3D Visual Grounding with Reasoning LLM

Anonymous submission

Paper ID

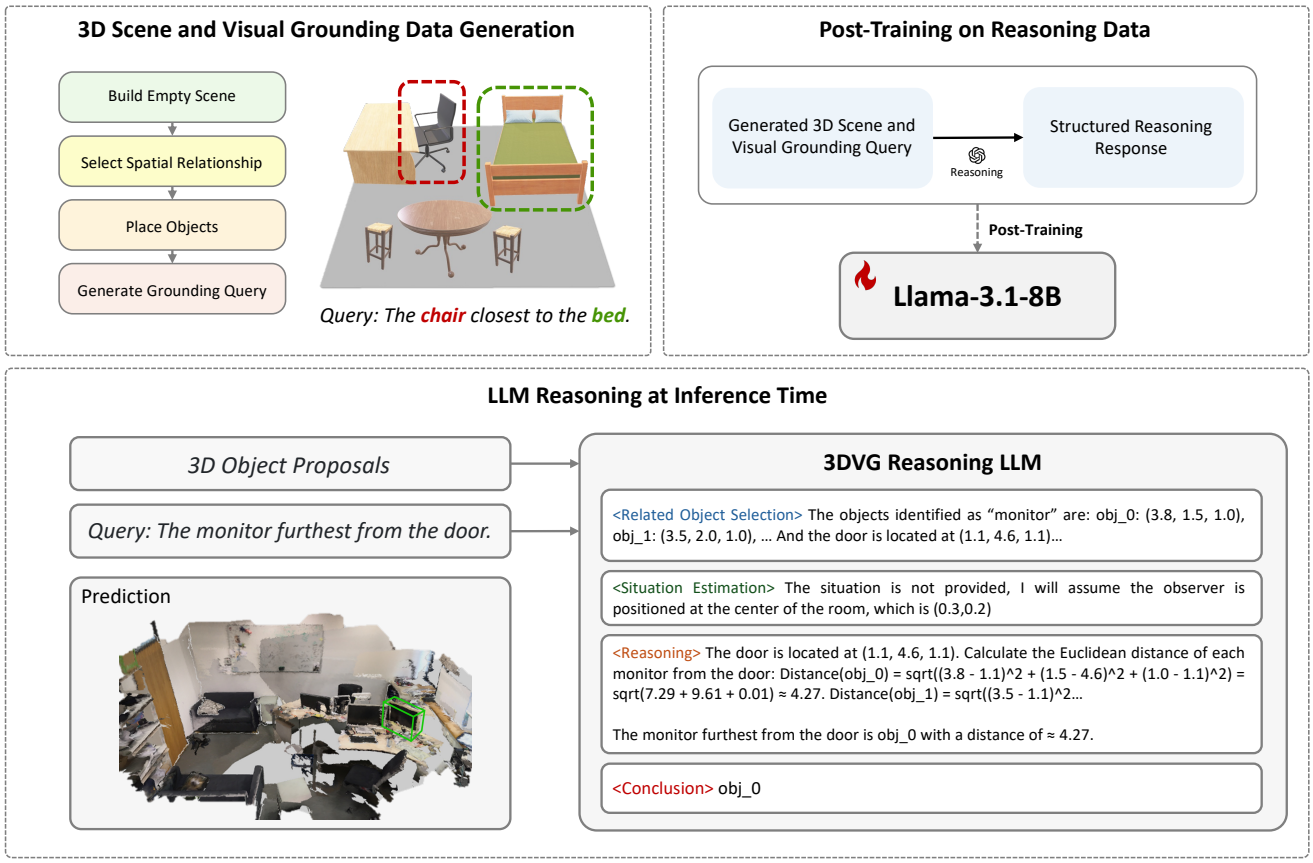


Figure 1. **Overview.** We propose a data generation pipeline that can generate visual grounding queries and reasoning responses. We use our collected data to conduct post-training, resulting in an LLM that can conduct structured reasoning in 3D visual grounding tasks.

Abstract

The recent development of Large Language Models (LLMs) with strong reasoning ability has driven research in various domains such as mathematics, coding, and scientific reasoning. Meanwhile, 3D visual grounding, as an important task in the 3D understanding area, still remains challenging due to the limited reasoning ability of recent 3D visual grounding models. Most of the current methods incorporate a text encoder and visual feature encoder to generate cross-modal fuse features and predict the referring object. These models often require supervised training on extensive 3D annotation data. Meanwhile, recent zero-shot visual grounding methods require access to proprietary LLM during test time, which leads to large inference costs. To overcome these limitations, we propose a 3D visual grounding data generation pipeline capable of synthesizing 3D scenes along with corresponding target queries and ground truth answers for training visual grounding models. Additionally, we leverage the generated visual grounding data and conduct post-training on Llama-3.1-8B-Instruct, resulting in a strong 3D visual grounding LLM that outperforms existing SoTA zero-shot methods, demonstrating its effectiveness.

1. Introduction

3D Visual Grounding (3DVG) is a crucial task in 3D understanding, aiming to identify a target object within a 3D scene based on a given textual query. Recent models [19, 26, 31, 32, 34, 41] typically integrate a text encoder and a visual encoder to generate cross-modal features for target object prediction. While these methods achieve strong performance, they rely on large-scale annotated 3D visual grounding datasets for supervised training. To mitigate this, recent research has explored zero-shot approaches, leveraging proprietary LLMs like GPT-4 [3] as agents to generate grounding predictions using visual grounding tools [36] or in-context examples and programs [40]. However, these approaches heavily depend on proprietary LLMs, increasing inference costs. In this work, we aim to explore using an 8B LLM to solve the challenging visual grounding task under zero-shot setting, by leveraging LLM's reasoning ability, we hope to provide a solution with strong visual grounding performance with existing zero-shot methods while being more cost-effective.

Recently, LLMs with reasoning ability [12, 18, 27, 35] have achieved great success in various domains including mathematics, coding, and scientific reasoning. DeepSeek-R1 [12] leverages reinforcement learning to achieve strong thinking ability and SoTA performance on various challenging benchmarks. However, despite the success of these reasoning models, they mainly focus on the text and image understanding domain, while solving the 3D visual grounding or understanding tasks by leveraging LLM's strong reasoning ability remains under-explored, with most of the current SoTA 3D visual grounding methods [19, 31] incorporates a text encoder to encode the given query and conduct direct prediction based on the cross-modal fused feature from various input modalities. In the 3D visual grounding task, the given query can contain sophisticated spatial relationships, and leveraging LLM to comprehend the input query is another potential solution that is not fully explored.

