Contrastive Mask Denoising Transformer for 3D Instance Segmentation

He Wang[1], Minshen Lin[2] and Guofeng Zhang[1*]

***Abstract*— In transformer-based methods for point cloud instance segmentation, bipartite matching is used to establish one-to-one correspondences between predictions and ground truths. However, in early training stages, matches can be unstable and inconsistent between epochs, requiring the model to frequently adjust its learning path, thus reducing the quality of model convergence. To address this challenge, we propose the contrastive mask denoising transformer for 3D instance segmentation, which utilizes a mask denoising module to guide the model towards a more stable optimization path in early training stages. Furthermore, we introduce a multi-pattern-aware query selection module to assist the model learn multiple patterns at one position such that clustered objects can be discerned. In addition, the proposed modules are "plug and play", which can easily be integrated into transformer-based architectures. Experimental results on ScanNetv2 dataset show that the proposed modules improve the performance of multiple pipelines, notably achieving +1.0 mAP on the main pipeline.**

I. INTRODUCTION

3D instance segmentation can enhance machines' spatial awareness and elevate automation efficiency, thereby demonstrating the vast potential for applications across fields such as robotics, autonomous driving, and augmented reality [1]. Contrary to the structured nature of 2D images, 3D point clouds are made up of numerous sparse and disorganized points. As a result, achieving precise instance masks in such cluttered and unstructured point clouds is a highly challenging task for 3D point cloud segmentation.

Classical methods for 3D point cloud segmentation [2, 3, 4, 5, 6] often rely on many hand-crafted components [7], preventing end-to-end training. Detection transformer (DETR) introduced the use of a set-based global loss that forces unique predictions via bipartite matching, effectively establishing an end-to-end 2D object detection pipeline and providing new directions for subsequent research [8]. Many researchers followed DETR to further improve its performance [9, 10, 11, 12, 13], and recently, it has been adapted to solve the challenging 3D instance segmentation task [14, 15].

Although DETR-like architectures have the advantage in streamlining the training process, they have some mutual shortcomings: 1) In the bipartite matching process [16, 17], inaccurate initial matches can lead to incorrect feedback, which in turn can negatively impact the learning path of the model and hinder performance improvements. 2) The queries for predicting instances are learnable or set to 0 vectors, which does not fully exploit prior information such as point cloud features. Furthermore, the architecture lacks a mechanism to effectively handle situations where multiple objects occupy nearby locations in one scene.

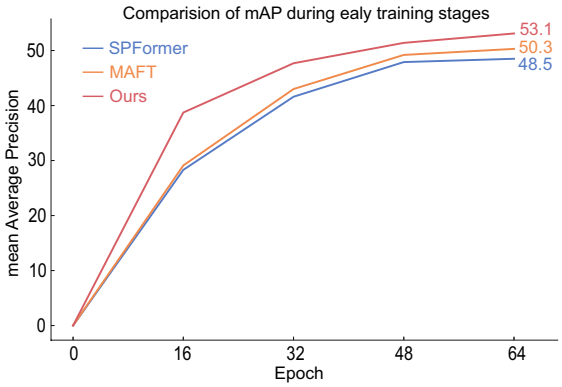

Fig. 1: **The mAP curves of different models during early training stages.** We evaluate the models every 16 epochs. With the contrastive denoising module, our model's performance improved by 2.8 and 4.6, respectively, compared to the other two models at 64 epochs.

Regarding 2D object detection task, numerous efforts have been made to address the aforementioned issues from various perspectives [18, 19]. However, for 3D detection and segmentation tasks, DETR-like models are in their nascent stage with limited works focusing on further optimizing these issues.

To address these two issues, we propose the contrastive mask denoising transformer (MaDFormer) for 3D instance segmentation. Our approach uses an auxiliary mask denoising task to tackle the mismatches caused by inaccurate predictions in early training stages and guide the model towards a better learning path. Simultaneously, we propose a multi-pattern-aware query selection module to fully leverage the backbone features during query initialization, improving the prediction results in early training stages. Additionally, to tackle the issue of identifying clustered objects near the same spatial location in one scene, we introduce multiple patterns for the query, which further enhances the model's accuracy for complex environments.

As shown in Fig. 1, in the early training stages, the mAP of our method is significantly improved compared to existing transformer-based methods. Moreover, experiments show

*Corresponding Author

[1]He Wang and Guofeng Zhang are with the State Key Lab of CAD&CG, Zhejiang University, Hangzhou, China 11921089@zju.edu.cn, zhangguofeng@zju.edu.cn

[2]Minshen Lin is with the College of Electrical Engineering, Zhejiang University, Hangzhou, China linminshen@zju.edu.cn

that our approach can effectively enhance the performance of DETR-like transformer models, achieving performance improvements of +0.9 and +1.0 mAP on different pipelines, respectively [14, 20].

