Supplementary Materials

A. Data Augmentation

A.1. Color Augmentation

We have further enhanced our data augmentation techniques by introducing color perturbations. Specifically, we add controlled noise to the color of each point and randomly adjust the contrast. See Eq. (1), where $\alpha \in [0.5, 1.5]$ and Noise $\sim \mathcal{N}(0, 1)$.

$$RGB_{aug} = \alpha \times RGB + \text{Noise} \tag{1}$$

A.2. Object Location Augmentation

Previous work by MVT [2, 4] demonstrates the benefits of a multi-view rotation method to enhance point cloud inputs. By rotating the object's location N_R times, we define the rotation angle θ as shown in Eq. (2).

$$\theta_j = \frac{j \times 2\pi}{N_R}, \quad \text{for } j \in \{1, 2, \ldots, N_R\} \tag{2}$$

With this augmentation, models can develop more robust object representations, less influenced by the initial viewpoint. In our study, we adopted this technique and confirmed MVT's findings that four rotations yield better results than two or three. However, increasing to eight rotations does not bring additional benefits and substantially increases computational load. See Table 1.

For an input $P = \{p_1, p_2, \ldots, p_{N_p}\}$, we construct the rotation matrix $R_{\theta_j} = R(\frac{j \times 2\pi}{N_v})$ for $j \in \{1, 2, \ldots, N_v\}$. For each p_i in P, and for each rotation j, we generate the rotated point p_i^j in the augmented set P_{aug}, as defined by Eq. (3), where P_{aug} has $N_R \times N_p$ elements.

$$p_i^j = R_{\theta_j}(p_i), \quad \begin{array}{l} \text{for } i \in \{1, 2, \ldots, N_p\}, \\ \text{for } j \in \{1, 2, \ldots, N_v\} \end{array} \tag{3}$$

Finally, we utilize Eq. (4) to aggregate multi-view object features O_i^j, which are D dimensional features, given by the networks.

$$O_i^{agg} = \frac{1}{N_v} \sum_{j=1}^{N_v} O_i^j, \quad \text{for } i \in \{1, 2, \ldots, N_p\} \tag{4}$$

Rotations	Overall1	Easy	Hard	VD	VI
1	60.0%	66.6%	53.7%	57.1%	61.5%
2	62.7%	69.4%	56.2%	62.2%	63.0%
3	63.1%	68.8%	57.5%	62.8%	63.2%
4	**64.4%**	**69.7%**	**59.4%**	**65.4%**	**64.0%**
8	62.6%	68.8%	56.6%	62.6%	62.6%

Table 1. Comparative analysis of our model's accuracy on the Nr3D [1] dataset for different numbers of rotations applied during the augmentation process. The optimal accuracy is achieved with four rotations, while more rotations do not significantly improve accuracy and considerably increase computational complexity.

B. Model Performance Across Datasets

In this section, we conduct a comprehensive examination of our model's adaptability and robustness by training across a variety of datasets. This approach not only demonstrates the model's generalizability but also provides insights into how different data characteristics influence performance outcomes. Specifically, we investigate three distinct training settings: joint training on Nr3D [1] and Sr3D/Sr3D+ [1], and individual training on each of the three datasets separately.

B.1. Training on Sr3D+

Previous works have primarily focused on models trained on the Nr3D and Sr3D datasets. However, the performance of models trained exclusively on the expanded Sr3D+ dataset, which augments Sr3D by including samples with no more than one distractor, remains unexplored. This section offers a comparative analysis of model accuracies when trained separately on Nr3D, Sr3D, and Sr3D+. As shown in Table 2, models trained on Sr3D

Dataset	Overall1	Easy	Hard	VD	VI
Nr3D	64.4%	69.7%	59.4%	65.4%	64.0%
Sr3D	75.2%	78.6%	67.3%	70.4%	75.4%
Sr3D+	72.7%	75.4%	66.4%	69.8%	72.8%

Table 2. Comparison of model accuracies when trained on Nr3D, Sr3D, and Sr3D+ datasets. The results illustrate that training on Sr3D and Sr3D+ yields similar and notably higher accuracies compared to Nr3D.