In this work, we aim to solve the 3D visual grounding task using a reasoning approach, we present an overview of our work in Fig. 1. Given the query of the target object in a 3D scene, we developed an LLM that can conduct a rigorous reasoning process before making the final prediction. To train such LLM, we design a 3D visual grounding reasoning data collection pipeline, that aims to address the current methods' dependency on 3D visual grounding data. We use our collected data and conduct post-training on the base model Llama-3.1-8B-Instruct [8], resulting in a 3D visual grounding LLM with strong reasoning ability, which achieves higher performance than existing SoTA zero-shot method on multiple 3D visual grounding benchmarks. Moreover, we discover the spatial hallucination phenomenon of proprietary LLM on spatial understanding tasks during our data collection process, we further design a prompt that can boost the accuracy of the response from LLM by a large margin in our data collection pipeline. We summarize our contribution as follows:

- We propose a novel 3D visual grounding reasoning data collection pipeline, which serves as a post-training dataset for our reasoning LLM on zero-shot 3D visual grounding.
- We discover the spatial hallucination of current SoTA LLM and design an effective prompt to reduce the spatial hallucination, enabling more accurate reasoning and response during data collection.
- We conduct post-training on our collected data and introduce **3DVG-LLM**, an LLM for 3D visual grounding task that achieves SoTA performance on 3D visual grounding benchmarks including NR3D and SR3D.
- To our best knowledge, this is the first successful attempt using synthetic 3D data to train a 3D understanding LLM with reasoning ability, we hope this work serves as a cornerstone for future 3D-LLM development with stronger reasoning and spatial understanding ability.

2. Related Work

2.1. 3D Visual Grounding

3D visual grounding is an important 3D understanding task that aims to predict the target object in the 3D scene based on a given language query of the target. Most of the recent 3D visual grounding methods [16, 19, 21, 34] adopt separate encoders for each input modality such as text and point cloud to generate cross-modal features and conduct prediction. Some end-to-end methods like SAT [38], LanguageRefer [29], and UniT3D [6] utilized a unified multimodal transformer to conduct 3D visual grounding. Recently, COT3DREF [1] and ReGround3D [42] proposed to leverage chain-of-thoughts to retrieve objects related to a given query, however, they did not leverage the language models to conduct reasoning for better spatial understanding. Furthermore, these methods utilize supervised training, which requires extensive annotations on the 3D data. Besides the supervised method, recent works also focus on the zero-shot approaches [36, 40]. These methods leverage LLM to generate code and solve the 3D visual grounding task, which achieves some success in zero-shot 3D visual grounding. However, these methods rely on in-context examples and proprietary LLMs [3, 24] during test time, which introduce more inference costs. In this work, we explore using an 8B open-source LLM to solve the 3D visual grounding task through inference time reasoning under zero-shot setting. Our method does not require proprietary LLM nor 3D data that requires extensive annotation, while achieving better performance compared with the SoTA zero-shot methods, showcasing the potential of open-source LLM for zero-shot 3D visual grounding, and paving the way for a more cost-effective solution in the field.

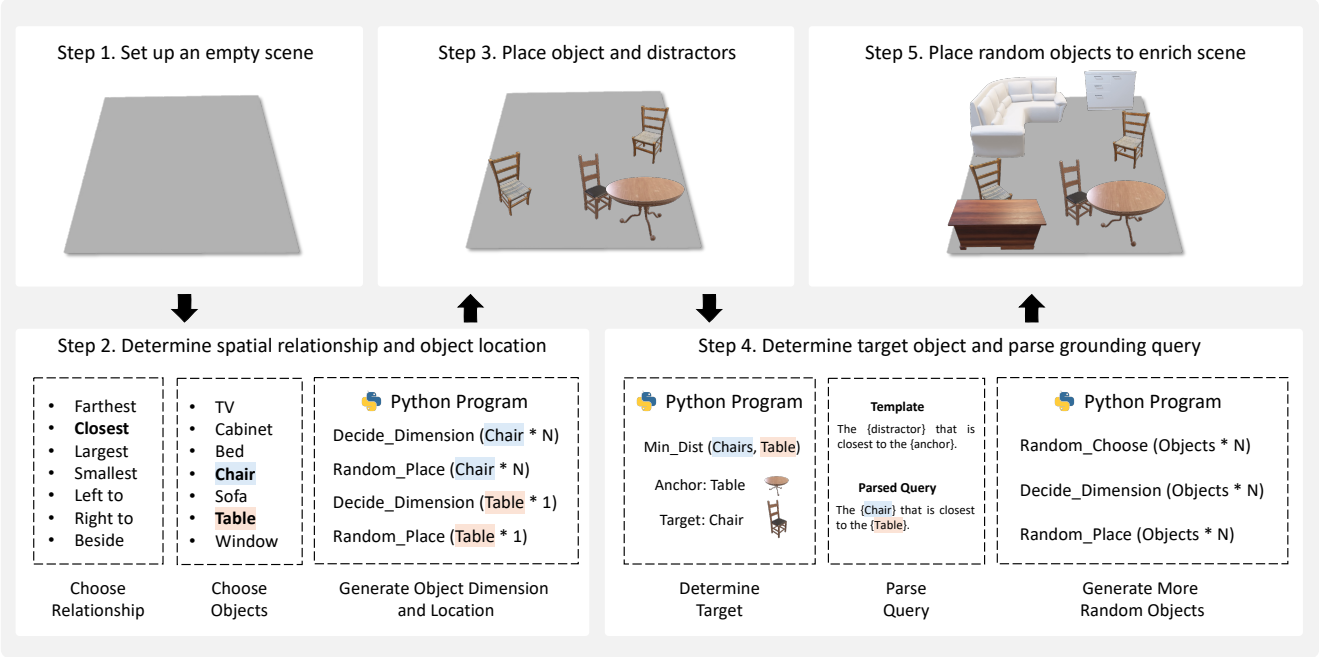


Figure 2. An illustration of our 3D scene layout generation pipeline, which contains 5 steps: **1)** Set up an empty 3D scene with certain dimension. **2)** Choose a spatial relationship and decide the anchor and distractor objects as well as their dimension and location. **3)** Place them in the 3D scene. **4)** Determine the target object from the distractors and parse query. **5)** Generate more random objects to enrich scene diversity. Note that some spatial relationships do not have anchor object and only have distractors, such as Largest/Smallest. Note that the furniture in the figure are for visualization purpose only, our data pipeline does not include generating the appearance of the furniture. Please refer to our attached scene generation code for more technical detail on the implementations.

