3D-STMN: Dependency-Driven Superpoint-Text Matching Network for End-to-End 3D Referring Expression Segmentation

Changli Wu*, Yiwei Ma*, Qi Chen, Haowei Wang, Gen Luo, Jiayi Ji†, Xiaoshuai Sun

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

{wuchangli, yiweima, chenqi, wanghaowei, luogen}@stu.xmu.edu.cn, jjyxmu@gmail.com, xssun@xmu.edu.cn

Abstract

In 3D Referring Expression Segmentation (3D-RES), the earlier approach adopts a two-stage paradigm, extracting segmentation proposals and then matching them with referring expressions. However, this conventional paradigm encounters significant challenges, most notably in terms of the generation of lackluster initial proposals and a pronounced deceleration in inference speed. Recognizing these limitations, we introduce an innovative end-to-end Superpoint-Text Matching Network (3D-STMN) that is enriched by dependency-driven insights. One of the keystones of our model is the Superpoint-Text Matching (STM) mechanism. Unlike traditional methods that navigate through instance proposals, STM directly correlates linguistic indications with their respective superpoints, clusters of semantically related points. This architectural decision empowers our model to efficiently harness cross-modal semantic relationships, primarily leveraging densely annotated superpoint-text pairs, as opposed to the more sparse instance-text pairs. In pursuit of enhancing the role of text in guiding the segmentation process, we further incorporate the Dependency-Driven Interaction (DDI) module to deepen the network's semantic comprehension of referring expressions. Using the dependency trees as a beacon, this module discerns the intricate relationships between primary terms and their associated descriptors in expressions, thereby elevating both the localization and segmentation capacities. Comprehensive experiments on the ScanRefer benchmark reveal that our model not only sets new performance standards, registering an mIoU gain of 11.7 points but also achieves a staggering enhancement in inference speed, surpassing traditional methods by 95.7 times. The code and models are available at https://github.com/sosppxo/3D-STMN.

1 Introduction

The goal of 3D visual grounding is to locate instances within a 3D scene based on given natural language descriptions (Chen, Chang, and Nießner 2020). In recent years, it has become a hot topic in academic research due to its wide-ranging application scenarios, including autonomous robotics, human-machine interaction, and self-driving systems, among others. Within this field, the task of 3D Referring Expression Segmentation (3D-RES) emerges as a formidable challenge. Compared to 3D visual detection tasks (Wang et al. 2022; He et al. 2022; Chen, Chang, and Nießner 2020; Achlioptas et al. 2020; Zhao et al. 2021; Luo et al. 2022; Huang et al. 2023), which merely locate the target objects with bounding boxes, 3D-RES demands a more complex understanding. It not only requires the identification of target instances within sparse point clouds, but it also requires the provision of precise 3D masks that correspond to each identified target instance.

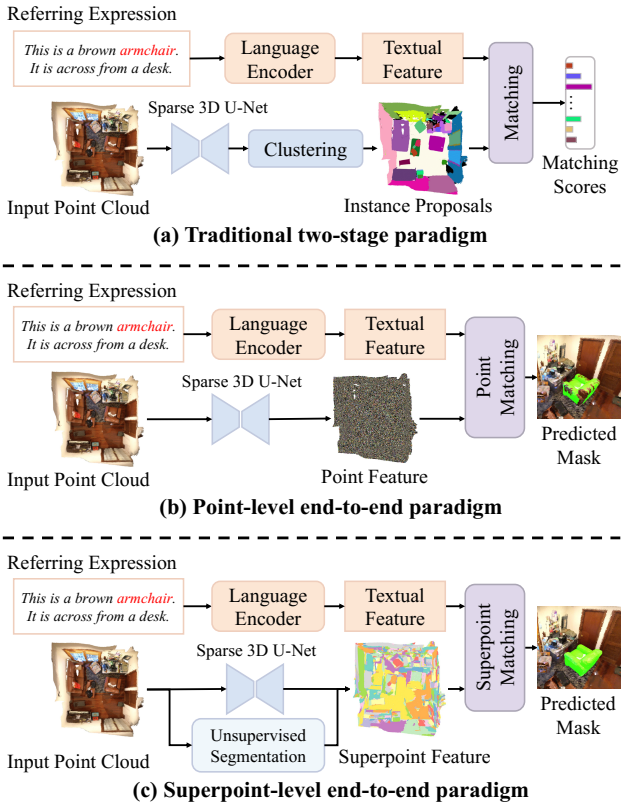

Figure 1: A comparison among (a) traditional two-stage paradigm, (b) point-level end-to-end paradigm, and (c) superpoint-level end-to-end paradigm.

*These authors contributed equally.

†The corresponding author.

At present, the only existing method referred to as TGNN (Huang et al. 2021), operates in a two-stage manner. In the initial stage, an independent text-agnostic segmentation model is trained to generate instance proposals. Then, in the second stage, a graph neural network is employed to forge links between the generated proposals and textual descriptions, as shown in Fig. 1-(a). Despite achieving good results, this two-stage paradigm still suffers from three primary issues: (1) The decoupling of segmentation from matching creates an over-reliance on the preliminary text-independent segmentation outcomes. Any inaccuracies or omissions in the first phase can insurmountably compromise the accuracy of the subsequent matching phase, irrespective of its intrinsic efficacy. (2) The model overlooks the inherent hierarchical and dependency structures within the referring sentence. Its linear language modeling strategy falls short in capturing intricate semantic nuances, leading to missteps in both localization and segmentation. (3) To amplify the recall efficiency in the secondary phase, the first stage extracts dense candidate masks through iterative clustering over several stages. This iterative process considerably decelerates the model's inference speed. Hence, despite its merits, the two-stage paradigm employed by TGNN leaves room for substantial improvement in both accuracy and efficiency.

A natural approach would be to employ an end-to-end method that directly matches textual features with points in the 3D point cloud, as shown in Fig. 1-(b). This approach has been widely proven effective in 2D-RES tasks (Ye et al. 2019; Liu et al. 2019; Ding et al. 2021; Yang et al. 2022). However, it has not been translated well to sparse, irregular 3D point cloud data, as it results in a low recall rate. As a solution, 3D-SPS (Luo et al. 2022) in 3D visual detection has suggested a method that progressively selects keypoints guided by language and regresses boxes using this keypoint information. However, this approach disrupts the continuity of the 3D mask in 3D-RES tasks, thereby deteriorating the segmentation results.

To tackle the aforementioned challenges, we present a dependency-driven Superpoint-Text Matching Network for an end-to-end 3D-RES. The idea of our approach is the matching of the expressions with over-segmented superpoints (Landrieu and Simonovsky 2018). As illustrated in Fig. 1-(c), these superpoints are initially aggregated through a clustering algorithm, thus attaining fine-grained semantic units. These superpoints, embodying semantics and being significantly fewer compared to the unordered points in a 3D point cloud, offer advantages in performance and speed during the matching process. In contrast to the proposals in TGNN, superpoints are fine-grained units derived from over-segmentation, capable of covering the entire scene, thereby averting the issues of inaccurate segmentation or missing instances. In light of this, we introduce a new Superpoint-Text Matching (STM) mechanism for 3D-RES, leveraging the aggregation of text features from superpoints to acquire the mask of the target instance. To bolster semantic parsing from the textual perspective, we devise a Dependency-Driven Interaction (DDI) module, achieving token-level interactions. This module exploits the prior information from the dependency syntax tree to steer the flow of text information. This structure further enhances inference on relationships among different instances via the network architecture, thus markedly improving the model's segmentation ability. We have conducted extensive quantitative and qualitative experiments on the classic ScanRefer dataset for investigation. It's noteworthy that our method achieves a remarkable 95.7-fold increase in inference speed while outperforming TGNN by an impressive 11.7 points.

