TransformerVG: 3D Visual Grounding with Transformers

Barry Shichen Hu Erik Schuetz
Technical University of Munich
{shichen.hu, erik.schuetz}@tum.de

Abstract

In this project, we perform the task of 3D visual grounding using an architecture that utilizes transformers. Existing approaches to this problem use an object detection module based on VoteNet and a fusion module that fuses language features with the detected object features to predict the final confidence scores. We propose TransformerVG, a transformer-based visual grounding pipeline that combines the 3DETR object detector with the transformer-based fusion model from the 3DVG pipeline. We outperform the ScanRefer baseline in the Acc@50 metric by 6% on the Benchmark through extensive experiments.

1. Introduction

3D Visual grounding is an emerging vision and language task that aims to locate objects in 3D scenes from an input point cloud using descriptions in natural language. It is challenging since, apart from accurate object detection, the models also need to understand the complex relations in 3D scenes to distinguish the true target from similar proposals. Due to their powerful relation modeling capability, Transformers can be an excellent method to apply to this task. This project combines 3DETR [6] and the fusion module of 3DVG [10] into one pipeline.
The contribution of this project is three-fold:

1. We conducted an extensive ablation study on the 3DETR pipeline with multiview data.

2. Various language modules are examined and compared for their language feature encoding performance.

3. A competitive Transformer-based pipeline that outperforms the ScanRefer pipeline [1] by a non-trivial margin.

2. Related Work

3D Visual Grounding. Various models use the "grounding-by-detection" strategy to solve the task in two stages. Chen et al. [1] first detect objects in the scene using VoteNet, then fuse the detected objects' features with the language embedding to produce the final localization proposal. Zhao et al. [10] follow the same structure using a Transformer-based vision-language fusion stage instead.

Transformer-based 3D object detection. The Transformer-based 3D object detection methods by Liu et al. [5] and Misra et al. [6] achieve great results on the ScanNet Benchmark. While Liu et al. [5] utilize an ensemble consisting of intermediate outputs from layers of decoder blocks for the final prediction, Misra et al. [6] make minimal changes to the vanilla transformer block and achieve end-to-end object detection on point clouds. Transformer-based methods incorporate the global relations through the attention mechanism while VoteNet-based detectors predict bounding boxes only based on votes on a local scale.

3. Method

3.1. Overview

The architecture proposed in this project contains three main modules: 1) detection module, 2) matching, and localization module, and 3) language module. Figure 1 shows the architecture used in this project. The model uses point clouds and textual object descriptions as inputs. Sampled point clouds are input into the detection module, and textual descriptions are input into the language module. The detection module outputs object proposals as bounding boxes and their corresponding features, while the language module encodes the descriptions and outputs features for each word. Finally, the matching and localization module fuses the outputs of both the detection and the language module to output the reference localization confidences.

3.2. Detection Module

As stated in the introduction, the Transformer-based detector by Misra et al. [6] is used as the detection module. 3DETR closely follows the vanilla Transformer structure introduced by Vaswani et al. [9] with an encoder and a decoder part. It takes as an input a 3D point cloud and outputs

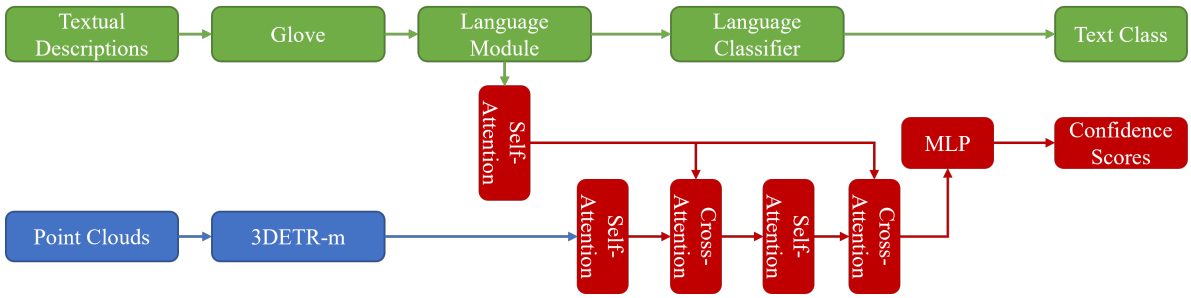

Figure 1. The architecture of TransformerVG that is used in this project. Steps related to the language module are shown in green, steps related to the detection module are in blue, and steps related to the fusion module are in red.