2.2. LLM with Reasoning Ability

Recent advancements in LLMs have significantly enhanced their reasoning capabilities, enabling them to tackle complex problems across various domains such as mathematics, coding, and scientific reasoning. Models like DeepSeek-R1 [12] incorporate reinforcement learning during training and achieve SoTA performance across multiple benchmarks. Similarly, LLaVA-CoT [35] and VisualCoT [30] introduce explicit reasoning processes to improve VLM on visual question-answering task. Prism [27] proposed to decouple the perception and reasoning stages to enhance VLM performance. While these models have demonstrated remarkable success in text and image understanding, leveraging reasoning LLMs for zero-shot 3D understanding remains largely unexplored, with most existing 3D visual grounding methods still require supervised training from the extensive annotated 3D data. To address this gap, we present a pioneering work that integrates reasoning LLM to tackle the challenging 3D visual grounding task. We hope this work serves as a strong baseline for reasoning-enabled LLMs in the 3D scene understanding domain and inspires further research at the intersection of reasoning-driven LLMs and 3D scene comprehension.

3. 3D Visual Grounding Data Collection

3.1. Overview

In this work, we design a novel 3DVG data collection framework for 3D reasoning visual grounding LLM training. In this section, we introduce our program-based 3D scene generation pipeline in Sec. 3.2, reasoning stages in Sec. 3.3, and the reasoning data collection in Sec. 3.4.

3.2. 3D Scene Generation

The collection of 3D visual grounding data requires a huge amount of effort including extensive human annotations and indoor 3D RGB-D scan collection. Unlike previous supervised methods that leverage human-annotated 3D data, we propose to generate synthetic 3D scene data for model training and zero-shot 3D visual grounding on real-world 3D scenes. There are many existing methods for 3D scene generation [10, 22, 37], however, they mostly focus on generating scene layout with realistic object arrangements or following the specified room style from users' generation prompt, while our goal is to collect object-centric 3D scene that focuses on objects' spatial arrangements as well as the target object's corresponding grounding query, which is an important component of 3D visual grounding data.

In this work, we design a program-based 3D scene layout generation pipeline that can generate an object-centric 3D scene following several common spatial relationships between objects. These common spatial relationships include: closest/farthest, largest/smallest, left/right, and beside. These common spatial relationships and corresponding scene layouts can be generated by our code program. Here, we provide a generation example of a 3D scene layout in Fig. 2, which contains the 3D scene layout following query *"The Chair that is closest to the Table."* with n distractor chairs in the scene:

1. Define a 3D scene size with a certain dimension $H \times W$.
2. Randomly generate n chairs and one table in the 3D scene.
3. Calculate the Euclidean distance between each chair and table to find the closest chair to the table; this chair will be the target object of the 3D visual grounding query.
4. Generate more objects (but not chairs or tables) to further enrich the 3D scene diversity.

Following this procedure, we collect a wide range of 3D scene layouts that follow our specified common spatial relationship and are not confined by the limited number of 3D scene layouts in the real-world available 3D scans. For each spatial relationship, we design multiple templates from which we can parse the target's object class to generate the final target object query. We prompt `GPT-4o` [17] to generate some indoor scene objects as well as their common dimension. When placing objects, we randomly select object classes and apply a random variation on their size to enhance diversity. We attached our 3D scene layout generation code in the appendix for reference.

Note that because our base model LLM already has very strong reasoning ability, we do not focus on designing complicated spatial relationships to strengthen its reasoning and spatial understanding ability, but only use several common relationships with reasoning data to encourage the LLM to think step-by-step. Our experiment results demonstrate that our LLM can still adapt to a 3D visual grounding query that has an unseen spatial relationship after post-training on our collected data. Furthermore, because our goal is to generate 3D scene data that follow our specified spatial relationship between objects, during the random object placements, we only refrain the object from being placed overlapping and do not focus on placing the object in a way that is realistic in real-world scenarios. Given our goal is to guide LLM's reasoning process, we believe the rationality of object arrangement will not affect LLM's reasoning process.

3.3. 3D Visual Grounding with Reasoning Stages

Recent advanced LLMs with reasoning ability adopt special tokens or tags [12, 35] to separate the reasoning and conclusion stage, which enables a more structured and organized thinking process of the LLM. In the 3D visual grounding domain, we believe it is also important to think step-by-step before generating the final prediction given the difficulty of interpreting the spatial relationship of the target query. Here, we proposed four different reasoning stages, which encourage the LLM to generate a more structured thinking process. The four reasoning stages include:

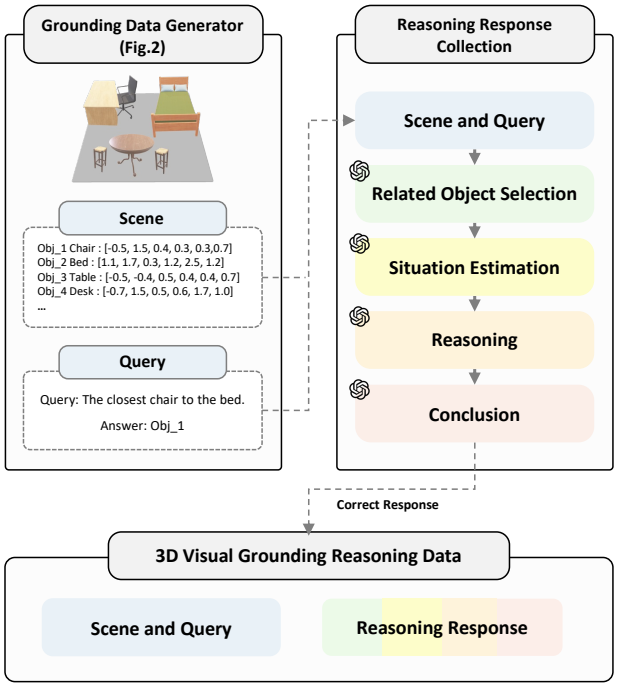


Figure 3. An overview of the reasoning data collection pipeline, after the grounding data generator produce the 3D scene and query, we use `GPT-4o` to generate structured reasoning response. Only the response with correct answer and format will be stored in our 3D visual grounding and reasoning data for later training.

