3D-GRAND: A Million-Scale Dataset for 3D-LLMs with Better Grounding and Less Hallucination

Supplementary Material

A. Detailed Comparison with SceneVerse

Among recent datasets, 3D-GRAND is most similar to SceneVerse [34]. They are both million-scale grounding datasets for 3D-LLMs. However, there are a few key differences: (1) SceneVerse [34] provides only sparse grounding, while 3D-GRAND is densely grounded. In 3D-GRAND, every noun phrase in the text—whether it's in captions, QAs, or object references—is explicitly grounded to a corresponding object in the 3D scene, whereas SceneVerse does not offer this level of grounding granularity. To elucidate this difference, we present Table 2 and 8 that compares SceneVerse and 3D-GRAND; (2) the language annotations of 3D-GRANDare more trustworthy and have higher quality. Hallucination is known as one of the most common mistakes of LLMs [32, 43, 58] In 3D-GRAND, we employ a hallucination filter to check and delete any annotations with hallucinated object IDs. This is not possible for SceneVerse since they have pure language output. 3D-GRAND is also quality-checked by humans to ensure the quality.

	Grounding Granularity	Object Reference Data
SceneVerse	Session-level many-to-one grounding	This is a big cotton sofa. It is between the window and the wooden table. → *sofa-3*
3D-GRAND	Noun-level one-to-one grounding	This is a ***big cotton sofa*** [*sofa-3*] . ***It*** [*sofa-3*] is between the ***window*** [*window-0*] and ***wooden table*** [*table-4*] .

Table 8. Example of grounding granularity. 3D-GRAND focuses on dense grounding.

Definitions of grounding granularity:

- **Paragraph-level set-to-set grounding**: Many sentences in a long paragraph, each containing several object nouns, are linked to a set of 3D objects without clear associations from specific sentences/noun phrases to objects.
- **Session-level many-to-one grounding**: Multiple sentences in one session, where each sentence can describe several objects (targets and landmarks), are associated with one 3D object.
- **Noun-level one-to-one grounding**: Each noun phrase in each sentence is explicitly matched with one 3D object.

B. Implementation of Our Model

B.1. Model input and output demonstration

In Figure 5, we show an example of 3D-GRAND model's input and output on the Grounded Object Reference task. Note how in the "Response", we train the model to generate a ⟨`detailed_grounding`⟩ pair of tags to densely ground every single object mentioned in the refer expression after generating the grounding answer in ⟨`refer_expression_grounding`⟩. The "ground first" "ground later" in Table 6 means whether the ⟨`detailed_grounding`⟩ tags happen before or after the ⟨`refer_expression_grounding`⟩ tags. Figure 5 is an example of "ground later", and Figure 6 shows an example of "ground first".

B.2. Training Data

There are two flavors of models that we fine-tuned: one grounded object reference model, and one grounded QA model. The grounded object reference model was trained using the grounded object reference data on 3D-FRONT train split, which consist of 234,791 3D-text pairs, each of which are densely grounded. This model was used to generate the ScanRefer results presented in Table 5, 6, and Figure 4 The grounded QA model was trained using a subset of 200k grounded QA: object existence data from the 3D-FRONT train split. The reason that we select a subset of 200k QAs is simply because the entire grounded QA dataset is too large and we do not have enough resource to train on all data. However, as shown in Table 3 and Figure 4, we find even such a subset is very effective in reducing hallucinations in 3D-LLMs.

We provide official data splits of train, val and test (90%, 5%, 5%) in our dataset release. The val and test proportion might seem small, but given our dataset's million-scale, they should be sufficiently large for any development and evaluation purposes.

B.3. Training Details

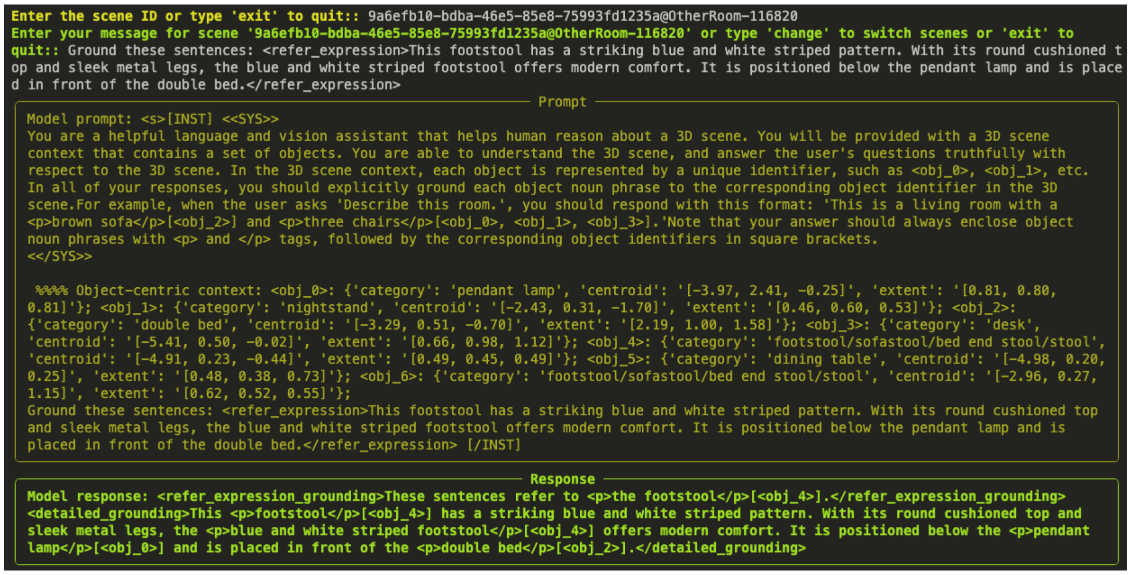
The two flavors of model mentioned above are LoRA-finetuned [29] based off Llama-2. We use DeepSpeed ZeRO-2 [57] and FlashAttention [17] to save GPU memory and speed up training. The model is trained in BF16 precision on 12 NVIDIA A40 GPUs with a combined batch size of 96 and a learning rate of 2e-4. We use the AdamW [47] optimizer with a weight decay of 0.01 and a cosine learning rate scheduler. We train the mode for 10k steps, which takes approximately 48 hours.

