Abstract
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. By simplifying and standardizing the process of benchmarking these models, EvalAI seeks to lower the barrier to entry for participating in the global scientific effort to push the frontiers of ML and AI, thereby increasing the rate of measurable progress in this domain. Our code is available at https://github.com/Cloud-CV/EvalAI.

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 [4, 11, 17, 19, 22]. 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 [11] or generating aesthetically pleasing images [15] coupled with the widespread proliferation of AI-driven smart applications, there is an imminent need 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[7]. Instead, to properly evaluate, they should be connected with a human workforce such as Amazon Mechanical Turk (AMT)[2] 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 environment 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 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 fulfills 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.

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 comparison in Table 1. Kaggle[16], CodaLab[8] and AICrowd[1] 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 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 [21], EvalAI integrates with Amazon Mechanical Turk (AMT) [2] 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 [5] 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.

To address this, we have developed the infrastructure to pair AMT users in real-time with artificial agents (for in-sent over to the leaderboard using the message queue. Once the evaluation is complete, the results are sent over to the leaderboard using the message queue.

Human-in-the-loop evaluation: Automatic evaluation of tasks like image captioning \cite{10, 18}, visual dialog \cite{11, 12} or image generation \cite{15} 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 \cite{14}! 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 \cite{11}). 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.

### 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 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.

**Human-in-the-loop evaluation:** Automatic evaluation of tasks like image captioning \cite{10, 18}, visual dialog \cite{11, 12} or image generation \cite{15} 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 \cite{14}! 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 \cite{11}). 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.

### 4 Impact

As shown in Table 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 challenges that EvalAI hosted are CARLA Autonomous Driving challenge \cite{13}, and Animal-AI Olympics \cite{9} 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.

**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 incomplete 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 \cite{23} have sensitive data which cannot be shared with the evaluation platform. Some other AI challenges like CARLA Autonomous Driving challenge \cite{13}, and Animal-AI Olympics \cite{9} 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.

### 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\cite{6}.

### Table 2. EvalAI growth statistics

<table>
<thead>
<tr>
<th>Year</th>
<th># submissions</th>
<th># participants</th>
<th># challenges</th>
<th># page views</th>
</tr>
</thead>
<tbody>
<tr>
<td>2018</td>
<td>12,516</td>
<td>357</td>
<td>11</td>
<td>306,517</td>
</tr>
<tr>
<td>2019 (YTD)</td>
<td>23,357</td>
<td>1,069</td>
<td>25</td>
<td>642,383</td>
</tr>
<tr>
<td>Growth</td>
<td>86%</td>
<td>186.5%</td>
<td>127%</td>
<td>109.6%</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Features</th>
<th>OpenML</th>
<th>Topcoder</th>
<th>Kaggle</th>
<th>AIcrowd</th>
<th>PaSail</th>
<th>CodaLab</th>
<th>EvalAI</th>
</tr>
</thead>
<tbody>
<tr>
<td>AI Challenge Hosting</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✓</td>
<td>✗</td>
<td>✗</td>
<td>✓</td>
</tr>
<tr>
<td>Custom metrics</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✓</td>
<td>✗</td>
<td>✗</td>
<td>✓</td>
</tr>
<tr>
<td>Multiple phases/splits</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✓</td>
<td>✗</td>
<td>✗</td>
<td>✓</td>
</tr>
<tr>
<td>Open Source</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✓</td>
<td>✗</td>
<td>✗</td>
<td>✓</td>
</tr>
<tr>
<td>Remote Evaluation</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✓</td>
<td>✗</td>
<td>✗</td>
<td>✓</td>
</tr>
<tr>
<td>Human Evaluation</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✓</td>
<td>✗</td>
<td>✗</td>
<td>✓</td>
</tr>
<tr>
<td>Environments</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✓</td>
<td>✗</td>
<td>✗</td>
<td>✓</td>
</tr>
</tbody>
</table>

*Image 54x461 to 294x565*
References