- **Related Object Selection.** Given all the objects in the 3D scene, the LLM listed the objects that might be relevant to the given query. Each listed object will include information such as its object ID, class name, location, and dimension.
- **Situation Estimation.** If a 3D situation (viewer's location) is provided, the LLM will estimate its possible coordinate in the 3D scene. Otherwise, the LLM will just assume it is standing in the middle of the 3D scene (which is usually the case).
- **Reasoning.** This is the main stage for LLM to conduct reasoning based on the previously given information. The LLM will conduct necessary math calculations and logical reasoning to derive the final answer.
- **Conclusion.** In this stage, the model outputs the final prediction in a structured format, which enables easier parsing and understanding for the user.

Input Prompt to GPT-4o

I have a 3D scene represents in a dictionary, each key is their object ID and the value contains the object name and 3D bounding box in center x, center y, center z, width, height, length. I will provide you a caption of an object which you will need to return the object ID of this object. I need you to strictly follow the format with four specific sections: SELECTION, SITUATION, REASONING, and CONCLUSION. It is crucial that you adhere to this structure exactly as outlined and that the final answer in the CONCLUSION matches the provided format precisely.
...
In SITUATION, if the situation information is not provided, if the situation information is not provided, assume your current location is in the middle of the floor. In REASONING, outline a step-by-step thought process you would use to solve the problem based on the 3D scene.
If you need to determine which object between obj_A and obj_B is on the left or right side, please calculate the cross product between your position O to obj_A vector OA and your position O to obj_B vector OB, if the cross product C between OA and OB is positve then obj_A is right and obj_B is left, if the cross product C is negative, obj_A is left and obj_B is right. Please put the smaller obj_ID as A and larger obj_ID as B.
...
3D scene: {"obj_0":... "obj_8": {"object_name": "**cabinet**", "box": [5.92, -4.87, 0.94, 0.83, 0.65, 1.89]}..."obj_20": {"object_name": " **cabinet** ", "box": [-2.88, -4.8, 1.0, 0.81, 0.72, 2.01]},...}

Caption: "The cabinet located **on the left side**."

GPT-4o response

<SELECTION>
The objects identified as cabinet in the scene are:
- **obj_8**: (5.92, -4.87, 0.94), dimensions (0.83, 0.65, 1.89)
- **obj_20**: (-2.88, -4.8, 1.0), dimensions (0.81, 0.72, 2.01)
</SELECTION>

<SITUATION>
Assuming I am in the middle of the floor, which is approximately at (1.09, 4.65). I am facing the cabinets. The average coordinate of the cabinets is:

\left(\frac{5.92 + (-2.88)}{2}, \frac{-4.87 + (-4.8)}{2}\right) = \left(\frac{3.04}{2}, \frac{-9.67}{2}\right) = (1.52, -4.835)

Since I am at (1.09, 4.65) and facing towards (1.52, -4.835), I am primarily facing in the **negative y-direction**.
</SITUATION>

<REASONING>
1. The caption states **"The cabinet located on the left side."**
2. Since I am facing in the negative y-direction, my **left side** corresponds to the **negative x-direction**.
3. The cabinet at **obj_8** is at (5.92, -4.87), while the cabinet at **obj_20** is at (-2.88, -4.8).
4. Comparing their x-coordinates, **obj_20 (-2.88, -4.8)** is more to the left than **obj_8 (5.92, -4.87)**
5. Therefore, the correct cabinet according to the caption is **obj_20**
</REASONING>

<CONCLUSION>
obj_20
</CONCLUSION>

GPT-4o response with Extra Prompt

<SELECTION>
The objects identified as cabinet in the scene are:
- **obj_8**: Located at (5.92, -4.87, 0.94) with dimensions (0.83, 0.65, 1.89).
- **obj_20**: Located at (-2.88, -4.8, 1.0) with dimensions (0.81, 0.72, 2.01).
</SELECTION>

<SITUATION>
The floor is given as **obj_40** with center coordinates at (1.09, 4.65). Thus, assume the current location is at **(1.09, 4.65)**.
</SITUATION>

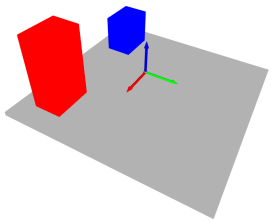
<REASONING>
Define vectors from our assumed position **O (1.09, 4.65)** to each cabinet:
OA (obj_8) = (5.92 - 1.09, -4.87 - 4.65) = (4.83, -9.52)
OB (obj_20) = (-2.88 - 1.09, -4.8 - 4.65) = (-3.97, -9.45)

Compute the cross product:
C = (4.83 \times -9.45) - (-3.97 \times -9.52)
C = -45.6435 - 37.7744
C = -83.4179

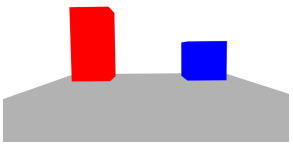
Since **C is negative**, **obj_8 is to the left** and **obj_20 is to the right**. The caption specifies "The cabinet located on the left side," so the correct object is **obj_8**.
</REASONING>

<CONCLUSION>
obj_8
</CONCLUSION>

Scene Visualization



Object_8 and Object_20 in the 3D scene.
x,y,z axis in Cartesian coordinate system correspond to Red, Green, Blue.



Egocentric View

Figure 4. A comparison of GPT-4o response with and without extra prompt. We highlight extra prompt in yellow. The correct reasoning from GPT-4o in green, and the error reasoning in red. The accuracy gain with extra prompt can be found in ablation study.

