Relation3D: Enhancing Relation Modeling for Point Cloud Instance Segmentation

Jiahao Lu
University of Science and Technology of China

Jiacheng Deng*
University of Science and Technology of China

Abstract

3D instance segmentation aims to predict a set of object instances in a scene, representing them as binary foreground masks with corresponding semantic labels. Currently, transformer-based methods are gaining increasing attention due to their elegant pipelines and superior predictions. However, these methods primarily focus on modeling the external relationships between scene features and query features through mask attention. They lack effective modeling of the internal relationships among scene features as well as between query features. In light of these disadvantages, we propose ***Relation3D: Enhancing Relation Modeling for Point Cloud Instance Segmentation****. Specifically, we introduce an adaptive superpoint aggregation module and a contrastive learning-guided superpoint refinement module to better represent superpoint features (scene features) and leverage contrastive learning to guide the updates of these features. Furthermore, our relation-aware self-attention mechanism enhances the capabilities of modeling relationships between queries by incorporating positional and geometric relationships into the self-attention mechanism. Extensive experiments on the ScanNetV2, ScanNet++, ScanNet200 and S3DIS datasets demonstrate the superior performance of Relation3D. Code is available at this website.*

1. Introduction

Point cloud instance segmentation aims to identify and segment multiple instances of specific object categories in 3D space. With the rapid development of fields such as robotic grasping [1], augmented reality [2, 3], 3D/4D reconstruction [4–8], and autonomous driving [9, 10], as well as the widespread application of LiDAR and depth sensor technologies [11, 12], point cloud instance segmentation has become a core technology for achieving efficient and accurate scene understanding. However, the unordered, sparse, and irregular nature of point cloud data, combined with the complex distribution of objects and the numerous categories in real-world applications, presents unique challenges for effective point cloud analysis. To tackle these challenges, early approaches mainly concentrated on accurately generating 3D bounding boxes (top-down) [13–15] or effectively grouping the points into instances with clustering algorithms (bottom-up) [16–18]. However, these methods have limitations: they either heavily rely on high-quality 3D bounding boxes, or require manual selection of geometric attributes.

Recently, researchers start to focus on the design of transformer-based methods [19–23]. These methods adopt an encoder-decoder framework (end-to-end), where each object instance is represented by an instance query. The encoder is responsible for learning the point cloud scene features, while the transformer decoder [24] iteratively at-

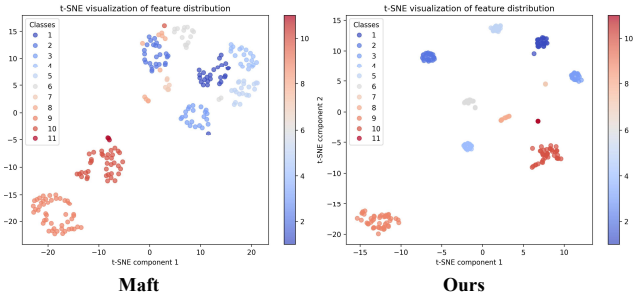


Figure 1. **T-SNE visualization of the superpoint-level feature distributions on ScanNetV2 validation set.** Different colors represent different instances. Our method highlights better inter-object diversity and intra-object similarity.

Point feature variation (Maft)	1.8603 (Excessive)			
Setting	Maft	Ours (Stage 1)	Ours (Stage 2)	Ours (Stage 3)
L_{cont}	1.057	0.7255	0.5841	**0.5739**

Table 1. **Excessive feature variation among points within the same supepoint and comparison of L_{cont} in different settings.** The experiment is conducted on ScanNetV2 validation set. L_{cont} measures the consistency of superpoint features within the same instance and the differences between features of different instances. Detailed information about L_{cont} can be found in Equation 5. Stage 1 represents the features output by ASAM, while stages 2 and 3 represent the features after refinement by CLSR.

*Corresponding Author

tends to the point cloud scene features to learn the instance queries. Ultimately, the instance queries can directly generate the masks for all instances in parallel. Current mainstream transformer-based methods commonly use mask-attention [25] to effectively model the external relationships between scene features and query features. However, they lack effective modeling for the internal relationships among scene features and between query features. As shown in Table 1 and the left panel of Figure 1, we observe insufficient consistency in superpoint features within the same instances, inadequate differentiation between features of different instances, and excessive feature variation among points within the same superpoint. These erroneous relationships between scene features undoubtedly increase the difficulty of instance segmentation. Besides, the effectiveness of self-attention lies in its establishment of relationships between query features. However, simply computing similarity between query features is too implicit and lacks adequate spatial and geometric relationship modeling, whose importance has been demonstrated in [26–28]. Although position embeddings are used to guide self-attention in transformer-based methods, the spatial information position embeddings provide is typically imprecise. For instance, SPFormer's [20] position embeddings are learnable and lack concrete spatial meaning, and in methods like Mask3D [19], Maft [22], and QueryFormer [21], discrepancies exist between the positions indicated by position embeddings and the actual spatial locations of each query's corresponding mask. This limitation in conventional self-attention prevents effective integration of implicit relationship modeling with spatial and geometric relationship modeling.

Based on the above discussion, we summarize two core issues that need to be considered and addressed in point cloud instance segmentation: *1) How to effectively model the relationships between scene features?* Most previous methods use pooling operations to obtain superpoint features [20, 22], but this pooling operation introduces unsuitable features and blurs distinctive features when there are large feature differences between points within a superpoint. Therefore, we need a new way to model superpoint features to emphasizing the distinctive point features. Additionally, considering the significant feature differences between superpoints within the same instance, we need to introduce scene feature relation priors to guide the superpoint features and model better superpoint relationships. *2) How to better model the relationships between queries?* Current self-attention designs rely on a simple computation of similarity between queries, but this implicit relationship modeling often requires extensive data and prolonged training to capture meaningful information. Therefore, integrating explicit spatial and geometric relationships is crucial, as it can refine attention focus areas and accelerate convergence. This motivates us to introduce instance-related biases to enhance the modeling of spatial and geometric relationships effectively.

