Supplementary Material for
EDA: Explicit Text-Decoupling and Dense Alignment for 3D Visual Grounding

Yanmin Wu[1] Xinhua Cheng[1] Renrui Zhang[2,3] Zesen Cheng[1] Jian Zhang[1]*

[1] Shenzhen Graduate School, Peking University, China
[2] The Chinese University of Hong Kong, China [3] Shanghai AI Laboratory, China

wuyanmin@stu.pku.edu.cn zhangjian.sz@pku.edu.cn

Section A of the supplementary material provides the implementation details of the individual modules and the network training details. In Section B , we supplement with additional experiments and quantitative analyses. Finally, in Section C , we present visualization results and qualitative analysis.

A. Implementation details

Text decoupling module. The maximum length of the text is l=256, and the absence bit of the position label $L \in \mathbb{R}^{1\times l}$ is padded with 0. Not every sentence can be decoupled into five semantic components, but the most fundamental "main object" is required.

Encoder-Decoder. We keep hyperparameters consistent with BUTD-DETR [1]. The point cloud is tokenized as $\mathcal{V} \in \mathbb{R}^{n\times d}$ by the PointNet++ [4] pre-trained on ScanNet. The text is tokenized as $\mathcal{T} \in \mathbb{R}^{l\times d}$ by the pre-trained RoBERTa [2]. Following object detection, the position and category of the boxes are embedded separately and concatenated as the box token $\mathcal{B} \in \mathbb{R}^{b\times d}$. The encoder, for visual-text feature extraction and modulation, is N_E=3 layers. The decoder with N_D=6 layers generates candidate object features $\mathcal{Q} \in \mathbb{R}^{k\times d}$. Where n=1024 denotes the number of seed points, l=256 the number of texts, b=132 the number of detection boxes, k=256 the number of candidate objects, and d=288 the feature dimension. Please refer to BUTD-DETR for more details.

Losses. 1) In the **position-aligned loss** $\mathcal{L}_{pos}$, the weights of each component in Eq. (1) are as follows: λ_1=0.6, λ_2=λ_3=0.2, λ_4=0.1. These values indicate that the weight of the "main object" component L_{main} is higher, which is obvious. The "relational" component L_{rel} with lower weight because it affects both the main and auxiliary objects. See Sec. B.1.(4) for parameter searching. **2)** In the **semantic-aligned loss** $\mathcal{L}_{sem}$, the weight w_+ follows a similar trend. The four features $\boldsymbol{t}_{main}, \boldsymbol{t}_{attri}, \boldsymbol{t}_{pron}, \boldsymbol{t}_{rel}$ are weighted by $1.0, 0.2, 0.2$, and 0.1, respectively. The weight w_- acts on the negative item, where the feature weight of the auxiliary object is 2 and the remainder weighs 1. The purpose is to differentiate the features of the main object from the auxiliary objects. **3)** We optimize the model with the following **total loss**:

$$\mathcal{L} = (\alpha(\mathcal{L}_{pos} + \mathcal{L}_{sem}) + 5\mathcal{L}_{box} + \mathcal{L}_{iou})/(N_D + 1) + 8\mathcal{L}_{pts}, \tag{8}$$

where $\mathcal{L}_{pos}$ and $\mathcal{L}_{sem}$ represent the visual-language alignment loss. $\mathcal{L}_{box}$ and $\mathcal{L}_{iou}$ indicate the object detection loss [3], with $\mathcal{L}_{box}$ representing the L1 regression loss of the object's position and size and $\mathcal{L}_{iou}$ representing the object's 3D IoU loss N_D is the layer number of the Decoder. $\mathcal{L}_{pts}$ is the KPS point samping loss [3]. α takes the value 1 in the SR3D/NR3D dataset and 0.5 in the ScanRefer dataset. Because the SR3D/NR3D dataset provides the bounding box of candidate objects, while the ScanRefer dataset requires detecting the bounding box, we give higher weights for the detection loss in the ScanRefer dataset.