Method	Nr3D with Sr3D					Nr3D with Sr3D+				
	Overall1	Easy	Hard	VD	VI	Overall1	Easy	Hard	VD	VI
ReferIt3D [1]	37.2%	44.0%	30.6%	33.3%	39.1%	37.6%	45.4%	30.0%	33.1%	39.8%
non-SAT [6]	43.9%	-	-	-	-	45.9%	-	-	-	-
TransRefer3D [3]	47.2%	55.4%	39.3%	40.3%	50.6%	48.0%	56.7%	39.6%	42.5%	50.7%
SAT [6]	53.9%	61.5%	46.7%	52.7%	54.5%	56.5%	64.9%	48.4%	54.4%	57.6%
MVT [4]	58.5%	65.6%	51.6%	56.6%	59.4%	59.5%	67.4%	52.7%	59.1%	60.3%
Ours	**63.4%**	**69.5%**	**57.6%**	**64.2%**	**63.1%**	**63.9%**	**71.0%**	**57.0%**	**63.1%**	**64.2%**
vs. second best	+4.9%	+3.9%	+6.0%	+7.6%	+3.7%	+4.4%	+3.6%	+4.3%	+4.0%	+3.9%

Table 3. Comparative analysis of accuracy for joint training on Nr3D with Sr3D and Nr3D with Sr3D+ datasets. The table highlights that although our model still outperforms previous works under the same settings, it does not yield better results compared to training solely on each of the three datasets.

x*	✓	✓	✓	-
y*	✓	✓	✓	-
z*	✓	✓	-	✓
d*	✓	-	✓	✓
Acc	**64.4%**	62.7%	63.3%	63.5%

Table 4. Model accuracies on Nr3D utilizing different spatial features. Features marked with an asterisk ('*') denote normalized data, while those without a checkmark indicate raw data.

consistently achieve the highest accuracy overall and across all categories. This is noteworthy considering that Sr3D+ expands the dataset; yet, the addition of these less complex samples does not translate to enhanced performance. Both Sr3D and Sr3D+ models exhibit comparable performance levels, significantly surpassing the accuracy achieved with Nr3D in all categories.

B.2. Joint Training on Nr3D and Sr3D/Sr3D+

We explored the impact of expanding Nr3D's training data with Sr3D and Sr3D+ datasets. The comparative results are detailed in Tab. 3. Our findings reveal that combining Nr3D with Sr3D or Sr3D+ offers no significant advantage over using each of them alone. The performance in these joint training setups was similar to that of Nr3D trained in isolation and was markedly lower than when the model was trained on Sr3D or Sr3D+ alone. This suggests that, with our method, Sr3D and Sr3D+ data do not contribute significant benefits when combined with Nr3D. Instead, the advantages of Sr3D and Sr3D+ seem to be diminished when merged with Nr3D, indicating that the augmentation does not necessarily translate to improved performance and may, in fact, hinder the distinct features of Sr3D/Sr3D+ datasets from being fully leveraged.

C. Spatial Feature Scaling & Normalization

To enhance the model's comprehension of spatial relationships, we define the relative coordinates (x, y, z) of an anchor object i with the target object as the reference origin. The distance d in the xy-plane between the target and the anchor object is represented by $d_{xy} = \sqrt{x^2 + y^2}$. With these definitions in place, we proceed to normalize and scale the spatial features as detailed in Equations Eq. (5), Eq. (6), and Eq. (7). The impact of these normalization steps on model accuracy is summarized in Table 4.

Determine the Direction:
For each potential anchor object, we scale (x, y) to obtain a unit vector that indicates the direction from the target object. As shown in Eq. (5).

$$(x^*, y^*) = \frac{(x, y)}{\|(x, y)\|_2} \tag{5}$$

Scale the Relative Height:
We also scale the relative height z to normalize the height differences with respect to the target object. This ensures that the range of the relative heights is normalized to 1. As shown in Eq. (6).

$$z^* = \frac{z}{\max_{1 \le i \le N}(z_i) - \min_{1 \le i \le N}(z_i)} \tag{6}$$

Calculate and Scale the Distance:
Finally, for each anchor object, we calculate the distance from the target object on the xy-plane and scale it, so that the anchor object with maximum distance $\max_{1 \le i \le N} \|d_{xy,i}\|_2$ is normalized to 1. As shown in Eq. (7).

$$d^*_{xy} = \frac{d_{xy}}{\max_{1 \le i \le N}(d_{xy,i})} \tag{7}$$

D. Target Category Score Extraction

In our 3D visual grounding model, accurately identifying the target category scores stands as a fundamental component. Importantly, the model is designed to ensure that the object scores are updated independently of other model aspects, including the spatial module and the fusion module, during backpropagation. This focused approach

in training is essential to maintain the precision and reliability of the model's output. By doing so, we safeguard the model's ability to discern and enhance its category recognition capabilities effectively, without interference from the other learning processes occurring simultaneously within the model's framework.

