

Explaining Crowdsourcing Behavior through Computational Rationality

MICHAEL A. HEDDERICH, Cornell University (USA) & Aalto University (Finland), mah499@cornell.edu

ANTTI OULASVIRTA, Aalto University (Finland), antti.oulasvirta@aalto.fi

Crowdsourcing has transformed whole industries by enabling the collection of human input at scale. Attracting high quality responses remains a challenge, however. Several factors affect which tasks a crowdsourcer chooses, how carefully they respond, and whether they cheat. In this work, we integrate many such factors into a simulation model of crowdsourcer behavior rooted in the theory of computational rationality. The root assumption is that crowdsourcers are rational and choose to behave in a way that maximizes their expected subjective payoffs. The model captures two levels of decisions: (i) a worker's choice among multiple tasks and (ii) how much effort to put into a task. We formulate the worker's decision problem and use deep reinforcement learning to predict worker behavior in realistic crowdsourcing scenarios. We examine predictions against empirical findings on the effects of task design and show that the model successfully predicts adaptive worker behavior with regard to different aspects of task participation, cheating, and task-switching. To support explaining crowdsourcer actions and other choice behavior, we make our model publicly available.

1 INTRODUCTION

Crowdsourcing services recruit human users, so-called workers, to participate in self-contained tasks on the web. They have become a crucial tool for many fields. Crowdsourcing has been established as a method for data collection, e.g., in social sciences [3] and human-computer interaction [15, 37]. Crowdsourcing services are also a – sometimes overlooked – key to many recent, data-hungry breakthroughs in machine learning. The popular machine learning datasets ImageNet [11] and COCO [41] contain millions of manually labeled information collected via crowdsourcing sites. The appeal of these services has been their cost-efficiency, which has been essential for scaling up efforts in collecting annotation and human input in general.

Understanding and anticipating the decision processes of crowdsourcers is essential for achieving good results and avoiding issues such as lack of participation [52, 63, 77] and incorrect labels due to lazy or malicious worker behavior [25, 43, 53]. Extensive empirical work has shown that workers on these platforms are guided by a complex set of motivations [16, 19, 46, 54, 56, 59]. They are also constantly taking and adapting their decisions to their environment, e.g., choosing the tasks they want to work on [8, 22, 23, 39, 76] or deciding on the amount of effort they want to put into a task – where they might trade-off minimizing their workload now with work opportunities in the future due to a better reputation [19, 22, 64]. They adapt the amount of effort they invest into tasks and learn from prior experiences. A quote from a worker interviewed in [21] is illuminating: *"I was careless when the Requester was testing Turkers with qualifications, so my score is less. I should have worked hard [...] but I was in a rush."* A better understanding of worker behavior can help task-givers and platform owners design crowdsourcing setups that benefit everyone involved.

Past works have aimed to provide such understanding by explaining, predicting, and simulating workers' decision processes through data-driven and mathematical models. Data-driven approaches learn the behavior of the workers from empirical data [18, 26, 44, 50, 54, 62, 68, 72, 78]. They require the collection of data in advance. Changes in the crowdsourcing setup would require a new collection of data to extract any changes in behavior. In contrast, for mathematical models, the author describes the worker's decision-making process through the definition of the model, or through writing a computational script as part of a simulation [29, 55, 61, 63, 78]. Scekic et al. [63], e.g., assume that a worker follows one of two pre-defined scripts, either working normally or maliciously. These scripted worker models often do not scale up to complex decision-making processes and do not take sparse future rewards into account. They

can also not reflect behavioral adaptation, which a changing crowdworking environment might trigger. Any changes in behavior require the manual definition of a new model.

The framework of computational rationality overcomes these challenges by allowing to model adaptive user behavior from first principles [51]. While this approach has been successfully leveraged for visuo-motor tasks [13, 30, 40, 51], there is only limited work on how people decide between tasks in interactive environments. In this work, we propose a novel theory of adaptive worker behavior on crowdworking platforms based on computational rationality and evaluate if it offers a good model of worker behavior. Until this paper, it was an open question of how to model the essential rewards and formation of beliefs via experience in this domain. Hence, prior to this paper, it was unclear if computational rationality offers a good model of crowdworkers' choices.

We assume the worker is (boundedly) rational and tries to maximize expected personal rewards, such as income or entertainment. Although workers act rationally, because they are not all-knowing, their choices are bounded by their limited knowledge of the world. We exploit this assumption to predict worker behavior by optimizing their choice behavior with reinforcement learning (RL) given an environment and a worker's assumed goals (rewards). Our work exploits modern deep learning based RL that allows modeling more complex and realistic scenarios [1, 9, 69].

As an example, Figure 1 shows the introduction of hidden gold-standard questions on the crowdworking platform to identify negligent or cheating workers. In contrast to existing work, our model adapts its actions to changing environments. While the modeled worker was acting negligently before, now the model adapts the behavior and puts more effort into answering. Our approach explains this change as the rational choice for the worker who does not want to be banned from future work.

We evaluate our new computational model for a set of crowdworking setups and worker behaviors. Our aim here is integrative. Instead of collecting and modeling a single dataset, we want to obtain a model that is adaptive and can explain several known results from a single model. We show that our model's emergent behavior and its predictions closely reflect the findings in the empirical literature on crowdworker behavior. Our model shows trivial adaptive responses, such as a worker choosing a specific task because it pays better. But when several goals and conditions are combined, the model gives rise to more intriguing behaviors, such as the emergence of 'cheating.' The model explains cheating as a rational response of a worker aiming to earn more with less work. We can also explain opportunistic behavior where workers drop out of tasks, for example, due to trying out several tasks because another task might be more beneficial in the long run.

The crowdworking scenario is a special case of behavior where multiple tasks are available, and one must decide how to invest time and energy under partial observability and delayed rewards. Such situations are commonplace throughout: in education (choosing between courses), in working life (choosing between tasks), or in domestic behavior (choosing between activities). To support future work in this area, we present open research directions and make our model platform publicly available.

Our contributions are thus:

- We propose a novel theory of crowdworker behavior based on computational rationality to explain a wide range of behaviors from a small set of first behavioral principles.
- We present a model utilizing modern deep reinforcement learning that can predict adaptive worker behavior in complex scenarios.

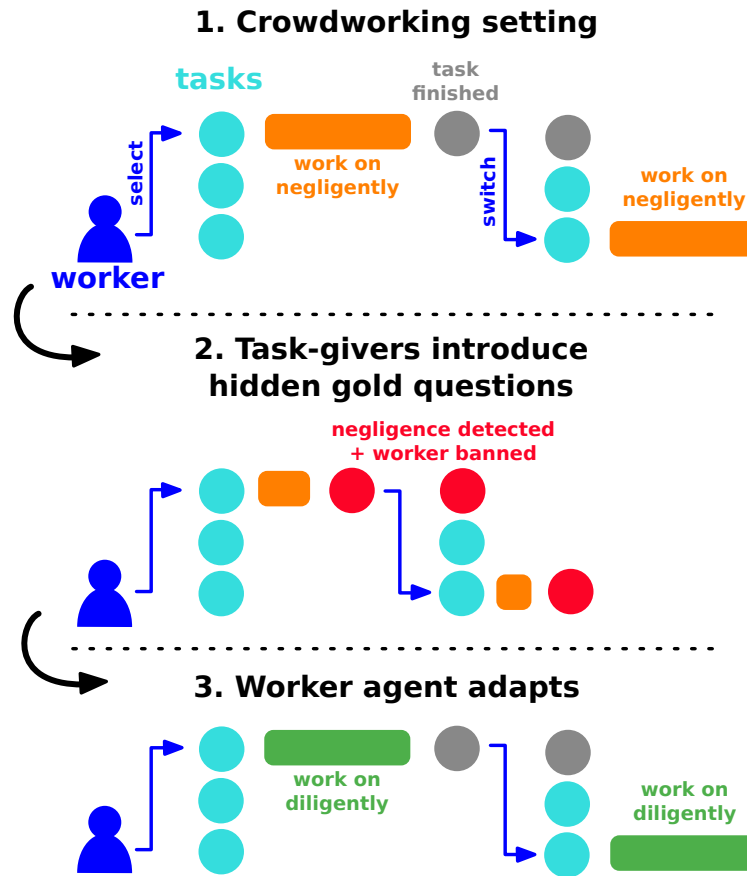


Fig. 1. We utilize computational rationality to explain adaptive answering behavior of a crowdworker: (1) The worker answers a task’s questions negligently. (2) The task-giver introduces hidden gold-standard questions to detect negligent behavior. This results in the worker being blocked from tasks. A scripted, always-cheating worker would continue answering negligently. (3) In contrast, our theory predicts that the worker adapts, stops cheating, and starts answering diligently. This replicates the effect found on real crowdsourcing platforms that cheating deterrents increase the quality of answers. Our theory explains this change in the worker’s behavior as the rational choice for the worker as it maximizes long-term, expected subjective payoffs.

- We show that our theory and model can explain findings from past crowdworker studies, specifically: (1) the influence of payout, effort, and interestingness on task participation, (2) negligent behavior and deterrents of cheating, and (3) task search.
- We identify new research opportunities for leveraging computational rationality to simulate and explain user behavior on crowdworking and other marketplace platforms. We make our computational modeling platform publicly available to foster research in this area.

2 RELATED WORK

In this section, we discuss crowdworking in general, existing crowdworking simulations as well as empirical reports on the behavior of crowdworkers. We also introduce reinforcement learning and the concept of computational rationality.

HIT Groups (1-20 of 821) Show Details Hide Details | Items Per Page: 20

Requester	Title	HITS	Reward	Created	Actions
GenQA Requeste	Question to Question Si...	57,163	\$0.15	1h ago	Preview Quality
Research Reward	Quick Market Research ...	23,774	\$0.02	6m ago	Preview Accept & Work
Lifeng Fan	Answer a survey after w...	19,162	\$0.03	9h ago	Preview Quality
James Billings	Market Research Survey	7,735	\$0.05	6d ago	Preview Accept & Work
Faax, Inc	Specify the Correct Infor...	6,791	\$0.01	3h ago	Preview Accept & Work

Fig. 2. Screenshot of the homepage for workers of MTurk. Different task-givers (“requesters”) list their tasks (“HIT groups”). For each answered question (“HIT”), the worker receives a small monetary reward visible in the list of tasks. A worker can decide which task to accept. Some tasks are blocked (lock icon) as the worker does not qualify for them.

2.1 Crowdfunding

Crowdfunding as a method to collect human input at scale has been used for a wide range of usecases from simple data annotation tasks [11, 41] to more creative tasks such as sketching [74], writing [36] and idea generation [5]. While crowdfunding can be used for complex tasks [35], in this work, we study what is likely the most popular form: microtask crowdfunding. A *question* is one unit of work that can be performed, e.g. an image that needs to be labeled or a multiple-choice question that needs to be answered. A *task* is a set of questions of the same type. The *requester* is offering a task on the crowdsourcing platform and decides on the properties of the task such as pay-out per question. The *worker* decides on which tasks to work, so called pull-crowdfunding [22]. The worker elects tasks on the platform and answers one or multiple questions per task. This setting is used by popular crowdfunding platforms such as Prolific or Amazon Mechanical Turk (MTurk, see Figure 2).

2.2 Empirical Findings on Crowdfunder Behavior

The behavior of humans in crowdfunding scenarios has been studied empirically. Here, we focus on three aspects that are crucial to task-givers and for which the worker needs to consider both short- and long-term consequences: task participation, cheating, and task search.

2.2.1 Task Participation. It has been shown that participation in tasks and the number of questions a worker answers depend on various factors.

Payout. Gadiraju et al. [19] surveyed 490 workers on CrowdFlower with a 5-point Likert scale identifying the monetary payout as the most important factor for workers when they decided which tasks to complete. Mason and Watts [46] studied an image sorting task on Amazon Mechanical Turk (MTurk) where they assigned each of the 611 participating workers to one of four different levels of payout. They found that a higher payout correlated significantly ($p < 0.001$) with a higher average completion rate. In a similar setup, Rogstadius et al. [59] tasked 158 workers on MTurk with counting items on images with three different payment levels (0, 3 and 10 cents). They found that payment had a significant effect on the number of completed questions (two-way between groups ANOVA, $p = 0.034$).

Effort. Gadiraju et al. [19] identified in their aforementioned survey the time it takes to complete a task as another important factor for the workers. Mason and Watts [46] also investigated three different difficulty levels for their image sorting task and found that the number of completed tasks decreased with increasing difficulty.

Interestingness. The third major factor regarding task completion identified by Gadiraju et al. [19] was the task's interestingness and topic. This is also reflected in an experiment in [46], where 320 MTurk workers solved a word puzzle task. The authors found a strong correlation between the number of puzzles solved and the worker's posthoc reported enjoyment of the task ($p < 0.001$). While workers who enjoyed the puzzles the most solved more than 19 of 24 possible puzzles on average, those who enjoyed it only a little completed just 6.2 puzzles on average.

Making tasks more interesting has also been actively used to improve crowdworking results. Feyisetan et al. [16], e.g., used gamification elements for an image labeling task, receiving on average 32 labeled images per worker, compared to 1 image per worker for a control group on CrowdFlower with the same payout. With a similar aim, Qiu et al. [56], proposed using a conversational interface and an involved conversation style to increase worker engagement. For 800 MTurk workers and across 4 different task types, they found statistically significant improvements regarding worker retention compared to traditional interfaces. Making the tasks more engaging resulted in more than twice as many workers answering additional questions. Phrasing the question in a more engaging way also worked for Park et al. [52], resulting in an increase in participation between 15% and 42% in a volunteer setting depending on the method for engagement change.