3.4. Reasoning Data Collection

To collect detailed reasoning data aligned with our four reasoning stages, we leverage the SoTA proprietary LLM GPT-4o [17] to generate structured reasoning responses, as illustrated in Fig. 3. We design a specialized prompt to guide GPT-4o in adhering to our four-stage reasoning process. The full prompt is provided in the appendix. Approximately 10% of responses that deviate from the specified format or contain incorrect predictions are automatically filtered. The remaining detailed visual grounding reasoning responses are incorporated into our final training data, which consists of 3.2K samples. Each sample includes a unique 3D scene layout with over 50 objects, a target object's language query, its corresponding object ID, and the reasoning response from GPT-4o. Note that since the primary goal of post-training is to enhance the LLM's reasoning capabilities, we only utilize a small-scale dataset for data efficient post-training, and leave the scale-up training using our data collection framework as a future direction.

3.5. Reducing The Spatial Hallucination

During the reasoning data collection, we found that despite the strong reasoning ability of the SoTA language model like GPT-4o, it still suffers from spatial hallucination and generates wrong predictions sometimes, especially when the given query is related to the left/right spatial relationship. This is likely caused by LLM's weak spatial understanding ability, leading to errors in recognizing left/right direction. For example, we found that GPT-4o often erroneously thinks *"when facing the negative Y direction, the left-hand side is negative X."* This leads to a lower accuracy when predicting target object related to left/right spatial relationship. To address this issue and ensure a more efficient data collection process, we modify the input prompt and encourage GPT-4o to determine the left/right object relationship by calculating the cross product between the location coordinate vectors toward two objects, in other words, we prompt GPT-4o to calculate the cross product between viewer's coordinate (which is estimated in the Situation Estimation Stage) and two objects' coordinate. Based on the cross product's value, LLM can determine two objects' left/right spatial relationship. We make a comparison of the GPT-4o response with and without our extra prompt in Fig. 4, and also conduct analysis in our ablation study and found out the extra prompt largely boosted the response accuracy, suggesting the effectiveness of our designed prompt.

Table 1. Performance comparison with existing methods on ScanRefer validation set. We follow previous work [40] and report the grounding accuracy under different IoU threshold, and also report the accuracy of *"Unique"* (scenes with a single target object) and *"Multiple"* (scenes with distractors of the same class) subsets, along with overall performance.

Method	Source	Supervision	LLM	Unique		Multiple		Overall	
				Acc@0.25	Acc@0.5	Acc@0.25	Acc@0.5	Acc@0.25	Acc@0.5
ScanRefer [5]	ECCV'20	Fully	-	67.6	46.2	32.1	21.3	39.0	26.1
InstanceRefer [39]	ICCV'21	Fully	-	77.5	66.8	31.3	24.8	40.2	32.9
3DVG-Transformer [41]	ICCV'21	Fully	-	77.2	58.5	38.4	28.7	45.9	34.5
BUTD-DETR [19]	ECCV'22	Fully	-	84.2	66.3	46.6	35.1	52.2	39.8
EDA [34]	CVPR'23	Fully	-	85.8	68.6	49.1	37.6	54.6	42.3
3D-VisTA [43]	ICCV'23	Fully	-	81.6	75.1	43.7	39.1	50.6	45.8
G3-LQ [32]	CVPR'24	Fully	-	88.6	73.3	50.2	39.7	56.0	44.7
MCLN [26]	ECCV'24	Fully	-	86.9	72.7	52.0	40.8	57.2	45.7
ConcreteNet [31]	ECCV'24	Fully	-	86.4	82.1	42.4	38.4	50.6	46.5
WS-3DVG [33]	ICCV'23	Weakly	-	-	-	-	-	27.4	22.0
LERF [20]	ICCV'23	Zero-Shot	CLIP [28]	-	-	-	-	4.8	0.9
OpenScene [25]	CVPR'23	Zero-Shot	CLIP [28]	20.1	13.1	11.1	4.4	13.2	6.5
LLM-Grounder [36]	ICRA'24	Zero-Shot	GPT-3.5 [24]	-	-	-	-	14.3	4.7
LLM-Grounder [36]	ICRA'24	Zero-Shot	GPT-4 [23]	-	-	-	-	17.1	5.3
ZSVG3D [40]	CVPR'24	Zero-Shot	GPT-4 [23]	63.8	58.4	27.7	24.6	36.4	32.7
3DVG-LLM	**Ours**	Zero-Shot	Llama-3.1-8B [8]	76.6	69.5	31.1	27.3	38.7	34.4

4. Experiments

4.1. Implementation Details

We use Llama-3.1-8B-Instruct [8] as our base model. We conduct post-training on Llama-3.1-8B-Instruct using our collected 3D visual grounding reasoning dataset, which consists of 3.2K 3D scenes, target queries, and structured reasoning processes. More details on training and implementation setup can be found in the appendix.

4.2. Benchmarks

We adopt two large-scale 3D visual grounding benchmarks in our experiments, including ScanRefer [5] and NR3D [4]. Both benchmarks leveraged 3D scenes collected from ScanNet [7]. We follow the previous zero-shot method ZSVG3D [40] and utilize the same object proposal generation procedure to obtain the 3D bounding boxes and their class label for consistent evaluation and fair comparison.

ScanRefer. ScanRefer provides 51,500 natural language descriptions from 800 different 3D scenes. We follow previous works [19, 36, 40] and adopt the validation set of ScanRefer for evaluation, which contains 9,508 language queries. We report ScanRefer's standard evaluation metrics: Accuracy@0.25 and Accuracy@0.5, where 0.25 and 0.5 are different IoU thresholds of 3D bounding boxes.

NR3D. NR3D contains 41.5K natural, free-form language captions collected from humans through a referring game, featuring multiple spatial relationships. Unlike ScanRefer, NR3D provides the GT 3D bounding boxes in the scene along with their corresponding object IDs. Given a language query, the visual grounding model is required to predict the target's object ID that the given query refers to. We adhere to the original NR3D benchmark's setup, categorizing text queries into "Easy" (scenarios with only one same-class distractor) and "Hard" (scenarios with multiple same-class distractors). Additionally, queries are also classified as "View-Dependent" or "View-Independent," depending on whether specific viewpoints are necessary for correctly identifying the target object.

4.3. Performance

We compare the performance of our **3DVG-LLM** with various recent and SoTA visual grounding methods on ScanRefer in Table. 1. And the performance comparison on NR3D in Table. 2. We categorize the existing methods into fully supervised, weakly supervised, and zero-shot depending on their training strategies. Note that we categorize our method as zero-shot because our model is not trained on ScanRefer, NR3D, or any other real-world 3D visual grounding data.