D.1. BERT [CLS] for Classification

In our model, we utilize the auxiliary loss term L_{text} along with the $[CLS]$ token representation from a pre-trained BERT model for training and identifying the target category mentioned in the textual description. This method achieves an accuracy rate of approximately 92.5% in correctly identifying the target category, highlighting the effectiveness of the $[CLS]$ token. For a predefined set of N_c object categories, our classifier, which comprises MLP layers, outputs logits for these N_c categories, denoted as L_{cls}, as defined in Eq. (8).

$$L_{cls} = \{L_i \mid L_i \in \mathbb{R},\ i = 1, 2, \ldots, N_c\} \tag{8}$$

Here, L_{cls} represents a set of scores, where each element L_i denotes the score for the i-th category. These scores reflect the model's assessment of how well each object matches a category based on the textual description.

D.2. Determination of Target Category

The index of the target category is determined using Eq. (9). This procedure identifies the index with the maximum logit value in L_{cls}, signifying the category most likely to be the target as per the textual description. This identified index is then utilized to extract the corresponding target category score for each object in the scene.

$$\mathcal{I} = \text{argmax}(L_{cls}) \tag{9}$$

D.3. Visual Data Processing

In a 3D space with N objects, the object classifier CLF utilizes the scene-aware object encoder's output O^{sa} to calculate a score matrix $\mathbf{C} \in \mathbb{R}^{N \times N_c}$, as defined in Eq. (10). Each row in the matrix, $\mathbf{C}_j$, corresponds to N_c category scores for the j-th object. The classifier CLF employs a multi-layer perceptron (MLP) to perform this task.

$$C_j = \text{CLF}(O^{sa}_j) \tag{10}$$

D.4. Extraction of Category Scores for Objects

For each object in the visual scene, the target category score is determined by the corresponding score from the object classifier's output, indexed by the target category index. This is represented as Eq. (11)

$$\mathcal{S} = \{\mathcal{S}_j \mid \mathcal{S}_j = C_{j,\mathcal{I}},\ j = 1, 2, \ldots, N\} \tag{11}$$

In this expression, $\mathcal{S}$ denotes the set of target category scores, with each $\mathcal{S}_j$ being the score for the j-th object, as determined by the target category index $\mathcal{I}$ and the object classifier's output $C_{j,\mathcal{I}}$. This method assigns a score to each object, indicating its match level with the identified target category.

E. Hyperparameter Selection

Selecting the right hyperparameters is crucial in optimizing the performance of our models. In this section, we discuss the hyperparameter selection process for two critical components of our model: Multi-Modal Predictions Fusion and the Loss Function. We explore how varying hyperparameters influence the fusion of different data modalities and the effectiveness of our loss function. This careful calibration is aimed at achieving the best balance between these elements, thereby enhancing the model's overall accuracy and robustness.

E.1. Multi-Modal Predictions Fusion

In Section 3.5, we introduce a fusion method delineated in Eq. (7). These hyperparameters are crucial for the optimization of our model. Fig. 1 demonstrates the model's accuracy across 100 epochs for different configurations of λ and μ, where λ is the weight of the spatial score and μ is the weight for the category score. Our empirical results suggest that the model's performance is robust for parameter settings where λ and μ are not significantly disparate. A noteworthy observation is that settings with moderately high but comparable values of both λ and μ manage to maintain stability in accuracy. However, increasing the discrepancy between λ and μ, especially with λ as high as 10, leads to a marked decline in performance. These insights accentuate the necessity of balanced hyperparameter tuning to harness the full potential of our fusion approach effectively.

