Supplementary Materials for paper ScanReason: Empowering 3D Visual Grounding with Reasoning Capabilities

Chenming Zhu[1,2], Tai Wang[2], Wenwei Zhang[2], Kai Chen[2], and Xihui Liu[1†]

[1] The University of Hong Kong
[2] Shanghai AI Laboratory

The supplementary material consists of quantitative evaluations on text description of 3D reasoning grounding (Sec. A), more visualization (Sec. B), the ScanReason annotations generation prompts (Sec. C) and details of instruction tuning datasets (Sec. D).

A More Evaluations

Considering the expected outputs of 3D reasoning grounding questions consist of not only the target objects' 3D bounding boxes but also text responses including either demonstrating the explanation (e.g. Why to choose these objects) or offering reasonable suggestions (e.g. How to use these objects.). We argue that it is also necessary to evaluate the text response's correctness. However, due to the complexity and diversity of the answers, it is non-trivial to use or design a proper evaluation method that can ensure the evaluation accuracy. To ensure the evaluation accuracy with limited human and time resources, we uniformly sample 100 reasoning grounding pairs from evaluation datasets and test GPT-4 [5], 3D-LLM [3] and ReGround3D on the datasets. Then we manually score the 300 responses using an integer ranging from 1 to 5, while 1 indicates an incorrect answer, 5 is a correct answer. The matching score λ_i represents levels of the similarity between the response and ground-truth answer. The correctness metric is denoted as :

$$S = \frac{1}{N} \sum_{i}^{N} \left(\frac{\lambda_i - 1}{4} \right) \times 100\% \tag{1}$$

The results in Tab. 1 demonstrate that even if GPT-4 could not access information of the 3D scene, it can "guess" the answer to the complex reasoning question based on its powerful world knowledge and common sense, and it could serve as a strong baseline for evaluation. ReGround3D achieves the superior performance (38.7 vs. 32.4) based on a much smaller LLM (FlanT5$_{\text{XL}}$-3B) and has the ability to localize the target objects in the 3D scene at the same time.

Table 1: Matching scores of text responses on 3D reasoning grounding task among ReGround3D(ours), 3D-LLM and GPT-4.

Methods	Spatial	Functional	Logical	Emotional	Safety	Overall
GPT-4 [5]	23.7	48.7	29.1	19.7	26.5	32.4
3D-LLM(vg) [3]	17.2	24.4	18.4	11.1	13.8	17.2
ReGround3D	34.9	49.2	35.1	30.1	30.2	38.7

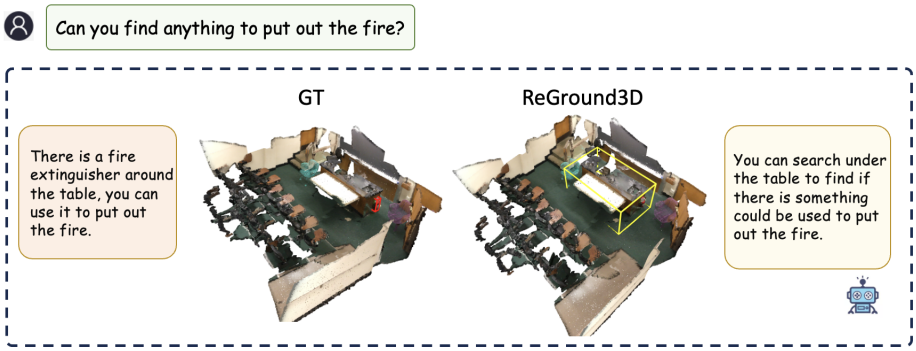

Fig. 1: This example demonstrates that our model sometimes struggles to localize the small and long-tailed objects in the 3D scene.

B More Visualizations

B.1 More Examples

In this section we will illustrate more examples of our ScanReason benchmark for each type of reasoning questions in Fig. 3, Fig. 4, Fig. 5, Fig. 6 and Fig. 7. Each example consists of the reasoning question, target object locations (3D bounding boxes), and the corresponding text response.

