Text-Guided Graph Neural Networks for Referring 3D Instance Segmentation

Pin-Hao Huang,[1*] **Han-Hung Lee,**[1,2*] **Hwann-Tzong Chen,**[2,4] **Tyng-Luh Liu**[1,3]
[1] Institute of Information Science, Academia Sinica, Taiwan
[2] Department of Computer Science, National Tsing Hua University, Taiwan
[3] Taiwan AI Labs
[4] Aeolus Robotics
{pinhao517, rexleeppp}@gmail.com, htchen@cs.nthu.edu.tw, liutyng@iis.sinica.edu.tw

Abstract

This paper addresses a new task called *referring 3D instance segmentation*, which aims to segment out the target instance in a 3D scene given a query sentence. Previous work on scene understanding has explored visual grounding with natural language guidance, yet the emphasis is mostly constrained on images and videos. We propose a Text-guided Graph Neural Network (TGNN) for referring 3D instance segmentation on point clouds. Given a query sentence and the point cloud of a 3D scene, our method learns to extract per-point features and predicts an offset to shift each point toward its object center. Based on the point features and the offsets, we cluster the points to produce fused features and coordinates for the candidate objects. The resulting clusters are modeled as nodes in a Graph Neural Network to learn the representations that encompass the relation structure for each candidate object. The GNN layers leverage each object's features and its relations with neighbors to generate an attention heatmap for the input sentence expression. Finally, the attention heatmap is used to "guide" the aggregation of information from neighborhood nodes. Our method achieves state-of-the-art performance on referring 3D instance segmentation and 3D localization on ScanRefer, Nr3D, and Sr3D benchmarks, respectively.

Introduction

Referring image segmentation aims to make pixel-level predictions of the target object in an image described by a natural language expression. With the development of convolutional neural networks and the assist from several useful language models, referring segmentation in 2D images has been widely studied in the past few years. Yet, the joint modeling of language and 3D vision is still a new topic waiting to be explored. In this work, we design a text-guided graph-based approach for referring 3D instance segmentation—a new task that, to our best knowledge, has not been well investigated before. Given a referring expression, the task is to predict the 3D mask of the target instance in a point cloud. Figure 1 illustrates an example of our referring task.

Unlike 2D images, where pixels are arranged in an organized configuration, 3D data mostly come in the form of point clouds, which tend to be unordered and are more challenging to learn. Recent techniques transform point clouds into volumetric cells (voxels) and perform sparse convolution on occupied cells (Graham, Engelcke, and van der Maaten 2018). Such an approach demonstrates outstanding performance on the indoor-scene 3D semantic segmentation task and is commonly employed as the backbone model.

We adopt the above technique as the backbone for feature extraction in the proposed two-phase approach. Specifically, in the first phase of our method, the backbone model extracts point features and semantic classes. In parallel, an instance centroid estimation branch predicts the offsets to shift points toward their associated instance center. With these predictions, a fast mask prediction algorithm is proposed to generate high-quality 3D instance masks. In the second phase, we propose the Text-guided Graph Neural Network that takes the instance features, centers, and text features as input for instance referring. We use ScanRefer (Chen, Chang, and Nießner 2020), Nr3D, and Sr3D (Achlioptas et al. 2020) datasets to formulate the task of referring 3D instance segmentation for evaluating our method. Note that no prior art is available to solve the proposed new task. Hence, for comparison we also apply our method to the problem of referring 3D object localization and identification, which dose not require precise mask predictions and has been addressed in a few recent works. The experimental results show that our method establishes a strong baseline for the task of referring 3D instance segmentation for future approaches and, with a simple modification, can also achieve state-of-the-art performance on referring 3D object localization and identification.

Related Work

Referring Localization and Segmentation in 2D

Referring image localization aims to estimate a bounding box on the specific object described by the referring expression (Hu et al. 2016; Nagaraja, Morariu, and Davis 2016; Yu et al. 2016; Hu et al. 2017; Yu et al. 2017, 2018; Deng et al. 2018; Zhuang et al. 2018; Liu et al. 2019b; Wang et al. 2019a; Yang, Li, and Yu 2019a,b; Liu et al. 2019a; Sadhu, Chen, and Nevatia 2019; Yang, Li, and Yu 2020). In contrast, referring image segmentation predicts a segmentation mask on the referred object to obtain more precise results (Hu, Rohrbach, and Darrell 2016; Liu et al. 2017; Li et al. 2018; Margffoy-

[*] Equal contribution.

Query: This is a brown armchair. It is across from a desk.

Instance Segmentation

Instance Referring

Closer View

Figure 1: The conventional task of 3D instance segmentation is to extract the 3D masks of all instances in a 3D scene. As a more challenging task to be addressed in this paper, *referring 3D instance segmentation* aims to single out the referred 3D instance based on the visual and textual cues derived from the input 3D point cloud and the query sentence.

Tuay et al. 2018; Ye et al. 2019; Chen et al. 2019; Hu et al. 2020; Huang et al. 2020; Hui et al. 2020).

Many of these works investigate the relationship between objects. One approach involves modular decomposition (Hu et al. 2017; Yu et al. 2018), which learns the alignment between pairwise region relationship and the parsed language expression (of object relationship). Recently, Graph Neural Networks (GNNs) have been widely used for modeling object relationships or reasoning sentence context for referring expression grounding. A common approach is to construct a graph over the objects or regions extracted from the backbone model (Wang et al. 2019a; Yang, Li, and Yu 2019a,b; Huang et al. 2020; Yang, Li, and Yu 2020), or construct a graph that represents the linguistic structure (Hui et al. 2020). Although GNNs are powerful in dealing with 2D vision and language reasoning, we argue that the spatial structures are more complicated in 3D settings, and it is rather challenging to tackle 3D tasks using general GNNs. Hence, we propose an enhanced GNN approach that adapts the linguistic representation with each instance node and emphasizes the spatial relationships between instances to perform reasoning for the referring 3D instance segmentation task.

