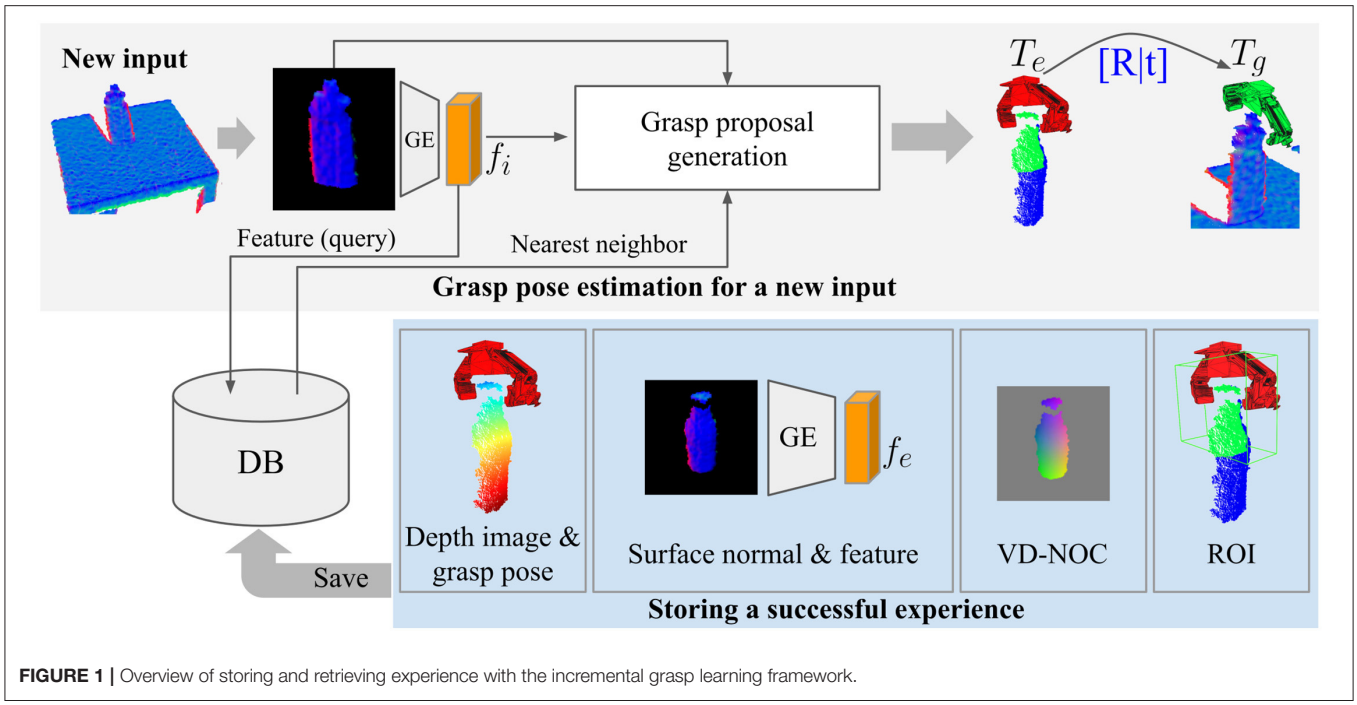


FIGURE 1 | Overview of storing and retrieving experience with the incremental grasp learning framework.

traditional NOC values represent the correspondence between the target object and another one in the target object's local frame. Typically this has been applied for object pose estimation where the target object is a reference model and the other object is an observation of the reference model in a scene.

To apply the same methodology without object models, we introduce the view-dependent normalized object coordinate values. The depth images for a reference and an input are converted to surface normal images. The VD-NOC values for the input V_i are computed using the 3D coordinates of each pixel I_i^{3D} from the input segmentation mask in the camera coordinate frame. Normalization is performed by setting the origin to the mean coordinate between the maximum and minimum values of I_i^{3D} according to,

$$V_i = \frac{I_i^{3D} - \bar{I}_i^{3D}}{\max |I_i^{3D} - \bar{I}_i^{3D}|}, \text{ where } \bar{I}_i^{3D} = \frac{\max(I_i^{3D}) + \min(I_i^{3D})}{2}. \quad (1)$$

Normalization is performed separately for each dimension resulting in different normalization factors for each axis. The direction of the z-axis is flipped to produce positive values for points that are nearer.

For grasping, the VD-NOC values are used to estimate the similarity between points on the target object in the input image and the points on the object in the experience database. A smaller distance between values in the VD-NOC values represents closer geometrical correspondence. These can be used to estimate the transformation of a set of points in the grasp pose ROI in order to transfer the grasp experience to the target object.

3.2.2. DGCM-Net Architecture

An overview of the dense geometric correspondence network is shown in **Figure 2**. DGCM-Net consists of a geometry encoder, VD-NOC encoder, and VD-NOC decoder. The purpose of the geometry encoder is to learn a representation that places images with similar geometry closer in feature space than images with dissimilar geometry. The purpose of the VD-NOC encoder-decoder is to reconstruct the VD-NOC values between a pair of images.

The input to the geometry encoder is a cropped surface normal image derived from the input depth image and segmentation mask. The cropped image is created from a 2D bounding box that is centered at the 2D projected point of the segmentation mask's centroid. The height and width of the image are adjusted to correspond to 30 cm spatial size in 3D space. The cropped image is then resized to 128 × 128 pixels. The first three blocks of the Resnet-50 (He et al., 2016) architecture is employed and initialized with the pre-trained weights using the ImageNet dataset (Deng et al., 2009). The output of the third block is passed to three convolution layers, kernel sizes = [3, 3, 2] and filter sizes = [256, 256, 128] with strides 2 for all, and two fully connected layers with 256 outputs. The *LeakyReLU* activation is applied to every layer output except the last layer that uses the tanh as an activation to transform feature descriptors to 256 dimensions.

The input to the VD-NOC encoder is a cropped VD-NOC image. The input is passed to five convolution layers, kernel sizes = [5, 3, 3, 3, 3] and filter sizes = [128, 256, 256, 256, 256] with strides 2 for all, and one fully connected layer with 256 outputs. The activation of each layer is the same for the geometry encoder. The VD-NOC decoder reconstructs the VD-NOC values for the input image with respect to the camera frame. The input to the decoder is the concatenated features from both geometry

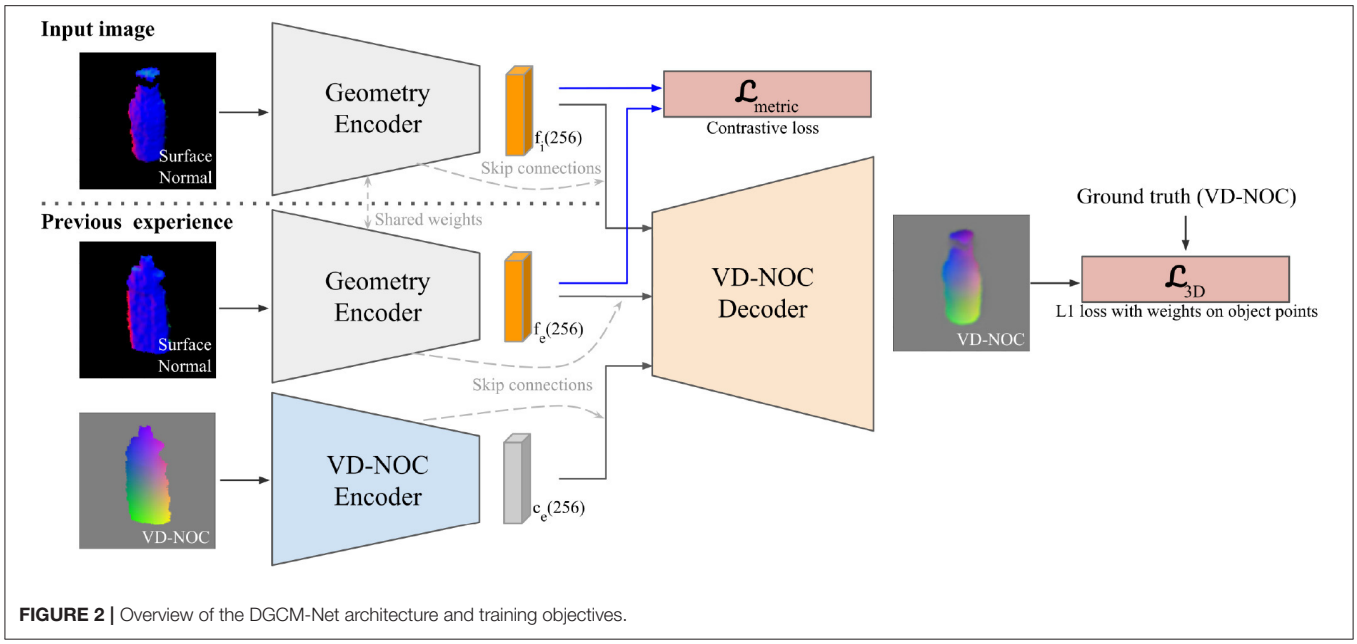


FIGURE 2 | Overview of the DGCM-Net architecture and training objectives.

encodings of the images and the output of the VD-NOC encoder for the reference. Skip connections (Ronneberger et al., 2015) are added by concatenating one half of the output channels of each intermediate layer of the encoders with corresponding layers in the decoder. This helps to predict fine details in local areas. The decoder ends with a fully connected layer with 2,048 outputs followed by five blocks of deconvolution and convolution layers. The output of the last convolution layer is the same size as the input image with three channels that represent the x, y, and z components of the VD-NOC values.

3.2.3. Training Objective

DGCM-Net has two tasks and therefore consists of two objectives in the training process. The first is the metric learning of feature descriptors to perform matching and the second is for reconstructing the VD-NOC values of an input image. For metric learning, the contrastive loss (Hadsell et al., 2006) is employed to minimize the Euclidean distance between features of similar geometry (a positive pair) while increasing the distance for a pair of different geometry (a negative pair) as formulated by,

$$\mathcal{L}_{\text{metric}} = \frac{1}{N}\sum_{i=1}^{N}(1-\omega_i)d_i^2 + \omega_i \max(10-d_i,0)^2, \qquad (2)$$

where ω denotes labels for pairs that are set to 0 for positive pairs and 1 for negative pairs. d denotes the Euclidean distance between encoded feature vectors (f_i,f_e $\in$ $\mathbb{R}^{256}$) of the target and experience images from the geometry encoder. The loss is computed for a mini-batch that consists of N pairs of training images.