B.2 Failure Case Analysis

Illustrated in the qualitative results in the paper, we find that our model tends to output much fewer 3D bounding boxes compared with the ground-truth 3D bounding boxes when multiple objects are regarded as the target objects. Besides, as shown in Fig. 1, even if the 3D grounding module is introduced to more accurately localize the target objects, ReGround3D still struggles to recognize and localize the small and long-tailed objects in the 3D scene, which has been a long-term challenge in the 3D object detection and grounding model. Besides, we found that sometimes the visual-centric reasoning module seems to misinterpret human intention, as illustrated Fig. 2, where we attribute this phenomenon to the lack of the corresponding type of training data, leading to incorrect responses to this type of problem.

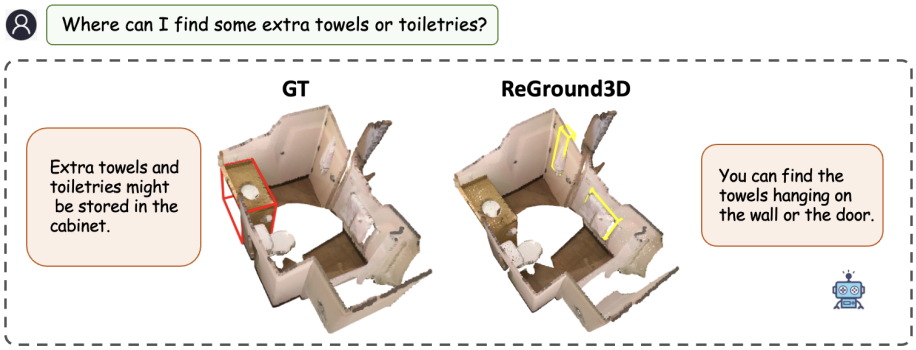

Fig. 2: This example demonstrates that our model sometimes misinterprets human intention.

C ScanReason Annotations Generation Prompts

We show five prompt templates for generating five types of reasoning question-answer-location pairs, each comprising system messages and manually crafted context examples. In our attempts, since GPT-3.5 struggles to understand the 3D spatial relationships of objects in the scene based on the provided 3D spatial coordinates of objects, we resort to GPT-4 for data generation, which is verified to be much better than GPT-3.5 in understanding the spatial relationships. We input the category information and 3D bounding boxes of the objects in the 3D scenes, providing information about the semantics and spatial locations of the scene in a textual representation. Then we provide specific instructions to the GPT-4 [5] to generate diverse data. As shown in Fig. 8, Fig. 9, Fig. 10, Fig. 11 and Fig. 12, to further make the generated question-answer-location pairs more accurate and responsive, we adopt prompt engineering by giving GPT-4 [5] about 3-5 few-shot examples to show what kind of data it is should generate. For each sample in the few shot samples, the "content" has the object ids, category information, and 3D bounding boxes of the objects in the scenes, and the "response" refers to human-written question-answer-location pairs. We include the 3D bounding boxes and categories information of all the objects in scenes into "query" and ask the GPT-4 [5] to give us 10 meaningful samples.

D Details of Instruction Tuning Datasets

D.1 Data Reformulation

3D Object Detection data. Generally speaking, 3D object detection datasets contain information about 3D bounding boxes of all objects in the pre-defined list of categories. In order to cover as many objects as possible, we chose to construct question and answer pairs based on the EmbodiedScan [6] dataset, which includes 160k 3D-oriented boxes spanning over 760 categories. During the model training process, we convert the annotations of 3D boxes into a specific

