Grounded, or a Good Guesser? A Per-Question Balanced Dataset to Separate Blind from Grounded Models for Embodied Question Answering

Miles Shelton[1] **Nate Wingerd**[1] **Kritim K. Rijal**[1] **Ayush Garg**[1]
Adelina Gutic[1] **Brett Barnes**[1] **Catherine Finegan-Dollak**[2]
University of Richmond Department of Computer Science
[1] {first.last}@richmond.edu
[2] Corresponding Author, cfinegan@richmond.edu

Abstract

Embodied question answering (EQA) means using *perception of* and *action in* an environment to answer natural language questions about that environment. However, previous work has demonstrated that blind language models (which do not incorporate perception, but predict an answer based solely on the question text) are a strong baseline for existing benchmarks, even compared against state-of-the-art vision and language models. To determine whether a model is grounding its answers in its specific environment, rather than relying on a language model's expectations about the world generally, we propose PQB-EQA, a *per-question balanced* EQA dataset. In this new benchmark, every question appears twice, paired with two different environments that yield two different answers. That is, the answer distribution is balanced for each question, not just across the whole dataset. We show both theoretically and empirically that grounding in the environment is necessary to perform better than chance on PQB-EQA.[1]

1 Introduction

Imagine a search and rescue robot that could answer the question, "What is behind the concrete slab?" by navigating to a location where it could find the answer. To answer the question correctly, the agent would need to (1) understand the natural language question, (2) perceive its environment (using, for instance, vision), and (3) select actions to find the necessary information; that is, it would need to engage in *embodied question answering* (EQA) (Das et al., 2018). Obviously, such an agent should ground its answers in the environment; we do not want the search and rescue robot to tell us what is *often* behind concrete slabs. There is therefore a need to test such grounding in EQA models.

Unfortunately, "blind" language models—that is, models that receive only the question text and no images—have been strong baselines for existing EQA benchmarks (Anand et al., 2018; Thomason et al., 2019; Ilinykh et al., 2022; Majumdar et al., 2024), showing that they do not require the model to use perception, let alone to act in its environment to find the answer. Performance on these benchmarks therefore does not tell us whether a multimodal language model is in fact grounding its answers in its environment or hallucinating based on patterns in language or datasets. Because EQA is meant to enable agents to answer questions *about an environment*, answers based solely on language priors are unreliable for real-world use (Thomason et al., 2019; Ilinykh et al., 2022); work in the related area of visual question answering (VQA) further supports this (Zhang et al., 2016; Goyal et al., 2017).

This paper introduces PQB-EQA, a new EQA benchmark with per-question balancing, a dataset design strategy to ensure that a model that does not perceive its environment cannot perform better than chance. Each example is a question-environment-answer tuple. Every question depends on its environment; for example, "Could you tell me if there are cobwebs on the houses to the southwest?" might be answered *yes* in reference to one environment and *no* in reference to another, as shown in Figure 1. PQB-EQA pairs each question with two different environments giving two different correct answers. Only a model that integrates language and perception can determine the correct answer for the given question-environment pair. In addition, human testers demonstrated that action in the environments is necessary to find the answers.

The contributions of this paper are the following:

- We construct the first per-question-balanced benchmark for EQA.
- We demonstrate that a state-of-the-art language model performs no better than chance

[1] Data and code available at https://milesshelton.github.io/pqb_eqa/

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 124–135
July 27 - August 1, 2025

(a) The agent turns to the southwest and quickly spots cobwebs on the sandstone house.

(b) The agent turns and walks into the house to look for cobwebs before determining there are none.

Figure 1: Screenshots collected from humans in two environments answering the question, "Could you tell me if there are cobwebs on the houses to the southwest? A. yes B. no."

on our new benchmark, while adding vision and a set of oracle actions greatly improves performance.

- We verify that humans need to take action in the environments to answer the questions.

2 Background: EQA Task Definition

The EQA task can be understood in terms of its inputs (an environment and a question about it) and outputs (an answer). An agent capable of certain actions is given a natural language question and placed in a partially-observable environment (as a fully-observable environment would obviate the need for actions to find the answer). The agent uses actions to explore the environment to find the answer. EQA specifically addresses questions *about the environment* (distinguishing it from text-only QA) and allows the agent to select actions to obtain information, distinguishing it from VQA.

