OpenEQA: Embodied Question Answering in the Era of Foundation Models

Supplementary Material

A. Acknowledgements

The Georgia Tech effort was supported in part by ONR YIP and ARO PECASE. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the U.S. Government, or any sponsor.

B. OpenEQA Benchmark Details

This section provides further details on the construction of the OpenEQA benchmark (Sec. 2.3). Specifically, we describe the process for generating human-like episode histories H for EM-EQA (Appendix B.1), the interface for collecting question-answer pairs (Q, A^*) (Appendix B.2), and the interface used to validate the dataset (Appendix B.3).

B.1. Generating Episode Histories H

Episode histories H provide agents with observations of the environment, and are used for the EM-EQA split of OpenEQA in both ScanNet and HM3D environments (see Sec. 1.1). The ScanNet dataset was originally collected by people who were asked to scan indoor environments with an RGB-D camera. We use the initial 30 seconds (or 600 frames) of these human trajectories from ScanNet as EM-EQA episode histories H.

HM3D consists of scanned 3D environments, but does not come with pre-collected environment tours. Thus, we generate episode histories H using a two-step, semi-automated process. First, we generate a shortest-path trajectory from a starting location x_{src} to a destination x_{dst} in the environment. We select locations such that the geodesic distance between x_{src} and x_{dst} is $> 10m$ and the path curves (enforced by the criteria that the geodesic path distance $\geq 1.1 \times$ Euclidean path distance). Under these constraints, the paths typically traverse multiple rooms in the environment. To collect an episode history H, an agent travels along the path, while scanning the scene every $1m$ by rotating up to $180°$. These scans are intended to mimic human-like exploration behavior. After collecting the trajectories, we manually inspect each trajectory to ensure they properly explore the scene; we exclude trajectories with extended periods closely facing walls. This process results in one episode history H for 63 different HM3D validation environments.

B.2. Collecting Question-Answer Pairs (Q, A^*)

We use a Google Form to collect question-answer pairs (Q, A^*) annotations from 8 different AI researchers. Specifically, the annotators watch a video of a given episode history H, and generate questions for the 7 categories listed in Sec. 2.3. In the form, each category is described and one to two *good* and *bad* example questions are provided.

B.3. Interface for Dataset Validation

After the initial collection of question-answer pairs Q, A^*, we ask two independent people to validate each question. Specifically, the validators are shown an episode history H and a corresponding question Q on a simple HTML page. They are asked to provide an answer or mark the question as invalid (i.e. ambiguous or unanswerable). For the subset of *object localization* questions, we ask the validators to provide two answers for each questions because referring expressions often have multiple valid options (e.g. an object may be both *'left of the sink'* and *'right of the stove'*). We remove any question marked invalid by either validator.

C. LLM-based Evaluation Details

OpenEQA questions often require open-ended answers, we use an LLM to evaluate correctness of answer produced by EQA agents. We prompt an LLM to compare human annotated answer A_i^* and model output A_i given a question Q_i and output a score σ_i on a scale of 1 to 5. On this scale, 1 indicates an incorrect response, 5 is a correct response and intermediate values represent different levels of similarity. Since questions can often have multiple correct answers, we also provide extra answers to the LLM prompt during scoring. We show the LLM prompt in Figure 6. Given the scores σ_i, we calculate an aggregate LLM-based **correctness** metric (LLM-Match) using Equation (1).

D. EQA Agent Function Signatures

In this section, we describe the function signature that is expected from an agent by OpenEQA benchmark.

Box 1 describes the function signature for the EM-EQA task. An agent is expected to produce a text answer to a question based on an episode history. The episode history generally consists of RGB, depth, camera pose, and camera intrinsic information. The benchmark does not prescribe any specific way to use the history. A variety of different approaches and representations of the history can be pursued by researchers, such as point clouds, NeRFs, or instance maps. Since all methods have the same set of episode history information at their disposal, it allows for a fair comparison of methods. The final natural language answer is evaluated using LLM-Match metric described in Sec. 2.4 and Appendix C.