3D bounding boxes around detected objects. For this, Misra et al. [6] use XYZ-coordinates and RGB raw values as input features. These features are grouped by the set-aggregation operation introduced in [8] and projected into the encoder dimension. The encoder applies multiple layers of self-attention and MLPs to the input data. As Misra et al. [6] explain, positional embeddings can be omitted since the XYZ-coordinates are present in the input features. A parallel decoder composed of multiple Transformer blocks produces a set of features using the XYZ-coordinates of the sampled points from the input as queries. Multiple MLP heads use these features to predict the properties of the 3D bounding boxes. Inspired by a performance increase recorded by Chen et al. [1], we decided to train our model using XYZ-coordinates with additional features provided by ScanNet: multiview, normals, and heights. The 3DETR model used in this project has been changed accordingly to be able to use these enriched input features.

3.3. Language Module

Following Chen et al. [1], the language model first encodes each word into a 300-dimensional feature vector using the embedding from Pennington et al. [7]. Then these word embeddings are further processed to extract features and relations. A separate feature vector for each word and a global language representation for the description are produced. This global representation is used for predicting the class of the queried object. Multiple models can be used for this operation: self-attention, transformer encoder, GRU, and Bert. The feature vector for each word is used for vision-language fusion in the matching module. The empirical results with these models are reported in Section 4.2 "Ablation Studies and Analysis".

3.4. Fusion Module

The project description sets the choice for the fusion module. TransformerVG uses the fusion module introduced by Zhao et al. [10]. The fusion model consists of self-attention and cross-attention layers. The self-attention layers follow the vanilla multi-head attention structure introduced in [9] with the aid of an additional spatial proximity matrix by Zhao et al. [10]. They refine and capture the spatial relations between bounding box proposals, which are then used for querying the language features in the cross-attention. Finally, the output of the last attention layer is passed through an MLP to produce the confidence scores for each object proposal.

3.5. Loss Function

To engineer the loss function for this pipeline, we closely follow the approaches of Chen et al. [1] and Zhao et al. [10]. Our final loss is calculated as:

$$\mathcal{L} = 0.3\mathcal{L}_{loc} + \mathcal{L}_{det} + 0.1\mathcal{L}_{cls} \quad (1)$$

where $\mathcal{L}_{loc}$, $\mathcal{L}_{det}$, and $\mathcal{L}_{cls}$ stand for matching loss, detection loss, and language classification loss. To achieve the final loss, $\mathcal{L}$ is amplified by a factor of 10, as proposed by Chen et al. [1].

4. Experiments

4.1. Datasets and Implementation Details

Datasets. We train and evaluate different configurations of 3DETR and TransformerVG on the ScanNet [2] and ScanRefer datasets [1].

-*ScanNet:* ScanNet is an RGB-D video dataset that can be used for object detection. It contains 2.5 million views in more than 1,500 scans, annotated with 3D camera poses, surface reconstructions, instance-level semantic segmentation, and axis-aligned bounding boxes. This dataset contains 9,677 train instances and 2,606 test instances from 20 different categories. We follow Zhao et al. [10] by using the Mean Average Precision (mAP) at IoU thresholds of 0.25 and 0.5, denoted as AP_{25} and AP_{50}, to evaluate the detection performance.

-*ScanRefer:* ScanRefer is a visual grounding dataset based on the RGB-D scans of ScanNet. It has 51,583 textual descriptions of 11,046 objects from 800 scenes. These descriptions are based on the objects' properties and surround-

ings. We use Acc@0.25IoU and Acc@0.5IoU as our metrics, which is the percentage of correctly predicted bounding boxes whose IoU with the ground-truth bounding boxes is above 0.25 and 0.5. As in Chen et al. [1], we differentiate between "unique" and "multiple" scenes. "Unique" scenes contain only a single object from the target class, while "multiple" scenes contain more than one object. We evaluate our model by evaluating its performance on both the validation set and the ScanRefer Benchmark website.

