Multi-Agent Embodied Question Answering in Interactive Environments

Sinan Tan[1,2*] , Weilai Xiang[3*†] , Huaping Liu[1,2 ‡] , Di Guo[1,2] and Fuchun Sun[1,2]

[1] Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China

[2] Beijing National Research Center for Information Science and Technology, Beijing, 100084, China

[3] Shenyuan Honors College, Beihang University, Beijing, 100191, China

Abstract. We investigate a new AI task — Multi-Agent Interactive Question Answering — where several agents explore the scene jointly in interactive environments to answer a question. To cooperate efficiently and answer accurately, agents must be well-organized to have balanced work division and share knowledge about the objects involved. We address this new problem in two stages: Multi-Agent 3D Reconstruction in Interactive Environments and Question Answering. Our proposed framework features multi-layer structural and semantic memories shared by all agents, as well as a question answering model built upon a 3D-CNN network to encode the scene memories. During the reconstruction, agents simultaneously explore and scan the scene with a clear division of work, organized by next viewpoints planning. We evaluate our framework on the IQuADv1 dataset and outperform the IQA baseline in a single-agent scenario. In multi-agent scenarios, our framework shows favorable speedups while remaining high accuracy.

Keywords: 3D Reconstruction, Embodied Vision, Question Answering

1 Introduction

For decades, one of our best wishes has been to develop robots that can assist humans with the ability to understand the scene, to interact with environments, and to communicate with humans. For instance, a domestic robot might be asked: *How many apples are in the house?* To answer it, the agent must explore the house, open fridges & cabinets for possibly hidden apples, check the occurrence of apples, and answer the question by natural language.

This sort of problem refers to Embodied Question Answering (EQA) [4] : Being asked *What color is the car?*, an agent navigates to the car and observes

*Equal contribution.

†This work was completed while Weilai Xiang was visiting Tsinghua University, Beijing.

‡Corresponding author. hpliu@tsinghua.edu.cn

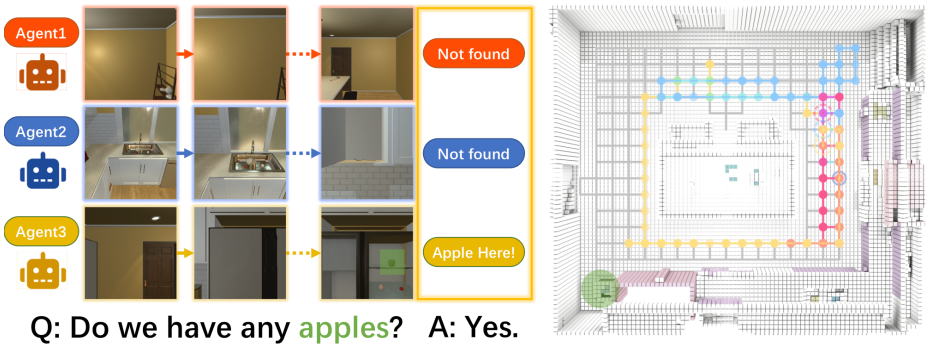


Fig. 1: A demonstration of the Multi-Agent Interactive Question Answering task. Three agents search the room simultaneously with a clear division of work, enabling them to answer the question *Do we have any apples?* more efficiently.

it before it answers the question. Since the car may be out of sight initially, the agent must have common sense about possible locations of the car and a way to get there. However, point-to-point navigation is not enough — what if we want the agent to search for a missing fork which may be *anywhere* in the kitchen?

To be more practical, Interactive Question Answering (IQA) [7] takes both interactive actions (e.g., open a cabinet) and more generic questions (e.g., *existence* and *counting*) into consideration. To answer *Is there a fork in the kitchen?*, the agent must have comprehensive cognition to the kitchen, without missing any place where the target may exist, including interactive objects like containers. However, this process could be time-costing.

Parallelism has always been a fundamental but effective idea. Since several agents can search for an object simultaneously, the question will soon be answered if multiple robots can explore collaboratively. Therefore, we introduce **Multi-Agent Interactive Question Answering**, which presents additional challenges to AI systems. **First**, the multi-agent system must be well-organized to avoid duplicate work and unbalanced work. **Second**, the multi-agent QA system must integrate information from all agents and answer the question accurately without a repeat or a miss. **Third**, the multi-agent system should achieve as high speedup as possible while keeping the high accuracy.

Very few studies have looked into multi-agent embodied question answering tasks. However, active 3D reconstruction [5][24] is not a novel problem. Here we propose a two-stage framework for Multi-Agent IQA, which firstly executes a multi-agent (embodied) 3D reconstruction to construct 3D global structural and semantic memories and secondly encodes the scene via 3D memories to answer the question. To support interactive objects, we propose a multi-layer data structure as an extension to traditional voxel-based reconstructions.

We train and evaluate our proposed two-stage framework on the IQuADv1 IQA dataset [7] in both single-agent and multi-agent scenarios and observe promising results of highly effective and efficient in both cases.

