Exploring Q-Learning for Adaptive Difficulty in a Tile-based Image Labeling Game

Sofia Eleni Spatharioti
Northeastern University
Boston, MA, USA
spatharioti.s@northeastern.edu

Sara Wylie
Northeastern University
Boston, MA, USA
s.wylie@northeastern.edu

Seth Cooper
Northeastern University
Boston, MA, USA
se.cooper@northeastern.edu

***Abstract*—Participant disengagement in citizen science tasks remains a significant challenge for crowdsourcing platform creators, in their efforts to generate meaningful data and connect with their membership. Reinforcement learning is increasingly used to take advantage of the plethora of available data to learn to sequence tasks for participants. To this end, we extend the reinforcement learning techniques used in Tile-o-Scope Grid, an image matching web game, by introducing an adaptive Q-learning based approach that incorporates participant performance in sequencing the difficulty of levels. We compared our adaptive version against both a previous non-adaptive algorithm, as well as a greedy approach. We found that the adaptive extension outperformed both, in terms of total reward. This work contributes to the growing literature on reinforcement learning approaches applied to citizen science.**

***Index Terms*—citizen science, image labeling, Q-learning, reinforcement learning, adaptive**

I. Introduction

The popularity of crowdsourced image labeling tasks has steadily risen over the past two decades, with more and more organizations and stakeholders turning to the crowd when in need of analyzing high volumes of data. Factors such as low complexity, low cost and high flexibility have spurred on a wide range of crowdsourcing image labeling platforms, with applications ranging from training Machine Learning models and image annotation [1]–[4], to disaster response [5]–[8], and citizen science [2], [9]–[11]. However, organizations may sometimes struggle to maintain high retention levels from participants, as a majority of them become disengaged early on in the task, leaving only a small subset to produce most of the resulting image labels [12], [13].

To mitigate this behavior, a significant body of work has turned to introducing gamification elements to increase motivation and mitigate participant disengagement [14], [15]. In particular, the ESP game [1] remains one of the most successful examples of gamified image labeling, having being able to attract roughly 30,000 players over a period of 11 months, resulting in extremely high precision labels for over four million images. Moreover, task variety has been found to be another effective method for increasing performance [16]–[19]. The plethora of data generated in gamified image labeling tasks offers promising directions for deploying reinforcement learning (RL) based algorithms, with an aim of bringing together gamification and task variety for improving performance [20]–[22].

To this end, we aimed at combining reinforcement learning and gamification, using an image matching web game called Tile-o-Scope Grid [22]. Tile-o-Scope Grid introduces task variety by serving sequences of levels of varying difficulty in completion, as well as meaningfulness in types of datasets. This work builds on a reinforcement learning approach used previously in Tile-o-Scope Grid, which used Q-learning to create sequences of levels, but did not adapt these sequences based on user performance [22]; we refer to the previous approach as *non-adaptive*. In this work we extend the non-adaptive approach by: 1) taking into consideration player performance into the reward and construction of the Q-table, 2) using player data in an online fashion to adapt the Q-table accordingly, beyond its initialization via training, and 3) adapting the remaining levels in a player's generated level sequence at the end of every level, to reflect the changes in the Q-table due to new data entering the system. We refer to this work's new approach as *adaptive*.

We conducted two sets of evaluations of the adaptive Q-learning based algorithm, by comparing it to the previously non-adaptive approach, and a greedy strategy (serving levels of the highest value). Our first set of comparisons was conducted using synthetic data, generated by simulating player actions, based on a set of *user personas*, motivated by work by Holmgård et al. on player personas [23]. This offered interesting insights on the algorithm's performance when a high volume of data is available. For the second set of comparisons, we recruited participants through Amazon Mechanical Turk.

We found that both Q-Learning based algorithms outperformed the Greedy approach, both in terms of total reward, when measured as a combination of contributed labels, difficulty and performance, as well as efficiency (when measured as labels over total time spent on task), and number of levels completed. Moreover, when comparing the two Q-Learning approaches, we found that the extension of the previous Q-Learning based algorithm to adapt to player performance, as well as new data, led to significantly higher total rewards, as well as reward efficiency. We observed highest retention of

We thank all participants, and all reviewers for their feedback. This material is based upon work supported by the National Science Foundation under grant no. 1816426.

978-1-6654-3886-5/21/$31.00 ©2021 IEEE

participants over total reward in the adaptive version, which indicates that such an approach may be able to better engage participants in image labeling tasks.

II. Related Work

A growing body of work has focused on Reinforcement Learning (RL) algorithms to enhance game experience. In the domain of education, such approaches have been deployed in games like Refraction, a puzzle-based game aimed at developing both mathematical and spatial reasoning in students [20]. RL based approaches are developed to order game levels, to mitigate high drop-off rates from students at the beginning of the game. Mandel et al. developed and compared three evaluation approaches for reinforcement learning algorithms used in Treefrog Treasure, an educational fractions game [21]. A multi-objective Reinforcement Learning tutorial planner is utilized in Crystal Island, a game targeted at middle school students about microbiology, to increase learning and engagement outcomes [24]. Interactive deep Reinforcement Learning is used in SanTrain, a serious game for tactical combat casualty care [25]. While our work also aims to create targeted sequences for players using Reinforcement Learning approaches to increase engagement, the primary focus of the tile-based game we use is image labeling instead of education. More specifically, we are particularly interested in applications of this tool in domains such as disaster response, with an emphasis on tracking industrial disasters.