3D Instance Localization and Segmentation

The development for 3D deep learning has progressed impressively over the past few years. Pioneering works like PointNet (Qi et al. 2016, 2017) have been proposed to directly utilize the raw point cloud as input for semantic segmentation, and extensively used as a backbone feature extractor for point-based methods (Wang et al. 2018; Yang et al. 2019; Yi et al. 2019; Pham et al. 2019). On the other hand, voxel-based methods operate on the volumetric grid parsed from the input 3D scene. SparseConvNet (SCN) (Graham, Engelcke, and van der Maaten 2018; Choy, Gwak, and Savarese 2019) leverages the sparsity of 3D data and applies efficient sparse convolution operation on occupied voxels. They demonstrate outstanding performance on 3D semantic segmentation. MTML (Lahoud et al. 2019) employs metric learning to learn voxel features for clustering voxels of the same instance, while MASC (Liu and Furukawa 2019) simultaneously predicts the voxel affinity for instance label prediction. Additionally, 3DSIS (Hou, Dai, and Nießner 2019) combines multiview 2D features with 3D features for instance segmentation.

Instance center prediction has proved useful for point cloud clustering. VoteNet (Qi et al. 2019) learns an offset for each point that shifts each point toward their corresponding instance center. OccuSeg (Han et al. 2020), PointGroup (Jiang et al. 2020) and 3D-MPA (Engelmann et al. 2020) all adopt SCN as the backbone and include center offset prediction in their framework to ameliorate the 3D instance segmentation results and obtain impressive performances on popular 3D instance segmentation benchmarks such as ScanNet (Dai et al. 2017) and S3DIS (Armeni et al. 2016).

Following these leading techniques, we also employ SCN as our backbone model. However, beyond this analogy, we introduce a fast instance mask prediction method that shows promising results for 3D instance segmentation.

3D Vision and Language

Compared to the counterpart of referring expression comprehension in images, joint inference of language and 3D vision is a relatively new research field. Existing works focus on generating fine-grained 3D objects (Chen et al. 2018) or distinguishing fine-grained differences between objects (Achlioptas et al. 2019) given the language descriptions. These tasks are confined to individual objects, while we focus on tasks of singling out items in natural 3D scenes. ScanRefer (Chen, Chang, and Nießner 2020) introduces a task of localizing objects within a 3D scene given the referring expressions. ReferIt3D (Achlioptas et al. 2020) proposes similar datasets, Sr3D, and Nr3D. Different from ScanRefer, the proposed task in ReferIt3D assumes that a well-segmented premise in which localization is not required.

Our work focuses on the referring 3D instance segmentation task extended from ScanRefer. We also test on Sr3D and Nr3D to show the effectiveness of our graph-based approach.

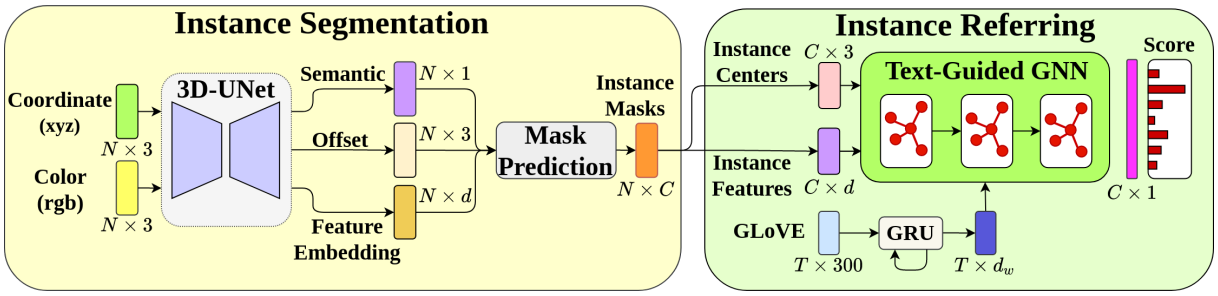

Figure 2: Illustration of the overall architecture. The architecture contains two main parts: (1) instance segmentation and (2) instance referring. First, given the input 3D scene, the backbone network predicts three attributes for each point: the semantic class, the offset between each point and its corresponding instance center, and the feature embedding. Based on these predictions, the points are then clustered to obtain the segmentation results. Next, the instance features and centers are fed into the Text-Guided Graph Neural Network (TGNN) along with the text features aggregated by the GRU. The TGNN aggregates the multi-modality features and finally predicts the results.

Method

Given a set of 3D point clouds about a scene and a language expression S, the referring task aims at segmenting the target instance identified by S. The proposed method proceeds in two phases. First, it is designed to perform 3D instance segmentation, where from the learned point features the point clouds are classified into individual instances and thus the predicted instance centers. Second, the instance-level features can be obtained by aggregating point features that belong to the same instance, which are then fed into a Text-guided Graph Neural Network (TGNN) to yield the final prediction of the referred 3D instance. Figure 2 illustrates that the various modules of the two phases can be effectively linked to form an end-to-end trainable neural network for achieving the referring 3D instance segmentation of point clouds.

Phase 1: 3D Instance Segmentation

To perform instance segmentation, it is feasible to apply any current leading techniques such as OccuSeg (Han et al. 2020). However, these techniques often require additional sophisticated modules to enhance their performance. We instead intend to make our segmentation model lightweight while maintaining competitive segmentation performance to accommodate for the text-guided GNN. Thus, our segmentation model contains only a single 3D Sparse U-Net (Graham, Engelcke, and van der Maaten 2018) for point feature extraction. The model is learned to encode three different representations to facilitate the following three tasks: (1) The semantic class each point belongs to. (2) The point features for grouping points within the same instance. (3) The coordinate offset between each point and its corresponding instance center.

Semantic Segmentation. The semantic representations are learned through a simple cross entropy loss denoted as $\mathcal{L}_{\text{ss}}$. Each predicted semantic class is obtained by choosing the one with the maximum score.