Inspired by the above discussion, we propose *Relation3D: Enhancing Relation Modeling for Point Cloud Instance Segmentation*, which includes an adaptive superpoint aggregation module (ASAM), a contrastive learning-guided superpoint refinement module (CLSR), and relation-aware self-attention (RSA). To address the first issue, we propose an adaptive superpoint aggregation module and a contrastive learning-guided superpoint refinement module. In the adaptive superpoint aggregation module, we adaptively calculate weights for all points within each superpoint, emphasizing distinctive point features while diminishing the influence of unsuitable features. In the contrastive learning-guided superpoint refinement module, we first adopt a dual-path structure in the decoder, with bidirectional interaction and alternating updates between query features and superpoint features. This design enhances the representation ability of superpoint features. Furthermore, to optimize the update direction of superpoint features, we introduce contrastive learning [29–31] to provide contrastive supervision for superpoint features, reinforcing the consistency of superpoint features within instances and the differences between features of different instances, as is shown in Figure 1. To address the second issue, in relation-aware self-attention, we first model the explicit relationships between queries. By obtaining the mask and its bounding box corresponding to each query, we can model the positional and geometric relationships between queries. Next, we embed these relationships into self-attention as embeddings. Through this approach, we achieve an effective integration of implicit relationship modeling with spatial and geometric relationship modeling.

The main contributions of this paper are as follows: (i) We propose Relation3D: Enhancing Relation Modeling for Point Cloud Instance Segmentation, which achieves accurate and efficient point cloud instance segmentation predictions. (ii) The adaptive superpoint aggregation module and the contrastive learning-guided superpoint refinement module effectively enhances the consistency of superpoint features within instances and the differences between features of different instances. The relation-aware self-attention improves the relationship modeling capability between queries by incorporating the positional and geometric relationships into self-attention. (iii) Extensive experimental results on four standard benchmarks, ScanNetV2 [32], ScanNet++ [33], ScanNet200 [34], and S3DIS [35], show that our proposed model achieves superior performance compared to other transformer-based methods.

2. Related Work

Proposal-based Methods. Existing proposal-based methods are heavily influenced by the success of Mask R-CNN [36] for 2D instance segmentation. The core idea of these methods is to first extract 3D bounding boxes and then use a

mask learning branch to predict the mask of each object within the boxes. GSPN [13] adopts an analysis-by-synthesis strategy to generate high-quality 3D proposals, refined by a region-based PointNet [37]. 3D-BoNet [15] employs PointNet++[38] for feature extraction from point clouds and applies Hungarian Matching[39] to generate 3D bounding boxes. These methods set high expectations for proposal quality.

Grouping-based Methods. Grouping-based methods follow a bottom-up processing flow, first generating predictions for each point (such as semantic mapping and geometric displacement), and then grouping the points into instances based on these predicted attributes. PointGroup [40] segments objects on original and offset-shifted point clouds and employs ScoreNet for instance score prediction. SoftGroup [18] groups on soft semantic scores and uses a top-down refinement stage to refine the positive samples and suppress false positives. ISBNet [41] introduces a cluster-free approach utilizing instance-wise kernels. Recently, Spherical Mask [42] has addressed the low-quality outcomes of coarse-to-fine strategies by introducing a new alternative instance representation based on spherical coordinates.

Transformer-based Methods. Following 2D instance segmentation techniques [24, 43, 44], in the 3D field, each object instance is represented as an instance query, with query features learned through a vanilla transformer decoder. Transformer-based methods require the encoder to finely encode the point cloud structure in complex scenes and use the attention mechanism in the decoder to continuously update the features of the instance queries, aiming to learn the complete structure of the foreground objects as much as possible. Mask3D [19] and SPFormer [20] are pioneering works utilizing the transformer framework for 3D instance segmentation, employing FPS and learnable queries, respectively, for query initialization. QueryFormer [21] and Maft [22] build on Mask3D [19] and SPFormer [20] by improving query distribution. However, these works have not thoroughly explored the importance of internal relationships between scene features and between query features. Our method aims to enhance relation modeling for both scene features and query features to achieve better instance segmentation.

Relation modeling. Many 2D methods have demonstrated the importance of relation modeling. CORE [26] first leverages a vanilla relation block to model the relations among all text proposals and further enhances relational reasoning through instance-level sub-text discrimination in a contrastive manner. RE-DETR [28] incorporates relation modeling into component detection by introducing a learnable relation matrix to model class correlations. Relation-DETR [27] explores incorporating positional relation priors as attention biases to augment object detection. Our method is the first to explore the significance of relation priors in 3D instance segmentation. Through the adaptive superpoint aggregation module and the contrastive learning-guided superpoint refinement module, we progressively enhance the relationships among scene features. Additionally, relation-aware self-attention improves the relationships among queries.

3. Method

3.1. Overview

The goal of 3D instance segmentation is to determine the categories and binary masks of all foreground objects in the scene. The architecture of our method is illustrated in Figure 2. Assuming that the input point cloud has N points, each point contains position (x, y, z), color (r, g, b) and normal (n_x, n_y, n_z) information. Initially, we utilize a Sparse UNet [45] to extract point-level feature $F \in \mathbb{R}^{N \times C}$. Next, we perform adaptive superpoint aggregation module (Section 3.3) to acquire the superpoint-level features $F_{\text{super}} \in \mathbb{R}^{M \times C}$. Subsequently, we initialize several instance queries $Q \in \mathbb{R}^{K \times C}$ and input Q and F_{super} into the transformer decoder. To improve the relationship modeling capability between queries, we propose the relation-aware self-attention (Section 3.5). To update the features of F_{super}, we design a superpoint refinement module (Section 3.4) in the decoder, which is also a cross attention operation. However, unlike conventional cross-attention, the scene features F_{super} act as the $\mathcal{Q}$, while the instance queries Q serve as the $\mathcal{K}$ and $\mathcal{V}$. To guide the update direction of the superpoint features F_{super}, we implement a contrastive learning approach, which enhances the consistency of superpoint features within instances and increases the differences between features of different instances.