To summarize, our main contributions are as follows:

- We propose a novel efficient end-to-end framework 3D-STMN based on Superpoint-Text Matching (STM) mechanism for aligning superpoint with textual modality, making superpoint a highly competitive player in multi-modal representation.
- We design a Dependency-Driven Interaction (DDI) module to exploit the prior information from the dependency syntax tree to steer the flow of text information, markedly improving the model's segmentation ability.
- Extensive experiments show that our method significantly outperforms the previous two-stage baseline in the ScanRefer benchmark, registering a mIoU gain of 11.7 points but also achieving a staggering enhancement in inference speed.

2 Related Work

2.1 2D Referring Expression Comprehension and Segmentation

Vision and language play a crucial role in human understanding of the environment (Ma et al. 2022; Ji et al. 2022; Ma et al. 2023; He et al. 2021; Wu et al. 2023a; Zhao et al. 2023; Zhang et al. 2023). In the context of 2D-REC tasks, the objective is to predict a bounding box corresponding to the object described in a given referring expression (Nagaraja, Morariu, and Davis 2016; Yu et al. 2016; Hu et al. 2017; Yu et al. 2017; Deng et al. 2018; Zhuang et al. 2018; Sadhu, Chen, and Nevatia 2019; Yang, Li, and Yu 2020; Luo et al. 2020b). Conversely, in 2D-RES tasks, the aim is to accurately predict a segmentation mask delineating the referred object for more precise localization (Hu, Rohrbach, and Darrell 2016; Yu et al. 2018; Ye et al. 2019; Shi et al. 2018). Many existing approaches adopt a two-stage paradigm involving segmentation followed by matching (Li et al. 2018; Margffoy-Tuay et al. 2018). To overcome the limitation of the quality of segmentation models, some methods have been proposed that refine segmentation masks using single-stage networks (Ye et al. 2019; Liu et al. 2019; Luo et al. 2020a). While these approaches have shown promise in 2D tasks, their direct application to 3D point cloud scenes is hindered by the inherent challenges posed by the sparse and irregular nature of 3D point clouds.

2.2 3D Referring Expression Comprehension and Segmentation

Recently, 3D REC has garnered significant attention, aiming to localize objects within a 3D scene based on referring expressions. ScanRefer (Chen, Chang, and Nießner 2020)

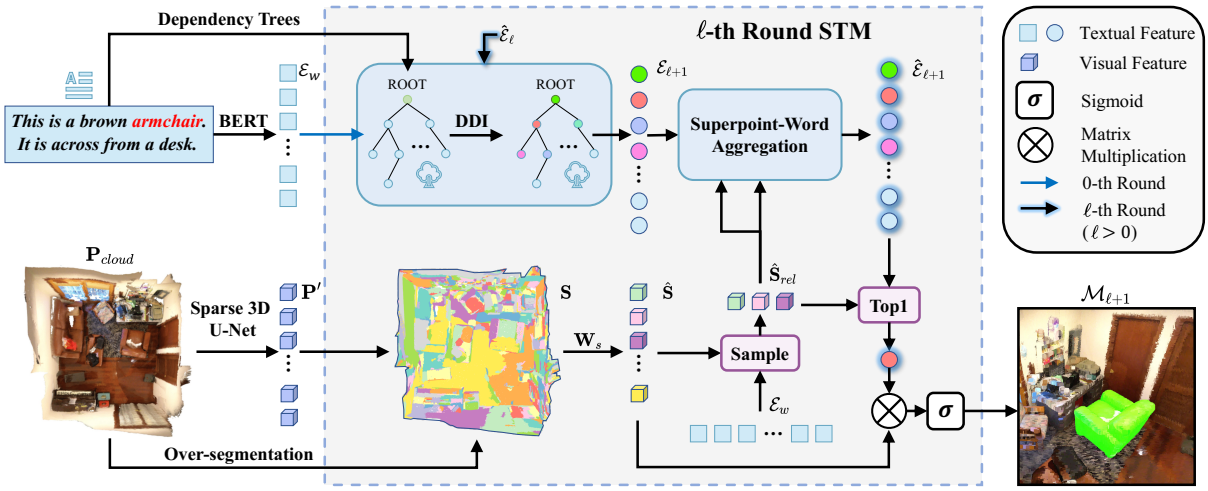

Figure 2: Overview of our 3D Superpoint-Text Matching Network (3D-STMN).

provides a dataset based on ScanNetv2 (Dai et al. 2017) for the Referring 3D Instance Localization task. Additionally, ReferIt3D (Achlioptas et al. 2020) proposes two datasets, Sr3D and Nr3D. Most existing methods (Chen, Chang, and Nießner 2020; Achlioptas et al. 2020; Zhao et al. 2021; Yuan et al. 2021; Yang et al. 2021; Huang et al. 2021; Feng et al. 2021) adopt a two-stage paradigm. Meanwhile, some methods have adopted an single-stage framework (Luo et al. 2022; Jain et al. 2022; Wu et al. 2023b). To address 3D-RES tasks, TGNN (Huang et al. 2021) proposed a two-stage model. However, both the accuracy and inference speed of TGNN exhibit inherent limitations of the segmentation model. To circumvent these challenges, we propose an end-to-end Superpoint-Text Matching Network in this paper.

2.3 Superpoint based 3D Scene Understanding

Similar to superpixels (Achanta et al. 2012; Tu et al. 2018), superpoints have been used for point cloud segmentation (Papon et al. 2013; Lin et al. 2018; Landrieu and Simonovsky 2018; Robert, Raguet, and Landrieu 2023) and object detection (Han et al. 2020; Engelmann et al. 2020). For 3D instance segmentation, superpoints have also demonstrated incredible potential (Liang et al. 2021; Sun et al. 2023). However, these works only applied superpoint to pure visual tasks and did not explore the ability of superpoint to align with text. In this paper, we first propose a framework for aligning superpoint with text, making superpoint a highly competitive player in multimodal representations.

3 Method

In this section, we provide a comprehensive overview of the 3D-STMN. The framework is illustrated in Fig. 2.

3.1 Feature Extraction

Visual Modality Given a point cloud scene with $\mathcal{N}_p$ points, it can be represented as $\mathbf{P}_{cloud} \in \mathbb{R}^{\mathcal{N}_p \times (3+F)}$. Here, each point comes with 3D coordinates along with an F-dimensional auxiliary feature that includes RGB, normal vectors, among others. Building on TGNN (Huang et al. 2021), we employ a singular Sparse 3D U-Net (Graham, Engelcke, and Van Der Maaten 2018) to extract point-wise features, represented as $\mathbf{P}' \in \mathbb{R}^{\mathcal{N}_p \times C_p}$.