Feature Embedding. The purpose of learning a proper feature embedding is to ensure that the resulting model can adequately group points of the same instance while differentiating those from different instances. The supervised learning is carried out via a discriminative loss function (De Brabandere, Neven, and Van Gool 2017), which has been recently proved effective for 3D instance segmentation (Lahoud et al. 2019; Han et al. 2020). Specifically, the feature embedding loss consists of three terms:

$$\mathcal{L}_{\text{fe}} = \mathcal{L}_{\text{var}} + \mathcal{L}_{\text{dist}} + \alpha \cdot \mathcal{L}_{\text{reg}} \,, \tag{1}$$

where α is a weighting parameter and following (Lahoud et al. 2019), we have

$$\mathcal{L}_{\text{var}} = \frac{1}{C} \sum_{c=1}^{C} \frac{1}{N_c} \sum_{i=1}^{N_c} [\|\boldsymbol{\mu}_c - \mathbf{f}_i\| - \epsilon_{\text{var}}]_+^2 \,, \tag{2}$$

$$\mathcal{L}_{\text{dist}} = \frac{1}{C(C-1)} \sum_{c=1}^{C} \sum_{c \neq c'=1}^{C} [\delta_{\text{dist}} - \|\boldsymbol{\mu}_c - \boldsymbol{\mu}_{c'}\|]_+^2 \,, \tag{3}$$

$$\mathcal{L}_{\text{reg}} = \frac{1}{C} \sum_{c=1}^{C} \|\boldsymbol{\mu}_c\| \,. \tag{4}$$

Note that the notations above assume there are C ground-truth instances/clusters and N_c is the total number of points in the cth instance, whose average feature is calculated by $\boldsymbol{\mu}_c = \frac{1}{N_c} \sum_{i=1}^{N_c} \mathbf{f}_i$. The distance parameter ϵ_{var} in (2) is the expected maximum radius of a cluster, while δ_{dist} in (3) is the minimum distance between two cluster centers. $[x]_+ = \max(0, x)$ is just like a linear rectifier.

Offset Prediction. Denote the center of ground-truth cluster $c \in \{1, \ldots, C\}$ as $\mathbf{o}_c \in \mathbb{R}^3$. The assignment $i \mapsto c(i)$ specifies that point $x_i \in \mathbb{R}^3$ is associated with the ground-truth cluster with center $\mathbf{o}_{c(i)}$. Similar to VoteNet (Qi et al. 2019), we predict for each point x_i its geometry offset Δx_i, to the ground-truth center $\mathbf{o}_{c(i)}$ by optimizing

$$\mathcal{L}_{\text{cen}} = \frac{1}{N} \sum_{i=1}^{N} \text{smooth}_{\ell_1}(\Delta x_i - (\mathbf{o}_{c(i)} - x_i)) \,, \tag{5}$$

where $N = \sum_{c=1}^{C} N_c$ is the total number of points in the point clouds. To ensure the predicted offset points toward the

correct direction, we add the following directional loss:

$$\mathcal{L}_{\text{dir}} = \frac{1}{N}\sum_{i=1}^{N} 1 - \left(\frac{\Delta x_i}{\|\Delta x_i\|} \cdot \frac{\mathbf{o}_{c(i)} - x_i}{\|\mathbf{o}_{c(i)} - x_i\|} \right). \quad (6)$$

We have so far described how our method yields the semantic class labels, the learned point features, and the point-wise offsets. These modules indeed function as essential pre-processing and are mostly benefited or motivated from existing techniques as specified in the relevant work. In the remaining of this section, we detail the central part of our approach to referring 3D instance segmentation.

3D Mask Prediction. Phase 1 concludes with the task to obtain the mask prediction for 3D instance segmentation. We focus on the point features, $\{\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_N\}$ and the offset predictions, $\{\Delta x_1, \Delta x_2, \ldots, \Delta x_N\}$. After shifting points to each predicted center, namely, $\{x_i + \Delta x_i\}_{i=1}^N$, points within the same instance are expected to be closely clustered both in the geometry space as well as the feature space. Under this premise, our approach first estimates the instance centers and then iteratively refine them by sampling points, favoring those that have shorter *distances* to their respective cluster center in both the geometry and feature space.

Due to the concern of computation efficiency in training the model, we need to develop an effective sampling scheme to well approximate the essential cluster distribution of the given point clouds. Observe that those points with a small offset magnitude $\|\Delta x\|$ are deemed to be closer to its instance center and assume higher confidence to be clustered. Thus, to begin with, we sample M points ($M \ll N$) according to the following Gaussian distribution:

$$x \sim p(\|\Delta x\|) \propto \exp\left(\frac{\|\Delta x\|^2}{-2\sigma^2}\right) \quad (7)$$

such that points with small $\|\Delta x\|$ have a higher probability to be selected. We arrange the set of sampled points in order of ascending $\|\Delta x\|$ and denote it as $\widetilde{Q} = \{(\tilde{x}_i, \tilde{\mathbf{f}}_i)\}_{i=1}^M$. Let $\tilde{x}_{i^*} \in \widetilde{Q}$ be the one with the smallest index and initially $i^* = 1$. We then cluster all other sampled points in $\widetilde{Q}$ that are close to $\tilde{x}_{i^*}$ in both the geometry and feature space, and use average pooling to obtain the center $\widetilde{X}$ and the feature $\widetilde{F}$ of the resulting cluster proposal. All these sampled points relevant to forming the proposal are then removed from $\widetilde{Q}$. The process is repeated until $\widetilde{Q}$ is empty, and the resulting set of cluster proposals is denoted as $\mathcal{P} = \{(\widetilde{X}_c, \widetilde{F}_c)\}$. In Algorithm 1, we list the steps of sequential re-sampling and the use of non-maximum suppression to refine $\mathcal{P}$ into $\mathbf{M} = \{\mathbf{m}_1, \mathbf{m}_2, \ldots, \mathbf{m}_C\}$, the set of C instance masks.

