3D-SPS: Single-Stage 3D Visual Grounding via Referred Point Progressive Selection

Junyu Luo[1,2] *, Jiahui Fu[1,2] *, Xianghao Kong[1,2] , Chen Gao[1,2] †,
Haibing Ren[3] , Hao Shen[3] , Huaxia Xia[3] , Si Liu[1,2]

[1]Institute of Artificial Intelligence, Beihang University [2]Hangzhou Innovation Institute, Beihang University [3]Meituan Inc.

Abstract

*3D visual grounding aims to locate the referred target object in 3D point cloud scenes according to a free-form language description. Previous methods mostly follow a two-stage paradigm, i.e., language-irrelevant detection and cross-modal matching, which is limited by the isolated architecture. In such a paradigm, the detector needs to sample keypoints from raw point clouds due to the inherent properties of 3D point clouds (irregular and large-scale), to generate the corresponding object proposal for each keypoint. However, sparse proposals may leave out the target in detection, while dense proposals may confuse the matching model. Moreover, the language-irrelevant detection stage can only sample a small proportion of keypoints on the target, deteriorating the target prediction. In this paper, we propose a **3D S**ingle-Stage Referred Point **P**rogressive **S**election (3D-SPS) method, which progressively selects keypoints with the guidance of language and directly locates the target. Specifically, we propose a Description-aware Keypoint Sampling (DKS) module to coarsely focus on the points of language-relevant objects, which are significant clues for grounding. Besides, we devise a Target-oriented Progressive Mining (TPM) module to finely concentrate on the points of the target, which is enabled by progressive intra-modal relation modeling and inter-modal target mining. 3D-SPS bridges the gap between detection and matching in the 3D visual grounding task, localizing the target at a single stage. Experiments demonstrate that 3D-SPS achieves state-of-the-art performance on both ScanRefer and Nr3D/Sr3D datasets.*

1. Introduction

Visual Grounding (VG) aims to localize the target object in the scene based on an object-related linguistic description. In recent years, the 3D VG task has received increasing attention owing to its wide applications, such as autonomous robots and human-machine interaction in AR/VR/Metaverse. Even though much progress [29, 33–38, 40, 41, 43] has been achieved in the 2D VG task, it is still challenging to locate the referred target object in 3D scenes since point clouds are irregular and large-scale.

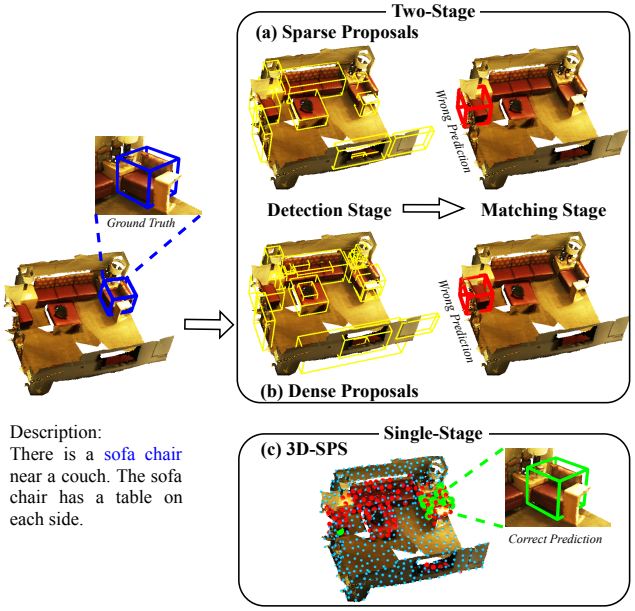

Figure 1. Traditional two-stage 3D VG methods are limited by the isolation of the detection stage and the matching stage. (a) Sparse proposals may leave out the target in detection. (b) Dense proposals could confuse the matching model. (c) 3D-SPS progressively selects keypoints (blue points→red points→green points) and performs referring at a single stage. Noted that dense surfaces are utilized only to help readers understand the example 3D scene, while the input of our method only contains sparse point clouds.

Existing 3D VG methods [2, 7, 11, 39, 42, 44] are mainly based on the *detection-then-matching* two-stage pipeline. The first stage is language-irrelevant detection, where general 3D object detectors [4, 20, 23] are adopted to produce numerous object proposals. The second stage is cross-modal matching, where specific vision-language attention

*Equal contribution
†Corresponding author: *Chen Gao*.

mechanisms are usually designed to match the proposal and the description. Previous methods primarily focus on the second stage, *i.e.*, exploring relations among proposals to distinguish the target object.

We argue that the separation of the two stages limits the existing methods. Previous 2D detection methods adopt data-independent anchor boxes as proposals on regular and well-organized images. However, the anchor-based fashion is generally impractical for the large-scale and irregular 3D point clouds. Consequently, the 3D detector utilized in the first stage needs to sample a limited number of keypoints to represent the whole scene and generate the corresponding proposal for each keypoint. However, sparse proposals may leave out the target in the detection stage (*e.g.*, the *sofa chair* in Figure 1 (a)), which leads to the inability to locate the target in the matching stage. Meanwhile, dense proposals may contain redundant objects, causing the inter-proposal relationship so complex that the matching module struggles to distinguish the target. As shown in Figure 1 (b), it is difficult to select the right *sofa chair* from these numerous proposals with similar appearances. Therefore, the two-stage grounding methods face a dilemma of deciding the proposal number. Besides, the keypoint sampling strategy (*e.g.*, Farthest Point Sampling (FPS) [25]) usually adopted in the detector at the first stage is also language-irrelevant. The strategy aims to sample keypoints to cover the entire scene as much as possible to detect all potential objects. Thus, the proportion of target keypoints is relatively small, which is unfavorable for the target prediction.

To address the aforementioned issues, we propose a **3D S**ingle-Stage Referred Point **P**rogressive **S**election (3D-SPS) method in this paper. Our main idea is to progressively select keypoints under the guidance of the language description throughout the whole process, as shown in Figure 1 (c). Based on this idea, we propose a Description-aware Keypoint Sampling (DKS) module to coarsely focus on the points of language-relevant objects, *e.g.*, *sofa chair, couch, and table* in Figure 1 (c). These keypoints provide significant clues for localizing the grounding target in the following cross-modal interaction. Besides, we devise a Target-oriented Progressive Mining (TPM) module, which conducts progressive mining to finely figure out the target. We leverage the self/cross-attention mechanism to model intra/inter-modal relationships respectively. In addition, we fuse the keypoint features with point features of the whole scene to achieve global localization perception. To progressively select keypoints of the target, we utilize the language-points cross-attention map to select the keypoints that the language pays more attention to and discard irrelevant points. The model gradually concentrates on the target and obtains a condensed set of keypoints through multiple layers. Thus, the proportion of target points will gradually increase with richer target-related features, which benefits the target box regression. Finally, 3D-SPS distinguishes the target from the condensed keypoint set and regresses its bounding box. Note that 3D-SPS is also consistent with the commonsense of how human finds the target object. Commonly, a human first selects a coarse candidate set according to the language description and then finely recognize and judge it to select the target object. [16, 31]