For the reconstruction of VD-NOC values, the standard L1 loss is applied for each pixel p. Since background pixels are masked out, their values are easy to predict. Hence, the loss values for pixels on the object masks M_i $\in$ $\mathbb{R}^{W\times H}$ are weighted by a factor of 3 to more precisely predict the values of pixels in the object masks (Park et al., 2019b). The reconstruction loss is thus given by,

$$\mathcal{L}_{3D} = \frac{1}{N\times W\times H}\sum_{i=1}^{N}\left[3\sum_{p\in M_i}||V_i^p - V_{gt}^p||_1 + \sum_{p\notin M_i}||V_i^p - V_{gt}^p||_1\right]. \qquad (3)$$

The reconstruction loss is computed only if the pair of samples is positive. Finally, the objective of the training is the weighted sum of two loss functions,

$$\mathcal{L} = \mathcal{L}_{\text{metric}} + \lambda\mathcal{L}_{3D}, \qquad (4)$$

where λ is a weight balancing the two objectives. We set λ to 1 in our experiments.

3.2.4. Training Using Synthetic Images

Synthetic depth images are created to train the network. 3D models are sampled such that no two models are the same even after a scale change[1]. Objects are selected from the YCB object and model set (Calli et al., 2017) and listed in **Figure 3** (left). Depth images are rendered in OpenGL[2] for each object model by uniformly sampling a pose and randomly selecting scale factors for each axis. Five scenes are rendered with different scales for each sampled object pose. To avoid ambiguous views of symmetric objects, view angles are limited between 0 and 45 degrees on each axis. For cylindrical objects, no variation around

[1]The set of 3D models only contains one box because any other box can be constructed just by manipulating the scale in the different dimensions.
[2]https://www.opengl.org

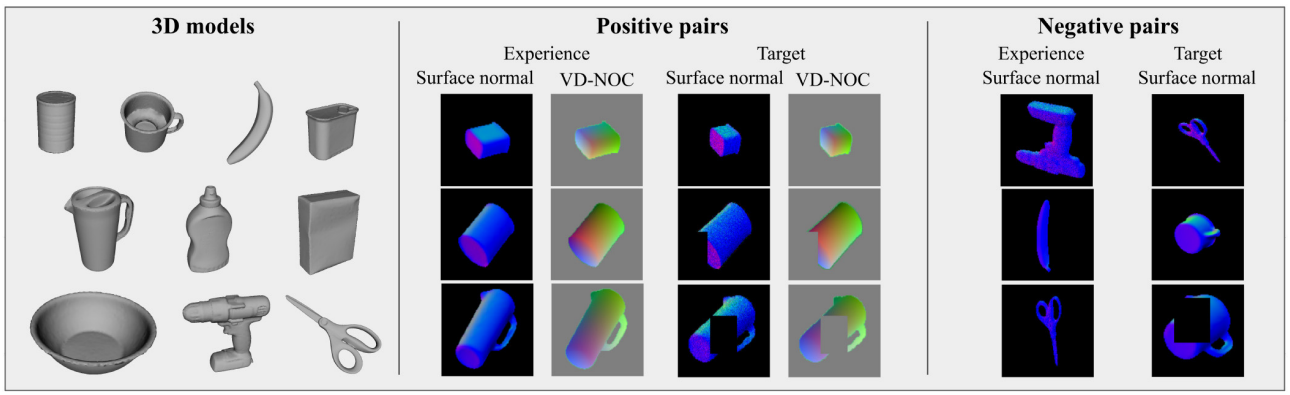


FIGURE 3 | (Left) Object models from the YCB dataset used to train DGCM-Net. **(Middle)** Examples of positive pairs with online augmentation, Gaussian noise and partial occlusion. **(Right)** Examples of negative pairs. The pairs are used to train feature vectors to have smaller distance for similar geometries.

TABLE 1 | Overview of the parameters used to generate the training data and for the online data augmentation.

Stage	Data generation		Online augmentation	
Parameters	**Scale (each axis)**	**Distance to camera**	**Frac. of occluded region**	**Gaussian noise**
Range	$\mathcal{U}$(0.8, 1.5)	$\mathcal{U}$ (1m, 1.7m)	$\mathcal{U}$ (0.0, 0.25)	$\mathcal{N}$ (μ = 0,σ = 0.01)

the rotational axis is applied. Parameters used in the generation process are summarized in **Table 1**. The result for every training sample is a depth image, VD-NOC image, annotated pose, annotated scale factors for each dimension and a look-up table of visible vertices. Approximately 166 k images are created and used for training.

Metric learning requires positive and negative pairs. Positive pairs are obtained from samples of the same object in different poses when a pair of images share more than half of the visible vertices. Negative pairs are obtained from different objects or different poses of the same object when images share less than half of the visible vertices. Examples of training samples of both types are given in **Figure 3** (middle and right). For positive pairs, the target VD-NOC values (i.e., the ground truth value) is computed using the relative pose of the object, which is known for the training samples. Thus, the VD-NOC values that are defined in the camera frame of the first element of the pair are transformed to the camera frame of the second element. For our grasping framework, this amounts to transforming the VD-NOC values from the object in the input image to the object in the experience database.

Further augmentation is applied to the image samples to improve robustness against occlusion and noise. Occlusion is simulated by setting a partial area in the surface normal image and the corresponding entries in the VD-NOC values to zero (i.e., the value for the background). This enables the network to learn features that still return good matches between an input and samples in the database even when one is occluded. Gaussian noise is also applied to both images to cope with the expected noise from real sensors. **Figure 3** (middle and right) presents examples after applying the augmentation. More details about the parameters used for augmentation are provided in **Table 1**.

We train the network for 35 epochs using the ADAM optimizer (Kingma and Ba, 2015) while assigning 25 positive pairs and 25 negative pairs for each batch. The learning rate is initially set to 0.0001 and divided by a factor of 10 every 5 epochs. After training the network once, the weights are fixed for all experiments in this paper without any fine-tuning.

3.3. Generating Grasp Proposals

The overview in **Figure 1** shows the process of retrieving and generating grasps given an input depth image. First, the surface normal image of the input is encoded to a feature map f_i by the geometry encoder. This is compared to all feature maps $\{f_e\}\ \forall\ e \in \mathcal{E}$ to find a set of nearest neighbors $\mathcal{N}_i \subset \mathcal{E}$. The stored VD-NOC values V_e of a sample $e \in \mathcal{N}_i$ is loaded to compute the VD-NOC feature map c_e. Given c_e, f_e, and f_i, the decoder predicts the VD-NOC values of the input depth image $V_{i;e}$ (VD-NOC values of D_i in the frame of D_e) as shown in **Figure 4**. The ROI of the experience $\mathcal{R}_e$ is used to compute the corresponding ROI for the input $\mathcal{R}_i = \{p \in V_i : \min_e |p - p_e| < \theta_c\ \forall\ p_e \in \mathcal{R}_e\}$, which is the subset of points whose distance to the nearest points in $\mathcal{R}_e$ is below a threshold θ_c. The predicted VD-NOC ROI points is denoted $V^{\mathcal{R}}_{i;e}$ and are defined in the camera frame of D_e. Each pixel of $V^{\mathcal{R}}_i$ forms a 3D-3D correspondence from the VD-NOC values $V^{\mathcal{R}}_{i;e}$ and $V^{\mathcal{R}}_{i;i}$ that are defined in D_e and D_i. Thus, an initial rotation from the camera frame of the experience to the camera frame of the input is derived by aligning the ROI VD-NOC images. The grasp pose T_e is then aligned to D_i by computing the rotation that minimizes the summation of distances of the correspondences given by,

$$R_{\text{init}}, t_{\text{init}} = \arg\min_{R,t} \sum_{\mathcal{R}_i} ||(RV^{\mathcal{R}}_{i;e} + t) - V^{\mathcal{R}}_{i;i}||_2. \tag{5}$$

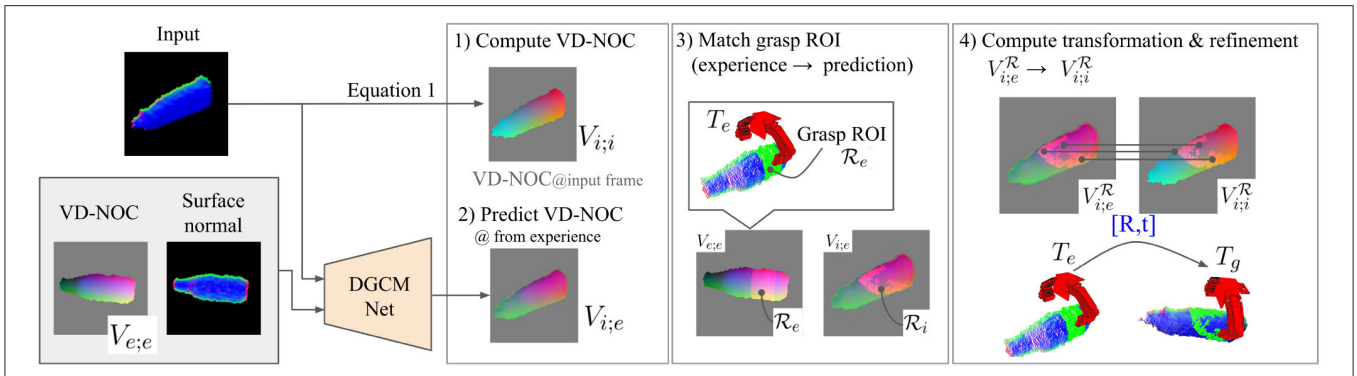


FIGURE 4 | Overview of the process for generating grasp proposals from the nearest neighbor experience.