Figure 6. **Prompt used for LLM-Match scoring.** The placeholders {question}, {answer}, {extra_answers}, and {prediction} are replaced by the question Q, ground truth answer A^*, additional answer, and the agent's predicted answer A, respectively. Note that the extra answers are only available for *object localization* questions. When not available, corresponding sections of the prompt are omitted.

```
You are an AI assistant who will help me to evaluate the response given the question,
the correct answer, and extra answers that are also correct.  To mark a response, you
should output a single integer between 1 and 5 (including 1, 5).  5 means that the
response perfectly matches the answer or any of the extra answers.  1 means that the
response is completely different from the answer and all of the extra answers.

Example 1:
Question:  Is it overcast?
Answer:  no
Extra Answers:  ['doesn't look like it', 'no',' it's sunny']
Response:  yes
Your mark:  1

Example 2:
Question:  Who is standing at the table?
Answer:  woman
Extra Answers:  ['a woman', 'a lady', 'woman']
Response:  Jessica
Your mark:  3

Example 3:
Question:  Are there drapes to the right of the bed?
Answer:  yes
Extra Answers:  ['yes, there are drapes', 'yeah', 'the drapes are to the right of the
king bed']
Response:  yes
Your mark:  5

Your Turn:
Question:  {question}
Answer:  {answer}
Extra Answers:  {extra_answers}
Response:  {prediction}
```

Similarly, Box 1 also describes the expected function signature for A-EQA task. Here, an agent does not receive an episode history and must generate its own experience through exploration. To allow standardization, we provide access to the simulation environment and start state as part of the benchmark. The state allows for instantiating an environment and fixing the starting location of the agent and various objects. We do not prescribe a specific navigation API for the benchmark, researchers are free to pursue different abstractions such as atomic navigation actions or navigation skills, as long as it doesn't use any privileged simulation information. The final answer is evaluated for correctness using LLM-Match, and the efficiency (see Sec. 2.5) is computed using the number of atomic actions taken by the agent (to allow for standardization).

E. Baseline Agent Details

This section provides additional details and LLM prompts for the blind LLM baseline (Appendix E.1), Socratic LLM w/ Frame Captions example (Appendix E.2), and GPT-4V (Appendix E.3).

E.1. Blind LLM Prompt and Details

The prompt used for both our LLaMA-2 and GPT-4 blind LLM baselines is illustrated in Fig. 7. We use the 70B parameter version of LLaMA-2 that is fine-tuned for chat, and the `gpt-4-0613` version of GPT-4.

E.2. Socratic LLM w/ Frame Captions Example

Figure 8 shows how Socratic LLM w/ Frame Captions baseline produces an answer to a question given K frames sam-

Algorithm 1 EQA Agent Signatures

```
def EMEQA_Agent(Q: str, H: dict) -> str:
    """ Function signature for EM-EQA Agents

    Args:
    - Q: EQA question
    - H: episodic memory (history)
        - keys -> rgb: image,
                  depth: image,
                  c_pose: camera pose,
                  c_in: camera intrinsics
        - H["rgb"] = np.array(T, H, W, 3)
        - H["depth"] = np.array(T, H, W, 1)
        - H["c_pose"] = np.array(T, 6)
        - H["c_in"] = np.array(T, 6)

    Returns:
    - answer: natural language
    """
    ...

    return answer

def AEQA_Agent(Q: str,
        S: dict) -> Tuple[str, int]:
    """ Function signature for A-EQA Agents

    Args:
    - Q: EQA question
    - S: initial state of simulator
        - keys -> metadata
        - S["metadata"] = Dict[str, Any]

    Returns:
    - answer: natural language
    - T: episode lifetime. Timesteps
         taken to answer the question
    """

    env = make_env(S["metadata"])
    env.set_state(S)
    ...
    return answer, T
```

pled from episodic memory H. We use LLaVa-1.5 to generate image captions. We use the 70B parameter version of LLaMA-2 that is fine-tuned for chat, and the `gpt-4-0613` version of GPT-4 for large language model.

E.3. GPT-4V Details

Given an episodic memory H, we draw K frames and pass it to GPT-4V (through the API) in addition to question Q and prompt ω. We use chain-of-thought prompting in ω. We choose $K = 50$ for EM-EQA and $K = 75$ for A-EQA. Figure 9 shows the prompt ω and the input format passed to GPT-4V.