More formally, we define the task as follows: a model, M, is given a partially-observable environment E, and a natural language question, Q, which is a sequence of words $w_1 ... w_n$ from the input vocabulary, V_i. The goal is to predict the correct answer Y conditioned on E. At each time step t, M receives an observation from E, which we call obs_t. M may choose an action, a_t, from an action space, A. The action at time t may affect the observation at time $t+1$. The goal remains to predict the answer Y in light of the environment E; action choice is a latent variable. Each example in a dataset is therefore a tuple, $\{Q, E, Y\}$.

3 Per-Question Balancing

Dataset balancing is a technique to prevent models that learn spurious correlations from appearing better than they are. If the answer "yes" shows up much more frequently than "no" in a dataset, a system that picks the majority answer will be a strong baseline without requiring language or vision skills. Some EQA datasets addressed this problem in their datasets by balancing their answers. For example, Tan et al. (2023) generated questions and sampled so that their final dataset contained precisely the same number of "Yes" answers and "No" answers, as well as equal numbers of counting questions answered 0, 1, 2, 3, or 4. This prevents a model from exploiting dataset-wide patterns in the answers.

However, balancing answers across the entire dataset is insufficient to prevent blind language models from falsely appearing to be effective. Consider an extreme example: Suppose a dataset had environments that always contained dogs but never contained cats. By sampling an equal number of questions that asked if there were dogs and questions that asked if there were cats, one could easily create a dataset that was perfectly balanced across answers but would not require a model to integrate language and vision, since language alone would be enough to answer the questions correctly.

Per-question dataset balancing, by contrast, addresses the blind language model problem by balancing the distribution of correct answers *for each question*. This means that every question must appear in multiple $\{Q, E, Y\}$ tuples; that is, for a given question q_i, there must exist two tuples, $\{q_i, e_i, y_i\}$ and $\{q_i, e_i', y_i'\}$, such that $e_i \neq e_i'$ and $y_i \neq y_i'$. Since every question has two distinct answers, recognizing the type of question—or indeed, the exact question asked—cannot be enough for a model to guess the correct answer. Although this technique has been used for static visual question answering (VQA) datasets (Hodosh and Hockenmaier, 2016; Zhang et al., 2016; Goyal et al., 2017), it has not previously been applied to EQA.

4 Dataset

We constructed a per-question balanced dataset comprised of 424 question-environment-answer tuples and recordings of human players.

4.1 Environments

Environments in the dataset are bounded sections of a world in the video game Minecraft. Minecraft has been a popular test bed for reinforcement learning (Guss et al., 2019; Johnson et al., 2016; Kiseleva et al., 2021; Baker et al., 2022; Fan et al., 2022; Wang et al., 2023) and grounded natural language processing (Kitaev and Klein, 2017; Szlam et al., 2019; Narayan-Chen et al., 2019; Srinet et al., 2020; Bonn et al., 2020; Jayannavar et al., 2020; Burns et al., 2021; Kiseleva et al., 2021; Shi et al., 2022), although it has not previously been used for EQA. Due to its highly engaged online community, an abundance of text, images, and videos related to the game are available for training (Fan et al., 2022).

There is a tradeoff between realism and variety of environments for EQA. Previous EQA datasets sought to use the most realistic possible visual simulations, which limited their environments to houses and offices (Gordon et al., 2018; Das et al., 2018; Ren et al., 2024; Majumdar et al., 2024). In contrast, using Minecraft as our simulation enables a wide range of environment types. Minecraft natively incorporates 64 different biome types, such as deserts and villages, and over 1,000 distinct items, from gold to mushroom stew. It also encourages imaginative construction, allowing new objects to be added. Since prior datasets for EQA have already provided realism, we contribute environments that enable greater variability.

Using the WorldEdit Minecraft mod,[2] we constructed 312 60-by-60 block, single-biome environments. These include city, town, cave, desert, mansion, nether, plains, and snow environments with variations in the items, structures, and creatures placed in each. All environments are compatible with the MineRL (Guss et al., 2019) v.1.0 framework for training RL agents, enabling EQA to benefit from existing action-selection modules.

4.2 Questions

To ensure truly natural language, all questions were collected from and curated by humans. For question collection, we recorded dialog between paid human participants playing cooperative mini-games where their scores depended on accurately gathering information. Details of the procedures are in Appendix A. We manually identified questions from the game dialogs that (a) depend on the environment (that is, the answer will be different in different environments), (b) are possible for the agent to answer without damaging the environment, and (c) do not require dialog context beyond the question. Some questions immediately met the criteria; others we modified to meet the criteria (for example, we replaced pronouns with their antecedents from the dialog context). See Appendix B for details of question curation.

