

CODE-UPLOAD AI CHALLENGES ON EVALAI

A Thesis
Presented to
The Academic Faculty

By

Rishabh Jain

In Partial Fulfillment
of the Requirements for the Degree
Master of Science in Computer Science in the
College of Computing

Georgia Institute of Technology

May 2021

© Rishabh Jain 2021

CODE-UPLOAD AI CHALLENGES ON EVALAI

Approved by:

Dr. Dhruv Batra, Advisor
School of Interactive Computing
Georgia Institute of Technology

Dr. Devi Parikh
School of Interactive Computing
Georgia Institute of Technology

Dr. Stefan Lee
School of Electrical Engineering
and Computer Science
Oregon State University

Date Approved: April 28, 2021

The journey of a thousand miles begins with one step.

Lao Tzu

To my parents, my elder sister, my brother-in-law who set me on this road.

To Dr. Dhruv Batra and Dr. Devi Parikh for their unwavering support

ACKNOWLEDGMENTS

I have received a great deal of support and encouragement from a lot of people throughout this journey. I would like to take a moment to thank them.

First and foremost, I would like to thank my advisor Dhruv for believing in me and giving me an opportunity to work at Machine Learning and Perception Lab. My deepest gratitude goes to Dhruv for not only supporting and teaching me academically but also encouraging and providing me with several opportunities to present my work at different venues. He always motivated me to pick the problems of my interest along with providing support and guidance while navigating through those problems. I have learnt a lot of things from him which helped me flourish both personally and professionally ranging from the importance of discipline in life to thinking clearly about the motivation, users and approach of a problem. I am grateful to be his student.

I am glad and have also enjoyed the opportunity to work closely with Devi on several projects. Devi has taught me a lot of things over the course of the past 3 years such as attention to detail, clarity of thought and communication in ideas, time management, ability to remain stress free even during a deadline, and responsiveness. She has groomed me in solving problems from microscopic level to macroscopic level such as in drafting emails to writing long proposals and was both patient and helpful to me in ironing out my flaws. She always provided invaluable suggestions and solutions in the projects along with the freedom I needed to move on.

I am thankful to Stefan for the constant motivation and guidance he provided me during my internship at Georgia Tech. His ability to help people in any situation is something I picked up from him. He also possesses marvelous writing skills. I have learnt quite a few things from him while working on a paper and I always look up to him for improving mine. I also enjoyed the conversations with him outside of work on our lab retreats. He also taught me a few board games such as exploding kittens during the 2019 CVPR conference.

Thank you Dhruv, Devi and Stefan for serving on my thesis committee and steering me in the right direction over the past few years. Due to these pillars, I was blessed with funding at Georgia Tech both during internship and masters'. Moreover, my experience at the CVMLP lab has been nothing short of amazing due to the friendly culture and learning environment in the lab. There are innumerable lessons I have learnt from them along the way that I will always be grateful for.

I have been given a unique opportunity to lead and represent an open-source organization, CloudCV, while working in the lab. Working on CloudCV with Harsh and Deshraj was both fun and exciting. Harsh is an amazing mentor, friend and a perfect flatmate one can desire for. I owe special thanks to him for helping me with my masters' SOP. I can't thank Deshraj enough for hand holding me during the early days of the EvalAI/CloudCV project. He always kept me motivated and helped me to handle difficult situations during my internship in 2018. Apart from work, I have always enjoyed playing cricket, FIFA, racquetball and conversations with both of them. I have found them to be the people whom I can always reach out for discussing ideas, career advice and guidance. I would also like to thank our impressive team of CloudCV, GSoC and GCI students and mentors, comprising of Akash, Taranjeet, Shiv, Kartik, Sanket, Vipin, Sanjeev, Adarsh, Kajol, Khalid, Ayush, Gautam, Ram, Utsav, Mayank, Shekhar, Mayank, Rishabh B., Prem, and Deepesh. Working with them was exciting as we tried to solve a bunch of problems for the AI community. Their motivation, dedication, and hard work led us to achieve various milestones.

I have been fortunate enough to work with extremely talented peers during my grad school. I would like to thank each and every member of CVMLP lab during my time encompassing Harsh Agrawal, Prithvijit Chattopadhyay, Ramprasaath Selvaraju, Jiasen Lu, Deshraj Yadav, Abhishek Das, Viraj Prabhu, Erik Wijmans, Aishwarya Agrawal, Yash Goyal, Nirbhay Modhe, Arjun Majumdar, Joanne Truong, Arjun Chandrasekaran, Akrit Mohapatra, Jianwei Yang, Ayush Shrivastava, Mohit Sharma, Vishvak Murahari, Samyak Datta, Karan Desai, Michael Cogswell, Satwik Kottur, Ramakrishna Vedantam, Yash Kant,

Abhinav Moudgil, Ram Ramrakhya, Ashwin Kalyan, Purva Tendulkar, Stefan Lee, Peter Anderson, and Zhile Ren. Special thanks to - Ayush for being my flatmate and supporting me during good and bad times in the last 3 years, Harsh for helping me to learn FIFA, to take me to six-flags and other bunch of activities, Ramprasath for making me feel like home when I first arrived in US, cooking amazing and delicious food for us, Deshraj for teaching me racquetball, Prithvijit for entertaining friday nights, Arjun for all discussions regarding work-life balance, Abhishek for the challenging 8 ball matches, Karan for introducing me to creating memes and emojis, Mohit, Purva, Stefan, Peter, Zhile, Ashwin, Vishwak, Yash, Abhinav, Satwik, Akrit for the amazing time during conferences, lab retreats, and to Jiasen, Jianwei, Samyak, Erik, Arjun, Joanne, Aishwarya, Yash, Rama, Nirbhay for providing stimulating discussions and happy distractions to rest my mind outside of work.

I am privileged to have friends outside of work who were always there during my good and bad times. A big shoutout to Shivani for always being available to me and listening to all my things patiently, Saloni for helping me understand life, Shubhi for sending me gifts on every birthday, Sahil for supporting me as brother, Awinash for inspiring me with his hard work, Shivangi, Shivanika, Nivedita for making me feel like home everytime I called them, Divyansh, Shantanu, Shivang, Saurav, Pankaj, Karamveer, Shaury for being another family to me since undergrad.

I also place on record, my sense of gratitude to one and all, who directly or indirectly, have lent their hand in this venture.

I finish with Rampur, (my hometown in India), where the most basic source of my life energy resides: my family. I have an amazing family, unique in many ways. My parents - Ajit Kumar Jain and Sarita Jain; my sister, brother-in-law - Amrita Jain and Abhishek Jain; whose support has been unconditional all these years; they have given up many things for me; they have cherished with me every great moment and supported me whenever I needed it.

Finally, I doubt that, without such a team behind me, I would be in this place today.

TABLE OF CONTENTS

Acknowledgments	v
List of Tables	xiii
List of Figures	xiv
List of Acronyms	xvi
Summary	1
Chapter 1: Introduction	2
1.1 Motivation	2
1.1.1 Challenges in current AI agent evaluation	3
1.1.2 Infrastructure challenges in evaluating agents' code	3
1.1.3 Reproducibility of results	4
1.1.4 Maintaining evaluation consistency	4
1.1.5 Measuring constant progress	5
1.2 Proposed System	5
1.3 Contributions	6
1.4 Related Publications	6
1.5 Thesis Outline	7

Chapter 2: Related Work	9
2.1 OpenML	9
2.2 CodaLab	9
2.3 AICrowd	10
2.4 Kaggle	10
Chapter 3: EvalAI Overview	11
3.1 Users	11
3.1.1 Challenge Organizers	11
3.1.2 Admins	11
3.1.3 Participants	12
3.2 AI Challenges	12
3.2.1 Prediction Upload AI Challenges	12
3.2.2 Code Upload AI Challenges	12
3.3 Submissions	13
3.3.1 Using UI	13
3.3.2 Using EvalAI Command Line Interface (EvalAI-CLI)	13
3.4 Evaluation	14
3.4.1 Using automatic metrics	14
3.4.2 Using Remote Evaluation	14
3.5 Enhancing the Automated Metrics Evaluation Infrastructure	14
3.5.1 AWS Fargate	15
Chapter 4: AI Agents' Code Evaluation on EvalAI	17

4.1	On Reinforcement Learning Tasks	17
4.2	System Architecture	18
4.3	Components	18
4.3.1	EvalAI-CLI	18
4.3.2	EvalAI Backend	18
4.3.3	Amazon Simple Queue Service (AWS SQS)	19
4.3.4	AWS Fargate	19
4.3.5	Amazon Elastic Kubernetes Service (AWS EKS)	19
4.3.6	Kubernetes [12]	19
4.3.7	Amazon Elastic Kubernetes Service (AWS EKS) [13] Worker Nodes	19
4.3.8	Agent Docker Container	20
4.3.9	Environment Docker Container	20
4.3.10	Google Remote Procedure Call (gRPC) [14]	20
4.3.11	Pod	20
4.3.12	Job	20
4.3.13	Container Networking Interface [15]	20
4.3.14	Fluentd [16]	21
4.3.15	AWS CloudWatch [17]	21
4.3.16	Amazon Elastic Container Repository (AWS ECR)	21
4.3.17	Amazon Virtual Private Cloud (AWS VPC)	21
4.4	Working	22
4.4.1	Setting up environment container	22
4.4.2	Evaluation using environment container	22

4.5	Features Offered	23
4.6	Evaluation Setups and Case Studies	23
4.6.1	Using our evaluation infrastructure	24
4.7	On Supervised and Unsupervised Learning Tasks	27
Chapter 5: Hosting a Code Upload Challenge on EvalAI		29
5.1	Simplifying the AI challenge creation using github	29
5.2	Uploading Submissions and Evaluation	30
5.3	Analyzing and Viewing the Agents' Performance	30
5.4	Downloading and Running the Agents in the Real World	30
Chapter 6: Challenge Analytics and EvalAI Hosting		32
6.1	Analytics Dashboards for the Users of the Platform	32
6.1.1	EvalAI Admins	32
6.1.2	Challenge Organizers	32
6.1.3	Challenge Participants	33
6.2	Easy Hosting of EvalAI on Private Servers	33
Chapter 7: Impact		34
7.1	AI Community	34
7.2	Open-Source and Google Summer of Code (GSoC)	35
Chapter 8: Future Work		37
8.1	Multi-agent Evaluation	37
8.2	Evaluation of AI agents on dynamic datasets	37

Chapter 9: Conclusion	39
Appendices	41
Appendix A: EvalAI: Towards Better Evaluation of AI Agents	42
Appendix B: Evaluating visual and text explanations in an interactive, goal-driven human-AI task	48
References	73

LIST OF TABLES

A.1	Head-to-head comparison of capabilities between existing platforms and EvalAI	44
A.2	EvalAI growth statistics	47
B.1	Mean fraction (in %) of high salience regions ($\mu_{I>\tau}$) and mean spread of high salience regions ($S_{I>\tau}$) in Grad-CAM heat maps for each game played by a subject. The error terms are the 95% confidence intervals around the mean (1.96 *std. error). The number of games belonging to each category are given in the third column (total number of games = 620). Details regarding $\mu_{I>\tau}$, $S_{I>\tau}$ are described in subsection B.5.3.	66

LIST OF FIGURES

1.1	Need for Code-Upload AI Challenges	4
3.1	Prediction Upload AI Challenges	12
3.2	Code Upload AI Challenges	13
3.3	Enhancing the Automated Metrics Evaluation Infrastructure	15
4.1	Agent Environment System	18
4.2	Code Upload Challenge Evaluation Infrastructure	22
4.3	AI Habitat Challenge Evaluation on EvalAI	25
4.4	Code Upload Remote Challenge Evaluation	26
4.5	AI agent Evaluation for Static Code Upload Challenges	27
5.1	Challenge Creation Using GitHub	29
5.2	Sim2Real Gibson challenge Real World Phase Leaderboard	31
5.3	Sim2Real Gibson challenge Simulation Phase Leaderboard	31
6.1	Public Docker Images for EvalAI	33
7.1	Impact in AI Community	34
7.2	Google Summer of Code (GSoC) Participation From 2017 - 2020	36

A.1	EvalAI is a platform to evaluate AI agents in dynamic environment with human-in-the-loop.	43
A.2	System architecture for code upload challenges	45
B.1	(a) The pool of images in the game among which the human subject attempts to identify the secret image (green border). The subject asks ‘Is someone playing tennis?’. The AI answers the question (‘yes’) for the secret image. The subject guesses the secret image that is most consistent with the question and answer. (b) Grad-CAM visual explanation for the <i>secret image</i> is overlaid on all images in the pool. The heat-maps highlight regions in the image that contribute most to the AI’s prediction. (c) Text explanation that reasons about the answer for the secret image is provided with all images. The human subject guesses the secret image that is most consistent with the question, answer and explanation.	49
B.2	Mean performance of the human-AI team at the goal-driven task before (orange) and after (blue) human subjects gain access to explanations. The improvement in performance is the per-game difference averaged across 20 games for each subject. The mean improvement across subjects is presented (green). Error bars denote the 95% confidence interval around the mean ($1.96 \times \text{std. error}$).	58
B.3	(a) Sample interaction between human (given the full pool which is not shown) and AI. (b) Image that the subject wrongly guessed as secret image based on interaction. (c) Subject’s correct guess following the text explanation.	67
B.4	Screenshot of game interface with text explanations. The subject’s question given the initial pool is, ‘what objects are in the image?’, to which the model’s response is ‘trees’. The text explanation ‘they are shaped like wings’ is relevant to the secret image (likely referring to the airplane) but do not appear relevant to the question or answer.	68

LIST OF ACRONYMS

API	Application Programming Interface
AWS	Amazon Web Services
AWS EC2	Amazon Elastic Cloud Compute
AWS ECR	Amazon Elastic Container Repository
AWS ECS	Amazon Elastic Container Service
AWS EFS	Amazon Elastic File System
AWS EKS	Amazon Elastic Kubernetes Service
AWS IAM	Amazon Identity and Access Management
AWS RDS	Amazon Relational Database Service
AWS S3	Amazon Simple Storage Service
AWS SQS	Amazon Simple Queue Service
AWS VPC	Amazon Virtual Private Cloud
EvalAI-CLI	EvalAI Command Line Interface
gRPC	Google Remote Procedure Call
GSoC	Google Summer of Code

SUMMARY

Artificial intelligence develops techniques and systems whose performance must be evaluated on a regular basis in order to certify and foster progress in the discipline. We have developed several tools such as EvalAI which helps us in evaluating the performance of these systems and to push the frontiers of machine learning and artificial intelligence. Initially, the AI community focussed on simple and traditional methods of evaluating these systems in the form of prediction upload challenges but with the advent of deep learning, larger datasets, and complex AI agents, etc. these methods are not sufficient for evaluation. A technique to evaluate these AI agents is by uploading their code, running it on the sequestered test dataset, and reporting the results on the leaderboard. In this work, we introduced code upload evaluation of AI agents on EvalAI for all kinds of AI tasks, i.e. reinforcement learning, supervised learning and unsupervised learning. We offer features such as scalable backend, prioritized submission evaluation, secure test environment, and running AI agents code in isolated sanitized environment. The end-to-end pipeline is extremely flexible, modular and portable which can later be extended to multi-agents setups and evaluation on dynamic datasets. We also proposed a procedure using github for AI challenge creation to version, maintain, and reduce the friction in this conglomerate process. Finally, we focused on providing several analytics to all the users of the platform along with easing the hosting of EvalAI on private servers as an internal evaluation platform.

CHAPTER 1

INTRODUCTION

Artificial Intelligence keeps on being an inexorably necessary segment of our lives, regardless of whether we are applying the methods to research or businesses. The applications are becoming more scalable and adaptable, but much more complicated and volatile, as more and better tools, more processing resources, and the use of more dynamic and large data sources are integrated. As a result, there is a growing need for a deeper understanding of their capabilities and shortcomings, as well as safety issues. Theoretical methods can yield useful information, but only through evaluation of AI agents' code we can get a more detailed picture of how a machine performs in a variety of tasks or environments. So, we have developed a centralized platform which can evaluate machine learning models or agents' acting in an unseen dynamic environment individually or against each other.

1.1 Motivation

Deep Learning models have made groundbreaking progress in AI and this is possible due to the availability of large datasets, and powerful neural models. These models have wide variety of applications ranging from voice-activated assistants, self-driving cars, to search and rescue robots, etc., but before these large scale models are enhanced and deployed in the wild, they are evaluated on a sequestered test dataset. There are a few platforms which provide the evaluation of AI agents' on hidden test-dataset such as Kaggle, AICrowd, ParlAI, etc. but they suffer from limitations such as public test dataset, dataset biases, etc. Our approach addresses several limitations of the existing platforms and also provides the mechanism for evaluating AI agents' by simply uploading their code to run it in real-time on static or dynamic datasets.

1.1.1 Challenges in current AI agent evaluation

Public test datasets

Most of the current AI tasks require the test datasets to be public so that participants can evaluate their model on it and submit the predictions from the model which gets compared with the ground truth annotations on the evaluation server. In order to get a higher accuracy, a subset of public test datasets can be labelled and used to train the model, which defeats the purpose of the test dataset. Also, there might be unintended overlap between train and test sets which can be easily determined by looking at the test sets. For instance, in the VQA v1 dataset, the answer to “how much” or “how many” questions is usually 2. Moreover, having a public dataset also creates issues in terms of privacy of the test dataset.

Evaluation on new test dataset

Researchers improve the training and testing datasets over a period of time. If a task doesn't support uploading of AI agents' for evaluation, then it becomes almost impossible to compare and contrast the performance of the current state-of-the-art agent on the newly released test dataset with the old test dataset.

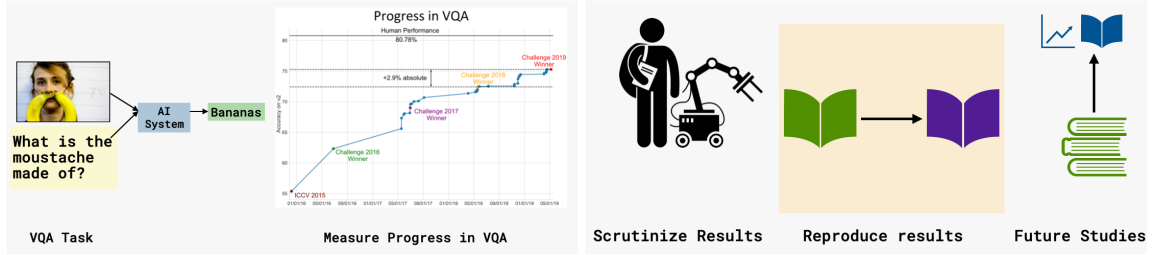
Identifying failure modes of the agent

Several studies have shown that playing with an AI agent in the form of an interactive demo enables researchers to identify the failure modes. It is impossible to create these demos in the current evaluation system which require uploading predictions from the agents'.