C. Additional 3D-GRAND Data Collection

C.1. Point Cloud Generation Pipeline for 3D-Front

Here, we present an expanded version of Section 4, focusing on the methodologies employed in the collection and cleaning of 3D scenes, specifically detailing our process for deriving 3D point clouds from existing datasets.

In our workflow with 3D-FRONT, layouts and meshes are initially processed in Blender to produce multi-view

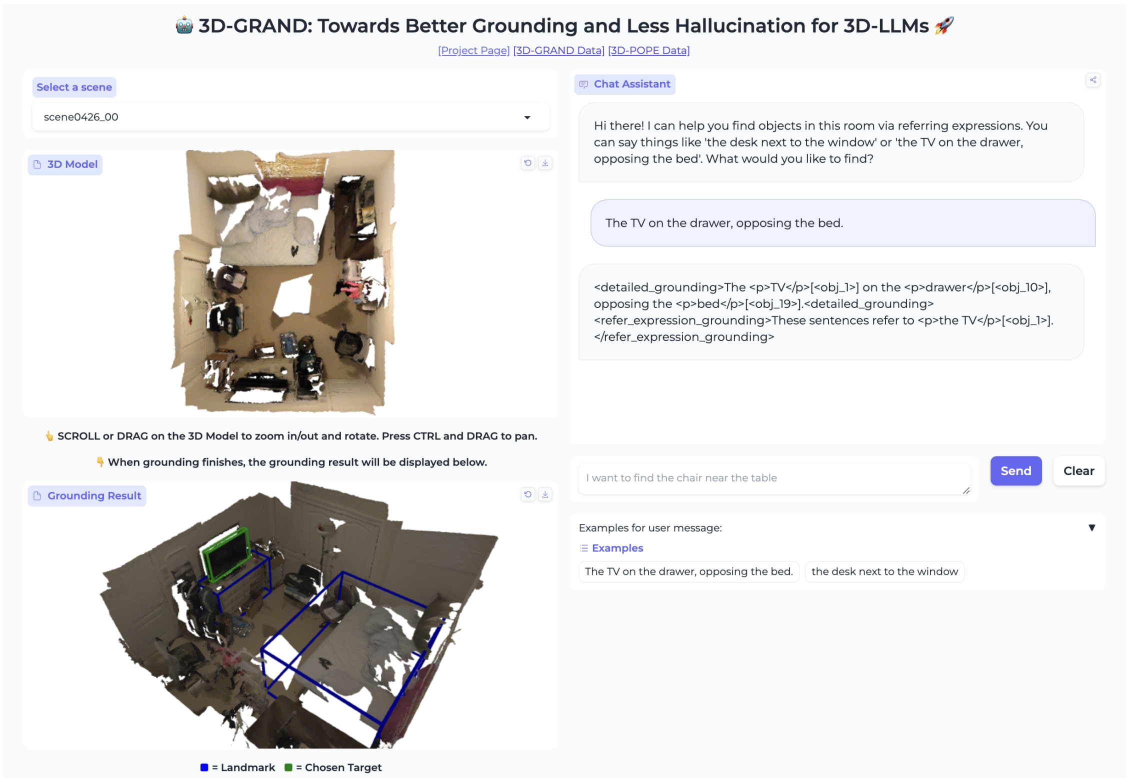


```
Enter the scene ID or type 'exit' to quit:: 9a6efb10-bdba-46e5-85e8-75993fd1235a@OtherRoom-116820
Enter your message for scene '9a6efb10-bdba-46e5-85e8-75993fd1235a@OtherRoom-116820' or type 'change' to switch scenes or 'exit' to
quit:: Ground these sentences: <refer_expression>This footstool has a striking blue and white striped pattern. With its round cushioned t
op and sleek metal legs, the blue and white striped footstool offers modern comfort. It is positioned below the pendant lamp and is place
d in front of the double bed.</refer_expression>
                                                        Prompt
  Model prompt: <s>[INST] <<SYS>>
  You are a helpful language and vision assistant that helps human reason about a 3D scene. You will be provided with a 3D scene
  context that contains a set of objects. You are able to understand the 3D scene, and answer the user's questions truthfully with
  respect to the 3D scene. In the 3D scene context, each object is represented by a unique identifier, such as <obj_0>, <obj_1>, etc.
  In all of your responses, you should explicitly ground each object noun phrase to the corresponding object identifier in the 3D
  scene.For example, when the user asks 'Describe this room.', you should respond with this format: 'This is a living room with a
  <p>brown sofa</p>[<obj_2>] and <p>three chairs</p>[<obj_0>, <obj_1>, <obj_3>].'Note that your answer should always enclose object
  noun phrases with <p> and </p> tags, followed by the corresponding object identifiers in square brackets.
  <</SYS>>

   %%%% Object-centric context: <obj_0>: {'category': 'pendant lamp', 'centroid': '[-3.97, 2.41, -0.25]', 'extent': '[0.81, 0.80,
  0.81]'}; <obj_1>: {'category': 'nightstand', 'centroid': '[-2.43, 0.31, -1.70]', 'extent': '[0.46, 0.60, 0.53]'}; <obj_2>:
  {'category': 'double bed', 'centroid': '[-3.29, 0.51, -0.70]', 'extent': '[2.19, 1.00, 1.58]'}; <obj_3>: {'category': 'desk',
  'centroid': '[-5.41, 0.50, -0.02]', 'extent': '[0.66, 0.98, 1.12]'}; <obj_4>: {'category': 'footstool/sofastool/bed end stool/stool',
  'centroid': '[-4.91, 0.23, -0.44]', 'extent': '[0.49, 0.45, 0.49]'}; <obj_5>: {'category': 'dining table', 'centroid': '[-4.98, 0.20,
  0.25]', 'extent': '[0.48, 0.38, 0.73]'}; <obj_6>: {'category': 'footstool/sofastool/bed end stool/stool', 'centroid': '[-2.96, 0.27,
  1.15]', 'extent': '[0.62, 0.52, 0.55]'};
  Ground these sentences: <refer_expression>This footstool has a striking blue and white striped pattern. With its round cushioned top
  and sleek metal legs, the blue and white striped footstool offers modern comfort. It is positioned below the pendant lamp and is
  placed in front of the double bed.</refer_expression> [/INST]

                                                        Response
  Model response: <refer_expression_grounding>These sentences refer to <p>the footstool</p>[<obj_4>].</refer_expression_grounding>
  <detailed_grounding>This <p>footstool</p>[<obj_4>] has a striking blue and white striped pattern. With its round cushioned top and
  sleek metal legs, the <p>blue and white striped footstool</p>[<obj_4>] offers modern comfort. It is positioned below the <p>pendant
  lamp</p>[<obj_0>] and is placed in front of the <p>double bed</p>[<obj_2>].</detailed_grounding>
```

