

Mask-Attention-Free Transformer for 3D Instance Segmentation

Xin Lai¹ Yuhui Yuan³ Ruihang Chu¹

¹The Chinese University of Hong Kong

Yukang Chen¹ Han Hu³ Jiaya Jia^{1,2*}

²SmartMore ³Microsoft Research Asia

Abstract

Recently, transformer-based methods have dominated 3D instance segmentation, where mask attention is commonly involved. Specifically, object queries are guided by the initial instance masks in the first cross-attention, and then iteratively refine themselves in a similar manner. However, we observe that the mask-attention pipeline usually leads to slow convergence due to low-recall initial instance masks. Therefore, we abandon the mask attention design and resort to an auxiliary center regression task instead. Through center regression, we effectively overcome the low-recall issue and perform cross-attention by imposing positional prior. To reach this goal, we develop a series of position-aware designs. First, we learn a spatial distribution of 3D locations as the initial position queries. They spread over the 3D space densely, and thus can easily capture the objects in a scene with a high recall. Moreover, we present relative position encoding for the cross-attention and iterative refinement for more accurate position queries. Experiments show that our approach converges 4× faster than existing work, sets a new state of the art on ScanNetv2 3D instance segmentation benchmark, and also demonstrates superior performance across various datasets. Code and models are available at [https://github.com/dvlab-research/](https://github.com/dvlab-research/Mask-Attention-Free-Transformer) Mask-Attention-Free-Transformer.

1. Introduction

Nowadays 3D point clouds can be conveniently collected. They have benefited various applications, such as autonomous driving, robotics, and augmented reality. As a fundamental task, 3D instance segmentation also poses great challenges simultaneously, such as geometric occlusion and semantic ambiguity.

Many works have been proposed to solve the 3D instance segmentation task. Grouping-based methods [25, 55, 5, 75] rely on heuristic clustering algorithms such as DBSCAN or Breadth-First Search (BFS) to generate instance proposals. They thus require sophisticated hyper-parameters

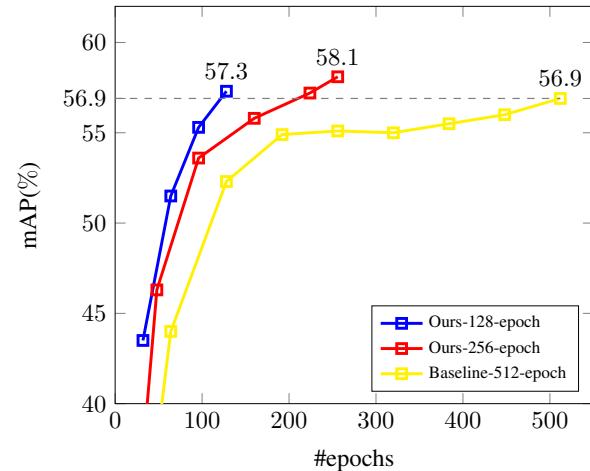


Figure 1. Validation curve of the baseline and ours on ScanNetv2 val set. With only 128-epoch training, ours outperforms the baseline trained with 512 epochs.

tuning and are prone to wrongly segment instances that are close to each other. Recently, transformer-based methods [49, 50] develop a fully end-to-end pipeline. With transformer decoder layers, a fixed number of object queries attend to global features iteratively and directly output instance predictions. It requires no post-processing for duplicate removal such as NMS, since it adopts one-to-one bipartite matching during training. Moreover, it employs mask attention, which uses the instance masks predicted in the last layer to guide the cross-attention.

However, we point out that current transformer-based methods suffer from the issue of slow convergence. As shown in Fig. 1, the baseline model manifests slow convergence and lags behind our method by a large margin, particularly in the early stages of training. We dive further and find that the issue is potentially caused by the low recall of the *initial instance masks*. Specifically, as shown in Fig. 2 (a), the initial instance masks are produced by the similarity map between the initial object queries and the point-wise mask features. Since the initial object queries are unstable in early training, we notice that the recall of initial instance masks is substantially lower than ours in Fig. 3, especially at the beginning of training (*i.e.*, the 32-th epoch). The low-quality initial instance masks increase the training difficulty,

*Corresponding Author

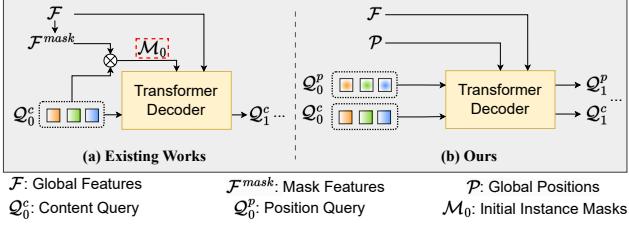


Figure 2. The framework of (a) existing works (based on mask attention) and (b) ours. Existing works have the issue of low-recall initial instance masks (*i.e.*, \mathcal{M}_0). Our approach resorts to an auxiliary center regression task to circumvent this issue.

thereby slowing down convergence.

Given the low recall of the initial instance masks, we abandon the mask attention design and instead construct an auxiliary center regression task to guide cross-attention, as depicted in Fig. 2 (b). To enable center regression, we develop a series of position-aware designs. Firstly, we maintain a set of learnable position queries, each of which denotes the position of its corresponding content query. They are densely distributed over the 3D space, and we require each query to attend to its local region. As a result, the queries can easily capture the objects in a scene with a higher recall, which is crucial in reducing training difficulty and accelerating convergence.

In addition, we design the contextual relative position encoding for cross-attention. Compared to the mask attention used in previous works, our solution is more flexible since the attention weights are adjusted by relative positions instead of hard masking. Furthermore, we iteratively update the position queries to achieve more accurate representation. Finally, we introduce the center distances between predictions and ground truths in both matching and loss.

In total, our contribution is three-fold.

- We observe that existing transformer-based methods suffer from the low recall of initial instance masks, which causes training difficulty and slow convergence.
 - Instead of relying on mask attention, we construct an auxiliary center regression task to overcome the low-recall issue and design a series of position-aware components accordingly. Our approach manifests faster convergence and demonstrates higher performance.
 - Experiments show our approach achieves a new state-of-the-art result and demonstrates superior performance on various datasets including ScanNetv2, ScanNet200, and S3DIS.

2. Related Work

2.1. 3D Instance Segmentation

3D instance segmentation is a fundamental task for 3D recognition [45, 46, 28, 6, 32, 36, 31, 35, 27, 24, 34,

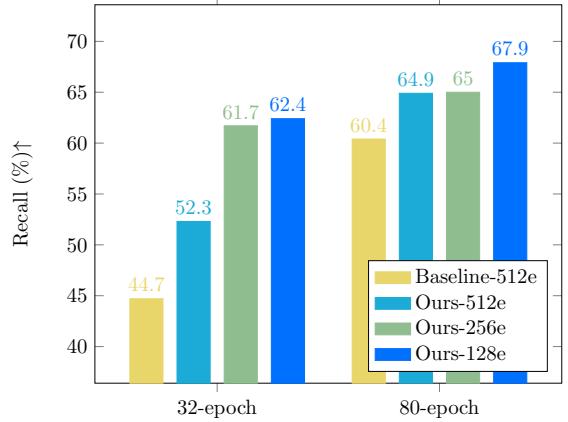


Figure 3. The recall of initial instance masks at the 32-th and 80-th training epoch. We compare the baseline trained with 512 epochs and ours trained with 128, 256, and 512 epochs. Since our approach does not produce initial instance masks before entering decoder layers, we make statistics on the instance masks output by the first decoder layer for both the baseline and ours.

