Is the House Ready For Sleeptime?
Generating and Evaluating Situational Queries for Embodied Question Answering

Vishnu Sashank Dorbala[1,2], Prasoon Goyal[2], Robinson Piramuthu[2], Michael Johnston[2],
Reza Ghanadan[1] and Dinesh Manocha[1]

[1]University of Maryland, College Park
[2]Amazon AGI

Supplementary materials can be found at https://gamma.umd.edu/seqa/

Abstract— We present and tackle the problem of Embodied Question Answering (EQA) with *Situational Queries* (S-EQA) in a household environment. Unlike prior EQA work tackling simple queries that directly reference target objects and properties ("What is the color of the car?"), situational queries (such as "Is the house ready for sleeptime?") are challenging as they require the agent to correctly identify multiple object-states (Doors: Closed, Lights: Off, etc.) and reach a *consensus* on their states for an answer. Towards this objective, we first introduce a novel Prompt-Generate-Evaluate (PGE) scheme that wraps around an LLM's output to generate *unique* situational queries and corresponding consensus object information. PGE is used to generate $2K$ datapoints in the VirtualHome simulator, which is then annotated for ground truth answers via a large scale user-study conducted on M-Turk. With a high rate of answerability (97.26%) on this study, we establish that LLMs are good at generating situational data. However, in evaluating the data using an LLM, we observe a low correlation of 46.2% with the ground truth human annotations; indicating that while LLMs are good at generating situational data, they struggle to answer them according to consensus. When asked for reasoning, we observe the LLM often goes against commonsense in justifying its answer. Finally, we utilize PGE to generate situational data in a real-world environment, exposing LLM hallucination in generating reliable object-states when a structured scene graph is unavailable. To the best of our knowledge, this is the first work to introduce EQA in the context of *situational queries* and also the first to present a *generative* approach for query creation. We aim to foster research on improving the real-world usability of embodied agents through this work.

I. INTRODUCTION

Embodied Question Answering (EQA) introduced by Das. et. al., [2] involves an agent exploring its surroundings to find an answer to a query. Prior work on this task [3]–[6] has primarily dealt with what we term "*simple queries*", that are directly answerable from states or properties of objects from the simulator. For instance, questions like "What is the color of the sofa?" or "Is there an object nearby to help me cut vegetables?" are directly related to an object present in the environment, requiring a mix of vision-language grounding with elementary commonsense to solve the task. While important, this type of question-answering capability does not add much utility to a real-world household robot, as the user asking questions is likely to already know answers to simple questions about their surroundings.

In contrast, answering queries such as "*Is the kitchen ready for meal preparation?*" requires a sense of situational awareness in gauging the status of multiple objects in the environment. While not uniformly defined, a *consensus* does exist on standard conditions that can be used to answer
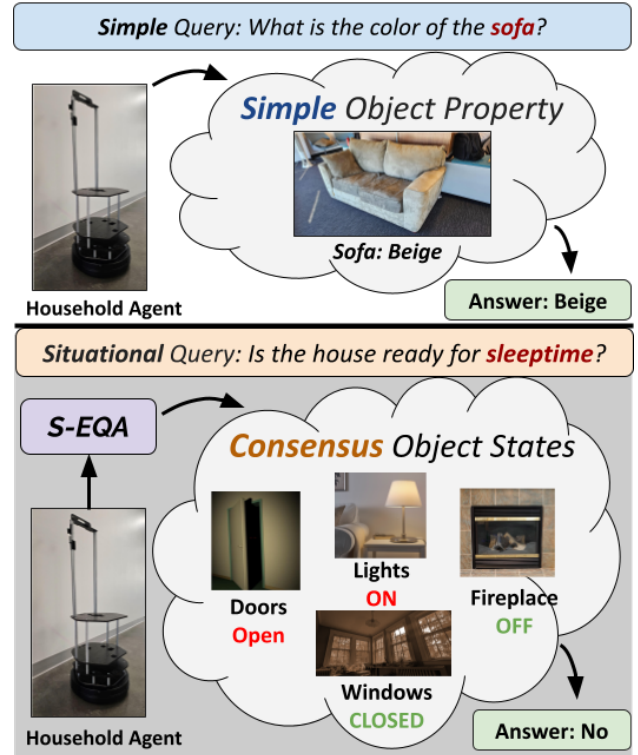

Fig. 1: Simple vs **Situational EQA**: We introduce *Situational Embodied Question Answering* (S-EQA), where queries require inspecting multiple objects and deriving *consensus* knowledge of their states. Using an LLM, we generate S-EQA on VirtualHome [1], with *consensus* object states and relationships. A large-scale MTurk study annotates and verifies data *authenticity*. Evaluating LLMs on S-EQA reveals their strength in generating queries and consensus but misalignment in answering them. LLM explanations further suggests poor commonsense reasoning on our task.

such questions. This refers to a set of generally accepted object states and conditions (such as a <u>Knife</u> being <u>Present</u>) for successfully evaluating the query. A household robot deployed in the real world is likely to encounter such queries that require it to utilize consensus knowledge pertaining which objects and associated states are commonly associated with the situation. We term these as *situational queries*, and in this work, tackle the twofold challenge of 1) conjuring *authentic* situational queries and corresponding consensus object data and 2) highlighting the role of consensus object states in simplifying the process of responding to situational queries.

