GSPN: Generative Shape Proposal Network for 3D Instance Segmentation in Point Cloud

Li Yi[1]   Wang Zhao[2]   He Wang[1]   Minhyuk Sung[1]   Leonidas Guibas[1,3]

[1]Stanford University   [2]Tsinghua University   [3]Facebook AI Research

Abstract

We introduce a novel 3D object proposal approach named Generative Shape Proposal Network (GSPN) for instance segmentation in point cloud data. Instead of treating object proposal as a direct bounding box regression problem, we take an analysis-by-synthesis strategy and generate proposals by reconstructing shapes from noisy observations in a scene. We incorporate GSPN into a novel 3D instance segmentation framework named Region-based Point-Net (R-PointNet) which allows flexible proposal refinement and instance segmentation generation. We achieve state-of-the-art performance on several 3D instance segmentation tasks. The success of GSPN largely comes from its emphasis on geometric understandings during object proposal, which greatly reducing proposals with low objectness.

1. Introduction

Instance segmentation is one of the key perception tasks in computer vision, which requires delineating objects of interests in a scene and also classifying the objects into a set of categories. 3D instance segmentation, with a huge amount of applications in robotics and augmented reality, is in tremendous demand these days. However, the progress of 3D instance segmentation lags far behind its 2D counterpart [15, 24, 20], partially because of the expensive computation and memory cost of directly applying 2D Convolutional Neural Networks (CNN) approaches to 3D volumetric data [39, 7]. Recently, [33, 34] proposed efficient and powerful deep architectures to directly process point cloud, which is the most common form of 3D sensor data and is very efficient at capturing details in large scenes. This opens up new opportunities for 3D instance segmentation and motivates us to work with 3D point clouds.

The great advance in 2D instance segmentation is largely driven by the success of object proposal methods [41], where object proposals are usually represented as 2D bounding boxes. Thus it is natural to seek for an effective way of representing and generating object proposals in the 3D point cloud. But, this is indeed very challenging since 3D object proposal approaches need to establish the understanding of semantics and objectness for a wide range of object categories with various scales in a cluttered
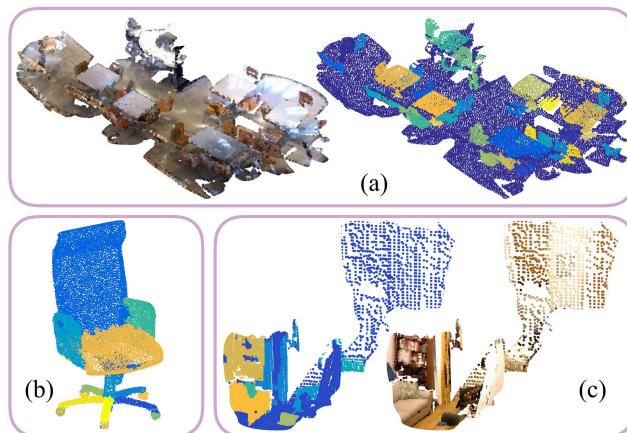

Figure 1. The flexibility of our instance segmentation framework R-PointNet allows it to well handle various types of input data including (a) a complete reconstruction for a real indoor scene, (b) objects with fine-grained part instances, (c) partial point cloud captured from a single view point.

scene under heavy sensor noise and data incompleteness. A straightforward method is to directly estimate simple geometric approximations to the objects such like 3D bounding boxes [39, 48]. These approximations are simple and not very faithful to most objects, meaning generating such proposals does not require a strong understanding of the underlying object geometry. This makes it easy to produce blind box proposals which either contain multiple objects or just a part of an object, influencing the differentiation among object instances. Moreover, we hardly know how well the 3D object proposer understands objectness.

In contrast, we take a different perspective for object proposal which emphasizes more on the geometric understanding. It is a commonplace that perception is in part constructive [16]. Therefore we leverage an analysis-by-synthesis strategy where we propose objects by first reconstructing them. Specifically, we leverage a generative model to explicitly depict the natural object distribution and propose candidate objects by drawing samples from the object distribution. The noisy observations in a scene will guide the proposal generation process by indicating where to sample in the object distribution. This idea is especially appealing in 3D since unlike 2D images, objects are more canonical in the 3D world with the right physical scale and more com-

plete geometry. As a result, the object distribution is more compact, making it feasible to capture.

We design a deep neural network named Generated Shape Proposal Network (GSPN) to achieve this purpose. Compared with direct 3D bounding boxes regression, the advantages of GSPN are two-fold. Firstly, it produces object proposals with higher objectness. The network is explicitly trained to understand how natural objects look like before it generates any proposal. By enforcing geometric understanding, we can greatly reduce blind box proposals not corresponding to a single object. Secondly, GSPN encodes noisy observations to distributions in the natural object space, which can be regarded as an instance-aware feature extraction process. These features delineate object boundaries and could serve as a very important cue for proposal refinement and segmentation mask generation.

To be able to reject, receive and refine object proposals and further segment out various instances in a 3D point cloud, we develop a novel 3D instance segmentation framework called Region-based PointNet (R-PointNet). From a high level, R-PointNet resembles image Mask R-CNN [15]; it contains object proposal and proposal classification, refinement and segmentation components. We carefully design R-PointNet so that it can nicely consume unstructured point cloud data and make the best use of object proposals and instance sensitive features generated by GSPN.

We have tested our instance segmentation framework R-PointNet with GSPN on various tasks including instance segmentation on complete indoor reconstructions, instance segmentation on partial indoor scenes, and object part instance segmentation. We achieve state-of-the-art performance on all these tasks.

Key contributions of our work are as follows:

- We propose a Generative Shape Proposal Network to tackle 3D object proposal following an analysis-by-synthesis strategy.

- We propose a flexible 3D instance segmentation framework called Region-based PointNet, with which we achieve state-of-the-art performance on several instance segmentation benchmarks.

- We conduct extensive evaluation and ablation study to validate our design choices and show the generalizability of our framework.

