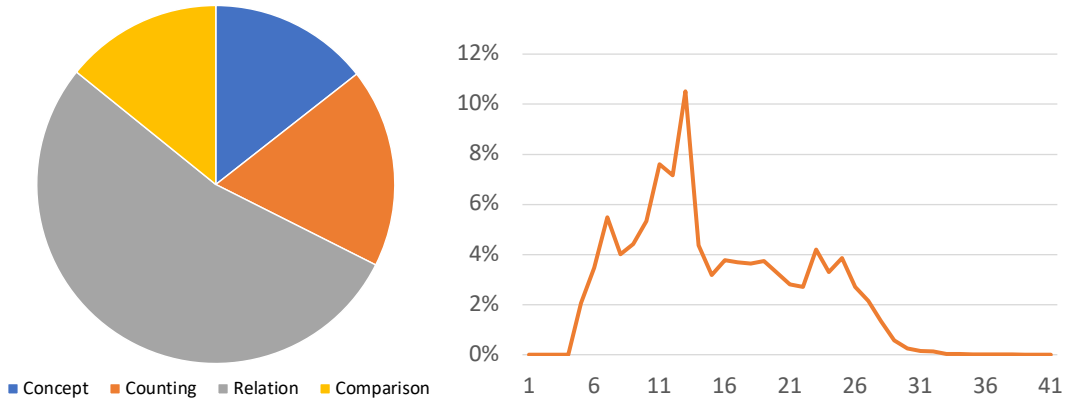
Supplementary Material for "3D Concept Learning and Reasoning from Multi-View Images"

Contents

A Dataset

A.1 Dataset Statistics

Figure 1, we show some statistics about our dataset. From Figure 1(a), we can see that relation type takes up the most portion of the questions, which is reasonable since our dataset focuses on 3D reasoning, and spatial relation is a crucial perspective. From Figure 1 (b), we can see that our questions cover a wide range of word lengths.



(a) Distribution of Question Type

(b) Distribution of Question Word Length

Figure 1: Dataset Statistics

A.2 More Dataset Examples

In Figure 2, we show some more examples of our 3DMV-VQA dataset.

B Implementation details

B.1 Reasoning Operators

- FILTER The FILTER operator takes a voxel grid and a concept as input, and outputs the filtered voxel grid where points do not belong to the input have zero density values.
- GET_INSTANCE The GET_INSTANCE operator takes a voxel grid as input and outputs a list of voxel grids of different instances. We use DBSCAN, an unsupervised algorithm to assign the voxel grid points with densities greater than 0.5 into different instances. If the input is of the same semantic object, we directly use DBSCAN to get the instances. Else, we first get the voxel grids of different classes from semantic concept grounding, and then get the instances of all the instances of each semantic class. After that we integrate the instances of all semantic classes.
- QUERY The QUERY operator takes the voxel grid of an instance and returns the semantic concept of the instance.
- COUNT The COUNT operator takes a list of voxel grids as input and returns the length of the list.

Concept:
Q: Is there a sofa?
A: Yes

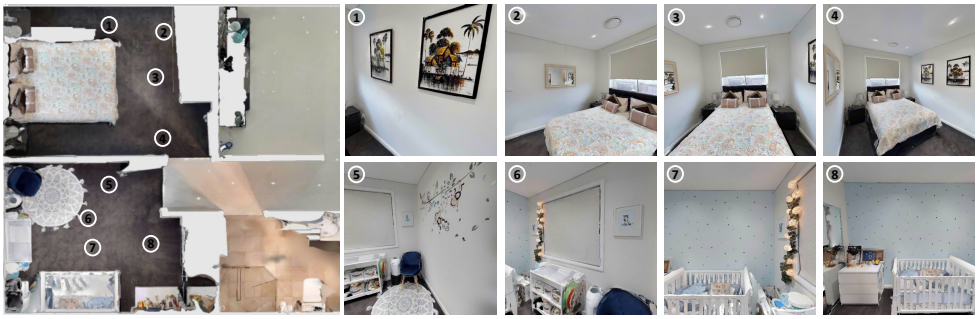
Q: Is there a room with bed?
A: No

Counting:
Q: How many pillows are on the sofa? A: 6

Q: How many rooms have beds? **A**: 0

Relation:
Q: How many pillows are on the sofa? A: 6
Q: How many chairs are close to the table in the room without a sofa ? **A**: 4

Comparison:
Q: Are there more chairs in the room with a sofa than the other?
A: No
Q: Is the sofa closer to a vase or a television?
A: Plant



Concept:
Q: Is there a vase?
A: No

Q: Is there a room without bed?
A: No

Counting:
Q: How many rooms have beds? **A**: 2

Q: How many rooms have mirrors? **A**: 2

Relation:
Q: Facing the smaller bed from the chair, is there a lamp on the right? **A**: No
Q: Is there a television between the chair and the smaller bed? **A:** No