Linguistic Modality Given a free-form plain text description of the target object with $\mathcal{N}_w$ words $\{c_i\}_{i=1}^{\mathcal{N}_w}$, we follow (Huang et al. 2021) to adopt a pre-trained BERT (Devlin et al. 2018) to extract the C_t-dimensional word-level embeddings $\mathcal{E}_w \in \mathbb{R}^{\mathcal{N}_w \times C_t}$, and description-level embedding $\mathbf{d}_0 \in \mathbb{R}^{C_t}$ which is the embeddings of [CLS] token.

3.2 Superpoint-Text Matching

Superpoints and Dependency-Driven Text After extracting the features, we perform over-segmentation to $\mathbf{P}_{cloud}$ to obtain $\mathcal{N}_s$ superpoints $\{\mathcal{K}_i\}_{i=1}^{\mathcal{N}_s}$(Landrieu and Simonovsky 2018).

To obtain the superpoint-level features $\mathbf{S} \in \mathbb{R}^{\mathcal{N}_s \times C_p}$, we directly feed point-wise features $\mathbf{P}'$ into superpoint pooling layer based on $\{\mathcal{K}_i\}_{i=1}^{\mathcal{N}_s}$, which can be formulated as:

$$\mathbf{S}^i = \text{AvgPool}(\mathbf{P}', \mathcal{K}_i), \tag{1}$$

where $\mathbf{S}^i$ denotes the feature of the i-th superpoint, $\mathcal{K}_i$ denotes the set of indices of points contained in the i-th superpoint, AvgPool$(\cdot)$ is superpoint average pooling operation.

To the text end, we feed the expression with word-level embeddings $\mathcal{E}_w$ into the proposed DDI module which aims to construct a description-dependency graph and outputs the dependency-driven feature $\mathcal{E}_0$, which is formulated as:

$$\hat{\mathcal{E}}_0 = (\mathcal{E}_{\text{root}} \,\|\, \mathcal{E}_w)\mathbf{W}_t, \tag{2}$$

$$\mathcal{E}_1 = \text{DDI}(\hat{\mathcal{E}}_0), \tag{3}$$

where $\mathbf{W}_t \in \mathbb{R}^{C_t \times D}$ is a learnable parameter, $\mathcal{E}_w \in \mathbb{R}^{\mathcal{N}_w \times C_t}$, $\mathcal{E}_1 \in \mathbb{R}^{(\mathcal{N}_w+1) \times D}$, $\mathcal{E}_{\text{root}}$ is the randomly initialized ROOT node feature, $\|$ denotes the concatenation operation. More details about DDI are presented in Sec. 3.3.

To enhance the efficiency of subsequent processing, we adopt a filtering approach on $\mathbf{S}$ after linear projection, which

is widely used in multimodal segmentation tasks (Ding et al. 2022; Luo et al. 2022). Specifically, we acquire the k_{rel} superpoints based on the relevance score s_r between the superpoints and their corresponding descriptions. The filtering process can be given by:

$$\hat{\mathbf{S}} = \mathbf{S}\mathbf{W}_s, \tag{4}$$

$$\mathcal{A} = \text{softmax}\Big(\frac{\hat{\mathbf{S}}\mathbf{Q}_s \cdot (\mathcal{E}_w\mathbf{K}_t)^T}{\sqrt{D}}\Big), \tag{5}$$

$$s_r^i = \sum_{j=1}^{\mathcal{N}_w} \mathcal{A}^{ij}, \tag{6}$$

$$\hat{\mathbf{S}}_{rel} = \hat{\mathbf{S}}[\text{ArgTopk}(s_r, k_{rel})] \parallel \text{AvgPool}(\hat{\mathbf{S}}), \tag{7}$$

where $\mathbf{W}_s \in \mathbb{R}^{C_p \times D}, \mathbf{Q}_s \in \mathbb{R}^{D\times D}, \mathbf{K}_t \in \mathbb{R}^{C_t \times D}$ denote learnable parameters. $\text{AvgPool}(\hat{\mathbf{S}})$ plays the role of golobal features, and $\parallel$ denotes concatenation. $\hat{\mathbf{S}}_{rel} \in \mathbb{R}^{(k_{rel}+1)\times D}$ denotes features of description-relevant superpoints.

Superpoint-Text Matching Process To perform Superpoint-Text matching, we initially project the superpoint features $\mathbf{S}$ to a D-dimensional subspace that corresponds to the text embedding $\mathcal{E}$. After a description-guided sampling of superpoints, we update the embedding of each text token using Superpoints-Word Aggregation (SWA) with adaptive attention weights. We design it as a multi-round refinement process:

$$\mathcal{E}_{\ell+1} = \text{DDI}(\hat{\mathcal{E}}_\ell), \tag{8}$$

$$\hat{\mathcal{E}}_{\ell+1} = \text{SWA}(\mathcal{E}_{\ell+1}, \hat{\mathbf{S}}_{rel}), \ell = 0, 1, ..., L-1, \tag{9}$$

where $\hat{\mathcal{E}}_\ell, \hat{\mathcal{E}}_{\ell+1} \in \mathbb{R}^{(\mathcal{N}_w+1)\times D}$ and L is the number of multiple rounds. The details about SWA are presented in the following subsection.

Next, we perform matrix multiplication between $\hat{\mathbf{S}}$ and $\hat{\mathcal{E}}$ to obtain the response maps that capture the relationship between all superpoints and word tokens. This computation can be described as follows:

$$\mathbf{M}_{\ell+1} = \sigma(\hat{\mathcal{E}}_{\ell+1}\hat{\mathbf{S}}^T), \tag{10}$$

where $\hat{\mathbf{S}}^T \in \mathbb{R}^{D\times\mathcal{N}_s}$ is the transpose of $\hat{\mathbf{S}}$, $\mathbf{M}_{\ell+1} \in \mathbb{R}^{(\mathcal{N}_w+1)\times\mathcal{N}_s}$ is the response maps, and $\sigma(\cdot)$ denotes sigmoid function. In particular, $\mathbf{M}^n_{\ell+1} \in \mathbb{R}^{\mathcal{N}_s}$ is the response map of the n-th token, based on which we can generate the segmentation result and attention mask $\mathbf{A}^n_{\ell+1} \in \mathbb{R}^{\mathcal{N}_s}$ corresponding to the n-th token.

