ELSEVIER

Computer Vision and Image Understanding

journal homepage: www.elsevier.com

Attention mechanism for collision-free grasp detection from 3D point clouds

Dinh-Cuong Hoang[a,**], Bao-Long Tran[a]

[a]*ICT Department, FPT University, Hanoi, Vietnam*

ABSTRACT

Grasp detection is a challenging and important task in robotics and computer vision. Many existing methods require time-consuming multi-stage processing for sampling grasp candidates and evaluating the grasp quality. While several works proposed end-to-end models for 6-DOF grasp detection and achieved state-of-the-art results in benchmarks. However, most of these models treat all points in a scene equally without focusing on the more relevant regions, which greatly harms the speed and accuracy. Inspired by the success of the attention mechanism in various computer vision tasks, this work discovers the power of the attention mechanism to boost the performance of grasp detection from 3D point clouds. To achieve this, taking the recent VoteGrasp (Hoang et al. (2022)) as a basic pipeline, we integrate different attention modules into the end-to-end grasp detection network and provides insights into the potential of these modules.

1. Introduction

Conventional methods for grasp detection consider accurate estimation of 6D pose as a fundamental. These approaches typically consist in recovering correspondences between CAD models and object point clouds to register grasps from pre-computed database (Muñoz et al. (2016), Zeng et al. (2017)). The grasps are previously calculated for CAD models and then to be regressed to determine high-quality one for objects (Bohg et al. (2013)). Some approaches obtain 6D pose estimation by employing algorithms such as iterative closest point (Besl and McKay (1992)) to align models to point clouds or exploiting local descriptors such as SIFT key points (Dias et al. (2014)) for 3D matching. Whereas, recent approaches leverage the advance of deep learning technique to design end-to-end models for predicting 6DoF rotation and translation vector in 3D space (Wu et al. (2019), Wang et al. (2019)). In general, however, the heavy dependencies of these methods on 3D CAD models, which might be not always available for all objects or might require intensive labor in the pre-scanning process, restricts their ability to be widely applied in all scenarios.

This motivated alternative approaches that directly detect grasps from sensor data thanks to CNN without estimating object pose. In other words, instead of registering point clouds to a pre-computed dataset and indexing grasps, these methods learn to identify a set of candidates and the probability of success of grasps. In recent works, (Mahler et al. (2017), Mahler et al. (2018)) apply deep CNNs to find features, while others (Redmon and Angelova (2015), Lenz et al. (2015)) develop end-to-end learning for estimating possible grasps. The results of learning-based approaches allow to grasp pre-known objects, which might be partially occluded, an unknown pose as well as fully novel objects.

**Corresponding author: Tel.: +0-000-000-0000; fax: +0-000-000-0000;
e-mail: hoangcuongbk80@gmail.com; cuonghd7@fe.edu.vn
(Dinh-Cuong Hoang)

Grasp detection area differs from object detection and pose estimation due to it requires both determining grasp candidates and maximining the probability of success of grasps. To generate quality candidates, which are understood as reasoning parts of objects for the gripper to appropriately operate, it seems to be inadequate to consider only 2D or 2.5D local features of objects since focusing on visual similarities might lead to failure in some cases, especially in occlusion context and texture-less objects. Therefore, a few approaches (ten Pas et al. (2017), Mousavian et al. (2019), Liang et al. (2019), Fang et al. (2020)) take 3D geometry analysis into account to robust the performance of predicting grasp by localizing grasp from 3D point sets.

On the other hand, achieving highly feasible grasps is a challenging problem that researchers have to cope with. The successful grasps are considered to be not involved with undesirable contacts with surroundings. This is hardly achievable because of measurement noise in acquiring data, occlusion environments, and cluttered scenarios. In attempt to deal with this problem, several methods intensively collect numerous grasp candidates, while others propose end-to-end models for 6-DOF grasp detection. These end-to-end approaches achieve state-of-the-art result in benchmarks thanks to employing backbone networks such as PoinNet++ (Qi et al. (2017b)) to take advantage of local property. Although these strategies could gain high-grade grasping points, they find hard to avoid unexpected collisions. Solely considering regional information of each candidate without examining the spatial relationship of neighboring candidates appears to be insufficient for system to be fully aware of context. Thus, the contextual information is a prospective material to be succeed in accomplishing collision-free grasps.

Motivated by the above perspective, we introduce an effective end-to-end model for 6-DOF grasp detection. It accomplishes highly competitive grasp configurations in severe scenarios of occlusion when compared with others in benchmarks. This success is attributed to the essence of leveraging both a voting mechanism architecture (Qi et al. (2019)) to elect candidates and a context learning module to encode the spatial relationship of neighboring candidates into feature vectors. Voting mechanism allows our model to widen our model's capability to beforehand unknown objects. Besides, fusing contextual spatial features into local features enables our method to enhance the performance in alleviating potential collision. The following highlights the main contributions of our work:

- Our research proposes a new robust framework VoteGrasp combing voting mechanism and contextual learning module for 6-DOF grasp detection and achieves remarkable results while operating in severe scenes of occlusions. Besides, our proposal demonstrates its generalization capability to novel objects.
- We develop a context learning module that contributes the spatial dependency of objects in candidates vicinity to the feature vectors to learn collision-free grasps.
- Experiments compare the results of attention modules to find out what is most robust and suitable to our model.

2. Related work

2.1. 3D Point Cloud Based Grasp Detection

Robotic grasping is conventionally involved in two related problems of perception and planning. The perceptual component aims to acquire the position and orientation of the object to be grasped. The planning component considers how to evaluate a good manipulation. This primitive idea comes to 3D retrieval methods that retrieve segmented point clouds to 3D CAD models to estimate the object poses and then decide grasps from a pre-defined grasp dataset. As mentioned in previous sections, this old-fashioned approach is impractical in some cases due to the problem of the existence of all accurate CAD models. Furthermore, these methods could not detec grasps for novel objects outside of the dataset.

To overcome these issues, machine learning-based approaches have been introduced to directly detect grasp from sensor data without estimating object pose with a conventional pipeline of grasp sampling process, extracting features of the

grasps, and evaluating the quality of grasps. (ten Pas et al. (2017)) proposed grasp pose detection (GPD) algorithm from point clouds. Unlike (Herzog et al. (2012)), which necessarily segments the object from the background, this method first identifies a ROI that could include multiple objects or even background to find a grasp. The large set of grasp candidates is interested to be found in ROI with respect to two conditions of not being in collision with the point cloud and there being no contact between grippers and point cloud. Then, a four-layer CNN decides whether or not a candidate is a grasp for observed and occluded surfaces acquired from the depth sensor. As an extension of PGD idea, PointNetGPD (Liang et al. (2019)) replaces the CNN-based evaluation model with a new network based on architecture of PointNet (Qi et al. (2017a)). Taking advantage of PointNet architecture, this evaluation model can directly perform geometry analysis, which the original PGD idea lacks, from the 3D point cloud and therefore can detect more reliable grasps. However, these approaches could not provide good enough grasp assessments, they need exhaustively manual sampling grasps, which sometimes is hard to acquire when the raw point cloud is spare. This motivates the method (Mousavian et al. (2019)) to introduce two network architectures for both sampling and evaluating grasps. Instead of manually sampling grasps, this one utilizes a variational auto-encoder (VAE) model to generate a diverse set of grasps as well as limit the number of failing grasps. Detecting grasps is important because not all grasps are kinematically feasible or collision-free for the manipulation of the robot. In terms of the evaluation model, this method not only classifies each grasp but also iteratively refines a significant portion of the rejected grasps to the successful ones. Nevertheless, all these approaches solely depend on the local visible parts of objects, which occasionally is imperfect due to the noisy depth value. Some methods sort to use high-quality depth sensors or utilize the multi-view technique (ten Pas et al. (2017)) to provide high-success rates of grasps.

The lack of geometric information about candidates surroundings and information about the scene mitigates the performance of the above methods. Perceiving this remaining issue, end-to-end methods (Fang et al. (2020), Ni et al. (2020)) take the whole point clouds as input and neglect the tradition pipeline of time-consuming grasp sampling process. They combine global data information to directly predict the poses and qualities of spatial grasps. Their PointNet++-based architectures allow them to immediately extract local spatial features from the raw data point clouds, while others such as (Choi et al. (2018)) employs a three-dimensional deep learning neural network to deal with voxelized point clouds, which might be less precise in detail due to voxelization. Although considering whole point clouds, these approaches dont take the relationship between objects into account and only focus on local representations of points. As a result, their performance in cluttered environments is unreliable. We address the remaining challenges by leveraging voting mechanisms combined with contextual information to ensure the generalization and reliability of grasps.

