There’s a Microwave in the Hallway

Information Use in Embodied Question Answering Task

Master’s thesis in Computer science and engineering

Yasmeen Emampoor

Department of Computer Science and Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
UNIVERSITY OF GOTHENBURG
Gothenburg, Sweden 2022

MASTER'S THESIS 2022

There's a Microwave in the Hallway

Information Use in Embodied Question Answering Task

Yasmeen Emampoor

UNIVERSITY OF
GOTHENBURG

Department of Computer Science and Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
UNIVERSITY OF GOTHENBURG
Gothenburg, Sweden 2022

There's a Microwave in the Hallway
Information Use in Embodied Question Answering Task
Yasmeen Emampoor

Supervisor: Simon Dobnik, *Philosophy, Linguistics, and Theory of Science*
Co-Supervisor: Nikolai Ilinykh, *Philosophy, Linguistics, and Theory of Science*
Examiner: Koen Claessen, *Computer Science*

Master's Thesis 2022
Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Typeset in LaTeX
Gothenburg, Sweden 2022

There's a Microwave in the Hallway
Information Use in Embodied Question Answering Task
Yasmeen Emampoor
Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg

Abstract

Embodied Question Answering (EQA) is a task in which an agent situated in virtual environment navigates from its current position to an object (Navigation), and then answer a question about it (Visual Question Answering, VQA), for example "What color is the table in the table in the kitchen?" This project examines how an agent modelled as a deep neural network uses semantic information from its language model and visual information to answer questions in the second task. This is important since due to the regular nature of the task and the dataset it could be that the model is answering questions purely based on general semantic information from its language model (tables are frequently brown) and not relying on the visual scene, a phenomenon that is commonly known as *hallucinating.*

This project first examines the quality of the current task dataset, EQA-MP3D, and presents a series of experiments where the visual information given to the model is manipulated or corrupted. Next, this model is extended, giving it new sources of information with an expectation that the model would use it to improve grounding of questions and answers in perception. Structured information is found to be particularly helpful, in the form of identified object regions.

Additionally, we examine the impact of question types on performance. The dataset includes 3 distinct question types, `color`, `color room`, and `location`. The baseline performance differs across types. The performance is also impacted by changes in the input differently by question type.

Keywords: embodied question answering, visual question answering, multi-modality, information fusion

Acknowledgements

Thank you to Simon and Nikolai for your supervison, to Ali Aruqi for sharing meetings and your insight on the dataset, to Koen and Joachim for all of your help, to Robert Adesam for your help with the MLT GPUs.
Finally, thank you to my English teacher mother for your proof-reading. All mistakes that remain are my own.

Yasmeen Emampoor, Gothenburg, February 2022

Contents

List of Figures

List of Tables

1

Introduction

Robots have the potential to be useful in many fields; they can perform tasks that are dangerous for humans, as well as tasks that are simply tedious. These robots are often operated based on pre-defined actions, or remotely. The potential increases in efficiency from being able to give a goal or task in natural language and have the robot interpret and determine how to carry out this task are huge. This would make robots more accessible generally, since it would reduce the learning curve for operating them. A patient in a hospital would likely find it much easier to simply tell the robot 'I need water', rather than having to find the 'water' option on some sort of touchscreen interface. It also would support situations where the instructions being given are too complex for this kind of interface; one example of this is human-robot teaming, in which groups of humans and robots cooperate to achieve tasks in situations such as disaster scenarios [1]. Communication in natural language can facilitate organisation where all of the humans are working with all of the robots, rather than an individual human having control over one or many robots. Situations like this require complex understanding of natural language instructions. For a household agent, it would be beneficial to be able to interact with a robot in a similar way to how one would with another person. However, this is only useful if it is giving a relevant answer!

```
Human: Go get my computer from the office.
Robot navigates to the office, finds the computer, brings it back.
Human: Were my keys in the office?
Robot: I don't know, I'll go check.
Robot navigates to the office, finds the keys. Returns.
Robot: Yes, they are on the bookshelf.
Human: Thanks.
```

For this project, we consider a simpler conversation: the human asks a question, and the robot answers.

```
Q: Where is the computer?
A: In the office.
```

In the Embodied Question Answering task, an embodied agent is asked a question, has to navigate to the appropriate place to answer the question, and answers the question. The project focuses on the question answering aspect of the task, after navigation to the object or location has already been performed. In this project, we examine how a simple agent model for the Embodied Question Answering task uses information encoded in the features that have been chosen and trained upon. This is an active area of research, since one concern with EQA models, and combined visual/linguistic models in general, is how much they are actually using the visual information, rather than relying on linguistic priors and biases. To be effective, an agent needs to able to identify specific objects and locations, without being directly trained on that environment, especially since environments change. Furniture might move, a wall might be painted, and the agent needs to be able to adapt. Just considering accuracy can give an inflated idea of an agent's performance on the task, since correct answers to questions could be due to linguistic biases or imbalances in the dataset. This project uses a number of methods to try to determine what features contribute to the model's performance.

The contributions of this work are as follows:

- We examine the performance of the VQA portion of Das et. al's EQA model. We look at how it performs under feature ablation where the visual component is blindfolded in several different ways.
- We study how additional additional structural and semantic representations affects the performance of the model.
- Through this, we point out several shortcomings of the MP3D-EQA dataset and identify possible directions for improvement.

Using a series of blindfolding experiments, experiments which disrupt the visual input, on the model presented by Das et. al, we investigate how the model uses the information given–visual (images from the end of navigation) and linguistic (the question) [2]. We next explore giving the model new information, through added input (semantic categories, object segmentation), modified points of view (turning the camera at the end of navigation), and transfer learning (using an object detector trained on a large image dataset). The main metrics considered in this project are mean rank and accuracy on the MP3D-EQA dataset's evaluation split [3].

The expectation was that the blindfolding experiments would show reduced performance on these metrics if the model was using the visual information. This was the case, and comparing the reductions across different blindfolding strategies gave insights into *how* the visual information was being used. For the experiments that increased or restructured the information available to the model, increases in performance on these metrics would show that the new information or structure was useful. Here we had mixed success. We found that the usefulness of information and structures varied by question type. This builds on previous work that has mainly discussed question types in terms of dataset generation, but has not heavily explored the impact the question types have on performance [2][3].

The structure of this report is as follows: we begin with background on question answering and embodied agents in Chapter 2, followed by an explanation and discussion of the tools and datasets used for the project in Chapter 3, the experiments conducted in Chapter 4, and finally ending with discussion of the results and suggested follow-up experiments in Chapter 5.

2

Background

This chapter presents background on Embodied Question Answering and related tasks, as well as the use of simulation in robotic research, with a presentation of the project's research questions in 2.6.

2.1 Grounding

Grounding is the process of giving symbols meaning. This is a fairly complicated undertaking. In 1980, Searle, presented the 'Chinese Room Argument' [4]. This argument proposes a situation in which a person who does not read any Chinese is put in a room and presented with a large amount of writing in Chinese. Next, they are given a second batch of Chinese writing, along with rules, in English, for how to connect the first batch to the second batch. Finally, they are given a third batch of Chinese writing, again with rules in English to connect to the previous batch, as well as how to produce 'responses' in Chinese based on the third batch. The participant is unaware that those providing the Chinese writing have set these batches up as a 'script', a 'story', and 'questions', and that the rules for the third batch produce 'answers'. Searle then poses that the participant could become so good at following the provided rules, that the answers they provide, in Chinese, could be as good as answers they could provide in English (assuming that is their native language), without ever understanding the content of the writing. Based on this, Searle argues that an AI system performing this task can not be seen to be understanding any more than this participant. From this, Harnad suggests that some form of sensory input is required to give symbols meaning, and says that 'there is really only one viable route from sense to symbols: from the ground up" [5][6]. Grounding is key in creating robots that can interact with humans in natural language. The robot must be able to link objects and actions to words. For example, what does 'turn' mean? In 2002, Lauria et al. presented a project in which a robot was given natural language instructions, which were mapped to procedures of identified primitives, such as 'turn' [7]. More recently, Hermann et al. presented a simulated embodied agent learning grounded language through a combination of unsupervised and reinforcement learning, with minimal prior knowledge [8]. Another approach with minimal prior knowledge was presented by Thomason et al., where the robot learns through dialogue with a human [9].

Grounding is also a challenge in tasks such as image captioning where models need to produce labels for both the objects in the image and descriptions of the objects' interactions with each other in natural language [10].

There has been argument that language models are able to encode perceptual information without grounding the text to perceptual input. [11] finds that large language models, trained only on text, no visual input, are able to produce information about color similarities/distances (e.g. red and orange are close colors).

2.2 Visual Question Answering

Visual Question Answering is a task in which an agent must answer a question based on an image; these questions are "free-form and open-ended" [12]. This task differs from image captioning in that answering these questions requires retrieving directed information, considering both the image and the question. The question may concern items that are not the obvious focus of the image. There is a huge variety in possible question types in this task, including object detection ("how many...?"), activity recognition ("Is this person doing ...?"), and knowledge-base reasoning ("what is ... made of?"). VQA is a somewhat unusual natural language processing task in that it is often implemented as a classification task, with the agent choosing from a large number of potential answers, rather than generating an answer. This means that the success of VQA can often be measured as accuracy to the ground truth answer. However, VQA can also be implemented as a generation task, for which other measures, like BLEU, which compares a generated output to one or more good human outputs, can be used [13][14].

2.3 Navigation

Navigation is a common task for robots. Bonin-Font et. al describe it as "the process of determining a suitable and safe path between a starting and a goal point for a robot travelling between them"[15]. A variety of navigation strategies exist, including map based strategies and strategies involving visual landmarks. Computer vision researchers also are exploring how different representations of visual input affect navigation [3].

One navigation task is Object Navigation, in which the agent, given an object's label, navigates to the object [16].

2.4 Embodied Question Answering

Embodied Question Answering combines the navigation and VQA tasks into one: the embodied agent must navigate to find the object that the question refers to [2]. For example, for the question 'What color is the refrigerator?' the agent must identify where the fridge is most likely to be, the kitchen, and navigate there before identifying the fridge and answering the color. EQA has been expanded to

Multi-Target Embodied Question Answering, in which questions can include multiple targets, allowing for comparison questions such as ’is the oven in the kitchen the same color as the sink in the bathroom?’ [17]. Current approaches for this task use templates for generating questions, due to the costs of collecting a large enough dataset from humans. The EQA dataset includes nine question types: `location`, `color`, `color_room`, `preposition`, `existence`, `logical`, `count`, `room_count`, and `distance`, though the EQA-V1 dataset, used for the experiments by Das et.al includes only the first five question types [2]. The MT-EQA dataset adds six comparison question types [17].

One concern with both EQA and VQA models is how much they actually incorporate the visual input in determining an answer. Anand et al. [18] conducted an experiment on the EQAv1 dataset that found equivalent to slightly better performance on the question answering task using simple question only models with no visual input. This is interesting in that it suggests that the models are doing a good job learning common sense knowledge from the textual content, but is a problem in this particular task, since it means that the agent is not actually adapting its answers to the specific situation. [19] creates a balanced VQA dataset, in which each question is paired with two images with different answers. For example, ’Is the umbrella upside down?’ has an image for yes, and an image for no. This dataset reduces models’ ability to exploit language priors.

Another concern is the ’naturalness’ of the questions. This is more of a concern for EQA; ’What color is the table in the living room?’ is just not a question that comes up particularly often. Color questions are also likely to have language priors to exploit. Although pink refrigerators exist, they are quite rare, and a model is likely to get the answer correct quite often just by learning that appliances are usually white or silver. A dataset with a number of unusually colored objects might be beneficial in learning grounding in vision, but it is also important that the questions also encourage use of vision. Prepositional questions, for example, would be useful for this. Although there may be patterns in whether items are to the left or right of something, they are not as strong as patterns in object colors.

An extended version of this task is Interactive Question Answering (IQA), in which the questions require interaction with the environment, such as ’Do we need milk?’, where the agent would have to open the refrigerator [20]. This extension should overcome some of the concerns about EQA models, since the agent’s interactions with the environment, such as opening the refrigerator, would be observable from the outside. There is, however, still the concern that the model could learn patterns such as ’people are more likely to ask if they need milk if there is little or no milk’.

