This ICCV paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

Query Refinement Transformer for 3D Instance Segmentation

Jiahao Lu[1], Jiacheng Deng[1], Chuxin Wang[1], Jianfeng He[1], Tianzhu Zhang[1,2,†]
[1]University of Science and Technology of China, [2]Deep Space Exploration Lab
{lujiahao, dengjc, wcx0602, hejf}@mail.ustc.edu.cn, tzzhang@ustc.edu.cn

Abstract

3D instance segmentation aims to predict a set of object instances in a scene and represent them as binary foreground masks with corresponding semantic labels. However, object instances are diverse in shape and category, and point clouds are usually sparse, unordered, and irregular, which leads to a query sampling dilemma. Besides, noise background queries interfere with proper scene perception and accurate instance segmentation. To address the above issues, we propose the Query Refinement Transformer termed QueryFormer. The key to our approach is to exploit a query initialization module to optimize the initialization process for the query distribution with a high coverage and low repetition rate. Additionally, we design an affiliated transformer decoder that suppresses the interference of noise background queries and helps the foreground queries focus on instance discriminative parts to predict final segmentation results. Extensive experiments on ScanNetV2 and S3DIS datasets show that our QueryFormer can surpass state-of-the-art 3D instance segmentation methods.

1. Introduction

3D instance segmentation is a fundamental task for 3D scene understanding, which aims to predict the semantic labels and the binary foreground masks for every object in the scene simultaneously. With the popularity of AR/VR [31], 3D indoor scanning [21], and autonomous driving [46], 3D instance segmentation has become a key technology facilitating scene understanding. However, the complex layout of the scene and the variety of object categories pose severe challenges to 3D instance segmentation in segmenting similar objects and accurate point cloud masking.

To overcome the above challenges, a series of 3D instance segmentation methods [11, 24, 4, 23, 39, 36] have been proposed. Generally, these methods can be divided into three categories: proposal-based [45, 11, 24], grouping-based [17, 4, 23, 39], and query-based [36]. Proposal-based methods [45, 11, 24] extract 3D bounding boxes and utilize a mask learning branch to predict the object mask inside each box. Grouping-based methods [17, 4, 23, 39] rely on a bottom-up pipeline that generates predictions for each point (e.g., semantic categories and geometric offsets) and then groups the points into instances. However, proposal-based and grouping-based methods strongly rely on high-quality proposals and require manual selection of geometric properties and tuning of hyperparameters. These drawbacks and limitations prompt query-based methods to be proposed and receive extensive attention from researchers. Query-based methods are regarded as a class of transformer query-based methods [2, 5] in which each object instance is represented as an instance query. Query-based methods require a large number of queries spread throughout the scene to cover a large part of foreground objects in the scene so that there are one or even more queries on each covered object instance. Then a transformer decoder learns the instance queries by iteratively focusing on multi-level point cloud features. Eventually, the instance queries aggregate point cloud features to produce all instance masks in parallel.

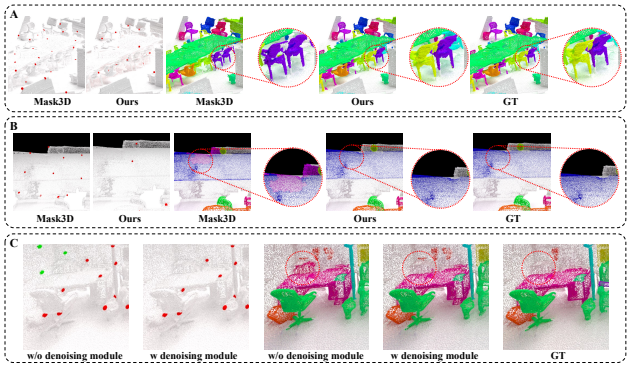

Figure 1. **The visualization of query distribution and segmentation results.** A. The query initialization module samples a query on each of the two chairs to distinguish the two instances precisely. B. Repeated queries incorrectly split the complete board into two pieces. C. The denoising module removes many noise background queries and optimizes the final segmentation results.

Based on the discussion of the above approaches, query-based methods show great promise for 3D instance segmentation tasks by offering superior performance while not re-

*Corresponding Author

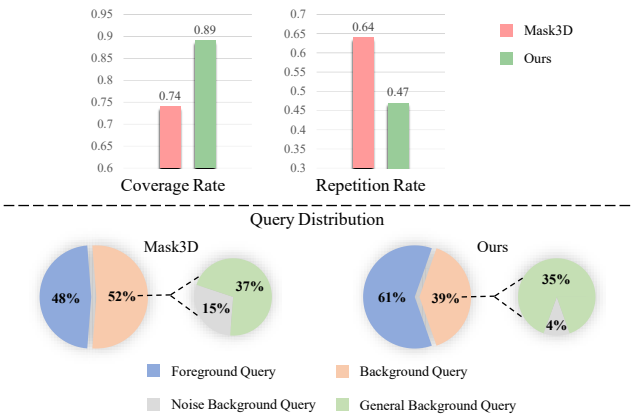

Figure 2. **Statistics on coverage rate&repetition rate and query distribution.** For query distribution, our method has more foreground distribution and less noise background queries and FPS initialization exists many noise background queries. Besides, our method guarantees a high coverage rate of foreground object instances while maintaining a low repetition rate.

quiring high-quality 3D proposals and reducing the need for extensive manual setup. By analyzing the query-based method, we summarize the following two core points of query-based 3D instance segmentation: 1) *How to effectively initialize queries?* For the query-based method, we conclude that coverage and repetition rates are two significant factors affecting query initialization. The coverage rate is defined as the proportion of foreground object instances with queries distributed to the total foreground object instances in the scene. As the statistical results in Figure 2, the coverage rate of FPS [35] is 74% (Ours is 89%), and the lower coverage rate causes some objects to be missed. For example, as shown in Figure 1(A), the two chairs are incorrectly segmented into one whole object because one chair is not covered by queries. The repetition rate is defined as the proportion of queries that have other queries coexisting in the same foreground object instance to the total number of foreground queries. The higher repetition rate (64% *vs* our 47%) results in low query utilization and unnecessary overlapping computation. Besides, repeated queries contradict one-to-one bipartite matching mechanism and will destroy the integrity of segmentation results. As shown in Figure 1(B), multiple queries split the board into two parts. In practice, a high coverage rate often leads to a high repetition rate, and a low repetition rate is accompanied by a low coverage rate, so it is necessary to design an initialization strategy that can guarantee a high coverage rate of foreground object instances while maintaining a low repetition rate. 2) *How to suppress the interference of noise background queries on accurate segmentation?* Query-based methods yield outstanding performance but sample more queries than object instances in the scene. For example, sampling results by the Mask3D [36] in Figure 2 show that 48% of the queries are distributed on foreground objects, while 15% of the 52% of the background queries are distributed near the instances. The background queries distributed near the instances are called noise background queries because the closer the queries are to the instances, the more negatively they affect the accurate segmentation of the instances. As shown in Figure 1(C), the foreground queries(red) incorrectly segment a part of the background as a foreground instance due to the interference of the noise background queries(green). Therefore, in addition to ensuring that foreground queries accurately segment foreground instances, it is equally essential to suppress the interference of noise background queries on the segmentation task.

