3D Visual Grounding with Transformers

Stefan Frisch
Technical University of Munich
stefan.frisch@tum.de

Florian Stilz
Technical University of Munich
florian.stilz@tum.de

Abstract

3D visual grounding lies at the intercept of 3D object detection and natural language understanding. This work focuses on developing a transformer architecture for bounding box prediction around a target object that is described by a text description. Transformers are the superior choice for 3D object detection compared to the previously often employed VoteNet as they are capable of capturing the long-range context better and can operate on variable-sized inputs. Therefore, they are well suited to operate on 3D point clouds. We show that employing 3DETR-m as an object detection framework improves the 3D visual grounding performance. Additionally, we introduce a transformer into the matching of the textual and visual features since it is uniquely suited to capture the relevant interdependencies. Our method achieves around 2.5% improvement over the ScanRefer architecture by replacing the object detection with 3DETR-m and adding a vanilla transformer encoder to the matching module.

1. Introduction

3D visual grounding is the task of localizing a target object in a 3D scene given a text description. Performing this task accurately will be beneficial for many real-world applications such as AR/VR, autonomous robots, etc. 3D scenes can be represented by point clouds, which encode both geometrical and other point features. One of the most recent models is ScanRefer by Chen et al. [2] which uses VoteNet [1] for 3D object detection, and a basic concatenation with MLP to fuse and localize the final bounding boxes. However, ScanRefer struggles to handle complex spatial relations within point clouds as well as differentiate accurately similar object proposals using solely a text description.

One way to address these issues is to use transformers since they can operate on variable-sized inputs and encode relational aspects in scenes that enrich the visual grounding task. We developed a transformer-based architecture for 3D visual grounding, which takes as input a point cloud and a text description of an object and outputs a bounding box around the described object. For the 3D object detection we utilize the **3DETR-m**, which outperforms VoteNet significantly on the ScanNetV2 dataset [5]. Furthermore, a **vanilla transformer encoder** is used to tackle the localization issue in ScanRefer by adding contextual awareness to our model before the final localization. Our model thus makes better use of the contextual knowledge within the point clouds and across the linguistic and visual features. This awareness is provided by the attention mechanism in the encoder architecture. Transformers have also become the default choice for natural language processing specifically large language models. We introduce BERT as a language encoder into the 3D grounding task and discuss the results. Our contributions are the following:

- Introduction of the transformer-based object detection framework 3DETR into the 3D visual grounding task.
- With the help of chunking the data reduction of training time by about 500% compared to ScanRefer.
- Our novel transformer-based architecture improves over the baseline ScanRefer by more than 2.5%.

2. Related work

2.1. 3D object detection

Point cloud based 3D object detection is a well-studied field. PointNet [6] and PointNet++ [7] have been used as the backbone for several 3D object detection and 3D object segmentation. VoteNet [1] is a 3D object detection that uses Hough Voting on sparse point cloud input. Misra et al. [5] introduced 3DETR, an end-to-end 3D object detection transformer-based model that achieves competitive results on the ScanNetV2 [3] and SUN RGB-D-v1 [8] datasets. Our object detection model is based on the 3DETR model.

2.2. 3D visual grounding

Chen et al. introduced the ScanRefer dataset [2] and proposed a 3D grounding framework with the identical name. As a first step, object proposals are generated as well as features for the description and the proposed objects. Afterwards, the object and textual features are fused to predict the target object. Zhao et al. introduced 3DVG [10], a model that follows the same paradigm but relies partly on a transformer architecture to better utilize the contextual clues.

3. Method

3.1. Network architecture

Our architecture is based on the ScanRefer [2]. We experiment with the replacement of each sub module (object detection, language encoding, multimodal fusion) with an attention-based model and also the combination of these. For the language encoding module, we tested the well-known Bert [4] architecture to get richer language features than the GRU module that is based on GloVE embeddings. Notably, the Bert model was not employed as a final language encoder, since we found that it hurts the performance while increasing the training time. A detailed ablation study of the language encoder can be found in section 4. In the object detection part we rely on the transformer-based 3DETR [5] architecture instead of VoteNet [1]. Similar to VoteNet 3DETR uses a PointNet++ backbone to get the number of points to a manageable level. This is especially important since 3DETR is an attention-based architecture with a quadratic runtime in the number of points. For details on the 3DETR architecture the reader may refer to Misra et al. [5]. In the fusion and localization module, we utilize a vanilla transformer encoder architecture as it was proposed by Vaswani et al. [9]. The self-attention from a transformer encoder is applied after the concatenation of the features to better represent dependencies between the features. We found that the best results where achieved with 5 encoder layers with 8 self-attention heads each. The final architecture can be observed in figure 1.