F. Sparse Voxel Maps

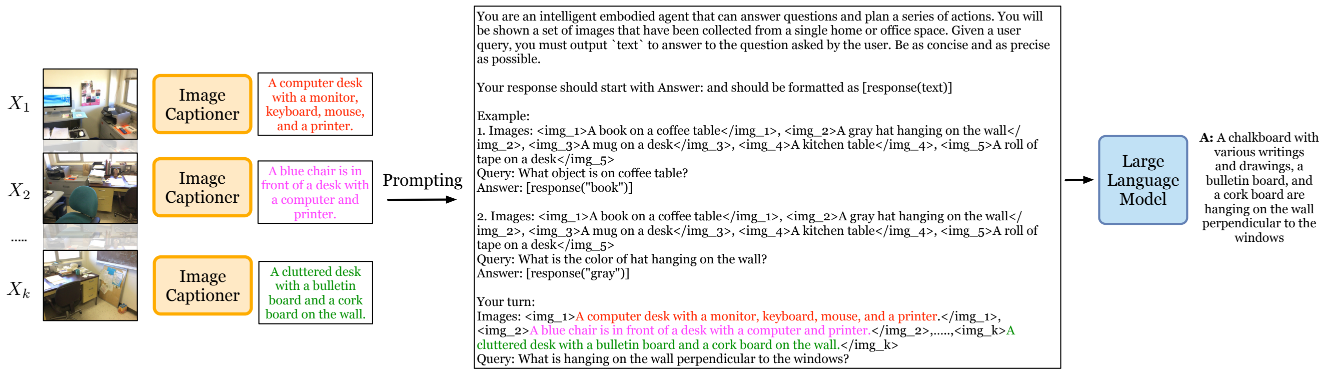
For building SVM, we use K uniformly-sampled frames from the episode history H. K varies across difference scenes but the principle is to find the minimum number of K (for reducing the run-time memory consumption) to cover the whole environment. We process each sampled frame with the following two steps:

Step 1. Detecting object views in the frame using Detic. Each object view v is a tuple of $\langle c, b\rangle$, where c is the 2D image crop of the object and b is the 3D bounding box in the world coordinate system. We first extract object masks from the frame by setting the vocabulary for Detic to more than 500 household object categories. Then we get the image crop c around each detected mask with an additional margin. We then use depth information to get a 3D point cloud where we run DBSCAN [10] to further filter out background points, and compute the bounding box b. Note that we only consider depths that are in the range of $[0.1m, 4m]$.

Step 2. Associating each object view v with a global object instance o. Most objects will be detected in more than one frame, and a main goal of SVM is to de-duplicate object views to create global object instances. Each global object instance o is a tuple of $\langle C, b^*\rangle$, where C is a list a image crops (i.e., c) from multiple viewpoints (i.e., v), and b^* is a re-computed 3D bounding box from a concatenated point cloud of different views. For matching v to o, SVM considers 3D bounding box overlapping and CLIP [35] embedding similarity.

Figure 7. **Prompt used for Blind LLM baselines.** The placeholder {`question`} is replaced by the question Q. The same prompt is used for LLaMA-2 and GPT-4.

```
You are an intelligent question answering
agent. I will ask you questions about
an indoor space and you must provide an
answer.

If the question does not provide enough
information to properly answer, provide
an appropriate guess.

Q: What machine is on top of the stove?
A: The microwave
Explanation: stoves are typically found
in kitchens and near microwaves.

Q: What piece of furniture is in the
middle of the bedroom?
A: a bed
Explanation: bedrooms almost always
contain a bed.

Q: Is the door open or closed?
A: open
Explanation: the door can be in either
state, so we just randomly pick one.

Q: {question}
```


X_1
X_2
.....
X_k
Image Captioner
Image Captioner
Image Captioner
A computer desk with a monitor, keyboard, mouse, and a printer.
A blue chair is in front of a desk with a computer and printer.
A cluttered desk with a bulletin board and a cork board on the wall.
Prompting
Large Language Model
A: A chalkboard with various writings and drawings, a bulletin board, and a cork board are hanging on the wall perpendicular to the windows

```
You are an intelligent embodied agent that can answer questions and plan a series of actions. You will
be shown a set of images that have been collected from a single home or office space. Given a user
query, you must output `text` to answer to the question asked by the user. Be as concise and as precise
as possible.

Your response should start with Answer: and should be formatted as [response(text)]

Example:
1. Images: <img_1>A book on a coffee table</img_1>, <img_2>A gray hat hanging on the wall</
img_2>, <img_3>A mug on a desk</img_3>, <img_4>A kitchen table</img_4>, <img_5>A roll of
tape on a desk</img_5>
Query: What object is on coffee table?
Answer: [response("book")]

2. Images: <img_1>A book on a coffee table</img_1>, <img_2>A gray hat hanging on the wall</
img_2>, <img_3>A mug on a desk</img_3>, <img_4>A kitchen table</img_4>, <img_5>A roll of
tape on a desk</img_5>
Query: What is the color of hat hanging on the wall?
Answer: [response("gray")]

Your turn:
Images: <img_1>A computer desk with a monitor, keyboard, mouse, and a printer.</img_1>,
<img_2>A blue chair is in front of a desk with a computer and printer.</img_2>,.....,<img_k>A
cluttered desk with a bulletin board and a cork board on the wall.</img_k>
Query: What is hanging on the wall perpendicular to the windows?
```