Comparison:
Q: Are there more chairs than beds? **A**: No

Q: Is the chair closer to a table or a bed?
A: Table

Concept:
Q: Are there any televisions?
A: Yes

Q: Is there a towel in the room with a toilet?
A: No

Counting:
Q: How many rooms have mirrors? **A**: 1

Q: How many pillows are on the bed? **A**: 4

Relation:
Q: Facing the sink from the toilet, is there a lamp on the right? **A**: Yes
Q: Is there a television between the shower and the bed? **A:** No

Comparison:
Q: Are there more tables than chairs? **A**: Yes

Q: Is the toilet closer to the sink or the shower?
A: Sink

Figure 2: More Examples of 3DMV-VQA Dataset.

- EXIST The EXIST operator takes a list of voxel grids as input and examines if the list is empty or not. If the list is empty, then the targeted concept doesn't exist.
- GET_ROOM_INSTANCE The GET_ROOM_INSTANCE operator takes a voxel grid as input and returns a list of new voxel grids of different room instances. To get the instances of the rooms, we use the results of 3D semantic grounding and extract all walls. We then segment the whole scene into rooms using the wall instances.
- FILTER_ROOM The FILTER_ROOM operator takes a list of voxel grids of different room instances and returns another list of voxel grids where there are non-zero density values.
- COUNT_ROOM The COUNT_ROOM operator takes a list of voxel grids of different room instances and returns the length of the list of voxel grids where there are non-zero density values. It's a combination of the FILTER_ROOM operator and the COUNT operator.
- EXIST_ROOM The EXIST_ROOM operator takes a list of voxel grids of different room instances and returns whether the list of voxel grids where there are non-zero density values is empty or not. It's a combination of the FILTER_ROOM operator and the EXIST operator.
- RELATION The RELATION operator takes a relation tuple (the voxel grids of two or three instances) and a relation, concatenate them and pass them into the relation module network of the specified relation, and outputs a score (which can be turned into True/False value according to whether the score is greater than 0.5 or not) indicating whether the two/three instances have the relation or not.
- FILTER_RELATION The FILTER_RELATION operator takes two/three lists of voxel grids of different semantic classes and a specified relation. For all possible tuples chosen from the lists, each containing the two or three instances of different semantic classes, we pass the concatenated tuple into the relation module network and collects all True/False values. We filter out all tuples with true values and returns a list of the concatednated voxel grids.
- EXIST_RELATION The EXIST_RELATION operator takes two/three lists of voxel grids of different semantic classes and a specified relation. For all possible tuples chosen from the lists, each containing the two or three instances of different semantic classes, we pass the concatenated tuple into the relation module network and collects all True/False values. We examine whether there's a tuple with true value. It's a combination of the FILTER_RELATION operator and the EXIST operator.
- COUNT_RELATION The COUNT_RELATION operator takes two/three lists of voxel grids of different semantic classes and a specified relation. For all possible tuples chosen from the lists, each containing the two or three instances of different semantic classes, we pass the concatenated tuple into the relation module network and collects all True/False values. We count how many tuples with true values we have. It's a combination of the FILTER_RELATION operator and the COUNT operator.
- RELATION_MORE The RELATION_MORE operator takes a specified relation in the comparison form (*e.g.,* closer, more left), a first voxel grid, together with two second voxel grids for comparison. For each of the two second voxel grids, it's concatenated with the first voxel grid and input into the relation network. Then we output the voxel grid with the higher score value.
- RELATION_MOST The RELATION_MOST operator takes a specified relation in the comparison form (*e.g.,* closest, leftmost), a first voxel grid, together with a list of second voxel grids for comparison. For each of the second voxel grids, it's concatenated with the first voxel grid and input into the relation network. Then we output the voxel grid with the higher score value.

- LARGER_THAN The LARGER_THAN operator takes as input two integers and returns whether the first integer is greater than the second.

- SMALLER_THAN The SMALLER_THAN operator takes as input two integers and returns whether the first integer is smaller than the second.

B.2 Baselines

CNN-LSTM, MAC & MAC(V) We use an ImageNet-pretrained ResNet-50 to extract $14 \times 14 \times 1024$ feature maps for MAC and MAC(V). We use the 2048-dimensional feature from the last pooling layer.

3D-Feature + LSTM We first use PCA to downgrade the feature size from 512 to 16. We then pass the the 3D-feature through three 3D-CNN (implemented by sparse convolutions) layers with intermediate size 64, to further downsample. We then concatenate this 3D feature with LSTM language feature and into them into a MLP to get the final answer.