The optimized solution for Equation (5) is obtained using singular value decomposition. The unit of t_{init} does not correspond to the scale of the 3D space because the VD-NOC values are normalized. Therefore, the translation t_{init} is separately computed using the difference between the mean coordinates between the maximum and minimum values of the ROI points as was applied in Equation (1). The computed rotation and translation are used to transform all ROI points of the experience $\mathcal{R}_e$ to the scene and the alignment is refined by applying the iterative closest point algorithm. The grasp pose in the experience T_e is transformed to create the grasp proposal T_g by applying the same refined transformation. Finally, the gripper position is moved a fixed distance from the object surface by translating along the approach direction with respect to the closest point in the input.

Each match in the database has an associated score in the range $[0, 1]$ that represents that similarity of the depth image to the input, which is used as a pseudo-measure for the quality of the grasp. This score is computed as,

$$S(i, e) = e^{-||f_i - f_e||_2}. \tag{6}$$

The final output is a set of grasps $\mathcal{G} = \{(T_g, s_g)\}$ where each grasp proposal is composed of a transformation of the gripper into the scene T_g as well as a score value s_g using Equation (6).

4. OFFLINE EXPERIMENTS

This section analyzes the grasp proposal method with a hand annotated dataset. Experiments are performed to first investigate the quality of grasp pose prediction with respect to the size of the grasp experience database and secondly to evaluate the ability to transfer grasps between observations of objects within the same and to different classes. The threshold for matching ROI correspondences θ_c is set to 0.3 for all experiments. This value produces reasonable separation of ROI areas and other parts of objects. Every stored experience is duplicated with in-plane rotations at angles between −90 and +90 degrees with a step size of 45 degrees. This enables grasp transfer to objects in new poses. Code for DGCM-Net is publicly available at https://rgit.acin.tuwien.ac.at/v4r/dgcm-net.

4.1. Dataset

A dataset is created to evaluate the quality of grasp prediction comprising depth images of the objects shown in **Figure 5**. These objects are organized into seven classes: `can`, `mug`, `cup`, `bottle`, `bowl`, `box`, and `clamp`. Four instances are used for each class and a number of these instances are from the YCB object dataset (Calli et al., 2017), while other instances are objects commonly found in homes. The dataset is available at https://www.acin.tuwien.ac.at/en/vision-for-robotics/software-tools/lfed-6d-dataset/.

Recordings are made by placing each object on a small table and capturing a depth image with an ASUS XTion Pro Live RGB-D camera. Each object is placed in various poses and locations, and the camera is moved between two different heights. The object is segmented in each depth image by detecting the table surface with RANSAC and selecting all points that remain above the table plane. The dataset does not require ground truth segmentation, but instead should be segmented by the same method that extracts the masks for the input images in order for the entries in the experience database to best resemble the inputs.

Grasp poses for a parallel-jaw gripper are manually annotated in the depth images. Each depth image consists of possibly multiple grasp annotations according to their direction, for example, from the top or from the side. The full dataset used for testing consists of depth images, segmentation masks, and grasp poses for 28 objects.

4.2. Measuring Grasp Pose Quality

Reporting quantitative statistics requires the quality of the estimated grasp poses to be measured. It is possible to execute physics simulation and to check for grasp success, however, to isolate the grasp prediction itself, we measure the difference in grasp pose for an input with respect to the annotated pose. The experiments are simplified by selecting only grasp annotations on the top of the objects when they are placed in their upright canonical pose. Even though the grasp proposals are limited to top down, multiple poses are available, especially for objects that are symmetric or are elongated in the x or y dimension such that translations of a top-down grasp are equivalent. Consistency of top-down grasp poses is

FIGURE 5 | Objects in the dataset used for the offline experiments and to supplement past experience during the online experiment.

ensured by testing with the subset of classes `can`, `mug`, `cup`, and `bottle`.

A grasp pose is regarded as correct when the translation error is less than 5 cm and the rotational error around the x- and y-axes is less than 15 degrees. A rotation error around the z-axis in the gripper frame, which is parallel to the rotational axis of an object, is ignored since it should be a successful grasp regardless of the rotation with respect to this axis.

4.3. Increasing Experience

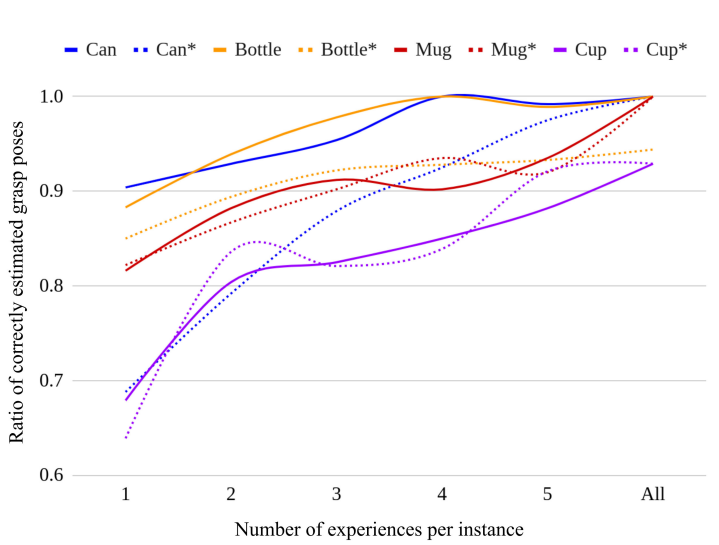
The left plot of **Figure 6** shows the ratio of correctly estimated grasp poses with increasing number of experience per instance. Solid lines show results when only the experience for the relevant class is considered and the dotted lines show the results when experience for all classes is considered. For each configuration (class and number of experience per instance), we perform 10 iterations using randomly selected samples in the iteration. The results with class specific experience demonstrate that the grasp poses are often correctly estimated even if only one experience is included per instance. For the `can` and `bottle` classes, the correct estimation is approximately 90%, while the worst performing class, `cup`, achieves 68%. However, as the number of experiences increases per instance in each class, the grasp pose estimation improves.

The dotted lines show the variation in performance when including other classes for experience, which reflects more practical scenarios in the real world. Except for the `mug` class, the performance slightly drops because the retrieval of experiences from different classes can cause inaccurate prediction of VD-NOC values. However, the accuracy still achieves more than 79% when two experiences are included per instance. This implies that the feature space encoded by the geometry encoder is sufficient to distinguish different geometrical shapes. The performance gap for the `can` class, which has the most simple shape, is comparably larger than for the other classes. This is because the mapping from more complex to simpler geometries produces inaccurate estimations by transferring detailed shapes into simpler geometries. We discuss more detail regarding this relationship between classes in the following section.

Figure 7 shows qualitative results. The figure shows the three nearest experiences and the best transformed pose for different instances from different classes. The first and second rows are obtained when the experience database contains samples from all classes and include the exact instance in the test image. The third and fourth rows are obtained after excluding the instance in the test image so that different instances in the same class must be retrieved to generate grasp proposals. The results reveal that the grasp poses are transformed to similar locations and directions even if object poses from experience are different (see the grasp poses for the `can` in the third row).

4.4. Transfer Between Instances and Classes

These experiments show that experience can be transferred between instances and classes. The experiments are conducted by using all experience from a single object instance while testing on different instances. The evaluation metric and subset of target grasp poses are the same as in the previous experiment. The matrix on the right of **Figure 6** shows that the experience of instances transfer well to other instances within the same class.



Test / Exp.	Can 1	Can 2	Can 3	Can 4	Bottle 1	Bottle 2	Bottle 3	Bottle 4	Mug 1	Mug 2	Mug 3	Mug 4	Cup 1	Cup 2	Cup 3	Cup 4
Can 1	1	.67	.33	0	.17	.33	0	0	.08	.14	0	0	0	.33	.29	0
Can 2	.50	1	.50	0	.83	1	1	.75	0	.14	0	0	0	.33	0	0
Can 3	0	1	1	0	.33	.67	1	.50	0	0	0	.23	0	0	0	0
Can 4	0	0	0	1	0	0	0	0	0	.14	0	0	.20	0	0	.10
Bottle 1	0	.33	.50	0	1	.83	1	1	0	0	0	0	.17	0	0	0
Bottle 2	0	0	.83	0	.83	1	1	.75	.08	.43	.09	0	.20	.50	0	.10
Bottle 3	0	.33	.17	0	1	.83	1	.75	0	0	0	0	0	.17	0	0
Bottle 4	0	.67	1	0	1	1	1	1	0	0	0	0	0	0	.29	0
Mug 1	0	0	0	.83	0	0	0	0	1	.64	.46	.23	.40	.17	.43	.30
Mug 2	.50	.83	.17	0	0	0	0	.50	.69	1	.64	0	.80	.83	1	0
Mug 3	0	0	0	0	0	0	0	0	.62	.29	1	.92	.40	0	0	.20
Mug 4	0	0	0	.17	0	0	0	0	.31	0	.46	1	0	0	0	.30
Cup 1	0	.17	.17	.50	0	0	0	0	.85	.93	.82	.23	1	.33	.86	0
Cup 2	0	0	0	0	0	0	0	0	.46	.79	.18	0	.60	1	.43	.50
Cup 3	0	0	.33	.83	0	0	0	0	.08	.50	.64	0	.20	.50	1	0
Cup 4	0	0	0	1	0	0	0	0	.23	.14	.09	.62	0	0	0	.80

FIGURE 6 | Results of the offline experiments. **(Left)** Ratio of accurately estimated grasp pose with increasing number of experiences per instance. Solid lines show results when class specific experience is used for each test class. Dotted lines show results when experience from all classes is used. **(Right)** Ratio of accurately estimated grasp poses using experience from each instance in all classes.