E.2. Loss Function

In Section 3.5, we introduce our loss function in Eq.(8), which incorporates the hyperparameters α, β, and γ. Through experimentation with various configurations of these hyperparameters, as shown in Fig. 2, we observed that our model exhibits considerable robustness in its performance, particularly with variations in α and γ. The results, as depicted in the accompanying figure, demonstrate that the accuracy remains relatively stable across a range of values for α and γ, indicating the model's insensitivity to these parameters. However, a noticeable decline in performance is observed when β is reduced from 1 to 0.1, suggesting that β plays a more critical role in the loss function's efficacy. This underscores the importance of β in our model's training and implies that while some hyperparameters can be adjusted

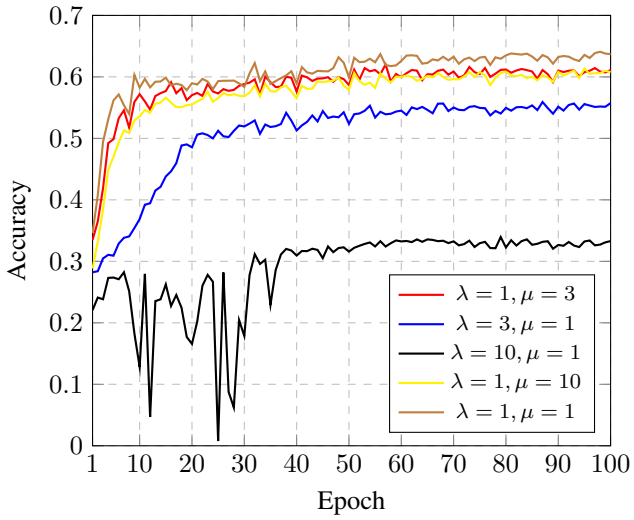


Figure 1. Model accuracy across different hyperparameter settings for λ (spatial score weight) and μ (category score weight). The model maintains robust performance when the values of λ and μ are relatively balanced.

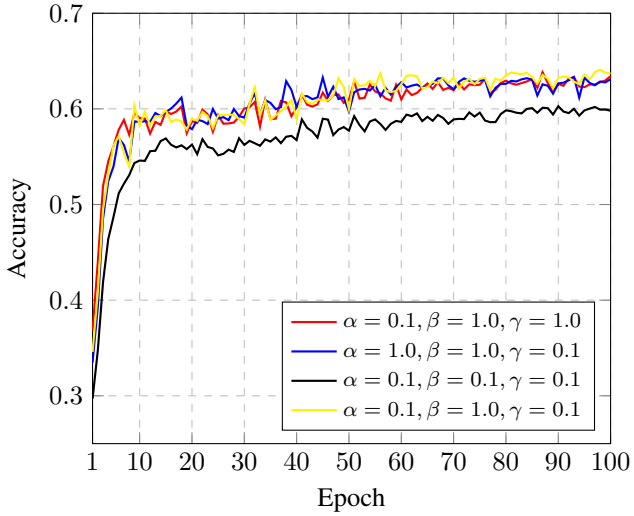


Figure 2. Model accuracy for varied hyperparameter settings of α, β, and γ. The model shows overall robustness to these hyperparameter adjustments, particularly for α and γ.

with minimal impact, others require careful calibration to maintain optimal performance.

F. Layer Ablation Study

In this section, we present an ablation study to investigate the influence of various layer configurations on model performance. To facilitate a fair comparison with established model [4], we have preserved the standard configuration of PointNet++ and the pre-trained BERT. While it remains uncertain whether a higher number of layers might yield further improvements, we have not pursued such configurations due to computational resource limitations and hardware constraints.

F.1. Scene-Aware Object Encoder

The results from Table 5, based on experiments with the Nr3D dataset, show a consistent performance of our model across various layer configurations in the scene-aware object encoder. The overall accuracy and sub-category performances like Easy, Hard, VD, and VI remain relatively steady, regardless of the number of layers. Notably, the highest accuracy is at 2 layers (64.4%), but the variance is marginal compared to other configurations. This pattern suggests that adding more layers, up to four, doesn't significantly improve the model's performance. Such stability in results across different layer numbers highlights the robustness of our model, suggesting it can be effective without extensive layer adjustments.

Layers	Overall1	Easy	Hard	VD	VI
1	63.6%	68.8%	58.6%	63.3%	63.8%
2	**64.4%**	**69.7%**	**59.4%**	**65.4%**	64.0%
3	64.1%	69.4%	59.0%	64.0%	**64.2%**
4	62.6%	68.1%	57.3%	62.3%	62.7%

Table 5. Comparison of result accuracy for different numbers of layers in the scene-aware object encoder. All experiments were conducted on the Nr3D dataset.