Contributions. In summary, our main contributions include:

- **Problem**. We introduce the Multi-Agent IQA, the task of organizing collaborative Interactive Question Answering for several agents.
- **Method**. We propose a two-stage framework for Multi-Agent IQA, a method to efficiently construct 3D global memories via multi-agent 3D reconstruction and to answer the question by encoding the scene memories with 3D-CNN.
- **Results**. Our 3D-memory-based framework surpasses the original IQA method in both answering accuracy and episode length, with a single agent on the IQuADv1 dataset. With 2, 3, and 4 agents, we show consistent high-level parallelism and affordable speedups in average episode length.

2 Related Work

2.1 Question Answering in Embodied Environments

Visual Question Answering. VQA requires the agent to observe the given visual contents (i.e., images [1] or videos [13][23]) and reason out the answer combining the multi-modal inputs. Common architectures for images VQA involve RNNs to encode questions, CNNs to encode images and fully connected layers to fuse language and visual features [15]. Our approach to Question Answering uses similar encoding and modality fusion strategies but uses a 3D-CNN to encode the scene with semantic memories instead of 2D-CNNs for images.
Embodied Question Answering. EQA [4] requires active perception of the environment instead of answering with images passively received. Similar to Visual Semantic Navigation [22], EQA requires the agent to navigate from the current location to the target specified by its semantic category. Some recent studies use deep Reinforcement Learning (RL) to generate navigational actions directly from visual observations [4][25]. However, for our problems which require **holistic** scene searching, point-to-point navigation is not enough.
Interactive Question Answering. IQA is an extension of EQA with actionable environments and requires the agent to discover underlying objects. The IQuADv1 dataset [7] consists of question types including *existence*, *counting* and *spatial relationship*. Therefore, it requires *holistic* scene understanding to cover all occurrences of the object instead of direct navigation to a single target. The IQA baseline maintains a 2D spatial memory to encode semantic representation at each location. However, the top-down memory may fail to complex concepts like "containing". In our work, a 3D semantic memory is constructed to provide more precise records.

2.2 Multi-Agent Systems

Multi-agent systems offer obvious advantages over single-agent ones including parallelism, robustness, scalability, and fitness for geographic distribution [18]. For IQA tasks, expecting a robot to visit every corner where the apple may occur

is unreal, but it will be possible to have several robots to answer the question quickly with parallelism when objects are scattered throughout the house.

Multi-agent reinforcement learning is a popular topic related. Some studies involve the communication of local knowledge between agents [6][16][19]. However, designing networks and protocols for communication becomes complicated for complex tasks when the number of agents increases. Meanwhile, many traditional multi-agent systems rely on optimization-based methods such as optimal mass transport [5], which can exploit collaborations of any number of agents for the 3D reconstruction task. In this paper, we adopt the optimization-based idea and formulate the multi-agent 3D reconstruction as a Set Cover Problem.

2.3 3D Computer Vision

3D Reconstruction. With RGB-D data available, 3D reconstruction becomes fundamental to 3D machine learning tasks. KinectFusion [12] is a typical real-time 3D reconstruction framework with TSDF [3] fusion. These volumetric-based methods result in voxel-wise data representing the structure of the target, denoted as "Structural Memory" in our work.

Active 3D Reconstruction. In recent years we witnessed the development of active reconstruction by robots. Quite a few studies focus on proposing a measurement (e.g., the score of uncertainty or variance) field in the 3D space and selecting Next Best Views as targets for each time step [5][24]. Inspired by these works, we evaluate voxel coverage from each view to select the next viewpoints with a set cover algorithm and assign them to agents by clustering.

3D Semantic Segmentation. With 3D datasets, 3D deep learning has made impressive progress. 3D-SIS [10] is one of those 3D instance segmentation frameworks which proposes 3D-RPN networks. Since we use 3D-CNN in question answering stage, here we use Mask R-CNN [9] to perform 2D instance segmentation and back-project the 2D semantic map to 3D voxels as Semantic Memory.

2.4 Environments and Datasets

There are several environments for embodied agents widely used such as AI2-THOR [14], Habitat [17] (a platform supporting Matterport3D [2] and Gibson [21]) and House3D [20]. However, only AI2-THOR explicitly supports multiple agents as well as interactions with objects. Therefore, we adopt the AI2-THOR interactive environment for our embodied AIs. AI2-THOR is a photo-realistic simulation environment consists of a variety of objects. We use the IQuADv1 dataset developed on AI2-THOR to evaluate our method with questions including *counting*, *existence* and *spatial relationships*. In our work, the simulator settings are slightly different from IQA [7]. We perform the OpenObject/CloseObject actions by specifying the object ID and allow agents to get the IDs from the simulator in already reconstructed areas to set free from the trouble of linking each object in 3D voxels to the object ID.

