Becoming the Super Turker: Increasing Wages via a Strategy from High Earning Workers

Saiph Savage[1],[2]
Universidad Nacional Autonoma de Mexico (UNAM)[1]
saiph.savage@mail.wvu.edu

Chun Chiang[2], Susumu Saito[3]
West Virginia University (WVU)[2]
Waseda University[3]

Carlos Toxtli[2], Jeffrey Bigham[4]
Carnegie Mellon University (CMU)[4]

ABSTRACT

Crowd markets have traditionally limited workers by not providing transparency information concerning which tasks pay fairly or which requesters are unreliable. Researchers believe that a key reason why crowd workers earn low wages is due to this lack of transparency. As a result, tools have been developed to provide more transparency within crowd markets to help workers. However, while most workers use these tools, they still earn less than minimum wage. We argue that the missing element is guidance on how to use transparency information. In this paper, we explore how novice workers can improve their earnings by following the transparency criteria of Super Turkers, i.e., crowd workers who earn higher salaries on Amazon Mechanical Turk (MTurk). We believe that Super Turkers have developed effective processes for using transparency information. Therefore, by having novices follow a Super Turker criteria (one that is simple and popular among Super Turkers), we can help novices increase their wages. For this purpose, we: *(i)* conducted a survey and data analysis to computationally identify a simple yet common criteria that Super Turkers use for handling transparency tools; *(ii)* deployed a two-week field experiment with novices who followed this Super Turker criteria to find better work on MTurk. Novices in our study viewed over 25,000 tasks by 1,394 requesters. We found that novices who utilized this Super Turkers' criteria earned better wages than other novices. Our results highlight that tool development to support crowd workers should be paired with educational opportunities that teach workers how to effectively use the tools and their related metrics (e.g., transparency values). We finish with design recommendations for empowering crowd workers to earn higher salaries.

ACM Reference Format:
Saiph Savage[1], [2], Chun Chiang[2], Susumu Saito[3], and Carlos Toxtli[2], Jeffrey Bigham[4]. 2020. Becoming the Super Turker: Increasing Wages via a Strategy from High Earning Workers. In *Proceedings of The Web Conference 2020 (WWW '20), April 20–24, 2020, Taipei, Taiwan.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3366423.3380199

1 INTRODUCTION

Amazon Mechanical Turk (MTurk) is the most popular crowd market [51]. It allows crowd workers (Turkers) to earn money from micro jobs involving human-intelligence-tasks (HITs). Although MTurk brings new jobs to the economy, most Turkers still struggle to earn the U.S. minimum wage ($7.25) [4, 30]. This is problematic considering that "earning good wages" is the primary motivator of crowd workers [3, 4, 49]. Many believe that the lack of transparency on MTurk is the root cause why Turkers are not being fairly compensated [30]. Economists consider that a market is transparent when all actors can access vast information about the market such as the products, services, or capital assets [74]. Similarly, Silberman et al. discusses how the lack of transparency on gig markets affects workers earnings: "A wide range of processes that shape platform-based workers' ability to find work and receive payment for work completed are, on many platforms, opaque [58]." MTurk has primarily focused on providing transparency information solely to requesters by allowing them to access in-depth knowledge about Turkers. However, MTurk has traditionally provided more limited information to workers (e.g., Turkers previously could not profit from knowledge about requesters' previous hiring record or the estimated hourly wage of the tasks on the market, although as of July 2019, this has started to change[1]). This lack of transparency for workers can lead them to invest significant time in a task but receive anywhere from inadequate to no compensation.

To begin addressing the issue of transparency, scholars and practitioners have developed web browser extensions [13, 39] or created online forums[2] to bring greater transparency to Turkers. These tools and forums provide Turkers with otherwise unavailable information about requesters, tasks, and expected payment. For instance, TurkerView allows workers to obtain an overview of the expected hourly wage they would receive if they worked for a particular requester. However, while an ever increasing number of workers are using these tools for transparency [44], only a fraction of Turkers' earnings are well above the minimum wage [30]. The problem is that utilizing transparency tools to earn higher wages is not straightforward. Each transparency tool displays several different metrics. This poses the question, which metric should a Turker use to ensure better wages? This complexity has likely led most Turkers to employ transparency tools ineffectively [44, 66].