2. Related Work

Object Detection and Instance Segmentation Recently, a great progress has been made for 2D object detection [9, 36, 35, 21, 22, 25, 19] and instance segmentation [6, 15, 20, 31]. R-CNN [10] firstly combines region proposal with CNN for 2D object detection. After this, a series of works including Fast R-CNN [9], Faster R-CNN [36] and Mask R-CNN [15] are proposed to accelerate region proposal, improve feature learning, and extend the detection framework for the instance segmentation task.

Following these progress in 2D, learning-based 3D detection and instance segmentation frameworks gradually emerged. People largely focused on 3D object bounding box detection where object proposal is essential and have come up with different approaches. [39] directly apply region proposal network (RPN) [36] on volumetric data, which is limited due to its high memory and computation cost. [7, 32, 48, 4, 45, 28] rely on the mature 2D object proposal approaches and obtain object proposal from projected views of a 3D scene. It is hard to apply these methods to segment cluttered indoor environment which cannot be fully covered by a few views. In addition, the projection loses information about the scene such as the physical sizes of objects and introduces additional difficulty in object proposal, making it less appealing than directly proposing objects in 3D. [23] first categorizes voxel cells and then merges them to form objects. As a pioneering work in 3D instance segmentation learning, [42] proposes a shape-centric way to directly propose objects in the 3D point cloud, where points are grouped with a learned similarity metric to form candidate objects. However this bottom-up grouping strategy cannot guarantee proposals with high objectness. In contrast to previous approaches, our approach directly proposes objects in 3D and emphasizes the proposal objectness through a generative model.

3D Generative Models Variational Autoencoder (VAE) [17] is one of the most popular generative models commonly used for image or shape generation [13, 29]. It learns to encode natural data samples x into a latent distribution where samples can be drawn and decoded to the initial data form. VAE explicitly models the data distribution and learns a proper parametrization via maximizing the data likelihood. However, VAE cannot add controls to the sampled data points, which usually restricts its usage. An extension called Conditional Variational Autoencoder (CVAE) was proposed in [38], where the generation is also conditioned on certain attributes.

Alternative to VAE and CVAE, GAN [12] and CGAN [26] could generate more faithful images or shapes by introducing an adversarial game between a discriminator and a generator. However, it is well-known that GAN suffers from mode collapse issues since it doesn't explicitly model the data likelihood, while likelihood-based models, such as VAE, can generally capture a more complete data distribution. We leverage CVAE instead CGAN since it complies with the condition more on average.

Deep Learning on Point Cloud Various 3D representations [40, 43, 37, 47, 33, 34, 8] have been explored recently for deep learning on 3D data. Among them, point cloud representation is becoming increasingly popular due to its memory efficiency and intuitiveness. We use existing deep architectures for 3D point cloud such as PointNet/PointNet++ [33, 34] and the Point Set Generation networks [8] as the bases of our framework.
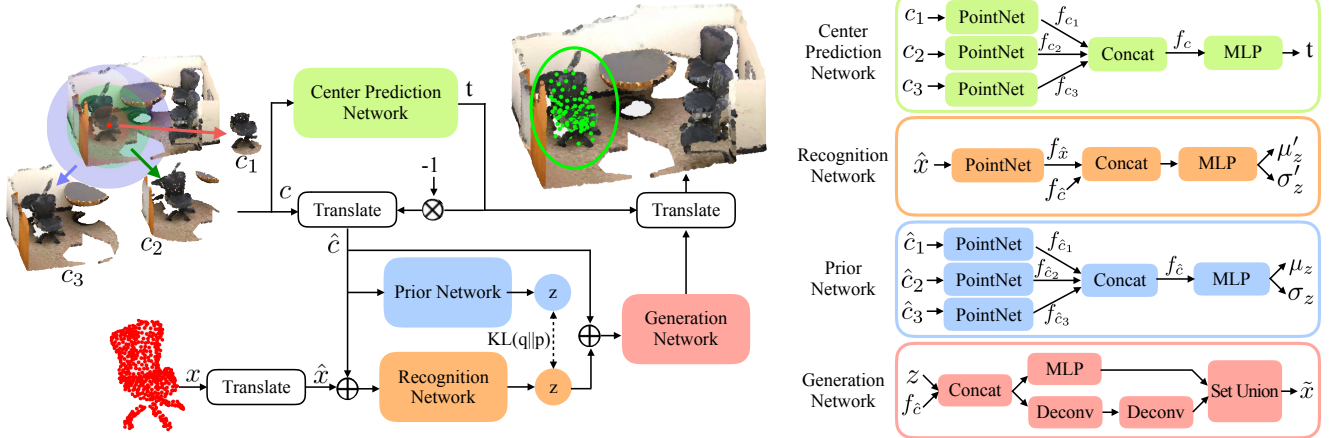
Figure 2. The architecture of GSPN. On the left we show the data flow in GSPN and the key building blocks, highlighted by colored rectangles. The detailed architecture of each building block is shown on the right.

3. Method

We propose R-PointNet, a 3D object instance segmentation framework that shares a similar high-level structure with 2D Mask R-CNN [15] but is carefully designed for unstructured point cloud data. Most importantly, it leverages a network named Generative Shape Proposal Network (GSPN) that efficiently generates 3D object proposals with high objectness. In addition, our Point RoIAlign layer is designed to collect features for proposals, allowing the network to refine the proposals and generate segments. Next, we will explain the design details of our network.