Training details. The code is implemented based on PyTorch. We set the batch size to 12 on four 24-GB NVIDIA-RTX-3090 GPUs. For ScanRefer, we use a $2e{-}3$ learning rate for the visual encoder and a $2e{-}4$ learning rate for all other layers. It takes about 15 minutes per epoch, and around epoch 60, the best model appears. The learning rates for SR3D are $1e{-}3$ and $1e{-}4$, 25 minutes per epoch, requiring around 45 epochs of training. The learning rates for NR3D are set at $1e{-}3$ and $1e{-}4$, 15 minutes per epoch, and around 180 epochs are trained. Since SR3D is composed of brief machine-generated sentences, convergence is easier. ScanRefer and NR3D are comprised of human-annotated free-form complex descriptions, respectively, and require more training time.

B. Additional experiments

B.1. Regular 3D Visual Grounding

(1) The explanation of the BUTD-DETR's performance. Given a sentence, such as *"It is a brown chair with armrests and four legs . It is directly under a blackboard"*, our text decoupling module determines that *"chair"* is the main object based on grammatical analysis and thus obtains the position label $L_{main} = 0000100....$ However, in the official implementation of BUTD-DETR, which requires an additional ground truth class for the target object, its input is: *"<object name> chair. <Description> It is a brown chair ..."*. Then search for the position where the object name *"chair"* appears in the sentence as a position label. This operation presents some problems:

- **i)** It is unfair to use GT labels during inference;
- **ii)** Descriptions may employ synonyms for the category *"chair,"* such as *"armchair, office-chair, and loveseat,"* leading to a failed search position label;
- **iii)** Sometimes, the object name is not mentioned, such as when it is replaced by the word *"object."* In the NR3D validation set, BUTD-DETR removed 800 such challenging samples, and about 5% did not participate in the evaluation.

To be fair, we re-evaluate it using the position labels obtained by the proposed text decoupling module, as displayed in the second row in Tab. 6.

	ScanRefer		SR3D	NR3D
	0.25IoU	0.5IoU		
Official	52.2	39.8	67.1	55.4
Re-evaluation	50.4	38.6	65.6	49.1

Table 6. Performance of BUTD-DETR, where SR3D/NR3D only use Acc@0.25IoU as the metric.

(2) Evaluation of ScanRefer using GT box. In the ScanRefer dataset, only the point cloud is provided as visual input, requiring object detection and language-based object grounding. Conversely, SR3D/NR3D offers additional GT boxes of candidate objects. Therefore, we further evaluate the ScanRefer dataset by GT boxes. As shown in Tab. 7, our performance improves significantly without retraining, particularly in the unique setting where accuracy exceeds 90%. This result demonstrates that more accurate object detection can further enhance our performance.

Method	Unique		Multiple		**Overall**	
	0.25	0.5	0.25	0.5	**0.25**	**0.5**
BUTD-DETR	85.62	68.64	46.07	35.51	52.0	40.5
EDA (Ours)	**90.91**	**75.33**	**51.71**	**40.66**	**57.6**	**45.8**

Table 7. Performance on ScanRefer using GT box. Our method presents significant advantages.

(3) Detailed results on the SR3D/NR3D dataset. Due to page limitations, we only report overall performance in Table 2. Table 8 breaks down the detailed results of our method into four subsets: easy, hard, view-dependent, and view-independent.

Dataset	Easy	Hard	View-dep.	View-indep.	**Overall**
SR3D	70.3	62.9	54.1	68.7	**68.1**
NR3D	58.2	46.1	50.2	53.1	**52.1**

Table 8. Detailed performances of our method on the SR3D/NR3D dataset with the metric of Acc@0.25IoU.