2.2. Hough Voting in Object Detection

The Hough Transform was originally informed to detect defined shapes in 2D space such as lines, circles, or eclipses (Hough (1959), Hough (1962), Duda and Hart (1972)). This technique is limited to shapes characterized by a small number of parameters. The Generalized Hough Transform is then introduced to extend the application of the primitive algorithm to arbitrary shapes. It, therefore, is widely extended and applied to computer vision tasks including object detection (Gall and Lempitsky (2013), Gall et al. (2011)), motion detection (Gall et al. (2011), Kalviainen (1996)), medical imaging (Golemati et al. (2006)), and robot navigation (Iocchi et al. (2001)). In terms of 3D scenes, methods (Deng et al. (2018), Rabbani and Van Den Heuvel (2005)) utilizes the original Hough Transform formulation in a straightforward way to deal with 3D analytical shapes like spheres and cylinders. However, these methods cannot be applied to generic free-form objects, which are common in realistic applications.

More recently, several methods (Silberberg et al. (1984), Tombari and Di Stefano (2010)) widen the use of Hough voting mechanisms to 3D object detection. (Tombari and Di Stefano (2010)) even proves the robustness of their approach in scenes

with a significant degree of occlusions. They aim to deploy 3D features of interesting point, which is chosen randomly or extracted by means of feature detectors, to compute the correspondences between 3D models and the current scene. The features of each point do not normally consist of 3D properties but include points relative spatial relationship with respect to centroids of 3D models. In that way, correspondences can cast a vote in 3D Hough space to accumulate evidence for feasible centroids in the scene. If enough features vote for the presence of the centroid of an object, the object is determined. Though this method performs well in cluttered scenarios, it requires the existence of 3D models, which are not always available in practice.

Deep learning technique allows (Kehl et al. (2016)) to generalize the use of voting mechanisms to novel objects. In the training phase, this research densely samples scale-invariant RGB-D patches from synthetic views of fixed size. Features of each patch is learned by CNNs and local votes describing the patch 3D center point offset to the object centroid are stored together into a codebook. In the practicing phase, patches from real data are fed into neural networks to regress features for a k-NN search in a pre-computed codebook. If the feature distance of retrieved nearest neighbors of patches is smaller than a certain thresh, these patches are understood to cast 6D votes. This work furthermore uses a vote filtering process to refine votes and reject implausible ones, so that detection results are more reliable.

A proposed end-to-end model VoteNet (Qi et al. (2019)) recently reaches state-of-the-art results in 3D point cloud detection from real 3D scans because of some main reasons. Firstly, it directly learns 3D features from raw point cloud data by adopting PointNet++ backbone to output a set of seed points. Whereas, (Liu et al. (2016)) learns 2D descriptors from RGB-D images and (Song and Xiao (2016), Hou et al. (2019)) require regularizing point clouds such as voxelization to learn features so that they ignore or sacrifice sufficient spatial information. Secondly, while (Kehl et al. (2016)) determines votes by looking up a pre-computed codebook, VoteNet generates votes by leveraging a shared deep network-based voting module. This approach is more efficient due to votes are trained jointly with the rest of the pipeline compared with (Kehl et al. (2016)) storing votes of each patch of images independently into a codebook. All these improvements allow this model to directly vote for virtual centroids of objects and achieve high-quality 3D object proposals. Inspired by the success of VoteNet, we leverage its voting architecture to strengthen grasp detection to occlusion.

2.3. Context and Attention in 3D point clouds

Contextual information is essential to be precisely aware of a particular location. Much research employs the use of contextual perception to improve the performance of computer vision tasks in 3D scenarios such as 3D point matching (Deng et al. (2018)), point cloud semantic segmentation (Ye et al. (2018)), instance segmentation of 3D point clouds (Hu et al. (2018b)), and 3D scene layout prediction (Shi et al. (2019)). Differ from conventional methods, which purely take local geometric features, these methods fuse global features including points and normals within a local vicinity into learned local descriptors to produce more discriminative local representations. As a result, challenging tasks for 3D perception are robustly solved. However, numerous methods making the use of contextual information equally treat neighbors of a location so that they cannot precisely reflect on the relationships between points in 3D contexts.

In order to cope with above issues, the attention mechanism is leveraged to meticulously investigate the dependency of a local position on each neighbor. In other words, instead of assuming that all neighbors have the same impact on a local representation, attention mechanism computes how much each neighbor affects a particular local feature. (Xie et al. (2018)) proposes ShapeContextNet which combines the attention idea with the concept of shape context to be applied in point cloud classification and segmentation. The shape context idea designs a discriminative descriptor with spatially inhomogeneous cells. This descriptor is actually a feature vector (histogram) that captures neighborhood information by counting the number of neighbor-

ing points in each cell. The descriptor is combined with self-attention idea to inform Attentional ShapeContextNet, which is the main contribution of this research. In terms of place recognition, (Zhang and Xiao (2019)) informs Point Contextual Attention Network to effectively handle this problem. This method utilizes contextual information and per-point local feature by adding attention networks to PointNetVLAD to produce the attention map that estimates an attention score for each point. By leveraging attention mechanism, this network can predict the significance of each local point and therefore pay more attention to the most informative points. (Paigwar et al. (2019)) perceives that although PointNet performs a fascinating result in 3D object detection, it is limited to the points in point clouds. Thus, this method develops Attentional PointNet for 3D object detection in spacious contexts taken from the LiDAR sensor. Instead of processing the whole point cloud, it learns to find possible locations of objects of interest. Utilizing attention mechanism facilitates this approach to sequentially attend to relevant smaller regions in a large point cloud so that meets the significance of both detection results and inference time. These successes inspire us to stipulate that incorporating contextual information and attention theory is prospective to our problem of interest.

3. Proposed method

In this work, we investigate the profound contribution of attention mechanism to grasp detection area. Based on the recent successful end-to-end model VoteGrasp (Hoang et al. (2022)), we examine a set of attention modules by integrating each of them into VoteGrasp. In this way, we convey insights about their essence to grasp detection as well as their operations. Concretely, the architecture of our proposal will be clarified in 3.1 and the attention modules will be deeply discussed in 3.2.

3.1. VoteGrasp

VoteGrasp is designed based on voting mechanism to overcome the obstacle of detecting grasps in occlusion and cluttered environments. Fig.1 illustrates the overall architecture of VoteGrasp. Given a point cloud input of size $N \times 3$, our method outputs a set of potential grasps, in which each grasp $G = (p, R, w, q)$ is accompanied by a center point $p = (x, y, z) \in \mathbb{R}^3$, gripper orientation $R \in SO(3)$, gripper width $w \in \mathbb{R}$, and grasp score $q \in [0, 1]$. In terms of gripper estimation, we reformulate R as in (Kehl et al. (2017)) instead of directly regressing it because of non-linearlity of the rotation space (Peng et al. (2019)).

Backbone Network: We take advantage of PointNet++ architecture as our backbone to extract geometric features. PointNet++ consists of set learning layers to combine features from multiple scales therefore able to enrich local features with increasing contextual scales. This backbone network prefers M seed points and extracts high-dimensional features $\{s_i\}_{i=1}^{M}$ where $s_i = [x_i, f_i]$ is feature of a seed point specified by seed location in 3D space $x_i \in \mathbb{R}^3$ and feature vector $f_i \in \mathbb{R}^F$.

Vote and Cluster: The M seed points are used as materials for computing votes for grasps. Each vote is characterized by a grasp center point and a feature vector for learning grasp. To obtain this, a multi-layer perceptron (MLP) containing fully connected layers, ReLU, and batch normalization is employed to compute J votes per seed. This allows us to estimate multiple grasp poses for each object. We collect a set of votes $\{\{v_{ij} = [y_{ij}, g_{ij}] \in \mathbb{R}^{3+F}\}_{i=1}^{M}\}_{j=1}^{J}$. Here v_{ij} is j^{th} vote in J set votes at i^{th} point in M seed points. y_{ij} and g_{ij} (F-dimensions) represent grasp center and feature vector learned for final grasp detection of vote v_{ij} correspondingly. The next vital step is to cluster the votes by uniform sampling and finding neighboring votes within a certain Eclidean distance. Basing on a grasp center y_i from a vote $\{v_i = [y_i, g_i] \in \mathbb{R}^{3+F}\}_{i=1}^{M \times J}$ calculated in previous step, we use iterative farthest point sampling (FPS) to select a subset of K votes $\{v_{ik}\}_{k=1}^{K}$ in the neighborhood. A ball query finds K neighboring votes v_{ik} within a radius vacinity. The output are K groups of vote sets of size $K \times n_k \times (3 + F)$, where each group elects a grasp center and n_k is number of neighbors of vote v_{ik}.