Implementation Details. The experiments are run with the PyTorch framework on three different GPUs: NVIDIA RTX 2080Ti, 3080Ti, and 3090. We use a 3DETR detection module, which is pretrained on ScanNet with multiview, normals, heights, and XYZ features for 1,080 epochs with a batch size of 6 on the 3080Ti, for all experiments. We further tuned the hyperparameter settings from Misra et al. [6] for this pretraining. We train our complete pipeline with a frozen detection module except for the MLP feature heads and the last decoder layer. We closely follow the training details in [6] and [10]. The model is optimized using the AdamW optimizer with learning rates of 1e-6, 5e-4 and 5e-4 for the detection module, the language module, and the matching module respectively. We apply a cosine learning rate scheduler with a weight decay factor of 1e-5 for every module except the detection module. In addition, we use gradient clipping at an l_2 norm of 0.1.

4.2. Ablation Studies and Analysis

In this section, we discuss the contribution and influence of each module on the performance of the complete pipeline. We also conduct further analysis of the performance with different parameters.

3DETR. For this study, we use the validation set of ScanNet to compare the performance of the different experiments. We have conducted an ablation study on different input features with two different versions of 3DETR. As shown in Table 1, 3DETR-m performs better than 3DETR with both sets of features. In addition, 3DETR-m produces a significantly higher mAP_{25} score and similar mAP_{50} score when trained with XYZ, multiview, normals, and heights. That is why we use these input features for all further studies.

To better cope with the increased number of input features, we compare different 3DETR-m models with varying pre-encoder dimensions. Table 2 shows the results of this analysis. The models were trained for 1,080 epochs on the ScanNet dataset using XYZ, normals, multiview, and heights as input features. Since the ScanRefer Benchmark uses the Acc@0.5IoU as their main criteria, we choose to use the detector with the best mAP_{50} score for the visual grounding pipeline.

The results displayed a significant amount of overfitting when training the different pre-encoder configurations. We evaluate the model performance with different dropout rates to reduce this overfitting, as shown in Table 3. Based on the mAP_{50} scores, we choose to use a dropout rate of 0.5 for our further experiments.

We implement the pretrained 3DETR-m in the ScanRefer pipeline and achieve better results with all metrics compared to the ScanRefer baseline as shown in Table 4.

3DVG. We incorporate the 3DVG matching module into our pipeline and compare its performance with the original ScanRefer pipeline. As indicated in Table 5, this pipeline outperforms the ScanRefer baseline. Furthermore, we evaluate the performance of the complete pipeline when changing the depth of the fusion module. As shown in Table 6, the performance in both metrics degrades with increasing depth, which complies with the conclusion of Zhao et al. [10]. Thus, the fusion module's depth is chosen as 2 in the further experiments.

Model and Feature Set	mAP_{25}	mAP_{50}
3DETR XYZ+rgb	61.29	35.92
3DETR XYZ+multiview+normals	66.68	28.37
3DETR-m XYZ+rgb	64.84	**45.71**
3DETR-m XYZ+multiview+normals	**70.39**	45.69

Table 1. 3DETR ablation study with different features. Heights are included as a default features.

Model	pre-encoder dimensions	mAP_{25}	mAP_{50}
3DETR-m	(135, 64, 128, 256)	**71.19**	40.46
3DETR-m	(135, 128, 128, 256)	70.62	**43.61**
3DETR-m	(135, 256, 256, 256)	68.18	40.69

Table 2. 3DETR ablation study with different features using an encoder dropout rate of 0.3.

Model	Encoder Dropout Rate	mAP_{25}	mAP_{50}
3DETR-m	0.3	70.62	43.61
3DETR-m	0.4	**70.75**	45.00
3DETR-m	0.5	70.39	**45.69**

Table 3. 3DETR ablation study with different encoder dropout rates using the (135,128,128,256) pre-encoder configuration.

	acc@0.25	acc@0.5
ScanRefer(Vanilla)	41.19	27.40
ScanRefer(3DETR-m)	**42.60**	**30.37**

Table 4. Comparison between ScanRefer with 3DETR-m and vanilla ScanRefer, trained with XYZ+multiview+normals+lobcls.