Inspired by the above discussion, we propose a Query Refinement Transformer for 3D Instance Segmentation, named QueryFormer, including a point encoder, a query initialization module, and an affiliated transformer decoder. We use a generic U-net point cloud representation network [7] as the point encoder of the method. In the query initialization module, we aggregate the superpoint features [20] and the multi-level features extracted by the point encoder to the seed points through the Aggregation Module. The Set Grouping module (SG) is designed to achieve a high coverage rate and low repetition rate distribution of seed points on instances. Specifically, the set grouping module shifts seed points to the centers of the objects to ensure a high coverage rate and filters out duplicate seed points to decrease the repetition rate and minimize the computational cost. The affiliated transformer decoder is dedicated to suppressing the impacts of noise background queries on the instance segmentation. Specifically, we generate some perturbed ground-truth centers around the instance centers during training. The perturbed centers are encoded in the same space as the query and input to the decoder for mapping to a consistent feature space. For each foreground instance, a unique query is matched by the Hungarian matching [18]. This query forms positive pairs with perturbed centers, while unmatched queries form several negative pairs with perturbed centers. We design a denoising module with contrastive loss [30] to drive the unmatched noise background queries away from the instance to reduce interference with the instance mask prediction.

In summary, the main contributions of this work are as follows: (i) We design a query initialization module to optimize the initialization process for a high coverage and low repetition rate query distribution, which further helps the proposed affiliated transformer decoder to achieve more accurate results. (ii) The proposed affiliated transformer decoder suppresses the interference of noise background queries and helps the foreground queries focus on discriminative parts for more accurate instance segmentation. (iii) Extensive experimental results on two standard benchmarks, ScanNetV2 [8] and S3DIS [1], demonstrate that the proposed model performs favorably against state-

of-the-art 3D instance segmentation methods.

2. Related Work

In this section, we briefly overview related works on 3D instance segmentation, including proposal-based methods, grouping-based methods, and instance segmentation with transformer.

Proposal-based Methods. Existing proposal-based methods are greatly influenced by the success of Mask R-CNN [14] for 2D instance segmentation. GSPN [45] takes an analysis-by-synthesis strategy to generate high-quality 3D proposals, which are refined by a region-based PointNet [34]. 3D-BoNet [44] uses PointNet++ [35] to extract features from point clouds and applies Hungarian matching [18] to generate 3D bounding boxes. GICN [24] approximates the instance center of each object as a Gaussian distribution. 3D-MPA [11] predicts centers of instances and employs a graph-based convolutional network to cluster points near the centers to refine proposal features. The proposal-based methods have high expectations for the quality of proposals.

Grouping-based Methods. Grouping-based methods produce per-point predictions, such as semantic categories and geometric offsets, and then group points into instances. To group points, MTML [19] uses a multi-task learning strategy. PointGroup [17] segments objects on the original and offset-shifted point clouds and uses the ScoreNet to predict scores for instances. HAIS [4] extends PointGroup by absorbing surrounding fragments of instances and then refining the instances based on intra-instance prediction. SSTNet [23] constructs a tree network from precomputed superpoints and splits non-similar nodes to get object instances. SoftGroup [39] groups on soft semantic scores instead of hard semantic prediction and processes each proposal to refine positive samples and suppress negative ones. Although the grouping-based approaches require many manual settings, they have dominated the field until recently.

Instance Segmentation with Transformer. Transformer [38] introduces the self-attention mechanism to model long-range dependencies and has been widely applied in computer vision tasks such as image classification [10, 3], object detection [2, 9], and segmentation [48, 6, 5]. Recently, DETR [2] has been proposed as a new paradigm to use object queries for object detection in an image. Based on the set prediction mechanism proposed in DETR, MaskFormer [6] employs a transformer decoder to compute a set of pairs, each consisting of a class prediction and a mask embedding vector, to solve both semantic and instance segmentation tasks in a unified manner. Mask2Former [5] outperforms state-of-the-art specialized architectures on all considered datasets for 2D image semantic, instance, and panoptic segmentation. The success of transformer has come to prominence in some 3D point cloud tasks such as 3D object detection [28, 25] and 3D semantic segmentation [47]. However, applications of transformer to 3D instance segmentation tasks have yet to be exploited profoundly. DyCo3D [15] uses a transformer at the bottleneck of the feature backbone to increase the receptive field size. Mask3D [36] proposes the first transformer framework for 3D instance segmentation and achieves the state-of-the-art performance. In Mask3D, each object instance is represented as an instance query, and a vanilla transformer decoder is applied to predict instance masks. In this paper, we design a novel query initialization module to effectively initialize queries and formulate a denoising module with contrastive loss to suppress the interference of noise background queries. Eventually, QueryFormer achieves superior results with better query refinement.

3. Method

This section provides the details of our proposed method. We first illustrate the overview of the proposed method in Section 3.1, and then introduce the proposed Query Initialization Module in Section 3.2. In Section 3.3, we explain how the Affiliated Transformer Decoder works to suppress the undesirable interference of noise background queries on isntance segmentation. Finally, we elaborate the model training and inference in Section 3.4.

3.1. Overview

The goal of 3D instance segmentation is to determine the categories and binary masks of all foreground objects in the scene. Unlike the bounding boxes of 3D object detection, 3D instance segmentation requires further accurate delineation of the mask. The architecture of our method is illustrated in Figure 3(a). Assuming that the input point cloud has N points, each point contains position x, y, z and color r, g, b information. First, we use a 3D-UNet based on sparse convolution for multi-level feature extraction F_0, F_1, F_2. To select the query points with high coverage rate and low repetition rate from the scene, we input the seed points P_{seed} sampled by Farthest Point Sampling (FPS) into the Query Initialization Module (QIM). Through feature fusion in the aggregation module and filtering in the set grouping, we get the high quality query points Q_1. Finally, we input the obtained query points into the proposed Affiliated Transformer Decoder (ATD) to get the final predictions.