ALPRO We end-to-end finetune ALPRO model on our dataset with the pre-trained checkpoint for 10 epochs. Our settings for finetuning is the same as the finetuning configurations for MSRVTT-QA in ALPRO paper. We take the 1500 answer candidates of MSRVTT-QA, and replace some irrelavent candidates with answers appear in our dataset. During inference, the model take as input image frames of shape 224×224, and output a set of classification probabilities of 1500. The predictions are obtained as the answer with the highest probability.

LGCN We take our QA-pairs as "FrameQA" task splited in LGCN paper on TGIF-QA dataset. For a QA-pair, we take image frames, each extracting 5 bounding boxes and their regional features of 1024-dim using MaskRCNN, and embed the question by word level and character level the same way as mentioned LGCN paper. Other settings for training and inference use same as FrameQA configurations.

NS-VQA We first use CLIP-LSeg to get per-pixel semantic label to perform semantic concept grounding. After Filtering with certain concepts, all the pixels that do not belong to the concepts are set to white-transparent pixels. For counting problems, we also use DBSCAN which takes the pixel x-y values concatenated by their color values as input, and assign instance clusters to all the non-transparent pixels. For training of the relation network, we use pretrained ResNet-50 to get the 2D features of the images with filtered instances, concatenate them with language features, and then go through one MLP to output a score. We "maxpool" the predictions in each image. For example, for concept questions that ask about whether there's a semantic class in the scene, we iterate through all images and get the prediction, if there's a prediction "yes" in one of the images then the final prediction is also "yes". For counting problems, we also iterate through and get integer predictions, and we get the largest prediction among all the images as our final prediction. For query problems, we iterate through all images and get the predictions which are concepts, we choose the concept which appears most frequently among the images.

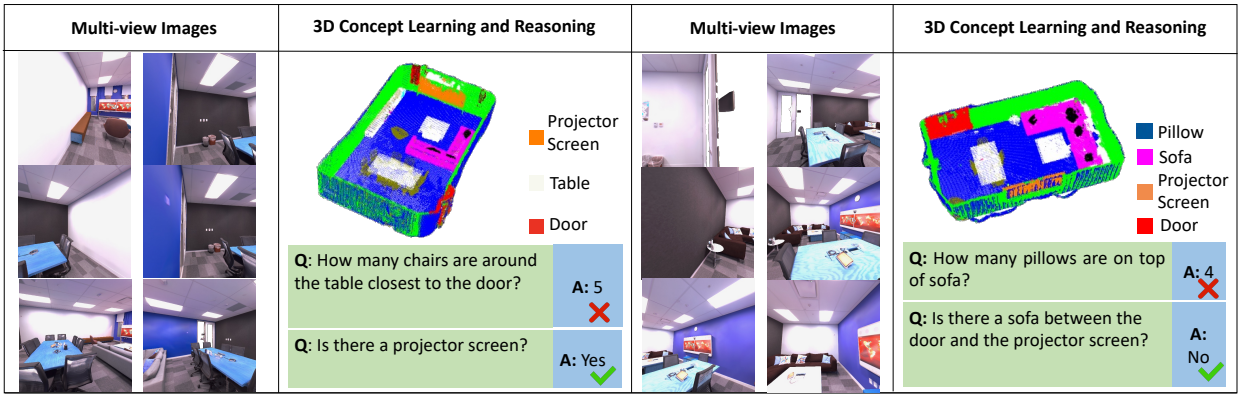


Figure 3: Qualitative examples of generalizing to Replica dataset.

C Experiments

C.1 Generalization Results

C.1.1 Generalization to Replica

Result Analysis To further show that 3D-CLR trained on HM3D can be generalized to new reasoning datasets, we further collect a small visual question answering dataset on Replica [1] with Habitat following the same data generation as HM3D, and with the same question type distribution as in Figure 1a. Table 1 shows the results. We can see that 3D-CLR can maintain the performance on Replica as it performs on HM3D, which shows 3D-CLR's good generalization ability. Specifically, 3D-CLR and NS-VQA can maintain the performance on the conceptual questions, suggesting that CLIP-LSeg is able to perform semantic concept grounding on new concepts in the new dataset. Moreover, we see that the performance on counting problem is even better than that of HM3D, this is probably because Replica scenes are simpler and contains fewer details about tiny objects, thus making instance segmentation easier. As for relational problems, 3D-CLR can also maintain good results, showing that the relation networks training on HM3D can also be utilized on other datasets and further suggesting that the vocabulary of relations is limited yet general across all scenes, and can be learned from scratch.

Methods	Concept	Counting	Relation	Comparison
MAC	55.7	16.4	40.9	58.8
MAC(V)	54.1	17.4	41.2	60.8
NS-VQA	57.2	18.7	30.4	62.3
3D-CLR	**65.3**	**45.1**	**53.6**	**73.5**

Table 1: Question-answering accuracy of 3D visual reasoning baselines on different question types when generalizing to Replica dataset.