While both monetary earnings and interest in a task motivate workers, the financial reward seems to be the dominating factor. Surveying 431 workers on MTurk, Kaufmann et al. [32] found that payment ranks highest before any enjoyment-based motivation. Similar results are also found in the previously mentioned studies [19, 46, 59].

2.2.2 Negligent Behavior & Cheating. A prevailing issue of crowdsourcing is the reliability of the workers' answers. Workers regularly try to obtain their payouts by answering questions against the intention of the task-givers. Gadiraju et al. [19] reported, e.g., that they removed 51% of the workers for whom they suspected malicious behavior. This cheating can range from simply choosing answers at random to more elaborate procedures. When requesting text translations to other languages, Pavlick et al. [53] suspected ca. 15% of the workers cheated by providing low-quality content such as random text and another 10% cheated by using the output of the Google Translate service instead of actual, human translations.

To prevent cheating, common practices have been developed to detect and penalize malicious worker behavior. A popular combination is gold-standard questions and reputation systems.

Gold-standard questions are questions where the true answer is known by the task-giver. They are chosen and inserted into the task so that it is difficult for the worker to distinguish them from the real questions [25]. These questions can be used to detect and discard the answers of workers that behave negligently or maliciously. Systems like [43] go even further and actively stop providing answers to users once their estimated reliability is too low.

The gold-standard questions are specific to one task. To take into consideration a worker's past history, several platforms have a *reputation system*, such as the Hit Approval Rate on MTurk. Task-givers check the quality of a worker's answers, e.g. via gold questions, and then rate the worker. A good rating will increase a worker's reputation. Other task-givers can then decide to enforce a minimum reputation for their tasks.

In a study on two crowdworking platforms, Schild et al. [64] found that a low approval rate (i.e. allowing workers with a low reputation) is associated with a higher proportions of dishonest individuals. Kazai et al. [34] ran crowdsourcing

tasks with and without a combination of gold standard questions and minimum reputation level. In most studied subgroups, the mean worker accuracy they reported was lower for the settings without cheating deterrents.

From a worker perspective, Gadiraju et al. [19] showed in their survey that 88% of their workers cared about their reputation level. Han et al. [22] surveyed workers on MTurk and Figure8 and found that workers tried to achieve a good reputation to qualify for future work.

2.2.3 Task Search. Given a pool of tasks, the worker needs to decide on which specific tasks they want to spend their time on. Hara et al. [23] identified task search as an important factor of unpaid work. In [39], the surveyed workers spent on average 27% of their time searching for tasks. Fully assessing a task's properties like interest, difficulty, and time effort, is, however, often only possible by trying out a few of the task's questions. Han et al. [22] found that most workers abandon tasks early after quickly assessing the effort. In the analysis of Kazai et al. [33], over 50% of workers only answered one question. In the log data analyzed by Hara et al. [23], 13% of the questions were not completed by workers.

On the other hand, workers might also get hooked and continue working on a task for a long time, e.g. because they enjoy the task as in the previously introduced word puzzle setting of [46]. This is reflected in a regularly occurring pattern in data collection where for a given task, most workers answer only very few questions and a small set of workers provides most of the answers, as also reported e.g. by Snow et al. [66] and Rogstadius et al. [59].

The depth with which workers search is, however, limited. Chilton et al. [8] collected task data on MTurk and surveyed workers on the platform. They found that the first two pages of task search results were the most important ones. If a task was no longer visible on one of the front pages, participation dropped drastically.

2.3 Explaining Crowdworker Behavior

Several models of crowdworker behavior have been proposed as part of simulation systems. Their goal is to support task designers by simulating crowdworking platforms. Qiu et al. [55] present a simulation system with a web-based interface. In their study, simulated workers participate in a task based on a Poisson distribution, and the quality of their answers depends on three manually chosen probabilistic parameters. In [63], incentive schemes are simulated. The behavior of the workers is set via parameters, and they use two different hand-crafted worker roles, namely normal and malicious users. In [61], the worker can take two actions: registering for a task and then submitting an answer. The authors use it to simulate competing tasks to predict when a task does not get enough participating workers. In the simulation by Zou et al. [78], the worker goes through a fixed behavioral path of searching for a task, working on it, and then submitting it. The authors add a data-driven aspect to their work by collecting behavioral data from workers and tuning the parameters of the script with it.

All of these simulations have in common that the worker's actions are limited to a specific, hand-crafted path of behavior. Changes in the environment can, at the most, influence the probability of an action being taken earlier or later. Any structural changes in the behavior require the designer to create a new worker model.

Jäger et al. [29] take an economics perspective, studying the process of workers joining and dropping out of the labor market. The workers search for and work on tasks and then drop out of the crowdworking platform if they can not cover their cost of living at the end of the month. Again, the worker's behavior is fixed. Yan et al. [71] are interested in the cognitive process of how the worker arrives at a solution, modeling it via an evolutionary process on a lower-dimensional task representation. They study, however, only how a fixed number of workers solves one question.

Data-driven approaches create models from datasets of behavioral worker information. Gadiraju et al. [18] build a worker typology based on collected behavioral traces for two specific tasks from the areas of image transcription and information finding. They leverage this typology to pre-select workers. In [72], the authors build a regression model to predict how a worker perceives the complexity of a task and explain the factors that contribute to task complexity. Posch et al. [54] build a model of crowdworker motivation by selecting important factors from a pool of possible factors. Their selection was based on a large data collection from 10 countries and different income groups. These approaches all require empirical data in advance, and the models are specific to the crowdworking setup in which the data was collected.

In the area of machine learning, some works collect user behavior data and estimate models of specific crowdworkers. This information is then utilized to correct annotated data [26, 44, 68] or to best allocate crowdworker resources [50, 62]. Given the difficulty of estimating complex models from a small set of data, these worker models are, however, on purpose simplistic [26]. The workers are only described by their probability of being incorrect [26, 50], their skill level for different topics [44, 68] or a predefined set of fixed answering behaviors [31]. The estimated behavior is also limited to a specific dataset.

2.4 Computational Rationality

Reinforcement learning (RL) is a subfield of machine learning where a learner maps situations to actions to maximize a numerical reward. For an in-depth introduction into RL, we refer to [67]. Recent advances in deep learning based RL have made it possible to scale up to more complex situations (so-called state-spaces) and thereby allowed studying realistic scenarios beyond simple toy examples [1, 9, 69]. As a theory of intelligent behavior, reinforcement learning is presently studied not only in machine learning but also in economics and cognitive science [27].

Bounded optimality is the idea that an agent takes the optimal decisions given their limited (bounded) available information and computational resources [60]. Following the footsteps of cognitive science [10], there is a growing interest in the field of human-computer interaction to explain user behavior with bounded rational choice, i.e. for user interface control like pointing-and-clicking [13], typing [30] or menu selection [40]. These approaches are synthesized in the framework of computational rationality [51], which assumes that users act in alignment with what benefits them most, taking into account the constraints imposed by their cognitive architecture and their familiarity with the task environment. Combining these first principles with modern reinforcement learning algorithms allows building adaptive models to predict user behavior in complex environments without having empirical data in advance.

While the aforementioned works have focused on visuo-motor tasks, we are interested in utilizing computational rationality to study how users decide between tasks in an interactive environment. Our approach is related to the work by Gebhardt et al. [20] who study task interleaving. While their model can explain the switching behavior of users between different tasks, their tasks are static, and they do not model a marketplace environment with other actors, such as task-givers limiting access to tasks. They are thus lacking core decision aspects of crowdworking platforms.

Individuals acting rationally is an assumption regularly taken when studying marketplaces, e.g., in the auction selection algorithms proposed by Chen et al. [7] and Yu et al. [73]. Our work differs in that we do not present a method that selects workers while taking into account individual rationality but rather use the rationality assumption to explain and predict complex worker behavior.

3 A THEORY OF CROWDWORKER BEHAVIOR

Our ambition is a theory that could explain the findings on crowdworker behavior reviewed in the previous section from a few assumptions about the underpinning mechanisms. We here develop a theory of crowdworker behavior within the framework of computational rationality [51]. When instantiated in a computational model, it allows predicting worker behavior in those situations. For worker behavior, rewards are delayed, which calls for sequential decision-making. To this end we deploy reinforcement learning, which is a theory of how agents can learn to act in complex environments where rewards are delayed. Reinforcement learning also allows us to model bounds to behavior, such as the worker’s limited knowledge of the tasks.

The key assumptions of our theory are:

Sequential Decision Making The worker interacts with the platform in a sequence of actions. At each step, the worker observes the environment and decides how to continue. Rewards might be delayed and only collected at a later step.

Bounded Observations The worker has a limited view of the surrounding environment and can only know what was seen so far. The future is also uncertain and the worker can only make educated guesses (or estimates) on what the result of the next actions will be.

Rationality The worker takes rational actions to the best of their ability. Rationality means that the worker is assumed to pick an action that maximizes the expected subjective rewards. Given the worker’s bounds, however, the actions might not be objectively speaking optimal.

Learning Through Experience The worker learns how to optimize goals by participating on the crowdworking platform and receiving feedback in the form of rewards. The worker learns from the experiences and adapts the behavior accordingly to maximize the payoffs.

These assumptions can be formally described as a sequential decision-making problem, which we propose in this paper.

To approach a workable model beyond these basic assumptions, we need to start filling in the elements of that decision-making problem. The first concerns the worker’s rewards. It has been shown that workers are motivated by both intrinsic and extrinsic rewards [32, 59, 77]. The *monetary reward* is most dominant [19, 32, 46, 59]. Beyond money, different other aspects have been found that drive workers. These include enjoyment of using a certain skill or being creative [32], the fun of solving a task [46] and how engaging the task is [52, 56]. Since these aspects overlap, we group them together as one *interestingness reward* a worker obtains from working on a task. Note that we do not include common reinforcement learning goals such as minimizing the number of actions or penalizing time spent. We purposefully assume a minimal set of goals and expect that secondary goals such as saving time derive automatically from our core goals.

We, thus, arrive at our proposed **core principle for crowdworker behavior**: *the worker maximizes the expected value of actions, measured in payout and interestingness, given his or her constrained knowledge about the crowdsourcing platform. Any specific behavior emerges and can be explained from this assumption.*

4 MODELLING THE WORKER’S DECISION PROBLEM

To operationalize our theory, we build a computational model, which we introduce here. We will first describe the crowdworking setting (Section 4.1). We will then describe the introduced theory and setting from a reinforcement learning perspective (4.2). Sections 4.3 till 4.6 give formal descriptions of these ideas with additional details for reproducibility in the Appendix.

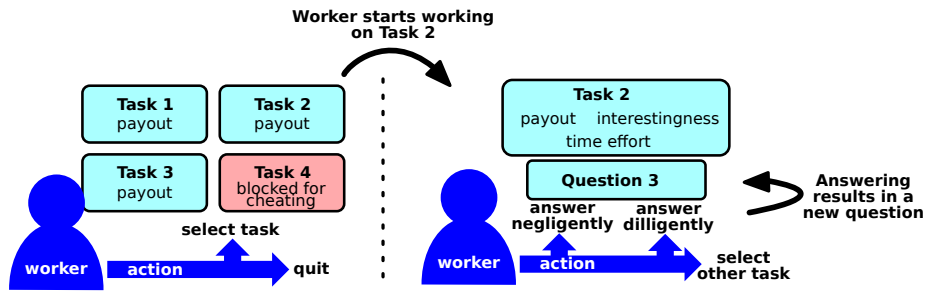


Fig. 3. Visualization of the worker decisions. Left: the worker first needs to decide which task to work on. For some of the tasks’ properties, such as their interestingness, the worker needs to try out the task to assess them. Right: once a task has been chosen, the worker needs to decide how much effort to put into answering each question.

4.1 Crowdsourcing Setting

On a crowdsourcing platform, task-givers provide tasks. Figure 2 shows such a list of tasks on the popular micro-task platform MTurk. These tasks vary in their payout, their interestingness, the time it takes to answer a question [19], the number of questions a task contains [12] and the task’s difficulty [18]. How difficult a task is to solve for a worker also depends on the worker’s expertise in the specific task. A task-giver can add mechanisms to deter negligent or cheating behavior such as introducing gold-standard questions with known answers [25].

From a worker’s perspective, the challenge is that of choice. The model covers two common and interlinked decision problems of a worker:

- *What task(s) should I work on?* In pull crowdsourcing [22], workers decide on which tasks they want to work on, choosing from a set of tasks that is available to them. As long as the worker is given questions, they can independently decide if they want to continue working on a task, switch to a different task, or quit working altogether.
- *How much effort should I put into each task?* Once the worker has decided on a specific task, the worker needs to decide how much effort to put into answering the questions. Workers can try to solve them diligently. The time it takes depends on the question’s effort. Whether the worker manages to give the correct answer then depends on the task’s difficulty. The worker can also choose to act negligently. For multiple-choice questions, they could e.g. always answer with the first or with a random option. This takes less time and likely results in an incorrect answer, and is, therefore, often seen as cheating by the task-giver.