(4) Parameter search for the weight λ. The representative results of a grid search on the weights in Eq. (1) are presented in Table 9. Either of these options outperforms existing methods, demonstrating the efficiency of our dense alignment. As seen in (a), it is not optimal to treat all components equally because their functions are not equivalent. When giving λ_1 a higher weight (see (f, g)), it turns out that a weight that is too high would also lead to a decrease in performance, which may compromise the functionality of other components. λ_1 takes 0.6 as the best option, and the other items take 0.1 or 0.2. We select option (d) for implementation.

	$\lambda_1, \lambda_2, \lambda_3, \lambda_4$	@0.25IoU	@0.5IoU
(a)	1.0, 1.0, 1.0, 1.0	53.6	40.6
(b)	0.5, 0.2, 0.2, 0.2	53.3	40.9
(c)	0.6, 0.2, 0.2, 0.2	53.5	41.2
(d)	0.6, 0.2, 0.2, 0.1	**54.6**	**42.3**
(e)	0.6, 0.1, 0.1, 0.1	54.5	42.0
(f)	0.8, 0.2, 0.2, 0.2	52.9	41.5
(g)	1.0, 0.2, 0.2, 0.2	53.2	40.3

Table 9. Grid search of the weight λ. Evaluated on the ScanRefer dataset. We select (d) for implementation.

(5) Does the text component really help? Based on the "main object", we densely align the other four text components ("Attribute", "Relationship", "Pronoun", and "Auxiliary object") with visual features. The question immediately arises whether the random alignment with some words yields the same gain. As a comparison, we randomly select four words in the sentence to align with the visual features. As shown in Table 10(b), although the performance is improved by 0.5%, it is lower than when only one of the four components is aligned (c,d) and substantially worse than aligning all four (e). This 0.5% gain may be due to the randomly aligned words with the possibility of involving four text components. The results demonstrate our insight into dense alignment and decoupling of meaningful components.

	Main	Random 4 words	One of {Attr, Pron, Auxi, Rel} (lowest)	 (best)	Attr + Pron + Auxi + Rel	Acc.
(a)	✓					51.5
(b)	✓	✓				52.0
(c)	✓		✓			52.8
(d)	✓			✓		53.1
(e)	✓				✓	**54.6**

Table 10. Comparison with the alignment of four random words. The metric is Acc@0.25IoU. (a): Baseline, only aligned with the "Main Object" text component; (b): aligned with four random words; (c-d): aligned with one of our four decoupled components; (e): aligned with all four components.

B.2. Language Modulated 3D Object Detection

We maintain the same experimental setup as BUTD-DETR to evaluate the performance of 3D object detection on ScanNet. The 18 classes in ScanNet are concatenation into a sentence: *"bed. bookshelf. cabinet. chair. counter. curtain. desk. door ..."* as text input. Output the bounding boxes and classes of all objects in the point cloud. As shown in Tab. 11, our model after text modulation on the ScanRefer achieves 1.1% and 1.5% higher performance than BUTD-DETR, to 64.1% and 45.3%. Note that the proposed method is not specifically designed for object detection, and the performance evaluation uses the **same model** as the visual grounding task (Table 1 and Table 5).

Method	mAP@0.25	mAP@0.5
DETR+KPS+iter †	59.9	-
3DETR with PointNet++ †	61.7	-
BUTD-DETR trained on ScanRefer	63.0	43.8
EDA trained on ScanRefer (Ours)	**64.1**	**45.3**

Table 11. 3D Object detection performance on ScanNet. † The accuracy is provided by BUTD-DETR.

C. Qualitative analysis

1) Regular 3D Visual Grounding. Qualitative results on the regular 3D visual grounding task are displayed in Fig. 5, 6, 7. **i)** Fig. 5 indicates that compared to BUTD-DETR, our method has a superior perception of appearance, enabling the identification of objects based on their attributes among several candidates of the same class. This improvement is made possible by the alignment of our decoupled text attribute component with visual features. **ii)** Fig. 6 demonstrates that our method exhibits excellent spatial awareness, such as orientation and position relationships between objects. The alignment of our decoupled relational component with visual features and the positional encoding of Transformer may be advantageous to this capability. **iii)** Furthermore, we surprisingly found that our method also has a solid understanding of ordinal numbers, as shown in Fig. 7, probably because we parsed ordinal numbers as part of the attribute component of the object. These examples demonstrate that text decoupling and dense alignment enable fine-grained visual-linguistic matching.

