R2D3: Imparting Spatial Reasoning by Reconstructing 3D Scenes from 2D Images

Arijit Ray[1] Dina Bashkirova[1] Reuben Tan[1] Kuo-Hao Zeng[2]
Bryan A. Plummer[1] Ranjay Krishna[2,3] Kate Saenko[1]

[1]Boston University, [2]Allen AI, [3]University of Washington

Abstract

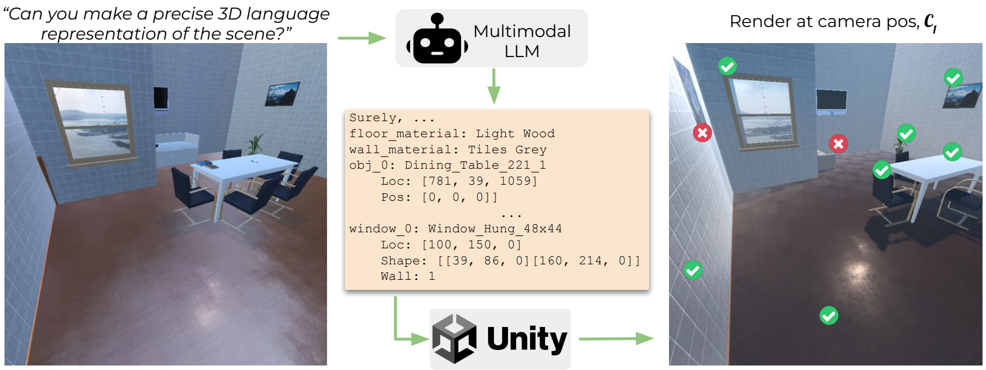
Multimodal large language models (MLMs), which have been widely adopted due to their impressive commonsense reasoning on 2D images, still struggle with 3D spatial reasoning. The research community lacks holistic evaluations that enable an analysis into what training data imparts 3D spatial capabilities. Existing benchmarks probe spatial understanding using only coarse-level spatial awareness (e.g. is something to the left or right of something else), or by predicting a bounding box for an object query. Hence, we propose R2D3, a holistic "analysis by synthesis" evaluation benchmark; R2D3 tasks an MLM to represent a 2D image as a set of semantic assets with precise 3D locations and pose such that a graphics engine can reconstruct the 3D scene. This task requires the model to have a comprehensive understanding of the elements that make up the scene and their precise 3D relative locations. Our benchmark includes 12K indoor scenes in the AI2THOR environment and is compatible with several downstream applications such as embodied AI, spatial reasoning, and navigation tasks. Using our benchmark, we explore finetuning techniques that encourage spatial reasoning. Surprisingly, we find that conventional fine-tuning on the training set of our benchmark, while enough to understand semantics, is not enough to learn the precise 3D locations and poses of the objects in a scene. However, conveying the camera-scene orientation by marking a point in the image and including its 3D coordinate in text during training allows the model to improve 3D spatial estimation at test time. We hope that the R2D3 benchmark will help drive progress in exploring design choices that improve the precise 3D spatial understanding of MLMs.

1 Introduction

Humans maintain a steady awareness of the 3D orientation of space around them to operate in the real world [78, 73]. Despite their widespread adoption, multimodal language models (MLMs) [2, 127, 67], with their impressive language and vision commonsense capabilities, still struggle to reason spatially as concluded by numerous studies [14, 45, 6]. There is a limited evaluation of what imparts 3D spatial reasoning skills to these models. Most 3D benchmarks focus on comprehending a 3D scan [41, 7, 9] while most MLMs today only accept image inputs and not 3D representations. Existing spatial tests with 2D images focus on bounding box prediction [126, 99, 9] or spatial relationship prediction [72, 34] or distance prediction [14] between two objects. Although a coarse-level spatial reasoning (left, right, behind, etc) evaluation set exists [34], there is no training data to study what factors impart precise pixel-level spatial reasoning into MLMs.

We propose a benchmark, R2D3 (Figure 1), where an MLM is tasked to represent a 2D input image such that a graphics simulator can reconstruct the 3D input scene. This task draws inspiration from fundamental schools of thought in computer vision–analysis by synthesis [120] and inverse rendering [55, 10]–that suggest that a precise reconstruction requires the model to have a comprehensive

Preprint. Under review.


"Can you make a precise 3D language representation of the scene?"
Multimodal LLM
Unity
Render at camera pos, C_l

```
Surely, ...
floor_material: Light Wood
wall_material: Tiles Grey
obj_0: Dining_Table_221_1
    Loc: [781, 39, 1059]
    Pos: [0, 0, 0]]
                   ...
window_0: Window_Hung_48x44
    Loc: [100, 150, 0]
    Shape: [[39, 86, 0][160, 214, 0]]
    Wall: 1
```

Figure 1: The task of the model is to precisely describe a 3D scene using the object names, attributes, and 3D locations that can be rendered in a graphics engine to reflect the image. We use this task to evaluate a multimodal language model's ability to understand the 3D nature of the scene holistically.

understanding of the elements that make up the scene. Precisely, the task expects the MLM to generate a structured output containing a collection of object and scene concepts with their 3D positions. The concepts are names of assets (objects, walls, windows, etc.).

Our benchmark is generated using 12K indoor rooms from ProcTHOR [24], compatible with the AI2THOR [1] simulator, providing support for downstream embodied AI applications such as object manipulation, navigation, and task completion [97]. Each sample in R2D3 is a room, represented with the graphics program that generates it and a 2D image view of the room. To drive further research in image, text, and 3D understanding, we also generate natural language descriptions of the rooms and each of the object assets using GPT4-V [2]. R2D3 is larger than existing open-source evaluation benchmarks for probing 3D spatial reasoning in MLMs [34], and also includes a training set. Furthermore, since our data curation is based on procedural generation, it does not require human annotation, can be scaled up with more assets, and extended to reason about interactive tasks.

Using R2D3, we explore finetuning techniques that encourage 3D spatial reasoning in MLMs. We focus on techniques that require no change to the MLM's architecture, allowing for an easier low-resource adaptation rather than large-scale retraining of the LLM, which is infeasible for most researcher labs due to resource limitations. We adopt standard metrics from localization and adapt them to our task. Specifically, we measure the L2 distance for object placement accuracy and use standard classification accuracies for semantics. We adopt the widely-used open-sourced LLaVA 1.5-13B [67] model for our experiments. We explore two scenarios during training, one where the camera orientation is not precisely known, and the MLM must infer it from the image, and another where we know the exact camera position and orientation during training.

Our results find that LLaVA struggles to estimate the camera position and orientation if this data is not available in its training data. Since recent work shows the image encoder [27] in LLaVA lacks 3D depth capabilities [17, 8], we overlay a depth estimation mask, generated from an off-the-shelf depth estimation model [114]. We see that this significantly improves LLaVA's ability to learn to orient the scene properly when camera information is not present during training. If camera information is present, we investigate the best way to convey it to the MLM during training. Since MLLMs primarily operate on the quality of the visual or language prompt, we explore visual and language prompting. We find that a simple visual prompting strategy, i.e., marking a point in the image with the 3D location specified in language - is more effective than specifying the entire camera orientation in language. This suggests that LLaVA is better at comprehending a visual mark to calculate the 3D orientation than being able to calculate it from language alone. We see that our best adaptation reaches only 74% accuracy in placing objects correctly and 51% pose accuracy, suggesting considerable room for improvement using our benchmark.