Despite the challenges associated with using transparency tools, top earning crowd workers on MTurk, those that make above the minimum wage, have emerged. These rare workers are commonly referred to as "Super Turkers" because they earn "superior" wages. They have earned this name despite market conditions such as limited availability of tasks and a greater percentage of low-paying tasks[4, 8, 30]. We argue that it is Super Turkers' ability to use transparency that brings them a unique advantage[7]. Our goal was to

WWW '20, April 20–24, 2020, Taipei, Taiwan

ACM ISBN 978-1-4503-7023-3/20/04.
https://doi.org/10.1145/3366423.3380199

[1] https://blog.mturk.com/new-feature-for-the-mturk-marketplace-aaa0bd520e5b
[2] www.turkernation.com/

uncover one of the ways in which Super Turkers used transparency, and then guide novices to follow that same criteria.

For this purpose, we first questioned Super Turkers on how they used specific transparency tools to decide which tasks to perform. We then conducted a data analysis of their responses to computationally identify a Super Turker criteria that denoted one of the ways in which Super Turkers decided to use transparency. We rooted our data analysis on the "Analytic Hierarchy Process", a well established multi-criteria decision-making approach [24, 45, 77]. By using the Analytic Hierarchy Process we identify in a hierarchical form the level of importance that Super Turkers give to different transparency metrics when deciding what tasks to perform. We utilized a hierarchical process given that transparency information is not always available and workers have limited time to decide what tasks to perform (especially as time spent finding labor is time where workers are not paid [28]). We argue that the hierarchy helps workers to more rapidly identify what transparency metrics they should analyze; and if any of those metrics are unavailable, they have a plan of what else to rapidly inspect [10]. We also aimed for this criteria to be simple and popular among Super Turkers. Simplicity was important to make it easier for novices to follow [61]. Popularity was important to have a decision criteria that was representative of Super Turkers (although it is possible and likely that Super Turkers also have other more complex criteria for deciding how to use transparency.)

Once we had an identified transparency criteria that Super Turkers utilized to select work, we conducted a field experiment to investigate how the hourly wage of novices changed when following such criteria. Notice that it is not simple to run a field experiment that can track how workers actually increment their hourly wages. MTurk does not provide any information about the hourly wage of a particular task nor how much time it would take workers to complete a given HIT. It is, therefore, not straightforward how one might calculate the change in workers' wages over time [30]. To overcome these challenges, we developed a plugin that logs workers' behavior, calculates how much time workers spend on each task, and estimates each worker's hourly wage per HIT[3]. The plugin is inspired on prior research and related tools[9].

Equipped with our plugin and the Super Turker criteria, we ran a two-week field experiment. We had real-world novices perform over 25,000 tasks on MTurk by 1,394 requesters, with the experimental group of workers following the Super Turker criteria, and the control group not receiving any additional guidance. Our study uncovered that having novices follow the Super Turker criteria did empower them to increase their income. We finish with design recommendations for tools and platforms to increase crowd workers' wages. We advocate for tools that bring transparency, and also teach workers how to best make use of that transparency.

2 RELATED WORK

2.1 Work Environment on Crowd Markets

Crowdsourcing not only facilitates the generation of ground truth for machine learning [17], but also enables novel crowd-powered technology [6, 36]. Technology companies use crowdsourcing as "ghost work", which is unperceived by end users [28, 56]. However, criticism surrounding crowd markets compares them to sweatshops or "markets for lemons"[15, 38, 71, 78]. Receiving wages that are less than the U.S. hourly minimum wage, which is $7.25 USD, is one of the most significant disadvantages for workers in crowd markets [5, 30, 32, 34, 35, 37, 39, 40, 47, 48, 67]. Besides the negative factor of low wages, requesters create tasks for workers to perform, but are able to arbitrarily reject the submitted work once the labor has been accomplished [28].

Additionally, crowd workers spend a significant amount of time performing invisible and unpaid labor, e.g., acquiring tasks, learning how to perform assigned tasks, and resolving conflicts with the platform or requesters when discrepancies concerning payment occur [25, 29, 30, 68]. One of the main reasons for this is that crowd markets have imposed transaction costs, which were traditionally assumed by companies, onto workers [16, 28]. Transaction costs are the expenses associated with managing the exchange of goods or services. Researchers have coined this situation "algorithmic cruelty" as the algorithms behind the crowd market are generating critical pain points for workers, such as having no recourse if their account becomes unfairly blocked, if their completed work is arbitrarily rejected, or if they are not fairly compensated.

