Bottom Up Top Down Detection Transformers for Language Grounding in Images and Point Clouds

Ayush Jain[†1], Nikolaos Gkanatsios[†1], Ishita Mediratta[§2], and Katerina Fragkiadaki[1]

[1]Carnegie Mellon University, [2]Meta AI

1 Supplementary file

1.1 Overview

In Section 1.2, we provide implementation details for BUTD-DETR on both the 3D and the 2D domain. In Section 1.3, we provide a detailed analysis of our results on SR3D, NR3D [2] and ScanRefer benchmarks [4]. In Section 1.4 we ablate the choice of the detection backbone and experiment with unfreezing it during the referential grounding training stage. In Section 1.5, we show the effect of corrupting the detector's proposals at training time. In Section 1.6, we discuss training with detection prompts that contain negative labels. We evaluate our model as a language-modulated object detector in Section 1.7. In Section 1.8, we show more qualitative results on both 3D point clouds and 2D images, including failure cases.

1.2 Implementation details

We report here architecture choices as well as training hyperparameters. We implement BUTD-DETR in PyTorch. For the 3D version, the point cloud is encoded with PointNet++ [18] using the same hyperparameters as in [14], pre-trained on ScanNet [5]. We use the last layer's features, resulting in 1024 visual tokens. The detected boxes are encoded using their spatial and categorical features. Specifically, we encode each box's coordinates with an MLP, then we concatenate this vector with projected RoBERTa [13] embeddings and feed to another MLP to obtain the box embeddings. For the cross-modality encoder, we use $N_E = 3$ layers. All attention layers are implemented using standard key-value attention [20,15]. In the decoder, the queries are formed from the 256 most confident visual tokens. To compute this confidence score, each visual token is fed to an MLP to give a scalar value. We supervise these values using Focal Loss [12]. Specifically, since each visual token corresponds to a point with known coordinates, we associate visual tokens to ground-truth object centers and keep the

[†]Equal contribution, order decided by `np.random.rand`

[§]Work done during an internship at CMU

Table 1: **Performance analysis on language grounding on SR3D.** We evaluate top-1 accuracy using ground-truth (GT) boxes, under the different setups introduced in [2]. See the main text for an explanation of each setup.

Method	Easy	Hard	View-Dep	View-Indep	Overall (GT)
ReferIt3DNet [2]	44.7	31.5	39.2	40.8	39.8
TGNN [10]	48.5	36.9	45.8	45.0	45.0
3DRefTransformer [1]	50.7	38.3	44.3	47.1	47.0
InstanceRefer [23]	51.1	40.5	45.4	48.1	48.0
LanguageRefer [19]	58.9	49.3	49.2	56.3	56.0
3DVG-Transformer [24]	54.2	44.9	44.6	51.7	51.4
TransRefer3D [8]	60.5	50.2	49.9	57.7	57.4
SAT 2D [21]	61.2	50.0	49.2	58.3	57.9
BUTD-DETR (ours)	**68.6**	**63.2**	**53.0**	**67.6**	**67.0**

4 closest points to each center. We consider these matched points as positives, i.e. here points with high ground-truth objectness. The same scoring method is employed in [14]. We use $N_D = 6$ decoder layers. Similar to encoder, all attention layers are implemented using standard self-/cross-attention.