2 Related work

Analysis by Synthesis. Our proposed R2D3 task is focused on evaluating the capability of MLMS to reason spatially about objects in a 3D environment. This falls under the general task of analysis

Dataset	Precise 3D	Real Apps	Captions/QA	Interactivity
2D Grounding Datasets [95, 65, 57]	No	Yes	No	No
3D Grounding Datasets [99, 126, 9]	Yes	Yes	No	No
ScanQA [7]	No	Yes	Yes	No
Neural De-render [113]	Yes	No	No	Yes
Spatial VLM [14]	No	Yes	Yes	No
Open EQA [72]	No	Yes	Yes	No
BLINK [34]	No	Yes	Yes	No
InverseRenderLLM [58]	Yes	No	No	Yes
R2D3 (Ours)	Yes	Yes	Yes	Yes

Table 1: Comparison of existing benchmark tasks to ours. Our dataset is aligned with AI2THOR, which enables interactive real applications like embodied AI, navigation, manipulation, as opposed to operating on simplistic domains like CLEVR or clipart. We have both precise 3D locations, pose and room layout as well natural language semantic captions and descriptions of each object. Our dataset requires no human annotation compared to many existing datasets.

by synthesis [79, 37], which aims to explain observed data such as an image using a set of physical variables. Related tasks include shape estimation of objects from images [70, 87, 101], pose estimation [125, 124, 102], multi-object scene recovery [35, 96, 26] and primitive reconstruction [105, 104, 86, 83, 85, 25, 56, 77, 106, 49, 48, 61, 38]. Based on the idea that a generative model can describe how variables produce the data, existing approaches [58, 113] propose to evaluate a model's capability to understand an image by generating an interpretable representation of it. While R2D3 is similar in nature to [113], our task focuses on images of indoor scenes instead of clip art which makes it more useful for downstream embodied AI and robotics tasks. Additionally, we are the first work to evaluate MLMs on their 3D spatial reasoning capabilities on the more application-oriented ProcThor dataset. In contrast, [58] investigate on the CLEVR [53] dataset, which contains simplified visual elements.

3D Understanding and Generation. The task of holistic 3D scene understanding [39, 74, 21, 92] requires the accurate generation of object entities along with the 3D scene layout. State-of-the-art approaches are often focused on reconstructing objects of arbitrary shapes [121, 66, 36] as well as segmentation maps [116, 81, 19, 60, 80]. Such approaches are similar in spirit to recent 3D generative approaches, including [76, 54]. Existing approaches have also widely addressed other aspects of 3D understanding including object localization [88, 94], dense segmentation [11, 13, 46, 68, 93, 109, 50, 42, 54, 75, 52, 89, 108, 123] and tracking [4, 62, 64]. Our R2D3 task also bears strong similarities to existing joint 3D and language understanding tasks including but not limited to fine-grained scene captioning [16], open-vocabulary classification and localization [15, 42, 3, 31, 47, 40] and question answering [118, 7]. Unlike ScanQA [7], which addresses semantic question answering for 3D scans, our framework emphasizes understanding the 3D structure from 2D images. In contrast to existing benchmarks [126, 99, 9], which only focus on object localization within scenes, R2D3 is built off the AI2Thor engine to enable alignment with multiple interactive applications [23, 112, 22]. More importantly, our benchmark does not require any human annotations and can be arbitrarily scaled up to include more assets and tasks. Compared to LayoutNet's focus on estimating room layout from panoramas [128], R2D3 evaluates a model's understanding of the orientation of all objects and not just the scene layout. In light of potential applications in embodied AI, we focus on adapting models to perform *precise* spatial reasoning under low-resource settings. This differs from existing work such as BLINK [34], which address *coarse* spatial reasoning and Cube-LLM [17], which does large-scale pretraining. Generation of 3D scenes has also been widely explored with input conditions of different modalities. For instance, Scenescript [6] generates 3D scenes from videos while Atiss [84] autocompletes scenes based on room types and floor plans. In contrast, our task synthesizes scenes based on single images while focusing on evaluative measures. Last but not least, R2D3 diverges from rule-based systems like WordsEye [18] and SceneSeer [12] and avoids reliance on purely text-driven or heavily engineered systems such as SceneCraft [59], Holodeck [115] and Ctrl-Room [30].

Vision and Language Models. Our task has also been heavily influenced by the emergence of multimodal foundation models [90, 51, 107, 119, 33, 111, 117]. To leverage the real world knowledge in LLMs and their generative capabilities, recent approaches have proposed to adapt pretrained visual encoders with LLMs for a wide range of downstream image [103, 28, 127, 64, 5, 20] and video [122, 71, 100, 110, 63, 69] understanding tasks. These MLMS have demonstrated impressive zero-shot results on downstream tasks. Recent work [32, 42, 17] have also proposed to advance 3D understanding by augmenting LLMs to reason about 3D scenes. Adjacent to finegrained semantic analysis of vision-language models [91, 43], we focus on precise 3D perception.

3 Approach

Our goal is to explore techniques during tuning that encourage multimodal language models (MLMs) to reason about the 3D nature of 2D images. Hence, we propose a testbed, R2D3, based on the task of 2D to 3D reconstruction where researchers can tune and test MLMs to discover strategies that lead to better 3D reasoning. In R2D3, given a 2D image, an MLM is tasked to predict a precise graphics description listing the entities that make up the scene such that the 3D scene can be reconstructed using a graphics engine. Therefore, to construct our testbed, we need tuples of a 2D scene, the corresponding 3D environment, and the graphics program that generates it. Compared to existing 2D to 3D estimation benchmarks, our testbed is built using physics engines and hence, requires no human supervision while remaining accurate. Further, it allows scaling up to arbitrarily more assets and to interact and tweak the environment, which is not possible with existing benchmarks.

Below, we first describe our format for the precise graphics description (a graphics program) that represents a 3D scene using entities that construct it. Next, we outline our data curation process, where we obtain paired data of the 3D graphics description for a 2D image. Finally, since MLLMs operate based on prompts, we use our testbed to analyze visual vs language prompting baselines that improve 3D reasoning in a state-of-the-art multimodal language model, LLaVA [67].

3.1 3D Graphics Program as Scene Representation

While 3D scenes can be modeled in various ways (meshes, point clouds, nerfs [76]), a graphics program-based representation has a few key advantages when it comes to evaluating MLMs: i) interpretability, i.e. each object, its attribute, and location can be analyzed; ii) the semantic and spatial understanding of the scene is disentangled with the mesh quality and the language model doesn't need to care about mesh quality; iii) compactness, i.e. a text description is more memory efficient that meshes or point clouds and can fit into context length of common MLMs; and iv) a natural language-based representation for MLMs that already operate in the language space to make it more readily usable for other downstream reasoning tasks.

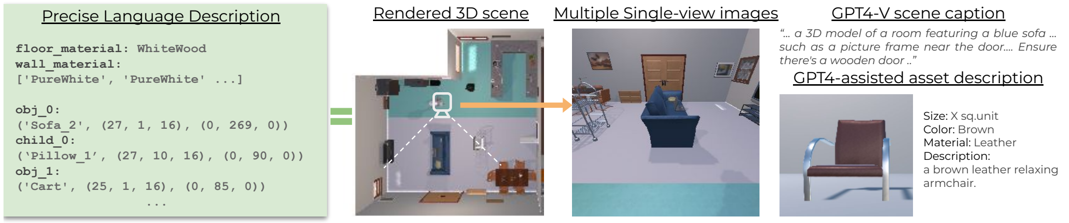
Specifically, we describe a 3D room using a standard YAML-like text file that lists the entities in the scene. For an indoor room, we start with the basic constituents that make the scene- the floor polygon, walls, objects and their children, and the windows and list them in a YAML format as shown in Figure 2 (left most block).

3.2 Generating paired image and 3D graphics program data

Figure 2 shows a summary of our curated dataset. We start with ProcTHOR scenes [24] and randomly take 12K rooms from these apartments. We formulate a light-weight graphics program for the rooms from the JSON representation of the scene in ProcTHOR [24] (see the left-most block in Figure 2). We do not use the JSON directly since it is too long (and redundant and over-parameterized) to fit into the context length for most state-of-the-art MLMs.

We render the 3D room in a graphics simulator, AI2THOR [1] (second left block in Figure 2), and take an image view of the room from the corner with the most objects visible (second right block in Figure 2). Please see the appendix to see how we calculate where to place the camera. Our camera always looks from a corner towards the opposite visible corner in the room. This 2D image view of the room and the corresponding graphics program form the basis of our curated dataset. In addition to the image, we also extract the semantic segmentation map to be of use for further research.