In summary, we make the following contributions:

- We propose the 3D-SPS method, which directly performs 3D VG at a single stage to bridge the gap between detection and matching. To the best of our knowledge, 3D-SPS is the first work investigating single-stage 3D VG.
- We treat the 3D VG task as a keypoint selection problem. Two selection modules, *i.e.*, DKS and TPM, are designed to progressively select target-related keypoints. DKS samples the coarse language-relevant keypoints, and TPM finely mines the cross-modal relationship to distinguish the target.
- Extensive experiments confirm the effectiveness of our method. 3D-SPS achieves the state-of-the-art performance on both *ScanRefer* [2] and *Nr3D/Sr3D* [1] datasets. The code is provided in https://github.com/fjhzhixi/3D-SPS.

2. Related Work

Visual Grounding on 2D Images. The goal of visual grounding on 2D images is to select a referred target according to the referring expression [8, 14, 22, 40]. Two mainstream frameworks have been proposed in succession: two-stage and one-stage methods. Specifically, two-stage methods [13, 19, 33–36, 40, 41, 43, 46] first generate region proposals with object detectors and then select the target region by matching the language features with the proposals. Each proposal is treated the same in the matching stage, despite their importance in the referring context varies. Besides, one-stage methods [3, 6, 17, 29, 37, 38] eliminate the proposal generation and feature extraction stage in two-stage frameworks. In these methods, linguistic features are densely fused with each pixel or patch to generate multi-modal feature maps for regressing the bounding box.

However, one-stage methods in 2D VG could not be directly lifted to 3D VG. Firstly, 3D point clouds are numerous and noisy. Therefore, it is computationally unacceptable [9, 10, 45] to treat each point as a candidate. Then, due to the large-scale and complexity of 3D scenes, it is not easy to model the relationship of all objects and figure out the target [11, 39, 44]. Moreover, 2D one-stage methods adopt the sliding-window manner like [12, 30], which cannot deal with 3D points since 2D input is highly regular while 3D points are inherently sparse, unordered, and

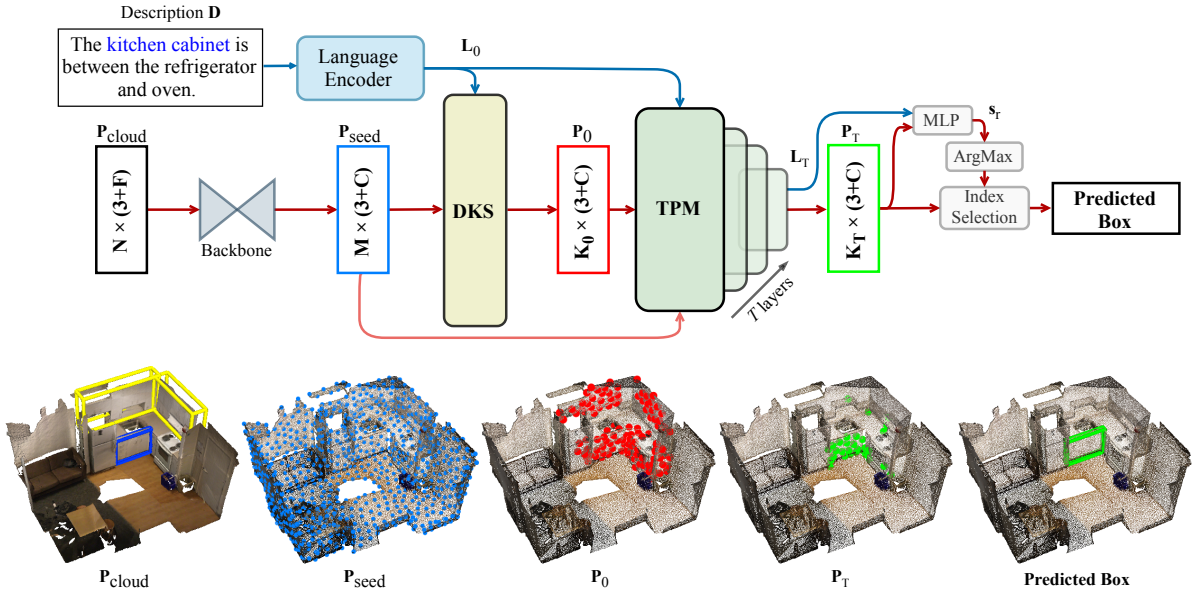

Figure 2. **3D-SPS framework.** We take the 3D VG task as a keypoint selection problem and avoid the separation of detection and matching. Specifically, we use PointNet++ as the backbone to extract point seeds $\mathbf{P}_{seed}$ from $\mathbf{P}_{cloud}$. After that, we coarsely sample the language-relevant keypoints $\mathbf{P}_0$ by DKS with word features $\mathbf{L}_0$, which are mostly on the *kitchen cabinets*, *refrigerator* and *oven* in the figure. Then, TPM finely selects target keypoints $\mathbf{P}_T$ and predict referring confidence scores $\mathbf{s}_r$. Here the keypoints are concentrated on the target *kitchen cabinet*. Finally, the target box is regressed from the keypoint with the highest $\mathbf{s}_r$ in $\mathbf{P}_T$. The blue box is the ground truth. The yellow boxes are objects of the same category as the target. The green box is our target prediction. Best viewed in color.

irregular [24, 25]. In this paper, we propose 3D-SPS to address the problems introduced by 3D point clouds, which becomes the leading 3D VG solution.

Visual Grounding on 3D Point Clouds. With the prevalence of deep learning technologies on 3D point clouds, the 3D VG task has attracted much attention. Chen *et al.* [2] released a 3D VG dataset *ScanRefer*, in which the bounding boxes of objects are referred by their corresponding descriptions in an indoor scene. ReferIt3D [1] also proposes two datasets, *i.e.*, *Sr3D* and *Nr3D*, for the 3D VG task.

Existing 3D VG works [2, 7, 11, 15, 28, 39, 42, 44] mainly focus on better modeling the relationship among objects to locate the target object, *e.g.*, adopting graph neural network [15], and attention mechanisms [44]. To the best of our knowledge, previous 3D grounding approaches can generally be concluded into a detection-then-matching two-stage framework. In these methods, the detection stage fails to leverage the language context to concentrate on the points that are more essential to the referring task. To overcome those shortcomings, we propose the first single-stage method in 3D VG to progressively select keypoints under the guidance of the description.