3.2. Query Initialization Module

Before we introduce the module, we give the definition of two concepts (coverage rate and repetition rate of query points). The Coverage Rate (CR) of query points indicates the number N_p of instances contained in the query points as a percentage of the total number N_{ins} of instances in the

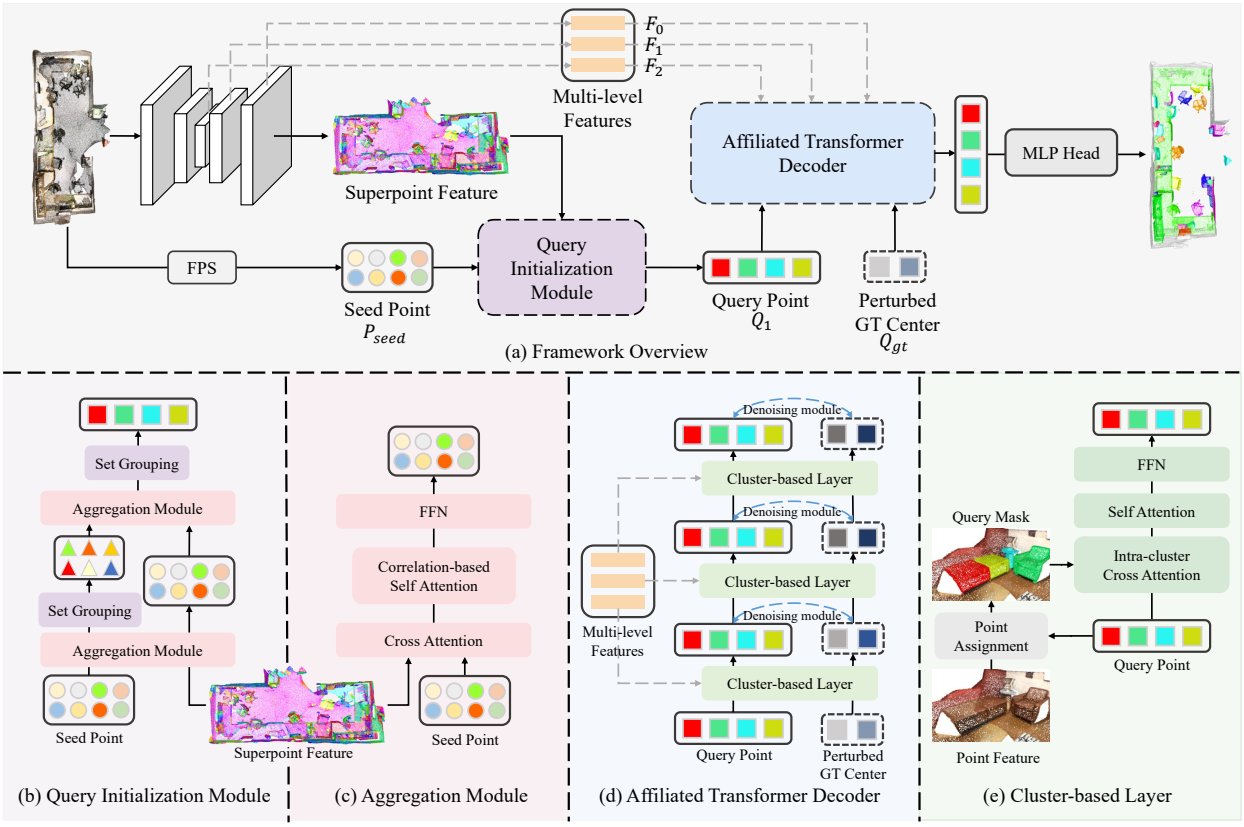

Figure 3. **The overview of our query refinement transformer for 3D instance segmentation.** As shown in (a), we first generate seed points P_{seed} by sampling from the raw point cloud. Next, we design the Query Initialization Module to improve the coverage rate and reduce the repetition rate of seed points. Afterwards, we send the generated high-quality query points Q_1 to the Affiliated Transformer Decoder to generate the final instance masks. Meanwhile, we design the Denoising Module to alleviate the interference of noise background queries on instance mask prediction. The details of each module are shown in (b), (c), (d) and (e).

scene, which can be formulated as:

$$CR = N_p / N_{ins}. \tag{1}$$

The Repetition Rate (RR) of query points indicates the proportion of instances covered by repetition in the query points, which can be formulated as:

$$RR = (N_f - N_p)/N_f, \tag{2}$$

where N_f denotes the number of foreground points in the query points. The previous methods [36] usually use FPS to select query locations. However, FPS makes it challenging to balance the coverage and repetition rates. As shown in Figure 3(b), to solve this problem, we propose the Query Initialization Module, which contains two main submodules, the aggregation module and the set grouping module.

Aggregation Module. In feature aggregation, we design a module that allows the point features in the same instance to fuse while reducing the interference of point features in other adjacent instances. As shown in Figure 3(c), the module contains three parts, cross-attention layer, correlation-based self-attention layer and feed-forward layer. First, we use the cross-attention mechanism to update the features of the seed points P_{seed} using the features of the superpoints in the scene. Here we initialize the seed point features by encoding the coordinates of the seed points. After that, the seed point features are input to the correlation-based self-attention mechanism. Unlike the previous self-attention mechanism, we use the information cues from the correlation relations in the attention map for feature fusion, and this design promotes the appropriate correlations of relevant seed-seed pairs and suppress the erroneous correlations of irrelevant seed-seed pairs. Specifically, we first calculate the similarity matrix of seed point features, as follows:

$$M = \frac{QK^T}{\sqrt{C}} w_M + P, \tag{3}$$

where w_M is a linear projection, P represents the position embeddings of seed points. We use the similarity matrix of features as the new features to calculate the similarity, with the following formula:

$$M^c = \mathrm{Softmax}(\frac{\mathrm{Q_M K_M^T}}{\sqrt{\mathrm{C_M}}})\mathrm{M}. \tag{4}$$

Then we calculate the updated features F^c_{seed} for each seed point based on the similarity matrix M^c. Finally we input F^c_{seed} into the feed-forward layer and then predict the mask, category and IoU scores by different MLP heads.