Algorithm 1 Sequential Re-sampling for Instance Masks

Input: Point clouds: $\mathbf{P} = \{(x_i, \mathbf{f}_i)\}_{i=1}^N \in \mathbb{R}^{N\times(3+d)}$; Cluster proposals: $\mathcal{P} = \{(\widetilde{X}_c, \widetilde{F}_c)\}$; $\delta_1, \delta_2, \sigma_1, \sigma_2$
Output: Masks $\mathbf{M} = \{\mathbf{m}_1, \mathbf{m}_2, \ldots \mathbf{m}_C\}$.

for $c = 1$ to $|\mathcal{P}|$ **do**
  Initialize $\epsilon = \infty$
  while $\epsilon > 10^{-2}$ **do**
    Sample $\mathbf{P}'_c$ w.r.t. bilateral Gaussian distribution:
    $(x, \mathbf{f}) \in \mathbf{P} \sim \frac{1}{Z}\exp\left(\frac{\|\widetilde{X}_c - x\|^2}{-2\sigma_1^2} + \frac{\|\widetilde{F}_c - \mathbf{f}\|^2}{-2\sigma_2^2}\right)$
    $(\widetilde{X}'_c, \widetilde{F}'_c) = \frac{1}{|\mathbf{P}'_c|}\sum_{(x,\mathbf{f})\in\mathbf{P}'_c}(x, \mathbf{f})$
    $\epsilon = \|\widetilde{X}'_c - \widetilde{X}_c\|$
    $(\widetilde{X}_c, \widetilde{F}_c) \leftarrow (\widetilde{X}'_c, \widetilde{F}'_c)$
  end while
end for
for $c = 1$ to $|\mathcal{P}|$ **do**
  $\mathbf{m}_c = [m_{c1}, m_{c2}, \ldots, m_{cN}]$, where
  $m_{ci} = \mathbb{1}[\,\|x_i - \widetilde{X}_c\| \le \delta_1 \wedge \|\mathbf{f}_i - \widetilde{F}_c\| \le \delta_2\,]$
  Add $\mathbf{m}_c$ to $\mathbf{M}$
end for
Apply NMS to $\mathbf{M}$ to remove overlapped masks.

Phase 2: Referring Segmentation via TGNN

To achieve referring instance segmentation, we first need a proper feature representation for the language input S, say, of length T. We use the pre-trained GloVE model (Pennington, Socher, and Manning 2014) to encode each word into a 300-d vector. These GloVE features are sequentially fed into a GRU network to aggregate the context of the sentence and output the textual embeddings $\mathbf{E} = \{\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_T\}$.

Considering now the predicted C instance features $\{\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \ldots, \boldsymbol{\mu}_C\}$ and the just-described textual embeddings, a simple scheme to pinpoint the referring target is to find the instance feature which *matches* the most to the textual embeddings. However, such a naïve approach ignores that spatial relationships between a candidate and its surrounding instances indeed play important roles in identifying the target instance such as in the case of referring *the chair next to the table*, *the bed under the window*, etc. To resolve this difficulty, we establish a Text-guided Graph Neural Network (TGNN) model to effectively correlate instance features with textual embeddings, while taking account of the spatial context among the object instances. An overview of our TGNN is shown in Figure 3. We next describe the key components leading to the proposed TGNN formulation.

Instance Graph. With the obtained C instances, we construct a directed instance graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ representing the overall scene structure, where the vertices represent the instances, and the directed edges are defined by each instance's K-nearest (based on Euclidean distance) neighbors. Note that a directed edge $(u, v) \in \mathcal{E}$ from node u to node v indicates that u is one of the K nearest neighbors of v, but the opposite may not hold, *i.e.*, (v, u) could be absent in $\mathcal{E}$.

Relative Coordinates Encoding. The center of the cth instance is $\mathbf{o}_c$ as in (6). Analogously, its K-nearest neighborhood, including the centers of the K nearby instances can be denoted as $\mathcal{N}_c = \{\mathbf{o}_c^1, \mathbf{o}_c^2, \ldots, \mathbf{o}_c^K\}$. In the proposed TGNN learning, we explore not only the coordinates relative to the whole scene, but also the relative positions between an underlying instance and its surrounding K neighboring objects. Taking the consideration into account, we use an MLP layer to encode the *relative instance coordinates* of the kth neigh-

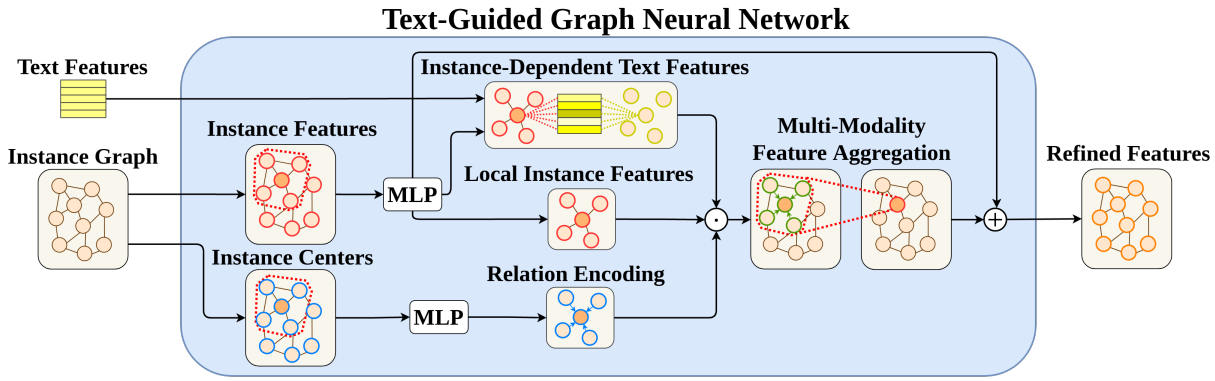

Figure 3: Illustration of the Text-Guided Graph Neural Network. Given the instance graph and textual features, our TGNN not only learns to match the instance features with textual embedding but also encodes the spatial context among the object instances. For each instance, we weight the text features based on the attention score related to the instances. On the other hand, we aggregate the spatial relation between the centering instance and the neighboring instances. The multi-modal features are then combined to form the refined instance features.

bor of instance c by

$$\mathbf{r}_c^k = \mathrm{MLP}(\mathbf{o}_c; \mathbf{o}_c^k; (\mathbf{o}_c - \mathbf{o}_c^k); \|\mathbf{o}_c - \mathbf{o}_c^k\|)\,, \tag{8}$$

where the notation ";" denotes the concatenation operation, and $\|\cdot\|$ is the Euclidean distance.