3. Method

In this section, we detail the 3D-SPS method. In Sec 3.1, we present an overview of 3D VG task and our method. In Sec 3.2 and Sec 3.3, we dive into the technical details and how we obtain the target by progressive keypoint selection. In Sec 3.4, we introduce the training objectives of 3D-SPS.

3.1. Overview

In the 3D VG task, the inputs are the point clouds $\mathbf{P}_{cloud} \in \mathbb{R}^{N\times(3+F)}$ and a free-form plain text description $\mathbf{D}$ of the target object with W words, where $\mathbf{P}_{cloud}$ contains 3D coordinates and F-dimensional auxiliary feature (RGB, normal vectors, *etc.*) of N points. The goal of this task is to locate the target object (*i.e.*, the most relevant object to the description) and predict its bounding box.

The main idea of 3D-SPS is the progressive keypoint selection process, as shown in Figure 2. *Firstly*, we adopt a widely used PointNet++ [25] as the backbone network to extract point features from $\mathbf{P}_{cloud}$. The backbone outputs M seed points with (x, y, z) coordinates and C-dimensional enriched local features $\mathbf{P}_{seed} \in \mathbb{R}^{M\times(3+C)}$. Meanwhile, we use the language encoder to extract H-dimensional word features $\mathbf{L}_0 \in \mathbb{R}^{W\times H}$ from W-length description $\mathbf{D}$. *Secondly*, DKS module selects K_0 language-relevant keypoints with features $\mathbf{P}_0 \in \mathbb{R}^{K_0\times(3+C)}$ from M seed points based on word features $\mathbf{L}_0$. These keypoints belong to the objects whose categories are mentioned in the description, providing significant clues to distinguishing the grounding target. *Thirdly*, TPM module takes point features $\mathbf{P}_0$ and word features $\mathbf{L}_0$ as inputs. The t-th layer of the TPM module takes $\mathbf{P}_{t-1}$ and $\mathbf{L}_{t-1}$ as inputs and outputs $\mathbf{P}_t$ and $\mathbf{L}_t$. TPM progressively distinguishes the grounding

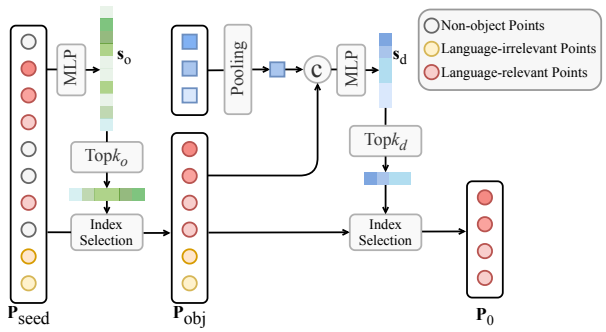

Figure 3. **The DKS module.** We use object confidence score $\mathbf{s}_o$ to select points near object centers and description relevance score $\mathbf{s}_d$ to select language-relevant points.

target by multi-layer cross-modal transformers. We select K_T keypoints with features $\mathbf{P}_T \in \mathbb{R}^{K_T \times (3+C)}$ and update the word features as $\mathbf{L}_T$. *Lastly*, we predict the referring confidence score $\mathbf{s}_r$ based on keypoint features $\mathbf{P}_T$ and cross-modally aligned word features $\mathbf{L}_T$ by a simple MLP head. The keypoint feature with the highest $\mathbf{s}_r$ is used to regress the bounding box of the grounding target as the center $\mathbf{c} \in \mathbb{R}^3$ and the size $\mathbf{s} \in \mathbb{R}^3$.

By treating the 3D VG task as a keypoint selection problem, our 3D-SPS concentrates on distinguishing the keypoints of the target object from point clouds for predicting the bounding box directly, which is more effective than traditional detection-then-matching two-stage methods.

3.2. Description-aware Keypoint Sampling

Since the search space of 3D anchor boxes is huge, the data-independent anchor assignment strategy widely adopted in 2D object detection [27] is impractical when lifted to 3D [20]. To this end, most 3D object detection methods [4, 20, 23] usually adopt sampling methods (*e.g.*, FPS [25]) to sample keypoints from seed points and generate a proposal for each selected point. Existing detection-then-matching methods for the 3D VG task usually use the same strategy at the detection stage. However, directly adopting the sampling strategy in detection to the 3D VG task is not sensible because of the divergence of interest of the two tasks. The sampling objective of 3D object detection is to cover the entire scene as much as possible for detecting potential objects, while the goal of 3D VG is to locate the referred target.

Therefore, we propose DKS to help the model focus on the keypoints of language-relevant objects instead of the whole scene. Specifically, we bring word features into the sampling process to select keypoints of the objects whose categories are mentioned in the description. These keypoints contain the information of not only the target object but also related objects to help determine the target.

Figure 3 details the DKS. We first obtain an object confidence score $\mathbf{s}_o$ based on point features $\mathbf{P}_{seed}$ to clarify

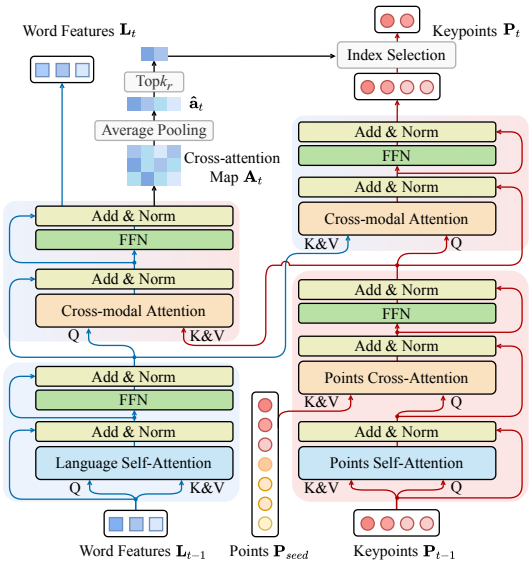

Figure 4. **The TPM module.** It is a two-stream cross-modal transformer model. We select the keypoints of the target based on the language-points cross-attention map $\mathbf{A}_t$ at the t-th layer.

whether the point is near an object center. The keypoint features $\mathbf{P}_{obj}$ with top k_o highest $\mathbf{s}_o$ are selected as:

$$\begin{aligned} \mathbf{s}_o &= \texttt{MLP}(\mathbf{P}_{seed}), \\ \mathbf{P}_{obj} &= \mathbf{P}_{seed}\left[\texttt{argtopk}(\mathbf{s}_o, k_o)\right]. \end{aligned} \tag{1}$$

Then a description relevance score $\mathbf{s}_d$ is utilized to select top k_d keypoints as $\mathbf{P}_0$ that are related to the description context $\mathbf{L}_0$. We jointly use point features $\mathbf{P}_{obj}$ and global word features to predict the $\mathbf{s}_d$ of each point, which can be formulated as:

$$\begin{aligned} \mathbf{s}_d &= \texttt{MLP}(\mathbf{P}_{obj} \,||\, \texttt{MaxPool}(\mathbf{L}_0)), \\ \mathbf{P}_0 &= \mathbf{P}_{obj}\left[\texttt{argtopk}(\mathbf{s}_d, k_d)\right]. \end{aligned} \tag{2}$$

3.3. Target-oriented Progressive Mining

With the coarsely selected language-relevant keypoints by DKS, we perform fine target mining with the TPM module. TPM is constructed by a T-layer stacked multi-modal two-stream transformer model, where both word features and keypoint features are processed in separate streams and interact through cross-modal attention layers to model the relationship and mine the target. At the t-th layer, TPM selects $\mathbf{P}_t$ from $\mathbf{P}_{t-1}$. TPM progressively selects the keypoints and concentrates the attention by discarding target-irrelevant keypoints in each layer.