[11]. The solutions can be categorized into detection-based, grouping-based and the new emerging transformer-based paradigms. Detection-based approaches [21, 2, 67] first detect the bounding boxes and then segment the fine-grained instance mask. On the other hand, grouping-based approaches [60, 61, 26, 17, 19, 25, 23, 20, 69, 33, 5, 55, 12, 10] employ clustering algorithms to group the points into a set of instance clusters. Before clustering, they either move 3D points towards the associated object center to form a more compact distribution [26, 19, 25, 23, 69, 33, 5, 55], or transform points to a high-dimension feature space [60, 61]. Further, a series of works leverage semantic priors to avoid the noisy points from other categories [26, 25, 33, 5, 55] or use advanced grouping strategies [55, 5, 33]. With these designs, the grouping-based paradigm has achieved leading performance across various evaluation benchmarks[1, 14] for a long time. Recently, transformer-based paradigm [49, 50] becomes another option and swiftly sets a new state-of-the-art. Compared with previous methods, it presents an elegant pipeline, and can directly output instance predictions. It relies on the transformer decoder and mask attention to aggregate information from global features. In this work, we construct an auxiliary center regression task to assist in cross-attention. Although the existing grouping-based methods also predict center offsets, we explain that ours is not for instance proposals, but to overcome the low-recall issue and provide positional priors for cross-attention.

2.2. Vision Transformer

Transformer has become a fundamental model in the vision area, thanks to its flexibility and power to model various scenarios using attention mechanisms [54]. Re-

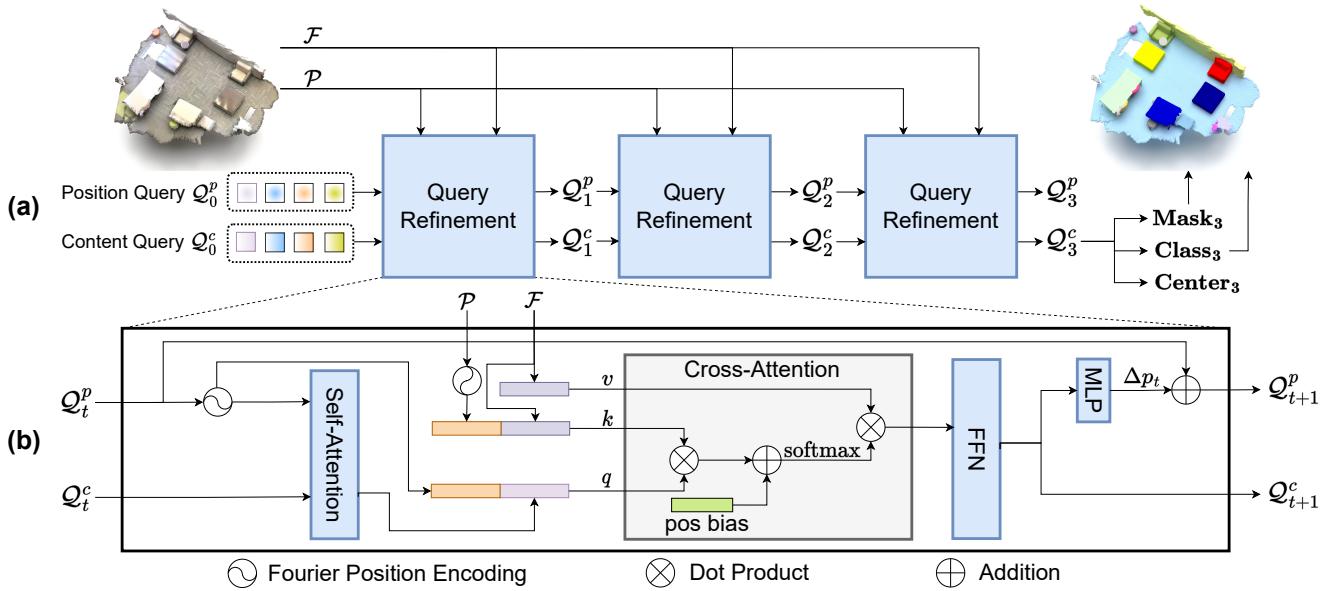


Figure 4. The overview of our framework. Besides content queries Q_0^c , we also maintain a set of learnable position queries Q_0^p . The content queries Q^c aggregate features from the global features \mathcal{F} . The position queries Q^p are designed to guide the cross-attention. The attention weights are adjusted based on the relative positions between the position queries Q^p and the global positions \mathcal{P} . Both content and position queries are iteratively refined in each layer. Only 3 decoder layers are shown (we use 6 layers in our experiments).

cently, many works [16, 52, 53, 59, 58, 13, 15, 51, 66] rely on the self-attention in transformers to develop vision fundamental models. Besides, DETR [3] proposes a fully end-to-end pipeline for object detection. It utilizes transformer decoders to dynamically aggregate features from images, and uses one-to-one bipartite matching for ground-truth assignment, yielding an elegant pipeline. To solve the notorious slow convergence of DETR, approaches [76, 62, 41, 37, 70, 68] propose deformable attention, impose strong prior or decrease searching space in cross-attention to accelerate convergence. Further, methods of [29, 39, 22, 71, 30] present several ways to stabilize matching and training. Moreover, masked attention [8, 7] are proposed to impose semantic priors to accelerate training for segmentation tasks. Recently, there are works [28, 73, 63, 64, 43, 49, 50] that develop transformer models tailored for 3D point clouds. Following this line of research, we observe the low recall of initial instance masks, and present solutions to circumvent the use of mask attention.

3. Method

We first review previous methods and present the overview of our method in Sec. 3.1. Then, we elaborate on the details of our position-aware designs in Sec. 3.2.

3.1. Overview

Preliminary. Recently, Mask3D [49] and SPFormer [50] present a fully end-to-end pipeline, which allows the object

queries to directly output instance predictions. With transformer decoders, a fixed number of object queries aggregate information from the global features (either multi-scale voxel features [49] or superpoint features [50]) extracted with the backbone. Moreover, similar to Mask2Former [8, 7], they adopt mask attention and rely on the instance masks to guide the cross-attention. Specifically, the cross-attention is masked with the instance masks predicted in the last decoder layer, so that the queries only need to consider the masked features. However, as shown in Fig. 3, the recall of initial instance masks is low in the early stages of training. It hinders the ability to achieve a high-quality result in the subsequent layers and thus increases training difficulty.

