3D Visual Prompting for Visual Question Answering

Bingning Huang, Nigar Doga Karacik, Ran Ding
Technische Universität München
bingninghuang68@gmail.com, ndoga.karacik@tum.de, ran.ding@tum.de

February 16, 2024

Abstract

Most existing vision and language (V&L) models focus primarily on 2D images, which limits their understanding of spatial information. To address this gap, we introduce a novel 3DVQA pipeline that directly uses 3D point clouds along with question text as input to enrich the model's spatial awareness, and outputs the answer text. In addition, GPT3.5 is prompted to generate question-answer pairs based on the Matterport3D [2] dataset. As a result, our model shows great ability in answering different types of questions.

Keywords– 3D Point Cloud, Visual Question Answering, Dataset Generation

1 Introduction

3D Visual Question Answering (VQA) integrates 3D scene understanding and natural language processing to enable systems to answer questions about 3D spatial data, such as point clouds. This technology has the potential to enhance virtual and augmented reality experiences and optimize robotic interactions within 3D environments. However, the advancement of 3D visual question answering faces obstacles, such as the absence of extensively annotated datasets that combine 3D visuals with relevant questions, as well as the complexity of deriving meaningful 3D features for accurate answer generation.

Nevertheless, promising new developments are emerging to address these challenges. Large language models (LLMs) can facilitate the creation of comprehensive 3D VQA datasets. Furthermore, Peng et al. [5] have developed a model that distils complex 3D point cloud features with 2D multiview CLIP features, allowing for the direct use of point clouds. Building on these advances, we have developed a 3D VQA pipeline that encodes the 3D scene point cloud using a pre-trained OpenScene [5] distill encoder. We then align the features using semantic pooling techniques, fuse with the text features, and predicts meaning answer. Our approach has produced good qualitative and quantitative results on the Matterport3D dataset [2], and we have compared our features with those from various sources. Additionally, we created a 3DVQA dataset based on the Matterport3D dataset[2], which has 10,800 aligned 3D panoramic views. We used GPT API to generate these answers using the object labels and the extracted coordinates from the 3D point cloud. Our dataset has 37,470 question-answer pairs from 1196 distinct scenes.

2 Related work

2.1 2D Visual Question Answering

The task of Visual Question Answering(VQA) is to provide the answer given an image and a related question.

Despite the strong zero-shot capability of CLIP on vision tasks, CLIP does not exhibit the same level of performance on certain visual and language downstream tasks. In CLIP-ViL [7], they propose to integrate CLIP's visual encoder with previous V&L models by replacing their visual encoder with CLIP's visual encoder.

MCAN [9] takes Fatser-RCNN as visual encoder, LSTM as question encoder and an encoder-decoder based modular co-attention network for fusing multiple representations. And employ an output classifier on top of the fused representation to predict the final answer. In CLIP-ViL, they replace the Fatser-RCNN with CLIP visual encoder. In the VQA task, the combination of the MCAN model with CLIP-Res50x4 yields the best-performing results.

2.2 3D visual feature processing

Our method draws on OpenScene [5], a simple yet effective zero-shot approach for open-vocabulary 3D scene understanding. This approach, as shown in Fig. 1, trains a 3D distilled encoder, which brings 3D points in alignment with pixels in the feature space, in turn are aligned with text features. It achieves state-of-the-art for zero-shot 3D semantic segmentation on standard benchmarks, outperforms supervised approaches in 3D semantic segmentation with many class labels.

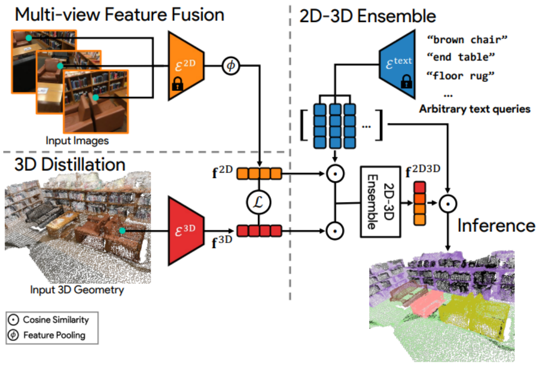


Figure 1: Openscene model.