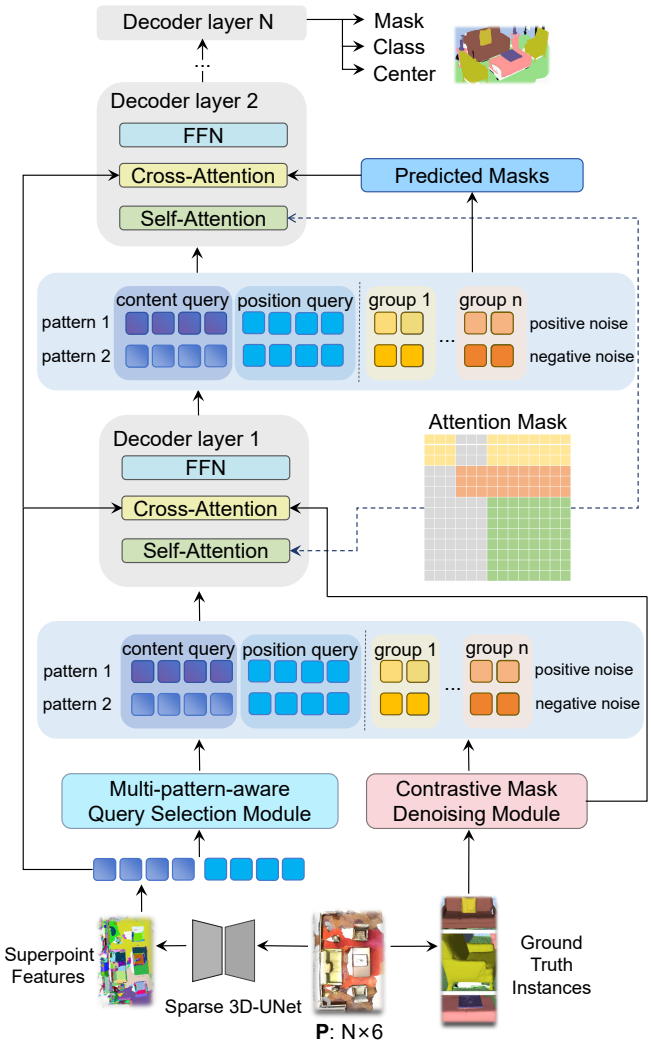

Fig. 2: **The overall architecture of MaDFormer.** We use n groups of noised queries as the input queries of the denoising-part, including both positive noise and negative noise. Simultaneously, we repeat selected queries for m times as input queries of matching-part, where m refers to the number of patterns. An attention mask is used to ensure there is no information leakage between these two parts.

The main contributions of this paper are summarized as follows:

1) We propose a contrastive mask denoising module to improve the accuracy of bipartite matching during the early training stages and enhance the final performance.
2) We design a multi-pattern-aware query selection module, further enhancing the model's ability to recognize complex scenes.
3) Our method can be used as a plug-and-play module in transformer-based models to further enhance their performance. By integrating our proposed modules into existing state-of-the-art transformer-based models for 3D instance segmentation, we achieve superior results with an increase in mAP by +1.0 on ScanNetv2 dataset.

II. Related Work

A. 3D Instance Segmentation

Classic methods. In 3D point cloud segmentation, classic approaches typically fall into two categories: proposal-based [21, 22, 2, 23] and grouping-based [24, 4, 25, 26] methods. Most of the proposal-based methods predict instance masks by generating 3D bounding boxes. Grouping-based methods first aggregate the points that have similar features into groups and then refine the prediction in a top-down [27] way. These approaches have achieved remarkable results and played dominant roles in the task for a long time. However, these two types of methods have obvious drawbacks: they usually require many hand-crafted components and complex post-processing steps [28], and the quality of the intermediate results greatly affects the final segmentation results.

Transformer-Based Methods. Several works attempt to apply the transformer architecture to 3D point cloud segmentation tasks [29]. Point Transformer [30] and point cloud transformer [31] made the first attempts to introduce the transformer layer and attention module to 3D segmentation tasks and achieved significant progress. Subsequently, Mask3D [15] and SPFormer [14] achieve superior results, utilizing the end-to-end training pipeline based on transformer structures adapted from Mask2Former [32]. Specifically, SPFormer [14] proposes using superpoint as an intermediate structure, combining the advantages of both bottom-up and top-down approaches in the transformer architecture. MAFT [20], an extension of SPFormer [14], proposes replacing the initial mask attention with a center regression task to improve convergence speed. QueryFormer [33] proposes adjusting the query distribution, to optimize the coverage and repetition rates of queries. Although existing methods have proposed several different solutions to refine queries through the decoder, they have not fully utilized the prior information in the backbone features, nor have they adequately considered the situation where multiple objects are present at close positions.