3.1. Generative Shape Proposal Network

GSPN treats the object proposal procedure as a object generation, sampling from a conditional probability distribution of natural objects $p_\theta(x|c)$ *conditioned* on the scene point cloud P and the seed point s, where c represents the context pair (P, s). The output point cloud $\tilde{x}$ generated as an object proposal approximates the object x in P containing s. This approach allows us to concretely see what a certain object proposal looks like and understand whether the network learns the objectness. Specifically, we formulate GSPN as a conditional variational autoencoder (CVAE) [38]. When approximating $p_\theta(x|c)$ as $\int_z p_\theta(x|z,c)p_\theta(z|c)dz$ with a latent representation z of natural objects, the proposals are generated by drawing a sample z from the conditional prior distribution $p_\theta(z|c)$, and then computing object proposals $\tilde{x}$ through the generative distribution $p_\theta(x|z,c)$. $p_\theta(z|c)$ and $p_\theta(x|z,c)$ are learned by maximizing the following variational lower bound of the training data conditional log-likelihood $\log p_\theta(x|c)$:

$$L = \mathbb{E}_{q_\phi(z|x,c)}[\log p_\theta(x|z,c)] - \text{KL}(q_\phi(z|x,c)||p_\theta(z|c)) \quad (1)$$

where $q_\phi(z|x,c)$ is a proposal distribution which approximates the true posterior $p_\theta(z|x,c)$.

The architecture of GSPN is shown in Figure 2. Two sub-networks, prior network and recognition network, parameterize $p_\theta(z|c)$ and $q_\phi(z|x,c)$ as Gaussian distributions

with predicted means and variances, respectively, and generation network learns $p_\theta(x|z,c)$. In addition, the center prediction network is used to centralize the context data and factor out the location information. The context pair c is represented by cropping P with spheres centered at s with $K = 3$ different radiuses to cover objects with various scales ($c_{k\in\{1\cdots K\}}$ will denote the context in each scale). We explain each sub-network next.

Center prediction network takes the context c as input and regresses the center t of the corresponding object x in the world coordinate system (the center is the axis-aligned bounding box center). The network employs K different PointNets each of which processes a context of each scale c_k and outputs a feature vector f_{c_k} independently, and it concatenates $\{f_{c_k}\}_{k=1}^K$ to form f_c, feeds the f_c through a multi-layer perceptron (MLP), and regresses the centroid location t. After this, the context c is centered at t, and the translated context $\hat{c}$ serves as the input to the prior network.

The prior network takes the same K-PointNet architecture for processing the input centered context $\hat{c}$, and maps the concatenated feature $f_{\hat{c}}$ to a Gaussian prior distribution $\mathcal{N}(\mu_z, \sigma_z^2)$ of $p_\theta(z|c)$ through an MLP. Recognition network shares the context encoder with the prior network, and it also consumes an centered object $\hat{x}$ and generates an object feature $f_{\hat{x}}$ with another PointNet. $f_{\hat{x}}$ is then concatenated with the context feature $f_{\hat{c}}$ and fed into an MLP for predicting the Gaussian proposal distribution $\mathcal{N}(\mu_z', \sigma_z'^2)$, which parametrizes $q_\phi(z|x,c)$.

After predicting $p_\theta(z|c)$ with the prior network, we sample z and feed it to the generation network. Again, the generation network shares the context encoder with the prior network. After concatenating the context features $f_{\hat{c}}$ from the prior network with z, it decodes a point cloud $\tilde{x}$ along with a per-point confidence score e representing the likelihood of appearance for each generated point. For decoding, we use point set generation architecture in [8] having two parallel branches, fully-connected (fc) branch and deconvolution (deconv) branches, and taking the union of two outputs. The resulting centralized point cloud is shifted back

(a)   (b)

Figure 3. A visualization for the learned instance sensitive features. (a) shows the context features $f_{\hat{c}}$ by first applying PCA to the high dimensional features and then converting the first three dimension into the color map here. (b) shows the predicted object centers. A clearer separation of different instances is shown in (b), which confirms that these features are instance sensitive.

to its original position with the predicted object center t.

GSPN takes an additional MLP taking $f_{\hat{c}}$ predicting objectness score for each proposal similarly with Mask R-CNN. The objectness score is supervised with axis-aligned bounding boxes; positive proposals come from seed points s belonging to foreground objects, and their bounding boxes overlap with some ground truth boxes with an intersection over union (IoU) larger than 0.5. Negative proposals are those whose IoUs with all the ground truth boxes are less than 0.5.

We emphasize that the center prediction network factoring out the location information in the generative model plays a very important role of simplifying the generation task by allowing contexts corresponding to the same object to be encoded with similar context features $f_{\hat{c}}$. We refer the feature $f_{\hat{c}}$ as *instance sensitive* features and visualize their predictions in Figure 3, where a clear difference can be observed among instances. From now on, we overload the symbol $f_{\hat{c}}$ to include the predicted object center t. We will also show how these instance sensitive feature could play a role for further proposal refinement and object segmentation in the next section.

Losses   GSPN is trained to minimize a multi-task loss function L_{GSPN} defined for each potential object proposal. L_{GSPN} is a summation of five terms including the shape generation loss L_{gen}, shape generation per-point confidence loss L_e, KL loss L_{KL}, center prediction loss L_{center}, and objectness loss L_{obj}. We use chamfer distance between the generated objects $\tilde{x}$ and the ground truth objects x as the generation loss L_{gen}, which serves as a surrogate to the negative log likelihood $-\log p_\theta(x|z, c)$. To supervise the per-point confidence prediction, we compute the distance from each predicted point to the ground truth object point cloud. Those points with distances smaller than a certain threshold ϵ are treated as confident predictions and others are unconfident predictions. KL loss essentially enforces the proposal distribution $q_\phi(z|x, c)$ and the prior distribution $p_\theta(z|c)$ to be similar. Since we have parametrized $q_\phi(z|x, c)$ and $p_\theta(z|c)$ as $\mathcal{N}(\mu'_z, \sigma'^2_z)$ and $\mathcal{N}(\mu_z, \sigma^2_z)$ respectively through neural networks, the KL loss can be easily computed as:

$$L_{KL} = \log \frac{\sigma'_z}{\sigma_z} + \frac{\sigma^2_z + (\mu_z - \mu'_z)^2}{2\sigma'^2_z} - 0.5 \qquad (2)$$

Average binary cross entropy loss is used for L_e. Smooth L1 loss [9] is used as the center prediction loss L_{center}. L_{obj} is also defined as an average binary cross entropy loss.