Dataset	Navigation + QA	Multi-Target	Interaction	Abstract Queries	Object State Consensus	Query Creation
EQA [2]	✓	-	-	-	-	*Rule-Based*
MT-EQA [5]	✓	✓	-	-	-	*Rule-Based*
IQA [6]	✓	✓	✓	-	-	*Rule-Based*
K-EQA [7]	✓	✓	-	✓	-	*Rule-Based*
S-EQA (**Ours**)	✓	✓	-	✓	✓	***Generative***

TABLE I: Comparison of S-EQA with existing EQA datasets. We introduce **situational queries** that require *consensus* object states for answers. While K-EQA asks abstract questions (e.g., "Is there an object used to cut food nearby?" → *Yes* if a *knife* is present), S-EQA extends this by requiring *consensus* on multiple object states (e.g., "Is the kitchen ready for meal prep?" → *Knives: Present, Oven: Preheated, etc.*). As real-world consensus data is scarce, we approximate it using an LLM and validate authenticity via a user study. We are the first to introduce a **generative** approach for EQA dataset creation, and discuss challenges associated with such schemes.

A query such as "*Is the house ready for sleeptime?*" is a *situational query*, since there exist generally accepted object states, such as dimmed lights, closed windows, locked doors, etc., which most people associate with a house being ready for nighttime. However, a query like "*Can I bring guests over?*", while situational, is too *personal*, as the answer relies on various user preferences on household norms and event-specific constraints. In our work, we aim to generate and human-validate queries where a general *consensus* exists on multiple object states.

Developing EQA datasets in the past has been limited to constructing simple queries using entities from a simulator [5], [6], with some work extending it to abstract queries built on scene-graphs [7]. Posing *situational queries* using these existing techniques is however nontrivial, as they stem from human intuition and contextual awareness of the environment. Setting up real-world experiments to collect this data is also cumbersome; if not having an intrusive agent capturing conversations at home, an offline approach would be cognitively heavy, requiring humans to imagine situations for queries. In our work, we develop a scheme that utilizes a Large Language Model (LLM) to generate such queries, providing in-context cures to cover unique concepts stemming from various situations. Table I compares our dataset with existing work.

Main Results: We tackle the issues presented above, utilizing LLMs as a prior for generating situational queries and associated consensus object states and relationships. We human-validate the generated data, and present a benchmark for EQA on situational queries. The main contributions of our work are as follows:-

- **Problem**: We tackle the novel problem of *posing* and evaluating *Situational Queries* for Embodied Question Answering (**S-EQA**). Unlike simple queries that directly reference particular objects, posing situational queries itself is challenging, requiring a *consensus* on multiple objects and their states, and cannot be formed easily using prior rule-based methods (Refer Table I).
- **Method**: We propose a novel Prompt-Generate-Evaluate (**PGE**) scheme that integrates an LLM with a feedback scheme that allows for the generation of several **unique** situational queries, along with corresponding *consensus* object-state and object-relationship information that needs to be satisfied to successfully answer the question.
- **Human-Validation**: Using PGE in the VirtualHome simulator, we produce a S-EQA dataset of 2000 situ-

ational queries and consensus data. We annotate these via a large scale MTurk [8] user study, which gives us a high authenticity percentage of 97.26%, indicating that the data generated corresponds to consensus.
- **LLM Eval**: We prompt the LLM to provide binary answers to S-EQA datapoints generated from Virtual-Home. Despite being able to generate highly authentic situational data, we observe a subpar correlation of 46.2% with the human annotations when it comes to answering. We infer that LLMs can **generate good situational queries** and consensus object-states, but **answer them poorly**.
- **LLM Reasoning**: When asked for an explanation, we observe the **poor commonsense reasoning of LLMs** in justifying their answer. This further suggests that LLMs exhibit poor situational awareness.
- **Real World Challenges**: Finally, we discuss the transfer of PGE to a real world real world lab environment, and discuss various challenges with LLM hallucination that obstruct the generation of quality consensus data.

II. RELATED WORK

Embodied Question Answering (EQA): EQA is a popular task [6], [9], [10] that involves a robot attempting to answer a query by exploring and gathering information from its surroundings. It was introduced by Das et. al [2] as a single-target question-answering dataset. Yu et. al. [5] further extended this task to a multi-object setting, MT-EQA, where correspondences between multiple target objects present in the question itself need to be analyzed before arriving at an answer. Both these works use a template-based approach to generating questions, and this was deemed sufficient since the queries are objective in nature mentioning the objects of interest in their query itself. This however is a limitation, since a majority of queries tend to be subjective or situational in nature [11], especially in a household environment.

Sintean et. al in K-EQA [7] attempt to tackle this problem, by using scene-graph data and formal grammar to postulate queries. However, these queries are still limited to simple object grounding within a scene, and do not require any form of situational awareness about the *readiness* of an environment. For instance, S-EQA considers queries like "Is the living room ready for a movie night?", that requires consensus on multiple objects and their states (TV: On, Popcorn: Cooked, On Table, etc.) to measure this readiness. A K-EQA query for the same example could be constructed as "Is there a device
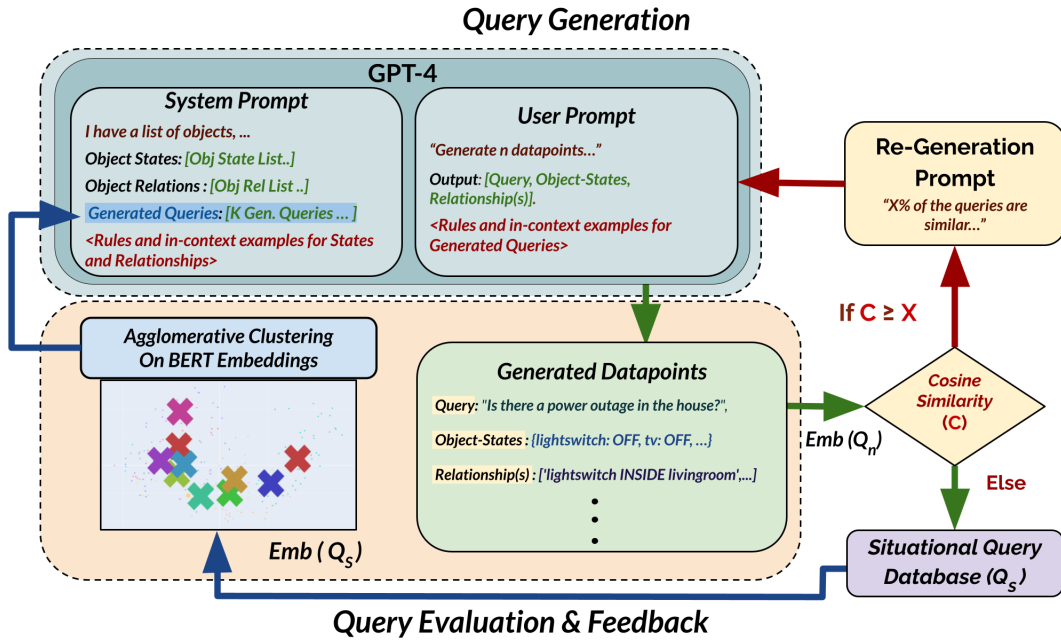
Query Generation