2) Grounding without Object Name (VG-w/o-ON). The visualization results of this challenging task are shown in Fig. 9. Since the target object's name is not provided, the model must make inferences based on appearance and positional relationships with auxiliary objects. However, other contrastive methods perform weakly on this task because they rely heavily on object names to exclude interference candidates, weakening the learning of other attributes.

3) Failure Case Analysis. Although our method delivers state-of-the-art performance, there are still a significant number of failure occurrences, which we analyze visually. **i)** Many language descriptions are intrinsically ambiguous, as illustrated in Fig. 8(a-c), especially in the "multiple" setting, the appearance attributes and spatial relationships of the target object are not unique, and there are multiple alternatives for candidate objects that match the requirements. **ii)** The text parsing error may occur owing to the language description's complexity and diversity. Such as, the GT object in Fig. 8(d) is a desk, but we parse it as a window; the GT object in Fig. 8(e) is a box, but we parse it as a piano. **iii)** There are also some cases where cross-modal feature matching fails even though the text parses well.

References

[1] Ayush Jain, Nikolaos Gkanatsios, Ishita Mediratta, and Katerina Fragkiadaki. Bottom up top down detection transformers for language grounding in images and point clouds. In *Proceedings of the European Conference on Computer Vision*, pages 417–433. Springer, 2022. 1

[2] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019. 1

[3] Ze Liu, Zheng Zhang, Yue Cao, Han Hu, and Xin Tong. Group-free 3d object detection via transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2949–2958, 2021. 1

