

IPDN: Image-enhanced Prompt Decoding Network for 3D Referring Expression Segmentation

Qi Chen^{*1}, Changli Wu^{*1}, Jiayi Ji^{†1 2}, Yiwei Ma¹, Danni Yang¹, Xiaoshuai Sun¹

¹Key Laboratory of Multimedia Trusted Perception and Efficient Computing, Ministry of Education of China, Xiamen University, 361005, P.R. China.

²National University of Singapore.

chenqi@stu.xmu.edu.cn, wuchangli@stu.xmu.edu.cn, jjyxmu@gmail.com, mayiwei@stu.xmu.edu.cn, yangdanni@stu.xmu.edu.cn, xssun@xmu.edu.cn

Abstract

3D Referring Expression Segmentation (3D-RES) aims to segment point cloud scenes based on a given expression. However, existing 3D-RES approaches face two major challenges: feature ambiguity and intent ambiguity. Feature ambiguity arises from information loss or distortion during point cloud acquisition due to limitations such as lighting and viewpoint. Intent ambiguity refers to the model’s equal treatment of all queries during the decoding process, lacking top-down task-specific guidance. In this paper, we introduce an Image-enhanced Prompt Decoding Network (IPDN), which leverages multi-view images and task-driven information to enhance the model’s reasoning capabilities. To address feature ambiguity, we propose the Multi-view Semantic Embedding (MSE) module, which injects multi-view 2D image information into the 3D scene and compensates for potential spatial information loss. To tackle intent ambiguity, we designed a Prompt-Aware Decoder (PAD) that guides the decoding process by deriving task-driven signals from the interaction between the expression and visual features. Comprehensive experiments demonstrate that IPDN outperforms the state-of-the-art by 1.9 and 4.2 points in mIoU metrics on the 3D-RES and 3D-GRES tasks, respectively.

Code — <https://github.com/80chen86/IPDN>

1 Introduction

3D Referring Expression Segmentation (3D-RES) presents significant potential applications in areas such as virtual reality, augmented reality, robotics navigation, and human-computer interaction. The goal of this task is to segment the object pointed to by a given textual description from a point cloud scene (Huang et al. 2021; Wu et al. 2024b).

The earliest approaches (Huang et al. 2021) to 3D-RES employed a two-stage paradigm: first, they used an instance segmentation network to generate proposals, and subsequently matched these proposals with the text to compute matching scores, leading to the final segmentation result. However, this methodology was found lacking in both efficiency and effectiveness (Wu et al. 2024b). Consequently,

^{*}These authors contributed equally.

[†]Corresponding author

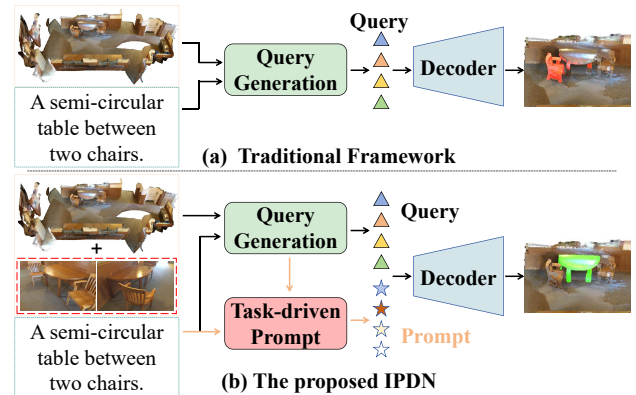


Figure 1: The pipeline of (a) the previous traditional query-based framework and (b) our method.

recent studies (Wu et al. 2024b; He and Ding 2024; He et al. 2024; Wu et al. 2024a) have pivoted toward adopting a one-stage query-based paradigm. For example, 3D-STMN (Wu et al. 2024b) achieves efficient segmentation by directly matching text with superpoints, while MCLN (Qian et al. 2024b) and some other works (He et al. 2024; Lin et al. 2023; Xu et al. 2024) enhance performance by coupling the 3D-RES task with other tasks for joint multi-task training.

However, despite the promising results these methods have achieved, they still come with certain limitations: (1) **Feature ambiguity**: Existing approaches rely solely on point cloud data for visual information extraction. However, point cloud data often suffers from information loss due to factors such as lighting, viewing angles, and sampling rates during collection, making it challenging to reproduce real-world scenes faithfully. Consequently, extracting high-quality features exclusively from point cloud data becomes difficult. Compared to 2D data, acquiring and annotating 3D data is far more challenging (Dong et al. 2022), limiting the rapid advancements seen in large-scale vision-language pretraining for 2D domains (Fei et al. 2024c,a,b). Therefore, purely visual 3D backbones (Qi et al. 2017a,b; Graham, Engelcke, and Van Der Maaten 2018; Deng et al. 2021; Shi et al. 2020; Yang et al. 2023; Zhao et al. 2021a; Lai et al. 2022) struggle to align extracted features with tex-

tual representations. (2) **Intent ambiguity**: For all queries, they are treated with equal importance, similar to purely visual 3D segmentation (Kolodiazhnyi et al. 2024; Sun et al. 2023; Lai et al. 2023; Schult et al. 2023; Lu et al. 2023a). However, in 3D-RES, only the target object described in the text needs to be segmented. Ideally, queries relevant to the text should be prioritized. Yet, current methods (Wu et al. 2024b,a; He and Ding 2024) do not highlight these relevant queries, leading to the model having to implicitly learn the distinction between relevant and irrelevant queries, significantly increasing the difficulty of the learning process.

To address the above issues, we introduce the Image-enhanced Prompt Decoding Network (IPDN), which leverages multi-view images and task-driven information in a top-down approach to unleash the model’s reasoning capabilities. As shown in Fig. 1, to tackle the feature ambiguity issue, we propose the Multi-view Semantic Embedding (MSE) strategy. MSE employs CLIP (Radford et al. 2021) to extract 2D image features, which are then fused with 3D point cloud features to significantly enhance visual representation. Additionally, Spatial-aware Attention is incorporated to address the absence of spatial positional relationships in 2D features. This approach results in visual features with superior representational power, enriched with text prior knowledge from CLIP, facilitating better alignment with textual features. To address the intent ambiguity issue, we designed a Prompt-aware Decoder (PAD) that guides the decoding process using task-driven signals. Through the Task-driven Prompt module, we generate prompts that emphasize the relevance of each query to the text, effectively injecting task-specific information into the model and significantly reducing the learning complexity. Extensive qualitative and quantitative experiments on the ScanRefer (Chen, Chang, and Nießner 2020) and Multi3DRefer (Zhang, Gong, and Chang 2023) datasets validate the superior performance of IPDN, surpassing the current state-of-the-art (SOTA) by 1.9 and 4.2 points in mIoU metrics on the 3D-RES and Generalized 3D Referring Expression Segmentation (3D-GRES) tasks, respectively.