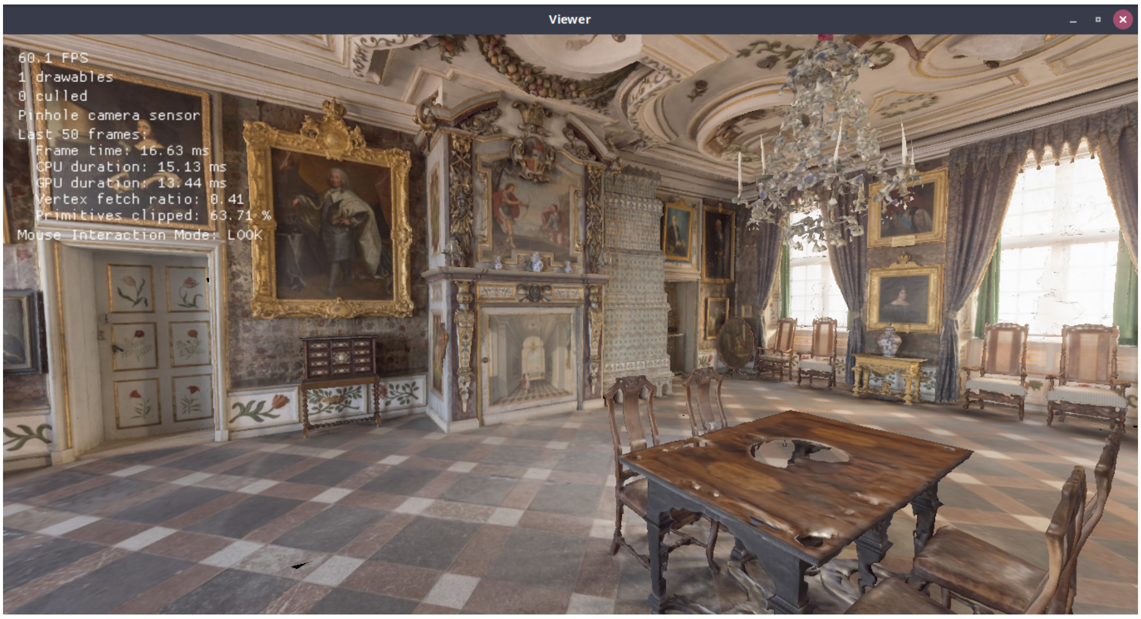
2.4.1 Dialogue

VQA and EQA can be seen as simplified dialogue tasks. They contain a limited number of question types, and the agent sticks to answering rather than employing more sophisticated strategies, such as asking questions to acquire more information when it is unsure. One interesting note about dialogue interactions between humans and robots is that humans automatically adjust their strategies when interacting with a computer. Tenbrink et al. found that people gave generally sparser commands when interacting with a computer system than they did interacting with another human, even when not given instructions to do so [21]. Humans also adjust while speaking to each other. One important aspect of dialogue is reference. In dialogue, references are often built by the participants together, via questions, clarifications, and agreement, among other strategies [22]. This shows that there is a clear step that will need to be taken between EQA and full interaction. Another important note is that dialogue is often spoken, and in the situation of an embodied agent, that would be the expectation. However, spoken language differs from written in a number of ways, including pronunciation, false starts, and interruptions [23].

A current area of research is Visual Dialog, in which an agent holds a conversation with a human about some visual content [24]. A relevant dataset for visual dialogue in an interior setting is Meet Up!, a corpus of dialogue and images from where two people played a game in a simulator–the participants were dropped into two separate rooms, and had to navigate to each other by describing the rooms that they were seeing [25]. A similar dataset is Where Are You (WAY), in which one participant is the Observer, who has a first person view of the space, and the Locator, who must determine where the Observer is in a top down map by asking the Observer questions [26]. This task leads into another area of research, navigation based on dialogue, in which the agent navigates based on language instructions, and can ideally ask questions for clarification (making it a dialogue). Like visual dialogue, datasets for dialogue-based navigation are often collected by having humans play both roles. Cooperative Vision-and-Dialog Navigation is a dataset of over 2000 navigation dialogues [27]. It was collected via Mechanical Turk crowd-sourcing; pairs used the Matterport3D simulator and a chat interface, with one person, the oracle, able to see the ideal moves for the navigation task, giving natural language instructions to the other person, the navigator'. The navigator could ask clarifying questions. The oracle was also shown the navigator's current visual frame. A similar but smaller dataset is RobotSlang, consisting of 169 dialogues between a commander referencing a static map and a human driver only able to see the camera view of the robot they were controlling [28]. The commander relayed instructions to the driver, based on their understanding of where the robot was. The driver was able to ask clarifying questions, such as where exactly to turn, and the commander could periodically ask localization questions, such as what color wall the driver could see.

2.5 Simulation

Working with embodied agents is resource intensive and makes reproducibility difficult to impossible, so simulation is beneficial for research. Within simulation, environments can be kept consistent, allowing for both reproducibility of an experiment and for comparison of different systems or methods. Simulation also allows for the reuse of datasets of human descriptions or labels, which are time-consuming and expensive to produce. Multiple simulation platforms for working with embodied AI are available, including AI Habitat, MINOS, and RoboTHOR [29][30][31].



```
60.1 FPS
1 drawables
0 culled
Pinhole camera sensor
Last 50 frames:
  Frame time: 16.63 ms
  CPU duration: 15.13 ms
  GPU duration: 13.44 ms
  Vertex fetch ratio: 0.41
  Primitives clipped: 63.71 %
Mouse Interaction Mode: LOOK
```

Figure 2.1: A screenshot from the habitat sim interactive viewer

AI Habitat is used for this project. Habitat's current main focus is navigation, mainly through indoor spaces, but there is some ability for object interactions–for example moving a chair from one point to another. New objects can also be added to the space. Fig. 2.1 shows a screenshot from the habitat sim interactive viewer. Habitat-PyRobot Bridge is a library, written by members of the Habitat team, to support the transfer of a simulated agent in Habitat to a physical robot [32]. Various scene datasets are supported by Habitat, the most used one being Matterport3D, a dataset of real interiors with human annotation of objects [33]. Other datasets are also available, such as Replica and Gibson [34][35].

2.6 Research Questions

Embodied Question Answering is interesting in that it requires specific identifications; people's homes will have multiple tables, and if someone asks their embodied agent about a specific table, they need the agent to be able to identify it. However, it is also a situation where one does not want to have to train their agent from scratch in every new location. If we see this as a step towards a house or office assistance agent, the typical user cannot generate their own dataset for their location and then spend days training the agent in its current environment. The agent needs to be able to give specifics about objects it may never have seen before. So, we need an agent to be able to leverage information from other locations or contexts, while being able to be specific about the current location. Broadly, the questions this project investigates are:

- What are the limitations of the current EQA dataset (in regards to the VQA task)?
- Is the model using available information in the expected way?

Is it identifying specific objects visually?

- Does a similar model given more information perform better?
  Information from the same context?
  Information from another context?

3

Toolbox

We are working with AI Habitat, a simulation platform for working with embodied AI [29]. It consists of two parts, Habitat-Sim, the 3D simulator, and Habitat-Lab, the library for embodied AI development.

3.1 The Model

We are starting with the Embodied Question Answering baseline in Habitat-Lab, which consists of three parts, a Convolutional Neural Network (CNN) for initial feature extraction, a navigation module, called PACMAN, and a question answering module [2]. The CNN feature extractor is trained on three tasks: RGB reconstruction, semantic segmentation, and depth estimation. The navigation module is trained to imitate shortest path navigation. The question answering module is given the last five frames of navigation (in training taken during ground-truth shortest path navigation, where a frame is the view the agent has after taking an action), and then predicts an answer from a set of possible answers (approaching this as a classification task). The training of the Habitat-Lab version of the model[1] differs to the version presented in the Embodied Question Answering paper [2]. For the original model, the CNN, question answering, and navigation modules were all trained separately, and then reinforcement learning was used to fine-tune the navigation module and more strongly link the question answering and navigation modules, by using successful question answering as part of the reward for the navigation module. This reinforcement learning is missing from the habitat version of the model; the components of the model are only trained separately.

A diagram of the Habitat version of the VQA portion of the baseline model can be seen in 3.1. Fully connected layer refers to a sequence of a linear layer, a ReLU layer, and a Dropout layer with p=0.5.

[1] available here: `https://github.com/facebookresearch/habitat-lab/tree/master/habitat_baselines/il`

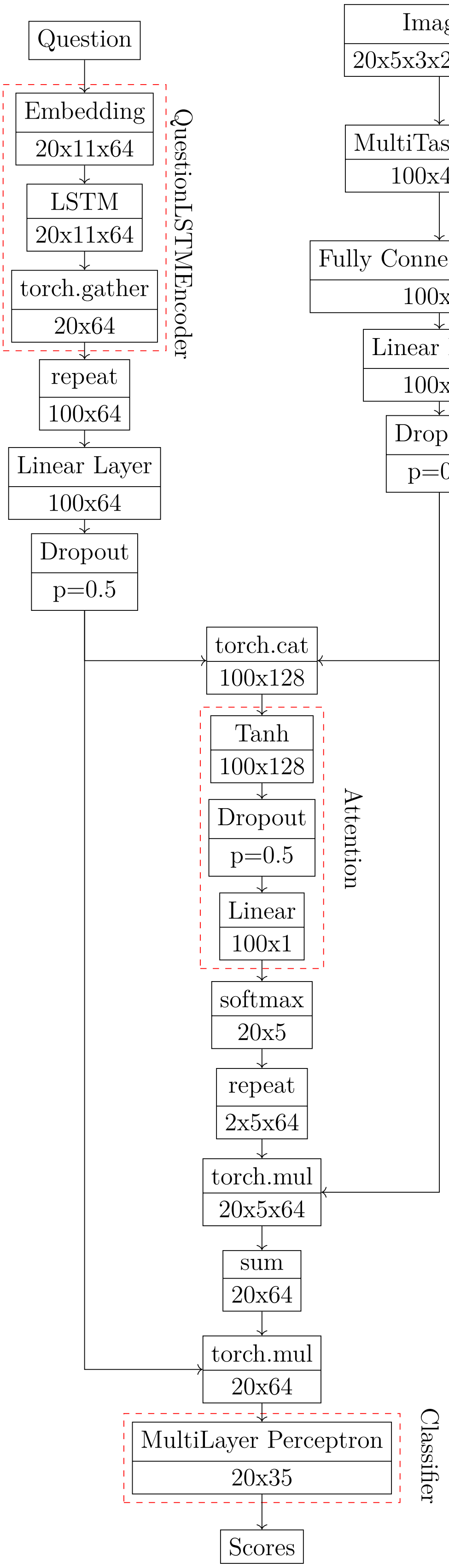


Figure 3.1: VQA Model

3.2 The Datasets

We are using the Matterport3D dataset, a dataset of real interiors with human annotation of objects, as our scene dataset [33]. We are using the MP3D-EQA task dataset, created using code to automatically generate questions and answers to correspond with annotated scenes in the Matterport3D dataset[2] [3]. An example episode from the task dataset can be seen in Appendix B. The scene dataset is used by Habitat to render the scenes. The task dataset is used to place the agent in the scene, and snapshots (frames) are taken at each position in the navigation path. These frames are then used as the visual input for the VQA model.

3.2.1 Interiors

3.2.2 MP3D-EQA Dataset

This dataset contains questions of three types: `color_room`: *What color is the <obj> in the <room>?*, `color`: *What color is the <obj>?*, and `location`: *What room is the <obj> located in?*. This dataset is based off of the EQA-V1 dataset, which was developed by Das et al. and used in the development of the EQA model described above[3] [2]. There are some differences between the datasets, however. The EQA-V1 dataset included a fourth type of question, `prepositional` questions: *What is <on/above/below/next to? the <obj> in the <room>?*, but these are not present in MP3D-EQA. They were removed because, [3] say, "we found those questions in MP3D to be relatively few, with strong biases in their answers". `Existence` questions, *Is there a <obj> in the <room>?*, were also included in EQA-V1, but [3] doesn't mention them at all. Other question types were proposed in [2] but not implemented. EQA-V1 was built based on the SUNCG scene dataset, which is no longer available, due to a legal dispute [36].