[4] 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

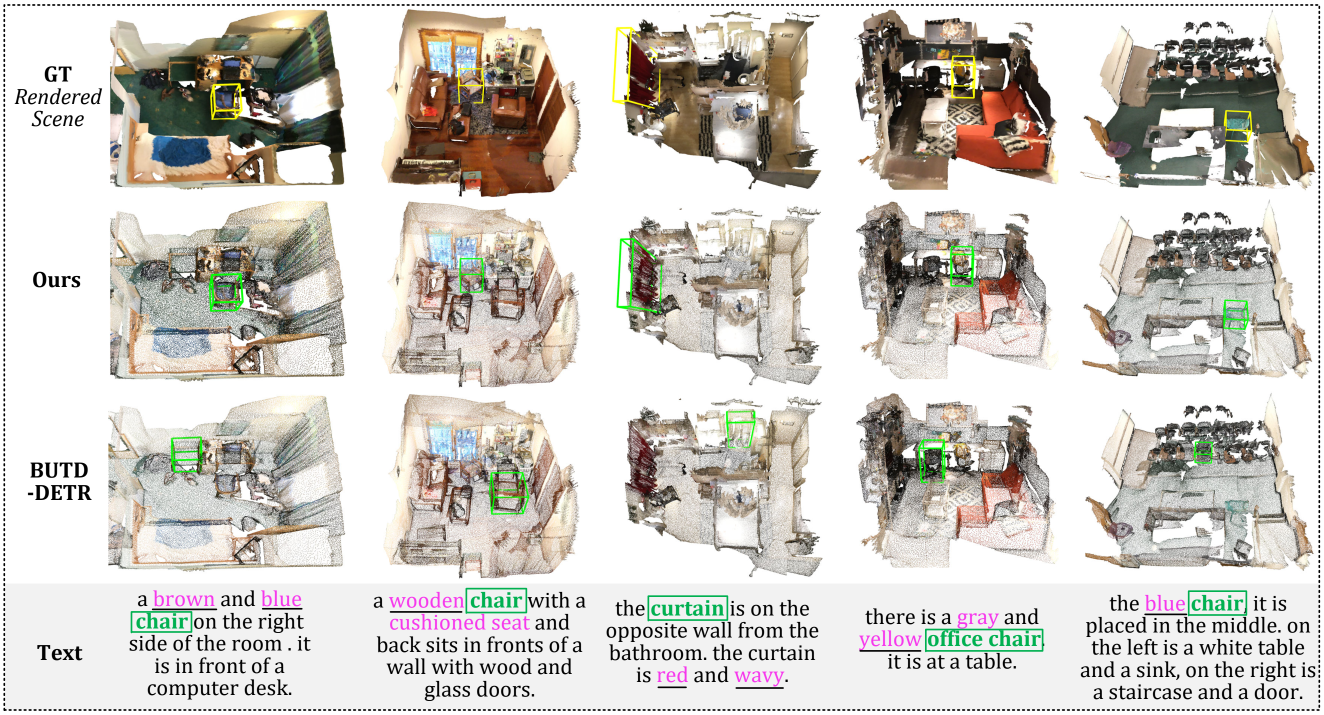

GT *Rendered Scene*
Ours
BUTD -DETR
Text
a brown and blue chair on the right side of the room . it is in front of a computer desk.
a wooden chair with a cushioned seat and back sits in fronts of a wall with wood and glass doors.
the curtain is on the opposite wall from the bathroom. the curtain is red and wavy.
there is a gray and yellow office chair. it is at a table.
the blue chair. it is placed in the middle. on the left is a white table and a sink, on the right is a staircase and a door.

Figure 5. Qualitative comparison of the **regular 3D VG task**. Our method has a superior perception of **appearance attributes**.