To drive further research in semantic understanding of 3D environments, we also generate natural language captions for the room that describe both semantics and the rough 3D relative locations of objects (rightmost block in Figure 2). Specifically, we caption the corner room view and the top-down view using GPT4-V. The corner room view caption captures the semantics of the image


Precise Language Description
Rendered 3D scene
Multiple Single-view images
GPT4-V scene caption

```
floor_material: WhiteWood
wall_material:
['PureWhite', 'PureWhite' ...]

obj_0:
('Sofa_2', (27, 1, 16), (0, 269, 0))
child_0:
('Pillow_1', (27, 10, 16), (0, 90, 0))
obj_1:
('Cart', (25, 1, 16), (0, 85, 0))
                 ...
```

"... a 3D model of a room featuring a blue sofa ... such as a picture frame near the door.... Ensure there's a wooden door .."
GPT4-assisted asset description
Size: X sq.unit
Color: Brown
Material: Leather
Description:
a brown leather relaxing armchair.

Figure 2: Method of generating our R2D3 benchmark data: We first generate a compact graphics program (left-most block) and render rooms in ProcTHOR [24]. We then render images of the rooms from corners that have the most objects visible. We also caption each of the assets and the overall semantics of the room.

being seen, and the top-down view caption is better at capturing the 3D relative locations of the objects in the scene. We also generate attribute-based phrases for each asset ID, further enabling fine-grained semantic analysis. This also makes the graphics program representation more useful for downstream reasoning and for zero-shot evaluations where a model just needs to predict the generic object class and attributes and not the exact asset ID. While generating the captions, we feed privileged information from the scene graph (such as object name) as a prompt to the GPT4-V to ensure the caption is of high accuracy.

In summary, our dataset contains 12K rooms (with an almost equal distribution of bathrooms, living rooms, bedrooms, and kitchens) composed of 996 object assets for 172 object classes, 14 kinds of windows, and 178 different wall and floor materials. On average, there are around 8 major objects and 5 children objects per room. Using MLMs' adaptation performance on this generated dataset, we wish to explore tuning strategies that lead to better 3D spatial and semantic performance.

4 Experiments

Using the R2D3 task we described above, we investigate tuning techniques that encourage accurate 3D spatial reasoning in a widely used MLM, LLaVA [67]. At test time, we only focus on the model's ability to estimate the 3D relative locations of objects without having access to the full 3D orientation of the camera. To keep the coordinate scale and range consistent with the GT for easier evaluation, we input the room polygon bounds and the corner position from which the image was taken (this can be thought of as a noisy approximate camera position). The model is then tasked to orient the image (and all objects in it) in that space. This intrinsically also requires the model to estimate the precise camera orientation to accurately align the image in the polygon layout. Specifically, here is the prompt we use at test time:

```
<image> The room polygon is [(x,z)...].  Image taken from corner (x, z) looking
inside the room.  Plausible 3D coordinates (x, y, z) for the image shown:
```

The model is then tasked to predict the graphics program as described in Section 3.1. While training the model for the task, we experiment with designing the multimodal prompt containing additional information that allows accurate estimation of the 3D scene in the MLM. The key intuition behind designing our prompt is to make it easier for the MLM to perform the spatial conversion of the 2D image scene to the 3D coordinates. Specifically, we analyze two scenarios: 1) Can LLaVA learn to estimate 3D without precise camera orientation during training, i.e., just from data alone? This would be useful for training on data with noisy camera information. 2) If we do have precise camera orientation during training, what is the best way to convey it to the MLM? Based on these questions, we describe the adaptation strategies we experiment with below.

4.1 Adaptation Strategies Explored

4.1.1 Learning without precise camera orientation in training

Specifically, we use the image and the prompt specifying the room polygon and only the coordinate from which the picture was taken- the exact angle is not specified, and the model needs to learn to

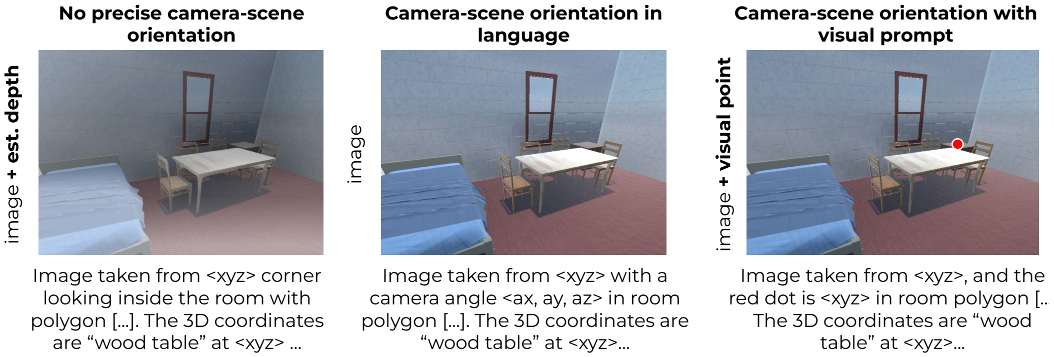

No precise camera-scene orientation
Camera-scene orientation in language
Camera-scene orientation with visual prompt
image + est. depth
image
image + visual point
Image taken from <xyz> corner looking inside the room with polygon […]. The 3D coordinates are "wood table" at <xyz> …
Image taken from <xyz> with a camera angle <ax, ay, az> in room polygon […]. The 3D coordinates are "wood table" at <xyz>…
Image taken from <xyz>, and the red dot is <xyz> in room polygon [.. The 3D coordinates are "wood table" at <xyz>…

Figure 3: Various ways to convey the 3D camera-scene orientation during training. Experiments show that overlaying depth helps when we do not have access to precise camera-scene orientation helps. When we do have precise camera-scene orientation, conveying it by marking a point in the image makes it easier for LLaVA to learn 3D estimation.

approximate how to orient the image it sees to the polygon shape. The model is then tuned to predict the precise 3D locations of all the entities in the scene. This method is denoted as **FT** in the tables.

Since recent work [8] shows ViT [27] in LLaVA lacks 3D perception, we explore whether we can simply convey depth alongside the ViT [27] features using a depth estimation model. Hence, we infer depth using the DepthAnything [114] model and overlay it on the image as shown in Figure 3 (left). The details for how we scale and overlay it can be found in the appendix. We denote this method as **Depth** in the tables.

Finally, as another ablation, recent work [17] suggests that switching out ViT [27] for DinoV2 [82] in LLaVA may improve 3D estimation. Hence, we also experiment with the same, but in a low-resource LORA [44] tuning setting and not full pretraining. This approach is denoted as **w/ Dino FT**.

4.1.2 How to best convey camera orientation during training?

We explore the setting where we do train with precise camera orientation. Since MLMs operate on language and visual promoting, we explore the two modalities to prompt to effectively convey the precise camera orientation to MLMs.

Conveying camera orientation using language. First, we experiment with feeding in the exact camera position and rotation angle specified in the language. Specifically, we add the following to the prompt: `... Image taken from corner (x,z) with a rotation around y of angle <a>. Plausible 3D coordinates ..`. This approach is denoted as **Orient Language** (Figure 3, middle).

Conveying camera and scene orientation using a visual prompt. Next, instead of specifying the camera position and rotation angle in language, we specify only the camera position in language and mark the 3D location of one random object in the image using a red dot as shown in Figure 3 (right). We specify the 3D location of the dot in the prompt like: `The red circular mark in the image is at 3D coordinates (x, y, z)`. The intuition is that the coordinate of the point marked along with the coordinate of the camera in the prompt gives the model a way to orient the image in the 3D space. Compared to specifying the rotation angle in language and making the model estimate 2D to 3D using that information, we wish to check if MLMs can perform the estimation better with the visual information instead. This approach is denoted as **Visual Point**.