The algorithms designed in this work are based on Q-Learning, an example of a model-free Reinforcement Learning algorithm. Chen et al. [26] developed a Q-learning based algorithm for generating optimal deck builds in Q-DeckRec, a recommendation system for Collectible Card Games (CCGs). Generated deck builds are then explored in their potential for increasing player engagement. Q-DeckRec uses a Multi-Layer Perceptron (MLP) approach to tackle the large state space in CCGs. The state space in Tile-o-Scope Grid is much smaller, which allows us to directly store the Q-table. In this work, we introduce an adaptive extension to the previous Q-Learning algorithm used in image matching games. Q-Learning based algorithms with updates are also explored in the context of real-time fighting games by Andrade et al. [27], though the emphasis of these types of games is on entertainment and not citizen science.

We use game levels that players encounter as actions for our learning algorithm. One of the available actions is built around a system where players can leave pre-set messages for others and read what others have left at specific points of the game. We view this type of level as an "intervention", as no actual labeling through playing is being generated. The benefits of different types of interventions have been previously explored, both in terms of messaging interventions and beyond. Prior work on showing encouraging messages closer to predicted intervention points on Galaxy Zoo has shown potential towards increasing contributions [28]. Interventions in the form of "micro-diversions", and their impact on user retention, have also been explored by Dai et al. [29]. However, the focus of these interventions was on entertainment rather than community building through crowd encouragement, and users had no communication with others throughout the task.

The main influence behind the design of Tile-o-Scope Grid is the game Dots [30]. Similar to Dots, players must connect tiles of the same category in order to collect them. A level ends when all collection requirements have been met. Instead of colored dots, Tile-o-Scope Grid uses a tile grid of images that become labeled as players connect images they believe belong to the same category. Another example of a tile-based game is Befaced [31]. Befaced follows an approach similar to Bejeweled [32], where players must collect tiles of different facial expressions. However, to complete a move, players must perform the corresponding facial expression, which is then captured by the game. Thus, Befaced produces a crowdsourced image set of facial expressions that can then be used in Machine learning algorithms for facial expression analysis systems. Tile-o-Scope Grid does not collect personal information about users and its main purpose lies in citizen science oriented projects, instead of applications in facial recognition or individual tracking.

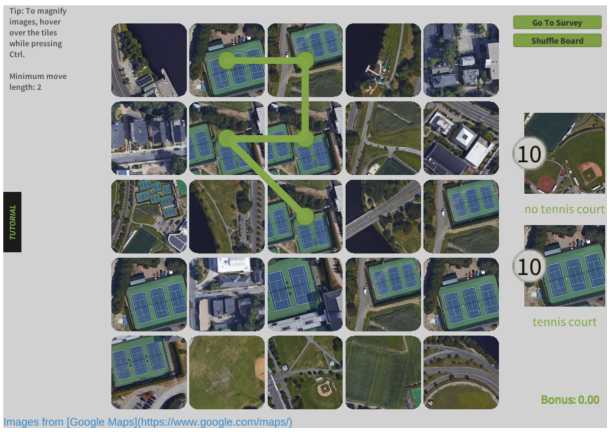
III. Game Setup

A. Tile-o-Scope Grid

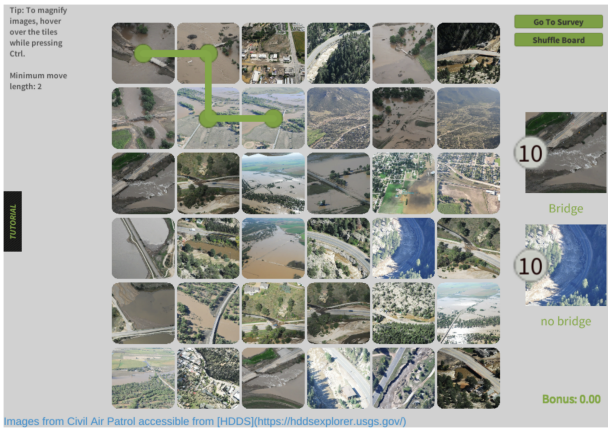
For this work, we used Tile-o-Scope Grid [22], an image matching web game. Tile-o-Scope Grid is built using Unity and has a design similar to the game Dots [30]. The game is structured in levels, with images placed on a grid. The goal is to label a predefined amount of images from each available category. To achieve that, players must connect tiles of images of the same category using a path of non-intersecting lines, in order to collect them. The difficulty of a level can be determined by a variety of factors, such as the size of the grid, number of images required to complete a level, type of dataset and number of categories present. Additional restrictions can be applied, such as limiting the number of moves or time to complete a level, as well as adding block tiles.

Tile-o-Scope Grid deploys an internal mechanism for converting image matches (i.e. connecting images in a line) to image labeling (i.e. assigning a label to each individual image in the match). This is achieved by inserting a percentage of images whose true label, the ground truth, is already known. Therefore, a match is *invalid* if at least two ground truth images in the match belong to separate categories, and *valid* if it consists of any number of ground truth images that belong to the same category, and any number of non-ground truth images. In this case, images with no ground truth are assigned to the unique category determined by the ground truth images. If there is no ground truth in a match, a dialog box pops up asking players to assign the image category. Every valid match reduces the collection requirement of the identified category by the number of tiles in the match. An invalid match induces a penalty by increasing all collection requirements by 1, up to the maximum number required by the level.

