3D Visual Grounding with Graph and Attention

Yue Chen
Technical University of Munich
yue.chen@tum.de

Tao Gu
Technical University of Munich
tao.gu@tum.de

Abstract

The motivation of this project is to improve the accuracy of 3D visual grounding. In this report, we propose a new model, named HAIS_2GNN based on the InstanceRefer model, to tackle the problem of insufficient connections between instance proposals. Our model incorporates a powerful instance segmentation model HAIS and strengthens the instance features by the structure of graph and attention, so that the text and point cloud can be better matched together. Experiments confirm that our method outperforms the InstanceRefer on ScanRefer validation datasets.

1. Introduction

3D Visual Grounding, which tries to segment out target objects in a point cloud using a linguistic description, has emerged as a new topic in the fields of 3D computer vision and natural language processing. Unlike 2D images, 3D point clouds are more likely to be unordered and sparse, making it more challenging to locate objects and capture their relations. It's also difficult to extract precise object relations from descriptions and combine them with detected objects.

Our work will concentrate on utilizing more powerful object detectors, taking advantage of the two-stage aggregation feature of HAIS network [3] to improve the quality of proposals and reduce noise. More informational relationships among proposals will also be collected by utilizing candidate instances feature, nearest neighbor features, global instance features on adopted graph neural network structure. And finally, we will experiment with some improved fusion models, e.g. multi-attention network. Chen *et al*. [2] offered the *ScanRefer* dataset as a benchmark for this study.

2. Related Work

3D Visual Grounding For the benchmark *ScanRefer*, Chen *et al*. [2] proposed the first effort by concatenating the proposal features and language features and calculating the score. TGNN [7] and InstanceRefer [11] explored the usage of graphs to obtain the relations among proposals. Instead of graphs, 3DVG-Transformer [12] provided a model inspired by attention and transformer in both proposal generation and fusion stage.

3D Instance Segmentation PointGroup [8] created a novel end-to-end bottom-up architecture to identify object proposals using the original and offset-shifted point coordinate sets. Based on PointGroup, HAIS [3] removed the original set and refined the offset-shifted set by absorbing fragments and extracting features inside proposals.

3. Method

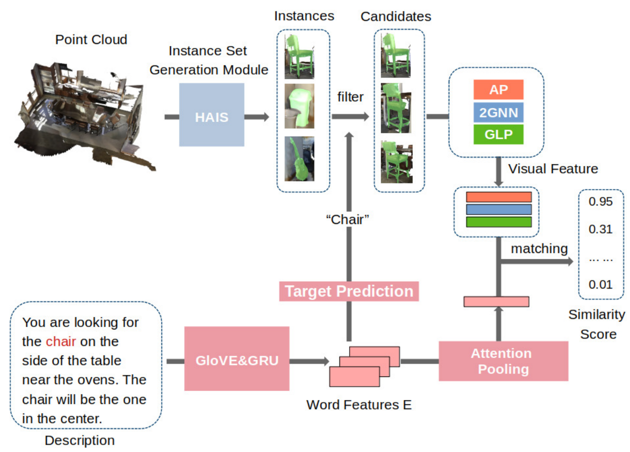
Our model HAIS_2GNN consists of four parts, the linguistic description part based on GRU for text feature extraction and target detection, the point group part based on the detector of HAIS model, the Multi-Level Visual Context module [11], and finally, a matching module, see Figure 1. Note that the InstanceRefer (baseline), for which the authors have released the source code to the open-source community, is not identical to the one described in its paper. We used the results of the open-source code as our baseline.

3.1. Instance Set Generation

Taking a point cloud $P \in \mathbb{R}^{N\times 3}$ and its features $F \in \mathbb{R}^{N\times 4}$ including RGB and height value as input, our model use HAIS to extract all instances point clouds $\mathbf{P}^I = \{P_i^I\}_{i=0}^M$, where P_i^I means the points of the i-th instance within the total M instances. Similarly, features and semantics of all instances are denoted as $\mathbf{F}^I$ and $\mathbf{S}^I$.

