3D-XVIGR: End-to-End Transformer Model for 3D Visual Grounding on Point Clouds

Tony Wang, Dave Zhenyu Chen
Technical University of Munich
{firstname}.{lastname}@tum.de

Abstract

In this work, we propose 3D-XVIGR[1], a novel transformer-based model for the 3D visual grounding task. The core objective of this task is to accurately localize 3D objects using text descriptions by aligning visual and natural language data. To achieve this, we formulate our approach by leveraging the aptitude of transformers for 3D point cloud detection and multi-modal matching. Moreover, we propose a novel decoder layer that inputs object query proposals, point cloud features, and word features to foster a more robust alignment between the textual descriptions and their corresponding 3D visual representations. Our model performs comparably on the ScanRefer dataset, even without using hand-crafted data augmentation techniques employed by the 3DVG Transformer, our baseline in this work. Consequently, our method presents a distinct and more streamlined approach for this task. 3D-XVIGR also yields promising results in the qualitative analysis, accurately associating textual descriptions with 3D point clouds. This work, therefore, provides insights into the challenges and opportunities associated with transformer-based solutions for 3D visual grounding. We believe that the findings offer valuable contributions to ongoing discourse and developments in this area, setting the stage for further enhancements to the 3D-XVIGR architecture. We have also released the code-base for 3D-XVIGR on GitHub[2].

1. Introduction

Visual grounding on point clouds (also known as referring 3D object localization) is an emerging task in 3D visual understanding. It aims to identify and locate objects or regions within an input point cloud based on given textual descriptions. This technology has significant implications for real-world applications like autonomous robots, augmented reality (AR), and virtual reality (VR).

Developing a reliable point-based visual grounding scheme still remains a challenge, for the main difficulty lies in comprehending the natural language and complex relations within 3D scenes and distinguishing the target object proposals from similar ones. Furthermore, the relatively small scales of recent visual grounding datasets have led to the overfitting problem, hindering these methods from learning a generalizable visual grounding model.

Transformers, renowned for their capability to handle irregular and unordered data, present a promising solution. Their self-attention mechanism can further be used as cross-attention to capture relationships between different data modalities like 3D geometry and natural language. These unique features of transformers make them particularly adept at processing point cloud data and performing 3D visual grounding.

We recognize this potential and introduce 3D-XVIGR, a simple end-to-end learnable transformer-based architecture for visual grounding tasks in 3D point clouds. Our model leverages the recent advancements in transformers for 3D point cloud detection [7] and 3D visual grounding [13] to explore the capabilities of transformers for 3D visual grounding. 3D-XVIGR addresses the complexity within 3D scenes by capturing the relations among points, queries, and proposals to distinguish the true target object. This transformer-based approach is a novel step towards understanding and optimizing the process of 3D visual grounding with transformers. Our primary contributions include:

- Introducing 3D-XVIGR, a novel and simple transformer-based architecture that leverages the advantages of Transformers for 3D visual grounding tasks.
- Comprehensive evaluations and comparisons with our baseline, demonstrating the effectiveness and applicability of 3D-XVIGR in addressing visual grounding challenges on ScanRefer.

[1] Pronunciation: *3D-Ex-Vigor*. The "X" signifies the use of a *Trans*former, while "VIGR" is an acronym derived from VIsual GRounding.