1.1.2 Infrastructure challenges in evaluating agents' code

One of the major bottlenecks in evaluating agents' code on the servers arises from the fact that these models and tasks are quite complex. Due to the complexity, they are very large in size and require a huge compute power for running. Moreover, setting up such large

servers is also cost heavy which is one of the major roadblocks for setting up code upload evaluation of AI agents’.



Code Upload AI Challenges

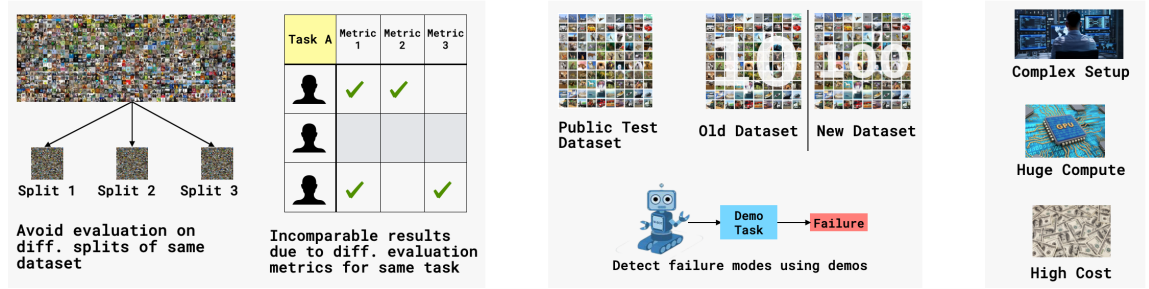


Figure 1.1: The code-upload challenges help in measuring constant progress on AI tasks, reproducibility of results, maintaining evaluation consistency, and solves the shortcomings and infrastructural challenges in the current evaluation system on EvalAI.

1.1.3 Reproducibility of results

Scientific progress depends upon the ability of independent researchers to scrutinize the results of a research study, reproduce the study’s main results using its materials, and build upon them in future studies. To reproduce the results from an AI model, we need to upload the code of the model in a specified format and run it on the test dataset.

1.1.4 Maintaining evaluation consistency

Large corpus of data is extremely useful for training AI agents’ but to provide a more realistic setting, we need to evaluate these agents’ against the hidden test dataset or in simulation environments so that they can generalize to the real world. We want to avoid evaluation of these models on the different splits of the same dataset which leads to inconsistent compar-

ison and also abstain the use of different evaluation metrics for the same task which leads to incomparable results.

1.1.5 Measuring constant progress

In order to develop any new capability in AI, we are able to engineer narrow and task-specific agents' because only within a very narrow and grounded context we can define our goal precisely. For instance, in a task such as VQA where we are given an image and a natural language question - "What is the moustache made of ?", the model has to predict a natural language correct answer - "Bananas". So, to measure constant progress on different AI tasks, we need to evaluate these agents'.

1.2 Proposed System

To address the aforementioned problems we propose to simplify and standardize the evaluation of AI agents' in static and dynamic datasets on EvalAI. We developed EvalAI [1], a couple of years back as a highly-extensible open-source platform which fulfills the critical need in the AI community for evaluation of machine learning models on static datasets. In this thesis, we improve the current evaluation methodology of AI agent on EvalAI. Concretely, we have developed an end-to-end system which will enable students, researchers, and data-scientists to upload the AI agents' on the platform, which will run on our scalable infrastructure and the results will be shown on leaderboard.

While evaluation of specialized AI agents' can be restricted to the task they were designed to perform, evaluation of more general abilities and adaptation requires testing across a large range of tasks. To be helpful in the development of general AI systems, we should not just evaluate performance on specific narrow tasks, but also facilitate the measurement of knowledge acquisition, cognitive growth, lifelong learning, and transfer learning. We propose an easy modular composition and scaling of agent-environments system for this purpose, where a wide range of task evaluations with variations can quickly be

constructed, administered, and compared.

1.3 Contributions

The main contribution of this thesis is code upload AI challenges on EvalAI with two settings; using our evaluation infrastructure and our servers, using challenge organizers infrastructure and challenge organizers servers; which address the desiderata of maintaining privacy of the data, automatic scaling of the infrastructure, and reducing the cost of evaluation of these AI agents’.

We also reduced friction in AI challenge creation by introducing challenge creation using GitHub. As the number of users and scale of the platform is growing, we have also automated the evaluation of prediction upload challenges and code upload challenges. We added several analytics dashboards for all the users of the platform and introduced effortless hosting of EvalAI on private servers for industrial organizations.

1.4 Related Publications

- EvalAI: Towards Better Evaluation Systems for AI agents’ [1]

Deshraj Yadav, Rishabh Jain, Harsh Agrawal, Prithvijit Chattopadhyay, Taranjeet Singh, Akash Jain, Shiv Baran Singh, Stefan Lee, Dhruv Batra

Workshop on AI Systems, SOSP 2019

- Dialog without Dialog: Learning Image-Discriminative Dialog Policies from Single-Shot Question Answering Data [2]

Michael Cogswell*, Jiasen Lu*, Rishabh Jain, Stefan Lee, Dhruv Batra, Devi Parikh

Advances in Neural Information Processing Systems 33, NeurIPS 2020

- nocaps: novel object captioning at scale [3]

Harsh Agrawal*, Karan Desai*, Yufei Wang, Xinlei Chen, Rishabh Jain, Mark Johnson, Dhruv Batra, Devi Parikh, Stefan Lee, Peter Anderson

IEEE/CVF International Conference on Computer Vision (ICCV), 2019

- Evaluating visual and text explanations in an interactive, goal-driven human-AI task

Arjun Chandrasekaran, Rishabh Jain, Karan Desai, Kerry Moffitt, Jeff Miller, David Diller, Bill Ferguson, Devi Parikh

1.5 Thesis Outline

The structure of the remainder of this work is as follows:

- In chapter 2, we compare our proposed approach with the existing platforms such as OpenML, CodaLab, AICrowd, Kaggle.
- In chapter 3, we give an overview of the existing EvalAI system, types of users, supported challenges, submissions, evaluations, and the enhancements in the automated metrics evaluation.
- In chapter 4, we describe our approach with key components, working, and features to standardize the AI agents' code evaluation on reinforcement learning tasks and supervised and unsupervised learning tasks along with case studies for each evaluation method.
- In chapter 5, we outline the method for hosting a code upload challenge on EvalAI describing the method of AI challenge creation using github.
- In chapter 6, we discuss the challenge analytics dashboard for the users of EvalAI and hosting of EvalAI on private servers.

- In chapter 7, we describe the impact of EvalAI on AI community, open-source and GSoC.
- In chapter 8, we talk about the future work on evaluating multi-agent systems using our current architecture. It also describes the evaluation of AI agents' on dynamic datasets.
- In chapter 9, we conclude the main contributions of this thesis.
- In Appendix A, We introduce EvalAI, an open source platform for evaluating artificial intelligence algorithms (AI) at scale. EvalAI is built to provide a scalable solution to the research community to fulfill the critical need of evaluating ML models and AI agents in a dynamic environment against ground-truth annotations or by interacting with a human. This will help researchers, students, and data scientists to create, collaborate, and participate in AI challenges organized around the globe.
- In Appendix B, We present our work on evaluating existing approaches that generate task-agnostic interpretable visual [4] and text explanations [5] for decisions from a deep neural network via metrics on static datasets or simple human judgements. In this work, we evaluate visual and text explanations in the context of an interactive, goal-driven, collaborative human-AI task, GuessWhich [6]. We make the following observations - (1) The performance of AI models on a metric from a static dataset is not well correlated with performance on the interactive task with lay human subjects. (2) In the interactive task, we find that subjects achieve higher performance when visual and text explanations are made available, (3) Overall, subjects obtain more useful information from text explanations than visual explanations, raising interesting open questions.

CHAPTER 2

RELATED WORK

Different general-purpose benchmarks and platforms have recently been launched, and they are rapidly being used to drive and measure development in AI research and hosting competitions.

2.1 OpenML

OpenML [7] is an online platform where researchers can automatically log and share machine learning data sets, code, and experiments. As a system, OpenML allows people to organize their experiments online, and build directly on top of the work of others. By readily integrating the online platform with several machine learning tools, large-scale collaboration in real-time is enabled, allowing researchers to share their very latest results while keeping track of their impact and use. While the focus of OpenML is on experiments and datasets, our solution focuses more on the end result - models, their predictions and subsequent evaluation.

2.2 CodaLab

CodaLab [8] is an open-source alternative to EvalAI which offers an environment for performing computational research that is more powerful, reproducible, and collaborative. Worksheets and AI challenges are two facets of CodaLab. Worksheets enable users to catch dynamic analysis pipelines in a repeatable manner, resulting in "executable papers." Users can catch the testing pipeline in an immutable manner by archiving the code, documents, and effects of an experiment. Through codaLab AI competitions are somewhat close to EvalAI in terms of features, it does not help testing interactive agents in various environ-

ments with or without humans in the process. As the AI community incorporates more complicated tasks, such as evaluating an agent within a simulation or running an agent on a real robot, a fully customizable backend like ours becomes increasingly relevant.

2.3 AICrowd

AICrowd [9] is another online platform for enabling data-science experts and enthusiasts to collaboratively solve real-world problems by hosting AI challenges. It also allows its users to host reinforcement learning challenges and static dataset challenges. Although, it also hosts code upload challenges but it binds the users to use Gitlab platform for submitting code whereas EvalAI doesn't have such requirements. Moreover, the submission on EvalAI is in the form of a docker image rather than a configuration file which generates a docker image because the dependencies might break over the period of time while creating the docker image from the configuration. Moreover, they also don't provide access to the docker images created from the code to the challenge organizers which can be used to deploy the model on the real robots to compare the performance in simulation and real world.

2.4 Kaggle

Kaggle [10] is one of the most common competition sites for data science and machine learning. It uses a cloud-based workbench that is functionally comparable to IPython notebooks to enable users to share their methodology with other data scientists. Despite its success, Kaggle [10] has a number of drawbacks. For starters, as a closed-source platform, and also unable to support code upload challenges. Moreover, it only supports challenge creation using templates which creates a limitation of maintaining challenge template versions if multiple versions of the same task is hosted over the years on the same platform.

CHAPTER 3

EVALAI OVERVIEW

EvalAI [1] is an open source platform for evaluating and comparing machine learning (ML) and artificial intelligence algorithms (AI) at scale. EvalAI is built to provide a scalable solution to the research community to fulfill the critical need of evaluating machine learning models and agents acting in an environment against annotations or with a human-in-the-loop. This will help researchers, students, and data scientists to create, collaborate, and participate in AI challenges organized around the globe. EvalAI also seeks to lower the barrier to entry for participating in the global scientific effort to push the frontiers of machine learning and artificial intelligence, thereby increasing the rate of measurable progress in this domain. The code is available [here](#).

3.1 Users

EvalAI has three kinds of users on the platform i.e challenge organizers, admins and participants.

3.1.1 Challenge Organizers

They are responsible for hosting and managing a challenge on the platform. More specifically, they create challenges using the challenge configuration templates or using github. Setting up submission evaluation using the evaluation script along with viewing and downloading the challenge analytics are some of responsibilities of challenge organizers.

3.1.2 Admins

EvalAI admins are in charge of helping challenge organizers to setup the challenges, view and approve the challenges on the platform.

3.1.3 Participants

Participants participate and compete in the challenges, create and download submissions on the platform.

3.2 AI Challenges

EvalAI supports hosting prediction upload challenges. We will describe hosting code upload challenges on EvalAI in chapter 5.

3.2.1 Prediction Upload AI Challenges

These challenges are set up for static datasets such as VQA, Visual Dialog, TextVQA, etc where the test dataset is public but the ground truth labels are private and stored on EvalAI. Participants run their models on the test datasets and submit the predictions file on EvalAI which gets compared to the ground truth labels in real-time and the accuracies are reported on the leaderboard as shown in Figure 3.1.

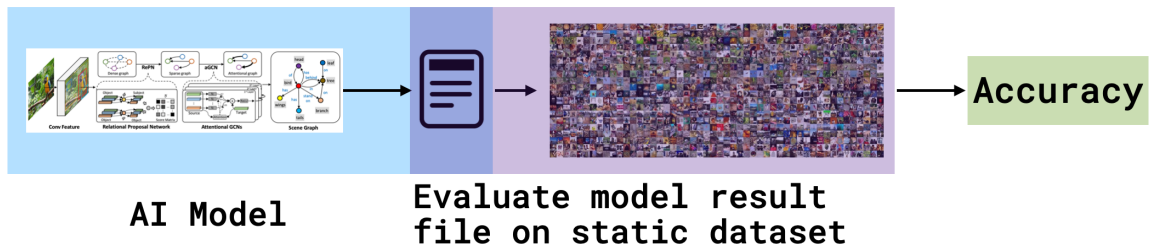


Figure 3.1: Prediction Upload AI Challenges

3.2.2 Code Upload AI Challenges

In this setup as shown in Figure 3.2, participants are asked to upload the AI model's code in the form of a docker image on the platform. The submission is then run in the simulation of a real world setting to evaluate the model. The models can later be downloaded and run in the real world to study the differences in the real world from simulation.

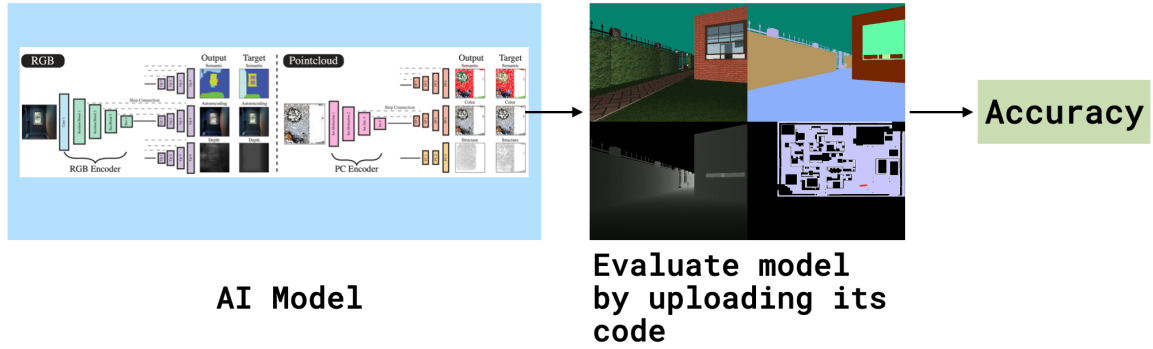


Figure 3.2: Code Upload AI Challenges

3.3 Submissions

Submissions are one of the most critical components on EvalAI. Submissions on EvalAI can be made through the UI, using the EvalAI-CLI tool or using the public submission file URL.

3.3.1 Using UI

It is the traditional way of creating submissions on any platform where a user simply uploads a predictions file on the platform.

3.3.2 Using EvalAI-CLI

We have built a command line tool known as EvalAI-CLI which can be used to push large submissions greater than 400 MB in size on the platform. Moreover, for uploading an agents' code, users have to submit it using this method. EvalAI-CLI provides flexibility to participants in automating the submissions on the platform while training an agent. Participants can also use it to view their submission status as well as for viewing the leaderboards of the challenge.

3.4 Evaluation

We allow challenge organizers to provide an implementation of their metric which is subsequently used to evaluate all submissions ensuring consistency in evaluation protocols.

3.4.1 Using automatic metrics

When a challenge is set up, challenge organizers can create an evaluation script in any language which is used to evaluate all the submissions. As soon as a submission occurs on the platform for a challenge, the challenge evaluation docker container picks that submission from the challenge queue, then runs it against the stored ground truth annotations and calculates the metrics defined in the evaluation script. Once the evaluation is completed, the results are displayed on the leaderboard.

3.4.2 Using Remote Evaluation

While hosting an AI challenge, organizers are concerned about the privacy of the test dataset and they don't want to share the data even with the challenge organizing platform members. Moreover, certain large-scale challenges require special compute capabilities for evaluation. EvalAI provides a unique solution for hosting these challenges, maintaining leaderboard while the actual evaluation of the submissions happen on challenge organizers servers without sharing the test dataset. Challenge organizers can poll EvalAI API's for the submission's data, evaluate it on their servers and then update the results in the database.

3.5 Enhancing the Automated Metrics Evaluation Infrastructure

A good distribution of challenges hosted on EvalAI uses the automated metrics evaluation of the tasks. With the increasing popularity of the platform, there has been a rise in the number of challenges to be hosted on EvalAI. In order to match the increasing demand and limited number of EvalAI admins, there is a need to automate the deployment of challenge

evaluation workers on EvalAI. EvalAI architecture in [1] suggests that whenever a challenge is created, an admin has to manually deploy the challenge evaluation worker on the AWS EC2 instance. We took the current evaluation infrastructure and automated it using AWS Fargate.

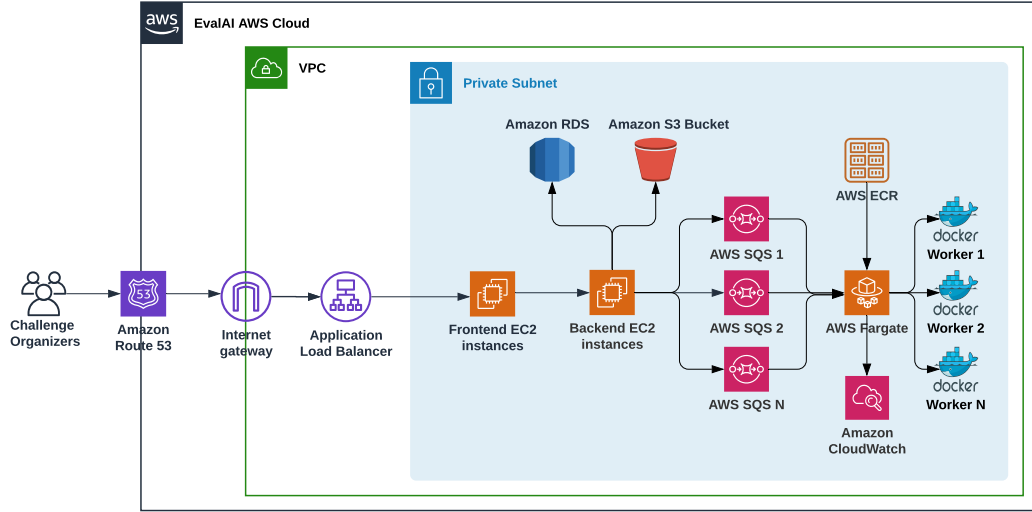


Figure 3.3: System architecture for enhancing the automated metrics evaluation infrastructure. The core pieces of the system are: (1) The frontend Amazon Elastic Cloud Compute (AWS EC2) instances which receives the request from the challenge organizer, (2) The backend AWS EC2 instances which frontend communicates with to store the data in the Amazon Relational Database Service (AWS RDS) and Amazon Simple Storage Service (AWS S3) bucket, (3) The Amazon Simple Queue Service (AWS SQS) queues which gets configured for each challenge, and (4) The AWS Fargate which pulls the evaluation worker docker image from Amazon Elastic Container Repository (AWS ECR) and deploys the docker containers. It also sends the docker container logs to Amazon CloudWatch.