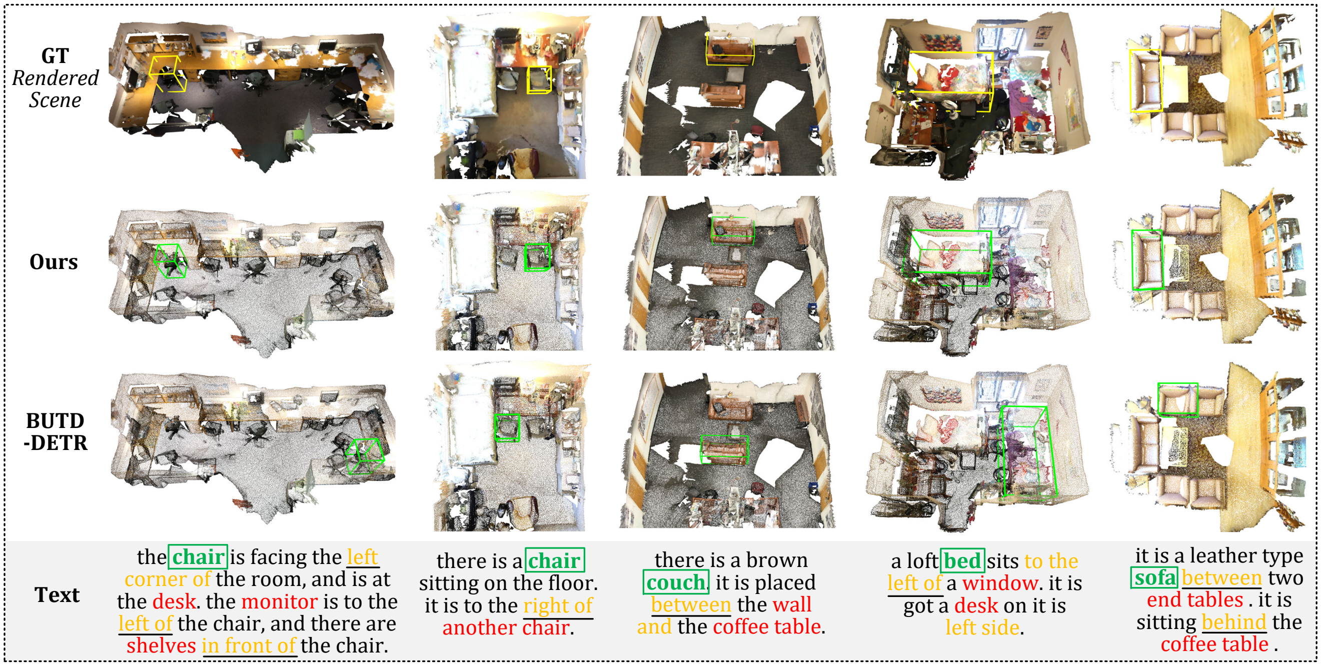

GT *Rendered Scene*
Ours
BUTD -DETR
Text
the chair is facing the left corner of the room, and is at the desk. the monitor is to the left of the chair, and there are shelves in front of the chair.
there is a chair sitting on the floor. it is to the right of another chair.
there is a brown couch it is placed between the wall and the coffee table.