To sum up, our main contributions are as follows:

- We identify two critical challenges in the 3D-RES task, *i.e.*, feature ambiguity and intent ambiguity, and propose a novel method, IPDN, to effectively address them.
- IPDN comprises two essential modules, *i.e.*, MSE and PAD. The MSE integrates multi-view image information into 3D representations while restoring spatial information lost. The PAD pre-processes task-related signals to guide the decoding process with greater precision.
- Extensive experiments show that our IPDN outperforms existing state-of-the-art methods, delivering significant improvements in both 3D-RES and 3D-GRES tasks.

2 Related Work

2.1 3D Referring Expression Comprehension

3D Referring Expression Comprehension (3D-REC) task is to predict a bounding box for objects indicated by text. Existing approaches to the 3D-REC task can largely be categorized into two types: two-stage (Chen et al. 2022; Feng et al.

2021; He et al. 2021; Yuan et al. 2021; Zhao et al. 2021b; Yang et al. 2024; Roh et al. 2022; Yang et al. 2021; Wu, Huang, and Wang 2024; Zhang et al. 2023) and one-stage (Luo et al. 2022; Wang et al. 2023). Two-stage methods first employ detection models to generate proposals, then use a series of strategies to semantically match the text with these proposals in order to identify the target object.

2.2 3D Referring Expression Segmentation

Unlike the relatively mature studies in 3D-REC and 2D-RES (Shah, VS, and Patel 2024; Jing et al. 2021; Ding et al. 2021; Yang et al. 2022; Wang et al. 2022a; Liu, Ding, and Jiang 2023; Lai et al. 2024; Chng et al. 2024), 3D-RES (Qian et al. 2024a; He and Ding 2024; He et al. 2024; Lin et al. 2023; Xu et al. 2024) is still in its infancy. As the pioneering work in this domain, TGNN (Huang et al. 2021) adopted a two-stage strategy, leveraging Graph Neural Networks for matching candidate instances with textual descriptions. 3D-STMN (Wu et al. 2024b) harnessed a one-stage method, significantly enhancing both inference speed and performance. Other approaches, such as MCLN (Qian et al. 2024b), capitalized on the similarity between 3D-RES and 3D-REC tasks to facilitate multitask joint learning.

To liberate the 3D-RES task from its constraint of having one and only one target object per sentence, the 3D-GRES task was introduced (Wu et al. 2024a). A distinctive feature of 3D-GRES is that the object referenced by the text may not exist or could be multiple objects, no longer restricted to a single object.

2.3 Prompt Learning

Prompt learning generally refers to the augmentation of models with specific prompt information. These prompts can be hand-crafted or automatically learned during the training process. Initially applied in the field of Natural Language Processing (NLP) (Lester, Al-Rfou, and Constant 2021; Li and Liang 2021; Liu et al. 2021), prompt learning has since been adapted for use in visual (Jia et al. 2022; Wang et al. 2022b; Zhang, Zhou, and Liu 2022) and vision-language (Zhou et al. 2022a,b; Zhu et al. 2023) models as well. In our model, we utilize a set of prompts generated under textual guidance to instruct the model in differentiating between relevant queries and irrelevant ones.

3 Method

In this section, we first introduce the inputs to the decoder, namely how visual features, textual features, and queries are obtained (Sec. 3.1). Secondly, we detail the Multi-view Semantic Embedding (MSE) strategy (Sec. 3.2). Then we describe the Prompt-aware Decoder (Sec. 3.3). Finally, we outline the loss function for the entire model (Sec. 3.4). An overview of our framework is shown in Fig. 2.

3.1 Feature Extraction

Textual Feature Given a textual description for the target object, we utilize a pre-trained RoBERTa (Liu et al. 2019) to extract word-level embeddings $E \in \mathbb{R}^{N_t \times C_t}$, where N_t

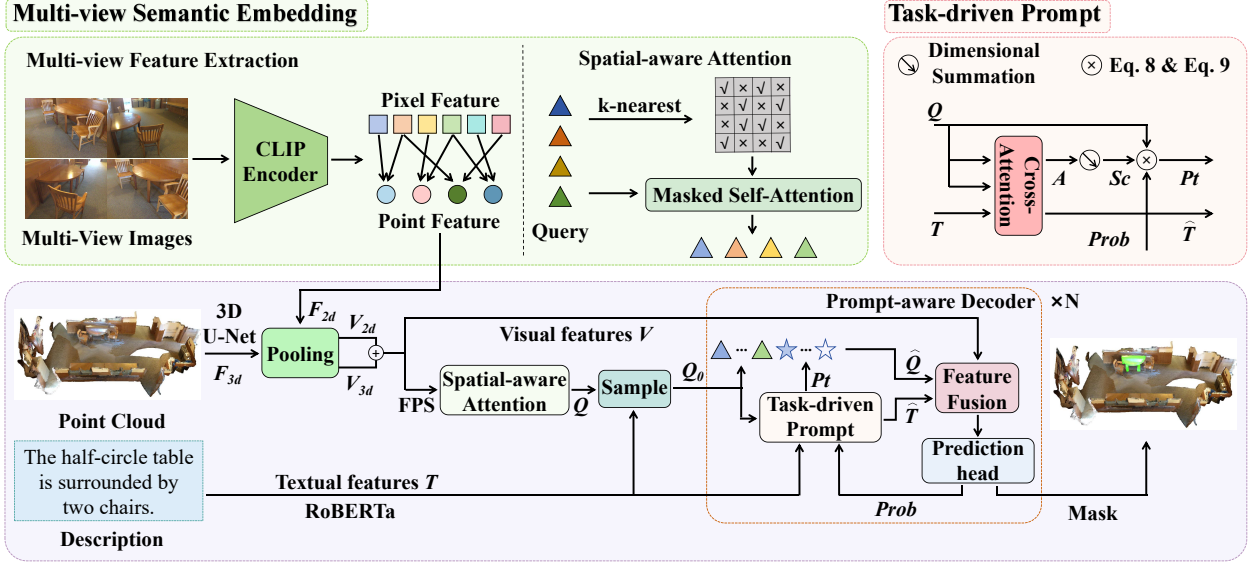


Figure 2: The overview of our framework.

denotes the number of tokens, and C_t indicates the C_t -dimensionality of each embedding. In order to have a unified feature dimension d in the decoder, we transform E into textual features $T \in \mathbb{R}^{N_t \times d}$ via a linear projection:

$$T = EW_t, \quad (1)$$

where $W_t \in \mathbb{R}^{C_t \times d}$ are learnable parameters.