3.2. Backbone

We employ Sparse UNet [45] as the backbone for feature extraction, yielding features F, which is consistent with SPFormer [20] and Maft [22]. Next, we aggregate the point-level features F into superpoint-level features F_{super} via adaptive superpoint aggregation module, which will be introduced in the subsequent section.

3.3. Adaptive Superpoint Aggregation Module

The purpose of this module is to aggregate point-level features into superpoint-level features. To emphasize distinctive and meaningful point features while diminishing the influence of unsuitable features, we design the adaptive superpoint aggregation module, as shown in Figure 2 (b). Specifically, we first perform max-pooling and mean-pooling on the point-level features F according to the pre-obtained superpoints, resulting in $F_{\max}$ and F_{mean} respectively. Next, we calculate the difference between the superpoint-level features and the original point-level features F. We then utilize two non-shared weight MLPs to predict the corresponding

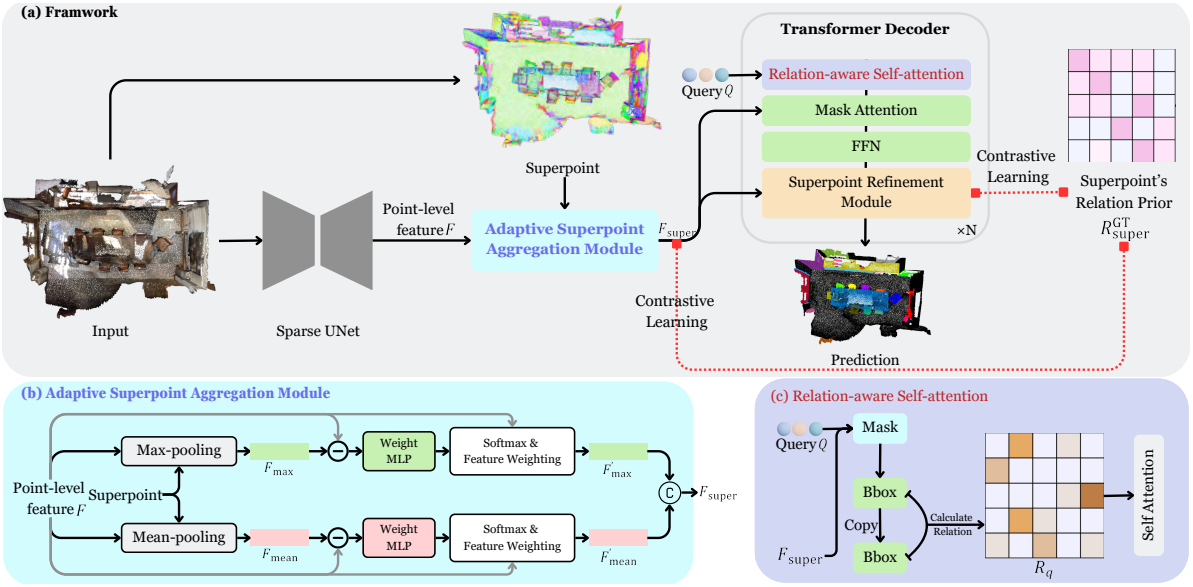


Figure 2. (a) The overall framework of our method Relation3D. (b) The details of our proposed adaptive superpoint aggregation module. (c) The details of our proposed relation-aware self-attention.

weights,

$$\mathcal{W}_{\max} = \mathrm{MLP}_1(F_{\max} - F), \tag{1}$$
$$\mathcal{W}_{\mathrm{mean}} = \mathrm{MLP}_2(F_{\mathrm{mean}} - F). \tag{2}$$

Getting the corresponding weights $\mathcal{W}_{\max}$ and $\mathcal{W}_{\mathrm{mean}}$, we apply a softmax operation to them in each superpoint. In this way, we can obtain the contribution of each point to its corresponding superpoint. We then use these weights, which sum to 1 in each superpoint, to perform feature weighting on F, resulting in $F'_{\max}$ and F'_{mean}. It's worth noting that the computation for each superpoint can be parallelized with point-wise MLP and torch-scatter extension library [46], so this superpoint-level aggregation is actually efficient. Finally, we concatenate $F'_{\max}$ and F'_{mean} to $[F'_{\max}, F'_{\min}]$ and input them into an MLP to reduce the $2C$ channels to C, obtaining the final superpoint-level features $F_{\mathrm{super}} \in \mathbb{R}^{M \times C}$.

3.4. Contrastive Learning-guided Superpoint Refinement Module

In the previous section, we introduce the adaptive superpoint aggregation module to emphasize distinctive point features within superpoints. Next, to further enhance the expressiveness of superpoints, we will leverage query features to update superpoint features within the transformer decoder. This design, in conjunction with the original mask attention, forms a dual-path architecture, enabling direct communication between query and superpoint features. This approach accelerates the convergence speed of the iterative updates.

Specifically, the superpoint refinement module employs a cross-attention mechanism for feature interaction. Here, we use the superpoint-level features F_{super} as the $\mathcal{Q}$ in the cross attention, while the instance queries Q serve as the $\mathcal{K}$ and $\mathcal{V}$. The specific structure is illustrated in Figure 3. To

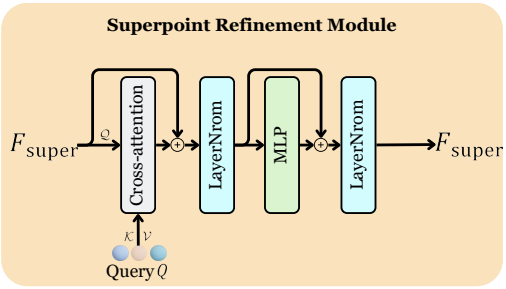


Figure 3. The superpoint refinement module. Superpoint-level features F_{super} serve as the $\mathcal{Q}$ in cross-attention, while the instance queries Q serve as the $\mathcal{K}$ and $\mathcal{V}$.

reduce computational and memory costs, we do not perform self-attention for self-updating F_{super}. Furthermore, the superpoint refinement module is not applied at every decoder layer. Instead, we perform the refinement of F_{super} every r layers to reduce computational resource consumption .