Intra/inter-modal Modeling. As Figure 4 shows, we employ the attention mechanism [32] to learn intra-modal relationships. For point features, the point self-attention block helps to refine point visual features and exploits their spatial relationship. For word features, the language self-attention block is used to extract context relationships.

Specially, we leverage a point cross-attention block to model the global location of keypoints in the scene because the interaction of selected keypoints could not well model descriptions which include the global location like "*in the center/corner of room*". Therefore, the scene point clouds $\mathbf{P}_{seed}$ (point features before DKS) are fused to acquire global scene features.

Next, point features and word features interact in cross-modal attention blocks. In these blocks, the points branch is assisted by word features to distinguish the target, while the language branch fuses the scene information by attending to point features.

Attention-guided Keypoint Selection. TPM reduces the keypoint set at each layer and gradually focuses on the target, as shown in Figure 4. We make use of the language-points cross-attention map $\mathbf{A}_t$, which represents the importance of keypoints to the referring task. Specifically, we perform average pooling on $\mathbf{A}_t$ and obtain point-wise attention scores $\hat{\mathbf{a}}_t \in \mathbb{R}^{K_{t-1}}$. Then the keypoints with top k_r highest $\hat{\mathbf{a}}_t$ are selected for the next layer as follow:

$$\begin{aligned} \hat{\mathbf{a}}_t &= \texttt{AvgPool}(\mathbf{A}_t), \\ \mathbf{P}_t &= \mathbf{P}_{t-1}\left[\texttt{argtopk}(\hat{\mathbf{a}}_t, k_r)\right]. \end{aligned} \tag{3}$$

3.4. Training Objectives

Visual Grounding Loss. 3D VG loss $\mathcal{L}_{\text{VG}}$ is the primary loss of our framework. In the training phase, we supervise referring confidence scores $\mathbf{s}_r$ predicted from $\mathbf{P}_T$ with the target label. During inference, we only choose the keypoint with the highest $\mathbf{s}_r$ from $\mathbf{P}_T$ to predict the target box. We adapt the loss in ScanRefer [2] to our framework. In ScanRefer, the target label of $\mathbf{s}_r$ is a one-hot label. The keypoint whose proposal box has the highest IoU with the ground truth target box is set to 1, and others are set to 0. However, in 3D-SPS, we usually obtain several feasible keypoints of the target after TPM since the model aims to select points on it. Therefore, we modify this target label from one-hot to multi-hot. Specifically, we assign 1 to keypoints whose predicted boxes' IoUs with the ground truth target box are the top k_1 highest and greater than the threshold θ.

DKS Loss. In the DKS module, we apply $\mathcal{L}_{\text{DKS}}$ to supervise the object confidence score $\mathbf{s}_o$ and the description relevance score $\mathbf{s}_d$ with Focal Loss [18]. The $\mathbf{s}_o$ is supervised by whether the point is inside an object box and belongs to the k_2-closest points to the object center. The $\mathbf{s}_d$ is supervised by whether the point belongs to any object whose category is mentioned in the description.

Detection Loss. Following the loss used in [20, 23], we use the object detection loss $\mathcal{L}_{\text{Det}}$ as an auxiliary loss for VG task. Specifically, $\mathcal{L}_{\text{Det}}$ comprises object semantic classification loss $\mathcal{L}_{\text{Cls}}$, objectness binary classification loss $\mathcal{L}_{\text{Obj}}$, center offset regression loss $\mathcal{L}_{\text{Center}}$, and bounding box regression loss $\mathcal{L}_{\text{Box}}$. In the training phase, we supervise the box of objects predicted by all keypoints of each TPM layer. During inference, we only use the box prediction of the keypoint with the highest $\mathbf{s}_r$ from the last TPM layer as our predicted grounding target.

Language Classification Loss. Following [2], we also introduce the language classification loss $\mathcal{L}_{\text{Lang}}$ as an auxiliary loss, which includes a multi-class object classification loss for the target category based on the updated language features of each TPM layer.

In summary, the total loss is: $\mathcal{L} = \alpha_1\mathcal{L}_{\text{VG}} + \alpha_2\mathcal{L}_{\text{DKS}} + \alpha_3\mathcal{L}_{\text{Det}} + \alpha_4\mathcal{L}_{\text{Lang}}$, where the weights $\alpha_1, \alpha_2, \alpha_3, \alpha_4$ are used for balancing different loss terms.

4. Experiments

4.1. Datasets

ScanRefer. The *ScanRefer* dataset [2] is a 3D visual grounding dataset with $51,583$ descriptions based on the 800 *ScanNet* [5] scenes. Each scene has 13.81 objects and 64.48 descriptions on average. The evaluation metric of the dataset is the Acc@mIoU, which means the fraction of descriptions whose predicted box overlaps the ground truth with IoU $> m$, where $m \in \{0.25, 0.5\}$. The accuracy is reported in *unique* and *multiple* categories. Specifically, a target object is classified as *unique* if it is the only object of its class in the scene; otherwise, it is classified as *multiple*.

Nr3D and Sr3D. The *ReferIt3D* dataset [1] is also based on the *ScanNet* [5] scenes. It contains two subsets: *Sr3D* and *Nr3D*. *Sr3D* (Spatial Reference in 3D) contains $83,572$ synthetic expressions generated by templates and *Nr3D* (Natural Reference in 3D) consists of $41,503$ human expressions. It directly provides segmented point clouds for each object as inputs rather than the whole scene. The evaluation metric of *ReferIt3D* is the accuracy, *i.e.*, whether the model correctly selects the target among objects.