Ours. Instead of relying on mask attention, we propose an auxiliary center regression task to guide instance segmentation. The overview of our method is shown in Fig. 4 (a). We first yield the global positions $\mathcal{P} \in \mathbb{R}^{N \times 3}$ from the input point cloud and extract the global features $\mathcal{F} \in \mathbb{R}^{N \times d}$ using the backbone (\mathcal{P} and \mathcal{F} can be either voxels [49] or superpoints [50] positions and features). In contrast to existing works, besides the content queries $Q_0^c \in \mathbb{R}^{n \times d}$, we also maintain a fixed number of position queries $Q_0^p \in [0, 1]^{n \times 3}$ that represent the normalized instance centers. Q_0^p is randomly initialized and Q_0^c is initialized with zero. Given the global positions \mathcal{P} and global features \mathcal{F} , our goal is to let the positional queries guide their corresponding content queries in cross-attention, and then iteratively refine both sets of queries, and finally predict the instance centers, classes and masks. For the t -th decoder layer, this process

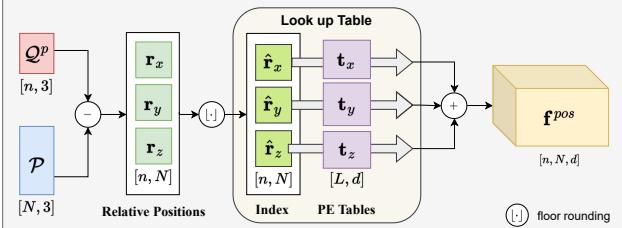


Figure 5. The illustration for relative position encoding. We note the shape of the corresponding tensor at the bottom.

is formulated as

$$\text{Center}_t = \text{MLP}_{center}(\mathbf{Q}_t^c) + \mathbf{Q}_{t-1}^p, \quad (1)$$

$$\text{Class}_t = \text{MLP}_{cls}(\mathbf{Q}_t^c), \quad (2)$$

$$\text{Mask}_t = \sigma(\mathbf{Q}_t^c \cdot \mathcal{F}_{mask}^T) < 0.5, \quad \mathcal{F}_{mask} = \text{MLP}_{mask}(\mathcal{F}), \quad (3)$$

where $\text{Center}_t \in \mathbb{R}^{n \times 3}$, $\text{Class}_t \in \mathbb{R}^{n \times K}$ and $\text{Mask}_t \in \{0, 1\}^{n \times N}$ are the predicted centers, classification logits, and the instance masks.

3.2. Position-aware Designs

To effectively support the center regression task and improve the recall of initial instance masks, we propose a series of position-aware designs as follows.

Learnable Position Query. Unlike previous works [49, 50], we introduce an additional set of position queries $\mathcal{Q}_0^p \in [0, 1]^{n \times 3}$. Since the range of points varies significantly among different scenes, the initial position queries are stored in a normalized form as learnable parameters followed by sigmoid function. Basically, we can obtain the absolute positions $\hat{\mathcal{Q}}_t^p \in \mathbb{R}^{n \times 3}$ from the normalized position queries \mathcal{Q}_t^p for a given input scene as

$$\hat{\mathcal{Q}}_t^p = \mathcal{Q}_t^p \cdot (p_{max} - p_{min}) + p_{min}, \quad (4)$$

where the $p_{min}, p_{max} \in \mathbb{R}^3$ represent the minimum and maximum coordinates of the input scene, respectively. The resultant $\hat{\mathcal{Q}}_t^p$ explicitly represents the positions of the corresponding content queries \mathcal{Q}_t^c .

It is notable that the initial position queries are densely spread throughout the 3D space. Also, every query aggregates features from its local region. This design choice makes it easier for the initial queries to capture the objects in a scene with a high recall, as shown in Fig. 3. It overcomes the low-recall issue caused by initial instance masks, and consequently reduces the training complexity of the subsequent layers.

Relative Position Encoding. Other than the absolute position encoding (*e.g.*, Fourier or sine transformations), we also adopt contextual relative position encoding in cross-attention. Inspired by [28], we first calculate the relative positions $\mathbf{r} \in \mathbb{R}^{n \times N \times 3}$ between the position queries $\hat{\mathcal{Q}}_t^p \in \mathbb{R}^{n \times 3}$ and the global positions $\mathcal{P} \in \mathbb{R}^{N \times 3}$, and quantize it into discrete integers $\hat{\mathbf{r}} \in \mathbb{Z}^{n \times N \times 3}$ as shown in Fig. 5. It is formulated as

$$\mathbf{r} = \hat{\mathcal{Q}}_t^p - \mathcal{P}, \quad \hat{\mathbf{r}} = \lfloor \frac{\mathbf{r}}{s} \rfloor + \frac{L}{2}, \quad (5)$$

where s denotes the quantization size, L denotes the length of position encoding table. We plus $\frac{L}{2}$ to ensure the discrete relative positions are non-negative.

Then, we use the discrete relative positions $\hat{\mathbf{r}}$ as indices to look up the corresponding position encoding tables $\mathbf{t} \in \mathbb{R}^{3 \times L \times d}$, as illustrated in Fig. 5. Formally, the relative position encoding $\mathbf{f}^{pos} \in \mathbb{R}^{n \times N \times d}$ is yielded as

$$\mathbf{f}^{pos} = \mathbf{t}[0, \hat{\mathbf{r}}_x] + \mathbf{t}[1, \hat{\mathbf{r}}_y] + \mathbf{t}[2, \hat{\mathbf{r}}_z], \quad (6)$$

where $\hat{\mathbf{r}}_x, \hat{\mathbf{r}}_y, \hat{\mathbf{r}}_z \in \mathbb{Z}^{n \times N}$ are the discrete relative positions along the x , y , and z -axis, respectively.

Further, the relative position encoding \mathbf{f}^{pos} performs dot product with the query features $\mathbf{f}^q \in \mathbb{R}^{n \times d}$ or key features $\mathbf{f}^k \in \mathbb{R}^{N \times d}$ in the cross-attention, which is formulated as

$$\text{pos_bias}_{i,j} = \mathbf{f}_{i,j}^{pos} \cdot \mathbf{f}_i^q + \mathbf{f}_{i,j}^{pos} \cdot \mathbf{f}_j^k, \quad (7)$$

where $\text{pos_bias} \in \mathbb{R}^{n \times N}$ is the positional bias. It is then added to the cross-attention weights, followed by the softmax function, as shown in Fig. 4 (b).

It is worth noting that the RPE offers a greater degree of flexibility and error-insensitivity, compared to mask attention. In essence, RPE can be likened to a soft mask that has the ability to adjust attention weights flexibly, instead of hard masking. Another advantage of RPE is that it integrates semantic information (*e.g.*, object size and class) and thus can harvest local information selectively. This is accomplished by the interaction between the relative positions and the semantic features (*i.e.*, \mathbf{f}^q and \mathbf{f}^k).

Iterative Refinement. Since the content queries in our decoder layers are updated regularly, it is not optimal to maintain frozen position queries throughout the decoding process. Additionally, the initial position queries are static, so it is beneficial to adapt them to the specific input scene in the subsequent layers. To that end, we iteratively refine the position queries based on the content queries. Specifically, as shown in Fig. 4 (b), we leverage an MLP to predict a center offset Δp_t from the updated content query \mathcal{Q}_{t+1}^c . We then add it to the previous position query $\hat{\mathcal{Q}}_t^p$ as

$$\begin{aligned} \Delta p_t &= \text{MLP}_{center}(\mathcal{Q}_{t+1}^c), \\ \hat{\mathcal{Q}}_{t+1}^p &= \hat{\mathcal{Q}}_t^p + \Delta p_t. \end{aligned} \quad (8)$$

Center Matching & Loss. To eliminate the need for duplicate removal methods such as non-maximum suppression (NMS), bipartite matching is adopted during training. Existing works [49, 50] rely on semantic predictions and binary masks to match the ground truths.