3.5.1 AWS Fargate

AWS Fargate is a technology that one can use with Amazon Elastic Container Service (AWS ECS) to run docker containers without having to manage servers or clusters of AWS EC2 instances. With Fargate, one no longer has to provision, configure, or scale clusters of virtual machines to run containers. This removes the need to choose server types, decide when to scale your clusters, or optimize cluster packing.

In the modified infrastructure, when a challenge is created, we pull the evaluation worker docker container from the AWS ECR storage. We package the ground truth annotations along with a few variables in the docker container and deploy it on the AWS Fargate. In order to remove EvalAI admin from the loop of challenge creation we also display the container logs on the UI so that organizers can make changes to the evaluation script until it successfully evaluates a submission. With AWS Fargate, we don't have to worry about the load on the system as it auto scales with the number of submissions waiting to be processed.

CHAPTER 4

AI AGENTS' CODE EVALUATION ON EVALAI

With the advent of new and complex problems in the AI community along with more robust models, the method to evaluate and compare the existing agents with the new ones require running agents' code on the test environment or static dataset concealed behind an evaluation server. We propose a novel approach to evaluate the AI agents on reinforcement learning tasks and on supervised and unsupervised learning tasks.

4.1 On Reinforcement Learning Tasks

Reinforcement learning is the training of machine learning models to make a sequence of decisions. The agent learns to achieve a goal in an uncertain, potentially complex environment. In reinforcement learning, an agent faces a game-like situation or completes a particular task. To get the machine to do what the programmer wants, the agent gets either rewards or penalties for the actions it performs to maximize the total reward. Formally, the learner or the decision maker is called the agent and the thing it interacts with, comprising everything outside the agent, is called the environment. These interact continually, the agent selecting actions and the environment responding to those actions and presenting new situations to the agent. The environment also gives rise to rewards, special numerical values that the agent tries to maximize over time as shown in Figure 4.1

Since these agents have to take decisions in an interactive environment, so to compare and evaluate them it is required to run these agents on the hidden test-environments. We propose a simplified novel system architecture to evaluate these agents in a single or multi-agent setup. The current setup can also be extended to dynamic environments where the test-environment is changing based on the decisions taken by the agent during evaluation.

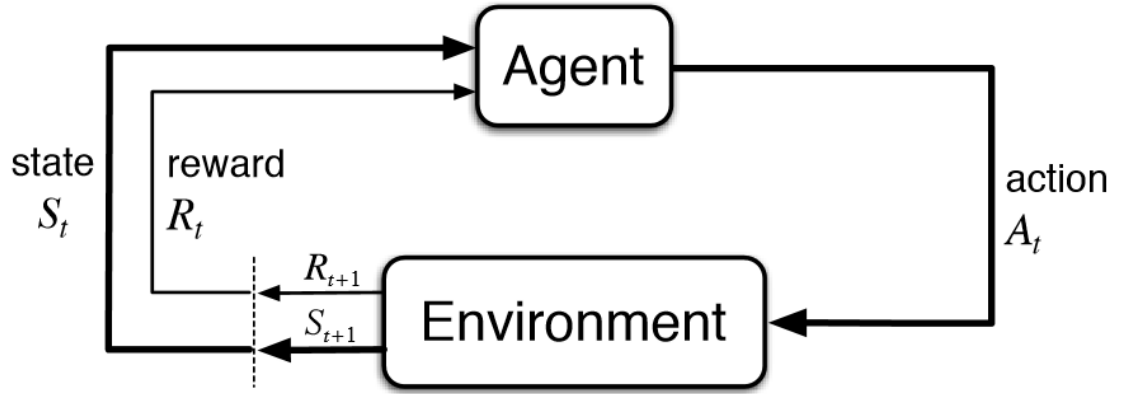


Figure 4.1: AI agent and environment basic configuration

4.2 System Architecture

The end-to-end evaluation system is built using the open-source technologies and it can be deployed on any cloud service with minimal changes. The various components of the evaluation pipeline along with their working are described as follows.

4.3 Components

4.3.1 EvalAI-CLI

It is the command line tool which is used to push the agents' code in the form of docker containers on EvalAI servers.

4.3.2 EvalAI Backend

It consists of Django [11] based REST APIs to interact with the database, pushing the submissions to AWS SQS queue, updating the status and result of the submissions in the database.

4.3.3 Amazon Simple Queue Service (AWS SQS)

AWS SQS is a fully managed message queuing service that enables us to decouple and scale microservices, distributed systems, and serverless applications. SQS eliminates the complexity and overhead associated with managing and operating message oriented middleware, and empowers developers to focus on differentiating work. Using SQS, we can send, store, and receive messages between software components at any volume, without losing messages or requiring other services to be available.

4.3.4 AWS Fargate

It is the service where the evaluation workers are deployed. The job of the challenge evaluation worker is to pick a message from the AWS SQS queue and deploy it on Amazon Elastic Kubernetes Service (AWS EKS) service in the form of a job.

4.3.5 Amazon Elastic Kubernetes Service (AWS EKS)

AWS EKS is a managed service that we use to run kubernetes [12] on Amazon Web Services (AWS) without needing to install, operate, and maintain our own kubernetes control plane or nodes.

4.3.6 Kubernetes [12]

It is an open-source system for automating the deployment, scaling, and management of containerized applications.

4.3.7 Amazon Elastic Kubernetes Service (AWS EKS) [13] Worker Nodes

The AWS EKS worker nodes are the managed AWS EC2 instances where the agents' code is run with the test environment in the form of a pod.

4.3.8 Agent Docker Container

It is the dockerized AI agents' code uploaded by the users which needs to be evaluated.

4.3.9 Environment Docker Container

It is the docker container which contains the test environment and also provides high-level Application Programming Interface (API) for the agent container to interact with.

4.3.10 Google Remote Procedure Call (gRPC) [14]

It is an open source high performance Remote Procedure Call (RPC) framework which is used to connect the environment and agent container during the evaluation.

4.3.11 Pod

Pods are the smallest, most basic deployable objects in kubernetes [12]. A Pod represents a single instance of a running process in your cluster. Pods contain one or more containers, such as docker containers. When a pod runs multiple containers, the containers are managed as a single entity and share the pod's resources.

4.3.12 Job

A Job creates one or more pods and will continue to retry execution of the pods until a specified number of them successfully terminate. As pods successfully complete, the job tracks the successful completions. When a specified number of successful completions is reached, the task (i.e, job) is complete. Deleting a job will clean up the pods it created.

4.3.13 Container Networking Interface [15]

It concerns itself only with network connectivity of containers and removing allocated resources when the container is deleted. It is used to restrict all the outgoing communication from the job pods to *eval.ai* domain in order to avoid any malicious activity by the users.

4.3.14 Fluentd [16]

Fluentd is an open source data collector for unified logging layers. It collects all the logs and system metrics for the agent and environment containers and pushes them to AWS cloudwatch. The logs can later be downloaded from AWS Cloudwatch interface for further analysis.

4.3.15 AWS CloudWatch [17]

Amazon CloudWatch is a monitoring and observability service built for engineers, developers, and IT managers. CloudWatch provides us with data and actionable insights to monitor the applications, respond to system-wide performance changes, optimize resource utilization, and get a unified view of operational health.

4.3.16 Amazon Elastic Container Repository (AWS ECR)

AWS ECR is a fully managed container registry that makes it easy to store, manage, share, and deploy our container images and artifacts anywhere. AWS ECR eliminates the need to operate our own container repositories or worry about scaling the underlying infrastructure.

4.3.17 Amazon Virtual Private Cloud (AWS VPC)

Amazon Virtual Private Cloud (AWS VPC) is a service that lets us launch AWS resources in a logically isolated virtual network that we define. We have complete control over our virtual networking environment, including selection of our own IP address range, creation of subnets, and configuration of route tables and network gateways.

4.4 Working

4.4.1 Setting up environment container

While setting up the AI challenge for reinforcement learning tasks, challenge organizers have to set up the environment container which contains the test environment, APIs for the agent to interact with the environment and code to update the results in the EvalAI database. They upload the docker image on AWS ECR and provide us the link for the submitted image in the challenge configuration.

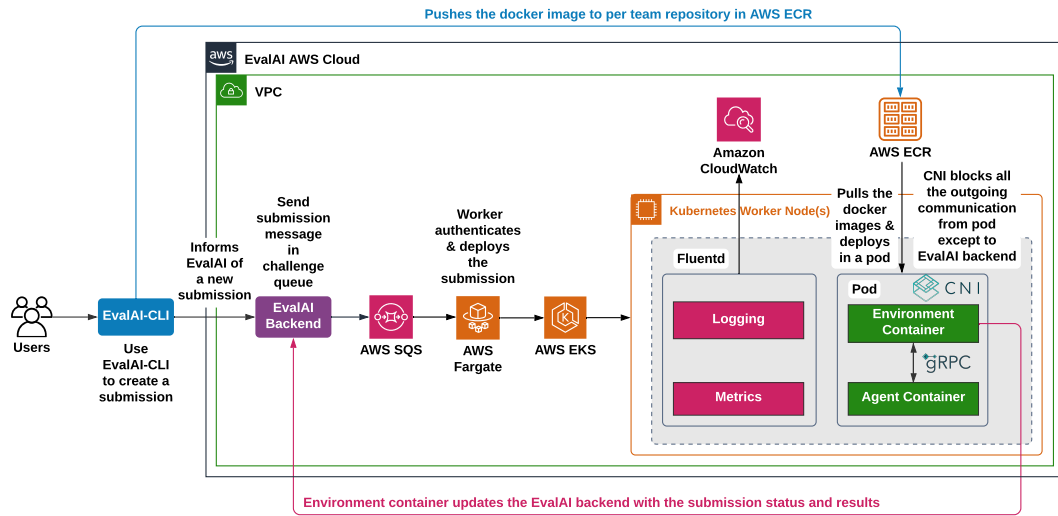


Figure 4.2: System architecture for evaluating code upload AI challenges on EvalAI. The users create a submission a submission in the form of a docker image using EvalAI-CLI tool which gets stored in AWS ECR repository. The EvalAI-CLI tool also informs EvalAI backend about the submission. A worker running on AWS Fargate picks up the submission and gives it to AWS EKS for evaluation. The AWS EKS deploys the submission on one of the node and finally results are updated in the EvalAI backend.

4.4.2 Evaluation using environment container

Once the participant pushes the docker image using EvalAI-CLI as described in Figure 4.2, the submissions are stored in AWS ECR repository, and a call to EvalAI backend is sent informing about the new submission which is then queued in the AWS SQS queue for the challenge. A worker running on AWS Fargate for the challenge is polling the queue for

the submissions. The worker picks up the submissions from the queue and gives it to AWS EKS which is responsible for deploying the submission on one of the AWS EKS nodes. As soon as the submission is deployed in the form of a pod, both the environment and agent docker images are pulled from AWS ECR repository and run on the machine. The agent and environment docker images communicate with each other using the Google Remote Procedure Call (gRPC) protocol. During the run, the logs are collected in real-time from both the containers by Fluentd and are sent to AWS cloudwatch. As soon as the submission completes, the environment container updates the results in EvalAI's database. In case of failure of any of the containers, the logs are sent to the participants. Once everything completes, the pod is deleted from the node and marks the evaluation as complete.

4.5 Features Offered

- The submission running backend is auto scalable depending upon the number of submissions or the usage of servers in the cluster.
- The agent code is run in a sanitized environment.
- Challenge organizers can configure which submissions to prioritize for evaluation in case of limited resources and time.
- The test environment is private and secure as only challenge organizers can access it and it isn't affected by agent docker containers.
- The evaluation infrastructure is modular and portable as it can be set up in any AWS cloud, while the challenge is hosted on EvalAI.

4.6 Evaluation Setups and Case Studies

We offer two types of setups for hosting code upload reinforcement learning challenges. In this section, we describe both the setups along a challenge using that setup.

4.6.1 Using our evaluation infrastructure

Challenge organizers can host reinforcement learning AI challenges using the evaluation infrastructure we have developed. Moreover, if they don't want to share the challenge data such as environment docker container, submissions, logs, etc. with us, then they can provide us the Amazon Identity and Access Management (AWS IAM) keys for their AWS account while challenge creation and our evaluation infrastructure will be automatically setup in their account. One such challenge using this setup is AI Habitat Challenge from Facebook AI Research which is an year long challenge hosted on EvalAI.

Case Study: AI Habitat Challenge

Embodied AI is the study of intelligent systems with a physical or virtual embodiment (robots and egocentric personal assistants). The embodiment hypothesis is the idea that intelligence emerges in the interaction of an agent with an environment and as a result of sensorimotor activity. AI Habitat is a simulator platform for research in Embodied AI. It enables training of such embodied AI agents (virtual robots and egocentric assistants) in a highly photorealistic and efficient 3D simulator, before transferring the learned skills to reality.

In order to test the trained agents, it uses EvalAI code upload challenge evaluation infrastructure hosted in Facebook's AWS cloud as shown in Figure 4.3. The challenge consists of two phases defined as PointNav which focuses on realism and sim2real predictivity and ObjectNav which focuses on egocentric object/scene recognition and a common sense understanding of object semantics. Each phase is divided into three splits i.e. minival, test-standard and test-challenge. Minival splits consist of a very small number of test-episodes (30 episodes) which is used to check the successful execution of the agent uploaded by challenge participants. Test-Standard splits contain a large number of test-episodes (2000 episodes) and are used to compare the state-of-the art (e.g. in papers) results. It is also used to maintain a public leaderboard that is updated on every submission. The test-challenge

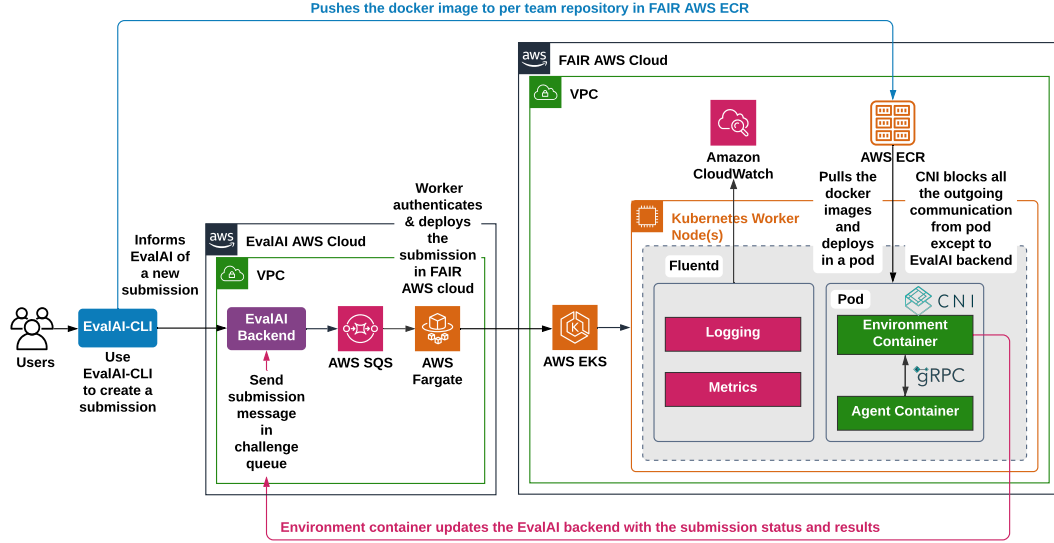


Figure 4.3: System architecture for evaluating AI Habitat code upload challenge on EvalAI.

split also contains a large number of test-episodes (2000 episodes) and is used to determine the winners of the challenge. EvalAI provides different visibility to the leaderboards. The minival and test-standard split leaderboards are public but the test-challenge leaderboard is private and gets revealed once the challenge completes each year. We have already hosted a couple of iterations of the challenge and we are hosting this year's challenge as well. In the previous iteration of the challenge, more than 30 teams participated and created over 500 submissions. None of the users or the teams reported issues with the evaluation and the challenge hosts were easily able to determine the winners for the challenge.

The use case for this setup arises from the fact that the data in certain domains such as medical domain, etc. are very confidential and sensitive. It is almost impossible to share such data with the admins of challenge evaluation platform. Moreover, sometimes challenge organizers have sufficient compute to evaluate the submissions on their internal servers which saves them the cost of using AWS resources. EvalAI also supports hosting of such challenges. The submissions will occur on the EvalAI platform which will be pulled by challenge organizers servers for evaluation. The organizers evaluate the submissions and then update the results in the EvalAI's database. We have hosted several challenges

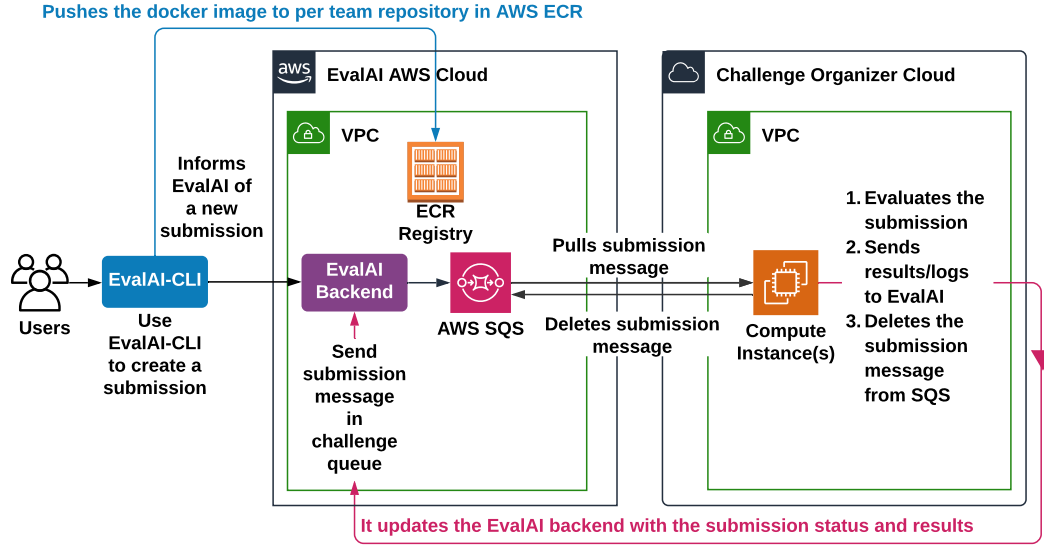


Figure 4.4: Code upload remote challenge evaluation.

using this setup from different organizations across the globe such as Animal-AI olympics challenge, Robothor challenge, Sim2Real challenge, GOSEEK challenge etc.