4.2. Implementation Details

Our model is trained end-to-end with the AdamW optimizer [21] and a batch size of 32 for 32 epochs. The initial learning rates of TPM layers and the rest of the model are empirically set to $1e-4$ and $1e-3$, respectively. We apply learning rate decay at epoch $\{16, 24, 28\}$ with a rate of 0.1. We adopt the pre-trained PointNet++ [25] following the settings in [20] and the language encoder in [26], while the rest of the network is trained from scratch. For the *ScanRefer* dataset, we use xyz coordinates, RGB values, normal vectors, and extracted multiview features as inputs following [2]. The number M of $\mathbf{P}_{seed}$ is empirically set to 1024. The number K_0 of $\mathbf{P}_0$ is empirically set to 512. The number T of TPM layers is set to 4, and we select 50% keypoints in each layer, *i.e.*, $\{K_t | t \in \{1,2,3,4\}\} = \{256, 128, 64, 32\}$. The loss weights are empirically set to $\alpha_1 = 0.1$, $\alpha_2 = 0.8$, $\alpha_3 = 5$, $\alpha_4 = 0.1$ for balancing terms. We set k_1 to 4, θ to

Method	Pub.	Input	Unique		Multiple		Overall	
			Acc@0.25	Acc@0.5	Acc@0.25	Acc@0.5	Acc@0.25	Acc@0.5
SCRC [14]	CVPR16	2D only	24.03	9.22	17.77	5.97	18.70	6.45
One-stage [38]	ICCV19	2D only	29.32	22.82	18.72	6.49	20.38	9.04
ScanRefer [2]	ECCV20	3D only	67.64	46.19	32.06	21.26	38.97	26.10
TGNN [15]	AAAI21	3D only	68.61	56.80	29.84	23.18	37.37	29.70
IntanceRefer [42]	ICCV21	3D only	77.45	**66.83**	31.27	24.77	40.23	32.93
SAT [39]	ICCV21	3D only	73.21	50.83	37.64	25.16	44.54	30.14
3DVG-Transformer [44]	ICCV21	3D only	77.16	58.47	38.38	28.70	45.90	34.47
3D-SPS (Ours)	-	3D only	**81.63**	64.77	**39.48**	**29.61**	**47.65**	**36.43**
ScanRefer [2]	ECCV20	2D + 3D	76.33	53.51	32.73	21.11	41.19	27.40
InstanceRefer [42]	ICCV21	2D + 3D	75.72	64.66	29.41	22.99	38.40	31.08
3DVG-Transformer [44]	ICCV21	2D + 3D	81.93	60.64	39.30	28.42	47.57	34.67
3D-SPS (Ours)	-	2D + 3D	**84.12**	**66.72**	**40.32**	**29.82**	**48.82**	**36.98**

Table 1. **Comparison on *ScanRefer*.** The *unique* stands for samples with no distracting objects and *multiple* for remaining samples. We measure the percentage of predictions whose IoU with the ground truth is greater than $\{0.25, 0.5\}$.

Method	Pub.	Easy	Hard	View-dep.	View-indep.	Overall
Nr3D						
ReferIt3DNet [1]	ECCV20	43.6% ± 0.8%	27.9% ± 0.7%	32.5% ± 0.7%	37.1% ± 0.8%	35.6% ± 0.7%
TGNN [15]	AAAI21	44.2% ± 0.4%	30.6% ± 0.2%	35.8% ± 0.2%	38.0% ± 0.3%	37.3% ± 0.3%
IntanceRefer [42]	ICCV21	46.0% ± 0.5%	31.8% ± 0.4%	34.5% ± 0.6%	41.9% ± 0.4%	38.8% ± 0.4%
3DVG-Transformer [44]	ICCV21	48.5% ± 0.2%	34.8% ± 0.4%	34.8% ± 0.7%	43.7% ± 0.5%	40.8% ± 0.2%
LanguageRefer [28]	CoRL21	51.0%	36.6%	41.7%	45.0%	43.9%
SAT [39]	ICCV21	56.3% ± 0.5%	42.4% ± 0.4%	46.9% ± 0.3%	50.4% ± 0.3%	49.2% ± 0.3%
3D-SPS (Ours)	-	**58.1% ± 0.3%**	**45.1% ± 0.4%**	**48.0% ± 0.2%**	**53.2% ± 0.3%**	**51.5% ± 0.2%**
Sr3D						
ReferIt3DNet [1]	ECCV20	44.7% ± 0.1%	31.5% ± 0.4%	39.2% ± 1.0%	40.8% ± 0.1%	40.8% ± 0.2%
TGNN [15]	AAAI21	48.5% ± 0.2%	36.9% ± 0.5%	45.8% ± 1.1%	45.0% ± 0.2%	45.0% ± 0.2%
IntanceRefer [42]	ICCV21	51.1% ± 0.2%	40.5% ± 0.3%	45.4% ± 0.9%	48.1% ± 0.3%	48.0% ± 0.3%
3DVG-Transformer [44]	ICCV21	54.2% ± 0.1%	44.9% ± 0.5%	44.6% ± 0.3%	51.7% ± 0.1%	51.4% ± 0.1%
LanguageRefer [28]	CoRL21	**58.9%**	49.3%	49.2%	56.3%	56.0%
SAT [39]	ICCV21	-	-	-	-	57.9% ± 0.1%
3D-SPS (Ours)	-	56.2% ± 0.6%	**65.4% ± 0.1%**	**49.2% ± 0.5%**	**63.2% ± 0.2%**	**62.6% ± 0.2%**

Table 2. **Comparison on *Nr3D* and *Sr3D*.** *Easy* samples contain no distractor, and the remaining belong to *Hard*. *View-dep./View-indep.* refer to whether the description is dependent or independent on the camera view.

0.25 in $\mathcal{L}_{\text{VG}}$, and k_2 to 5 in $\mathcal{L}_{\text{DKS}}$. All experiments are implemented with PyTorch on a single NVIDIA V100 GPU. More implementation details on the *ReferIt3D* dataset can be obtained in the supplementary material.

4.3. Quantitative Comparison

In Table 1 and 2, we compare 3D-SPS with existing 3D VG works on *ScanRefer* and *Nr3D/Sr3D* datasets. The methods involved are 2D-based methods SCRC [14] and One-stage [38], the segmentation-based two-stage methods TGNN [15] and InstanceRefer [42], the detection-based two-stage methods SAT [39], 3DVG-Transformer [44], ScanRefer [2], and ReferIt3DNet [1].

ScanRefer. 3D-SPS outperforms the existing methods by a large margin, as shown in Table 1. In the *Input* column, *3D only* stands for *xyz + RGB + normals*, and *2D + 3D* means an extra 128-dimensional *multiview* feature for each point is added to *3D only*. We concatenate these multiview features with our point features from the backbone and feed them into TPM together. In the *3D only* setting, 3D-SPS has improved by $+1.96\%$ at Acc@0.5 and $+1.75\%$ at Acc@0.25 compared to the existing state-of-the-art methods. In the *2D+3D* setting, 3D-SPS outperforms the existing methods by 2.31% at Acc@0.5 and 1.25% at Acc@0.25.