3.2. Region-based PointNet

In the second part of R-PointNet, the object proposals from GSPN are further processed to identify the object class, refine the proposal, and segment the foreground objects in the proposals from the initial point cloud P. We first obtain candidate RoIs by computing axis-aligned bounding boxes from the object proposals (points which confidence score e is greater than 0.5 are only used). Then, from each RoI, our Point RoIAlign layer extracts region features, which are fed through PointNet-based classification, regression, and segmentation sub-networks. Bounding boxes are refined by predicting the relative center and size changes as done in [36], and also the segmentation is obtained by predicting a per-point binary mask for each category label similarly with [15]. We visualize the architecture of R-PointNet in Figure 4. R-PointNet is trained to minimize a multi-task loss function defined within each RoI as $L = L_{cls} + L_{box} + L_{mask}$, which is the same as [15]. Next, we explain design details about R-PointNet.

Feature Backbone   Before computing region features in each RoI, we first augment the context feature $f_{\hat{c}}$ of GSPN with a *semantic* feature f_{sem} coming from a network pre-trained on a semantic segmentation task. Specifically, we pre-train a PointNet++ classifying each point into object classes with 4 sample-and-group layers and 4 feature interpolation layers. Then, we obtain the semantic feature for each point as a combination of each of the sample-and-group layers outputs in order to capture information at various scales. Since the point cloud is downsampled after a sample-and-group layer, for covering every point in P, we upsample the feature set after each sample-and-group layer through a feature interpolation operation; find the three closest points and interpolate with weights inversely proportional to distances. This allows us to concatenate features with different scales and form hypercolumns [14] for points. The concatenation of the context feature $f_{\hat{c}}$ and the semantic feature f_{sem} builds our feature backbone, and the feature backbones are aggregated in the next Point RoIAlign step. In ablation study 4.4, we demonstrate that both of the context and semantic features play an important role in obtaining good instance segmentation.

Point RoIAlign   For obtaining a fixed-size feature map within each RoI, our point RoIAlign layer samples N_{RoI} points equipped with a feature vector from P. Since the computation of context feature $f_{\hat{c}}$ is very expensive, practically we compute the features just for a set of *seed* points P_{sub}, and obtaining features of the sampled points using the feature interpolation step described in the previous paragraph. After point sampling and feature extraction, each RoI is normalized to be a unit cube centered at (0,0,0).
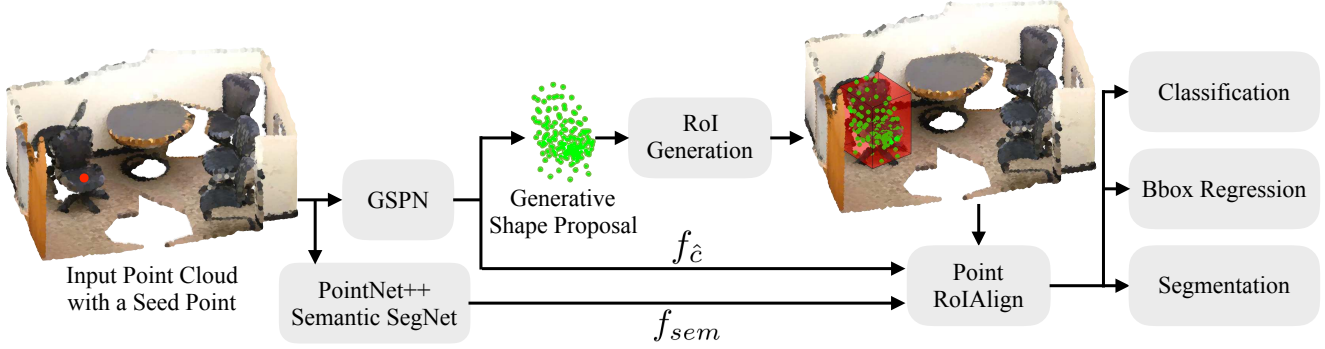
Figure 4. The architecture of R-PointNet. For each seed point in the scene, GSPN would generate a shape proposal along with instance sensitive features $f_{\hat{c}}$. The shape proposal is converted into an axis-aligned 3D bounding box, where Point RoIAlign can be applied to extract RoI features for the final segmentation generation. In addition to instance sensitive feature $f_{\hat{c}}$, semantic features obtained from a pretrained PointNet++ segmentation network are also used in the feature backbone.

3.3. Implementation Details

In all our experiments, we first train GSPN and Point-Net++ semantic segmentation network and then fix their weights during R-PointNet training.

Training   While training GSPN, we randomly sample 512 seed points for each training scene in each mini-batch, which gives us 512 shape proposals. We use a resolution of 512 points for these shape proposals. The context around each seed point is represented as a multi-scale cropping of the scene, where the cropping radius is set so that the smallest scale is on a par with the smallest object instance in the scenes and the largest scale roughly covers the largest object. To predict the object center for each seed point, we simply regress the unit direction vector from the seed point to the object center as well as the distance between the two similar to [44]. We also adopt KL-annealing [2] to stabilize the GSPN training.

During the training procedure of R-PointNet, we apply non-maximum suppression [11] on all the object proposals and keep a maximum number of 128 proposals for training. We select positive and negative RoIs in the same way as [15] where positive RoIs are those intersecting with ground truth bounding boxes with an IoU greater than 0.5, and negative RoIs are those with less than 0.5 IoU with all the ground truth bounding boxes. The ratio between positive and negative RoIs is set to be 1:3.

Inference   During the inference time, we randomly sample 2048 seed points in each test scene and keep at most 512 RoIs after non-maximum suppression for RoI classification, refinement, and segmentation. It usually takes ~1s on a Titan XP GPU to consume an entire scene with ~20k points. After obtaining a binary segmentation within each RoI, we project the segmentation mask back to the initial point cloud through the nearest neighbor search. All the points outside the RoI will be excluded from the projection.

4. Experiment

Our object proposal module GSPN and the instance segmentation framework R-PointNet are very general and could handle various types of data. To show their effectiveness, we experiment on three different datasets including:

- ScanNet [5]: This is a large scale scan dataset containing 3D reconstructions of 1613 indoor scenes. Each reconstruction is generated by fusing multiple scans from different views. The scenes are annotated with semantic and instance segmentation masks. They are officially split into 1201 training scenes, 312 validation scenes and 100 test scenes, where ground truth label is only publicly available for the training and validation sets.