B. 2D Vision Transformer

Transformer-based models can flexibly capture global information and long-range relations within a scene using the attention mechanism [34], whereby they have gradually become the mainstream frameworks in the 2D vision field [35, 36, 37]. Subsequently, DETR and its variants [8, 11, 19], with the distinctive end-to-end training feature, have been extensively applied in tasks such as 2D instance segmentation, object detection, and panoptic segmentation. As DETR-like models evolve, they have also revealed several

new challenges, such as high memory consumption and slow convergence. To address these issues, Deformable DETR [9] proposes to improve the algorithm's efficiency by concentrating the attention modules on a limited number of important sampling points near a reference point. To solve the issue of ambiguous query meanings in DETR that are difficult to optimize, Anchor DETR [13] uses fixed anchor points to initialize queries so that the queries can focus on the objects near the anchor point. DN-DETR [10] and DINO [18] use auxiliary denoising tasks to solve the problem of slow convergence and unstable bipartite matching. By introducing noised ground truth labels and bounding boxes as denoising-part queries, the model learns to reconstruct objects without bipartite matching as an auxiliary task, making relative offset learning in decoder easier [10]. In addition, negative query denoising has been proposed to train the model to predict "no object" if the noise is large, further enhancing the stability of bipartite matching [18]. To unify the modeling framework for object detection and segmentation in 2D, MaskFormer [38], Mask2Former [32], OneFormer [39], and MaskDINO [40] have further modified the DETR framework, making the models compatible with multiple tasks. Though many studies focus on DETR-like models for 2D tasks, their applications to 3D tasks have not been fully investigated yet.

Inspired by the works for 2D tasks, we analyze the drawbacks of the current 3D instance segmentation models. Specifically, we explore the instability of bipartite matching and ambiguous query meanings and thereby propose a contrastive mask denoising module and a multi-pattern-aware query selection module to enhance the model's performance.

III. Methods

A. Overview

The architecture of our proposed model is shown in Fig. 2. We adopt MAFT [20] as our baseline model, which uses a sparse U-Net feature backbone and superpoint-pooling layer to extract and aggregate features, followed by query decoders and prediction heads to predict masks and labels. To address the challenge of bipartite matching instability and the difficulty of multiple objects in one region, we propose a mask denoising module to guide the model towards a more stable optimization path in early training stages and a multi-pattern-aware query selection module to help predict multiple objects at one position.

As shown in Fig. 2, given a point cloud, our model first extracts spatial features using the U-Net backbone. The superpoint-pooling layer then aggregates point-wise features into superpoint features $\mathcal{F} \in \mathbb{R}^{m \times d}$ and yields the corresponding positions $\mathcal{P} \in \mathbb{R}^{m \times 3}$ for each superpoint, where m is the number of superpoints. Subsequently, the query decoder utilizes the superpoint features to update the queries iteratively via masked cross-attention with superpoint features. Apart from the N matching-part queries $\mathcal{Q}_m \in \mathbb{R}^{N \times d}$ from the baseline, we include n denoising-part queries $\mathcal{Q}_{dn} \in \mathbb{R}^{n \times d}$, which are concatenated with $\mathcal{Q}_m$ for iterative update in the decoder and final loss computation. These denoising-part queries are generated based on noised ground-truth labels and positions, guiding the model to stabilize bipartite matching via auxiliary noised-object reconstructing tasks. Importantly, we introduce noised ground truth masks $\mathcal{M}_{dn}$ for the denoising-part queries to better feed the ground-truth information into the decoder for instance segmentation task. Besides, the matching-part queries are selected from the superpoint features with different strategies as better priors, and multiple patterns for each selected feature are used to predict clustered objects at one position.

B. Contrastive Mask Denoising Module

The slow convergence problem of DETR-like models persists in SPFormer and MAFT. One reason for slow convergence is that the bipartite matching component necessary for end-to-end training is discrete and stochastic in nature, leading to unstable matching, especially in the early training stages [10]. This means that a query can be matched with different objects from epoch-to-epoch such that the optimization process is ambiguous and inconsistent.

Following these works in 2D object detection task [10, 18, 40], we propose a contrastive mask denoising module to stabilize the bipartite matching for 3D instance segmentation task. This module mainly consists of three components.

1) The first component includes several groups of denoising-part queries. Each group has n positive queries $\mathcal{Q}_{dn}^{pos} \in \mathbb{R}^{n \times d}$ and n negative queries $\mathcal{Q}_{dn}^{neg} \in \mathbb{R}^{n \times d}$. For a scene with n ground truth instances $I_1, I_2, ..., I_n$, the i-th denoising-part query is generated based on the label L_i and the position $\mathcal{P}_i$ of the instance I_i. The label noise is introduced by randomly changing the label to any label with a probability of λ_1. Another hyperparameter, λ_2, controls the positional noise scale. For positive queries, the positional noise is introduced by randomly shifting the position $\mathcal{P}_i$ within the range $\pm(\frac{\lambda_2}{2}w_i, \frac{\lambda_2}{2}h_i, \frac{\lambda_2}{2}l_i)$, where w_i, h_i, l_i are the width, height, and depth of the bounding box of I_i; for negative queries, the positional noise is introduced similarly but using a noise scale of $2\lambda_2$.
2) The second component is a set of denoising-part cross-attention masks $\mathcal{M}_{dn}$, and each mask corresponds to a denoising-part query. For the i-th positive query, the positive mask is generated based on the instance mask of I_i. The mask noise is introduced by randomly flipping the positions where the instance exists with a probability of λ_3, which essentially corrupts the ground truth mask, and randomly flipping the positions where the instance doesn't exist with a probability of $\frac{\lambda_3}{2}$. By setting a small value for λ_3, such a flipping scheme can ensure that most of the mask information is retained. For the i-th negative query, the negative mask is also generated based on the instance mask of I_i, and the mask noise