Visual Feature Given a point cloud scene $P \in \mathbb{R}^{N_p \times (3+f)}$, where N_p denotes the number of points. Each point carries a 3D coordinate as well as an auxiliary feature vector of f dimensions, such as RGB values and normal vectors. We first employ a Sparse 3D U-Net (Graham, Engelcke, and Van Der Maaten 2018) to extract point-wise features $F_{3d} \in \mathbb{R}^{N_p \times C_p}$, where C_p represents the feature dimensionality. Subsequently, following the approach of (Sun et al. 2023), we generate N_s superpoints $\{SP_i\}_{i=1}^{N_s}$ (Landrieu and Simonovsky 2018) from the original point cloud and perform superpoint pooling on F_{3d} to obtain 3D superpoint features $S_{3d} \in \mathbb{R}^{N_s \times C_p}$. Then, a multi-layer perceptron (MLP) is utilized to transform the dimensionality to d , yielding the 3D visual features $V_{3d} \in \mathbb{R}^{N_s \times d}$:

$$V_{3d} = \text{MLP}(\text{SPPool}(F_{3d})), \quad (2)$$

where $\text{MLP}(\cdot)$ is a learnable multi-layer perceptron, and $\text{SPPool}(\cdot)$ is superpoint pooling operation. The final visual feature $V \in \mathbb{R}^{N_s \times d}$ is obtained by the sum of V_{3d} and V_{2d} (introduced in Sec. 3.2).

Sparse Query Generation After obtaining the visual features V and textual features T , our next step is to utilize both to generate the queries that will be used in the decoder. Specifically, We first perform farthest point sampling (Moening and Dodgson 2003) on the superpoints (correspond one-to-one with the visual features), followed by spatial-aware attention (introduced in Sec. 3.2), then resample the results using the sampling module from MDIN (Wu

et al. 2024a), and generate the queries through an MLP:

$$Q_{seed} = V[\text{FPS}(p_{sp})], \quad (3)$$

$$Q_0 = \text{MLP}(\text{Sample}(\text{SPA}(Q_{seed}), T)), \quad (4)$$

where $\text{FPS}(\cdot)$, $\text{Sample}(\cdot)$ and $\text{SPA}(\cdot)$ denote the Farthest Point Sampling algorithm, the sampling module in MDIN and the spatial-aware attention respectively, $[\cdot]$ denotes accessing elements by the index within it, $p_{sp} \in \mathbb{R}^{N_s \times 3}$ represents the coordinates of the superpoints, $Q_{seed} \in \mathbb{R}^{2m \times d}$ is the seed query, and $Q_0 \in \mathbb{R}^{m \times d}$ ($m \ll N_s$) is initial query.

3.2 Multi-view Semantic Embedding

Multi-view Feature Extraction The 3D features extracted solely from point cloud data are limited in representational capacity due to the information loss of point clouds and insufficient alignment with the language modality. To address this issue, we propose a Multi-View Semantic Embedding (MSE) strategy. This approach enhances the visual features by extracting well-aligned multi-view semantics and injecting them back into the original 3D features through 2D-3D projection.

Specifically, given N_I images $\{I_i\}_{i=1}^{N_I}$ of the point cloud scene from different perspectives, we first extract patch-level 2D features using the CLIP (Radford et al. 2021) visual encoder, which is pre-aligned with visual-language tasks. To accommodate camera parameters, we upsample these features to the original image resolution via interpolation, resulting in pixel-level 2D features $\{F_i^{img} \in \mathbb{R}^{H \times W \times C_I}\}_{i=1}^{N_I}$, where C_I denotes the feature dimension, and H and W represent the height and width of the image, respectively. Next, we project the 2D pixel coordinates into the 3D point cloud space using the camera parameters. Similar to previous works (Wang et al. 2024; Yu et al. 2024; Peng et al. 2023; Zhang, Dong, and Ma 2023), for a pixel coordinate (u, v) , given the intrinsic camera parameters $\mathcal{K} \in \mathbb{R}^{3 \times 3}$,

extrinsic parameters $\mathcal{R} \in \mathbb{R}^{3 \times 3}$ and $\mathcal{T} \in \mathbb{R}^{3 \times 1}$, and depth $\mathcal{D} \in \mathbb{R}$, we obtain the corresponding 3D coordinates (x, y, z) through 2D-3D projection:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \text{Project}(u, v) = \mathcal{R}(\mathcal{K}^{-1} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \cdot \mathcal{D}) + \mathcal{T}. \quad (5)$$

After the projection, all 2D pixel features are assigned 3D coordinates $p_{3d} \in \mathbb{R}^{HWN_I \times 3}$. To inject these multi-view features into the point cloud, we apply spherical querying to p_{3d} in the point cloud scene. This technique assigns each pixel feature to the points within a sphere centered at its 3D coordinate, thus embedding multi-view semantic information. For points residing in multiple spheres, the final multi-view feature is computed as the average of the pixel features associated with that point. In this way, we obtain the multi-view semantic features $F_{2d} \in \mathbb{R}^{N_p \times C_I}$ for all points, which are then processed similarly to F_{3d} (Eq. 2) to derive the 2D visual features $V_{2d} \in \mathbb{R}^{N_s \times d}$. Finally, we get the visual feature $V \in \mathbb{R}^{N_s \times d}$ by summing the V_{3d} and V_{2d} .

Spatial-aware Attention While incorporating multi-view semantics improves visual representation and visual-language alignment, it also introduces limitations inherent to 2D images, such as the absence of spatial positional information and potential multi-view conflicts. Specifically, each image has a restricted field of view and lacks depth information, complicating the determination of 3D object positions and inter-object distances. To mitigate these issues, we use a spatial-aware attention mechanism to incorporate explicit 3D spatial relationships, enhancing spatial positioning. Additionally, due to the high computational cost and inefficiency of operating directly at the superpoint level, we implement efficient spatial-aware attention on the sparse seed query Q_{seed} , which is more manageable on our GPU.

First, we construct a k -nearest neighbor matrix $M \in \mathbb{R}^{2m \times 2m}$, where the element M_{ij} in the i^{th} row and j^{th} column indicates whether the j^{th} query is among the k nearest queries to the i^{th} query. If the j^{th} query is within the k nearest neighbors of the i^{th} query, M_{ij} is set to True; otherwise, it is set to False. The coordinates of the queries are obtained from the corresponding superpoint coordinates. Then, we use M as a mask to perform self-attention on the seed queries Q_{seed} , producing the output $Q \in \mathbb{R}^{2m \times d}$ as the input to the sample module:

$$Q = \text{SPA}(Q_{seed}) = \text{Masked_Self}(Q_{seed}, M), \quad (6)$$

where $\text{Masked_Self}(\cdot)$ denotes the masked self-attention.

3.3 Prompt-aware Decoder

Previous query-based methods (Wu et al. 2024b; He and Ding 2024; Qian et al. 2024b) inherit the instance segmentation approach (Sun et al. 2023; Lai et al. 2023; Schult et al. 2023) to handling queries, which does not distinguish the importance of different queries. However, this approach is not well-suited for the 3D-RES task, which aims to segment objects indicated by text rather than all objects. This means that queries related to the text should be prioritized. To help the model better differentiate the importance of queries

and reduce the learning difficulty, we introduce task-driven prompt learning in the decoder. By dynamically generating a set of text-relevant prompts, these prompts guide the model during the decoding process to identify which queries are more important and more likely to correspond to the target object.