Furthermore, to guide the update direction, we have designed a contrastive learning mechanism, which constrains the consistency of superpoint features within the same instance and enlarge the differences between features of different instances First, we can obtain the superpoint's relation prior $R^{\mathrm{GT}}_{\mathrm{super}}$ based on instance annotations. If the current scene has M superpoints, then $R^{\mathrm{GT}}_{\mathrm{super}}$ is an $M \times M$ binary matrix defined as follows,

$$R^{\mathrm{GT}}_{\mathrm{super}}(i,j) = \begin{cases} 1, & \text{if } i \text{ and } j \text{ are in the same instance;} \\ 0, & \text{otherwise,} \end{cases} \tag{3}$$

where i and j represent two different superpoints. Next, we will compute the similarity between F_{super} features, defined

as follows,

$$\mathcal{S} = \mathbf{Norm}(F_{\text{super}})@\mathbf{Norm}(F_{\text{super}})^{\text{T}}. \tag{4}$$

Here, $\mathcal{S}$ represents the similarity matrix where each element quantifies the relationship between pairs of superpoint features. Finally, we will apply contrastive learning by comparing $\mathcal{S}$ and $R^{\text{GT}}_{\text{super}}$ as follows,

$$L_{cont} = \text{BCE}(\frac{\mathcal{S}+1}{2}, R^{\text{GT}}_{\text{super}}), \tag{5}$$

where L_{cont} is the contrastive loss computed using binary cross-entropy (BCE). This loss will encourage the model to enhance the consistency of superpoint features within the same instance while reinforcing the differences between features of different instances. Notably, we also add this loss function after the adaptive superpoint aggregation module, which can guide ASAM to focus on meaningful features within the superpoint that help enhance the consistency of superpoint features within the same instance.

3.5. Relation-aware Self-attention

Previous methods use traditional self-attention to model the relationships between queries, where each query contains a content embedding and a position embedding. They first add the position embedding to the content embedding before computing the attention map. However, in most methods [19–22], the position embedding does not accurately match the actual position of the mask predicted by the corresponding query, leading to imprecise implicit modeling of positional relationships. More explanation can be found in the supplemental materials. Inspired by Relation-DETR [27], to enhance the self-attention's ability to model positional relationships and to improve geometric relationship modeling, we propose a relation-aware self-attention (RSA).

To be specific, we calculate the binary mask $\mathbb{M}$ for each instance query Q. Next, we calculate the bounding box (bbox) corresponding to each mask, including its center point and scale: x, y, z, l, w, h. With the bbox calculated, we compute the relative relationships between queries as follows,

i. Positional Relative Relationship:

$$\left[\log\left(\frac{|x_i - x_j|}{l_i} + 1\right), \log\left(\frac{|y_i - y_j|}{w_i} + 1\right), \log\left(\frac{|z_i - z_j|}{h_i} + 1\right)\right];$$

ii. Geometric Relative Relationship:

$$\left[\log\left(\frac{l_i}{l_j}\right), \log\left(\frac{w_i}{w_j}\right), \log\left(\frac{h_i}{h_j}\right)\right],$$

where i, j represents two different queries. Next, we concatenate these two sets of relationships to form an embedding, denoted as $\mathfrak{T} \in \mathbb{R}^{K\times K\times 6}$. Then, following past methods, we use conventional sine-cosine encoding to increase the dimensionality of $\mathfrak{T} \in \mathbb{R}^{K\times K\times 6d}$,

$$\mathfrak{T}' = \sin\cos(\mathfrak{T}). \tag{6}$$

Finally, the embedding $\mathfrak{T}'$ undergoes a linear transformation to obtain $R_q \in \mathbb{R}^{K\times K\times \mathcal{H}}$, where $\mathcal{H}$ denotes the number of attention heads.

After obtaining R_q, we incorporate it into the traditional self-attention mechanism. The specific formula is as follows,

$$\text{RSA}(Q) = \text{Softmax}(\frac{\mathcal{Q}\mathcal{K}^T}{\sqrt{\mathcal{C}}} + R_q)\mathcal{V}. \tag{7}$$

In this formulation, we have $\mathcal{Q} = QW_q$, $\mathcal{K} = QW_k$, and $\mathcal{V} = QW_v$, where W_q, W_k, and W_v denote the linear transformation matrices for query, key, and value respectively.

3.6. Model Training and Inference

Apart from Maft's losses [22], our method includes an additional contrastive loss L_{cont},

$$\begin{aligned} L_{all} &= \lambda_1 L_{ce} + \lambda_2 L_{bce} + \lambda_3 L_{dice} \\ &+\lambda_4 L_{center} + \lambda_5 L_{score} + \lambda_6 L_{cont}, \end{aligned} \tag{8}$$

where λ_1, λ_2, λ_3, λ_4, λ_5, λ_6 are hyperparameters. During the model inference phase, we use the predictions from the final layer as the final output. In addition to the normal forward pass through the network, we also employ NMS [47] on the final output as a post-processing operation.

4. Experiments

4.1. Experimental Setup

Datasets and Metrics. We conduct our experiments on ScanNetV2 [32], ScanNet++ [33], ScanNet200 [34], and S3DIS [35] datasets. **ScanNetV2** comprises 1,613 scenes with 18 instance categories, of which 1,201 scenes are used for training, 312 for validation, and 100 for testing. **ScanNet++** contains 460 high-resolution (sub-millimeter) indoor scenes with dense instance annotations across 84 unique instance categories. **ScanNet200** uses the same point cloud data, but it enhances annotation diversity, covering 200 classes, 198 of which are instance classes. **S3DIS** is a large-scale indoor dataset collected from six different areas, containing 272 scenes with 13 instance categories. Following previous works [22], we use the scenes in Area 5 for validation and the remaining areas for training. AP@25 and AP@50 represent the average precision scores with IoU thresholds of 25% and 50%, respectively. mAP is the mean of all AP scores, calculated with IoU thresholds ranging from 50% to 95% in 5% increments. On ScanNetV2, we report mAP, AP@50, and AP@25. Additionally, we report Box AP@50 and AP@25 results, as done in SoftGroup [18] and Maft [22]. For ScanNet200 and ScanNet++, we report mAP, AP@50, and AP@25. On S3DIS, we report AP@50 and AP@25.