Note that TGNN and InstanceRefer both rely on a pre-fixed 3D instance segmentation model. Thus InstanceRefer performs better on the Acc@0.5 score in the *Unique* subset.

Nr3D & Sr3D. The task of the *ReferIt3D* dataset (*Nr3D* & *Sr3D*) is to identify the target object among the given ground truth object bounding boxes. We modify 3D-SPS

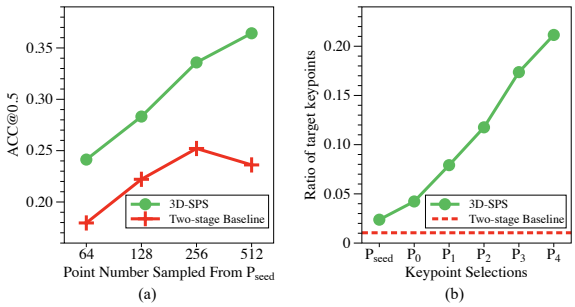

Figure 5. **Effectiveness Validation.** (a) As the point number sampled from $\mathbf{P}_{seed}$ increases, our 3D-SPS performs better. The performance of the two-stage baseline first increases and then decreases. (b) As the progressive language-relevant keypoint selection goes, the ratio of target keypoints in our 3D-SPS increases after each selection. Also, this ratio keeps outperforming language-irrelevant sampling (*e.g.*, FPS) used in the two-stage baseline.

	Acc@0.25	Acc@0.5
FPS	43.83	31.88
DKS (w/o $\mathbf{s}_d$)	46.15	34.95
DKS (w/o $\mathbf{s}_o$)	46.06	35.19
DKS	**47.65**	**36.43**

Table 3. Ablations on the sampling strategy of DKS.

T	1	2	3	4	5
Acc@0.25	45.37	45.99	46.48	**47.65**	47.02
Acc@0.5	33.13	33.97	34.53	**36.43**	36.07

Table 4. Ablations on the layer number T in TPM.

accordingly, removing DKS and only verifying the effectiveness of TPM. For fair comparisons, we adopt 2D semantic assisted training proposed by SAT [39] in the training process and only use 3D inputs in the inference process. Results in Table 2 show progressive selection is effective for referring tasks. 3D-SPS significantly improves the grounding accuracy by $+2.3\%$ in *Nr3D* and $+4.7\%$ in *Sr3D*. Although LanguageRefer performs better on the *Easy* subset of the synthetic dataset *Sr3D*, 3D-SPS outperforms it by a large margin on the more challenging *Hard* subset.

Effectiveness Validation. Figure 5 confirms that our main idea, *i.e.*, progressive keypoint selection, can address the issues from the motivation in Sec. 1. We analyze 3D-SPS and the two-stage method baseline [2] on the entire validation set of *ScanRefer*. As shown in Figure 5 (a), the two-stage baseline faces the dilemma of the point number sampled from $\mathbf{P}_{seed}$. In contrast, 3D-SPS benefits from more sampled points. According to Figure 5 (b), the two-stage baseline is limited by the small ratio of target keypoints due to the language-irrelevant keypoint sampling, while the ratio in 3D-SPS increases significantly after each selection.

Keypoints Num	w/o selection					w/ selection
	32	64	128	256	512	$512 \rightarrow 32$
Acc@0.25	42.06	44.77	46.30	46.38	46.09	**47.65**
Acc@0.5	31.89	33.88	34.99	35.53	34.98	**36.43**

Table 5. Ablations of TPM on whether to select keypoints and different keypoint numbers. Our default setting is *w/ selection*, where we progressively select keypoints from 512 to 32.

4.4. Ablation Study

In this subsection, we investigate the contribution of the proposed DKS and TPM module. We take *ScanRefer* as an example and report the *Overall* accuracy in *3D only* setting.

Sampling Strategy of DKS. Table 3 shows the ablations of sampling strategy in the DKS module. FPS [25] is a widely adopted point sampling method, which makes an effort to cover the whole scene without special attention to the language-relevant points. *DKS (w/o* $\mathbf{s}_d$*)* means that only the object confidence score $\mathbf{s}_o$ is utilized, and *DKS (w/o* $\mathbf{s}_o$*)* represents that only the description relevance score $\mathbf{s}_d$ is used. *DKS* means that both $\mathbf{s}_o$ and $\mathbf{s}_d$ are adopted and is the full version of the proposed DKS module. According to the results in Table 3, $\mathbf{s}_o$ and $\mathbf{s}_d$ are both beneficial to the referring task, helping DKS select description-related keypoints near object centers. The joint use of $\mathbf{s}_o$ and $\mathbf{s}_d$ can produce promising results.

Layer Number of TPM. We investigate the performance on different TPM layer numbers $T \in \{1, 2, 3, 4, 5\}$. As shown in Table 4, more TPM layers bring higher accuracy, which demonstrates that TPM and the progressive mining are essential to grounding. We take $T = 4$ as the default setting since more layers might force the model to leave out some keypoints of the target object and miss the best bounding box.

Progressive Selection of TPM. To further confirm the effectiveness of progressive keypoint selection, we compare the results on whether to adopt keypoint selection, as shown in Table 5. In detail, for the *w/o selection* setting, we only conduct multi-modal self/cross-attention. In this way, the number of keypoints does not change in TPM, and the predicted box is chosen from all keypoints after TPM. From Table 5, with the increase of keypoint numbers, the performance of the *w/o selection* setting rises at first and then declines. 3D-SPS (*w/ selection*) achieves significant improvement compared to the *w/o selection* settings. This observation proves the benefits of progressive keypoint selection.

4.5. Qualitative comparison

In this subsection, we perform a qualitative comparison on *ScanRefer* validation set to show how 3D-SPS works.