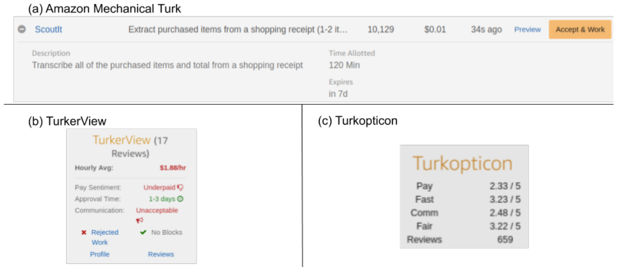
To further complicate the situation, crowd markets do not provide the same information to workers than to requesters. Usually, requesters are granted access to a large amount of information concerning the events in the marketplace; while workers have a much more limited perspective [39, 40]. For example, MTurk allows requesters to view the previous performances and interactions that workers have had on the platform [30]; while workers can only discern very little about what requesters have done previously (e.g., amount of rejected work, amount of unfairly paid tasks, or whether they are frauds[26, 39]).

A consequence of this limited information (coined a "lack of transparency" by researchers) is that crowd workers struggle to find fairly paid tasks or even be paid. Additionally, crowd workers lack basic benefits, e.g., paid sick leave, time off, and health insurance [28, 31]. Also, the work on crowdsourcing platforms typically does not help workers to advance their career [47]. As crowd markets continue to grow, they threaten the hard earned workers' rights attained through the labor movements [31]. Additionally, labor market oversight is more difficult in a crowd market economy[46].

2.2 Tools for Crowd Market Transparency

To begin addressing the unfairness that crowd workers experience, researchers have created tools that bring more transparency to the crowd market (e.g., tools that help workers better comprehend information about requesters and the market in general.) In this paper we refer to these tools as "transparency tools." These approaches believe that through transparency workers can learn how to avoid unreliable requesters and earn better wages [57]. Researchers and practitioners have developed different forums and browser extensions to help workers measure the reputation of requesters (e.g., how they previously interacted with workers) [57]. Crowd workers use Turkopticon [39] and TurkerView [13] to evaluate requesters

[3] Since calculating hourly wage of Turkers has proven a difficult task for researchers in the past, we have released the plugin to help other researchers: http://toxtli.github.io/superturker/


(a) Amazon Mechanical Turk

ScoutIt | Extract purchased items from a shopping receipt (1-2 it... | 10,129 | $0.01 | 34s ago | Preview | Accept & Work

Description
Transcribe all of the purchased items and total from a shopping receipt

Time Allotted
120 Min

Expires
in 7d

(b) TurkerView

TurkerView (17 Reviews)
Hourly Avg: $1.88/hr
Pay Sentiment: Underpaid
Approval Time: 1-3 days
Communication: Unacceptable
✗ Rejected Work ✓ No Blocks
Profile Reviews

(c) Turkopticon

Turkopticon	
Pay	2.33 / 5
Fast	3.23 / 5
Comm	2.48 / 5
Fair	3.22 / 5
Reviews	659

Figure 1: What a turker sees: (a) the HITs, (b) requester's reputation metrics on TurkerView, and (c) requester's reputation metric on Turkopticon.

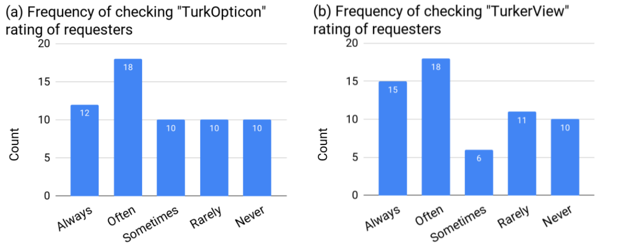


Figure 2: Frequency of how often Super Turkers checked requesters' ratings on Turkopticon and TurkerView.

