Joint Architecture for Visual Grounding and Dense Captioning

Nikola Nincic, Zhihao Liang

Abstract

3D visual grounding and dense captioning are two separate 3D vision-language tasks that share a common understanding of object features and spatial relationships. In this work, we introduce our simple fully unified transformer-based architecture that jointly solves these two tasks. It consists of both shared task-agnostic transformer encoder layers and lightweight task-specific MLP decoder layers. Experiments on ScanNet and ScanRefer datasets demonstrate that our method achieves the state-of-the-art results for visual grounding on uniquely described objects.

1. Introduction

Existing works on 3D visual grounding (VG) [4, 17] and dense captioning (DC) [18] have achieved impressive results by separately solving the two tasks. The 3D VG task takes a point cloud and an object description as input, its goal is to output a bounding box of the described object. DC, on the other hand, has only a point cloud as input and generates bounding boxes, as well as descriptions for all objects in the scene. Both tasks share a common understanding of object features and spatial relationships. Although previous attempts [3, 6] to combine them have shown encouraging results by using task-specific neural modules in a unified framework to exploit partly shared spatial object information, it is desirable to develop a method to learn a shared representation for the tasks. Using this idea, Chen et al. [5] implemented an architecture that computes a joint representation of 3D data and text, based on which the predicted bounding boxes and captions are generated. Using the method described in their paper as inspiration, we created our own architecture to jointly tackle DC and VG. To summarize, our achievements are:

- Implementing a functional fully unified transformer-based model that solves the combined tasks of visual grounding and dense captioning.

- Train the model to achieve visual grounding performance that is comparable to state-of-the-art models, such as 3DJCG [3].

- Generate visualizations that demonstrate the quality of our predictions.

2. Related work

2.1. 3D visual grounding

Over the past few years, several methods to solve the task of 3D visual grounding have been published [4, 9, 13, 15]. Some of the approaches use object detection modules, such as PointGroup [10] or VoteNet [12], while others just take the ground truth bounding boxes as input. Finally, these bounding boxes are used to predict the bounding box of the object that corresponds to the textual description.

2.2. 3D dense captioning

The related task of 3D dense captioning has also been explored by recent works [16, 18]. The most challenging part of solving this task is to include the spatial relationships between objects in the generated caption. Correctly using keywords, such as ”under” or ”behind”, requires a deeper understanding of the full scene, rather than just the features of the captioned object. For this reason, many attempts to densely caption objects in 3D scenes fail to include words that describe inter-relationships between objects, leading to unnatural and imprecise captions.

2.3. Joint architecture for 3D vision-language

Since both VG and DC share many sub-tasks, such as identifying objects, and learning their features + relationships, it is natural, to solve them in a unified manner. One very recent work that makes use of these similarities is UniT3D [5], a unified transformer-based architecture that uses PointGroup [10] to obtain object proposals from the point cloud, and a BERT-module [8] to generate a text embedding from the object description. The output from the two modules is then concatenated and fed into the transformer that computes the fused representation. To decode that fused output for each task, the authors simply apply a lightweight grounding and captioning head. With this novel approach, UniT3D managed to outperform 3DJCG [3] and D3Net [19] on the ScanRefer validation set [4], which are taking up high spots on the official ScanRefer benchmark challenge [1]. For this reason, we chose to explore this approach further and experiment with possible modifications, like using an object detection module with better performance and extracting proposal features from point features.

3. Method

3.1. A unified transformer-based model

Our architecture components as shown in Figure 1 include SoftGroup [14] for object detection and a pre-trained BERT module for generating text embedding. While BERT delivers a text embedding with 768-dimensional features per word, SoftGroup gives only 32-dimensional features per point and per instance proposal. In order to obtain instance proposals with a similar size of feature dimension as the word embeddings from BERT, we use MLPs to extract instance features from clustered point features given by SoftGroup and match the dimensions of the two. The resulting embeddings will be referred to as box tokens and text tokens, where a box token represents one object proposal and a text token represents one word.

Following Chen et al.'s idea [5], we do two forward passes before computing the gradients for each update step in the training:

Visual grounding pass: The box tokens are averaged and added to each of the text tokens as the global visual cue. Then the concatenated box and text tokens are fed to the transformer encoder that generates fused tokens. The first *number_of_proposals* fused tokens are then taken as input for the lightweight MLP to compute the confidence scores for each proposal. Since the SoftGroup-module already captures the object proposals in a scene, the grounding head mainly focuses on matching the given language descriptions to the detected object proposals. Thus, we use a cross-entropy loss L_{match} to compute the distance between the proposal with the highest confidence and the most accurate (= the proposal with the highest IoU with the queried object) object proposal. Also, we followed the idea to predict the object semantic class from the text query by applying an MLP with the [CLS] token as input. We supervise this object classification with another cross-entropy loss L_{cls}.