To obtain the final mask, we choose the response map $\mathcal{M}_{\ell+1} \in \mathbb{R}^{\mathcal{N}_s}$ associated with the word token that has the highest correlation score with all description-relevant superpoints:

$$\mathbf{A}_{v,\ell+1} = \text{softmax}\Big(\frac{\hat{\mathcal{E}}_{\ell+1}\mathbf{Q}_t^{\ell+1} \cdot (\hat{\mathbf{S}}_{rel}\mathbf{K}_s^{\ell+1})^T}{\sqrt{D}}\Big), \tag{11}$$

$$\mathbf{s}_v^i = \sum_{j=1}^{k_{rel}+1} \mathbf{A}^{ij}_{v,\ell+1}, \tag{12}$$

$$\mathcal{M}_{\ell+1} = \mathbf{M}_{\ell+1}[\text{ArgMax}(\mathbf{s}_v)], \tag{13}$$

where $\text{ArgMax}(\cdot)$ returns the index corresponding to the maximum value. $\mathbf{Q}_t^{\ell+1}, \mathbf{K}_s^{\ell+1} \in \mathbb{R}^{D\times D}$ are learnable parameters. $\mathbf{A}^{ij}_{v,\ell}$ means the attention score between the i-th word and the j-th description-relevant superpoint, and $\mathbf{s}_v^i$ denotes the visual correlation score of the i-th word.

Superpoint-Word Aggregation To enhance the discriminative power of the textual segmentation kernel, we introduce a Superpoint-Word Aggregation module, which is designed to refine the multi-round modality interaction between superpoints and textual descriptions.

At the ℓ-th layer, SWA adaptively aggregates the superpoint features to enables each word to absorb the visual information of the related superpoint features.

As depicted in Fig. 2, the adaptive superpoint-word cross-attention block utilizes the dependency-driven feature $\mathcal{E}$ to refine the word features by incorporating information from the related superpoints:

$$\hat{\mathcal{E}}_{\ell+1} = \text{softmax}\Big(\frac{\mathcal{E}_{\ell+1}\mathbf{Q}_{\ell+1} \cdot (\hat{\mathbf{S}}_{rel}\mathbf{K}_{\ell+1})^T}{\sqrt{D}} + \mathbf{A}_\ell\Big) \cdot \hat{\mathbf{S}}_{rel}\mathbf{V}_{\ell+1}, \tag{14}$$

where $\hat{\mathcal{E}}_{\ell+1} \in \mathbb{R}^{(\mathcal{N}_w+1)\times D}$is the output of superpoint-word cross-attention. $\mathbf{Q}_{\ell+1}, \mathbf{K}_{\ell+1}, \mathbf{V}_{\ell+1} \in \mathbb{R}^{D\times D}$ are learnable parameters. $\mathbf{A}_\ell \in \mathbb{R}^{(\mathcal{N}_w+1)\times(k_{rel}+1)}$ is superpoint attention masks. Given the predicted superpoint masks $\mathbf{M}_\ell$ from the prediction head, superpoint attention masks $\mathbf{A}_\ell$ filter superpoint with a threshold τ, as

$$\mathbf{A}_\ell^{ij} = \begin{cases} 0 & \text{if} \quad \mathbf{M}_\ell^{ij} \geq \tau \\ -\infty & \text{otherwise} \end{cases}. \tag{15}$$

$\mathbf{A}_\ell^{ij}$ indicates i-th word token attending to j-th superpoint where $\mathbf{M}_\ell^{ij}$ is higher than τ. Empirically, we set τ to 0.5. With transformer decoder layer stacking, superpoint attention masks $\mathbf{A}_\ell$ adaptively constrain cross-attention within the target instance.

3.3 Dependency-Driven Interaction

To explicitly decouple the textual description and effectively capture the dependency between words, we propose the Dependency-Driven Interaction module.

Description-Dependency Graph Given a free-form plain text description of the target object consisting of $\mathcal{N}_t$ sentences and a total of $\mathcal{N}_w$ words, we first use the Stanford CoreNLP (Manning et al. 2014) toolkit to obtain $\mathcal{N}_t$ dependency trees. Then we merge these $\mathcal{N}_t$ dependency trees into one graph by combining their ROOT nodes, as shown in Fig 2. Thus, for every description, the dependency graph has $\mathcal{N}_w + 1$ nodes $\{u\}$ with $\mathcal{N}_w$ edges $\{e\}$. Each node represents a word including the special token "ROOT", while each edge represents a type of dependency relationship.

Graph Transformer Layer with edge features Inspired by (Dwivedi and Bresson 2020), we adopt a Graph Transformer Layer with edge features to more effectively leverage the abundant feature information available in Description-Dependency Graphs, which is stored in the form of edge

attributes including dependency relationship. Given the textual features $\hat{\mathcal{E}}_0 = \{\hat{\mathcal{E}}_0^0, \hat{\mathcal{E}}_0^1, \cdots, \hat{\mathcal{E}}_0^{\mathcal{N}_w+1}\}$, we directly derive the node features $\hat{\mathbf{h}}_i^0 = \{\hat{\mathbf{h}}_0^0, \hat{\mathbf{h}}_1^0, \cdots, \hat{\mathbf{h}}_{\mathcal{N}_w+1}^0\}$ based on their corresponding indices. For the edge features$\{\beta_{ij}\}$, we assign a unique ID to each dependency relationship which is passed via a linear projection to obtain D-dimensional hidden features $\mathbf{e}_{ij}^0$.

$$\hat{\mathbf{h}}_i^0 = \hat{\mathcal{E}}_0^i, \tag{16}$$

$$\mathbf{e}_{ij}^0 = \beta_{ij}\mathbf{B}^0 + \mathbf{b}^0, \tag{17}$$

where $\mathbf{B}^0 \in \mathbb{R}^{1\times D}$ and $\mathbf{b}^0 \in \mathbb{R}^D$ are the parameters of the linear projection layers. We also add the precomputed node positional encodings to the node features following (Dwivedi and Bresson 2020).

Next, we proceed to define the update equations for the ℓ-th layer.

$$\mathbf{h}_i^\ell = \hat{\mathcal{E}}_\ell^i, \tag{18}$$

$$\hat{\mathbf{w}}_{ij}^\ell = \Big(\frac{\mathbf{h}_i^\ell \mathbf{Q}_h^\ell \cdot \mathbf{h}_j^\ell \mathbf{K}_h^\ell}{\sqrt{D}}\Big) \cdot e_{ij}^\ell \mathbf{E}_e^\ell, \tag{19}$$

$$\mathbf{w}_{ij}^\ell = \text{softmax}_j(\hat{\mathbf{w}}_{ij}^\ell), \tag{20}$$

$$\hat{\mathbf{h}}_i^{\ell+1} = \Big(\sum_{j\in\mathcal{N}_i} \mathbf{w}_{ij}^\ell (\mathbf{h}_j^\ell \mathbf{V}_h^\ell)\Big)\mathbf{O}_h^\ell, \tag{21}$$

$$\hat{\mathbf{e}}_{ij}^{\ell+1} = \hat{\mathbf{w}}_{ij}^\ell \mathbf{O}_e^\ell, \tag{22}$$

where $\mathbf{Q}_h^\ell, \mathbf{K}_h^\ell, \mathbf{V}_h^\ell, \mathbf{E}_e^\ell, \mathbf{O}_h^\ell, \mathbf{O}_e^\ell \in \mathbb{R}^{D\times D}$ denote learnable parameters.