On ScanRefer, our model outperforms the SoTA zero-shot method ZSVG3D [40], while using a smaller open-source LLM Llama-3.1-8B. Notably, unlike ZSVG3D, which requires multiple in-context examples during inference, our model conducts reasoning and prediction without any in-context example at test time. However, a performance gap remains between our model and SoTA fully supervised methods. We attribute this gap to the advantages of SoTA supervised methods, which either leverage RGB visual features [19] or benefit from supervised training to generate more accurate object proposals [26, 34]—both of which are critical for optimal grounding performance.

On NR3D, our model also outperforms the zero-shot SoTA method ZSVG3D. Moreover, we also tested the per-

Table 2. Performance on NR3D benchmark. Queries are labeled as *"Easy"* (one distractor) or *"Hard"* (multiple distractors), and as *"View-Dependent"* or *"View-Independent"* based on viewpoint requirements for grounding. † denotes the use of the oracle class label. We utilize oracle class label to estimate the upper bound performance of our model.

Method	Source	Supervision	LLM	Easy	Hard	Dep.	Indep.	Overall
ReferIt3DNet [4]	ECCV'20	Fully	-	43.6	27.9	32.5	37.1	35.6
TGNN [15]	AAAI'21	Fully	-	44.2	30.6	35.8	38.0	37.3
3DRefTransformer [2]	WACV'22	Fully	-	46.4	32.0	34.7	41.2	39.0
InstanceRefer [39]	ICCV'21	Fully	-	46.0	31.8	34.5	41.9	38.8
FFL-3DOG [9]	ICCV'21	Fully	-	48.2	35.0	37.1	44.7	41.7
LanguageRefer [29]	CoRL'22	Fully	-	51.0	36.6	41.7	45.0	43.9
3DVG-Transformer [41]	ICCV'21	Fully	-	48.5	34.8	34.8	43.7	40.8
TransRefer3D [13]	MM'21	Fully	-	48.5	36.0	36.5	44.9	42.1
BUTD-DETR [19]	ECCV'22	Fully	-	60.7	48.4	46.0	58.0	54.6
WS-3DVG [33]	ICCV'23	Weakly	-	27.3	18.0	21.6	22.9	22.5
ZSVG3D [40]	CVPR'24	Zero-Shot	GPT-4 [3]	46.5	31.7	36.8	40.0	39.0
3DVG-LLM	Ours	Zero-Shot	Llama-3.1-8B [8]	50.4	31.0	37.3	42.0	40.4
3DVG-LLM†	Ours	Zero-Shot	Llama-3.1-8B [8]	63.0	36.7	40.1	54.5	49.3

Table 3. Ablation study on the performance after post-training. We report the accuracy on NR3D of (1) base model Llama-3.1-8B, (2) base model after post-training on our collected data.

Post-training	Easy	Hard	Dep.	Indep.	Overall
	44.6	22.7	31.1	34.5	33.3
✓	63.0	36.7	40.1	54.5	49.3

Table 4. Performance comparison on NR3D when training the base model with and without reasoning process.

Reasoning	Easy	Hard	Dep.	Indep.	Overall
	42.2	25.4	30.2	35.4	33.5
✓	63.0	36.7	40.1	54.5	49.3

Table 5. Experiments on extra prompt's effectiveness on reducing the spatial hallucination of `GPT-4o`. We report the accuracy on 50 left relationship queries and 50 right relationship queries.

Extra Prompt	Left Query Acc	Right Query Acc
	40.0	52.0
✓	82.0	90.0

formance of our model using the oracle class label to estimate the upper bound performance. The increased accuracy after incorporating the oracle class label suggests that our performance is mainly limited by the quality of object proposal, which can be improved by using a better detector.

4.4. Ablation Studies

Performance under different training settings. To verify the effectiveness of post-training on our collected data, we evaluate our model and base model's performance on NR3D in Table. 3. After post-training on our collected data, the model's performance increased by a notable margin across all categories in NR3D, showcasing that conducting post-training on our data can effectively enhance the model's performance on 3D visual grounding task.

Importance of Reasoning. We also train the base model on our collected data without the reasoning process. In this training setting, the model directly generates the final prediction without the reasoning process. The resulting model is similar to the existing 3D-LLMs [11, 14], which leans to generate short and concise answers due to the characteristics of the existing 3D understanding training data. Table. 4 showcases that involving the reasoning process during training can largely improves the performance, which demonstrates the importance of our collected data.

Data collection prompt. To verify the effectiveness of our proposed extra prompt on reducing the spatial hallucination of `GPT-4o` during the data collection process, we conduct ablation study on a subset of our collected data, which consists of 100 visual grounding queries sampled from left and right spatial relationship (50 queries each). In Table. 5, `GPT-4o` achieves a large accuracy boost on both types of relationship queries with our extra prompt, showcasing the effectiveness of our prompt design in reducing spatial hallucination in our data collection framework.

4.5. Qualitative Results

We provide some qualitative results in Fig. 5. We include an in-domain query with similar spatial relationships in our

When facing the desks choose the one on the right. (In-domain)

To determine the object ID based on the given description, we need to identify the desks and then choose the one on the right.

The desks are obj_10 and obj_11. Since obj_11 is on the right of obj_10, the object ID is obj_11.

Answer: obj_11. *Base Model*

The objects that are desks are "obj_10" and "obj_11". Their coordinates are:
- obj_10: 2.8, 4.1, 0.8
- obj_11: 1.3, 4.0, 1.1

Assume my current location is at the center of the floor. The coordinates are:
- obj_14: 2.4, 3.3, 0.5

To determine which desk is on the right from my current location:
1. Calculate vectors from my position (O) to the desks:
  - Vector OA for obj_10: (2.8 - 2.4, 4.1 - 3.3) = (0.4, 0.8)
  - Vector OB for obj_11: (1.3 - 2.4, 4.0 - 3.3) = (-1.1, 0.7)
2. Calculate the cross product OA x OB to determine relative position:
  - Cross product C = (0.4 * 0.7) - (0.8 * -1.1) = 0.28 + 0.88 = 1.16

Since the cross product C is positive, obj_10 is on the right, and obj_11 is on the left.

Answer: obj_10. *Ours*

The chair that is next to the TV, it is between the table and the tv. (Out-of-domain)

To find the object ID of the chair that is next to the TV, we need to identify the chair and the TV in the scene.