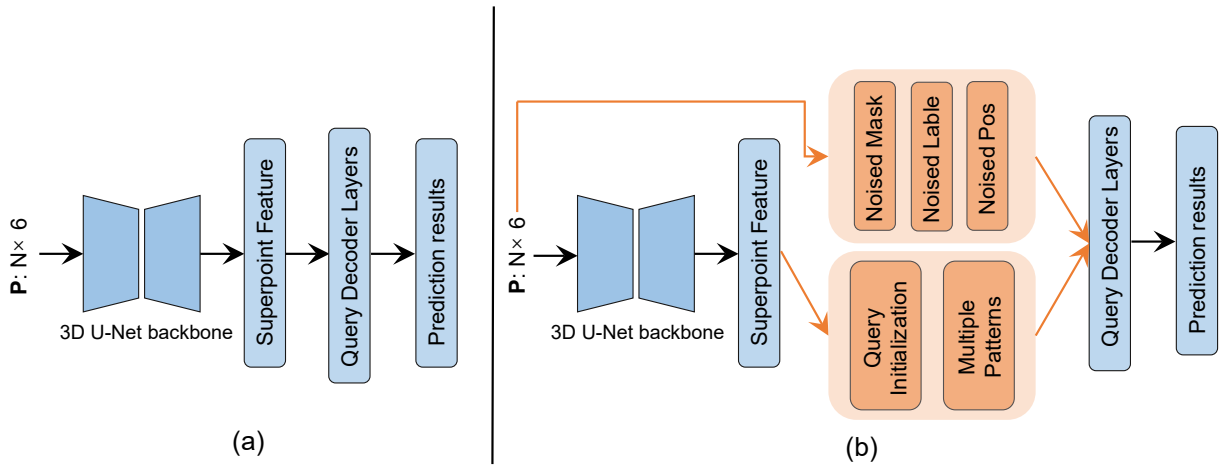

Fig. 3: Our framework is based on MAFT, (a) is the network architecture of MAFT and (b) is ours. The orange parts are our improvements.

is introduced in the same way but using a noise scale of $2\lambda_3$.

3) The third component is a self-attention mask $\mathcal{M}_{sa}$ for both denoising-part and matching-part queries. Since the denoising-part queries are derived from ground truth, we should ensure that no information is leaked from denoising-part queries to matching-part queries. Besides, it should also be ensured that different groups of denoising queries do not share information such that the reconstruction of one group is standalone. This can be achieved by introducing an attention mask in the self-attention module, which resembles the case where attention masks are used to mask future tokens to maintain causality in sequence generation tasks [41]. Assuming we have k groups denoising-part queries, and each group contains all the noised ground truth objects in the scene, so the total number of queries is $W = k \times 2n + N$, where n is the number of objects in the scene and N is the number of matching-part queries. Therefore, as shown in Fig. 2, if the first $k \times 2n$ rows and columns are denoising-part, $\mathcal{M}_{sa} = [m_{ij}]_{W\times W}$ is of size $W \times W$, and the entries of $\mathcal{M}_{sa}$ are given by

$$m_{ij} = \begin{cases} 1, \text{ if } j < k \times 2n \text{ and } \lfloor \frac{i}{2n} \rfloor \neq \lfloor \frac{j}{2n} \rfloor \\ 1, \text{ if } j < k \times 2n \text{ and } i \geq k \times 2n \\ 0, \text{ otherwise} \end{cases}, \quad (1)$$

where $m_{ij} = 1$ means the i-th query is masked from attending to the j-th query, while $m_{ij} = 0$ indicates no masking.

C. Multi-Pattern-Aware Query Selection Module

Query Initialization. Content queries in previous works [15, 14, 20] are learnable embeddings or setting as 0 vectors. These content queries have ambiguous physical meanings, which are refined by several layers of decoder to capture information in the scene. Since the final predicted masks are computed by multiplying the content queries with the superpoint features, it can be beneficial to use superpoint features as better priors to initialize content queries. Specifically, we propose three methods to initialize matching-part queries using superpoint features: 1) randomly select N queries; 2) use FPS algorithm to sample N queries based on their positions; 3) rank all the features based on the objectiveness score and select top-N queries. In addition, we can initialize both the positional and the content queries, or only one of them. The results are shown in Table V.