Figure 5. 3D-GRAND model input and output on Grounded Object Reference task.



3D-GRAND: Towards Better Grounding and Less Hallucination for 3D-LLMs

[Project Page] [3D-GRAND Data] [3D-POPE Data]

Select a scene
scene0426_00
3D Model
SCROLL or DRAG on the 3D Model to zoom in/out and rotate. Press CTRL and DRAG to pan.
When grounding finishes, the grounding result will be displayed below.
Grounding Result
■ = Landmark ■ = Chosen Target
Chat Assistant
I want to find the chair near the table
Send
Clear
Examples for user message:
Examples
The TV on the drawer, opposing the bed.
the desk next to the window

Hi there! I can help you find objects in this room via referring expressions. You can say things like 'the desk next to the window' or 'the TV on the drawer, opposing the bed'. What would you like to find?

The TV on the drawer, opposing the bed.

<detailed_grounding>The <p>TV</p>[<obj_1>] on the <p>drawer</p>[<obj_10>], opposing the <p>bed</p>[<obj_19>].<detailed_grounding>
<refer_expression_grounding>These sentences refer to <p>the TV</p>[<obj_1>].
</refer_expression_grounding>

Figure 6. Demo of interactive chat interface with the 3D-GRAND model.

images. These images are subsequently used to construct comprehensive point clouds for entire houses. Both point clouds and per-room meshes are utilized to generate scene-level point clouds. We avoid direct use of room meshes because they lack color information in ceilings, walls, and floors, necessitating the final output to be a point cloud.

For Structure3D, while per-scene multi-view images facilitate direct rendering of per-scene point clouds, we frequently encounter issues where parts of adjacent scenes are inadvertently reconstructed due to window transparency. To address