Case Study: Animal-AI Olympics

The Animal-AI Olympics is an AI competition with tests inspired by animal cognition. Participants are given a small environment with just seven different classes of objects that can be placed inside. In each test, the agent needs to retrieve the food in the environment, but to do so there are obstacles to overcome, ramps to climb, boxes to push, and areas that must be avoided. The real challenge is that they don't provide the tests in advance. It's up to participants to play with the environment and build interesting setups that can help create an agent that understands how the environment's physics work and the affordances that it has. The submission should be an agent capable of robust food retrieval behaviour similar to that of many kinds of animals. In order to set up the evaluation we provided the starter code to the challenge hosts which automatically pulled the submissions from the challenge queue and gave it to the evaluation code. The average evaluation time for the agents was an hour and then results are updated in the database by challenge organizers.

The challenge lasted for 4 months and received a huge participation from more than 100 teams who created more than 1700 submissions.

4.7 On Supervised and Unsupervised Learning Tasks

Supervised learning is the machine learning task of learning a function that maps an input to an output based on example input-output pairs. It infers a function from labeled training data consisting of a set of training examples. In unsupervised learning the AI model learns patterns from untagged data. The hope is that through mimicry, the machine is forced to build a compact internal representation of its world. A way to evaluate the models on these tasks is to upload predictions from the AI model on the EvalAI platform and compare them with ground truth annotations. In this generation of deep learning with more robust AI models we can enhance this evaluation method to upload the models rather than uploading the predictions from the models. In order to evaluate these models using code,

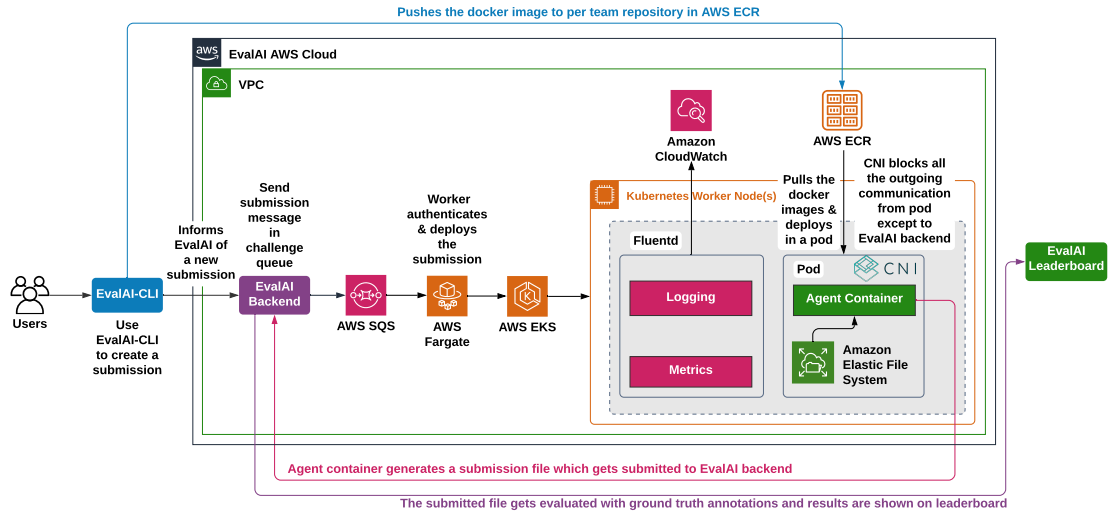


Figure 4.5: System architecture for evaluation AI agents on static datasets.

we can extend the reinforcement learning code evaluation infrastructure. The environment container in that infrastructure will correspond to the static test dataset stored on Amazon Elastic File System (AWS EFS)[18]. The infrastructure will differ when we run the AI agent docker image. In this case, we will attach an AWS EFS volume which contains the

static test dataset to the agent image and run the inference on it. A predictions file will be generated on the server which gets submitted to the EvalAI backend automatically. A worker on the AWS Fargate will evaluate it with the ground truth annotations and the results are finally shown on the leaderboard.

CHAPTER 5

HOSTING A CODE UPLOAD CHALLENGE ON EVALAI

EvalAI provides an easy method to host code upload challenges. We will describe the AI challenge creation process using github, uploading and evaluating submissions, analysing agents performance and finally downloading and running the agents in the real world.

5.1 Simplifying the AI challenge creation using github

AI challenge creation is a complex process which involves a lot of moving components such as challenge phases, data splits, leaderboards, etc. Also, it is becoming difficult to manage the hosted versions of an AI challenge over the years. Some other factors which add to it are the release of a bigger and newer version of the same dataset, introduction of new evaluation metrics in the community when a challenge is running, change in terms and conditions of the dataset over the period of time and bugs in the evaluation script while the challenge is running. In order to solve the aforementioned issues, we introduce AI challenge creation using github. It solves the existing problems and provides a streamlined method to challenge organizers for creating and managing the challenge.

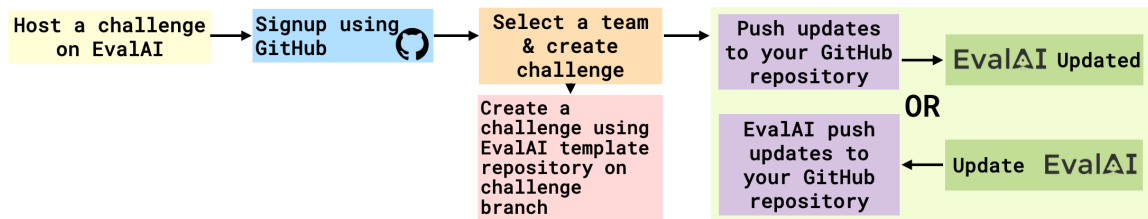


Figure 5.1: Workflow for challenge creation using github on EvalAI.

In this method, a challenge organizer can signup using github on EvalAI and use our EvalAI-Starters github repository as a template. To authenticate with EvalAI, they have to add the EvalAI authentication token in the repository along with a unique identifier

for the organizer team which can be fetched from EvalAI. Challenge organizers can now make changes to the challenge configuration, evaluation script, templates, etc. and push the commit on a new *challenge* branch. Once they commit on github, we run the challenge configuration validation build on the server and in case of any errors, we open the issues in the repository using their github token. Moreover, if they make changes to the challenge on the UI, then the same changes are reflected in the github repository.

5.2 Uploading Submissions and Evaluation

Once the participants participate in the challenge, they have to upload the AI agents in the form of docker images to EvalAI. The agents are evaluated using one of the code upload evaluation infrastructure described above.

5.3 Analyzing and Viewing the Agents' Performance

One of the use cases for storing logs from the agents is to create videos of the AI agents performing in the unseen test environment. This is useful for challenge organizers in order to analyze the top performing AI agents for weakness and improvements. Moreover, certain challenge analysis such as number of teams participated in a challenge, total number of submissions, etc. and graphs such as submission accuracy over time, rank of participants over time, etc. are useful for drawing insights from the challenge.

5.4 Downloading and Running the Agents in the Real World

The performance of the AI agents doesn't fully translate from the simulation environment to the real world due to the additional complexities of the real world. In order to evaluate the agents in the real world, the challenge organizers have to download the agents and deploy them on real robots to perform the same task. Since we accept the agents submissions in the docker image format, it becomes easy for challenge organizers to download that agents'



image and run it on the real robot or build a web demonstration from it.

Case Study: Sim2Real Challenge on Gibson Dataset

One of the challenges we hosted from Stanford i.e Sim2Real Challenge on Gibson dataset have utilized this feature. They used LocoBot as their robotic platform for the real-world testing of the AI agents. The challenge had three phases, Dev Phase, Challenge in Simulation and Challenge in Real World phase and received more than 100 submissions from 10 teams. The top performing team in the real world setting as shown in Figure 6.1 isn't the top performing team in the simulation setting Figure 5.3 which demonstrates the need for evaluating AI agents in the real world.

Leaderboard

Phase: Challenge in Real World, Split: Dev Split



 - Baseline submission
  - Private submission

Rank	Participant team	Scenario1	Scenario2	Scenario3	Total	Last submission at
1	DAN	0.44	0.33	0.11	0.89	10 months ago
2	inspir.ai.robotics	0.33	0.33	0.22	0.89	11 months ago
3	VGAI - TCS Research Kolkata	0.03	0.02	0.01	0.06	10 months ago
4	Joanne	0.00	0.00	0.00	0.00	10 months ago

Figure 5.2: Sim2Real Gibson challenge Real World Phase Leaderboard

Leaderboard

Phase: Challenge in Simulation, Split: Dev Split

 - Baseline submission
  - Private submission



Rank	Participant team	Scenario1	Scenario2	Scenario3	Total	Last submission at
1	inspir.ai.robotics	0.76	0.73	0.44	1.93	10 months ago
2	DAN	0.61	0.57	0.24	1.42	11 months ago
3	Baseline - Soft Actor-Critic	 0.35	0.29	0.11	0.75	11 months ago
4	Baseline - Soft Actor-Critic	 0.35	0.29	0.11	0.75	11 months ago

Figure 5.3: Sim2Real Gibson challenge Simulation Phase Leaderboard

CHAPTER 6

CHALLENGE ANALYTICS AND EVALAI HOSTING

6.1 Analytics Dashboards for the Users of the Platform

To increase the engagement of the users with the platform, to improve our services and to provide challenge organizers with a comprehensive view of the progress in the hosted task, we have built an analytics dashboard for the users.

6.1.1 EvalAI Admins

Observability has become one of the most important areas of our application and infrastructure landscape, and the open-source tools such as prometheus and grafana provide a robust solution for it. The metrics from different services used on EvalAI are stored using prometheus and grafana provides a flexible and visually pleasing interface to view the graphs.

6.1.2 Challenge Organizers

Challenge analysis helps challenge organizers to track the progress of a particular task over the years. EvalAI provides a visually appealing and easy to use interface to view and download the graphs. Graphs such as accuracy on a task over time, evaluation time of submissions with time and number of submissions in a challenge on a daily, monthly and yearly basis provides the understanding of the AI community working on the task to challenge organizers.

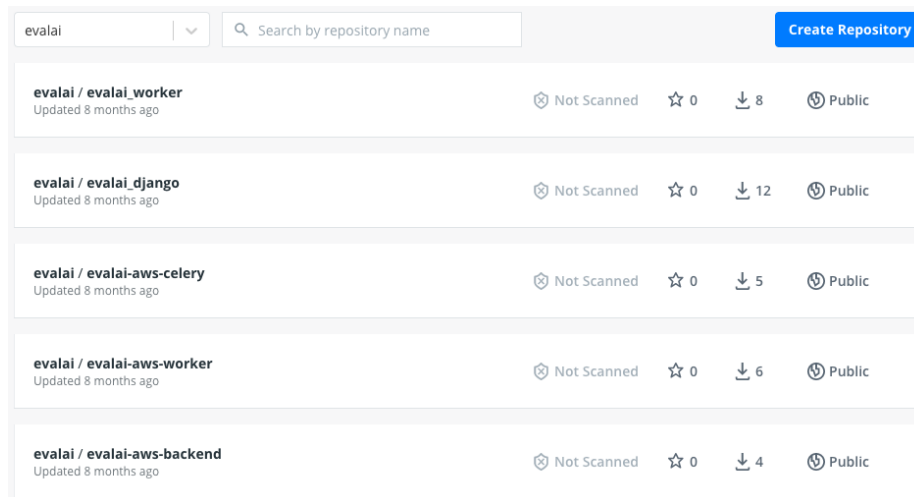
6.1.3 Challenge Participants

Challenge participants are curious to know the statistics such as the accuracy of their submissions over time, number of successful, failed, cancelled submissions and the trend of submissions in a challenge with the corresponding rank on the leaderboard.

6.2 Easy Hosting of EvalAI on Private Servers

Industry organizations have several limitations such as sensitive dataset, tasks which can only be accessed by their employees, and the limitation to use proprietary cloud infrastructure. Furthermore, the terms and conditions covering the usage of such datasets and tasks may be unfavorable for public hosting.

In order to provide a clean setup of setting up EvalAI within an organization, we provide a python script to package the EvalAI codebase in the form of docker containers which can be deployed on any machine. Moreover, we also provide the frontend, backend, etc. docker containers of EvalAI on a public docker repository which can be downloaded and run internally. Some of the industrial organizations such as eBay, ITU/WHO AI for health initiative, Astrazenica, etc, are using it instead of reinventing the wheel.



The screenshot shows the Docker Hub interface for the 'evalai' organization. At the top, there is a search bar with the text 'evalai' and a dropdown arrow, and a 'Create Repository' button. Below the search bar, there is a list of five Docker images, each with its name, update time, status, star count, download count, and visibility.

Repository Name	Updated	Status	Stars	Downloads	Visibility
evalai / evalai_worker	Updated 8 months ago	Not Scanned	0	8	Public
evalai / evalai_django	Updated 8 months ago	Not Scanned	0	12	Public
evalai / evalai-aws-celery	Updated 8 months ago	Not Scanned	0	5	Public
evalai / evalai-aws-worker	Updated 8 months ago	Not Scanned	0	6	Public
evalai / evalai-aws-backend	Updated 8 months ago	Not Scanned	0	4	Public

Figure 6.1: Public Docker Images for EvalAI

CHAPTER 7

IMPACT

As an open-source tool EvalAI has reformed the way for the evaluation of AI agents in the community. It has created its impression from hosting homework assignments from Georgia Tech, USC, UIUC, etc. to the evaluation of massive datasets such as VQA, Visual Dialog, etc., from evaluating static predictions files to evaluating agents' code and from free hosting of challenges to automating the infrastructure set up in challenge organizers account.

7.1 AI Community

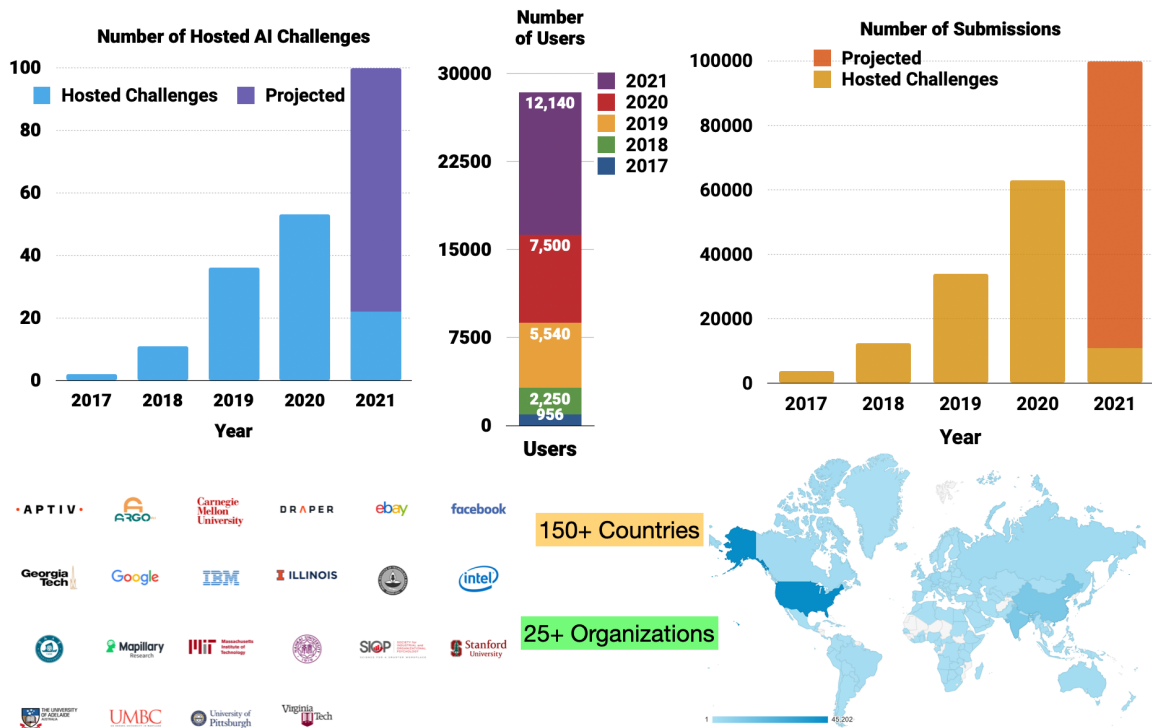


Figure 7.1: Impact in AI Community

EvalAI has been live for 4 years now as it started with the VQA challenge in 2017,

since then it has hosted **100+ AI challenges** consisting of **20+ code upload challenges** and **80+ prediction upload challenges**. The user base has grown from 2000 users in 2018 to **11000+** users in 2021 from **150+ countries**. There are more than **5000 participant teams** who have created more than **100 thousand submissions** on the platform.

7.2 Open-Source and Google Summer of Code (GSoC)

Being an open-source tool, EvalAI is housed under the CloudCV organization which provides it the support of **100+ contributors** ranging from frontend, backend developers and designers on Github with **1k+ stars** and **500+ forks**. The open source community also helps in building new features and maintaining the source code.

EvalAI has been an active participant of GSoC program under CloudCV organization for the last 4 years. GSoC is a global program focused on bringing more student developers into open source software development. Students work with an open source organization on a 10 week programming project during their break from school. I have served in GSoC in different roles starting from a student, mentor and organization administrator.