Language-relevant Keypoints. We visualize the progressive keypoint selection process of 3D-SPS in Figure 6 and compare it with the two-stage baseline ScanRefer [2]. En-

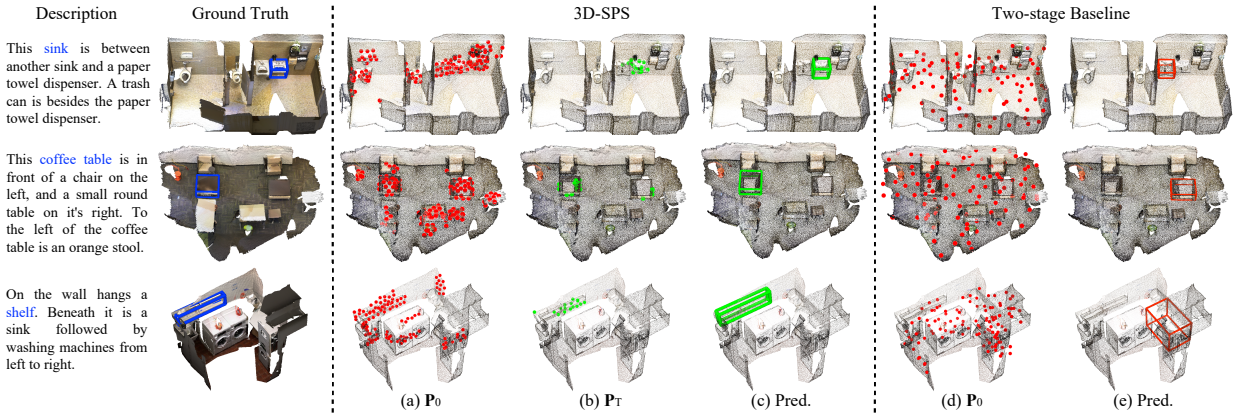

Figure 6. The two-stage baseline (ScanRefer) fails while our 3D-SPS predicts correctly since 3D-SPS can select more valuable keypoints. (a) Language-relevant keypoints $\mathbf{P}_0$ sampled by DKS. (b) Target keypoints $\mathbf{P}_T$ selected by TPM. (c) Bounding boxes predicted by 3D-SPS. (d) Language-irrelevant keypoints sampled by FPS. (e) Bounding boxes predicted by ScanRefer.

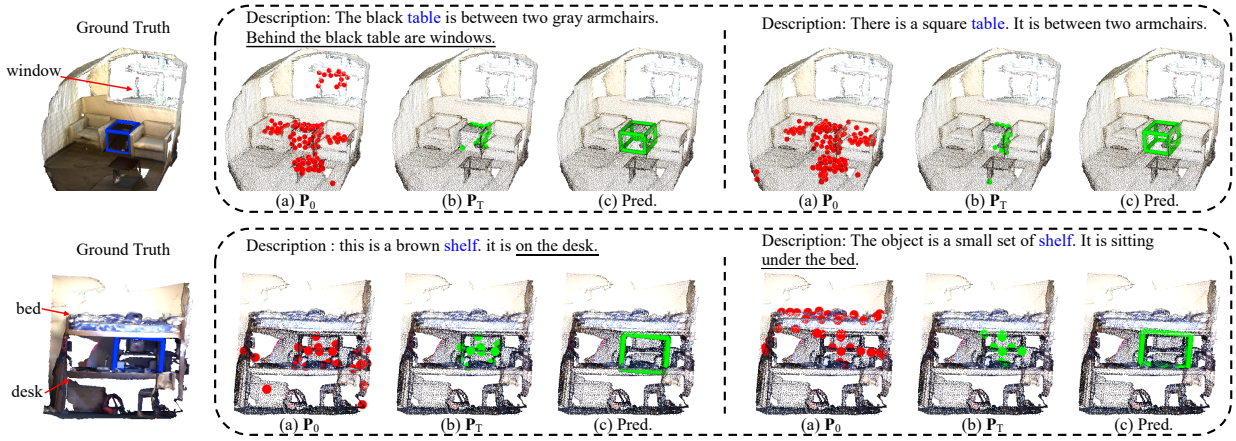

Figure 7. Visualization of the same referring target with different descriptions in 3D-SPS. (a) $\mathbf{P}_0$ sampled by DKS. Comparing the left and right subfigures in each row, when the language-relevant objects change (*e.g.*, *window*, *desk*, *bed*), 3D-SPS focuses on different keypoints (red keypoints). (b) $\mathbf{P}_T$ selected by TPM. (c) The predicted target bounding box.

abled by DKS and TPM, 3D-SPS gradually focuses on the target. In contrast, the attention of ScanRefer is scattered everywhere in the scene and ultimately fails to locate the target due to the separation of detection and matching.

Language-adapted Keypoints. 3D-SPS selects different keypoints for the same target with different descriptions. As shown in Figure 7 (upper), to locate the *table*, 3D-SPS selects some keypoints on the *window* for subsequent mining when *window* is mentioned in the left sample. On the right, when only *armchairs* is mentioned, 3D-SPS only selects keypoints on *armchairs* and *tables*. In Figure 7 (lower), for the target *shelf*, 3D-SPS finds more keypoints related to the *desk* when the shelf is described as *on the desk* in the left sample. When the description contains *under the bed*, the model pays more attention to the *bed*.

5. Conclusion and Discussion

In this work, we propose a brand new 3D visual grounding framework on point clouds named 3D Single-Stage Referred Point Progressive Selection method (3D-SPS). Under the guidance of language, it progressively selects keypoints following a coarse-to-fine pattern and directly localizes the target at a single stage. Comprehensive experiments reveal that our method outperforms the existing 3D VG methods on both *ScanRefer* and *Nr3D / Sr3D* datasets by a large margin, leading to the new state-of-the-art performance.

Limitation. The limitation of 3D-SPS exists due to the complexity of 3D point clouds and free-form description, although we have made significant improvements on existing methods. The view-dependent descriptions and the ambiguous queries can both confuse the model. These limitations could guide our future work.

Acknowledgement. This research is partly supported by National Natural Science Foundation of China (Grant 62122010, 61876177), Fundamental Research Funds for the Central Universities, and Key R & D Program of Zhejiang Province(2022C01082).

References

[1] Panos Achlioptas, Ahmed Abdelreheem, Fei Xia, Mohamed Elhoseiny, and Leonidas Guibas. Referit3d: Neural listeners for fine-grained 3d object identification in real-world scenes. In *ECCV*, 2020. 2, 3, 5, 6

[2] Dave Zhenyu Chen, Angel X. Chang, and Matthias Nießner. Scanrefer: 3d object localization in RGB-D scans using natural language. In *ECCV*, 2020. 1, 2, 3, 5, 6, 7

[3] Xinpeng Chen, Lin Ma, Jingyuan Chen, Zequn Jie, Wei Liu, and Jiebo Luo. Real-time referring expression comprehension by single-stage grounding network. *arXiv preprint arXiv:1812.03426*, 2018. 2

[4] Bowen Cheng, Lu Sheng, Shaoshuai Shi, Ming Yang, and Dong Xu. Back-tracing representative points for voting-based 3d object detection in point clouds. In *CVPR*, 2021. 1, 4

[5] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *CVPR*, 2017. 5

[6] Jiajun Deng, Zhengyuan Yang, Tianlang Chen, Wengang Zhou, and Houqiang Li. Transvg: End-to-end visual grounding with transformers. In *ICCV*, 2021. 2

[7] Mingtao Feng, Zhen Li, Qi Li, Liang Zhang, XiangDong Zhang, Guangming Zhu, Hui Zhang, Yaonan Wang, and Ajmal Mian. Free-form description guided 3d visual graph network for object grounding in point cloud. In *ICCV*, 2021. 1, 3