In contrast, to support center regression, we also incorporate center distance in bipartite matching. Since we require the queries to only attend to a local region, it is critical to ensure that they only match with nearby ground-truth objects. To achieve this, we adapt the matching costs formulation as follows

$$\begin{aligned} \mathcal{C}_{cls}(k, \hat{k}) &= \text{CE}(\text{Class}_k, \hat{k}), \\ \mathcal{C}_{dice}(k, \hat{k}) &= \text{DICE}(\text{Mask}_k, \text{Mask}_{\hat{k}}), \\ \mathcal{C}_{bce}(k, \hat{k}) &= \text{BCE}(\text{Mask}_k, \text{Mask}_{\hat{k}}), \\ \mathcal{C}_{center}(k, \hat{k}) &= \text{L}_1(\text{Center}_k, \text{Center}_{\hat{k}}), \\ \mathcal{C}(k, \hat{k}) &= \lambda_{cls}\mathcal{C}_{cls}(k, \hat{k}) + \lambda_{dice}\mathcal{C}_{dice}(k, \hat{k}) \\ &\quad + \lambda_{bce}\mathcal{C}_{bce}(k, \hat{k}) + \lambda_{center}\mathcal{C}_{center}(k, \hat{k}), \end{aligned} \quad (9)$$

where k and \hat{k} denotes a predicted and ground-truth instance, respectively, $\mathcal{C} \in \mathbb{R}^{n \times n_{inst}}$ denotes the matching cost matrix, and λ denotes the cost weights.

The Hungarian Algorithm is then applied on \mathcal{C} to yield the one-to-one matching result $\hat{\sigma} \in \mathbb{Z}^n$, which is followed by the loss function as

$$\begin{aligned} \hat{\sigma} &= \arg \min_{\sigma: \sigma_i \neq \sigma_j, \forall i \neq j} \sum_{i=1}^n \mathcal{C}(i, \sigma_i), \\ \mathcal{L} &= \lambda_{cls} \text{CE}(\text{Class}_i, \hat{\sigma}_i) \\ &\quad + \lambda_{dice} \text{DICE}(\text{Mask}_i, \text{Mask}_{\hat{\sigma}_i}) \\ &\quad + \lambda_{bce} \text{BCE}(\text{Mask}_i, \text{Mask}_{\hat{\sigma}_i}) \\ &\quad + \lambda_{center} \text{L}_1(\text{Center}_i, \text{Center}_{\hat{\sigma}_i}). \end{aligned} \quad (10)$$

4. Experiment

This section first provides an overview of the experimental setup in Sec. 4.1. We then present the 3D instance segmentation results in Sec. 4.2. Additionally, we conduct an extensive ablation study in Sec. 4.3. Furthermore, we showcase the object detection results and visual comparisons in Sections 4.4 and 4.5, respectively. Code and models will be made publicly available.

4.1. Experimental Setting

Network Architecture. For both ScanNetv2 [14] and ScanNet200 [47], we follow previous works [50, 25, 55, 5, 33] to use 5-layer U-Net as the backbone. The initial channel is set to 32. Unless otherwise specified, we use the coordinates and colors as the input features. We use 6 layers of Transformer decoders, where the head number is set to 8 and the hidden and feed-forward dimensions are set

to 256 and 1024, respectively. We adopt Fourier absolute position encoding with the temperature set to 10,000. The quantization size for RPE is set to 0.1m, and the length of the RPE table is 48. Unless otherwise specified, we choose [50] as the baseline model, since it has achieved the best performance on ScanNetv2 *val* set so far. For the S3DIS [1] dataset, following Mask3D [49], we use Res16UNet34C [9] as the backbone and employ 4 decoders to attend to the coarsest four scales, and this is repeated 3 times with the shared parameters. The decoder hidden and feed-forward dimensions are set to 128 and 1024, respectively.

Datasets. We use the ScanNetv2 [14], ScanNet200 [47] and S3DIS [1] datasets for evaluation. All of them are challenging large-scale indoor scene datasets.

The ScanNetv2 dataset comprises 1201 scenes for training, and an additional 312 and 100 indoor scenes for validation and testing, respectively. The scenes are captured with RGB-D cameras and annotated with 20 semantic labels, 18 of which are instance classes. The ScanNet200 dataset adopts the same point cloud data, but it offers more diverse annotations, covering 200 classes, 198 of which are instance classes.

The S3DIS dataset contains 271 rooms in 6 areas of three buildings, and 13 semantic categories are annotated. Following previous works, the scenes in Area 5 are used for validation and the others are for training.

Implementation Details. We adopt one RTX 3090 GPU for training on ScanNet and ScanNet200, and one A100 GPU on S3DIS. Following previous works, we use AdamW [40] optimizer with the learning rate and weight decay set to 0.0001 and 0.05, respectively. We adopt poly scheduler on ScanNet and ScanNet200, and onecycle scheduler on S3DIS. The batch size is set to 4. For the weights of matching costs and losses, $(\lambda_{cls}, \lambda_{bce}, \lambda_{dice}, \lambda_{center})$ are set to $(0.5, 1.0, 1.0, 0.5)$ on ScanNet and ScanNet200, and $(2.0, 5.0, 1.0, 0.5)$ on S3DIS. The voxel size is set to 0.02m. We limit the points number up to 250,000. Otherwise, we crop the scene by cubic windows iteratively until the point number is lower than the limit. During inference, we select the top 100 instances with the highest scores and set the minimum points number per instance to 100.

4.2. Instance Segmentation Results

ScanNetv2. We present the results of instance segmentation on both the ScanNetv2 *test* and *val* sets in Tables 1 and 2, respectively. Our method achieves a considerable increase in mAP compared to previous works, suggesting a superior ability to capture fine-grained details and produce high-quality instance segmentation. While Mask3D [49] slightly outperforms our model in terms of mAP₅₀, it is worth noting that this is potentially due to their use of a

Method	mAP	mAP ₅₀	bath	bed	bksf	cabinet	chair	counter	curtain	desk	door	other	picture	fridge	s. cur.	sink	sofa	table	toilet	wind.
3D-BoNet [2]	25.3	48.8	51.9	32.4	25.1	13.7	34.5	3.1	41.9	6.9	16.2	13.1	5.2	20.2	33.8	14.7	30.1	30.3	65.1	17.8
MTML [26]	28.2	40.2	57.7	38.0	18.2	10.7	43.0	0.1	42.2	5.7	17.9	16.2	7.0	22.9	51.1	16.1	49.1	31.3	65.0	16.2
GICN [38]	34.1	63.8	58.0	37.1	34.4	19.8	46.9	5.2	56.4	9.3	21.2	21.2	12.7	34.7	53.7	20.6	52.5	32.9	72.9	24.1
3D-MPA [17]	35.5	61.1	45.7	48.4	29.9	27.7	59.1	4.7	33.2	21.2	21.7	27.8	19.3	41.3	41.0	19.5	57.4	35.2	84.9	21.3
Dyco3D [20]	39.5	64.1	64.2	51.8	44.7	25.9	66.6	5.0	25.1	16.6	23.1	36.2	23.2	33.1	53.5	22.9	58.7	43.8	85.0	31.7
PE [69]	39.6	64.5	66.7	46.7	44.6	24.3	62.4	2.2	57.7	10.6	21.9	34.0	23.9	48.7	47.5	22.5	54.1	35.0	81.8	27.3
PointGroup [25]	40.7	63.6	63.9	49.6	41.5	24.3	64.5	2.1	57.0	11.4	21.1	35.9	21.7	42.8	66.6	25.6	56.2	34.1	86.0	29.1
HAIS [5]	45.7	69.9	70.4	56.1	45.7	36.4	67.3	4.6	54.7	19.4	30.8	42.6	28.8	45.4	71.1	26.2	56.3	43.4	88.9	34.4
OccuSeg [19]	48.6	67.2	80.2	53.6	42.8	36.9	70.2	20.5	33.1	30.1	37.9	47.4	32.7	43.7	86.2	48.5	60.1	39.4	84.6	27.3
SoftGroup [55]	50.4	76.1	66.7	57.9	37.2	38.1	69.4	7.2	67.7	30.3	38.7	53.1	31.9	58.2	75.4	31.8	64.3	49.2	90.7	38.8
SSTNet [33]	50.6	69.8	73.8	54.9	49.7	31.6	69.3	17.8	37.7	19.8	33.0	46.3	57.6	51.5	85.7	49.4	63.7	45.7	94.3	29.0
SPFormer [50]	54.9	77.0	74.5	64.0	48.4	39.5	73.9	31.1	56.6	33.5	46.8	49.2	55.5	47.8	74.7	43.6	71.2	54.0	89.3	34.3
Mask3D* [49]	56.6	78.0	92.6	59.7	40.8	42.0	73.7	23.9	59.8	38.6	45.8	54.9	56.8	71.6	60.1	48.0	64.6	57.5	92.2	36.4
Ours	57.8	77.4	77.8	64.9	52.0	44.9	76.1	25.3	58.4	39.1	53.0	47.2	61.7	49.9	79.5	47.3	74.5	54.8	96.0	37.4
Ours [‡]	59.6	78.6	88.9	72.1	44.8	46.0	76.8	25.1	55.8	40.8	50.4	53.9	61.6	61.8	85.8	48.2	68.4	55.1	93.1	45.0