The worker’s concrete decisions in our scenario are visualized in Figure 3. These can be split into two levels. On the higher level, the worker can *select a task* to work on next. Alternatively, the worker can also *quit* working. On the lower level, for a specific task, the worker can decide how to answer. *Answering negligently* is fast but likely produces a wrong answer. Taking the question properly into consideration and *answering diligently* takes more time but produces an answer that actually depends on the worker’s skill. Workers have a limited overall time budget and stop working when out of time. The worker’s actions are guided following the core principle established at the end of Section 3.

4.2 Reinforcement Learning Overview

We model the worker as an agent in a reinforcement learning framework, as visualized in Figure 4. The crowdworker is, in this case, the agent while other parts of the crowdsourcing platform, such as task-givers and tasks form the

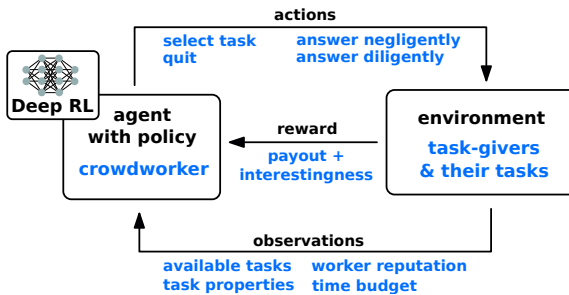


Fig. 4. Visualization of the reinforcement learning setup for modeling a worker as an agent in a crowdsourcing environment.

environment. The agent observes part of the environment, such as the number of available tasks or the tasks’ payouts. Other information from the environment might not be visible (such as measurements against cheating) or only visible once the agent interacts with the environment (such as the interestingness factor of a task).

The agent interacts with the environment through a fixed set of actions. These capture the two levels of decision: selecting a task or quitting; answering negligently or diligently. When the agent performs an action, this changes the environment. Answering a question can make, e.g., a new question available. The agent also receives a reward. Answering a question diligently provides a reward that combines both a monetary payout and an interestingness factor. Answering negligently only provides the payout since the worker agent does not properly experience the question.

The agent learns a so-called policy that tells the agent in each situation the actions it needs to take to increase its rewards. The policy defines the behavior of the agent. The policy is learned during the training phase by interacting with the environment and learning from the rewards it obtains. The learning of the policy also allows the model to adapt its behavior when the crowdsourcing environment changes, e.g. when more tasks become available or when a method to deter cheating is introduced.

4.3 Formalization of the Worker’s Decision-Making Problem

We now formalize our model. Further details for reproducibility are given in the Supplementary Material. Our model is based on a Markov Decision Process (MDP) [67]. The MDP consists of a tuple $\langle S, A, T, R \rangle$ with a set of states (S), a finite set of actions (A), environmental transition dynamics (T), and rewards (R). The crowdsourcing worker is the acting agent in this setting while the tasks and task-givers are part of the environment. At each timestep, the agent can take an action $a \in A$ to interact with the environment, e.g. providing an answer to a question. This will cause the environment’s state to change from $s \in S$ to $s' \in S$ with a probability $T(s, a, s') = p(s'|s, a)$. The behavior of the task-givers is modeled as part of these transitions, e.g. deciding to ban a user for cheating. An episode ends when the agent decides to quit or runs out of time.

The agent has access to some of the task’s properties, e.g. its payout. The agent’s information is, however, limited and it does not know, e.g., the specifics of the anti-cheating measures. We, therefore, split the state $s \in S$ where $s^v \subset s$ is visible to the agent (observations) and $s^h = s \setminus s^v$ is not.

The reward function R models the worker’s extrinsic and intrinsic reward. It specifies the scalar reward $r_t \in \mathbb{R}$ the agent receives in timestep t after it has performed an action a_t and the environment has transitioned from s_t to s_{t+1} . We assume that the agent acts rationally given its restricted knowledge of the environment and tries to maximize its

long-term rewards. The agent’s actions are chosen based on a policy π . An optimal policy π^* maximizes the expected cumulative discounted reward

$$\pi^* = \operatorname{argmax}_{\pi} \mathbb{E}_{\pi} \left[\sum_{t=0}^{\infty} \gamma^t r_t \right], \quad (1)$$

where γ is a discount factor to trade-off immediate versus long-term rewards.

Given the complexity of the crowdsourcing scenario, it is non-trivial to find the optimal policy. Therefore, we train the agents using reinforcement learning to obtain a policy π that approximates π^* . The policy describes the worker’s behavior. The agent’s learned policy is specific to a certain crowdsourcing setting. This gives us the possibility to simulate different environments and evaluate the agent’s behavior without the need to manually adapt or modify the worker. Below, we will see how the simulated workers adapt their behavior to changes in the crowdsourcing scenario in a realistic way.

4.4 Task-Givers & Tasks

The task-givers provide the tasks \mathcal{T} on the crowdsourcing platform. Each task $\tau \in \mathcal{T}$ has a set of properties:

- $\rho_{\tau} \geq 0$, the payout the worker obtains for solving a question,
- $\epsilon_{\tau} > 0$, the time needed to solve a question,
- $p_r \in [0, 1]$, the probability of being correct when answering negligently, and
- n_{τ} , the number of questions the task consists of.

Two additional properties of the task are specific for each worker w :

- $e_{\tau,w} \in [0, 1]$, the worker’s expertise in the task,
- $l_{\tau,w} \in [-0.5, 0.5]$, the interest the worker has in the task.

When all of a task’s questions have been answered, the task becomes inactive.

If not specified otherwise, we use 5 task-givers and each task-giver provides one task in each episode. The properties of the tasks are sampled from beta-distributions in each episode with the distributions being specific to the task-giver. A task-giver might e.g. pay on average higher payouts. In this case, their beta-distribution mean is shifted compared to the other task-givers. Further details on the distributions are given in the Supplementary Material.

The payout is visible to the worker from the beginning (i.e. $\rho \in s^v$). It is often challenging for a worker to assess from the outside how interesting, difficult, and time-consuming a specific task is. Therefore, ϵ_{τ} , $e_{\tau,w}$ and $l_{\tau,w}$ only become visible once the worker has looked at and interacted with the task, by selecting it and submitting an answer to the first question. The worker also knows how many questions they have already answered for a task and whether they can continue working on the task. The other properties are hidden from the worker.

4.5 Workers

Each worker agent w has two properties, λ_w^{ρ} and λ_w^l , how much it values the monetary and interestingness reward respectively. The set of actions A the agent can perform consists of the following:

- Task Selection Actions
 - *select task*, the agent selects a task $\tau \in \mathcal{T}$. The agent can switch tasks at any time.

- *quit*, the agent decides to stop answering questions. This ends the simulation’s episode, i.e. we go to the end state.
- Task-Specific Actions
 - *answer diligently*, the agent tries to answer the current question of the selected task properly. Based on its expertise, the agent provides the correct answer in the task with probability $e_{\tau,w}$. Otherwise, the agent does not know how to solve the question and the probability of still being correct is small (p_r).
 - *answer negligently*, the agent does not follow the task-giver’s instructions properly and cheats. The probability of still providing the right answer, e.g. by chance clicking the right option in a multiple-choice question, is again small (p_r).

For *answer diligently*, a question of task τ , the agent receives the payout and the interestingness or enjoyment reward

$$r = \lambda_w^p \rho_\tau + \lambda_w^l \iota_\tau . \quad (2)$$

For *answer negligently*, the reward is limited to the monetary part.

Each worker agent w has a time budget $b_w \in s^v$ that defines how long it can do crowdworking. The action *answer diligently* reduces the budget by ϵ_τ . Answering negligently reduces the time budget by a fixed amount $c_{negligent}$ which is usually much smaller than ϵ_τ . Selecting a different task takes a fixed time c_{select} . When the time budget is depleted, the episode ends.

4.6 Policy Estimation

Our simulation’s concept is independent of the specific reinforcement learning algorithm. To be able to scale up our approach to the complexity of our crowdworking scenario, we chose a machine learning-based method. In our default scenario, using the algorithms Advantage Actor Critic [70] and Deep Q Network [49] evaluated over $1k$ episodes only achieved a mean reward of 10.7 and 12.0 respectively. Quantile Regression Deep Q-Network [9] better approximated the optimal policy and the worker agent obtained a mean reward of 49.3. We, therefore, chose the latter algorithm for all experiments. The simulation is implemented in Python 3, using OpenAI Gym [4] and StableBaselines3 [58].

5 EVALUATION AGAINST EMPIRICAL FINDINGS ON CROWDWORKER BEHAVIOR

For a suitable sequential model of human decision-making, the worker agents should develop a similar behavior to their human counterparts given the same environment. Our aim here is integrative. Instead of collecting and modeling a single dataset, our objective is to develop a single, adaptive model that is capable of explaining several known results. For evaluation, we thus extracted eight effects from the empirical literature (as discussed in Section 2.2) to verify that our model replicates crowdworker behavior in a realistic way. These effects were chosen as they are crucial to task design and they involve the worker reasoning over the short- and long-term effects of their actions:

A Task Participation

- 1 Higher payout increases task participation.
- 2 Higher time effort decreases task participation.
- 3 More interesting tasks increase participation.
- 4 Payout is dominant over interest in regards to increasing participation.

B Negligent Behavior & Cheating

- 1 Workers cheat.

2 Gold questions and reputation systems prevent cheating.

C Task Search

1 Workers search for tasks.

2 The length of the task search is limited.

For choosing which task to work on next (task participation), it is often sufficient to pick the current best option. When considering cheating, however, one needs to take into account long-term effects, such as being banned from work. Task search exhibits more subtle long-term effects. Like in the idiom “a bird in the hand is worth two in the bush,” working on a non-optimal task now might be preferable over wasting time hoping for a better one. Our goal for all three aspects is to validate whether the same, human-like behavior emerges from our agents as reported in the literature on crowdsourcing. We will further study the adaptive nature of our model in Section 6.

Simulating the exact environment from a reported study is often not possible as many aspects are not detailed and might not even be measurable in a study, such as the properties of competing tasks. We, therefore, do not compare our results in a numeric form but instead focus on the qualitative trends. This also shares a motivation with the simulation by Scekic et al. [63] who are interested not in the absolute numbers but the impacts that crowdsourcing design decisions have.

5.1 Task Participation

For task-givers, it is important to understand which factors influence the willingness of workers to participate in their tasks.

A.1 Higher payout increases task participation. In the previously introduced literature (Section 2.2.1), it has been established that a higher payout results in a task being worked on more. We now evaluate if this effect can be reproduced by our model. We expected that an increase in payout for a task results in the simulated worker answering more questions for it, preferring this task over its lower-paying competitors.

We measured this by increasing the mean payout for a target task-giver from 0.5 to 0.9 monetary units while keeping the other task-givers’ payout distributions unchanged. The worker should adapt accordingly and prefer the tasks of the target task-giver. We measured the percentage of questions answered by the agent for the target task-giver in comparison to all. In Figure 5, one can see that the agent chose to answer more questions for the task-giver with a higher mean payout. It strongly deviated from the theoretically expected amount of work each task-giver would obtain (in the long run) if all task-givers were equal. The participation for the other, unchanged, task-givers went down accordingly. Thus, our agent behaved realistically, clearly preferring tasks with a higher payout.

A.2 Higher time effort decreases task participation. It has also been reported that a task’s effort is an important factor for the worker and that more difficult tasks have been worked on less. We, therefore, expected that an increase in a task’s time effort in our simulation should result in a measurable preference of the agent to not work on this task. Similarly to the previous payout experiment, here, we increased the mean time effort of a target task-giver. Figure 6 shows that the agent successfully replicated the empirical findings, working fewer tasks for the task-giver with higher effort.

The standard error is comparatively high as the task parameters were drawn from beta-distributions and the agent had to consider the complex trade-off between varying payouts, interestingness, and efforts of the target as well as the

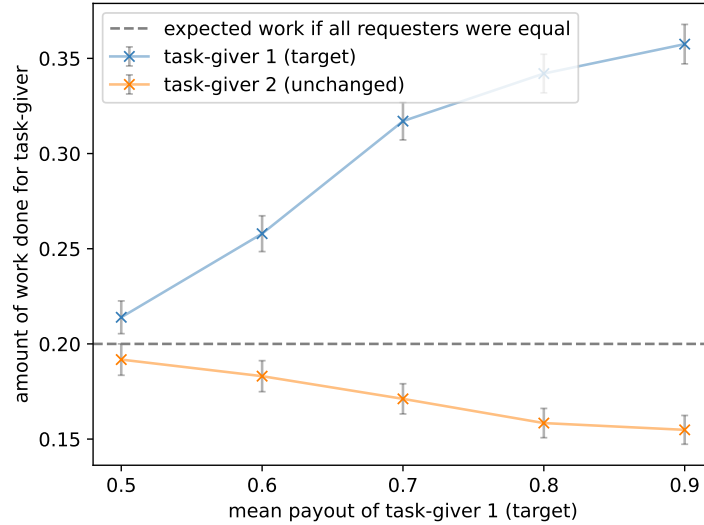


Fig. 5. *Higher payout increases task participation.* An increase in mean payout results in the agent working more on these tasks. The mean payout of the target task-giver was shifted by setting the first parameter of the payout’s beta-distribution from 10 to 15, 23.33, 40, and 90 resulting in mean payouts of 0.5 to 0.9. The parameter of the payout distribution for all other task-givers is kept unchanged by the default value 10 (plot visualizes results for task-giver 2). The dotted line shows the theoretically expected participation if all task-givers were worked on identically. The grey bars show the empirical standard error over 1000 episodes. The model replicated human crowdworker behavior, preferring to work on tasks with better payout (target) compared to competing tasks.

competing tasks. Note that the agent could not see which task belongs to which task-giver and could, therefore, not take shortcuts by learning a task-giver’s underlying task property distribution.