3 Overview of the Proposed Framework

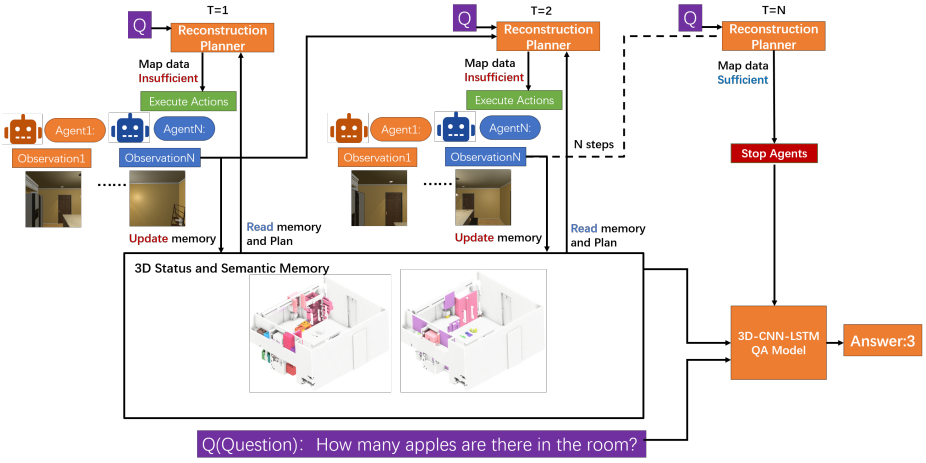


Fig. 2: Overview of the framework. The navigational actions for agents are planned step by step, according to the partially reconstructed memories. The agents execute these actions and update 3D memories along their routes. This procedure is repeated until the termination model decided there is enough data to answer the question or the whole scene is scanned. Then all agents are stopped, and the QA model encodes 3D memories and the question to predict the answer.

Our framework features enriched structural and semantic memories built along with 3D Reconstruction. Afterward, the QA model gives the answer based on memories constructed. Therefore, our framework consists of these two parts:

- **Multi-Agent 3D Reconstruction in Interactive Environments:** Our agents scan and reconstruct the interactive scene via voxel-based reconstruction, resulting in a global multi-layer structural memory. To divide labor for multiple agents and avoid duplicate work, we introduce a scalable optimization-based planner to select next-step viewpoints for each agent. They are assigned to agents and agents execute actions to navigate towards these viewpoints. During this procedure, global semantic memory is being constructed as well for semantic-related questions, by back-projecting 2D instance segmentation results to the 3D volume. Meanwhile agents open every openable object they meet and a new exclusive layer in both memories is created to record the object's inside structure and contents. After the data in memories is sufficient to answer the question, the reconstruction stops.
- **Question Answering with 3D-CNN and LSTM:** A 3D-CNN network is used to encode the semantic memory and an LSTM network is used to encode questions. Then we concatenate the semantic feature and the language feature and predict the final answer by an MLP.

4 Multi-Agent 3D Reconstruction in Interactive Environments

4.1 Data Structure in Support of Interactive Environments

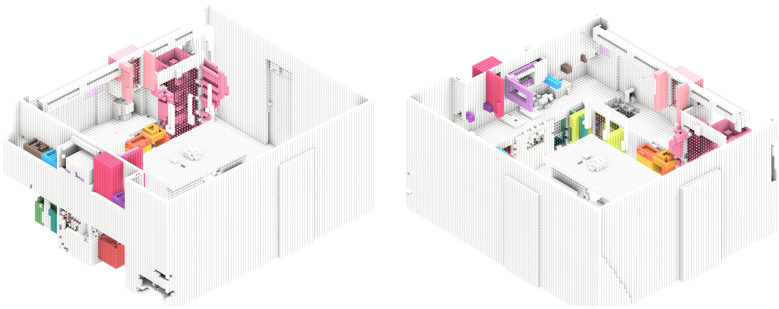


Fig. 3: The proposed multi-layer data structure from different angles. The "background" layer is demonstrated in white, recording the scene with all objects in default. The "dynamic" voxels in layers for interactive objects are shown in color. Large colored cubes represent **CONCRETE** voxels, while smaller ones are **EMPTY** voxels inside.

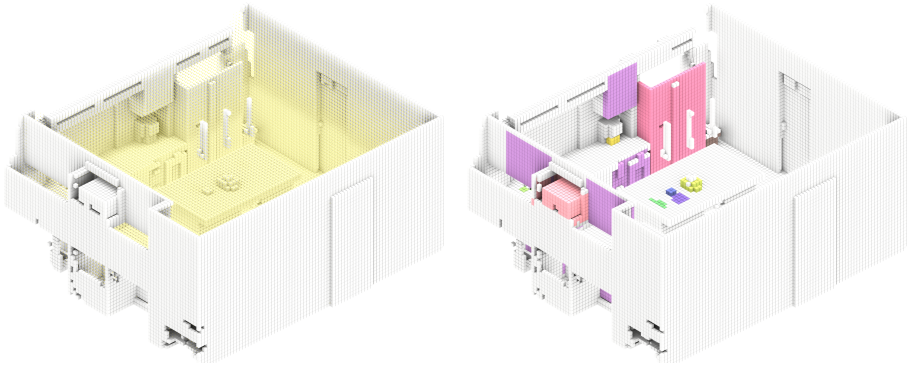
Traditional voxel-based reconstruction does not support interactive scenes, because voxels occupied by openable objects may be in different states when they are open and closed (denoted as "dynamic" voxels in our paper). However, we have to record both situations, otherwise, we will miss apples in cabinets/fridges. To address this issue, we develop an extended data structure for 3D reconstruction, introducing the concept of "layer".