3.2. Description Encoding

We follow the language model in InstanceRefer and use the pre-trained GloVE word embedding with Bidirectional GRU layers and attention pooling to obtain the global representation of the query description, i.e., $\hat{E} \in \mathbb{R}^{1\times D}$. In additional, this model will predict a target category of the query, helping filter out the candidates from all instances.



You are looking for the chair on the side of the table near the ovens. The chair will be the one in the center.

Figure 1. The pipeline of our model HAIS_2GNN. It firstly uses the instance set generation module HAIS to extract all the instance point clouds in the large 3D scene. Under the guidance of target prediction from language description, the instances belonging to the target category are filtered out to form the initial candidates. In parallel, summarized language features are achieved through attention pooling. Subsequently, a visual-language matching module outputs the similarity score Q by comparing visual features from visual models (AP, 2GNN, GLP) against language features. Eventually, the 3D bounding box of the instance with the highest score is regarded as the final grounding result.

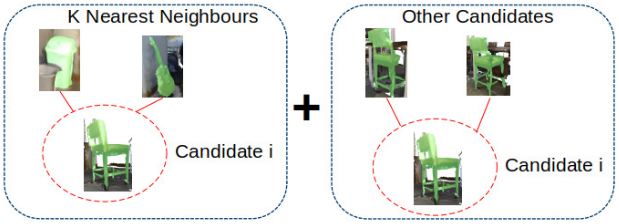


Figure 2. Illustration of 2GNN module. The first GNN (left) captures the relation between each candidate and its k nearest neighbours. The second GNN (right) passes message among all candidates in a scene.

3.3. Multi-Level Visual Context

AP Module We applied the same AP module in InstanceRefer. This module aims to capture features from attribute phrases. The candidate point cloud features $\hat{P}_i$ will be fed into a four-layer Sparse Convolution [6] and a max-pooling to obtain a global representation vector $\hat{F}_i^A \in \mathbb{R}^{1\times D}$.

2GNN module We enhanced the original RP module with an extra graph neural network (GNN). The RP module is proposed for capturing the relation among objects.

As shown in figure 2, for each candidate, the original RP module constructs a graph by connecting it and its K instance-level nearest neighbors. The node features $\bar{P}_i \in \mathbb{R}^{1\times(D+C)}$ for the i-th instance were generated by concatenating the average of its point cloud features $P_i \in \mathbb{R}^{1024\times D}$ and its semantic instance mask $S_i^I \in \mathbb{R}^{1\times C}$. The GNN with one single edge convolutional layer [9] can be formulated as:

$$\begin{aligned} r_{ik} &= \mathbf{MLP}([S_i^I; S_k^I; \text{RelativePosition}_{ik}]) \\ m_i &= \mathbf{MLP}([\bar{P}_i; \bar{P}_k; r_{ik}]) \\ \hat{F}_i^R &= \mathbf{MaxPool}(m_i) \end{aligned}$$

where r_{ik}, m_i, and $\hat{F}_i^R$ represent the edge features, the message, and the updated candidate features, respectively. S_i^I and S_k^I donate the semantic instance masks, indicating the predicted object class of the candidate i and its k-th neighbor. RelativePosition$_{ik}$ is the vector difference of the object center of candidate i and its k-th neighbor. The sign $[\cdot;\cdot]$ means the channel-wise concatenation.

On top of the original RP module, we additionally add a second GNN. The second GNN contributes to the relation encoding among candidates. As shown in figure 2, each candidate is now linked to other candidates. The node features $\bar{P}_i \in \mathbb{R}^{1\times(D+C)}$ remain the identical to the first graph. We use the same GNN layer in the previous graph, except that the index k ($k \neq i$) now corresponds to the k-th candidate. The final output $\hat{F}_i^R$ is the concatenation of the outputs of both graphs.