Figure 8. **Input example for Socratic LLMs w/ Frame Captions baseline.** We first caption each of the K frames with an image captioner and then prompt the LLM with those captions along with the question. The large langauge model produces an answer.

Figure 9. GPT4V input prompt.

```
You are an intelligent embodied agent
that can answer questions.  You will
be shown a set of images that have been
collected from a single location.  Given
a user query, you must output 'text' to
answer to the question asked by the user.

User Query: {question}
Think step by step.
```

After SVM is constructed, we then select the best crop from C per global instance o, where the object mask takes up the largest number of pixels. Each selected crop is passed to LLAVA-1.5 [26] to get the textual description, and all the descriptions with the instances' 3D coordinates (center of the bounding box b^*) are wrapped in a prompt for an LLM to answer the question Q. Limited by the LLM's capacity, we only consider topN ($N = 75$) instances ranked by the CLIP similarity between their visual feature and Q from all the instances we detect in SVM.

G. Force-A-Guess Details

As discussed in Sec. 3, we force baseline agents to guess an answer if they initial abstain – i.e. respond with an explanation for why the question is unanswerable. Specifically, we first ask an LLM if the initial answer is an abstaining response, and if so we replace the answer with a guess from a blind LLM. For step 1, use the prompt shown in Fig. 10. We provide a comparison baseline performance with and without this procedure in Appendix H.

H. Force-A-Guess Results

In Tab. 3, we present results illustrating the performance drop for baseline methods when they are allowed to abstain, rather than being forced to guess an answer. As expected, performance drops for most methods. We find that GPT-4-based methods (rows 3, 5, and 7) show the largest drop in performance, which corresponds with GPT-4's propensity to abstain. Specifically, for EM-EQA, GPT-4 abstains 36% to 55% of the time (as measured by GPT-4). LLaMA-2-based methods abstain 3% to 12% of the time (as measured by LLaMA-2). Thus, we observe minimal changes in LLaMA-2-based method scores. Finally, GPT-4V abstains 12% of the time (as measured by GPT-4), corresponding with a small drop in LLM-Match scores. Similar trends are

Figure 10. **Prompt used for Force-A-Guess.** The placeholders {question} and {old_answer} are replaced by the question Q and initial answer A, respectively. The same prompt is used for LLaMA-2 and GPT-4.

```
You are an intelligent question answering
agent.  I need you to fix the answers to
these question.

If the proposed answer says the question
is unanswerable you should output the
action ''guess''.  Otherwise, output the
action ''keep''.

Question:  What machine is on top of the
stove?
Proposed Answer:  the microwave
Action:  keep

Question:  What piece of furniture is in
the middle of the bedroom?
Proposed Answer:  The question is
unanswerable from the provided images.
Action:  guess

Question:  {question}
Proposed Answer:  {old_answer}
```

Table 3. **LLM-Match scores without forcing agents to guess.** *GPT-4V results are calculated on a subset of 500 examples.

# method	EM-EQA	EM-EQA (w/o guess)	A-EQA	A-EQA (w/o guess)
Blind LLMs				
1 GPT-4	33.5	-	35.5	-
2 LLaMA-2	27.7	-	28.8	-
Socratic LLMs w/ Frame Captions				
3 GPT-4 w/ LLaVA-1.5	43.6	29.3 (-14.3)	38.1	23.7 (-14.3)
4 LLaMA-2 w/ LLaVA-1.5	36.7	36.2 (-0.6)	30.9	31.2 (+0.4)
Socratic LLMs w/ Scene-Graph Captions				
5 GPT-4 w/ ConceptGraphs	36.5	18.5 (-18.0)	34.4	12.4 (-21.9)
6 LLaMA-2 w/ ConceptGraphs	28.7	26.6 (-2.0)	23.8	18.9 (-4.8)
7 GPT-4 w/ Sparse Voxel Maps	38.9	27.3 (-11.5)	34.2	21.2 (-13.0)
8 LLaMA-2 w/ Sparse Voxel Maps	34.3	34.6 (+0.3)	29.9	29.3 (-0.6)
Multi-Frame VLMs				
9 GPT-4V*	49.5	46.7 (-2.8)	41.8	40.6 (-1.2)
Human	86.8	-	85.1	-

observe in the A-EQA setting for all methods.