For an interactive scene with M interactive objects, we use $M + 1$ layers to store its structure, where each layer is a $W \times L \times H$ array. Layer 0 represents the "background", i.e., the voxels when all openable objects are closed. Layer 1 to Layer M record M interactive objects to be open, in the order in which they are discovered during 3D reconstruction. In AI2Thor environments, all instances of certain categories (including "Fridge", "Cabinet" and "Microwave Oven") are guaranteed to be interactive (openable). Thus, when agents discover an object in those categories, a new layer is added.

This data structure is applied to both Structural Memory for 3D reconstruction and Semantic Memory for semantic(instance) segmentation in the scene.

4.2 Structural Memory and Semantic Memory

Each voxel in the multi-layer volume has multiple information stored, and one of the most important is its scan status. Here we call it the **Structural Memory**,



(a) Structural Memory (b) Semantic Memory

Fig. 4: (a) Demonstration of a fully reconstructed Structural Memory in Layer 0. Large white cubes represent **CONCRETE** voxels, while small yellow cubes represent **EMPTY** voxels. The vast parts outside the volume are **UNKNOWN**. (b) Demonstration of the corresponding Semantic Memory. Each color represents one of the 20 semantic categories in the IQuADv1 dataset. White cubes represent background voxels or voxels of other unspecified semantic categories.

which monitors and records whether a voxel is scanned. This information is crucial to plan actions for the agents. To be specific, we assign a status to each voxel in the multi-layer volume, which is one of these following statuses:

- **UNKNOWN**: The initial status of all voxels, representing that the voxel has not been scanned yet.
- **EMPTY**: Indicating that in all scans involving this voxel, no object is found.
- **CONCRETE**: Indicating that in at least one scan, a concrete object occupies this voxel.

Hence, the complete structure of the scene is modeled via this voxel-based memory. However, the Structural Memory itself only records the geometrical structure, which is not enough for downstream tasks (i.e. Question Answering in our case). Therefore a piece of extra information is recorded, named **Semantic Memory**.

For each scan of the 3D scene, an instance segmentation model will be applied to the observed RGB image. These semantic labels acquired is written into the Semantic Memory by back projecting labels to all "CONCRETE" voxels. The Semantic Memory provides visual information of the 3D scene, therefore it is used for question answering in our QA model.

4.3 Scanning Boundaries and Scanning Tasks

Scanning boundaries are voxels at the border of the scanned part and the unscanned part of a layer in the 3D scene. Denoting $status(l, v)$ to be the status

of a voxel with coordinate $v = (x, y, z)$ on layer l, the actions of agents are planned according to these two kinds of scanning boundaries:

- Scene Scanning Boundaries (B_S), the border between scanned and unscanned parts in Layer 0.

$$B_S = \left\{ v \;\middle|\; \begin{array}{l} v \in \mathcal{V}, \\ status(0, v) = \text{UNKNOWN}, \\ \text{CONCRETE} \in status(0, Adj(v)) \end{array} \right\} \tag{1}$$

- Interactive Scanning Boundaries (B_I), the border between scanned and unscanned voxels in an object's "dynamic" part. For example, when a cabinet is open, the border between the scanned part inside the cabinet and the unscanned part of the cabinet is considered Interactive Scanning Boundary.

$$B_I = \left\{ v \;\middle|\; \begin{array}{l} v \in \mathcal{V}, \\ status(i, v) = \text{UNKNOWN}, \\ \exists v_a \in Adj(v), \\ status(i, v_a) \neq status(0, v_a), \\ status(i, v_a) = \text{CONCRETE} \end{array} \right\} \tag{2}$$

One task for our agents is to cover voxels on scanning boundaries which represent the unfinished parts of reconstruction. Furthermore, agents must visit the unopened interactive objects (e.g. cabinets that have never been opened so far) because these unopened objects have no scanning boundaries yet. Therefore a new kind of task is created to ensure they will be opened at least once, and the **scanning tasks** are:

- Voxels on Scanning Boundaries ($B_S \cup B_I$): They need to be observed on later scans to complete the memories.
- Unopened interactive objects (T_U): These objects must be opened at least once to create new layers in memories and examine their inside.

Therefore scanning tasks can be formulated as a set of voxels to be scanned:

$$T = B_S \cup B_I \cup T_U \tag{3}$$

4.4 Viewpoint-Voxel Coverage Matrix

The target of the planning algorithm for scene reconstruction is to move the agent. Yet in our work we do not plan every single specific move, instead, we choose the target of a series of moves. To be convenient, we denote the discretized observed part of the scene as V, which contains all possible *viewpoints*. A viewpoint is the combination of position and rotation of the camera.