4.2 Metrics

Recall that for all settings, we wish to evaluate both the spatial and semantic accuracy of the MLM. We display the spatial accuracy results in Table 2 and the semantic accuracy results in Table 3.

Semantics

For semantics, we measure standard metrics used in classification - object recall. We measure the recall of both the broad class of object (chair, fridge etc) (denoted as **Class** under **Object Recall** in

Tuning Strategy	Absolute		Relative		Pose	
	ACC ↑	L2 ↓	ACC ↑	L2 ↓	ACC (<10°) ↑	ERR (deg) ↓
a. FT	0.5805	0.1466	0.7024	0.0917	0.3880	75.68
b. w/ Dino FT	0.2110	0.3198	0.4299	0.1643	0.1262	100.46
c. Depth	0.6541	0.1176	0.7503	0.0781	0.4409	71.63
d. Orient Language	0.6960	0.1100	0.7595	0.0801	0.4804	66.71
e. Visual Point	**0.7421**	**0.0973**	**0.7782**	**0.0748**	**0.5102**	**61.98**

Table 2: Table showing the spatial accuracy of reconstructing the full 3D scene from a single image. Overlaying depth helps when precise camera orientation is not known over simply fine-tuning. If camera orientation is known, marking a visual point is better than specifying in language.

Tuning Strategy	Object Recall		Count Acc ↑	WallMaterial ↑	FloorMaterial ↑
	Class ↑	Finegrained ↑			
a. FT	0.8733	0.6250	0.6038	0.7545	0.8283
b. w/ Dino FT	0.7328	0.1240	0.1240	0.6338	0
c. Depth	0.8803	0.6238	0.5964	0.7784	0.8343
d. Orient Language	0.8823	0.6386	0.6101	0.7645	0.8703
e. Visual Point	0.8815	0.6496	0.6128	0.7645	0.8403

Table 3: Table showing the semantic accuracy of the entities in the scene. All tuning approaches are good at understanding semantics. Our prompts do not affect the accuracy significantly.

the tables) as well as the fine-grained asset based on the attributes (e.g. armchair01) (denoted as **Finegrained**). We also measure the count error of the objects between predicted and GT - denoted by **Count Err** in the tables. Finally, we also measure the material accuracy of the walls (denoted as **WallMaterial**) and the floors (**FloorMaterial**).

Spatial

We measure the spatial accuracy of the objects placed in two ways - Absolute and Relative. For **Absolute**, we compute the location distances for each object class between prediction and GT. In AI2THOR, we only need to provide the center coordinate of an asset, and each asset is a fixed size. Hence, for each object class in GT, we compute the L2 distances of object centers between predicted and GT after Hungarian matching (as commonly done in standard detection tasks). We normalize the L2 distances by the max dimension of the room. We consider an object to be placed correctly if the normalized error is below 10% of the max dimension (reported as **ACC** in the tables).

Since we would like our approach to be generalizable to scene generations at arbitrary coordinate spaces (especially for cases without precise camera orientation), we also introduce the **Relative** metric. To evaluate the relative positions of objects, we compute the L2 distance between every object-object pair in the image for pairs of object classes in the GT. For each object class, we compare the pairwise distances to all other objects classes for the predicted and GT scenes. If the predicted layout of objects is similar to GT, the hope is that all pairwise relative distances should be similar. Once again, we report the average pairwise **L2** and **ACC** ($< 10\%$ L2 normalized by max dimension).

For pose, we measure the error in degrees for rotation along the y-axis (vertical height axis). Once again, an absolute degree error less than 10 degrees counts as accurate placement. We report both **ACC** and degree **ERR** under **Pose** in Tables 2.

Tuning details We split our dataset into a training set of 11K rooms, a validation set of 500 rooms, and test set of 500 rooms. All numbers reported in the tables are on the test set. Since we focus on low-resource adaptation techniques, we use LORA tuning [44] with a rank of 16 and alpha 32 on the entire model, instead of fully tuning the LLM and vision encoder backbones. During training, we feed in the entire sequence of image tokens, language prompt, and graphics program of the scene and compute cross-entropy loss for the next token prediction. During testing, we input the image and prompt and generate the language representation of the scene using a greedy algorithm. We keep the generation parameters standard to the official implementation of LLaVA [67]. More details are in the

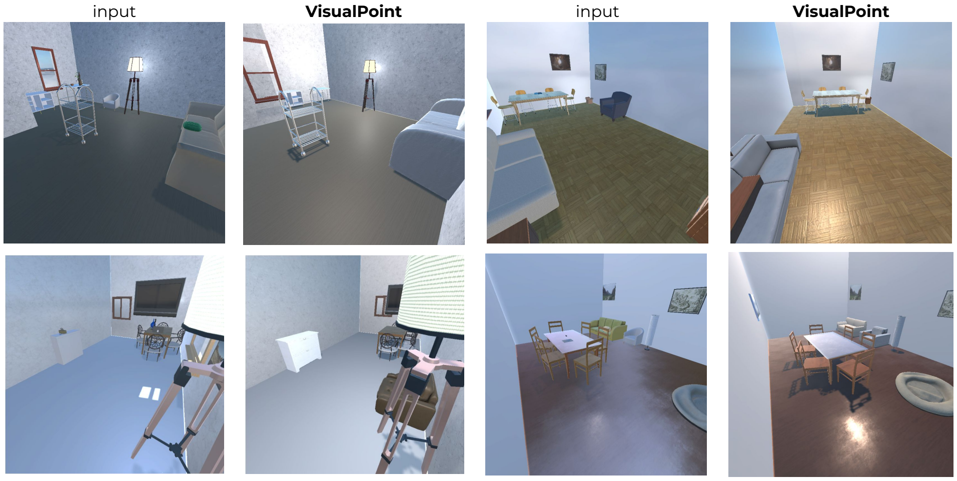
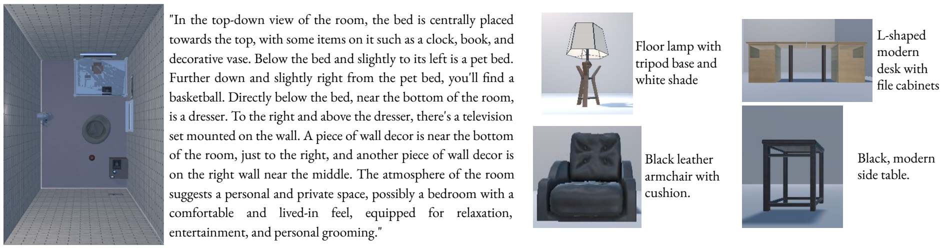


Figure 4: Qualitative results showing the input 2D image and the view rendered from the graphics program output of our VisualPoint strategy. VisualPoint is accurate at reconstructing the scene, although there are occasional errors in pose and object hallucination.


"In the top-down view of the room, the bed is centrally placed towards the top, with some items on it such as a clock, book, and decorative vase. Below the bed and slightly to its left is a pet bed. Further down and slightly right from the pet bed, you'll find a basketball. Directly below the bed, near the bottom of the room, is a dresser. To the right and above the dresser, there's a television set mounted on the wall. A piece of wall decor is near the bottom of the room, just to the right, and another piece of wall decor is on the right wall near the middle. The atmosphere of the room suggests a personal and private space, possibly a bedroom with a comfortable and lived-in feel, equipped for relaxation, entertainment, and personal grooming."

Floor lamp with tripod base and white shade
L-shaped modern desk with file cabinets
Black leather armchair with cushion.
Black, modern side table.

Figure 5: Some examples of the GPT4-V[2] generated captions for the 3D relative locations of assets in the rooms (left) and the attribute-based description of the ProcTHOR [24] assets.

appendix. In all our experiments, due to GPU memory constraints for context length, we only predict the major objects per room (around 8 per room on average) from a choice of 472 assets across 172 object classes.