Table 1. 3D instance segmentation results on ScanNet *test* set. * denotes using Res16UNet34C (twice as many parameters as ours) as the backbone. [‡] denotes using surface normal. Methods published before the submission deadline (03/08/2023) are listed.

Method	mAP	mAP ₅₀
GSPN [67]	19.3	37.8
MTML [26]	20.3	40.2
3D-MPA [17]	35.5	59.1
Dyco3D [20]	35.4	57.6
PointGroup [25]	34.8	56.7
MaskGroup [75]	42.0	63.3
HAIS [5]	43.5	64.1
OccuSeg [19]	44.2	60.7
SoftGroup [55]	46.0	67.6
SSTNet [33]	49.4	64.3
SPFormer [50]	56.3	73.9
Mask3D* [49]	55.2	73.7
Ours	58.4	75.9
Ours [‡]	59.9	76.5

Table 2. 3D instance segmentation results on ScanNet *val* set. * denotes using Res16UNet34C (twice as many parameters as ours) as the backbone. [‡] denotes using surface normal.

Method	mAP	mAP ₅₀	mAP ₂₅
SPFormer [†] [50]	25.2	33.8	39.6
Mask3D* [49]	27.4	37.0	42.3
Ours	29.2	38.2	43.3

Table 3. 3D instance segmentation results on ScanNet200 *val* set. [†] denotes reproduced results. * denotes using Res16UNet34C.

stronger backbone (*i.e.*, Res16UNet34C with twice as many parameters as ours) and DBSCAN post-processing. Despite this, our approach produces significantly better performance on the ScanNetv2 *val* set than Mask3D, as seen in Table 2.

Method	mAP ₅₀	mAP ₂₅
PointGroup [25]	57.8	-
MaskGroup [75]	65.0	-
SoftGroup [55]	66.1	-
SSTNet [33]	59.3	-
SPFormer [50]	66.8	-
Mask3D [49]	68.4	75.2
Ours	69.1	75.7

Table 4. 3D instance segmentation results on S3DIS Area5.

ScanNet200. Table 3 presents our comparison with previous state-of-the-art methods on the *val* set of ScanNet200. Our method achieves a significant improvement in comparison to the other methods. Consistent conclusion is also seen on this challenging dataset. It is important to note that previous works employ mask attention, while our approach does not. This verifies the success of our auxiliary center regression task in replacing mask attention.

S3DIS. As shown in Table 4, our method is evaluated on S3DIS Area5. Our approach outperforms previous works. This consistently shows the superiority of our method.

4.3. Ablation Study

We conduct an extensive ablation study to verify each component of our method as follows.

Learnable Position Query. The position query aims to provide an explicit center representation to the content query counterpart. Making it learnable intends to learn an optimal initial spatial distribution. We notice that some previous works [49, 42] adopt *non-parametric* initial queries,

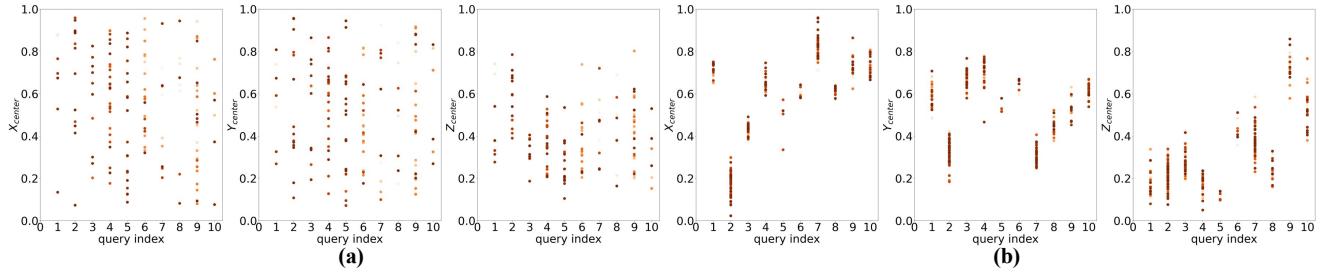


Figure 6. Spatial distribution of ground truths that are matched to the same query at different training iterations. (a): Baseline. (b): Ours. We randomly select 10 object queries and record their matched ground-truth instances in the early training. The three scatter plots for each method represent the center XYZ coordinates of their matched ground-truth instances at different training iterations (the scatter points for a query correspond to different training iterations). More visualization details are given in supplementary material. Our method manifests strong positional matching consistency among different training iterations.

Position	Content	mAP	mAP ₅₀	mAP ₂₅
FPS	zero	57.3	74.9	84.2
FPS	learnable	57.1	75.0	83.4
learnable	learnable	58.1	75.4	84.3
learnable	zero	58.4	75.9	84.5

Table 5. Ablation study on different initializations for position and content queries.

Position Encoding	mAP	mAP ₅₀	mAP ₂₅
No PE	0.0	0.0	0.0
Fourier APE	57.7	76.0	83.8
Content-conditioned APE	58.0	75.7	84.3
RPE	58.4	75.9	84.5

Table 6. Ablation study on different position encodings.

ID	iter. refine	center match	center loss	mAP	mAP ₅₀	mAP ₂₅
1		✓	✓	57.5	75.3	84.0
2	✓		✓	56.7	74.8	84.1
3	✓	✓		56.8	74.6	84.5
4	✓			56.4	74.7	83.7
5	✓	✓	✓	58.4	75.9	84.5

Table 7. Ablation study on iterative refinement and center matching & loss.

where Furthest Point Sampling (FPS) is used to sample a number of points and transform them into position encodings via Fourier transformation followed by an MLP. We make comparisons in Table 5. The results show that learnable position query and zero-initialized content query perform best. A potential reason why ‘FPS’ lags behind ‘learnable’ is that the latter learns an optimal spatial distribution.