By participating in GSoC, EvalAI benefits from the tremendous enthusiasm, engagement, and involvement of the GSoC student community. Last four years with GSoC has given us a unique opportunity to build an open-source community that is focussed on solving challenging problems to help make AI more reproducible and reliable. We also look at GSoC as an opportunity to introduce students to cutting-edge AI research and help prepare them for whatever comes next in their careers, be it industry, graduate school or independent research.

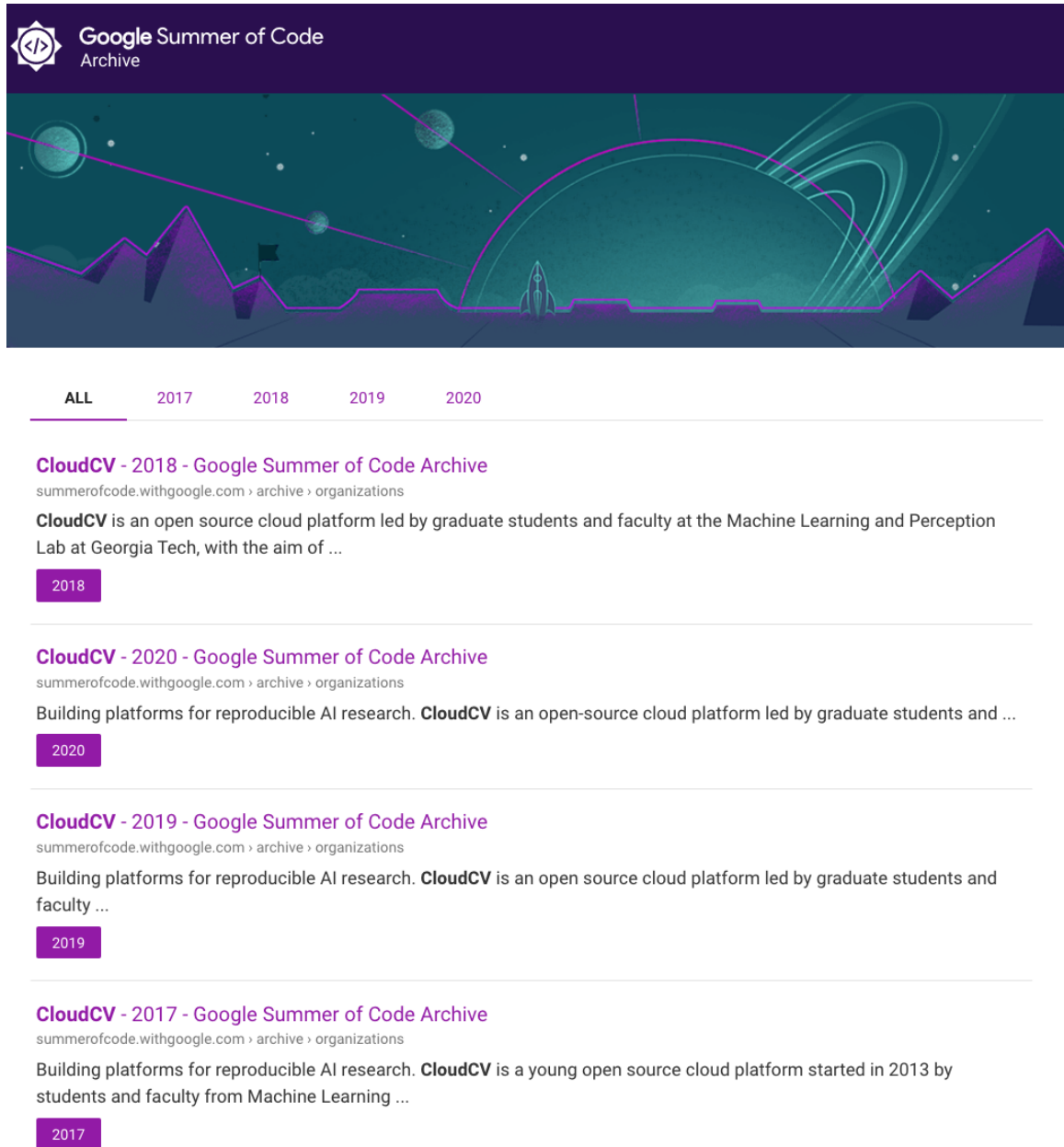


Figure 7.2: GSoC Participation from 2017 - 2020

CHAPTER 8

FUTURE WORK

8.1 Multi-agent Evaluation

Real-time strategy games remain a difficult domain for AI, owing to the large search space and confusion of what the opponent is doing. Multi-agent systems will offer a level of abstraction that will allow humans to design intelligent behaviours and systems in this domain. These agents have to be innovative, capable of learning the information from the opponents, and suggesting new issues to each other. The modeling of such agents and systems is not trivial and their evaluation is also a challenging task. However, existing evaluation platforms are either not compatible with multi-agent settings, or limited to a specific setting which motivates the need for a standard way of evaluating these agents at scale.

EvalAI's code upload challenge infrastructure paves the way for evaluation of such agents. Since we follow the two container (environment and agent) strategy for the evaluation of agents, we can extend this to upload multiple agents paired with the same environment in order to evaluate them against each other. Since the existing infrastructure is modular, portable, and scalable which will enable challenge organizers to modify it according to their needs.

8.2 Evaluation of AI agents on dynamic datasets

Artificial Intelligence is a fast moving field and with the release of robust and larger AI agents, the agents might overfit on the existing hidden test dataset. Using dynamic test datasets at each timestep according to the actions taken by the agent provides a more realistic method to evaluate agents in the simulation and prevents saturation of the benchmark

and overfitting on the test dataset. Therefore, we want to evaluate the AI agents on the dynamic test datasets.

The proposed evaluation infrastructure can be extended to build such a system. The environment docker container can be configured to download more data or change the data according to the actions taken by the agent. This would allow the AI community to check where a specific agent fails, allowing them to build improved model architectures and, as a result, pushing the state-of-the-art in Artificial Intelligence.

CHAPTER 9

CONCLUSION

In this thesis, we developed the evaluation infrastructure for setting up code-upload AI challenges on EvalAI. We proposed a modular, flexible, and portable system architecture which can be used to run the agents code in isolation, evaluate them in both simulation and real environments, and can be easily extended to evaluate static prediction based challenges. Interactive demos and videos for the agent helps challenge organizers to better visualize the behaviour of AI agents in unseen test environments. The architecture provides the features such as auto scaling of backend machines, prioritized submission evaluations, and private test environment and dataset. We also developed a remote code-upload evaluation pipeline in which the challenge organizers don't have to share the test data even with the challenge hosting platform. Both these pipelines are currently used in production settings for several challenges including Habitat AI Challenge from Facebook AI Research, iGibSon Challenge from Stanford, etc.

Moreover, we aim to reduce friction in the AI challenge creation process by introducing challenge creation using GitHub. This will help challenge organizers to update, version and manage the complex challenge configurations over the years. Challenge analytics is also another feature that we added for all the users of the platform. We believe that by providing the insights about the challenge to organizers, the progress in the domain can be easily measured.

EvalAI is an open-source platform which is used by several industrial organizations as an internal tool. We have automated the process for hosting EvalAI internally for prediction upload and code-upload challenges. We are also participating in Google Summer of Code (GSoC) program from Google since 2016. These programs are helping us in building a community of open-source developers for the project which will ultimately help us in

adding more relevant features for the community and fixing the issues with the existing ones.

Appendices

APPENDIX A

EVALAI: TOWARDS BETTER EVALUATION OF AI AGENTS

A.1 Introduction

Progress on several important problems in Computer Vision (CV) and Artificial Intelligence (AI) has been driven by the introduction of bold new tasks coupled with the curation of large, realistic datasets [19, 20, 21, 22, 23]. Not only do these tasks and datasets establish new problems and provide data necessary to analyze them, but more importantly they also establish reliable benchmarks where proposed solutions and hypothesis can be tested – an essential part of the scientific process. In recent years, the development of centralized evaluation platforms have lowered the barrier to compete and share results on these problems. As a result, a thriving community of researchers has grown around these tasks, thereby increasing the pace of progress and technical dissemination.

With the success of deep learning techniques on a wide variety of complex AI tasks such as grounded dialog generation [22] or generating aesthetically pleasing images [24] coupled with the widespread proliferation of AI-driven smart applications, there is an imminent to evaluate AI systems in the context of human collaborators. These tasks cannot be evaluated accurately using automatic metrics as performance on these metrics do not correlate well with human-judgment in practice[25]. Instead, to properly evaluate, they should be connected with a human workforce such as Amazon Mechanical Turk (AMT)[26] to mimic a setup which is closest to the one in which they may be eventually deployed.

Furthermore, the rise of reinforcement learning (RL) based problems in which an agent must interact with an environments introduces additional challenges for benchmarking. Unlike supervised learning, the performance in this setup cannot be measured by evaluating on a static test set. Evaluating these agents involves running the users code on a

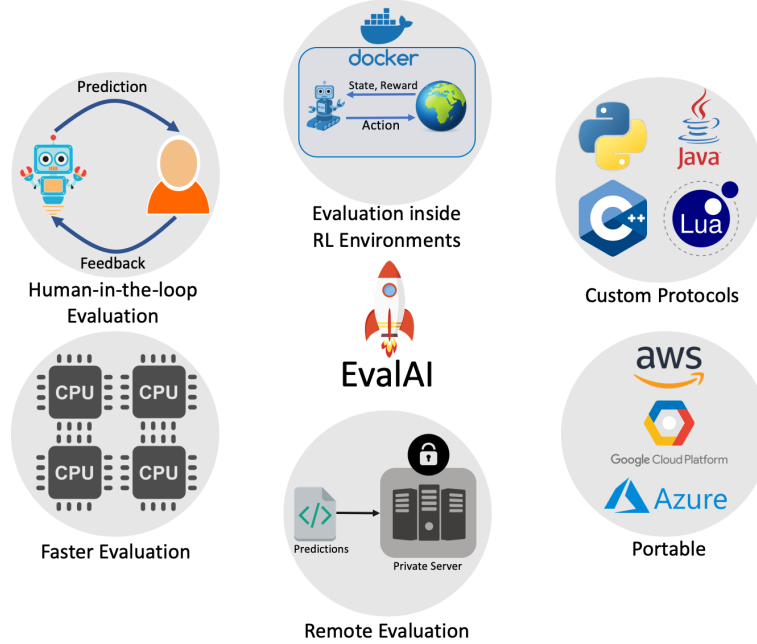


Figure A.1: EvalAI is a platform to evaluate AI agents in dynamic environment with human-in-the-loop.

collection of unseen environments such that one can check if algorithms “overfit” on training environments.

To address the aforementioned problems, we introduce a new evaluation platform called EvalAI that fullfills the critical need in the community for (1) human-in-the-loop evaluation of machine learning models and (2) the ability to run user’s code in a dynamic environment instead of a static dataset enabling the evaluation of interactive agents.

A.2 Related work

In light of the requirements highlighted in the previous section, we compare EvalAI with existing platforms. We also provide a head-to-head ocmparison in Table A.1. Kaggle[10], CodaLab[8] and AICrowd[9] are some of the most popular platforms for hosting machine learning competitions but they have several limitations. Kaggle doesn’t support custom evaluation metrics and multiple challenge phases – a common practice in popular challenges like COCO Caption Challenge, VQA etc. CodaLab provides an open-source

Features	OpenML	Topcoder	Kaggle	AICrowd	ParlAI	CodaLab	EvalAI
AI Challenge Hosting	✗	✓	✓	✓	✗	✓	✓
Custom metrics	✗	✗	✗	✓	✓	✓	✓
Multiple phases/splits	✗	✗	✗	✓	✗	✓	✓
Open Source	✓	✗	✗	✓	✓	✓	✓
Remote Evaluation	✗	✗	✗	✗	✓	✓	✓
Human Evaluation	✗	✗	✗	✗	✓	✗	✓
Environments	✗	✗	✗	✓	✗	✗	✓

Table A.1: Head-to-head comparison of capabilities between existing platforms and EvalAI

alternative to Kaggle and fixes several of their limitations but doesn’t support evaluating interactive agents in dynamic environments. EvalAI not only supports custom evaluation protocol but also allows evaluation of interactive agents in dynamic environments. In addition, we also support human-in-the loop evaluation of prediction based or code-upload based challenges, something AICrowd doesn’t support. Similar to ParlAI [27], EvalAI integrates with Amazon Mechanical Turk (AMT) [26] for human based evaluation. However, unlike EvalAI, ParlAI is not a challenge hosting platform and only supports evaluation of dialog models, not for any AI task in general. OpenAI gym [28] and EvalAI have the same underlying philosophy of encouraging easy accessibility and reproducibility of Reinforcement Learning (RL) agents but OpenAI gym is not a dedicated evaluation platform and lacks support for prediction based challenges, custom evaluation protocol, and human-in-the loop evaluation.

A.3 Key features

Evaluation inside RL environments: We have developed an evaluation framework to evaluate agents for tasks situated in active environments instead of static datasets (Figure A.2). Participants upload Docker images with their pre-trained models using a command line interface. At the time of evaluation, the instantiated worker evaluates the user-submitted model against test-environment provided by the challenge organizer. Once the evaluation is complete, the results are sent over to the leaderboard using the message queue.

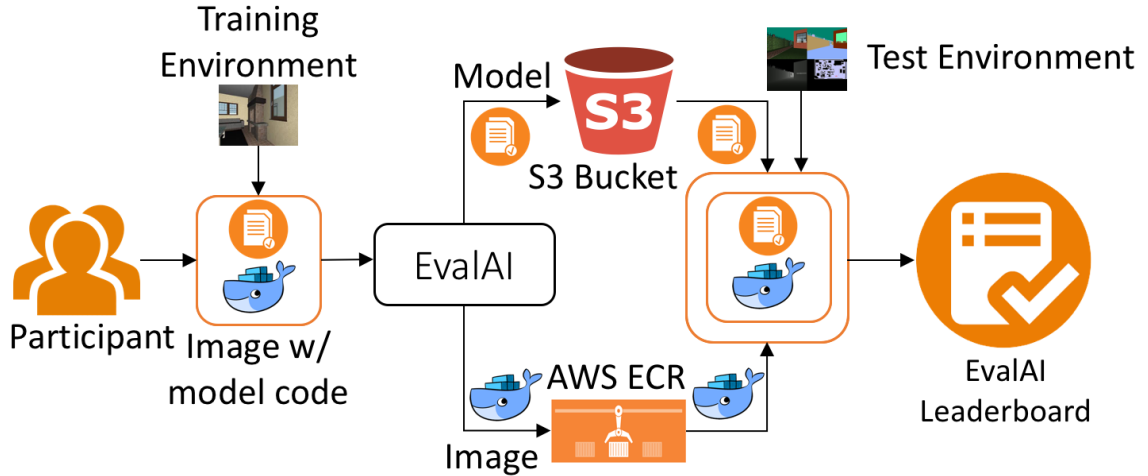


Figure A.2: System architecture for code upload challenges.

Human-in-the-loop evaluation: Automatic evaluation of tasks like image captioning [29, 30], visual dialog [22, 31] or image generation [24] is complicated by the huge set of possibly ‘correct’ responses and relatively sparse ground truth annotations. Given the interactive nature of tasks, it is clear that the most appropriate way to evaluate these kind of tasks is with a human in the loop, i.e. a Visual Turing Test [32]! Unfortunately, human-in-the-loop evaluation is still limited by financial and infrastructural challenges that must be overcome by each interested research group independently.

To address this, we have developed the infrastructure to pair AMT users in real-time with artificial agents (for instance, visual conversational agents [22]). We provide:

- **Custom HTML Templates:** Organizers can choose to provide their own HTML templates satisfying the unique requirements specific to their challenge.
- **Worker Pool:** We maintain a pool of good quality workers which have a history of high quality work and strong acceptance rate. Additionally, organizers can provide us with a list of whitelisted and blocked workers.
- **Uninterrupted back-and-forth communication:** For tasks that require multiple rounds of human-AI communication, we do a lot of book-keeping to ensure that

incompleted HITs are re-evaluated and turkers can reconnect with the same agent after temporary network failure.

- **Flexible schema:** We provide a flexible JSON based schema and APIs to fetch the results from the evaluation tasks once they are completed. These results are automatically updated on the leaderboard for each submission.

Private and Remote Evaluation: Certain large-scale challenges have special compute requirements for evaluation. For instance, challenges in medical domain such as FastMRI Image Reconstruction challenge [33] have sensitive data which cannot be shared with the evaluation platform. Some other AI challenges like CARLA Autonomous Driving challenge [34], and Animal-AI Olympics [35] need to run RL agents in a dynamic environment - requiring powerful clusters with GPUs. For these types of challenges, organizers can easily setup their own cluster of worker nodes to process participant submissions while we take care of hosting the challenge, handling user submissions and the maintaining the leaderboard. On submission, all related metadata is relayed to an external pool of workers through dedicated message queues - decoupling the worker nodes from the challenge front-end.

A.4 Impact

As shown in Table A.2, EvalAI has already hosted 35+ challenges, with over 1400 participants from 84 countries who have created over 35000 submissions. Some of the large scale challenge that EvalAI hosted are CARLA Autonomous Driving Challenge [34], Animal-AI Olympics Competition [35], Vision and Language Navigation [36], Habitat Challenge [37], Visual Question Answering Challenge [21] and many more.

Year	# submissions	# participants	# challenges	# page views
2018	12,516	357	11	306,517
2019 (YTD)	23,357	1,069	25	642,383
Growth	86%	186.5%	127%	109.6%

Table A.2: EvalAI growth statistics

A.5 Conclusion

While traditional platforms were adequate for evaluation of tasks using automatic metrics, there is a critical need to support human-in-the-loop evaluation for more free-form multimodal tasks such as (visual) dialog and image generation. We develop, EvalAI, a large-scale evaluation platform to support the same. To this end, EvalAI supports pairing an AI agent with thousands of workers in an interactive dynamic environment so as to rate or evaluate the former over multiple rounds of interaction. By providing a scalable platform that supports such evaluations will eventually encourage the community to benchmark performance on tasks extensively, leading to better understanding of a model’s performance both in isolation and in human-AI teams[38].