Context learning: In cluttered environments, grasping is inherently challenging because a successfull grasp has to be

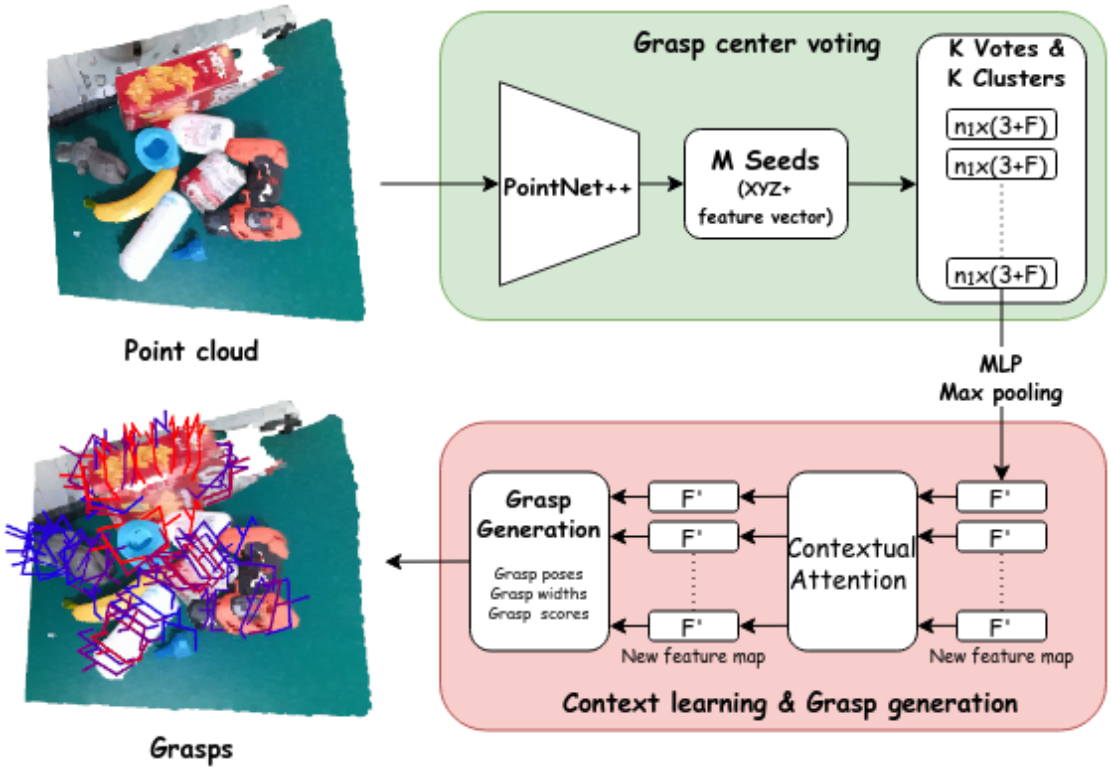


Fig. 1: VoteGrasp model using voting architecture and attention module for 6-DOF grasp detection in point cloud data. Our model conducts several self-attention modules discussed in section 3.2 following Hough voting network (Qi et al. (2019)) to learn contextual information. Green grasps refer to highest quality grasps and red ones refer to lowest quality grasps.

aware of both invisible object parts and potential collisions. The relationships between objects in the scene play an essential role in detecting collision-free grasps. Therefore, the correlations between objects and contextual information outside of interest regions are encoded into features to facilize this critical information to be learned. However, VoteNet (Qi et al. (2019)) is originally designed to detect objects independently thanks to grouping votes which respond to one object centroid. Each cluster C_k is fed into MLP layers to immediately regress its object class and bounding box. In our work, instead process each cluster instantly, we compute a new feature map by attaching relationship information between all clusters. Inspired by self-attention-based models (Vaswani et al. (2017),Xie et al. (2018), Wang et al. (2018), Fu et al. (2019)), we integrate a contextual module into our framework to acquire interdependencies between clusters. In more detail, votes $\{v_i = [y_i, g_i] \in \mathbb{R}^{3+F}\}_{i=1}^{n_k}$ in each clusters K are firstly fed into a MLP and max-pooling layer to aggregate a single feature vector $C_k \in \mathbb{R}^{F'}$. Summarizing all these single vector of K clusters, we gain a feature map $C = [C_1; C_2; ...; C_K] \in \mathbb{R}^{F \times F'}$. The next step is to learn correlations between clusters in C by conducting a context learning module. New rich contextual feature map is generally formulated as Eq.1. The particular form of formulation depends on the certain attention module applied and it is deeply discussed in 3.2.

$$C_i^{context} = \sum f(\theta(C_i), \psi(C_j)) \odot g(C_j) \tag{1}$$

Where $\theta(.)$, $\psi(.)$, $g(.)$ are learnable transformations and $f(i)$ is relation function to encode relation between all positions. The widespread relation function is the dot-product family, but in our research, we further examine other relation functions. The new feature map $C_{context} = [C_1^{context}; C_2^{context}; ...; C_K^{context} \in \mathbb{R}^{K \times K'}$ has same size with input feature. By leveraging self-attention mechanism, our network enables features of different clusters to communicate with each other. The contribution and effectiveness of the context learning module will be thoroughly illustrated in later sections.

Grasp Detection: New feature map $C^{context}$, which is enlarged with relationships between clusters, is then passed through a multi-layer perceptron (MLP) network to detect a ranked list of grasps $G = (p, R, w, q)$. More specifically, our model detects a grasp center $p = (x, y, z) \in \mathbb{R}^3$, gripper orientation $R \in SO(3)$, gripper width $w \in \mathbb{R}$, and grasp quality $q \in [0, 1]$. The MLP is implemented with 3 fully connected layers and two first of them are followed by batch normalization and ReLU. The last one - the prediction layer has $5 + V + 2A$ channels, in particular they are 3 grasp center regression values, 1 gripper width regression value, 1 grasp cinfidence regression value, V viewpoint scores, A angle scores (in-plane rotation), and A angle residual regression values (in-plane rotation). V and A represent the numbers of sampled viewpoints and in-plane rotations respectively.

Loss Function: The grasp detecton is supervised with multi-task loss:

$$L_{votegrasp} = L_{vote} + L_{grasp} \tag{2}$$

The VoteGrasp loss $L_{votegrasp}$ includes a voting loss L_{vote} and a grasp estimation loss L_{grasp}. Voting loss is built as a regression loss:

$$L_{vote} = \frac{1}{M_s} \sum_i \|y_i - c_i^g\|_H \cdot \mathbb{1}(x_i) \tag{3}$$

Where M_s denotes the total number of seeds on the object surface, c_i^g is the closest ground truth grasp center, $\|\cdot\|_H$ is the Huber norm and $\mathbb{1}(\cdot)$ is a binary function that indicates whether or not a seed point s_i belongs to an object. In terms of grasp loss function, it is defined as follows:

$$L_{grasp} = L_center + \alpha L_{rot} + \beta L_{width} + \gamma L_{score} \tag{4}$$

The grasp loss consists of a grasp center loss (regression) L_{center}, a rotation loss L_{rot}, a gripper width loss (regression) L_{width}, and a grasp confidence score (regression) L_{score}. The grasp center loss is conducted with two elements $L_{center} = L_{viewpoint} + L_{in-plane}$. While $L_{viewpoint}$ represents for viewpoint classification, the in-plane rotation estimation is designed as a combination of classification and regression as $L_{in-plane} = 0.1L_{angle-cls} + L_{angle-reg}$ (Qi et al. (2018)). We conduct $L1$-smooth loss (Ren et al. (2015)) for all regression ingredients and standard cross entropy for classification losses.

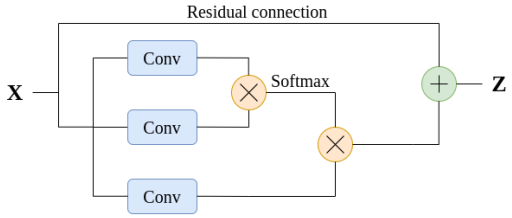


Fig. 2: Non-local block diagram. The input feature is transformed by $1 \times 1 \times 1$ convolutions. " $\otimes$ " denotes matrix multiplication, and " $\oplus$ " denotes element-wise sum.