3.2. Loss Function

We adjusted the loss function from ScanRefer to accommodate the 3DETR object detection part. In addition, we increased the loss weight of the localization loss in order to stronger emphasize the localization task. For more details on the object detection loss we refer to the Misra et al. [5] and for the language classification loss as well as the localization loss to Chen et al. [2]. The final loss of our method is then built as follows:

$$\mathcal{L}_{final} = 1 \cdot \mathcal{L}_{obj} + 0.1 \cdot \mathcal{L}_{cls} + 1 \cdot \mathcal{L}_{loc}$$

3.3. Training

Our final model was trained with a learning rate of 2e-4 and a batch size of 7.

Chunking We implemented a chunking mechanism for the ScanRefer architecture following Zhao et al. [10] to reduce the training time for the larger models to more manageable level. Chunking exploits the fact that in the ScanRefer dataset there are multiple objects in one 3D scene. So the object detection part in theory only has to run once per scene. In practice the chunk size determines how many objects are simultaneously processed. With a chunk size of 8 the training time for 50 epochs could be reduced by more than 500% from 25 hours to 4 hours and 17 minutes on one current GPU (NVIDIA Tesla T4), while losing 1-2% IoU accuracy. The detailed results can be observed in table 1.

Model	Acc@0.25	Acc@0.5	Duration 50 epochs
ScanRefer	**36.65**	**23.71**	25h
ScanRefer chunking 8	35.66	22.01	4h 17min
ScanRefer chunking 16	33.08	18.08	**3h 11min**

Table 1. ScanRefer training duration and accuracy with different chunk sizes.

Model	AP@0.25	AP@0.5
VoteNet	52.29	26.27
3DETR-m	**56.36**	**31.67**

Table 2. Pretraining 3D Object Detection on 3D scans in ScanRefer dataset with xyz + rgb as input. The metric is the average precision for the IoU@0.5 and IoU@0.25 scores

Pretraining We believe that the loss in accuracy while chunking stems from the fact that the object detection part is trained less often than regular training. We mitigate this loss by using a pretrained object detection model. The pretraining of the object detection part was performed by training solely on the 3D scans in the ScanRefer dataset using the corresponding settings from Misra et al. [5] for 3DETR-m and Qi et al. [1] for VoteNet. The comparison hereby showed that 3DETR-m clearly outperforms VoteNet as can be seen in table 2.

4. Experiments

All our results are reported on the validation set of the ScanRefer dataset following Chen et al. [2].

Metric The metric used in order to evaluate and compare model performance is the IoU metric. It indicates the overlapping area of the predictions and ground truth boxes divided by the union area of the those two boxes. Acc@0.5 hereby means the accuracy for having an IoU score greater than or equal to 0.5.

Bert The Bert architecture as a language encoder was extensively tested to check whether Bert can produce richer language features than the baseline GRU. The tests for the optimal number of Bert layers can be observed in the table 3. Bert did not improve the quality of the overall predictions while increasing the training time significantly. Our hypoth-

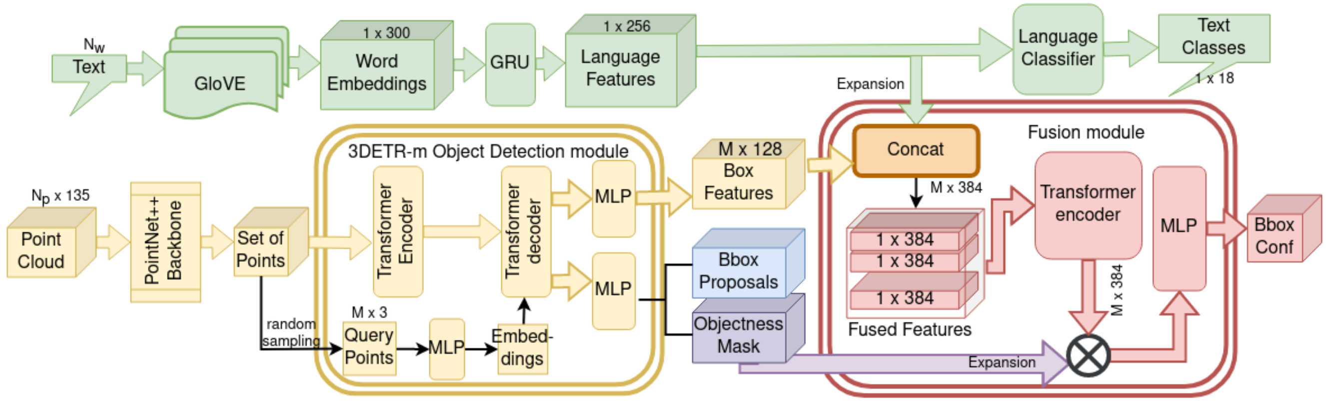

Figure 1. Our final architecture: The PointNet++ [7] aggregates the pointcloud to higher level features that are then passed to the 3DETR-m [5] object detection module. The object detection module passes high level features to the proposal module, which essentially is a set of MLP's that output bounding box proposals and an objectness mask. The descriptions are first passed through a pretrained GloVE and are then processed by a GRU to obtain the language features like in ScanRefer [2]. The language features and the object features are then concatenated and run through a vanilla transformer encoder [9]. As a final step non objects are masked by the objectness mask and a MLP is applied to output confidence scores.