F.2. Fusion Module

In Table 6, using data from the Nr3D dataset, we see how different numbers of layers in the fusion module affect our model's performance. With just 1 layer, the model's accuracy is lower, especially in the Hard and view-dependent (VD) categories. This suggests that a single layer might not be enough for complex tasks. A notable point is the big drop in accuracy with 6 layers in all categories, which also requires more training time and resources. This indicates that too many layers can harm the model, because of overfitting or unnecessary complexity. The best results are seen with 2 to 4 layers, with 3 layers performing the best in Hard and VD categories. This shows that a balance in the number of layers is key for good performance without using too many resources.

Layers	Overall1	Easy	Hard	VD	VI
1	62.4%	68.8%	56.3%	62.0%	62.6%
2	63.5%	69.8%	57.4%	62.7%	63.9%
3	**64.4%**	69.7%	**59.4%**	**65.4%**	64.0%
4	63.8%	**70.5%**	57.4%	63.2%	**64.1%**
6	58.9%	65.3%	52.7%	59.6%	58.5%

Table 6. Comparison of result accuracy for different numbers of layers in the fusion module. All experiments were conducted on the Nr3D dataset.

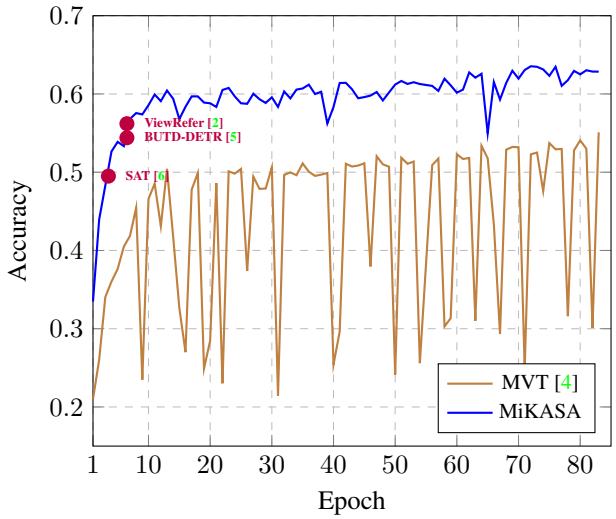


Figure 3. Comparative training curves highlighting early convergence and superior stability. Our model reaches superior performance within 7 epochs, as indicated by the labeled curves.

G. Extended Results Analysis

This section presents a comprehensive analysis of our model, examining its performance from multiple perspectives. In Sec. G.1, we not only highlight the model's rapid convergence and stable training but also compare its efficiency against standard benchmarks. This subsection emphasizes the robustness of our approach in the initial phases of learning. Moving to Sec. G.2, we delve deeper into the model's processing of different scores, providing insights into the nuanced decision-making mechanisms at play and how they contribute to overall accuracy. Sec. G.3 showcases the model's ability to adeptly handle viewpoint variations, demonstrating its flexibility and sophistication in complex visual environments. Finally, in Sec. G.4, we engage in a critical examination of instances where the model underperforms, exploring these scenarios to pinpoint limitations and propose potential avenues for future enhancements, thereby strengthening our understanding of the model's capabilities and boundaries.

G.1. Early Convergence and Superior Stability

Fig. 3 illustrates the comparative training dynamics of our model against MVT [4]. Despite sharing the same configuration in object and text encoders, our model demonstrates a notable smoothness in its training curve, indicative of enhanced stability during the learning process. This is particularly evident in the early stages of training, where our approach achieves superior performance benchmarks. Impressively, our model reaches superior performance within 7 epochs.

G.2. Interplay of Category & Spatial Scores

In examining Tab. 7, the model trained on the Nr3D dataset reveals a nuanced interplay between category and spatial scores in its decision-making mechanism. This uncovers interesting patterns and unexpected anomalies, offering insights into the complex cognitive strategies employed by the model in reference identification.

G.2.1 Correctly Predicted Target

The analysis of the Nr3D dataset reveals instances labeled as "Correctly Predicted Target", where the model demonstrates accurate referencing of targets. Intriguingly, in 97.4% of the cases, the spatial score provides the correct answer. This accuracy is maintained even in cases where the targets do not attain the highest category score, owing to the corrective influence of the spatial score. This finding illustrates the crucial role of spatial insight within the model, particularly when the clarity in categorical identification is ambiguous, and it emphasizes the harmonious interplay between category and spatial scores in achieving correct target predictions.