Then for every viewpoint, we compute the visible voxels from it according to the reconstruction of the scene. Then we construct a matrix about whether a voxel in scanning tasks T can be seen from a viewpoint in V. This is denoted as C, a $|V| \times |T|$ matrix, the **Viewpoint-Voxel Coverage Matrix**.

4.5 Termination Condition

With the definition of scanning tasks, it's obvious that, when there are no remaining tasks for interactive scene reconstruction, the reconstruction process can be terminated. Therefore, the termination condition can be formulated as:

$$T = \emptyset \tag{4}$$

Besides, for some questions, the scan can be terminated before the environment is completely scanned. We propose a special learning-based Termination Model in Section 5 to achieve the early stopping.

4.6 Multi-Agent 3D Reconstruction

After introducing the data structures, tasks, and termination conditions, we propose the multi-agent reconstruction algorithm that repeats the routes planning procedure over and over again until either termination condition is satisfied.

In each iteration, we investigate the voxels to be scanned next and assign them to agents. They are introduced as Scanning Tasks and can be retrieved from the semi-finished Structural Memory. Afterward, we evaluate the visibility of those voxels from each possible viewpoint. To avoid duplicate work and maximize the efficiency, we expect agents to cover as more Scanning Tasks voxels as possible while having little intersection, so we convert it into a Set Cover Problem and solve it by a greedy algorithm.

To regroup the selected viewpoints into N groups (suppose that we have N agents), we use the K-means algorithm to execute a spatial clustering. Then we assign a cluster to each agent by solving a Balanced Assignment Problem to minimize the total route length from current locations to their target viewpoints. We plan the route for each agent with a TSP solver. Agents execute actions to navigate along the route and update the Structural Memory and Semantic Memory. After an agent reaches a viewpoint or when an agent gets stuck due to wrong route planning, we clear the routes and repeat the procedure to re-plan the moves.

A 2D map is maintained according to the reconstructed 3D scene, determining which location the agent can pass through. This 2D map is used for route planning. At each time step, the 2D map is updated, and all newly discovered 3D objects will be created a new layer for, and added to the multi-layer 3D data structure.

Algorithm 1: Multi-Agent 3D Reconstruction

Result: SceneMap
Initialize SceneMap;
while $T \neq \emptyset \wedge \neg$ *TerminationModel(SceneMap)* **do**
  Generate B_S, B_I, and T_U;
  Generate $T = B_S \cup B_I \cup T_U$ and Count all viewpoints V;
  Evaluate the coverage of T from each viewpoint to form matrix C;
  Choose a subset $v \subseteq V$ by running Set Cover Problem solver on C;
  Regroup v into N clusters: $v = \cup_{j=1}^{N} v_j$ by K-means Algorithm, and assign v_j to agent A_i by Hungarian Algorithm;
  Plan route for agent A_i to travel a series of viewpoints $v_j = \{v_{j,1}, v_{j,2}, ..., v_{j,n_j}\}$ with TSP solver;
  repeat
    Execute actions along the planned routes;
    Update the SceneMap for each step;
    Add new layers to for newly discovered interactive objects on SceneMap;
    Update the 2D Map for navigation according to SceneMap;
  until *An agent reaches a selected viewpoint* $\vee$ *One gets blocked*;
end

5 Question Answering with 3D-CNN and LSTM

5.1 3D-CNN Scene Encoder and Question Encoder

The question answering model generates the answer according to the semantic volume and the given question. Here the question, denoted as Q, is encoded with an LSTM, getting the question feature f_Q. For CNN, we first need to process all the observations we get in the process of 3D reconstruction.

Imagine a voxel $v = (x, y, z)$ in the 3D volume with shape $W \times L \times H$, where W, L, H is the size of each dimension, respectively. Then this voxel v must have been observed several times in the multi-layer Semantic Memory with shape $M \times W \times L \times H$, not only from the multi-layer scans but also within each layer. We denote the total number of observations to v in Semantic Memory layers with class label c as $N(v, c)$, and each observation has a confidence score of $s_i(v, c)$. We build a tensor $\mathcal{V}$ with shape $C \times W \times L \times H$ to integrate all information about voxel v by averaging all these observations, i.e.:

$$\mathcal{V}(c, x, y, z) = \frac{1}{N(v, c)} \left(\sum_{i=1}^{N(v,c)} s_i(v, c) \right) \tag{5}$$

We encode the semantic map with a 3D-CNN network similar to ResNet-18, yet replacing all 2D Convolution layers with 3D Convolution layers, yielding the scene feature vector f_S. Since the 3D volume can be huge, we use submanifold sparse convolutions [8] instead of traditional convolutions to process those sparse data.