Many-to-one Matching. Different from the one-to-one matching[18] used previously, we use many-to-one match-

ing in the aggregation module. Formally, we introduce a pairwise cost matrix CM_k to evaluate the similarity of seed point q and the k-th ground-truth. CM_k is determined by classification probability, superpoint mask cost and distance cost. We will introduce how to obtain the cost matrix in the supplementary material. With the cost matrix, we assign corresponding instance to each seed point,

$$Ins_q = \begin{cases} \arg\min_k CM_k, if \min_k D_k < \tau, \\ -1, else, \end{cases} \tag{5}$$

where τ represents a threshold, D_k represents the distance between the coordinate of seed point q and the centerpoint of the k-th ground-truth, and -1 represents that no instance matches the seed point.

Set Grouping Module. The set grouping module aims to improve coverage rate and reduce repetition rate, which contains shift and IoU-guided NMS [29] operations. The shift operation offsets seed points to the centers of the instances to ensure a high coverage rate so that most foreground instances can have a corresponding seed point. While the IoU-guided NMS operation filters out duplicate seed points to ensure that each foreground instance retains only one high-quality seed point thus reducing the repetition rate of seed points. Specifically, we first use a MLP to predict each seed point q relative to its matching instance Ins_q coordinate offset of the center point. After obtaining this coordinate offset, add it to the original coordinate p:

$$p = MLP(s) + p. \tag{6}$$

In this way, we can not only shift some background points to the foreground, but also close the points on the same instance, which will promote these points to learn similar features and help NMS remove redundant seed points. Then we get the corresponding instance mask M_{ins} of the seed point q by calculating the dot product between superpoint features and seed point features. And we obtain the final mask through sigmoid and a threshold of 0.5, as follows:

$$M_b = \phi(M_{ins}) > 0.5. \tag{7}$$

Finally, we adopt the IoU-guided NMS to filter out a large number of redundant seed points. Here, the confidence of NMS is the multiplication of class score and IoU score. In order to enable the network to operate in parallel, we add extra seed points to satisfy the same number. Specifically, we do a $\bigcup$ operation on the mask M_b of all seed points and then operate FPS on the complement M^- of the whole scene to sample extra seed points.

3.3. Affiliated Transformer Decoder

In the Query Initialization Module, we have obtained queries with high coverage rate and low repetition rate through the Aggregation Module and Set Grouping Module. In this section, we describe how the Affiliated Transformer Decoder generates the final instance segmentation results with these queries. As shown in Figure 3(d), the Affiliated Transformer Decoder mainly contains the Cluster-based Layer and the Denoising Module.

Cluster-based Layer. Existing query-based methods [36] tend to compute similarities across the entire point cloud, which introduces large computational redundancy. Therefore, we propose the Cluster-based Layer. As shown in Figure 3(e), the method is based on the idea of clustering, assigning each point in the scene to its most similar query.

Specifically, we first calculate the similarity of the backbone features and the query features. Next, we assign each backbone feature via $\arg\max$ operation as following,

$$Idx_k = \arg\max_i M_{i,k}, \tag{8}$$

where $M_{i,k}$ is a dot product of query i and backbone feature k. Therefore, we divide all backbone features into disjoint sets by clustering. Next, we do cross-attention for queries and their corresponding sets. By this way, we only need to calculate attention map on some similar backbone features for each query. Finally we input quries into the self-attention layer and the feed-forward layer.

One-to-one Matching. In this section, we use one-to-one matching to replace many-to-one matching in Section 3.2. Concretely, we obtain the cost matrix CM the same as Aggregation Module but replace $\arg\min$ with Hungarian matching[18] to obtain the query-GT pairs.

Denoising Module. In section 3.2, we shift the background points around the foreground points to the center of the instance to improve the coverage rate. However, during the offset, not every point is well migrated to the center due to the different scales and classes of objects [40]. This leads to some background points that may affect the information aggregation of the foreground points. Therefore, we deal with the interference caused by background point offset in this module. We take the center of the instance from ground-truth and resample it, as follows:

$$Q_{gt} \sim N(C_{gt}, \sigma^2), \tag{9}$$

where N means the Gaussian distribution and σ represents deviation. We encode the perturbed GT centers Q_{gt} together with the query points Q_1 obtained in the query initialization module and feed them into the affiliated transformer decoder. Then, we match the query outputs with the ground-truth in the dataset. The query with the minimal matching cost forms several positive pairs with the ground-truth perturbed centers. We encourage the output features of the positive queries to be consistent with the output features of the corresponding center points and drive the unmatched noise background queries away from the instance to reduce interference with the instance mask prediction. Therefore, we use the contrastive loss. Suppose the number of queries is K, the number of ground-truth is N_{gt}, ground-truth j generates β perturbed GT centers, query i and the perturbed

GT centers of ground-truth j are positive pairs,

$$L_{cont} = -\frac{1}{|\beta N_{gt}|}\sum_{j=1}^{N_{gt}}\sum_{m=1}^{\beta}\log(\frac{\exp(d(q_i, q_{j,m}^{gt})/\varepsilon)}{\sum_{s=1}^{K}\exp(d(q_s, q_{j,m}^{gt})/\varepsilon)}), \tag{10}$$

where $d(.,.)$ is a distance measurement and ε is the temperature in contrastive learning.

3.4. Training and Inference

As to query initialization module, only the aggregation modules are supervised. We assign the unmatched queries with $N_{class}+1$ class label and compute the cross-entropy loss L_{cls} for each query. Then we compute the superpoint mask loss which consists of binary cross-entropy loss L_{bce} and dice loss L_{dice} [27] for each query-GT pair. Meanwhile, we compute the score loss L_{sco} using binary cross-entropy loss. Suppose that the Hungarian matching assigns the k-th ground-truth to the i-th query,

$$L_{sco,i} = BCE(o_i, IoU_{i,k}), \tag{11}$$

where o_i represents the IoU prediction of query i. In addition, we add the offset loss L_{off} which uses L1loss to close the distance between the coordinates of queries and the instance center points.

$$L_{off,i} = \left|p_i - p_k^{GT}\right| \tag{12}$$

The overall loss is defined as:

$$\begin{aligned} L = \lambda_1 \cdot L_{dice} + \lambda_2 \cdot L_{bce} + \lambda_3 \cdot L_{cls} \\ +\lambda_4 \cdot L_{off} + \lambda_5 \cdot L_{sco}. \end{aligned} \tag{13}$$

To Affiliated Transformer Decoder, we also add losses after each Cluster-based Layer. Compared to L, we add contrastive loss which is introduced in Section 3.3.

$$L^* = L + \lambda_6 \cdot L_{cont} \tag{14}$$

Following DN-DETR[22], we use the same losses as query-GT pairs to supervise the perturbed GT centers. This can ensure the perturbed GT centers would not diverge.