Model	Acc@0.25	Acc@0.5	Duration 50 epochs
ScanRefer (VoteNet + **GRU** + concat)	**35.66**	**22.01**	**4h 17min**
VoteNet + **Bert layer 3** + concat	34.45	20.93	8h 50min
VoteNet + **Bert layer 5** + concat	34.75	21.06	10h 50min
VoteNet + **Bert layer 12** + concat	34.22	21.10	18h

Table 3. Comparison of accuracy and training time for different numbers of Bert layers. All results are reported with chunk size 8 and learning rate 1e-3. Only the Bert language encoder was trained with a learning rate of 5e-5 in every case.

Model	unique		multiple		overall	
	Acc@0.25	Acc@0.5	Acc@0.25	Acc@0.5	Acc@0.25	Acc@0.5
pretrained VoteNet + **GRU** + concat	62.71	41.84	**30.22**	**19.57**	**36.53**	**23.90**
pretrained VoteNet + **BERT layer 12** + concat	**63.79**	**42.16**	28.57	17.80	35.40	22.53

Table 4. Comparison of accuracy for one unique object of one category in the scene with multiple objects of one category.

esis is that Bert focuses too much on the semantic meaning rather than on the relationship between objects. We tested the performance if there is a single object of its class in the scene compared to if there are multiple objects of the same class in the scene. The results from table 4 seem to confirm the hypothesis since Bert outperformed GRU in the unique object category while failing multiple object cases.

3DETR-m The following section provides an overview over the best results when using 3DETR-m as the object detection module. The comparison between 3DETR-m and VoteNet for different input features shows that 3DETR-m helps to provide slightly better bounding boxes as it's results show some improvements on the validation set as can be seen in table 5. The improvement of 3DETR-m over VoteNet is smaller than predicted especially considering the promising results from table 2.

Model	Acc@0.25	Acc@0.5
Input: xyz		
pretrained **VoteNet** + GRU + concat	**37.77**	24.69
pretrained **3DETR-m** + GRU + concat	35.53	**25.25**
Input: xyz + rgb		
pretrained **VoteNet** + GRU + concat	**37.11**	25.21
pretrained **3DETR-m** + GRU + concat	35.00	**25.50**

Table 5. Ablation study with pretrained 3DETR-m and VoteNet in comparison

Matching Module We found that adding a vanilla transformer encoder after the concatenation and before the localization module, helped the localization of the right object. The intuition here is that the self-attention within the encoder helps to get better features due to the stronger emphasize on the dependency relations between the box and texual features.

Final Model Table 7 depicts that our architecture improves the baseline ScanRefer by almost 2.5% on the IoU 0.5 accuracy indicating that a transformer architecture helps the 3D visual grounding task.