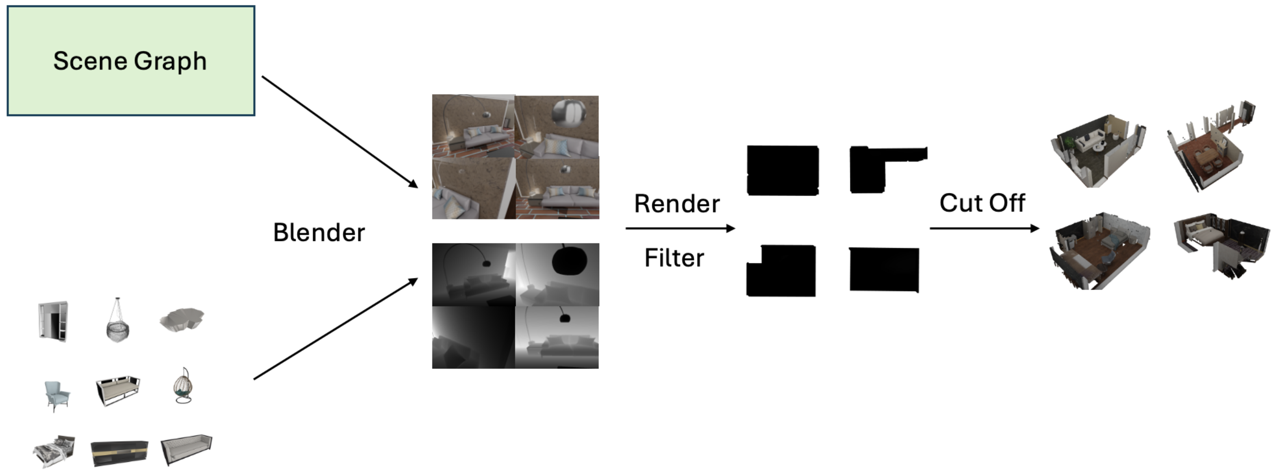


Figure 7. Point Cloud Generation for 3D-FRONT.

this, we employ the layout of each scene to trim extraneous points, thus enhancing the precision of the resulting point clouds.

D. Additional 3D POPE results

D.1. 3D Pope Results on NYU40

Table 9 presents evaluation results for 3D POPE using the NYU40 class set. NYU40 includes a subset of the classes from ScanNet200 featured in the main results table. The NYU40 class set consolidates many fine-grained classes into an "other" category, potentially reducing the challenge of negative sampling in the *Popular* and *Adversarial* settings compared to the ScanNet200 scenario.

Dataset	3D-POPE	Model	Accuracy	Precision	Recall	F1 Score	Yes (%)
ScanNet Val (NYU40)	*Random*	3D-LLM	50.00	50.00	100.00	66.67	100.00
		3D-VisTA	50.12	50.08	77.13	60.73	77.01
		LEO	54.03	52.70	78.52	63.07	74.50
		Ours zero-shot (No Grounding)	86.45	87.26	85.36	86.30	48.91
		Ours zero-shot (Grounding)	85.68	88.22	82.34	85.18	46.67
	Popular	3D-LLM	50.00	50.00	100.00	66.67	100.00
		3D-VisTA	50.27	50.23	77.13	60.84	76.91
		LEO	48.86	49.28	77.44	60.23	78.58
		Ours zero-shot (No Grounding)	80.85	78.30	85.35	81.68	54.50
		Ours zero-shot (Grounding)	81.69	81.32	82.28	81.80	50.59
	Adversarial	3D-LLM	50.00	50.00	100.00	66.67	100.00
		3D-VisTA	50.44	50.48	77.14	61.03	76.86
		LEO	49.77	49.85	77.67	60.73	77.91
		Ours zero-shot (No Grounding)	81.47	78.98	85.78	82.24	54.31
		Ours zero-shot (Grounding)	82.10	81.72	82.72	82.22	50.61

Table 9. Results of 3D-LLMs under three evaluation settings of 3D-POPE on the validation set of ScanNet using NYU40 class set. Yes denotes the proportion of answering "Yes" to the given question. The best results in each block are denoted in bold.

E. Human Validation

Because our dataset generation process involves GPT-4V, there is a potential for hallucinations. We identify three types of possible hallucinations that could impact our dataset: the text might inaccurately describe an object's property, such as color or size (termed incorrect object attribute); it might incorrectly depict the spatial relationship between two objects (termed incorrect spatial relation); or it might describe an object that does not exist in the referenced scene at all (termed incorrect object existence). Additionally, inaccuracies in our dataset may also arise from incorrectly grounding the wrong object.

To validate our dataset against these potential failures, we plan to verify a subset of our data through crowdsourcing to ascertain the frequency of these failure cases.

E.1. Crowd-sourcing

We crowd-source the validation of annotations using Hive, a platform commonly used for sourcing annotations for computer vision tasks. The platform can be accessed at https://thehive.ai/.

We conceptualize our dataset validation as a data annotation problem, employing scene-text pairs as the data unit. Annotators are instructed to label these pairs as "True" or "False" to indicate the presence or absence of hallucinations or inaccuracies. Additionally, a "Cannot Decide" option is provided to accommodate cases where the scene view is unclear.