Instance-Dependent Language Attention. Besides providing the relative geometry information, the K-nearest instance neighborhood can also be used to evaluate the language attention. To this end, let the corresponding (local) instance features of $\mathcal{N}_c$ be $\{\boldsymbol{\mu}_c^1, \boldsymbol{\mu}_c^2, \ldots, \boldsymbol{\mu}_c^K\}$. Then, the kth instance of $\mathcal{N}_c$ influences the feature representation of $\mathbf{e}_t \in \mathbf{E}$ by exerting the following attention weight:

$$a_{c,t}^k = \frac{\exp(\phi(\boldsymbol{\mu}_c^k)^\top \psi(\mathbf{e}_t))}{\sum\limits_{\mathbf{e}_t \in E} \exp(\phi(\boldsymbol{\mu}_c^k)^\top \psi(\mathbf{e}_t))}\,, \tag{9}$$

where ϕ and ψ are MLPs. Each attention weight $a_{c,t}^k$ can be considered the relevance score between the tth textual embedding $\mathbf{e}_t$ and the kth neighbor of the cth instance in the scene. We then obtain the instance-dependent local sentence embedding $\mathbf{S}_c = \{\mathbf{s}_c^1, \mathbf{s}_c^2, \ldots, \mathbf{s}_c^K\}$ with respect to the cth instance and its neighborhood $\mathcal{N}_c$ where

$$\mathbf{s}_c^k = \sum_{t=1}^{T} a_{c,t}^k \psi(\mathbf{e}_t), \quad \text{for } 1 \le k \le K. \tag{10}$$

Multi-modality Aggregation. From (8) and (10), we can refine each instance feature $\boldsymbol{\mu}_c$ into $\hat{\boldsymbol{\mu}}_c$ by incorporating the respective relative coordinates encodings as well as the instance-dependent local sentence embeddings. We have

$$\hat{\boldsymbol{\mu}}_c = \phi(\boldsymbol{\mu}_c) + \sum_{k=1}^{K} \phi(\boldsymbol{\mu}_c^k) \odot \mathbf{s}_c^k \odot \mathbf{r}_c^k\,, \tag{11}$$

where $\odot$ is the element-wise multiplication. In our formulation, the feature aggregation of cluster/instance c in (11) is implemented via a graph neural network where the multi-modality fusion is realized by the summation of the accumulated messages from the K neighboring instances. Despite that the message fusion involves features accounting for discriminative embedding, sentence embedding, and relative coordinate geometry, we choose to characterize the GNN as text-guided to manifest the pivotal role of the language cue in solving referring 3D instance segmentation.

Score Prediction. The final refined features are then passed into an MLP followed by the softmax function to obtain the final score for each instance:

$$y_c = \mathrm{MLP}(\hat{\boldsymbol{\mu}}_c) \tag{12}$$

$$\hat{y}_c = \frac{\exp(y_c)}{\sum_{c=1}^{C} \exp(y_c)}. \tag{13}$$

The referring loss is the cross entropy loss denoted as $\mathcal{L}_{\mathrm{ref}}$.

Experiments

Training Details

We pre-train a sparse 3D UNet feature extractor for 3D instance segmentation. The pre-trained weights are loaded and kept fixed during the training of TGNN for referring 3D instance segmentation. Each training iteration involves a batch of scenes where all the referring sentences for those scenes are fed into the network to compute the loss. For the experiments using GRU as the language extractor, we use a batch size of 8 and an initial learning rate of 0.001 with decay of 0.1 every 100 epochs. The maximum timestep and sentence length for GRU are set to 80. For the experiments with BERT (Vaswani et al. 2017; Devlin et al. 2018), the weights of the BERT model and TGNN are updated separately. The initial learning rate is 0.0002 for BERT with decay of 0.5 every 10 epochs, while the initial learning rate is 0.001 for TGNN with decay of 0.5 every 50 epochs. The batch size is 16, and the maximum sentence length is 80 as in GRU. The number of nearest-neighbors is 16 unless specified. The number of layers in the GNN is set to 3.

Dataset

We evaluate our method using recent 3D referring datasets including ScanRefer (Chen, Chang, and Nießner 2020) and Nr3D/Sr3D of ReferIt3D (Achlioptas et al. 2020). The datasets are based on ScanNetv2 (Dai et al. 2017), which contains 1,513 richly-annotated 3D reconstructions of indoor scenes. These datasets all follow the official ScanNet splits.

ScanRefer. ScanRefer contains 51,583 natural language expressions of 11,046 objects from 800 ScanNet scenes. The dataset is originally introduced for the task of referring object localization. We extend the task to referring 3D instance segmentation by taking the instance masks from ScanNet.

Sr3D and Nr3D. Sr3D (Spatial Reference in 3D) contains 83.5K synthetic expressions generated by a simple composition template. Nr3D (Natural Reference in 3D) has 41.5K human expressions collected as ReferItGame (Kazemzadeh et al. 2014). Both datasets' goal is to ground a natural language expression to an object in the 3D scene.

Results and Ablation Study

To our knowledge this is the first work to solve the task of referring 3D instance segmentation. Without available prior work for comparison, we evaluate our method against a carefully designed baseline and conduct ablation study to demonstrate the effectiveness of our method.

The ablation results are shown in Table 1. We use *mean IOU* and *Acc@kIOU* as the evaluation metrics. For the baseline "concat" model, we concatenate the instance features with the output of the last layer of the language extractor to obtain a fused vector for each object. The fused vector is passed through an MLP to obtain the predictions. Our full TGNN model achieves an increase of **+4 mIOU** with GRU and **+5.3 mIOU** with BERT over the baseline model.

We also evaluate the effects of different modules. In table 1, **R** denotes the relation coordinate encoding and **A** denotes the Instance-Dependent Language Attention in each graph layer. For the ablation without relation, we simply remove the relation encoding branch in the GNN layers. For the ablation without language attention, the output of the language extractor goes through an MLP and tiles up to match the shape of relation and instance features for graph feature aggregation. The ablation analysis shows that the relation encoding affects the most and gives an increase of **+1.4 mIOU**.