Moreover, to show the pattern of the learnable position query, we visualize the spatial distribution of center coordinates of the matched ground truths for a query in Fig. 6. It shows that each query consistently attends to a local region.

Relative Position Encoding. We compare various position encodings that are employed in previous works [49, 41], such as Fourier Absolute Position Encoding (APE) and the content query-conditioned APE. Specifically, the latter uses an MLP to project the content query into a d -dim diagonal matrix, which then transforms the original absolute position encoding into a new one. It incorporates semantic information into the position encoding but does not consider relative relation. As shown in Table 6, RPE outperforms the others, which implies that both semantic information and relative relation are beneficial. Also, we notice that if we do not apply any position encoding, the training corrupts. This shows that positional prior is crucial in our framework.

Iterative Refinement. We remove the iterative refinement and freeze the position query in all decoder layers, and we find that it causes a performance drop of 0.9% mAP as shown in the first row of Table 7. This verifies the effectiveness of iterative refinement.

Center Matching & Loss. Moreover, to manifest the importance of center matching and center loss, we also conduct ablation studies in Table 7. We first remove the center matching and keep the center loss in the second row of the table, and we find that the performance drops by 1.7% mAP. Then we keep the center matching and remove the center loss. The performance also decreases by 1.6% mAP as shown in the 3-rd row. When both are absent, we observe an even larger performance drop (2.0% mAP) in the 4-th row. The results reveal that both center matching and loss are important to our framework.

4.4. Object Detection Results

The instance predictions of instance segmentation can be easily transformed into bounding box predictions, by obtaining the minimum and maximum coordinates of the masked instances. We empirically find that the generated object detection results from the instance predictions work

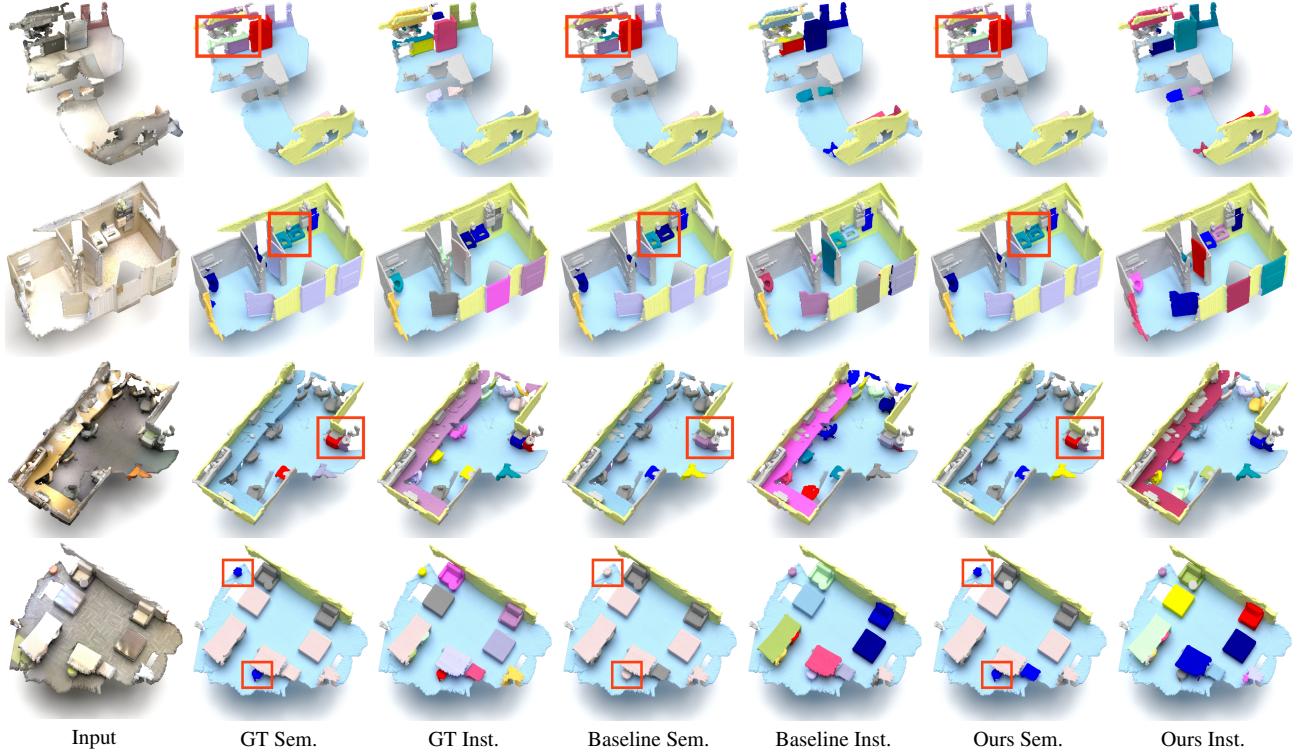


Figure 7. Visual comparison between baseline and ours (best viewed in color and by zoom-in). GT: Ground Truth. Sem.: Semantic labels. Inst.: Instance labels. The main difference is highlighted with a red bounding box. The bottom color map is for semantic labels. More examples are given in the supplementary material.

Method	task	box mAP ₅₀	box mAP ₂₅
VoteNet [44]	det	33.5	58.6
HGNet [4]	det	34.4	61.3
MLCVNet [65]	det	41.4	64.5
GSDN [18]	det	34.8	62.8
H3DNet [72]	det	48.1	67.2
3DETR [42]	det	47.0	65.0
Group-free [39]	det	52.8	69.1
RBGNet [57]	det	55.2	70.6
HyperDet3D [74]	det	57.2	70.9
FCAF3D [48]	det	57.3	71.5
CAGroup3D [56]	det	61.3	75.1
3D-MPA [17]	inst	49.2	64.2
Mask3D [49]	inst	56.2	70.2
Ours	inst	63.9	73.5

Table 8. 3D object detection results on ScanNetv2. For methods designed for the instance segmentation task, the bounding boxes are generated by the instance mask predictions.

significantly better than previous methods tailored for 3D object detection in terms of mAP₅₀, as shown in Table 8. This finding also shows that our approach outputs high-quality instance segmentation results with fewer artifacts.

4.5. Visual Comparison

We visually compare our approach with previous state-of-the-art methods in Fig. 7. More examples are given in

the supplementary material. The visualizations demonstrate that our method tends to correctly recognize the classes of the instances. It implies that our approach is able to generate more high-quality instance segmentation results.

5. Conclusion

In this work, we have presented a mask-attention-free transformer for the 3D instance segmentation task. We first observe the issue of low-recall of the initial masks in existing works. It adds training difficulty and slows down convergence. We thus avoid using mask attention and instead propose an auxiliary center regression task to guide the cross-attention. To fit center regression, we develop a series of designs. A dense distribution of position queries is learned to yield a higher recall of the perceived instances. Also, relative position encoding and iterative refinement are designed to further boost the performance. Each component is verified to be effective.

Acknowledgements

This work was supported in part by the Research Grants Council under the Areas of Excellence scheme grant AoE/E-601/22-R and Shenzhen Science and Technology Program KQTD20210811090149095.