2.3 Question&Answer prompting and generation

To better activate CLIP's textual encoder to align with inputs, PointCLIP V2[10] aims to utilize 3D-specific description with category-wise shape characteristics as the textual input of CLIP. Considering the powerful descriptive capacity of LLMs, they leverage GPT-3.5 to generate 3D specific text with sufficient 3D semantics for CLIP's textual encoder.

Besides, many previous works have used GPT to expand the capacity of their current VQA dataset and made their prompts capable of reasoning to generate responses in autonomous driving settings [4]. We do not have access to enough data to make VQA about the 3D scene like in the 2D VQA case. The ScanQA[8] dataset, built on the ScanNet dataset, has 10,062 fully human-annotated question-answer pairs about 3D scenes. ScanQA-3D QA [1] dataset has human edited 41,363 questions and 58,191 answers. GPT enables us to create the desired type and huge amount of data much faster and cheaper than human annotators. 3DLLM [3] also used GPT API to generate questions using 3D scene information, but for the question-answering part, they used 2D images. In this study, we used GPT API to create answers about given question templates and 3D scene information that only includes 3D point cloud information and labels.

3 Method

An overview of our approach is illustrated in Fig. 2. We first extract the 3d features encoded by 3D distillation encoder.

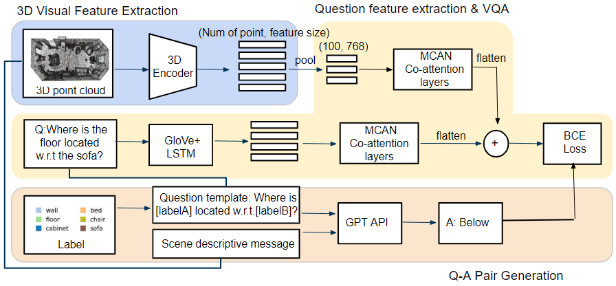

3D Visual Feature Extraction
Question feature extraction & VQA
3D point cloud
3D Encoder
(Num of point, feature size)
pool
(100, 768)
MCAN Co-attention layers
flatten
Q:Where is the floor located w.r.t the sofa?
GloVe+ LSTM
MCAN Co-attention layers
flatten
+
BCE Loss
wall
floor
cabinet
bed
chair
sofa
Label
Question template: Where is [labelA] located w.r.t.[labelB]?
Scene descriptive message
GPT API
A: Below
Q-A Pair Generation

Figure 2: Method overview.

At the same time, question features are extracted through GloVe and LSTM. After pooling processing of visual features, multi-modal features could be input to MCAN co-attention modules and then aggregated to yield the final answer. In order to train the network, we prompt ChatGPT with predefined question template and some scene descriptive message to get the ground truth answer.

3.1 3D Visual distillation

Since the open-vocabulary image embeddings are co-embedded with CLIP features in OpenScene[5], the output of distilled 3D model naturally lives in the same embedding space as CLIP. Therefore, we can distill such 2D visual-language knowledge into a 3D point network that only takes 3D point positions as input

We use the 3D encoder as shown in Fig.2 of pretrained OpenScene model to extract per-point features for Matterport 3D dataset, such that even without any 2D observations, the text-3D co-embeddings allow 3D scene level understanding given arbitrary text prompts.

Since Openscene model has a good perfoemance in 3D semantic segmentation task, we use the predicted semantic label for pooling. Also, we extract multiview features and 2D-3D ensemble features for further ablation analysis.

3.2 Pooling technique

Due to the feature size mismatch, we need some pooling technique $\phi : F_{3D} \mapsto F_{MCAN}$ to convert the 3D distilled feature: $F_{3D} : R^{p*f}$ to feature required for MCAN multi-modal fusion: $F_{MCAN} : R^{ROI*f}$, where p is the number of points in a scene, f is the feature dimension embedded by 3D distilled encoder, ROI is the Region Of Interest.