Multiple Patterns. The positional queries help DETR-like models to attend to specific positions, but multiple objects can exist at one position. Therefore, we include a multi-pattern design for the queries, intended to guide one positional query to operate in different modes so that different objects can be distinguished at one position. Specifically, the original content queries, $\mathcal{Q}_c^{ori} \in \mathbb{R}^{N_q \times d}$, has only one pattern. Inspired by [13], we incorporated shared learned pattern embeddings $\mathcal{Q}_{pat} \in \mathbb{R}^{N_p \times d}$ to predict multiple objects for one position, i.e.,

$$\mathcal{Q}_{pat} = \text{Embedding}(N_p, d), \quad (2)$$

where $N_p \geq 1$ is the number of patterns. To match the dimension, the content queries and the pattern embeddings are expended such that $\mathcal{Q}_c^{ori} \in \mathbb{R}^{N_q N_p \times d}$ and $\mathcal{Q}_{pat} \in \mathbb{R}^{N_q N_p \times d}$. The final content queries are the summation of the pattern embedding and the original content queries, which is given by

$$\mathcal{Q}_c = \mathcal{Q}_c^{ori} + \mathcal{Q}_{pat}. \quad (3)$$

The effective number of queries is $N_q \times N_p$; therefore, to fairly compare the effect of patterns, we need to ensure that the effective numbers of content queries are consistent.

D. Training and Inference

Bipartite matching is used in DETR-like models to establish unique predictions for each query and realize end-to-end training [8]. The key to bipartite matching is the

matching cost matrix $\mathcal{C}_{N\times n}$, where each entry $\mathcal{C}_{ij}$ evaluates the similarity between the i-th predicted instance and the j-th ground truth. The matching cost is given by

$$\mathcal{C}_{ij} = \lambda_{CE}\mathcal{L}_{CE} + \lambda_{Dice}\mathcal{L}_{Dice} + \lambda_{BCE}\mathcal{L}_{BCE} + \lambda_p\mathcal{L}_p, \quad (4)$$

where $\mathcal{L}_{CE}$ is the cross-entropy loss to supervise classification, $\mathcal{L}_{Dice}$ and $\mathcal{L}_{BCE}$ are the Dice loss and binary cross-entropy loss to supervise mask prediction, and $\mathcal{L}_p$ is the L-1 loss to supervise center position. The weights λ_{CE}, λ_{Dice}, λ_{BCE} and λ_p are set to 0.5, 1.0, 1.0, and 0.5, respectively. It should be noted that only the matching-part queries participate in bipartite matching, and the denoising-part queries make predictions and compute losses directly. After bipartite matching, the final loss can be computed by

$$\mathcal{L} = \mathcal{L}_{match} + \mathcal{L}_{dn}, \quad (5)$$

where $\mathcal{L}_{match} = \beta_1\mathcal{L}_{CE} + \beta_2\mathcal{L}_{Dice} + \beta_3\mathcal{L}_{BCE} + \beta_4\mathcal{L}_p + \beta_5\mathcal{L}_s$, $\mathcal{L}_s$ is the IoU-aware score loss [14], and $\mathcal{L}_{dn} = \gamma_1\mathcal{L}^{dn}_{CE} + \gamma_2\mathcal{L}^{dn}_{Dice} + \gamma_3\mathcal{L}^{dn}_{BCE} + \gamma_4\mathcal{L}^{dn}_p$. The coefficients are $\beta_1 = \beta_4 = \beta_5 = 0.5$, $\beta_2 = \beta_3 = 1.0$, $\gamma_1 = 1.3$, $\gamma_2 = \gamma_3 = \gamma_4 = 0.4$, respectively. In addition, we down-weight the classification loss of predictions for non-object by a factor of 10 to account for the class imbalance.

During inference, the contrastive mask denoising branch is inactive and the matching-parting queries predict N instances with labels, scores, and masks. These predictions are ranked by the scores to produce the final top-k instances, eliminating the need for post-processing steps and ensuring fast inference speed.

IV. Experiments

A. Experimental Settings

Dataset & Metrics

We evaluate our method using the challenging and large-scale indoor scene dataset ScanNetV2 and S3DIS. ScanNetV2 [42] is a representative and widely acknowledged public dataset, consisting of 25,000 scans from a variety of indoor settings. We train the model using 1202 training scenes provided by ScanNet and test the accuracy of the model using 312 validation scenes. S3DIS [43] dataset contains 271 rooms in 6 areas of three buildings; we evaluated our model on Area 5. Referring to the previous work, we adopt mean average precision (mAP) as the primary evaluation metric for instance segmentation performance. Specifically, we report mAP, AP_{50}, and AP_{25} scores on ScanNetV2 dataset and mAP, AP_{50} on S3DIS dataset.