Dense captioning pass: We use the teacher-forcing scheme to predict the next words for DC. In detail, we pad the previous text tokens with [CLS] at the beginning as input text tokens and the future text tokens with [SEP] at the end as target. By applying a triangular mask for the text tokens in the transformer encoder, each text token is only allowed to attend to previous tokens in the input. For the DC pass, the target box token is added to all text tokens as a captioning cue. Again, the box and text tokens are concatenated and fed into the transformer encoder. This time, we use the lightweight captioning head to decode the last *number_of_words* fused tokens to obtain the predicted next words. Here, we apply a word-level cross-entropy loss $L_{caption}$ on each predicted next word against the target word.

The final loss is a linear combination of these loss terms, i.e., $L_{total} = L_{match} + L_{cls} + L_{caption}$.

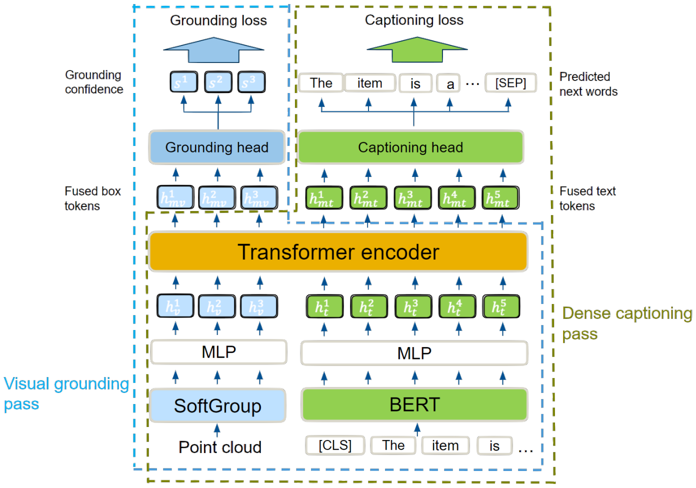


Figure 1. **Model architecture**

3.2. Data augmentation

As we have a rather low number of captioned scenes at our disposal, data augmentation is needed to avoid fast overfitting. However, the SoftGroup inference takes long. Therefore, we moved the computation in SoftGroup and BERT for every scene and description into a pre-processing step. This enabled us to train our model in a timely efficient fashion, but at the same time this removes the option to augment the scene data, as SoftGroup outputs would need to be re-computed. The solution we found to this problem is to pre-compute multiple SoftGroup outputs using augmented data per scene. This would make it harder for the model to overfit while keeping the advantages of the pre-processing step.

3.3. Data loading

In each step, a description and a random pre-computed augmentation variant of the scene it belongs to are loaded into the RAM. Since the pre-computed data is quite large and we only had 48GB of RAM, we had to resort to loading from disc at every step of our training. However, as we found out by doing time measurements, this only lead to about 10% longer training time per epoch.

3.4. Inference

We use the grounding head and captioning head for decoding visual grounding and dense captioning outputs from the task-agnostic transformer encoder module. For VG inference, we use the minimum and maximum coordinates in the predicted instance masks to construct the object bounding boxes, where we keep only the bounding box with the highest IoU for the GT box. For DC inference, we generate words from the [CLS] token. In our inference method, we keep track of the best sequences, until the sequence with the highest score ends with [SEP] token or reaches the predefined maximum length.

4. Experiments

4.1. Implementation details

In order to reach our goal of having a single network that supports VG and DC, we implement a unified transformer-based model using PyTorch [11]. Chen et al.'s UniT3D paper [5] was used as a guideline and the minsu3d repository [2] as a framework for this project. The pre-trained SoftGroup module and the method for loading ScanNet data were provided by minsu3d.

We use 20 descriptions per scene for training our network. Having a fixed number of descriptions effectively counters the strong imbalance in the distribution of the number of descriptions per scene that is present in the vanilla ScanRefer dataset. With a learning rate of 1e-4 and frozen BERT + SoftGroup, we trained our model for 85 epochs. In order to achieve more robust performance during inference, we use truncated text input of random length and replace the last text token with the predicted text token from epoch 45 during the training process. In case the same description was used for multiple objects in one scene, the number of targets for visual grounding was increased, guaranteeing uniqueness in the mapping between input and target.

Our full architecture contains 52M trainable parameters. All our experiments are conducted on an RTX 3060 GPU and 48GB of RAM.

4.2. Dataset

For all of our experiments we used the ScanRefer [4] dataset, which contains about 51k object descriptions for 11k objects in 800 scenes. These descriptions describe objects based on their appearance and position relative to other objects. Another notable point is that one description might match multiple objects, which makes an accurate prediction for all samples highly unlikely. Further, all of the scenes are part of the ScanNet [7] dataset and the training/validation split is chosen to be identical to the official ScanRefer split.