3.2. Attentions

In this section, we want to thoroughly discuss attention modules that we employ in our research. This provides insights into how attention mechanism operates in each module and its potentiality in collision-free grasp detection. We will examine how differently they capture interdependencies inside feature map and operate relation function. The detail of implementation and results of plugging each module into VoteGrasp are evaluated in section 4.

Non-local (Wang et al. (2018)): The idea of this method is inspired by non-local means algorithm (Buades et al. (2005)) for denoising images. Non-local means computes the denoised value at a position as a weighted average of all pixels in the image. The family of weights between two pixels depends on the similarity between them. Non-local Neural Network forms this idea into deep stacks of convolutional operations to capture long-range dependencies in sequential data.

$$y_i = \frac{1}{C(x')} \sum_{\forall j} f(x_i, x'_j) g(x'_j) \tag{5}$$

$$z_i = W_z y_i + x_i \tag{6}$$

Eq.5 computes responses y based on relationships between different locations. It is proved to be a generic non-local operation due to the fact that it takes all positions $\forall j$ into consideration. x is the input signal, which could be images, videos, but often features. To reduce the computation, the authors use x'

as a subsampled version of x by using pooling technique (max pooling). z is the output feature that has variable sizes as input x and therefore keeps the positional correspondences of input data to be intact. i is the index of an output position and j is the index that enumerates all positions. f is the pairwise function (relation function) that represents the relationship between i and all j. In this research, authors examine the use of several pairwise functions of Gaussian, Embedded Gaussian, Dot product, Concatenation, or simply done by matrix multiplication as shown in Fig.2. g represents the input signal at position j. $C(x')$ is normalizing factor and softmax function is used here. The non-local operation in Eq.5 is wrapped into a non-local block by adding residual connection component $+x_i$ as define in Eq.6. This residual connection facilitates non-local block to insert a new one into the pre-trained model and does not interrupt its initial weight matrix W_z. This lightweight and easy-to-implemented block is plugged into many architectures such as classification, and segmentation to learn where they should pay more attention.

Criss-cross (Huang et al. (2019)): Similar to Non-local block, this building block is conducted to capture pixel-wise dependencies between all positions in images. Non-local block performs significantly in classification tasks. Nonetheless, confronting dense prediction tasks such as segmentation, which heavily requires high resolution of feature maps, pressures this method to reveal its weakness in consuming huge computation complexity and GPU memory. Therefore, criss-cross attention module harvests contexts in images criss-cross paths to obtain dense contextual information while keeping feature maps to be lightweight. In other words, this architecture prefers aggregating contextual information in horizontal and vertical directions to averaging all pixels. As a result, this is 11× faster and saves 85% FLOPs in comparison with non-local block. Given an input feature map $X \in \mathbb{R}^{H\times W}$, non-local produces $H \times W$ weights, while criss-cross method generates $H + W - 1$ weights. However, the pixels not in a criss-cross path are still absent in the aggregation of one pixel. The authors use ($R = 2$) loops of criss-cross attention modules to guarantee full-image pixels are harvested to produce feature maps with dense contextual information. Adding this block to architectures allows them to boost their performance in both result accuracy and inference time.

Squeeze-and-Excitation (SE) (Hu et al. (2018a)): This research strengthens the power of the convolutional operator by exploiting the interdependencies between the channels of convolutional features. Instead of focusing on spatial relationships like non-local and criss-cross network, it looks into another aspect the relationship between channels. This block consists of two modules: Squeeze and Excitation. Squeeze operation aims to produce channel descriptors that embed global distribution feature responses. Using global average pooling allows this module to generate channel-wise statistics y. Given a input feature map $X \in \mathbb{R}^{H\times W\times C}$ and $x_c \in R^{H\times W}$ is the feature map at c^{th} channel. A c^{th} element of y is calculated at c^{th} channel of input data as following formula. Thus, $Y \in \mathbb{R}^C$ is a collection of average representation of all pixels of all channels.

$$y_c = F_{sq}(x_c) = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} x_c(i, j) \tag{7}$$

$$s = F_{ex}(y, W) \tag{8}$$

$$z_c = s_c y_c \tag{9}$$

The next operation Excitation aggregates the information in the previous one. This one focuses on totally gathering channel-wise dependencies. To obtain this purpose, it makes use of a simple self-gating mechanism that takes embedding features, which is the output of Squeeze operation, as input to generate per-channel modulation weights $s \in \mathbb{R}^C$ as Eq.8. At the end of the operation, input feature map X is rescaled with modulation weight s. The final feature map $Z \in \mathbb{R}^{H\times W\times C}$ can indicate to what extent each channel is informative. This output map can be used by subsequent layers and help them learn what they should look at. This block is proved to be insertable to neural network architectures to boost their performance in scene classification and object detection.

Compact Generalize Non-local (CGNL) (Yue et al. (2018)): This compact block is designed to achieve high accurate object recognition, especially in fine-grained objects and actions. The research notices the lack of considering interac-

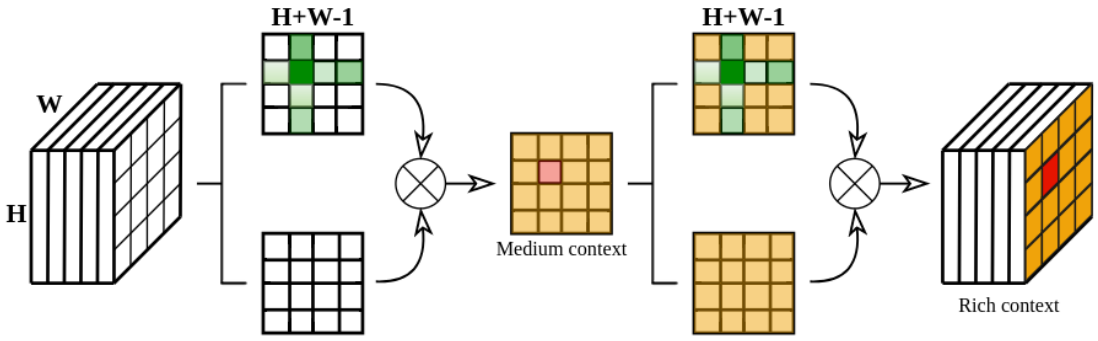


Fig. 3: Criss-cross block architecture captures interdependencies in vertical and horizontal directions. The block consists of two criss-cross operations to capture interdependencies of all pixels. The differences in shades of green represent different meanings that a pixel contributes to the target pixel (red). Similarly, the differences in shades of yellow illustrate the wealth of contextual information.

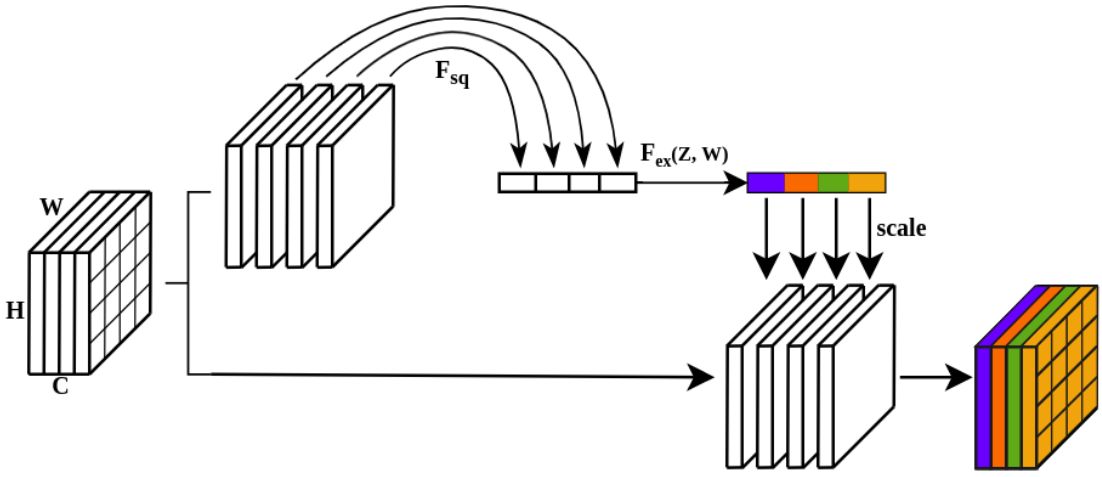


Fig. 4: Squeeze-and-Excitation block focuses on the channel relationship of input feature. Different colors depict the distribution of channel information.