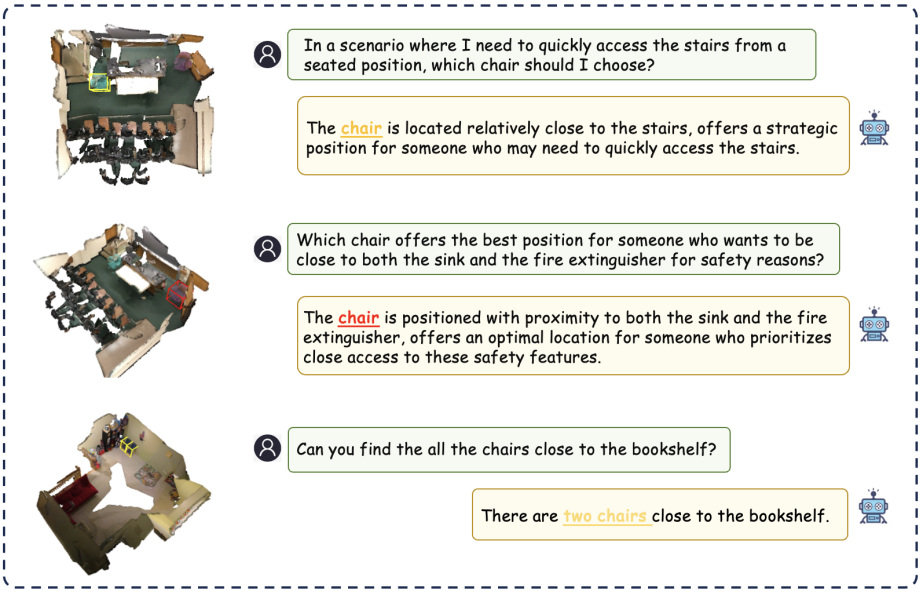

Fig. 3: Examples of spatial reasoning data. Each row represents one question-answer-location pair. The left column represents the 3D scene and target object location, and the right column shows the question and corresponding text answer.

question answering pair template: "User: `<scene>` Where is the `<category>` in this 3D Scene? Assistant: Sure, `<LOC>`." Here, `<category>` is randomly selected from the ground-truth categories contained in the current 3D scene, `<scene>` is the placeholder of 3D scene tokens.

3D Visual Grounding data. 3D visual grounding data aims at localizing the unique object in 3D scenes given the descriptive object expression. The descriptions of objects in these data typically explicitly include the object attributes and their spatial relationships with other objects. To ensure diversity in training data, we selected ScanRefer [2], SR3D [1], NR3D [1] as training data. Considering the variety of object descriptions, it is difficult to simply reformulate the object expression using a simple template like: " User: `<scene>` Where is `<expr>` in the 3D Scene?. Assistant: It is `<LOC>`." Therefore, we choose to retain the original object description as much as possible and use a template: "Here is a description about an object: " `<expr>` ", where is the object in the 3D Scene? Assistant: It is `<LOC>`.", where `<expr>` represents the object description in the data. Besides, we have created a range of similar question templates that are randomly selected during the training process.

Spatial Question Answering data We hope the model can understand 3D position in a more natural way using numerical values in Natural Language. We use $[x, y, z, dx, dy, dz]$ to represent a 3D box, where $[x, y, z]$ represents the cen-