The pooling techniques can be divided into two types, one is based on Farthest Point Sampling(FPS) and k-Nearest-Neighborhood(kNN) [6], another one is based on semantic label. Since FPS pooling could not give an insight to the underlying structure of the scene, we turn to semantic pooling. Although semantic label from ground truth can show us an upper bound performance, it is not applicable in real life. For better availability, we pool the 3d feature making use of Openscene prediction of 3D semantic segmantation task.

3.3 Multi-modal fusion

We use Global Vectors for Word Representation(GloVe) to obtain question embedding, then employ LSTM network for processing sequences of embedding.

Then, MCAN model[9] , which is stacked by self attention and self-guided-attention modules, are used for joint 3D distilled features and question features processing. An attention flattening layer is used within the MCAN model to flatten the sequence representations obtained from both visual and language features. After layer normalization for concatenation of V&L, a linear projection layer produces the final outputs representing the probability distribution over possible answers.

3.4 Visual question answering

During training, we use BCE loss to supervise the network. During evaluation, we calculate per-answer type accuracy respectively and also the overall accuracy.

Since GPT could only give one correct answer for each question, we perform accurate matching between output and ground truth answer, i.e. the matching result is either true or false.

3.5 Question&Answer generation

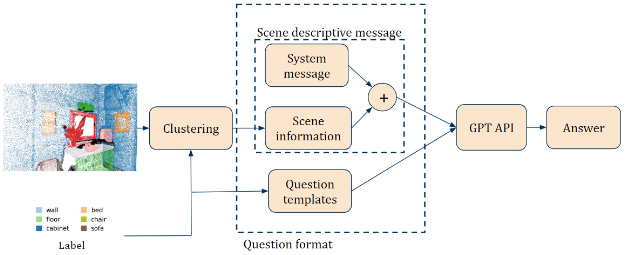


Figure 3: Process of the answer generation.

Feature	Pooling Method	Overall	Count	Spatial	Yes/No	Action
3D Distillation	FPS	55.3	**57.88**	44.21	91.36	33.5
3D Distillation label	GT Semantic Label	**58.82**	57.72	**48.02**	92.3	**41.99**
3D Distillation	Predicted Semantic Label	58.73	56.9	**48.02**	**93.58**	41.3

Table 1: Comparison of Pooling Methods on Q&A Pair Accuracy.

Feature	Pooling Method	Overall	Count	Spatial	Yes/No	Action
3D Distillation	GT Semantic Label	58.82	57.72	48.02	92.3	41.99
3D Distillation	Predicted Semantic Label	58.73	56.9	48.02	93.58	41.3
2D Multiview	GT Semantic Label	59.08	58.24	50.79	94.14	40.32
2D Multiview	Predicted Semantic Label	58.45	58.2	50.53	94.35	38.96
2D Singleview	GT Semantic Label	57.03	57.72	46.51	91.96	37.26
2D Singleview	Predicted Semantic Label	55.85	52.38	44.76	92.3	38.86
2D-3D Ensemble	GT Semantic Label	**61.7**	58.7	**52.54**	**94.78**	**45.33**
2D-3D Ensemble	Predicted Semantic Label	61.14	**59.2**	52.14	93.76	44.15

Table 2: Performance Evaluation of Feature Types Across Question Categories.

We can divide the question-answer generation into three parts to prompt GPT API, as shown in Fig. 3.

(1) Creation of the system message: the system message includes a task description, instructions about the task expected from GPT API, and some example cases A.1. **(2) Creation of scene information:** scene information includes scene ID, [X, Y, Z] center coordinates of clustered objects from every object category, bounding box information, and their object labels (A.1 Fig. 5). Since Matterport 41 labels provide us with extensive object options, we created our dataset based on Matterport41 from the Matterport 3D dataset [2]. We used DBScan and plane segmentation from Open3D for clustering the objects. **(3) Creation of the question templates:** it includes manually created questions templates for the different cases. We created four different question categories; yes/no, spatial, count, and action. We tried to use GPT API to generate the questions similar to related work [3]. It had created nice questions but could not follow the instructions properly and mixed the questions and question types up. Also, it had more hallucinations than the answers of the manually created questions.