E.1.1. Task Generation

Hive only supports presenting static images to annotators, so we generate annotation tasks by composing snapshots of a scene with corresponding text annotations. For each task, we take snapshots from four different angles and pair them with a corresponding annotation. To maintain simplicity and conciseness, we require validation of just one sentence per

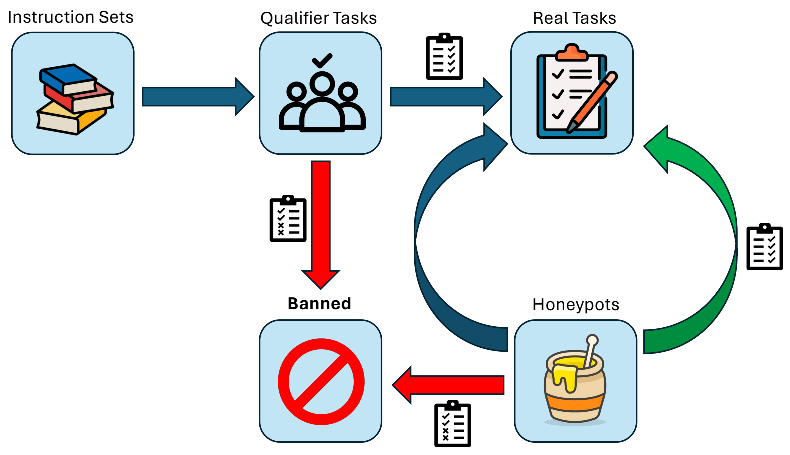
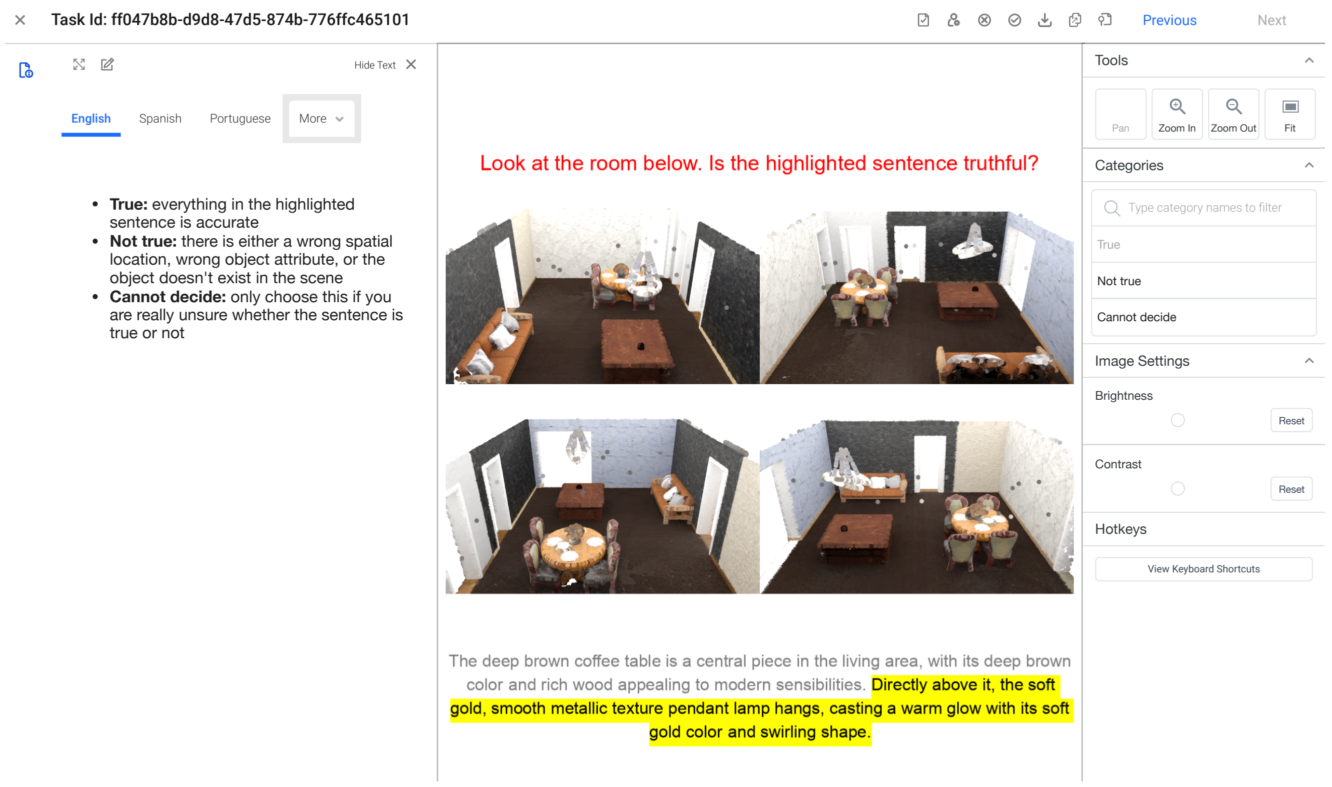


Figure 8. Illustration of the crowd-sourcing process. Annotators are first shown instruction sets that describe both how the task should be completed and the possible inaccuracies that could appear in the data. They are then presented with qualifier tasks, and annotators who do not get a high enough accuracy on these tasks are banned from annotating our dataset. Annotators who pass the qualifier are able to annotate real tasks, but are randomly presented with honeypots that are indistinguishable from real tasks. Annotators who do not get a high enough accuracy on honeypots are also banned from our dataset.