For the purposes of this work, we designed three different types of levels, to represent three difficulties on a scale



(a) *Tennis*



(b) *Bridges*

Fig. 1. Examples of the two highest valued different types of levels. (a) Identifying tennis courts. Imagery ©2019 Google, Map data ©2019 Google). (b) Identifying bridges. Imagery publicly available through the U.S. Geological Survey.

(a) Animals

(b) Cairns

Fig. 2. Examples of the lowest valued types of levels. (a) Animal classification, which was the lowest valued label contributing level. (b) Leaving messages on cairns. This type of level served as a break from image labeling, as well as a community building tool among participants.

from easy, to medium to hard based on previous work using the game [22]. All image labeling types of levels had a requirement of 10 tiles of each category:

- Animals (*A*) – easy: Matching images of cats and dogs on a 4×4 grid (Figure 2a). Images sourced from the Oxford-IIIT Pet Dataset, available under a Creative Commons Attribution-ShareAlike 4.0 International License [33].
- Tennis Courts (*T*) – medium: Matching images with tennis courts on a 5×5 grid (Figure 1a). Geo-located aerial images from near the campus area, sourced from Google Maps.
- Bridges (*B*) – hard: Matching images containing bridges, on a 6×6 grid (Figure 1b). Geo-located aerial images sourced from publicly available data of Civil Air Patrol's 2013 Colorado flood imagery, provided by the U.S. Geological Survey's Hazards Data Distribution System [34].

In addition to the above types of levels, we added a separate type of level, called Cairns (*C*), where no image labeling is conducted. Cairns are human-made piles of rocks which are often encountered in long-distance hiking trails. Hikers reaching a certain point in the trail with an existing human-made pile often place their own rock to the top of the pile. The growing pile serves as an indicator of the number of hikers that have reached this far. This type of level is used as a communication tool among players. More specifically, players are asked to leave a message on a "cairn" by selecting one of four options randomly drawn from a list of predetermined messages. Then, their "cairn" is animated as falling on top of a cairn formation of messages other players have left at that point in the game, which the player can read.

We view the purpose of these types of levels as twofold: First, it serves as a break from repeatedly labeling images, which can be often taxing due to the nature of certain datasets (i.e. identifying damage to structures after a natural disaster). These types of breaks have been found to potentially increase productivity in crowdsourced tasks [29]. Second, it serves as a community building tool, as it gives participants a sense that others are also working on the same task, sharing the same path, encouraging them to continue, celebrating reaching certain milestones, or even acknowledging the difficulty and

sharing frustrations. Such tools of community building have been found to positively impact engagement with projects [28]. An example of a Cairn level can be found in Figure 2b.

B. Non-adaptive Q-learning

Prior work on Tile-o-Scope Grid has used a Q-learning based approach for generating the entire sequence of levels for each player at once when they start the game [22]. Training data are initially collected that are used to construct a Q-table of $Q(s_t, a_t)$ values of pairs of states s_t and actions a_t, using the equation:

$$\begin{aligned} Q(s_t, a_t) \leftarrow &(1-\alpha) \cdot Q(s_t, a_t) \\ &+ \alpha \cdot (r_t + \lambda \cdot \max_a Q(s_{t+1}, a)) \end{aligned} \tag{1}$$

where r_t is the reward of taking action a_t from state s_t, α is the learning rate and λ is the discount factor.

States are determined as the history of (up to) the last K difficulties encountered by the player, with an additional *terminal* state X that represents the player quitting. Possible actions a_t are the possible types of levels a player may encounter (A, T, B, C), and reward r_t is set as the number of $tiles_t$ collected in the level, weighted by a difficulty weight w_{a_t}, as well as a performance weight p_{a_t}. The first weight is used to reflect the varying difficulty of collecting tiles of different categories, with tiles from more difficult levels weighted higher. The second weight is used to penalize bad performances. Taking an action that results in the player quitting (moving to the X state) leads to $r_t = 0$.

In the non-adaptive case, once the Q-table is constructed, using only the training data, each subsequent player is served a sequence of levels generated on the constructed Q-table, at the beginning of the game. Neither the Q-table nor a player's sequence is updated as more game data become available after that point. We define **total reward** for a user as the sum of all rewards r_t collected while playing as follows:

$$\text{Total Reward} = \sum_{t=1}^{T} r_t = \sum_{t=1}^{T} tiles_t \times w_{a_t} \times p_{a_t} \tag{2}$$

C. Adaptive Q-learning

In this work, we introduce an adaptive extension of the previous, non-adaptive Q-learning based approach, that differs in the following ways:

- *State Definition:* We expand the definition of state s_t to also include information about the player performance, in addition to the history of (up to) the last K difficulties encountered. Player performance is encoded using three factors (G, B, U), based on the number of mistakes a player makes in a level. The first (G) represents a player making no mistakes, the second (B) represents a player making up to m mistakes, and the third (U) represents a player making more than m mistakes.
- *Q-table Updates:* The Q-table is updated every time new data becomes available, as players complete levels. This means that the generated sequences further adapt as more players come into the game.
- *Sequence Updates:* While in the non-adaptive approach a player is served a level sequence at the start of the game that remains unchanged throughout, the adaptive version updates the remaining player sequence at the end of every level, to account for player performance and changes in the Q-Table that occur as more data become available.

