

SPAZER: Spatial-Semantic Progressive Reasoning Agent for Zero-shot 3D Visual Grounding

Appendix

A. Implementation details

B. Additional experimental results and analysis:

- B.1: Error type analysis
- B.2: Performance on subset vs. full dataset
- B.3: Inference time

C. Limitations and border impact:

- C.1: Analysis of failure cases
- C.2: Broader impact

A Implementation details

Computational resources. Note that most of our experiments adopt GPT-4o as the VLM and it requires no GPU-based computation. The experiments involving Qwen2-VL-72B and Qwen2.5-VL-72B are conducted on multiple NVIDIA H100 GPUs.

Model details. The default VLM of our agent is GPT-4o (gpt-4o-2024-08-06). And the temperature is set to 0.2 to improve the reproducibility of the results. On ScanRefer dataset [6], we use Mask3D [29] to obtain 3D bounding box predictions, which is consistent with prior works [41, 19].

Prompt design. Our agent adopts several different prompts for the VLM. We first conduct target class prediction using the prompt in Tab. 5, which is similar with VLM-Grounder [35]. For **view selection** (Sec. 3.2), we simply tell the VLM to select the view that can observe the query-described object most clearly using the prompt in Tab. 6. In **candidate object screening** (Sec. 3.3), we prompt the VLM to select Top- k candidate objects based on the annotated object IDs, as shown in Tab. 7. Eventually, **3D-2D joint decision-making** (Sec. 3.4) is achieved using the detailed prompt in Tab. 8.

Table 5: Prompt for reasoning the object category from the query description. "{text}" represents the input query.

Prompt Template for Target Class Prediction

You are working on a 3D visual grounding task, which involves receiving a query that specifies a particular object by describing its attributes and grounding conditions to uniquely identify the object.

Now, I need you to first parse this query and return the category of the object to be found. Sometimes the object’s category is not explicitly specified, and you need to deduce it through reasoning. If you cannot deduce after reasoning, you can use "unknown" for the category. Your response should be formatted in JSON.

Here are some examples:

Input: Query: this is a brown cabinet. it is to the right of a picture.

Output:

```
{
  "target_class": "cabinet"
}
```

Input: Query: it is a wooden computer desk. the desk is in the sleeping area, across from the living room. the desk is in the corner of the room, between the nightstand and where the shelf and window are.

Output:

```
{
  "target_class": "desk"
}
...
```

Now start your task:

Input: "{text}"

Table 6: View selection prompt for identifying the best 3D view to locate the target object. {target_class} is the predicted object class and "{text}" denotes the query text.

Prompt Template for View Selection

You are good at finding the object in a 3D scene based on a given query description. These images show different views of a room. You need to find the {target_class} in this query description: "{text}"

Please review all view images to find the target object and select the view that you can see the target object most clearly.

Output your answer in JSON format with these keys:

```
{
  "reasoning": "Explain how you identified the target object,
and why you choose this view.",
  "view": "2" // The number of the view is in the top left
corner of the corresponding image.
}
```

Table 7: Candidate screening prompt for identifying the Top- k object IDs based on a given query. {target_class} is the predicted object class and "{text}" denotes the query text. {n_topk} is set to 4. {object_id_list} contains the valid object IDs after anchor filtering.

Prompt Template for Candidate Screening

Here is the annotated image of the selected view. All objects belonging to the {target_class} class are labeled by a unique number (ID) in red color on them.

Please select the object ID that best matches the given query description: "{text}"

Carefully analyze the specified conditions (such as shape, color, relative position with surrounding objects) in the given query, then select top-{n_topk} best-matched object IDs. The selected top-{n_topk} object IDs should be sorted in descending order of confidence. The object ID should be chosen from this list: {object_id_list}

Output your answer in JSON format with these keys:

```
{
  "reasoning": "Explain how you identified and ranked the
top-{n_topk} target object IDs.",
  "object_id": [1, 2, 3, 4, 5] // A list of {n_topk}
selected target object IDs.
}
```

Table 8: Input prompt for 3D-2D joint decision-making. {target_class} is the predicted object class and "{text}" denotes the query text. {object_id_list} contains the valid object IDs after anchor filtering.