- **True:** everything in the highlighted sentence is accurate
- **Not true:** there is either a wrong spatial location, wrong object attribute, or the object doesn't exist in the scene
- **Cannot decide:** only choose this if you are really unsure whether the sentence is true or not

Look at the room below. Is the highlighted sentence truthful?

The deep brown coffee table is a central piece in the living area, with its deep brown color and rich wood appealing to modern sensibilities. Directly above it, the soft gold, smooth metallic texture pendant lamp hangs, casting a warm glow with its soft gold color and swirling shape.

Figure 9. Example of a dataset validation task presented to crowd-sourcing annotators: It displays a scene from four different angles alongside the sentence to be validated, which is highlighted. Annotators have the options to select "True," "Not True," or "Cannot Decide." On the left side of the screen, the instructions are repeated for annotators' reference. In this example, "True" is selected.

task, providing some surrounding context and highlighting the target sentence. For grounding validation, the grounded object is outlined in the scene with a bounding box, and the referring phrase in the sentence is emphasized. An example of such a task, along with the annotation interface, is depicted in Figure 9. Figure 10 displays two text validation tasks and two grounding validation tasks that were presented to annotators.

E.1.2. Crowd-sourcing Validity

Validating a dataset necessitates a high level of attention from annotators. We curate sets of instructions, qualifying tasks, and honeypot tasks to ensure that the annotations obtained from crowdsourcing are reliable. The crowdsourcing process is illustrated in Figure 8.

Before presenting any tasks to the workers, we present them with a set of instruction tasks that show an example annotation, the correct response (as determined by us), and the reason why that response is correct. They are paired with an incorrect example and an explanation of why it is incorrect in order to ensure unbiased annotations. Examples of qualifying instructions are shown in 11. These instructions are intentionally brief, as we found through trial-and-error that longer, paragraph-based instructions were largely ignored by annotators.

Qualifying tasks are presented to the annotators before they are shown any real tasks in order to train them to complete the real task with a high accuracy. Annotators are both shown the correct answer and a reasoning as to why it is correct for every qualifier. We set the minimum qualifier accuracy to 0.75 to ensure that annotators must achieve a minimum competency before annotating real tasks. Every dataset is given between 12 and 30 specially crafted qualifying tasks that demonstrate the possible inaccuracies that could appear in the data. These qualifiers are divided equally between true and false examples so as not to bias workers towards any one answer.

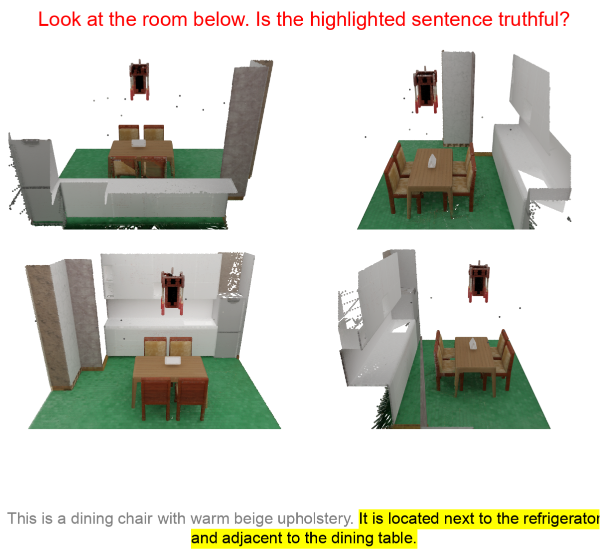
Honeypot tasks are randomly mixed in with real tasks in order to ensure that annotators are maintaining a high quality of annotations throughout the entire job. Because we annotate the honeypot tasks before showing them to annotators, we are able to evaluate any given worker's accuracy on honeypot tasks. We set the minimum honeypot accuracy to 0.89 to ensure that annotators are maintaining correct annotations. Workers that do not maintain this accuracy are banned from annotating our tasks. This is higher than the required accuracy for qualifiers because we expect annotators to already be well trained in our annotation tasks from the instructions and qualifiers. Every data type is given between 18 and 35 honeypot tasks. The honeypots are also approximately divided equally between true and false examples so that workers who consistently select a single answer without paying attention to the task (e.g., someone who always selects "True") will be banned.

To further ensure high-quality annotations, we send each question to 3 different annotators and only accept an annotation if at least 2 out of the 3 annotators agree with each other on the truthfulness of an item. If agreement is not reached, the task is returned as inconclusive.

E.2. Results

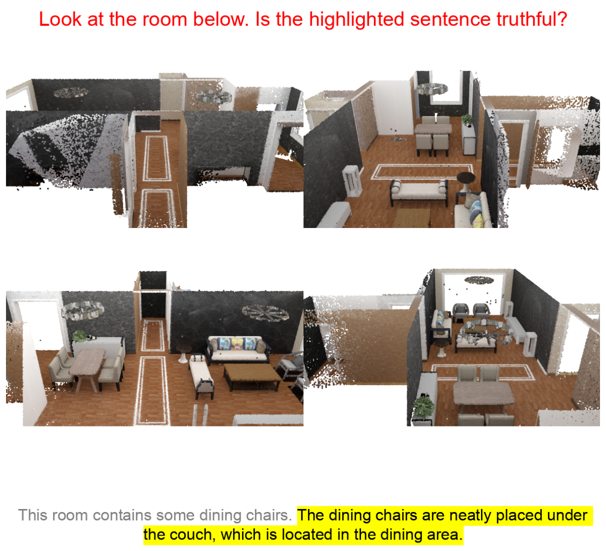
We perform validation on 10,200 room-annotation pairs. From each of the three data types, 1,700 pairs are sampled for validation of both text truthfulness and grounding accuracy. A subset of 800 rooms is uniformly chosen, with 400 designated for text truthfulness and another 400 for grounding accuracy. The text data is uniformly sampled from these rooms. We report accuracies for both text truthfulness and grounding accuracy in Table 10.