tions between positions across channels of non-local module or even criss-cross module. They solely capture dependencies between spatial pixels and temporal frames by merging channels. Hence, they pay more attention to object part relations but neglect crucial clues for recognizing actions - the interactions between objects, which correspond to different channels. Although SE network mentioned previously learns interdependencies between channels, it treats all positions in a similar way by global averaging as Eq.7. Therefore, to acquire long-range correlations as well as interactions, CGNL learns correlations among all elements across the channels by merging channels into positions. It could be clearly understood that this method fuses channel information into positional features. The input feature map $X \in \mathbb{R}^{H\times W\times C}$ is firstly divided into G groups and each sub-feature map X' contains $C' = C/G$ channels. In comparison with 5, CGNL reshapes the output of transformation function to $HWC' - D$ vector column as Eq.10 to fuses channel into position.

$$\theta(X') = vec(X'W_\theta) \in \mathbb{R}^{HWC'} \tag{10}$$

$$\phi(X') = vec(X'W_\phi) \in \mathbb{R}^{HWC'} \tag{11}$$

$$g(X') = vec(X'W_g) \in \mathbb{R}^{HWC'} \tag{12}$$

$$vec(Y') = f(\theta(X'), \phi(X'))g(X') \tag{13}$$

Pairwise function (relation function) $f : \mathbb{R}^{HWC'} \times \mathbb{R}^{HWC'} \rightarrow \mathbb{R}^{HWC'} \times \mathbb{R}^{HWC'}$ can distinguish pairs of same location but at different channels so that can enrich greatly feature map for action recognizing or fine-grained object classification. Each feature map Y' computed from each group is then concatenated along the channel dimension to restore Y. By informing this formulation, this module produces competitive or state-of-the-art result

on benchmark datasets.

Dual Attention Network (DANet) (Fu et al. (2019)): Methods discussed above solely use either position attention or channel attention. While position attention selectively calculates the feature of each pixel by a weighted average of all pixels, channel attention selectively emphasizes interdependencies between channels among all channel maps. Despite the fact that dual attention network further improves the discriminative representation of feature map by applying self-attention mechanism for both positional and channel dependencies. To obtain this, it feeds input data through two parallel modules of position attention and channel attention to capture long-range contextual information. The attention module commonly operates as the attentional component in previously reviewed methods. In the research, the authors examine the contribution of each attention module to the precise segmentation results. They perceive that the absence of any module would harm the accuracy of prediction task. Using both modules outcome highest Mean IoU indicator in scene segmentation among applying none, solely position, or solely channel attention. It could be interpreted that dual attention network allows architectures to emphasize where and what to be most meaningful.

Convolutional Block Attention Module (CBAM) (Woo et al. (2018)): Similar to DANet for upgrading discriminative features, this module employs both spatial and channel information for attention mechanism. However, this one is different from DANet that it conducts spatial and channel attention modules in a sequential arrangement. Research also investigates diverse combinations of these two attention modules: placing the channel module first, placing the spatial module first, placing them in parallel, and placing them in sequence. Practical experiments reveal that organizing the spatial module following up channel module achieves the lowest error rate. Moreover, plugging into pre-existing architecture such as MobileNet, Faster-RCNN, StairNet, SSD, and CBAM reach slight successful detection in comparison with Squeeze-and-Excitation block.

Point-Attention (Feng et al. (2020)): Attention mechanism obviously demonstrates its strength in upgrading the performance of prediction architectures. However, most of them are being used for 2D data due to applying for 3D data is not straightforward. 3D point clouds inherently are irregular, sparse, unordered, and non-grid structures. Immediately exploiting the above methods for 3D point clouds might require voxelization which losses informative components of input data. Information about neighbors is crucial in attention operation so that the process of converting point clouds into a discrete grid appears to be destructive to final results. Therefore, Point Attention Network (PAN) is introduced to employ self-attention mechanism while directly dealing with 3D point clouds. It captures neighbors contextual correlation in multi-directions to enrich local shape features by layers called Local Attention-Edge Convolution (LAE-Conv). Then, the following point-wise spatial attention module obtains long-range spatial contextual dependencies to achieve more precise segmentation. In 3D point set, some points far away from a particular point pi appear to be not meaningful to the representation of p_i. Therefore, PAN conducts LAE-Conv to search for meaningful neighbors beforehand to enhance the local representation of point p_i. A ball space of a central point with r radius is formed and divided into K uniform bins. Radius r is selected to ensure that each bin contains at least m neighbors and only m nearest neighbors contribute to the representation of the central point. Experiments yield $K = 16$ and $m = 1$ gain the highest accuracy. In comparison of this searching operation with other methods, we have some clear discussion. PointNet++ searches neighbors within a ball query radius. A PointNet-based hierarchical network separately processes local points. It, however, ignores the relationships between points. Moreover, the ball query selects all points inside the ball which would be tricky if the number of neighbors is small. On the other hand, although K-nearest neighbors (KNN) outputs a fixed number of neighbors, it does not guarantee that neighboring information comes from all directions. Neighborhood points found by LAE-Conv are not treated equally, the contribution of neighbors is computed by attention mechanism and the updated features of central point

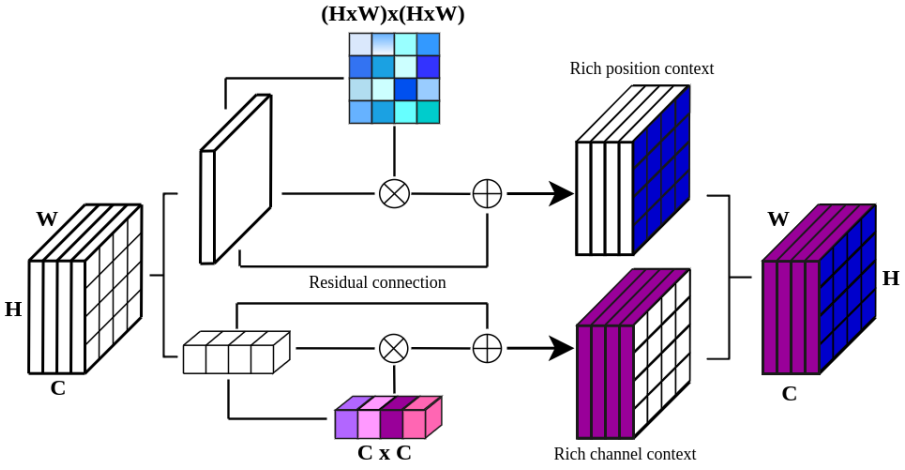


Fig. 5: Architecture of dual attention network includes two module of position attention and channel attention. The output feature is rich of position context and channel context.

p_i are aggregated with respect to K neighbors.

$$\alpha_{ij} = softmax(a(W(p_j - p_i))) \tag{14}$$

$$p'_i = \sum_{j \in N_{p_i}} \alpha_{ij} W_{p_j} \tag{15}$$

Where p_i is central point and its neighbor p_j. α_{ij} is weight coefficient of point p_j to contribute to p_i and it is normalized by softmax function. W is a learnable weight matrix that transforms the input point to a higher-level local feature. $(p_j - p_i)$ is a function to transform neighbor to local coordinate systems. p'_i is the higher-level feature of central point p_i. The wealthy representation of local geometric features extracted from LAE-Conv is fed into Point-wise spatial module to further capture long-range contextual information. This is obtained by the prevalent attention mechanism as shown in Fig.6. Weight map S is computed weight function added softmax normalization. The presence of PLAE in Fig.6 refers to residual connection. Practicing PAN on challenging benchmarks proves its ability to achieve 3D object detection is competitive with other state-of-the-art methods.

Point Transformer (Zhao et al. (2021)): A self-attention-based backbone for diverse tasks with 3D point clouds such as object classification, object part segmentation, and semantic segmentation. The research explores the performance of two type of attention operation: scalar attention (Vaswani et al. (2017)) and vector attention (Zhao et al. (2020)) and adding position encoding element as well. The attention layer of this method can be represented as follow.

$$y_i = \sum \rho(\gamma(f(x_i, x_j) + \varepsilon) \odot (g(x_j) + \varepsilon)) \tag{16}$$

Where y_i is responses based on different locations, ρ is the normalizing function, γ is the mapping function to transform the output of the relation function to right dimensionality, and ε is the position encoding element. To clearly understand the examining scalar attention and vector attention, we look closer into relation function f. There are two forms of attention operator: pairwise self-attention and patchwise self-attention.