Fig. 2: Prompt-Generate-Evaluate (PGE) Scheme: We utilize an LLM (GPT-4) to generate *situational datapoints* comprised of queries, corresponding consensus object states and relationships. The data generation occurs over m iterations, refining LLM prompts after evaluating every batch of n datapoints. For each batch, BERT embeddings of the n queries are computed and denoted as $Emb(Q_n)$. A cosine similarity $\mathbb{C}$ is then computed between $Emb(Q_n)$ and $Emb(\mathbb{Q}_S)$, the embeddings of existing queries in the *Situational Query Database* $\mathbb{Q}_S$. C is a threshold used to determine if the generated queries are sufficiently different from those in the database and if not, we continue *conversation* with the LLM (indicated with the red arrows). Concurrently, we cluster $Emb(\mathbb{Q}_S)$ into **k** categories, selecting the query nearest to the centroid in each cluster for feedback, labeled as "Generated Queries" in the System Prompt (denoted by blue arrows). The ideal datapoint generation pathway is highlighted by the green arrows.

for watching movies in the living room?", which requires only scene knowledge about the availability of a TV, but does not require *situational* knowledge about the various object-states involved in a movie night.

In our work, we utilize an LLM to generate situational queries and approximate human consensus on object states. Our queries are an abstraction above the scene-knowledge based K-EQA queries, requiring *consensus knowledge* about multiple object *states* for validation.

LLMs for Query Generation: A recent trend has seen the emergence of Large Language Models (LLMs) such as GPT-3.5 [12], GPT-4 [13], LLaMA [14], Palm [15] and Palm-e [16]. These models are well known for their commonsense reasoning capabilities [17], especially in the context of question-answering. In embodied AI, these capabilities have recently been leveraged to produce superior planning and decision-making strategies during navigation [18]–[21]. In our case, we wish to generate situational queries for embodied question answering and ask if the common sense reasoning capabilities of LLMs can both generate such queries as well as the the consensus object-states needed for *sufficiently* answering them.

III. SITUATIONAL EQA: GENERATION & VALIDATION

Having motivated the problem of *Situational EQA*, we now present a detailed description of our approach. We present three phases — generating situational data using an LLM (GPT-4), annotating the data via a large-scale user survey to produce a S-EQA dataset, and finally evaluating it on the VirtualHome simulator.

A. Situational Query Definition

Situational queries in our work are defined as questions requiring the assessment of a collection of consensus objects and states to reach an existential 'Yes/No' answer. For example, a query such as *"Is the livingroom ready for movie night?"* has a Yes/No answer based on whether consensus objects like the TV, lights, etc. are in their appropriate state. Additionally, we want to ensure that the queries are related to a situation and are not simple in nature, i.e., the target objects and states are not referenced in the query itself. For example, "*Is the sofa blue?*" [2] or "*How many objects can be used to hold rice?*" [7] are simple queries, as they do not deal with a *situation*, but rather pose a query referencing objects in the environment. In contrast, "*Is the dining area set up for dinner?*" is situational, as it doesn't explicitly mention any objects, but requires the identification of *relevant* objects (eg. *table, napkins*), and a *consensus* on what their states should be (*open, present*) for the answer. Considering the LLM as a function $\mathscr{L}$ that generates queries q, and $\mathbb{Q}$ to be a set of all possible queries generated by $\mathscr{L}$, we want *situational* queries $\mathbb{Q}_S \subset \mathbb{Q}$, which satisfies the following conditions -

- **Abstraction**: The query must not directly reference any object, i.e., it must omit words from a set $\mathbb{W}$ containing words such as "object", "table", "chair", etc.
- **Contextual**: The query **requires** an assessment of relevant objects and their states collectively satisfy a scenario (e.g., "Is the dining area set up for dinner?"), requiring identification of appropriate objects and consensus on their expected states (e.g., table present, napkins, lighting, etc.).
- **Binary**: The answer to the query must be binary i.e.,

Yes/No, and not subjective in nature.

Formally, we can define $\mathbb{Q}_S$ as:

$$\mathbb{Q}_S = \{q \in \mathbb{Q} \mid \text{Abstraction}(q, \mathbb{W})$$
$$\wedge \text{Contextual}(q) \wedge \text{Binary}(q)\} \qquad (1)$$

This formulation ensures that $\mathbb{Q}_s$ consists only of queries that are abstract, questioning in form, and expect a binary Yes/No response. We present these conditions as in-context rules to the LLM to curate it's responses.