(a) Language inputs (b) ground truth bounding boxes (c) predicted bounding boxes

Figure 2. Qualitative results from TransformerVG. The textual descriptions and their corresponding bounding boxes are in the same color.

	acc@0.25	*acc*@0.5
ScanRefer(Vanilla)	41.19	27.40
3DETR-m+3DVG	**42.24**	**29.16**

Table 5. Comparison between 3DETR-m+3DVG and vanilla ScanRefer, trained with XYZ+multiview+normals+lobcls.

Model	Transformer Depth	*acc*@0.25	*acc*@0.5
TransformerVG	2	**44.20**	**33.71**
TransformerVG	4	43.34	29.73
TransformerVG	6	41.56	29.10

Table 6. Ablation study on the depth of the fusion module, trained with XYZ+multiview+normals+lobcls.

Language Module. We conducted an ablation study on different language modules to evaluate their effects on the complete pipeline's performance. For attention-based and Transformer-based language modules, we used PyTorch's implementation. For the Bert-based language module, we used the pretrained model by Huggingface [3]. As expected, attention masks improve the performance of all types of language modules, and we conclude that GRU provides the best results. Table 7 shows the detailed results.

4.3. Qualitative results

We evaluate the performance of TransformerVG on the validation set of ScanRefer. Our model is able to correctly localize both unique objects and multiple objects. A sample visualization is in Figure 2 .

4.4. Comparisons with the state-of-the-art methods

We submitted our model to the ScanRefer benchmark. As shown in Table 8, our model can outperform the original ScanRefer pipeline and TGNN [4] yet still performs worse than state-of-the-art methods like 3DVG-Transformer and D3NET (acc@0.25IoU: 0.4806, acc@0.5IoU: 0.3919).

language module	overall *acc*@0.25	*acc*@0.5
gru	**44.20**	**33.71**
gru (w/o attention mask)	43.35	29.59
self-attention	38.66	29.99
self-attention (w/o attention mask)	36.56	25.75
transformer encoder	42.33	32.45
transformer encoder (w/o attention mask)	40.88	28.32
pretrained bert	42.56	29.33
pretrained bert (w/o attention mask)	42.26	29.55

Table 7. Ablation study on TransformerVG with different language modules.

Model	*acc*@0.25	*acc*@0.5
InstanceRefer	44.27	**35.80**
3DVG-transformer	**49.72**	35.12
ScanRefer	42.44	26.03
TGNN	41.02	32.81
TransformerVG (Ours)	45.62	33.79

Table 8. Comparison of TransformerVG with other methods on the Benchmark.

5. Conclusion

We have introduced a new visual grounding pipeline, called TranformerVG, that outperforms the competitive baseline ScanRefer. It consists of a Transformer-based object detector (3DETR) and a Transformer-based fusion module (3DVG). Furthermore, we have conducted an ablation study on 3DETR and have concluded that using multi-view data with a higher dropout rate further boosts the detection performance recorded by Misra et al. [6]. In addition, our work showcases the capacity of transformers in fusing multi-modality data, in this case, natural language and 3D object detection. We are expecting future research that explores the performance of Transformers on this task.

References

[1] 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, 2, 3

[2] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2017. 2

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019. 4

[4] Pin-Hao Huang, Han-Hung Lee, Hwann-Tzong Chen, and Tyng-Luh Liu. Text-Guided Graph Neural Networks for Referring 3D Instance Segmentation. In *AIII*, 2021. 4

[5] Ze Liu, Zheng Zhang, Yue Cao, Han Hu, and Xin Tong. Group-free 3d object detection via transformers. *arXiv preprint arXiv:2104.00678*, 2021. 1

[6] Ishan Misra, Rohit Girdhar, and Armand Joulin. An End-to-End Transformer Model for 3D Object Detection. In *ICCV*, 2021. 1, 2, 3, 4

[7] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014. URL http://www.aclweb.org/anthology/D14-1162. 2

[8] Charles R. Qi, Li Yi, Hao Su, and Leonidas J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space, 2017. 2