During inference, it should be noted that perturbed GT centers will not be generated, that is, we only infer queries generated by Query Initialization Module.

4. Experiments

4.1. Experimental Setup

Dataset and Metrics. We conduct our experiments on ScanNetV2 [8] and S3DIS [1] datasets. ScanNetV2 includes 1,613 scenes with 18 instance categories. Among them, 1,201 scenes are used for training, 312 scenes are used for validation, and 100 scenes are used for test. S3DIS is a large-scale indoor dataset collected from six different areas. It contains 272 scenes with 13 instance categories. Following previous works [39], we evaluate our approach: testing on Area 5 and 6-fold cross-validation. AP@25 and AP@50 represent the average precision scores with IoU thresholds 25% and 50%, and mAP represents the average of all the APs with IoU thresholds ranging from 50% to 95% with a step size of 5%. On ScanNetV2, we report mAP, AP@50 and AP@25. Moreover, we also report the Box AP@50 and AP@25 results following SoftGroup [39] and DKNet [43]. On S3DIS, we report mAP, AP@50, mean precision (mPrec), and mean recall (mRec).

Table 1. **Comparison on ScanNetV2 benchmark.**

Method	ScanNetV2				
	mAP	AP@50	AP@25	Box AP@50	Box AP@25
F-PointNet [33]	/	/	/	10.8	19.8
GSPN [45]	/	37.8	53.4	17.7	30.6
3D-SIS [16]	/	18.7	35.7	22.5	40.2
VoteNet [32]	/	/	/	33.5	58.6
3D-MPA [11]	35.3	51.9	72.4	49.2	64.2
DyCo3D [15]	40.6	61.0	/	45.3	58.9
PointGroup [17]	34.8	56.9	71.3	48.9	61.5
MaskGroup [49]	42.0	63.3	74.0	/	/
OccuSeg [13]	44.2	60.7	/	/	/
HAIS [4]	43.5	64.4	75.6	53.1	64.3
SSTNet [23]	49.4	64.3	74	52.7	62.5
SoftGroup [39]	45.8	67.6	78.9	59.4	71.6
DKNet [43]	50.8	66.9	76.9	59.0	67.4
Mask3D [36]	55.2	73.7	82.9	56.6	71.0
Ours	**56.5**	**74.2**	**83.3**	**61.7**	**73.4**

Implementation Details. We train our model on a single RTX3090 with a batch size of 5 for 600 epochs. We use AdamW [26] and a one-cycle learning rate schedule [37] with a maximal learning rate of 10^{-4}. We voxelize the point clouds with the size of 0.02m. For a fair comparison, the point encoder is a Minkowski Res16UNet34C [7], which is the same as Mask3D [36]. For hyperparameters, we tune $\tau, \sigma, \varepsilon, \beta$ as 0.6, 0.05, 0.5, 3 respectively. $\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5$ in Equation 13 and 14 are set as 2, 5, 2, 3, 2. λ_6 in Equation 14 is set as 2. More implementation details are stated in the supplementary material.

4.2. Comparison with the state-of-the-art methods.

ScanNetV2. As shown in Table 1, we compare our approach with existing state-of-the-art methods on the ScanNetV2 validation set. Our approach attains relative 2.4% improvements on mAP and shows relative 3.9% and 2.5% improvements on Box AP@50 and Box AP@25, respectively. On these metrics, our proposed model achieves state-of-the-art results. From Figure 4, we can see visualization of instance segmentation results. Compared to Mask3D, our approach correctly segments each instance and produces finer segmentation results.

S3DIS. We evaluate our method on S3DIS using Area

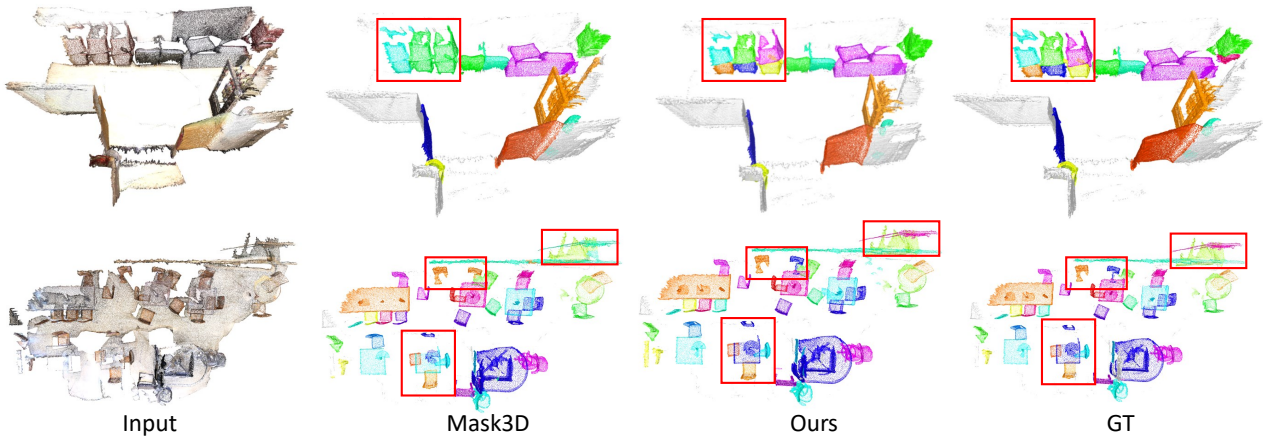

Figure 4. **Visualization of instance segmentation results on the ScanNetV2 validation set.** The red boxes highlight the key regions.