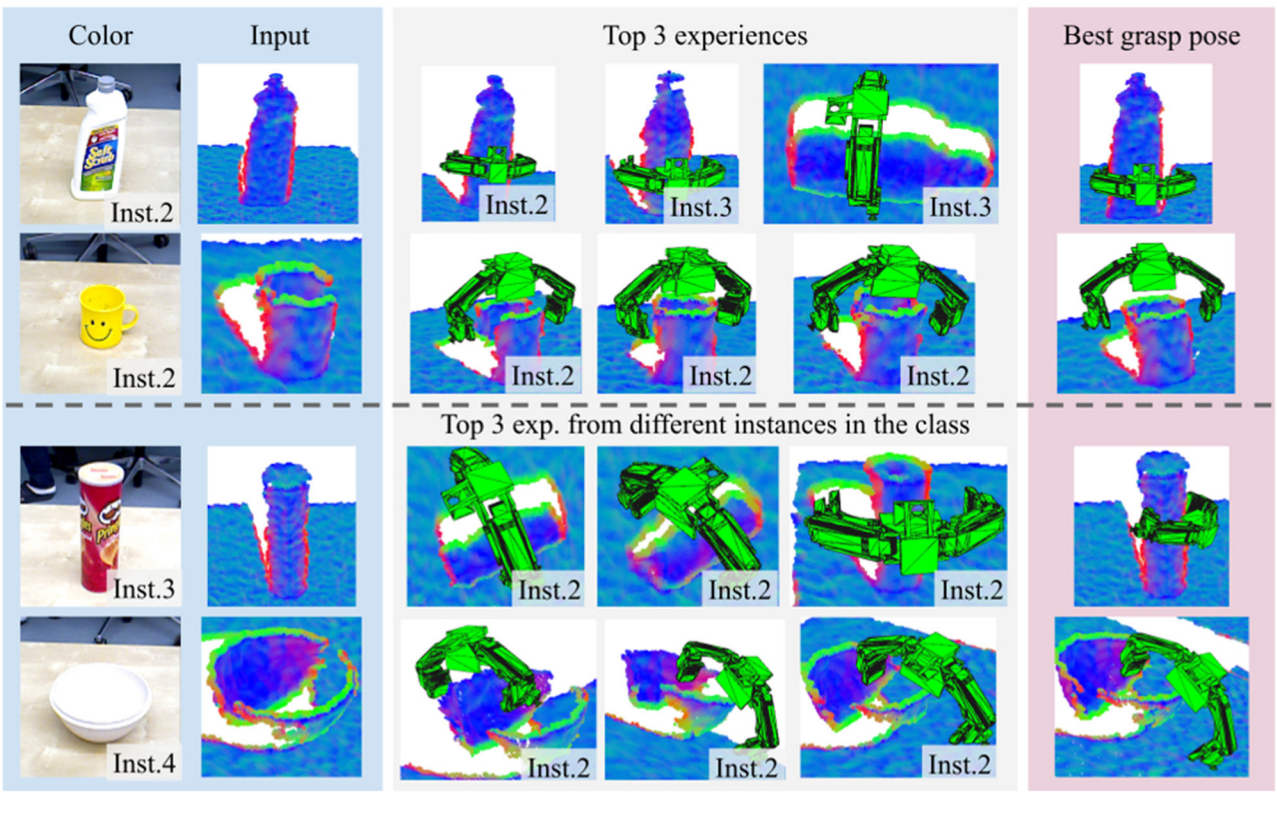


FIGURE 7 | Examples of the three nearest experiences and estimated grasp poses for example input images. Top two rows show examples when the same instance is included in the experience database. Bottom two rows show examples when the instance is not in the experience database.

Furthermore, experience also transfers beyond the class. For example, many good grasps are found for `bottle` instances when provided by experience from `can` instances (both types of objects have a closed surface on the top) and that instances of the `cup` class provide sufficient experience for grasping `mug` instances (both types of objects have no surface on the top).

However, the results show that it is difficult to transfer the experience of a class to an instance in the same class when the geometry and scale of the instance are different from the other instances in the class (e.g., `Can4` and `Cup4`). Thus, better grasp poses are obtained when the experience is obtained from geometrically similar objects regardless of explicit classes. It is also observed that grasps for simpler geometries (e.g., from `can` to `bottle` and `cup` to `mug`) are more accurately transferred, while the other direction from complex geometry to simpler geometry is more difficult. This is because the network tries to predict VD-NOC values of detailed shapes of objects in the experience set, such as handles of `mug` instances, which potentially causes errors by predicting corresponding points even if the shapes are missing in the new object.

5. ROBOT EXPERIMENTS

This section presents results of real-world grasping experiments with a mobile manipulator. First, we describe the hardware set up used for the experiments. Second, we compare our method to baseline approaches. Third, we evaluate the full pipeline of online incremental grasp learning. Finally, we demonstrate the extension to semantic grasp learning.

5.1. Experimental Details

The robot experiments are performed with the Toyota Human Support Robot (Yamamoto et al., 2019). The platform consists of a 4-DOF arm but motions are computed including the omni-directional base, which effectively offers seven degrees of freedom. Motions for grasp execution are planned using MoveIt (Chitta et al., 2012b)[3]. The end-effector is a parallel-jaw gripper and grasp success is measured by checking the distance between the tips of the gripper after the target object is lifted. If the distance is non-zero, then the grasp is declared successful, otherwise, it is a failure. Depth images are captured with the onboard ASUS XTion Pro Live RGB-D sensor positioned on the head of the robot.

For all grasping experiments, individual objects are placed on a small table that has a height of 45 cm. The robot is approximately positioned 30 cm from the table (edge of robot base to edge of table). The head of the robot is tilted such that the camera faces the center of the table. The torso of the robot is raised to give an approximate distance from the camera to an objects of 1 m to suit the optimal range of the sensor. Objects are segmented from the table with the same procedure for generating segmentation masks for the dataset.

All code is written in C++ and Python, and is running on the robot in Ubuntu 16.04. ROS (Quigley et al., 2009)[4] is used for process communication. DGCM-Net is implemented in Tensorflow and is running on an external PC with an NVIDIA GTX 1050 Ti.

[3]http://moveit.ros.org
[4]https://www.ros.org/

5.2. Comparison to Baselines

Experiments are conducted to measure the grasp performance of our framework. For comparison, experiments are also performed with a number of baselines. The full set of methods is as follows:

- HAF: The approach introduced in Fischinger et al. (2015), where height accumulated features are extracted from point clouds to abstract grasp-relevant structure. The features are computed on different regions of the input and a support vector machine is trained to predict the quality of the grasp for each feature. Both top-down and forward-facing grasps are enabled, and the output with highest score is executed. We use the original code provided[5].
- GPD: The approach introduced in ten Pas et al. (2017), where grasps are sampled using the surface geometry of the input point cloud. Grasp success for each sample is classified using a convolutional neural network (CNN). This takes as input three images: an averaged height map of occupied points, averaged height map of the unobserved region, and averaged surface normals. Given this input, the CNN generates a score value. Finally, grasps are clustered and the highest scoring cluster is selected. We use the original code provided[6] and the full 15 channel version.
- DGCM-Net: The grasp proposals from DGCM-Net using pre-collected experience for the relevant object classes in the experiments. Similar to GPD, the set of proposals from DGCM-Net are clustered and the highest scoring cluster is executed. Clustering is performed by grouping all grasps within 5 cm translation and 15 degrees rotation. The grasp of the cluster is the mean pose of the proposals that make up the cluster. The cluster with the highest summed score is executed. The number of nearest neighbors to be retrieved by DGCM-Net is set to 10.
- GPD + DGCM-Net: Grasps are proposed using GPD and the scores are modified by the predictions from DGCM-Net. First, the grasps from the GPD method are computed and the scores are normalized to the range $[0, 1]$. Then DGCM-Net is run on the same input and for each GPD candidate, we find all DGCM-Net proposals within 5 cm translation and 15 degrees rotation. The experience score is the average of the scores for all DGCM-Net grasps deemed to be nearby. The final score for each GPD candidate is the average of the normalized GPD score and the summed experience score. The grasp with the highest final score is executed.

Many robotic grasping approaches are successful for the bin-picking task (e.g., Mahler et al., 2017). However, these are focused on 2D grasping and therefore expect a top-down view of the scene and only generate a grasp parallel to the camera axis. This are unsuitable for our robot platform due to the position of the arm on the front of the body that occludes the scene when facing the camera directly downwards. Additionally, bin-picking methods are at a disadvantage because they only generate grasps for a single approach direction. It

[5]https://github.com/davidfischinger/haf_grasping
[6]https://github.com/atenpas/gpd

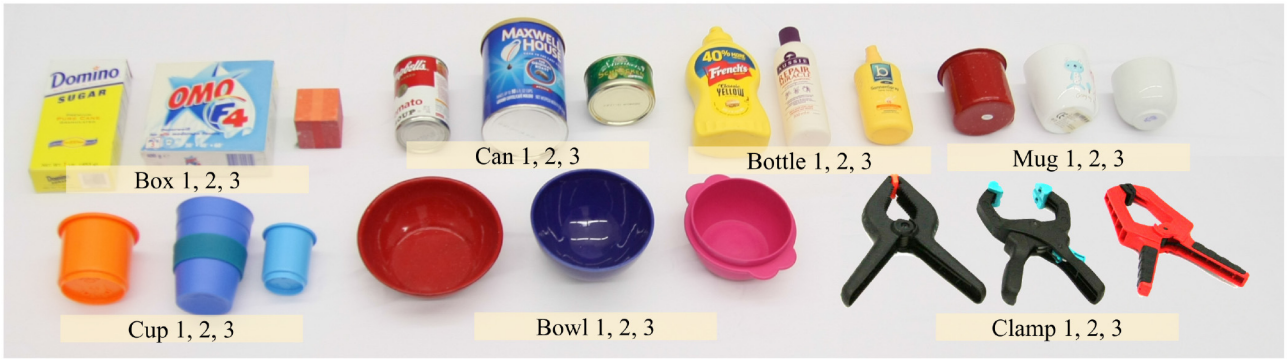


FIGURE 8 | Test objects used for the online grasping experiments. Three instances in seven classes are used.

TABLE 2 | Grasp success rate of our framework and baseline methods for different target object classes.