Prompt Template for 3D-2D Joint Decision-Making
<p>You are provided with a set of images depicting an indoor scene:</p> <ul style="list-style-type: none"> • A global view image showing the room’s 3D layout from a fixed perspective. • Several camera images captured from different viewpoints around the room. <p>All objects of interest in the scene are labeled with unique object IDs (in red), which are consistent across both the global and camera images.</p> <p>Your task is to identify the object ID that best matches the given query description. Follow the steps below:</p> <hr/> <p>1. Start with the global view image:</p> <ul style="list-style-type: none"> • Analyze the overall spatial layout and object distribution in the room. • Use the global view to evaluate view-independent spatial relationships, which do not rely on a specific viewpoint: <i>Examples include:</i> near, close to, next to, far, above, below, under, on, top of, middle, opposite <p>2. Then examine the camera images:</p> <ul style="list-style-type: none"> • Validate candidate objects identified from the global view. • Evaluate visual features: color, shape, size, texture, and material. • Use camera views to judge view-dependent spatial relationships, which depend on the camera perspective: <i>Examples include:</i> left, right, in front of, behind, back, facing, looking, between, across from, leftmost, rightmost • Tip: Annotations may not always be at the center of the object. Focus on the full <i>spatial extent</i> and choose the ID that best represents the <i>main body</i> of the described object across both views. <p>3. Iterate if needed:</p> <ul style="list-style-type: none"> • If no candidate fully matches the query, return to the global view and reassess alternatives. • Repeat verification with camera images until you confidently identify the best match. <hr/> <p>Task: Select the object ID of the target class: {target_class} Query description: "{text}" Object IDs to choose from: {object_id_list}</p> <p>Output format (JSON):</p> <pre>{ "reasoning": "Explain how you analyzed spatial relationships (view-dependent vs view-independent), cross-verified the object across views, and selected the best-matched ID.", "object_id": ID // e.g., 10 }</pre>

855 B Additional experimental results and analysis

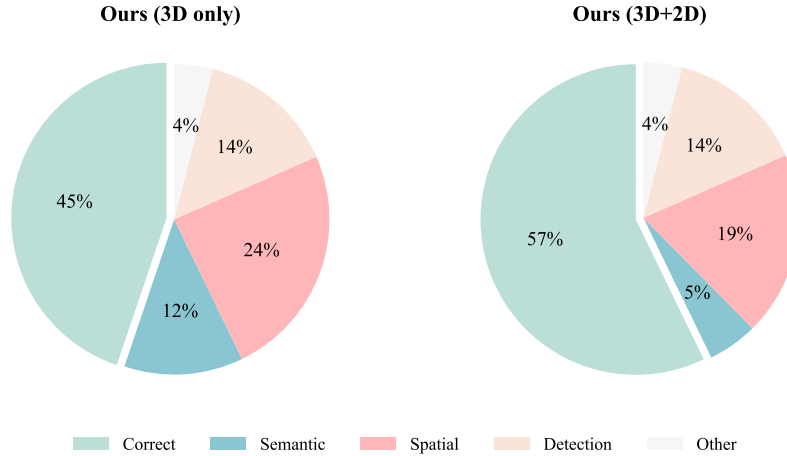


Figure 5: Error type distribution on ScanRefer dataset. Ours (3D only) indicates that our agent selects the target object directly at the candidate screening stage, without performing the subsequent 3D-2D joint decision-making. Ours (3D+2D) represents the full pipeline of our SPAZER method.