[9] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2017. 1, 2

[10] 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

Supplementary Material

A. Additional Visualization

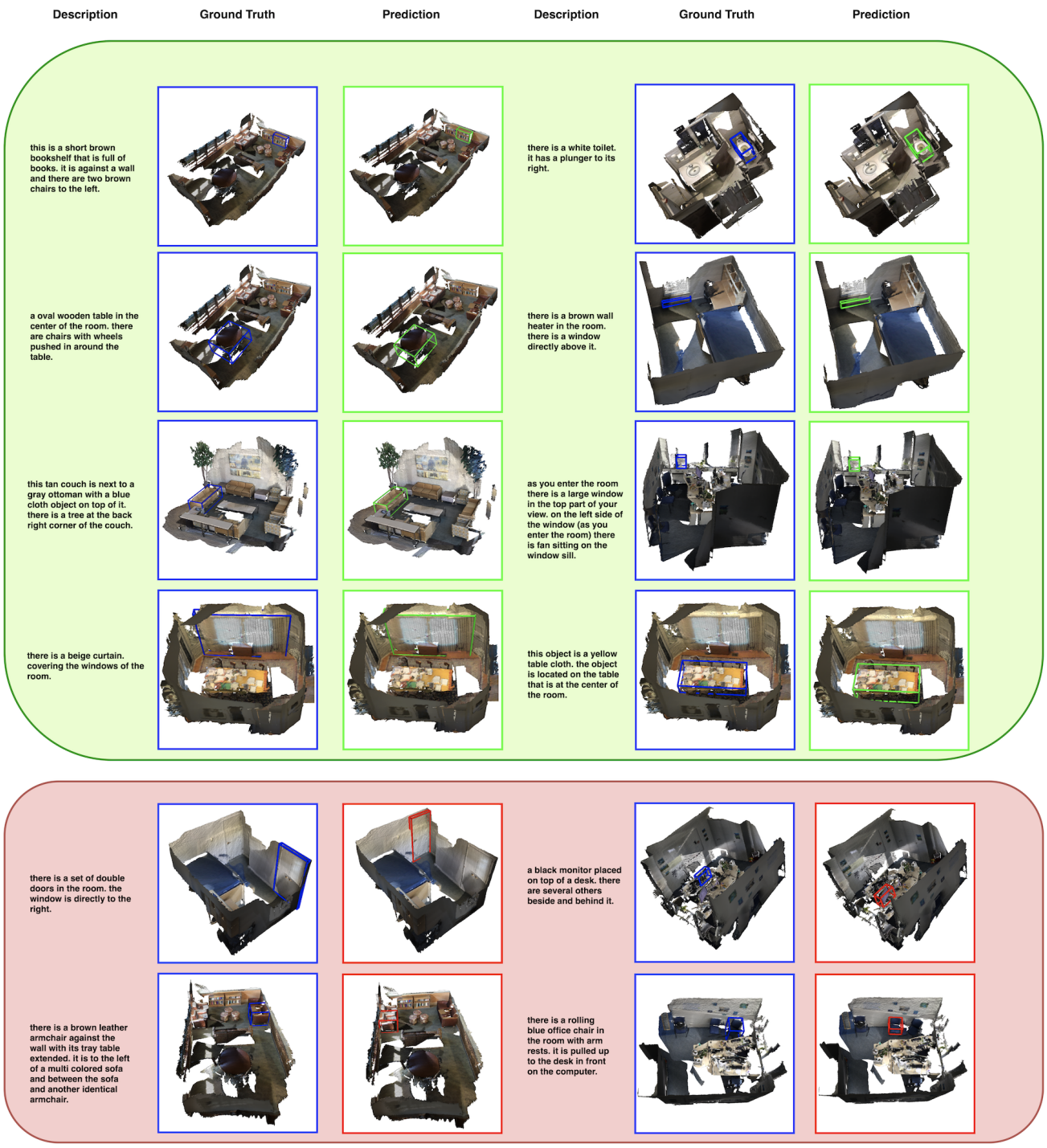

Fig. 3. Visualization of TransformerVG. Green predictions are correct predictions, and red predictions are wrong. Ground truth bounding boxes are in blue.