I. Full Results

Table 4 and Table 5 breaks down performance of different EQA agents, as described in Section 3, on EM-EQA and A-EQA respectively by the seven question categories described in Section 2.3. Due to API limitations, we only evaluate GPT4V on a subset of 500 OpenEQA questions in EM-EQA and 184 OpenEQA questions in A-EQA. We find that EQA agents with visual information excel at localizing and recognizing objects and attributes, and make better use of this information to answer questions that require world knowledge. However, on other categories, their performance is closer to the blind LLM baseline (GPT-4), indicating substantial room for improvement on OpenEQA.

J. Discussion on Blind LLMs.

We found blind LLMs to be a surprisingly strong baseline, considering they have no access to visual information about the scene. Upon closer inspection, we found that blind LLMs are good at "guessing" answers to EQA questions. For instance, consider the question: *'Q: What is the color of the staircase railing?'* GPT-4 answers *'brown'*, and because many houses have a *brown staircase railing*, this guess is often correct. This indicates a certain degree of regularity in the world such that answers to many questions are similar across different environments. Nevertheless, we establish a strong lower bound of performance achievable without perception, and can infer that additional gains are due perception and semantic grounding.

K. Does explicit coordinate information help?

The primary motivation for object-centric scene-graph representations is to have fine-grained perceptual understanding of objects and their locations. Thus, we intuitively expect that agents equipped with explicit object locations will fare better in questions that require spatial understanding. Surprisingly, we find in Tab. 2 that such agents fare no better than Socratic LLMs that simply use frame-level captions. We run an ablation experiment (Appendix N) where we remove explicit bounding box and size information from the scene graph, and find that this does not significantly affect performance, indicating that these LLMs are not able to effectively use 3D coordinate information in text.

L. LLM-Match Human Alignment and Details

Evaluating open-vocabulary responses to questions is an open challenge in AI, and in particular for question-answering. While human evaluation remains the gold-standard, it is also expensive and time consuming. An automatic evaluation metric is preferable for benchmarking, fast iteration, and model selection. We thus use an automatic LLM-Based evaluation metric in this work as described in Section 2.4. We performed analysis experiments to test the quality of this metric along two axis: (1) How closely aligned is the LLM-Match metric with human evaluators? (2) How sensitive is the LLM-Match metric towards specific choice of prompts and the LLM?
Human Alignment. To answer the first question, we designed an experiment to measure the agreement between LLM-Match metric and human evaluators. For this analysis, we uniformly sampled a subset of 300 questions from OpenEQA. To ensure coverage of the answer distributions (i.e. poor, fair, and good answers), we sampled 100 responses from a blind LLM (LLaMA-2), multi-frame VLM (GPT-4V), and human baseline answers. In a double blind study, we then asked 4 human evaluators to score the 300 responses using an evaluation prompt similar to the one used by LLM-Match. The evaluators were provided no information about the source of the response (except an MD5 hash of the question ID, response source, and annotator ID). We found a **Spearman's** $\rho = 0.909$ **between human and LLM evaluation** (bootstrap CI=(0.883,0.928), N=9999), indicating excellent agreement with human judgement. Table 7 shows the Spearman's ρ (a measure of correlation) between (1) each annotator and other humans and (2) each annotator and GPT-4 scoring. Human evaluators correlated with other humans in $\rho \in [0.91, 0.93]$, and with LLMs in $\rho \in [0.90, 0.94]$.
Choice of LLM. Table 6 shows rho between human evaluators and different LLMs, on the subset of 100 questions from GPT4V. GPT-4 scoring shows good agreement with human scoring ($\rho = 0.88$), while GPT3.5 (ρ=0.66) and LLaMA 2 (ρ=0.68) show lower correlation. We believe that future LLMs will show higher agreement with human annotators, and in the meantime we recommend only using GPT-4 for scoring.