Considering the lack of long-range connections in dependency graph structures, we introduce self-attention mechanism and combine it with graph attention in parallel. The outputs $\hat{\mathbf{h}}_i^{\ell+1}$ are added by a self-attention outputs of $\mathbf{h}_i^\ell$ and succeeded by residual connections and normalization layers to get the outputs $\widetilde{\mathbf{h}}_i^{\ell+1}$. $\widetilde{\mathbf{h}}_i^{\ell+1}$ and $\hat{\mathbf{e}}_{ij}^{\ell+1}$ are then passed to separate Feed Forward Networks preceded and succeeded by residual connections and normalization layers, as:

$$\widetilde{\mathbf{h}}_i^{\ell+1} = \text{Norm}\Big(\mathbf{h}_i^\ell + \text{SA}(\mathbf{h}_i^\ell) + \hat{\mathbf{h}}_i^{\ell+1}\Big), \tag{23}$$

$$\overline{\mathbf{h}}_i^{\ell+1} = \text{GeLU}(\widetilde{\mathbf{h}}_i^{\ell+1}\mathbf{W}_{h1}^\ell)\mathbf{W}_{h2}^\ell, \tag{24}$$

$$\mathbf{h}_i^{\ell+1} = \text{Norm}\Big(\widetilde{\mathbf{h}}_i^{\ell+1} + \overline{\mathbf{h}}_i^{\ell+1}\Big), \tag{25}$$

$$\widetilde{\mathbf{e}}_{ij}^{\ell+1} = \text{Norm}\Big(\mathbf{e}_{ij}^\ell + \hat{\mathbf{e}}_{ij}^{\ell+1}\Big), \tag{26}$$

$$\overline{\mathbf{e}}_{ij}^{\ell+1} = \text{ReLU}(\widetilde{\mathbf{e}}_{ij}^{\ell+1}\mathbf{W}_{e1}^\ell)\mathbf{W}_{e2}^\ell, \tag{27}$$

$$\mathbf{e}_{ij}^{\ell+1} = \text{Norm}\Big(\widetilde{\mathbf{e}}_{ij}^{\ell+1} + \overline{\mathbf{e}}_{ij}^{\ell+1}\Big), \tag{28}$$

where $\mathbf{W}_{h1}^\ell \in \mathbb{R}^{D\times 2D}$, $\mathbf{W}_{h2}^\ell \in \mathbb{R}^{2D\times D}$, $\mathbf{W}_{e1}^\ell \in \mathbb{R}^{D\times 2D}$, $\mathbf{W}_{e2}^\ell \in \mathbb{R}^{2D\times D}$ are learnable parameters, $\widetilde{\mathbf{h}}_i^{\ell+1}, \overline{\mathbf{h}}_i^{\ell+1}, \widetilde{\mathbf{e}}_{ij}^{\ell+1}$, $\overline{\mathbf{e}}_{ij}^{\ell+1}$ denote intermediate representations, $\text{SA}(\mathbf{h}_i^\ell)$ means the i-th outputs of self-attention of $\hat{\mathcal{E}}^\ell$.

Finally, the textual output of ℓ-th layer DDI is obtained by concatenation of $\{\mathbf{h}_i^{\ell+1}\}_{i=1}^{\mathcal{N}_w+1}$.

$$\mathcal{E}_{\ell+1} = \mathbf{h}_1^{\ell+1} \parallel \mathbf{h}_2^{\ell+1} \parallel \ldots \parallel \mathbf{h}_{\mathcal{N}_w+1}^{\ell+1}. \tag{29}$$

3.4 Training Objective

It is straight-forward to train a superpoint-referring expression matching network: given ground-truth binary mask of the referring expression $\mathbf{Y} \in \mathbb{R}^{\mathcal{N}_p}$, we first get the corresponding superpoint mask $\mathbf{Y}_s \in \mathbb{R}^{\mathcal{N}_s}$ by superpoint pooling followed by a 0.5-threshold binarization, and then we apply the binary cross-entropy (BCE) loss on the final response map $\mathcal{M}$. The operation can be written as:

$$\mathcal{L}_{bce}(\mathcal{M}, \mathbf{Y}_s) = BCE(\mathcal{M}, \mathbf{Y}_s), \tag{30}$$

$$\mathbf{Y}_s^i = \mathbb{I}(\sigma(\text{AvgPool}(\mathbf{Y}, \mathcal{K}_i))), \tag{31}$$

where $\text{AvgPool}(\cdot)$ denotes superpoint average pooling operation, $\mathbf{Y}_s^i$ denotes binarized mask value of the i-th superpoint $\mathcal{K}_i$. $\mathbb{I}(\cdot)$ indicates whether the value is higher than 50%.

While BCE loss treats each superpoint separately, it falls short in addressing the issue of foreground-background sample imbalance. To tackle this problem, we can use Dice loss (Milletari, Navab, and Ahmadi 2016):

$$\mathcal{L}_{dice}(\mathcal{M}, \mathbf{Y}_s) = 1 - \frac{2\sum_{i=1}^{\mathcal{N}_s} \mathcal{M}^i \mathbf{Y}_s^i}{\sum_{i=1}^{\mathcal{N}_s} \mathcal{M}^i + \sum_{i=1}^{\mathcal{N}_s} \mathbf{Y}_s^i}. \tag{32}$$

In the STM module, we apply $\mathcal{L}_{rel}$ following (Luo et al. 2022) to supervise the description relevance score s_r with cross-entropy (BCE) loss. The supervision of s_r is based on whether the point belongs to a mentioned category.

In addition, we add a simple auxiliary score loss $\mathcal{L}_{score}$ for proposal quality prediction following (Sun et al. 2023).

Overall, the final training loss function $\mathcal{L}$ can be formulated as:

$$\mathcal{L} = \lambda_{bce}\mathcal{L}_{bce} + \lambda_{dice}\mathcal{L}_{dice} + \lambda_{rel}\mathcal{L}_{rel} + \lambda_{score}\mathcal{L}_{score}, \tag{33}$$

where λ_{bce}, λ_{dice}, λ_{rel} and λ_{score} are hyperparameters used to balance these four losses. Empirically, we set $\lambda_{bce} = \lambda_{dice} = 1, \lambda_{rel} = 5, \lambda_{score} = 0.5$.

4 Experiments

4.1 Experiment Settings

We use the pre-trained Sparse 3D U-Net to extract pointwise features (Sun et al. 2023). Meanwhile, we adopt the pre-trained BERT (Devlin et al. 2018) as text encoder following the settings in (Huang et al. 2021). The rest of the network is trained from scratch. The initial learning rate is 0.0001. We apply learning rate decay at epoch {26, 34, 40} with a rate of 0.5. The number k_{rel} of $\hat{\mathbf{S}}_{rel}$ in STM is set to 512. The default number of multiple rounds L is 6. The batch size is 64, and the maximum sentence length is 80. All experiments are implemented with PyTorch, trained on a single NVIDIA Tesla A100 GPU.

4.2 Dataset

We evaluate our method using the recent 3D referring dataset ScanRefer (Chen, Chang, and Nießner 2020; Huang et al. 2021) which comprises 51,583 natural language expressions that refer to 11,046 objects in 800 ScanNet (Dai et al. 2017) scenes. The evaluation metric is the mean IoU (mIoU) and Acc@kIoU, which means the fraction of descriptions whose predicted mask overlaps the ground truth with IoU$> k$, where $k \in \{0.25, 0.5\}$.