Network Architecture

We adopt MAFT [20] as our baseline model, which uses a five-layer U-Net backbone and a six-layer transformer decoder, and introduce a contrastive mask denoising module and a multi-pattern-aware query selection module to optimize model performance. Fig. 3 provides an intuitive comparison of our model architecture against the baseline architecture. The left side of the figure illustrates the structure of the pipeline, and the right side illustrates our model structure. The components highlighted in orange indicate the modules we added.

TABLE I: Comparison on ScanNetv2 validation set

Method	mAP	AP_{50}	AP_{25}
3D-SIS [23]	/	18.7	35.7
PointGroup [4]	35.2	57.1	71.4
3D-MPA [21]	35.3	59.1	72.4
DyCo3D [44]	40.6	61.0	72.9
Mask-Group [6]	42.0	63.3	74.0
HAIS [24]	44.1	64.4	75.7
OccuSeg [45]	44.2	60.7	/
SoftGroup [26]	46.0	67.6	67.9
SSTNet [25]	49.4	64.3	74.0
Mask3D [15]	55.2	73.7	82.9
QueryFormer [33]	56.5	74.2	83.3
SPFormer [14]	56.3	73.9	82.9
MAFT [20]	57.9	74.7	84.0
Ours	**58.9**	**76.3**	**85.1**

TABLE II: 3D instance segmentation results on S3DIS Area5

Method	mAP	AP_{50}
SoftGroup [26]	51.6	66.1
SSTNet [25]	42.7	59.3
Mask3D [15]	56.6	68.4
QueryFormer [33]	57.7	69.9
SPFormer [14]	/	66.8
MAFT [20]	/	69.1
Ours	**58.3**	**70.1**

Implementation Details.

We carried out all the experiments on a single RTX 3090. Following previous works [20, 14], we fixed the length of the voxel to 0.02 meters and limited the number of points within a voxel to 250,000. We adopted the AdamW optimizer and a polynomial scheduler to train the model. For denoising-part hyperparameters, it is experimentally found that the best results are achieved when we set the scalar of the denoising module to 100, and the value of the weight coefficient γ_1, $\gamma_2, \gamma_3, \gamma_4$ mentioned in Eq.5 to 1.3, 0.4, 0.4, 0.4 respectively. During the experiment, we noticed that the results of some existing works show certain degrees of randomness. To ensure the accuracy of the experimental results, we repeated the training of these methods 10 times and took the average performance as the final comparison result.

B. Main Results

As shown in Table I and Table II, our method achieves state-of-the-art results for 3D point cloud instance segmentation on the validation set of ScanNetV2 and S3DIS Area 5, showing the generalization ability of our method. Our model performs well on all the metrics. Notably, our model achieves 58.9 on mAP, which is 1.0 percent higher than baseline [20] on ScanNetV2 validation set. Regarding AP_{50} and AP_{25}, our model also leads with scores of 76.3 and 85.1, with improvements of at least 1.6 and 1.1 percent compared to existing methods.

TABLE III: Ablation study of different modules

Contrastive Mask Denoising	Multi-Pattern-Aware Query Selection		mAP	AP_{50}	AP_{25}
	Query Init.	Multiple Patterns			
			57.9	74.7	84.0
✓			58.6	75.5	84.6
	✓		58.0	75.6	84.1
	✓	✓	58.2	75.7	84.3
✓	✓	✓	**58.9**	**76.3**	**85.1**

TABLE IV: Comparison of early-stage training performance

Method	Epochs	mAP	AP_{50}	AP_{25}
SPFormer	64	48.5	68.4	79
MAFT	64	50.3	68.5	77.9
Ours (without QS module)	64	**53.1**	**71.5**	**80.5**

C. Ablation Study

Ablation

Table III presents the results of comprehensive ablation experiments on ScanNetv2 validation set. The results show that contrastive mask denoising module and multi-pattern-aware query selection module can effectively improve the performance of the model. By only introducing the contrastive mask denoising module, the mAP of the model increases by 0.7. Besides, introducing the multi-pattern-aware query selection module can increase the mAP by 0.3 as indicated by the fourth row of Table III. These results demonstrate the effectiveness of the two modules respectively. The last row of the table shows that the model performs the best when these two modules are incorporated simultaneously, improving the mAP by +1.0. In addition, as shown in the third row of Table III, removing the multiple patterns in the query selection module can lead to a decrease in mAP by 0.2, which proves the effectiveness of multiple patterns.

Contrastive Mask Denoising Module

In Table IV, we add the contrastive mask denoising module to the pipeline for the comparative experiment. The experimental results show that the denoising module can help the model learn better results in early training stages and guide the model towards a more stable optimization path. At the 64-th epoch, our model improved over SPFormer [14] by +4.6 mAP and over MAFT [20] by +2.8 mAP, respectively.