B. Generating Situational Queries: S-EQA

We prompt the LLM to generate datapoints of the format - *[Situational Query, Object-States, Object-Relationships]*. For example,

> **Situational Query**:
> *"Is the bathroom ready for a shower?"*
> **Object-States** : [lightswitch: ['On'], towels: ['Present'], Soap: ['Present'] ...]
> **Object-Relationships** : [lightswitch inside bathroom, towels inside bathroom, ...]

The object states and relationships (such as 'towels inside bathroom being present') being generated represent scene-graph requirements that the LLM believes need to be checked for the query to satisfy equation 1.

VirtualHome Setup: We use the VirtualHome [1] simulator for defining ground truth objects, states and properties in our experiments. It contains 7 different household environments containing over 390 objects. Unlike other embodied simulators, VirtualHome is unique in that it contains several household objects including interactable ones, with easily *modifiable* locations and states. The objects are also visually different upon modification, and this allows us to perform VQA later on to evaluate our dataset.

All objects in VirtualHome have a set of state properties, which are either physical *Open/Closed* (eg. for a Door), and/or *On/Off* (such as lights) or *Present/None* (such as Apple). These objects also have a relationship with either the room or another object. For example, *Apple INSIDE Fridge* and *Fridge INSIDE livingroom* are both valid relationships. The simulator has 4 different rooms into which all objects are placed - [livingroom, bedroom, kitchen, bathroom]. We use these room markings to group and analyze the generated queries later on.

Prompt-Generate-Evaluate (PGE) Scheme: Figure 2 presents an overview of our *Prompt-Generate-Evaluate* scheme for generating data. We analyze the LLM's generated datapoints to provide feedback for ensuring the data fits into our definition of situational queries, while also maintaining diversity.

The *System Prompt* for our LLM contains a list of all object states and relationships, which in this case is the scene-graph provided by VirtualHome. We also provide additional rules and in-context examples to improve the generation of prompts. For instance, explicitly mentioning "Do not conjure any new objects or states" ensures that the generated data does not contain objects not present or states that are not achievable in VirtualHome. Additionally, we also provide a set of k *"Generated Queries"* as feedback to the system prompt. These queries are representative of what the LLM has already generated, and we encourage the LLM to generate queries different from these.

Our *User Prompt* asks to generate *n* datapoints in the format specified earlier, and while providing a few manually annotated sample datapoints. Additionally, we provide some in-context rules to ensure that the generated queries are not too simplistic, not too ambiguous, and are also unique. For instance, we explicitly state '*The situational query must not reference any object or even contain the word 'object'.*' and '*The query must have a Yes/No answer.*'. We also ask it to generate queries very different from what is already present in the *k Generated Queries* in the System Prompt and come up with 'unique and creative household situations' with the object data provided.

Fig. 3: Influence of PGE: Notice the queries without PGE directly reference objects (microwave and computer) and are not situational. The queries are also just rephrased. PGE incorporates feedback measures to generate unique and diverse queries. Some queries may not hold general consensus on object states (such as *Was there a break-in at the house?*), and these are filtered via human validation.

To ensure that our *Situational Query Database* $\mathbb{Q}_S$ contains diverse data points adhering to specifications presented in equation 1, we take the following two **feedback measures**:

- **Generated Query Update:** After generating *n* data points, we embed the situational queries using RoBERTa [22] to get sentence embeddings $Emb(\mathbb{Q}_S)$. We perform agglomerative clustering [23] on $Emb(\mathbb{Q}_S)$ to accumulate them into *k* groups of queries based on sentence similarity. We then pick the query closest to the centroid of each of these clusters to get *k* queries that best represent the diversity in the generated data. Finally, we update the system prompt with these queries.
- **Continued Conversation:** Simultaneously, we compute the *cosine similarity* $\mathbb{C}$ of the *n* generated query embeddings $Emb(Q_n)$ with the queries already present in the *Situational Query Database* $\mathbb{Q}_S$. If $\mathbb{C}$ is greater than a particular threshold (*X*), we continue the *conversation* with the LLM, by presenting it with a re-generation prompt. We mention the percentage of similarity, and ask the LLM to re-generate a new batch of datapoints.

These *feedback measures* ensure $\mathbb{Q}_S$ contains unique situational queries that are not simple rewordings, while also encouraging diversity in the generated situations. Contrasting examples with and without PGE are illustrated in Fig. 3.

The total number of datapoints generated $\mathbb{Q}_S$ is always less than or equal to the product of the looping parameters *n* and *m*, i.e., $\mathbb{Q}_S \leq n * m$. While our approach allows for the generation of potentially infinite datapoints, we observe a saturation point where the LLM starts to generate similar-looking queries (*"Is the bathroom clean and dry?"* vs *"Has the bathroom been cleaned today?"*). This is indeed a drawback of such a generative approach, and can be attributed

to the abstract nature of the queries being generated, which do not pertain a particular object, but related to a household situation. Therefore, to effectively assess the utility of the data, we cut off generation at 2000 data points. We validate this data via a large-scale MTurk user study, and present this as **S-EQA**, a dataset containing diverse and challenging situational queries along with consensus object data to help answer them.

C. Validating S-EQA: t-SNE plots and M-Turk Annotation

Our objective with validating S-EQA is not just to annotate the queries, but to also measure the *authenticity* of the generated data. We first visualize the queries in $\mathbb{Q}_S$ using t-SNE plots to analyze their diversity. We then conduct a user study via Mechanical Turk [8] to annotate the generated data.

t-SNE Plots: We embed the generated queries with BERT and visualize them using t-SNE. They are then are categorized by Room (pick from [kitchen, living room, bedroom, bathroom, multi-room, no room]), Situational/Simple (Yes/No) and Spatio-Temporal (time-sensitivity). We observe that GPT-4 tends to generate queries specific to objects in a single room over multi-room or no-room ones like "*Is the house ready for sleeptime?*". Further, we note that most queries (82.2%) are classified as situational and not simple. Around 78% are categorized as spatial queries (like "*Is the kitchen ready for cooking?*") that refer to the current environment state, as opposed queries like "*Was someone in the kitchen today?*" which require deeper situational understanding involving time. Overall, we observe that most queries adhere to the definition provided in Equation 1. More details including the plots are in the technical report[1].