4.3 Answers

To achieve per-question balancing, we matched each question to two environments that yielded different answers, using a mix of automation and manual review. Using information from the Minecraft save file about items in the environment, we can programmatically determine the answers to some questions; for instance, "are there pigs?" is a look-up to see if any entities in the environment are pigs. However, other questions are more complex; for instance, "are all rooms the exact same?" is hard to automate when the specification lists one structure as "house," not multiple structures as "rooms." We manually found two environments with different answers for these questions. We used the two answers for each question, each correct in a different environment, to construct a multiple choice question.

Answer types are varied. Approximately half are yes-or-no, but 30% of answers appear only once in the dataset. For examples of questions and answer choices, see Appendix C.

To ensure tuple quality, we played through the environments and answered the questions. If the reviewer's answer matched the annotated answer, we kept the tuple; otherwise, a second reviewer answered the questions, and we kept the majority label. After review, we removed all questions that had the same answer in both environments.

4.4 Human Play Data

We recorded human players navigating the environments to answer the questions during tuple review. Once per second, we saved a screenshot. Once per in-game tick (1/20$^{\text{th}}$ of a second), we logged all actions taken by the user during that tick. The action space includes actions like "forward," "jump", "left", "right", and "camera." Camera movement is controlled by mouse; the other actions are controlled with keypresses. Ongoing actions, such as forward movement that lasted more than a single tick, appear in each tick's log until they end.

[2] https://github.com/EngineHub/WorldEdit

Model	Acc.	p
Blind GPT4-o	0.507	$p = 0.8082$
GPT4-o with Vision	0.827	$p < 0.0001$

Table 1: Accuracy and p of the two models. The blind model does not significantly outperform chance, while the grounded model does.

Benchmark (Reported in)	Δ Perf.
EQA v1 (Ilinykh et al., 2022)	1.8
A-EQA (Majumdar et al., 2024)	6.3
PQB-EQA *(ours)*	**32.0**

Table 2: Reported difference in scores between models with and without vision on previous benchmarks and on PQB-EQA.

	Yes/No	Other
Blind GPT4-o	0.509	0.505
GPT4-o with Vision	0.796	0.861

Table 3: Results of each model on yes/no questions and all other types of questions. The model with vision outperforms the blind model by a wide margin on both categories of question.

5 Experiments

We conducted experiments to verify that (a) a blind model would *not* perform better than chance, and (b) a model that successfully navigated and perceived its environment *would* perform better than chance. Both elements are needed to ensure that the benchmark cannot be gamed by a blind model but will fairly evaluate truly grounded models.

Text-only GPT4-o was our blind model. We provided it with the questions and answer choices described in Sections 4.2 and 4.3 and instructed it to choose the best answer and guess if it is unsure. Prompt details are in Appendix D. This is a state-of-the-art large language model, but if our hypothesis is correct, it will not outperform chance, since it does not observe the environment.

To verify the dataset's feasibility for a model that perceives and acts in its environment, we provided sequences of oracle screenshots from the human play recordings to GPT4-o with vision.[3] We adjusted the prompt to refer to the images, but kept it otherwise the same as blind GPT4-o. If our hypothesis is correct, this model should achieve significantly better than random accuracy.

We evaluated accuracy and measured significance using a two-tailed binomial test.

6 Results and Discussion

Table 1 summarizes results. Blind GPT4-o achieved accuracy of 50.7%, which is not significantly different from chance ($p = 0.81$). GPT4-o with oracle screenshot sequences achieved accuracy of 82.7%, significantly outperforming chance ($p < 0.0001$). We therefore conclude that the dataset is feasible for grounded EQA models but not blind models.

Works comparing blind models against models with vision on previous benchmarks have observed notably smaller differences, as summarized in Table 2. Ilinykh et al. (2022)[4] reported that on EQA v1, their language-only model achieved accuracy of 36.2, while their vision + language model achieved accuracy of 38.0, a difference of 1.8 percentage points. Majumdar et al. (2024) reported that on their A-EQA dataset, blind GPT-4 achieved an LLM-Match score of 41.8, while GPT4-V achieved 35.5, a difference of 6.3 percentage points. On our dataset, the difference in accuracy is 32 percentage points.