4.3. Evaluation metrics

For visual grounding, we chose Acc@0.25IoU and Acc@0.5IoU. The computation works as follows: We count the number of times our model predicted a bounding box that has at least 25%/50% IoU with the ground truth and divide it by the total number of predictions. We further divide the scores into *Unique, Multiple and Overall*. *Unique* means only one object of the same class exists in the scene, while *Multiple* means there are more. *Overall* represents the overall accuracy.

To jointly measure the quality of the generated descriptions and the detected bounding boxes, we evaluate them by using standard image captioning metrics such as CIDEr and BLEU-4 under different Intersection- over-Union (IoU) scores between predicted bounding boxes and the matched ground truth bounding boxes.

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

To compare the performance of our architecture to those of state-of-the-art (SOTA) methods, we conducted a quantitative analysis, the results of which can be seen in table 1 and 2.

3D visual grounding: We can see that our model outperforms both 3DJCG and ScanRefer in the category *Unique*, but struggles when the description is not referencing the only object of a class. This pattern can also be observed for UniT3D, although it's scores are generally higher.

3D dense captioning: When looking at the performance comparison in table 2, it is quickly visible that our captioning scores are significantly lower than the ones of the other architectures. We believe that the reason for this lies in insufficient training for the dense captioning inference. Since we used the teacher-forcing scheme during the joint training, our model relies on the previous ground truth words as input in order to generate good predictions. Therefore, training the model in an inference-like scheme, meaning without knowledge of the ground truth description, is necessary to enable it to generate good captions during inference.

	Val Acc@0.25IoU			Val Acc@0.5IoU		
	Unique	*Multiple*	*Overall*	*Unique*	*Multiple*	*Overall*
ScanRefer [4]	76.33	32.73	41.19	53.51	21.11	27.40
3DJCG [3]	78.75	**40.13**	**47.62**	61.30	30.08	36.14
UniT3D(w/ pre-training) [5]	**82.75**	36.36	45.27	**73.14**	**31.05**	**39.14**
Ours	80.10	29.27	39.13	69.05	23.08	32.00

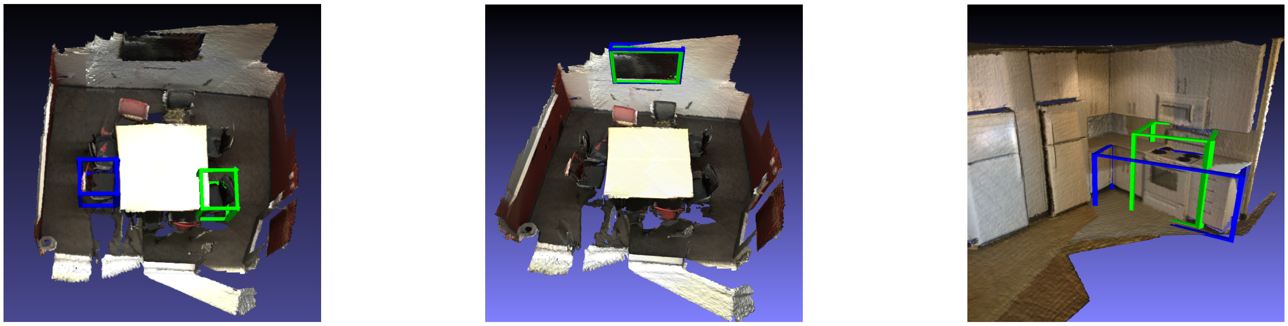
Table 1. Visual Grounding results comparison

	Dense Captioning Recall-Scores @0.5IoU				
	CiDEr	*BLEU-4*	*ROUGE-L*	*METEOR*	*mAP@0.5IoU*
Scan2Cap [18]	35.20	22.36	43.57	21.44	32.09
3DJCG [3]	**47.68**	**31.53**	**51.08**	**24.28**	39.75
UniT3D(w/ pre-training) [5]	46.69	27.22	45.98	21.91	54.03
Ours (before extra-training)	0.60	3.41	25.76	11.82	**65.1***
Ours	13.05	8.06	33.33	16.45	**65.1***

Table 2. Dense captioning recall results comparison

4.5. Visualizations

How the captions and predicted bounding boxes compare against the ground truth in praxis, can be observed in figure 2. Taking a look at the images, we can quickly discover that our model is accurate at finding objects that are unique in the scene, such as the TV in figure 2b and the stove in figure 2c. Additionally, when the description is not unique, it still identifies an object of the same class correctly (see



(a) **Ground-truth**: "There is a gray chair. It is at a table with its back facing a white wall" **Prediction**: "Placed there is a square chair . it is at the long table with the brown chair is at the"

(b) **Ground-truth**: "This is a black tv. It is mounted on a white wall" **Prediction**: "Its there is a rectangular. it picture on the wall"

(c) **Ground-truth**: "There is a rectangular white stove. It is between two kitchen cabinets" **Prediction**: ", this it stove , . it this kitchen is stove that this is it counter that is . it"

Figure 2. Visualizations of predicted bounding boxes and their corresponding descriptions.

fig.2a. The captions on the other hand often include the correct words (see fig.2c) but have seemingly no structure.