Method	Unique (∼19%)			Multiple (∼81%)			**Overall**			Inference Time		
	0.25	0.5	mIoU	0.25	0.5	mIoU	**0.25**	**0.5**	**mIoU**	Stage-1	Stage-2	All
TGNN (GRU)	-	-	-	-	-	-	35.0	29.0	26.1	-	-	-
TGNN (GRU) †	67.2	54.1	48.7	29.1	23.9	21.8	36.5	29.8	27.0	26139ms	125ms	26264ms
3D-STMN(GRU)	88.3	82.8	73.0	45.5	25.8	29.5	53.8	36.8	38.0	-	-	277ms
TGNN (BERT)	-	-	-	-	-	-	37.5	31.4	27.8	-	-	-
TGNN (BERT) †	69.3	57.8	50.7	31.2	26.6	23.6	38.6	32.7	28.8	26862ms	235ms	27097ms
3D-STMN	**89.3**	**84.0**	**74.5**	**46.2**	**29.2**	**31.1**	**54.6**	**39.8**	**39.5**	-	-	**283ms**

Table 1: The 3D-RES results on ScanRefer, including mIoU and accuracy evaluated by IoU 0.25 and IoU 0.5. † The mIoU and accuracy are reevaluated on our machine.

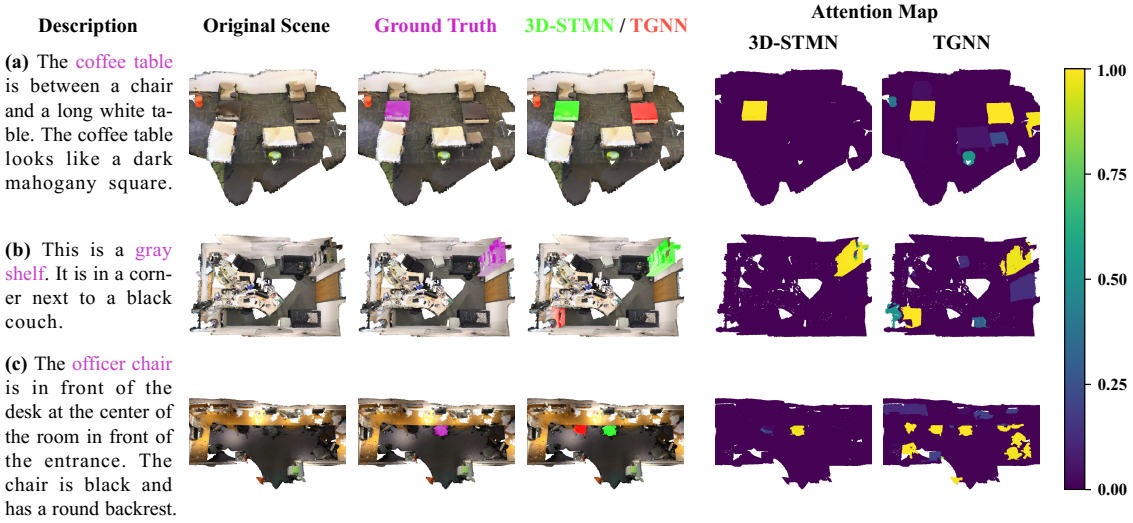

Figure 3: Visualization of the prediction results and attention maps of our 3D-STMN and TGNN. Zoom in for best view.

4.3 Quantitative Comparison

We report the results on the ScanRefer dataset in Tab. 1. Our proposed 3D-STMN achieves state-of-the-art performance by a substantial margin, with an overall improvement of **17.1%**, **8.4%**, **11.7%** in terms of **Acc@0.25**, **Acc@0.5** and **mIoU**, respectively. In terms of speed, our 3D-STMN is **95.7** times faster than the two-stage TGNN (Huang et al. 2021). With an average inference time of **0.3** seconds, our model enables **real-time** applications of 3D-RES. Our 3D-STMN consistently outperforms TGNN, whether using BERT or GRU features, demonstrating its robustness and inference power. In the "Unique" setting, our model boosts Acc@0.25 by 30 points, underscoring its precision with unique objects.

4.4 Ablation Study

STM Mechanism As shown in Tab. 2, under the same settings, the second row (using superpoint-level features) outperforms significantly in all metrics, demonstrating the effectiveness of using superpoints as representations.

Next, in rows 3-6, we added the DDI module. Regardless of the structure of the DDI module, it greatly enhances the performance of the segmentation kernels, leading to significant improvements in all metrics, demonstrating fine-grained discriminability of dependency-driven features.

We also tested three distinct strategies of segmentation kernel in STM: **i)** Root: This employs the embedding of the root node to formulate the segmentation kernel; **ii)** Top1: Leverages the word embedding with the highest score, which is derived by averaging the word-superpoint attention map along the superpoint dimension; **iii)** Average: Utilizes an embedding computed by averaging embeddings of all words. Our findings, presented in Tab. 2, reveal that the Top1 strategy emerges as the most effective due to its innate ability to adapt visually. Consequently, we've chosen this setup for subsequent experiments in our study.

Structure of DDI In Tab. 3, we explored four different versions of the structure of DDI module: **i)** GA (graph-attention only), **ii)** SA - GA (series of self-attention followed by graph-attention), **iii)** GA - SA (series of graph-attention followed by self-attention), and **iv)** GA ‖ SA (graph-attention and self-attention running in parallel).

Our findings reveal that the GA setting, when compared to the absence of the DDI module, brings about a marked enhancement in performance. This underscores the pivotal role of detailed dependency-driven interactions in our model.

After concatenating SA with dense connections on GA (SA - GA and GA - SA), the "Overall" performance of

Method	Superpoint	Segmentation Kernel	Overall 0.5	Overall mIoU
w/o DDI		CLS	22.0	25.6
w/o DDI	✓	CLS	32.9	33.1
3D-STMN	✓	Root	37.3	38.0
3D-STMN	✓	Avg	39.5	38.6
3D-STMN	✓	Top1	**39.8**	**39.5**

Table 2: Ablation study of STM, where "w/o DDI" denotes directly using the [CLS] token to generate segmentation kernel instead of using the proposed DDI module.

DDI Structure	Unique mIoU	Multiple mIoU	Overall 0.25	Overall 0.5	Overall mIoU
w/o DDI	66.2	25.1	46.8	32.9	33.1
GA	72.8	27.6	51.0	36.7	36.4
SA - GA	72.6	29.1	51.0	38.3	37.5
GA - SA	72.7	29.1	50.9	38.9	37.6
GA ‖ SA	**74.5**	**31.1**	**54.6**	**39.8**	**39.5**

Table 3: Ablation study of DDI module, where "w/o DDI" denotes not using the proposed DDI module.