5.2 Question Answering Model

Here we concatenate the scene feature vector f_S and the language feature vector f_Q and use a simple multi-layer perceptron (MLP), to get the joint representation h of the scene and the question.

$$h = \text{MLP}([f_S; f_Q]) \tag{6}$$

Finally, a fully connected layer is applied to produce the probability distribution of the final answer.

$$p(ans) = \text{Softmax}(W_a h + b_a) \tag{7}$$

5.3 Termination Model

Similarly, we apply another fully connect layer to predict the probability for agents to stop:

$$p(stop) = \text{Softmax}(W_s h + b_s) \tag{8}$$

5.4 Training the QA Model and the Termination Model

Training the QA Model and the Termination Model is not a trivial task. Among thousands of voxels, sometimes there could be only less than ten voxels related to a given question. With such sparsity of interested voxels, end-to-end training of the QA network does not work in our experiments. Below are the steps we go through to train the QA model.

Pretraining the Instance Segmentation Model We use Mask R-CNN for instance segmentation. The Mask R-CNN is trained on more than 10k images sampled from the 3D scenes in the training set, with annotations automatically generated from the output of the simulators. The pretrained model achieves 56.7% mAP on our validation set.

Preparing training data For each question in the IQuADv1 dataset, we perform scene reconstruction with the proposed interactive reconstruction algorithm to generate semantic memory for 3D scenes in the IQuADv1 dataset. Since the termination model may decide to early stop the navigation, the intermediate reconstruction results are also saved for the QA model. This provides data for pretraining the 3D-CNN. Here we use the ground truth segmentation provided by the AI2Thor simulator.

Pretraining the 3D-CNN and the LSTM End-to-end training of the 3D-CNN & LSTM QA model from scratch is very hard to converge. Therefore, we split these two networks. For 3D-CNN network, we add three auxiliary branches, corresponding to three different kinds of questions in the IQuADv1 dataset. These branches respectively predict whether an object of a category exists, the number of objects of that category, and all containers holding that category of objects. We design loss functions for these three branches and pretrain the 3D-CNN alone. For the LSTM for language understanding, we pretrain it in the

way similar to IQA, which is, using fully-connected layers on f_Q to predict the question type and all involved object categories corresponding to the type.
Training the models The weights of pretrained networks are transferred to the 3D-CNN-LSTM QA Network. The whole QA network is then trained with the answers as supervision. The Termination Model is trained in a similar way, with the supervision being no further exploration is needed for a given semantic memory and question (e.g. for "existence" problems, the reconstruction process can stop immediately when the object we are interested in is already found).

6 Experimental Results

6.1 Single-Agent IQA

To investigate the performance of our 3D-memory-based QA framework, we perform experiments with single-agent set-ups and compare it with the original IQA model proposed in [7]. Results are shown in Table 1. Accuracy and episode length for three question types in IQuADv1 dataset are reported separately.

Table 1: Experiment results with single-agent set-up

	Existence		Counting		Containing	
Model	Accuracy	Length	Accuracy	Length	Accuracy	Length
IQA (GT Detection) [7]	86.56%	679.70	35.31%	604.79	70.94%	311.03
IQA (Pred. Detection) [7]	68.47%	318.33	30.43%	926.11	58.67%	516.23
Human [7]	90.00%	58.40	80.00%	81.90	90.00%	43.00
Ours (GT Segmentation)	98.75%	166.31	88.28 %	237.40	91.88%	195.89
Ours (Pred. Segmentation)	79.53%	159.85	45.62%	220.95	77.50%	204.87

IQA (Pred. Detection) uses predicted depth, while others use GT depth

When ground truth (GT) semantic segmentation is available, our proposed framework not only outperforms the baseline method with GT detections, but also achieves higher accuracy than humans, showing the potential advantages of our model. Still, it takes more actions than humans to answer a question, indicating that its efficiency can be improved.

When replacing GT segmentation with results predicted by Mask R-CNN, we notice obvious performance drops. It indicates that the bottleneck of our method is the accuracy of semantic segmentation. With more advanced methods such as multi-view based image segmentation, our method may perform better.

However, even with predicted segmentation, the overall performance of our method still outperforms the IQA baseline with GT detection (better in *Counting* and *Containing*, worse in *Existence*), showing that the rest part of our model is robust enough to tolerate imperfect segmentation. Note that we use GT depth

images to perform the 3D Reconstruction, but the IQA model noted as "GT Detection" uses GT depth as well. Since our method doesn't require very high reconstruction precision, noisy depth sensor is unlikely to cause severe problems. However, when it comes to predicted depth data, registration between different predicted depth frames, or MVS-based techniques like [11] would be required.