- PartNet[27]: This dataset provides fine-grained part instance annotations for 3D objects from ShapeNet [3]. It covers 24 categories. And the number of part instances per-object ranges from 2 to 220 with an average of 18.

- NYUv2 [30]: This dataset contains 1449 RGBD images with 2D semantic instance segmentation annotations. We use the improved annotation from [7]. Partial point cloud could be obtained by lifting the depth channel using the camera information. And we follow the standard train test split.

We also conduct extensive ablation study to validate different design choices of our framework.

4.1. Instance Segmentation on ScanNet

We first evaluate our approach on ScanNet 3D semantic instance segmentation benchmark, where algorithms are evaluated and compared on 18 common object categories. In this task, colored point clouds are provided as input and the goal is to segment out every instance in the scene belonging to the 18 object categories, along with its semantic label. The list of categories captures a variety of objects from small-scale pictures to large-scale doors, making the task very challenging. The average precision (AP) with an IoU threshold 0.5 is used as the evaluation metric. Different from detection tasks, the IoU here is computed based on segmentations instead of bounding boxes, emphasizing

	Mean	cabinet	bed	chair	sofa	table	door	window	bookshelf	picture	counter	desk	curtain	fridge	shower curtain	toilet	sink	bathtub	other
PMRCNN	5.3	4.7	0.2	0.2	10.7	2.0	3.1	0.4	0.0	**18.4**	0.1	0.0	2.0	6.5	0.0	10.9	1.4	33.3	2.1
SGPN	13.3	6.0	36.1	25.7	33.5	16.1	7.9	12.2	14.9	1.3	2.6	0.0	6.2	2.6	0.0	16.1	10.4	19.4	3.8
Ours	**28.5**	**31.1**	**38.9**	**54.6**	**38.7**	**26.7**	**23.3**	**21.3**	**27.9**	2.7	**5.3**	**12.4**	**6.6**	**21.4**	**20.0**	**77.0**	**32.7**	**50.0**	**23.3**

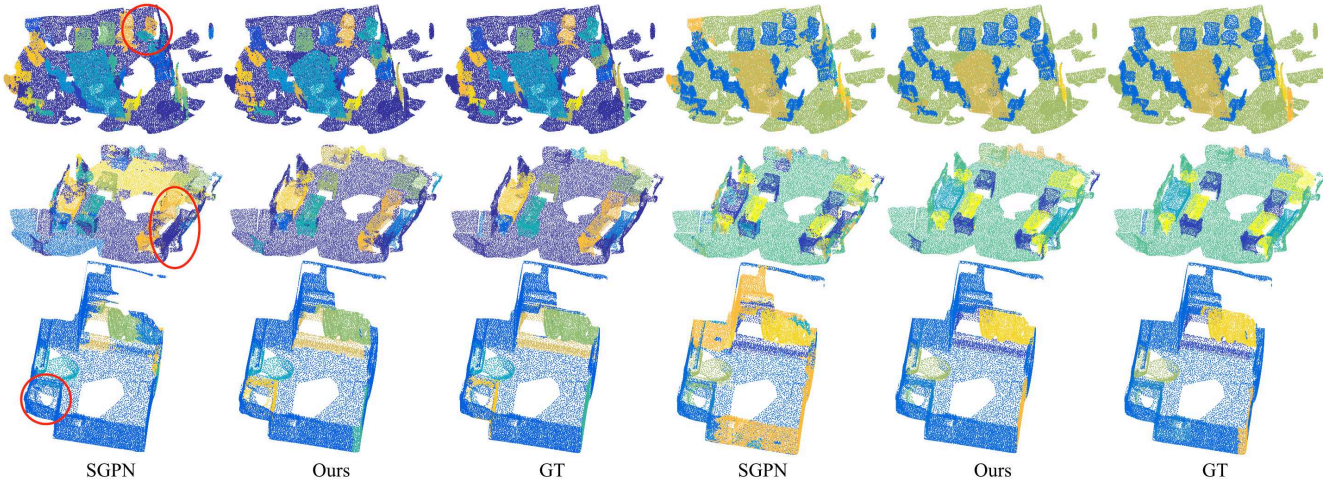Table 1. Instance segmentation results on ScanNet (v2) 3D semantic instance benchmark.

Figure 5. Visualization for ScanNet instance segmentation results. The first three columns show the instance segmentation results where different colors represent different object instances and the last three columns show semantic segmentation results. We highlight SGPN's failure case with red circles in the first column. It is frequent for SGPN to break one object into multiple pieces or miss certain objects.

more on detailed understandings. Unlike previous methods [33, 42], we do not cut scene into cubes but directly consume the whole scene, which avoids the cube merging process and is much more convenient. We compare our approach with the leading players on the ScanNet (v2) benchmark, including SGPN [42] and a projected Mask R-CNN (PMRCNN) approach [1]. SGPN learns to group points from the same instance through a variation of metric learning. PMRCNN first predicts instances in 2D colored images and then project the predictions back to 3D point clouds followed by an aggregation step. To the best of our knowledge, currently these are the only learning based instance segmentation approaches for 3D point cloud that could handle arbitrary object categories in an indoor environment. We report the results in Table 1.

Our R-PointNet outperforms all the previous state-of-the-arts on most of the object categories by a large margin. R-PointNet achieves very high AP for categories with small geometric variations such like toilet since the GSPN only needs to capture a relatively simple object distribution and could generate very high-quality object proposals. For categories requiring strong texture information during segmentation such like window and door, SGPN fails to get good scores since their similarity metric can not effectively encode colors. Our approach achieves much better results on these categories which shows GSPN not only leverages geometry but also the color information while generating object proposals. PMRCNN works in 2D directly instead of in 3D and fails to leverage the 3D information well, leading to very low AP on most of the categories. Interestingly,

PMRCNN achieves the best score for the picture category, which is not surprising since pictures lie on 2D surfaces where appearance instead of geometry acts as the key cue for segmentation. 2D based approaches currently are still more capable of learning from the appearance information.