D. Parameters

For both adaptive and non-adaptive approaches, we used a history of up to $K = 2$ difficulties encountered in the state definition. The discount factor λ was set to 0.95 and the learning rate was set to $\alpha = 0.1$. For difficulty weights, we used $w_C = 0.0$, $w_A = 0.5$, $w_T = 1.0$ and $w_B = 1.2$. The C difficulty was given weight $w_C = 0.0$, since no actual labeling is produced when encountering Cairn levels, and the remaining difficulties were weighted such as to reflect the increase in difficulty. These parameters were set based on prior work on Tile-o-Scope Grid [22]. For performance weights we used $p_G = 1$, $p_B = 0.5$, $p_U = 0.2$, to scale the value of tiles collected based on the number of mistakes users are making. User performance weights were chosen to offer maximum reward if no mistakes were made (P_G), and scaled down as more mistakes are made. We followed a similar pattern of 0.5 for the next performance level (p_B), and chose 0.2 for the weight of the worst performance (P_U) instead of 0, to not completely disregard contributions in this case. Performance threshold m was set to 4, based on collected performance data from participants, by looking at the inflection point of the mistakes distribution graph.

Finally, when generating level sequences using a Q-learning approach, instead of picking the highest valued action for the current state based on the Q-table (i.e. $\text{argmax}_{a_t} Q(s_t, a_t)$), the algorithm uses a weighted random selection, using a normalized exponential function (softmax). This allows a small level of exploration, while retaining a strong preference for higher valued actions. To further incentivize exploration and mitigate potential bias towards already explored states, we adapted the sequence generation function to prioritize actions that lead to unexplored states first.

IV. Evaluating Performance

For comparisons, we focused on the following conditions:

- **Q-learn Adaptive (QA)** : Sequences generated using the adaptive Q-learning based algorithm.
- **Q-learn Non-adaptive (QN)** : Sequences generated using the non-adaptive Q-learning based algorithm.
- **Greedy (G)**: Serving only highest difficulty levels (*B*).

A. Simulating Game Data

A pilot round of evaluations was conducted using synthetic data, by simulating different types of users playing the actual game. We simulated the performance of each synthetic player in terms of tiles collected and mistakes made for each level they encountered, as well as the probability of them quitting at

any point. Conducting simulations allowed us to observe how the different algorithms would behave when a bigger volume of player data becomes available, and identify potential issues, as running costs may render recruiting so many users through Mechanical Turk prohibitive.

To get a better sense of how actual users perform on the web game overall, we first ran a HIT on Mechanical Turk collecting data from 197 participants. Each participant was served a randomly generated sequence of types of levels. We used a payment scheme of \$0.10 plus a bonus of \$0.01 for each required tile collected, for a maximum bonus of \$1.9. For example, if a participant completed one level requiring 20 tiles, they were awarded a total payment of $\$0.10 + \$0.01 \times 20 = \$0.3$. We then used the resulting data to design our simulations.

More specifically, we simulated a player completing a level by collecting a randomly generated number of tiles collected, within a given range, based on the type of level completed. To ensure simulated players would be somewhat realistic, we set the minimum and maximum for possible number of tiles collected based on the first and third quartile of tiles collected from the data generated from Mechanical Turk, per type of level. To simulate a player quitting, we used quit rates for each type of level, based on the generated data from the HIT described above.

To simulate the number of mistakes made, and therefore the performance, we created a set of *user personas*, representing different types of players that may play the game. Our personas included:

- *Good*: High probability of making zero mistakes across all types of levels.
- *Bad*: High probability of having a bad performance (more than m mistakes) across all types of levels.
- *Tennis*: High probability of zero mistakes in the *T* difficulty, high probability of bad performance in the *B* difficulty.
- *Quit-Tennis*: Always quits when *T* difficulty is encountered. High probability of zero mistakes on *A* difficulties, high probability of bad performance in *B* difficulties.
- *Quit-Bridge*: Always quits when *B* difficulty is encountered. High probability of zero mistakes on *A* and *T* difficulties.

To train the two algorithms, we used data from the above mentioned HIT to train the non-adaptive version, and data from 50 synthetic users for the adaptive version. Each user was randomly assigned to one of the 5 available user personas, and followed player behavior accordingly. Training the adaptive version of the algorithm consisted of completing sequences generated and continuously updated based on the aforementioned algorithm extension. After the training of the algorithms was completed, we simulated 600 users for each of the 3 conditions (Q-Learn Adaptive, Q-learn Non-adaptive, Greedy), each randomly assigned to a user persona. The median **total reward** per user per condition, calculated using formula (2), was QA:104 QN:100 and G: 86. Looking at the two quit-based user personas, we qualitatively observed that the adaptive algorithm behaved as expected, by serving fewer *T* difficulties to Quit-Tennis users and fewer *B* difficulties to Quit-Bridges users.