To verify that the difference in performance holds across question types, we split our dataset into yes/no questions and all other types. Based on the results in Table 3, we conclude that the findings hold over different question types.

To verify that the dataset is suited to EQA and not simply static VQA, we analyzed the human action logs. On average, humans took 278.8 actions to answer a question. Less than 25% of recordings included under 50 actions, and only three did not require action. We therefore conclude that in order for an agent to succeed at this task when it is not given the oracle screenshots, it will need to integrate language, perception, *and* action.

7 Related work

In recent years, many EQA datasets have been released, beginning with Gordon et al. (2018)'s in-

[3]For these experiments, we use an oracle sequence of actions, as our goal is not to evaluate an EQA model, but rather to verify qualities of the benchmark.

[4]The reported model was not GPT-4, so some caution should be used in comparing their results to ours; however, running new experiments using GPT-4 on the EQA v1 benchmark is infeasible due to its dependence on currently unavailable SUNCG environments.

teractive question answering (IQA) and Das et al. (2018)'s EQA v1. Variations on the EQA task have focused on multi-target EQA (Yu et al., 2019), EQA with fine-grained robotic manipulation (Deng et al., 2021), EQA with a knowledge base (Tan et al., 2023), EQA with "situational" questions (Dorbala et al., 2024), EQA in a photo-realistic simulation (Ren et al., 2024), and "episodic memory" EQA (Majumdar et al., 2024).

Several EQA datasets are balanced in some way, but none are balanced per question. Das et al. (2018) excluded questions with low normalized entropy for the answer distribution; however, the resulting dataset was still susceptible to blind models (Anand et al., 2018; Thomason et al., 2019; Ilinykh et al., 2022). Yu et al. (2019) applied the same technique with a higher entropy threshold. Deng et al. (2021) manipulated the distribution of answers to be uneven in ways based on anticipated practical applications. Tan et al. (2023) balanced answers across the dataset by over-generating questions and answers and sampling to get equal numbers of "yes" and "no" answers and a uniform distribution over the numbers 0-4 for counting questions. As noted (Section 3), this form of balancing could leave more subtle patterns in the data that a blind model can still exploit. The closest to a per-question balanced dataset for EQA is IQUAD v1 (Gordon et al., 2018), which associated each question with multiple environments; however, an analysis of that dataset shows that only 5 out of 128 questions are equally likely to be true or false.

Previous work on per-question balancing (Hodosh and Hockenmaier, 2016; Zhang et al., 2016; Goyal et al., 2017) applied to visual question answering (VQA) for static images. It does not incorporate environments that agents may act in. Such environments present additional challenges; for example, an EQA agent may have to choose different actions to answer the same question in two different environments, as in Figure 1.

8 Conclusion

This work presented the first per-question balanced dataset for embodied question answering. Per-question balancing ensures that ungrounded models cannot perform well. We gathered natural language questions from human participants and aligned each one with two different environments yielding two different answers. Experiments with GPT-4o verify that even a state-of-the-art hyper-LLM performs no better than chance when using only language, and that the task would be feasible for a model that successfully navigated its environment to find the visual inputs needed to answer the questions. Thanks to its compatibility with MineRL, PQB-EQA is suited for experiments that test the integration of all three parts of the EQA task: language, perception, and action. Future work should evaluate models other than GPT-4 on this benchmark to provide a wider variety of baselines. In addition, we plan to evaluate using end-to-end EQA systems where an agent chooses the actions in place of the oracle action selection used here.

9 Limitations

Embodied question answering has three components: natural language understanding, perception, and action. Our evaluation uses oracle action sequences; future work should incorporate action selection and the perception generated from the selected actions.

Because the questions are multiple-choice, this dataset is not suitable for evaluating generation of answers. This limitation was necessary to ensure per-question answer balancing. Where generation of answers and ensuring that answers are grounded are both important, this dataset should be used in combination with other evaluations.

The dataset is only in English.

Environments are not photorealistic; however, we believe that the greater breadth of environments this enables balances this limitation.

The present dataset's size makes it suitable for evaluation, but it is not large enough to train new models.

10 Ethics

All EQA-based technology carries the risk of misuse for purposes such as inappropriate surveillance by bad actors. In addition, EQA systems that are used for ethical applications but provide inaccurate information may cause harm.