6.2 Multi-Agent IQA on IQuADv1 Dataset

Table 2: Multi-agent experiments with N agents

	Existence		Counting		Containing		Overall		Overall Speedup		
N	Acc.	Length	Acc.	Length	Acc.	Length	Acc.	Length	Ideal	Actual	% of Ideal
1	79.53%	159.85	45.62%	220.95	77.50%	204.87	67.55%	195.22	1.0	1.00	100%
2	78.59%	93.99	46.56%	127.57	77.50%	116.69	67.55%	112.75	2.0	1.73	87%
3	78.28%	69.24	45.47%	90.33	77.03%	86.36	66.93%	81.98	3.0	2.38	79%
4	78.44%	60.41	43.91%	79.63	76.25%	75.46	66.20%	71.83	4.0	2.72	68%

We test the proposed framework on multi-agent set-ups. The results are shown in Table 2. There are no significant differences in *accuracy* for different numbers of agents, which indicates that our constructed semantic memory is sufficient and stable. Despite the similar accuracy, it takes much fewer steps for each agent to finish the task when more agents are available.

To examine the parallelism in our framework, we calculate the *speedup in length* with 2, 3, and 4 agents. The length is the maximum number of actions taken among all agents. When the task assignment is unbalanced, some agents may be in heavy load while other agents are idle. Therefore using the maximum number of actions as the metric can more accurately measure the speedup in terms of time consumption. As the number of agents increases, the speedup also becomes higher. However, the percentage of the actual speedup compared to the ideal speedup drops from 87% to 68%, when the number of agents increases from 2 to 4. These results show that our clustering and task assignment based multi-agent schedule algorithm is scalable, but still has room for improvement.

6.3 Qualitative Examples

To illustrate how our agents navigates in Embodied Environments, we select a question within the IQuADv1 dataset — *How many eggs are there in the room?*. The question is answered correctly with 1, 2, 3 and 4 agents set-ups. All agents are spawn at the same location as defined in the IQuADv1 dataset. We record the track of each agent and visualize it in the following figure.

As is shown in Fig. 5, the searching area of agents is relatively scattered. For example, with three agents (colored in red, yellow and blue), they act in

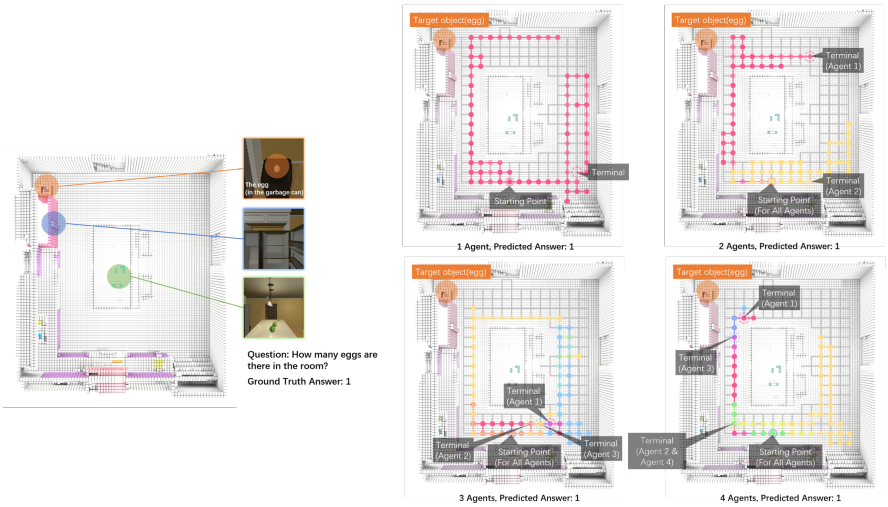


Fig. 5: A qualitative example with the question and some rendered image of the scene on the left side. The track of each agent, with single-agent set-up and multi-agent set-up with 2, 3 and 4 agents are shown on the right.

the bottom part, the top-left part and the right part of the room respectively, indicating that the proposed task assignment algorithm is effective in allocating the scanning tasks to each agent. With more agents joining the reconstruct process, their searching area has more overlaps at the left and bottom part of the scene. That's reasonable because more interactive objects (mainly cabinets) exist in this region, and agents are required to head to this region when other parts have been fully reconstructed.

7 Conclusion

In this work, we introduce a new task of Multi-Agent Interactive Question Answering. We propose a novel two-stage framework to solve this problem, where firstly Multi-Agent 3D Reconstruction is performed to build a semantic memory, and then a 3D-CNN based QA model is used to generate the answer. Experiments show that our framework achieves high accuracy with single-agent set-up, and it is scalable to extend to multi-agent scenarios. Additionally, with GT semantic segmentation our proposed framework surpasses human performance, indicating that accurate 3D Semantic Segmentation is the bottleneck in our method.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China under Grants U1613212 and 61703284.