GLP Module We applied the same GLP module following InstanceRefer, which predicts the localization of target in one of the nine areas using the entire point cloud as input, then aggregates the language features to generate a global representation $\hat{F}_i^G \in \mathbb{R}^{1\times D}$.

3.4. Matching

With multiple visual instance features($\hat{F}^A$, $\hat{F}^R$, $\hat{F}^G$) and the language feature ($\hat{E}$), we perform a matching operation to get a confidence score to select the instance described. We directly compute the cosine similarity between visual and language features and sum them as a score. For the loss calculation of the matching part, we follow the contrastive manner in InstanceRefer. [11].

3.5. Loss Function

Loss is mainly composed of three parts, which are the matching loss $L_{\text{mat}} = L_{\text{AP}} + L_{\text{RP(2GNN)}} + L_{\text{GLP}}$, the target prediction loss from Language model L_{lang} and the target localization prediction loss L_{seg} in GLP module.

$$L = L_{\text{det}} + 0.1 * L_{\text{lang}} + 0.1 * L_{\text{seg}}$$

4. Experiments

In this section, we demonstrate the implementation details, the comparative analysis of our results as well as other experiments we investigated but performed negative results.

4.1. Implementation

Official pre-trained point groups are used in our experiments to perform panoptic segmentation. We use the same network architecture as the open-source code InstanceRefer for language encoding, AP, GLP, and matching. In the 2GNN module, the kNN instance number K is 8 for the first graph, the channel numbers of the edge convolutional layer are (25,128) and (128,128).

We train the network for 25 epochs with a batch size of 32 using the Adam optimizer. For the majority of experiments, the learning rate is set at 0.001. (In the Attention-based Fusion we used 0.0005 to acquire a similar speed of the loss gradient descent). All experiments are performed using PyTorch and an NVIDIA 3070Ti graphics card.

4.2. Dataset and Evaluation Metrics

ScanRefer. The ScanRefer dataset [2] is a newly proposed 3D scene visual grounding dataset, which consists of 51,583 descriptions of 11,046 objects from 800 ScanNet [5] scenes.

For the evaluation metrics, it calculates the 3D intersection over union (IoU) between the predicted bounding box and ground truth. The Acc@mIoU is adopted as the evaluation metric, where $m \in \{0.25, 0.5\}$. Accuracy is reported in “unique” and “multiple” categories. If only a single object of its class exists in the scene, we regard it as “unique”, otherwise “multiple”.

4.3. Quantitative Analysis

As demonstrated in table 1, our model achieved the highest Acc@0.5 scores on the validation set. Compared to our baseline, every score was improved apparently, reporting 8.1% in ”Unique”, 8.4% in ”Multiple”, and 8.3% in ”Overall” in Acc@0.5. Because our baseline results are similar to the results of InstanceRefer w/o MAT (paper), which has the exact same modules as our baseline, our results are reliable. In model InstanceRefer w/ MAT (paper), a co-attention MCAN [10] matching module (MAT) is implemented. The results of our model still outperform in both ”Unique” and ”Multiple” in Acc@0.5. Importantly, our efforts were not concentrated on the matching module. We may incorporate the MAT into our model to take advantage of the benefits of both.

Table 2 presents the ablation study for the two modules we proposed. There is a noticeable improvement by using HAIS. The 2GNN module can improve baseline performance marginally, and maintain the benefits for HAIS in ”Multiple” and ”Overall”. Although the score in ”Unique” was not increased by using 2GNN after HAIS, the overall improvement is remarkable.

4.4. Other Experiments

Module: Add extra node feature We noticed that messages in the original RP module were calculated by predicted object classes, object centers, averaged RGB color values and averaged height of neighbors. We thought the node features were too simplistic to encode meaningful neighbor information. Therefore, we added a two-layer sparse convolution [6] to acquire a rich feature for each instance. Concatenating with the original node features, we set the new features as node features and trained them with the same hyperparameters.