"

4 Results

We constructed a dataset comprising 32,039 Q&A pairs across 1,025 diverse scenes for the training set, and 5,431 Q&A pairs from 171 unique scenes for the validation set. Detailed dataset composition are elaborated in the appendix A.2.

For evaluation purposes, we assess the accuracy of predictions against ground truth (GT) values. A match is scored as '1' (correct), while a mismatch is scored as '0' (incorrect). Our evaluation metrics include overall accuracy and category-specific accuracy for question types such as count, spatial, yes/no, and action-related queries. These categories present varying levels of difficulty due to the differing degrees of scene understanding required.

As demonstrated in Table 1, various pooling methods were compared, highlighting that semantic pooling outperforms other techniques. Furthermore, Table 2 explores the performance of features derived from different sources, such as multi-view, single-view, and ensemble features, indicating that ensemble features yield the best results, with single-view features performing the least effectively.

5 Conclusion

In summary, our contributions can be summarized as follows: 1. present a new pipeline that directly utilizes 3D point cloud to realize visual question answer, 2. generate annotated question and answer pairs using GPT3.5. However, since we use semantic pooling to process the visual features, it can't achieve instance-level understanding. Further research should be conducted to solve this problem.

References

[1] Daichi Azuma, Taiki Miyanishi, Shuhei Kurita, and Motoaki Kawanabe. Scanqa: 3d question answering for spatial scene understanding, 2022. 2

[2] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3d: Learning from rgb-d data in indoor environments. *International Conference on 3D Vision (3DV)*, 2017. 1, 4

[3] 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, 2023. 2, 4

[4] Yingzi Ma, Yulong Cao, Jiachen Sun, Marco Pavone, and Chaowei Xiao. Dolphins: Multimodal language model for driving, 2023. 2

[5] Songyou Peng, Kyle Genova, Chiyu ”Max” Jiang, Andrea Tagliasacchi, Marc Pollefeys, and Thomas Funkhouser. Openscene: 3d scene understanding with open vocabularies, 2023. 1, 2, 3

[6] Charles R. Qi, Li Yi, Hao Su, and Leonidas J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space, 2017. 3

[7] Sheng Shen, Liunian Harold Li, Hao Tan, Mohit Bansal, Anna Rohrbach, Kai-Wei Chang, Zhewei Yao, and Kurt Keutzer. How much can clip benefit vision-and-language tasks?, 2021. 1

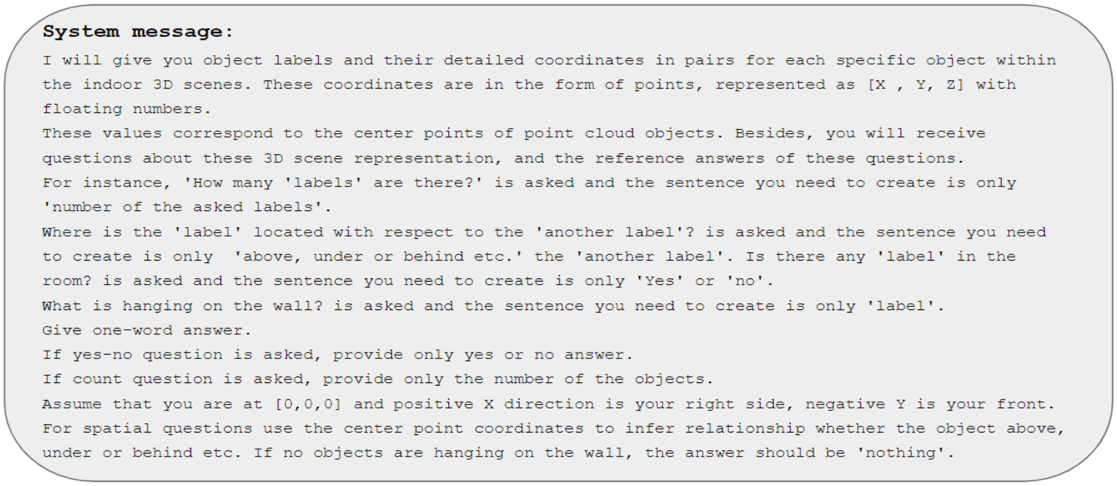
[8] Shuquan Ye, Dongdong Chen, Songfang Han, and Jing Liao. 3d question answering, 2022. 2