[2] https://github.com/wngTn/3d-xvigr

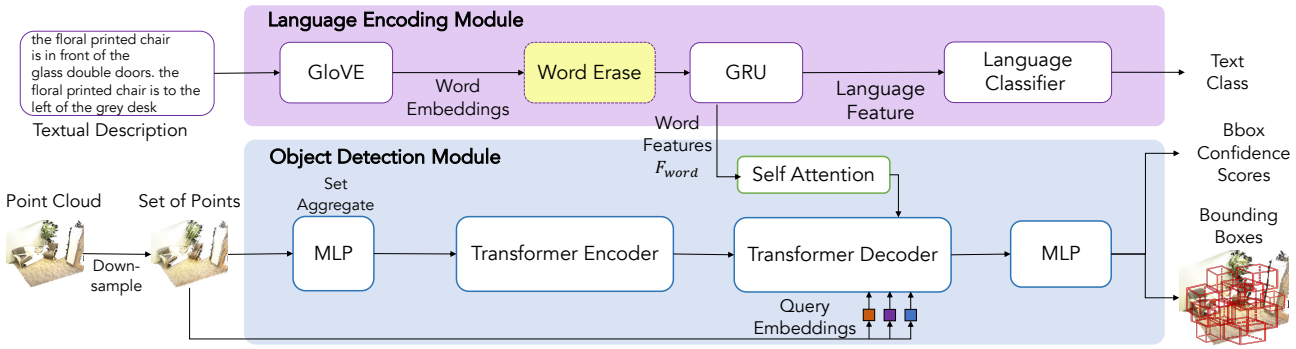

Figure 1. **Overview of the 3D-XVIGR architecture.** Our model consists of a Language Encoding Module to create word embeddings and a transformer-based Object Detection Module, which encodes the input point cloud and combines the encoded point cloud and word features to output the object bounding boxes and their respective confidence score.

2. Related Works

3D Visual Grounding. The domain of 3D point cloud visual grounding has seen increased attention, pioneered by Chen et al. [2], who released the ScanRefer [2] dataset and proposed an end-to-end grounding model, which uses PointNet++ [9] and VoteNet [8] for object detection. ReferIt3D [1] launched two datasets, Nr3D and Sr3D, that resemble ScanRefer but use ground-truth bounding boxes rather than predicted ones. Other proposed methods include the Text-guided Graph Neural Network (TGNN) [5], which segments target objects from 3D scenes based on query sentences, and InstanceRefer [12], which uses a pre-trained panoptic segmentation model. Hierarchical Attention Model (HAM) [3] introduces hierarchical representations for both vision and language inputs. Finally, 3DVG-Transformer [13] leverages transformers to utilize the contextual clues for relation-enhanced proposal generation and cross-modal proposal disambiguations.

Transformers in Computer Vision The Transformer architecture by Vaswani et al. [11] has gained immense prominence not only in Natural Language Processing (NLP) but also in various other domains such as speech recognition, image recognition, and cross-domain applications. Specifically in the field of point clouds, transformers are well suited for operating on 3D points due to their inherent permutation invariance. Notably, the Transformer3D-Det [14] effectively harnessed a transformer to substantially enhance the performance of VoteNet [8] for 3D object detection. In comparison, 3DETR [7] has discarded the VoteNet [8] backbone entirely in favor of a fully transformer-based encoder-decoder architecture, using non-parametric queries to perform 3D object detection.

3. Methods

Figure 1 depicts an overview of 3D-XVIGR. We employ a simple architecture that contains two modules: the language encoding module (LEM) and the object detection module (ODM). The language encoding module obtains the word embeddings of the textual description. It uses a GRU cell (as in ScanRefer [2]) to encode them as a set of word features F_{word} and a global language feature. The language classifier then uses the global language feature to generate the output text class, whereas the word features are used in the object detection module. The object detection module follows an encoder-decoder architecture. It first encodes the input point cloud and outputs proposal bounding boxes and their respective confidence scores given the word features. To achieve this, we adopt the (masked) encoder from 3DETR [7] and design a new decoder that inputs query embeddings, point features, and word features.

3.1. Object Detection Module

The Object Detection Module takes a down-sampled 3D point cloud as input and predicts the positions of objects in the form of 3D bounding boxes and their confidence scores, taking a textual query into account. As the number of points of the down-sampled 3D point cloud is still very large, we use a set aggregation down-sampling operation from Pointnet++ [9] to further down-sample the point cloud to N‘ points and project them to N‘ dimensional features. The resulting subset of N‘ features is passed through an encoder to also obtain a set of N‘ features. A decoder takes these features and the word features as input and predicts multiple bounding boxes using a parallel decoding scheme. Both encoder and decoder are based on standard transformer blocks with ’pre-norm’ [6], performing layer-norm before applying self-attention or non-linear projections.

Encoder. We adopt the masked encoder of 3DETR-m [7], which implements an inductive bias and has proven to improve results significantly. It uses masked self-attention to attend to local rather than global feature aggregation. The encoder input are the point features resulting from the down-sampling and set-aggregation steps. The

Table 1. comparison between 3DVG [13], 3DVG + 3DETR-m, 3D-XVIGR and ScanRefer [2] on the ScanRefer (val) dataset [2] We report the accuracy of correct predicted bounding boxes at IoU 0.25 and 0.5 with only 3D information as input.