We also show qualitative comparisons between SGPN and our approach in Figure 5. SGPN needs to draw a clear boundary in the learned similarity metric space to differentiate object instances, which is not easy. We could observe a lot of object predictions either include a partial object or multiple objects. Compared with SGPN, our GSPN generates proposals with much higher objectness, leading to much better segmentation quality. We also find SGPN having a hard time learning good similarity metric when there are a large number of background points since it purely focuses on learning the semantics and similarity for foreground points and ignores background points during training, which could increase false positive predictions on backgrounds. GSPN, on the other hand, explicitly learns the objectness of each proposal and could easily tell foreground from the background. Therefore it won't be easily influenced by the background points.

4.2. Part Instance Segmentation on PartNet

Our R-PointNet could not only handle object instance segmentation in indoor scenes, but also part instance segmentation in an object. Different from objects in scenes, object parts are more structured but less separated, e.g. a chair seat always lies above chair legs while closely connecting with them, which introduces new challenges for in-
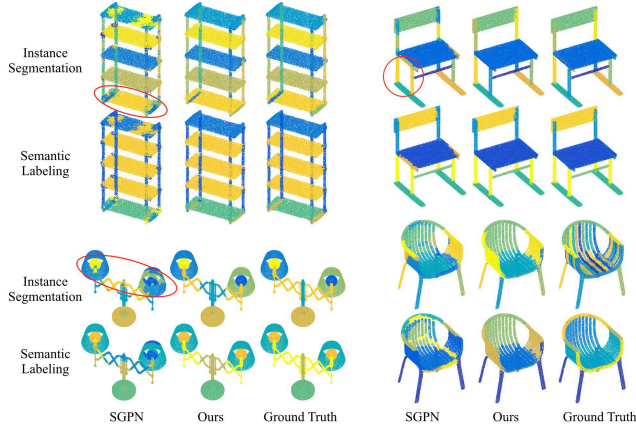
Figure 6. Visualization for part instance segmentation results. As highlighted by the red circles, SGPN does not delineate object parts as good as ours and usually fail to differentiate part instances with the same semantic meaning.

stance segmentation. [27] introduces PartNet, a large scale fine-grained part annotation dataset, where 3D objects from [3] are segmented into fine-grained part instances. Different from previous large-scale part annotation dataset [46], PartNet provides ground truth part instance annotation in addition to semantic labeling and the segmentation granularity is more detailed, making it more suitable for testing instance segmentation approaches. We take the four largest categories from PartNet and evaluate our approach on the semantic part instance segmentation task. Still, we use AP as the evaluation metric with an IoU threshold of 0.5. And we compare with SGPN, which also claims to be able to handle part instance segmentation tasks. We report the qualitative comparison in Table 2 and also visualize the predictions of both our approach and SGPN in Figure 6.

	Chair	Table	Lamp	Storage
SGPN	0.194	0.146	0.144	0.215
Ours	**0.268**	**0.219**	**0.183**	**0.267**

Table 2. Part instance segmentation on PartNet. Our approach outperforms SGPN on all categories.

Our approach outperforms SGPN on all categories by a large margin. As shown in Figure 6, our approach could successfully segment part instances with various scales, from small base panels of the storage furniture in the top left corner to large chair seat in the top right corner. Even for part instances enclosed by other parts, e.g. the light bulbs inside the lamp shades in the bottom left corner, we could still segment them out while SGPN fails. SGPN usually groups part instances with the same semantic label. As highlighted by the red circles. We also show a failure case for both our approach and SGPN in the bottom right corner, where each bar on the back, arm, and seat of the chair are treated as an individual part instance. This causes great challenge to both semantic labeling and instance segmentation. Compared with SGPN, we obtain smoother instance segments with much less noise.

4.3. Instance Segmentation on NYUv2

In this experiment, we focus on colored partial point cloud data lifted from RGBD images. Different from Scan-Net, each RGBD image only captures a scene from a single viewpoint, causing a large portion of data missing in the lifted point cloud. This is especially severe in a cluttered indoor environment with heavy occlusions. To show the effectiveness of our approach, we follow the setting of [42] and conduct instance segmentation for 19 object categories on the colored partial point cloud. AP with an IoU threshold of 0.25 is used as the evaluation metric and again the IoU is computed between predicted segmentations and ground truth segmentations.

Same as [42], to better make use of the color information, we extract features from RGB images directly with a pre-trained AlexNet [18]. We use features from the conv5 layer and we concatenate the image feature with the PointNet++ semantic segmentation feature to augment f_{sem}, which serves as the semantic feature backbone of R-PointNet. The concatenation between the image feature and the point feature happens within the corresponding pixel and 3D point pair, which could be obtained by projecting each 3D point back to 2D and search for its closest pixel. We compare our approach with SGPN-CNN [42], the previous state-of-the-art approach on this partial point cloud instance segmentation task. In SGPN-CNN, 2D CNN features are incorporated into SGPN for better leverage of the color information. While conducting the comparison with [42], we found issues within the authors' evaluation protocol. With the help of the authors, we re-evaluate SGPN-CNN and report the quantitative comparison in Table 3.

Our R-PointNet outperforms SGPN-CNN with respect to mAP and provides better results on 14/19 classes. Even on partial point cloud with severe data missing, R-PointNet still captures the shape prior, generates object proposals and predicts final segmentations reasonably well. R-PointNet achieves higher scores on categories with small geometric variations such like bathtub and toilet, whose shape distributions are relatively easier for GSPN to capture.

4.4. Ablation Study

To validate various design choices of ours, we conduct ablation study on the ScanNet (v2) validation set and discuss the details below. We also refer readers to the supplementary material for more ablation studies such as the generative model design.