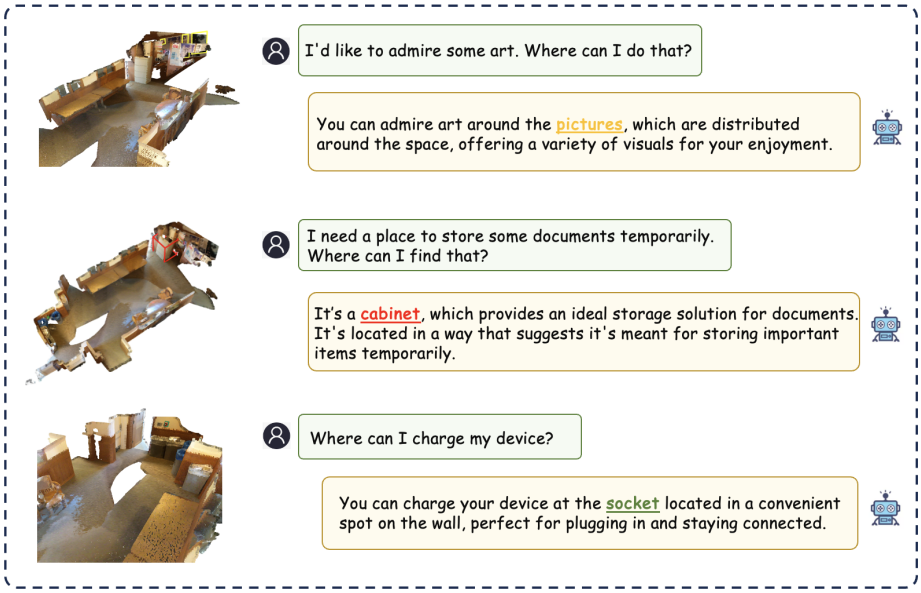

Fig. 4: Examples of functional reasoning data. Each row represents one question-answer-location pair. The left column represents the 3D scene and target objects location, and the right column shows the question and corresponding text answer.

ter of a 3D area and $[dx, dy, dz]$ represents the 3D box size, these coordinates can appear anywhere in the input text. Since there are no explicit coordinate question answering pairs in the 3D vision-language datasets, we turned our attention to the SR3D dataset. SR3D is a template-based generated dataset that not only provides expressions of target objects but also provides object ids for target objects and anchor objects. Based on SR3D, we constructed a 3D QA dataset focusing on 3D positional relationships between objects. For example, the query "select the trash can that is beneath the desk" from SR3D dataset can be transformed into "User: Is the trash can $[x_1, y_1, z_1, dx_1, dy_1, dz_1]$ beneath the desk $[x_2, y_2, z_2, dx_2, dy_2, dz_2]$?" Assistant: Yes." with the assistance of GPT-3.5.

3D Question Answering data. Considering that we expect the model can also output reasonable answers in the conversation, we also introduce 3D QA data during the training process to further enhance the model's 3D visual question answering and scene understanding capabilities. We reformulate CLEVR3D [7] data into a simple question-answer template: "User: `<scene>` `<question>`. Assistant: `<answer>`." The SQA3D [4] dataset not only provides questions but also provides the situation in which the questions are asked. We reformulate it into a template: "User: `<scene>` `<situation>` `<question>`. Assistant: `<answer>`." wherein the `<question>` is the placeholder for the question and `<situation>` is the placeholder of the situation while raising the corresponding question.

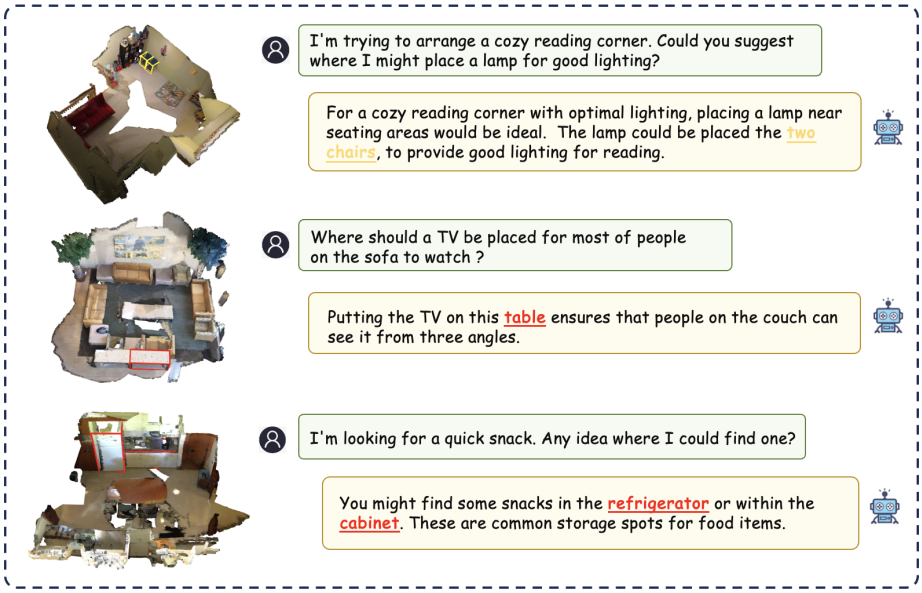

Fig. 5: Examples of logical reasoning data. Each row represents one question-answer-location pair. The left column represents the 3D scene and target objects location, and the right column shows the question and corresponding text answer.