TABLE V: Comparison of different query selection

Type	Method	mAP
Position	Fps	55.3
Position	Score	55.7
Position	Random	55.7
Position & Content	Random	57.6
Position & Content	Fps	57.7
Position & Content	Score	57.8
Content	Score	57.4
Content	Random	57.6
Content	Fps	**58.2**

TABLE VI: Comparison of number of patterns

Number of pattern	mAP
1	58.0
2	**58.2**
4	58.0
8	57.9

TABLE VII: Comparison of different pipelines

Method	mAP	AP_{50}	AP_{25}
SPFormer	56.3	73.9	82.9
MAFT	57.9	74.7	84.0
SPFormer + DN + QS	57.2	74.6	83.4
MAFT + DN + QS	58.9	76.3	85.1

Multi-Pattern-Aware Query Selection Module

Table V shows that different designs of the query selection (QS) module can impact the performance of the model. "Query selection type" indicates the queries being selected: position queries, content queries, or both. In addition, we adopt three methods for query selection: random selection, farthest point sampling, or ranking based on the scores.

In our experiments, we observe that schemes that only select the position queries are generally less effective. The performance is improved when both position and content queries are selected. This indicates that the features extracted by the backbone can provide richer prior information for query refinement, which improves the final performance of the model. Most notably, the best result of 58.2 is achieved when the query selection is only applied to content queries. This also proves the necessity of using query selection module to make full use of the backbone's feature information for query initialization. Additionally, we conduct experiments with different numbers of patterns, as shown in Table VI. The results indicate that the performance is optimal when the number of patterns is set to 2, while the effective number of queries is fixed at 400.

D. Analysis

The experiments from Table I to Table VI prove that our model can effectively improve performance. In addition, as a generalizable approach, we expect compatibility and performance improvement by applying it to other similar frameworks. To verify this, we choose SPFormer [14], which also uses the transformer architecture, as another pipeline to test the effectiveness of our method. Table VII shows that our method increases mAP by +0.9 for this new pipeline, which validates that our approach has compatibility and generalizability to transformer architectures, improving their performance. Many works focus on how to create a better model, but few aim to propose a general module that effectively enhances the performance of a certain type of method, which is a very meaningful endeavor.

Despite our model demonstrating superior segmentation capabilities, there are still limitations in certain aspects. Specifically, the performance of the model fluctuates with different batch sizes and number of queries. This indicates

that the model is sensitive to the batch size of training, and the dependence on the number of queries may limit the adaptability of the model. This is a universal problem in DETR-like models. Additionally, we achieved 58.2 mAP when fixing the total number of queries to 400 in our experiments with multiple patterns, and we observed that using more queries with multiple patterns can slightly benefit the final performance but occupy more memory space, which can hinder training on a single GPU while using contrastive mask denoising. These aspects reveal potential weaknesses in the stability and scalability of our model, pointing towards directions for future improvement. In response to these issues, future research could explore how to further improve the model's stability and spatio-temporal efficiency, aiming to achieve better performance across a wider range of application scenarios.

V. Conclusions

In this work, we propose a transformer-based 3D point cloud instance segmentation model using the contrastive mask denoising task and the multi-patterns-aware query selection module, to address the challenge of unstable bipartite matching in early training stages and unclear query physical meaning. The contrastive mask denoising task can guide the model towards a more stable optimization path in early training stages. Simultaneously, the multi-pattern-aware query selection module leverages backbone features, enabling the model to recognize multiple objects in the same region with greater precision. The experimental results prove that our method can effectively enhance the model's ability to understand complex scenes. Additionally, our approach is generalizable and can be integrated into most transformer-based architectures as a "plug and play" module to enhance their performance.

ACKNOWLEDGMENT

This work was partially supported by Key R&D Program of Zhejiang Province (No. 2023C01039).

References

[1] Ekim Yurtsever et al. "A survey of autonomous driving: Common practices and emerging technologies". In: *IEEE access* 8 (2020), pp. 58443–58469.

[2] Bo Yang et al. "Learning object bounding boxes for 3D instance segmentation on point clouds". In: *Advances in neural information processing systems* 32 (2019).

[3] Charles R Qi et al. "Deep hough voting for 3d object detection in point clouds". In: *proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 9277–9286.

[4] Li Jiang et al. "Pointgroup: Dual-set point grouping for 3d instance segmentation". In: *Proceedings of the IEEE/CVF conference on computer vision and Pattern recognition*. 2020, pp. 4867–4876.

[5] Weiyue Wang et al. "Sgpn: Similarity group proposal network for 3d point cloud instance segmentation". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 2569–2578.

[6] Min Zhong et al. "Maskgroup: Hierarchical point grouping and masking for 3d instance segmentation". In: *2022 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE. 2022, pp. 1–6.

[7] Meiling Gong et al. "A review of non-maximum suppression algorithms for deep learning target detection". In: *Seventh Symposium on Novel Photoelectronic Detection Technology and Applications*. Vol. 11763. SPIE. 2021, pp. 821–828.