The dataset has the most 'color_room' questions, as shown in Tab. 3.1.

Table 3.1: Question Type Breakdown

question type	**percentage of training set**	**percentage of evaluation set**
color_room	69.85908	68.46154
color	15.91858	17.69231
location	14.22234	13.84615

The dataset has a fixed train/evaluation split. There is one object which only occurs in the evaluation set ('toaster'), and one answer that only occurs in the evaluation set ('gym'). More details about the answers can be found in Appendix C.

[2] The dataset is available in the habitat-lab repository here: `https://github.com/facebookresearch/habitat-lab/#data`

[3] Code for generating EQA-V1 questions is available here: `https://github.com/facebookresearch/EmbodiedQA`

3.2.2.1 Limitations

The paper which introduces the MP3D-EQA dataset was mainly focused on the navigation aspect of the task, and the question types reflect that. A `color` or `location` question should theoretically be a good indication of whether or not the agent has successfully navigated to the object. On the other hand, actually answering the question may not require as complex of reasoning in that it shouldn't require a long memory to answer.

However, color identification is actually a very difficult task. One issue is purely visual–an object's color looks different in different lighting conditions, so someone might see something as light grey in good light, but in dimmer light only see it as grey. Another, more complex issue, is related to language: the way that humans identify and refer to colors is context dependent. Monroe et al. created a dataset of color descriptions using pairs of Amazon Mechanical Turk participants [37]. One member of the pair was the speaker, the other was the listener. Both people could see the same three color samples, and the speaker was given one of them as the 'target', which they needed to convey to the listener. There were three possible situations: baseline, or 'far', where all three samples were very distinct, such as pink, green, and yellow; split, where one of the other colors (called distractors) was close to the target, such as two shades of blue, and then the last distractor was far, for example yellow; and close, where both distractors were similar to the target (all shades of blue, for example). They found that in the 'split' and 'close' conditions, the speaker used more comparatives (i.e. lighter blue) and superlatives (i.e. the lightest one), and also was more likely to use 'high specificity' color terms, such as 'magenta' or 'teal', using more basic terms, such as 'red' or 'blue', in the 'far' condition. This suggests that people would adjust their descriptions of an object's color based on what is around it, which is not taken into account by this dataset.

Another color issue, specific to the MP3D-EQA dataset, is that the colors used for annotation (done by Amazon Mechanical Turk workers), come from Kenneth L. Kelly's 'Twenty-two colors of maximum contrast', with the addition of 'off-white' and 'slate grey', since they are very common in indoor scenes [38][3]. However, Kelly's colors were not designed to be natural color descriptions; they were developed as a set of colors that could be used in situations where contrast was needed–for example color coding of graphs. This is shown by the inclusion of both 'buff' and 'yellowish pink' in the color set, both of which are unlikely to be common color descriptions. (The names have been adjusted slightly in the EQA dataset. Buff is tan, and all of the 'color-ish color' names have been changed to simply 'color color', making them less natural.) Another issue is that the set contains 'white' and 'off-white', as well as 'grey' and 'slate grey', pairs which are likely to be confused in different lighting conditions.

Another issue with the dataset is the answer distribution. As can be seen in Fig. 3.2, some answers, such as brown, are overrepresented in the dataset. The dataset was developed in an experiment focused on navigation, so some items in the dataset are identical except for the navigation path. For the VQA portion, this means that the

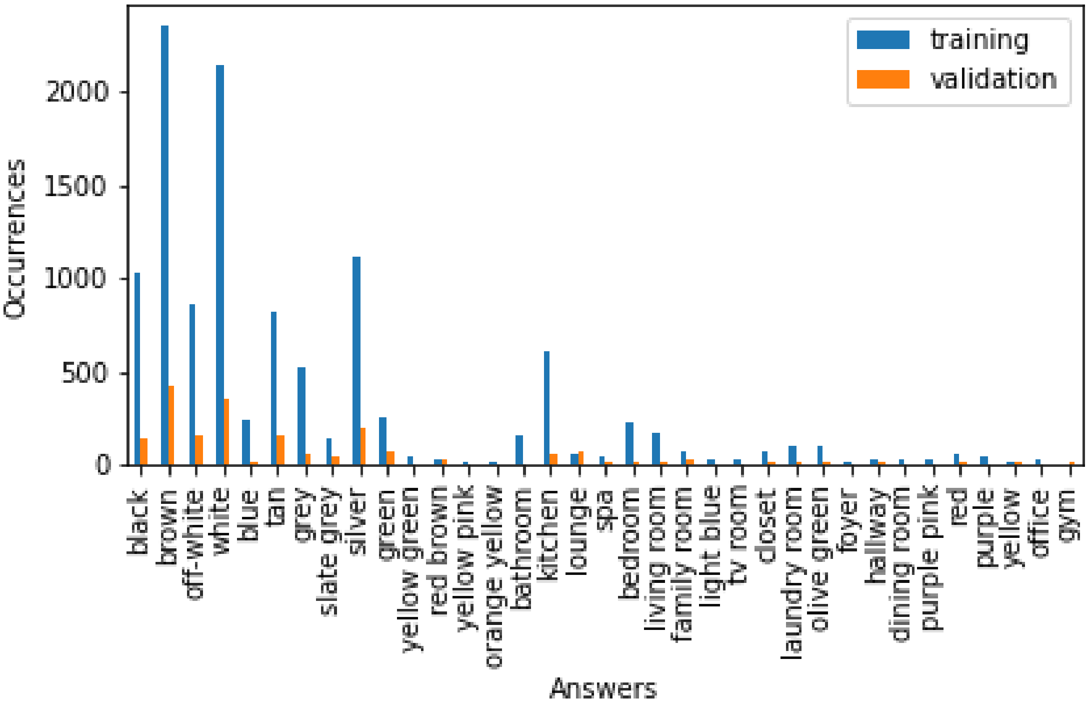
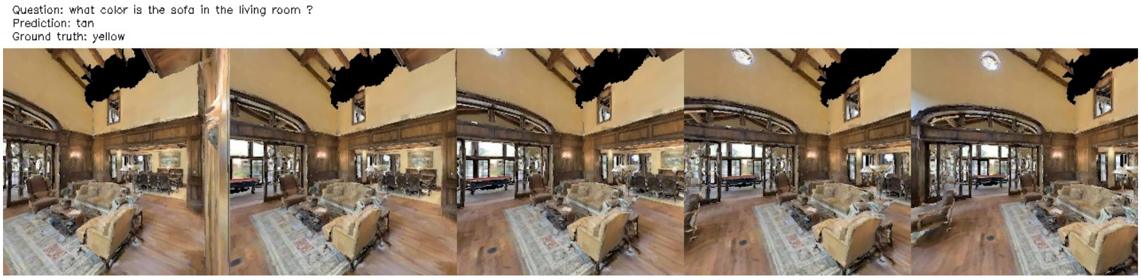


Figure 3.2: Answer Distribution

images likely have limited differences from each other and the training of the VQA model contains sets of 15 nearly identical training items for each base question.

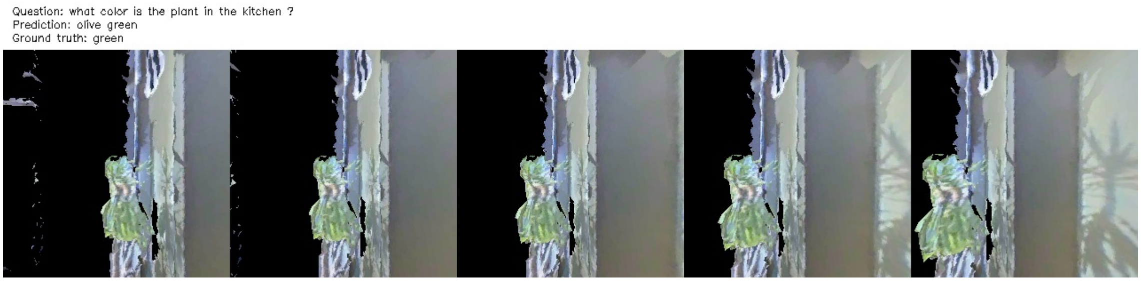
The dataset of questions and answers for EQA in Habitat was automatically generated, and may contain some errors. One example of this is shown in Fig. 3.3, in which the VQA model answered that the sofa in the living room is tan, but the ground truth answer is that it is yellow. However, looking at the image, I see a tan sofa and a yellow armchair. It seems that at some point, the armchair was annotated as a sofa, but the model is identifying the tan sofa as the sofa being asked about.



Question: What color is the sofa in the living room?
Prediction: tan
Ground Truth: yellow

Figure 3.3: Error Example

One last issue with the dataset is that the render quality is sometimes poor. Most scenes render well, but an example of a very badly rendered scene is Fig. 3.4.



Question: What color is the plant in the kitchen?
Prediction: olive green
Ground Truth: green

Figure 3.4: Rendering Error Example

4

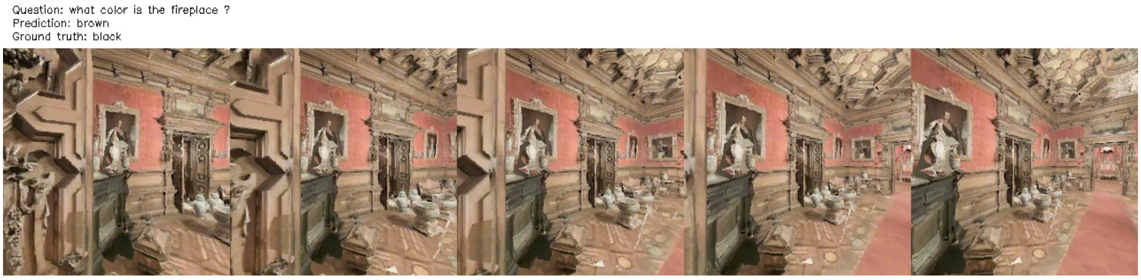
Experiments

These experiments are designed to investigate different aspects of information given to the agent, in order to improve performance in the question answering portion of an embodied question answering task. The first experiment establishes the baseline for comparison. The second experiment is really a series of experiments, in which visual information is removed or disrupted, in order to establish how the baseline model is making use of visual and textual information. The third experiment gives the agent categorical information about the things it sees–another type of information about its current location. The fourth experiment gives the agent new viewpoints during question answering, broadening the visual information the agent has about its current location. The fifth experiment replaces the original CNN with a pre-trained object detector CNN, and experiments with different initial processing of the visual information, as well as the use of transfer learning from a broader context. Also included is a short study to determine the impact of the dataset imbalance on learning by balancing the question types.

4.1 Experiment 1: Baseline

4.1.1 Method

The first step is to train and evaluate the baseline to be used as the point of comparison for all following experiments. This is the CNN and VQA portions of the EQA baseline in habitat-lab, described in 3.1. A diagram of the baseline VQA model can be seen in Fig. 3.1. An example of the input frames and question and output answer can be seen in Fig. 4.1.



Question: What color is the fireplace?
Prediction: Brown
Ground Truth: Black

Figure 4.1: Example VQA Result

4.1.2 Results

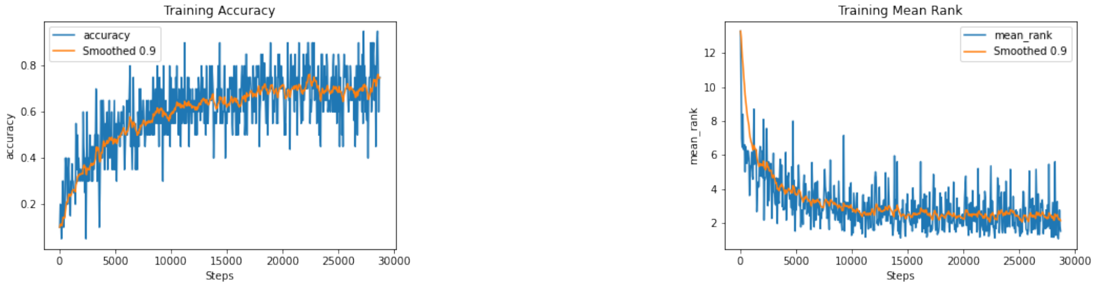
Fig. 4.2 shows metrics for each batch during training, with a weighted average shown in orange. Fig. 4.3 shows metrics averaged for each epoch during the baseline evaluation. As can be seen in Fig. 4.4, the evaluation loss decreases until epoch 7, where it spikes, the model achieves its lowest loss at epoch 8.