On the other hand, there are occurrences where the spatial score is not dominant (2.6%), but the robust category score rectifies. This reciprocity between categorical recognition and spatial localization in the model's architecture is pivotal, allowing the model to uphold its accuracy even amidst ambiguity.

Moreover, about 3.39% of cases marked as "Correctly Predicted Target" don't secure a position within the top 6 category scores. This suggests the model's capability to accurately identify targets even without a definite categorical conviction, illuminating the multifaceted and nuanced decision-making process of the model.

G.2.2 Incorrectly Predicted Target

Before exploring cases classified as "Incorrectly Predicted Target", it is important to clarify that this term refers to the scores assigned by the model to the correct ground truth targets that were not selected, rather than the scores of the incorrect objects that the model erroneously chose.

	Correctly Predicted Target						Incorrectly Predicted Target					
	NrsD			Sr3D			NrsD			Sr3D		
	Top 1	Top 3	Top 6	Top 1	Top 3	Top 6	Top 1	Top 3	Top 6	Top 1	Top 3	Top 6
Category Score	42.0%	85.9%	96.7%	48.5%	96.0%	99.1%	25.0%	60.9%	80.9%	21.4%	65.3%	80.7%
Spatial Score	97.4%	99.9%	99.9%	98.2%	99.9%	100%	3.1%	36.4%	39.4%	3.6%	42.9%	47.2%

Table 7. A summary of model performance based on category and spatial scores for the NrsD and Sr3D datasets. The table presents the proportion of correctly and incorrectly predicted targets that achieved Top 1, Top 3, and Top 6 scores in both category and spatial assessments. These results provide insights into the model's decision-making process and emphasize the interplay between categorical recognition and spatial understanding in predicting object references.

Analyzing how the actual targets scored when the model did not select them provides profound insights into the model's cognition during incorrect references. For example, 36.4% of these are within the top 3 spatial scores. This pattern suggests that, in these instances, the model tends to be proximal to the correct spatial localization but fails to distinguish the true target from other distractors effectively. This frequent proximity to correct spatial localization implies the presence of refined spatial awareness in the model; however, the inability to differentiate targets accurately in such situations points to potential areas for improvement in the spatial encoding mechanism. This examination reveals the nuanced collaboration between category and spatial scores, showcasing the model's aptitude in resolving uncertainties and ensuring accurate referencing, even when clear certainties in category or space are lacking. The insights gained spotlight the model's flexibility and sophisticated approach to 3D visual grounding. Importantly, this analysis not only enhances our understanding of the decision-making process but also guides refinements, allowing investigations into which scores may have failed, and fostering advancements in scenarios where spatial information is ambiguous, thus contributing to the model's explainability and improvement.

G.3. View-Dependent Samples

In Fig. 4, we display the capability of our model in handling view-dependent scenarios, emphasizing its precision in adapting to varying viewpoints. These examples are particularly illustrative of the effectiveness of our multi-key-anchor technique and the spatial module. By accurately recalibrating the interpretation of spatial relationships and object positions in response to changes in perspective, our model demonstrates its robustness and versatility. This adaptability, pivotal in dynamic visual environments, showcases the strengths of our approach in 3D visual grounding, validating the model's suitability for complex, real-world applications.

G.4. Analysis of Model Limitations: Failed Samples

G.4.1 Complex Scenes

Complex scenes pose a challenge due to their dense information and often ambiguous spatial details. These scenarios require the model to differentiate and accurately identify objects amidst a multitude of overlapping or closely situated elements. Figure 5a showcases examples of such complex scenes.

G.4.2 Non-Rigid Subjects: Animals & Humans

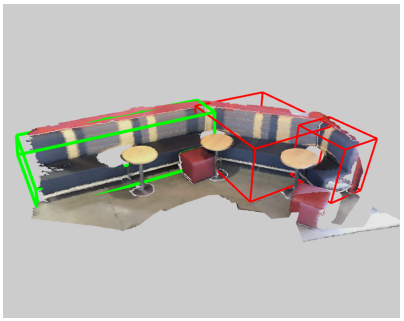
Non-rigid subjects like animals and humans present a unique challenge due to their variable shapes and postures. The model's ability to recognize and categorize these subjects is often hindered by limited training data and the inherent flexibility of these subjects. Figures 5b to 5d demonstrate instances involving these non-rigid subjects.