MTurk Validation: We conduct a large-scale Mechanical Turk [8] study to annotate the generated data and evaluate the authenticity of real-world queries. The annotators are chosen from the US-East region, with at least a 95% approval rate on previous work.

Annotators are shown a generated query, along with consensus object-states and relationships. Given options of *Yes*, *No*, and *Cannot Answer*, a worker's capacity to give a definitive response, i.e., choosing *Yes* or *No*, serves as an indicator of whether the generated data is in alignment with general user consensus. For instance, the query "Is the house ready for sleeptime?", if agreed upon by workers when shown images of dimly-lit rooms, closed doors, and windows, validates the data point.

We obtain a low 'Cannot Answer' percentage of 2.74% among the $5 \times 2K = 10K$ participants. As such, a high 97.26% percentage of generated queries being answerable strongly indicates that the LLM is capable of generating object-states and queries that are a good representation of human consensus.

Subsequently, we conduct a second **comparison study** with the same workers, where they are presented with an image of a room accompanied by a situational S-EQA query (For eg. *Is someone working in the bedroom?* with top-views of the bedroom) and, separately, images of objects paired with consensus queries formulated based on their state (For eg., *Is the Computer On?* with an image of a computer being
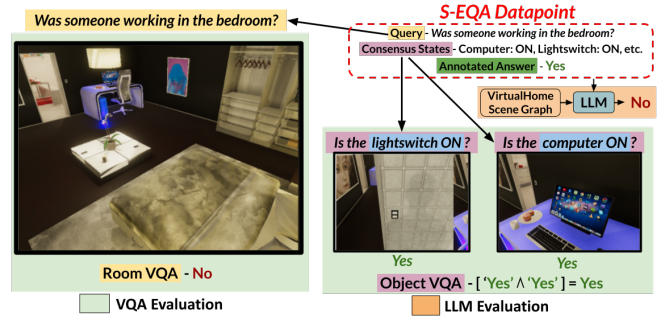
Fig. 4: S-EQA Simulator Eval: We evaluate S-EQA generated from VirtualHome using the LLM with Scene Graphs (Orange) and VQA (Green). In this example, the situational query asks if someone was working in the bedroom, to which the annotation was positive (Yes). For the LLM Eval, we pass the entire VirtualHome scene graph with modified consensus states along with the query as input to the LLM. Note the negative response of the LLM here, despite it generating valid consensus states. This mismatch is present in **46.8%** of cases, reflecting poor LLM answering capability. For VQA Eval, we query various VLMs with Room and Object images (See Table II). Note the simpler Object VQA queries that are answered correctly, suggesting breaking down situational queries into consensus ones can simplify the task via indirect answering.

on). We sample additional 'Yes/No' questions the SimVQA [24] dataset to even out the number of situational and consensus queries. An improved annotator performance in answering consensus queries over situational ones would indicate S-EQA's *utility* in breaking down subjective queries into simpler consensus queries for improved question-answering. We discuss this later in the results section.

In both our MTurk experiments, we measure average reviewer agreement across 5 users on each question to obtain ground truth labels on the data. For instance, if a question has 3 *Yes's*, and 2 *No's*, we label the ground truth as 'Yes'. On the other hand, if even one of the annotations is 'Cannot Answer', we label the human-annotated output by the same.

IV. SITUATIONAL EQA: EVALUATION

Figure 4 presents an overview of our evaluation of the simulator-based dataset, and Figure 5 presents our real-world setup for evaluation. We evaluate queries both via VQA as well as by using an LLM. Further, LLM answers are qualitatively studied via reasoning to infer situational awareness.

A. VQA Evaluation:

We perform VQA on images taken from VirtualHome with situational and consensus queries. The VirtualHome simulator is unique in enabling this, as it allows us to modify consensus object states pertaining to subjective queries such that they also visually change upon modifying states.
Situational → Consensus Queries: Consensus object states associated with the generated situational queries are used to formulate binary objective questions for evaluation. These are of the form -

$$Is\ the\ <OBJECT>\ <STATE>?$$

For instance, for a query q :'*Was someone working in the bedroom?*', the consensus queries would be '*Is the Computer On?*' and '*Is the Lightswitch On?*'.

We present VQA models with both situational and consensus queries, and compare the results.

Room VQA (Situational Query): Here, we present the VQA model with multiple top-view images of a relevant room as input, and ask it to answer the associated situational query with Yes/No/Cannot Answer. The assumption here is that the agent might have access to overhead security cameras which it could use to answer the queries. Most of our queries are associated with one particular room, which is often part of the query itself. During our experimentation, we observe a bias in model performance when the name of the room itself is present in the query. For instance, if the query contained 'living room', and the image was that of the living room, the answer would always be yes, irrespective of the actual context of the query. To mitigate this, we replace any mention of a room in the query with the generic term - "*this place*".

Object VQA (Consensus Query): We provide the VQA model with these consensus queries along with images of the corresponding objects in their designated states.

For both **Room VQA** and **Object VQA**, we measure the accuracy and F1-scores of prediction across all the images, using the human-verified answers as ground truth. We use a threshold of 0.5 on the accuracy and F1 scores to determine if the query has successfully been answered. Additionally, we also compute the **Joint Accuracy** (J_A), and **Joint F1** (J_{F1}) scores which is a success case when either of the Room or Object VQA are successful. Formally, let R be the accuracy or F1 result for Room VQA and O be the same result for Object VQA, where $R, O \in \{0, 1\}$. $R = 1$ indicates a match with the ground truth for Room VQA, and similarly, $O = 1$ indicates a match for Object VQA.