For the ablation with different language extractors, the language attention module yields a minor gain of **+0.5 mIOU** when using GRU as the language extractor. In contrast, the gain from adopting an language attention module is larger (**+1.0 mIOU**) when using BERT for language feature extraction owing to the more powerful language representations learned by BERT. We visualize several qualitative results in Figures 4 and 5.

Referring 3D Localization and Identification

Comparison with ScanRefer Network. Although our task of interest is referring 3D instance segmentation, our method can also predict 3D bounding boxes as by-products for comparing with the ScanRefer network (Chen, Chang, and Nießner 2020) on the task of referring localization. To obtain a 3D bounding box for a referred instance, we simply take the minimum and maximum of the XYZ values of points in the segmentation mask. Table 2 shows the results. The "unique" and "multiple" subsets indicate whether or not multiple instances of the same object class appear in the scene for the referred object. Our model with a BERT language extractor achieves better performance compared to ScanRefer on *Acc@0.5*. The performance on *Acc@0.25* is slightly worse but on-par with ScanRefer, probably due to the over-simplified strategy we use for deriving bounding boxes from masks and the differences in the evaluation criteria of localization and segmentation. If a segmentation mask misses a few points at the boundary, it only results in a small change in segmentation IOU; however, the bounding box size might vary a lot because of those missing boundary points. Since this work focuses on referring 3D instance segmentation, we do not seek to minimize the bounding-box loss like the ScanRefer network. Also, the ScanRefer network uses XYZ, RGB, multiview, and normal features while we only use XYZ and RGB. Nevertheless, the better accuracy of

Method	mIOU (%)	Acc@0.25	Acc@0.5
Concat	22.6 ± 0.3	30.4 ± 0.5	24.8 ± 0.4
GRU w/o R/A	24.0 ± 0.2	32.1 ± 0.2	26.7 ± 0.3
GRU w/o A	25.6 ± 0.2	34.2 ± 0.3	28.6 ± 0.4
GRU w/o R	24.8 ± 0.4	33.4 ± 0.4	27.5 ± 0.6
GRU	$\mathbf{26.1 \pm 0.2}$	$\mathbf{35.0 \pm 0.4}$	$\mathbf{29.0 \pm 0.3}$
BERT w/o A	26.8 ± 0.3	35.9 ± 0.3	30.1 ± 0.4
BERT	$\mathbf{27.8 \pm 0.2}$	$\mathbf{37.5 \pm 0.4}$	$\mathbf{31.4 \pm 0.3}$

Table 1: Ablations and comparison with baseline on the ScanRefer validation set. (R: Relation; A: Attention).

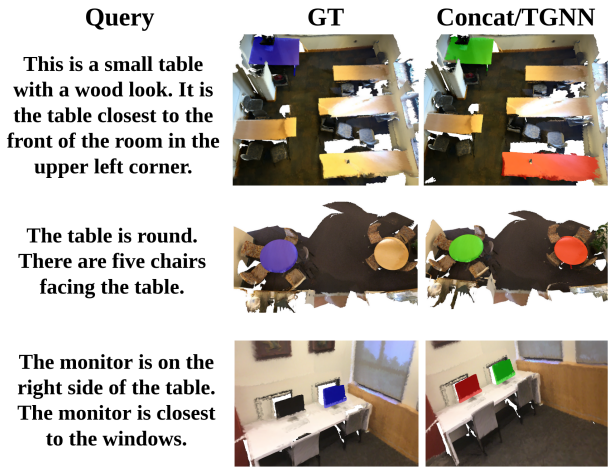

Figure 4: Examples of our predictions on the ScanRefer validation set. The blue, red, and green masks indicate the ground truth, the prediction of the baseline "concat" model, and our TGNN, respectively. In these examples, TGNN predicts correct results by successfully modeling the relationship around the referred instances while the baseline "concat" model fails due to its lack of spatial context.

	Method	Unique		Multiple		Overall	
		Acc@0.25 (%)	Acc@0.5	Acc@0.25	Acc@0.5	Acc@0.25	Acc@0.5
Validation	ScanRefer	63.04	39.95	28.91	18.17	35.53	22.39
	OursGRU	64.50	53.01	27.01	21.88	34.29	27.92
	OursBERT	**68.61**	**56.80**	**29.84**	**23.18**	**37.37**	**29.70**
Test	ScanRefer	**68.59**	43.53	**34.88**	20.97	**42.44**	26.03
	OursGRU	62.40	53.30	28.20	21.30	35.90	28.50
	OursBERT	68.30	**58.90**	33.10	**25.30**	41.00	**32.80**

Table 2: ScanRefer object localization results evaluated by accuracy at IOU 0.25 and IOU 0.5

	Method	Overall	Easy	Hard	View-dep.	View-indep.
Nr3D	ReferIt3D	$35.6\% \pm 0.7\%$	$43.6\% \pm 0.8\%$	$27.9\% \pm 0.7\%$	$32.5\% \pm 0.7\%$	$37.1\% \pm 0.8\%$
	Ours	$\mathbf{37.3\% \pm 0.3\%}$	$\mathbf{44.2\% \pm 0.4\%}$	$\mathbf{30.6\% \pm 0.2\%}$	$\mathbf{35.8\% \pm 0.2\%}$	$\mathbf{38.0\% \pm 0.3\%}$
Sr3D	ReferIt3D	$40.8\% \pm 0.2\%$	$44.7\% \pm 0.1\%$	$31.5\% \pm 0.4\%$	$39.2\% \pm 1.0\%$	$40.8\% \pm 0.1\%$
	Ours	$\mathbf{45.0\% \pm 0.2\%}$	$\mathbf{48.5\% \pm 0.2\%}$	$\mathbf{36.9\% \pm 0.5\%}$	$\mathbf{45.8\% \pm 1.1\%}$	$\mathbf{45.0\% \pm 0.2\%}$

Table 3: Comparison with ReferIt3D on the accuracy of referring object identification