For the 2D version, the image is encoded using ResNet-101 [9] pretrained on ImageNet [6]. We use multi-scale features as in [25]. The feature maps of the different scales are flattened and concatenated in the spatial dimension, leading to 17821 visual tokens. The feature dimension of each token is 256. To obtain the box proposals, we use the detector of [3] trained on 1601 classes of Visual Genome [11]. The detected boxes are encoded using their spatial and categorical features. Specifically, we compute the 2D Fourier features of each box and feed them to an MLP, then we concatenate this vector with projected RoBERTa [13] embeddings and feed to another MLP to obtain the box embeddings. To form queries, we rank visual tokens based on their confidence score and keep the 300 most confidence ones. This confidence layer is supervised using Focal Loss [12]: we assign a positive objectness scores to every point that lies inside a ground-truth answer box. We set $N_E = 6$ and $N_D = 6$. All attention layers to the visual stream are implemented with deformable attention [25], attention to either the language stream or detected boxes is the standard attention of [20,15]. We do not use deformable attention in the 3D domain since computing it requires pooling features and doing bilinear interpolation from neighbouring pixels. In 2D, finding neighbouring pixels can be trivially done by simply looking up neighbouring indices due to its continuous grid structure. However, in discontinous domains like 3D, we would need to compute all pairs of distances between the points in a given pointcloud and rank them to obtain nearest neighbours. This is computationally expensive. Moreover, since pointclouds have irregular density, using a fixed number of neighbours is sub-optimal. These issues can be resolved by using specialised data-structures like KD-Trees and by using adaptive neighbourhood sampling, however they are beyond the scope of this work.

Table 2: **Performance analysis on language grounding on NR3D.** We evaluate top-1 accuracy using ground-truth (GT) boxes, under the different setups introduced in [2]. See the main text for an explanation of each setup.

Method	Easy	Hard	View-Dep	View-Indep	Overall (GT)
ReferIt3DNet [2]	43.6	27.9	32.5	37.1	35.6
TGNN [10]	44.2	30.6	35.8	38.0	37.3
3DRefTransformer [1]	46.4	32.0	34.7	41.2	39.0
InstanceRefer [23]	46.0	31.8	34.5	41.9	38.8
FFL-3DOG [7]	48.2	35.0	37.1	44.7	41.7
LanguageRefer [19]	51.0	36.6	41.7	45.0	43.9
3DVG-Transformer [24]	48.5	34.8	34.8	43.7	40.8
TransRefer3D [8]	48.5	36.0	36.5	44.9	42.1
SAT 2D [21]	56.3	42.4	**46.9**	50.4	49.2
BUTD-DETR (ours)	**60.7**	**48.4**	46.0	**58.0**	**54.6**

For the 3D model, we freeze the text encoder and use a learning rate of $1e{-}3$ for the visual encoder and $1e{-}4$ for all other layers. We are able to fit a batch size of 6 on a single GPU of 12GB and 24 on an NVIDIA A100. Under these conditions, each epoch takes around 50 minutes on an A100. For the 2D model, we use a learning rate of $1e{-}6$ for Resnet101 visual encoder, $5e{-}6$ for RoBERTa text encoder and $1e{-}5$ for rest of the layers. We pre-train on 64 NVIDIA V100 GPUs with a batch size of 1, and finetune on RefCOCO/RefCOCO+ with a batch size of 2 on 16 V100s. The total training time is included in the respective tables. We release pre-trained checkpoints for both 3D and 2D models.

Table 3: **Performance analysis on language grounding on ScanRefer.** We evaluate top-1 accuracy using detected boxes, under the different setups introduced in [4]. See the main text for an explanation of each setup.

Method	Unique@0.25	Unique@0.5	Multi@0.25	Multi@0.5	Overall@0.25	Overall@0.5
ReferIt3DNet [2]	53.8	37.5	21.0	12.8	26.4	16.9
ScanRefer [4]	63.0	40.0	28.9	18.2	35.5	22.4
TGNN [10]	68.6	56.8	29.8	23.2	37.4	29.7
InstanceRefer [23]	77.5	66.8	31.3	24.8	40.2	32.9
FFL-3DOG [7]	78.8	**67.9**	35.2	25.7	41.3	34.0
3DVG-Transformer [24]	77.2	58.5	38.4	28.7	45.9	34.5
SAT 2D [21]	-	-	-	-	44.5	30.1
BUTD-DETR (ours)	**84.2**	66.3	**46.6**	**35.1**	**52.2**	**39.8**

1.3 Detailed results on SR3D/NR3D and ScanRefer

We include results on SR3D/NR3D [2] and ScanRefer [4] under the different evaluation protocols specified in the original papers. Similar to prior works, we report results using overall accuracy metric. In det setup, we threshold over the