The questions in PQB-EQA were collected from paid human participants aged at least 18 years. Our data collection procedure was reviewed by our institution's IRB and considered to pose no more than minimal risk to participants. Participants were compensated $15 per hour, exceeding the local minimum wage. Informed consent was obtained from all participants. They were made aware that the data collected would be made publicly avail-

able for research. The nature of the data collection (asking questions about a Minecraft game) made disclosure of personally identifiable information unlikely; however, we manually reviewed the results and deleted the sole instance where a participant's name appeared in a question.

Acknowledgments

We thank the anonymous reviewers for their thoughtful feedback, which helped improve the paper. We are grateful to Laura Biester for her comments on early drafts and to Amerah Rutherford for her assistance with testing data collection instructions and performing quality assurance on code. Some of the authors were supported by University of Richmond Arts & Sciences Summer Fellowships. This work was also supported in part by a grant from the University of Richmond Undergraduate Research Committee.

References

Ankesh Anand, Eugene Belilovsky, Kyle Kastner, Hugo Larochelle, and Aaron Courville. 2018. Blindfold Baselines for Embodied QA. ArXiv:1811.05013 [cs].

Bowen Baker, Ilge Akkaya, Peter Zhokhov, Joost Huizinga, Jie Tang, Adrien Ecoffet, Brandon Houghton, Raul Sampedro, and Jeff Clune. 2022. Video PreTraining (VPT): Learning to Act by Watching Unlabeled Online Videos. ArXiv:2206.11795 [cs].

Steven Bird and Edward Loper. 2004. NLTK: The Natural Language Toolkit. In *Proceedings of the ACL Interactive Poster and Demonstration Sessions*, pages 214–217, Barcelona, Spain. Association for Computational Linguistics.

Julia Bonn, Martha Palmer, Zheng Cai, and Kristin Wright-Bettner. 2020. Spatial AMR: Expanded Spatial Annotation in the Context of a Grounded Minecraft Corpus. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4883–4892, Marseille, France. European Language Resources Association.

Kaylee Burns, Christopher D. Manning, and Li Fei-Fei. 2021. Neural Abstructions: Abstractions that Support Construction for Grounded Language Learning. ArXiv:2107.09285 [cs].

Abhishek Das, Samyak Datta, Georgia Gkioxari, Stefan Lee, Devi Parikh, and Dhruv Batra. 2018. Embodied question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–10.

Yuhong Deng, Di Guo, Xiaofeng Guo, Naifu Zhang, Huaping Liu, and Fuchun Sun. 2021. MQA: Answering the Question via Robotic Manipulation. In *Robotics: Science and Systems XVII*. Robotics: Science and Systems Foundation.

Vishnu Sashank Dorbala, Prasoon Goyal, Robinson Piramuthu, Michael Johnston, Dinesh Manocha, and Reza Ghanadhan. 2024. S-EQA: Tackling Situational Queries in Embodied Question Answering. ArXiv:2405.04732 [cs].

Linxi Fan, Guanzhi Wang, Yunfan Jiang, Ajay Mandlekar, Yuncong Yang, Haoyi Zhu, Andrew Tang, De-An Huang, Yuke Zhu, and Anima Anandkumar. 2022. MineDojo: Building Open-Ended Embodied Agents with Internet-Scale Knowledge. ArXiv:2206.08853 [cs] version: 2.

Daniel Gordon, Aniruddha Kembhavi, Mohammad Rastegari, Joseph Redmon, Dieter Fox, and Ali Farhadi. 2018. IQA: Visual Question Answering in Interactive Environments. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4089–4098, Salt Lake City, UT. IEEE.

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. 2017. 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)*, pages 6904–6913.

William H. Guss, Brandon Houghton, Nicholay Topin, Phillip Wang, Cayden Codel, Manuela Veloso, and Ruslan Salakhutdinov. 2019. MineRL: A Large-Scale Dataset of Minecraft Demonstrations. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, pages 2442–2448, Macao, China. International Joint Conferences on Artificial Intelligence Organization.

Micah Hodosh and Julia Hockenmaier. 2016. Focused Evaluation for Image Description with Binary Forced-Choice Tasks. In *Proceedings of the 5th Workshop on Vision and Language*, pages 19–28, Berlin, Germany. Association for Computational Linguistics.

Nikolai Ilinykh, Yasmeen Emampoor, and Simon Dobnik. 2022. Look and Answer the Question: On the Role of Vision in Embodied Question Answering. In *Proceedings of the 15th International Conference on Natural Language Generation*, pages 236–245, Waterville, Maine, USA and virtual meeting. Association for Computational Linguistics.