Edge Type	Unique mIoU	Multiple mIoU	Overall 0.25	Overall 0.5	Overall mIoU
Bi-directional	72.7	29.4	51.3	38.7	37.8
Forward	74.2	30.8	54.2	39.3	39.2
Reverse	**74.5**	**31.1**	**54.6**	**39.8**	**39.5**

Table 4: Analyzing the edge direction of Dependency Graph.

the model has improved due to the addition of long-range connections, demonstrating the complementary role of SA in enhancing the effectiveness of the GA structure. Finally, by incorporating parallel self-attention (GA ‖ SA), a notable boost in performance was achieved across all settings. This highlights the efficacy of utilizing a parallel connection while simultaneously supplementing long-range connections, which preserves the explicit modeling capability of the dependency tree and maintains the ordered interaction of information.

Edge Direction of Dependency Graph The ablation of edge direction in the dependency graph is presented in Tab. 4. The findings are as follows: **i)** the **Bi-directional** setting may seem reasonable, but it doubles the number of edge types and leads to overly chaotic information flow, significantly increasing the difficulty of learning. **ii)** The **Forward** direction setting outperforms the bidirectional one, but its top-down information flow from the root node leads to inefficient updates in the upper-level nodes, which are often more relevant to the target object. **iii)** The **Reverse** setting is optimal because it facilitates bottom-up information flow from leaf nodes to higher-level ones. This results in a progressive accumulation of richer information at each level, mirroring the way humans comprehend complex sentences.

Sampling Number	Unique mIoU	Multiple mIoU	Overall 0.25	Overall 0.5	Overall mIoU
64	71.7	28.0	50.9	35.7	36.5
128	72.5	27.9	51.7	36.2	36.6
256	71.9	29.0	50.6	38.3	37.3
512	**74.5**	**31.1**	**54.6**	**39.8**	**39.5**
1024	72.2	29.4	52.0	37.8	37.8
w/o sampling	72.6	29.4	50.8	39.0	37.8

Table 5: Ablation study of sampling number of superpoints, where "w/o sampling" means using all superpoints.

Sampling Number of Superpoints Within the ScanRefer dataset, the count of superpoints highlighted in the description varies. Investigating the optimal sampling number in STM is vital. As displayed in Tab. 5, our model's performance initially rises with increased superpoints, peaking at 512, then declines. Notably, using our sampling strategy yields significantly better results than not sampling at all.

4.5 Qualitative Comparison

In this subsection, we compare our 3D-STMN to TGNN qualitatively on the ScanRefer validation set. Fig. 3 visually shows that our 3D-STMN outperforms TGNN in accurately localizing target objects on attention maps, regardless of the difficulty level of the test samples. The attention generated by 3D-STMN is highly focused and precise. On the other hand, TGNN struggles with discernment, as it assigns high attention values to multiple semantically similar objects, as observed in cases **(a)**, **(b)**, and **(c)**. Notably, when faced with scenes containing multiple objects similar to the target, accompanied by longer and more complex textual descriptions as in cases **(a)** and **(c)**, TGNN fails to distinguish and accurately localize the target, performing no better than random guessing. In contrast, our 3D-STMN can accurately segment these challenging samples. Similar to humans, it focuses subtly but distinctly on objects closely adjacent to the target, distinguishing them from the background.

5 Conclusion

We present 3D-STMN, an efficient and dense-aligned end-to-end method for 3D-RES. By employing the Superpoint-Text Matching (STM) mechanism, our model successfully breaks free from the limitations of the traditional two-stage paradigm. This liberates us to leverage end-to-end dense supervision, harnessing the advantages of precise segmentation and rapid inference speed. Specifically, our model achieves an impressive inference speed of less than 1 second per scene, rendering it well-suited for real-time applications and highly applicable in time-critical scenarios. Furthermore, the proposed Dependency-Driven Interaction (DDI) module substantially enhances our model's comprehension of referring expressions. By explicitly modeling dependency relationships, our model exhibits improved localization and segmentation capabilities, demonstrating a significant advancement in performance.

Acknowledgments

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. U21B2037, No. U22B2051, No. 62072389), the National Natural Science Fund for Young Scholars of China (No. 62302411), China Postdoctoral Science Foundation (No. 2023M732948), the National Natural Science Foundation of China (No. 62072386, No. 62176222, No. 62176223, No. 62176226, No. 62072387, No. 62002305 and No. 62272401), the Natural Science Foundation of Fujian Province of China (No.2021J01002, No.2022J06001), and partially sponsored by CCF-NetEase ThunderFire Innovation Research Funding (NO. CCF-Netease 202301).

References

Achanta, R.; Shaji, A.; Smith, K.; Lucchi, A.; Fua, P.; and Süsstrunk, S. 2012. SLIC superpixels compared to state-of-the-art superpixel methods. *IEEE TPAMI*, 34(11): 2274–2282.

Achlioptas, P.; Abdelreheem, A.; Xia, F.; Elhoseiny, M.; and Guibas, L. 2020. Referit3d: Neural listeners for fine-grained 3d object identification in real-world scenes. In *ECCV*, 422–440. Springer.

Chen, D. Z.; Chang, A. X.; and Nießner, M. 2020. Scanrefer: 3d object localization in rgb-d scans using natural language. In *ECCV*, 202–221. Springer.

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*, 5828–5839.

Deng, C.; Wu, Q.; Wu, Q.; Hu, F.; Lyu, F.; and Tan, M. 2018. Visual grounding via accumulated attention. In *CVPR*, 7746–7755.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv*.

Ding, H.; Liu, C.; Wang, S.; and Jiang, X. 2021. Vision-language transformer and query generation for referring segmentation. In *ICCV*, 16321–16330.

Ding, Z.; Ding, Z.-h.; Hui, T.; Huang, J.; Wei, X.; Wei, X.; and Liu, S. 2022. Ppmn: Pixel-phrase matching network for one-stage panoptic narrative grounding. In *ACM MM*, 5537–5546.

Dwivedi, V. P.; and Bresson, X. 2020. A generalization of transformer networks to graphs. *arXiv*.

Engelmann, F.; Bokeloh, M.; Fathi, A.; Leibe, B.; and Nießner, M. 2020. 3d-mpa: Multi-proposal aggregation for 3d semantic instance segmentation. In *CVPR*, 9031–9040.

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*, 3722–3731.

Graham, B.; Engelcke, M.; and Van Der Maaten, L. 2018. 3d semantic segmentation with submanifold sparse convolutional networks. In *CVPR*, 9224–9232.

Han, L.; Zheng, T.; Xu, L.; and Fang, L. 2020. Occuseg: Occupancy-aware 3d instance segmentation. In *CVPR*, 2940–2949.

He, C.; Li, R.; Li, S.; and Zhang, L. 2022. Voxel set transformer: A set-to-set approach to 3d object detection from point clouds. In *CVPR*, 8417–8427.

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*, 2344–2352.

Hu, R.; Rohrbach, M.; Andreas, J.; Darrell, T.; and Saenko, K. 2017. Modeling relationships in referential expressions with compositional modular networks. In *CVPR*, 1115–1124.

Hu, R.; Rohrbach, M.; and Darrell, T. 2016. Segmentation from natural language expressions. In *ECCV*, 108–124. Springer.

Huang, L.; Wang, H.; Zeng, J.; Zhang, S.; Cao, L.; Ji, R.; Yan, J.; and Li, H. 2023. Geometric-aware Pretraining for Vision-centric 3D Object Detection. *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 *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 1610–1618.