Task-driven Prompt To design reliable prompts tailored for 3D-RES task, we first measure the relevance between the text and queries using cross-attention scores. Specifically, we perform a cross-attention operation by using the text features T as the query and the Sparse Queries Q_l from the l^{th} layer as the keys and values within the attention mechanism:

$$\hat{T}_l, A_l = \text{Cross}(T, Q_l, Q_l), \quad (7)$$

where $\hat{T}_l \in \mathbb{R}^{N_t \times d}$, $A_l \in \mathbb{R}^{N_t \times m}$, and $Q_l \in \mathbb{R}^{m \times d}$ denote the text features, attention scores, and sparse queries at the l^{th} layer, respectively. $\text{Cross}(\cdot)$ denotes the cross-attention operation (Vaswani et al. 2017). Then, by summing the attention scores A_l across the first dimension, we initially obtain the relevance scores $S_{c_l} \in \mathbb{R}^m$ indicating how closely each query is associated with the given textual description.

After obtaining the scores S_{c_l} , a intuitive approach would be to directly apply the Softmax function to determine the desired relevance. However, most queries are irrelevant to the text description, and their scores essentially act as noise, which should be minimized. To address this, we introduce a threshold filtering operation to filter out irrelevant queries as much as possible, making the prompts more reliable.

Specifically, we utilize the probability $Prob_{l-1} \in \mathbb{R}^m$, generated by the prediction head of the upper layer queries, to filter out queries. This probability represents the likelihood that a query corresponds to the target instance. For queries with probabilities below the threshold r , their relevance scores are set to negative infinity, meaning their values will be 0 after applying the Softmax function. Finally, the Softmax function is applied to the relevance scores, and the results are multiplied by the queries, producing prompts that guide the model in distinguishing between relevant and irrelevant queries. The process can be formulated as follows:

$$\hat{S}_{c_l}^j = \begin{cases} -\infty, & Prob_{l-1}^j < r \\ S_{c_l}^j, & Prob_{l-1}^j \geq r \end{cases}, \quad (8)$$

$$Pt_l = Q_l \cdot \text{Softmax}(\hat{S}_{c_l}), \quad (9)$$

$$\hat{Q}_l = \text{Concat}(Q_l, Pt_l), \quad (10)$$

where r is a hyperparameter, j denotes the j^{th} element, $Pt_l \in \mathbb{R}^{m \times d}$ represents the prompts, $\text{Concat}(\cdot)$ denotes the Concatenation operation, $\hat{Q}_l \in \mathbb{R}^{2m \times d}$ stands for the queries with the prompts attached, and all subscripts l indicate the l^{th} layer.

Feature Fusion & Prediction Head We follow the feature fusion method outlined in MDIN (Wu et al. 2024a), utilizing the query \hat{Q}_l to integrate the textual features \hat{T}_l and visual features V , thereby updating the queries under the guidance

of the prompts. The specific formula is presented as follows:

$$\begin{aligned} \bar{Q}_l = & \text{Abandon}(\text{Cross}(\hat{Q}_l, \hat{T}_l, \hat{T}_l) \\ & + \text{Self}(\hat{Q}_l) \\ & + \text{Cross}(\hat{Q}_l, V, V)), \end{aligned} \quad (11)$$

where $\text{Self}(\cdot)$ denotes the self-attention operation, $\text{Abandon}(\cdot)$ denotes the discarding of the prompts, and $\bar{Q}_l \in \mathbb{R}^{m \times d}$ represents the updated queries. Considering the difference in scale between the queries and superpoints, we apply Spatial-aware Attention once again at the end of each layer for feature enhancement and to generate the query Q_{l+1} for the next layer:

$$Q_{l+1} = \text{SPA}(\bar{Q}_l). \quad (12)$$

Before going to the next layer, Q_{l+1} will pass through a prediction head to generate $Mask_l$ and $Prob_l$ in l^{th} layer:

$$Mask_l = Q_{l+1}(VW_{mask})^T, \quad (13)$$

$$Prob_l = Q_{l+1}W_{prob}, \quad (14)$$

where $W_{mask} \in \mathbb{R}^{d \times d}$ and $W_{prob} \in \mathbb{R}^{d \times 1}$ are learnable parameters, superscript T indicates matrix transpose, $Mask_l \in \mathbb{R}^{m \times N_s}$ represents the predicted masks for every query and $Prob_l \in \mathbb{R}^m$ represents the likelihood that a query corresponds to the target instance.

Following MDIN (Wu et al. 2024a), we select the query with the highest $Prob$ value and binarize its corresponding mask to generate the prediction during inference in the 3D-RES task. In the 3D-GRES task, we merge the binary masks of all queries with $Prob$ values greater than 0.5 to produce the final prediction.

3.4 Loss

The loss of our method primarily consists of three components. The first component is the basic loss \mathcal{L}_b , which is applied only on the queries corresponding to the target instance (Wu et al. 2024a):

$$\mathcal{L}_b = \text{BCE}(M^+, M^{tgt}) + \text{DICE}(M^+, M^{tgt}), \quad (15)$$

where M^+ represents the mask output by the prediction head for the query corresponding to the target instance, M^{tgt} is the ground truth mask for the target instance, $\text{BCE}(\cdot)$ denotes the Binary Cross-Entropy loss, and $\text{DICE}(\cdot)$ refers to the Dice loss (Milletari, Navab, and Ahmadi 2016).

The second part is the probability loss \mathcal{L}_p , which is used to supervise $Prob$, that is, the probability associated with the query corresponding to the target instance:

$$\mathcal{L}_p = \text{BCE}(Prob, L^{tgt}), \quad (16)$$

where the label $L^{tgt} \in \{0, 1\}^m$ indicates whether the query corresponds to the target instance, with 1 representing a positive match and 0 representing a negative match, and $Prob$ is the probability output by the prediction head.

The third part is the contrastive learning loss \mathcal{L}_c , which is used to align the text features with their corresponding queries. Here, we adopt the approach used in EDA (Wu et al. 2023).

The final loss \mathcal{L} is calculated as the weighted sum of \mathcal{L}_b , \mathcal{L}_p and \mathcal{L}_c :

$$\mathcal{L} = \lambda_b \mathcal{L}_b + \lambda_p \mathcal{L}_p + \lambda_c \mathcal{L}_c, \quad (17)$$

where λ_b , λ_p and λ_c are hyperparameters.