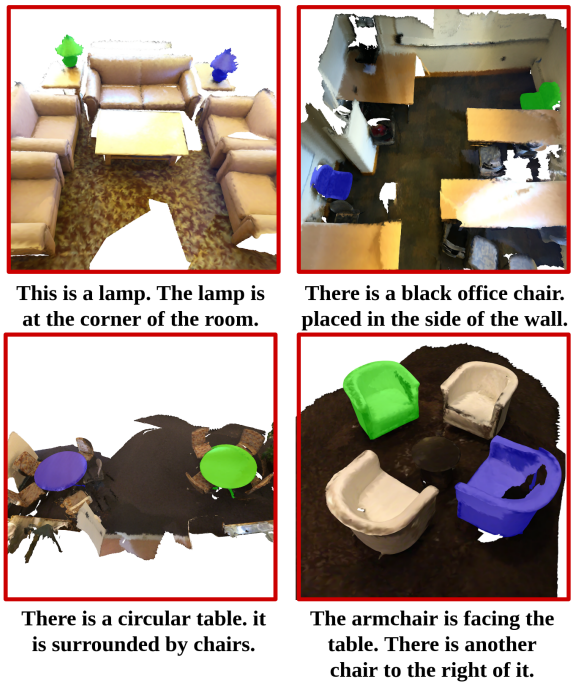

Figure 5: Reasonable failure cases of our predictions on the ScanRefer validation set. The blue masks indicate the ground truths while the green masks refer to our predicted instances. The predictions are incorrect but reasonable since the referring descriptions are ambiguous to pinpoint a unique target among multiple objects that conform with the expression.

our method at higher IOU shows that our method can more precisely localize the referred target.

Comparison with ReferIt3D. ReferIt3DNet (Achlioptas et al. 2020) assumes that object masks for each scene are already given in the input, and the task is just to select which object is the referred object. PointNet++ (Qi et al. 2016) is used to extract the feature vector for each candidate. Object vectors are then fed as nodes into a dynamic graph-convolutional network (DGCNN) (Wang et al. 2019b). To compare with their method, we replace their DGCNN with our proposed TGNN. We set the number of nearest-neighbors $K = 7$ to match their settings. The results are shown in Table 3. For results not presented in the original ReferIt3D paper, we re-run their code and report the results. Our model shows improvements over ReferIt3DNet, especially when the description context is more difficult. In the "hard" context cases (more than two instances of the same object class in the scene), we achieve improvements of **+2.7%** on Nr3D and **+5.4%** on Sr3D. For the view-dependent context (the referring expression depending on camera view), we see gains of **+3.3%** and **+6.6%** for Nr3D and Sr3D. The results suggest that our graph formulation of TGNN is more effective than DGCNN in modeling the context information.

Conclusions

The core of our method is the Text-guided Graph Neural Network (TGNN)—a graph-based approach for referring instance segmentation in 3D scenes. Given the segmented 3D instances in the first phase of our method, TGNN not only learns to aggregate textual features based on the neighboring local structure of each instance but also captures the spatial interactions centering on each object. Our method shows significant improvements over the baselines on ScanRefer, NR3D, and SR3D datasets, justifying its effectiveness in modeling instance relationships for referring 3D instance segmentation. Besides, since TGNN operates under the premise that all instances are well segmented, our Sequential Re-sampling Mask Prediction algorithm plays an indispensable role in the entire framework to produce fast and promising instance segmentation results. Referring 3D instance segmentation is a relatively new task to be explored. Further improvements can certainly be achieved with the advances in computer vision and natural language processing, whereas the crux will still be on the novel integration of techniques from both fields.

Acknowledgements

This work was supported in part by the MOST, Taiwan under Grant 110-2634-F-001-009. We are grateful to the National Center for High-performance Computing for computer time and facilities.

References

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. *16th European Conference on Computer Vision (ECCV)* .

Achlioptas, P.; Fan, J.; Hawkins, R.; Goodman, N.; and Guibas, L. J. 2019. ShapeGlot: Learning language for shape differentiation. In *Proceedings of the IEEE International Conference on Computer Vision*, 8938–8947.

Armeni, I.; Sener, O.; Zamir, A. R.; Jiang, H.; Brilakis, I.; Fischer, M.; and Savarese, S. 2016. 3d semantic parsing of large-scale indoor spaces. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1534–1543.

Chen, D.-J.; Jia, S.; Lo, Y.-C.; Chen, H.-T.; and Liu, T.-L. 2019. See-Through-Text Grouping for Referring Image Segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

Chen, D. Z.; Chang, A. X.; and Nießner, M. 2020. ScanRefer: 3D Object Localization in RGB-D Scans using Natural Language. *16th European Conference on Computer Vision (ECCV)* .

Chen, K.; Choy, C. B.; Savva, M.; Chang, A. X.; Funkhouser, T.; and Savarese, S. 2018. Text2shape: Generating shapes from natural language by learning joint embeddings. In *Asian Conference on Computer Vision*, 100–116. Springer.

Choy, C.; Gwak, J.; and Savarese, S. 2019. 4d spatio-temporal convnets: Minkowski convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3075–3084.

Dai, A.; Chang, A. X.; Savva, M.; Halber, M.; Funkhouser, T.; and Nießner, M. 2017. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 5828–5839.

De Brabandere, B.; Neven, D.; and Van Gool, L. 2017. Semantic instance segmentation with a discriminative loss function. *arXiv preprint arXiv:1708.02551* .

Deng, C.; Wu, Q.; Wu, Q.; Hu, F.; Lyu, F.; and Tan, M. 2018. Visual grounding via accumulated attention. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 7746–7755.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv preprint arXiv:1810.04805* .

Engelmann, F.; Bokeloh, M.; Fathi, A.; Leibe, B.; and Niessner, M. 2020. 3D-MPA: Multi-Proposal Aggregation for 3D Semantic Instance Segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Graham, B.; Engelcke, M.; and van der Maaten, L. 2018. 3D Semantic Segmentation with Submanifold Sparse Convolutional Networks. *CVPR* .