[44, 68]. Fig. 1 displays the interface used on the Mturk, Turkopticon, and TurkerView. Turkopticon is an opensource tool that allows crowd workers to rate the requesters with 4 "attributes" in a 5 point Likert-scale: generosity ("pay"), promptness ("fast"), communicativity ("comm") and fairness ("fair") [39]. Crowd workers can also leave text descriptions to illustrate each requester and how they interact with workers, as well as the type of tasks they post on the platform. TurkerView is another transparency tool that allows workers to visualize requesters' reputations and offers metrics that are similar to Turkopticon. TurkerView does have some differences. First, it is not opensource. The tool has focused on commercializing its intelligent algorithms that predict how a requester will behave based on her interactions with workers who are using Turkerview[13]. Turkerview also offers a metric that predicts the hourly wage that a given requester is likely to pay (which is not exact per task but provides an overall picture of how that requester operates). Browser extension tools and forums have increased transparency on MTurk. However, despite the availability of such tools, novices still fail to recognize which requester will pay fairly [30, 44]. (These labor conditions might change in the future with the introduction of "One Line Of Code" to automatically ensure fair wages[79]. But these approaches depend on the requester and the platforms wanting to be "fair", which is not always the case[28]).

Together, this suggests that unguided transparency is not sufficient to ensure higher wages. We argue that adopting a Super Turker criteria in a Analytical Hierarchy Process approach offers the necessary guidance to novices to utilize transparency tools to earn higher wages. Our study aims to:

- identify a common and simple criteria that Super Turkers employ for using transparency tools to find work
- guide novices to follow the identified criteria
- increase novices' earning potential

3 UNCOVERING SUPER TURKER PRACTICES

Our goal was to identify a common set of criteria that Super Turkers implemented when using transparency tools. Each Super Turker might value many different criteria. However, understanding that novices progress to experts through repetition[27, 62], we were interested in identifying criteria that were simple and widely accepted, so that novices could easily apply the criteria to earn higher wages. For this purpose, we created a survey that questioned Super Turkers on their criteria for using transparency tools, and then conducted a data analysis over the responses to identify a simple yet popular criteria that novices could easily follow.

3.1 Survey: Method

The survey contained 18 required questions and took 3 to 5 minutes to complete. Our survey was rooted in prior research [65] and based on our research questions. Participants were paid $0.60 USD according to a legal hourly wage of $7.25 USD. The survey was only available to workers, deemed Super Turkers, i.e., workers who had done over 10,000 HITs and who earned more than the U.S minimum wage. Similar to [44, 65], our survey began by asking Super Turkers demographic information and questions about their MTurk work experiences. These questions asked crowd workers about their weekly working hours and how long they had been on MTurk. Next, we asked workers to create flowcharts by selecting and sorting through a list of steps. They denoted the order in which they used different transparency tools and the metrics they analyzed from each tool. For validity purposes, we based the list on prior work that has studied workers' actions around HITs [66, 81] and the metrics that transparency tools, e.g., Turkopticon or TurkerView, share with workers [13, 39]. Participants could either select and sort all steps on the list or select and sort the specific few steps that were key to them. After this question, each Super Turker had a flowchart denoting her process for using transparency tools to select HITs. Next, we studied the importance that Super Turkers gave to different transparency tools and their associated metrics. For each type of transparency tool that Super Turkers stated that they used in their flowchart, the survey asked them how they handled such information. The questions included how often they checked the information and how important the particular metrics associated with the information were. We questioned Super Turkers about the minimum acceptable scores they had for each metric. We used 5-point Likert scale questions to ask Super Turkers about how often they checked the metric (frequency) and the importance they gave to the metric when making decisions. The options in our Likert scales questions were based on the anchors created by Vagias [76]. The minimum acceptable score questions were slider questions that ranged between scores 1 and 5 or "not applicable."

3.2 Survey: Findings

100 Super Turkers completed the survey. To avoid malicious or distracted workers, we added two attention check questions into the

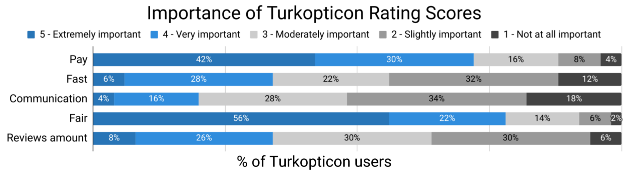


Figure 3: Overview of the importance that Super Turkers gave to different "Turkopticon" metrics about requesters: *fair* was the most important, followed by *pay*.

survey. This resulted in us keeping 68 responses for the analysis, the other responses were excluded because of failed attention checks.