IoU between the box regressed by BUTD-DETR and the ground truth box. In `GT` setup, we select the ground truth box that the has highest IoU with the most confident box regressed by BUTD-DETR and check if it matches with the target box. Besides overall accuracy, we additionally report accuracy on the following contexts for SR3D/NR3D:

- Easy: there is only one "distractor", i.e. object belonging to the same class as the target instance
- Hard: there are two or more distractors
- View-dependent: cases for which rotating the scene around the z axis would lead to a different answer, e.g. "tv left of sofa"
- View-independent: rotation does not affect the answer, e.g. "chair closest to table"

We evaluate on the following contexts for ScanRefer:

- Unique: there is no "distractor", i.e. object belonging to the same class as the target instance
- Multi: there is at least one distractor

We compare BUTD-DETR against prior approaches in Table 1 for SR3D, Table 2 for NR3D and Table 3 for ScanRefer. For SR3D and NR3D, all models are trained and tested with access to ground-truth object proposals, as in [2]. For ScanRefer, all models are trained and tested with detected objects, so we report accuracy under the 0.25 and 0.5 IoU thresholds. We vastly outperform all competitors under all setups on SR3D. On NR3D, we show clear gains on all protocols except for view-dependent. Performance on this setup could be improved by incorporating a view prediction network, but we aimed to have a model that works for both 3D and 2D with as least domain-specific design choices as possible. On ScanRefer, we clearly outperform all previous approaches under all setups except for Unique@0.5, where we perform on par with the best-performing competitor.

1.4 Effect of detection backbone

To examine the importance of the detection backbone, since previous work use VoteNet [17] as their detector, we evaluate our model using VoteNet boxes on ScanRefer and get 50.0% Acc@0.25 and 37.5% Acc@0.5 (in comparison to 50.9% and 38.8% with Group-Free boxes), which still outperforms all competitors. On SR3D and NR3D all previous works use GT boxes; hence we re-run all baselines of Table 1 with the same detector as our model.

Additionally, we try to unfreeze the object detector backbone during training with language. Inspired by [22], we added a box regression layer in our baseline "w/o visual tokens" of Table 2. This achieves 46.4% on SR3D, which is indeed better than our previous baseline by 4.5%. However, it still underperforms our proposed model by 4.7%. This result indicates that box-bottlenecked baselines still underperform, even when the object detector is finetuned.

Table 4: **Effect of detection augmentation on SR3D.**

Method	Overall (Det)
BUTD-DETR w/o box stream	51.0
BUTD-DETR w/o detection augmentation	51.1
BUTD-DETR	52.1

1.5 Effect of detection augmentation

As we mention in the main paper, the 3D detector is trained on ScanNet and thus the proposals are of much better quality at train time and worse at test time. To mitigate overfitting, we randomly replace 30% of the detected boxes at training time with random ones. Quantitatively, this gives a boost of 1% absolute, as seen in Table 4. Note that this augmentation can only be applied when the box stream is employed.

1.6 Negative training with detection prompts

We devise object detection as language grounding of an utterance formed by concatenating a sequence of category labels, e.g. "Chair. Dining table. Bed. Plant. Sofa.". The task is again to i) detect the mentioned objects in the scene, i.e. return the bounding boxes of their instances, and ii) associate each localized box to a span, i.e. an object category in the utterance.

To form these detection prompts, one solution could be to concatenate all object classes into a long utterance. However, this can be impractical if the domain-vocabulary is "open", or, in practice, very large (485 classes in ScanNet, 1600 in Visual Genome and so on). Instead, assuming that we have object annotations, we sample out of the positive labels that are annotated for a scene and a number of negative ones, corresponding to class names that are not associated with any instances in the scene. Having negative classes in the detection prompts helps the precision of the model, as it learns not to fire for every noun phrase that appears in an utterance. More specifically, the contrastive losses described in the main paper push the negative class' text representation away from the query representation of existing objects.