- The more common pairwise self-attention, in which weight computation $f(x_i, x_j)$ is computed by aggregating all feature vectors of positions x_i, x_j within whole image as Eq.16(without γ and ε ingredients). The dimensionality of relation function output depends on the form of relation function f. The pre-dominant formulation is dot-product attention that produces output with dimensionality equals to 1. This construction shares its output across all channels and does not adapt the attention weights at different channels. The specific choice of dot-product is termed scalar attention. The other cases of relation function such as summation, subtraction, and Hadamard production produce vector output (Zhao et al. (2020)) can be additionally

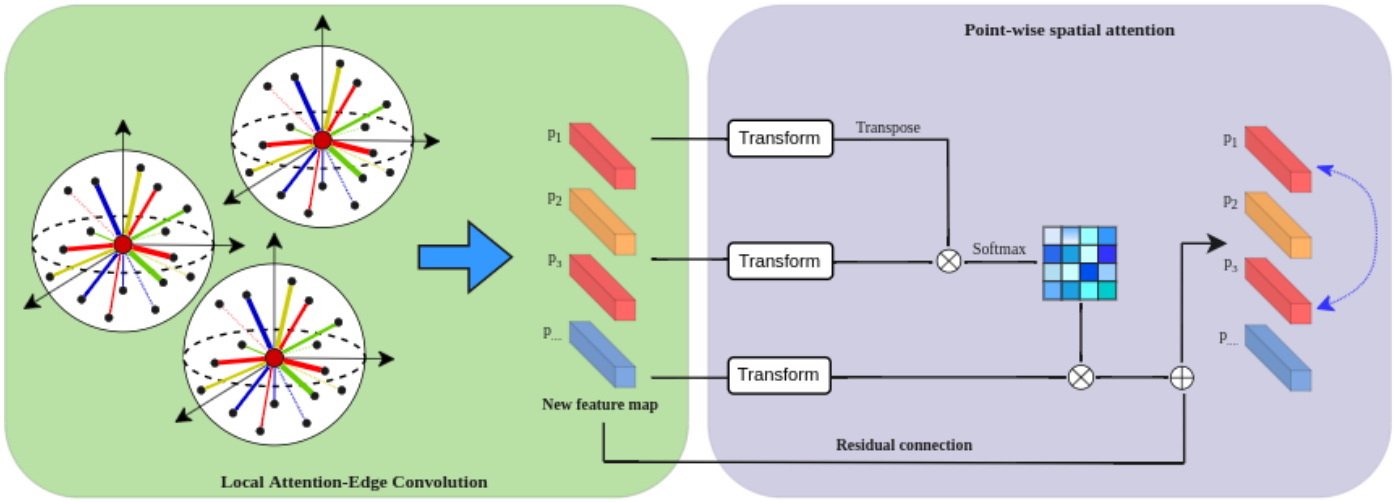


Fig. 6: Local-global architecture of Point-attention block contains Local Attention-Edge Convolution module and Point-wise spatial attention. The first module searches a target point's 16 neighbors within a ball and computes a new feature map, which is enriched with geometric information of neighbors. The thickness of lines connecting the center point to neighbors refers to different contributing values. After enlarging local representation, the second module captures global correlations.

processed to map right dimensionality to input features by γ function. Therefore, vector weights can vary along the channel dimensions, and it is termed vector attention. In Point Transformer, subtraction relation is selected for relation function to apply vector attention. Experiments yield that using vector attention (Subtraction relation) outperforms using scalar attention (Dot-product relation).

$$\alpha(x_i, x_j) = \gamma(f(x_i, x_j)) \quad (17)$$

$$\textbf{Dot-product: } f(x_i, x_j) = \varphi(x_i)^\top \psi(x_j) \quad (18)$$

$$\textbf{Summation: } f(x_i, x_j) = \varphi(x_i) + \psi(x_j) \quad (19)$$

$$\textbf{Subtraction: } f(x_i, x_j) = \varphi(x_i) - \psi(x_j) \quad (20)$$

$$\textbf{Hadamard product: } f(x_i, x_j) = \varphi(x_i) \odot \psi(x_j) \quad (21)$$

- Another form of attention is patchwise attention, in which the input of relation function is the patch of feature vectors $x_{R(i)}$ instead of feature vector $x(i)$ at a particular position. Moreover, the output of wight computation function alpha(xR(i)) is a tensor of same spatial dimensionality as the patch $x_{R(i)}$. Thus, patchwise attention inherently is vector attention. Several selections for relation function are Star-product, Clique-product, Concatenation (Zhao et al. (2020)).

- **Position encoding.** In pairwise attention, feature vectors $x(j)$ are processed independently so that the weight computation cannot leverage information from any location except for i and j. Therefore, position encoding facilizes augmenting feature maps with position information. The importance of position encoding is examined in experiments suggest that applying relative position encoding for both attention $(f(x_i, x_j) + \varepsilon)$ and feature $(g(x_j) + \varepsilon)$ accomplishes the highest performance.

4. Evaluation

In this section, we would like to answer the following questions: (1) How do attention modules would affect the performance of grasp detection? What sort of attention modules is suitable for 3D point cloud data? How well do the learned models generalize to novel object categories? To answer the above questions, we conduct the experiments of grasp detection on the public dataset GraspNet-1Billion Fang et al. (2020). This is a large-scale grasp dataset collected from cluttered scenes considering multi-object-multi-grasp setting. The objects in GraspNet-1Billion have varying shapes, textures, sizes, materials, and underlying different occlusion conditions. Hence, it

can be used to evaluate robustness to occlusion and the generalization ability of trained models.

4.1. GraspNet-1Billion

The GraspNet-1Billion (Fang et al. (2020)) consists of 97,280 RGB-D images captured from 190 cluttered scenes. The dataset provides over one billion grasp poses for 88 objects presented in the scenes and an accurate 3D mesh model of each object is available as well. Besides, it also provides relevant information including camera poses, 6D object poses, object masks, and bounding boxes for all frames. The rich annotations allow us to generate ground truth votes and grasp configurations easily. Following (Fang et al. (2020)) we split the dataset into 100 scenes for training and 90 scenes for testing. To evaluate model generalizability, the test sets are divided into 30 scenes with novel objects, 30 for unseen but similar objects, and the rest for seen objects.

4.2. Implementation

Table 1: Layer parameters of the PointNet++ (Qi et al. (2017b)) based feature learning network.

layer name	input layer	layer params
SA1	point cloud	(2048,0.025,[64,64,128])
SA2	SA1	(1024,0.05,[128,128,256])
SA3	SA2	(512,0.1,[128,128,256])
SA4	SA3	(256,0.2,[128,128,256])
FP1	SA3, SA4	[256,256]
FP2	SA2, SA3	[256,256]

In our implementation, we randomly choose N=50k points from each raw point cloud. We then apply the PointNet++ (Qi et al. (2017b)) based feature learning network, which has 4 set abstraction layers (SA) and 2 feature propagation layers (FP). The detailed layer parameters are shown in Table 1. The FP2 outputs $M = 1024$ seeds with $F = 256 - dim$ features and 3D coordinates that will be transformed to votes. The voting module generates $J = 10$ votes per seed with an MLP layer spec: $[256, 256, 259 \times 10]$. In the attention module, we form $K = 1024$ clusters and output a new feature map $C_{context} \in K \times F'$ where $K = 1024$, $F' = 128$. In the last step, 1024 grasps are detected from the new feature map. The prediction layer has $5 + V + 2A$ channels where $V = 120$, and $A = 6$.

The implementations are realized by PyTorch and Python platforms on one Nvidia GeForce RTX 2080 Ti 10GB GPU using CUDA and Linux operating system. All the experiments adopt similar training settings. The networks are trained from scratch in an end-to-end manner. We train each model over 200 epochs with stochastic gradient descent using a batch size of 8 and the Adam optimizer with a learning rate of 0.001.

4.3. Metrics

To evaluate the performance of grasp detection, we follow prior work (Fang et al. (2020)) and report results using $Precision@k$. This metric measures the precision of top-k ranked grasps. We first check whether a predicted grasp (G_p) is true positive or not. It is considered a true positive only if the grasp satisfies three conditions: (i) there is an object inside the gripper; (ii) it is collision-free; (iii) the grasp is antipodal under a given friction coefficient μ. The third condition is computed based on the prior works (ten Pas et al. (2017); Fang et al. (2020)). We let AP_μ denote the average $Precision@k$ for k ranges from 1 to 50 given a friction coefficient μ. We report the average of AP_μ with $\mu = \{0.2, 0.4, 0.6, 0.8, 1.0\}$, denoted as **AP**.