Table 3 shows a negative result. It might be due to the enormously increased trainable parameters. Additional sparse convolutional layers added 1/4 more parameters, making the training harder. Another possible reason is purposed in the next paragraph.

Module: 2GNN Series In the 2GNN module, instead of concatenating the outputs of both graphs (see figure 2), we attempted to let our candidates know each other after obtaining information about their neighbors. Therefore, we used the output of the first graph (containing neighbor information) as the node features for the second graph. This idea did not improve our model as shown in table 3. Similar to the previous idea, where we added more features to the nodes, the node features became long and noisy. The GNN or the edge convolutional layer [9] we applied in our model, might not be able to filter out the noise.

Module: Orientation Inspired by Scan2Cap [4], we take the message m_i obtained from the graph module in 3.3 as the output relation features, and pass it through an additional MLP to predict the angular between objects. We discretize the output angular deviations ranges from 0° to 180° into 6 classes and use a cross-entropy loss as the classification loss. We construct the ground truth labels using the transformation matrices of the aligned CAD models in Scan2CAD [1], and mask out objects not provided in Scan2CAD in the loss function.

As shown in table 3, the experimental results were not improved, and we assumed that this is because for each instance input $\hat{P}_i^I \in \mathbb{R}^7$ of the graph neural network, its features are only its center x, y, z, its size dx, dy, dz, and its semantic class S^I. It is difficult for the network to predict the angle between objects with this limited information, so this additional module failed to stabilize the learning process of the relational graph module as we expected.

Method	Unique		Multiple		Overall	
	Acc@0.25	Acc@0.5	Acc@0.25	Acc@0.5	Acc@0.25	Acc@0.5
Validation results						
ScanRefer [2]	65.00	43.31	30.63	19.75	37.30	24.32
InstanceRefer (baseline)	76.10	64.99	28.79	22.93	37.97	31.09
InstanceRefer w/o MAT (Paper) [11]	-	66.80	-	22.18	-	31.04
InstanceRefer w/ MAT (paper) [11]	77.45	66.83	**31.27**	24.77	**40.23**	32.93
HAIS_2GNN (ours)	**80.16**	**70.24**	29.39	**24.85**	39.24	**33.66**

Table 1. Comparative analysis of results. The results of InstanceRefer (baseline) are achieved by training on our hardware platform using the authors' open-source code. The results of InstanceRefer (paper) were released by the original paper [11]. Notably, the open-source code missed the visual-language matching (MAT) part and used different hyperparameters, resulting in lower scores. Since our model was based on the open-source code, we used the InstanceRefer (baseline) results as the baseline for our experiments.

HAIS	2GNN	Unique	Multiple	Overall
		64.99	22.93	31.09
✓		**70.41**	24.05	33.05
	✓	65.47	23.40	31.56
✓	✓	70.24	**24.85**	**33.66**

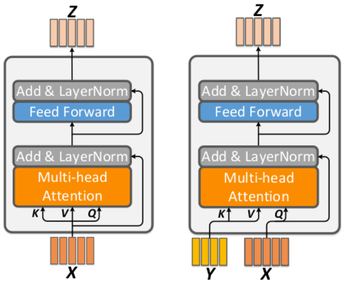
Table 2. Ablation study for different network architecture on ScanRefer validation set, where Acc@0.55 is used as metrics.

Method	Unique	Multiple	Overall
InstanceRefer (baseline)	64.99	22.93	31.09
Extra Node Feature	64.93	20.32	28.98
2GNN Series	64.66	21.68	30.02
Orientation	64.82	22.25	30.51
Attention-based Fusion	65.26	22.64	30.91

Table 3. Results of other experiments on the validation set, where Acc@0.5 is used as metrics. The first row represents the results of InstanceRefer (baseline). There are three experiments related to the RP model, besides adding more node features, we have the 2GNN Series, where we directly stack two GNNs. The Orientation method adds additional angular information between the objects. The last experiment is about the self-attention based fusion model.