5 Results

Overlaying depth can help LLaVA estimate camera orientation significantly. As seen in Table 2, rows a vs c, overlaying the estimated depth helps spatial estimation of the objects by 7% absolute and 5% relative. Both these approaches assume access to no precise camera-scene orientation data, and the model must learn to estimate it based on the image it sees and the knowledge of the corner the image was taken from in the room polygon. While standard fine-tuning underperforms to learn from the 3D locations in GT, estimating the depth improves the ability to orient the image correctly.

Having precise camera-scene orientation during training is beneficial to estimate it during testing. As seen in Table 2, specifying the exact camera-scene orientation parameters during training in the language prompt (row d) understandably helps significantly over training approaches without (row a, c). Recall that we only assume access to noisy camera-scene orientation during test time.

Marking the 3D location of a random object is an effective way to convey camera-scene orientation. As seen in Table 2, we see that marking a random object with the 3D coordinate (row e) during training outperforms specifying the exact camera-scene orientation in the language (row e). This suggests that LLaVA is better at estimating the camera-scene orientation by interpolating between the camera position and the visual point marked in the image than at interpreting the position and angle specified in language alone.

For LORA-based tuning, providing estimated depth is much better than switching ViT for DinoV2. While recent work [8] suggests DinoV2 [82] has superior 3D capabilities to ViT [27], we see that LLaVA has a difficult time interpreting DinoV2 features if not pre-trained completely like in CubeLLM [17]. As seen in Table 2 and 3 (rows b vs c), LLaVA with DinoV2 [27] tuned using LORA [44] underperforms in all metrics.

Pose accuracy leaves room for improvement. We see that in Table 2, the pose accuracy of our best method is only at 51%. On average, we see a $61°$ error in the pose for objects, leaving considerable room for improvement.

There is no significant change in semantic accuracy between standard FT and our prompting methods. As seen in Table 3, there is no significant change in the semantic capabilities of the standard LLaVA, whether we fine-tune or use our prompting strategies. This suggests that while our prompting methods do not affect the semantic accuracy of LLaVA, they improve the spatial accuracy (Table 2).

LLaVA is good at object classes, but not at fine-grained recognition and counting. In Table 3, a high object recall suggests that LLaVA is already pretty good at recognizing broad classes of objects. However, the accuracy of fine-grained recognition is lower. The accuracy is also lower for counting objects precisely across all adaptation methods.

Qualitative examples of our results and data While our dataset is procedurally generated using physics engines (which is always accurate), we show some qualitative examples of the captions generated by GPT4-V (which may be noisy) for the rooms and the assets in Figure 5. We also show some qualitative results of our best-performing model outputs in Figure 4. The input is from our R2D3 dataset, and the corresponding rendering is from the graphics program output of our MLM. While locations are mostly accurate, there are errors in pose and object hallucination.

6 Discussion

Limitations Our work fine-tunes LLaVA. Hence, understandably, LLaVA loses some general question-answering and conversational capabilities. However, the lessons from the adaptation can be applied to a larger-scale adaptation where we mix in some of the original natural language data to retain the original LLM capabilities. Our task and benchmark are also based on simulators and graphics engines. While recent works [29, 98] show that training in simulation can transfer to real, further investigation is needed to analyze transfer to real environments and other downstream tasks. Further analysis is also required to test other state-of-the-art MLMs [2, 127, 20].

Future work While we use our benchmark here to reconstruct 3D scenes from a 2D image, it can be extended in various ways. Interesting avenues for future work can look at utilizing the interactive nature of our scenes and reasoning about tasks using them. We can also view our work as image to 3D interactive scene generation. In that case, an exciting future work would be to look into generalizing real apartment images in the wild to create digital replicas.

Conclusion We hope that our benchmark paves the way to explore strategies that impart better 3D reasoning in multimodal language models (MLMs) and improve further on our baselines to make them deployable for real-life applications.

Acknowledgments and Disclosure of Funding

This project was partially funded by Amazon Science and NSF award number 2120322. We wish to thank Devdeep Ray and Jang Hyun Cho for helpful discussions on 3D and depth estimation. The contents of the paper do not represent the views of the US Government or the funding agencies.

References

[1] AI2-THOR: An Interactive 3D Environment for Visual AI. *ArXiv*, abs/1712.05474, 2017.

[2] 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.

[3] 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.

[4] Christopher Agia, Krishna Murthy Jatavallabhula, Mohamed Khodeir, Ondrej Miksik, Vibhav Vineet, Mustafa Mukadam, Liam Paull, and Florian Shkurti. Taskography: Evaluating robot task planning over large 3d scene graphs. In *Conference on Robot Learning*, pages 46–58. PMLR, 2022.

[5] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems*, 35:23716–23736, 2022.

[6] Armen Avetisyan, Christopher Xie, Henry Howard-Jenkins, Tsun-Yi Yang, Samir Aroudj, Suvam Patra, Fuyang Zhang, Duncan Frost, Luke Holland, Campbell Orme, et al. Scenescript: Reconstructing scenes with an autoregressive structured language model. *arXiv preprint arXiv:2403.13064*, 2024.

[7] Daichi Azuma, Taiki Miyanishi, Shuhei Kurita, and Motoaki Kawanabe. Scanqa: 3d question answering for spatial scene understanding. In *proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 19129–19139, 2022.

[8] Mohamed El Banani, Amit Raj, Kevis-Kokitsi Maninis, Abhishek Kar, Yuanzhen Li, Michael Rubinstein, Deqing Sun, Leonidas Guibas, Justin Johnson, and Varun Jampani. Probing the 3d awareness of visual foundation models. *arXiv preprint arXiv:2404.08636*, 2024.

[9] Gilad Baruch, Zhuoyuan Chen, Afshin Dehghan, Tal Dimry, Yuri Feigin, Peter Fu, Thomas Gebauer, Brandon Joffe, Daniel Kurz, Arik Schwartz, et al. Arkitscenes: A diverse real-world dataset for 3d indoor scene understanding using mobile rgb-d data. *arXiv preprint arXiv:2111.08897*, 2021.

[10] Bruce Guenther Baumgart. *Geometric modeling for computer vision.* Stanford University, 1974.

[11] Valts Blukis, Chris Paxton, Dieter Fox, Animesh Garg, and Yoav Artzi. A persistent spatial semantic representation for high-level natural language instruction execution. In *Conference on Robot Learning*, pages 706–717. PMLR, 2022.

[12] Angel X Chang, Mihail Eric, Manolis Savva, and Christopher D Manning. Sceneseer: 3d scene design with natural language. *arXiv preprint arXiv:1703.00050*, 2017.

[13] Boyuan Chen, Fei Xia, Brian Ichter, Kanishka Rao, Keerthana Gopalakrishnan, Michael S Ryoo, Austin Stone, and Daniel Kappler. Open-vocabulary queryable scene representations for real world planning. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11509–11522. IEEE, 2023.

[14] Boyuan Chen, Zhuo Xu, Sean Kirmani, Brian Ichter, Danny Driess, Pete Florence, Dorsa Sadigh, Leonidas Guibas, and Fei Xia. Spatialvlm: Endowing vision-language models with spatial reasoning capabilities. *arXiv preprint arXiv:2401.12168*, 2024.

[15] 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.

[16] Zhenyu Chen, Ali Gholami, Matthias Nießner, and Angel X Chang. Scan2cap: Context-aware dense captioning in rgb-d scans. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3193–3203, 2021.

[17] Jang Hyun Cho, Boris Ivanovic, Yulong Cao, Edward Schmerling, Yue Wang, Xinshuo Weng, Boyi Li, Yurong You, Philipp Krähenbühl, Yan Wang, et al. Language-image models with 3d understanding. *arXiv preprint arXiv:2405.03685*, 2024.

[18] Bob Coyne and Richard Sproat. Wordseye: An automatic text-to-scene conversion system. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 487–496, 2001.

[19] Manuel Dahnert, Ji Hou, Matthias Nießner, and Angela Dai. Panoptic 3d scene reconstruction from a single rgb image. *Advances in Neural Information Processing Systems*, 34:8282–8293, 2021.