The behavioral effect of time-consuming tasks being less appealing can be explained by the task effort’s indirect influence on the monetary income, and therefore the agent’s reward. The higher the effort for a question, the less time the worker can spend answering more questions or working on other tasks.

A.3 More interesting tasks increase participation. The third studied factor that affects participation is the task’s interestingness. It has been shown that workers are motivated by tasks they are interested in. To evaluate this, we increased the mean interestingness of the target task-giver. In Figure 7, one can see that we successfully replicated the human behavior. The participation increased for the target task-giver. The agent decided to work more on tasks that are of higher interest to it.

A.4 Payout is dominant over interest in regards to increasing participation. Across the literature, it has been shown that for many crowdworkers the payout is more important than the task’s interestingness. We simulated this by introducing to the default setting a task-giver with a higher mean payout and a task-giver with a higher mean interestingness for their tasks. Figure 8 shows that indeed the increased payout clearly dominated as a factor that increases participation. Tasks that paid more were clearly preferred while tasks from task-givers with unchanged mean payout are less and less worked on. Increasing the interestingness of the tasks helped to dampen somewhat the drop.

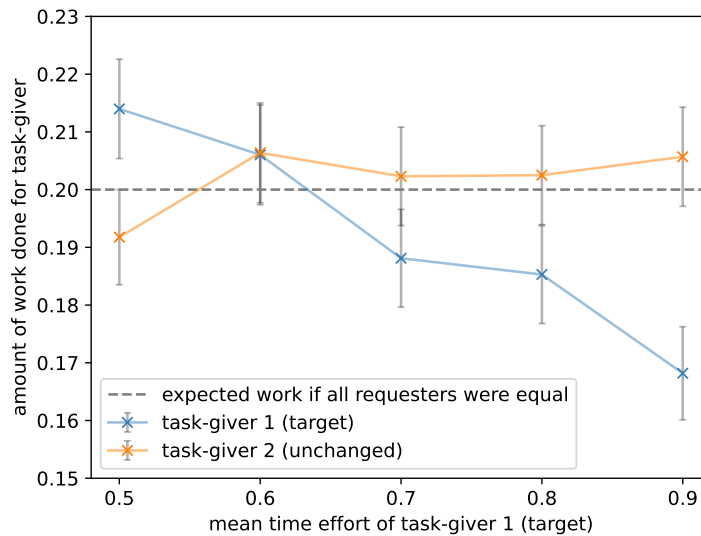


Fig. 6. *Higher effort decreases task participation.* An increase in mean effort results in the agent working less on these tasks. The mean effort of the target task-giver was shifted by setting the first parameter of the time effort’s beta-distribution from 10 to 15, 23.33, 40, and 90 resulting in mean time efforts of 0.5 to 0.9. The parameter of the effort distribution for all other task-givers is kept unchanged by the default value 10 (plot visualizes results for task-giver 2). The dotted line shows the expected participation if all task-givers were worked on identically. The grey bars show the standard error over 1000 episodes. The model replicated human crowdsourcing behavior, avoiding tasks that require more effort.

5.2 Negligent Behavior & Cheating

Negligent workers are a recurring issue in real-world crowdsourcing, e.g. when workers answer multiple choice questions by selecting an arbitrary option instead of properly thinking the questions through. Task-givers often classify this behavior as cheating.

B.1 Workers cheat. The worker in our model has the option to either answer the questions diligently (spending more time coming up with an answer) or to answer negligently (and therefore quickly). We expected the same cheating behavior to emerge that is so problematic in real-life crowdsourcing. In Figure 9, one can see that in a setting without penalization for cheating, our agent, like its human counterparts, also chose to cheat, answering nearly all questions negligently.

B.2 Gold questions and reputation systems prevent cheating. Given the popularity of gold-standard questions and reputation systems and the evidence that these result in higher quality answers, we expected our simulated workers to also change their behavior in the presence of these systems, to start caring for their reputation, and to be thus deterred from answering in a negligent way.

To evaluate this, we introduced hidden gold standard questions with known answers in the task and a reputation system. There was a 10% chance that a task’s question is a gold question. If the agent answered 3 gold questions incorrectly, it was banned from a task. For the reputation system, the worker started with a reputation of 1 which was

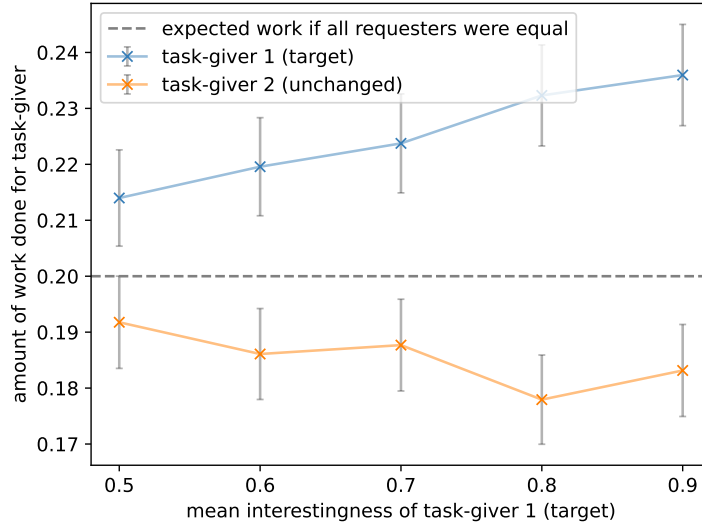


Fig. 7. *Higher interestingness increases task participation.* An increase in mean interestingness results in the agent working more on these tasks. The mean interestingness of the target task-giver was shifted by setting the first parameter of the shifted interestingness’ beta-distribution from 10 to 15, 23.33, 40, and 90 resulting in mean interestingness rewards (or enjoyment) of 0 to 0.4. The parameter of the interestingness distribution for all other task-givers is kept unchanged by the default value 10 (plot visualizes results for task-giver 2). The dotted line shows the theoretically expected participation if all task-givers were worked on identically. The grey bars show the standard error over 1000 episodes. The model replicated human crowdworker behavior, preferring to work on tasks that are of higher interest to it (target).

reduced by 0.05 for each incorrectly answered gold question and increased by the same amount for each correctly answered gold question (with a minimum of 0 and a maximum of 1).

Figure 9 shows that we replicated the same effect as observed in real-life crowdworking: these measurements deter from cheating. The agent answered most questions diligently even though this took more time. The number of negligently answered questions dropped drastically to a small percentage the agent could still get away with. In Section 6, we will explore the topic of cheating further and explain the reasons behind it.

Comparison to other models. The aspect of cheating crowdworkers highlights the adaptive nature of our model, where other models fall short. Here, we compare the behavior of different models in crowdworking environments with and without cheating deterrents.

The models discussed in Section 2.3 have a fixed path of behavior, and while some of the modeled workers can act in a probabilistic way, their possibility to adapt is minimal. We designed the models “diligent worker” and “cheating worker” in this experiment after the two model behaviors described in [63]: one acts diligently, trying to answer all questions to the best of their abilities; and one answers negligently, therefore cheating on all questions. For both models, the worker selects the tasks that pay best at each time point and finishes once all tasks are closed or time runs out.

Figure 10 shows the modeled workers’ average reward. The cheating worker performs well in an environment without cheating deterrents but obtains only a minimal reward when gold questions and a reputation system are introduced. The diligent worker performs well under cheating deterrents but is sub-optimal (from a worker perspective)

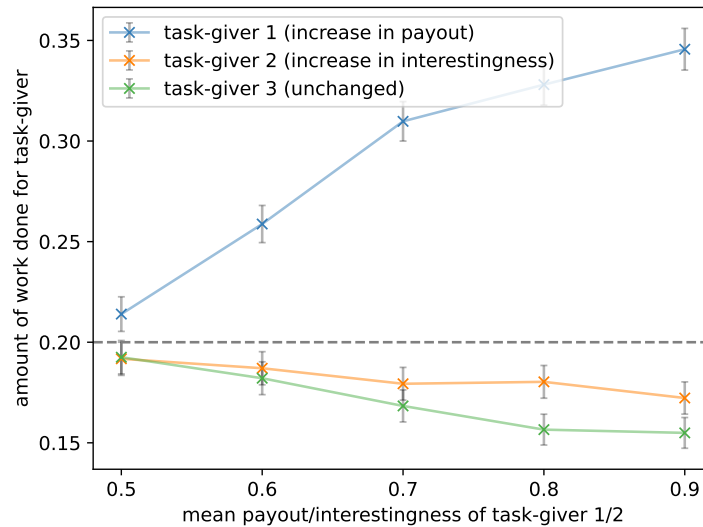


Fig. 8. *Payout is dominant over interest in regards to increasing participation.* For one task-giver, the mean payout was increased and for another the mean interest value. This was performed by setting the first parameter of the beta distribution from 10 to 15, 23.33, 40, and 90 resulting in mean payout rewards of 0.5 to 0.9 and mean interestingness rewards of 0 to 0.4. The parameters of the distributions for the other three task-givers were left unchanged (plot visualizes task-giver 3). The dotted line shows the theoretically expected participation if all task-givers were worked on identically. The grey bars show the standard error over 1000 episodes. The model replicated human crowdsourcing behavior, considering the payout more important than the interestingness of a task.

in environments without cheating deterrents, obtaining less than optimal rewards. By the nature of their design, these models can not adapt to changes in the environment. In contrast, our adaptive worker model can adjust to the change of the crowdsourcing setting without the need to change the model manually. It changes its strategy (see Figure 9), obtaining close to the optimal reward in each situation.

We also study a myopic or greedy worker model. There, the worker chooses as the next action the one that gives the highest reward (normalized by the time effort). It is myopic in that the agent can not take into account future rewards. This results in a behavior of only answering negligently and, therefore, in the scenario with cheating deterrents in a suboptimal global reward at the end of the episode. In contrast, one of the core principles of our model is that it can consider future rewards. This results in only our model in this experiment behaving and adapting like real human crowdsourcers.

5.3 Task Search

In Section 2.2.3, we discussed that crowdsourcers can often not directly identify all of a task’s properties, such as difficulty or effort. They try out different tasks in search of the ones they would like to work on. Given that the agent in our setting does not have full information on all available tasks, we expected it to also exhibit a searching behavior.

C.1 Workers search for tasks. In our simulation, only the payout was visible to the worker from the beginning. For all other properties, the worker needed to start and try out the task to assess them. For human-like behavior, we expected

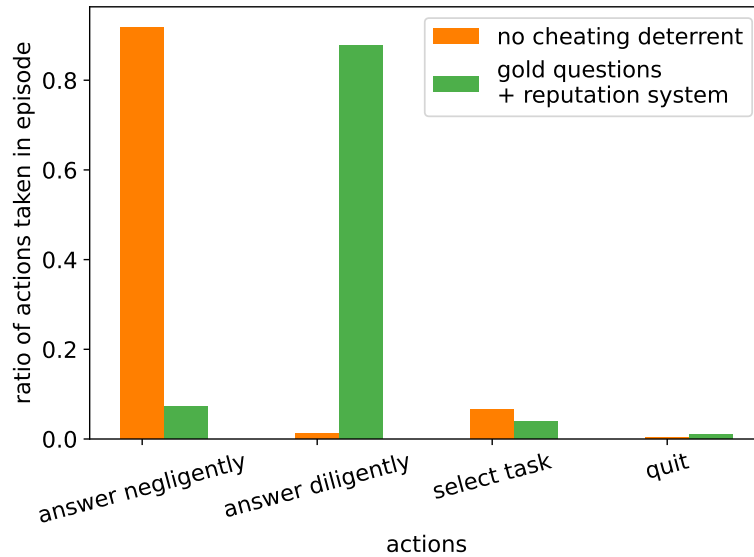


Fig. 9. Ratios of actions taken by the agent. Orange bars are without measurements that deter cheating and green bars with 0.1 gold questions and 0.9 minimum reputation level. Mean over 1000 episodes. Like it is regularly observed for actual crowdworkers, the agent often answered negligently, i.e. cheated. Introducing gold questions and a reputation system, which are popular in real crowdworking settings, also helped here to deter cheating.

our agent to start working on different tasks, abandon early those which it does not like, and continue working on those it prefers.

In the previous experiments, we saw that the agent’s preferences rely on a combination of multiple factors including payout and interestingness. In order to have an experimental setting where there is a clearly preferable task, we set the parameters for all tasks to the same fixed values (instead of distributions) and just differentiated our target task by a higher interestingness. This allowed us, from the outside, to easily identify in each episode the preferable task. As before, the worker agent could only identify the interestingness of a task by trying it out. The agent was also not able to identify the target task by its task number, as these were randomized every episode. We also reduced the agent’s time budget so that it could not work on all tasks. The agent was retrained for this modified scenario.

For two randomly sampled episodes, Figure 11 shows the trajectories of the tasks the simulated worker answered questions for. One can see that our model replicated search behavior: at first, the agent tried out two tasks in a negligent way without putting much effort in. Once it found its preferred task, it put effort into it answering deliberately.