4 Experiments

4.1 Implementation details

In our experiments, we apply the PolyRL strategy to adjust the learning rate starting from 0.0001, with a decay power of 4.0. The batch size is set to 16. The number of queries m is set to 128. The decoder consists of 6 layers. The hyperparameter k in sec.3.2 is set to 8, and the hyperparameter r in sec. 3.3 is 0.75. In the loss function, the weights λ_b , λ_p , and λ_c are set to 1.0, 0.1, and 0.1 respectively. All experiments are conducted using the PyTorch framework on an NVIDIA GeForce RTX 3090 GPU.

4.2 Dataset and Evaluation Metrics

ScanRefer We utilize the ScanRefer dataset (Chen, Chang, and Nießner 2020) to evaluate our method, which consists of 51,583 natural language expressions, encompassing 11,046 objects across 800 ScanNet (Dai et al. 2017) scenes. The evaluation metrics include mean Intersection over Union (mIoU), Acc@0.25, and Acc@0.5.

Multi3DRefer We use the Multi3DRefer (Zhang, Gong, and Chang 2023) dataset to evaluate our model’s performance on the 3D-GRES task, which differs from 3D-RES in that the number of targets referenced by the text can be arbitrary. The dataset consists of a total of 61926 language descriptions, of which 51583 are directly obtained from ScanRefer. Among these, 6688 descriptions match zero targets, 13178 match multiple targets, and the rest match a single target. The evaluation metric is the same as that used in ScanRefer. When the text refers to zero object, the sample’s mIoU is 1 if the model correctly identifies this, otherwise, it is 0.

4.3 Quantitative Comparison

As shown in Tab. 1, our model significantly outperforms the existing SOTA methods on the 3D-GRES task, achieving an improvement of 4.2 points in mIoU and even 5.3 points in Acc@0.5. It can be observed that, apart from the zero target scenario with distractors where our model performs below, it significantly surpasses the MDIN (Wu et al. 2024a) in all other cases. Especially in single-target scenarios, without distractors, our model outperforms the MDIN by nearly eight points on the Acc@0.5 metric. This demonstrates that our task-driven prompt effectively guides the model to focus on more significant queries, thereby enabling more accurate localization of key targets.

We also conducted experiments on the traditional 3D-RES task, as shown in Table 2. On the ScanRefer dataset, our proposed IPDN model achieved state-of-the-art performance overall. Specifically, our model outperformed the previous best model, MDIN (Wu et al. 2024a), by 2.6, 1.8, and 1.9 points in terms of Acc@0.25, Acc@0.5, and mIoU, respectively. Notably, we observed more significant improvements in challenging scenes with multiple distracting objects. This indicates that our model benefits from more robust multi-view semantic integration and the task-driven prompt, which effectively guides the model to focus on more critical information, thereby enhancing its discriminative ability to accurately identify the target object among multiple instances of the same category.

Method	Acc@0.25				Acc@0.5				mIoU
	ZT	ST	MT	All	ZT	ST	MT	All	
ReLA (Liu, Ding, and Jiang 2023)	36.2 / 72.7	48.3 / 83.4	73.0	61.8	36.2 / 72.7	20.4 / 65.5	42.4	37.4	42.8
M3DRef-CLIP (Zhang, Gong, and Chang 2023)	39.2 / 81.6	50.8 / 77.5	66.8	55.7	39.2 / 81.6	29.4 / 67.4	41.0	37.5	37.4
3D-STMN (Wu et al. 2024b)	42.6 / 76.2	49.0 / 77.8	68.8	60.4	42.6 / 76.2	24.6 / 69.2	43.9	40.9	43.0
MDIN (Wu et al. 2024a)	47.9 / 78.8	55.5 / 84.4	76.3	67.0	47.9 / 78.8	29.5 / 71.7	46.8	44.7	47.5
IPDN (Ours)	39.4 / 84.1	61.5 / 88.9	79.6	71.5	39.4 / 84.1	34.7 / 79.5	52.1	50.0	51.7

Table 1: The 3D-GRES results on Multi3DRefer. ZT, ST, and MT represent zero target, single target, and multiple targets, respectively. The left and right sides of the “/” represent the situations with and without distractor objects, respectively.

Method	Venue	Unique (~19%)			Multiple (~81%)			Overall		
		0.25	0.5	mIoU	0.25	0.5	mIoU	0.25	0.5	mIoU
TGNN† (Huang et al. 2021)	AAAI2021	69.3	57.8	50.7	31.2	26.6	23.6	38.6	32.7	28.8
InstanceRefer† (Yuan et al. 2021)	ICCV2021	81.6	72.2	60.4	29.4	23.5	21.5	40.2	33.5	30.6
3DRefTR (Lin et al. 2023)	Arxiv	89.6	77.0	-	52.3	43.7	-	57.9	48.7	41.2
X-RefSeg3D (Qian et al. 2024a)	AAAI2024	-	-	-	-	-	-	40.3	33.8	29.9
3D-STMN (Wu et al. 2024b)	AAAI2024	89.3	84.0	74.5	46.2	29.2	31.1	54.6	39.8	39.5
Reanson3D (Huang et al. 2024)	Arxiv	88.4	84.2	74.6	50.5	31.7	34.1	57.9	41.9	42.0
SegPoint (He et al. 2024)	ECCV2024	-	-	-	-	-	-	-	-	41.7
MCLN (Qian et al. 2024b)	ECCV2024	89.6	78.2	-	53.3	45.9	-	58.7	50.7	44.7
RefMask3D (He and Ding 2024)	ACMMM2024	89.6	84.7	-	48.1	40.8	-	55.9	49.2	44.9
MDIN (Wu et al. 2024a)	ACMMM2024	91.0	87.2	76.7	50.1	44.9	41.4	58.0	53.1	48.3
IPDN (Ours)	-	91.5	88.0	77.9	53.1	47.0	43.6	60.6	54.9	50.2

Table 2: The 3D-RES results on ScanRefer. † The mIoU and accuracy are reevaluated on our machine.

Thanks to the integration of well-established large-scale pre-trained models from the 2D domain within the MSE module, the visual representations in our model are more robust, enabling it to perform reliably even on rarely seen classes in the training set. To validate this, inspired by (Rozenberszki, Litany, and Dai 2022; Yan et al. 2024; Lu et al. 2023b; Takmaz et al. 2023), We categorized object classes based on their frequency of appearance in the training set and conducted testing accordingly, as shown in Tab. 3. Specifically, we categorized all target classes in ScanRefer into three groups. The first group, labeled “High”, consists of classes that make up more than 1% of the training set, accounting for approximately 75% of the total samples. The second group, labeled “Mid”, includes classes that comprise less than 1% but more than 0.1% of the training set, representing about 20%. The remaining classes, labeled “Low”, make up less than 0.1% of the training set and account for about 5% of the samples.