Jain, A.; Gkanatsios, N.; Mediratta, I.; and Fragkiadaki, K. 2022. Bottom up top down detection transformers for language grounding in images and point clouds. In *ECCV*, 417–433. Springer.

Ji, J.; Ma, Y.; Sun, X.; Zhou, Y.; Wu, Y.; and Ji, R. 2022. Knowing what to learn: a metric-oriented focal mechanism for image captioning. *IEEE TIP*, 31: 4321–4335.

Landrieu, L.; and Simonovsky, M. 2018. Large-scale point cloud semantic segmentation with superpoint graphs. In *CVPR*, 4558–4567.

Li, R.; Li, K.; Kuo, Y.-C.; Shu, M.; Qi, X.; Shen, X.; and Jia, J. 2018. Referring image segmentation via recurrent refinement networks. In *CVPR*, 5745–5753.

Liang, Z.; Li, Z.; Xu, S.; Tan, M.; and Jia, K. 2021. Instance segmentation in 3D scenes using semantic superpoint tree networks. In *ICCV*, 2783–2792.

Lin, Y.; Wang, C.; Zhai, D.; Li, W.; and Li, J. 2018. Toward better boundary preserved supervoxel segmentation for 3D point clouds. *ISPRS*, 143: 39–47.

Liu, D.; Zhang, H.; Wu, F.; and Zha, Z.-J. 2019. Learning to assemble neural module tree networks for visual grounding. In *ICCV*, 4673–4682.

Luo, G.; Zhou, Y.; Ji, R.; Sun, X.; Su, J.; Lin, C.-W.; and Tian, Q. 2020a. Cascade grouped attention network for referring expression segmentation. In *ACM MM*, 1274–1282.

Luo, G.; Zhou, Y.; Sun, X.; Cao, L.; Wu, C.; Deng, C.; and Ji, R. 2020b. Multi-task collaborative network for joint referring expression comprehension and segmentation. In *CVPR*, 10034–10043.

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*, 16454–16463.

Ma, Y.; Xu, G.; Sun, X.; Yan, M.; Zhang, J.; and Ji, R. 2022. X-clip: End-to-end multi-grained contrastive learning for video-text retrieval. In *ACM MM*, 638–647.

Ma, Y.; Zhang, X.; Sun, X.; Ji, J.; Wang, H.; Jiang, G.; Zhuang, W.; and Ji, R. 2023. X-Mesh: Towards Fast and Accurate Text-driven 3D Stylization via Dynamic Textual Guidance. In *ICCV*, 2749–2760.

Manning, C. D.; Surdeanu, M.; Bauer, J.; Finkel, J. R.; Bethard, S.; and McClosky, D. 2014. The Stanford CoreNLP natural language processing toolkit. In *ACL*, 55–60.

Margffoy-Tuay, E.; Pérez, J. C.; Botero, E.; and Arbeláez, P. 2018. Dynamic multimodal instance segmentation guided by natural language queries. In *ECCV*, 630–645.

Milletari, F.; Navab, N.; and Ahmadi, S.-A. 2016. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *3DV*, 565–571. Ieee.

Nagaraja, V. K.; Morariu, V. I.; and Davis, L. S. 2016. Modeling context between objects for referring expression understanding. In *ECCV*, 792–807. Springer.

Papon, J.; Abramov, A.; Schoeler, M.; and Worgotter, F. 2013. Voxel cloud connectivity segmentation-supervoxels for point clouds. In *CVPR*, 2027–2034.

Robert, D.; Raguet, H.; and Landrieu, L. 2023. Efficient 3D Semantic Segmentation with Superpoint Transformer. *arXiv*.

Sadhu, A.; Chen, K.; and Nevatia, R. 2019. Zero-shot grounding of objects from natural language queries. In *ICCV*, 4694–4703.

Shi, H.; Li, H.; Meng, F.; and Wu, Q. 2018. Key-word-aware network for referring expression image segmentation. In *ECCV*, 38–54.

Sun, J.; Qing, C.; Tan, J.; and Xu, X. 2023. Superpoint transformer for 3d scene instance segmentation. In *AAAI*, volume 37, 2393–2401.

Tu, W.-C.; Liu, M.-Y.; Jampani, V.; Sun, D.; Chien, S.-Y.; Yang, M.-H.; and Kautz, J. 2018. Learning superpixels with segmentation-aware affinity loss. In *CVPR*, 568–576.

Wang, Y.; Ye, T.; Cao, L.; Huang, W.; Sun, F.; He, F.; and Tao, D. 2022. Bridged transformer for vision and point cloud 3d object detection. In *CVPR*, 12114–12123.

Wu, S.; Fei, H.; Cao, Y.; Bing, L.; and Chua, T.-S. 2023a. Information Screening whilst Exploiting! Multimodal Relation Extraction with Feature Denoising and Multimodal Topic Modeling. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 14734–14751.

Wu, Y.; Cheng, X.; Zhang, R.; Cheng, Z.; and Zhang, J. 2023b. EDA: Explicit Text-Decoupling and Dense Alignment for 3D Visual Grounding. In *CVPR*, 19231–19242.

Yang, S.; Li, G.; and Yu, Y. 2020. Graph-structured referring expression reasoning in the wild. In *CVPR*, 9952–9961.

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*, 18155–18165.

Yang, Z.; Zhang, S.; Wang, L.; and Luo, J. 2021. Sat: 2d semantics assisted training for 3d visual grounding. In *ICCV*, 1856–1866.

Ye, L.; Rochan, M.; Liu, Z.; and Wang, Y. 2019. Cross-modal self-attention network for referring image segmentation. In *CVPR*, 10502–10511.

Yu, L.; Lin, Z.; Shen, X.; Yang, J.; Lu, X.; Bansal, M.; and Berg, T. L. 2018. Mattnet: Modular attention network for referring expression comprehension. In *CVPR*, 1307–1315.

Yu, L.; Poirson, P.; Yang, S.; Berg, A. C.; and Berg, T. L. 2016. Modeling context in referring expressions. In *ECCV*, 69–85. Springer.

Yu, L.; Tan, H.; Bansal, M.; and Berg, T. L. 2017. A joint speaker-listener-reinforcer model for referring expressions. In *CVPR*, 7282–7290.

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*, 1791–1800.

Zhang, A.; Fei, H.; Yao, Y.; Ji, W.; Li, L.; Liu, Z.; and Chua, T. 2023. Transfer Visual Prompt Generator across LLMs. *CoRR*, abs/2305.01278.

Zhao, L.; Cai, D.; Sheng, L.; and Xu, D. 2021. 3DVG-Transformer: Relation modeling for visual grounding on point clouds. In *ICCV*, 2928–2937.

Zhao, Y.; Fei, H.; Ji, W.; Wei, J.; Zhang, M.; Zhang, M.; and Chua, T.-S. 2023. Generating Visual Spatial Description via Holistic 3D Scene Understanding. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 7960–7977.

Zhuang, B.; Wu, Q.; Shen, C.; Reid, I.; and Van Den Hengel, A. 2018. Parallel attention: A unified framework for visual object discovery through dialogs and queries. In *CVPR*, 4252–4261.