Table 2. **Comparison on S3DIS benchmark.**

Method	S3DIS Area 5				S3DIS 6-fold CV			
	mAP	AP@50	mPrec	mRec	mAP	AP@50	mPrec	mRec
SGPN [41]	/	/	36.0	28.7	/	/	38.2	31.2
ASIS [42]	/	/	55.3	42.4	/	/	63.6	47.5
3D-Bonet [44]	/	/	57.5	40.2	/	/	65.6	47.6
OccuSeg [13]	/	/	/	/	/	/	72.8	60.3
3D-MPA [11]	/	/	63.1	58.0	/	/	66.7	64.1
PointGroup [17]	/	57.8	61.9	62.1	/	64.0	69.6	69.2
DyCo3D [15]	/	/	64.3	64.2	/	/	/	/
MaskGroup [49]	/	65.0	62.9	64.7	/	69.9	66.6	69.6
SSTNet [23]	42.7	59.3	65.5	64.2	54.1	67.8	73.5	73.4
SoftGroup [39]	51.6	66.1	**73.6**	66.6	54.4	68.9	**75.3**	69.8
DKNet [43]	/	/	70.8	65.3	/	/	**75.3**	71.1
Mask3D [36]	56.5	69.3	68.7	70.7	60.7	72.0	70.5	72.5
Ours	**57.7**	**69.9**	70.5	**72.2**	**62.0**	**73.3**	72.7	**73.4**

Table 3. **Comparison on ScanNet200 benchmark.**

Method	mAP	AP@50	AP@25
Mask3D [36]	27.4	37.0	42.3
Ours	**28.1**	**37.1**	**43.4**

5 and 6-fold cross-validation respectively. For a fair comparison, Mask3D and our method are all supervised by instance labels at the superpoint level instead of the point level. As shown in Table 2, our superior results on some important metrics validate the effects and generalization of our method.

ScanNet200. We also compare our approach with existing state-of-the-art method on the ScanNet200 validation set, which includes a magnitude more class categories than ScanNetV2. As shown in Table 3, the results demonstrate the remarkable generalization of our model.

4.3. Effects of DBSCAN [12].

DBSCAN is a time-consuming density-based clustering algorithm that can separate multiple instances which are incorrectly predicted as a whole. Some methods with low coverage rates can easily divide multiple instances into a whole, so these methods rely heavily on DBSCAN for post-processing. As shown in Table 4, the high performance of Mask3D must be guaranteed by DBSCAN. However, profiting from our designed query initialization module and denoising module, our method can achieve an superior performance without support from DBSCAN. In order to show the effects of DBSCAN more explicitly, as shown in Figure 5, two chairs in Mask3D are not covered by queries, so the three chairs are jointly predicted. After the post-processing of DBSCAN, this problem has been alleviated. Although DBSCAN is effective, there are two problems. The first is that multiple hyperparameters need to be readjusted for different scenarios. The other is the time-consuming problem. For example, DBSCAN takes about 1 minute to process an indoor scenario. Therefore, our method can achieve superior performance at a low time cost.

Table 4. **Effects of DBSCAN.** In this experiment, the number of queries in both methods is set to 100.

Method	DBSCAN	ScanNetV2			S3DIS Area 5		
		mAP	AP@50	AP@25	mAP	AP@50	AP@25
Mask3D [36]	✗	53.7	73.0	82.5	53.9	67.0	73.9
	✓	54.7	74.0	82.9	56.5	69.3	75.6
	Improvements	**+1**	**+1**	**+0.4**	**+2.6**	**+2.3**	**+1.7**
Ours	✗	56.2	73.8	83.2	57.1	69.3	76.6
	✓	56.5	74.2	83.3	57.7	69.9	77.1
	Improvements	**+0.3**	**+0.4**	**+0.1**	**+0.6**	**+0.6**	**+0.5**

4.4. Ablation Study.

Components Analysis. In this section, we perform extensive ablation studies on the ScanNetV2 dataset to evaluate the effects of each design. Table 5 demonstrates the performance of the model with different designs. Concretely, the second row demonstrates that the query initialization module improves by 1.6 on mAP with a high coverage rate and low repetition rate query distribution. The cluster-based layer is designed to reduce the complexity of calculations and therefore only makes a modest contribution to accuracy. In the fifth row, the denoising module

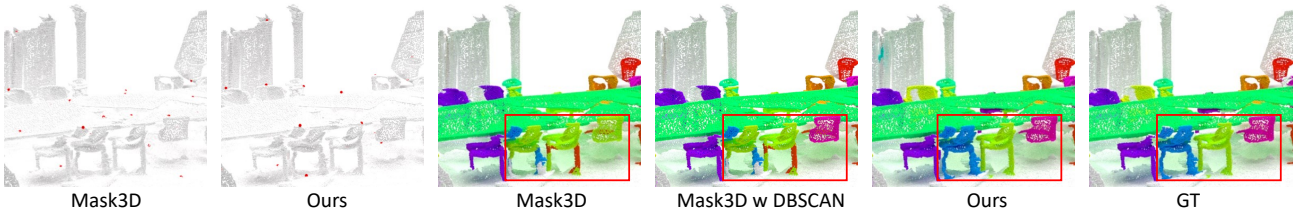

Figure 5. **Visualization of query distribution and instance segmentation results on the ScanNetV2 validation set.** The red points represent the position of queries, and the red boxes highlight the key regions.

Table 5. **Evaluation of the model with different designs on ScanNetV2.** QIM represents the Query Initialization Module, CL represents the Cluster-based Layer, DM is the Denoising Module.

QIM	CL	DM	DBSCAN	ScanNetV2		
				mAP	AP@50	AP@25
✗	✗	✗	✗	53.7	73.0	82.5
✓	✗	✗	✗	55.3	73.5	83.2
✗	✓	✗	✗	54.0	73.0	82.7
✓	✓	✗	✗	55.4	73.6	83.2
✓	✓	✓	✗	56.2	73.8	83.2
✓	✓	✓	✓	**56.5**	**74.2**	**83.3**

Table 6. **Effects of QIM.** Here, the experiment is conducted on ScanNetV2 without DBSCAN. CR means the coverage rate, while RR refers to the repetition rate. Higher CR and lower RR reflect a better result.

Method	Num	ScanNetV2				
		CR↑	**RR↓**	mAP	AP@50	AP@25
Mask3D [36]	50	0.54	0.48	47.6	65.5	75.2
Mask3D	100	0.74	0.64	53.7	73.0	82.5
Mask3D	150	0.83	0.72	54.3	73.0	82.8
Mask3D	500	**0.97**	0.99	52.8	70.6	79.9
Ours	50	0.76	**0.43**	54.0	72.9	82.5
Ours	100	0.89	0.47	**56.2**	**73.8**	**83.2**
Ours	150	0.92	0.70	55.3	72.5	83.0

improves the performance by 0.8 on mAP through noise background query suppression. Since the query initialization module improves the sampling quality of the query, our method does not rely on DBSCAN post-processing. Therefore, after time-consuming DBSCAN post-processing, our results are only marginally improved.