a loft bed sits to the left of a window. it is got a desk on it is left side.
it is a leather type sofa between two end tables . it is sitting behind the coffee table .

Figure 6. Qualitative comparison of the **regular 3D VG task**. Our method has a superior perception of **spatial relationships**.

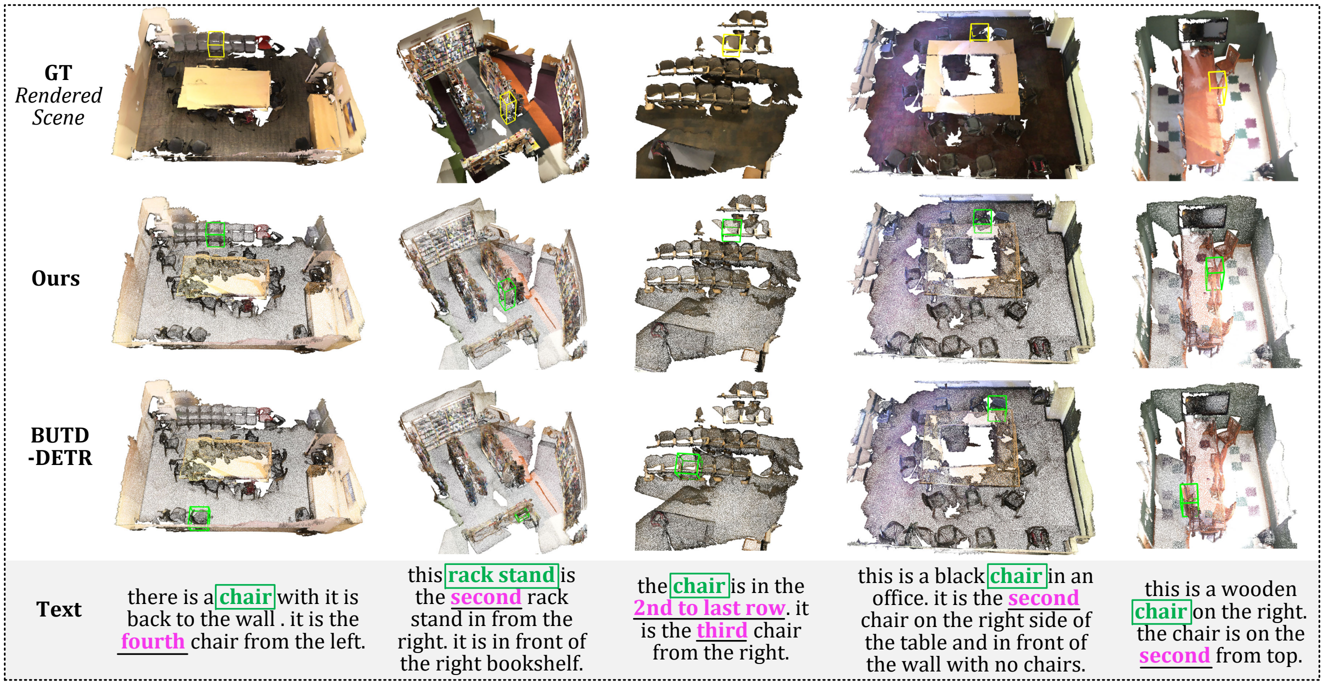
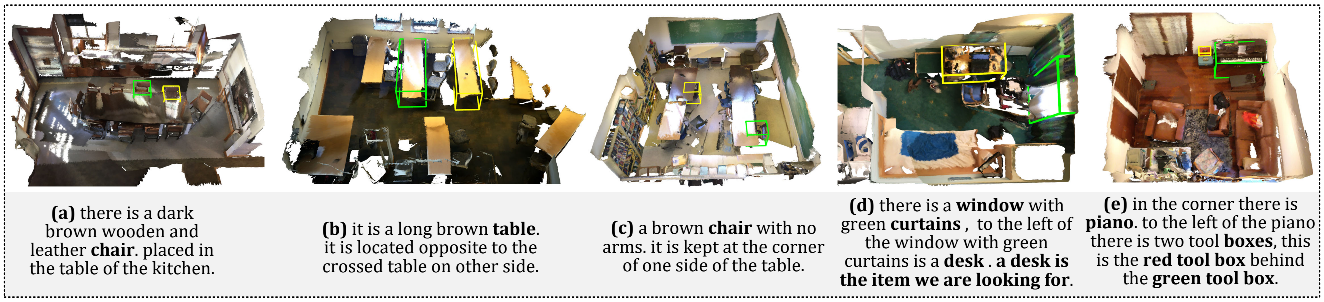