MDETR also considers an object detection evaluation. However, there are two noticeable differences. First, they use only single-category utterances, e.g. "Dog.". This category can be either positive (appears in the annotations) or negative (does not appear in the annotations), according to a sampling ratio. Opposite to that, our detection prompts are longer, consisting of multiple object categories, both positive and negative. Second, MDETR employs these sentences after pre-training, to train and evaluate their model as an object detector. Instead, we mix detection prompts through the training, leading to considerable quantitative gains in both 3D and 2D.

Lastly, although the ratio r of positive to negative classes that appear in a detection phrase is a hyperparameter, we report results only for $r = 1$ and

Table 5: **Object detection performance on ScanNet.** We evaluate BUTD-DETR trained with detection prompts on different datasets. Training on referential data and detection prompts offers a consistent gain on detection mAP.

Method	mAP@0.25
DETR+KPS+iter [14]	59.9
3DETR with PointNet++ [16]	61.7
BUTD-DETR trained on ScanNet	59.3
BUTD-DETR trained on ScanNet with softmax	61.0
BUTD-DETR trained on SR3D	61.1
BUTD-DETR trained on NR3D	61.3
BUTD-DETR trained on ScanRefer	**63.0**

sample at most 10 positive classes. We leave tuning of this hyperparameter for future research.

1.7 Detection results

A benefit of i) being able to ground all objects mentioned in the phrase and not only the target object, as well as ii) being trained with detection prompts, is that BUTD-DETR can operate as an object detector. We evaluate its performance on ScanNet benchmark which has 18 classes. Specifically, for each scene, we form a detection prompt that contains all 18 classes. The objective is to find all instances in the scene, as explained in Section 1.6.

We first train BUTD-DETR on ScanNet using the same prompt of 18 classes. This is analogous to a 3DETR [16] model with PointNet++ backbone or the DETR+KPS+iter ablation in Table 10 of [14]. Additionally, we evaluate BUTD-DETR trained on a language grounding benchmark. The results are shown in Table 5. BUTD-DETR performs on par with the ablation of [14], but worse than 3DETR. Note that our objectives, i.e. contrastive losses, are not optimized for classification across a fixed number of classes, but for query-span alignment. Instead, detectors use softmax layers over a known number of classes. For comparison, we train BUTD-DETR on ScanNet with a softmax loss over the 18 benchmark classes to observe an improvement of 1.7%. However, softmax losses are not suitable for language grounding, where the labels are not a priori known or limited to a specific set. When BUTD-DETR is trained on the 3D referential datasets, the performance on ScanNet improves up to 3.7%, without having access to more scenes. This suggests that co-training with grounding and detection prompts is beneficial for both tasks.

1.8 More qualitative results

We show qualitative results of the 2D version of BUTD-DETR on RefCOCO in Figure 1. We also show failure cases on SR3D in Figure 2. More qualitative results on SR3D and NR3D are shown in Figures 3, 4, 5.