APPENDIX B

EVALUATING VISUAL AND TEXT EXPLANATIONS IN AN INTERACTIVE, GOAL-DRIVEN HUMAN-AI TASK

B.1 Introduction

Humans are now collaborating with AI systems with increasing frequency across applications ranging from medical diagnosis, driving vehicles, to scheduling meetings. Analogous to human tools, the effectiveness of the human-AI collaboration depends on the capability of the AI and its *usability* by a lay person. Recent works in Explainable AI have made exciting progress in developing approaches that explain decisions of deep neural networks such as saliency-based methods, attention-based methods, etc. Progress is typically measured by evaluating explanation modalities via quantitative metrics [5] on static datasets, human studies with quality judgements [4] or proxy tasks like predictability [39] and verification [40]. While these works improve researchers’ understanding of explanation modalities, it is unclear to what extent they are useful to a lay person in an interactive downstream task. In this work, we evaluate the utility of visual and text explanations in an interactive, goal-driven, collaborative human-AI task with a lay person.

GuessWhich [6] is a multi-modal task, that is similar to the ‘20 questions’ game. Given a pool of N images, a human subject attempts to guess the ‘secret’ image which is known to the AI. The human asks a question regarding the secret image, which the AI answers. Figure B.1a shows the pool of images and the human subject’s question, followed by the AI’s answer for the secret image (bottom image with green border). After T rounds, the subject makes a guess. The accuracy of the guess depends on the subject’s questions and the usefulness of the AI’s answers to the human. GuessWhich has a number of desirable properties – it is complex, goal-driven, interactive, involves a real human subject, and lends

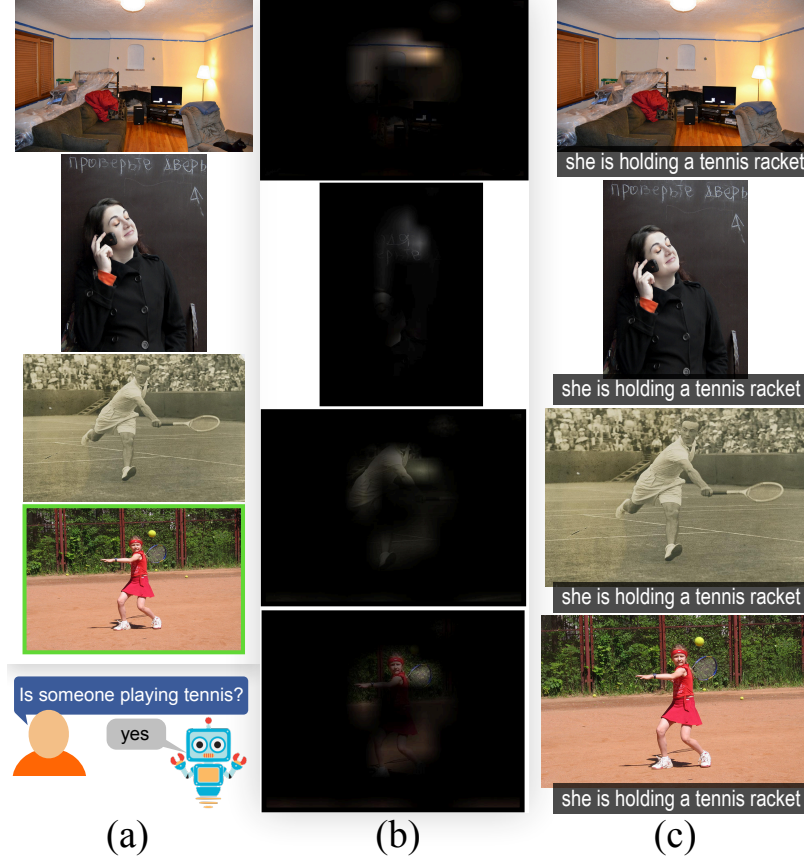


Figure B.1: (a) The pool of images in the game among which the human subject attempts to identify the secret image (green border). The subject asks ‘Is someone playing tennis?’. The AI answers the question (‘yes’) for the secret image. The subject guesses the secret image that is most consistent with the question and answer. (b) Grad-CAM visual explanation for the *secret image* is overlaid on all images in the pool. The heat-maps highlight regions in the image that contribute most to the AI’s prediction. (c) Text explanation that reasons about the answer for the secret image is provided with all images. The human subject guesses the secret image that is most consistent with the question, answer and explanation.

itself to remote experiments via crowd-sourcing platforms. An evaluation approach with these characteristics can provide valuable insights regarding the utility of explanations to a lay person.

We adapt the GuessWhich game to also provide subjects the AI’s explanations regarding its answer to the subject’s question. After the subject asks a question, receives an answer and makes a guess, we display the AI’s explanations on the game interface by default (with an option to toggle the explanations on/off). Subjects then attempt to guess the secret image by identifying the image that is most consistent with the question, answer and

explanation. Note that only one explanation is provided for each image in the pool, that corresponds to the subject’s question and answer on the secret image. We evaluate the extent to which the answer and explanation are useful to the subject in the context of the game. We evaluate performance of the human-AI team before and after the subject has access to explanations. An increase in performance demonstrates the utility of explanations from the AI in the context of a collaborative human-AI team. We implement a Visual Question Answering (VQA) [41] model as the AI (described in subsection B.3.3), pools containing 4 images (described in subsection B.3.2), and number of rounds $T = 1$. Details regarding the game-play are provided in subsection B.3.1.

We evaluate Grad-CAM [4], a popular task-agnostic saliency-based visual explanation that is faithful to the model. As shown in Figure B.1 (right column), the visual explanation for the secret image which is in the form of a heat-map, is overlaid on every image in the pool. The subject then selects the image that is most consistent with their question, the model’s answer and the overlaid heat-map. We also evaluate an approach that generates text rationales [5] for the VQA model’s answer, in the form of a natural language sentence. Similar to the visual explanations, we provide subjects with the text rationale that explains the VQA model’s answer for the secret image. As shown in Figure B.1 (middle column), the text explanation is provided below each image in the pool. Details regarding the explanation modalities are provided in subsection B.3.4 and subsection B.3.4.

We recruit lay human subjects from the Amazon Mechanical Turk (AMT)¹ crowd-sourcing platform. Each subject plays a total of 20 games, playing each game exactly once.

With the exception of concurrent work by Ray et al. [42]², to our best knowledge, this work is the first to evaluate the effect of existing task-agnostic visual and text explanations from deep neural networks in the context of a goal-driven, collaborative human-AI task. We believe that our work contributes to a better understanding regarding the influence of

¹<https://www.mturk.com/>

²Although the proposed evaluation paradigm is similar, Ray et al. [42] differ from this work in a number of ways. The salient differences are discussed in section B.2.

interpretable explanations in the context of an interactive, concrete task with a real human. Specifically, the contributions of this work are as follows:

1. We investigate the under-explored problem of analyzing the utility of visual and text explanations in an interactive, multi-modal goal-driven, collaborative human-AI task.
2. We provide empirical evidence for, and analyses of, the discrepancy between metrics on a static dataset for the VQA task and performance in the goal-driven task (subsection B.5.2).
3. We find that human subjects utilize existing visual and text explanations regarding the AI’s answer to achieve higher performance on the task (subsection B.4.3).
4. We find that subjects obtain more useful information from text explanations than visual explanations, raising interesting questions regarding the nature of actionable information conveyed via different modalities of explanations (subsection B.4.4).
5. We analyze human strategies and find that (i) subjects often form a hypothesis regarding the secret image and ask questions that seek to confirm this hypothesis, (ii) subjects do not often adapt their question-asking strategy across games despite observing the AI’s mistakes on similar questions from previous games. (subsection B.5.2).

B.2 Related Work

End-goal of explanations. Prior work has evaluated various aspects of explanations, such as the ability to improve trust [4, 43] or predictability [39] of the model to a lay person, etc. We evaluate the *utility* of explanations in improving *performance* of a human-AI team in a goal-driven, interactive task. This implicitly measures whether the explanation, output answer and input question is more consistent with the secret image or a distractor image from the pool. In the context of verification tasks, this can be considered a (variant of) relative *consistency* [40].

Evaluation paradigms for explanation modalities. Doshi-Velez and Kim [44] present a taxonomy of evaluation approaches in increasing order of specificity and cost – functionally-

grounded evaluation (no real humans, proxy tasks), human-grounded evaluation (real humans, simple tasks) and application-grounded evaluation (real humans, real tasks). We present an evaluation paradigm intermediate between human evaluation with simple tasks and real-world tasks – we evaluate interpretability approaches with real humans performing complex, interactive, goal-driven tasks.

Saliency-based visual explanations. Simonyan et al. [45] presented a gradient-based approach to visualize saliency maps in a convolutional neural network (CNN), based on earlier work [46]. Following this, several gradient-based to visualize saliency heat maps have been proposed, e.g., Guided Backprop [47], Integrated Gradients [48], SmoothGrad [49], etc. In this paper we use Grad-CAM [4], a popular approach which has been reported to have desirable properties [50].

Apart from gradient-based approaches, Ribeiro et al. [43] present a model-agnostic approach to interpret the decisions from a deep model by learning an interpretable linear decision boundaries that locally approximates the CNN.

Text explanations. Several recent works have focused on generating text explanations for deep multi-modal models. Hendricks et al. [51] presented an early approach that justified a classifier’s decision in terms of visual attributes, and also ground the decision in these attributes [52]. Park et al. [53] present a dataset consisting of human-written explanations that reason about a VQA model’s decision. Wu et al. [5] use only rationales that are ‘relevant’ (based on the image, question, model) from this dataset to train an explanation module, which we utilize in this work.

Image-guessing games. We propose the use of GuessWhich [6], which can be considered a form of Lewis signaling game [54] with a sender (questioner) and receiver (answerer). Das et al. [55] train RL dialog agents to improve communication by cooperatively playing a similar image-guessing game. The goal in GuessWhich is to identify the secret image, given a set of distractors. De Vries et al. [56] implement as an AI evaluation test-bed, a similar two-player question-based guessing game in which the objective is to identify the

location of an object in the image.

VQA. We use a VQA model as the multi-modal AI in our work. The model is trained to answer questions given an image and question [41]. Several models have been proposed over the years to solve the VQA task. We implement the CNN-LSTM model [57] for its interpretable Grad-CAM visual explanations. We also implement a variant of the Bottom-Up-Top-Down (BUTD) attention model [58] in the text explanations experiments as per [5]. **Ray et al. [42].** Concurrent work [42] proposes an approach to evaluate explanations that also utilizes the GuessWhich game [6]. Unlike their work which evaluates attention maps from the model, and custom-designed explanations for GuessWhich, we are interested in evaluating the performance of *existing task-agnostic* explanation modalities in a downstream task with a human subject. In addition, our experimental setup differs – Ray et al. present different explanations for each image, each corresponding to the respective image in the pool. Thus, the task of the human subject is to identify the (image, explanation) pair that is consistent with (a) the question and (b) the answer. In contrast, in our experiments, we present subjects with a single explanation for the secret image. Subjects thus, have to identify the secret image that is consistent with (a) the explanation (b) the question (c) the answer.

B.3 Approach

In this section, we first describe relevant details regarding the GuessWhich game [6], then our modifications to the game to evaluate the utility of explanations in the context of the interactive task. We then describe details regarding the game’s image pool construction, AI models, explanations, human players and back-end infrastructure.

B.3.1 Gameplay

The AI is assigned a secret image – one among the pool of images in the game. The objective of the game is for the human subject to accurately identify the unknown secret image.

The subject asks a question about the secret image and obtains the AI’s answer. Based on this, they select an image as their guess of the secret image. Following this, explanations from the AI are displayed on the interface by default (with the option of toggling them on/off).

The Grad-CAM [4] visual explanation is a heat-map which is overlaid on all images of the pool (see 1(b)). The heat-map represents regions in the secret image that contribute most to the model’s prediction. The natural language sentence of the text explanation is shown underneath all images in the pool (see 1(c)).

The subject considers both the model’s answer and the explanation for the answer, to guess the secret image successively (with feedback) until they guess the secret image correctly. We display the subject’s running score which is proportional to the accuracy of their guesses. The score serves as a tool to gamify the task and also as an incentive for subjects to perform well, since they are provided a task bonus that is proportional to their game score.

Overall, a human subject plays 20 different games, and guesses the secret image before and after having access to explanations. Crucially, every task (of 20 games) across different experimental settings is completed by a unique subject to prevent leakage of information regarding the model across tasks.

B.3.2 Pool Selection

The pool of images in a game are selected in a manner that attempts to keep the game challenging, yet engaging for a human subject on a crowd-sourcing platform. In pilot studies, we found that the GuessWhich interface with a pool of 20 images resulted in a task that was too challenging. After varying pool sizes and rounds, we ultimately determined that games with 4 images in the pool, and a single round of question-answering were of optimal difficulty. The pool of images are sampled from the validation split of the COCO dataset [20] to avoid overlap with images that the VQA models were trained on. Images

in their original aspect ratio are placed in random order in a rectangular grid, as shown in Figure B.4.

The pool is constructed by first determining the secret image, then sampling a hard-negative ‘neighbor’. The remaining two images are randomly sampled to complete the pool.

Secret image. In an effort to create diverse games for subjects, we sample a diverse set of secret images. First, we compute the average representation of all images belonging to a particular category in the COCO validation set, obtaining the ‘canonical representation’ for that category. The image with representation most similar to the canonical representation is considered a ‘canonical image’. An image is represented by the set of activations from the penultimate layer of VGG Net [59], a popular convolutional neural network.

Overall, we acquire 80 canonical images corresponding to each of the 80 COCO categories. From this set, we sample 20 images uniformly randomly for use as secret images in our 20 games.

Hard negatives. The GuessWhich game involves identifying the secret image from among distractors. Thus, the presence of images that are visually similar to the secret image likely increases difficulty of the game. We ensure that the game is sufficiently challenging by sampling an image from among the neighbors of the secret image in image representation space.

Random images. For each pool, 2 images (distinct from the secret image and hard-negative) are sampled uniformly randomly from the COCO validation set.

B.3.3 AI models

In our experiments, we employ deep neural network models that are trained to perform the task of Visual Question Answering (VQA) [41], i.e., they predict an answer, given an image and a question about the image. The architecture of the model implemented in the Grad-CAM is a CNN-LSTM model [57]. The VQA model in the text explanations experiments

is based on the Bottom-Up-Top-Down (BUTD) model introduced by Anderson et al. [58]. **CNN-LSTM model.** The CNN-LSTM VQA model is a two-stream architecture consisting a VGG-19 Net [59] that encodes the input image, and an LSTM [60] that encodes the input question. The question and image representations are then fused together via a point-wise multiplication. A multi-layer perceptron (MLP) projects the fused representation into an output score over 1000 categories (the 1000 most frequent answers in the training set). The highest scoring answer from the model is provided to the human subject.

The CNN-LSTM model achieves an accuracy of 58.16% on the test-standard of the VQA 1.0 dataset. Selvaraju et al. [4] find that the Grad-CAM heat-maps from this model are weakly correlated with human attention maps. We use this model due to its simplicity and interpretability of heat-maps. We also tried visualizing Grad-CAM explanations from the Hierarchical Co-Attention model (a popular VQA model introduced by Lu et al. [61]) but from visual inspection, found the heat-maps to be less interpretable.

Bottom-Up-Top-Down model. With text explanations, we employ the VQA model architecture described in [5] which is similar to BUTD [58]. In the BUTD model, a set of object-like image regions are first proposed as candidates, and visual features from these regions are weighted using an attention mechanism. While in [58], candidate regions are obtained from Faster R-CNN [62] the architecture in our experiments uses segmented image regions from [63]. The question is encoded via a Gated Recurrent Unit [64], and this representation is used as context for attending over visual features. The weighted sum of attended visual features is fused with the question representation using point-wise multiplication. An MLP over the fused representation outputs a score over 3127 categories (answers) which includes every correct answer that appeared more than 8 times in the training set. The highest scoring output is provided to the subject as the response of the AI. We refer readers to Wu et al. [5] for further details regarding the model.

B.3.4 Explanations

Grad-CAM. Grad-CAM is a saliency-based visual explanation that is faithful to the model’s prediction. Given an output, Grad-CAM computes the support for it in terms of spatial regions of the image. The contributions from different regions are visualized as a grayscale heat-map, which is back-projected into the size of the original image. For further details regarding Grad-CAM, see [4].

We compute Grad-CAM heat-maps that explain the model’s highest scoring prediction for the subject’s question on the secret image. Image regions that contribute highly to the prediction are shaded lightly in the heat-map, and regions with smaller contributions are shaded darker. As described earlier, we overlay the heat-maps over each image in the pool. When Grad-CAM heat-maps are overlaid on the original image, regions that contribute highly to the prediction are more visible than regions with smaller contributions.

Text explanations. A thought-provoking recent approach in explainable AI produces text explanations for a model’s prediction [5]. The rationale is generated by an ‘explanation module’ that is conditioned on the image, the VQA model’s attention over segmented objects, the input question, and the output answer. Human-written rationales that attempt to explain the model’s prediction are first collected. Then, to determine whether a rationale is consistent with the VQA model’s prediction, the visual features that contribute most to the rationale are identified via a sensitivity analysis. The rationales for which the highly contributing visual features also have high attention weights from the VQA model, are deemed to be consistent with the VQA model. The explanation module is trained using the consistent rationales. For further details, please see [5].

B.3.5 Crowd-sourcing

Human subjects. We recruit human subjects from Amazon Mechanical Turk (AMT)³. To ensure that we recruit subjects who read instructions carefully, we constrain our subjects to

³<https://www.mturk.com/>

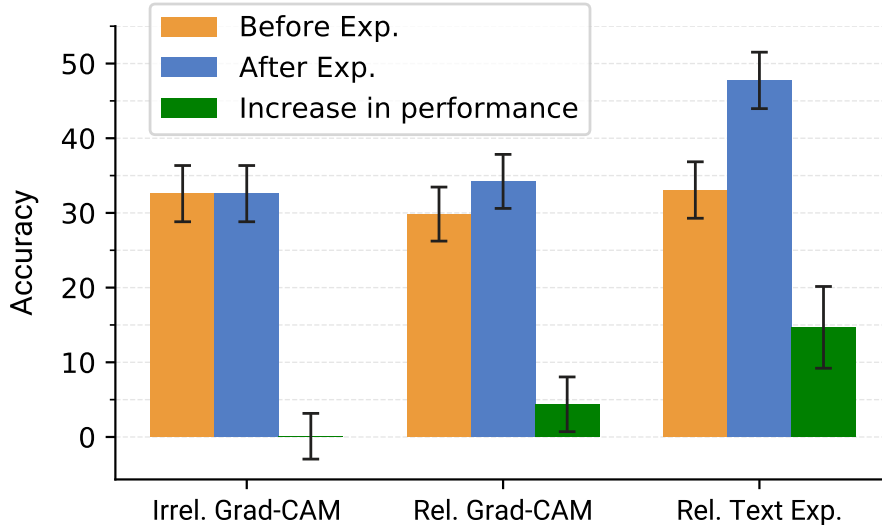


Figure B.2: Mean performance of the human-AI team at the goal-driven task before (orange) and after (blue) human subjects gain access to explanations. The improvement in performance is the per-game difference averaged across 20 games for each subject. The mean improvement across subjects is presented (green). Error bars denote the 95% confidence interval around the mean ($1.96 \times \text{std. error}$).

those who have completed at least 5000 HITs (tasks) on AMT, have a task approval rating of above 98%. Each subject is assigned a single task of 20 games to prevent the familiarity from these games from ‘leaking over’ to other tasks, leading to potentially inflated game scores.