	QA	QN	G	Comparisons		
Participants	92	119	71	QA–G	QA–QN	QN–G
Total Reward	156	106	39.6	***	**	**
Total Time (m)	6.37	6.19	17.41	***		***
# Levels	10	11	4	*		**
# Tiles	222	238	225			
Throughput	0.56	0.57	0.23	***		***
Total Reward Throughput	0.33	0.25	0.05	***	***	***
IMI Enjoyment (mean)	5.61	5.81	5.29			**
IMI Competence (mean)	5.79	5.81	4.97	***		***
IMI Effort (mean)	5.05	5.02	5.09			

TABLE I
SUMMARY OF PERFORMANCE METRICS FOR PARTICIPANTS THAT COMPLETED AT LEAST ONE LEVEL PER CONDITION. MEDIANS REPORTED UNLESS NOTED OTHERWISE. **BOLD** INDICATES $p < 0.05$ FOR OMNIBUS TESTS. (***:$p < 0.001$; **:$p < 0.01$; *:$p < 0.05$)

B. Recruiting Participants through Amazon Mechanical Turk

A second round of evaluations was conducted by recruiting participants through Amazon Mechanical Turk. We posted a HIT using the aforementioned payment scheme. Participants were randomly sorted to one of the three conditions (Q-Learn Adaptive, Q-Learn Non-adaptive, Greedy) and could quit at any time and proceed to a post task survey. As we were interested in identifying potential differences in intrinsic motivation among conditions, we used the Enjoyment, Competence and Effort subscales of the Intrinsic Motivation Inventory (IMI) [35] for the post task survey. After the completion of the survey, they received a code for payment. Study procedures were approved by Northeastern University's Institutional Review Board (IRB). We did not collect any demographics from participants.

To train the adaptive Q-Learning algorithm, we first ran a separate HIT recruiting 50 participants and used the adaptive algorithm to order levels. The non-adaptive Q-Learning algorithm was trained on a random subset of 50 users from the initial HIT which served random sequences. To compare the approaches, we ran a comparison HIT that collected data from 342 participants, with 300 submitting codes for payment. For our analysis, we looked at participants that completed at least one level. This resulted in 282 participants. The mean payment rate across all conditions for the comparison HIT was \$11.07/hr.

In addition to the total reward metric, we also focused on the following metrics:

- *Total Time*: The total time spent on the task, determined by the times the first and last actions were recorded.
- *# Levels*: The total number of levels played, including the *Cairn* type of level.
- *# Tiles*: The total number of tiles collected.
- *Throughput*: Tiles collected divided by total time spent on the task. We viewed this metric as an indicator of user efficiency.

- *Total Reward Throughput*: Total reward divided by total time spent on the task.

For statistical analysis, we conducted a Kruskal-Wallis omnibus test to identify initial significant differences. We then performed post-hoc pairwise comparisons using pairwise Wilcoxon rank sum tests with a Holm correction, for any metric whose omnibus test was significant ($p < .05$).

V. Results

A summary of results from recruiting participants through Mechanical Turk can be found in Table I. Our findings are summarised as follows:

Q-learning based approaches outperformed serving only levels of highest difficulty: We found that both Q-Learning based conditions significantly outperformed the Greedy condition in a variety of metrics. More specifically, we observed significantly higher *total rewards* per user, as well as significantly more levels completed. Regarding user efficiency, we observed a significant drop in throughput rates, i.e. total tiles collected divided by total time spent on the task, in users attempting to label images of only the highest valued dataset (Greedy). Although participants spent significantly more time on this condition, this translated neither in increased tile collections (# Tiles), nor total rewards. This is another indication of the increased difficulty of the *Bridges* dataset. We saw that the Q-learning based approaches served more of a combination of easier level difficulties, with the non-adaptive serving mostly Animals, and adaptive mostly a combination of Tennis Courts and Animals. An overview of level distributions per condition can be found in Figure 4.

Adapting player sequences led to higher total reward collections: When comparing the two Q-Learning based algorithms, we found that extending the existing algorithm to adapt to user performance and incoming data led to a significant increase in the total award collected per user. Participants in both conditions completed similar amounts of levels (# Levels) and spent comparable amounts of time (Total Time), with no significant differences detected in these metrics. User efficiency was also comparable (Throughput). However, we did find a significant increase in total reward throughput in the adaptive version, when measured as total reward over time spent on task.

An overview of participant retention rate over the total reward metric, can be found in Figure 3. We observed an earlier and sharper drop-off of participants remaining above a given total reward in the Greedy condition, when compared to the conditions utilizing Q-Learning based algorithms. The behavior observed in this condition is in line with existing literature highlighting challenges of retaining participants in citizen science projects, with a majority exiting the task early in the process, with a small subset of people remaining to contribute most of the work [9]. On the contrary, the adaptive extension we introduced is able to consistently retain more participants, suggesting that such an approach may be more suitable for mitigating disengagement.

Participants' self-reported measures of enjoyment were highest in the Q-Learning conditions: To gauge levels of intrinsic motivation in the task, we asked participants to complete the enjoyment, competence and effort subscales of the Intrinsic Motivation Inventory (IMI) [35], the results of which can be found in Table I. We found a significant drop in self-reported enjoyment in the Greedy condition, when compared to the non-adaptive Q-Learn condition. The adaptive version also scored higher enjoyment levels, though not significantly so, compared to Greedy. Perceived Competence was significantly higher in both the adaptive and non-adaptive Q-Learning approaches, when compared to Greedy. Finally, participants reported comparable effort levels across all conditions. Our results from the IMI questionnaire are another indication of the potential positive impact task variety may have on participants' intrinsic motivation, which may then lead to higher contribution and engagement levels.