Implementation Details. We build our model on PyTorch framework [46] and train our model on a single

Method	ScanNetV2 validation					ScanNetV2 test		
	mAP	AP@50	AP@25	Box AP@50	Box AP@25	mAP	AP@50	AP@25
3D-SIS [14]	/	18.7	35.7	22.5	40.2	16.1	38.2	55.8
3D-MPA [16]	35.3	51.9	72.4	49.2	64.2	35.5	61.1	73.7
DyCo3D [49]	40.6	61.0	/	45.3	58.9	39.5	64.1	76.1
PointGroup [40]	34.8	56.9	71.3	48.9	61.5	40.7	63.6	77.8
MaskGroup [50]	42.0	63.3	74.0	/	/	43.4	66.4	79.2
OccuSeg [51]	44.2	60.7	/	/	/	48.6	67.2	74.2
HAIS [52]	43.5	64.4	75.6	53.1	64.3	45.7	69.9	80.3
SSTNet [17]	49.4	64.3	74	52.7	62.5	50.6	69.8	78.9
SoftGroup [18]	45.8	67.6	78.9	59.4	71.6	50.4	76.1	86.5
DKNet [53]	50.8	66.9	76.9	59.0	67.4	53.2	71.8	81.5
ISBNet [41]	54.5	73.1	82.5	62.0	78.1	55.9	75.7	83.5
Spherical Mask [42]	**62.3**	**79.9**	**88.2**	/	/	**61.6**	**81.2**	**87.5**
Mask3D [19]	55.2	73.7	82.9	56.6	71.0	56.6	78.0	87.0
QueryFormer [21]	56.5	74.2	83.3	61.7	73.4	58.3	78.7	87.4
SPFormer [20]	56.3	73.9	82.9	/	/	54.9	77.0	85.1
Maft [22]	58.4	75.9	84.5	63.9	73.5	57.8	77.4	/
Maft‡ [22]	59.9	76.5	/	/	/	59.6	78.6	86.0
Ours	**62.5**	**80.2**	**87.0**	**66.7**	**75.3**	**62.2**	**81.6**	**90.1**

Table 2. **Comparison on ScanNetV2 validation and hidden test set.** The second and third rows are the non-transformer-based and transformer-based methods, respectively. ‡ denotes using surface normal.

RTX4090 with a batch size of 6 for 512 epochs. We employ Maft [22] as the baseline. We employ AdamW [48] as the optimizer and PolyLR as the scheduler, with a maximum learning rate of 0.0002. Point clouds are voxelized with a size of 0.02m. For hyperparameters, we tune K, r as 400, 3 respectively. $\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5, \lambda_6$ in Equation 8 are set as 0.5, 1, 1, 0.5, 0.5, 1. Since ScanNet++ and ScanNet200 have more categories and instances, we set K as 500. All the other hyperparameters are the same for all datasets.

4.2. Comparison with existing methods.

Results on ScanNetV2. Table 2 reports the results on ScanNetV2 validation and hidden test set. Due to our focus on modeling the internal relationships between the scene features and between the queries, our approach outperforms other transformer-based methods, achieving an increase in mAP by 2.6, AP@50 by 3.7, AP@25 by 2.5, Box AP@50 by 2.8 and Box AP@25 by 1.8 in the validation set, and a rise in mAP by 2.6, AP@50 by 3.0 and AP@25 by 4.1 in the hidden test set. To vividly illustrate the differences between our method and others, we visualize the qualitative results in Figure 4. From the regions highlighted in red boxes, it is evident that our method can generate more accurate predictions.

Results on ScanNet++. Table 3 presents the results on ScanNet++ validation and hidden test set. The notable performance enhancement underscores the efficacy of our method in handling denser point cloud scenes.

Results on ScanNet200. Table 4 reports the results on ScanNet200 validation set. The significant performance improvement demonstrates the effectiveness of our method in handling complex and challenging scenes with a broader range of categories.

Results on S3DIS. We evaluate our method on S3DIS using Area 5 in Table 5. Our proposed method achieves better performance compared to previous methods, with gains in both AP@50 and AP@25, demonstrating the effectiveness and generalization of our method.

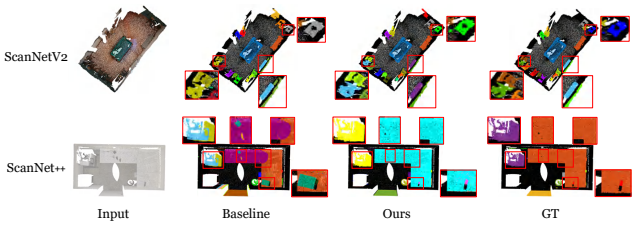


Figure 4. **Visualization of instance segmentation results on ScanNetV2 and ScanNet++ validation set.** The red boxes highlight the key regions.

Method	ScanNet++ validation			ScanNet++ test		
	mAP	AP@50	AP@25	mAP	AP@50	AP@25
PointGroup [40]	/	/	/	8.9	14.6	21.0
HAIS [52]	/	/	/	12.1	19.9	29.5
SoftGroup [18]	/	/	/	16.7	29.7	38.9
Maft [22]	23.1	32.6	39.7	20.9	31.3	40.4
Ours	**28.2**	**39.3**	**46.1**	**24.2**	**35.5**	**44.0**

Table 3. **Comparison on ScanNet++ validation and hidden test set.** ScanNet++ contains denser point cloud scenes and wider instance classes than ScanNetV2, with 84 distinct instance classes.