The TV is located at obj_9 with the box coordinates [3.9, 4.1, 1.5, 0.6, 0.4, 0.5].
The chair is located at obj_2 with the box coordinates [4.4, 3.7, 0.6, 0.7, 0.7, 0.9].
However, this chair is not next to the TV.

The next chair is located at obj_3 with the box coordinates [5.3, 3.2, 0.5, 0.6, 0.6, 0.9].
This chair is indeed between the table and the TV. The table is located at obj_8 with the box coordinates [4.9, 3.4, 0.5, 1.0, 1.0, 0.9].

Answer: obj_3. *Base Model*

The objects that might be related to the provided caption are:
- obj_2: Chair at (4.4, 3.7, 0.6) with dimensions (0.7, 0.7, 0.9).
- obj_3: Chair at (5.3, 3.2, 0.5) with dimensions (0.6, 0.6, 0.9).
- obj_9: TV at (3.9, 4.1, 1.5) with dimensions (0.6, 0.4, 0.5).
- obj_8: Table at (4.9, 3.4, 0.5) with dimensions (1.0, 1.0, 0.9).
...
The caption describes a chair next to the TV, between the table and the TV. Given the 2D positions:
- obj_2 is very close to obj_9 (TV) and lies between obj_8 (table) and obj_9 (TV).
- obj_3 is closer to obj_8 (table) and not directly next to obj_9 (TV).

Thus, obj_2 is the chair next to the TV, between the table and the TV.

Answer: obj_2. *Ours*

The chair that is most in the middle of the room. (Out-of-domain)

To find the object ID of the chair in the middle of the room, we need to identify which chair is in the middle.
...
To confirm that this is the middle chair, we need to look at the x and y coordinates of the chairs. obj_10 has an x coordinate of 1.4, which is relatively close to the middle of the room.
...
Based on this information, we can conclude the object ID of the dark color chair in the middle of the room is:

Answer: obj_10. *Base Model*

The relevant objects are those with the name "chair" which are:
- obj_9: "chair", box: 3.7, 1.2, 0.4, 0.7, 0.7, 0.8
...
- obj_12: "chair", box: 3.0, 1.9, 0.6, 0.7, 0.6, 0.5
...
Assuming the current location is the center of "floor" which is at (2.8, 2.3).
To determine which chair is in the middle of the room, we can calculate the Euclidean distance of each chair ...
1. Distance from obj_9 (3.7, 1.2):
  Distance = sqrt((3.7 - 2.8)^2 + (1.2 - 2.3)^2) = sqrt(0.9^2 + 1.1^2) = sqrt(0.81 + 1.21) = sqrt(2.02) ≈ 1.42
...
4. Distance from obj_12 (3.0, 1.9):
  Distance = sqrt((3.0 - 2.8)^2 + (1.9 - 2.3)^2) = sqrt(0.2^2 + 0.4^2) = sqrt(0.04 + 0.16) = sqrt(0.2) ≈ 0.44

The closest chair to the assumed location is obj_12 with a distance of approximately 0.44.

Answer: obj_12. *Ours*

Figure 5. Qualitative results from NR3D. Green box and red box represents the prediction from our model and Llama-3.1-8B, respectively. Boxes are re-plotted for better visualization and some sentences are skipped due to space limits. Best viewed when zoomed in.

training data, and out-of-domain queries that have unseen spatial relationships in training data. The out-of-domain examples demonstrate the effectiveness of our post-training strategy to enable LLM reasoning.

5. Conclusion

In this work, we introduce a novel 3D visual grounding and reasoning data collection pipeline, which can generate post-training data to enhance LLM for 3D visual grounding task. In the data collection process, we also identify and mitigate spatial hallucination in SoTA proprietary LLMs by designing an effective prompt strategy, leading to a more reliable and efficient data collection process. Leveraging our collected training data, we conduct post-training and introduce **3DVG-LLM**, a powerful reasoning LLM specifically tailored for 3D visual grounding tasks. In our experiments, **3DVG-LLM** achieves SoTA performance among all zero-shot methods on ScanRefer and NR3D, demonstrating its effectiveness in reasoning for 3D visual grounding task.

References

[1] Eslam Abdelrahman, Mohamed Ayman, Mahmoud Ahmed, Habib Slim, and Mohamed Elhoseiny. Cot3dref: Chain-of-thoughts data-efficient 3d visual grounding. *arXiv preprint arXiv:2310.06214*, 2023. 2

[2] Ahmed Abdelreheem, Ujjwal Upadhyay, Ivan Skorokhodov, Rawan Al Yahya, Jun Chen, and Mohamed Elhoseiny. 3dreftransformer: Fine-grained object identification in real-world scenes using natural language. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 3941–3950, 2022. 7

[3] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 2, 7

[4] Panos Achlioptas, Ahmed Abdelreheem, Fei Xia, Mohamed Elhoseiny, and Leonidas Guibas. Referit3d: Neural listeners for fine-grained 3d object identification in real-world scenes. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16*, pages 422–440. Springer, 2020. 6, 7

[5] Dave Zhenyu Chen, Angel X Chang, and Matthias Nießner. Scanrefer: 3d object localization in rgb-d scans using natural language. In *European conference on computer vision*, pages 202–221. Springer, 2020. 6

[6] Zhenyu Chen, Ronghang Hu, Xinlei Chen, Matthias Nießner, and Angel X Chang. Unit3d: A unified transformer for 3d dense captioning and visual grounding. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 18109–18119, 2023. 2

[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. 6

[8] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024. 2, 6, 7

[9] Mingtao Feng, Zhen Li, Qi Li, Liang Zhang, XiangDong Zhang, Guangming Zhu, Hui Zhang, Yaonan Wang, and Ajmal Mian. Free-form description guided 3d visual graph network for object grounding in point cloud. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3722–3731, 2021. 7

[10] Weixi Feng, Wanrong Zhu, Tsu-jui Fu, Varun Jampani, Arjun Akula, Xuehai He, Sugato Basu, Xin Eric Wang, and William Yang Wang. Layoutgpt: Compositional visual planning and generation with large language models. *Advances in Neural Information Processing Systems*, 36:18225–18250, 2023. 3

[11] Rao Fu, Jingyu Liu, Xilun Chen, Yixin Nie, and Wenhan Xiong. Scene-llm: Extending language model for 3d visual understanding and reasoning. *arXiv preprint arXiv:2403.11401*, 2024. 7