VI. Discussion

One of the main differences between the two Q-Learning implementations regards updating the Q-Table and the player sequences. In the previous non-adaptive approach, the Q-Table is generated using only the initial training data and ignores any new data that enter the system. Each player's task sequence is also generated once. By updating the Q-Table as more players play the game (QA), the mechanism has access to a wider pool of player data and may learn over time that some sequences are better than others. On the contrary, the non-adaptive version relies solely on the initial training data and cannot correct itself over time. Updating the Q-Table helps mitigate potential bias from the initial training group. By updating the player sequence at the end of each level, we allow the sequence to adapt to player performance as well, which is not taken into consideration in the state definition in the non-adaptive version. We found that Total Reward was significantly higher in the adaptive version, and highest overall, which suggests that these updates are indeed beneficial. Looking at how the percentage of levels changes if we include these adaptive updates (Figure 4), we observe that the adaptive version learns to serve a reduced number of Animals levels and increased number of Tennis Courts, relative to non-adaptive, indicating that the updates allow it to change its behavior.

The Cairns system was introduced as a rewarding micro-diversion for increasing engagement as a community building tool. It is included as part of the actions that the mechanism picks in order to optimize the sequences it generates. Therefore, the percentage of Cairns levels served is affected by the sequencing mechanism used in each condition. Greedy serves no Cairns by design, as it only serves levels of the highest difficulty. Both non-adaptive (QN) and adaptive (QA) Q-Learning mechanisms have a variety of differences, such as user performance, training, sequence updates, Q-Table updates etc., that affect the serving of Cairns. We observed that the non adaptive version (QN) did not end up valuing Cairns enough to serve them. The updates in the adaptive version (QA) seem to suggest that Cairns become more valuable. We

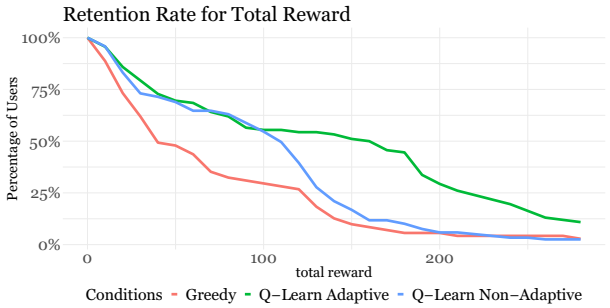


Fig. 3. Retention rate of participants over total reward (i.e. tiles collected multiplied by the relevant difficulty weight and relevant performance weight). The adaptive version achieved, visually, higher participant retention rates over total reward.

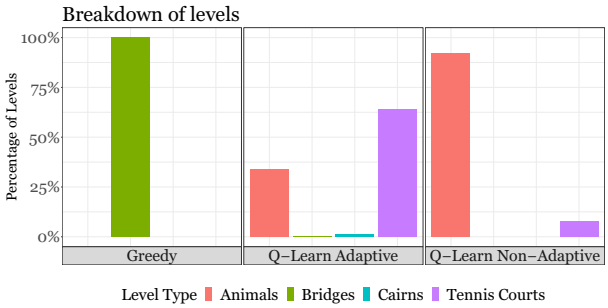


Fig. 4. Breakdown of types of levels served per different mechanism.

found that the Total Reward metric was significantly higher in QA compared to QN, even though Cairns do not add anything to that total reward, as their difficulty weight is 0. This may suggest that despite their zero value in terms of total reward, the Cairns messaging system is potentially valuable for gaining user contributions. However, we did not explicitly ask users to measure this system.

The design of our Q-Learning based algorithms was not aimed towards incentivizing contributions on some datasets over others, but rather to incentivize continued contributions on the tool, by taking into consideration elements such as level difficulty, contributions and user performance. We view all datasets as meaningful in terms of contributions needed, and in different applications (i.e. animal identification, facility tracking, disaster response). Dataset priority, if desired, can be added to the current implementation by introducing an additional weight.

We opted for a payment scheme that allowed players to collect a bonus, depending on their contributions. However, players could quit at any time to get paid, even without playing the game at all. Moreover, the payment scheme was the same across all conditions, therefore we expect changes in engagement to be due to the types of levels players encountered, which are controlled by the difficulty adjustment methods we aimed to evaluate in this work.

VII. Conclusion

In this work, we presented an adaptive extension to the Q-Learning based algorithm deployed in Tile-o-Scope Grid, an image matching web game, that takes into consideration player performance, as well as incoming gameplay data, with a goal of producing more user tailored level sequences. Our evaluation of the new algorithm, using both synthetic data as well as data from participants recruited from Amazon Mechanical Turk, reveal great promise for using simple Reinforcement Learning techniques towards designing effective task variety mechanisms for image labeling in citizen science settings.

The recruitment of participants was conducted via Amazon Mechanical Turk, which is primarily a crowdsourcing marketplace and was therefore limited to some extent, based on recruitment cost. Studies on Mechanical Turk have identified worker motivations that go beyond external rewards [36]. However, exploring other crowdsourcing sources, or targeting a more citizen science oriented user pool can offer more insights on how such task variety mechanisms may be further utilized to achieve secondary goals, such as raising awareness and advocating about specific topics tied to datasets used.