Table 4.1 shows the model's epoch with the lowest loss.

Table 4.1: Lowest Loss Epoch During Baseline Evaluation

Checkpoint	8
Loss	2.204141
Overall Mean Rank	4.35231
Mean Rank on Color Room Questions	3.611236
Mean Rank on Color Questions	2.692754
Mean Rank on Location Questions	10.137037
Overall Accuracy	0.38
Accuracy on Color Room Questions	0.373783
Accuracy on Color Questions	0.527536
Accuracy on Location Questions	0.222222
Kappa Score	-0.004667

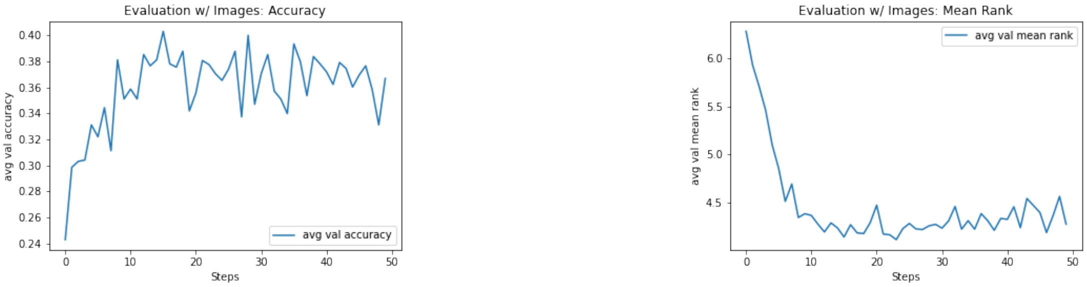
The mean rank metric shows how well the model is ranking the answers. This is a useful metric to consider, since the answer distribution is unbalanced, as seen in



(a) Accuracy

(b) Mean Rank

Figure 4.2: Training Metrics



(a) Accuracy

(b) Mean Rank

Figure 4.3: Baseline Evaluation Metrics

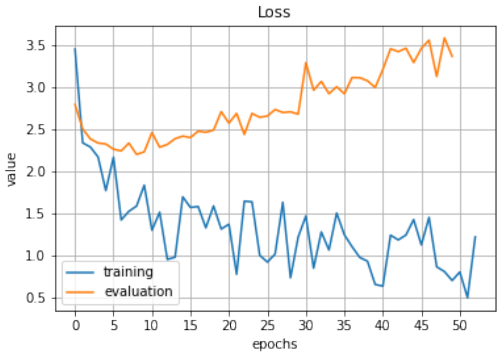


Figure 4.4: Baseline Training and Evaluation Loss

Fig. 3.2. For this metric, the lower the rank, the better. In this case, the maximum possible rank is 35, because that is the number of possible answers. An average mean rank of 35 would mean that the correct answer was in last place. We can see that the baseline model performs slightly better on the `color` questions than on `color room`, and much better than on `location questions`.

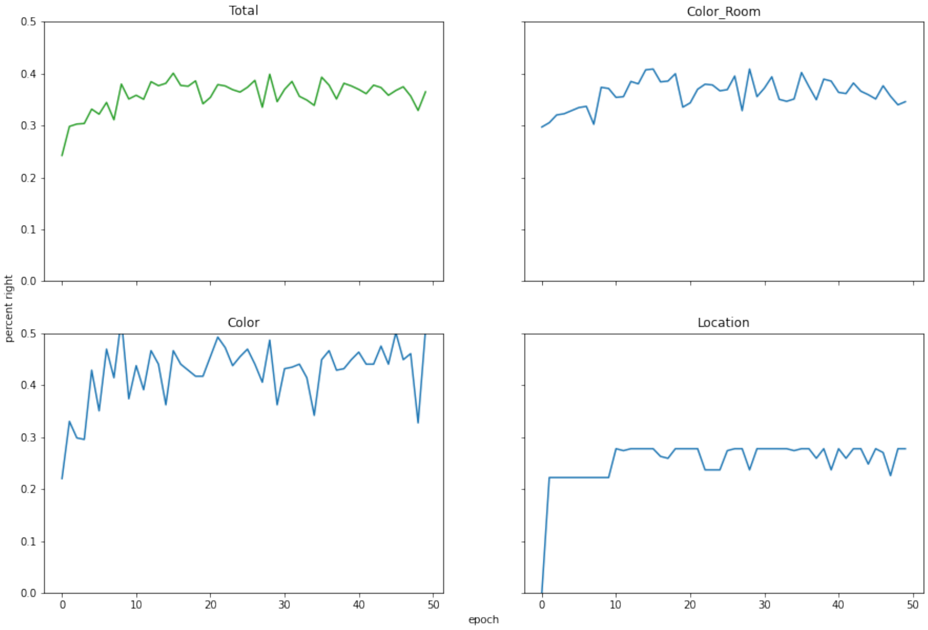


Figure 4.5: Baseline Correct Answers by Question Type

When exploring accuracy, there are a number of different factors to consider. One is question type. As can be seen in Fig. 4.5, using accuracy as a metric, the question type performance aligns with the performance considering mean rank.

`Color` is the simplest question type, requiring only that the object is recognized, and that the color is recognized. This may help to explain its higher performance compared to the other question types. `Location` questions in particular require recognition of multiple features and objects to determine the room, rather than only the target object. However, at this point, the `color room` question type should be the same as `color`–the navigation to the correct room is already done. The extra information in the question may be hampering performance. However, the reduced performance could also be due to larger variety in the `color room` questions; the category is much larger. An experiment reducing the size of the `color room` question type category could be done to explore this further. One note here is that `color room` questions might be more interesting in models with memory–a model where the agent is not expected to be looking at an item when answering questions about it.

Another factor to consider is the distribution of potential answers, since the problem is being attacked as a classification task. There is the possibility of getting the correct answers by chance, and inflating the accuracy. A metric that takes this into account is Cohen's Kappa, shown in Eq. 4.1 [39].

$$\kappa = (p_0 - p_e)/(1 - p_e) \tag{4.1}$$

where p_0 is the observed agreement, in this case accuracy, and p_e is the expected agreement, calculated by Eq. 4.2 [40] [41].

$$p_e = \sum_{k \in K} P(k|classifier) \cdot P(k|ground\_truth) \tag{4.2}$$

A kappa value below zero means chance agreement, and the closer the value is to 1, the higher agreement is found. Kappa is most commonly used in annotation tasks, to measure inter-annotater agreement, but our case, one of the annotators is the model, and the other is the ground-truth for the evaluation set. The kappa value is a harsh metric, focused on how well the model is learning specifics. Learning distributions is part of what the model is expected to do, but this metric factors that out of the model performance, and estimates how much the model is able to learn from features. Kappa values are also lower for imbalanced datasets, which, as can be seen in Fig. 3.2, ours is.

A table showing suggested division for interpreting the strength of a kappa value, suggested by Landis and Koch, can be seen in Table 4.2 [42]. As can be seen in Table 4.1 the kappa score for the baseline model reports poor/chance agreement. This range from Landis and Koch is designed for interpretation of kappa results for human annotation, however, and does not address the issue of dataset balance.

Table 4.2: Kappa Interpretation

Kappa Statistic	Strength of Agreement
<0.00	Poor
0.00-0.20	Slight
0.21-0.40	Fair
0.41-0.60	Moderate
0.61-0.80	Substantial
0.81-1.00	Almost Perfect

4.2 Experiment 2: Manipulation of Visual Input

In this series of connected experiments, we examine if and how the model is using visual input.

4.2.1 Blindfolded Evaluation

4.2.1.1 Method

In this experiment, we would like to determine if the model is actually considering the visual input when answering questions. This is done via a blindfolding test. There are a number of different ways to conduct blindfolding tests, depending on what you want to test. In this case, to investigate what the model is learning during training, an experiment is conducted in which the model is trained normally (the baseline model was used), and then blindfolded only during evaluation. If the model's performs worse blindfolded, it is using and learning from visual information.

The model is given zeroes instead of the visual content of the scene. This is done by duplicating and modifying the method that converts the .jpgs into numpy arrays to be input to the model, so that it produces an array of zeroes of the same size instead, sending a black image as the input to the model. An example can be seen in Fig. 4.6. The point of this blindfolding is to determine if the VQA model is considering the visual input when answering the questions. However, one consideration here is that the incorrect visual input could be disruptive to the model. The corrupted images are sent into the CNN at the beginning of the model, and the output values from the CNN are propogated through the model. The question is multiplied by disrupted values, and this output is what is sent to the classifier. This means that disrupting the initial visual input also affects the language later. Fig. 4.7 shows the path the information from this modified visual input takes through the model, highlighted in red.

Question: what color is the plant in the hallway ?
Prediction: green
Ground truth: white

Question: What color is the plant in the hallway?
Prediction: green
Ground Truth: white

Figure 4.6: Example Blindfolded Result

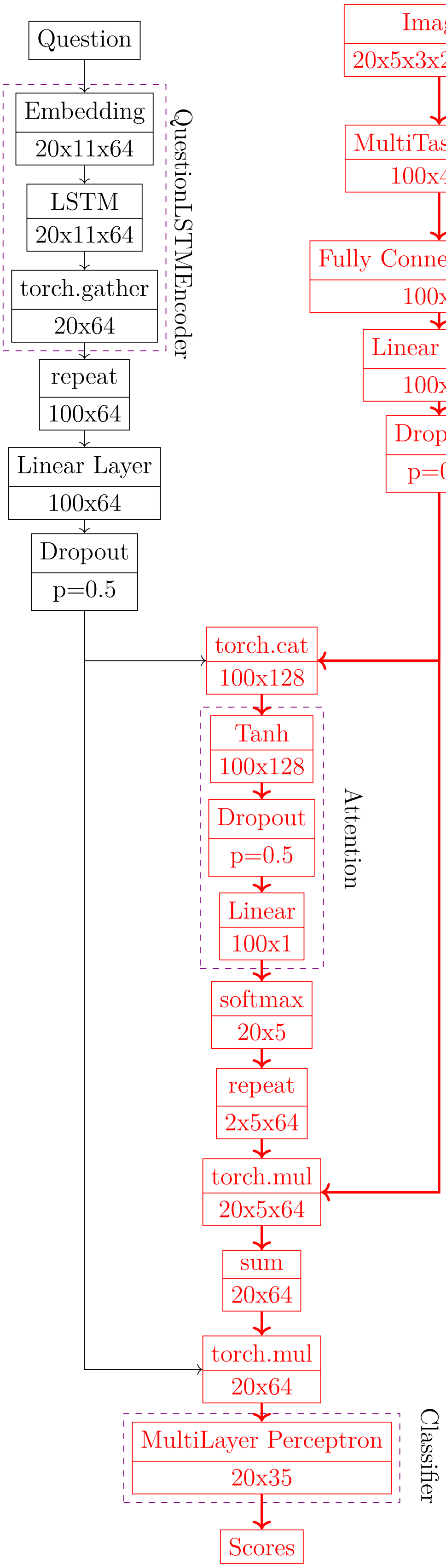


Figure 4.7: VQA Model with path of modified visual input traced in red

4.2.1.2 Results

Table 4.3 shows the lowest loss epoch of the model when evaluated on black images. We can see that, on black images, the model has reduced accuracy on all question types except `location`, and higher mean rank on all types.

Table 4.3: Evaluation Metrics for Model evaluated on Black Images

Metric	Value	Difference from Baseline
Checkpoint	9	1
Loss	2.364372	0.160231
↓ Overall Mean Rank	5.507692	1.155382
Mean Rank on Color Room Questions	4.561798	0.950562
Mean Rank on Color Questions	3.086957	0.394203
Mean Rank on Location Questions	13.277778	3.140741
↑ Overall Accuracy	0.246154	-0.133846
Accuracy on Color Room Questions	0.258427	-0.115356
Accuracy on Color Questions	0.217391	-0.310145
Accuracy on Location Questions	0.222222	0
↑ Kappa Score	0.003501	0.008168

Evaluation results are reported for the lowest loss epoch during evaluation. The second column indicates absolute difference from the baseline. We also report relative percent difference for mean ranks. Green rows indicate improvement. Intensity of row color is scaled by magnitude of difference from baseline. The same applies to all evaluation tables later in this report.