As shown, the performance of 3D-STMN (Wu et al. 2024b) and MDIN (Wu et al. 2024a) significantly drops for the “Low” frequency categories, decreasing by 17.5 and 14.9 points, respectively, compared to the “High” group. In contrast, our model shows a decrease of only 6.3 points. When comparing across models, our model outperforms MDIN by more than 10 points in the “Low” group. This substantial improvement highlights the enhanced robustness of our model, attributed to the multi-view semantic integration, enabling it to handle infrequent, long-tail samples effectively.

Method	High	Mid	Low	Overall
3D-STMN	39.1	46.7	21.6	39.5
MDIN	46.9	58.2	32.0	48.3
IPDN (Ours)	48.5	60.5	42.2	50.2

Table 3: Test results of the subsets of ScanRefer, divided by frequency, with the mIoU metric. High, Mid, and Low refer to categories that account for more than 1%, between 0.1% and 1%, and less than 0.1% of the training set, respectively.

4.4 Ablation Study

All of our ablation experiments were conducted on ScanRefer dataset (Chen, Chang, and Nießner 2020).

Component Ablation In our proposed IPDN, the main components include MSE and PAD. To assess the impact of these two components, we conducted an ablation study, as shown in Tab. 4. The results indicate that omitting both components results in a 2.1-point decrease in mIoU. Introducing MSE improves mIoU by 1.1 points, demonstrating its effectiveness in enhancing visual features. Further inclusion of PAD leads to an additional 1.0-point increase in mIoU, indicating that task-driven prompts effectively guide the model to focus on more important queries, thereby improving segmentation accuracy.

Spatial-aware Attention Ablation We conducted an ablation study on the hyperparameter k in the Spatial-aware

MSE	PAD	Acc@0.25	Acc@0.5	mIoU
×	×	58.2	52.9	48.1
✓	×	58.9	53.7	49.2
✓	✓	60.6	54.9	50.2

Table 4: Ablation study on the proposed components. Not using PAD signifies removing the prompt from the decoder.

k	Acc@0.25	Acc@0.5	mIoU
1	59.1	53.9	49.3
2	59.2	54.1	49.4
3	59.9	54.5	49.7
4	60.6	54.9	50.2
5	60.5	54.6	50.0

Table 5: Ablation study on the hyperparameter k .

Attention, and the results are shown in Tab. 5. From row 1 and row 2, we can see that when k is small, the expected performance is not achieved. This is because a smaller k represents a smaller receptive field for objects, which may be even smaller than the field of view of the image itself, thus leading to minimal improvement. At the same time, from row 3 \sim 5, it is evident that a larger k is not always better. When k is too large, objects may attend to distant irrelevant objects, causing a negative effect. In summary, setting k to 8 is considered reasonable.

PAD Ablation We conducted an ablation study on the hyperparameter r in the PAD, and the results are shown in Tab. 6. From the row 1, we can see that when r is 0 (no filtering), there is a significant amount of noise in the prompts, leading to minimal effect, with only a 0.2 mIoU improvement (compared to row 2 in Tab. 4). From row 2 to row 4, we observe that as r increases, more irrelevant queries are filtered out, resulting in improved prompting effects. From the row 5, we learn that r is not necessarily better when it is larger, because when r is too large, some relevant queries are also filtered out, leading to a decrease in the prompting effect. In summary, 0.75 is a suitable threshold, effectively filtering out irrelevant queries while retaining relevant ones.

4.5 Qualitative Comparison

In this section, we visualized a set of representative examples in ScanRefer dataset, as shown in Fig. 3. It can be seen

r	Acc@0.25	Acc@0.5	mIoU
1	59.3	54.3	49.4
2	59.8	54.2	49.6
3	59.7	54.5	49.6
4	60.6	54.9	50.2
5	60.3	54.4	50.0

Table 6: Ablation study on the hyperparameter r .

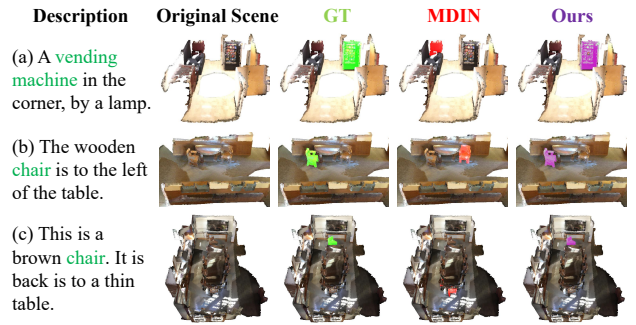


Figure 3: Qualitative comparison between MDIN and ours.

from the figure that our model demonstrates stronger reasoning capabilities compared to MDIN (Wu et al. 2024a). Specifically, in case (a), there is no distracting object present in the scene, only a vending machine, but MDIN still fails to identify it. This is because, within the ScanRefer training dataset of over 30,000 samples, there are only 10 samples where the target is a vending machine, which is insufficient for the model to recognize such an object. However, models with large-scale 2D pre-training do not suffer from this issue and can well identify vending machines, allowing our model to accurately locate the target. In case (b), the concept of “left” is involved, which is perspective-dependent. Since three-dimensional space theoretically contains an infinite number of perspectives, purely 3D models have difficulty distinguishing left from right. In contrast, the perspective in 2D images is fixed, which provides significant assistance in handling such cases. Finally, in case (c), thanks to the powerful prompting ability of our task-driven prompts, even when there are nearly ten distractor objects present, our model can still accurately locate the target object.

5 Conclusion

In this paper, we focus on addressing feature ambiguity and intent ambiguity by introducing the Image-enhanced Prompt Decoding Network (IPDN). To overcome feature ambiguity, we propose the Multi-view Semantic Embedding (MSE) module, which incorporates multi-view 2D image information into the 3D scene, compensating for any potential spatial information loss. To resolve intent ambiguity, we developed the Prompt-Aware Decoder (PAD), which guides the decoding process by generating task-driven signals from the interaction between the expression and visual features. Extensive experiments demonstrate the superiority of IPDN.

Acknowledgements

This work was supported by National Key R&D Program of China (No.2023YFB4502804), the National Science Fund for Distinguished Young Scholars (No.62025603), the National Natural Science Foundation of China (No. U22B2051, No. U21B2037, No. 62072389, No. 62302411, No. 624B2118), the Natural Science Foundation of Fujian Province of China (No.2021J06003), and China Postdoctoral Science Foundation (No. 2023M732948).