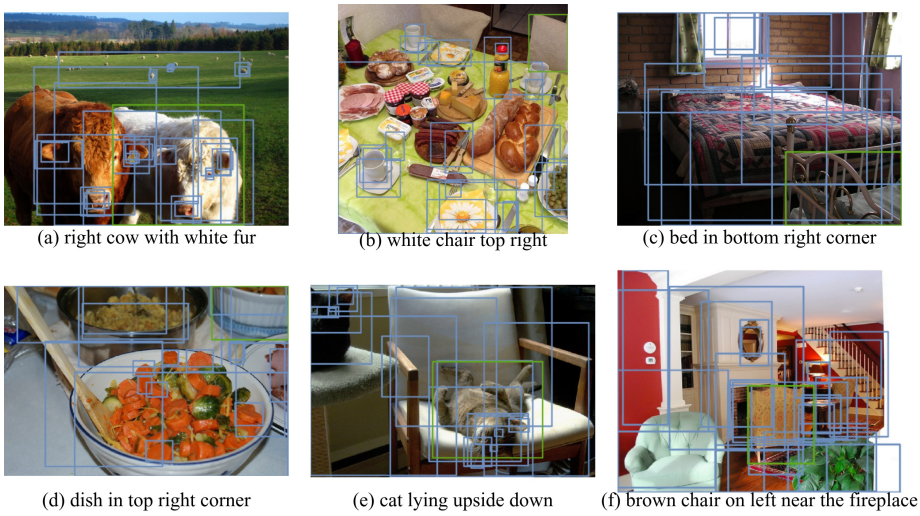


Fig. 1: **Qualitative results of BUTD-DETR on RefCOCO.** The detector's proposals are shown in blue, our model's prediction in green. BUTD-DETR can predict boxes that the detector misses, e.g. in (b), the chair is missed by the detector so none of the previous detection-bottlenecked approaches could ground this phrase. In (a) and (c) the detector succeeds with low IoU but BUTD-DETR is able to predict a tight box around the referent object.

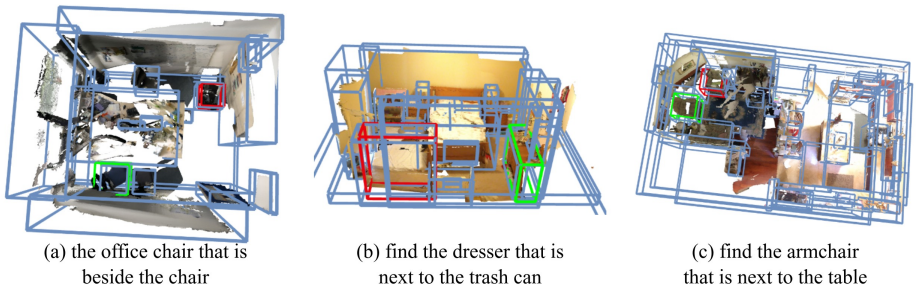


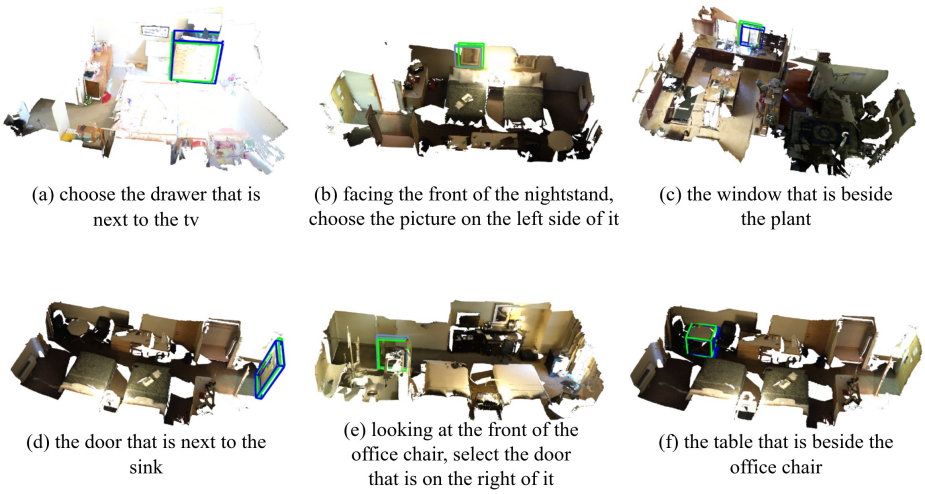
Fig. 2: Failure cases of BUTD-DETR on SR3D. Our predictions with red, ground-truth with green. Even if the box is there, still our model can fail, proving that ranking the correct boxes over other proposals remains a hard problem.

(a) choose the desk on the opposite side of the couch

(b) Facing the beds you want the front pillow on the left bed.

(c) Standing at the foot of the bed looking at the head on the bed. It is the pillow in the from on the bed on the right.