Prashant Jayannavar, Anjali Narayan-Chen, and Julia Hockenmaier. 2020. Learning to execute instructions in a Minecraft dialogue. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2589–2602, Online. Association for Computational Linguistics.

Matthew Johnson, Katja Hofmann, Tim Hutton, and David Bignell. 2016. The Malmo Platform for Artificial Intelligence Experimentation. In *Proceedings of*

the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI'16, pages 4246–4247. AAAI Press. Place: New York, New York, USA.

Julia Kiseleva, Ziming Li, Mohammad Aliannejadi, Shrestha Mohanty, Maartje ter Hoeve, Mikhail Burtsev, Alexey Skrynnik, Artem Zholus, Aleksandr Panov, Kavya Srinet, Arthur Szlam, Yuxuan Sun, Katja Hofmann, Michel Galley, and Ahmed Awadallah. 2021. NeurIPS 2021 Competition IGLU: Interactive Grounded Language Understanding in a Collaborative Environment. ArXiv:2110.06536 [cs].

Nikita Kitaev and Dan Klein. 2017. Where is Misty? Interpreting Spatial Descriptors by Modeling Regions in Space. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 157–166, Copenhagen, Denmark. Association for Computational Linguistics.

Arjun Majumdar, Anurag Ajay, Xiaohan Zhang, Pranav Putta, Sriram Yenamandra, Mikael Henaff, Sneha Silwal, Paul Mcvay, Oleksandr Maksymets, Sergio Arnaud, Karmesh Yadav, Qiyang Li, Ben Newman, Mohit Sharma, Vincent Berges, Shiqi Zhang, Pulkit Agrawal, Yonatan Bisk, Dhruv Batra, Mrinal Kalakrishnan, Franziska Meier, Chris Paxton, Sasha Sax, and Aravind Rajeswaran. 2024. OpenEQA: Embodied Question Answering in the Era of Foundation Models. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.

Anjali Narayan-Chen, Prashant Jayannavar, and Julia Hockenmaier. 2019. Collaborative Dialogue in Minecraft. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5405–5415, Florence, Italy. Association for Computational Linguistics.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2022. Robust Speech Recognition via Large-Scale Weak Supervision. ArXiv:2212.04356 [cs, eess].

Allen Z. Ren, Jaden Clark, Anushri Dixit, Masha Itkina, Anirudha Majumdar, and Dorsa Sadigh. 2024. Explore until Confident: Efficient Exploration for Embodied Question Answering. ArXiv:2403.15941 [cs].

Zhengxiang Shi, Yue Feng, and Aldo Lipani. 2022. Learning to Execute Actions or Ask Clarification Questions. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 2060–2070, Seattle, United States. Association for Computational Linguistics.

Kavya Srinet, Yacine Jernite, Jonathan Gray, and Arthur Szlam. 2020. CraftAssist Instruction Parsing: Semantic Parsing for a Voxel-World Assistant. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4693–4714, Online. Association for Computational Linguistics.

Arthur Szlam, Jonathan Gray, Kavya Srinet, Yacine Jernite, Armand Joulin, Gabriel Synnaeve, Douwe Kiela, Haonan Yu, Zhuoyuan Chen, Siddharth Goyal, Demi Guo, Danielle Rothermel, C. Lawrence Zitnick, and Jason Weston. 2019. Why Build an Assistant in Minecraft? ArXiv:1907.09273 [cs].

Sinan Tan, Mengmeng Ge, Di Guo, Huaping Liu, and Fuchun Sun. 2023. Knowledge-Based Embodied Question Answering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(10):11948–11960.

Jesse Thomason, Daniel Gordon, and Yonatan Bisk. 2019. Shifting the Baseline: Single Modality Performance on Visual Navigation & QA. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1977–1983, Minneapolis, Minnesota. Association for Computational Linguistics.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. 2023. Voyager: An Open-Ended Embodied Agent with Large Language Models. ArXiv:2305.16291 [cs].

Licheng Yu, Xinlei Chen, Georgia Gkioxari, Mohit Bansal, Tamara L. Berg, and Dhruv Batra. 2019. Multi-Target Embodied Question Answering. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6302–6311, Long Beach, CA, USA. IEEE.

Licheng Yu, Patrick Poirson, Shan Yang, Alexander C. Berg, and Tamara L. Berg. 2016. Modeling Context in Referring Expressions. In *Computer Vision – ECCV 2016*, Lecture Notes in Computer Science, pages 69–85, Cham. Springer International Publishing.