We report comprehensive statistics from the annotation process in Table 11. We observe a very low qualifier pass rate ranging from 11 - 20 % across the different tasks in our data, suggesting that our qualifiers were effective in allowing only the most attentive annotators qualify to annotate real tasks. In addition, none of these annotators were banned due to honeypots. This increases our confidence that our qualification process is effective in training annotators and filtering out those who were not attentive. We also observe that workers spend roughly the same time on real tasks and honeypot tasks, suggesting that the honeypots are indistinguishable from real tasks for the annotators. This further supports the validity of our annotations.


Look at the room below. Is the highlighted sentence truthful?

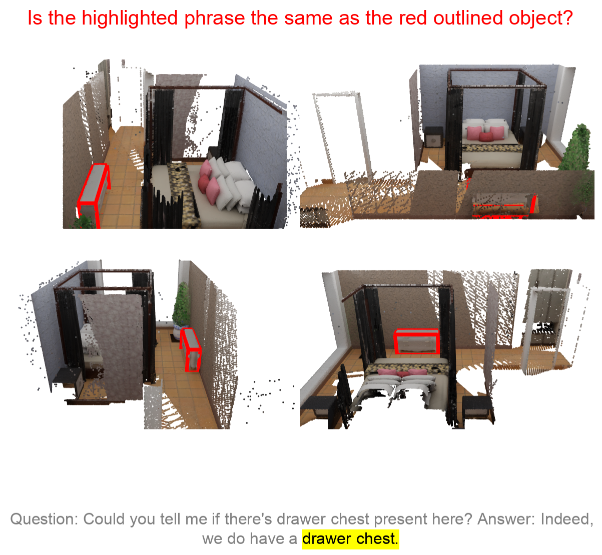
This is a dining chair with warm beige upholstery. It is located next to the refrigerator and adjacent to the dining table.

(a) "True" example of a text validation task.


Look at the room below. Is the highlighted sentence truthful?

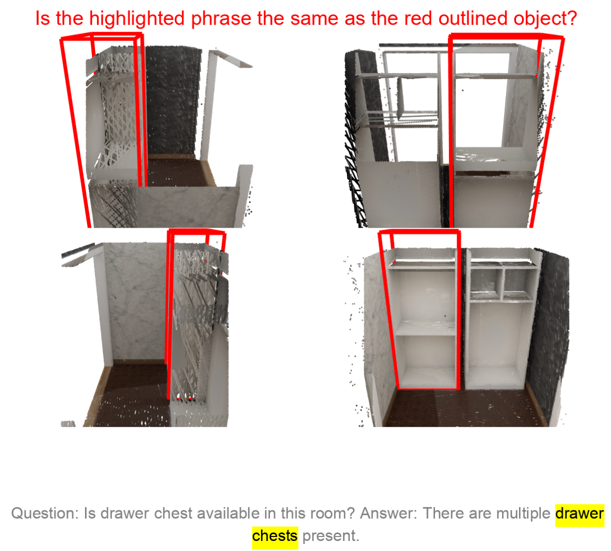
This room contains some dining chairs. The dining chairs are neatly placed under the couch, which is located in the dining area.

(b) "False" example of a text validation task.


Is the highlighted phrase the same as the red outlined object?

Question: Could you tell me if there's drawer chest present here? Answer: Indeed, we do have a drawer chest.

(c) "True" example of a grounding validation task.


Is the highlighted phrase the same as the red outlined object?

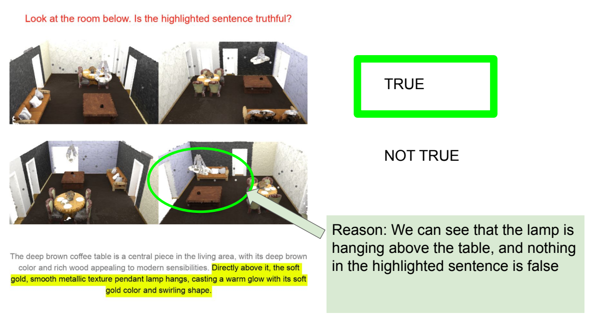
Question: Is drawer chest available in this room? Answer: There are multiple drawer chests present.

(d) "False example of a grounding validation task.

Figure 10. Examples of tasks presented to annotators for validating both text accuracy and grounding accuracy. The instruction is displayed at the top of the task, while the center showcases four different views of the scene to ensure comprehensive coverage of all relevant areas. At the bottom, the annotation is presented with the pertinent section highlighted.