[12] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025. 2, 3, 4

[13] 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*, pages 2344–2352, 2021. 7

[14] Yining Hong, Haoyu Zhen, Peihao Chen, Shuhong Zheng, Yilun Du, Zhenfang Chen, and Chuang Gan. 3d-llm: Injecting the 3d world into large language models. *Advances in Neural Information Processing Systems*, 36:20482–20494, 2023. 7

[15] Pin-Hao Huang, Han-Hung Lee, Hwann-Tzong Chen, and Tyng-Luh Liu. Text-guided graph neural networks for referring 3d instance segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 1610–1618, 2021. 7

[16] 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. 2

[17] Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024. 4, 5

[18] Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, et al. Openai o1 system card. *arXiv preprint arXiv:2412.16720*, 2024. 2

[19] Ayush Jain, Nikolaos Gkanatsios, Ishita Mediratta, and Katerina Fragkiadaki. Bottom up top down detection transformers for language grounding in images and point clouds. In *European Conference on Computer Vision*, pages 417–433. Springer, 2022. 2, 6, 7

[20] Justin Kerr, Chung Min Kim, Ken Goldberg, Angjoo Kanazawa, and Matthew Tancik. Lerf: Language embedded radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19729–19739, 2023. 6

[21] Junyu Luo, Jiahui Fu, Xianghao Kong, Chen Gao, Haibing Ren, Hao Shen, Huaxia Xia, and Si Liu. 3d-sps: Single-stage 3d visual grounding via referred point progressive selection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16454–16463, 2022. 2

[22] Başak Melis Öcal, Maxim Tatarchenko, Sezer Karaoğlu, and Theo Gevers. Sceneteller: Language-to-3d scene generation. In *European Conference on Computer Vision*, pages 362–378. Springer, 2024. 3

[23] OpenAI. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 6

[24] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language

models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022. 2, 6

[25] Songyou Peng, Kyle Genova, Chiyu Jiang, Andrea Tagliasacchi, Marc Pollefeys, Thomas Funkhouser, et al. Openscene: 3d scene understanding with open vocabularies. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 815–824, 2023. 6

[26] Zhipeng Qian, Yiwei Ma, Zhekai Lin, Jiayi Ji, Xiawu Zheng, Xiaoshuai Sun, and Rongrong Ji. Multi-branch collaborative learning network for 3d visual grounding. In *European Conference on Computer Vision*, pages 381–398. Springer, 2025. 2, 6

[27] Yuxuan Qiao, Haodong Duan, Xinyu Fang, Junming Yang, Lin Chen, Songyang Zhang, Jiaqi Wang, Dahua Lin, and Kai Chen. Prism: A framework for decoupling and assessing the capabilities of vlms. *Advances in Neural Information Processing Systems*, 37:111863–111898, 2025. 2, 3

[28] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 6

[29] 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. 2, 7

[30] Hao Shao, Shengju Qian, Han Xiao, Guanglu Song, Zhuofan Zong, Letian Wang, Yu Liu, and Hongsheng Li. Visual cot: Advancing multi-modal language models with a comprehensive dataset and benchmark for chain-of-thought reasoning. *Advances in Neural Information Processing Systems*, 37:8612–8642, 2025. 3

[31] Ozan Unal, Christos Sakaridis, Suman Saha, and Luc Van Gool. Four ways to improve verbo-visual fusion for dense 3d visual grounding. In *European Conference on Computer Vision*, pages 196–213. Springer, 2025. 2, 6

[32] Yuan Wang, Yali Li, and Shengjin Wang. G3-lq: Marrying hyperbolic alignment with explicit semantic-geometric modeling for 3d visual grounding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13917–13926, 2024. 2, 6

[33] Zehan Wang, Haifeng Huang, Yang Zhao, Linjun Li, Xize Cheng, Yichen Zhu, Aoxiong Yin, and Zhou Zhao. Distilling coarse-to-fine semantic matching knowledge for weakly supervised 3d visual grounding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2662–2671, 2023. 6, 7

[34] Yanmin Wu, Xinhua Cheng, Renrui Zhang, Zesen Cheng, and Jian Zhang. Eda: Explicit text-decoupling and dense alignment for 3d visual grounding. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 19231–19242, 2023. 2, 6

[35] Guowei Xu, Peng Jin, Hao Li, Yibing Song, Lichao Sun, and Li Yuan. Llava-cot: Let vision language models reason step-by-step, 2025. 2, 3, 4

[36] Jianing Yang, Xuweiyi Chen, Shengyi Qian, Nikhil Madaan, Madhavan Iyengar, David F Fouhey, and Joyce Chai. Llm-grounder: Open-vocabulary 3d visual grounding with large language model as an agent. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7694–7701. IEEE, 2024. 2, 6

[37] Yue Yang, Fan-Yun Sun, Luca Weihs, Eli VanderBilt, Alvaro Herrasti, Winson Han, Jiajun Wu, Nick Haber, Ranjay Krishna, Lingjie Liu, et al. Holodeck: Language guided generation of 3d embodied ai environments. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16227–16237, 2024. 3

[38] 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 (ICCV)*, pages 1856–1866, 2021. 2

[39] Zhihao Yuan, Xu Yan, Yinghong Liao, Ruimao Zhang, Sheng Wang, Zhen Li, and Shuguang Cui. Instancerefer: Cooperative holistic understanding for visual grounding on point clouds through instance multi-level contextual referring. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1791–1800, 2021. 6, 7

[40] Zhihao Yuan, Jinke Ren, Chun-Mei Feng, Hengshuang Zhao, Shuguang Cui, and Zhen Li. Visual programming for zero-shot open-vocabulary 3d visual grounding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20623–20633, 2024. 2, 6, 7

[41] 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. 2, 6, 7

[42] Chenming Zhu, Tai Wang, Wenwei Zhang, Kai Chen, and Xihui Liu. Scanreason: Empowering 3d visual grounding with reasoning capabilities. In *European Conference on Computer Vision*, pages 151–168. Springer, 2024. 2

[43] Ziyu Zhu, Xiaojian Ma, Yixin Chen, Zhidong Deng, Siyuan Huang, and Qing Li. 3d-vista: Pre-trained transformer for 3d vision and text alignment. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2911–2921, 2023. 6