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 *CVPR*, 2016. 2, 5
- [2] Yang Bo, Wang Jianan, Clark Ronald, Hu Qingyong, Wang Sen, Markham Andrew, and Trigoni Niki. Learning object bounding boxes for 3d instance segmentation on point clouds. In *NeurIPS*, pages 6737–6746, 2019. 2, 6
- [3] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, 2020. 3
- [4] Jintai Chen, Biwen Lei, Qingyu Song, Haochao Ying, Danny Z Chen, and Jian Wu. A hierarchical graph network for 3d object detection on point clouds. In *CVPR*, 2020. 8
- [5] Shaoyu Chen, Jiemin Fang, Qian Zhang, Wenyu Liu, and Xinggang Wang. Hierarchical aggregation for 3d instance segmentation. In *ICCV*, pages 15467–15476, 2021. 1, 2, 5, 6
- [6] Yukang Chen, Yanwei Li, Xiangyu Zhang, Jian Sun, and Jiaya Jia. Focal sparse convolutional networks for 3d object detection. In *CVPR*, 2022. 2
- [7] Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *CVPR*, 2022. 3
- [8] Bowen Cheng, Alex Schwing, and Alexander Kirillov. Per-pixel classification is not all you need for semantic segmentation. *NeurIPS*, 2021. 3
- [9] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4d spatio-temporal convnets: Minkowski convolutional neural networks. In *CVPR*, 2019. 5
- [10] Ruihang Chu, Yukang Chen, Tao Kong, Lu Qi, and Lei Li. Icm-3d: Instantiated category modeling for 3d instance segmentation. *IEEE Robotics and Automation Letters*, 7(1):57–64, 2021. 2
- [11] Ruihang Chu, Zhengzhe Liu, Xiaoqing Ye, Xiao Tan, Xiaojuan Qi, Chi-Wing Fu, and Jiaya Jia. Command-driven articulated object understanding and manipulation. In *CVPR*, 2023. 2
- [12] Ruihang Chu, Xiaoqing Ye, Zhengzhe Liu, Xiao Tan, Xiaojuan Qi, Chi-Wing Fu, and Jiaya Jia. Twist: Two-way inter-label self-training for semi-supervised 3d instance segmentation. In *CVPR*, 2022. 2
- [13] Xiangxiang Chu, Zhi Tian, Yuqing Wang, Bo Zhang, Haibing Ren, Xiaolin Wei, Huaxia Xia, and Chunhua Shen. Twins: Revisiting the design of spatial attention in vision transformers. *arXiv:2104.13840*, 2021. 3
- [14] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *CVPR*, 2017. 2, 5
- [15] Xiaoyi Dong, Jianmin Bao, Dongdong Chen, Weiming Zhang, Nenghai Yu, Lu Yuan, Dong Chen, and Baining Guo. Cswin transformer: A general vision transformer backbone with cross-shaped windows. *arXiv:2107.00652*, 2021. 3
- [16] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*, 2021. 3
- [17] Francis Engelmann, Martin Bokeloh, Alireza Fathi, Bastian Leibe, and Matthias Nießner. 3d-mpa: Multi-proposal aggregation for 3d semantic instance segmentation. In *CVPR*, pages 9031–9040, 2020. 2, 6, 8
- [18] JunYoung Gwak, Christopher Choy, and Silvio Savarese. Generative sparse detection networks for 3d single-shot object detection. In *ECCV*, 2020. 8
- [19] Lei Han, Tian Zheng, Lan Xu, and Lu Fang. Occuseg: Occupancy-aware 3d instance segmentation. In *CVPR*, pages 2940–2949, 2020. 2, 6
- [20] Tong He, Chunhua Shen, and Anton van den Hengel. Dyco3d: Robust instance segmentation of 3d point clouds through dynamic convolution. In *CVPR*, pages 354–363, 2021. 2, 6
- [21] Ji Hou, Angela Dai, and Matthias Nießner. 3d-sis: 3d semantic instance segmentation of rgb-d scans. In *CVPR*, pages 4421–4430, 2019. 2
- [22] Ding Jia, Yuhui Yuan, Haodi He, Xiaopei Wu, Haojun Yu, Weihong Lin, Lei Sun, Chao Zhang, and Han Hu. Detrs with hybrid matching. *arXiv preprint*, 2022. 3
- [23] Haiyong Jiang, Feilong Yan, Jianfei Cai, Jianmin Zheng, and Jun Xiao. End-to-end 3d point cloud instance segmentation without detection. In *CVPR*, pages 12796–12805, 2020. 2
- [24] Li Jiang, Shaoshuai Shi, Zhuotao Tian, Xin Lai, Shu Liu, Chi-Wing Fu, and Jiaya Jia. Guided point contrastive learning for semi-supervised point cloud semantic segmentation. In *ICCV*, 2021. 2
- [25] Li Jiang, Hengshuang Zhao, Shaoshuai Shi, Shu Liu, Chi-Wing Fu, and Jiaya Jia. Pointgroup: Dual-set point grouping for 3d instance segmentation. In *CVPR*, pages 4867–4876, 2020. 1, 2, 5, 6
- [26] Jean Lahoud, Bernard Ghanem, Marc Pollefeys, and Martin R Oswald. 3d instance segmentation via multi-task metric learning. In *ICCV*, pages 9256–9266, 2019. 2, 6
- [27] Xin Lai, Yukang Chen, Fanbin Lu, Jianhui Liu, and Jiaya Jia. Spherical transformer for lidar-based 3d recognition. In *CVPR*, 2023. 2
- [28] Xin Lai, Jianhui Liu, Li Jiang, Liwei Wang, Hengshuang Zhao, Shu Liu, Xiaojuan Qi, and Jiaya Jia. Stratified transformer for 3d point cloud segmentation. In *CVPR*, 2022. 2, 3, 4
- [29] Feng Li, Hao Zhang, Shilong Liu, Jian Guo, Lionel M Ni, and Lei Zhang. Dn-detr: Accelerate detr training by introducing query denoising. In *CVPR*, 2022. 3
- [30] Feng Li, Hao Zhang, Shilong Liu, Lei Zhang, Lionel M Ni, Heung-Yeung Shum, et al. Mask dino: Towards a unified transformer-based framework for object detection and segmentation. *arXiv preprint*, 2022. 3
- [31] Yanwei Li, Yilun Chen, Xiaojuan Qi, Zeming Li, Jian Sun, and Jiaya Jia. Unifying voxel-based representation with transformer for 3d object detection. *NeurIPS*, 2022. 2