[20] Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale N Fung, and Steven Hoi. Instructblip: Towards general-purpose vision-language models with instruction tuning. *Advances in Neural Information Processing Systems*, 36, 2024.

[21] Saumitro Dasgupta, Kuan Fang, Kevin Chen, and Silvio Savarese. Delay: Robust spatial layout estimation for cluttered indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 616–624, 2016.

[22] Matt Deitke, Winson Han, Alvaro Herrasti, Aniruddha Kembhavi, Eric Kolve, Roozbeh Mottaghi, Jordi Salvador, Dustin Schwenk, Eli VanderBilt, Matthew Wallingford, et al. Robothor: An open simulation-to-real embodied ai platform. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3164–3174, 2020.

[23] Matt Deitke, Rose Hendrix, Ali Farhadi, Kiana Ehsani, and Aniruddha Kembhavi. Phone2proc: Bringing robust robots into our chaotic world. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9665–9675, 2023.

[24] Matt Deitke, Eli VanderBilt, Alvaro Herrasti, Luca Weihs, Kiana Ehsani, Jordi Salvador, Winson Han, Eric Kolve, Aniruddha Kembhavi, and Roozbeh Mottaghi. Procthor: Large-scale embodied ai using procedural generation. *Advances in Neural Information Processing Systems*, 35:5982–5994, 2022.

[25] Boyang Deng, Kyle Genova, Soroosh Yazdani, Sofien Bouaziz, Geoffrey Hinton, and Andrea Tagliasacchi. Cvxnet: Learnable convex decomposition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 31–44, 2020.

[26] Maximilian Denninger and Rudolph Triebel. 3d scene reconstruction from a single viewport. In *European Conference on Computer Vision*, pages 51–67. Springer, 2020.

[27] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2021.

[28] Danny Driess, Fei Xia, Mehdi SM Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, et al. Palm-e: An embodied multimodal language model. *arXiv preprint arXiv:2303.03378*, 2023.

[29] Kiana Ehsani, Tanmay Gupta, Rose Hendrix, Jordi Salvador, Luca Weihs, Kuo-Hao Zeng, Kunal Pratap Singh, Yejin Kim, Winson Han, Alvaro Herrasti, Ranjay Krishna, Dustin Schwenk, Eli VanderBilt, and Aniruddha Kembhavi. Imitating shortest paths in simulation enables effective navigation and manipulation in the real world, 2023.

[30] Chuan Fang, Xiaotao Hu, Kunming Luo, and Ping Tan. Ctrl-room: Controllable text-to-3d room meshes generation with layout constraints. *arXiv preprint arXiv:2310.03602*, 2023.

[31] 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.

[32] 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.

[33] Tsu-Jui Fu, Linjie Li, Zhe Gan, Kevin Lin, William Yang Wang, Lijuan Wang, and Zicheng Liu. Violet: End-to-end video-language transformers with masked visual-token modeling. *arXiv preprint arXiv:2111.12681*, 2021.

[34] Xingyu Fu, Yushi Hu, Bangzheng Li, Yu Feng, Haoyu Wang, Xudong Lin, Dan Roth, Noah A Smith, Wei-Chiu Ma, and Ranjay Krishna. Blink: Multimodal large language models can see but not perceive. *arXiv preprint arXiv:2404.12390*, 2024.

[35] Georgia Gkioxari, Jitendra Malik, and Justin Johnson. Mesh r-cnn. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9785–9795, 2019.

[36] Georgia Gkioxari, Nikhila Ravi, and Justin Johnson. Learning 3d object shape and layout without 3d supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1695–1704, 2022.

[37] U Grenander. Lectures in pattern theory i, ii and iii: Pattern analysis. *Pattern Synthesis and Regular Structures*, 1981, 1976.

[38] Can Gümeli, Angela Dai, and Matthias Nießner. Roca: Robust cad model retrieval and alignment from a single image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4022–4031, 2022.

[39] Varsha Hedau, Derek Hoiem, and David Forsyth. Recovering the spatial layout of cluttered rooms. In *2009 IEEE 12th international conference on computer vision*, pages 1849–1856. IEEE, 2009.

[40] Yining Hong, Yilun Du, Chunru Lin, Josh Tenenbaum, and Chuang Gan. 3d concept grounding on neural fields. *Advances in Neural Information Processing Systems*, 35:7769–7782, 2022.

[41] Yining Hong, Chunru Lin, Yilun Du, Zhenfang Chen, Joshua B Tenenbaum, and Chuang Gan. 3d concept learning and reasoning from multi-view images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9202–9212, 2023.

[42] 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.

[43] Cheng-Yu Hsieh, Jieyu Zhang, Zixian Ma, Aniruddha Kembhavi, and Ranjay Krishna. Sugarcrepe: Fixing hackable benchmarks for vision-language compositionality, 2023.

[44] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models, 2021.

[45] Ziniu Hu, Ahmet Iscen, Aashi Jain, Thomas Kipf, Yisong Yue, David A Ross, Cordelia Schmid, and Alireza Fathi. Scenecraft: An llm agent for synthesizing 3d scene as blender code. In *ICLR 2024 Workshop on Large Language Model (LLM) Agents*, 2024.

[46] Chenguang Huang, Oier Mees, Andy Zeng, and Wolfram Burgard. Visual language maps for robot navigation. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 10608–10615. IEEE, 2023.

[47] 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*, volume 35, pages 1610–1618, 2021.

[48] Siyuan Huang, Siyuan Qi, Yixin Zhu, Yinxue Xiao, Yuanlu Xu, and Song-Chun Zhu. Holistic 3d scene parsing and reconstruction from a single rgb image. In *Proceedings of the European conference on computer vision (ECCV)*, pages 187–203, 2018.

[49] Hamid Izadinia, Qi Shan, and Steven M Seitz. Im2cad. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5134–5143, 2017.

[50] Krishna Murthy Jatavallabhula, Alihusein Kuwajerwala, Qiao Gu, Mohd Omama, Tao Chen, Alaa Maalouf, Shuang Li, Ganesh Iyer, Soroush Saryazdi, Nikhil Keetha, Ayush Tewari, Joshua B. Tenenbaum, Celso Miguel de Melo, Madhava Krishna, Liam Paull, Florian Shkurti, and Antonio Torralba. Conceptfusion: Open-set multimodal 3d mapping, 2023.

[51] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *International conference on machine learning*, pages 4904–4916. PMLR, 2021.

[52] Li Jiang, Hengshuang Zhao, Shaoshuai Shi, Shu Liu, Chi-Wing Fu, and Jiaya Jia. Pointgroup: Dual-set point grouping for 3d instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and Pattern recognition*, pages 4867–4876, 2020.

[53] Justin Johnson, Bharath Hariharan, Laurens Van Der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2901–2910, 2017.

[54] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4):1–14, 2023.

[55] D Knill D Kersten and A Yuille. Introduction: A bayesian formulation of visual perception. *Perception as Bayesian inference*, pages 1–21, 1996.

[56] Florian Kluger, Hanno Ackermann, Eric Brachmann, Michael Ying Yang, and Bodo Rosenhahn. Cuboids revisited: Learning robust 3d shape fitting to single rgb images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13070–13079, 2021.

[57] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International journal of computer vision*, 123:32–73, 2017.

[58] Peter Kulits, Haiwen Feng, Weiyang Liu, Victoria Abrevaya, and Michael J Black. Re-thinking inverse graphics with large language models. *arXiv preprint arXiv:2404.15228*, 2024.

[59] Vikram Kumaran, Jonathan Rowe, Bradford Mott, and James Lester. Scenecraft: automating interactive narrative scene generation in digital games with large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, volume 19, pages 86–96, 2023.

[60] Abhijit Kundu, Kyle Genova, Xiaoqi Yin, Alireza Fathi, Caroline Pantofaru, Leonidas J Guibas, Andrea Tagliasacchi, Frank Dellaert, and Thomas Funkhouser. Panoptic neural fields: A semantic object-aware neural scene representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12871–12881, 2022.