References

1. Antol, S., Agrawal, A., Lu, J., Mitchell, M., Batra, D., Lawrence Zitnick, C., Parikh, D.: Vqa: Visual question answering. In: Proceedings of the IEEE international conference on computer vision. pp. 2425–2433 (2015)
2. Chang, A., Dai, A., Funkhouser, T., Halber, M., Niessner, M., Savva, M., Song, S., Zeng, A., Zhang, Y.: Matterport3d: Learning from rgb-d data in indoor environments. arXiv preprint arXiv:1709.06158 (2017)
3. Curless, B., Levoy, M.: A volumetric method for building complex models from range images. In: Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques. pp. 303–312 (1996)
4. Das, A., Datta, S., Gkioxari, G., Lee, S., Parikh, D., Batra, D.: Embodied question answering. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. pp. 2054–2063 (2018)
5. Dong, S., Xu, K., Zhou, Q., Tagliasacchi, A., Xin, S., Nießner, M., Chen, B.: Multi-robot collaborative dense scene reconstruction. ACM Transactions on Graphics (TOG) **38**(4), 1–16 (2019)
6. Foerster, J., Assael, I.A., De Freitas, N., Whiteson, S.: Learning to communicate with deep multi-agent reinforcement learning. In: Advances in neural information processing systems. pp. 2137–2145 (2016)
7. Gordon, D., Kembhavi, A., Rastegari, M., Redmon, J., Fox, D., Farhadi, A.: Iqa: Visual question answering in interactive environments. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4089–4098 (2018)
8. Graham, B., van der Maaten, L.: Submanifold sparse convolutional networks. arXiv preprint arXiv:1706.01307 (2017)
9. He, K., Gkioxari, G., Dollar, P., Girshick, R.: Mask r-cnn. In: The IEEE International Conference on Computer Vision (ICCV) (Oct 2017)
10. Hou, J., Dai, A., Niessner, M.: 3d-sis: 3d semantic instance segmentation of rgb-d scans. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2019)
11. Huang, P.H., Matzen, K., Kopf, J., Ahuja, N., Huang, J.B.: Deepmvs: Learning multi-view stereopsis. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2821–2830 (2018)
12. Izadi, S., Kim, D., Hilliges, O., Molyneaux, D., Newcombe, R., Kohli, P., Shotton, J., Hodges, S., Freeman, D., Davison, A., et al.: Kinectfusion: Real-time 3d reconstruction and interaction using a moving depth camera. In: Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology. pp. 559–568 (2011)
13. Jang, Y., Song, Y., Yu, Y., Kim, Y., Kim, G.: Tgif-qa: Toward spatio-temporal reasoning in visual question answering. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (July 2017)
14. Kolve, E., Mottaghi, R., Han, W., VanderBilt, E., Weihs, L., Herrasti, A., Gordon, D., Zhu, Y., Gupta, A., Farhadi, A.: Ai2-thor: An interactive 3d environment for visual ai. arXiv preprint arXiv:1712.05474 (2017)
15. Malinowski, M., Rohrbach, M., Fritz, M.: Ask your neurons: A neural-based approach to answering questions about images. In: The IEEE International Conference on Computer Vision (ICCV) (December 2015)
16. Mousavi, H.K., Nazari, M., Takáč, M., Motee, N.: Multi-agent image classification via reinforcement learning. arXiv preprint arXiv:1905.04835 (2019)

17. Savva, M., Kadian, A., Maksymets, O., Zhao, Y., Wijmans, E., Jain, B., Straub, J., Liu, J., Koltun, V., Malik, J., et al.: Habitat: A platform for embodied ai research. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 9339–9347 (2019)
18. Stone, P., Veloso, M.: Multiagent systems: A survey from a machine learning perspective. Autonomous Robots **8**(3), 345–383 (2000)
19. Sukhbaatar, S., Fergus, R., et al.: Learning multiagent communication with backpropagation. In: Advances in neural information processing systems. pp. 2244–2252 (2016)
20. Wu, Y., Wu, Y., Gkioxari, G., Tian, Y.: Building generalizable agents with a realistic and rich 3d environment. arXiv preprint arXiv:1801.02209 (2018)
21. Xia, F., Li, C., Chen, K., Shen, W.B., Martın-Martın, R., Hirose, N., Zamir, A.R., Savarese, L.F.F.S.: Gibson env v2: Embodied simulation environments for interactive navigation (2019)
22. Yang, W., Wang, X., Farhadi, A., Gupta, A., Mottaghi, R.: Visual semantic navigation using scene priors. arXiv preprint arXiv:1810.06543 (2018)
23. Zhao, Z., Yang, Q., Cai, D., He, X., Zhuang, Y., Zhao, Z., Yang, Q., Cai, D., He, X., Zhuang, Y.: Video question answering via hierarchical spatio-temporal attention networks. In: IJCAI. pp. 3518–3524 (2017)
24. Zheng, L., Zhu, C., Zhang, J., Zhao, H., Huang, H., Niessner, M., Xu, K.: Active scene understanding via online semantic reconstruction. In: Computer Graphics Forum. vol. 38, pp. 103–114. Wiley Online Library (2019)
25. Zhu, Y., Mottaghi, R., Kolve, E., Lim, J.J., Gupta, A., Fei-Fei, L., Farhadi, A.: Target-driven visual navigation in indoor scenes using deep reinforcement learning. In: 2017 IEEE international conference on robotics and automation (ICRA). pp. 3357–3364. IEEE (2017)