- [32] Yanwei Li, Xiaojuan Qi, Yukang Chen, Liwei Wang, Zeming Li, Jian Sun, and Jiaya Jia. Voxel field fusion for 3d object detection. In *CVPR*, 2022. 2
- [33] Zhihao Liang, Zhihao Li, Songcen Xu, Mingkui Tan, and Kui Jia. Instance segmentation in 3d scenes using semantic superpoint tree networks. In *ICCV*, 2021. 2, 5, 6
- [34] Jiahui Liu, Chirui Chang, Jianhui Liu, Xiaoyang Wu, Lan Ma, and Xiaojuan Qi. Mars3d: A plug-and-play motion-aware model for semantic segmentation on multi-scan 3d point clouds. In *CVPR*, 2023. 2
- [35] Jianhui Liu, Yukang Chen, Xiaoqing Ye, and Xiaojuan Qi. Prior-free category-level pose estimation with implicit space transformation. *arXiv preprint arXiv:2303.13479*, 2023. 2
- [36] Jianhui Liu, Yukang Chen, Xiaoqing Ye, Zhuotao Tian, Xiao Tan, and Xiaojuan Qi. Spatial pruned sparse convolution for efficient 3d object detection. *NeurIPS*, 2022. 2
- [37] Shilong Liu, Feng Li, Hao Zhang, Xiao Yang, Xianbiao Qi, Hang Su, Jun Zhu, and Lei Zhang. Dab-detr: Dynamic anchor boxes are better queries for detr. In *ICLR*, 2022. 3
- [38] 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*, 2020. 6
- [39] Ze Liu, Zheng Zhang, Yue Cao, Han Hu, and Xin Tong. Group-free 3d object detection via transformers. In *ICCV*, 2021. 3, 8
- [40] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv:1711.05101*, 2017. 5
- [41] Depu Meng, Xiaokang Chen, Zejia Fan, Gang Zeng, Houqiang Li, Yuhui Yuan, Lei Sun, and Jingdong Wang. Conditional detr for fast training convergence. In *ICCV*, 2021. 3, 7
- [42] Ishan Misra, Rohit Girdhar, and Armand Joulin. An end-to-end transformer model for 3d object detection. In *ICCV*, 2021. 6, 8
- [43] Chunghyun Park, Yoonwoo Jeong, Minsu Cho, and Jaesik Park. Fast point transformer. In *CVPR*, 2022. 3
- [44] Charles R Qi, Or Litany, Kaiming He, and Leonidas J Guibas. Deep hough voting for 3d object detection in point clouds. In *ICCV*, 2019. 8
- [45] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *CVPR*, 2017. 2
- [46] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *NeurIPS*, 2017. 2
- [47] David Rozenberszki, Or Litany, and Angela Dai. Language-grounded indoor 3d semantic segmentation in the wild. In *ECCV*, 2022. 5
- [48] Danila Rukhovich, Anna Vorontsova, and Anton Konushin. Fcaf3d: fully convolutional anchor-free 3d object detection. In *ECCV*, 2022. 8
- [49] Jonas Schult, Francis Engelmann, Alexander Hermans, Or Litany, Siyu Tang, and Bastian Leibe. Mask3d for 3d semantic instance segmentation. *ICRA*, 2023. 1, 2, 3, 4, 5, 6, 7, 8
- [50] Jiahao Sun, Chunmei Qing, Junpeng Tan, and Xiangmin Xu. Superpoint transformer for 3d scene instance segmentation. *AAAI*, 2023. 1, 2, 3, 4, 5, 6
- [51] Shuyang Sun, Xiaoyu Yue, Song Bai, and Philip Torr. Visual parser: Representing part-whole hierarchies with transformers. *arXiv:2107.05790*, 2021. 3
- [52] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Herve Jegou. Training data-efficient image transformers & distillation through attention. In *ICML*, 2021. 3
- [53] Hugo Touvron, Matthieu Cord, Alexandre Sablayrolles, Gabriel Synnaeve, and Hervé Jégou. Going deeper with image transformers. *arXiv:2103.17239*, 2021. 3
- [54] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 2
- [55] Thang Vu, Kookhoi Kim, Tung M Luu, Thanh Nguyen, and Chang D Yoo. Softgroup for 3d instance segmentation on point clouds. In *CVPR*, 2022. 1, 2, 5, 6
- [56] Haiyang Wang, Lihe Ding, Shaocong Dong, Shaoshuai Shi, Aoxue Li, Jianan Li, Zhenguo Li, and Liwei Wang. Cagroup3d: Class-aware grouping for 3d object detection on point clouds. In *NeurIPS*, 2022. 8
- [57] Haiyang Wang, Shaoshuai Shi, Ze Yang, Rongyao Fang, Qi Qian, Hongsheng Li, Bernt Schiele, and Liwei Wang. Rbgnet: Ray-based grouping for 3d object detection. In *CVPR*, 2022. 8
- [58] Wenhui Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pvtt2: Improved baselines with pyramid vision transformer. *arXiv:2106.13797*, 2021. 3
- [59] Wenhui Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *ICCV*, 2021. 3
- [60] Weiyue Wang, Ronald Yu, Qiangui Huang, and Ulrich Neumann. Sgpn: Similarity group proposal network for 3d point cloud instance segmentation. In *CVPR*, pages 2569–2578, 2018. 2
- [61] Xinlong Wang, Shu Liu, Xiaoyong Shen, Chunhua Shen, and Jiaya Jia. Associatively segmenting instances and semantics in point clouds. In *CVPR*, pages 4096–4105, 2019. 2
- [62] Yingming Wang, Xiangyu Zhang, Tong Yang, and Jian Sun. Anchor detr: Query design for transformer-based detector. In *AAAI*, 2022. 3
- [63] Wenxuan Wu, Qi Shan, and Li Fuxin. Pointconformer: Revenge of the point-based convolution. *arXiv preprint*, 2022. 3
- [64] Xiaoyang Wu, Yixing Lao, Li Jiang, Xihui Liu, and Hengshuang Zhao. Point transformer v2: Grouped vector attention and partition-based pooling. *NeurIPS*, 2022. 3
- [65] Qian Xie, Yu-Kun Lai, Jing Wu, Zhoutao Wang, Yiming Zhang, Kai Xu, and Jun Wang. Mlcvnet: Multi-level context votenet for 3d object detection. In *CVPR*, 2020. 8
- [66] Jianwei Yang, Chunyuan Li, Pengchuan Zhang, Xiyang Dai, Bin Xiao, Lu Yuan, and Jianfeng Gao. Focal self-attention for local-global interactions in vision transformers. In *NeurIPS*, 2021. 3
- [67] 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 *CVPR*, pages 3947–3956, 2019. 2, 6
- [68] Qihang Yu, Huiyu Wang, Siyuan Qiao, Maxwell Collins, Yukun Zhu, Hartwig Adam, Alan Yuille, and Liang-Chieh Chen. k-means mask transformer. In *ECCV*, 2022. 3
- [69] Biao Zhang and Peter Wonka. Point cloud instance segmentation using probabilistic embeddings. In *CVPR*, pages 8883–8892, 2021. 2, 6
- [70] Gongjie Zhang, Zhipeng Luo, Yingchen Yu, Kaiwen Cui, and Shijian Lu. Accelerating detr convergence via semantic-aligned matching. In *CVPR*, 2022. 3
- [71] Hao Zhang, Feng Li, Shilong Liu, Lei Zhang, Hang Su, Jun Zhu, Lionel Ni, and Harry Shum. Dino: Detr with improved denoising anchor boxes for end-to-end object detection. In *ICLR*, 2022. 3
- [72] Zaiwei Zhang, Bo Sun, Haitao Yang, and Qixing Huang. H3dnet: 3d object detection using hybrid geometric primitives. In *ECCV*, 2020. 8
- [73] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip HS Torr, and Vladlen Koltun. Point transformer. In *ICCV*, 2021. 3
- [74] Yu Zheng, Yueqi Duan, Jiwen Lu, Jie Zhou, and Qi Tian. Hyperdet3d: Learning a scene-conditioned 3d object detector. In *CVPR*, 2022. 8
- [75] Min Zhong, Xinghao Chen, Xiaokang Chen, Gang Zeng, and Yunhe Wang. Maskgroup: Hierarchical point grouping and masking for 3d instance segmentation. In *ICME*, 2022. 1, 6
- [76] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. In *ICLR*, 2020. 3