Method	ScanNet200 validation		
	mAP	AP@50	AP@25
SPFormer [20]	25.2	33.8	39.6
Mask3D [19]	27.4	37.0	42.3
QueryFormer [21]	28.1	37.1	43.4
Maft [22]	29.2	38.2	43.3
Ours	**31.6**	**41.2**	**45.6**

Table 4. **Comparison on ScanNet200 validation set.** ScanNet200 employs the same point cloud data as ScanNetV2 but enhances more annotation diversity, with 198 instance classes.

Method	AP@50	AP@25
PointGroup [40]	57.8	/
MaskGroup [50]	65.0	/
SoftGroup [18]	66.1	/
SSTNet [17]	59.3	/
SPFormer [20]	66.8	/
Mask3D [19]	68.4	75.2
QueryFormer [21]	69.9	/
Maft [22]	69.1	75.7
Spherical Mask [42]	72.3	/
Ours	**72.5**	**78.5**

Table 5. **Comparison on S3DIS Area5.** S3DIS contains 13 instance categories.

	ASAM	CLSR	RSA	mAP	AP@50	AP@25
[A]	✗	✗	✗	59.8	77.4	85.4
[B]	✓	✗	✗	60.1	77.9	85.6
[C]	✗	✓	✗	60.9	78.7	86.2
[D]	✓	✓	✗	61.5	78.8	86.3
[E]	✗	✗	✓	61.0	78.5	86.0
[F]	✓	✓	✓	**62.5**	**80.2**	**87.0**

Table 6. **Evaluation of the model with different designs on ScanNetV2 validation set.** ASAM refers to the adaptive superpoint aggregation module. CLSR refers to the contrastive learning-guided superpoint refinement module. RSA refers to the relation-aware self-attention.

Setting	mAP	AP@50	AP@25
W/o contrastive loss	60.1	77.9	85.6
W contrastive loss	**60.5**	**78.4**	**85.9**

Table 7. **Effectiveness of contrastive loss to ASAM.**

Setting	mAP	AP@50	AP@25
W max-pooling	62.2	79.9	86.7
W mean-pooling	62.3	79.7	86.7
W max-pooling & mean-pooling	**62.5**	**80.2**	**87.0**

Table 8. **Ablation study on ASAM.**

4.3. Ablation Studies

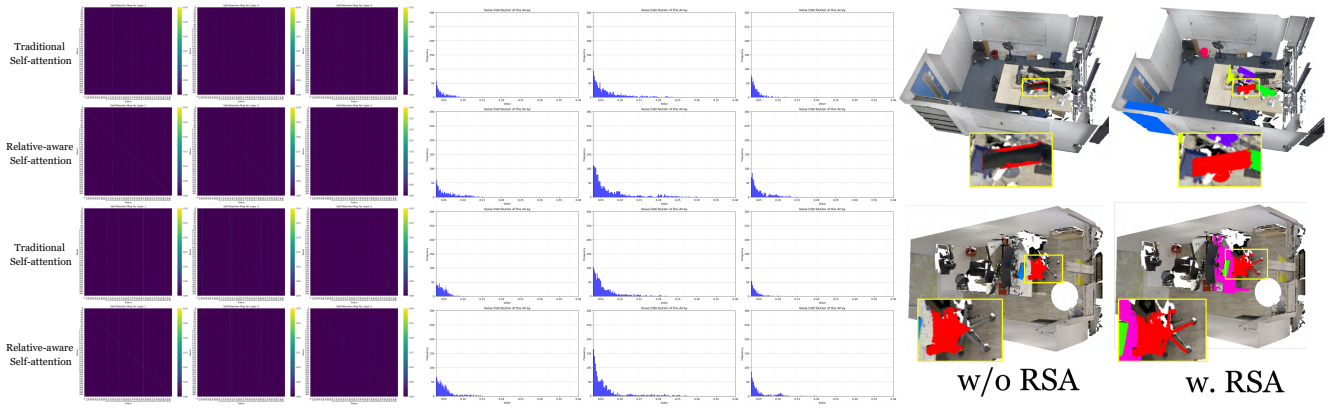
Evaluation of the model with different designs. To further study the effectiveness of our designs, we conduct ablation studies on ScanNetV2 validation set. As shown in Table 6, [A] represents the baseline of our method, which is Maft [22] using surface normals and NMS. [B] demonstrates that with the assistance of ASAM, which aims to better aggregate point-level features into superpoint-level features and emphasize distinctive and meaningful point features while diminishing the influence of unsuitable features, there is a performance improvement: mAP increases by 0.3, AP@50 by 0.5, and AP@25 by 0.2. However, due to the lack of guidance from contrastive loss (introduced in the CLSR), the aggregation direction of superpoints cannot be effectively controlled as expected, so the performance gain is limited. To validate this point, as shown in Table 7, adding contrastive loss to guide ASAM leads to further performance enhancement.

[C] shows that with the help of CLSR, we can interactively update superpoint features, and the use of contrastive learning guides the update direction by enforcing consistency of superpoint features within the same instance and increasing the difference between features of different instances. Compared to the baseline [A], the performance improves significantly, with mAP increasing by 1.1, AP@50 by 1.3, and AP@25 by 0.8. [D] combines ASAM and CLSR. In this design, not only the contrastive learning embedded within CLSR provides ASAM with a clear direction for feature aggregation but also ASAM can offer better-initialized superpoint features to CLSR. This synergistic design cooperates well and results in a 0.6 mAP improvement. [E] demonstrates the effectiveness of RSA, which enhances the self-attention mechanism's ability to model positional relationships and improves geometric relationship modeling. Compared to [A], RSA leads to an improvement of 1.2 mAP and 1.1 AP@50. Finally, in [F], we present the performance of the complete model, underscoring the essential roles played by each module in 3D instance segmentation.