	HAF	GPD	DGCM-Net	GPD + DGCM-Net
Box	0.87	0.80	0.67	1.00
Can	0.87	0.67	0.93	0.73
Bottle	0.87	0.93	0.93	0.93
Mug	0.80	0.80	0.87	1.00
Cup	0.80	0.80	0.73	1.00
Bowl	0.40	0.87	0.80	0.87
Clamp	0.40	0.60	0.60	0.67
Average	0.71	0.79	0.79	0.89

The bottom row shows the average for all classes.

is left to future work to extend the evaluation to this type of scenario.

The experiments are performed for objects from the classes `box`, `can`, `bottle`, `mug`, `cup`, `bowl`, and `clamp`. Three instances are chosen per class and five poses are considered per instance. The five poses for each instance are kept constant for the experiments with each grasping method. The objects selected for the experiments are shown in **Figure 8**. These include one object from the YCB dataset for each class from the objects used in section 4, in particular, the `sugar box`, `spam can`, `mustard bottle`, `red mug`, `orange cup`, `red bowl`, and `XL clamp`. The other two objects for each class are a mixture of YCB objects and common objects found in homes.

The experience used for our method is an extension of the database from section 4 that includes instances from the additional classes of `box`, `bowl`, and `clamp`. Since we are interested in observing the grasp performance for unseen objects, the YCB objects selected as target objects are removed from the experience database.

Performance is measured by grasp success rate, which is the number of successful grasps divided by the total number of attempts. **Table 2** reports the average grasp success rate for each class and the average for all classes (bottom row). The results show that our method performs equivalently to GPD and that both methods outperform HAF (+8%). However, combining experience with GPD achieves a much higher grasp success rate overall. In comparison to the original GPD method, this is an increase of 10%.

For most classes, the combined approach performs either the same as the best performing individual method or better. The only exception is the `can` class, which has 20% lower grasp success rate than our direct method. Our observation during the experiments is that GPD often proposed grasps on and orthogonal to the rim of the `can` objects, which resulted in failures. This exposes the flaw that if the initial candidates are unfavorable, the combined approach cannot improve. For these objects, when grasp experience on the top of the `can` are stored, the grasps on the rim are still similar in position and orientation to warrant their selection.

Surprisingly, the `box` class is the most difficult for our approach despite having easy geometry to compute a grasp as shown by the high success rate of HAF. This can be explained by the fact that all objects of this type can be represented by a single `box` by changing the scale in the different dimensions. Thus, the network has to decide whether to regard a new instance as a scaled version of an experience or as a transformed (i.e., rotated) instance. The ambiguity causes noisy predictions of VD-NOC values. Furthermore, since grasp proposals are transformed from previous experiences and are ideally in a similar grasp location, a scale change may cause the prediction to exceed the range of the gripper, resulting in its rejection due to the collision.

5.3. Incremental Learning

This set of experiments demonstrate the full incremental learning framework. The test objects chosen are the `gray clamp` from our dataset, the `plastic drill` from the YCB object dataset and a `gaming controller`. Past experience is stored in the database, however, not for the classes of the test objects. Therefore, experience from the `clamp` class is removed. Since good grasps may not be generated for the unseen objects, GPD is used in the beginning until DGCM-Net makes reasonable predictions. A threshold of 0.9 is set as the minimum feature distance that must be achieved by the output of DGCM-Net,

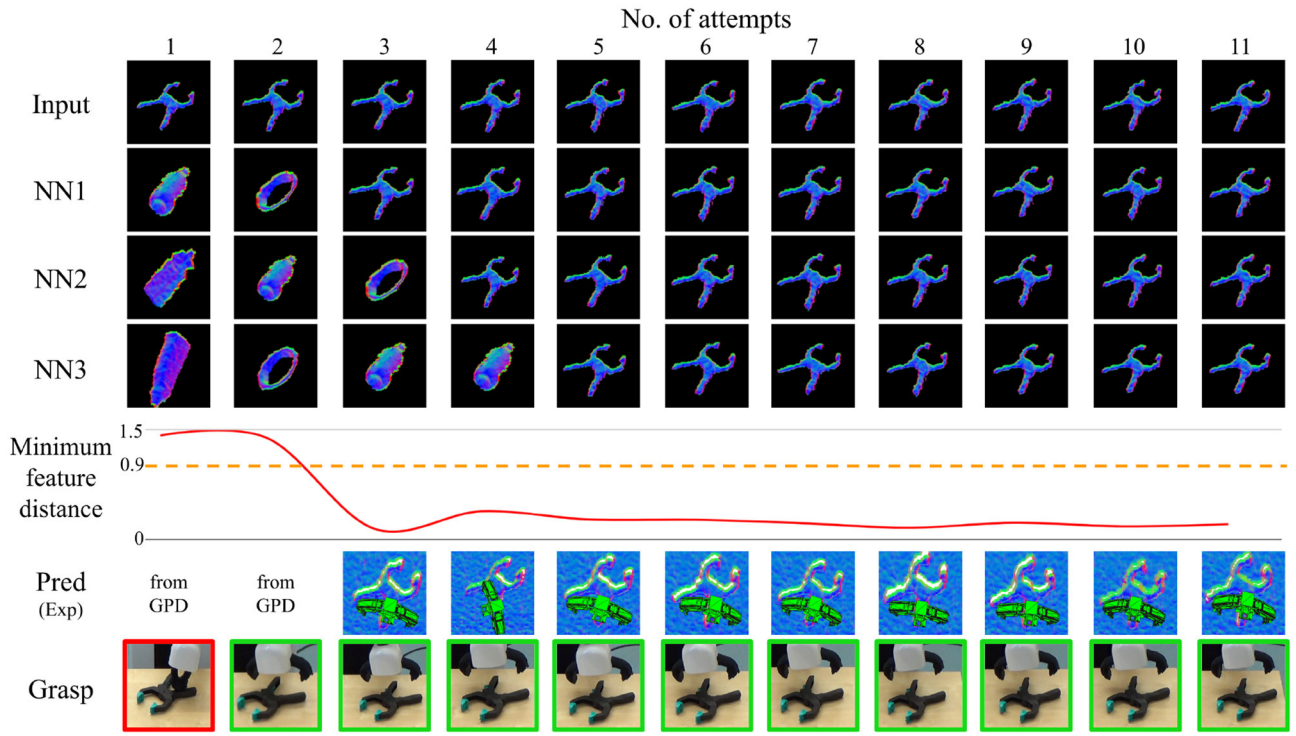


FIGURE 9 | Summary of results for the incremental learning experiment with the `gray clamp`.

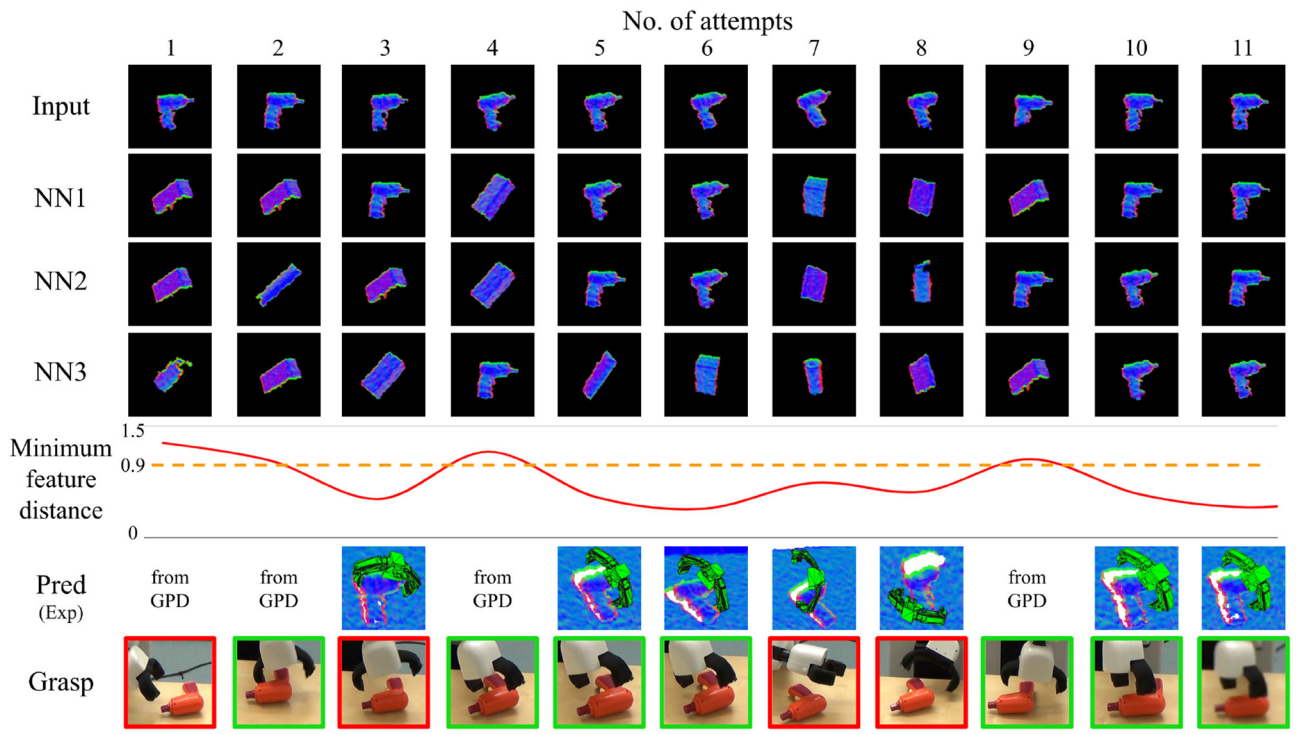


FIGURE 10 | Summary of results for the incremental learning experiment with the YCB `plastic drill`.