Method	Unique		Multiple		Overall	
	Acc.@.25	Acc.@.50	Acc.@.25	Acc.@.50	Acc.@.25	Acc.@.50
3-DVG [13]	0.79	0.58	0.39	0.28	0.47	0.33
3-DVG + 3DETR-m	0.76	0.56	0.37	0.26	0.44	0.32
3D-XVIGR (Ours)	0.74	0.49	0.35	0.23	0.43	0.28
ScanRefer [2]	0.67	0.46	0.32	0.21	0.39	0.26

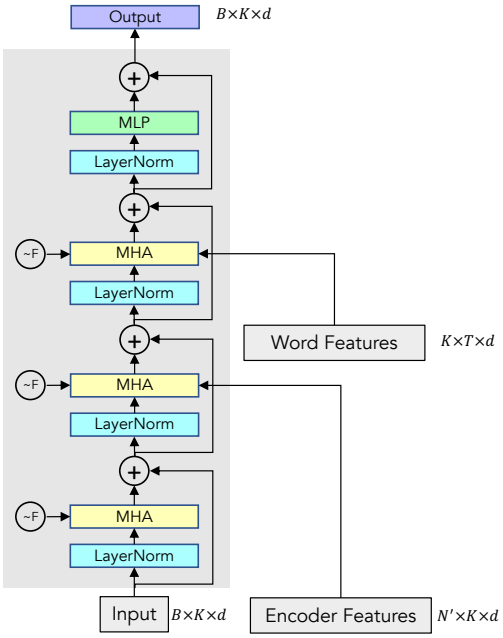

Figure 2. **Our proposed Decoder Language Layer.** The decoder language layer performs cross-attention between the query embeddings, encoder features as well as word features.

transformer encoder then applies multiple layers of self-attention and non-linear projections to encode the point features. We omit any positional encodings in the encoder since the input already contains information about the XYZ coordinates.

Decoder. The decoder fuses a set of B query embeddings, N' point features from the encoder, and K word features together. The output is then fed into an MLP to predict a set of bounding boxes and their respective confidence scores. In order to accommodate the complexity of word features in our model, we designed the *Decoder Language Layer*. This novel layer incorporates the word features of the LEM into the query embeddings and point features, leveraging the attention mechanism to model interdependencies between these sequences. In the decoder, cross-attention layers between the LEM are particularly useful, as this enables query embeddings to additionally attend to the word features and generate more meaningful embeddings.

We design the decoder language layer similar to the decoder layer in 3DETR [7] by adding another block of cross-attention layer before the last MLP block (see Figure 2) and also using Fourier positional encodings [10] of the XYZ coordinates. The overall transformer decoder comprises 8 layers, with 6 decoder layers and 2 decoder language layers.

3.2. Loss Function

We inspire our loss design from 3DVG [13], which contains the reference loss $\mathcal{L}_{ref}$ for visual grounding, the object detection loss $\mathcal{L}_{det}$ for training object detection, and the language to object classification loss $\mathcal{L}_{cls}$ to ensure the word features can be well-matched with the target objects. The object detection loss follows the loss used in 3DETR [7] on the ScanNetv2 [4] dataset: $\mathcal{L}_{det} = 5 * \|\hat{\mathbf{c}} - \mathbf{c}\|_1 + \|\hat{\mathbf{d}} - \mathbf{d}\|_1 - 5 * \mathbf{s}_c^T \log \hat{\mathbf{s}}_c$, where $\mathbf{c}, \mathbf{d}, \mathbf{s}$ are the center, dimensions, and semantic terms, respectively. We omit the angular term from the original loss since we output axis-aligned bounding boxes. We then set the final loss as a combination of these terms to $\mathcal{L} = 0.3 * \mathcal{L}_{ref} + \mathcal{L}_{det} + 0.1 * \mathcal{L}_{cls}$.