4.2.2 Blindfolded Training

4.2.2.1 Method

For this experiment, the same function used to blindfold the model in Experiment 1 is also applied during training, in order to to attempt to address the question of how much the model is being confused by the 'new' case of a black image after training with normal images during Experiment 1, as well as to have a 'text-only' baseline without changing the architecture of the model. However, the pre-trained CNN is not retrained.

4.2.2.2 Results

As can be seen in Table 4.4, the model trained on black images has reduced accuracy on all question types, but slightly lower mean rank on `color` and `color room` questions. The loss is incredibly high. When trained on black images, the model's decrease in performance from baseline is less than when the model trained normally is evaluated on black images, except for `location` questions, which are not effected by the blindfolding in the first experiment, but have a 22% decrease in accuracy here.

Table 4.4: Evaluation Metrics for Model Trained and Evaluated on Black Images

Metric	Value	Difference from Baseline
Checkpoint	9	1
Loss	67201.890625	67199.686484
↓ Mean Rank	4.453846	0.101536
Mean Rank on Color Room Questions	3.157303	-0.453933
Mean Rank on Color Questions	2.26087	-0.431884
Mean Rank on Location Questions	13.666667	3.52963
↑ Overall Accuracy	0.323077	-0.056923
Accuracy on Color Room Questions	0.348315	-0.025468
Accuracy on Color Questions	0.478261	-0.049275
Accuracy on Location Questions	0	-0.222222
↑ Kappa Score	0.014279	0.018946

4.2.3 Random Noise

4.2.3.1 Method

In this experiment, instead of black images, which would give a static, theoretically recognizable color, on evaluation the model is given images filled with random noise. Like in the first blindfolding experiment, this is a new case for the model, since it was trained on normal images from the dataset. Figure 4.8 shows an example.

Question: what color is the plant in the dining room ?
Prediction: olive green
Ground truth: green

Question: What color is the plant in the dining room?
Prediction: olive green
Ground Truth: green

Figure 4.8: Random Noise Example

4.2.3.2 Results

When evaluated on random noise, the accuracy is reduced on all question types, and the mean rank is also higher across the board, with the most significant change being to `location` types, where the mean rank has increased by around 8. These differences from baseline are greater in magnitude than the differences from baseline in evaluation with black images. This makes sense considering that this random noise is a fully new case for the model. Although a fully black image is new for the model in previous experiment, the model is still expecting dark areas, such as shadows, or even large sections of black, such as in the bad render shown in Fig. 3.4. CNNs are designed to find patterns, and although black is a simple pattern, it is still a pattern, unlike the randomness of the noise here.

Table 4.5: Evaluation Metrics for Model Evaluated on Random Noise

Metric	Value	Difference from Baseline
Checkpoint	2	-6
Loss	3.076424	0.872283
↓ Mean Rank	6.898974	2.546664
Mean Rank on Color Room Questions	5.51236	1.901124
Mean Rank on Color Questions	3.318841	0.626087
Mean Rank on Location Questions	18.32963	8.192593
↑ Overall Accuracy	0.211282	-0.168718
Accuracy on Color Room Questions	0.258427	-0.115356
Accuracy on Color Questions	0.194203	-0.333333
Accuracy on Location Questions	0	-0.222222
↑ Kappa Score	-0.004550	0.000117

4.2.4 Shuffled Frames from the Dataset

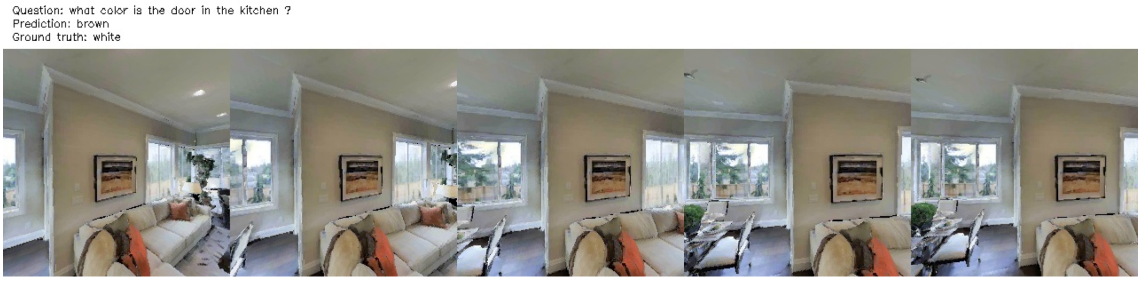
4.2.4.1 Method

In this experiment, the model is given images from the dataset, but not the correct images for the question. This gives the model visual structure, but still not actual views on the objects. This is done to give insight into whether the model is actually doing object recognition, or if it is learning other helpful patterns in the images. Work in multi-modal (vision and language) machine translation has found that multi-modal architectures are often insensitive to incongruent images [43].

This is done was by modifying the original dataset file, creating a new shuffled one. As can be seen in Appendix B, episodes have a `question`, which includes the question, answer, question type, and question and answer token IDs. These `questions` are shuffled between episodes. Since there are duplicate questions, with differing navigation paths, it is possible that a question could still have a valid set of images (meaning a set of images originally matched to the same question), but this possibility is low–for an individual question, the probability of a match is less than one percent. An example episode result can be seen in Fig. 4.9.

4.2.4.2 Results

Table 4.6 shows the lowest loss epoch of the model evaluated on shuffled scenes. Again, this disruption of the visual input results in higher mean ranks for all question types, and reduced accuracy on all categories except `location`. However, the decrease in performance is lower than the decrease in either the case of evaluation on black images or evaluation on random noise, supporting the idea that the model is using patterns in the images as support. [44] finds that models are not very good at



Question: what color is the door in the kitchen ?
Prediction: brown
Ground truth: white

Question: What color is the door in the kitchen?
Prediction: brown
Ground Truth: white

Figure 4.9: Example Shuffled Result

identifying conflicts between visual and linguistic information; models were bad at identifying incorrect captions. This suggests that the issue of relevant visual input extends beyond our model; the use of visual and linguistic information together is an area for future work.

Table 4.6: Evaluation Metrics for Model Evaluated on Shuffled Scenes

Metric	Value	Difference from Baseline
Checkpoint	10	2
Loss	2.393173	0.189032
↓ Mean Rank	5.144615	0.792305
Mean Rank on Color Room Questions	4.157303	0.546067
Mean Rank on Color Questions	3.034783	0.342029
Mean Rank on Location Questions	12.722222	2.585185
↑ Overall Accuracy	0.266154	-0.113846
Accuracy on Color Room Questions	0.264419	-0.109364
Accuracy on Color Questions	0.307246	-0.22029
Accuracy on Location Questions	0.222222	0
↑ Kappa Score	0.012831	0.017498

4.2.5 Summary

These experiments do show that the model is learning to use visual information. All methods of blindfolding or visual manipulation result in lower performance. Beyond this, these experiments also suggest that the model is not just using the visual information the way we as humans would expect, identifying objects and their properties, but also is using patterns to support its predictions in some other way. This is based on the fact that when given shuffled scenes from the dataset, the decrease in performance was lowest (around 11.5%), and when given random noise, the decrease was highest (around 17%).

These experiments also suggest that the visual information may not be used as much as we would hope, since when the model is trained on black images, the decrease in accuracy is only 5% (without the training the accuracy decreases by 13.5%). Since the model can tell you what color something is with only 5% less accuracy when it can't actually see it, the dataset appears to have a lot of exploitable biases.

`Location` questions generally appear to suffer less: there are likely two reasons for this. One, initial performance was so bad that there isn't as much room for decrease in performance. Another is that there are only 15 distinct question-answer pairs in the evaluation set. Of these, seven are also present in training, and two of these only have one possible answer in training. This may result in more exploitable language biases in this category than `color` or `color room`.

4.3 Experiment 3: Basic Semantic Categories

4.3.1 Method

In this section we examine if supplying the model with additional knowledge about objects in the scene can benefit its learning. Information fusion is an actively developing topic, focused on developing models that have and use more knowledge about the real world [45]. Alongside perceptual knowledge, language is a useful source of knowledge, since it is semantically dense [46]. In this experiment, we use ID-based encoding as a simple form of semantic information.

Schüz and Zarrieß found that models with prior knowledge about objects were able to make better predictions about those objects, and they found that an 'Early Fusion' strategy of integrating the object type information allowed the model to make better predictions about atypical colors of common objects [47]. Since one of the concerns with EQA models is that they are relying too heavily on common sense knowledge of objects, this is exactly what we would like to achieve–improving predictions about specific objects, which, if the prior knowledge about colors is not correct for the object, may be of atypical color. Based on this, we combine category information about objects in the scene with the visual features before attending with the question, as shown in Fig. 4.10. The categorical information comes from the 'semantic sensor' of the agent in `habitat-sim`, which reads annotations from the dataset. It reads annotations of object instances, which can then be mapped to category ids. The list of category IDs can be found in appendix A. This new input is a tensor with an integer category label for each pixel of the image.

The categories are broad, however there is overlap between the labelled categories and the items that questions ask about. For example, fireplaces are labelled (27), and the question 'What color is the fireplace in the living room?' occurs in the dataset. On the other hand, all appliances are labelled the same (37), but questions ask about specific appliances, such as 'What color is the oven?'

4.3.2 Results

Table 4.7 shows the epoch with the lowest loss for the model given semantic categories. The model achieves slightly lower mean ranks for all categories, but also around 1% lower accuracy for all question types except `location`, which has the same accuracy as baseline. This means that the model is ranking answers better, but the number one answer is wrong slightly more often. However, these differences are so small, that, especially with the kappa values being below zero, no strong conclusions can be drawn. From the improved mean ranks, we can see that the model is trying to fuse information together, and that very basic semantic category knowledge is helpful for ranking. A potential follow-up is discussed in 5.1.1.

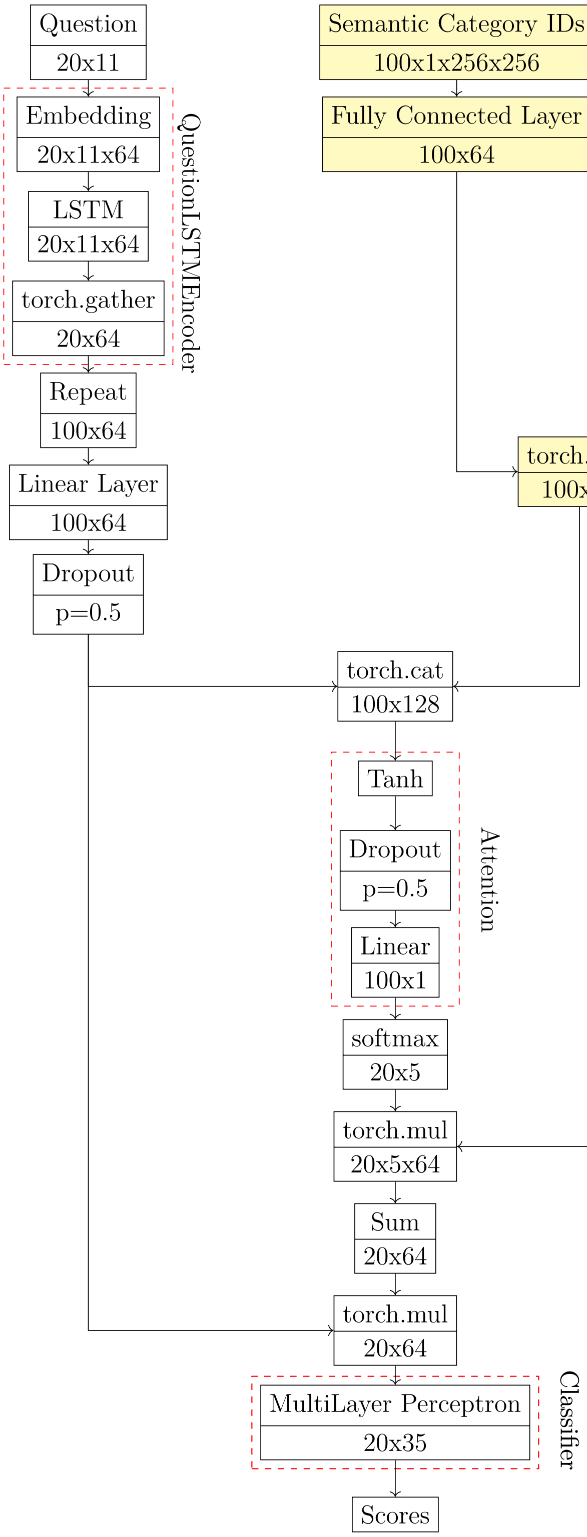


Figure 4.10: Model With Semantic Category IDs as 3rd input. Additions to baseline model in yellow.