Comparison with Other 3D Proposal Approaches We conduct ablation study by replacing GSPN with other types of object proposal networks. To be specific, we implemented two alternatives. One is to directly regress 3D bounding box for each object, mimicking the Mask R-CNN. For each seed point in the 3D space, we define 27 axis-aligned anchor boxes associated with three scales and 9 aspect ratios. We design a proposal network, which is essentially an MLP, for bounding box regression. It takes in the context features $f_{\hat{c}}$ and directly regress center shift,

	Mean	bath-tub	bed	book-shelf	box	chair	coun-ter	desk	door	dres-ser	gar-bin	lamp	moni-tor	night-stand	pil-low	sink	sofa	table	TV	toilet
SGPN-CNN	33.6	45.3	**62.5**	**43.9**	0.0	45.6	40.7	30.0	20.2	42.6	**8.8**	28.2	15.5	43.0	30.4	**51.4**	**58.9**	25.6	6.6	39.0
Ours	**39.3**	**62.8**	51.4	35.1	**11.4**	**54.6**	**45.8**	**38.0**	**22.9**	**43.3**	8.4	**36.8**	**18.3**	**58.1**	**42.0**	45.4	54.8	**29.1**	**20.8**	**67.5**

Table 3. Instance segmentation results on NYUv2 dataset.

box deltas and objectness score for each anchor box, which is used for anchor box refinement. Then we apply non-maximum suppression and point RoI align to the refined anchor boxes and feed them into the final classification, box regression, and segmentation heads. The other way is to conduct a binary segmentation within a specific context centered at each seed point and convert the resulting segmentation into axis-aligned bounding box for object proposals. We choose the largest context c_3 in our case to guarantee that the largest object in the scene could be fully included by the context. We compare the RoI generated by GSPN with the above two alternatives through two evaluation metrics. One is the mean 3D IoU (mIoU) between the proposal bounding boxes and ground truth bounding boxes at all seed points. The other is the final instance segmentation mAP. For direct bounding box regression approach, we select the refined anchor box with the highest objectness score for mIoU computation. We report the quantitative comparison in Table 4 and visualize proposals from different approaches in Figure 7.

Our GSPN achieves the best performance with respect to both evaluation metrics. It generates better object proposals more similar to the underlying objects, which indeed improves the final segmentation performance. Both binary segmentation based object proposal and bounding box regression based object proposals could generate boxes covering partial or multiple objects, influencing their usage for the downstream segmentation task.

	mIoU	AP	AP@0.5	AP@0.25
Bbox Reg	0.514	15.8	33.1	51.3
Binary Seg	0.543	14.9	30.9	47.7
Ours	**0.581**	**19.3**	**37.8**	**53.4**

Table 4. Evaluation of different 3D proposal approaches. Compared with straightforward bounding box regression and binary segmentation based bounding box proposal, our GSPN not only generates object proposals overlapping more with the ground truth objects, but also largely improves the final segmentation mAP.

Choices of Feature Backbone We use a combination of instance sensitive context feature $f_{\hat{c}}$ and semantic feature f_{sem} as the feature backbone, which both play an important role in achieving good instance segmentation. We validate their importance by removing each of them and evaluating the influence to the final segmentation mAP. The results are shown in Table 5. It can be seen that removing either of them from the backbone will cause a performance degeneration. This confirms that the instance sensitive feature GSPN learns is complementary to the semantic feature
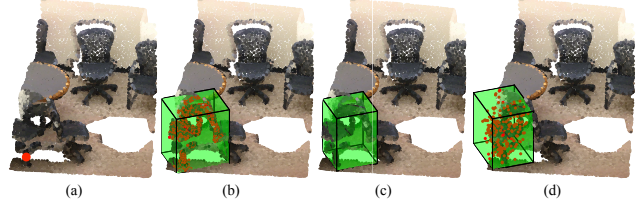

Figure 7. Visualization of object proposals with the induced 3D bounding boxes generated by different approaches. (a) shows the input point cloud and the red point on the chair leg serves as the seed point. We show proposals generated by (b) binary segmentation based object proposal, (c) direct bounding box regression, (d) GSPN. The 3D bounding boxes given by (b) and (c) include other objects while GSPN generates a faithful approximation to the underlying object thus successfully avoids including other objects.

f_{sem}, and using both is important. In addition, we also remove the pretraining step for f_{sem} and train the pointnet++ semantic segmentation network in an end-to-end fashion with R-PointNet. We observe a performance drop as well as is shown in Table 5.

	AP	AP@0.5	AP@0.25
w/o $f_{\hat{c}}$	0.178	0.349	0.515
w/o f_{sem}	0.161	0.319	0.477
w/o pretraining	0.180	0.364	0.517
Ours	**0.191**	**0.376**	**0.541**

Table 5. Comparison of different choices for the feature backbone. Both context feature $f_{\hat{c}}$ and semantic feature f_{sem} play important roles in our feature backbone. We also find pretraining the semantic feature with a semantic segmentation task improves the segmentation performance.

5. Conclusion

We present GSPN, a novel object proposal network for instance segmentation in 3D point cloud data. GSPN generates good quality object proposals with high objectness, which could greatly boost the performance of an instance segmentation framework. We demonstrate how GSPN could be incorporated into a novel 3D instance segmentation framework: R-PointNet, and achieve state-of-the-art performance on several instance segmentation benchmarks.