Only at the end of its time budget did it sometimes answer negligently since there were no longer consequences for malicious behavior since its reputation did not carry over to the next episode.

C.2 The length of task search is limited. While workers actively search for tasks they like, Chilton et al. [8] showed that this search is focused on the first pages of the search results and seldomly beyond. For a realistic behavior, we expected our agent to adapt its behavior equivalently and limit its search if the number of tasks it needs to search through increases.

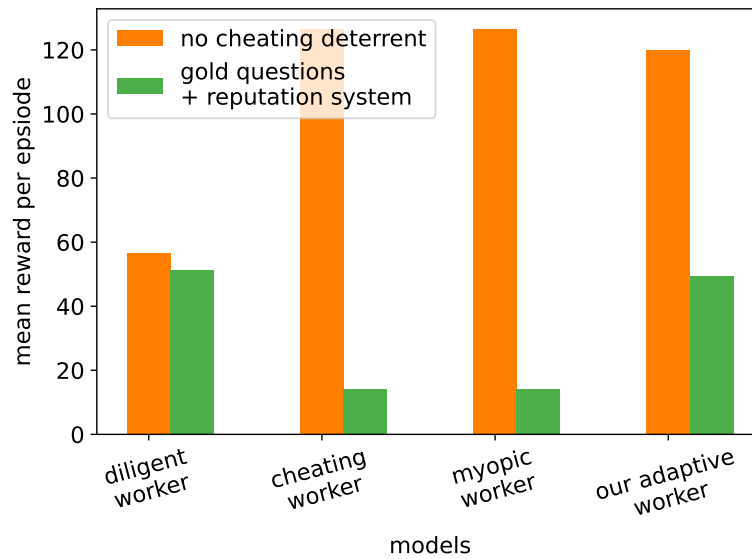


Fig. 10. Mean reward per episode by different crowdworker models in an environment without (orange) and with (green) cheating deterrents. The diligent worker model always answers diligently, the cheating worker model always answers negligently and the myopic worker answers in the way that gives the highest reward for the next action. Only our adaptive worker model can replicate the behavior of human crowdworkers, considering long term rewards and adapting accordingly.

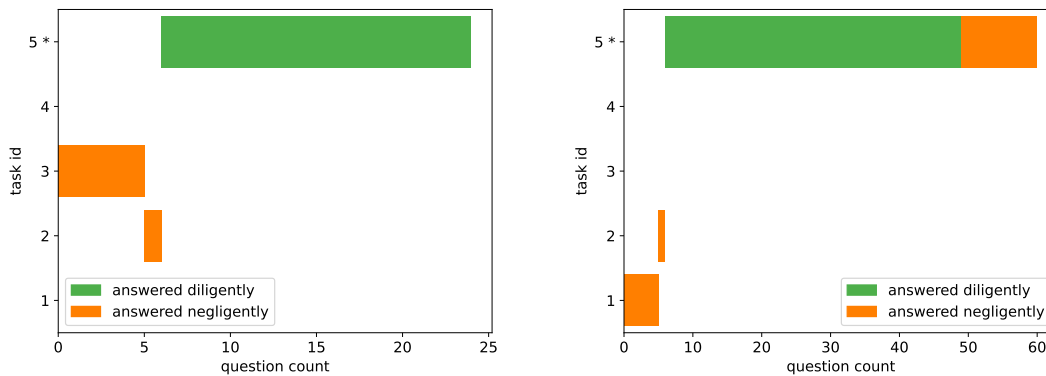


Fig. 11. C.1 Workers search for tasks: The trajectories of the tasks worked on for two randomly sampled episodes. The agent tried out tasks until it found the preferred task (marked with *).

We repeated the previous simulation experiment C.1 but increased the number of tasks available to the worker. Figure 12 shows how many times the preferred task is found and how many times the agent switched to another task. The percentage of times the preferred task was found and worked on dropped drastically when the number of available tasks was increased. The number of switches to other tasks, which are necessary when searching, stayed, however, in

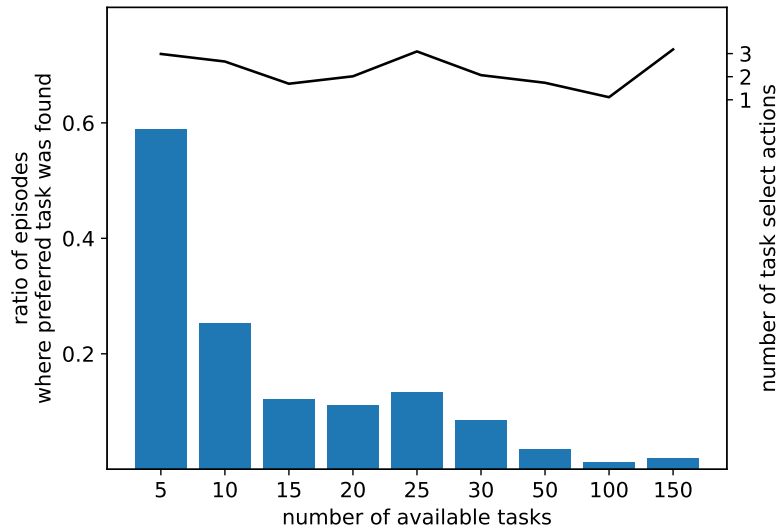


Fig. 12. *C.1 The length of task search is limited*: Rate of episodes (over 1000 episodes) where the preferred task was found and worked on (blue bars) and the number of task select actions performed by the agent (black line). The number of task changes stayed in the same range while the times the preferred task was found decreases strongly, showing that the agent does not search for the preferred task if the search takes too long.

the same small range. Like the human workers who did not bother to go beyond the first result pages, the agent, thus, gave up searching for the one preferred task if the pool of available tasks and thus the search took too long.

This behavior can be explained by it being more beneficial for the agent to work on a suboptimal task than spending more unpaid time searching for the preferred task.

5.4 Discussion

For our three areas of interest – task participation, cheating, and task search – we simulated changes in crowdsourcing environments to observe the behavior that emerges from our worker model. For all studied effects, we reproduced the same qualitative effects that were observed in the literature. Our theory, that workers choose their actions to maximize their goals with limited knowledge, can thus be used to explain the studied behavior. In the next section, we will employ our model as a tool for explaining further aspects of crowdsourcing.

6 EXPLAINING THE EMERGENCE OF ‘CHEATING’

In this section, we dive deeper, using our proposed and validated model to explain phenomena around cheating on crowdsourcing platforms. This highlights the adaptive nature of our model and how it can be used to better understand crowdworker behavior.

In Section 5.2, we saw that, like their human counterparts, our simulated workers cheat as long as there are no repercussions for it. From the worker’s perspective, this is a rational decision. Looking at the payout, the agent obtained on average 120 monetary units per episode (standard error 0.60) in that experiment; compared to only 49 units (standard

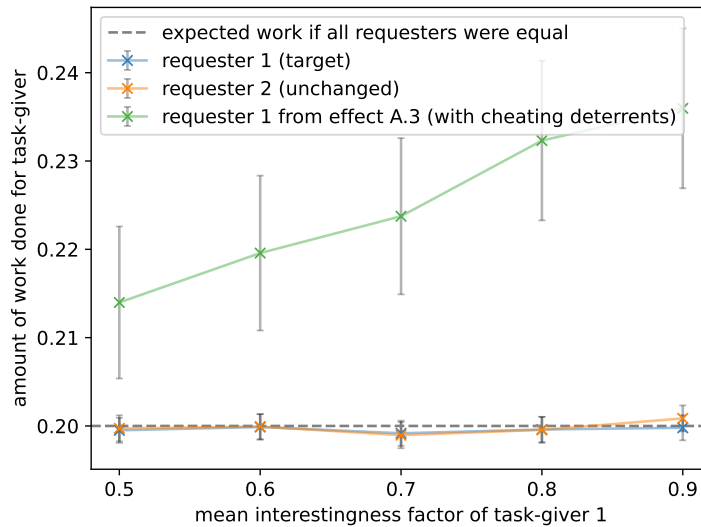


Fig. 13. *Cheating behavior eliminates interestingness effect.* The experiment for effect A.3 (Section 5.1) was repeated but without cheating deterrents (gold questions and reputation system). The mean interestingness reward (or enjoyment) of the target task-giver was increased from 0.5 to 0.9. In the previous scenario that discouraged cheating, making the task more interesting had a clear effect (green line in this plot). Here, the interestingness effect disappeared as the agent preferred to cheat instead of working on and enjoying the task.

error 0.51) when it was forced to abstain from cheating due to the reputation system. In the latter case, continuing to cheat would not have been a reasonable choice as this results in being barred from tasks early and obtaining an even lower reward.

An interesting aspect to note is the interaction between cheating and the interestingness factor. For the effect A.3, we replicated from the existing literature that task participation increases if the worker is more interested in the task. In this setting, we used gold questions and a reputation system to prevent malicious behavior. Here, we repeated the experiment without those cheating deterrents. Figure 13 shows that the positive effect of interesting questions on task participation disappeared. The monetary reward was the stronger incentive and the agent preferred to earn more by answering quickly and negligently instead of properly working (and possibly enjoying) the task. This is an insight to consider for task designers. While making a task more interesting, e.g. through gamification, can be beneficial to the participation rate, the additional effort of increasing the interestingness can be wasted if one does not prevent malicious behavior at the same time.

Our model can also help explain why crowdsourcing is so prone to cheating compared to other employment and marketplace situations. In Figure 14, we measured the amount of cheating in relation to the number of task-givers. Each task-giver evaluated the worker's performance using gold questions, but no reputation system is used. One can see that in a small market, the worker agent is deterred from cheating. Increasing the number of available tasks, however, correlated directly with the amount of cheating. The agent no longer had to fear that being excluded from one task would result in a substantial drop in income. Instead, it could just move on to the next of the many available tasks.

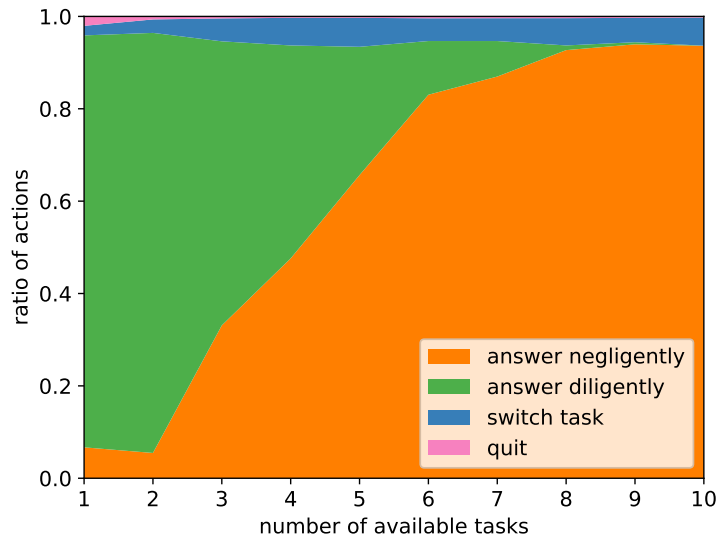


Fig. 14. *The larger the pool of available tasks, the more the agent cheats.* Measuring the percentage of actions taken for an increasing number of available tasks. Each task contained 10% gold questions and a worker was blocked from the task if they failed 3 gold questions. However, no reputation system across tasks was used. In a small market with few tasks, the agent answered mostly diligently and there was nearly no cheating. In a market with a large pool of tasks, like in crowdworking, the worker had a high incentive to cheat, i.e. answer negligently.

The mostly anonymous setting and the many available tasks allow these dynamics on real crowdworking platforms. This also explains why reputation systems or the selection process of high-quality workers are so important on these platforms to bring back accountability.

Based on this result, one could assume that the higher the required reputation for a task, the better. However, to err is famously human and not all incorrect answers are due to malicious behavior. Looking at the number of obtained, correct answers with regards to reputation levels in Figure 15, one can see that there is actually a drop in answers for high reputation levels. This is due to the worker agent being excluded not because of malicious behavior but because it made mistakes. A strong mechanism of cheating prevention, like the requirement of a high reputation score, removes workers with malicious behavior. However, a high threshold can also eliminate honest and good workers that just made a few mistakes; thus reducing the pool of available workers. It can, therefore, be beneficial to not require a perfect reputation score if the aim is to obtain large numbers of answers. This is reflective of real-world crowdsourcing settings. Kazai et al. [34], e.g., set their required score to $> 95\%$. Gadiraju et al. [18] identified less-competent workers that have a genuine intent to complete work correctly but lack the necessary skills. And even their most competent worker group was not perfect and had an average accuracy of around 90% for simple and below 70% for their most complex task. For machine learning datasets, it has been shown that perfect answers are not necessary to obtain useful training data [2, 17, 24] and imperfection can even be encouraged in favor of speed [38].