Effects of QIM. As shown in Table 6, a large number of queries can not only bring a high coverage rate but also lead to a high repetition rate. Fewer queries always bring a lower repetition rate, but a lower coverage rate of the scene may follow. Therefore, we should balance the relationship between repetition and coverage rates when selecting the number of queries. The experimental results of Table 6 on ScanNetV2 also verify that higher experimental performance can be obtained by balancing the coverage and repetition rates. Thanks to the high-quality queries obtained in QIM, our method performs best when the number of queries is 100, while Mask3d performs best when the number of queries is 150. Therefore, our method achieves better performance with less computational overhead.

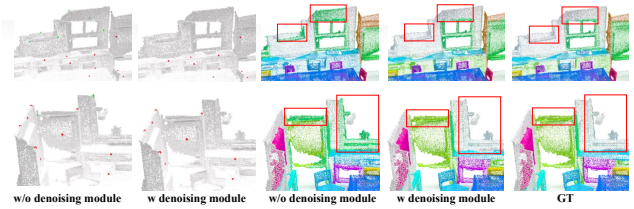

Figure 6. **The visualization of segmentation results related to the denoising module in the affiliated transformer decoder.** The red points represent the position of queries, the green points represent the position of noise background queries, and the red boxes highlight the key regions.

Table 7. **Efficiency comparison on a single cluster-based layer.**

Method	Memory(KB)	FLOPs(G)	Inferance Time(ms)
Mask3D [36]	48.8	2.3	1.1
Ours	**0.49**	**1.3**	**0.72**

Effects of ATD. The affiliated transformer decoder is conceived to suppress the interference of noise background queries and thus maintain a robust and desirable instance segmentation result. In order to vividly and detailedly validate the importance of suppressing this interference, relevant visualization experiments are carried out in Figure 6. The denoising module in the affiliated transformer decoder pulls away the noise background points (green) for the two scenes to suppress the interference with the foreground queries (red). Furthermore, the segmentation visualization in the third and fourth columns verifies the suppression beneficial for more accurate and robust segmentation results and validates the positive effects of our designed affiliated transformer decoder.

Efficiency comparison. As shown in Table 7, we compare the efficiency of the cluster-based layer provided by our method and the normal layer used in Mask3D. It can be seen from the results that the cluster-based layer is superior to the normal layer in terms of memory, FLOPs and inference time.

4.5. Parameter and Runtime Analysis.

Table 8 reports the model parameter and the runtime per scan of different methods on the ScanNetV2 validation set. For a fair comparison, the reported runtime is measured on the same RTX 3090 GPU. Compared with Mask3D, our method achieves noticeable performance improvement with only a 2.7M parameter increment. As to the inference

Table 8. **Parameter and runtime analysis of different methods on the ScanNetV2 validation set.** The runtime is measured on the same RTX 3090 GPU.

Method	Parameter(M)	Runtime(ms)
HAIS [4]	30.9	578
SoftGroup [39]	30.9	588
SSTNet [23]	/	729
Mask3D [36]	39.6	578
Ours	42.3	487

speed, our model is the fastest among all methods by reducing the decoder size. Concretely, Mask3D uses 12 layers of decoder, while our model only uses 8 layers. Although we add the Query Initialization Module, we adopt the superpoint features as the keys and values of cross attention, and down-sampling is done for each layer.

5. Conclusion

In this paper, we propose the Query Refinement Transformer for 3D instance segmentation. To solve the query sampling dilemma, we design a query initialization module to guarantee a high coverage rate of object instances while maintaining a low repetition rate. Furthermore, the well-designed affiliated transformer decoder suppresses the interference of noise background queries for better instance segmentation results. Extensive experiments conducted on two widely used 3D instance segmentation benchmarks demonstrate the superior performance of QueryFormer.

6. Acknowledgements

This work was partially supported by the National Nature Science Foundation of China (Grant 62022078), National Defense Basic Scientific Research Program of China (Grant JCKY2021601B013) and the National Nature Science Foundation of China (Grant 62021001).

References

[1] Iro Armeni, Ozan Sener, Amir R Zamir, Helen Jiang, Ioannis Brilakis, Martin Fischer, and Silvio Savarese. 3d semantic parsing of large-scale indoor spaces. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1534–1543, 2016. 2, 6

[2] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16*, pages 213–229. Springer, 2020. 1, 3

[3] Chun-Fu Richard Chen, Quanfu Fan, and Rameswar Panda. Crossvit: Cross-attention multi-scale vision transformer for image classification. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 357–366, 2021. 3

[4] Shaoyu Chen, Jiemin Fang, Qian Zhang, Wenyu Liu, and Xinggang Wang. Hierarchical aggregation for 3d instance segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15467–15476, 2021. 1, 3, 6, 8

[5] Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1290–1299, 2022. 1, 3

[6] Bowen Cheng, Alex Schwing, and Alexander Kirillov. Per-pixel classification is not all you need for semantic segmentation. *Advances in Neural Information Processing Systems*, 34:17864–17875, 2021. 3

[7] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4d spatio-temporal convnets: Minkowski convolutional neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3075–3084, 2019. 2, 6

[8] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5828–5839, 2017. 2, 6

[9] Jian Ding, Nan Xue, Yang Long, Gui-Song Xia, and Qikai Lu. Learning roi transformer for oriented object detection in aerial images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2849–2858, 2019. 3

[10] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 3

[11] Francis Engelmann, Martin Bokeloh, Alireza Fathi, Bastian Leibe, and Matthias Nießner. 3d-mpa: Multi-proposal aggregation for 3d semantic instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9031–9040, 2020. 1, 3, 6, 7

[12] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *kdd*, volume 96, pages 226–231, 1996. 7

[13] Lei Han, Tian Zheng, Lan Xu, and Lu Fang. Occuseg: Occupancy-aware 3d instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2940–2949, 2020. 6, 7

[14] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 3

[15] Tong He, Chunhua Shen, and Anton Van Den Hengel. Dyco3d: Robust instance segmentation of 3d point clouds through dynamic convolution. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 354–363, 2021. 3, 6, 7