Infrastructure. We set up a live interaction between a human subject on AMT and a VQA model running on an AWS (EC2 GPU)⁴ instance. The interface back-end is connected to RabbitMQ⁵ which queues the subject’s questions. RabbitMQ provides a channel for the user’s questions to the VQA model, which predicts the answer for the secret image. The answers are returned to the subject using web-sockets in real-time. We replicate the infrastructure in Chattopadhyay et al., and refer readers to their paper [6] for details.

B.4 Experiments

Recall from subsection B.3.1, that subjects playing the game guess the secret image in two stages – first, *only* based on the model’s answer to the subject’s question, and second, after the model’s explanation for the answer becomes available. We compare the overall performance of the human-AI team before and after explanations are available.

B.4.1 Evaluation metrics

Performance of the human-AI team is defined as the fraction (in %) of correct guesses of the secret image to the total number of games played. Figure B.2 reports the mean performance before and after explanations across all subjects and games. Another way to view this difference in performance is by directly computing the improvement in performance after the explanations are available. The improvement in performance for each subject is measured by the mean improvement in accuracy (across 20 games) after explanations are provided. Figure B.2 reports the mean improvement in performance across all subjects (in green).

B.4.2 Irrelevant visual explanation

To benchmark the performance of the human-AI teams at the modified GuessWhich task, we implement a baseline (control) experiment. We overlay an unrelated ‘explanation’ heatmap over images in the pool instead of the actual explanation. Irrelevant Grad-CAM explanations are generated via the following procedure – we sample a random question, and a random image from the COCO validation set. The question and image are provided as input to the VQA model. The Grad-CAM map corresponding to the most confident answer from the model is computed for this question about the image.

Overall, 31 subjects play a total of 20 games each. I.e., we have a total of $31 * 20 = 620$

⁴<https://aws.amazon.com/ec2/instance-types/>

⁵<https://www.rabbitmq.com/>

data points. We observe that subjects frequently change their guess of the secret image after considering the (unrelated) explanation. However, as we see in Fig Figure B.2 (‘Irrel. Grad-CAM.’), the total number of correct guesses before equals the number of correct guesses after explanations are available.

Despite the variance in performance improvement across individual subjects, the mean improvement in performance is 0 as shown in Fig Figure B.2. Interestingly, subjects do not seem to be misled by the irrelevant Grad-CAM maps on significantly many occasions.

B.4.3 Relevant visual explanation

In this experiment, the Grad-CAM heat-map is computed for the subject’s question and the model’s most confident prediction (answer) for the secret image. Thus, the explanation highlights regions in the secret image that contribute to the model’s answer to the subject’s question. Overall, 32 subjects play a total of 20 games each. First, we compare the performance *before* the explanation is available with the baseline experiment presented earlier, i.e., ‘Rel. Grad-CAM.’ vs. ‘Irrel. Grad-CAM’ in Fig Figure B.2. We note that although the VQA model is identical in both cases, there exists an inter-trial variance due to the variability in human subjects. Second, we observe that the performance *after* the subject considers explanations, is higher than *before* explanations are available to the subject. However, there also exists a significant variance in performance across subjects, as indicated by the error bars (95% confidence interval around the mean).

The mean improvement in accuracy across subjects $+/- 1.96 * \text{std. error.} = 4.375 + / - 3.664$. Thus, we also observe a small but significant improvement in the subjects’ ability to guess the secret image after they gain access to the relevant Grad-CAM explanations.

B.4.4 Relevant text explanation

In this experiment, subjects are shown the relevant text explanation, i.e., the rationale that explains the VQA model’s predicted answer for the subject’s question on the secret image.

Overall, 31 subjects play a total of 20 games each.

The performance of the team is significantly higher after the subjects are provided with the text explanations, compared to before, as we see in Figure B.2 (‘Rel. text exp.’). Text explanations seem to provide the human subject with information that helps them better identify the secret image as shown by the large mean improvement in performance (green) in Figure B.2. Specifically, in comparison with Grad-CAM visual explanations, the information gain from text explanations seems significantly higher to the subject.

We present some analyses of subjects’ interactions with text explanations that may partly explain the large improvement in performance (section B.5) and identify new directions for future work utilizing text explanations in interactive settings (section B.6).

B.4.5 Chance performance

To contextualize the performance of the human-AI team reported in Figure B.2, it is important to consider how much better the guessing strategy is, compared to random (chance) performance. At first glance, it may appear that this is 25% since there are 4 images in the pool. However, this may not be the case.

The pool construction strategy (described in subsection B.3.2) may have an unintended consequence – on observing the images, it may be possible for a subject to discern that a pool contains two images that are similar (the secret image and hard negative often belong to the same class), and two images that are unrelated to the rest.

In the event that human subjects recognize this similarity, and further reason that the secret image is one among the two similar images, we would expect the performance *before explanations* to be $\geq 50\%$. In our experiments, we observe that the mean performance before explanations is in the range of (29% to 32%). Thus, on average, the human subjects in the reported experiments do not seem to accurately identify the strategy for pool construction.

Note that the game only elicits a guess from subjects *after* one round of question-

answering, i.e., the user first observes the pool, then asks a question, and guesses the secret image based on the model’s answer to the question. To accurately estimate subjects’ ability to identify the pool construction strategy, one could perform the following experiment – given only the pool, ask human subjects to guess the secret image. We do not perform this experiment for two reasons. First, our primary interest is in the interactions between a human and AI. So, we refrain from encouraging subjects to guess the secret image from the pool *before* interaction.

Second, our objective is to evaluate the utility of explanations in improving performance of the human AI team. When evaluating the *improvement* in performance with the information provided by explanations, the potential added benefit of recognizing the pool construction mechanism would likely not change this difference. Thus, estimating the extent to which human subjects identify our strategy for pool construction is orthogonal to the research question in this work.

Sensitivity to pool. Recall that we attempt to ensure that the GuessWhich games are of moderate difficulty by generating pools accordingly. Based on pilot studies and prior work [6], we find that a game containing a large number of images (especially hard-negatives) is very challenging, given the imperfect AI models. The moderate performance of the human-AI teams (reported in section B.4), suggests that the games are neither too easy nor too difficult (given the VQA model that the humans interact with). Based on our experience from playing the GuessWhich game with explanations, we formed a few hypotheses regarding the gameplay and performance.

Consider the situation before explanations, when the model’s answer for the subject’s question on the secret image is correct. We hypothesize that the human subject can narrow down the plausible candidate images to the secret image and the hard-negative. Since the hard-negative is relatively close (in representation space) to the secret image, we expect that an answer to a question on the secret image, might often also be applicable to the hard-negative image. Similarly, in the event that the question on the secret image is answered

incorrectly by the model, it is plausible that the human subjects narrow down the candidates to the two random images.

Further, in the event that the human subject successfully narrows down the plausible candidates to two images – one of which is the secret image – they can effectively utilize the explanation from the model to accurately identify the secret image. Recall that the explanation available to the subject is with respect to the answer predicted by the model for the question asked on the secret image. Thus, in cases where the explanation is consistent with the predicted answer, question, and the secret image, we expect human subjects to accurately identify the secret image. Specifically, human subjects will likely be able to guess the secret image correctly when the explanation is more consistent with the predicted answer for the secret image than with other plausible candidate images.

In cases where the explanation is equally (or more) consistent with a candidate (distractor) image compared to the secret image, we do not expect the human subject’s guess to be accurate.

Overall, the performance of the human-AI team depends on the pool of images, i.e., the choice of secret image, the hard-negative and the 2 random images.

B.5 Analysis

We examine aspects of the AI, gameplay, explanations and subjects’ perceptions in some detail via quantitative and qualitative analysis.

B.5.1 AI performance on VQA vs. GuessWhich

AI models are often evaluated on quantitative metrics from static datasets, e.g., VQA [41]. A critical question that can evaluate progress is – how well do these models work in practice? In this section, we analyze the extent to which performance of the AI on the VQA task translates to performance of the human-AI team in GuessWhich.

We consider the performance of the VQA models *before* explanations, which corre-

sponds to Grad-CAM (CNN-LSTM) and Text Exp. (BUTD) in Figure B.2. Despite the differences in architecture and the datasets on which the two models were trained on, the performance of the two models are quite similar. The CNN-LSTM model was trained on VQA v1, a subset of the VQA v2 dataset that the BUTD model was trained on. The BUTD model achieves a VQA accuracy of 70.34 on the VQA v2 test set, compared to the CNN-LSTM’s 58.16 on the VQA v1 set (despite being typically, an easier task).

Thus, differences in performance of the models on VQA benchmarks do not translate to the interactive task of GuessWhich. The next section analyzes reasons for why this might be the case.

B.5.2 Accuracy of VQA model in GuessWhich

Performance of the human-AI team (before explanations) depends on the pool of images, question, answer, and the subject’s ability to integrate information to make the right guess. In this section, we analyze the accuracy of the VQA model on the questions asked by the human subjects while interacting with the model during the GuessWhich game. We sample of 15 subjects ($15 * 20 = 300$ games) randomly from the set of 31 subjects who played the GuessWhich game with Grad-CAM explanations. For these games, we manually verify if the answer predicted by the VQA model is correct or wrong.

We find that the mean accuracy of the model is $53.33\% + / - 4.46\%$ (error is the $1.96 * \text{standard error of the mean}$ that corresponds to a 95% CI). As a reference, the human subject guesses the secret image correctly (before explanations), between 29% to 32% of the time (Fig Figure B.2). This demonstrates that accurate answers from the model are not always *useful* to the subject in the context of the game.

While it is important for the VQA model to accurately answer the subject’s question for the secret image, accuracy alone does not reflect the various aspects that are involved in making an accurate guess in the GuessWhich game. Consider the following scenarios:

Incorrect yet informative. Even when the model’s answer is incorrect, it could be infor-

mative in the context of the game. For instance, consider the question, “What is in the image?” for an image that contains a sandwich. The answer “hot dog”, although incorrect, provides a clue regarding the secret image. In another game, the same question is asked for an image of a boat in water under a cloudy sky. The model’s response “overcast” is again incorrect but conveys useful information to the subject.

Binary question bias. A common strategy among subjects is to form a hypothesis regarding the secret image, then (dis)confirm this via yes/no questions. While the VQA model performs well on questions that elicit information about the image, e.g., ‘What is in the image?’, the model is often inaccurate with yes/no questions. Despite receiving feedback that exposes the model’s inaccuracy at the end of every game, most subjects continue their strategy of asking yes/no questions for all their 20 games. The strategy might be a result of subjects’ familiarity with the real-life 20-questions game where players are typically only allowed to ask yes/no questions. This demonstrates a compelling point regarding the effect of existing biases that do not align well with model capabilities.

Accurate yet uninformative. An accurate answer to a binary question that is not perfectly discriminative, might not be a useful in the context of the pool. Consider a pool where a distractor image contains books that are visually salient and the secret image contains books in the background. The accurate answer to the question, “Are there books in the image?” is “yes”. The answer, however, can be misleading to the subject who might expect the model to respond in a pragmatic manner like a human.

B.5.3 Information gain in visual explanations

We identify characteristics of Grad-CAM maps that are correlated with improvement in performance in GuessWhich. Recall that Grad-CAM highlights locations in the secret image that contribute to the AI’s answer. (1(b)). The intensity at a particular location of a heat-map corresponds to its saliency, i.e., the extent to which it contributes to the prediction of the AI’s answer. We study the relationship between high-saliency regions and

Table B.1: Mean fraction (in %) of high salience regions ($\mu_{I>\tau}$) and mean spread of high salience regions ($S_{I>\tau}$) in Grad-CAM heat maps for each game played by a subject. The error terms are the 95% confidence intervals around the mean (**1.96***std. error). The number of games belonging to each category are given in the third column (total number of games = 620). Details regarding $\mu_{I>\tau}$, $S_{I>\tau}$ are described in subsection B.5.3.

Accuracy	$\mu_{I>\tau}$	$S_{I>\tau}$	# games
× before, × after exp.	4.98 +/- 0.34	2.12 +/- 0.12	373
× before, ✓ after exp.	5.72 +/- 0.65	2.50 +/- 0.26	62
✓ before, × after exp.	5.08 +/- 0.85	2.29 +/- 0.36	38
✓ before, ✓ after exp.	4.64 +/- 0.52	1.97 +/- 0.17	147

performance.

High saliency regions. We define a location in a heat-map to have ‘high saliency’ if it exceeds an intensity threshold τ^6 . We denote the fraction of pixels in a Grad-CAM map that is highly salient by $\mu_{I>\tau}$, and find that $\mu_{I>\tau} = 4.98\% + / - 0.25\%$ (mean $+ / - 1.96*$ std. error) across all Grad-CAM maps in our experiments.

We computed $\mu_{I>\tau}$ for each of the following settings – when the subject guessed the secret image correctly before explanations, wrongly before explanations, correctly after explanations and wrongly after explanations. We observe in that the fraction is similar to the overall $\mu_{I>\tau}$ across all Grad-CAM heat maps (around 5% with no statistically significant deviation).

We also computed trends comparing finer-grained gameplay. Specifically, we consider the trend in *each game* and compute $\mu_{I>\tau}$ for each of the four possibilities shown in Table B.1 – when a subject guesses the secret image incorrectly both before and after they see explanations (in the same game), when they guess it wrongly before explanations but correctly after, and so on. Interestingly, we observe that compared to all other games, $\mu_{I>\tau}$ is higher for the set of games where subjects guessed the secret image incorrectly before explanations, but correctly after (row 2 in Table B.1). This difference is significant when

⁶we choose $\tau = 118$ which corresponds to the 95th percentile of intensity across all Grad-CAM maps in our experiments.

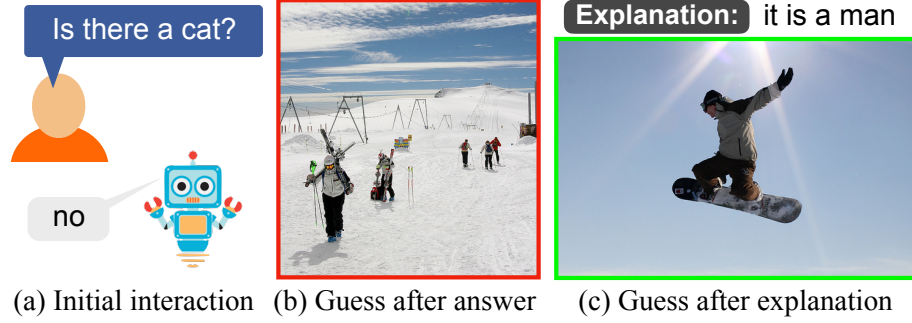


Figure B.3: (a) Sample interaction between human (given the full pool which is not shown) and AI. (b) Image that the subject wrongly guessed as secret image based on interaction. (c) Subject’s correct guess following the text explanation.

compared with games where the subjects’ guesses were either correct or incorrect, both before and after explanations (rows 1 and 4 in Table B.1). Thus, in cases where the subject’s guess before explanations is incorrect, Grad-CAM maps with high $\mu_{I>\tau}$ are correlated with a correct guess after explanations.

Note that the above trends are contingent on the definition of ‘high salience’ and hence, choice of τ . We expect that a small τ (most pixels are considered high salience) or very large τ (very few pixels considered high salience) would not be meaningful since the model’s answer is sensitive to only a limited region in the image. We choose an intermediate value, i.e., 95th percentile of intensities.

Spread of high saliency regions. In this section we investigate the correlation between the ‘spread’ of the high salience regions in a Grad-CAM heat-map and performance in GuessWhich. To compute the ‘spread’, we first binarize the image using the threshold τ (described above) – pixels with intensity above τ are set to 255 and the rest to 0. We then down-sample the image via a max-pooling operation – we use a max-pool kernel of size (28, 28) that produces an $8 * 8$ grid of pixels containing the largest intensities from their respective receptive fields of the original $224 * 224$ image. Finally, we find the number of connected components (CCs) in this binarized, downsampled image, and call it the spread score ($S_{I>\tau}$). A Grad-CAM heat map with larger spread of high salience has larger numbers of CCs, and smaller spread has fewer CCs.

The mean spread score $S_{I>\tau}$ is similar (around 2.1) for the settings when the subject

guessed the secret image correctly before explanations, wrongly before explanations, correctly after explanations and wrongly after explanations. This implies that there are 2.1 distinct high salience regions in a down-sampled $8 * 8$ heat-map.

We also compute the spread score of high intensity pixels for each game (Table B.1), and observe that the trends are similar to the analysis of high salience regions, but less significant. Specifically, $S_{I>\tau}$ for the set of games where the subject’s guess was incorrect before explanations and correct after (row 2), is higher than all other sets of games. This is especially the case when compared with the games where the guesses are either both incorrect or correct both before and after explanations (rows 1, 4 respectively).

Note that trends are likely sensitive to τ and kernel size.

Overall, we find that when the initial guess of the subject is wrong, the more *useful* explanations to the subject have high $\mu_{I>\tau}$. Similar to Lage et al. [65], this raises questions regarding the characteristics of visual explanations that lead to higher interpretability and has implications for designing human *usable* explanations.