G.4.3 Action-Oriented Descriptions

Interpreting action-oriented descriptions such as "laying" or "sitting" is a challenge for existing models, which currently lacks the capability to contextualize how humans interact with objects in performing these actions. For example, understanding phrases like "laying on the bed" involves not just recognizing the bed but also comprehending the typical orientation of a person on it. This advanced level of contextual understanding, especially in human-object interactions, remains a developmental goal for our model. Figures 5d to 5f provide illustrations of such scenarios.

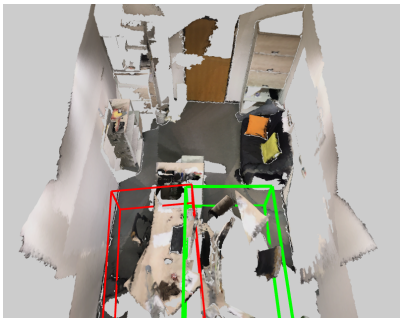
G.4.4 Symbolic and Textual Details

Processing symbolic and textual details, such as logos, signs, or written text on objects, is a complex task that our model struggles with. This limitation affects the model's ability to interpret directives involving object identification based on such details. Figures 5g to 5i provide examples of this challenge. These cases collectively underscore the importance of developing more sophisticated training strategies. Enriching the dataset and enhancing the model's capability to process complex contextual, symbolic, and textual information is key to advancing its proficiency in 3D visual grounding tasks.

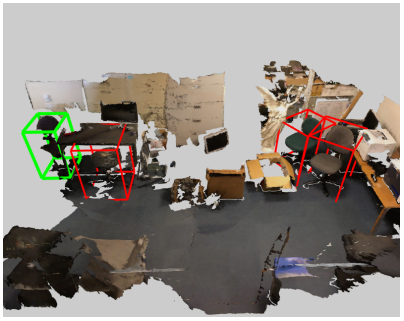


(a) “The couch along the wall and on the left hand side of the other couches, when you are facing the group of couches.”

(b) “Facing fridge, this cabinet is middle of the three upper cabinets to the right.”

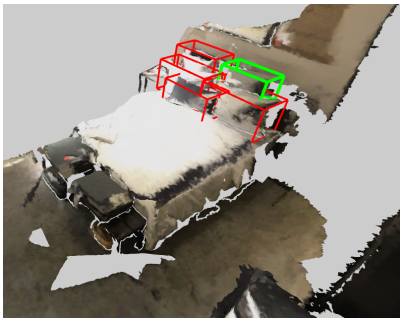


(c) “Facing the door, the desk on the right side.”



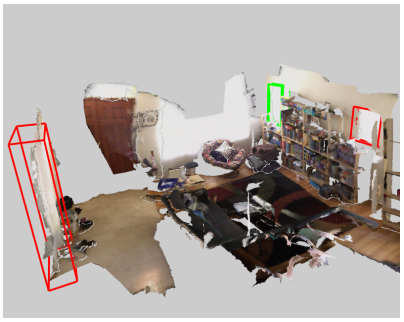
(d) “Facing the row of 4 chairs, it’s (the chair) furthest left.”

(e) “Facing the windows, chose the center desk under the clock.”

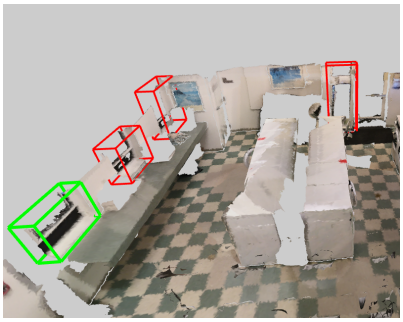


(f) “If you are facing the bed, it is the pillow in the back, on the right.”

(g) “When you are facing the white board, look at the 3 desks on your right and pick the one in the middle.”

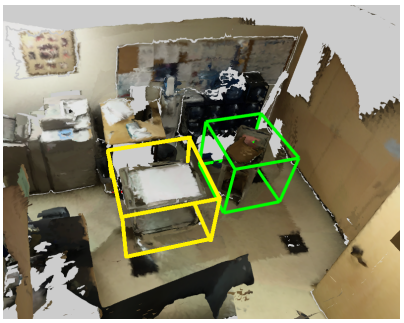


(h) “If you are facing the bookshelves, it is the window on the left.”