[8] Chen Gao, Jinyu Chen, Si Liu, Luting Wang, Qiong Zhang, and Qi Wu. Room-and-object aware knowledge reasoning for remote embodied referring expression. In *CVPR*, 2021. 2

[9] Ben Graham. Sparse 3d convolutional neural networks. In *BMVC*, 2015. 2

[10] Benjamin Graham, Martin Engelcke, and Laurens van der Maaten. 3d semantic segmentation with submanifold sparse convolutional networks. In *CVPR*, 2018. 2

[11] Dailan He, Yusheng Zhao, Junyu Luo, Tianrui Hui, Shaofei Huang, Aixi Zhang, and Si Liu. Transrefer3d: Entity-and-relation aware transformer for fine-grained 3d visual grounding. In *ACM MM*, 2021. 1, 2, 3

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 2

[13] Richang Hong, Daqing Liu, Xiaoyu Mo, Xiangnan He, and Hanwang Zhang. Learning to compose and reason with language tree structures for visual grounding. In *TPAMI*, 2019. 2

[14] Ronghang Hu, Huazhe Xu, Marcus Rohrbach, Jiashi Feng, Kate Saenko, and Trevor Darrell. Natural language object retrieval. In *CVPR*, 2016. 2, 6

[15] Pin-Hao Huang, Han-Hung Lee, Hwann-Tzong Chen, and Tyng-Luh Liu. Text-guided graph neural networks for referring 3d instance segmentation. In *AAAI*, 2021. 3, 6

[16] Georgin Jacob, RT Pramod, Harish Katti, and SP Arun. Qualitative similarities and differences in visual object representations between brains and deep networks. In *Nat. Commun.*, 2021. 2

[17] Yue Liao, Si Liu, Guanbin Li, Fei Wang, Yanjie Chen, Chen Qian, and Bo Li. A real-time cross-modality correlation filtering method for referring expression comprehension. In *CVPR*, 2020. 2

[18] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollar. Focal loss for dense object detection. In *ICCV*, 2017. 5

[19] Daqing Liu, Hanwang Zhang, Feng Wu, and Zheng-Jun Zha. Learning to assemble neural module tree networks for visual grounding. In *ICCV*, 2019. 2

[20] Ze Liu, Zheng Zhang, Yue Cao, Han Hu, and Xin Tong. Group-free 3d object detection via transformers. In *ICCV*, 2021. 1, 4, 5

[21] Ilya Loshchilov and Frank Hutter. Fixing weight decay regularization in adam. *ArXiv*, abs/1711.05101, 2017. 5

[22] Varun K Nagaraja, Vlad I Morariu, and Larry S Davis. Modeling context between objects for referring expression understanding. In *ECCV*, 2016. 2

[23] Charles R. Qi, Or Litany, Kaiming He, and Leonidas J. Guibas. Deep hough voting for 3d object detection in point clouds. In *ICCV*, 2019. 1, 4, 5

[24] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *CVPR*, 2017. 3

[25] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *NeurIPS*, 2017. 2, 3, 4, 5, 7

[26] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021. 5

[27] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster R-CNN: towards real-time object detection with region proposal networks. In *NeurIPS*, 2015. 4

[28] Junha Roh, Karthik Desingh, Ali Farhadi, and Dieter Fox. Languagerefer: Spatial-language model for 3d visual grounding. In *CoRL*, 2021. 3, 6

[29] Arka Sadhu, Kan Chen, and Ram Nevatia. Zero-shot grounding of objects from natural language queries. In *ICCV*, 2019. 1, 2

[30] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 2

[31] Shimon Ullman, Liav Assif, Ethan Fetaya, and Daniel Harari. Atoms of recognition in human and computer vision. In *PNAS*, 2016. 2

[32] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 4

[33] Liwei Wang, Yin Li, Jing Huang, and Svetlana Lazebnik. Learning two-branch neural networks for image-text matching tasks. In *TPAMI*, 2018. 1, 2

[34] Peng Wang, Qi Wu, Jiewei Cao, Chunhua Shen, Lianli Gao, and Anton van den Hengel. Neighbourhood watch: Refer-

ring expression comprehension via language-guided graph attention networks. In *CVPR*, 2019. 1, 2

[35] Sibei Yang, Guanbin Li, and Yizhou Yu. Cross-modal relationship inference for grounding referring expressions. In *CVPR*, 2019. 1, 2

[36] Sibei Yang, Guanbin Li, and Yizhou Yu. Dynamic graph attention for referring expression comprehension. In *CVPR*, 2019. 1, 2

[37] Zhengyuan Yang, Tianlang Chen, Liwei Wang, and Jiebo Luo. Improving one-stage visual grounding by recursive sub-query construction. In *ECCV*, 2020. 1, 2

[38] Zhengyuan Yang, Boqing Gong, Liwei Wang, Wenbing Huang, Dong Yu, and Jiebo Luo. A fast and accurate one-stage approach to visual grounding. In *ICCV*, 2019. 1, 2, 6

[39] Zhengyuan Yang, Songyang Zhang, Liwei Wang, and Jiebo Luo. Sat: 2d semantics assisted training for 3d visual grounding. In *ICCV*, 2021. 1, 2, 3, 6, 7

[40] Licheng Yu, Zhe Lin, Xiaohui Shen, Jimei Yang, Xin Lu, Mohit Bansal, and Tamara L Berg. Mattnet: Modular attention network for referring expression comprehension. In *CVPR*, 2018. 1, 2

[41] Licheng Yu, Patrick Poirson, Shan Yang, Alexander C Berg, and Tamara L Berg. Modeling context in referring expressions. In *ECCV*, 2016. 1, 2

[42] Zhihao Yuan, Xu Yan, Yinghong Liao, Ruimao Zhang, Sheng Wang, Zhen Li, and Shuguang Cui. Instancerefer: Cooperative holistic understanding for visual grounding on point clouds through instance multi-level contextual referring. In *ICCV*, 2021. 1, 3, 6

[43] Hanwang Zhang, Yulei Niu, and Shih-Fu Chang. Grounding referring expressions in images by variational context. In *CVPR*, 2018. 1, 2

[44] Lichen Zhao, Daigang Cai, Lu Sheng, and Dong Xu. 3dvg-transformer: Relation modeling for visual grounding on point clouds. In *ICCV*, 2021. 1, 2, 3, 6

[45] Yin Zhou and Oncel Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *CVPR*, 2018. 2

[46] Bohan Zhuang, Qi Wu, Chunhua Shen, Ian Reid, and Anton Van Den Hengel. Parallel attention: A unified framework for visual object discovery through dialogs and queries. In *CVPR*, 2018. 2