[8] Nicolas Carion et al. "End-to-end object detection with transformers". In: *European conference on computer vision*. Springer. 2020, pp. 213–229.

[9] Xizhou Zhu et al. "Deformable detr: Deformable transformers for end-to-end object detection". In: *arXiv preprint arXiv:2010.04159* (2020).

[10] Feng Li et al. "Dn-detr: Accelerate detr training by introducing query denoising". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 13619–13627.

[11] Peng Gao et al. "Fast convergence of detr with spatially modulated co-attention". In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2021, pp. 3621–3630.

[12] Shilong Liu et al. "Dab-detr: Dynamic anchor boxes are better queries for detr". In: *arXiv preprint arXiv:2201.12329* (2022).

[13] Yingming Wang et al. "Anchor detr: Query design for transformer-based detector". In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 36. 3. 2022, pp. 2567–2575.

[14] Jiahao Sun et al. "Superpoint transformer for 3d scene instance segmentation". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 37. 2. 2023, pp. 2393–2401.

[15] Jonas Schult et al. "Mask3d: Mask transformer for 3d semantic instance segmentation". In: *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2023, pp. 8216–8223.

[16] Harold W Kuhn. "The Hungarian method for the assignment problem". In: *Naval research logistics quarterly* 2.1-2 (1955), pp. 83–97.

[17] Francesc Serratosa. "Fast computation of bipartite graph matching". In: *Pattern Recognition Letters* 45 (2014), pp. 244–250.

[18] Hao Zhang et al. "Dino: Detr with improved denoising anchor boxes for end-to-end object detection". In: *arXiv preprint arXiv:2203.03605* (2022).

[19] Depu Meng et al. "Conditional detr for fast training convergence". In: *Proceedings of the IEEE/CVF*

International Conference on Computer Vision. 2021, pp. 3651–3660.

[20] Xin Lai et al. "Mask-attention-free transformer for 3d instance segmentation". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2023, pp. 3693–3703.

[21] Francis Engelmann et al. "3d-mpa: Multi-proposal aggregation for 3d semantic instance segmentation". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 9031–9040.

[22] Shih-Hung Liu et al. "Learning gaussian instance segmentation in point clouds". In: *arXiv preprint arXiv:2007.09860* (2020).

[23] 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*. 2019, pp. 4421–4430.

[24] Shaoyu Chen et al. "Hierarchical aggregation for 3d instance segmentation". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 15467–15476.

[25] Zhihao Liang et al. "Instance segmentation in 3D scenes using semantic superpoint tree networks". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 2783–2792.

[26] Thang Vu et al. "Softgroup for 3d instance segmentation on point clouds". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 2708–2717.

[27] Maksim Kolodiazhnyi et al. "Top-down beats bottom-up in 3d instance segmentation". In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 2024, pp. 3566–3574.

[28] Alexander Neubeck and Luc Van Gool. "Efficient non-maximum suppression". In: *18th international conference on pattern recognition (ICPR'06)*. Vol. 3. IEEE. 2006, pp. 850–855.

[29] Ze Liu et al. "Group-free 3d object detection via transformers". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 2949–2958.

[30] Hengshuang Zhao et al. "Point transformer". In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2021, pp. 16259–16268.

[31] Meng-Hao Guo et al. "Pct: Point cloud transformer". In: *Computational Visual Media* 7 (2021), pp. 187–199.

[32] Bowen Cheng et al. "Masked-attention mask transformer for universal image segmentation". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2022, pp. 1290–1299.

[33] Jiahao Lu et al. "Query refinement transformer for 3d instance segmentation". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2023, pp. 18516–18526.

[34] Ashish Vaswani et al. "Attention is all you need". In: *Advances in neural information processing systems* 30 (2017).

[35] Alexey Dosovitskiy et al. "An image is worth 16x16 words: Transformers for image recognition at scale". In: *arXiv preprint arXiv:2010.11929* (2020).

[36] Ze Liu et al. "Swin transformer: Hierarchical vision transformer using shifted windows". In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2021, pp. 10012–10022.

[37] Ze Liu et al. "Swin transformer v2: Scaling up capacity and resolution". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2022, pp. 12009–12019.

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

[39] Jitesh Jain et al. "Oneformer: One transformer to rule universal image segmentation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 2989–2998.

[40] Feng Li et al. "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*. 2023, pp. 3041–3050.

[41] Rundi Wu et al. "Pq-net: A generative part seq2seq network for 3d shapes". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 829–838.

[42] Angela Dai et al. "Scannet: Richly-annotated 3d reconstructions of indoor scenes". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 5828–5839.

[43] Iro Armeni et al. "3D Semantic Parsing of Large-Scale Indoor Spaces". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016.

[44] 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*. 2021, pp. 354–363.

[45] Lei Han et al. "Occuseg: Occupancy-aware 3d instance segmentation". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 2940–2949.