Table 4. **Category-level Performance on EM-EQA** Rows represent the different agents as described in Section 3 and columns represent the different category of questions in the dataset, as described in Section 2.3. *GPT-4V scores are calculated on a subset of 500 OpenEQA question due to API limitations. Bold numbers indicate max in section.

	EQA Category							
# method	object recognition	object localization	attribute recognition	spatial understanding	object state recognition	functional reasoning	world knowledge	LLM-Match (C)
Blind LLMs								
1 GPT-4	**15.4**	**20.3**	**31.5**	**31.4**	51.0	**52.2**	**34.2**	**33.5±1.0**
2 LLaMA-2	10.7	15.3	22.3	25.0	**51.7**	44.1	29.7	28.3±1.0
Average	13.0	17.8	26.9	28.2	51.3	48.2	31.9	
Socratic LLMs w/ Frame Captions								
3 GPT-4 w/ LLaVA-1.5	**36.5**	**31.9**	**45.8**	**36.1**	**56.0**	**54.8**	**44.8**	**43.6±1.1**
4 LLaMA-2 w/ LLaVA-1.5	30.5	18.8	39.4	31.4	50.1	47.4	41.7	36.8±1.1
Average	33.5	25.4	42.6	33.8	53.0	51.1	43.3	
Socratic LLMs w/ Scene-Graph Captions								
5 GPT-4 w/ ConceptGraphs	26.4	17.0	40.6	29.1	**55.5**	**48.4**	39.9	36.5±1.0
6 LLaMA-2 w/ ConceptGraphs	17.1	13.9	24.4	27.2	43.5	38.1	39.0	28.7±1.0
7 GPT-4 w/ Sparse Voxel Maps	**30.0**	**20.0**	**49.6**	**31.7**	**55.5**	45.4	**40.8**	**38.9±1.0**
8 LLaMA-2 w/ Sparse Voxel Maps	23.4	11.7	38.9	30.8	52.8	45.4	39.1	34.3±1.1
Average	24.2	15.6	38.4	29.7	51.8	44.3	39.7	
Multi-Frame VLMs								
9 GPT-4V*	43.4	42.0	57.2	33.6	63.2	57.4	50.7	49.6±2.0
Average All Agents	29.6	22.2	42.3	31.4	53.8	48.1	42.3	
Human	87.9	77.3	87.9	86.7	98.7	81.8	87.2	86.8±0.6

Table 5. **Category-level Performance on A-EQA.** Rows represent the different agents as described in Section 3 and columns represent the different category of questions in the dataset, as described in Section 2.3. *GPT-4V scores are calculated on a subset of 184 OpenEQA question due to API limitations. Bold numbers indicate max in section.

	EQA Category							
# method	object recognition	object localization	attribute recognition	spatial understanding	object state recognition	functional reasoning	world knowledge	LLM-Score (C)
Blind LLMs								
1 GPT-4	**25.3**	**28.4**	**27.3**	**37.7**	47.2	**54.2**	**29.5**	**35.5±1.7**
2 LLaMA-2	13.7	22.1	16.2	29.7	**43.3**	50.4	28.8	29.0±1.6
Average	19.5	25.2	21.8	33.7	45.3	52.3	29.2	
Socratic LLMs w/ Frame Captions								
3 GPT-4 w/ LLaVA-1.5	**25.0**	**24.0**	**34.1**	**34.4**	**56.9**	**53.5**	**40.6**	**38.1±1.7**
4 LLaMA-2 w/ LLaVA-1.5	19.7	11.7	31.2	28.3	48.1	46.1	35.8	30.9±1.7
Average	22.3	17.8	32.6	31.3	52.5	49.8	38.2	
Socratic LLMs w/ Scene-Graph Captions								
5 GPT-4 w/ ConceptGraphs	25.3	16.5	29.2	37.0	52.2	**46.8**	37.8	34.4±1.8
6 LLaMA-2 w/ ConceptGraphs	13.3	11.9	18.8	27.9	31.7	31.7	36.5	23.9±1.6
7 GPT-4 w/ Sparse Voxel Maps	**29.0**	**17.2**	**31.5**	**31.5**	**54.2**	39.8	**38.9**	**34.2±1.8**
8 LLaMA-2 w/ Sparse Voxel Maps	16.7	9.7	33.4	29.0	47.2	40.5	37.5	29.9±1.7
Average	21.1	13.8	28.2	31.3	46.3	39.7	37.7	
Multi-Frame VLMs								
9 GPT-4V*	34.0	34.3	51.5	39.5	51.9	45.6	36.6	41.8±3.2
Average All Agents	23.3	17.9	32.8	32.5	48.9	43.4	37.7	
Human	89.7	72.8	85.4	84.8	97.8	78.9	88.5	85.1±1.1

Table 6. **Varying LLM used for scoring.** On a subset of 100 questions with answers from GPT-4, GPT-4 scoring shows excellent agreement with human judgement, while using other LLMs shows lower correlation (Spearman correlation coefficient).