[61] Abhijit Kundu, Yin Li, and James M Rehg. 3d-rcnn: Instance-level 3d object reconstruction via render-and-compare. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3559–3568, 2018.

[62] Andrey Kurenkov, Michael Lingelbach, Tanmay Agarwal, Emily Jin, Chengshu Li, Ruohan Zhang, Li Fei-Fei, Jiajun Wu, Silvio Savarese, and Roberto Martın-Martın. Modeling dynamic environments with scene graph memory. In *International Conference on Machine Learning*, pages 17976–17993. PMLR, 2023.

[63] KunChang Li, Yinan He, Yi Wang, Yizhuo Li, Wenhai Wang, Ping Luo, Yali Wang, Limin Wang, and Yu Qiao. Videochat: Chat-centric video understanding. *arXiv preprint arXiv:2305.06355*, 2023.

[64] Shuang Li, Xavier Puig, Chris Paxton, Yilun Du, Clinton Wang, Linxi Fan, Tao Chen, De-An Huang, Ekin Akyürek, Anima Anandkumar, et al. Pre-trained language models for interactive decision-making. *Advances in Neural Information Processing Systems*, 35:31199–31212, 2022.

[65] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft coco: Common objects in context, 2015.

[66] Haolin Liu, Yujian Zheng, Guanying Chen, Shuguang Cui, and Xiaoguang Han. Towards high-fidelity single-view holistic reconstruction of indoor scenes. In *European Conference on Computer Vision*, pages 429–446. Springer, 2022.

[67] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36, 2024.

[68] Zhijian Liu, Haotian Tang, Yujun Lin, and Song Han. Point-voxel cnn for efficient 3d deep learning. *Advances in neural information processing systems*, 32, 2019.

[69] Ruipu Luo, Ziwang Zhao, Min Yang, Junwei Dong, Minghui Qiu, Pengcheng Lu, Tao Wang, and Zhongyu Wei. Valley: Video assistant with large language model enhanced ability. *arXiv preprint arXiv:2306.07207*, 2023.

[70] Wufei Ma, Angtian Wang, Alan Yuille, and Adam Kortylewski. Robust category-level 6d pose estimation with coarse-to-fine rendering of neural features. In *European Conference on Computer Vision*, pages 492–508. Springer, 2022.

[71] Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahad Shahbaz Khan. Video-chatgpt: Towards detailed video understanding via large vision and language models. *arXiv preprint arXiv:2306.05424*, 2023.

[72] Arjun Majumdar, Anurag Ajay, Xiaohan Zhang, Pranav Putta, Sriram Yenamandra, Mikael Henaff, Sneha Silwal, Paul Mcvay, Oleksandr Maksymets, Sergio Arnaud, Karmesh Yadav, Qiyang Li, Ben Newman, Mohit Sharma, Vincent Berges, Shiqi Zhang, Pulkit Agrawal, Yonatan Bisk, Dhruv Batra, Mrinal Kalakrishnan, Franziska Meier, Chris Paxton, Sasha Sax, and Aravind Rajeswaran. Openeqa: Embodied question answering in the era of foundation models. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[73] Hanspeter A Mallot. *From geometry to behavior: An introduction to spatial cognition*. MIT Press, 2024.

[74] Arun Mallya and Svetlana Lazebnik. Learning informative edge maps for indoor scene layout prediction. In *Proceedings of the IEEE international conference on computer vision*, pages 936–944, 2015.

[75] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4460–4470, 2019.

[76] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021.

[77] Tom Monnier, Jake Austin, Angjoo Kanazawa, Alexei Efros, and Mathieu Aubry. Differentiable blocks world: Qualitative 3d decomposition by rendering primitives. *Advances in Neural Information Processing Systems*, 36:5791–5807, 2023.

[78] Daniel R. Montello. Spatial cognition. In James D. Wright, editor, *International Encyclopedia of the Social & Behavioral Sciences (Second Edition)*, pages 111–115. Elsevier, Oxford, second edition edition, 2015.

[79] Vinod Nair, Josh Susskind, and Geoffrey E Hinton. Analysis-by-synthesis by learning to invert generative black boxes. In *Artificial Neural Networks-ICANN 2008: 18th International Conference, Prague, Czech Republic, September 3-6, 2008, Proceedings, Part I 18*, pages 971–981. Springer, 2008.

[80] Yinyu Nie, Angela Dai, Xiaoguang Han, and Matthias Nießner. Learning 3d scene priors with 2d supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 792–802, 2023.

[81] Yinyu Nie, Xiaoguang Han, Shihui Guo, Yujian Zheng, Jian Chang, and Jian Jun Zhang. Total3dunderstanding: Joint layout, object pose and mesh reconstruction for indoor scenes from a single image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 55–64, 2020.

[82] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

[83] Despoina Paschalidou, Luc Van Gool, and Andreas Geiger. Learning unsupervised hierarchical part decomposition of 3d objects from a single rgb image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1060–1070, 2020.

[84] Despoina Paschalidou, Amlan Kar, Maria Shugrina, Karsten Kreis, Andreas Geiger, and Sanja Fidler. Atiss: Autoregressive transformers for indoor scene synthesis. *Advances in Neural Information Processing Systems*, 34:12013–12026, 2021.

[85] Despoina Paschalidou, Angelos Katharopoulos, Andreas Geiger, and Sanja Fidler. Neural parts: Learning expressive 3d shape abstractions with invertible neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3204–3215, 2021.

[86] Despoina Paschalidou, Ali Osman Ulusoy, and Andreas Geiger. Superquadrics revisited: Learning 3d shape parsing beyond cuboids. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10344–10353, 2019.

[87] Georgios Pavlakos, Xiaowei Zhou, Aaron Chan, Konstantinos G Derpanis, and Kostas Daniilidis. 6-dof object pose from semantic keypoints. In *2017 IEEE international conference on robotics and automation (ICRA)*, pages 2011–2018. IEEE, 2017.

[88] Charles R Qi, Or Litany, Kaiming He, and Leonidas J Guibas. Deep hough voting for 3d object detection in point clouds. In *proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9277–9286, 2019.

[89] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017.

[90] 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.

[91] Arijit Ray, Filip Radenovic, Abhimanyu Dubey, Bryan A. Plummer, Ranjay Krishna, and Kate Saenko. Cola: A benchmark for compositional text-to-image retrieval, 2023.

[92] Yuzhuo Ren, Shangwen Li, Chen Chen, and C-C Jay Kuo. A coarse-to-fine indoor layout estimation (cfile) method. In *Computer Vision–ACCV 2016: 13th Asian Conference on Computer Vision, Taipei, Taiwan, November 20-24, 2016, Revised Selected Papers, Part V 13*, pages 36–51. Springer, 2017.

[93] Gernot Riegler, Ali Osman Ulusoy, and Andreas Geiger. Octnet: Learning deep 3d representations at high resolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3577–3586, 2017.

[94] Mehdi SM Sajjadi, Daniel Duckworth, Aravindh Mahendran, Sjoerd Van Steenkiste, Filip Pavetic, Mario Lucic, Leonidas J Guibas, Klaus Greff, and Thomas Kipf. Object scene representation transformer. *Advances in Neural Information Processing Systems*, 35:9512–9524, 2022.

[95] Shuai Shao, Zeming Li, Tianyuan Zhang, Chao Peng, Gang Yu, Xiangyu Zhang, Jing Li, and Jian Sun. Objects365: A large-scale, high-quality dataset for object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8430–8439, 2019.

[96] Daeyun Shin, Zhile Ren, Erik B Sudderth, and Charless C Fowlkes. 3d scene reconstruction with multi-layer depth and epipolar transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 2172–2182, 2019.

[97] Mohit Shridhar, Jesse Thomason, Daniel Gordon, Yonatan Bisk, Winson Han, Roozbeh Mottaghi, Luke Zettlemoyer, and Dieter Fox. Alfred: A benchmark for interpreting grounded instructions for everyday tasks, 2020.