4.6. Ablation and analysis

Does inference-like training help the dense captioning? Before training our model in an inference-like scheme, we tested its performance on ScanRefer's validation set and saw significantly worse performance in all captioning metrics. Therefore, we have strong reason to believe that additionally training the model for DC inference is increasing the captioning scores. The difference in performance is visualized in table 2.

5. Conclusion

In conclusion, our work shows the potential of a joint architecture for 3D dense captioning and 3D visual grounding. Without many resources at our disposal, we managed to achieve SOTA-comparable results in VG and some minor achievements in DC. Therefore, it is very likely that with more data, resources, and training, our architecture could achieve competitive scores, especially in VG.

Limitations: While our model achieves reasonable results in visual grounding, it is still lacking in dense captioning. It has to be trained more to be capable of generating complete sentences without the teacher-forcing scheme. Further, additional scene data, such as normals or multiview could be used for training. We believe this could further improve our performance on both tasks. Finally, more hyperparameter tuning could be done to maximize the architecture's potential.

References

[1] Scanrefer benchmark. https://kaldir.vc.in.tum.de/scanrefer_benchmark/. Accessed: 2023-07-04. 1

[2] Scanrefer benchmark. https://github.com/3dlg-hcvc/minsu3d. Accessed: 2023-11-04. 3

[3] Daigang Cai, Lichen Zhao, Jing Zhang, Lu Sheng, and Dong Xu. 3djcg: A unified framework for joint dense captioning and visual grounding on 3d point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16464–16473, 2022. 1, 3

[4] Dave Zhenyu Chen, Angel X Chang, and Matthias Nießner. Scanrefer: 3d object localization in rgb-d scans using natural language. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XX*, pages 202–221. Springer, 2020. 1, 3

[5] Dave Zhenyu Chen, Ronghang Hu, Xinlei Chen, Matthias Nießner, and Angel X Chang. Unit3d: A unified transformer for 3d dense captioning and visual grounding. *arXiv preprint arXiv:2212.00836*, 2022. 1, 2, 3

[6] Dave Zhenyu Chen, Qirui Wu, Matthias Nießner, and Angel X Chang. D 3 net: A unified speaker-listener architecture for 3d dense captioning and visual grounding. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXII*, pages 487–505. Springer, 2022. 1

[7] 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

[8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 1

[9] Shijia Huang, Yilun Chen, Jiaya Jia, and Liwei Wang. Multi-view transformer for 3d visual grounding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15524–15533, 2022. 1

[10] Li Jiang, Hengshuang Zhao, Shaoshuai Shi, Shu Liu, Chi-Wing Fu, and Jiaya Jia. Pointgroup: Dual-set point grouping for 3d instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and Pattern recognition*, pages 4867–4876, 2020. 1

[11] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019. 3

[12] Charles R Qi, Or Litany, Kaiming He, and Leonidas J Guibas. Deep hough voting for 3d object detection in point clouds. In *proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9277–9286, 2019. 1

[13] Junha Roh, Karthik Desingh, Ali Farhadi, and Dieter Fox. Languagerefer: Spatial-language model for 3d visual grounding. In *Conference on Robot Learning*, pages 1046–1056. PMLR, 2022. 1

[14] Thang Vu, Kookhoi Kim, Tung M Luu, Thanh Nguyen, and Chang D Yoo. Softgroup for 3d instance segmentation on point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2708–2717, 2022. 2

[15] Zhengyuan Yang, Songyang Zhang, Liwei Wang, and Jiebo Luo. Sat: 2d semantics assisted training for 3d visual grounding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1856–1866, 2021. 1

[16] Zhihao Yuan, Xu Yan, Yinghong Liao, Yao Guo, Guanbin Li, Shuguang Cui, and Zhen Li. X-trans2cap: Cross-modal knowledge transfer using transformer for 3d dense captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8563–8573, 2022. 1

[17] 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*, pages 2928–2937, 2021. 1

[18] Dave Zhenyu Chen, Ali Gholami, Matthias Nießner, and Angel X Chang. Scan2cap: Context-aware dense captioning in rgb-d scans. *arXiv e-prints*, pages arXiv–2012, 2020. 1, 3

[19] Dave Zhenyu Chen, Qirui Wu, Matthias Nießner, and Angel X Chang. D3net: A unified speaker-listener architecture for 3d dense captioning and visual grounding. *arXiv e-prints*, pages arXiv–2112, 2021. 1