3.2.1 Understanding Super Turkers' Key Transparency Metrics. 88% of the Super Turkers that participated in our survey stated that they used transparency tools. Fig. 2 presents an overview of how frequently Super Turkers used each of these tools to access transparency information about requesters. Super Turkers primarily used both TurkerView and Turkopticon plugins to evaluate requesters, although TurkerView was used slightly more frequently than Turkopticon. Next, we analyzed the type of metrics that Super Turkers took into account when using these tools.

Fig. 3 presents an overview of how important each Turkopticon metric was for Super Turkers. The most important metrics were *TO_fair* and *TO_pay* (Notice that we use the term "*TO*" to distinguish Turkopticon metrics from TurkerView "*TV*" metrics). More than half of the respondents (78%) considered that the *TO_fair* metric was either "5 - extremely important" or "4 - very important" when selecting HITs. Meanwhile, 72% of the respondents deemed the *TO_pay* metric as "5- extremely important" and "4 - very important." *TO_fair* describes whether the requester rejects or approves the work in a fair way, and *TO_pay* metric represents how well the requester pays. On Mturk, requesters can gratuitously reject workers and can indiscriminately refuse payment upon completion of the tasks. Thus, this metric measures how likely is an individual requester to reject or accept work in reasonable (fair) manner.

After understanding how important these different transparency metrics were to Super Turkers, we sought to understand the values that Super Turkers expected or looked for in each metric. Recognizing these thresholds can help novice workers to better utilize the metrics and be more effective at finding fair HITs. Our results indicated that Super Turkers had different expectations for each metric. 92% of our Super Turkers who used Turkopticon expressed that they had a basic requirement score for *TO_fair* (e.g., they only considered HITs who had a *TO_fair* score above a certain threshold), and 90% of the Super Turkers had a basic requirement score for *TO_pay*. The average requirement score of *TO_fair* was 3.69 (SD=0.78) and of *TO_pay* was 3.2 (SD=0.9).

We conducted a similar analysis for TurkerView. Fig. 4 shows that for Super Turkers, the most valuable transparency metric on TurkerView was: *TV_hourly_pay*, followed closely by *TV_fair*, *TV_rejection_rate* and *TV_block_rate*. (Note the use of "*TV*" in this case.) The *TV_hourly_pay* metric estimates the hourly wage in USD that a requester will pay Turkers. This is based on previous workers' experiences performing tasks. To calculate this metric, TurkerView averages the hourly wage of each HIT posted by the requester and the amount of time it took workers to finish the task. *TV_fair* is a metric denoting whether workers believe that a requester pays fairly for the assigned task. *TV_rejection_rate* and *TV_block_rate* display the frequency that a particular requester rejects or blocks workers. When rejected or blocked by a requester, workers not only lose the remuneration for the completed work, but they also receive a bad record on MTurk. This hurts workers future employment opportunities. When analyzing the details of the metrics, we find that Super Turkers, on average, only accept the tasks of requesters whose *TV_hourly_pay* was over $8.29 USD/h (SD = 3.03).

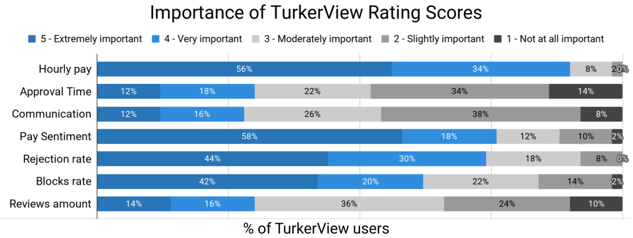


Figure 4: The importance of each "TurkerView" reputation attributes. *Hourly pay* and *Fair* are crucial when Super Turkers select a HIT to work.