[9] Zhou Yu, Jun Yu, Yuhao Cui, Dacheng Tao, and Qi Tian. Deep modular co-attention networks for visual question answering, 2019. 2, 3

[10] Xiangyang Zhu, Renrui Zhang, Bowei He, Ziyu Guo, Ziyao Zeng, Zipeng Qin, Shanghang Zhang, and Peng Gao. Pointclip v2: Prompting clip and gpt for powerful 3d open-world learning, 2023. 2

A Appendix

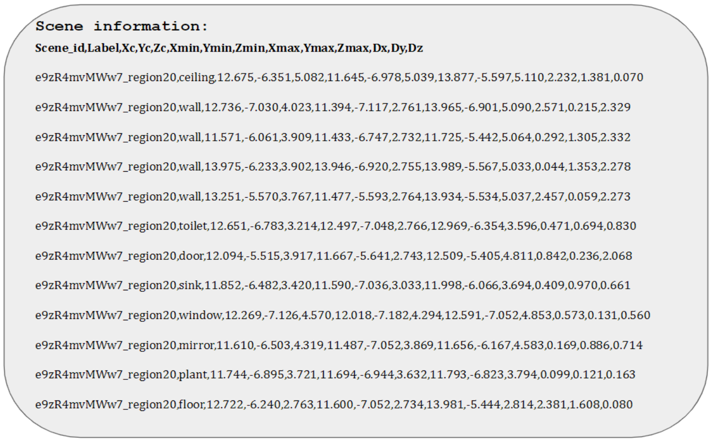
A.1 Scene descriptive message details



System message:

I will give you object labels and their detailed coordinates in pairs for each specific object within the indoor 3D scenes. These coordinates are in the form of points, represented as [X , Y, Z] with floating numbers.
These values correspond to the center points of point cloud objects. Besides, you will receive questions about these 3D scene representation, and the reference answers of these questions.
For instance, 'How many 'labels' are there?' is asked and the sentence you need to create is only 'number of the asked labels'.
Where is the 'label' located with respect to the 'another label'? is asked and the sentence you need to create is only 'above, under or behind etc.' the 'another label'. Is there any 'label' in the room? is asked and the sentence you need to create is only 'Yes' or 'no'.
What is hanging on the wall? is asked and the sentence you need to create is only 'label'.
Give one-word answer.
If yes-no question is asked, provide only yes or no answer.
If count question is asked, provide only the number of the objects.
Assume that you are at [0,0,0] and positive X direction is your right side, negative Y is your front.
For spatial questions use the center point coordinates to infer relationship whether the object above, under or behind etc. If no objects are hanging on the wall, the answer should be 'nothing'.

Figure 4: The system message that we used to send GPT API.