856 B.1 Error type analysis

857 In Fig. 5 we present the distribution of different error types in SPAZER’s predictions to provide
 858 insights into potential directions for further improvement. The main error types include: 1) Semantic:
 859 Errors caused by the model overlooking critical semantic cues, such as color, shape, *etc.*; 2) Spatial:
 860 Errors where the model fails to correctly interpret spatial relationships, including relative positions
 861 between objects or absolute directions (*e.g.*, northwestern-most); 3) Detection: Cases where the 3D
 862 detector fails to detect the target object or predicts the wrong category; 4) Other: Mainly due to
 863 referring ambiguities in the query text, where multiple objects in the scene could reasonably match
 864 the description based on human judgment.

865 **Uni-modal and multi-modal.** Compared to the 3D-only paradigm, incorporating 3D+2D
 866 significantly reduces errors in the Semantic category, indicating that 2D images provide important
 867 supplementary semantic cues for the agent. Additionally, the reliable view-dependent relational
 868 information from 2D images also helps reduce the occurrence of Spatial errors.

869 **Analysis and future work.** Based on the error type distribution of our SPAZER (right side of Fig. 5),
 870 Spatial errors account for the largest proportion, indicating that the primary challenge in 3DVG lies
 871 in understanding complex spatial relationships. To address this, we plan to further explore more
 872 effective 3D representations in future work. In addition, a considerable portion of errors is caused by
 873 the detector, suggesting that reducing the agent’s reliance on the detector is one of the key issues to
 874 be addressed in future work.

875 B.2 Performance on subset vs. full dataset

876 In consideration of budget and evaluation efficiency, we follow previous work VLM-Grounder [35]
 877 to evaluate our agent on the subset (250 selected samples) of each dataset. To further verify whether
 878 the results on the subset are comparable to those on the full dataset, we conduct additional
 879 experiments using open-source VLMs, as shown in Tab. 9. We observe that both our method and
 880 prior work SeeGround [19] exhibit consistent performance across the two dataset partitions, with
 881 overall accuracy variations under 2.0, which is negligible compared to the improvement achieved by
 882 our method.

883 B.3 Inference time

884 In Tab. 10, we break down the inference time of each step in SPAZER. The time consumption across
 885 different steps is relatively balanced, with the view selection step being faster since it requires no
 886 additional computation. Moreover, compared to VLM-Grounder [35], which also leverages 2D
 887 camera images for reasoning, our method achieves significantly higher inference efficiency. This is

Table 9: Performance comparison on the full set and subset of the Nr3D [1] dataset.

Dataset	Method	VLM	Easy	Hard	Dep.	Indep.	Overall
Full	SeeGround [19]	Qwen2-VL-72B	54.5	38.3	42.3	48.2	46.1
	Ours	Qwen2-VL-72B	59.3	44.8	46.3	54.9	51.8
	Ours	Qwen2.5-VL-72B	62.4	46.9	49.9	56.8	54.3
Subset	SeeGround [19]	Qwen2-VL-72B	51.5	37.7	44.8	45.5	45.2
	Ours	Qwen2-VL-72B	62.5	43.0	51.0	55.2	53.6
	Ours	Qwen2.5-VL-72B	60.3	50.9	54.2	57.1	56.0

Table 10: Inference time of each step in our SPAZER. Ours significantly outperforms VLM-Grounder in efficiency. Both methods adopt GPT-4o as the VLM.

Method	Step	Time (s)	Total (s)
Ours (SPAZER)	3D Holistic View Selection	5.2	23.5
	Candidate Object Screening	8.5	
	3D-2D Joint Decision-Making	9.8	
VLM-Grounder [35]	-	-	50.3

888 because we rely on VLM-selected anchors and require only a small number of images, avoiding the
889 need to sample and filter all video frames.