Module: Attention-based Fusion We consider implementing the co-attention layer in MCAN [10] as originally mentioned in [11]. For the i-th instance, we concatenate the three visual features together $\hat{F}_i \in \mathbb{R}^{3\times D}$, and then employ three self-attention layers to aggregate the relationship between different visual contexts, as shown in figure 3 left side, the input of self-attention layer will be $\hat{F}_i$ as X.

Table 3 shows that the unique task improves, but overall it does not exceed the original model. We think the reason is that self-attention only considers attention within three perception models and does not aggregate the features of the language description.



(a) Self-Attention(SA) (b) Guided-Attention(GA)

Figure 3. Two basic attention units with multi-head attention for different types of inputs. SA takes one group of input features X and output the attended features Z for X; GA takes two groups of input features X and Y and output the attended features Z for X guided by Y. [10]

Therefore, a more sophisticated fusion model may aggregate language features with the help of guided-attention in MCAN [10]. As shown in figure 3 right side, the input of guided attention layer will be the original language features $E \in \mathbb{R}^{N\times D}$ as Y, where N is the query length, and the concatenated visual features $\tilde{F}_i \in \mathbb{R}^{1\times(3\times D)}$ as X for the i-th instance.

5. Conclusion

In this report, we propose an improved framework HAIS_2GNN based on the InstanceRefer model, for 3D visual grounding. Our model performs more accurate localization prediction via using a more powerful panoptic segmentation model HAIS and enhanced relation module. Specifically, our model innovatively adds a relation graph module between candidate instances to enrich the intra-candidate instance relations. We also discuss the possibility of adding additional features to the graph module, and directions for improving the fusion module.

References

[1] Armen Avetisyan, Manuel Dahnert, Angela Dai, Manolis Savva, Angel X Chang, and Matthias Nießner. Scan2cad:

Learning cad model alignment in rgb-d scans. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2614–2623, 2019. 3

[2] Dave Zhenyu Chen, Angel X Chang, and Matthias Nießner. Scanrefer: 3d object localization in rgb-d scans using natural language. *16th European Conference on Computer Vision (ECCV)*, 2020. 1, 3, 4

[3] Shaoyu Chen, Jiemin Fang, Qian Zhang, Wenyu Liu, and Xinggang Wang. Hierarchical aggregation for 3d instance segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15467–15476, 2021. 1

[4] Zhenyu Chen, Ali Gholami, Matthias Nießner, and Angel X Chang. Scan2cap: Context-aware dense captioning in rgb-d scans. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3193–3203, 2021. 3

[5] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5828–5839, 2017. 3

[6] Benjamin Graham, Martin Engelcke, and Laurens van der Maaten. 3d semantic segmentation with submanifold sparse convolutional networks, 2017. 2, 3

[7] Pin-Hao Huang, Han-Hung Lee, Hwann-Tzong Chen, and Tyng-Luh Liu. Text-guided graph neural networks for referring 3d instance segmentation. In *Proceedings of the Thirty-Fifth Conference on Association for the Advancement of Artificial Intelligence (AAAI)*, pages 1610–1618, 2021. 1

[8] Li Jiang, Hengshuang Zhao, Shaoshuai Shi, Shu Liu, Chi-Wing Fu, and Jiaya Jia. Pointgroup: Dual-set point grouping for 3d instance segmentation. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1

[9] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E. Sarma, Michael M. Bronstein, and Justin M. Solomon. Dynamic graph cnn for learning on point clouds, 2019. 2, 3

[10] Zhou Yu, Jun Yu, Yuhao Cui, Dacheng Tao, and Qi Tian. Deep modular co-attention networks for visual question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6281–6290, 2019. 3, 4

[11] 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1791–1800, 2021. 1, 2, 4

[12] Lichen Zhao, Daigang Cai, Lu Sheng, and Dong Xu. 3dvg-transformer: Relation modeling for visual grounding on point clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2928–2937, October 2021. 1