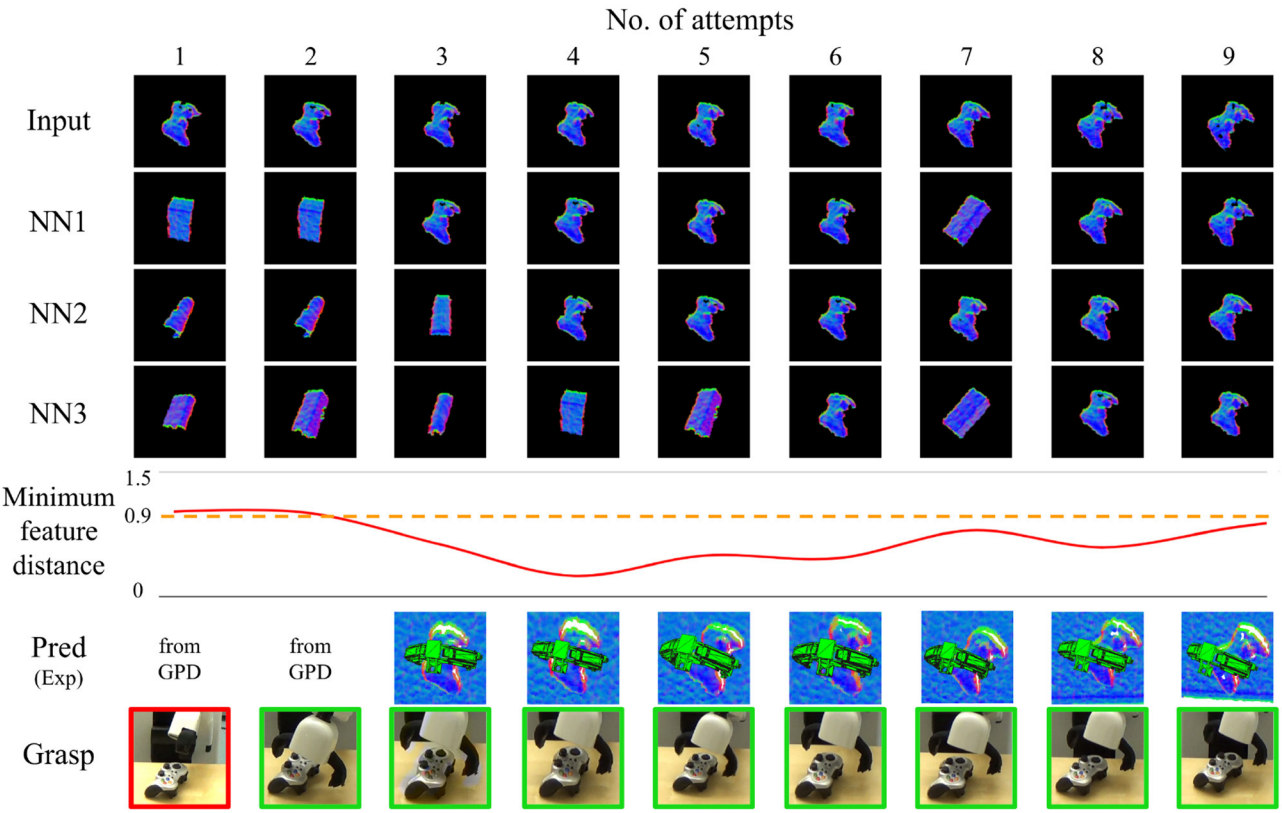


FIGURE 11 | Summary of results for the incremental learning experiment with the `gaming controller`.

otherwise, the best grasp from GPD is executed. Typically it only takes one or two successful attempts for the system to switch from GPD to DGCM-Net. The objects are placed randomly on the table at the beginning of each experiment and the system runs autonomously, with the robot grasping the object from where it lies after a successful or failed attempt. The object is only handled by a person if it is unintentionally moved near the edge of the table and presents a risk of falling. After successful grasps, objects are placed on the table by the robot and receive a slight variation in pose; failed grasps typically cause considerable object movement. After any grasp attempt, the robot base returns to the start position and the localization inaccuracy generates further viewpoint variation.

Figures 9–11 show the evolution of grasp success for the three objects. In these figures, the first row shows the surface normal image of the input and the second to fourth rows show the nearest three matches in the database. Below this we plot the minimum feature distance of the nearest neighbor. Lastly, we show the best grasp proposal from DGCM-Net and the actual gripper position during the grasp captured by an external camera (red border indicates failure and green border indicates success). A video of the experiments is provided in the **Supplementary Material** and is also available at https://youtu.be/iI_P1UVXfjo.

From these experiments, we make two observations. Firstly, after the first successful attempt from GPD is recorded, the robot typically continues to grasp the target objects successfully. The predicted grasps for each attempt confirms that our method reliably predicts the same successful experience so long as the object and its shape is correctly identified. The second observation is that the minimum feature distance drops below the threshold after as many as one sample is in the database. This is most apparent for the `gray clamp` in which the minimum feature distance is very small for all subsequent trials.

The grasping for the `plastic drill` and `gaming controller` are less reliable than for the `gray clamp`. For the `plastic drill`, the system still exhibits some failures even after accumulating experience. In both cases, the nearest feature distance does not converge to the same low value as was observed for the `gray clamp`. The reason is that the `plastic drill` and `gaming controller` have less distinct shape and therefore are more difficult to match. This is especially noticeable when the objects have rotated. The objects are often confused as an instance from the `box` class and the grasps for the matching object is executed. Fortunately for the `gaming controller`, the execution still results in success. However, for the `plastic drill`, the predicted grasp is not very good and the grasp fails.

5.4. Semantic Grasping

A final set of experiments demonstrate the extension of our method to generate semantic grasps for instances belonging to

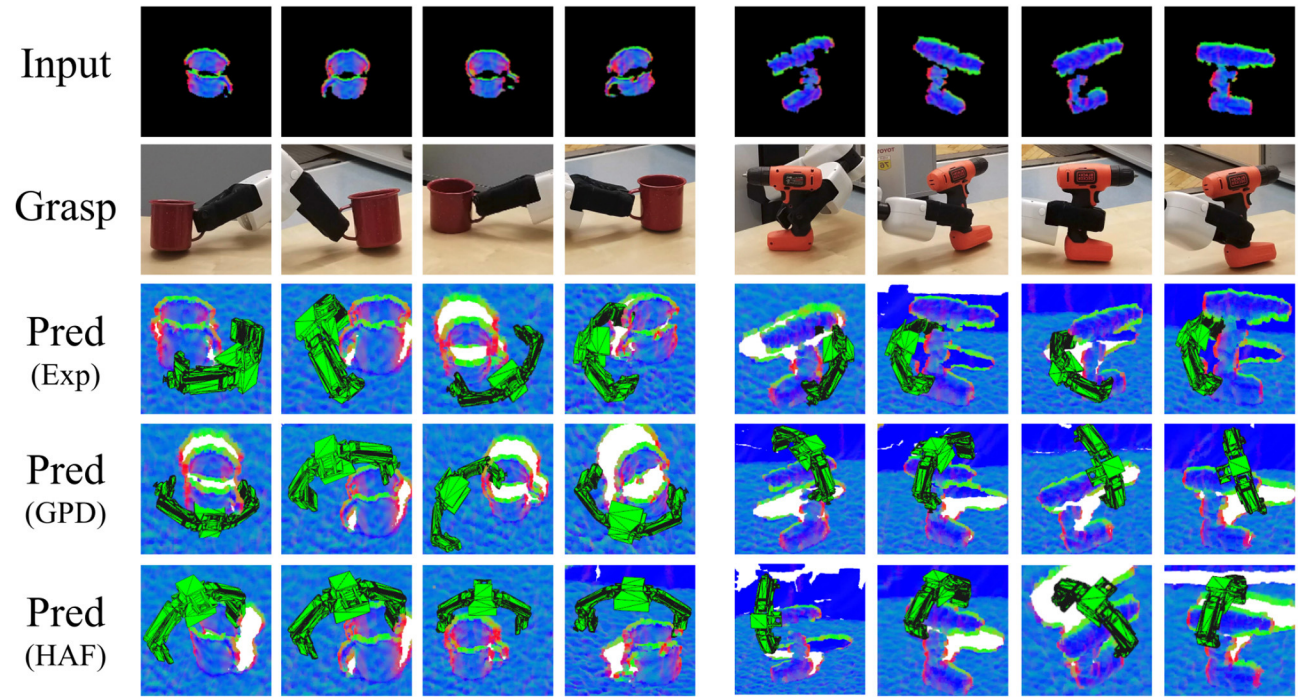


FIGURE 12 | Examples of semantic grasping for the YCB red `mug` and the YCB orange `drill` in four different poses. **(First row)** Surface normal image. **(Second row)** Grasp pose executed by the robot. **(Third row)** Predicted grasp from DGCM-Net. **(Fourth row)** Predicted grasp from GPD. **(Fifth row)** Predicted grasp from HAF.

the same functional class. For these experiments, we investigate grasps on the handles of `mug` and `drill` objects. The experience is hand annotated for test exemplars of instances similar to the target object. It is possible for the robot to self-learn semantic grasping by incorporating, for example, affordance detection to indicate whether the location of the grasp matches the affordance of the object part (Do et al., 2018). This is out of scope for the present article and left for future work.

Example grasp proposals generated by our method as well as GPD and HAF for a `mug` and `drill` in various poses are shown in **Figure 12**. Our method reliably generates grasp poses on the relevant object part, while both GPD and HAF fail to do so. Although the grasps from GPD and HAF may result in success, they do not support the functional use of the object. For the `mug`, it is understandable that the handle is not grasped because the quality of the depth data on that part of the object is very poor and does not characterize a stable grasp. Our method, on the other hand, does not only rely on the local structure to estimate the grasp. So long as there is some cue about the handle, as is present in these selected examples, the handle grasp is generated. The `drill` offers more depth data on the handle but the baselines still prefer to grasp the head. HAF is executed to find both top and front grasps, and the best scoring grasp is shown. Nonetheless, a top grasp or a front grasp on the head is preferred instead of the handle. A video of the grasps executed by the robot is provided in the **Supplementary Material** and at https://youtu.be/iI_P1UVXfjo.