GT
Rendered
Scene
Ours
BUTD
-DETR
Text
there is a chair with it is back to the wall . it is the fourth chair from the left.
this rack stand is the second rack stand in from the right. it is in front of the right bookshelf.
the chair is in the 2nd to last row. it is the third chair from the right.
this is a black chair in an office. it is the second chair on the right side of the table and in front of the wall with no chairs.
this is a wooden chair on the right. the chair is on the second from top.

Figure 7. Qualitative comparison of the **regular 3D VG task**. Challenging cases with **ordinal numbers**.



(a) there is a dark brown wooden and leather **chair**. placed in the table of the kitchen.

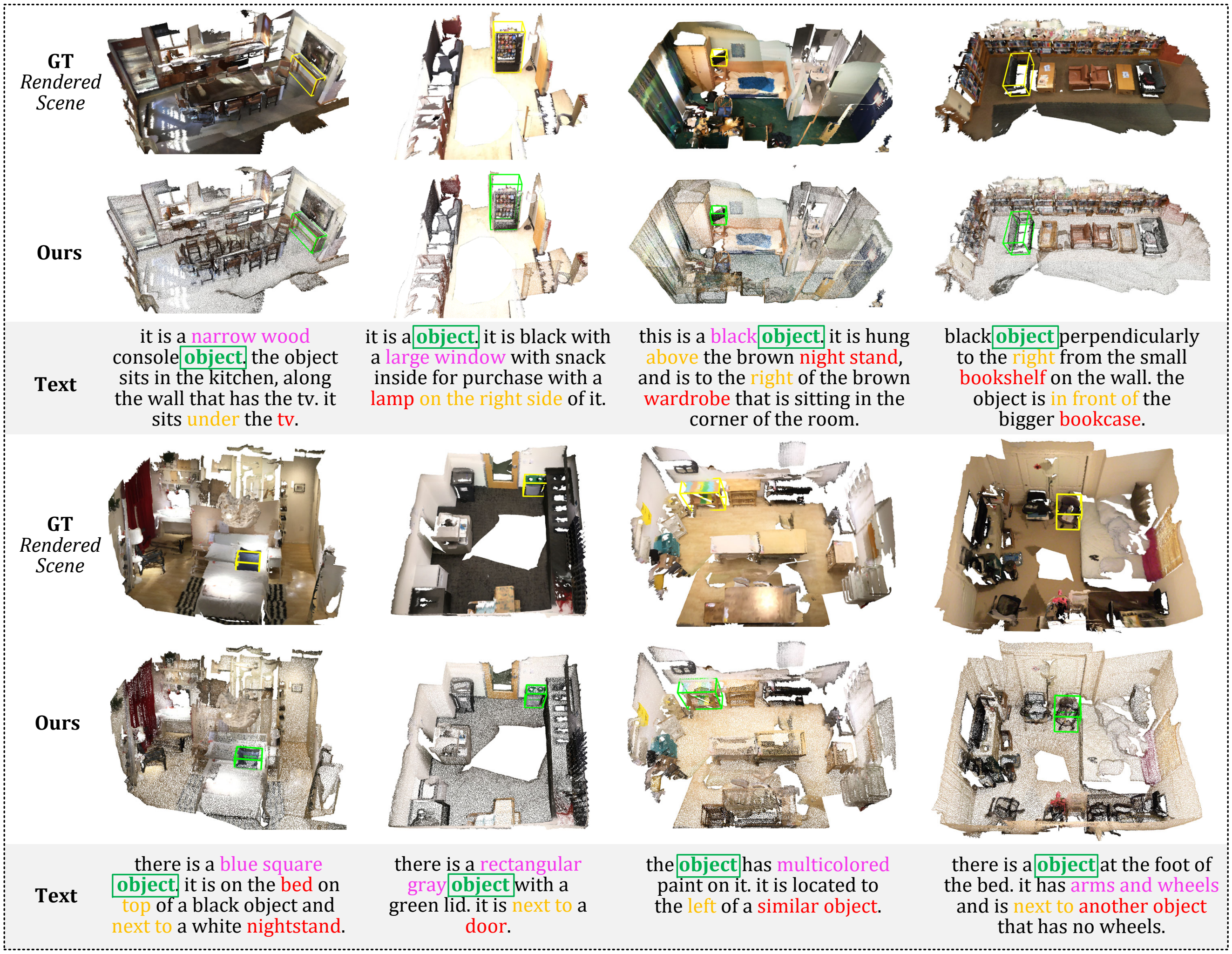
(b) it is a long brown **table**. it is located opposite to the crossed table on other side.

(c) a brown **chair** with no arms. it is kept at the corner of one side of the table.

(d) there is a **window** with green **curtains** , to the left of the window with green curtains is a **desk** . **a desk is the item we are looking for**.

(e) in the corner there is **piano**. to the left of the piano there is two tool **boxes**, this is the **red tool box** behind the **green tool box**.

Figure 8. **Failure cases**, with the GT box and the predicted box shown in yellow and green, respectively. (a-c): Failure due to ambiguity of reference. (d-e): Failure due to text parsing error for complex and long sentences.



it is a narrow wood console object. the object sits in the kitchen, along the wall that has the tv. it sits under the tv.

it is a object. it is black with a large window with snack inside for purchase with a lamp on the right side of it.

this is a black object. it is hung above the brown night stand, and is to the right of the brown wardrobe that is sitting in the corner of the room.

black object perpendicularly to the right from the small bookshelf on the wall. the object is in front of the bigger bookcase.

there is a blue square object. it is on the bed on top of a black object and next to a white nightstand.

there is a rectangular gray object with a green lid. it is next to a door.

the object has multicolored paint on it. it is located to the left of a similar object.

there is a object at the foot of the bed. it has arms and wheels and is next to another object that has no wheels.

Figure 9. 3D Visual grounding without object name (**VG-w/o-ON**), where the word “object” replaces the target’s name.