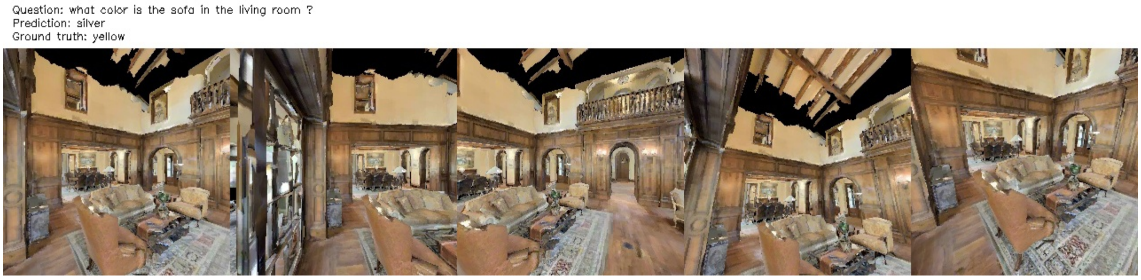
Table 4.7: Evaluation Metrics for Model with Semantic Categories

Metric	Value	Difference from Baseline
Checkpoint	6	-2
Loss	2.108628	-0.095513
↓ Mean Rank	3.985128	-0.367182
Mean Rank on Color Room Questions	3.388764	-0.222472
Mean Rank on Color Questions	2.368116	-0.324638
Mean Rank on Location Questions	9.0	-1.137037
↑ Overall Accuracy	0.371282	-0.008718
Accuracy on Color Room Questions	0.365543	-0.00824
Accuracy on Color Questions	0.510145	-0.017391
Accuracy on Location Questions	0.222222	0
↑ Kappa Score	-0.001182	0.003485

4.4 Experiment 4: Look Around

4.4.1 Method

In this experiment we attempt to increase the usefulness of the visual information being provided to the model. The VQA model takes in the last five frames of navigation as its visual input. However, as seen in Fig. 4.1, these images are very similar to each other, and often have odd angles of the object in consideration. This experiment implements a look around procedure, where the agent takes a series of moves (look left, look right, etc) at the end of navigation, so that the last five frames give more varied viewpoints of the room and object. The hypothesis here is that the model should, given the larger visual context, perform better on `location` questions. The 'frame queues' for each episode are generated before beginning training or evaluation, by saving the RGB (red-green-blue) observations of the agent at specified positions and rotations [1]. In the baseline model, this frame queue is those last five frames of navigation. The attention section of the model then chooses which frames to focus on when answering the questions–it weights the most important frames. For the look around experiment, this queue is: the final position and rotation of navigation, a frame from the same position turned 0.523599 radians (30°) to the left, same position turned 0.523599 radians (30°) to the right (from the original rotation), same position turned up 0.2617995 radians (15°) (from the original rotation), and the same position turned down 0.2617995 radians (15°) (from the original rotation). As can be seen in Fig. 4.11, this gives more variation in the final five frames.



Question: What color is the sofa in the living room?
Prediction: silver
Ground Truth: yellow

Figure 4.11: Look Around Example

Rotations are represented as quaternions which are a commonly used representation in game animation and similar applications [48]. These quaternions are given as `[x, y, z, w]`[2]. Although one of the benefits of quaternions is the ability to do rotation around all axes at once, for this experiment, only one axis is ever rotated around at a time. The equations below show the calculation for rotating around the `y` axis:

[1] These are given in a global coordinate frame.

[2] This corresponds to $w + x\mathbf{i} + y\mathbf{j} + z\mathbf{k}$, where $\mathbf{i}, \mathbf{j}, \mathbf{k}$ are unit vectors pointing along the three axes.

```
original_rotation = [x0, y0, z0, w0]
rotation = [0, 1, 0, rotation_amount]
new_quaternion = original_rotation * rotation
```

4.4.2 Results

Giving the model greater variety in views results in lower mean ranks, at the lowest loss epoch, for all question types, and higher accuracy for `color room` and `location` questions (as well as overall). `Color` questions have a decrease in accuracy. These differences are very small, however. It is possible a look around that gives viewpoints that vary more would provide more useful information for the model. A few possibilities for this are discussed in 5.1.1.

[49] found that the visual context is not a large factor in a model generating referring expressions. They suggest two potential conclusions from this: one, that humans also do not consider the context when generating their references, so the models don't need it, or two, that the problem is too complicated to model with simple representations of visual context. Our increase in performance could be due to giving the model more patterns to latch onto and use, even if they aren't being used in a way recognisable to a person. Another possibility is that our visual context is more complicated than that of [49], closer to that experienced by humans. In our case, `location` questions were most improved by the increased visual context. This could be due to the fact that locations are not identifiable 'objects', they are actually concepts built by multiple objects–yes, if you see a refrigerator, you are probably in a kitchen, but this dataset actually includes a refrigerator in a lounge, and to recognise this, you would need to notice other objects in the room, such as the couch. The conclusion here is that the context required to answer a question varies based on the task and question itself.

Table 4.8: Evaluation Metrics for Model using Look Around

Metric	Value	Difference from Baseline
Checkpoint	7	-1
Loss	2.140735	-0.063406
↓ Mean Rank	3.890256	-0.462054
Mean Rank on Color Room Questions	3.26367	-0.347566
Mean Rank on Color Questions	2.281159	-0.411595
Mean Rank on Location Questions	9.044444	-1.092593
↑ Overall Accuracy	0.405128	0.025128
Accuracy on Color Room Questions	0.40824	0.034457
Accuracy on Color Questions	0.492754	-0.034782
Accuracy on Location Questions	0.277778	0.055556
↑ Kappa Score	-0.009582	-0.004915

4.5 Experiment 5: Faster R-CNN

This experiment replaces the original CNN with an object detecting CNN, pretrained on a large image dataset.

4.5.1 Method

The object detecting CNN used is Faster R-CNN [50]. It uses a Region Proposal Network to predict object bounds and likelihoods, and then region of interest pooling to extract features for each bounding box. The version of Faster R-CNN used in this experiment is the one described in [51]. This version is used together with ResNet-101 [52]. The Faster R-CNN network is used to identify object bounds, regions exceeding a confidence threshold are identified, and then these regions are given to the ResNet-101 CNN to produce feature vectors representing the regions, giving a set of $V = \{v_1, ..., v_m\}$, where $v_m \in \mathbb{R}^{1 \times D}$, with $M = 36$ and $D = 2048$. (36 objects from the Faster R-CNN network are used.) The model also produces attributes and object labels, but these are not used in this experiment.

In this experiment, instead of using the last five frames of navigation as input, only the final frame was used. The original model runs the images through a CNN pretrained on the dataset, the CNN outputs features for the entire image, and the rest of the model uses these as input. In this experiment, the final frame was run through Faster R-CNN instead, and instead of features for the entire image, the set of individual object features (V) were used as input to the VQA model. The model can be seen in Fig. 4.12. This change to the architecture means that the attention portion, instead of weighting the five frames for relevance, weights the 36 objects from the single frame.

One consideration here was the summation layer. In the original model, this was used to combine the five frames, after the previous steps have determined the most relevant ones. In this situation, the frames likely have very similar features–they're not fully distinct. However, in this experiment, we don't have frames, we have objects, and these objects are distinct. Combining these distinct objects, after the relevant ones have been determined, could give a nice summary of the relevant objects in the scene, or could cause objects to be combined in ways that make them difficult to learn between scenes. Experiments both summing and not summing these weighted object feature vectors would be beneficial, but, due to time constraints, only the experiment with summation was done.

4.5.2 Results

This architecture improves some aspects of performance. As seen in Table 4.9, the model using FRCNN object features performs similarly to baseline in terms of overall mean rank, with decreased performance on `color room` and `color` questions, and increased on `location` questions. The model also does 15% better than baseline on `location` questions, but 5.9% and 10.7% worse than baseline on `color` and `color room` type questions respectively. This suggests that individual object features are

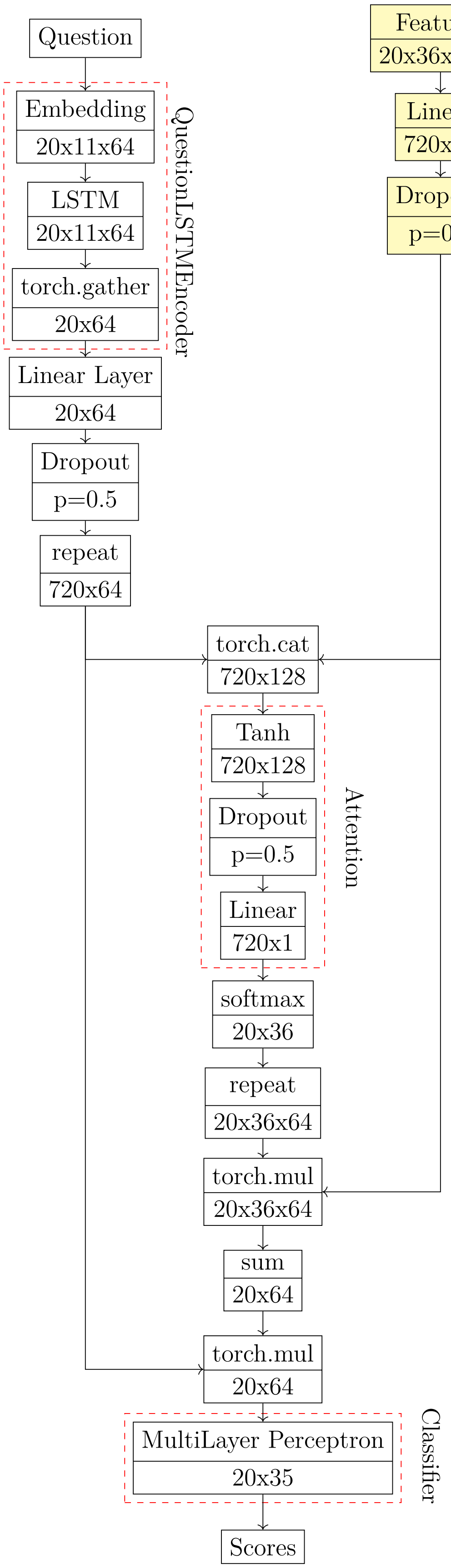


Figure 4.12: Model with Object Features
The parts highlighted in yellow are new to this model. They replace the original CNN and its input.

useful for determining location. The lowered performance on questions related to color may be due to the way that the model summarizes the objects in the scene. The kappa scores also show that the model's performance is more likely to be due to learning, rather than chance, compared to the baseline.

By identifying objects, we get a deeper semantic understanding of the scene. However, since we only use a single frame, we lose information from the previous frames. This would be a concern, but for this model, it does not seem to have a great impact. The original frames are all very similar, and even implementing the look around procedure to expand the visual context had a limited effect.

Table 4.9: Evaluation Metrics for Model using FRCNN Object Features as Input

Metric	Value	Difference from Baseline
Checkpoint	18	10
Loss	2.116223	-0.087918
↓ Overall Mean Rank	4.294872	-0.057438
Mean Rank on Color Room Questions	3.931086	0.31985
Mean Rank on Color Questions	3.681159	0.988405
Mean Rank on Location Questions	6.877778	-3.259259
↑ Overall Accuracy	0.341026	-0.038974
Accuracy on Color Room Questions	0.313858	-0.059925
Accuracy on Color Questions	0.42029	-0.107246
Accuracy on Location Questions	0.374074	0.151852
↑ Kappa Score	0.019255	0.023922

One point to mention here is that the original CNN was designed to be used for both question answering and navigation. Since this replacement CNN does not do depth estimation, it likely wouldn't work for navigation. Having two separate CNNs makes this model bulkier.