Importance of different designs. I: RSA incorporates explicit relationship modeling between queries, which helps the network learn and converge more easily by focusing on more relevant queries, compared to purely implicit modeling. II: ASAM and CLSR are essentially a unified entity (designed to solve the same problem: better modeling of scene relationships). We separated them only for clarity in description. Both I and II are equally important for instance segmentation, and their contributes comparably to the final performance.

Ablation study on the adaptive superpoint aggregation module. In this section, we conduct experiments on the adaptive superpoint aggregation module (ASAM). First, we perform an ablation study on max-pooling and mean-pooling, as shown in Table 8, where "W max-pooling" indicates that ASAM includes only the max-pooling branch. The results show that both mean-pooling and max-pooling contribute to performance gains. Furthermore, to illustrate the characteristics of the learned weight distribution in ASAM, we present corresponding visualization in Figure 6. From the figure, it is evident that ASAM places greater emphasis on the edges and corner regions of objects—areas that are typically distinctive for each instance. Therefore, with the assistance of ASAM and contrastive learning, the model is able to aggregate more discriminative superpoint features.

Effectiveness of the relation-aware self-attention. As shown in Figure 5, we compare the attention maps and attention weight distributions between traditional self-attention and relation-aware self-attention. From Figure 5(a), it can be observed that our proposed relation-aware self-attention has more high-weight focal points in its attention map, which is further supported by the data in Figure 5(b). Notably, we have excluded points with attention values ranging from 0 to 0.03 from our statistical analysis, as these account for the vast majority (approximately 99%) of the attention map and would otherwise obscure the meaningful patterns in our study. Furthermore, we substantiate this from a visualization perspective. As demonstrated in Figure 5(c), with the aid of RSA, the representative query can forge connections with a broader set of relevant queries, in contrast to traditional attention mechanisms that concentrate on a limited number of specific queries. This enhancement facilitates the generation of superior instance masks. These observations indicate



(a) Attention Map (b) Attention Weights Distribution (c) Query (yellow bbox) with its highly related queries

Figure 5. **(a) Comparison of attention maps for traditional self-attention vs. relation-aware self-attention.** We display the progression of attention maps from layer 1, 3, 5. **(b) Comparison of attention weight distributions for traditional self-attention vs. relation-aware self-attention.** The attention weight distributions are also shown from layer 1, 3, 5. **(c) Query (yellow bbox) with its highly related queries.**

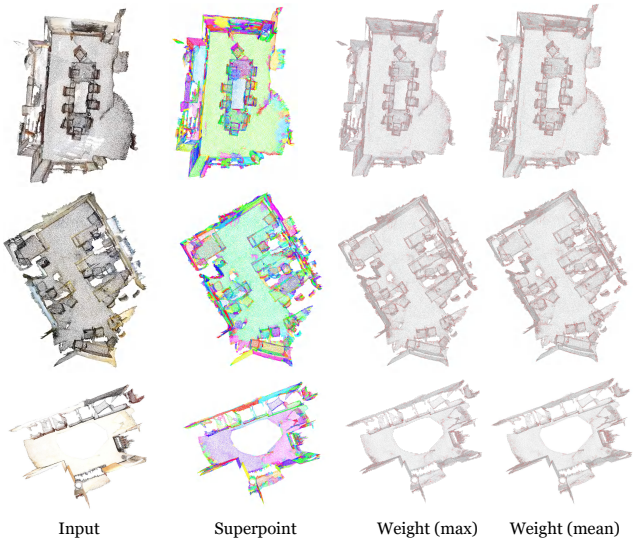


Figure 6. **Visualization of weights in the adaptive superpoint aggregation module.** A deeper red color indicates a higher weight assigned to point feature during the "Softmax & Feature Weighting" stage (Figure 2 (b)).

that relation-aware self-attention achieves a more focused attention when modeling position and geometric relationships. Unlike traditional self-attention, which has relatively dispersed attention without any specific focal query, relation-aware self-attention selectively emphasizes relevant queries, resulting in a more precise and meaningful representation.

Contribution to the convergence speed. As shown in Figure 7, our method demonstrates a faster convergence speed compared to the baseline. This improvement can be attributed to the relation priors introduced by CLSR and RSA: contrastive learning provides relation priors for superpoints to guide feature aggregation, while RSA introduces position and geometric relation priors for query features, enhancing self-attention. Additionally, the superpoint refinement module in CLSR forms a dual-path architecture, enabling direct communication between query features and superpoint features, speeding up the convergence.

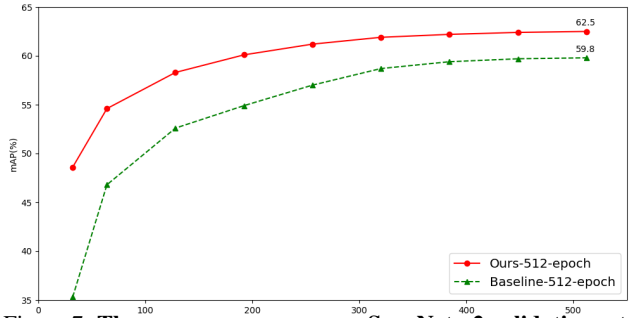


Figure 7. **The convergence curve on ScanNet-v2 validation set.**

5. Conclusion

In this paper, we propose a novel 3D instance segmentation method called Relation3D. We focus on modeling the internal relationships among scene features as well as between query features, an aspect that past methods have not explored sufficiently. Specifically, we introduce an adaptive superpoint aggregation module to better represent superpoint features and a contrastive learning-guided superpoint refinement module that updates superpoint features in dual directions while guiding the direction of these updates with the help of contrastive learning. Additionally, our proposed relation-aware self-attention mechanism enhances the modeling of relationships between queries by improving the representation of positional and geometric relationships. Extensive experiments conducted on the several datasets demonstrate the effectiveness of Relation3D.

6. Acknowledgements

This work was partially supported by Wenfei Yang and Tianzhu Zhang.