Table 10. Text Truthfulness and Grounding Accuracy from crowdsourcing. Accuracy is computed by dividing the number of "True" responses by the total number of tasks (1700).

Method	**Text Truthfulness**	**Grounding Accuracy**
Grounded Scene Description	0.877	0.944
Grounded QA	0.852	0.956
Grounded Object Reference	0.863	0.918



Look at the room below. Is the highlighted sentence truthful?

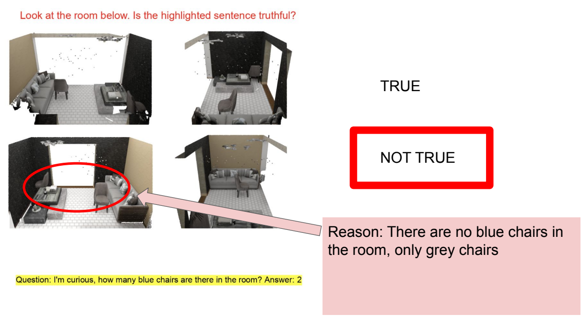
TRUE

NOT TRUE

Reason: We can see that the lamp is hanging above the table, and nothing in the highlighted sentence is false

The deep brown coffee table is a central piece in the living area, with its deep brown color and rich wood appealing to modern sensibilities. Directly above it, the soft gold, smooth metallic texture pendant lamp hangs, casting a warm glow with its soft gold color and swirling shape.

(a) "True" example instruction for the text validation task.



Look at the room below. Is the highlighted sentence truthful?

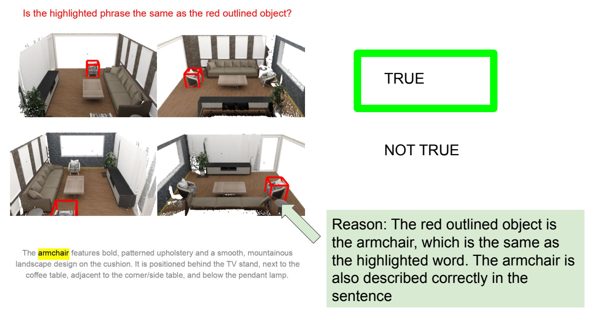
TRUE

NOT TRUE

Reason: There are no blue chairs in the room, only grey chairs

Question: I'm curious, how many blue chairs are there in the room? Answer: 2

(b) "False" example instruction for the text validation task.



Is the highlighted phrase the same as the red outlined object?

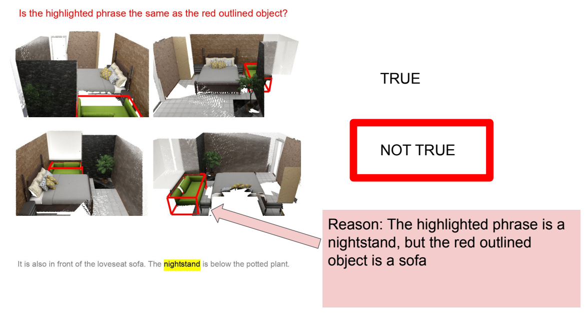
TRUE

NOT TRUE

Reason: The red outlined object is the armchair, which is the same as the highlighted word. The armchair is also described correctly in the sentence

The armchair features bold, patterned upholstery and a smooth, mountainous landscape design on the cushion. It is positioned behind the TV stand, next to the coffee table, adjacent to the corner/side table, and below the pendant lamp.

(c) "True" example instruction for the grounding validation task.



Is the highlighted phrase the same as the red outlined object?

TRUE

NOT TRUE

Reason: The highlighted phrase is a nightstand, but the red outlined object is a sofa

It is also in front of the loveseat sofa. The nightstand is below the potted plant.

(d) "False" example instruction for the grounding validation task.

Figure 11. Examples of instructions presented to annotators before they are shown any actual tasks for annotation. For every possible kind of hallucination (incorrect object attribute, spatial relation, or object existence), an illustrative positive and negative example are presented in order to instruct the annotator to look for all possible failure cases.

Table 11. Comprehensive annotation metrics. Includes qualifier pass rate, honeypot count, honeypot ban rate, percent of tasks marked inconclusive (where workers could not come to an agreement on the label), and the average time that workers spend on both real tasks and honeypot tasks. Each dataset was evaluated on 1700 annotations. At least 2 workers must agree on the label for an annotation to be considered valid.

Category	**Type**	**% Qualifier Pass Rate**	**# Honeypots**	**% Honeypot Ban Rate**	**% of Inconclusive Tasks**	**Avg. Real Task Speed (s)**	**Avg. Honeypot Speed (s)**
		(Pass)	(Total)	(Ban)	(Tasks)	(Real)	(Honeypot)
Text Accuracy	Scene Description	17	35	0	2.03	17.91	17.08
	QA	19	18	0	1.35	16.22	16.64
	Object Reference	10	38	0	0.82	19.88	17.57
Grounding Accuracy	Scene Description	20	18	0	1.66	9.69	12.84
	QA	16	18	0	1.11	6.65	11.14
	Object Reference	20	18	0	1.88	9.69	12.84