[98] Sneha Silwal, Karmesh Yadav, Tingfan Wu, Jay Vakil, Arjun Majumdar, Sergio Arnaud, Claire Chen, Vincent-Pierre Berges, Dhruv Batra, Aravind Rajeswaran, Mrinal Kalakrishnan, Franziska Meier, and Oleksandr Maksymets. What do we learn from a large-scale study of pre-trained visual representations in sim and real environments?, 2023.

[99] Shuran Song, Samuel P Lichtenberg, and Jianxiong Xiao. Sun rgb-d: A rgb-d scene understanding benchmark suite. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 567–576, 2015.

[100] Reuben Tan, Ximeng Sun, Ping Hu, Jui-hsien Wang, Hanieh Deilamsalehy, Bryan A Plummer, Bryan Russell, and Kate Saenko. Koala: Key frame-conditioned long video-llm. *arXiv preprint arXiv:2404.04346*, 2024.

[101] Alykhan Tejani, Danhang Tang, Rigas Kouskouridas, and Tae-Kyun Kim. Latent-class hough forests for 3d object detection and pose estimation. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part VI 13*, pages 462–477. Springer, 2014.

[102] Shashank Tripathi, Lea Müller, Chun-Hao P Huang, Omid Taheri, Michael J Black, and Dimitrios Tzionas. 3d human pose estimation via intuitive physics. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4713–4725, 2023.

[103] Maria Tsimpoukelli, Jacob L Menick, Serkan Cabi, SM Eslami, Oriol Vinyals, and Felix Hill. Multimodal few-shot learning with frozen language models. *Advances in Neural Information Processing Systems*, 34:200–212, 2021.

[104] Shubham Tulsiani, Hao Su, Leonidas J Guibas, Alexei A Efros, and Jitendra Malik. Learning shape abstractions by assembling volumetric primitives. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2635–2643, 2017.

[105] Anton van den Hengel, Chris Russell, Anthony Dick, John Bastian, Daniel Pooley, Lachlan Fleming, and Lourdes Agapito. Part-based modelling of compound scenes from images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 878–886, 2015.

[106] Vaibhav Vavilala and David Forsyth. Convex decomposition of indoor scenes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9176–9186, 2023.

[107] Junke Wang, Dongdong Chen, Zuxuan Wu, Chong Luo, Luowei Zhou, Yucheng Zhao, Yujia Xie, Ce Liu, Yu-Gang Jiang, and Lu Yuan. Omnivl: One foundation model for image-language and video-language tasks. *Advances in neural information processing systems*, 35:5696–5710, 2022.

[108] Peng-Shuai Wang. Octformer: Octree-based transformers for 3d point clouds. *ACM Transactions on Graphics (TOG)*, 42(4):1–11, 2023.

[109] Peng-Shuai Wang, Yang Liu, Yu-Xiao Guo, Chun-Yu Sun, and Xin Tong. O-cnn: Octree-based convolutional neural networks for 3d shape analysis. *ACM Transactions On Graphics (TOG)*, 36(4):1–11, 2017.

[110] Yi Wang, Kunchang Li, Xinhao Li, Jiashuo Yu, Yinan He, Guo Chen, Baoqi Pei, Rongkun Zheng, Jilan Xu, Zun Wang, et al. Internvideo2: Scaling video foundation models for multimodal video understanding. *arXiv preprint arXiv:2403.15377*, 2024.

[111] Yi Wang, Kunchang Li, Yizhuo Li, Yinan He, Bingkun Huang, Zhiyu Zhao, Hongjie Zhang, Jilan Xu, Yi Liu, Zun Wang, et al. Internvideo: General video foundation models via generative and discriminative learning. *arXiv preprint arXiv:2212.03191*, 2022.

[112] Luca Weihs, Matt Deitke, Aniruddha Kembhavi, and Roozbeh Mottaghi. Visual room rearrangement. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5922–5931, 2021.

[113] Jiajun Wu, Joshua B Tenenbaum, and Pushmeet Kohli. Neural scene de-rendering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 699–707, 2017.

[114] Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything: Unleashing the power of large-scale unlabeled data. In *CVPR*, 2024.

[115] 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 *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2024)*, volume 30, pages 20–25. IEEE/CVF, 2024.

[116] Shunyu Yao, Tzu Ming Hsu, Jun-Yan Zhu, Jiajun Wu, Antonio Torralba, Bill Freeman, and Josh Tenenbaum. 3d-aware scene manipulation via inverse graphics. *Advances in neural information processing systems*, 31, 2018.

[117] Qinghao Ye, Haiyang Xu, Guohai Xu, Jiabo Ye, Ming Yan, Yiyang Zhou, Junyang Wang, Anwen Hu, Pengcheng Shi, Yaya Shi, et al. mplug-owl: Modularization empowers large language models with multimodality. *arXiv preprint arXiv:2304.14178*, 2023.

[118] Shuquan Ye, Dongdong Chen, Songfang Han, and Jing Liao. 3d question answering. *arXiv preprint arXiv:2112.08359*, 2021.

[119] Lu Yuan, Dongdong Chen, Yi-Ling Chen, Noel Codella, Xiyang Dai, Jianfeng Gao, Houdong Hu, Xuedong Huang, Boxin Li, Chunyuan Li, et al. Florence: A new foundation model for computer vision. *arXiv preprint arXiv:2111.11432*, 2021.

[120] Alan Yuille and Daniel Kersten. Vision as bayesian inference: analysis by synthesis? *Trends in cognitive sciences*, 10(7):301–308, 2006.

[121] Cheng Zhang, Zhaopeng Cui, Yinda Zhang, Bing Zeng, Marc Pollefeys, and Shuaicheng Liu. Holistic 3d scene understanding from a single image with implicit representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8833–8842, 2021.

[122] Hang Zhang, Xin Li, and Lidong Bing. Video-llama: An instruction-tuned audio-visual language model for video understanding. *arXiv preprint arXiv:2306.02858*, 2023.

[123] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip HS Torr, and Vladlen Koltun. Point transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 16259–16268, 2021.

[124] Qitao Zhao, Ce Zheng, Mengyuan Liu, and Chen Chen. A single 2d pose with context is worth hundreds for 3d human pose estimation. *Advances in Neural Information Processing Systems*, 36, 2024.

[125] Qitao Zhao, Ce Zheng, Mengyuan Liu, Pichao Wang, and Chen Chen. Poseformerv2: Exploring frequency domain for efficient and robust 3d human pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8877–8886, 2023.

[126] Jia Zheng, Junfei Zhang, Jing Li, Rui Tang, Shenghua Gao, and Zihan Zhou. Structured3d: A large photo-realistic dataset for structured 3d modeling. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IX 16*, pages 519–535. Springer, 2020.

[127] Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*, 2023.

[128] Chuhang Zou, Alex Colburn, Qi Shan, and Derek Hoiem. Layoutnet: Reconstructing the 3d room layout from a single rgb image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2051–2059, 2018.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [NA] Our dataset is a synthetic benchmark to test spatial perception in MLMs. Hence, we do not deal with any individual, cultural, or societal topics. The only bias could be that our apartment indoor scenes reflect primarily North American homes.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [NA] We have no theoretical results. Only empirical.
   (b) Did you include complete proofs of all theoretical results? [NA]
3. If you ran experiments (e.g. for benchmarks)...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] . In the supplementary.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] . In the supplementary.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No] Our training runs are computationally expensive and hence we only run them a couple times. The differences in accuracy are large for our case and evaluated on a relatively large number of 3d rooms (500). Hence, we do not compute error bars from multiple runs. However, we will release training code and checkpoints for reproducibility.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [Yes]
   (c) Did you include any new assets either in the supplemental material or as a URL? [No]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [NA] . We use publicly available open-source datasets.
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [NA] . Our datasets are synthetic and no personally identifiable content is present.
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [NA] . Our dataset is synthetically generated and doesn't need any human annotation.
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [NA]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [NA] .