References

- Chen, D. Z.; Chang, A. X.; and Nießner, M. 2020. Scanrefer: 3d object localization in rgb-d scans using natural language. In *ECCV*.
- Chen, S.; Guhur, P.-L.; Tapaswi, M.; Schmid, C.; and Laptev, I. 2022. Language conditioned spatial relation reasoning for 3d object grounding. *NeurIPS*.
- Chng, Y. X.; Zheng, H.; Han, Y.; Qiu, X.; and Huang, G. 2024. Mask grounding for referring image segmentation. In *CVPR*.
- Dai, A.; Chang, A. X.; Savva, M.; Halber, M.; Funkhouser, T.; and Nießner, M. 2017. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *CVPR*.
- Deng, J.; Shi, S.; Li, P.; Zhou, W.; Zhang, Y.; and Li, H. 2021. Voxel r-cnn: Towards high performance voxel-based 3d object detection. In *AAAI*.
- Ding, H.; Liu, C.; Wang, S.; and Jiang, X. 2021. Vision-language transformer and query generation for referring segmentation. In *ICCV*.
- Dong, R.; Qi, Z.; Zhang, L.; Zhang, J.; Sun, J.; Ge, Z.; Yi, L.; and Ma, K. 2022. Autoencoders as cross-modal teachers: Can pretrained 2d image transformers help 3d representation learning? *arXiv*.
- Fei, H.; Wu, S.; Ji, W.; Zhang, H.; Zhang, M.; Lee, M.-L.; and Hsu, W. 2024a. Video-of-thought: Step-by-step video reasoning from perception to cognition. In *ICML*.
- Fei, H.; Wu, S.; Zhang, H.; Chua, T.-S.; and Shuicheng, Y. 2024b. Vitron: A unified pixel-level vision llm for understanding, generating, segmenting, editing. In *NeurIPS*.
- Fei, H.; Wu, S.; Zhang, M.; Zhang, M.; Chua, T.-S.; and Yan, S. 2024c. Enhancing video-language representations with structural spatio-temporal alignment. *TPAMI*.
- Feng, M.; Li, Z.; Li, Q.; Zhang, L.; Zhang, X.; Zhu, G.; Zhang, H.; Wang, Y.; and Mian, A. 2021. Free-form description guided 3d visual graph network for object grounding in point cloud. In *ICCV*.
- Graham, B.; Engelcke, M.; and Van Der Maaten, L. 2018. 3d semantic segmentation with submanifold sparse convolutional networks. In *CVPR*.
- He, D.; Zhao, Y.; Luo, J.; Hui, T.; Huang, S.; Zhang, A.; and Liu, S. 2021. Transrefer3d: Entity-and-relation aware transformer for fine-grained 3d visual grounding. In *ACM MM*.
- He, S.; and Ding, H. 2024. RefMask3D: Language-Guided Transformer for 3D Referring Segmentation. *arXiv*.
- He, S.; Ding, H.; Jiang, X.; and Wen, B. 2024. SegPoint: Segment Any Point Cloud via Large Language Model. *arXiv*.
- Huang, K.-C.; Li, X.; Qi, L.; Yan, S.; and Yang, M.-H. 2024. Reason3D: Searching and Reasoning 3D Segmentation via Large Language Model. *arXiv*.
- Huang, P.-H.; Lee, H.-H.; Chen, H.-T.; and Liu, T.-L. 2021. Text-guided graph neural networks for referring 3d instance segmentation. In *AAAI*.
- Jia, M.; Tang, L.; Chen, B.-C.; Cardie, C.; Belongie, S.; Hariharan, B.; and Lim, S.-N. 2022. Visual prompt tuning. In *ECCV*.
- Jing, Y.; Kong, T.; Wang, W.; Wang, L.; Li, L.; and Tan, T. 2021. Locate then segment: A strong pipeline for referring image segmentation. In *CVPR*.
- Kolodiazhnyi, M.; Vorontsova, A.; Konushin, A.; and Rukhovich, D. 2024. Oneformer3d: One transformer for unified point cloud segmentation. In *CVPR*.
- Lai, X.; Liu, J.; Jiang, L.; Wang, L.; Zhao, H.; Liu, S.; Qi, X.; and Jia, J. 2022. Stratified transformer for 3d point cloud segmentation. In *CVPR*.
- Lai, X.; Tian, Z.; Chen, Y.; Li, Y.; Yuan, Y.; Liu, S.; and Jia, J. 2024. Lisa: Reasoning segmentation via large language model. In *CVPR*.
- Lai, X.; Yuan, Y.; Chu, R.; Chen, Y.; Hu, H.; and Jia, J. 2023. Mask-attention-free transformer for 3d instance segmentation. In *ICCV*.
- Landrieu, L.; and Simonovsky, M. 2018. Large-scale point cloud semantic segmentation with superpoint graphs. In *CVPR*.
- Lester, B.; Al-Rfou, R.; and Constant, N. 2021. The power of scale for parameter-efficient prompt tuning. *arXiv*.
- Li, X. L.; and Liang, P. 2021. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv*.
- Lin, H.; Luo, Y.; Zheng, X.; Li, L.; Chao, F.; Jin, T.; Luo, D.; Wang, C.; Wang, Y.; and Cao, L. 2023. A unified framework for 3d point cloud visual grounding. *arXiv*.
- Liu, C.; Ding, H.; and Jiang, X. 2023. Gres: Generalized referring expression segmentation. In *CVPR*.
- Liu, X.; Ji, K.; Fu, Y.; Tam, W. L.; Du, Z.; Yang, Z.; and Tang, J. 2021. P-tuning v2: Prompt tuning can be comparable to fine-tuning universally across scales and tasks. *arXiv*.
- Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; and Stoyanov, V. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv*.
- Lu, J.; Deng, J.; Wang, C.; He, J.; and Zhang, T. 2023a. Query refinement transformer for 3d instance segmentation. In *ICCV*.
- Lu, S.; Chang, H.; Jing, E. P.; Boularias, A.; and Bekris, K. 2023b. Ovir-3d: Open-vocabulary 3d instance retrieval without training on 3d data. In *CoRL*.
- Luo, J.; Fu, J.; Kong, X.; Gao, C.; Ren, H.; Shen, H.; Xia, H.; and Liu, S. 2022. 3d-sps: Single-stage 3d visual grounding via referred point progressive selection. In *CVPR*.
- Milletari, F.; Navab, N.; and Ahmadi, S.-A. 2016. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *3DV*.
- Moening, C.; and Dodgson, N. A. 2003. Fast marching farthest point sampling. Technical report, University of Cambridge, Computer Laboratory.
- Peng, S.; Genova, K.; Jiang, C.; Tagliasacchi, A.; Pollefeys, M.; Funkhouser, T.; et al. 2023. Openscene: 3d scene understanding with open vocabularies. In *CVPR*.