Scorer LLM	ChatGPT-4	ChatGPT3.5	LLaMA 2	Human
ChatGPT-4	1.00	0.66	0.68	0.88
ChatGPT3.5	-	1.00	0.66	0.61
LLaMA 2	-	-	1.00	0.63
Human	-	-	-	1.00

Table 7. **Per-annotator Spearman-ρ.** Human scoring has excellent agreement with both other humans and with LLM scoring.

Annotator	**vs. Other Humans**	**vs. LLM**
0	0.91	0.91
1	0.91	0.91
2	0.92	0.90
3	0.93	0.94

M. LLM-Match Robustness Details

Table 8. **LLM Role.** Correlation between scores when changing the 'role' of the LLM in the scoring prompt (Spearman correlation coefficient).

Role	AI	"Score Master"	Professional
AI	1.00	0.97	0.96
"Score Master"	-	1.00	0.97
Professional	-	-	1.00

Table 9. **Match criterion for a '5'.** Correlation between scores when changing the criterion in the scoring prompt (Spearman correlation coefficient).

Match Crit.	Perfect	Contains	Pro	Person
Perfect	1.00	0.96	0.95	0.96
Contains	-	1.00	0.97	0.97
Pro	-	-	1.00	0.98
Person	-	-	-	1.00

Table 10. **Temperature of scoring LLM.** Changing the temperature of GPT-4 used in scoring (Spearman correlation coefficient).

Temp	0.01	0.1	0.2	0.3
0.01	1.00	0.98	0.98	0.98
0.1	-	1.00	0.97	0.98
0.2	-	-	1.00	0.97
0.3	-	-	-	1.00

Our LLM-Match uses the specific evaluation prompt described in Fig. 6. The metric is stable under small permutations of the prompt and LLM-Match settings as illustrated in Table 8, Table 9 and Table 10, which show the correlation in LLM-Match scores using different prompting strategies, assessed on 500 GPT-4V answers.

Role: Table 8 demonstrates that changing the LLM's role from 'AI' to 'Score Master' or 'professional evaluator' does not significantly change the results, and scores between any two treatments have a tight correlation with a Spearman's ρ all above 0.95.

Match criterion: Similarly, Table 9 shows analogous results ($\rho > 0.95$) when changing the description of a '5' from 'perfect match' to 'contains correct answer', 'similar to a reasonable person', or 'reasonable professional'.

Temperature: The stochasticity in the evaluation function has negligible impact as well, as shown by varying the temperature and seed. Table 10 shows results when varying the temperature used in the GPT-4 scorer from 0.01-0.3, with results all >0.97.

N. 3D Coordinate Ablation

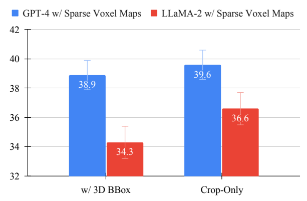


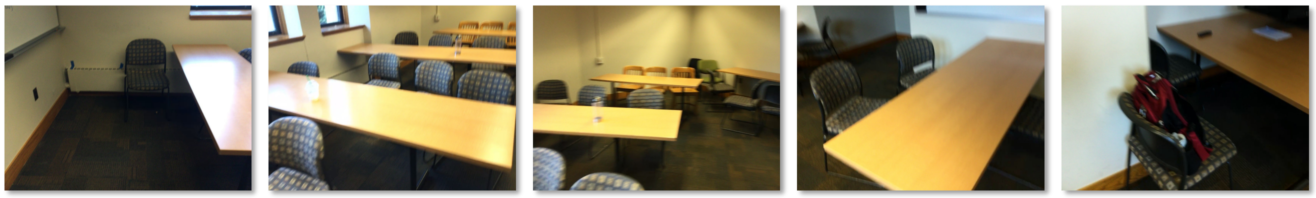
Figure 11. **Ablating 3D location for scene-graph agents.** Removing bounding box locations and extent had no significant effect for agents using either LLM.