J is then defined as:

$$J(R,O) = \begin{cases} 1 & \text{if } R = 1 \text{ or } O = 1 \\ 0 & \text{otherwise} \end{cases}$$

J_a and J_{F1} give us an upper bound on the performance of VQA models. For J_{F1}, we set a threshold of 0.5 above which the score would be deemed satisfactory.

B. LLM Evaluation:

Motivated by K-EQA [7], we utilize scene graphs provided by VirtualHome to develop an LLM-based action planner.

For S-EQA, we have situational queries generated along with consensus object states and relationships. VirtualHome provides us with a *fully observable scene graph* for this, i.e., the states and relationships of all objects are known.

Our aim is to measure how well the LLM is able to approximate the human consensus behind answering situational queries. For this, we first ask the LLM to provide a Yes/No answer to each of the the datapoints generated. We then measure the correlation between the LLM and human annotated answer. Further, we also ask the LLM to reason about its answers for a subjective analysis.

C. Real World Experiments:

Unlike VirtualHome, where a fully observable scene graph is readily available, the real-world environment does not have object states in a predefined format. Instead, we gather images from various locations in our environment which is used as context for GPT-4V to generate situational datapoints. A representative subset of scenes and datapoints being generated are illustrated in Fig 5. We aimed to create diverse situations including group meetings, robot operations,

cafeteria related events across various locations in our lab environment.

The VLM predicts Yes/No responses for each datapoint. These are then compared against human-annotated ground truth responses to assess alignment. We then analyze failure cases where the LLM misinterprets missing information or hallucinates object states, showcasing the difficulty in accurately comprehending situational queries in the real world.

V. RESULTS AND INFERENCES

GPT-4 [13] is our LLM of choice to both generate and evaluate situational queries in our experiments. Our ablations with GPT-3.5 showed poor quality in terms of generation, with many outputs not fitting the specified data format.

A. Simulator Setup

We first program the object states and relationships from the generated data points into the VirtualHome environment. This also includes manually filtering out a small subset of bad states which are not feasible. For instance, *'computer INSIDE fridge'* is not a valid relationship for us to set, and the simulator prevents this.

For VQA evaluation, we require images of the room and corresponding consensus objects set to their appropriate states. We gather room images by considering the 4 corners of each room, with a camera pointing downwards towards the center. Choosing images in this way allows us to achieve maximum coverage of objects present.

Then for object images, we use the first-person view of a human placed in the environment. We consider the human to be our robot agent here. They are instructed to walk to each of the objects present in a S-EQA datapoint, where we grab an image when they are present in front of them. This approach for capturing images proved to be simpler to implement over simulator camera placement and gives us a visual perspective of an agent grabbing an object.

B. VQA Results

We use BLIP-2 [25], BLIP [26] and ViLT [27] to evaluate S-EQA, presented in Table II below.

	VQA Model	Accuracy % ↑	F1-Score % ↑
Room	BLIP-2 [25]	**42.31**	**52.51**
	BLIP [26]	41.82	49.63
	ViLT [27]	39.80	42.31
Object	BLIP-2 [25]	**58.26**	**68.75**
	BLIP [26]	51.33	63.75
	ViLT [27]	47.27	58.31
Joint	BLIP-2 [25]	**89.6**	82.5
	BLIP [26]	83.29	**83.75**
	ViLT [27]	72.91	80.35

TABLE II: VQA performance on Room and Object VQA.

Observe the higher Object-VQA accuracies (15.31% on average) and F1 scores over Room-VQA across each approach. This suggests that answering situational queries by looking at images at a room level is significantly harder than answering queries about the states of target objects. Our **comparison study** with MTurk (Refer III-C) further corroborates this result, with 74.3% of users finding consensus queries to be simpler to answer over situational ones. This suggests that our approach of breaking down

Real World Environment
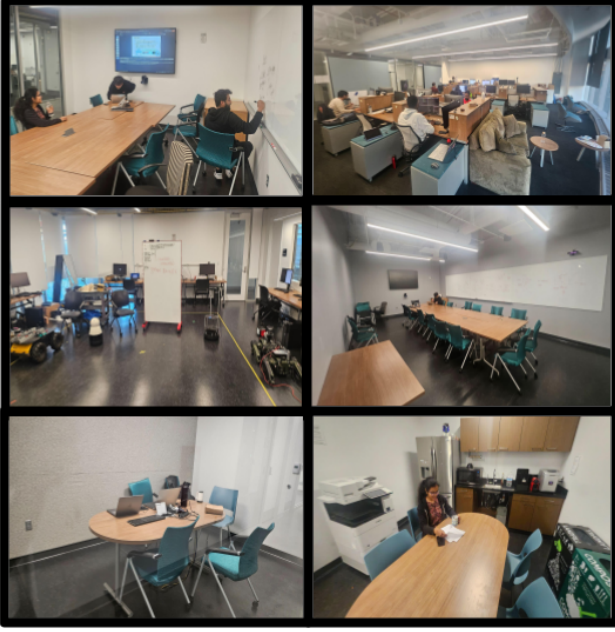
Situational Queries & Consensus Object States

```
1. Can I use the huddle room for a private call?
Object-States [TABLE: OCCUPIED, CHAIRS: AVAILABLE,
LAPTOPS: PRESENT, NOISE LEVEL: LOW, DOOR: CLOSED,
LIGHTING: SUFFICIENT]

2. Is the cafeteria a good place for a team lunch right
now?
Object-States: [TABLES: PARTIALLY OCCUPIED, CHAIRS:
AVAILABLE, FOOD STATION: STOCKED, NOISE LEVEL: MODERATE,
LIGHTING: BRIGHT]

3. Is the conference room set up for a virtual meeting?
Object-States: [TABLE: OCCUPIED, CHAIRS: ARRANGED, SCREEN:
ON, CAMERA: AVAILABLE, MICROPHONE: PRESENT, LIGHTING:
BRIGHT]

4. Is the lab organized for a safe robotics demonstration?
Object-States: [ROBOTS: READY, FLOOR: CLEAR, TOOLS:
ORGANIZED, WORKBENCH: ACCESSIBLE, COMPUTERS: ON, SAFETY
EQUIPMENT: AVAILABLE]

5. Can I conduct a quick brainstorming session in the
huddle room?
Object-States: [TABLE: OCCUPIED, CHAIRS: ARRANGED,
WHITEBOARD: AVAILABLE, MARKERS: PRESENT, SCREEN: OFF,
NOISE LEVEL: LOW]
```