However, we must also contend with the situation that TurkerView and Turkopticon do not hold information about all requesters on the platform. When this is the case, the primary metric to measure whether a HIT is worth doing is the reward it offers and the description of the task. From our survey, 82% of Super Turkers had fundamental requirements on the metrics of the HIT rewards (i.e., for them to do a HIT, the reward needed to be above a certain threshold.) The minimum acceptable reward for a task averaged $0.23 (SD = 0.23, median = 0.2). Note that Super Turkers took into account the overall reward rather than the hourly wage, because MTurk only provides information on rewards (how much the requester will pay in total if the worker completes the HIT).

3.2.2 Computing Super Turker Transparency Criteria. Once we had an overview of the type of transparency metrics that Super Turkers considered, we wanted to identify the sequential order in which they evaluated these metrics. It is possible that Super Turkers have various sequences for how they use the available transparency metrics. Our goal was to identify criteria that was common (i.e.,used by many Super Turkers) and concise. Most important was a concise set of criteria for novices to efficiently and effectively implement. For this purpose, we took all the flowcharts that Super Turkers had generated, converted the steps into a text sequence, and used the longest common subsequence algorithm [43] to identify the criteria that was common and shortest among the Super Turkers in our study. The algorithm computed the following criteria:

- Work only with requesters whose:
  - "hourly pay" on TurkerView is over $8.29 USD/h (averaged from values that Super Turkers provided for this metric).
  - If such transparency data is unavailable:
    - work only with requesters whose "fair"score on Turkopticon is over 3.69 (averaged).
  - If such transparency data is unavailable:
    - perform tasks with reward > $0.23 USD (averaged.)

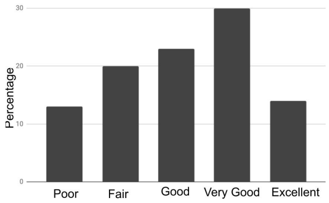


Figure 5: Super Turkers' view on the computed criteria. Most Super Turkers approved the criteria.

Notice that the computed criteria defines a hierarchy of transparency metrics. The hierarchy helps in this case because not all transparency metrics are always available to workers, i.e., some can be missing. The criteria offers a way to potentially find "good work" even under labor market conditions of limited information.

3.2.3 Super Turkers' Impressions of the Criteria. We also asked Super Turkers for feedback on the computed criteria (see Fig 5). In general, Super Turkers approved the criteria. They also approved of doing HITs that paid slightly more than $0.23 cents when no other information about the requester was available. Their logic, in such a situation, was to accept a task that paid slightly higher than the average HIT in order to start earning money rather than waiting. For instance, one Super Turker mentioned:

"Being somewhat informed about a requester and the quality of work will help you make more money. Sometimes you just have to go with it because otherwise you won't make anything." -Super Turker A

Super Turkers commented that accepting HITs that offered lower than their preferred pay just to ensure income was at times necessary, especially as the market might not offer anything better:

"I think it's a good strategy, but sometimes you may have to do work for requesters whose hourly pay is lower. Sometimes you have to do what work is available." -Super Turker B

The Super Turkers who found the criteria "poor" or "fair" were primarily workers whose strategy focused on doing batch HITs. They felt the criteria did not represent their process. This is expected given that the algorithm focused on computing a criteria that was most common; instead of something that was representative of all Super Turkers. But the criteria we identified served the purpose of being computationally appropriate in representing the majority.

"...Setting it at $0.23 may cause workers to miss out on an excellent, one click batch that may be $0.05..." -Super Turker C

Given these positive results, we tested the criteria with novices in the real world to see how it might help increase their wages.

4 NOVICES USE SUPER TURKER CRITERIA

The first study allowed us to compute a common and concise criteria of how the average Super Turkers decided what transparency metrics to consider. The criteria was algorithmically constructed based on the input from Super Turkers.

After computing the Super Turker criteria, we conducted a two-week field experiment on MTurk and investigated:

- Do novices following the Super Turker criteria perform individual tasks that pay more?
- Do control group novices, who utilize their own criteria, discover upon the identical Super Turker criteria?
- Do novices following the Super Turker criteria increase their hourly wage?