Figure 11 compares the EM-EQA performance of the Socratic baseline that uses Sparse Voxel Map captions with and without including 3D bounding box information in the text descriptions. Results show that explicit bounding box location and size information from the scene graph does not significantly change the performance of scene-graph based agents. This suggests that neither LLM, trained with only text information, is able to effectively use the 3D location information.

O. OpenEQA Dataset Examples

Additional examples from the ScanNet and HM3D splits of OpenEQA are provided in the Figures 12, 13, and 14.

Episode History H



Question-Answer Pairs (Q, A^*)

[Object Recognition]
Q: What is the red object on the chari?
A*: a backpack

[Attribute Recognition]
Q: Among all the chairs, what is the unique color of the chair?
A*: green

[Spatial Understanding]
Q: Can 10 people sit in this room?
A*: yes

[Object State Recognition]
Q: Is the plastic water bottle open?
A*: no

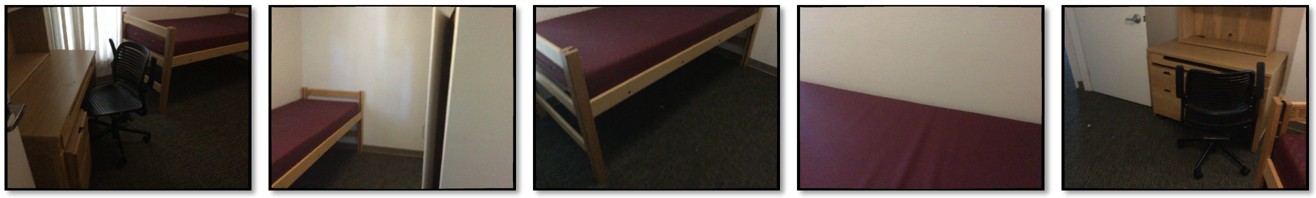
[Functional Reasoning]
Q: What can I use to write something on using my pencil?
A*: the piece of paper

[World Knowledge]
Q: Were students here lately?
A*: yes

[Object Localization]
Q: Where is my unfinished Starbucks drink?
A*: on the table near the front whiteboard
Extra Answers: ['On the second table from the front', 'In the center of the second table.', 'On the second table from the front', 'On the table near the windows']

Figure 12. **OpenEQA dataset examples from a ScanNet scene.** Note that only a subset of frames from the episode history H are displayed. Thus, some questions may require additional visual information to answers.

Episode History H



Question-Answer Pairs (Q, A^*)

[Object Recognition]
Q: What object is on top of the work desk?
A*: shelf

[Attribute Recognition]
Q: what color is the task chair?
A*: black

[Spatial Understanding]
Q: which object is closer to the window, the bed or the trash can?
A*: bed

[Object State Recognition]
Q: Is the closet door fully closed?
A*: no

[Functional Reasoning]
Q: The closet is full. Where can I store a suitcase in this room?
A*: Under the bed

[World Knowledge]
Q: What are the two main functions or purposes of this room? Why did you arrive at those conclusions?
A*: Sleeping, since there is a bed. Working, since there is a task chair and desk.

[Object Localization]
Q: Where is the two-tier shelf?
A*: on top of the desk
Extra Answers: ['On top of the desk.', 'On the left when you first walk in to the room.', 'above the desk infront of the chair', 'on top of the desk']

Figure 13. **OpenEQA dataset examples from a ScanNet scene.** Note that only a subset of frames from the episode history H are displayed. Thus, some questions may require additional visual information to answers.

Question-Answer Pairs (Q, A^*)

[Object Recognition]
Q: what is on the chair?
A^*: a soft pillow

[Attribute Recognition]
Q: is the outside door open or closed?
A^*: open

[Spatial Understanding]
Q: is the table in the living room clean?
A^*: yes

[World Knowledge]
Q: what is special about the wall in the living room?
A^*: it seems to be made of stone

[Object Localization]
Q: where is the standing lamp?
A^*: next to the bed in the bedroom
Extra Answers: ['in the bedroom',
'to the left of the bed',
'the bedroom',
'The room with the bed and the bathroom']

Figure 14. **OpenEQA dataset examples from an HM3D scene.** Note that only a subset of frames from the episode history H are displayed. Thus, some questions may require additional visual information to answers.