Scene information:

```
Scene_id,Label,Xc,Yc,Zc,Xmin,Ymin,Zmin,Xmax,Ymax,Zmax,Dx,Dy,Dz

e9zR4mvMWw7_region20,ceiling,12.675,-6.351,5.082,11.645,-6.978,5.039,13.877,-5.597,5.110,2.232,1.381,0.070

e9zR4mvMWw7_region20,wall,12.736,-7.030,4.023,11.394,-7.117,2.761,13.965,-6.901,5.090,2.571,0.215,2.329

e9zR4mvMWw7_region20,wall,11.571,-6.061,3.909,11.433,-6.747,2.732,11.725,-5.442,5.064,0.292,1.305,2.332

e9zR4mvMWw7_region20,wall,13.975,-6.233,3.902,13.946,-6.920,2.755,13.989,-5.567,5.033,0.044,1.353,2.278

e9zR4mvMWw7_region20,wall,13.251,-5.570,3.767,11.477,-5.593,2.764,13.934,-5.534,5.037,2.457,0.059,2.273

e9zR4mvMWw7_region20,toilet,12.651,-6.783,3.214,12.497,-7.048,2.766,12.969,-6.354,3.596,0.471,0.694,0.830

e9zR4mvMWw7_region20,door,12.094,-5.515,3.917,11.667,-5.641,2.743,12.509,-5.405,4.811,0.842,0.236,2.068

e9zR4mvMWw7_region20,sink,11.852,-6.482,3.420,11.590,-7.036,3.033,11.998,-6.066,3.694,0.409,0.970,0.661

e9zR4mvMWw7_region20,window,12.269,-7.126,4.570,12.018,-7.182,4.294,12.591,-7.052,4.853,0.573,0.131,0.560

e9zR4mvMWw7_region20,mirror,11.610,-6.503,4.319,11.487,-7.052,3.869,11.656,-6.167,4.583,0.169,0.886,0.714

e9zR4mvMWw7_region20,plant,11.744,-6.895,3.721,11.694,-6.944,3.632,11.793,-6.823,3.794,0.099,0.121,0.163

e9zR4mvMWw7_region20,floor,12.722,-6.240,2.763,11.600,-7.052,2.734,13.981,-5.444,2.814,2.381,1.608,0.080
```

Figure 5: Scene information as an example.

A.2 Dataset details

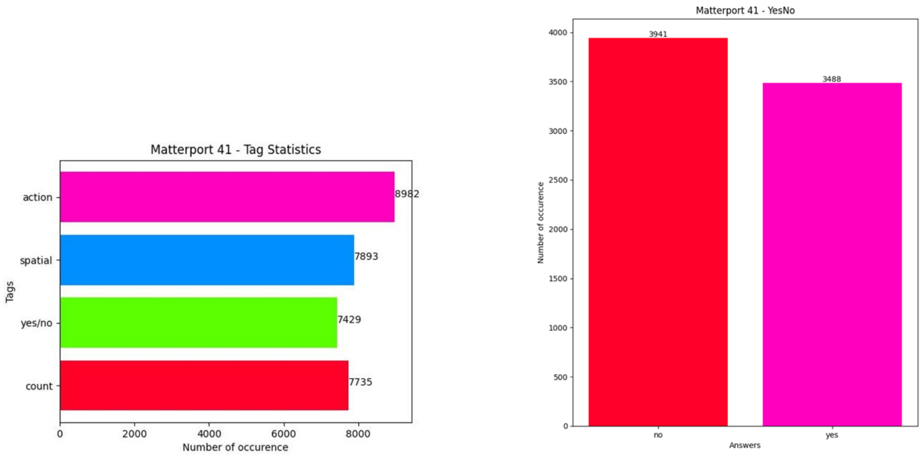


Figure 6: Distribution of the question types and ’yes/no’ questions.

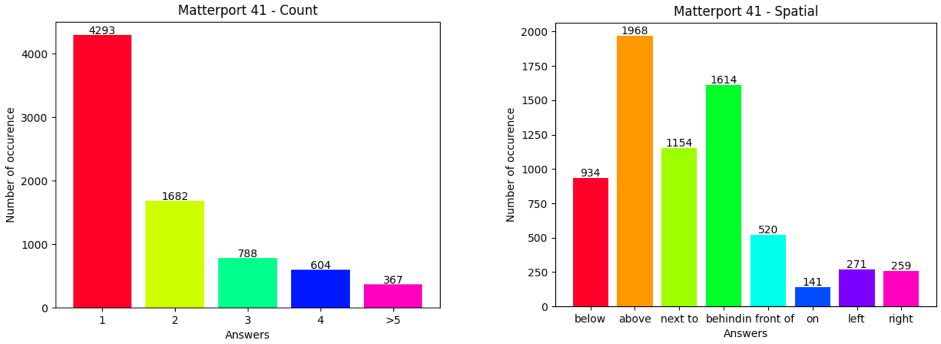


Figure 7: Distribution of the question types and ’count’ and ’spatial’ questions (unaligned version).