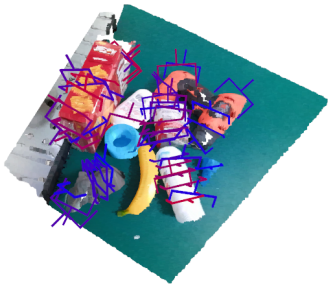
4.4. Results

We examine VoteGrasp with different self-attention modules in GraspNet-1Billion dataset captured by RealSense sensor and Kinect sensor and the results are shown in Table. 3 and Table. 4 respectively. Two tables witness a similar trend of grasp precision of attention modules in testing with seen, unseen (but similar), and novel objects. Point attention module yields severest grasp detection accuracy in all practicing conditions. Non-local and Criss-cross networks both using only position attention gain approximate accuracy and are slightly higher than point attention. Whereas, SE instead focuses on

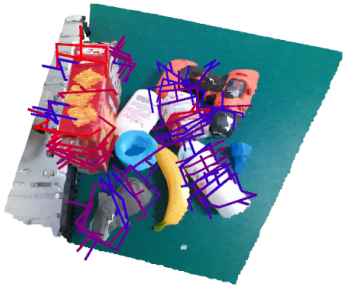
(a)

(b)

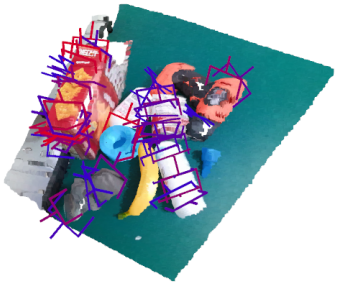
(c)



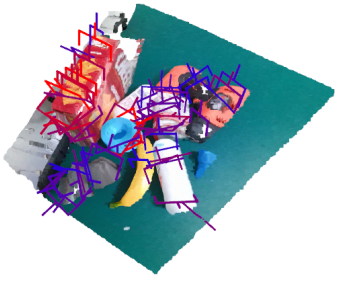
(d)



(e)



(f)



(g)

(h)

(i)

(j)

Fig. 7: Examples of input point clouds and predicted grasps from VoteGrasp combining with different attention module; (a) experiment objects; (b) input point cloud; (c) Non-local; (d) Criss-cross; (e) Squeeze-and-Excitation; (f) Compact Generalized Non-local; (g) Dual Attention Network; (h) Convolution Block Attention Module; (i) Point attention; (j) Point transformer. The different intensity of grasp color denotes the confident score of grasps. Red refers to the highest quality grasps and blue refers to the lowest ones.

channel correlations and meets more reliable grasps than solely employing positional distribution. CGNL fusing information of pixels' position into feature vectors obtains better grasp detection. Leveraging both position attention module and channel attention module, CBAM and DANet acquire significant grasp accuracy than the previously mentioned blocks. However, ar-

Table 2: Inference Time of VoteGrasp with different attention modules, evaluated on GraspNet-1Billion (Fang et al. (2020)) dataset.

Method	Inference Time (ms)
Non-local	140
CGNL	150
Criss-cross	138
Squeeze-and-Excitation (SE)	135
CBAM	143
DANet	145
Point Transformer	170
Point-Attention	148

ranging two attention modules in sequence as CBAM achieves marginally reasonable grasps than placing them in parallel as DANet. Finally, point transformer outperforms all others with huge significance of grasp accuracy at 41.63% averaged AP from both cameras. Besides, we investigate inference time of VoteGrasp with different attention modules. We perceive that all versions consume approximate time around $135ms$ to $170ms$ as shown in Table. 2. Point transformer, which achieves best result of grasp detection, takes $170ms$. Therefore, this block is considered as the preferred solution to be combined with VoteGrasp to achieve optimal robustness.

5. Conclusions

In this work, we conduct VoteGrasp with a set of attention modules. Thereby, we provide insights about attention mechanism and its ability to be integrated with grasp detection architecture as well. Taking advantage of VoteGrasp in grasp detection, we examine the performance of different attention modules to discover the optimal combination for collision-free grasps. Through experiments, we verify that point transformer is the ideal choice to achieve high-quality grasps in occlusions and cluttered scenes. This attention module guarantees VoteGrasp's ability to generalize to highly occluded objects and even novel objects while keeping the model to be lightweight. Interesting future work is to consider adding a reachability predictor to the grasping network and explore the use of our approach in task planning applications.

References

Besl, P.J., McKay, N.D., 1992. Method for registration of 3-d shapes, in: Sensor fusion IV: control paradigms and data structures, Spie. pp. 586–606.

Bohg, J., Morales, A., Asfour, T., Kragic, D., 2013. Data-driven grasp synthesisa survey. IEEE Transactions on robotics 30, 289–309.

Buades, A., Coll, B., Morel, J.M., 2005. A non-local algorithm for image denoising, in: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), IEEE. pp. 60–65.

Choi, C., Schwarting, W., DelPreto, J., Rus, D., 2018. Learning object grasping for soft robot hands. IEEE Robotics and Automation Letters 3, 2370–2377.

Deng, H., Birdal, T., Ilic, S., 2018. Ppfnet: Global context aware local features for robust 3d point matching, in: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 195–205.

Dias, A.S., Brites, C., Ascenso, J., Pereira, F., 2014. Sift-based homographies for efficient multiview distributed visual sensing. IEEE Sensors Journal 15, 2643–2656.

Duda, R., Hart, P., 1972. [a]use of the hough transform to detect lines and curves in pictures, [o] comm. ACM .

Fang, H.S., Wang, C., Gou, M., Lu, C., 2020. Graspnet-1billion: A large-scale benchmark for general object grasping, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 11444–11453.

Feng, M., Zhang, L., Lin, X., Gilani, S.Z., Mian, A., 2020. Point attention network for semantic segmentation of 3d point clouds. Pattern Recognition 107, 107446.

Fu, J., Liu, J., Tian, H., Li, Y., Bao, Y., Fang, Z., Lu, H., 2019. Dual attention network for scene segmentation, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 3146–3154.

Gall, J., Lempitsky, V., 2013. Class-specific hough forests for object detection, in: Decision forests for computer vision and medical image analysis. Springer, pp. 143–157.

Gall, J., Yao, A., Razavi, N., Van Gool, L., Lempitsky, V., 2011. Hough forests for object detection, tracking, and action recognition. IEEE transactions on pattern analysis and machine intelligence 33, 2188–2202.

Golemati, S., Stoitsis, J., Balkizas, T., Nikita, K., 2006. Comparison of b-mode, m-mode and hough transform methods for measurement of arterial diastolic and systolic diameters, in: 2005 IEEE Engineering in Medicine and Biology 27th Annual Conference, IEEE. pp. 1758–1761.

Herzog, A., Pastor, P., Kalakrishnan, M., Righetti, L., Asfour, T., Schaal, S., 2012. Template-based learning of grasp selection, in: 2012 IEEE International Conference on Robotics and Automation, IEEE. pp. 2379–2384.

Hoang, D.C., Stork, J.A., Stoyanov, T., 2022. Context-aware grasp generation in cluttered scenes, in: IEEE International Conference on Robotics and Automation (ICRA 2022), Philadelphia, USA, May 23-27, 2022.

Hou, J., Dai, A., Nießner, M., 2019. 3d-sis: 3d semantic instance segmentation

Table 3: The table shows the results on GraspNet-1Billion test set captured by RealSense sensor.

	Seen			Unseen (but similar)			Novel		
	AP	$AP_{0.8}$	$AP_{0.4}$	AP	$AP_{0.8}$	$AP_{0.4}$	AP	$AP_{0.8}$	$AP_{0.4}$
Non-local	28.06	32.24	18.43	26.30	35.29	14.92	11.79	11.83	6.62
Criss-cross	29.01	33.37	18.84	27.44	36.62	15.72	12.26	12.43	6.77
Squeeze-and-Excitation (SE)	32.18	37.32	22.01	30.60	39.73	18.98	15.91	16.06	8.05
CGNL	34.13	38.87	24.04	33.03	40.78	20.54	16.92	17.03	10.01
CBAM	36.70	41.08	27.04	35.23	43.62	22.83	21.08	20.69	10.25
DANet	34.02	38.51	24.00	32.87	40.01	20.12	16.50	17.00	9.83
Point Transformer	**41.45**	**45.63**	**32.42**	**40.12**	**48.40**	**27.21**	**26.02**	**26.23**	**14.18**
Point-Attention	26.34	30.40	17.18	24.26	33.35	12.72	10.68	10.82	6.05

Table 4: The table shows the results on GraspNet-1Billion test set captured by Kinect sensors respectively.