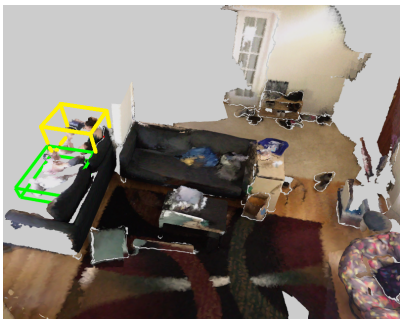


(i) “If you are facing the three windows on one wall, it is the window on the left.”

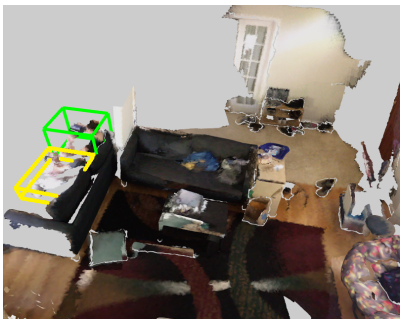
Figure 4. Examples of view-dependent scenarios. In these illustrations, the target categories are indicated with purple markings, and the objects used as anchors for locating these targets are highlighted in orange. The green bounding box refers to the correctly chosen object, and the red bounding box refers to the unchosen distractors.



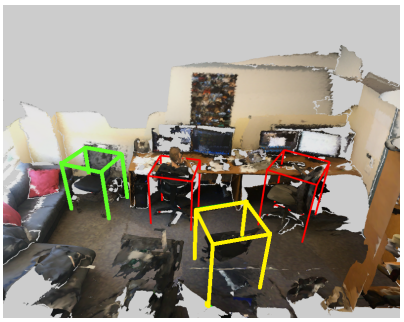
(a) “In the middle of the room is a beige cart. square, top has papers on it and bottom shelf is empty.”



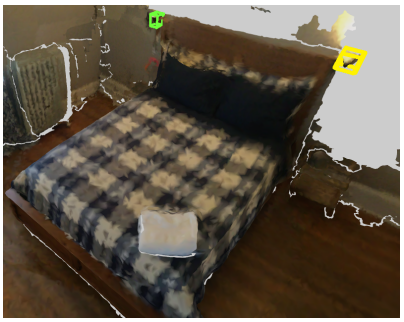
(b) “2 white dogs sitting at the person’s feet.”



(c) “The taller person sitting on the left side of the couch when you are facing it.”



(d) “The black chair facing the chair with a person in it.”



(e) “If you are laying in bed with your head by the headboard, it is the light that is just above the bed and to the left.”

(f) “There are two pillows by the headboard of the bed. If you are laying in bed, it is the pillow on the left side.”

(g) “Facing the picture with blue sky, clouds and yellow flowers, choose the computer tower furthest to the right of the room.”

(h) “The box that has a FedEx label on it.”

(i) “The trash can is near the hallway with the logo on the wall. It’s the one on the left.”

Figure 5. A collection of challenging scenarios where the model demonstrates limitations. Featured are complex scenes with ambiguous spatial details, difficulties in recognizing animals and humans due to limited training data and non-rigidity, and challenges in interpreting action-oriented descriptions and textual/symbolic details on objects. These examples highlight key areas for the model’s further development and refinement. In these illustrations, the target categories are indicated with purple markings, and the objects used as anchors for locating these targets are highlighted in orange. The yellow bounding boxes refer to the ground truth, the green bounding boxes refer to the chosen objects, and the red bounding boxes refer to the unchosen distractors.

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. 1, 2

[2] Zoey Guo, Yiwen Tang, Ray Zhang, Dong Wang, Zhigang Wang, Bin Zhao, and Xuelong Li. Viewrefer: Grasp the multi-view knowledge for 3d visual grounding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 15372–15383, October 2023. 1, 5

[3] Dailan He, Yusheng Zhao, Junyu Luo, Tianrui Hui, Shaofei Huang, Aixi Zhang, and Si Liu. TransRefer3d: Entity-and-relation aware transformer for fine-grained 3d visual grounding. In *Proceedings of the 29th ACM International Conference on Multimedia*. ACM, oct 2021. 2

[4] 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, 2, 4, 5

[5] Ayush Jain, Nikolaos Gkanatsios, Ishita Mediratta, and Katerina Fragkiadaki. Bottom up top down detection transformers for language grounding in images and point clouds, 2021. 5

[6] Zhengyuan Yang, Songyang Zhang, Liwei Wang, and Jiebo Luo. Sat: 2d semantics assisted training for 3d visual grounding. In *ICCV*, 2021. 2, 5