Another crucial aspect for task-givers is cost. The task-giver pays for each question that is answered. The higher the percentage of gold-standard questions, the easier it is to detect negligent behavior. These questions with known

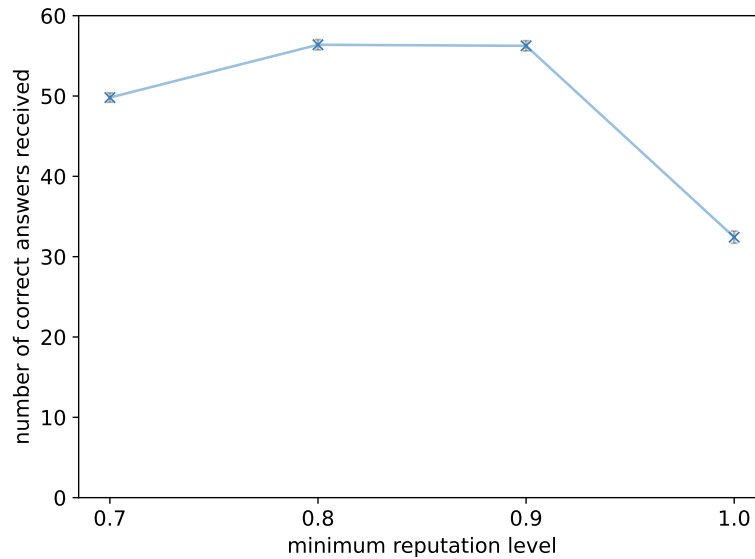


Fig. 15. *Too high reputation levels are detrimental to the number of answers.* Our results show that a high threshold can also eliminate honest and good workers that just made a few mistakes; thus reducing the pool of available workers. This is reflective of real-world crowdsourcing settings. The plot shows average number of correct answers received by all task-givers for different minimum reputation levels and 10% gold-standard questions. The grey bars show the empirical standard error over 1000 episodes. Requiring a perfect reputation of the worker actually decreases the number of received *correct* answers since the worker makes non-malicious mistakes.

answers, however, do not provide new information about the actual task of interest. The task-giver, therefore, needs to choose a suited ratio between gold questions and normal questions with unknown answer [42]. We inspected this by running our approach with different probabilities for gold questions and measuring for a task the ratio of correct answers received and the number of questions answered (both normal and gold-standard). This is equivalent to the ratio of new task information and costs for the task-giver. In Figure 16, one can see the dynamics that emerged. If the probability for gold questions was very low ($p_{\text{gold}} = 0.01$ in the plot), the task-givers had low overhead costs, but the quality of the answers was very poor due to the agent being inclined to cheat. Increasing the minimum level of reputation and increasing the percentage of gold questions could increase the amount of correct, new information received. This was due to the agent being controlled more strongly and, therefore, less prone to cheat. However, if the probability for gold questions was too high ($p_{\text{gold}} = 0.3$ in the plot) the advantage of higher quality answers was diminished by the large overhead costs. A balanced approach between cheating deterrents and costs obtained the best results here.

7 DISCUSSION, LIMITATIONS & FUTURE WORK

This paper has proposed a theory and a model of worker behavior within the framework of computational rationality. The core assumption is that workers are rational, but in a bounded way. In other words, workers choose actions that optimize their expected rewards – in this case payout and interestingness – within the bounds of their observations.

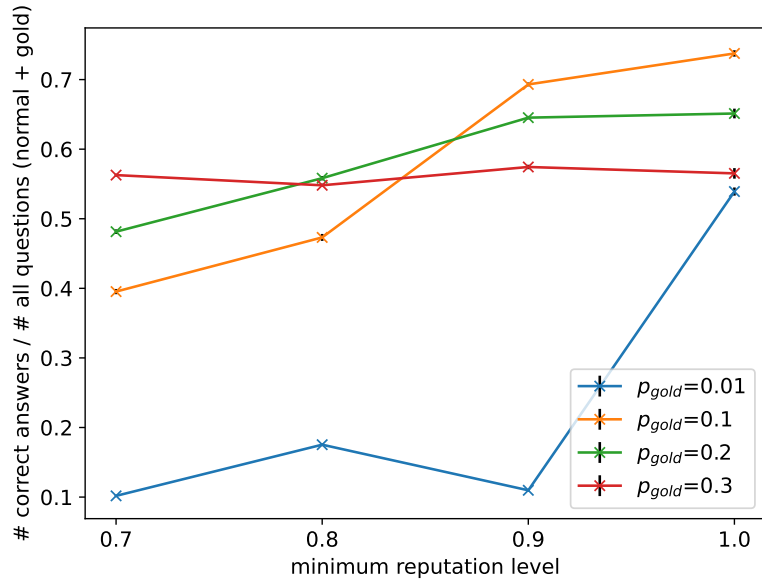


Fig. 16. *Costs and quality of answers need to be balanced.* Average ratio between the number of correct answers received (new information) and the number of questions (normal and gold = costs). Settings with few gold questions were cheap but obtained few correct information. Settings with many gold questions were too expensive compared to the amount of new information obtained.

Using reinforcement learning we can utilize this theory to demonstrate adaptive behavior. Our approach explains the behavior of crowdworkers with regard to

- task participation and the joint effect of payout, time effort, and interestingness on task choice,
- task search and the impact of search length, and
- negligent behavior and cheating, i.a.
 - their prevalence in crowdworking compared to other labor markets,
 - the impact on the worker’s experience, and
 - the effect of control mechanisms like reputation systems.

These explanations help in better understanding the behavior of crowdworkers. They can also be used by task-givers to better understand the implications of their task design in order to improve both their yield and the experience for the crowdworkers.

We speculate that our theory also applies to further aspects of crowdsourcing and that these phenomena can be modeled via rational, optimizing worker behavior. Extending the model on the same theoretical basis will help better model and understand the actions of crowdworkers and human behavior in general.

(1) *Extending the model to a continuous time setting will allow the agent to take fine-grained decisions on how to distribute its effort.* We currently simulate our crowdworking setting in a round-based fashion where the agent takes discrete actions. While answering negligently takes less time than answering diligently, the agent can not decide how much time to spend specifically on an action. Gadiraju et al. [18] found more fine-grained behavior, e.g. both their “competent worker” and “diligent worker” are accurate, but the latter is much slower in answering. This could be explained by the “diligent worker” being less competent and making the rational decision to compensate for their lack

of skills with more effort to ensure the same level of quality and therefore long-term earnings. Similarly, they found “fast deceivers” and “smart deceivers.” While the former behaved equivalently to the negligent behavior we encountered, the latter decided to spend extra time to circumvent the task-giver’s quality checks. In a continuous time setting, the agent can take rational and more fine-grained choices on how to answer a question.

(2) *Modeling properties of individuals will provide more specific explanations.* In our model, individual workers can differ in how they value payout and interestingness, how skilled they are, and how much time they can spend on crowdworking. Studies found significant differences in related metrics like accuracy and participation depending on factors such as country of origin or age group [34, 54, 59]. In this work, we focused on the impacts that crowdwork design decisions have (similar to Scekic et al. [63]) and thus aimed at explaining findings from a large set of empirical studies instead of reproducing absolute numbers from a specific dataset. Therefore, we did not study individuals’ properties but chose distributions for the parameters describing the workers and task-givers. Past works that used the framework of computational rationality on other settings fitted empirical parameters to specific datasets [20, 30]. Using such a data-fitting approach could consider the distinctive characteristics of a crowdworking platform and its users. This could deliver more accurate predictions when simulating a specific crowdworking environment. While this would break with our goal of not requiring a priori behavioral data collection, combining our approach with a small amount of data collection might allow modeling more complex behaviors than with a purely data-driven approach.

(3) *Breaking the assumption of constant skills will enable modeling learning effects in workers.* We currently simulate the expertise of a worker by a value $e_{\tau,w}$ that gives the probability of a worker w answering a task τ correctly if they answer diligently. This allows us to model workers of different skill levels and different areas of expertise. It assumes, however, that the expertise is constant. This might not always be the case. Han et al. [22] found a positive learning effect where workers provide better answers as they progress through the questions. Similarly, Rübiger et al. [57] identified a learning and an exploitation phase for the workers. Workers who had reached the exploitation phase were faster and more reliable. Task-givers are often aware of the need for learning and provide instructions or even training phases (e.g. in [41]) and ideas have been proposed to actively train workers on platforms [14, 47, 65].

Besides the possibility to model learning effects in workers, removing the assumption of constant skill will also open the approach to investigating interesting behavior in task-givers. Task-givers could take rational decisions on how much to invest in the training of their crowdworkers, e.g. through paid example questions, for higher quality results in the long term.

(4) *Multi-agent reinforcement learning will make it possible to model interactions between crowdworkers.* Our modeled crowdworkers currently interact solely with the crowdworking platform, selecting and working directly for task-givers. However, not only the task-givers are competing on the crowdsourcing market, but also the workers. Hara et al. [23] report that workers registered for multiple tasks of interest at the same time to save them for later and to prevent them from being taken by others. Workers can also interact in collaborative ways, such as using outside platforms to rate and discuss task-givers [21, 28, 45, 48]. In works such as [6], task-givers even encourage on-platform collaboration between workers on tasks. Multi-agent reinforcement learning [75] provides a challenging but exciting option to extend our approach; keeping the simplicity of our agent’s rational goals but allowing inter-worker relations to emerge.

(5) *Beyond Crowdworking.* In this work, we focused on crowdworking as a marketplace that is highly relevant to fields such as social sciences, human-computer interaction, and machine learning, and that has also received extensive study in the literature. In Section 6, we already hinted at the possibility of studying smaller markets when scaling the number of task-givers. More generally, crowdworking is just one type of scenario where there are multiple tasks available and a human actor must, under partial observability and delayed rewards, decide how to invest their time and

energy. The same principles apply to many other situations in life: a student must choose between courses with goals ranging from life enjoyment to future career perspectives. In the working life, an employee must choose how to attend to different tasks as part of complex business environments. And in the domestic setting, a person chooses every day between a large set of possible activities with varying types of rewards. We see the transfer of the presented approach to these settings as fascinating future work to help understand and explain human behavior in different environments.

8 OPEN SCIENCE

We make our model platform publicly available as open source online¹ to foster further explorations in the area of crowdworking and simulation of human behavior in general. The model can be used as a component for a larger simulation or alternatively directly with our crowdworking environment. It also contains a code tutorial to introduce the model platform.

9 CONCLUSION

In this work, we proposed a theory to explain crowdworker behavior: the worker maximizes the expected value of their actions, measured in payout and interest, under the limitation that they have restricted knowledge of the crowdworking platform. Using the recent advances in deep reinforcement learning, we showed that a model based on this theory replicates the behavior reported in empirical works on human crowdworkers. We studied three areas that are relevant to task-givers and that have interesting long-term effects, namely task participation, negligent behavior, and task search. We further explore the issue of cheating, explaining facets like its prevalence on crowdworking platforms compared to other marketplaces or why requiring a perfect worker reputation can be detrimental to task-givers. We also discuss how our theory could explain a wider range of aspects in the crowdworking space in future work. To foster further research, we make our modeling platform publicly available.

ACKNOWLEDGMENTS

MAH was supported by a fellowship within the IFI program of the German Academic Exchange Service (DAAD). AO was supported by the Research Council of Finland (flagship program: Finnish Center for Artificial Intelligence, FCAI, grants 328400, 345604, 341763; Subjective Functions, grant 357578).

REFERENCES

- [1] Leonard Adolphs and Thomas Hofmann. 2020. LeDeepChef Deep Reinforcement Learning Agent for Families of Text-Based Games. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*. AAAI Press, 7342–7349. <https://ojs.aaai.org/index.php/AAAI/article/view/6228>
- [2] Görkem Algan and Ilkay Ulusoy. 2021. Image classification with deep learning in the presence of noisy labels: A survey. *Knowl. Based Syst.* 215 (2021), 106771. <https://doi.org/10.1016/j.knosys.2021.106771>
- [3] John Bohannon. 2016. Mechanical Turk upends social sciences. *Science* 352, 6291 (2016), 1263–1264. <https://doi.org/10.1126/science.352.6291.1263>
- [4] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. 2016. OpenAI Gym. arXiv:arXiv:1606.01540
- [5] Joel Chan, Steven Dang, and Steven P Dow. 2016. Improving crowd innovation with expert facilitation. In *Proceedings of the 19th ACM Conference on Computer-Supported Cooperative Work & Social Computing*. 1223–1235.
- [6] Joseph Chee Chang, Saleema Amershi, and Ece Kamar. 2017. Revolt: Collaborative Crowdsourcing for Labeling Machine Learning Datasets. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems, Denver, CO, USA, May 06-11, 2017*, Gloria Mark, Susan R. Fussell, Cliff Lampe, m. c. schraefel, Juan Pablo Hourcade, Caroline Appert, and Daniel Wigdor (Eds.). ACM, 2334–2346. <https://doi.org/10.1145/3025453.3026044>