Fig. 3: Qualitative results of BUTD-DETR on NR3D. Our predictions are shown blue, ground-truth in green. The language of NR3D is more complex and the utterances are longer. Case (c) is a failure case.



(a) choose the drawer that is next to the tv

(b) facing the front of the nightstand, choose the picture on the left side of it

(c) the window that is beside the plant

(d) the door that is next to the sink

(e) looking at the front of the office chair, select the door that is on the right of it

(f) the table that is beside the office chair

Fig. 4: More qualitative results of BUTD-DETR on SR3D. Our predictions are shown in blue, ground-truth in green.

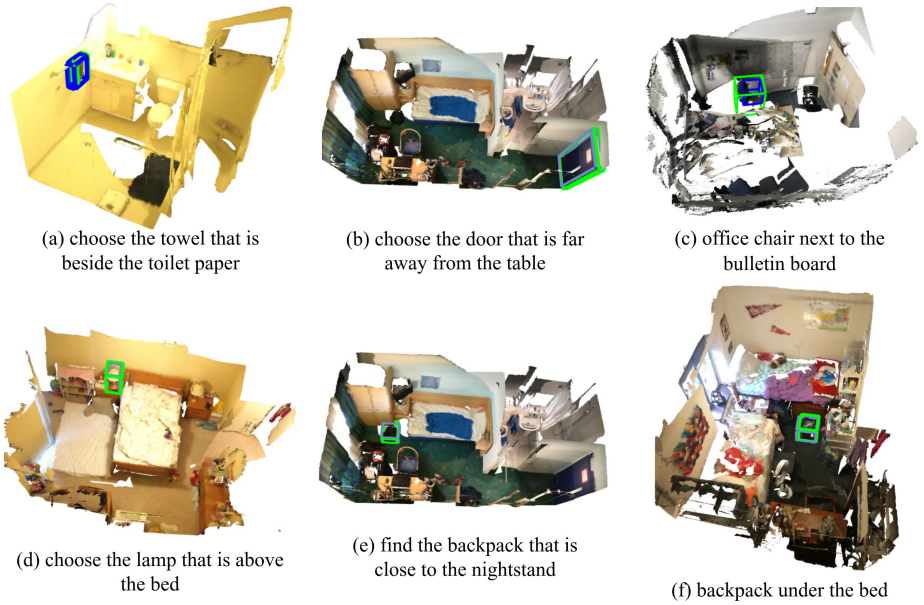


Fig. 5: More qualitative results of BUTD-DETR on SR3D. Our predictions are shown in blue, ground-truth in green.