Fig. 5: Real World Setup: Since we do not have a scene graph readily available in the real world, we pass the images as context to GPT-4o. On the left are images showing various situations in different areas in our environment. On the right are examples of the generated situational queries and consensus object state data. Note the object-states marked in red, indicating **LLM hallucination** generating consensus information that is outside the scope of the data provided. VirtualHome datapoints did not have this issue since the scene graph was explicitly given. Directly using a VLM to generate this data seems to produce such hallucinations.

situational queries into simpler consensus ones could be used to indirectly answer difficult situational queries.

The high values on joint scores J indicate an upper bound on performance. While still better than RoomVQA, ObjectVQA models have mediocre performance overall, despite the queries being straightforward in nature. This can be attributed to inference on synthetic images from VirtualHome which these models have not been trained on, opening an avenue for future work.

C. LLM Eval Results

We obtain a low correlation of 46.2% between GPT-4 generated answers and the human-annotated ones. This is despite the LLM having complete access to the entire scene graph which it used earlier to generate situational queries.

Furthermore, the low 'Cannot Answer' percentage of 2.74% on the annotated responses shows that most queries and consensus data generated did indeed make sense to most users. We can infer from these two values that while GPT-4 is good at generating situational queries and the potential consensus data needed to answer them, it is poor at answering it's own generated queries. To understand this further, we prompt for reasoning as follows:-

LLM Reasoning: To analyze the poor LLM performance on query answering, we reason about the generated answers by prompting GPT-4 to provide reasoning as follows:

> *<Situational Query and Object Data>*
> *Your answer is: <LLM Answer (Yes/No)>.*
> *Can you provide a brief reason for your answer focusing only on the object states and relationships provided?*

Upon inspecting the generated reasoning, we notice that GPT-4 often goes against the consensus while predicting the answer. For example, consider the following datapoint -

> *Query: "Is the living room prepared for a movie night?"*
> *Object-States : [tv: ['ON'], lights: ['OFF'], remotecontrol: ['ON'], ...]*
> *Object-Relationships : [tv INSIDE livingroom, lights INSIDE livingroom, remotecontrol INSIDE livingroom, ...]*
> *LLM Answer: 'NO'*

The reasoning generated for the LLM Answer is -

> *The lights are off which is suitable for a movie night, but the TV is on which is not suitable for movie watching as it may create distractions.*

GPT-4 suggests that the TV being ON may create distractions for movie watching and hence responds with No. However, a more plausible situation would be that the user would watch the movie on the TV itself, and that the living room is indeed prepared for a movie night.

On the flip side, since our object states are limited to the ones present in VirtualHome, we are inclined to believe that the quality of the output depends on the richness of the object set. In this example, for instance, additional object information such as *DVD: ['Present']* and *Popcorn ON Table* would improve the consensus of a movie night about to happen. Since such objects do not exist in VirtualHome, it is reasonable to assume that GPT-4's explanation is sufficient.

These results suggest that while GPT-4 is indeed capable of producing situational queries and *authentic* consensus information, its displays **poor commonsense reasoning** while answering such queries.

D. Real World Results

We experiment with situational queries in our real-world lab environment, assuming the presence of an assistive workplace robot agent. We gather a total of 10 images across various situations, corresponding to 6 different locations in our workplace, and feed these into PGE as visual context for GPT-4V. We then generate 100 datapoints using PGE and manually annotate them.

During annotation, we note several hallucinations in the object states that did not occur earlier when the scene graph was explicitly defined in VirtualHome. Some instances of this are highlighted in red in Figure 5. For example, note the noise level indication is present as an object state in many cases, but can not be discerned by looking at the image. Ghost objects such as *microphones* and *safety equipment* also appear in the generated object-states. We infer here that a *grounded* scene graph, like what VirtualHome provides is important to avoid poor data generation.

Evaluating the generated datapoints with GPT-4 presents a mismatch of 57.8%, a similar trend as that with the VirtualHome data, and solidifies our claim that LLM generated binary answers are misaligned with human consensus. More technical details are in the extended version of our work[1].

VI. Conclusion, Limitations and Future Work

We introduce and tackle the novel and challenging problem of Embodied Question Answering with Situational Queries — **S-EQA**. Unlike prior work that dealt with simple objective queries that directly reference target objects, situational queries necessitate an approximation of a *consensus* on multiple object states. Given the challenge in gathering such queries from the real world, we rely on commonsense knowledge from an LLM, GPT-4 to develop a Prompt-Generate-Evaluate (PGE) framework for posing unique situational queries and corresponding consensus object information. In the VirtualHome simulator, we use PGE to generate 2K unique queries, and annotate them via a large scale MTurk user study with $10K$ participants. The high answerability (97.26%) lets us infer that LLMs are good at generating situational queries and corresponding consensus data.