References

[1] Scannet 3d semantic instance benchmark leader board. http://kaldir.vc.in.tum.de/scannet_benchmark, Accessed 2018-11-16, 6pm. 6

[2] S. R. Bowman, L. Vilnis, O. Vinyals, A. M. Dai, R. Jozefowicz, and S. Bengio. Generating sentences from a continuous space. *arXiv preprint arXiv:1511.06349*, 2015. 5

[3] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015. 5, 7

[4] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia. Multi-view 3d object detection network for autonomous driving. In *IEEE CVPR*, volume 1, page 3, 2017. 2

[5] A. Dai, A. X. Chang, M. Savva, M. Halber, T. A. Funkhouser, and M. Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *CVPR*, volume 2, page 10, 2017. 5

[6] J. Dai, K. He, Y. Li, S. Ren, and J. Sun. Instance-sensitive fully convolutional networks. In *European Conference on Computer Vision*, pages 534–549. Springer, 2016. 2

[7] Z. Deng and L. J. Latecki. Amodal detection of 3d objects: Inferring 3d bounding boxes from 2d ones in rgb-depth images. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, page 2, 2017. 1, 2, 5

[8] H. Fan, H. Su, and L. J. Guibas. A point set generation network for 3d object reconstruction from a single image. In *CVPR*, volume 2, page 6, 2017. 2, 3

[9] R. Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015. 2, 4

[10] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014. 2

[11] R. Girshick, F. Iandola, T. Darrell, and J. Malik. Deformable part models are convolutional neural networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 437–446, 2015. 5

[12] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014. 2

[13] I. Gulrajani, K. Kumar, F. Ahmed, A. A. Taiga, F. Visin, D. Vazquez, and A. Courville. Pixelvae: A latent variable model for natural images. *arXiv preprint arXiv:1611.05013*, 2016. 2

[14] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik. Hypercolumns for object segmentation and fine-grained localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 447–456, 2015. 4

[15] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, pages 2980–2988. IEEE, 2017. 1, 2, 3, 4, 5

[16] W. James. The principles of psychology, vol. 2. ny, us: Henry holt and company, 1890. 1

[17] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 2

[18] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012. 7

[19] A. Kundu, Y. Li, and J. M. Rehg. 3d-rcnn: Instance-level 3d object reconstruction via render-and-compare. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3559–3568, 2018. 2

[20] Y. Li, H. Qi, J. Dai, X. Ji, and Y. Wei. Fully convolutional instance-aware semantic segmentation. *arXiv preprint arXiv:1611.07709*, 2016. 1, 2

[21] T.-Y. Lin, P. Dollár, R. B. Girshick, K. He, B. Hariharan, and S. J. Belongie. Feature pyramid networks for object detection. In *CVPR*, volume 1, page 4, 2017. 2

[22] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár. Focal loss for dense object detection. *IEEE transactions on pattern analysis and machine intelligence*, 2018. 2

[23] O. Litany, T. Remez, D. Freedman, L. Shapira, A. Bronstein, and R. Gal. Asist: Automatic semantically invariant scene transformation. *Computer Vision and Image Understanding*, 157:284–299, 2017. 2

[24] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia. Path aggregation network for instance segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8759–8768, 2018. 1

[25] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016. 2

[26] M. Mirza and S. Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014. 2

[27] K. Mo, S. Zhu, A. X. Chang, L. Yi, S. Tripathi, L. J. Guibas, and H. Su. Partnet: A large-scale benchmark for fine-grained and hierarchical part-level 3d object understanding. *arXiv preprint arXiv:1812.02713*, 2018. 5, 7

[28] A. Mousavian, D. Anguelov, J. Flynn, and J. Košecká. 3d bounding box estimation using deep learning and geometry. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pages 5632–5640. IEEE, 2017. 2

[29] C. Nash and C. K. Williams. The shape variational autoencoder: A deep generative model of part-segmented 3d objects. In *Computer Graphics Forum*, volume 36, pages 1–12. Wiley Online Library, 2017. 2

[30] P. K. Nathan Silberman, Derek Hoiem and R. Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012. 5

[31] P. O. Pinheiro, T.-Y. Lin, R. Collobert, and P. Dollár. Learning to refine object segments. In *European Conference on Computer Vision*, pages 75–91. Springer, 2016. 2

[32] C. R. Qi, W. Liu, C. Wu, H. Su, and L. J. Guibas. Frustum pointnets for 3d object detection from rgb-d data. *arXiv preprint arXiv:1711.08488*, 2017. 2

[33] C. R. Qi, H. Su, K. Mo, and L. J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 1(2):4, 2017. 1, 2, 6

[34] C. R. Qi, L. Yi, H. Su, and L. J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in Neural Information Processing Systems*, pages 5099–5108, 2017. 1, 2

[35] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016. 2

[36] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015. 2, 4

[37] G. Riegler, A. O. Ulusoy, and A. Geiger. Octnet: Learning deep 3d representations at high resolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 3, 2017. 2

[38] K. Sohn, H. Lee, and X. Yan. Learning structured output representation using deep conditional generative models. In *Advances in Neural Information Processing Systems*, pages 3483–3491, 2015. 2, 3

[39] S. Song and J. Xiao. Deep sliding shapes for amodal 3d object detection in rgb-d images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 808–816, 2016. 1, 2

[40] H. Su, S. Maji, E. Kalogerakis, and E. Learned-Miller. Multiview convolutional neural networks for 3d shape recognition. In *Proceedings of the IEEE international conference on computer vision*, pages 945–953, 2015. 2

[41] J. R. Uijlings, K. E. Van De Sande, T. Gevers, and A. W. Smeulders. Selective search for object recognition. *International journal of computer vision*, 104(2):154–171, 2013. 1

[42] W. Wang, R. Yu, Q. Huang, and U. Neumann. Sgpn: Similarity group proposal network for 3d point cloud instance segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2569–2578, 2018. 2, 6, 7

[43] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1912–1920, 2015. 2

[44] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox. Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. *Robotics: Science and Systems (RSS)*, 2018. 5

[45] B. Yang, W. Luo, and R. Urtasun. Pixor: Real-time 3d object detection from point clouds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7652–7660, 2018. 2

[46] L. Yi, V. G. Kim, D. Ceylan, I. Shen, M. Yan, H. Su, C. Lu, Q. Huang, A. Sheffer, L. Guibas, et al. A scalable active framework for region annotation in 3d shape collections. *ACM Transactions on Graphics (TOG)*, 35(6):210, 2016. 7

[47] L. Yi, H. Su, X. Guo, and L. J. Guibas. Syncspeccnn: Synchronized spectral cnn for 3d shape segmentation. In *CVPR*, pages 6584–6592, 2017. 2

[48] Y. Zhou and O. Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. *arXiv preprint arXiv:1711.06396*, 2017. 1, 2