Peng Zhang, Yash Goyal, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. 2016. Yin and Yang: Balancing and Answering Binary Visual Questions. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5014–5022, Las Vegas, NV, USA. IEEE.

A Question Collection

We recruited[5] twelve pairs of self-identified experienced Minecraft players on a college campus in the United States to play mini-games in Minecraft. Participants were at least 18 years old and were compensated $15 in gift cards for a one-hour session, with the opportunity to win an additional $50 gift card for high scores. The prize incentivized

[5]All procedures were reviewed and approved according to the policies of the IRB at the university where the research was conducted. Informed consent was obtained from all participants. Recruitment materials included posters, emails, and an in-class announcement. Templates of consent forms, recruitment materials, and full instructions to the participants are available by request to the corresponding author.

Q:	What do you see?
A:	Grass fields, lots of trees, and flowers.
Q:	What's the flowers or colors?
A:	Red ones, yellow ones, pinkish ones, yellow and white ones.
	...
Q:	Do you see any animals?
A:	I see... Sheep...

Table 4: Dialog from a session of the can-you-do-it game where the questioner is trying to find out if it is possible to dye a sheep orange given what's available in the environment.

teams to work together to achieve their assigned tasks.

Player roles were questioner and agent. Games were designed so that a questioner needs to obtain information about an environment that only the agent can observe. The questioner could ask questions about the agent's environment but could not directly observe it. The agent could move freely around the environment and answer questions but not volunteer information.

(a) Initial screenshot from agent's environment

(b) Screenshot when agent tries to answer "Do you see any animals?"

Figure 2: Screenshots from the can-you-do-it game. The questioner knows the task is "dye a sheep orange," but the agent does not.

build	create	get	make	open	smelt
cook	destroy	grow	mine	place	trade
craft	dig	kill	obtain	put	

Table 5: Teams lost points for using these words that encouraged modification of the environment.

Mini-game 1: In the *can-you-do-it* game, the questioner is given a secret goal task and must determine whether it can be accomplished in the provided environment. To increase the variety of questions asked, we follow Yu et al. (2016) in providing a set of "taboo" words that the questioner may not say unless the agent says them first. For example, Table 4 and Figure 2 show dialog and screenshots from a team trying to determine whether the task "dye a sheep orange" can be completed in the given environment. Taboo words were *yellow*, *red*, *orange*, *sheep*, and *wool*. As an experienced Minecraft player, the questioner knew that orange dye could be made from red and yellow flowers. The agent does not know what the task is or what the taboo words are. The questioner needs the agent's help to gather information about the environment in order to correctly determine whether the materials necessary to dye a sheep orange can be obtained. The team gains points for correctly determining whether a task is or is not feasible and loses points for an incorrect answer. Points are also deducted if the questioner says any taboo words before the agent says them.

In our internal testing, we found that the Minecraft game mechanics make modifying the environment a valuable way of gathering information. However, models for active question answering should ideally limit the changes they make to their environment; a robot butler should not destroy the kitchen wall to determine how much milk is in the refrigerator. The game therefore applied a point penalty for modifying the environment. Points are also deducted if the questioner uses "doing words" from the set listed in Table 5; this discourages the questioner from disguising instructions as questions (e.g., asking "Can you mine three stone?" to instruct the agent to collect stone).

We recorded and transcribed spoken dialog for each session. Transcription was done in two stages: one automated pass using Whisper[6] (Radford et al., 2022), followed by manual corrections. We used

[6] https://github.com/openai/whisper, version 20231117

NLTK v.3.8.1 (Bird and Loper, 2004) to split the transcript into sentences and treated any sentence with a question mark as a question.

Mini-game 2: In the *spot-the-difference* game, the questioner and agent are placed in variations of the same environment. They earn points by identifying true differences and lose points for false positives (Figure 3). Due to the difficulties with transcription in the can-you-do-it game, we shifted to using a custom chat interface (Figure 4). The questioner's interface would only send messages that included at least four words and included a question mark. The agent's interface would only send messages that included at most ten words. Both interfaces required correct spelling before they would send a message. Both players could only see one question and answer at a time, in an effort to reduce the number of context-dependent questions. An admin could see the entire dialog and the questioner's guesses for how the two environments differed.