Han, L.; Zheng, T.; Xu, L.; and Fang, L. 2020. OccuSeg: Occupancy-Aware 3D Instance Segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Hou, J.; Dai, A.; and Nießner, M. 2019. 3d-sis: 3d semantic instance segmentation of rgb-d scans. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4421–4430.

Hu, R.; Rohrbach, M.; Andreas, J.; Darrell, T.; and Saenko, K. 2017. Modeling relationships in referential expressions with compositional modular networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1115–1124.

Hu, R.; Rohrbach, M.; and Darrell, T. 2016. Segmentation from natural language expressions. In *European Conference on Computer Vision*, 108–124. Springer.

Hu, R.; Xu, H.; Rohrbach, M.; Feng, J.; Saenko, K.; and Darrell, T. 2016. Natural language object retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4555–4564.

Hu, Z.; Feng, G.; Sun, J.; Zhang, L.; and Lu, H. 2020. Bi-Directional Relationship Inferring Network for Referring Image Segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Huang, S.; Hui, T.; Liu, S.; Li, G.; Wei, Y.; Han, J.; Liu, L.; and Li, B. 2020. Referring Image Segmentation via Cross-Modal Progressive Comprehension. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Hui, T.; Liu, S.; Huang, S.; Li, G.; Yu, S.; Zhang, F.; and Han, J. 2020. Linguistic Structure Guided Context Modeling for Referring Image Segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*.

Jiang, L.; Zhao, H.; Shi, S.; Liu, S.; Fu, C.-W.; and Jia, J. 2020. PointGroup: Dual-Set Point Grouping for 3D Instance Segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Kazemzadeh, S.; Ordonez, V.; Matten, M.; and Berg, T. L. 2014. ReferIt Game: Referring to Objects in Photographs of Natural Scenes. In *EMNLP*.

Lahoud, J.; Ghanem, B.; Pollefeys, M.; and Oswald, M. R. 2019. 3D Instance Segmentation via Multi-Task Metric Learning. In *The IEEE International Conference on Computer Vision (ICCV)*.

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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 5745–5753.

Liu, C.; and Furukawa, Y. 2019. MASC: multi-scale affinity with sparse convolution for 3d instance segmentation. *arXiv preprint arXiv:1902.04478* .

Liu, C.; Lin, Z.; Shen, X.; Yang, J.; Lu, X.; and Yuille, A. 2017. Recurrent multimodal interaction for referring image segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, 1271–1280.

Liu, D.; Zhang, H.; Wu, F.; and Zha, Z.-J. 2019a. Learning to assemble neural module tree networks for visual grounding. In *Proceedings of the IEEE International Conference on Computer Vision*, 4673–4682.

Liu, X.; Wang, Z.; Shao, J.; Wang, X.; and Li, H. 2019b. Improving referring expression grounding with cross-modal attention-guided erasing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1950–1959.

Margffoy-Tuay, E.; Pérez, J. C.; Botero, E.; and Arbeláez, P. 2018. Dynamic multimodal instance segmentation guided by natural language queries. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 630–645.

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

Pennington, J.; Socher, R.; and Manning, C. D. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 1532–1543.

Pham, Q.-H.; Nguyen, T.; Hua, B.-S.; Roig, G.; and Yeung, S.-K. 2019. JSIS3D: joint semantic-instance segmentation of 3d point clouds with multi-task pointwise networks and multi-value conditional random fields. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 8827–8836.

Qi, C. R.; Litany, O.; He, K.; and Guibas, L. J. 2019. Deep Hough Voting for 3D Object Detection in Point Clouds. In *Proceedings of the IEEE International Conference on Computer Vision*.

Qi, C. R.; Su, H.; Mo, K.; and Guibas, L. J. 2016. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. *arXiv preprint arXiv:1612.00593* .

Qi, C. R.; Yi, L.; Su, H.; and Guibas, L. J. 2017. PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. *arXiv preprint arXiv:1706.02413* .

Sadhu, A.; Chen, K.; and Nevatia, R. 2019. Zero-shot grounding of objects from natural language queries. In *Proceedings of the IEEE International Conference on Computer Vision*, 4694–4703.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *Advances in neural information processing systems*, 5998–6008.

Wang, P.; Wu, Q.; Cao, J.; Shen, C.; Gao, L.; and Hengel, A. v. d. 2019a. Neighbourhood watch: Referring expression comprehension via language-guided graph attention networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1960–1968.

Wang, W.; Yu, R.; Huang, Q.; and Neumann, U. 2018. Sgpn: Similarity group proposal network for 3d point cloud instance segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2569–2578.

Wang, Y.; Sun, Y.; Liu, Z.; Sarma, S. E.; Bronstein, M. M.; and Solomon, J. M. 2019b. Dynamic Graph CNN for Learning on Point Clouds. *ACM Transactions on Graphics* .

Yang, B.; Wang, J.; Clark, R.; Hu, Q.; Wang, S.; Markham, A.; and Trigoni, N. 2019. Learning object bounding boxes for 3d instance segmentation on point clouds. In *Advances in Neural Information Processing Systems*, 6740–6749.

Yang, S.; Li, G.; and Yu, Y. 2019a. Cross-modal relationship inference for grounding referring expressions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4145–4154.

Yang, S.; Li, G.; and Yu, Y. 2019b. Dynamic graph attention for referring expression comprehension. In *Proceedings of the IEEE International Conference on Computer Vision*, 4644–4653.

Yang, S.; Li, G.; and Yu, Y. 2020. Graph-Structured Referring Expression Reasoning in The Wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9952–9961.

Ye, L.; Rochan, M.; Liu, Z.; and Wang, Y. 2019. Cross-modal self-attention network for referring image segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 10502–10511.

Yi, L.; Zhao, W.; Wang, H.; Sung, M.; and Guibas, L. J. 2019. Gspn: Generative shape proposal network for 3d instance segmentation in point cloud. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3947–3956.

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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1307–1315.

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

Yu, L.; Tan, H.; Bansal, M.; and Berg, T. L. 2017. A joint speaker-listener-reinforcer model for referring expressions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 7282–7290.

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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4252–4261.