However, using GPT-4 to answer the simulator queries shows a low correlation with the human annotations (46.2%); we infer that **LLMs are good at generating situational data but poor at answering them**. When asked to explain its answers, we observe the **LLM's poor commonsense capability**, exhibiting counterintuitive reasoning. This suggests that LLMs show poor situational awareness.

Finally, we discuss challenges with transferring our generation and evaluation schemes to a real world lab environment. Since a grounded scene graph isn't directly available for use in the real-world, we provide images of various locations as context for query generation, and observe several unintended hallucinations with *ghost* object states as consensus. Future work will focus on limiting these by first gathering structured scene graphs using existing visual map building techniques.

References

[1] X. Puig, K. Ra, M. Boben, J. Li, T. Wang, S. Fidler, and A. Torralba, "Virtualhome: Simulating household activities via programs," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8494–8502.

[2] A. Das, S. Datta, G. Gkioxari, S. Lee, D. Parikh, and D. Batra, "Embodied question answering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 1–10.

[3] H. Luo, G. Lin, F. Shen, X. Huang, Y. Yao, and H. Shen, "Robust-eqa: robust learning for embodied question answering with noisy labels," *IEEE Transactions on Neural Networks and Learning Systems*, 2023.

[4] D. Gao, R. Wang, Z. Bai, and X. Chen, "Env-qa: A video question answering benchmark for comprehensive understanding of dynamic environments," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 1675–1685.

[5] L. Yu, X. Chen, G. Gkioxari, M. Bansal, T. L. Berg, and D. Batra, "Multi-target embodied question answering," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 6309–6318.

[6] D. Gordon, A. Kembhavi, M. Rastegari, J. Redmon, D. Fox, and A. Farhadi, "Iqa: Visual question answering in interactive environments," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4089–4098.

[7] S. Tan, M. Ge, D. Guo, H. Liu, and F. Sun, "Knowledge-based embodied question answering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[8] "Amazon Mechanical Turk — mturk.com," https://www.mturk.com/.

[9] J. Duan, S. Yu, H. L. Tan, H. Zhu, and C. Tan, "A survey of embodied ai: From simulators to research tasks," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 6, no. 2, pp. 230–244, 2022.

[10] Q. Sima, S. Tan, H. Liu, F. Sun, W. Xu, and L. Fu, "Embodied referring expression for manipulation question answering in interactive environment," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 7635–7641.

[11] Y. Li, A. X. Feng, J. Li, S. Mumick, A. Halevy, V. Li, and W.-C. Tan, "Subjective databases," *arXiv preprint arXiv:1902.09661*, 2019.

[12] OpenAI, "Language models are unsupervised multitask learners," *OpenAI Blog*, vol. 23, no. 6, 6 2020, https://openai.com/blog/chatgpt.

[13] ——, "Gpt-4 technical report," 2023.

[14] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar *et al.*, "Llama: Open and efficient foundation language models," *arXiv preprint arXiv:2302.13971*, 2023.

[15] A. Chowdhery, S. Narang, J. Devlin, M. Bosma, G. Mishra, A. Roberts, P. Barham, H. W. Chung, C. Sutton, S. Gehrmann *et al.*, "Palm: Scaling language modeling with pathways," *arXiv preprint arXiv:2204.02311*, 2022.

[16] D. Driess, F. Xia, M. S. Sajjadi, C. Lynch, A. Chowdhery, B. Ichter, A. Wahid, J. Tompson, Q. Vuong, T. Yu *et al.*, "Palm-e: An embodied multimodal language model," *arXiv preprint arXiv:2303.03378*, 2023.

[17] J. Huang and K. C.-C. Chang, "Towards reasoning in large language models: A survey," *arXiv preprint arXiv:2212.10403*, 2022.

[18] V. S. Dorbala, J. F. Mullen Jr, and D. Manocha, "Can an embodied agent find your" cat-shaped mug"? llm-based zero-shot object navigation," *arXiv preprint arXiv:2303.03480*, 2023.

[19] G. Zhou, Y. Hong, and Q. Wu, "Navgpt: Explicit reasoning in vision-and-language navigation with large language models," *arXiv preprint arXiv:2305.16986*, 2023.

[20] D. Shah, B. Osiński, S. Levine *et al.*, "Lm-nav: Robotic navigation with large pre-trained models of language, vision, and action," in *Conference on Robot Learning*. PMLR, 2023, pp. 492–504.

[21] C. H. Song, J. Wu, C. Washington, B. M. Sadler, W.-L. Chao, and Y. Su, "Llm-planner: Few-shot grounded planning for embodied agents with large language models," *arXiv preprint arXiv:2212.04088*, 2022.

[22] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized bert pretraining approach," *arXiv preprint arXiv:1907.11692*, 2019.

[23] D. Müllner, "Modern hierarchical, agglomerative clustering algorithms," *arXiv preprint arXiv:1109.2378*, 2011.

[24] P. Cascante-Bonilla, H. Wu, L. Wang, R. S. Feris, and V. Ordonez, "Simvqa: Exploring simulated environments for visual question answering," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 5056–5066.

[25] J. Li, D. Li, S. Savarese, and S. Hoi, "Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models," *arXiv preprint arXiv:2301.12597*, 2023.

[26] J. Li, D. Li, C. Xiong, and S. Hoi, "Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation," in *International Conference on Machine Learning*. PMLR, 2022, pp. 12 888–12 900.

[27] W. Kim, B. Son, and I. Kim, "Vilt: Vision-and-language transformer without convolution or region supervision," in *International Conference on Machine Learning*. PMLR, 2021, pp. 5583–5594.