4.6 Study: Dataset Resampling

Since the model has varying accuracy by question type, as seen in Fig. 4.5 and the question types are not balanced in the dataset, with the `color room` type being significantly larger than the other two categories, we conduct an additional experiment in which the other two categories are made larger, to match the size of the `color room` question category. One note is that in previous experiments, `color` questions actually perform highest, although they are not the largest category. This may be due to them being a simpler question to answer, and they do share some possible answers with the `color room` questions, which may make the impact of the dataset imbalance lower on the `color` questions than the `location` questions.

Imbalanced datasets in classification problems often result in classifiers ignoring smaller classes. There are a number of strategies to mitigate this, and one of the simplest is sampling to adjust the dataset [53]. In this experiment, rather than sampling based on the classes (the answers), we sample based on the question type. This indirectly also affects the answer distribution, since, for example, 'brown' is not an answer for any location questions.

4.6.1 Method

The original training set contains 8031 `color_room` questions, 1830 `color` questions, and 1635 `location` questions. Using random sampling with replacement from the original dataset, the `color` and `location` types are bootstrapped to be contain 8031 questions as well[3]. The original baseline model is then trained on this larger dataset. The new model is then evaluated on the original evaluation set.

4.6.2 Results

Evaluation metrics for this experiment can be seen in Table 4.10. Boosting the smaller question types improves accuracy, but accuracy is reduced on `color` questions, and the overall boost is less than 1%. `Color room` and `location` improve 2.5% and 2.9% respectively. The positive kappa score does suggest that the model is now performing better than random sampling from the distribution. This suggests that bias has been reduced, which is interesting given that the category with reduced performance here is not the one that was over-represented in the original training set. This suggests that from the agent's perspective, potential biases are not as simple as one might assume. Another interesting note is that Mean Ranks all suffer. This, combined with the accuracy increase, means that the model is right more often, but when it is wrong, it is ranking the correct answers worse. The reason for this is unclear, but it may be due to shifting distribution patterns. From this experiment, we can see that adding more data per question type could significantly improve the model's learning.

[3] This was done using Imbalanced-learn's RandomOverSampler [54]

Table 4.10: Evaluation Metrics for Model trained on Boosted Dataset

Metric	Value	Difference from Baseline
Checkpoint	8	0
Loss	2.262098	0.057957
↓ Mean Rank	4.833846	0.481536
Mean Rank on Color Room Questions	4.248689	0.637453
Mean Rank on Color Questions	2.831884	0.13913
Mean Rank on Location Questions	10.285185	0.148148
↑ Overall Accuracy	0.388205	0.008205
Accuracy on Color Room Questions	0.399251	0.025468
Accuracy on Color Questions	0.452174	-0.075362
Accuracy on Location Questions	0.251852	0.02963
↑ Kappa Score	0.003186	0.007853

5

Conclusion

5.1 Discussion

The questions posed at the beginning of this report were:

- What are the limitations of the current EQA dataset (in regards to the VQA task)?
- Is the model using available information in the expected way?
  Is it identifying specific objects visually?
- Does a similar model given more information perform better?
  Information from the same context?
  Information from another context?

We answer the first question, **What are the limitations of the current EQA dataset (in regards to the VQA task)?**, via the examination of the dataset in 4.6, and in 3.2.2.1. We have found that the dataset is very unbalanced, both in terms of question types and answer distribution. Basic resampling is unable to significantly improve results, since the variety of questions is very limited. A resampling weighted by an estimate of question difficulty might have better results, but due to the limited number of questions in the dataset, this might not be worthwhile. The dataset also focuses on the navigation aspect of the EQA task, with questions repeated with different navigation paths. There are also some errors in annotation and issues with rendering quality.

The second question, **Is the model using available information in the expected way?**, is addressed in 4.2, where we deliberately confuse the model by manipulating the visual information given to it. We blindfold the model using two methods, giving it black images and images filled with random noise. As a third method of manipulation we change the view, so that the agent is being asked a question about a location other than its current viewpoint. We found that the model does rely on patterns in the visual information, and does seem to be relying somewhat on identification of specific objects, since the shuffling experiment does decrease accuracy. However, it appears that what is most important to the model is the presence of patterns in the visual input, since the random noise experiment results in the worst performance. The shuffling experiment has a comparatively lower magnitude decrease in accuracy to both the black and random noise blindfolding experiments. A contributing factor to this may be that even the correct final frames may not be very helpful viewpoints on the objects, since they are chosen automatically at the end of navigation. The blindfolding results suggest that the baseline model is learning to use information from the images in some way. However, these experiments do not address the fact that it may be possible to develop text only models that perform equally well to models with visual input, as suggested by [18]. Even within these methods of using the same model with blindfolding functions, the ideal learning parameters may vary based on how much vision is present.

The third question, **Does a similar model given more information perform better?**, is addressed by experiments 3, 4, and 5. The subquestion about **Information from the same context?** is addressed by all three. In Experiment 3, we provide additional information through object IDs from Habitat's semantic parser. Giving these labels results in lower accuracy, but also lower mean ranks. This suggests that the model is ranking answers better, but the number one answer has gotten slightly worse. However, the decreases in accuracy are very low, with a 0.87% decrease overall, so this decrease could just be due to chance. In comparison, the overall improvement in mean rank is 8.5%. This suggests that semantic category IDs are helpful, but not enough to overcome dataset bias. In Experiment 4, Look Around, we give a broader visual context (from the same scene). The agent is given new viewpoints from its final position, though these views are not guaranteed to show the object the question is about. This model improved performance both in terms of mean rank and accuracy. These results indicate that the broader view is beneficial for the model. In Experiment 5, we use Faster R-CNN object features from a single frame as input, instead of sending five frames through the original pretrained CNN. This model performs worse overall, but shows significant improvement (15% increase on accuracy) on location questions, suggesting that object level features are useful in determining location. Experiment 5 also attempts to address the subquestion about **Information from another context?**, since it uses transfer learning from large image datasets, but results here were inconclusive.

Model performance varies significantly from epoch to epoch, suggesting that the parameters used for this model, such as learning rate, are not ideal to have a reliable architecture. At the same time, keeping the model architecture as consistent as possible while adjusting input sources, may also contribute to lack of stability.

We have found that different question types benefit from different sources of information. For example, as mentioned above, object detections and object level features are helpful in answering `location` questions, as is more visual knowledge, as shown in 4.4. Generally, we can conclude that adding visual information, both in terms of granularity of represenation (4.5) and a broader perspective (4.4), is beneficial. At the same time, adding simple representations such as IDs of object categories in the scene (4.3) improves the ranking of correct answers, suggesting that the model can learn from more semantic information as well.

It is clear that there is room for continued work on the topic of EQA. The impact of question types on performance is visible, so datasets with greater variety in question types will be important. Consideration of how to balance performance on each question type will also be key–the current dataset strongly encourages focus on `color room` questions due to its imbalance. These results suggest that models synthesizing information from a variety of sources will perform better on this task.

5.1.1 Follow Up Experiments

As this was a series of connected experiments, there were of course other experiments that could have been conducted, but were excluded due to time constraints.

- One of these is an extension of the Semantic Categories experiment. In that experiment, categories were represented by integer IDs, rather than words. This would allow the model to learn topic-level classifications for questions: this question is about appliances, but would not give direct linking of the objects to the question. Using word embeddings as input, using the same embedding scheme as the questions, might provide more useful information for the model.

- A few potential follow-ups are related to the Look Around experiment. One potential experiment would be to add more rotations, using a larger frame queue as the input. Another would be to take frames from different points in the navigation. This would be particularly interesting given the dataset, since by the end of the navigation, the frames for the same question given different starting points are likely very similar, but frames sampled from other portions of the navigation should have lower similarity. A potential hypothesis is that pulling frames from earlier in the sequence could give better results on location questions, since more of the room containing the object should be seen.

- There is quite a lot of room for follow-up on the Faster R-CNN experiment. As well as examining the impact of summing the objects, Faster R-CNN produces a number of other outputs which could be used as input, including textual labels for the objects. One experiment would be to add these labels, potentially using word2vec to create embeddings, and modifying the portion of the model for question processing to also use word2vec, so that these labels could be matched to words in the questions.

- Finally, although for this thesis experiments were designed to be self-contained to better identify sources of improvement, combining aspects of all three experiments could be interesting and potentially improve performance.

- The model used for this project was chosen due to the ease of introduction of new inputs, but state-of-the-art models are more complex. Transformer based models such as [55] perform better on question answering tasks. Conducting similar experiments to the semantic category and look around experiments with a complex model could be useful, to see if these new sources of information can improve performance there as well.

5.2 Conclusion

We find that the EQA dataset has room for improvement. It would benefit from more balance in question types and possibly additional question types. It could also use some human correction, as there are a few annotation errors or confusing questions (a wardrobe in the closet?). We establish that the model does use visual information, as manipulation of the visual input impacts performance. However, we find that it is possible to train a model without visual information that achieves only slightly reduced performance. We also establish that providing more information can benefit the model, but the impact varies by question type. The Faster R-CNN experiment suggests that structured information is beneficial. Although the only category to improve in performance is location, the kappa score suggests that the implied structure provided by object detections improves learning. In these experiments, the model responds to both increased visual information and semantic information, and tries to fuse them. Since both types of information are valuable, a next step would be to combine increased visual information with semantic information, potentially actual linguistic information, rather than simply objects.

Bibliography

[1] I. Kruijff-Korbayová, F. Colas, M. Gianni, F. Pirri, J. de Greeff, K. Hindriks, M. Neerincx, P. Ögren, T. Svoboda, and R. Worst, "Tradr project: Long-term human-robot teaming for robot assisted disaster response," *KI - Künstliche Intelligenz*, vol. 29, no. 2, pp. 193–201, 2015.

[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 (CVPR)*, 2018.

[3] E. Wijmans, S. Datta, O. Maksymets, A. Das, G. Gkioxari, S. Lee, I. Essa, D. Parikh, and D. Batra, "Embodied Question Answering in Photorealistic Environments with Point Cloud Perception," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[4] J. R. Searle, "Minds, brains, and programs," *Behavioral and Brain Sciences*, vol. 3, no. 3, p. 417424, 1980.

[5] S. Harnad, "Grounding symbols in the analog world with neural nets," 1993.

[6] S. Harnad, "The symbol grounding problem," *Physica D: Nonlinear Phenomena*, vol. 42, no. 1, pp. 335–346, 1990.

[7] L. Stanislao, G. Bugmann, T. Kyriacou, and E. Klein, "Mobile robot programming using natural language," *Robotics and Autonomous Systems*, vol. 38, no. 3, pp. 171 – 181, 2002. Advances in Robot Skill Learning.

[8] K. M. Hermann, F. Hill, S. Green, F. Wang, R. Faulkner, H. Soyer, D. Szepesvari, W. M. Czarnecki, M. Jaderberg, D. Teplyashin, *et al.*, "Grounded language learning in a simulated 3d world," *arXiv preprint arXiv:1706.06551*, 2017.

[9] J. Thomason, A. Padmakumar, J. Sinapov, N. Walker, Y. Jiang, H. Yedidsion, J. Hart, P. Stone, and R. J. Mooney, "Improving grounded natural language understanding through human-robot dialog," in *2019 International Conference on Robotics and Automation (ICRA)*, pp. 6934–6941, 2019.

[10] R. Socher, A. Karpathy, Q. V. Le, C. D. Manning, and A. Y. Ng, "Grounded compositional semantics for finding and describing images with sentences," *Transactions of the Association for Computational Linguistics*, vol. 2, pp. 207–218, 2014.

[11] M. Abdou, A. Kulmizev, D. Hershcovich, S. Frank, E. Pavlick, and A. Søgaard, "Can language models encode perceptual structure without grounding? a case study in color," *arXiv preprint arXiv:2109.06129*, 2021.

[12] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. L. Zitnick, and D. Parikh, "Vqa: Visual question answering," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[13] Q. Wu, D. Teney, P. Wang, C. Shen, A. Dick, and A. van den Hengel, "Visual

question answering: A survey of methods and datasets," *Computer Vision and Image Understanding*, vol. 163, pp. 21–40, 2017. Language in Vision.