B Question Review

Data from both games required review to identify usable questions. A question is suitable for EQA only if the correct answer to the question depends on the environment; thus, general knowledge questions such as "What is a baby tree called?" are not EQA questions. In the current work, we ensure that each question can stand alone without dialog context; EQA in dialog is a more difficult problem that we leave for future work. We therefore annotated each question as (1) suitable for EQA, (2) suitable for EQA if context is provided, or (3) unsuitable for EQA. At least two annotators categorized each question. Where we found disagreements, an additional annotator broke the tie. Category 1 questions went directly into the dataset. Category 3 questions were discarded. One annotator reviewed the five previous dialog turns for all category 2 questions and rephrased them to contain context where possible (e.g., turning "What colors are they?" into "What colors are the flowers?"); the rephrased questions became part of the dataset. Questions that could not be rephrased based on the previous five turns were discarded.

C Example Data

Table 6 provides ten randomly selected multiple choice questions from PQB-EQA.

D GPT Settings

We used gpt-4o-2024-08-06, the default GPT4-o model as of this writing, with temperature set to 0.2 and max_tokens at 10 for all experiments.

For the blind condition, the prompt for system role was *Your goal is to answer multiple choice questions about a Minecraft environment. Each question has two possible answers, A and B. Your response should be a letter, either A or B. If you cannot determine the correct answer, guess.*

For the vision condition, the prompt for system role was *Your goal is to answer multiple choice questions based on Minecraft screenshots. Each image is a sequence of screenshots from a Minecraft session. Each question has two possible answers, A and B. Your response should be a letter, either A or B. If you cannot determine the correct answer, guess.*

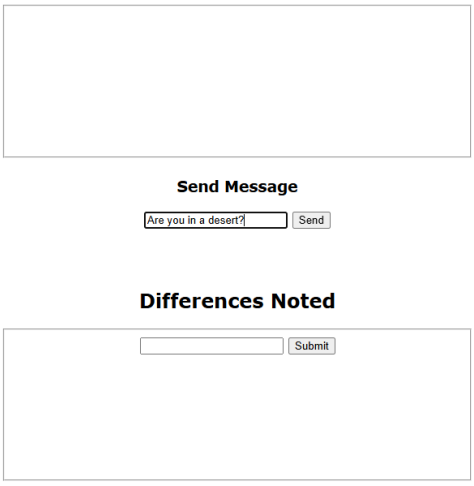
Both models returned text that began with either A or B for every question. Some included additional text after the letter. For the first 350 questions, we confirmed that the trailing text was the answer choice associated with that letter or the first several words of it. We therefore interpret the letter as the model's answer and discard any trailing text.

The cost for all experiments using GPT-4o models totaled $1.49 U.S.

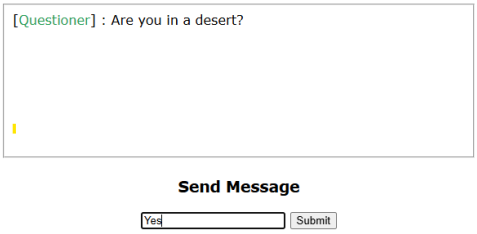
(a) Screenshot from questioner's environment

(b) Screenshot from agent's environment

Figure 3: Example of two environments from the spot-the-difference game. The team would earn points for noting that one environment includes a hay wagon and the other does not or that only one environment has cobwebs on the buildings, but would lose them if they said the environments were different biomes.



(a) Questioner chat interface for asking questions and noting differences



(b) Agent chat interface for answering questions

Figure 4: Chat interface for the spot-the-difference game.

what type of fence is around the garden? A. stone brick wall B. cobblestone wall
Can you access stone? A. yes B. no
when you are on the top of the higher side, how many cacti are there? A. 18 cacti B. 17 cacti
Okay, are there any mobs passive or non-passive in your environment? A. yes B. no
And there are no chests? A. incorrect B. correct
what types of trees are around you? A. there are warped trees B. there are none
do you have redstone? A. yes B. no
what does your environment look like, are there any structures? A. i see ice, blue ice, spruce leaves, stone, snow i also see a(n) house and a(n) mountain, and 4 rabbits, and 1 polar bear B. i see cut sandstone, sandstone, cactus, sand, smooth sandstone i also see a(n) house and a(n) cactus, and 5 rabbits
You don't see any stone at all in any chests or in the ground? A. incorrect B. correct
Is there any vegetation? A. there are warped trees, and crimson trees B. there are none

Table 6: Example questions and answers from PQB-EQA