3D Reasoning Grounding data. In addition to the types of data mentioned above, We also employed our own proposed 3D reasoning grounding data to train the model, further enhancing its capability to handle complex reasoning questions. The output format is more akin to a combination of 3D Question Answering (QA) and localization, where the model's response not only includes the target object but also provides an explanation for the selection of the target object. We adopted a template: "User: `<scene>` `<question>`. Assistant: Sure, `<LOC>`, `<reason>`."

D.2 Output Type Templates

In the actual interaction between users and the model, questions are generally not divided according to the aforementioned tasks but are more concerned with whether the model's output format meets the user's needs. For example, in the 3D QA data, there exists a question: "Where is the pillow on the bed?", with the corresponding answer being "near the headboard". Simultaneously, such questions may also appear in our reformulated 3D Object Detection and Visual Grounding datasets, where the desired model output is the specific location coordinates of the object in the scene. To make the interaction between the model and users more natural and the outputs more in line with user needs, we accordingly employ output type templates appended to user instructions. Such instructions enable the training data to break free from the constraints of its original task and integrate more naturally according to the output type,

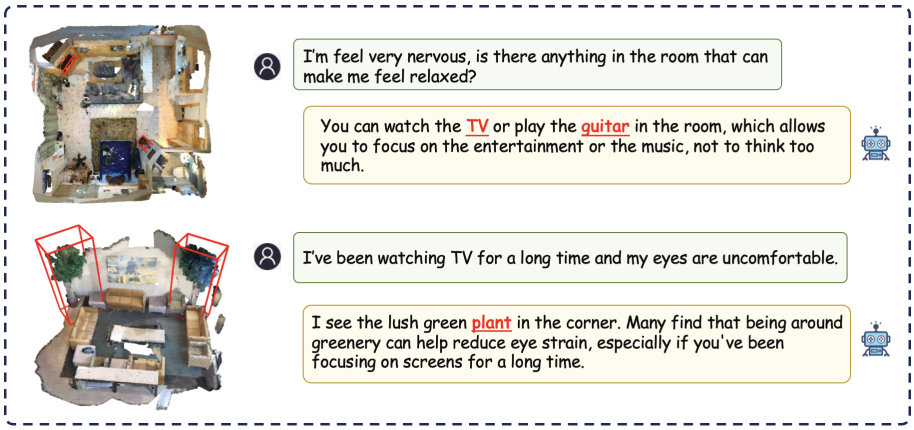

Fig. 6: Examples of emotional reasoning data. Each row represents one question-answer-location pair. The left column represents the 3D scene and target objects location, and the right column shows the question and corresponding text answer.

thereby further enhancing the model's understanding and response to complex and varied inputs in natural dialogue.