¹<https://github.com/m-hedderich/crowdworkersimulation>

- [7] Xiao Chen, Min Liu, Yaqin Zhou, Zhongcheng Li, Shuang Chen, and Xiangnan He. 2017. A Truthful Incentive Mechanism for Online Recruitment in Mobile Crowd Sensing System. *Sensors* 17, 1 (2017). <https://doi.org/10.3390/s17010079>
- [8] Lydia B. Chilton, John J. Horton, Robert C. Miller, and Shiri Azenkot. 2010. Task search in a human computation market. In *Proceedings of the ACM SIGKDD Workshop on Human Computation, HCOMP '10, Washington DC, USA, July 25, 2010*. ACM, 1–9. <https://doi.org/10.1145/1837885.1837889>
- [9] Will Dabney, Mark Rowland, Marc G. Bellemare, and Rémi Munos. 2018. Distributional Reinforcement Learning With Quantile Regression. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, Sheila A. McIlraith and Kilian Q. Weinberger (Eds.). AAAI Press, 2892–2901. <https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/17184>
- [10] Peter Dayan and Nathaniel D Daw. 2008. Decision theory, reinforcement learning, and the brain. *Cognitive, Affective, & Behavioral Neuroscience* 8, 4 (2008), 429–453.
- [11] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. ImageNet: A large-scale hierarchical image database. In *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20-25 June 2009, Miami, Florida, USA*. IEEE Computer Society, 248–255. <https://doi.org/10.1109/CVPR.2009.5206848>
- [12] Djellel Eddine Difallah, Michele Catasta, Gianluca Demartini, Panagiotis G. Ipeirotis, and Philippe Cudré-Mauroux. 2015. The Dynamics of Micro-Task Crowdsourcing: The Case of Amazon MTurk. In *Proceedings of the 24th International Conference on World Wide Web (Florence, Italy) (WWW '15)*. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, 238–247. <https://doi.org/10.1145/2736277.2741685>
- [13] Seungwon Do, Minsuk Chang, and Byungjoo Lee. 2021. A Simulation Model of Intermittently Controlled Point-and-Click Behaviour. In *CHI '21: CHI Conference on Human Factors in Computing Systems, Virtual Event / Yokohama, Japan, May 8-13, 2021*, Yoshifumi Kitamura, Aaron Quigley, Katherine Isbister, Takeo Igarashi, Pernille Bjørn, and Steven Mark Drucker (Eds.). ACM, 286:1–286:17. <https://doi.org/10.1145/3411764.3445514>
- [14] Shayan Doroudi, Ece Kamar, Emma Brunskill, and Eric Horvitz. 2016. Toward a Learning Science for Complex Crowdsourcing Tasks. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (San Jose, California, USA) (CHI '16)*. Association for Computing Machinery, New York, NY, USA, 2623–2634. <https://doi.org/10.1145/2858036.2858268>
- [15] Serge Egelman, Ed H. Chi, and Steven Dow. 2014. *Crowdsourcing in HCI Research*. Springer New York, New York, NY, 267–289. https://doi.org/10.1007/978-1-4939-0378-8_11
- [16] Oluwaseyi Feyisetan, Elena Simperl, Max Van Kleek, and Nigel Shadbolt. 2015. Improving Paid Microtasks through Gamification and Adaptive Furtherance Incentives. In *Proceedings of the 24th International Conference on World Wide Web, WWW 2015, Florence, Italy, May 18-22, 2015*, Aldo Gangemi, Stefano Leonardi, and Alessandro Panconesi (Eds.). ACM, 333–343. <https://doi.org/10.1145/2736277.2741639>
- [17] Benoît Frénay and Michel Verleysen. 2014. Classification in the Presence of Label Noise: A Survey. *IEEE Trans. Neural Networks Learn. Syst.* 25, 5 (2014), 845–869. <https://doi.org/10.1109/TNNLS.2013.2292894>
- [18] Ujwal Gadiraju, Gianluca Demartini, Ricardo Kawase, and Stefan Dietze. 2019. Crowd Anatomy Beyond the Good and Bad: Behavioral Traces for Crowd Worker Modeling and Pre-selection. *Comput. Support. Cooperative Work*, 28, 5 (2019), 815–841. <https://doi.org/10.1007/s10606-018-9336-y>
- [19] Ujwal Gadiraju, Ricardo Kawase, and Stefan Dietze. 2014. A Taxonomy of Microtasks on the Web. In *Proceedings of the 25th ACM Conference on Hypertext and Social Media (Santiago, Chile) (HT '14)*. Association for Computing Machinery, New York, NY, USA, 218–223. <https://doi.org/10.1145/2631775.2631819>
- [20] Christoph Gebhardt, Antti Oulasvirta, and Otmar Hilliges. 2021. Hierarchical reinforcement learning explains task interleaving behavior. *Computational Brain & Behavior* 4, 3 (2021), 284–304.
- [21] Neha Gupta, David Martin, Benjamin V Hanrahan, and Jacki O'Neill. 2014. Turk-life in India. In *Proceedings of the 18th International Conference on Supporting Group Work*. 1–11.
- [22] Lei Han, Kevin Roitero, Ujwal Gadiraju, Cristina Sarasua, Alessandro Checco, Eddy Maddalena, and Gianluca Demartini. 2019. All Those Wasted Hours: On Task Abandonment in Crowdsourcing. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining (Melbourne VIC, Australia) (WSDM '19)*. Association for Computing Machinery, New York, NY, USA, 321–329. <https://doi.org/10.1145/3289600.3291035>
- [23] Kotaro Hara, Abigail Adams, Kristy Milland, Saiph Savage, Chris Callison-Burch, and Jeffrey P. Bigham. 2018. A Data-Driven Analysis of Workers' Earnings on Amazon Mechanical Turk. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems, CHI 2018, Montreal, QC, Canada, April 21-26, 2018*, Regan L. Mandryk, Mark Hancock, Mark Perry, and Anna L. Cox (Eds.). ACM, 449. <https://doi.org/10.1145/3173574.3174023>
- [24] Michael A. Hedderich, Lukas Lange, Heike Adel, Jannik Strötgen, and Dietrich Klakow. 2021. A Survey on Recent Approaches for Natural Language Processing in Low-Resource Scenarios. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*. Association for Computational Linguistics, 2545–2568. <https://doi.org/10.18653/v1/2021.naacl-main.201>
- [25] Danula Hettiachchi, Vassilis Kostakos, and Jorge Goncalves. 2022. A Survey on Task Assignment in Crowdsourcing. *ACM Comput. Surv.* 55, 3, Article 49 (feb 2022), 35 pages. <https://doi.org/10.1145/3494522>
- [26] Dirk Hovy, Taylor Berg-Kirkpatrick, Ashish Vaswani, and Eduard H. Hovy. 2013. Learning Whom to Trust with MACE. In *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 9-14, 2013, Westin Peachtree Plaza Hotel, Atlanta, Georgia, USA*, Lucy Vanderwende, Hal Daumé III, and Katrin Kirchhoff (Eds.). The Association for Computational Linguistics, 1120–1130. <https://aclanthology.org/N13-1132/>

- [27] Thomas F. Icard. 2023. Resource Rationality. (2023).
- [28] Lilly C. Irani and M. Six Silberman. 2013. Turkopticon: Interrupting Worker Invisibility in Amazon Mechanical Turk. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (Paris, France) (CHI '13)*. Association for Computing Machinery, New York, NY, USA, 611–620. <https://doi.org/10.1145/2470654.2470742>
- [29] Georg Jäger, Laura S Zilian, Christian Hofer, and Manfred Füllsack. 2019. Crowdfunding: working with or against the crowd? *Journal of Economic Interaction and Coordination* 14, 4 (2019), 761–788.
- [30] Jussi P P Jokinen, Aditya Acharya, Mohammad Uzair, Xinhui Jiang, and Antti Oulasvirta. 2021. Touchscreen Typing as Optimal Supervisory Control. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems (CHI '21)*. ACM. <https://doi.org/10.1145/3411764.3445483>
- [31] Yunus Emre Kara, Gaye Genc, Oya Aran, and Lale Akarun. 2015. Modeling annotator behaviors for crowd labeling. *Neurocomputing* 160 (2015), 141–156. <https://doi.org/10.1016/j.neucom.2014.10.082>
- [32] Nicolas Kaufmann, Thimo Schulze, and Daniel Veit. 2011. More than fun and money. Worker Motivation in Crowdsourcing - A Study on Mechanical Turk. In *A Renaissance of Information Technology for Sustainability and Global Competitiveness. 17th Americas Conference on Information Systems, AMCIS 2011, Detroit, Michigan, USA, August 4-8 2011*, Vallabh Sambamurthy and Mohan Tanniru (Eds.). Association for Information Systems. http://aisel.aisnet.org/amcis2011_submissions/340
- [33] Gabriella Kazai, Jaap Kamps, and Natasa Milic-Frayling. 2011. Worker Types and Personality Traits in Crowdsourcing Relevance Labels. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management (Glasgow, Scotland, UK) (CIKM '11)*. Association for Computing Machinery, New York, NY, USA, 1941–1944. <https://doi.org/10.1145/2063576.2063860>
- [34] Gabriella Kazai, Jaap Kamps, and Natasa Milic-Frayling. 2012. The face of quality in crowdsourcing relevance labels: demographics, personality and labeling accuracy. In *21st ACM International Conference on Information and Knowledge Management, CIKM'12, Maui, HI, USA, October 29 - November 02, 2012*, Xue-wen Chen, Guy Lebanon, Haixun Wang, and Mohammed J. Zaki (Eds.). ACM, 2583–2586. <https://doi.org/10.1145/2396761.2398697>
- [35] Vassillis-Javed Khan, Konstantinos Papangelis, Ioanna Lykourantzou, and Panos Markopoulos. 2019. *Macrotask Crowdsourcing: Engaging the Crowds to Address Complex Problems*. Springer.
- [36] Joy Kim, Justin Cheng, and Michael S Bernstein. 2014. Ensemble: exploring complementary strengths of leaders and crowds in creative collaboration. In *Proceedings of the 17th ACM conference on Computer supported cooperative work & social computing*. 745–755.
- [37] Aniket Kittur, Ed H. Chi, and Bongwon Suh. 2008. Crowdsourcing User Studies with Mechanical Turk. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (Florence, Italy) (CHI '08)*. Association for Computing Machinery, New York, NY, USA, 453–456. <https://doi.org/10.1145/1357054.1357127>
- [38] Ranjay A. Krishna, Kenji Hata, Stephanie Chen, Joshua Kravitz, David A. Shamma, Li Fei-Fei, and Michael S. Bernstein. 2016. Embracing Error to Enable Rapid Crowdsourcing. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (San Jose, California, USA) (CHI '16)*. Association for Computing Machinery, New York, NY, USA, 3167–3179. <https://doi.org/10.1145/2858036.2858115>
- [39] Pavel Kucherbaev, Florian Daniel, Maurizio Marchese, Fabio Casati, and Brian Reavey. 2014. Toward Effective Tasks Navigation in Crowdsourcing. In *Proceedings of the 2014 International Working Conference on Advanced Visual Interfaces (Como, Italy) (AVI '14)*. Association for Computing Machinery, New York, NY, USA, 401–404. <https://doi.org/10.1145/2598153.2602249>
- [40] Zhi Li, Yu-Jung Ko, Aini Putkonen, Shirin Feiz, Vikas Ashok, IV Ramakrishnan, Antti Oulasvirta, and Xiaojun Bi. 2023. Modeling Touch-based Menu Selection Performance of Blind Users via Reinforcement Learning. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. 1–18.
- [41] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. 2014. Microsoft COCO: Common Objects in Context. In *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V (Lecture Notes in Computer Science, Vol. 8693)*, David J. Fleet, Tomás Pajdla, Bernt Schiele, and Tinne Tuytelaars (Eds.). Springer, 740–755. https://doi.org/10.1007/978-3-319-10602-1_48
- [42] Qiang Liu, Alexander T Ihler, and Mark Steyvers. 2013. Scoring Workers in Crowdsourcing: How Many Control Questions are Enough?. In *Advances in Neural Information Processing Systems*, C.J. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger (Eds.), Vol. 26. Curran Associates, Inc. <https://proceedings.neurips.cc/paper/2013/file/cc1aa436277138f61cda703991069eaf-Paper.pdf>
- [43] Xuan Liu, Meiyu Lu, Beng Chin Ooi, Yanyan Shen, Sai Wu, and Meihui Zhang. 2012. CDAS: A Crowdsourcing Data Analytics System. *Proc. VLDB Endow.* 5, 10 (2012), 1040–1051. <https://doi.org/10.14778/2336664.2336676>
- [44] Fenglong Ma, Yaliang Li, Qi Li, Minghui Qiu, Jing Gao, Shi Zhi, Lu Su, Bo Zhao, Heng Ji, and Jiawei Han. 2015. FaitCrowd: Fine Grained Truth Discovery for Crowdsourced Data Aggregation. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Sydney, NSW, Australia, August 10-13, 2015*. ACM, 745–754. <https://doi.org/10.1145/2783258.2783314>
- [45] David B. Martin, Benjamin V. Hanrahan, Jacki O'Neill, and Neha Gupta. 2014. Being a turker. In *Computer Supported Cooperative Work, CSCW '14, Baltimore, MD, USA, February 15-19, 2014*, Susan R. Fussell, Wayne G. Lutters, Meredith Ringel Morris, and Madhu C. Reddy (Eds.). ACM, 224–235. <https://doi.org/10.1145/2531602.2531663>
- [46] Winter A. Mason and Duncan J. Watts. 2009. Financial incentives and the "performance of crowds". *SIGKDD Explor.* 11, 2 (2009), 100–108. <https://doi.org/10.1145/1809400.1809422>
- [47] Masaki Matsubara, Ria Mae Borromeo, Sihem Amer-Yahia, and Atsuyuki Morishima. 2021. Task Assignment Strategies for Crowd Worker Ability Improvement. *Proc. ACM Hum.-Comput. Interact.* 5, CSCW2, Article 375 (oct 2021), 20 pages. <https://doi.org/10.1145/3479519>