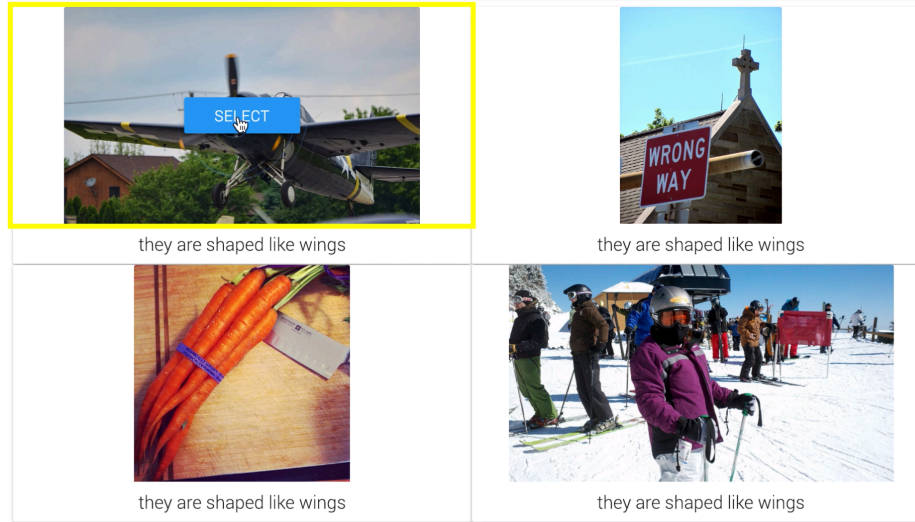


Figure B.4: Screenshot of game interface with text explanations. The subject’s question given the initial pool is, ‘what objects are in the image?’, to which the model’s response is ‘trees’. The text explanation ‘they are shaped like wings’ is relevant to the secret image (likely referring to the airplane) but do not appear relevant to the question or answer.

B.5.4 Information gain in text explanations

Recall from subsection B.4.4 that text explanations significantly improve the guessing accuracy of the human subject. The explanation module is trained with human-written rationales that intend to explain the reasoning behind the model’s answer. E.g., Figure B.3 shows a game where the user’s initial interaction causes them to guess a wrong image. However, the text explanation that reasons about the model’s answer, provides enough information regarding the secret image.

The game in Figure B.1, presents a slightly different type of text explanation. The answer ‘yes’ to the question “Is someone playing tennis?” narrows down the target to the bottom two images. The explanation “she is holding a racket” which seemingly refers to the young girl (bottom image), provides a clue regarding the target.

Occasionally, the text explanation volunteers information regarding the secret image which does not appear to be an explanation or even relevant to the answer or question. Consider the game in Figure B.4. The answer ‘trees’ to the question regarding the objects in the image is likely not discriminative since trees are not the salient object in any image. Further, trees are present in the background of two images, resulting in ambiguity. The text explanation ‘they are shaped like wings’, while irrelevant to the answer, clearly indicates to the subject, the secret image – the top-left image containing an airplane. This raises an interesting and open question regarding the scope of an explanation, and measuring the consistency of a text explanation with the image, question, and answer.

B.5.5 Open-ended question strategy

Subjects in our studies largely follow the strategy of asking ‘yes’/‘no’ questions to identify the secret image. These questions often serve to narrow down the target to two images in the pool, e.g., Figure B.1. In other cases, questions are more discriminative, often focusing on fine-grained details in the scenes or invoking common-sense, e.g., ‘Does it fly?’, ‘Is it a mess?’, etc., which is challenging to the model.

To study the effect of the subjects’ questioning strategy on performance, we experiment with a fixed question strategy. We choose an open ended question that elicits information from the image, ‘What is in the image?’. Based on the VQA model’s answer to this question, the subject guesses the secret image. Following this, as in previous experiments, relevant Grad-CAM explanations are presented.

This questioning strategy results in a performance of $71.00 + / - 6.42$ before explanations which is significantly higher than when subjects freely question the model $29.844 + / - 3.617$ (Figure B.2). Note that the two experiments have identical settings except for the source of the question and scale (this experiment has 10 subjects, compared to 31 subjects in the previous experiment). In contrast, performance after explanations drops to $67.00 + / - 6.65$ while explanations in earlier experiments improved performance.

We identify the potential for significantly better performance with a simple questioning strategy that is possibly more aligned with the capabilities of the VQA model. Since explanations do not improve performance in this setting, leads to exploring questions regarding calibrating a lay person with a AI’s capabilities [39], and the circumstances when explanations may help a lay person.

B.6 Discussion and Future work

We discuss considerations regarding the nature of explanations and their interpretability to a human, the presentation of information in text explanations, and the utility of counterfactual explanations.

Saliency vs. generated text explanations. Apart from being different modalities, Grad-CAM and the text explanations are also qualitatively different in the information that they can convey. Recall that a Grad-CAM heat-map highlights saliency, i.e., *regions in the input* that contribute to a model’s prediction. Grad-CAM (and other saliency-based) are solely a function of the model (*faithful*) and the input. In contrast, a text explanation is a *generated sentence rationale* that is grounded in the regions attended by the VQA model. The natural

language rationale decoded by the explanation module is a function of the input, VQA model and human rationales that the explanation module was trained on. In practice, we observe that the Grad-CAM heat maps convey information regarding ‘where’ the model looked during a prediction while the text rationales also include relevant information about the ‘what’ and ‘why’.

In the context of downstream tasks, an interesting open question is the trade-off between faithful, limited, saliency-based explanations and flexible, expressive, loosely-grounded explanations.

What type of text explanation is more useful? Generating full sentence rationales for a deep network’s decision is a recent, interesting direction. At times, the useful information in a rationale is conveyed by a single word or phrase in the sentence. E.g., for the question, ‘Is there food in this image?’, the explanation is ‘It is a hot dog’. In another instance, for the question, ‘Is there food in the image?’, the explanation is ‘the food is full of vegetables’. In both these cases, the explanations mention a hypernym of the object in the question. Other text explanations refer to the context surrounding the concept. E.g., for the question, ‘Is there books?’, the explanation is ‘there is a lot of books on the desk’ and for ‘is there a boat?’, the explanation is ‘there is a boat on the water’. Others attempt to reason about the answer. E.g., for the question, ‘Is it outside?’, the explanation for the answer ‘yes’ is ‘sky is blue’.

Categorizing the different types of rationales, and evaluating their utility in the context of human trust, performance on a downstream human-AI task, etc. are some of the future directions.

Counterfactual explanations. In the context of conveying information via explanations, the following question arises – is it possible to convey more information by presenting a counterfactual explanation? Specifically, can counterfactual explanations, i.e., explanations for an alternate answer, provide the human subject with additional useful information for a downstream task (such as GuessWhich)? Consider a concrete example where the question

‘Is there food in the image?’ results in the answer ‘yes’, for a secret image containing food and drink. The explanation for this would likely be about the food. The counterfactual answer ‘no’ would likely be about non-food objects like drinks. The additional information regarding the other objects in the image might help the subject better guess the secret image. It would be interesting to answer the question – what is the optimal counterfactual that can convey the most information?

B.7 Conclusion

We evaluate the utility of existing, task-agnostic visual and text explanation modalities in the context of an interactive, collaborative, goal-driven human-AI task, GuessWhich [6]. We evaluate Grad-CAM, a widely used saliency-based explanation modality and a text explanation modality that generates a sentence rationale explaining the model’s prediction. We provide empirical evidence for, and qualitative analyses that attempt to explain, the discrepancy in performance of VQA models on existing metrics on a static dataset and performance in the proposed interactive task. We find that human subjects utilize relevant visual and text explanations to achieve higher performance on the task. In a control experiment with irrelevant (random) visual explanations, subjects effectively disregard the explanations without drop in performance. Overall, subjects obtain more useful information from text explanations to achieve higher performance. Our findings validate the utility of existing task-agnostic visual and text explanations in interactive tasks and identify several research questions that further investigate the mechanisms of explanations in interactive human-AI tasks.

REFERENCES

- [1] D. Yadav, R. Jain, H. Agrawal, P. Chattopadhyay, T. Singh, A. Jain, S. B. Singh, S. Lee, and D. Batra, *Evalai: Towards better evaluation systems for ai agents*, 2019. arXiv: 1902.03570 [cs.AI].
- [2] M. Cogswell, J. Lu, R. Jain, S. Lee, D. Parikh, and D. Batra, *Dialog without dialog data: Learning visual dialog agents from vqa data*, 2020. arXiv: 2007.12750 [cs.CV].
- [3] H. Agrawal, K. Desai, Y. Wang, X. Chen, R. Jain, M. Johnson, D. Batra, D. Parikh, S. Lee, and P. Anderson, “Nocaps: Novel object captioning at scale,” *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, Oct. 2019.
- [4] R. R. Selvaraju, A. Das, R. Vedantam, M. Cogswell, D. Parikh, and D. Batra, “Grad-cam: Why did you say that? visual explanations from deep networks via gradient-based localization,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2017.
- [5] J. Wu and R. J. Mooney, “Faithful multimodal explanation for visual question answering,” *arXiv preprint arXiv:1809.02805*, 2018.
- [6] P. Chattopadhyay, D. Yadav, V. Prabhu, A. Chandrasekaran, A. Das, S. Lee, D. Batra, and D. Parikh, “Evaluating visual conversational agents via cooperative human-ai games,” *CoRR*, vol. abs/1708.05122, 2017.
- [7] *Openml, website - <https://www.openml.org/>*.
- [8] *Codalab, website - <https://codalab.org/>*.
- [9] *Aicrowd, website - <https://www.aicrowd.com/>*.
- [10] *Kaggle, website - <https://www.kaggle.com/>*.
- [11] *Django: The web framework for perfectionists with deadlines. website - <https://www.djangoproject.com/>*.
- [12] *Kubernetes, website - <https://kubernetes.io/docs/home/>*.
- [13] *Amazon elastic kubernetes service, website - <https://aws.amazon.com/eks/>*.
- [14] *Google remote procedure call, website - <https://grpc.io/>*.

- [15] *Container networking interface*, website - <https://github.com/containernetworking/cni>.
- [16] *Fluentd*, *fluentd is an open source data collector for unified logging layer*, website - <https://www.fluentd.org/>.
- [17] *Cloudwatch*, *observability of your aws resources and applications on aws and on-premises*, website - <https://aws.amazon.com/cloudwatch/>.
- [18] *Amazon elastic file system*, *simple, serverless, set-and-forget, elastic file system*, website - <https://aws.amazon.com/efs/>.
- [19] A. Krizhevsky, I. Sutskever, and G. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks,” in *Advances in Neural Information Processing Systems (NIPS)*, 2012.
- [20] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, “Microsoft coco: Common objects in context,” in *European conference on computer vision*, Springer, 2014, pp. 740–755.
- [21] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. L. Zitnick, and D. Parikh, “VQA: Visual Question Answering,” in *International Conference on Computer Vision (ICCV)*, 2015.
- [22] A. Das, S. Kottur, K. Gupta, A. Singh, D. Yadav, J. M. F. Moura, D. Parikh, and D. Batra, “Visual dialog,” *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1080–1089, 2017.
- [23] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, “Squad: 100, 000+ questions for machine comprehension of text,” in *EMNLP*, 2016.
- [24] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. C. Courville, and Y. Bengio, “Generative adversarial nets,” in *Advances in Neural Information Processing Systems (NIPS)*, 2014.
- [25] P. Chattopadhyay, D. Yadav, V. Prabhu, A. Chandrasekaran, A. Das, S. Lee, D. Batra, and D. Parikh, “Evaluating visual conversational agents via cooperative human-ai games,” in *Proceedings of the Fifth AAAI Conference on Human Computation and Crowdsourcing (HCOMP)*, 2017.
- [26] *Amazon Mechanical Turk (AMT)*, Website - <https://www.mturk.com/>.
- [27] A. H. Miller, W. Feng, A. Fisch, J. Lu, D. Batra, A. Bordes, D. Parikh, and J. Weston, “Parlai: A dialog research software platform,” *arXiv preprint arXiv:1705.06476*, 2017.

- [28] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, “Openai gym,” *CoRR*, vol. abs/1606.01540, 2016.
- [29] B. Dai, D. Lin, R. Urtasun, and S. Fidler, “Towards diverse and natural image descriptions via a conditional gan,” *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 2989–2998, 2017.
- [30] D. Li, X. He, Q. Huang, M.-T. Sun, and L. Zhang, “Generating diverse and accurate visual captions by comparative adversarial learning,” *arXiv preprint arXiv:1804.00861*, 2018.
- [31] A. Das, S. Kottur, J. M. F. Moura, S. Lee, and D. Batra, “Learning cooperative visual dialog agents with deep reinforcement learning,” *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 2970–2979, 2017.
- [32] D. Geman, S. Geman, N. Hallonquist, and L. Younes, “Visual turing test for computer vision systems,” *Proceedings of the National Academy of Sciences*, p. 201 422 953, 2015.
- [33] J. Zbontar, F. Knoll, A. Sriram, M. J. Muckley, M. Bruno, A. Defazio, M. Parente, K. J. Geras, J. Katsnelson, H. Chandarana, *et al.*, “Fastmri: An open dataset and benchmarks for accelerated mri,” *arXiv preprint arXiv:1811.08839*, 2018.
- [34] A. Dosovitskiy, G. Ros, F. Codevilla, A. López, and V. Koltun, “Carla: An open urban driving simulator,” in *CoRL*, 2017.
- [35] M. Crosby, B. Beyret, and M. Halina, “The animal-ai olympics,” *Nature Machine Intelligence*, vol. 1, p. 257, 2019.
- [36] P. Anderson, Q. Wu, D. Teney, J. Bruce, M. Johnson, N. Sünderhauf, I. D. Reid, S. Gould, and A. van den Hengel, “Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments,” *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3674–3683, 2017.
- [37] 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.
- [38] A. Chandrasekaran, V. Prabhu, D. Yadav, P. Chattopadhyay, and M. Johnson, “Do explanations make vqa models more predictable to a human?” In *EMNLP*, 2018.
- [39] A. Chandrasekaran, V. Prabhu, D. Yadav, P. Chattopadhyay, and D. Parikh, “Do explanations make vqa models more predictable to a human?” In *Proceedings of*

the 2018 Conference on Empirical Methods in Natural Language Processing, 2018, pp. 1036–1042.

- [40] M. Narayanan, E. Chen, J. He, B. Kim, S. Gershman, and F. Doshi-Velez, “How do humans understand explanations from machine learning systems? an evaluation of the human-interpretability of explanation,” *arXiv preprint arXiv:1802.00682*, 2018.
- [41] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. Lawrence Zitnick, and D. Parikh, “VQA: Visual question answering,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 2425–2433.
- [42] A. Ray, Y. Yao, R. Kumar, A. Divakaran, and G. Burachas, “Can you explain that? lucid explanations help human-ai collaborative image retrieval,” in *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing*, vol. 7, 2019, pp. 153–161.
- [43] M. T. Ribeiro, S. Singh, and C. Guestrin, “Why should i trust you?: Explaining the predictions of any classifier,” in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM, 2016, pp. 1135–1144.
- [44] F. Doshi-Velez and B. Kim, “Towards a rigorous science of interpretable machine learning,” *arXiv preprint arXiv:1702.08608*, 2017.
- [45] K. Simonyan, A. Vedaldi, and A. Zisserman, “Deep inside convolutional networks: Visualising image classification models and saliency maps,” *arXiv preprint arXiv:1312.6034*, 2013.
- [46] D. Erhan, Y. Bengio, A. Courville, and P. Vincent, “Visualizing higher-layer features of a deep network,” 2009.
- [47] J. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller, “Striving for simplicity: The all convolutional net,” in *ICLR (workshop track)*, 2015.
- [48] M. Sundararajan, A. Taly, and Q. Yan, “Axiomatic attribution for deep networks,” in *Proceedings of the 34th International Conference on Machine Learning*, D. Precup and Y. W. Teh, Eds., ser. Proceedings of Machine Learning Research, vol. 70, International Convention Centre, Sydney, Australia: PMLR, Jun. 2017, pp. 3319–3328.
- [49] D. Smilkov, N. Thorat, B. Kim, F. B. Viégas, and M. Wattenberg, “Smoothgrad: Removing noise by adding noise,” *CoRR*, vol. abs/1706.03825, 2017. arXiv: 1706.03825.

- [50] J. Adebayo, J. Gilmer, M. Muelly, I. Goodfellow, M. Hardt, and B. Kim, “Sanity checks for saliency maps,” in *Advances in Neural Information Processing Systems*, 2018, pp. 9505–9515.
- [51] L. A. Hendricks, Z. Akata, M. Rohrbach, J. Donahue, B. Schiele, and T. Darrell, “Generating visual explanations,” in *European Conference on Computer Vision*, Springer, 2016, pp. 3–19.
- [52] L. Anne Hendricks, R. Hu, T. Darrell, and Z. Akata, “Grounding visual explanations,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 264–279.
- [53] D. Huk Park, L. Anne Hendricks, Z. Akata, A. Rohrbach, B. Schiele, T. Darrell, and M. Rohrbach, “Multimodal explanations: Justifying decisions and pointing to the evidence,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8779–8788.
- [54] D. Lewis, *Convention: A philosophical study*. John Wiley & Sons, 2008.
- [55] A. Das, S. Kottur, J. M. Moura, S. Lee, and D. Batra, “Learning cooperative visual dialog agents with deep reinforcement learning,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2951–2960.
- [56] H. De Vries, F. Strub, S. Chandar, O. Pietquin, H. Larochelle, and A. Courville, “Guesswhat?! visual object discovery through multi-modal dialogue,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5503–5512.
- [57] J. Lu, X. Lin, D. Batra, and D. Parikh, *Deeper lstm and normalized cnn visual question answering model*, <https://github.com/VT-vision-lab/VQA.LSTM.CNN>, 2015.
- [58] 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*, 2018, pp. 6077–6086.
- [59] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *CoRR*, vol. abs/1409.1556, 2014.
- [60] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural computation*, 1997.
- [61] J. Lu, J. Yang, D. Batra, and D. Parikh, “Hierarchical question-image co-attention for visual question answering,” in *Advances In Neural Information Processing Systems*, 2016, pp. 289–297.

- [62] 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*, 2015, pp. 91–99.
- [63] R. Hu, P. Dollár, K. He, T. Darrell, and R. Girshick, “Learning to segment everything,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4233–4241.
- [64] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, “Learning phrase representations using rnn encoder-decoder for statistical machine translation,” in *EMNLP*, 2014.
- [65] I. Lage, E. Chen, J. He, M. Narayanan, B. Kim, S. Gershman, and F. Doshi-Velez, “An evaluation of the human-interpretability of explanation,” *arXiv preprint arXiv:1902.00006*, 2019.