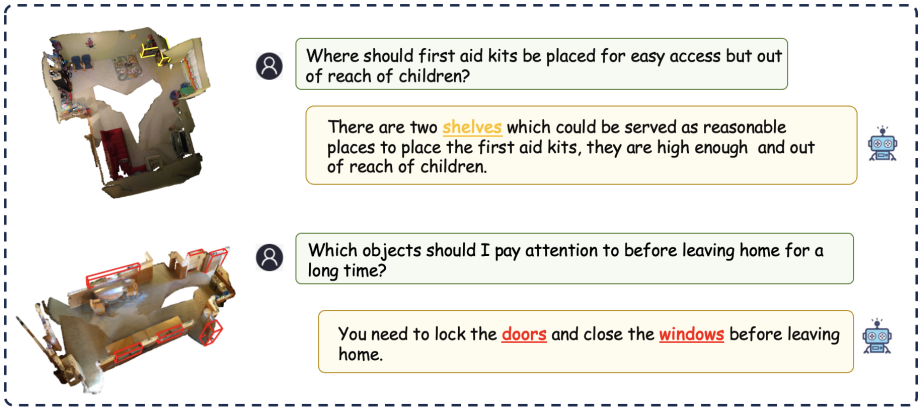

Fig. 7: Examples of safety reasoning data. Each row represents one question-answer-location pair. The left column represents the 3D scene and target object location, and the right column shows the question and corresponding text answer.

```
messages=[{"role": "system", "content" "You are an AI visual assistant that can analyze a
3D scene. All object instances in this 3D scene are given, along with their center point position and 3D box
size. Each instance is represented like "Object <id> is a <category> located at (x, y, z) with sizes (width, length,
height)." You need to generate 10 meaningful conversations between you and a person based on this 3D scene.
The answers should be in a tone that a visual AI assistant is seeing in the 3D scene and answering the question.
Ask diverse questions and give corresponding answers. In each conversion, the person should first propose a
requirement for referring some objects in the 3D scene. Such conversation should meeting the following
requirements: (1) the question should be implicit complex referring some existing 3D objects based on the
object function, not explicitly including the object category and exact 3D location. (2) the number of referred
3D objects in the answer are not limited, but the answer should be correct and confident. you can directly use
object id to represent the object.
Do not ask any question that cannot be answered confidently."}]

for sample in fewshot_samples:
      messages.append({"role": "user", "content": sample['content']})
      messages.append({"role": "assistant", "content":
      sample['response']})
      messages.append({"role": "user",  "content": '\n'.join(query)})
```

Fig. 8: Prompts on generating functional reasoning question-answer-location pairs data..

```
messages=[{"role": "system", "content" "You are an AI visual assistant that can analyze a
3D scene. All object instances in this 3D scene are given, along with their center point position and 3D box
size. Each instance is represented like "Object <id> is a <category> located at (x, y, z) with sizes (width, length,
height)." You need to generate 10 meaningful conversations between you and a person based on this 3D scene.
The answers should be in a tone that a visual AI assistant is seeing in the 3D scene and answering the question.
Ask diverse questions and give corresponding answers. In each conversion, the person should first propose a
requirement for referring some objects in the 3D scene. Such conversation should meeting the following
requirements: (1) the question should be implicit complex referring some existing 3D objects based on the
spatial relationship, not explicitly including the object category and exact 3D location. For example, the
question could be similar like 3D visual grounding task, or it could be offered an assumed human position,
question the object spatial relationship with human. (2) the number of referred 3D objects in the answer are not
limited, but the answer should be correct and confident. you can directly use object id to represent the object.
Do not ask any question that cannot be answered confidently and do not ask any question about object
function."}]
for sample in fewshot_samples:
      messages.append({"role": "user", "content": sample['content']})
      messages.append({"role": "assistant", "content":
      sample['response']})
      messages.append({"role": "user", "content": '\n'.join(query)})
```

Fig. 9: Prompts on generating spatial reasoning question-answer-location pairs data.

```
messages=[{"role": "system", "content" "You are an AI visual assistant that can analyze a 3D
scene. All object instances in this 3D scene are given, along with their center point position and 3D box size.
Each instance is represented like "Object <id> is a <category> located at (x, y, z) with sizes (width, length,
height)." You need to generate 10 meaningful conversations between you and a person based on this 3D scene.
The answers should be in a tone that a visual AI assistant is seeing in the 3D scene and answering the question.
Ask diverse questions and give corresponding answers. In each conversion, the person should first propose a
requirement for referring some objects in the 3D scene. Such conversation should meeting the following
requirements: (1) the questions should implicit complex referring some existing 3D objects based on the
application of logistic reasoning to infer the functions of objects and their utility, meanwhile considering spatial
relationships and the person's current or assumed position, not explicitly including the object category and
exact 3D location. (2) the number of referred 3D objects with their object ids in the answer are not limited. Do
not ask any question that cannot be answered confidently."}]

for sample in fewshot_samples:
messages.append({"role": "user", "content": sample['content']})
      messages.append({"role": "assistant", "content":
      sample['response']})
      messages.append({"role": "user", "content": '\n'.join(query)})
```