- [48] Brian James McInnis, Dan Cosley, Chaebong Nam, and Gilly Leshed. 2016. Taking a HIT: Designing around Rejection, Mistrust, Risk, and Workers' Experiences in Amazon Mechanical Turk. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems, San Jose, CA, USA, May 7-12, 2016*, Jofish Kaye, Allison Druin, Cliff Lampe, Dan Morris, and Juan Pablo Hourcade (Eds.). ACM, 2271–2282. <https://doi.org/10.1145/2858036.2858539>
- [49] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. 2013. Playing Atari with Deep Reinforcement Learning. <https://doi.org/10.48550/ARXIV.1312.5602>
- [50] An Nguyen, Byron Wallace, and Matthew Lease. 2015. Combining crowd and expert labels using decision theoretic active learning. In *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing*, Vol. 3. 120–129.
- [51] Antti Oulasvirta, Jussi P. P. Jokinen, and Andrew Howes. 2022. Computational Rationality as a Theory of Interaction. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems (New Orleans, LA, USA) (CHI '22)*. Association for Computing Machinery, New York, NY, USA, Article 359, 14 pages. <https://doi.org/10.1145/3491102.3517739>
- [52] Junwon Park, Ranjay Krishna, Pranav Khadpe, Li Fei-Fei, and Michael Bernstein. 2019. AI-Based Request Augmentation to Increase Crowdsourcing Participation. *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing* 7, 1 (Oct. 2019), 115–124. <https://ojs.aaai.org/index.php/HCOMP/article/view/5282>
- [53] Ellie Pavlick, Matt Post, Ann Irvine, Dmitry Kachaev, and Chris Callison-Burch. 2014. The Language Demographics of Amazon Mechanical Turk. *Trans. Assoc. Comput. Linguistics* 2 (2014), 79–92. https://doi.org/10.1162/tacl_a_00167
- [54] Lisa Posch, Arnim Bleier, Clemens M. Lechner, Daniel Danner, Fabian Flöck, and Markus Strohmaier. 2019. Measuring Motivations of Crowdsworkers: The Multidimensional Crowdsworker Motivation Scale. *Trans. Soc. Comput.* 2, 2, Article 8 (sep 2019), 34 pages. <https://doi.org/10.1145/3335081>
- [55] Sihang Qiu, Alessandro Bozzon, and Geert-Jan Houben. 2020. VirtualCrowd: A Simulation Platform for Microtask Crowdsourcing Campaigns. In *Companion of The 2020 Web Conference 2020, Taipei, Taiwan, April 20-24, 2020*, Amal El Fallah Seghrouchni, Gita Sukthankar, Tie-Yan Liu, and Maarten van Steen (Eds.). ACM / IW3C2, 222–225. <https://doi.org/10.1145/3366424.3383546>
- [56] Sihang Qiu, Ujwal Gadiraju, and Alessandro Bozzon. 2020. Improving Worker Engagement Through Conversational Microtask Crowdsourcing. In *CHI '20: CHI Conference on Human Factors in Computing Systems, Honolulu, HI, USA, April 25-30, 2020*, Regina Bernhaupt, Florian 'Floyd' Mueller, David Verweij, Josh Andres, Joanna McGrenere, Andy Cockburn, Ignacio Avellino, Alix Goguy, Pernille Bjon, Shengdong Zhao, Briane Paul Samson, and Rafal Kocielnik (Eds.). ACM, 1–12. <https://doi.org/10.1145/3313831.3376403>
- [57] Stefan Rübiger, Myra Spiliopoulou, and Yücel Saygin. 2018. How do annotators label short texts? Toward understanding the temporal dynamics of tweet labeling. *Inf. Sci.* 457–458 (2018), 29–47. <https://doi.org/10.1016/j.ins.2018.05.036>
- [58] Antonin Raffin, Ashley Hill, Adam Gleave, Anssi Kanervisto, Maximilian Ernestus, and Noah Dormann. 2021. Stable-Baselines3: Reliable Reinforcement Learning Implementations. *Journal of Machine Learning Research* 22, 268 (2021), 1–8. <http://jmlr.org/papers/v22/20-1364.html>
- [59] Jakob Rogstadius, Vassilis Kostakos, Aniket Kittur, Boris Smus, Jim Laredo, and Maja Vukovic. 2011. An Assessment of Intrinsic and Extrinsic Motivation on Task Performance in Crowdsourcing Markets. In *Proceedings of the Fifth International Conference on Weblogs and Social Media, Barcelona, Catalonia, Spain, July 17-21, 2011*, Lada A. Adamic, Ricardo Baeza-Yates, and Scott Counts (Eds.). The AAAI Press. <http://www.aaai.org/ocs/index.php/ICWSM/ICWSM11/paper/view/2778>
- [60] Stuart J. Russell. 1997. Rationality and intelligence. *Artificial Intelligence* 94, 1 (1997), 57–77. [https://doi.org/10.1016/S0004-3702\(97\)00026-X](https://doi.org/10.1016/S0004-3702(97)00026-X) Economic Principles of Multi-Agent Systems.
- [61] Razieh L. Saremi, Ye Yang, Gregg Vesonder, Guenther Ruhe, and He Zhang. 2021. CrowdSim: A Hybrid Simulation Model for Failure Prediction in Crowdsourced Software Development. *CoRR* abs/2103.09856 (2021). arXiv:2103.09856 <https://arxiv.org/abs/2103.09856>
- [62] Burcu Sayin, Evgeny Krivosheev, Jie Yang, Andrea Passerini, and Fabio Casati. 2021. A review and experimental analysis of active learning over crowdsourced data. *Artificial Intelligence Review* 54 (2021), 5283–5305.
- [63] Ognjen Scekic, Christoph Dorn, and Schahram Dustdar. 2013. Simulation-Based Modeling and Evaluation of Incentive Schemes in Crowdsourcing Environments. In *On the Move to Meaningful Internet Systems: OTM 2013 Conferences - Confederated International Conferences: CoopIS, DOA-TrusteD Cloud, and ODBASE 2013, Graz, Austria, September 9-13, 2013. Proceedings (Lecture Notes in Computer Science, Vol. 8185)*, Robert Meersman, Hervé Panetto, Tharam S. Dillon, Johann Eder, Zohra Bellahsene, Norbert Ritter, Pieter De Leenheer, and Dejing Dou (Eds.). Springer, 167–184. https://doi.org/10.1007/978-3-642-41030-7_11
- [64] Christoph Schild, Lau Lilleholt, and Ingo Zettler. 2021. Behavior in cheating paradigms is linked to overall approval rates of crowdsworkers. *Journal of Behavioral Decision Making* 34, 2 (2021), 157–166. <https://doi.org/10.1002/bdm.2195> arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/bdm.2195
- [65] Adish Singla, Ilija Bogunovic, Gábor Bartók, Amin Karbasi, and Andreas Krause. 2014. Near-Optimally Teaching the Crowd to Classify. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014 (JMLR Workshop and Conference Proceedings, Vol. 32)*. JMLR.org, 154–162. <http://proceedings.mlr.press/v32/singla14.html>
- [66] Rion Snow, Brendan O'Connor, Daniel Jurafsky, and Andrew Y. Ng. 2008. Cheap and Fast - But is it Good? Evaluating Non-Expert Annotations for Natural Language Tasks. In *2008 Conference on Empirical Methods in Natural Language Processing, EMNLP 2008, Proceedings of the Conference, 25-27 October 2008, Honolulu, Hawaii, USA, A meeting of SIGDAT, a Special Interest Group of the ACL. ACL, 254–263*. <https://aclanthology.org/D08-1027/>
- [67] Richard S Sutton and Andrew G Barto. 2018. *Reinforcement learning: An introduction*. MIT press.
- [68] Ryutaro Tanno, Ardavan Saeedi, Swami Sankaranarayanan, Daniel C. Alexander, and Nathan Silberman. 2019. Learning From Noisy Labels by Regularized Estimation of Annotator Confusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

- [69] Oriol Vinyals, Timo Ewalds, Sergey Bartunov, Petko Georgiev, Alexander Sasha Vezhnevets, Michelle Yeo, Alireza Makhzani, Heinrich Küttler, John P. Agapiou, Julian Schrittwieser, John Quan, Stephen Gaffney, Stig Petersen, Karen Simonyan, Tom Schaul, Hado van Hasselt, David Silver, Timothy P. Lillicrap, Kevin Calderone, Paul Keet, Anthony Brunasso, David Lawrence, Anders Ekermo, Jacob Repp, and Rodney Tsing. 2017. StarCraft II: A New Challenge for Reinforcement Learning. *CoRR* abs/1708.04782 (2017). arXiv:1708.04782 <http://arxiv.org/abs/1708.04782>
- [70] Yuhuai Wu, Elman Mansimov, Shun Liao, Alec Radford, and John Schulman. 2017. *OpenAI Baselines: ACKTR & A2C*. Technical Report. OpenAI. <https://openai.com/blog/baselines-acktr-a2c/>
- [71] Jie Yan, Renjing Liu, and Guangjun Zhang. 2018. Task Structure, Individual Bounded Rationality and Crowdsourcing Performance: An Agent-Based Simulation Approach. *J. Artif. Soc. Soc. Simul.* 21, 4 (2018). <https://doi.org/10.18564/jasss.3854>
- [72] Jie Yang, Judith Redi, Gianluca Demartini, and Alessandro Bozzon. 2016. Modeling task complexity in crowdsourcing. In *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing*, Vol. 4. 249–258.
- [73] Jiayi Yu, Anfeng Liu, Neal N. Xiong, Shaobo Zhang, Tian Wang, and Mianxiong Dong. 2023. Employing Social Participants for Timely Data Collection Using Pub/sub Solutions in Dynamic IoT Systems. *Computer Networks* 220 (2023), 109501. <https://doi.org/10.1016/j.comnet.2022.109501>
- [74] Lixiu Yu and Jeffrey V. Nickerson. 2011. Cooks or cobblers?: crowd creativity through combination. In *Proceedings of the International Conference on Human Factors in Computing Systems, CHI 2011, Vancouver, BC, Canada, May 7-12, 2011*, Desney S. Tan, Saleema Amershi, Bo Begole, Wendy A. Kellogg, and Manas Tungare (Eds.). ACM, 1393–1402. <https://doi.org/10.1145/1978942.1979147>
- [75] Kaiqing Zhang, Zhuoran Yang, and Tamer Basar. 2019. Multi-Agent Reinforcement Learning: A Selective Overview of Theories and Algorithms. *CoRR* abs/1911.10635 (2019). arXiv:1911.10635 <http://arxiv.org/abs/1911.10635>
- [76] Xuefeng Zhang and Qian Chen. 2022. Towards an understanding of the decision process of solvers’ participation in crowdsourcing contests for problem solving. *Behaviour & Information Technology* 41, 12 (2022), 2635–2653.
- [77] Yuxiang Zhao and Qinghua Zhu. 2014. Effects of extrinsic and intrinsic motivation on participation in crowdsourcing contest A perspective of self-determination theory. *Online Information Review* 38 (11 2014), 896–917. <https://doi.org/10.1108/OIR-08-2014-0188>
- [78] Guangyu Zou, Alvaro Gil, and Marina Tharayil. 2014. An agent-based model for crowdsourcing systems. In *Proceedings of the 2014 Winter Simulation Conference, Savannah, GA, USA, December 7-10, 2014*, Stephen J. Buckley and John A. Miller (Eds.). IEEE/ACM, 407–418. <https://doi.org/10.1109/WSC.2014.7019907>

A CODE & TUTORIAL

The online repository² contains the documented code of the crowdsourcing RL platform along with a tutorial (as Jupyter notebook and PDF) on how to modify, train and evaluate the model. This will be made available online under an open-source license.

B MODEL AND IMPLEMENTATION DETAILS

The specific parameters and distributions differ largely between crowdsourcing platforms and between studies in the literature. We, therefore, tried to choose reasonable default settings.

If not specified otherwise, we used the same parameters and distributions for all experiments. For the properties that are beta-distributed, we chose $\beta(10, 10)$ which has a mean of 0.5 and a symmetric shape similar to a normal distribution. For the number of questions per task, the distributions range was scaled to $[0, 100]$, and for the interest factor $\iota_{\tau, w}$, it was shifted by -0.5 . For the worker’s expertise $e_{\tau, w}$, we chose $\beta(40, 10)$, which has a mean of 0.8, as we assumed that most task-givers create their tasks so that they match the skills of the workers on a platform. We set $|\mathcal{T}| = 5$, $c_{negligent} = 0.1$, $c_{select} = 1$, $b_w = 50$, $\lambda_w^p = 1$ and $\lambda_w^l = 0.5$. If not specified otherwise, the task-giver introduced gold questions with a probability of 0.1 and banned the worker if 3 of these questions were answered incorrectly. The workers started with a reputation of $r = 1$ which was decreased by 0.05 for every incorrectly answered gold question and increased by the same value for a correct answer. All values were reset or redrawn from the corresponding distribution at the beginning of each episode.

We trained each agent for 10M timesteps, reducing the random exploration rate from 1 to 0.05 over the first 20% of the timesteps. The learning starts after 500 timesteps. For all other hyperparameters, we follow the default settings of StableBaselines3 for QR-DQN: discount factor 0.99, 200 quantiles, learning rate $5e - 5$, ReLU activation function, and Adam optimizer. For the analysis, we ran the trained agents for 1000 epochs and reported the mean over the episodes of our analyzed metrics as well as the empirical standard error.

²<https://github.com/m-hedderich/crowdworkersimulation>