3.3. Experiments

Dataset. Our model is evaluated on the ScanRefer dataset [2], comprising 3D indoor scene scans with annotated bounding boxes for 11,000+ objects over 800 scenes; the scenes originate from the Scannet [4] dataset. These annotations next to the bounding box include the object name and multiple descriptions for that specific object.

Metric. To evaluate the performance of 3D-XVIGR, we gauge the accuracy over different intersection over union (IoU) thresholds. A prediction only counts as true positive when its IoU with the ground truth is above a certain threshold. Similarly to 3DVG [13], we set the threshold values to 0.25 and 0.5. As per ScanRefer [2] and 3DVG [13], we classify scenes as "unique" if they contain a single object of a specific class and "multiple" otherwise. We report metrics for each category, including an "overall" measure encompassing all scenes.

Models. Our evaluation employs four models: 3DVG [13], ScanRefer [2], 3DVG + 3DETR-m, and 3D-XVIGR (Ours). We train 3DVG [13] using the original code-base from

GitHub[3] following the authors. For ScanRefer [2], we report the metrics taken directly from the paper and do not train the model ourselves. We also use 3DVG [13] and replace its original Transformer3D-Det [14] object detection backbone with 3DETR-m [7] - this variant is called 3DVG + 3DETR-m. 3D-XVIGR constitutes our proposed model. Both 3DVG + 3DETR-m and 3D-XVIGR were initialized with the pre-trained weights of 3DETR after 1080 epochs, as provided by the authors. Aside from the epoch count, both models were trained with the same hyper-parameters as 3DETR. All models were trained end-to-end on ScanRefer (train) [2] for 200 epochs.

Results. Table 1 illustrates our ScanRefer (val) [2] results against the baselines 3DVG [13], 3DVG + 3DETR-m, and ScanRefer [2]. Although our model does not exceed baseline performance, it presents comparable results, with notable deficiencies at 0.5 IoU—decreasing from 0.58 to 0.49 in 'unique' and 0.28 to 0.23 in 'multiple.' This underperformance could be attributed to 3D-XVIGR's limited training (200 epochs due to hardware constraints), implying incomplete convergence. 3DVG and 3DVG + 3DETR-m's superior performance could be due to the feature selection and proposal copy & paste [13] data augmentation in the cross-modal fusion module, which are absent in 3D-XVIGR. While 3DVG + 3DETR-m underperforms 3DVG, it is due to Transformer3D-Det [14], the original 3DVG object detection backbone, outperforming 3DETR-m [7] on the ScanNetv2 [4] dataset. Compared to ScanRefer [2], our model exhibits superior results, most likely due to ScanRefer being released in 2020, thus using less sophisticated modules.

Figure 3 shows qualitative results of our method on the ScanRefer dataset. These results show that 3D-XVIGR can correctly put the textual description and 3D point cloud into relation. Moreover, 3D-XVIGR can perform better than 3DVG, as shown in the third column, where it could correctly identify the referred kitchen cabinet. However, mirroring Table 1, 3D-XVIGR struggles with detecting high-IoU bounding boxes.

3.4. Ablation Experiments

Table 2. Results of 3D-XVIGR with different numbers of Decoder Layers (Dec. Lay.) and Decoder Language Layers (Dec. Lan. Lay.).

# Dec. Lay.	# Dec. Lan. Lay.	Overall	
		Acc.@.25	Acc.@.50
6	2	0.43	0.28
8	2	0.42	0.26
8	4	0.42	0.24

[3] https://github.com/zlccccc/3DVG-Transformer

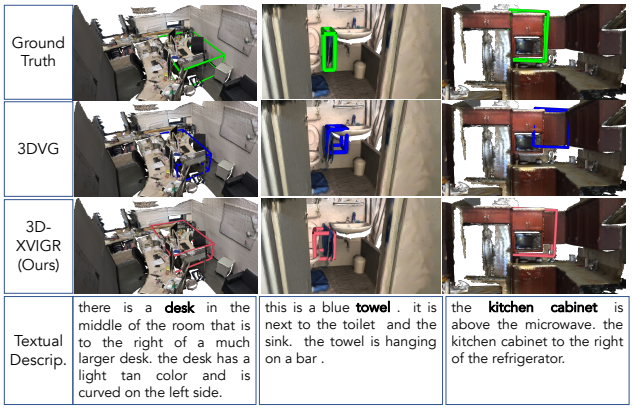

Figure 3. **Qualitative Results on ScanRefer.** We show comparisons with our baseline 3DVG [13] on the ScanRefer [2] (val) dataset.