[16] Ji Hou, Angela Dai, and Matthias Nießner. 3d-sis: 3d semantic instance segmentation of rgb-d scans. In *Proceedings*

of the IEEE/CVF conference on computer vision and pattern recognition, pages 4421–4430, 2019. 6

[17] Li Jiang, Hengshuang Zhao, Shaoshuai Shi, Shu Liu, Chi-Wing Fu, and Jiaya Jia. Pointgroup: Dual-set point grouping for 3d instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and Pattern recognition*, pages 4867–4876, 2020. 1, 3, 6, 7

[18] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955. 2, 3, 4, 5

[19] Jean Lahoud, Bernard Ghanem, Marc Pollefeys, and Martin R Oswald. 3d instance segmentation via multi-task metric learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9256–9266, 2019. 3

[20] Loic Landrieu and Martin Simonovsky. Large-scale point cloud semantic segmentation with superpoint graphs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4558–4567, 2018. 2

[21] Ville V Lehtola, Harri Kaartinen, Andreas Nüchter, Risto Kaijaluoto, Antero Kukko, Paula Litkey, Eija Honkavaara, Tomi Rosnell, Matti T Vaaja, Juho-Pekka Virtanen, et al. Comparison of the selected state-of-the-art 3d indoor scanning and point cloud generation methods. *Remote sensing*, 9(8):796, 2017. 1

[22] Feng Li, Hao Zhang, Shilong Liu, Jian Guo, Lionel M Ni, and Lei Zhang. Dn-detr: Accelerate detr training by introducing query denoising. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13619–13627, 2022. 6

[23] Zhihao Liang, Zhihao Li, Songcen Xu, Mingkui Tan, and Kui Jia. Instance segmentation in 3d scenes using semantic superpoint tree networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2783–2792, 2021. 1, 3, 6, 7, 8

[24] Shih-Hung Liu, Shang-Yi Yu, Shao-Chi Wu, Hwann-Tzong Chen, and Tyng-Luh Liu. Learning gaussian instance segmentation in point clouds. *arXiv preprint arXiv:2007.09860*, 2020. 1, 3

[25] Ze Liu, Zheng Zhang, Yue Cao, Han Hu, and Xin Tong. Group-free 3d object detection via transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2949–2958, 2021. 3

[26] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 6

[27] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *2016 fourth international conference on 3D vision (3DV)*, pages 565–571. Ieee, 2016. 6

[28] Ishan Misra, Rohit Girdhar, and Armand Joulin. An end-to-end transformer model for 3d object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2906–2917, 2021. 3

[29] Alexander Neubeck and Luc Van Gool. Efficient non-maximum suppression. In *18th international conference on pattern recognition (ICPR'06)*, volume 3, pages 850–855. IEEE, 2006. 5

[30] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. 2

[31] Kyeong-Beom Park, Minseok Kim, Sung Ho Choi, and Jae Yeol Lee. Deep learning-based smart task assistance in wearable augmented reality. *Robotics and Computer-Integrated Manufacturing*, 63:101887, 2020. 1

[32] Charles R Qi, Or Litany, Kaiming He, and Leonidas J Guibas. Deep hough voting for 3d object detection in point clouds. In *proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9277–9286, 2019. 6

[33] Charles R Qi, Wei Liu, Chenxia Wu, Hao Su, and Leonidas J Guibas. Frustum pointnets for 3d object detection from rgb-d data. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 918–927, 2018. 6

[34] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017. 3

[35] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017. 2, 3

[36] Jonas Schult, Francis Engelmann, Alexander Hermans, Or Litany, Siyu Tang, and Bastian Leibe. Mask3d for 3d semantic instance segmentation. *arXiv preprint arXiv:2210.03105*, 2022. 1, 2, 3, 4, 5, 6, 7, 8

[37] Leslie N Smith and Nicholay Topin. Super-convergence: Very fast training of neural networks using large learning rates. In *Artificial intelligence and machine learning for multi-domain operations applications*, volume 11006, pages 369–386. SPIE, 2019. 6

[38] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 3

[39] Thang Vu, Kookhoi Kim, Tung M Luu, Thanh Nguyen, and Chang D Yoo. Softgroup for 3d instance segmentation on point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2708–2717, 2022. 1, 3, 6, 7, 8

[40] Haiyang Wang, Lihe Ding, Shaocong Dong, Shaoshuai Shi, Aoxue Li, Jianan Li, Zhenguo Li, and Liwei Wang. Ca-group3d: Class-aware grouping for 3d object detection on point clouds. *arXiv preprint arXiv:2210.04264*, 2022. 5

[41] Weiyue Wang, Ronald Yu, Qiangui Huang, and Ulrich 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. 7

[42] Xinlong Wang, Shu Liu, Xiaoyong Shen, Chunhua Shen, and Jiaya Jia. Associatively segmenting instances and semantics in point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4096–4105, 2019. 7

[43] Yizheng Wu, Min Shi, Shuaiyuan Du, Hao Lu, Zhiguo Cao, and Weicai Zhong. 3d instances as 1d kernels. In *Computer*

Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXIX, pages 235–252. Springer, 2022. 6, 7

[44] Bo Yang, Jianan Wang, Ronald Clark, Qingyong Hu, Sen Wang, Andrew Markham, and Niki Trigoni. Learning object bounding boxes for 3d instance segmentation on point clouds. *Advances in neural information processing systems*, 32, 2019. 3, 7

[45] Li Yi, Wang Zhao, He Wang, Minhyuk Sung, and Leonidas J Guibas. Gspn: Generative shape proposal network for 3d instance segmentation in point cloud. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3947–3956, 2019. 1, 3, 6

[46] Ekim Yurtsever, Jacob Lambert, Alexander Carballo, and Kazuya Takeda. A survey of autonomous driving: Common practices and emerging technologies. *IEEE access*, 8:58443–58469, 2020. 1

[47] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip HS Torr, and Vladlen Koltun. Point transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 16259–16268, 2021. 3

[48] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip HS Torr, et al. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6881–6890, 2021. 3

[49] Min Zhong, Xinghao Chen, Xiaokang Chen, Gang Zeng, and Yunhe Wang. Maskgroup: Hierarchical point grouping and masking for 3d instance segmentation. In *2022 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6. IEEE, 2022. 6, 7