6. CONCLUSION

This article presented an approach for incrementally learning grasps by leveraging past experience. In our system, every successful grasp is stored in a database and retrieved to guide future grasps. This is accomplished with the dense geometric correspondence network that is trained to predict the similarity between newly acquired input depth images and stored experiences as well as to predict 3D-3D correspondences to transform grasp poses. A descriptive feature space is constructed for the retrieval task using metric learning and correspondences are established by predicting view-dependent normalized object coordinate values.

Offline studies with a dataset showed that our approach precisely recovers grasps from experiences with the same object and also transfers well to unseen objects from the same or different class. Furthermore, results showed that more experience leads to more reliable grasp proposals. Hardware experiments with a mobile manipulator showed that our experience-based grasping method performs equally successful as the baselines and integration with the baselines shows overall superior performance. Additional experiments demonstrated the full

online capability to efficiently learn grasps for unseen objects, often needing only one or two successful grasps to reliably re-grasp the same object. Finally, an extension was demonstrated whereby specific grasps, such as those on handles, can be desired in order to achieve semantically meaningful grasps.

One direction for future work is to include more instances per class when training DGCM-Net to better generalize over varied shapes of a class instead of simply manipulating scales of an object for each class. Furthermore, it has been observed that points on the bottom of objects are missing from the object masks since the points are regarded as table or background. Thus, more detail about objects would be extracted by applying a segmentation method that refines the mask using color information. Another avenue of future work is to include failed experience during the learning phase. Particularly for objects that are difficult to grasp, it may take a large number of attempts to finally succeed. By also considering failures, it would be possible to reject grasp candidates and thus more quickly guide grasping to successful regions. Currently our method was only tested with a parallel-jaw gripper. It would be interesting to extend this work to other hardware, such as three-finger grippers or anthropomorphic hands. It would be even more interesting to investigate how to transfer grasps between grippers so that experience learned by one platform can be exploited by another platform with different hardware.

DATA AVAILABILITY STATEMENT

The dataset generated for this study is available at https://www.acin.tuwien.ac.at/en/vision-for-robotics/software-tools/lfed-6d-dataset/.

AUTHOR CONTRIBUTIONS

TP, KP, and MV conceived the idea of the presented work, contributed to the analysis of the results, and to the writing of the manuscript. TP and KP implemented the system and conducted the experiments.

FUNDING

The research leading to these results has received funding from the Austrian Science Fund (FWF) under grant agreement Nos. I3969-N30 (InDex), I3967-N30 (BURG), and I3968-N30 (HEAP).

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/frobt.2020.00120/full#supplementary-material

REFERENCES

Antonova, R., Kokic, M., Stork, J. A., and Kragic, D. (2018). "Global search with Bernoulli alternation kernel for task-oriented grasping informed by simulation," in *Proceedings of the Conference on Robot Learning* (Zurich), 641–650.

Balntas, V., Doumanoglou, A., Sahin, C., Sock, J., Kouskouridas, R., and Kim, T.-K. (2017). "Pose guided RGBD feature learning for 3D object pose estimation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision* (Venice), 3876–3884. doi: 10.1109/ICCV.2017.416

Bohg, J., and Kragic, D. (2009). "Grasping familiar objects using shape context," in *Proceedings of the International Conference on Advanced Robotics* (Munich), 1–6.

Bohg, J., Morales, A., Asfour, T., and Kragic, D. (2014). Data-driven grasp synthesis–A survey. *IEEE Trans. Robot.* 30, 289–309. doi: 10.1109/TRO.2013.2289018

Boularias, A., Bagnell, J. A., and Stentz, A. (2015). "Learning to manipulate unknown objects in clutter by reinforcement," in *Proceedings of the AAAI Conference on Artificial Intelligence* (Austin), 1336–1342.

Cai, J., Cheng, H., Zhang, Z., and Su, J. (2019). "MetaGrasp: Data efficient grasping by affordance interpreter network," in *2019 International Conference on Robotics and Automation (ICRA)* (Montreal), 4960–4966. doi: 10.1109/ICRA.2019.8793912

Calli, B., Singh, A., Bruce, J., Walsman, A., Konolige, K., Srinivasa, S., et al. (2017). Yale-CMU-Berkeley dataset for robotic manipulation research. *Int. J. Robot. Res.* 36, 261–268. doi: 10.1177/0278364917700714

Chitta, S., Jones, E. G., Ciocarlie, M., and Hsiao, K. (2012a). Mobile manipulation in unstructured environments: perception, planning, and execution. *IEEE Robot. Autom. Mag.* 19, 58–71. doi: 10.1109/MRA.2012.2191995

Chitta, S., Sucan, I., and Cousins, S. (2012b). Moveit! [ROS topics]. *IEEE Robot. Autom. Mag.* 19, 18–19. doi: 10.1109/MRA.2011.2181749

Dang, H., and Allen, P. K. (2012). "Semantic grasping: planning robotic grasps functionally suitable for an object manipulation task," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems* (Vilamoura), 1311–1317. doi: 10.1109/IROS.2012.6385563

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). "ImageNet: a large-scale hierarchical image database," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Miami), 248–255. doi: 10.1109/CVPR.2009.5206848

Detry, R., Ek, C. H., Madry, M., and Kragic, D. (2013). "Learning a dictionary of prototypical grasp-predicting parts from grasping experience," in *Proceedings of the IEEE International Conference on Robotics and Automation* (Karlsruhe), 601–608. doi: 10.1109/ICRA.2013.6630635

Detry, R., Ek, C. H., Madry, M., Piater, J., and Kragic, D. (2012). "Generalizing grasps across partly similar objects," in *Proceedings of the IEEE International Conference on Robotics and Automation* (Saint Paul), 3791–3797. doi: 10.1109/ICRA.2012.6224992

Detry, R., and Piater, J. (2013). "Unsupervised learning of predictive parts for cross-object grasp transfer," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems* (Tokyo), 1720–1727. doi: 10.1109/IROS.2013.6696581

Do, T., Nguyen, A., and Reid, I. (2018). "Affordancenet: an end-to-end deep learning approach for object affordance detection," in *Proceedings of the IEEE International Conference on Robotics and Automation* (Brisbane), 5882–5889. doi: 10.1109/ICRA.2018.8460902

Fang, K., Bai, Y., Hinterstoisser, S., Savarese, S., and Kalakrishnan, M. (2018). "Multi-task domain adaptation for deep learning of instance grasping from simulation," in *Proceedings of the IEEE International Conference on Robotics and Automation* (Brisbane), 3516–3523. doi: 10.1109/ICRA.2018.8461041

Fischinger, D., Weiss, A., and Vincze, M. (2015). Learning grasps with topographic features. *Int. J. Robot. Res.* 34, 1167–1194. doi: 10.1177/0278364915577105

Florence, P. R., Manuelli, L., and Tedrake, R. (2018). "Dense object nets: Learning dense visual object descriptors by and for robotic manipulation," in *Proceedings of the Conference on Robot Learning* (Zurich), 373–385.

Hadsell, R., Chopra, S., and LeCun, Y. (2006). "Dimensionality reduction by learning an invariant mapping," in *Proceedings of the IEEE Conference on*

Computer Vision and Pattern Recognition (New York, NY), 1735–1742. doi: 10.1109/CVPR.2006.100

He, K., Zhang, X., Ren, S., and Sun, J. (2016). "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Las Vegas), 770–778. doi: 10.1109/CVPR.2016.90

Herzog, A., Pastor, P., Kalakrishnan, M., Righetti, L., Asfour, T., and Schaal, S. (2012). "Template-based learning of grasp selection," in *Proceedings of the IEEE International Conference on Robotics and Automation* (Saint Paul), 2379–2384. doi: 10.1109/ICRA.2012.6225271

Iqbal, S., Tremblay, J., To, T., Cheng, J., Leitch, E., Campbell, A., et al. (2019). Directional semantic grasping of real-world objects: from simulation to reality. *arXiv:1909.02075*.

James, S., Davison, A. J., and Johns, E. (2017). "Transferring end-to-end visuomotor control from simulation to real world for a multi-stage task," in *Proceedings of the Conference on Robot Learning* (Mountain View).

James, S., Wohlhart, P., Kalakrishnan, M., Kalashnikov, D., Irpan, A., Ibarz, J., et al. (2019). "Sim-To-Real via Sim-To-Sim: Data-efficient robotic grasping via randomized-to-canonical adaptation networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (Long Beach), 12619–12629. doi: 10.1109/CVPR.2019.01291

Jang, E., Devin, C., Vanhoucke, V., and Levine, S. (2018). "Grasp2vec: Learning object representations from self-supervised grasping," in *Proceedings of the Conference on Robot Learning* (Zurich), 99–112.

Jang, E., Vijayanarasimhan, S., Pastor, P., Ibarz, J., and Levine, S. (2017). "End-to-end learning of semantic grasping," in *Proceedings of the Conference on Robot Learning* (Mountain View), 119–132.

Jiang Y., Moseson, S., and Saxena, A. (2011). "Efficient grasping from RGBD images: learning using a new rectangle representation," in *Proceedings of the IEEE International Conference on Robotics and Automation* (Shanghai), 3304–3311. doi: 10.1109/ICRA.2011.5980145

Johns, E., Leutenegger, S., and Davison, A. J. (2016). "Deep learning a grasp function for grasping under gripper pose uncertainty," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems* (Daejeon), 4461–4468. doi: 10.1109/IROS.2016.7759657

Kalashnikov, D., Irpan, A., Pastor, P., Ibarz, J., Herzog, A., Jang, E., et al. (2018). "Scalable deep reinforcement learning for vision-based robotic manipulation," in *Proceedings of the Conference on Robot Learning* (Zurich), 651–673.

Kingma, D. P., and Ba, J. L. (2015). "Adam: A method for stochastic optimization," in *Proceedings of the International Conference on Learning Representations* (San Diego).