Fig. 10: Prompts on generating logical reasoning question-answer-location pairs data..

```
messages=[{"role": "system", "content" "You are an AI visual assistant that can analyze a 3D scene. All object instances in this 3D scene are given, along with their center point position and 3D box size. Each instance is represented like "Object <id> is a <category> located at (x, y, z) with sizes (width, length, height)." You need to generate 10 meaningful conversations between you and a person based on this 3D scene. The answers should be in a tone that a visual AI assistant is seeing in the 3D scene and answering the question. Ask diverse questions and give corresponding answers. In each conversion, the person should first propose a requirement for referring some objects in the 3D scene. Such conversation should meeting the following requirements: (1) Each conversation starts with a human expressing an emotion, a preference, an activity or a behaviour pattern, and your response should give the suggesting target objects within a given 3D scene that could positively influence their emotional state. (2) Make sure your questions probe the emotional and psychological dimensions of human-AI interaction, avoiding purely logistical or spatial queries. The aim is to foster a deeper connection and understanding between the AI and humans, providing responses that are not only logical but also emotionally intelligent and considerate.
Do not ask any question that cannot be answered confidently."}]
for sample in fewshot_samples:
messages.append({"role": "user", "content": sample['content']})
      messages.append({"role": "assistant", "content":
      sample['response']})
      messages.append({"role": "user",  "content": '\n'.join(query)})
```

Fig. 11: Prompts on generating emotional reasoning question-answer-location pairs data..

```
messages=[{"role": "system", "content" "You are an AI visual assistant that can analyze a 3D scene. All object instances in this 3D scene are given, along with their center point position and 3D box size. Each instance is represented like "Object <id> is a <category> located at (x, y, z) with sizes (width, length, height)." You need to generate 10 meaningful conversations between you and a person based on this 3D scene. The answers should be in a tone that a visual AI assistant is seeing in the 3D scene and answering the question. Ask diverse questions and give corresponding answers. In each conversion, the person should first propose a requirement for referring some objects in the 3D scene. Such conversation should meeting the following requirements: (1) the questions should implicit complex referring some existing 3D objects based on highlighting a safety concern or requirement, not explicitly including the object category and exact 3D location. (2) the number of referred 3D objects with their object ids in the answer are not limited. Do not ask any question that cannot be answered confidently."}]


for sample in fewshot_samples:
messages.append({"role": "user", "content": sample['content']})
      messages.append({"role": "assistant", "content":
      sample['response']})
      messages.append({"role": "user",  "content": '\n'.join(query)})
```

Fig. 12: Prompts on generating safety reasoning question-answer-location pairs data.

References

1. Achlioptas, P., Abdelreheem, A., Xia, F., Elhoseiny, M., Guibas, L.: 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. pp. 422–440. Springer (2020)
2. Chen, D.Z., Chang, A.X., Nießner, M.: Scanrefer: 3d object localization in rgb-d scans using natural language. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XX. pp. 202–221. Springer (2020)
3. Hong, Y., Zhen, H., Chen, P., Zheng, S., Du, Y., Chen, Z., Gan, C.: 3d-llm: Injecting the 3d world into large language models. Advances in Neural Information Processing Systems **36** (2024)
4. Ma, X., Yong, S., Zheng, Z., Li, Q., Liang, Y., Zhu, S.C., Huang, S.: Sqa3d: Situated question answering in 3d scenes. arXiv preprint arXiv:2210.07474 (2022)
5. OpenAI: Gpt-4 technical report (2023)
6. Wang, T., Mao, X., Zhu, C., Xu, R., Lyu, R., Li, P., Chen, X., Zhang, W., Chen, K., Xue, T., et al.: Embodiedscan: A holistic multi-modal 3d perception suite towards embodied ai. arXiv preprint arXiv:2312.16170 (2023)
7. Yan, X., Yuan, Z., Du, Y., Liao, Y., Guo, Y., Li, Z., Cui, S.: Clevr3d: Compositional language and elementary visual reasoning for question answering in 3d real-world scenes. arXiv preprint arXiv:2112.11691 (2021)