The reward function used for both Q-Learning based algorithms was designed based on several factors, the number of tiles collected, as well as level difficulty and user performance, when viewed in terms of mistakes made in a given level. Investigating other reward functions, such as taking into consideration time spent on the task, or promoting equal contributions across all datasets and how these impact performance is also of interest. Moreover, the state definition of the current Q-Learning approach focused on a history of up to 2 levels encountered, which allowed direct storing of the constructed Q-Tables. A potential future direction concerns taking into consideration levels further in the past, which would effectively increase the state space, potentially requiring some level of approximation, as seen in related work on Q-Learning algorithms for games [26]. Moreover, while in this work we only explored one type of game, Tile-o-Scope Grid, we are interested in exploring the adaptability of our algorithms to other games, which would require setting rewards, weights and the action space accordingly.

One of the types of levels used in the game (Cairns) did not involve image labeling. On the contrary, it served as a form of break, as well as a tool for players to offer messages of encouragement, building on existing work on the benefits of such interventions on crowd engagement [28], [29]. However, these messages were not part of the reward function used in any of the Q-Learning based approaches. Further integrating this secondary stream of data to enhance the design of adaptive task variety mechanisms opens up promising new avenues towards mitigating crowd disengagement, by utilizing concepts such as Self Determination Theory's connectedness aspect [37].

In this work, we deployed a bonus payment scheme per level completed, in an effort to appropriately compensate participants relevant to the effort and time spent on the task. Special considerations need to be made when moving away from paid to volunteer recruitment settings, as such mechanisms may promote keeping people engaged in unpaid tasks. Tile-o-Scope

Grid is geared towards applications in specific citizen science projects, such as investigating and assessing the impact of different environmental disasters. For example, it has been deployed as part of a wider effort in identifying damage to industrial facilities after Hurricane events. We thus view value in engaging volunteers for producing work that is beneficial to a wider collective group.

REFERENCES

[1] L. von Ahn and L. Dabbish, "Labeling images with a computer game," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. Vienna, Austria: ACM, 2004, pp. 319–326.

[2] M. J. Raddick, G. Bracey, P. L. Gay, C. J. Lintott, P. Murray, K. Schawinski, A. S. Szalay, and J. Vandenberg, "Galaxy Zoo: exploring the motivations of citizen science volunteers," *Astronomy Education Review*, vol. 9, no. 1, Dec. 2010.

[3] D. Mitry, T. Peto, S. Hayat, J. E. Morgan, K.-T. Khaw, and P. J. Foster, "Crowdsourcing as a novel technique for retinal fundus photography classification: analysis of images in the EPIC Norfolk cohort on behalf of the UKBiobank Eye and Vision Consortium," *PLOS ONE*, vol. 8, no. 8, p. e71154, 2013.

[4] C.-J. Ho, T.-H. Chang, and J. Y.-j. Hsu, "Photoslap: A multi-player online game for semantic annotation," in *Proceedings of the National Conference on Artificial Intelligence*, vol. 22, no. 2, 2007, p. 1359.

[5] F. Ofli, P. Meier, M. Imran, C. Castillo, D. Tuia, N. Rey, J. Briant, P. Millet, F. Reinhard, M. Parkan, and S. Joost, "Combining human computing and machine learning to make sense of big (aerial) data for disaster response," *Big Data*, vol. 4, no. 1, pp. 47–59, Feb. 2016.

[6] R. Munro, T. Schnoebelen, and S. Erle, "Quality analysis after action report for the crowdsourced aerial imagery assessment following Hurricane Sandy," in *Proceedings of the 10th International Conference on Information Systems for Crisis Response and Management*, 2013.

[7] P. Meier, "Crowdsourcing the evaluation of post-Sandy building damage using aerial imagery," Nov. 2012. [Online]. Available: https://irevolutions.org/2012/11/01/crowdsourcing-sandy-building-damage/

[8] M. Imran, C. Castillo, J. Lucas, P. Meier, and S. Vieweg, "AIDR: Artificial Intelligence for Disaster Response," in *Proceedings of the 23rd International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 2014, pp. 159–162.

[9] T. Sturn, M. Wimmer, C. Salk, C. Perger, L. See, and S. Fritz, "Cropland Capture – a game for improving global cropland maps," in *Proceedings of the 10th International Conference on the Foundations of Digital Games*, 2015.

[10] The SciStarter Team, "Top 18 Projects of 2018 on SciStarter," Jan. 2019. [Online]. Available: http://blogs.discovermagazine.com/citizen-science-salon/2019/01/14/top-18-projects-2018-scistarter/

[11] A. B. Swanson, "Living with lions: spatiotemporal aspects of coexistence in savanna carnivores," Ph.D. dissertation, University of Minnesota, Jul. 2014.

[12] H. Sauermann and C. Franzoni, "Crowd science user contribution patterns and their implications," *Proceedings of the National Academy of Sciences*, vol. 112, no. 3, pp. 679–684, Jan. 2015.

[13] P. G. Ipeirotis, "Analyzing the amazon mechanical turk marketplace," *XRDS: Crossroads, The ACM Magazine for Students*, vol. 17, no. 2, pp. 16–21, 2010.

[14] E. D. Mekler, F. Brühlmann, K. Opwis, and A. N. Tuch, "Disassembling gamification: the effects of points and meaning on user motivation and performance," in *CHI'13 extended abstracts on human factors in computing systems*, 2013, pp. 1137–1142.