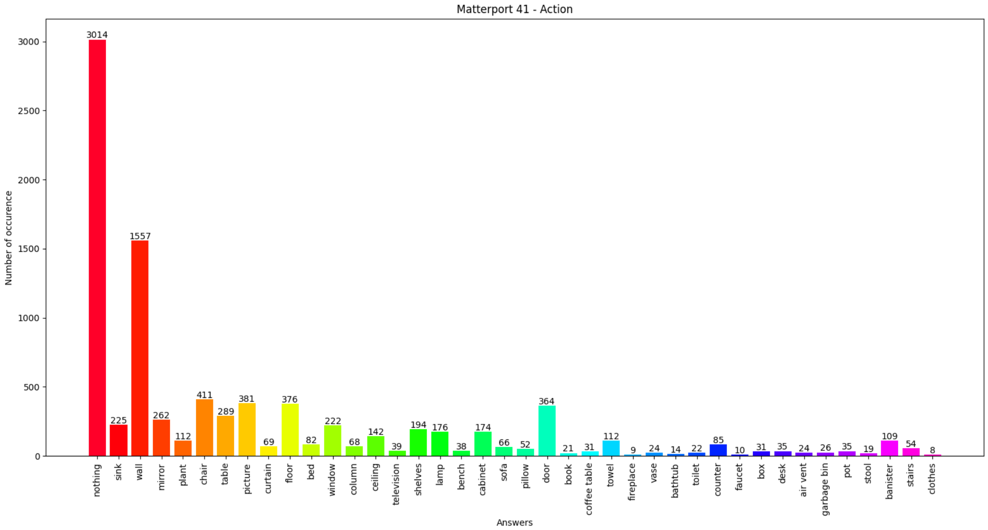
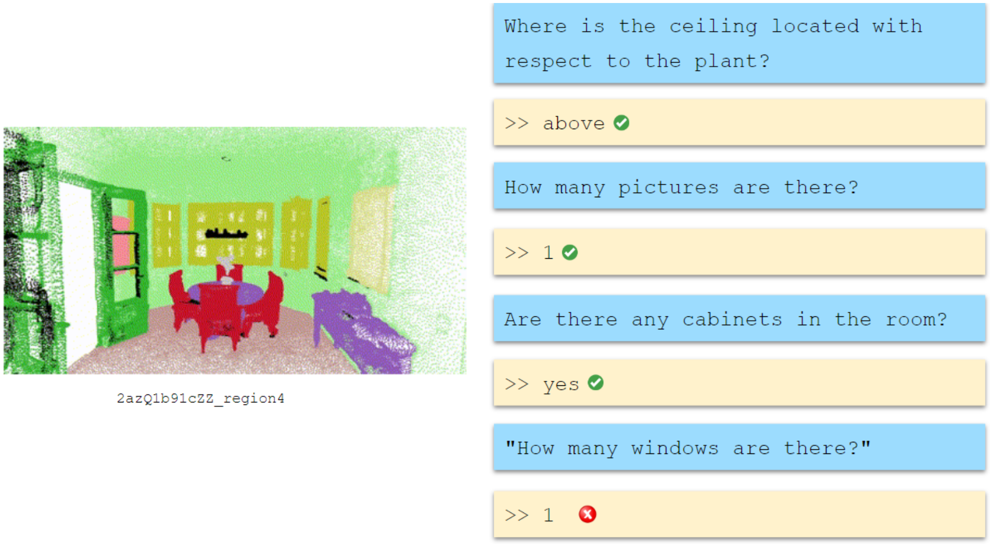


Figure 8: Distribution of ’action’ questions (unaligned version).



Where is the ceiling located with respect to the plant?

>> above ✓

How many pictures are there?

>> 1 ✓

Are there any cabinets in the room?

>> yes ✓

"How many windows are there?"

>> 1 ✗

Figure 9: Answers of our model in the inference time with unaligned data.