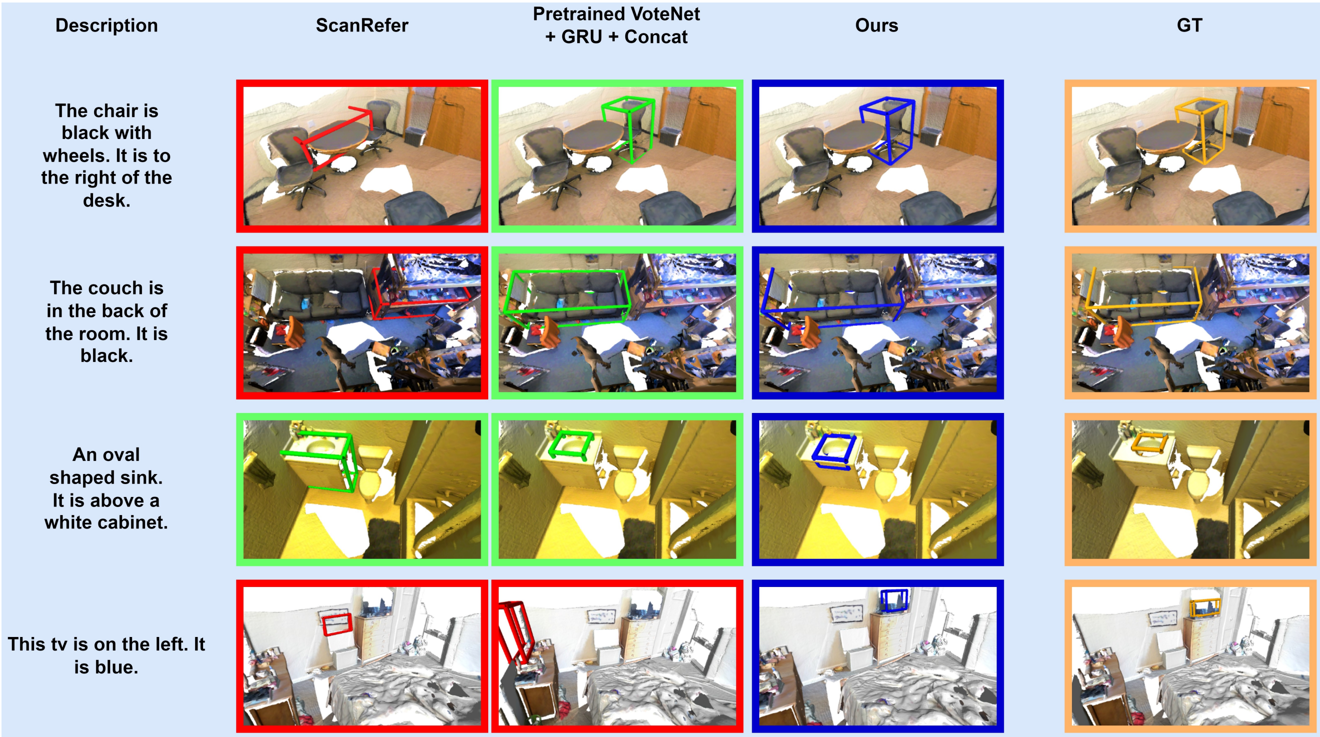

Figure 2. Qualitative Analysis for baseline model ScanRefer, pretrained VoteNet with chunking and our final transformer architecture. The predicted bounding boxes are marked green if they predict the correct object, red if they predict a wrong object, and blue if it is the best prediction out of all three predictions. The ground truth (GT) box is orange

Model	Acc@0.25	Acc@0.5
pretrained 3DETR-m + GRU + **concat**	35.00	25.50
pretrained 3DETR-m + GRU + **vTransformer 2 Layers**	36.59	26.23
pretrained 3DETR-m + GRU + **vTransformers 4 Layers**	36.57	26.23
pretrained 3DETR-m + GRU + **vTransformer 5 Layers**	37.08	**26.56**
pretrained 3DETR-m + GRU + **vTransformer 6 Layers**	**37.11**	26.15

Table 6. Ablation study with vanilla transformer using xyz + rgb as input

5. Qualitative Analysis

Figure 2 shows our qualitative analysis for the ScanRefer baseline model, the optimized baseline namely ScanRefer with pretrained VoteNet as well as our final model. The blue bounding boxes under the section ”Ours” indicate that our model not only managed to predict the correct objects, but also managed to produce the best bounding boxes in comparison to the other models in each scenario displayed.

Model	Acc@0.25	Acc@0.5
ScanRefer	37.05	23.93
pretrained VoteNet + GRU + concat	**37.11**	25.21
pretrained 3DETR-m + GRU + concat	35.00	25.50
Ours (pretrained 3DETR-m + GRU + vTransformer)	37.08	**26.56**

Table 7. Model comparison on ScanRefer dataset with xyz + rgb as input

6. Conclusion

In this project we showed that a transformer-based architecture for object detection as well as the matching module helps to achieve better results in 3D visual grounding. We could also significantly decrease the duration of training with the help of chunking. The performance loss of chunking could be eliminated by employing pretrained object detection models. The replacement of VoteNet with 3DETR-m for the object detection part and the addition of a vanilla transformer encoder for the matching module improved the overall performance on the ScanRefer dataset compared to the original ScanRefer model by more than 2.5%.

References

[1] Kaiming He Leonidas J. Guibas Charles R. Qi, Or Litany. Deep hough voting for 3d object detection in point clouds. In *In 2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019. 1, 2

[2] Dave Zhenyu Chen, Angel X Chang, and Matthias Nießner. Scanrefer: 3d object localization in rgb-d scans using natural language. In *European Conference on Computer Vision*, pages 202–221. Springer, 2020. 1, 2, 3

[3] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes, 2017. 1

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2018. 2

[5] Ishan Misra, Rohit Girdhar, and Armand Joulin. An end-to-end transformer model for 3d object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2906–2917, 2021. 1, 2, 3

[6] Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation, 2016. 1

[7] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017. 1, 3

[8] Shuran Song, Samuel P. Lichtenberg, and Jianxiong Xiao. Sun rgb-d: A rgb-d scene understanding benchmark suite. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 567–576, 2015. 1

[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. 2, 3

[10] Lichen Zhao, Daigang Cai, Lu Sheng, and Dong Xu. 3dvg-transformer: Relation modeling for visual grounding on point clouds. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2908–2917, 2021. 2