Klank, U., Pangercic, D., Rusu, R. B., and Beetz, M. (2009). "Real-time CAD model matching for mobile manipulation and grasping," in *Proceedings of IEEE-RAS International Conference on Humanoid Robots* (Paris), 290–296. doi: 10.1109/ICHR.2009.5379561

Kopicki, M., Detry, R., Adjigble, M., Stolkin, R., Leonardis, A., and Wyatt, J. L. (2016). One-shot learning and generation of dexterous grasps for novel objects. *Int. J. Robot. Res.* 35, 959–976. doi: 10.1177/0278364915594244

Kroemer, O., Ugur, E., Oztop, E., and Peters, J. (2012). "A kernel-based approach to direct action perception," in *Proceedings of the IEEE International Conference on Robotics and Automation* (Saint Paul), 2605–2610. doi: 10.1109/ICRA.2012.6224957

Kumra, S., and Kanan, C. (2017). "Robotic grasp detection using deep convolutional neural networks," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems* (Vancouver), 769–776. doi: 10.1109/IROS.2017.8202237

Lenz, I., Lee, H., and Saxena, A. (2015). Deep learning for detecting robotic grasps. *Int. J. Robot. Res.* 34, 705–724. doi: 10.1177/0278364914549607

Levine, S., Pastor, P., Krizhevsky, A., Ibarz, J., and Quillen, D. (2018). Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection. *Int. J. Robot. Res.* 47, 421–436. doi: 10.1177/0278364917710318

Li, Z., Wang, G., and Ji, X. (2019). "CDPN: Coordinates-based disentangled pose network for real-time RGB-based 6-DoF object pose estimation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision* (Seoul), 7678–7687. doi: 10.1109/ICCV.2019.00777

Liang, H., Ma, X., Li, S., Gorner, M., Tang, S., Fang, B., et al. (2019). "PointNetGPD: Detecting grasp configurations from point sets," in *Proceedings of the IEEE International Conference on Robotics and Automation* (Montreal), 3629–3635. doi: 10.1109/ICRA.2019.8794435

Liu, C., Fang, B., Sun, F., Li, X., and Huang, W. (2019). Learning to grasp familiar objects based on experience and objects' shape affordance. *IEEE Trans. Syst. Man Cybern.* 49, 2710–2723. doi: 10.1109/TSMC.2019.2901955

Mahler, J., Liang, J., Niyaz, S., Laskey, M., Doan, R., Liu, X., et al. (2017). "Dex-Net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics," in *Proceedings of Robotics: Science and Systems* (Cambridge). doi: 10.15607/RSS.2017.XIII.058

Mahler, J., Pokorny, F. T., Hou, B., Roderick, M., Laskey, M., Aubry, M., et al. (2016). "Dex-Net 1.0: A cloud-based network of 3D objects for robust grasp planning using a multi-armed bandit model with correlated rewards," in *Proceedings of the IEEE International Conference on Robotics and Automation* (Stockholm), 1957–1964. doi: 10.1109/ICRA.2016.7487342

Makhal, A., Thomas, F., and Gracia, A. P. (2018). "Grasping unknown objects in clutter by superquadric representation," in *Proceedings of the IEEE International Conference on Robotic Computing* (Laguna Hills), 292–299. doi: 10.1109/IRC.2018.00062

Manuelli, L., Gao, W., Florence, P. R., and Tedrake, R. (2019). kPAM: Keypoint affordances for category-level robotic manipulation. *arXiv:1903.06684*.

Morales, A., Chinellato, E., Fagg, A. H., and Pobil, A. P. D. (2004). Using experience for assessing grasp reliability. *Int. J. Human. Robot.* 1, 671–691. doi: 10.1142/S0219843604000290

Morrison, D., Leitner, J., and Corke, P. (2018). "Closing the loop for robotic grasping: A real-time, generative grasp synthesis approach," in *Proceedings of Robotics: Science and Systems* (Pittsburgh, PA). doi: 10.15607/RSS.2018.XIV.021

Mousavian, A., Eppner, C., and Fox, D. (2019). "6-DOF GraspNet: Variational grasp generation for object manipulation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision* (Seoul), 2901–2910. doi: 10.1109/ICCV.2019.00299

Park, K., Patten, T., Prankl, J., and Vincze, M. (2019a). "Multi-task template matching for object detection, segmentation and pose estimation using depth images," in *Proceedings of the IEEE International Conference on Robotics and Automation* (Montreal), 7207–7213. doi: 10.1109/ICRA.2019.8794448

Park, K., Patten, T., and Vincze, M. (2019b). "Pix2Pose: Pixel-wise coordinate regression of objects for 6D pose estimation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision* (Seoul), 7668–7677. doi: 10.1109/ICCV.2019.00776

Pinto, L., and Gupta, A. (2016). "Supersizing self-supervision: Learning to grasp from 50K tries and 700 robot hours," in *Proceedings of the IEEE International Conference on Robotics and Automation* (Stockholm), 3406–3413. doi: 10.1109/ICRA.2016.7487517

Prankl, J., Aldoma, A., Svejda, A., and Vincze, M. (2015). "RGB-D object modelling for object recognition and tracking," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems* (Hamburg), 96–103. doi: 10.1109/IROS.2015.7353360

Quigley, M., Conley, K., Gerkey, B. P., Faust, J., Foote, T., Leibs, J., et al. (2009). "ROS: An open-source robot operating system," in *Proceedings of the IEEE International Conference on Robotics and Automation, Workshop on Open Source Software* (Kobe).

Redmon, J., and Angelova, A. (2015). "Real-time grasp detection using convolutional neural networks," in *Proceedings of the IEEE International Conference on Robotics and Automation* (Seattle), 1316–1322. doi: 10.1109/ICRA.2015.7139361

Ronneberger, O., Fischer, P., and Brox, T. (2015). "U-Net: Convolutional networks for biomedical image segmentation," in *Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention* (Munich), 234–241. doi: 10.1007/978-3-319-24574-4_28

Rusu, R. B., Holzbach, A., Diankov, R., Bradski, G., and Beetz, M. (2009). "Perception for mobile manipulation and grasping using active stereo," in *Proceedings of IEEE-RAS International Conference on Humanoid Robots* (Paris), 632–638. doi: 10.1109/ICHR.2009.5379597

Saxena, A., Driemeyer, J., and Ng, A. Y. (2008). Robotic grasping of novel objects using vision. *Int. J. Robot. Res.* 27, 157–173. doi: 10.1177/0278364907087172

Song, D., Huebner, K., Kyrki, V., and Kragic, D. (2010). "Learning task constraints for robot grasping using graphical models," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems* (Taipei), 1579–1585. doi: 10.1109/IROS.2010.5649406

Srinivasa, S. S., Ferguson, D., Helfrich, C. J., Berenson, D., Collet, A., Diankov, R., et al. (2010). HERB: A home exploring robotic butler. *Auton. Robots* 28, 5–20. doi: 10.1007/s10514-009-9160-9

ten Pas, A., Gualtieri, M., Saenko, K., and Platt, R. (2017). Grasp pose detection in point clouds. *Int. J. Robot. Res.* 36, 1455–1473. doi: 10.1177/0278364917735594

Tremblay, J., To, T., Sundaralingam, B., Xiang, Y., Fox, D., and Birchfield, S. T. (2018). "Deep object pose estimation for semantic robotic grasping of household objects," in *Proceedings of the Conference on Robot Learning* (Zurich), 306–316.

Wang, C., Xu, D., Zhu, Y., Martín-Martín, R., Lu, C., Fei-Fei, L., et al. (2019). "DenseFusion: 6D object pose estimation by iterative dense fusion," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (Long Beach), 3338–3347. doi: 10.1109/CVPR.2019.00346

Wang, F., and Hauser, K. (2019). "In-hand object scanning via RGB-D video segmentation," in *Proceedings of the IEEE International Conference on Robotics and Automation* (Montreal), 3296–3302. doi: 10.1109/ICRA.2019.8794467

Wang, H., Sridhar, S., Huang, J., Valentin, J., Song, S., and Guibas, L. J. (2019). "Normalized object coordinate space for category-level 6D object pose and size estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (Long Beach), 2637–2646. doi: 10.1109/CVPR.2019.00275

Wang, Z., Li, Z., Wang, B., and Liu, H. (2017). Robot grasp detection using multimodal deep convolutional neural networks. *Adv. Mech. Eng.* 8, 1–12. doi: 10.1177/1687814016668077

Wohlhart, P., and Lepetit, V. (2015). "Learning descriptors for object recognition and 3D pose estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Boston), 3109–3118. doi: 10.1109/CVPR.2015.7298930

Xiang, Y., Schmidt, T., Narayanan, V., and Fox, D. (2018). PoseCNN: "A convolutional neural network for 6D object pose estimation in cluttered scenes," in *Proceedings of Robotics: Science and Systems* (Pittsburgh, PA). doi: 10.15607/RSS.2018.XIV.019

Yamamoto, T., Terada, K., Ochiai, A., Saito, F., Asahara, Y., and Murase, K. (2019). Development of human support robot as the research platform of a domestic mobile manipulator. *ROBOMECH J.* 6, 1–15. doi: 10.1186/s40648-019-0132-3

Zakharov, S., Shugurov, I., and Ilic, S. (2019). "DPOD: 6D pose object detector and refiner," in *Proceedings of the IEEE/CVF International Conference on Computer Vision* (Seoul), 1941–1950. doi: 10.1109/ICCV.2019.00203

Zeng, A., Song, S., Nießner, M., Fisher, M., Xiao, J., and Funkhouser, T. (2017). "3DMatch: Learning local geometric descriptors from RGB-D reconstructions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Honolulu), 199–208. doi: 10.1109/CVPR.2017.29

Zeng, A., Song, S., Welker, S., Lee, J., Rodriguez, A., and Funkhouser, T. (2018a). "Learning synergies between pushing and grasping with self-supervised deep reinforcement learning," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems* (Madrid), 4238–4245. doi: 10.1109/IROS.2018.8593986

Zeng, A., Song, S., Yu, K., Donlon, E., Hogan, F. R., Bauza, M., et al. (2018b). "Robotic pick-and-place of novel objects in clutter with multi-affordance grasping and cross-domain image matching," in *Proceedings of the IEEE International Conference on Robotics and Automation* (Brisbane), 3750–3757. doi: 10.1109/ICRA.2018.8461044

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Copyright © 2020 Patten, Park and Vincze. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.