[14] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," in *ACL*, pp. 311–318, 2002.

[15] G. O. Francisco Bonin-Font, Alberto Ortiz, "Visual navigation for mobile robots: A survey," *Journal of Intelligent and Robotic Systems*, vol. 53, no. 3, 2008.

[16] D. Batra, A. Gokaslan, A. Kembhavi, O. Maksymets, R. Mottaghi, M. Savva, A. Toshev, and E. Wijmans, "Objectnav revisited: On evaluation of embodied agents navigating to objects," *arXiv preprint arXiv:2006.13171*, 2020.

[17] L. Yu, X. Chen, G. Gkioxari, M. Bansal, T. L. Berg, and D. Batra, "Multi-target embodied question answering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[18] A. Anand, E. Belilovsky, K. Kastner, H. Larochelle, and A. C. Courville, "Blindfold baselines for embodied QA," *CoRR*, vol. abs/1811.05013, 2018.

[19] Y. Goyal, T. Khot, D. Summers-Stay, D. Batra, and D. Parikh, "Making the v in vqa matter: Elevating the role of image understanding in visual question answering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[20] 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 (CVPR)*, June 2018.

[21] T. Tenbrink, R. J. Ross, K. E. Thomas, N. Dethlefs, and E. Andonova, "Route instructions in map-based human-human and human-computer dialogue: A comparative analysis," *Journal of Visual Languages and Computing*, vol. 21, no. 5, pp. 292–309, 2010.

[22] H. H. Clark and D. Wilkes-Gibbs, "Referring as a collaborative process," *Cognition*, vol. 22, no. 1, pp. 1–39, 1986.

[23] A. H. Anderson, M. Bader, E. G. Bard, E. Boyle, G. Doherty, S. Garrod, S. Isard, J. Kowtko, J. McAllister, J. Miller, C. Sotillo, H. S. Thompson, and R. Weinert, "The hcrc map task corpus," *Language and Speech*, vol. 34, no. 4, pp. 351–366, 1991.

[24] A. Das, S. Kottur, K. Gupta, A. Singh, D. Yadav, J. M. F. Moura, D. Parikh, and D. Batra, "Visual dialog," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[25] N. Ilinykh, S. Zarrieß, and D. Schlangen, "Meetup! A corpus of joint activity dialogues in a visual environment," *CoRR*, vol. abs/1907.05084, 2019.

[26] M. Hahn, J. Krantz, D. Batra, D. Parikh, J. M. Rehg, S. Lee, and P. Anderson, "Where are you? localization from embodied dialog," *arXiv preprint arXiv:2011.08277*, 2020.

[27] J. Thomason, M. Murray, M. Cakmak, and L. Zettlemoyer, "Vision-and-dialog navigation," *arXiv preprint arXiv:1907.04957*, 2019.

[28] S. Banerjee, J. Thomason, and J. J. Corso, "The RobotSlang Benchmark: Dialog-guided robot localization and navigation," *arXiv preprint arXiv:2010.12639*, 2020.

[29] Manolis Savva*, Abhishek Kadian*, Oleksandr Maksymets*, Y. Zhao, E. Wijmans, B. Jain, J. Straub, J. Liu, V. Koltun, J. Malik, D. Parikh, and D. Batra, "Habitat: A Platform for Embodied AI Research," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

[30] M. Savva, A. X. Chang, A. Dosovitskiy, T. Funkhouser, and V. Koltun, "Minos: Multimodal indoor simulator for navigation in complex environments," *arXiv preprint arXiv:1712.03931*, 2017.

[31] M. Deitke, W. Han, A. Herrasti, A. Kembhavi, E. Kolve, R. Mottaghi, J. Salvador, D. Schwenk, E. VanderBilt, M. Wallingford, L. Weihs, M. Yatskar, and A. Farhadi, "Robothor: An open simulation-to-real embodied ai platform," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[32] A. Kadian, J. Truong, A. Gokaslan, A. Clegg, E. Wijmans, S. Lee, M. Savva, S. Chernova, and D. Batra, "Sim2real predictivity: Does evaluation in simulation predict real-world performance?," *IEEE Robotics and Automation Letters*, vol. 5, p. 66706677, Oct 2020.

[33] A. X. Chang, A. Dai, T. A. Funkhouser, M. Halber, M. Nießner, M. Savva, S. Song, A. Zeng, and Y. Zhang, "Matterport3d: Learning from RGB-D data in indoor environments," *CoRR*, vol. abs/1709.06158, 2017.

[34] J. Straub, T. Whelan, L. Ma, Y. Chen, E. Wijmans, S. Green, J. J. Engel, R. Mur-Artal, C. Ren, S. Verma, *et al.*, "The replica dataset: A digital replica of indoor spaces," *arXiv preprint arXiv:1906.05797*, 2019.

[35] F. Xia, A. R. Zamir, Z. He, A. Sax, J. Malik, and S. Savarese, "Gibson env: Real-world perception for embodied agents," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[36] W. H. Orrick, "UAB "Planner5D" v. Facebook, Inc.."

[37] W. Monroe, R. X. Hawkins, N. D. Goodman, and C. Potts, "Colors in Context: A Pragmatic Neural Model for Grounded Language Understanding," *Transactions of the Association for Computational Linguistics*, vol. 5, pp. 325–338, 09 2017.

[38] K. L. Kelly, "Twenty-two colors of maximum contrast," *Color Engineering*, vol. 3, pp. 26–27, 1965.

[39] R. A. annd Massimo Poesio, "Technical report csm-437: Kappa3 = alpha (or beta)," tech. rep., University of Essex, September 2005.

[40] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[41] R. Artstein and M. Poesio, "Inter-Coder Agreement for Computational Linguistics," *Computational Linguistics*, vol. 34, pp. 555–596, 12 2008.

[42] J. R. Landis and G. G. Koch, "The measurement of observer agreement for categorical data," *Biometrics*, vol. 33, no. 1, pp. 159–174, 1977.

[43] D. Elliott, "Adversarial evaluation of multimodal machine translation," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2974–2978, 2018.

[44] R. Shekhar, S. Pezzelle, Y. Klimovich, A. Herbelot, M. Nabi, E. Sangineto, and R. Bernardi, "Foil it! find one mismatch between image and language caption," *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2017.

[45] N. Ilinykh and S. Dobnik, "When an image tells a story: The role of visual and semantic information for generating paragraph descriptions," in *Proceedings of the 13th International Conference on Natural Language Generation*, (Dublin, Ireland), pp. 338–348, Association for Computational Linguistics, Dec. 2020.

[46] K. Desai and J. Johnson, "Virtex: Learning visual representations from textual annotations," 2021.

[47] S. Schüz and S. Zarrieß, "Knowledge supports visual language grounding: A case study on colour terms," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, (Online), pp. 6536–6542, Association for Computational Linguistics, July 2020.

[48] K. Shoemake, "Animating rotation with quaternion curves," *SIGGRAPH Comput. Graph.*, vol. 19, p. 245254, July 1985.

[49] H. Viethen, R. Dale, and M. Guhe, "The impact of visual context on the content of referring expressions," in *Proceedings of the 13th European Workshop on Natural Language Generation*, pp. 44–52, 2011.

[50] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in Neural Information Processing Systems* (C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, eds.), vol. 28, Curran Associates, Inc., 2015.

[51] P. Anderson, X. He, C. Buehler, D. Teney, M. Johnson, S. Gould, and L. Zhang, "Bottom-up and top-down attention for image captioning and visual question answering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[52] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[53] G. Menardi and N. Torelli, "Training and assessing classification rules with imbalanced data," *Data mining and knowledge discovery*, vol. 28, no. 1, pp. 92–122, 2014.

[54] G. Lemaître, F. Nogueira, and C. K. Aridas, "Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning," *Journal of Machine Learning Research*, vol. 18, no. 17, pp. 1–5, 2017.

[55] H. Tan and M. Bansal, "Lxmert: Learning cross-modality encoder representations from transformers," *arXiv preprint arXiv:1908.07490*, 2019.

Appendices

A

Matterport3D Semantic Categories

id	category
0	void
1	wall
2	floor
3	chair
4	door
5	table
6	picture
7	cabinet
8	cushion
9	window
10	sofa
11	bed
12	curtain
13	chest of drawers
14	plant
15	sink
16	stairs
17	ceiling
18	toilet
19	stool
20	towel
21	mirror
22	TV monitor
23	shower
24	column
25	bathtub
26	counter
27	fireplace
28	lighting
29	beam
30	railing
31	shelving
32	blinds
33	gym equipment
34	seating
35	board panel
36	furniture
37	appliances
38	clothes
39	objects
40	misc
41	unlabelled

B

Example Episode from Task Dataset

The shortest path and viewpoint lists have been shortened for purposes of the example.

```
{'episode_id': '640',
 'scene_id': 'mp3d/5LpN3gDmAk7/5LpN3gDmAk7.glb',
 'start_position': [15.50573335967819, -0.7660300302505512, 8.392731789742543],
 'start_rotation': [-5.312086480921031e-17,
  -0.8526401643962381,
  -0.0,
  0.522498564647173],
 'info': {'bboxes': [{'type': 'object',
    'box': {'centroid': [13.2358, -14.5238, 0.497693],
     'a0': [1.0, 0.0, 0.0],
     'a1': [0.0, 1.0, 0.0],
     'a2': [0.0, 0.0, 1.0],
     'radii': [0.593273, 0.243441, 1.68627],
     'obj_id': 305,
     'level': 0,
     'room_id': 18},
    'name': 'door',
    'target': True},
   {'type': 'room',
    'box': {'centroid': [10.874245, -11.97072, 0.5380600000000001],
     'a0': [1.0, 0.0, 0.0],
     'a1': [0.0, 1.0, 0.0],
     'a2': [0.0, 0.0, 1.0],
     'radii': [3.1686549999999998, 3.26178, 1.95437],
     'room_id': 18,
     'level': 0},
    'name': ['kitchen'],
    'target': False}],
  'question_meta': [{'name': 'color', 'diffuse': 'grey'}],
  'question_answers_entropy': 0.8303560860446519,
  'level': 0},
 'goals': [{'position': [13.2358, 0.4976929999999973,
   ↪ 14.5238],
  'radius': 0.6412771421234348,
  'object_id': 305,
  'object_name': 'door',
  'object_category': 'object',
  'room_id': 18,
  'room_name': 'kitchen',
  'view_points': [{'position': [12.985883260576134,
     -1.246680130110505,
     14.494095338174798],
    'rotation': [-2.855981544936522e-28,
     -0.7071067811874078,
     -0.0,
     0.7071067811856873]},
       ...
   {'position': [13.089462756345679,
      ↪ -1.246680130110505, 13.976197859327065],
    'rotation': [-1.2227381688226952e-16,
     -0.8910065241891411,
     -0.0,
     0.45399049973802935]}]}],
'start_room': 'R22',
'shortest_paths': [[{'position': [15.50573335967819,
    -0.7660300302505512,
    8.392731789742543],
   'rotation': [-5.312086480921031e-17,
    -0.8526401643962381,
    -0.0,
    0.522498564647173],
   'action': 2},
       ...
  {'position': [13.042462387438766,
     ↪ -0.7660300302505512, 13.951177365325918],
   'rotation': [-1.2227381690007914e-16,
    -0.891006524222839,
    -0.0,
    0.45399049967190386],
   'action': 3}]],
'question': {'question_text': 'what color is the
   ↪ door in the kitchen?',
 'answer_text': 'grey',
 'question_tokens': [4, 5, 6, 7, 19, 9, 7, 10],
 'answer_token': [0, 0, 0, 0],
 'question_type': 'color_room'}}
```

C

Instances of Answers in Training Data

Answer	Counts
brown	2352
white	2143
silver	1110
black	1031
off-white	855
tan	825
kitchen	600
grey	525
green	255
blue	240
bedroom	225
living room	165
bathroom	150
slate grey	135
olive green	105
laundry room	105
family room	75
closet	75
lounge	60
red	60
yellow green	45
purple	45
spa	45
light blue	30
office	30
hallway	30
purple pink	30
dining room	30
red brown	30
tv room	30
orange yellow	15
foyer	15
yellow	15
yellow pink	15