Number of Layers. We investigate various decoder configurations in Table 2. The results show a slight performance dip with increased decoder layers, possibly due to weight initialization from 3DETR pre-trained weights. 3DETR comprises 8 total decoder layers, so models with this layer count are better initialized. Considering the model's limited training duration (200 epochs), the extra layers might not be fully trained.

4. Conclusion

This work introduced 3D-XVIGR, a transformer-based model for the 3D visual grounding tasks. We proposed a new decoder language layer, which fosters a robust alignment between textual descriptions and 3D point clouds. This decoder language layer allows 3D-XVIGR to leverage a streamlined architecture for aligning visual and natural language processing tasks. Notably, even without relying on highly engineered solutions or hand-crafted data augmentation techniques, 3D-XVIGR demonstrates comparable performance with baseline methods on the ScanRefer dataset. The results also underscore the potential of transformer-based solutions for 3D visual grounding. However, our model's performance could be improved with extended training and more sophisticated techniques, which suggests areas for future enhancements. Nevertheless, 3D-XVIGR represents a significant step in 3D visual grounding, offering valuable insights and a foundation for further research in this area.

Acknowledgement. Sincere gratitude is extended to Yushan Zheng for her continuous support and assistance throughout this work, particularly during challenging phases. Her enduring encouragement and patience have been paramount to this project's outcome.

References

[1] Panos Achlioptas, Ahmed Abdelreheem, Fei Xia, Mohamed Elhoseiny, and Leonidas J. Guibas. ReferIt3D: Neural listeners for fine-grained 3d object identification in real-world scenes. In *16th European Conference on Computer Vision (ECCV)*, 2020. 2

[2] Dave Zhenyu Chen, Angel X. Chang, and Matthias Nießner. ScanRefer: 3D Object Localization in RGB-D Scans Using Natural Language. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, volume 12365, pages 202–221. Springer International Publishing, Cham, 2020. Series Title: Lecture Notes in Computer Science. 2, 3, 4

[3] Jiaming Chen, Weixin Luo, Xiaolin Wei, Lin Ma, and Wei Zhang. HAM: Hierarchical Attention Model with High Performance for 3D Visual Grounding, Oct. 2022. arXiv:2210.12513 [cs]. 2

[4] 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. 3, 4

[5] Pin-Hao Huang, Han-Hung Lee, Hwann-Tzong Chen, and Tyng-Luh Liu. Text-guided graph neural networks for referring 3d instance segmentation. In *AAAI Conference on Artificial Intelligence*, 2021. 2

[6] Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander Rush. OpenNMT: Open-source toolkit for neural machine translation. In *Proceedings of ACL 2017, System Demonstrations*, pages 67–72, Vancouver, Canada, July 2017. Association for Computational Linguistics. 2

[7] Ishan Misra, Rohit Girdhar, and Armand Joulin. An End-to-End Transformer Model for 3D Object Detection, Sept. 2021. arXiv:2109.08141 [cs]. 1, 2, 3, 4

[8] Charles R. Qi, Or Litany, Kaiming He, and Leonidas J. Guibas. Deep Hough Voting for 3D Object Detection in Point Clouds, Aug. 2019. arXiv:1904.09664 [cs]. 2

[9] Charles R Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *arXiv preprint arXiv:1706.02413*, 2017. 2

[10] Matthew Tancik, Pratul P. Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan T. Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. *NeurIPS*, 2020. 3

[11] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6000–6010, Red Hook, NY, USA, 2017. Curran Associates Inc. 2

[12] Zhihao Yuan, Xu Yan, Yinghong Liao, Ruimao Zhang, 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. 2

[13] 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, Montreal, QC, Canada, Oct. 2021. IEEE. 1, 2, 3, 4

[14] Lichen Zhao, Jinyang Guo, Dong Xu, and Lu Sheng. Transformer3d-det: Improving 3d object detection by vote refinement. *IEEE Transactions on Circuits and Systems for Video Technology*, 31(12):4735–4746, 2021. 2, 4