- Qi, C. R.; Su, H.; Mo, K.; and Guibas, L. J. 2017a. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *CVPR*.
- Qi, C. R.; Yi, L.; Su, H.; and Guibas, L. J. 2017b. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *NeurIPS*.
- Qian, Z.; Ma, Y.; Ji, J.; and Sun, X. 2024a. X-RefSeg3D: Enhancing Referring 3D Instance Segmentation via Structured Cross-Modal Graph Neural Networks. In *AAAI*.
- Qian, Z.; Ma, Y.; Lin, Z.; Ji, J.; Zheng, X.; Sun, X.; and Ji, R. 2024b. Multi-branch Collaborative Learning Network for 3D Visual Grounding. *arXiv*.
- Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. 2021. Learning transferable visual models from natural language supervision. In *ICML*.
- Roh, J.; Desingh, K.; Farhadi, A.; and Fox, D. 2022. LanguageRefer: Spatial-language model for 3d visual grounding. In *CoRL*.
- Rozenberszki, D.; Litany, O.; and Dai, A. 2022. Language-grounded indoor 3d semantic segmentation in the wild. In *ECCV*.
- Schult, J.; Engelmann, F.; Hermans, A.; Litany, O.; Tang, S.; and Leibe, B. 2023. Mask3d: Mask transformer for 3d semantic instance segmentation. In *ICRA*.
- Shah, N. A.; VS, V.; and Patel, V. M. 2024. LQMFormer: Language-aware Query Mask Transformer for Referring Image Segmentation. In *CVPR*.
- Shi, S.; Guo, C.; Jiang, L.; Wang, Z.; Shi, J.; Wang, X.; and Li, H. 2020. Pv-rcnn: Point-voxel feature set abstraction for 3d object detection. In *CVPR*.
- Sun, J.; Qing, C.; Tan, J.; and Xu, X. 2023. Superpoint transformer for 3d scene instance segmentation. In *AAAI*.
- Takmaz, A.; Fedele, E.; Sumner, R. W.; Pollefeys, M.; Tombari, F.; and Engelmann, F. 2023. Openmask3d: Open-vocabulary 3d instance segmentation. *arXiv*.
- Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *NeurIPS*.
- Wang, Z.; Huang, H.; Zhao, Y.; Li, L.; Cheng, X.; Zhu, Y.; Yin, A.; and Zhao, Z. 2023. 3drp-net: 3d relative position-aware network for 3d visual grounding. *arXiv*.
- Wang, Z.; Li, Y.; Liu, T.; Zhao, H.; and Wang, S. 2024. OV-Uni3DETR: Towards Unified Open-Vocabulary 3D Object Detection via Cycle-Modality Propagation. *arXiv*.
- Wang, Z.; Lu, Y.; Li, Q.; Tao, X.; Guo, Y.; Gong, M.; and Liu, T. 2022a. Cris: Clip-driven referring image segmentation. In *CVPR*.
- Wang, Z.; Zhang, Z.; Lee, C.-Y.; Zhang, H.; Sun, R.; Ren, X.; Su, G.; Perot, V.; Dy, J.; and Pfister, T. 2022b. Learning to prompt for continual learning. In *CVPR*.
- Wu, C.; Liu, Y.; Ji, J.; Ma, Y.; Wang, H.; Luo, G.; Ding, H.; Sun, X.; and Ji, R. 2024a. 3D-GRES: Generalized 3D Referring Expression Segmentation. *arXiv*.
- Wu, C.; Ma, Y.; Chen, Q.; Wang, H.; Luo, G.; Ji, J.; and Sun, X. 2024b. 3d-stmn: Dependency-driven superpoint-text matching network for end-to-end 3d referring expression segmentation. In *AAAI*.
- Wu, T.-Y.; Huang, S.-Y.; and Wang, Y.-C. F. 2024. DOrA: 3D Visual Grounding with Order-Aware Referring. *arXiv*.
- Wu, Y.; Cheng, X.; Zhang, R.; Cheng, Z.; and Zhang, J. 2023. Eda: Explicit text-decoupling and dense alignment for 3d visual grounding. In *CVPR*.
- Xu, W.; Shi, C.; Tu, S.; Zhou, X.; Liang, D.; and Bai, X. 2024. A Unified Framework for 3D Scene Understanding. *arXiv*.
- Yan, M.; Zhang, J.; Zhu, Y.; and Wang, H. 2024. Maskclustering: View consensus based mask graph clustering for open-vocabulary 3d instance segmentation. In *CVPR*.
- Yang, L.; Zhang, Z.; Qi, Z.; Xu, Y.; Liu, W.; Shan, Y.; Li, B.; Yang, W.; Li, P.; Wang, Y.; et al. 2024. Exploiting contextual objects and relations for 3d visual grounding. *NeurIPS*.
- Yang, Y.-Q.; Guo, Y.-X.; Xiong, J.-Y.; Liu, Y.; Pan, H.; Wang, P.-S.; Tong, X.; and Guo, B. 2023. Swin3d: A pre-trained transformer backbone for 3d indoor scene understanding. *arXiv*.
- Yang, Z.; Wang, J.; Tang, Y.; Chen, K.; Zhao, H.; and Torr, P. H. 2022. Lavt: Language-aware vision transformer for referring image segmentation. In *CVPR*.
- Yang, Z.; Zhang, S.; Wang, L.; and Luo, J. 2021. Sat: 2d semantics assisted training for 3d visual grounding. In *ICCV*.
- Yu, Q.; Du, H.; Liu, C.; and Yu, X. 2024. When 3D Bounding-Box Meets SAM: Point Cloud Instance Segmentation with Weak-and-Noisy Supervision. In *WACV*.
- Yuan, Z.; Yan, X.; Liao, Y.; Zhang, R.; Wang, S.; Li, Z.; and Cui, S. 2021. Instancerefer: Cooperative holistic understanding for visual grounding on point clouds through instance multi-level contextual referring. In *ICCV*.
- Zhang, J.; Dong, R.; and Ma, K. 2023. Clip-fo3d: Learning free open-world 3d scene representations from 2d dense clip. In *ICCV*.
- Zhang, J.; Fan, G.; Wang, G.; Su, Z.; Ma, K.; and Yi, L. 2023. Language-assisted 3D feature learning for semantic scene understanding. In *AAAI*.
- Zhang, Y.; Gong, Z.; and Chang, A. X. 2023. Multi3drefer: Grounding text description to multiple 3d objects. In *ICCV*.
- Zhang, Y.; Zhou, K.; and Liu, Z. 2022. Neural prompt search. *arXiv*.
- Zhao, H.; Jiang, L.; Jia, J.; Torr, P. H.; and Koltun, V. 2021a. Point transformer. In *ICCV*.
- Zhao, L.; Cai, D.; Sheng, L.; and Xu, D. 2021b. 3DVG-Transformer: Relation modeling for visual grounding on point clouds. In *ICCV*.
- Zhou, K.; Yang, J.; Loy, C. C.; and Liu, Z. 2022a. Conditional prompt learning for vision-language models. In *CVPR*.
- Zhou, K.; Yang, J.; Loy, C. C.; and Liu, Z. 2022b. Learning to prompt for vision-language models. *IJCV*.
- Zhu, B.; Niu, Y.; Han, Y.; Wu, Y.; and Zhang, H. 2023. Prompt-aligned gradient for prompt tuning. In *ICCV*.