References

1. Abdelreheem, A., Upadhyay, U., Skorokhodov, I., Yahya, R.A., Chen, J., Elhoseiny, M.: 3DRefTransformer: Fine-Grained Object Identification in Real-World Scenes Using Natural Language. In: Proc. WACV (2022) 2, 3
2. Achlioptas, P., Abdelreheem, A., Xia, F., Elhoseiny, M., Guibas, L.: ReferIt3D: Neural Listeners for Fine-Grained 3D Object Identification in Real-World Scenes. In: Proc. ECCV (2020) 1, 2, 3, 4
3. Anderson, P., He, X., Buehler, C., Teney, D., Johnson, M., Gould, S., Zhang, L.: Bottom-Up and Top-Down Attention for Image Captioning and Visual Question Answering. In: Proc. CVPR (2018) 2
4. Chen, D.Z., Chang, A., Nießner, M.: ScanRefer: 3D Object Localization in RGB-D Scans using Natural Language. In: Proc. ECCV (2020) 1, 3
5. Dai, A., Chang, A.X., Savva, M., Halber, M., Funkhouser, T.A., Nießner, M.: ScanNet: Richly-Annotated 3D Reconstructions of Indoor Scenes. In: Proc. CVPR (2017) 1
6. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: A large-scale hierarchical image database. In: Proc. CVPR (2009) 2
7. Feng, M., Li, Z., Li, Q., Zhang, L., Zhang, X., Zhu, G., Zhang, H., Wang, Y., Mian, A.: Free-form Description Guided 3D Visual Graph Network for Object Grounding in Point Cloud. In: Proc. ICCV (2021) 3
8. He, D., Zhao, Y., Luo, J., Hui, T., Huang, S., Zhang, A., Liu, S.: TransRefer3D: Entity-and-Relation Aware Transformer for Fine-Grained 3D Visual Grounding. In: Proc. ACMMM (2021) 2, 3
9. He, K., Zhang, X., Ren, S., Sun, J.: Deep Residual Learning for Image Recognition. In: Proc. CVPR (2016) 2
10. Huang, P.H., Lee, H.H., Chen, H.T., Liu, T.L.: Text-Guided Graph Neural Networks for Referring 3D Instance Segmentation. In: Proc. AAAI (2021) 2, 3
11. Krishna, R., Zhu, Y., Groth, O., Johnson, J., Hata, K., Kravitz, J., Chen, S., Kalantidis, Y., Li, L.J., Shamma, D.A., Bernstein, M.S., Fei-Fei, L.: Visual Genome: Connecting Language and Vision Using Crowdsourced Dense Image Annotations. International Journal of Computer Vision **123** (2016) 2
12. Lin, T.Y., Goyal, P., Girshick, R.B., He, K., Dollár, P.: Focal Loss for Dense Object Detection. In: Proc. ICCV (2017) 1, 2
13. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V.: RoBERTa: A Robustly Optimized BERT Pretraining Approach. ArXiv **abs/1907.11692** (2019) 1, 2
14. Liu, Z., Zhang, Z., Cao, Y., Hu, H., Tong, X.: Group-Free 3D Object Detection via Transformers. In: Proc. ICCV (2021) 1, 2, 6
15. Lu, J., Batra, D., Parikh, D., Lee, S.: ViLBERT: Pretraining Task-Agnostic Visiolinguistic Representations for Vision-and-Language Tasks. In: Proc. NeurIPS (2019) 1, 2
16. Misra, I., Girdhar, R., Joulin, A.: An End-to-End Transformer Model for 3D Object Detection. In: Proc. ICCV (2021) 6
17. Qi, C., Litany, O., He, K., Guibas, L.J.: Deep Hough Voting for 3D Object Detection in Point Clouds. In: Proc. ICCV (2019) 4
18. Qi, C., Yi, L., Su, H., Guibas, L.J.: PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. In: Proc. NIPS (2017) 1
19. Roh, J., Desingh, K., Farhadi, A., Fox, D.: LanguageRefer: Spatial-Language Model for 3D Visual Grounding. In: Proc. CoRL (2021) 2, 3

20. Vaswani, A., Shazeer, N.M., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is All you Need. In: Proc. NIPS (2017) 1, 2
21. Yang, Z., Zhang, S., Wang, L., Luo, J.: SAT: 2D Semantics Assisted Training for 3D Visual Grounding. In: Proc. ICCV (2021) 2, 3
22. Yu, Z., Yu, J., Xiang, C., Zhao, Z., Tian, Q., Tao, D.: Rethinking Diversified and Discriminative Proposal Generation for Visual Grounding. In: Proc. IJCAI (2018) 4
23. Yuan, Z., Yan, X., Liao, Y., Zhang, R., Li, Z., Cui, S.: InstanceRefer: Cooperative Holistic Understanding for Visual Grounding on Point Clouds through Instance Multi-level Contextual Referring. In: Proc. ICCV (2021) 2, 3
24. Zhao, L., Cai, D., Sheng, L., Xu, D.: 3DVG-Transformer: Relation Modeling for Visual Grounding on Point Clouds. In: Proc. ICCV (2021) 2, 3
25. Zhu, X., Su, W., Lu, L., Li, B., Wang, X., Dai, J.: Deformable DETR: Deformable Transformers for End-to-End Object Detection. In: Proc. ICLR (2021) 2