[15] B. Fatehi, C. Holmgård, S. Snodgrass, and C. Harteveld, "Gamifying psychological assessment: insights from gamifying the thematic apperception test," in *Proceedings of the 14th International Conference on the Foundations of Digital Games*, 2019, pp. 1–12.

[16] W. S. Lasecki, A. Marcus, J. M. Rzeszotarski, and J. P. Bigham, "Using microtask continuity to improve crowdsourcing," School of Computer Science, Carnegie Mellon University, Pittsburgh, Pennsylvania, Tech. Rep. CMU-HCII-14-100, Jan. 2014.

[17] S. E. Spatharioti and S. Cooper, "On variety, complexity, and engagement in crowdsourced disaster response tasks," in *Proceedings of the 14th International Conference on Information Systems for Crisis Response And Management*, Albi, France, 2017, pp. 489–498.

[18] P. Dai, Mausam, and D. S. Weld, "Decision-theoretic control of crowdsourced workflows," in *Proceedings of the 24th AAAI Conference on Artificial Intelligence*, 2010.

[19] C. J. Cai, S. T. Iqbal, and J. Teevan, "Chain reactions: the impact of order on microtask chains," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2016.

[20] T. Mandel, Y.-E. Liu, S. Levine, E. Brunskill, and Z. Popovic, "Offline policy evaluation across representations with applications to educational games," in *Proceedings of the 2014 International Conference on Autonomous Agents and Multi-agent Systems*, 2014, pp. 1077–1084.

[21] T. Mandel, Y.-E. Liu, E. Brunskill, and Z. Popović, "Offline evaluation of online reinforcement learning algorithms," in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, 2016, pp. 1926–1933.

[22] S. E. Spatharioti, S. Wylie, and S. Cooper, "Using Q-Learning for Sequencing Level Difficulties in a Citizen Science Matching Game," in *Extended Abstracts of the Annual Symposium on Computer-Human Interaction in Play Companion Extended Abstracts*, ser. CHI PLAY '19 Extended Abstracts. New York, NY, USA: ACM, 2019, pp. 679–686, event-place: Barcelona, Spain.

[23] C. Holmgård, A. Liapis, J. Togelius, and G. Yannakakis, "Evolving Models of Player Decision Making: Personas versus Clones," *Entertainment Computing*, vol. 16, 2015.

[24] R. Sawyer, J. Rowe, and J. Lester, "Balancing learning and engagement in game-based learning environments with multi-objective reinforcement learning," in *International Conference on Artificial Intelligence in Education*. Springer, 2017, pp. 323–334.

[25] A. Dobrovsky, U. M. Borghoff, and M. Hofmann, "Improving adaptive gameplay in serious games through interactive deep reinforcement learning," in *Cognitive infocommunications, theory and applications*. Springer, 2019, pp. 411–432.

[26] Z. Chen, C. Amato, T.-H. D. Nguyen, S. Cooper, Y. Sun, and M. S. El-Nasr, "Q-DeckRec: a fast deck recommendation system for collectible card games," in *2018 IEEE Conference on Computational Intelligence and Games*, 2018, pp. 1–8.

[27] G. Andrade, G. Ramalho, H. Santana, and V. Corruble, "Extending Reinforcement Learning to Provide Dynamic Game Balancing," in *IJCAI 2005 Workshop on Reasoning, Representation, and Learning in Computer Games*, 2005, pp. 7–12.

[28] A. Segal, Y. Gal, E. Kamar, E. Horvitz, A. Bowyer, and G. Miller, "Intervention strategies for increasing engagement in crowdsourcing: platform, predictions, and experiments," in *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, ser. IJCAI'16. New York, New York, USA: AAAI Press, 2016, pp. 3861–3867.

[29] P. Dai, J. M. Rzeszotarski, P. Paritosh, and E. H. Chi, "And now for something completely different: improving crowdsourcing workflows with micro-diversions," in *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing*, ser. CSCW '15. Vancouver, BC, Canada: ACM, 2015, pp. 628–638.

[30] Playdots, Inc., "*Dots*," Game [Mobile], 2013.

[31] C. T. Tan, D. Rosser, and N. Harrold, "Crowdsourcing facial expressions using popular gameplay," in *SIGGRAPH Asia 2013 Technical Briefs*, 2013, pp. 26:1–26:4.

[32] Electronic Arts, "Bejeweled," 2001. [Online]. Available: https://www.ea.com/games/bejeweled

[33] O. M. Parkhi, A. Vedaldi, A. Zisserman, and C. V. Jawahar, "Cats and dogs," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

[34] "Hazards Data Distribution System Explorer." [Online]. Available: http://hddsexplorer.usgs.gov/

[35] E. Deci and R. M. Ryan, *Intrinsic Motivation and Self-Determination in Human Behavior*, ser. Perspectives in Social Psychology. Springer US, 1985.

[36] N. Kaufmann, T. Schulze, and D. Veit, "More than fun and money. Worker motivation in crowdsourcing – a study on Mechanical Turk," in *Proceedings of the Americas Conference on Information Systems*, Aug. 2011.

[37] R. M. Ryan and E. L. Deci, "Self-determination theory and the facilitation of intrinsic motivation, social development, and well-being," *American Psychologist*, vol. 55, no. 1, pp. 68–78, 2000.