890 C Limitations and border impact

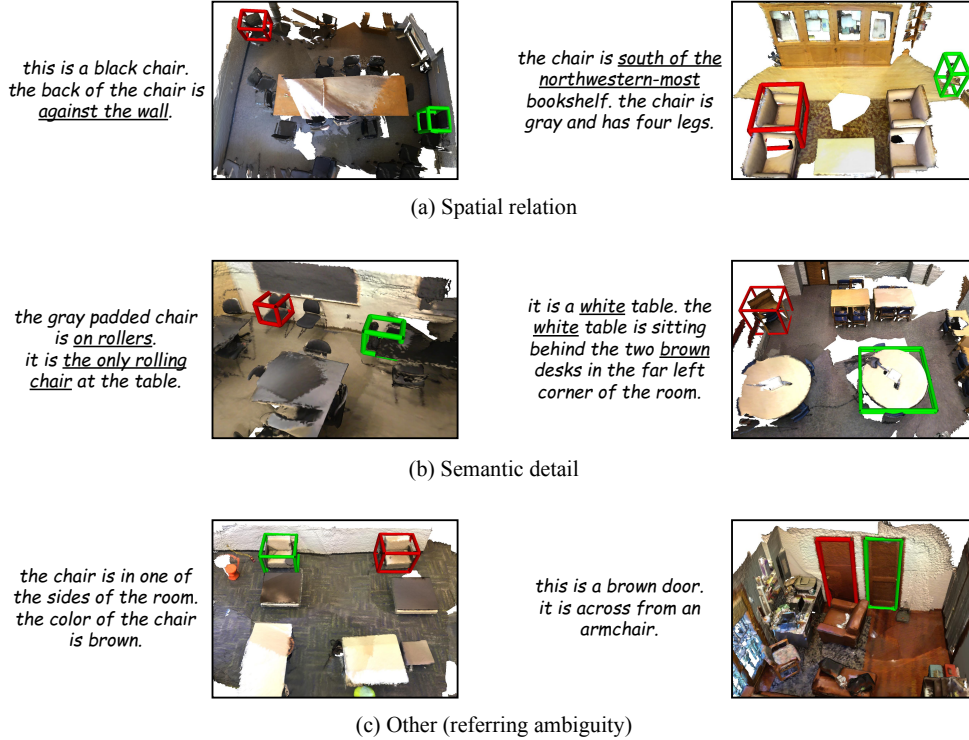


Figure 6: Typical types of failure cases. The prediction and ground-truth are highlighted in red and green, respectively. (a) Relation error includes relative relation (e.g., against the wall) and absolute relation (e.g., northwestern-most). (b) Semantic error mainly involves detailed object attributes, such as shape (on rollers), color (white), material, etc. (c) Other errors are primarily caused by the referring ambiguity, i.e., multiple objects in the scene satisfy the query.

891 C.1 Analysis of failure cases

892 Based on Fig. 5, there are mainly four types of failure cases. Since we adopt a detect-and-match
893 paradigm similar to previous works [41, 19], detection-related errors are currently unavoidable. In
894 future work, we plan to explore how to enable the agent to directly produce localization results. The
895 remaining three types of errors are illustrated through case studies in Fig. 6.
896 Regarding spatial relations, failures mainly occur in: 1) complex positional relationships, which
897 often involve both the orientation of the target object and its relation to surrounding objects; 2)
898 directional terms (e.g., south, northwest). In future research, we plan to incorporate visual prompts
899 indicating orientation into the 3D representation.
900 For semantic details, when the scene contains multiple visually similar objects, the candidate
901 screening stage may fail to include the correct target into the Top- k list, preventing effective semantic
902 verification in subsequent steps. This reflects a limitation of our current method.
903 Lastly, we observed that some samples in the dataset exhibit referring ambiguity. As shown in
904 Fig. 6(c), both the predicted result and the ground truth can satisfy the query description based on
905 human interpretation. Therefore, the construction of higher-quality 3DVG datasets stands out as a
906 critical challenge that needs to be addressed in future research.

907 C.2 Broader impact

908 Our VLM-driven agent SPAZER for 3D visual grounding offers potential benefits in areas such as
909 human-robot interaction, augmented reality, and assistive technologies by enabling more intuitive
910 object localization from language. However, it may also carry risks, such as biases inherited from
911 pre-trained models, which could affect performance in diverse environments. Future work should
912 address these concerns through fairness-aware training, improved interpretability, and responsible
913 deployment.