References

[1] Chungang Zhuang, Shaofei Li, and Han Ding. Instance segmentation based 6d pose estimation of industrial objects using point clouds for robotic bin-picking. *Robotics and Computer-Integrated Manufacturing*, 82:102541, 2023.

[2] 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.

[3] Alessandro Manni, Damiano Oriti, Andrea Sanna, Francesco De Pace, and Federico Manuri. Snap2cad: 3d indoor environment reconstruction for ar/vr applications using a smartphone device. *Computers & Graphics*, 100:116–124, 2021.

[4] Guanjun Wu, Taoran Yi, Jiemin Fang, Lingxi Xie, Xiaopeng Zhang, Wei Wei, Wenyu Liu, Qi Tian, and Xinggang Wang. 4d gaussian splatting for real-time dynamic scene rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20310–20320, 2024.

[5] Ruijie Zhu, Yanzhe Liang, Hanzhi Chang, Jiacheng Deng, Jiahao Lu, Wenfei Yang, Tianzhu Zhang, and Yongdong Zhang. Motiongs: Exploring explicit motion guidance for deformable 3d gaussian splatting. *arXiv preprint arXiv:2410.07707*, 2024.

[6] Jiahao Lu, Tianyu Huang, Peng Li, Zhiyang Dou, Cheng Lin, Zhiming Cui, Zhen Dong, Sai-Kit Yeung, Wenping Wang, and Yuan Liu. Align3r: Aligned monocular depth estimation for dynamic videos. *arXiv preprint arXiv:2412.03079*, 2024.

[7] Jiahao Lu, Jiacheng Deng, Ruijie Zhu, Yanzhe Liang, Wenfei Yang, Tianzhu Zhang, and Xu Zhou. Dn-4dgs: Denoised deformable network with temporal-spatial aggregation for dynamic scene rendering. *arXiv preprint arXiv:2410.13607*, 2024.

[8] Jonathon Luiten, Georgios Kopanas, Bastian Leibe, and Deva Ramanan. Dynamic 3d gaussians: Tracking by persistent dynamic view synthesis. *arXiv preprint arXiv:2308.09713*, 2023.

[9] Davy Neven, Bert De Brabandere, Stamatios Georgoulis, Marc Proesmans, and Luc Van Gool. Towards end-to-end lane detection: an instance segmentation approach. In *2018 IEEE intelligent vehicles symposium (IV)*, pages 286–291. IEEE, 2018.

[10] 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.

[11] 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.

[12] Maciej Halber, Yifei Shi, Kai Xu, and Thomas Funkhouser. Rescan: Inductive instance segmentation for indoor rgbd scans. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2541–2550, 2019.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

[18] 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.

[19] 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.

[20] Jiahao Sun, Chunmei Qing, Junpeng Tan, and Xiangmin Xu. Superpoint transformer for 3d scene instance segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 2393–2401, 2023.

[21] Jiahao Lu, Jiacheng Deng, Chuxin Wang, Jianfeng He, and Tianzhu Zhang. Query refinement transformer for 3d instance segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 18516–18526, 2023.

[22] Xin Lai, Yuhui Yuan, Ruihang Chu, Yukang Chen, Han Hu, and Jiaya Jia. Mask-attention-free transformer for 3d instance segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3693–3703, 2023.

[23] Jiahao Lu, Jiacheng Deng, and Tianzhu Zhang. Beyond the final layer: Hierarchical query fusion transformer with agent-interpolation initialization for 3d instance segmentation. *arXiv preprint arXiv:2502.04139*, 2025.

[24] 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.

[25] 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.

[26] Jingyang Lin, Yingwei Pan, Rongfeng Lai, Xuehang Yang, Hongyang Chao, and Ting Yao. Core-text: Improving scene text detection with contrastive relational reasoning. In *2021 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6. IEEE, 2021.

[27] Xiuquan Hou, Meiqin Liu, Senlin Zhang, Ping Wei, Badong Chen, and Xuguang Lan. Relation detr: Exploring explicit position relation prior for object detection. *arXiv preprint arXiv:2407.11699*, 2024.

[28] Xixuan Hao, Danqing Huang, Jieru Lin, and Chin-Yew Lin. Relation-enhanced detr for component detection in graphic design reverse engineering. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*, pages 4785–4793, 2023.

[29] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *Advances in neural information processing systems*, 33:18661–18673, 2020.

[30] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

[31] Jiahao Lu, Jiacheng Deng, and Tianzhu Zhang. Bsnet: Box-supervised simulation-assisted mean teacher for 3d instance segmentation. *arXiv preprint arXiv:2403.15019*, 2024.

[32] 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.

[33] Chandan Yeshwanth, Yueh-Cheng Liu, Matthias Nießner, and Angela Dai. Scannet++: A high-fidelity dataset of 3d indoor scenes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12–22, 2023.

[34] David Rozenberszki, Or Litany, and Angela Dai. Language-grounded indoor 3d semantic segmentation in the wild. In *European Conference on Computer Vision*, pages 125–141. Springer, 2022.

[35] 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.

[36] 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.

[37] 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.

[38] 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.

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

[40] 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.

[41] Tuan Duc Ngo, Binh-Son Hua, and Khoi Nguyen. Isbnet: a 3d point cloud instance segmentation network with instance-aware sampling and box-aware dynamic convolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13550–13559, 2023.

[42] Sangyun Shin, Kaichen Zhou, Madhu Vankadari, Andrew Markham, and Niki Trigoni. Spherical mask: Coarse-to-fine 3d point cloud instance segmentation with spherical representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4060–4069, 2024.

[43] 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.

[44] Feng Li, Hao Zhang, Huaizhe Xu, Shilong Liu, Lei Zhang, Lionel M Ni, and Heung-Yeung Shum. Mask dino: Towards a unified transformer-based framework for object detection and segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3041–3050, 2023.

[45] Spconv Contributors. Spconv: Spatially sparse convolution library. `https://github.com/traveller59/spconv`, 2022.

[46] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

[47] 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.

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

[49] 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.

[50] 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.

[51] 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.

[52] 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.

[53] 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.