	Seen			Unseen (but similar)			Novel		
	AP	$AP_{0.8}$	$AP_{0.4}$	AP	$AP_{0.8}$	$AP_{0.4}$	AP	$AP_{0.8}$	$AP_{0.4}$
Non-local	31.52	40.37	21.19	30.57	38.40	18.67	13.05	13.19	7.03
Criss-cross	32.29	40.85	21.91	31.23	39.07	19.09	13.82	13.93	7.08
Squeeze-and-Excitation (SE)	35.93	43.85	24.05	33.68	41.09	21.58	16.00	16.09	8.29
CGNL	37.52	45.61	27.68	35.86	43.31	24.74	18.46	18.49	10.63
CBAM	39.98	48.60	28.79	38.24	46.02	26.25	21.30	20.59	10.34
DANet	37.36	45.17	27.32	35.38	43.02	24.15	18.02	18.09	10.23
Point Transformer	**44.80**	**52.47**	**33.03**	**42.19**	**50.18**	**30.58**	**26.21**	**26.69**	**14.50**
Point-Attention	29.01	37.62	19.19	27.63	36.21	16.32	11.35	11.68	6.80

of rgb-d scans, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 4421–4430.

Hough, P.V., 1959. Machine analysis of bubble chamber pictures, in: Proc. of the International Conference on High Energy Accelerators and Instrumentation, Sept. 1959, pp. 554–556.

Hough, P.V., 1962. Method and means for recognizing complex patterns. US Patent 3,069,654.

Hu, J., Shen, L., Sun, G., 2018a. Squeeze-and-excitation networks, in: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 7132–7141.

Hu, S.M., Cai, J.X., Lai, Y.K., 2018b. Semantic labeling and instance segmentation of 3d point clouds using patch context analysis and multiscale processing. IEEE transactions on visualization and computer graphics 26, 2485–2498.

Huang, Z., Wang, X., Huang, L., Huang, C., Wei, Y., Liu, W., 2019. Ccnet: Criss-cross attention for semantic segmentation, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 603–612.

Iocchi, L., Mastrantuono, D., Nardi, D., 2001. A probabilistic approach to hough localization, in: Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No. 01CH37164), IEEE. pp. 4250–4255.

Kalviainen, H., 1996. Motion detection using the randomised hough transform: exploiting gradient information and detecting multiple moving objects. IEE Proceedings-Vision, Image and Signal Processing 143, 361–369.

Kehl, W., Manhardt, F., Tombari, F., Ilic, S., Navab, N., 2017. Ssd-6d: Making rgb-based 3d detection and 6d pose estimation great again, in: Proceedings of the IEEE international conference on computer vision, pp. 1521–1529.

Kehl, W., Milletari, F., Tombari, F., Ilic, S., Navab, N., 2016. Deep learning

of local rgb-d patches for 3d object detection and 6d pose estimation, in: European conference on computer vision, Springer. pp. 205–220.

Lenz, I., Lee, H., Saxena, A., 2015. Deep learning for detecting robotic grasps. The International Journal of Robotics Research 34, 705–724.

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

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C., 2016. Ssd: Single shot multibox detector, in: European conference on computer vision, Springer. pp. 21–37.

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

Mahler, J., Matl, M., Liu, X., Li, A., Gealy, D., Goldberg, K., 2018. Dex-net 3.0: Computing robust vacuum suction grasp targets in point clouds using a new analytic model and deep learning, in: 2018 IEEE International Conference on robotics and automation (ICRA), IEEE. pp. 5620–5627.

Mousavian, A., Eppner, C., Fox, D., 2019. 6-dof graspnet: Variational grasp generation for object manipulation, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 2901–2910.

Muñoz, E., Konishi, Y., Murino, V., Del Bue, A., 2016. Fast 6d pose estimation for texture-less objects from a single rgb image, in: 2016 IEEE International Conference on Robotics and Automation (ICRA), IEEE. pp. 5623–5630.

Ni, P., Zhang, W., Zhu, X., Cao, Q., 2020. Pointnet++ grasping: learning an end-to-end spatial grasp generation algorithm from sparse point clouds, in: 2020 IEEE International Conference on Robotics and Automation (ICRA), IEEE. pp. 3619–3625.

Paigwar, A., Erkent, O., Wolf, C., Laugier, C., 2019. Attentional pointnet for 3d-object detection in point clouds, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, pp. 0–0.

ten Pas, A., Gualtieri, M., Saenko, K., Platt, R., 2017. Grasp pose detection in point clouds. The International Journal of Robotics Research 36, 1455–1473.

Peng, S., Liu, Y., Huang, Q., Zhou, X., Bao, H., 2019. Pvnet: Pixel-wise voting network for 6dof pose estimation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4561–4570.

Qi, C.R., Litany, O., He, K., Guibas, L.J., 2019. Deep hough voting for 3d object detection in point clouds, in: proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 9277–9286.

Qi, C.R., Liu, W., Wu, C., Su, H., Guibas, L.J., 2018. Frustum pointnets for 3d object detection from rgb-d data, in: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 918–927.

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

Qi, C.R., Yi, L., Su, H., Guibas, L.J., 2017b. Pointnet++: Deep hierarchical feature learning on point sets in a metric space, in: Advances in neural information processing systems, pp. 5099–5108.

Rabbani, T., Van Den Heuvel, F., 2005. Efficient hough transform for automatic detection of cylinders in point clouds. Isprs Wg Iii/3, Iii/4 3, 60–65.

Redmon, J., Angelova, A., 2015. Real-time grasp detection using convolutional neural networks, in: 2015 IEEE international conference on robotics and automation (ICRA), IEEE. pp. 1316–1322.

Ren, S., He, K., Girshick, R., Sun, J., 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. Advances in neural information processing systems 28.

Shi, Y., Chang, A.X., Wu, Z., Savva, M., Xu, K., 2019. Hierarchy denoising recursive autoencoders for 3d scene layout prediction, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 1771–1780.

Silberberg, T.M., Davis, L., Harwood, D., 1984. An iterative hough procedure for three-dimensional object recognition. Pattern Recognition 17, 621–629.

Song, S., Xiao, J., 2016. Deep sliding shapes for amodal 3d object detection in rgb-d images, in: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 808–816.

Tombari, F., Di Stefano, L., 2010. Object recognition in 3d scenes with occlusions and clutter by hough voting, in: 2010 Fourth Pacific-Rim Symposium on Image and Video Technology, IEEE. pp. 349–355.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I., 2017. Attention is all you need. Advances in neural information processing systems 30.

Wang, C., Xu, D., Zhu, Y., Martín-Martín, R., Lu, C., Fei-Fei, L., Savarese, S., 2019. Densefusion: 6d object pose estimation by iterative dense fusion, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 3343–3352.

Wang, X., Girshick, R., Gupta, A., He, K., 2018. Non-local neural networks, in: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 7794–7803.

Woo, S., Park, J., Lee, J.Y., Kweon, I.S., 2018. Cbam: Convolutional block attention module, in: Proceedings of the European conference on computer vision (ECCV), pp. 3–19.

Wu, D., Zhuang, Z., Xiang, C., Zou, W., Li, X., 2019. 6d-vnet: End-to-end 6-dof vehicle pose estimation from monocular rgb images, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, pp. 0–0.

Xie, S., Liu, S., Chen, Z., Tu, Z., 2018. Attentional shapecontextnet for point cloud recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4606–4615.

Ye, X., Li, J., Huang, H., Du, L., Zhang, X., 2018. 3d recurrent neural networks with context fusion for point cloud semantic segmentation, in: Proceedings of the European conference on computer vision (ECCV), pp. 403–417.

Yue, K., Sun, M., Yuan, Y., Zhou, F., Ding, E., Xu, F., 2018. Compact generalized non-local network, in: Advances in Neural Information Processing Systems, pp. 6510–6519.

Zeng, A., Yu, K.T., Song, S., Suo, D., Walker, E., Rodriguez, A., Xiao, J., 2017. Multi-view self-supervised deep learning for 6d pose estimation in the amazon picking challenge, in: 2017 IEEE international conference on robotics and automation (ICRA), IEEE. pp. 1386–1383.

Zhang, W., Xiao, C., 2019. Pcan: 3d attention map learning using contextual information for point cloud based retrieval, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 12436–12445.

Zhao, H., Jia, J., Koltun, V., 2020. Exploring self-attention for image recognition, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10076–10085.

Zhao, H., Jiang, L., Jia, J., Torr, P.H., Koltun, V., 2021. Point transformer, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 16259–16268.