Qualitative Examples In Figure 3, we show some qualitative examples of generalizing to Replica scenes. From the examples, we can come to several conclusions. First, 3D-CLR can perform zero-short semantic concept grounding

on unseen concepts in HM3D, such as "projector screen" and capture relations such as "between". Second, it still performs poorly in counting questions. In the example on the left, it cannot count the instances of "chairs" and on the right, it cannot count the instances of "pillows". This is because the objects are "sticked" to each other and are not separated in space. Therefore, DBSCAN cannot tell the objects apart.

C.1.2 Generalization to Unseen Concepts

To demonstrate 3D-CLR's zero-shot concept grounding ability and generalization ability, we generate some more question-answer pairs on unseen concepts. Recall that in the dataset section in the main paper, we would merge some concepts with similar concepts (*e.g.,* "stuffed animal" with "toy"). To generate the datasets with unseen concepts, we use these merged concepts instead of the concepts in the proposed 3DMV-VQA dataset. We also append some unseen concepts manually to the new dataset. We assure that there are no overlapping words between the seen concepts and unseen concepts. Table 2 shows the generalization result. We can see that 3D-CLR and NS-VQA can still have good results on the conceptual problems, suggesting that they could perform zero-shot concept grounding. However, MAC and MAC(V) has very poor performances, much worse than the performances than HM3D. This suggests that the modular design and incorporation of CLIP-LSeg equips 3D-CLR with zero-shot generalization ability.

Methods	Concept	Counting	Relation	Comparison
MAC	51.3	15.6	36.2	53.5
MAC(V)	51.4	16.1	38.5	54.2
NS-VQA	58.6	19.2	29.7	58.1
3D-CLR	**63.4**	**37.7**	**55.1**	**68.9**

Table 2: Question-answering accuracy of 3D visual reasoning baselines on different question types when generalizing to unseen categories.

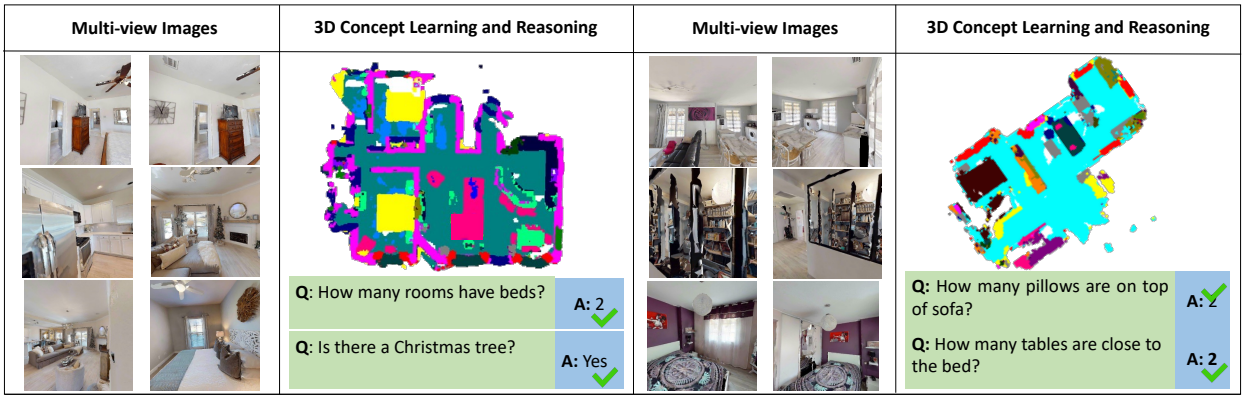


Figure 4: More Qualitative Examples on 3DMV-VQA.

C.2 More Qualitative Examples on 3DMV-VQA

In Figure 4, we show more qualitative examples on our 3DMV-VQA dataset. As we can see, 3D-CLR can generalize well to unseen concepts like "Christmas tree", and can perform well on counting problems if the instances are well

apart from each other.

References

[1] Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J. Engel, Raul Mur-Artal, Carl Yuheng Ren, Shobhit Verma, Anton Clarkson, Ming Yan, Brian Budge, Yajie Yan, Xiaqing Pan, June Yon, Yuyang Zou, Kimberly Leon, Nigel Carter, Jesus Briales, Tyler Gillingham, Elias Mueggler, Luis Pesqueira, Manolis Savva, Dhruv Batra, Hauke Malte Strasdat, Renzo De Nardi, Michael Goesele, S. Lovegrove, and Richard A. Newcombe. The replica dataset: A digital replica of indoor spaces. *ArXiv*, abs/1906.05797, 2019. 6