4.1 Field Experiment: Methods

Our field experiment followed a randomized control-group pretest-test design that is characterized by being similar to a between subjects study, but with the addition that measurements are taken both before and after a treatment [18]. This setup facilitates better understanding of the change generated from experiments. For this purpose, we split our experiment into two stages: a six-day pretest stage to understand novices' behavior and wages before our intervention, and a six-day test stage to understand novices' behavior and wages after our intervention (i.e., after telling novices to follow the criteria). We divided the subjects into two groups:

(1) **Control group.** During the entire study (i.e., throughout the pretest and test stages), novice crowd workers used transparency tools and completed tasks as normal;
(2) **Experimental group.** In the pretest, novice crowd workers used transparency tools and completed tasks as normal; but, in the test stage, they were instructed to follow the decision-making criteria of Super Turkers.

We recruited 100 novice Turkers. Similar to prior work, we considered novices were workers who had completed less than 500 HITs [12, 75]. The completion of 500 HITs is also within the probation period of MTurk [1, 63] We randomized novices across each of our two conditions (50 workers in each groups). All novices in our study reported using Turkopticon and TurkerView (which is aligned with the findings of previous work that identified that most workers are using transparency tools [44]).

Pretest Stage. In this stage, participants across conditions were asked to: (1) install a Google Chrome plugin we developed; (2) perform HITs as normal. The plugin allowed us to track participants' behavior (types of tasks they did, requesters they worked with, and earnings made.) We used this information to study how workers' wages changed over time and track how they utilized transparency tools. Participants were allowed to uninstall our plugin at any time, and we rewarded them for the period of time that they had our plugin installed on their computer. We paid novices $0.60 USD for installing our extension, accounting for the US federal minimum wage ($7.25/hour) as installation took less than 4 minutes.

Test Stage. In this stage, novices in the control group were asked to continue working using their customary method; while in the experimental group, novices were asked to make decisions using the identified Super Turker criteria. Participants were also informed that the criteria came from Super Turkers and could help them to increase their hourly wages. For this purpose, at the beginning of the test stage, we emailed participants and informed them of the activity they would do with us that week. Similar to prior work that has run field experiments on MTurk, we paid participants in the experimental group another $0.5 for reading the criteria and an additional bonus of $0.10 each time they followed the criteria to select a HIT and completed it. On average, we paid participants a total of $4.90 for following the criteria; the participant who received the most for following the criteria earned $24.70. Similar to [20], only participants in the experimental group were paid extra to follow the criteria. In both the control and experiment

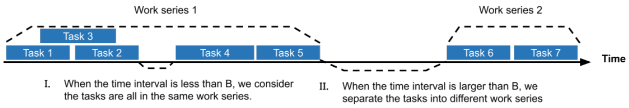

Work series 1
Work series 2
Task 3
Task 1
Task 2
Task 4
Task 5
Task 6
Task 7
Time
I. When the time interval is less than B, we consider the tasks are all in the same work series.
II. When the time interval is larger than B, we separate the tasks into different work series

Figure 6: We separate work series based on time intervals: When the time interval is less than B, we consider the tasks are all within the same work series. If not, we separate them into different work series.

group, we continued paying participants each day they kept our plugin installed. To avoid our remuneration interfering with our study, we paid these bonuses at the end of our experiment and did not include our HIT remuneration when calculating workers' hourly wages.

4.1.1 Collecting and Quantifying Workers' Behavior. For our study, we needed methods for: (1) collecting and quantifying workers' behaviors and the HITs they performed; (2) flagging when workers utilized the Super Turker criteria; (3) measuring how much workers' hourly wages changed when following the criteria. We created a Google Chrome extension[3] (i.e. plugin) to collect crowd workers' behavior on Mturk. The plugin tracked the metadata about the HITs that workers previewed or accepted, and the timestamps of when workers accepted, submitted or returned HITs (i.e., tasks which were accepted but for various reasons not completed). In specific, the plugin collected:

- HIT information, such as title, rewards, timestamps (accept/ submit/ return), requester IDs, HIT Group IDs, and HIT IDs.
- Worker information, such as daily earnings on the dashboard, installed extensions, approval rate, and worker IDs.
- Requester reputation information, such as ratings on Turkopticon and TurkerView.

There were certain things our plugin did not record: the required specific qualifications for a HIT and the HITs that participants decided to not take (preview or accept). To maintain workers' privacy, our browser extension also did not record workers' browsing records outside of MTurk or workers' personal data (such as worker ID, qualifications, among other personal metrics).

4.1.2 Identifying Whether Workers Follow The Criteria. Once we had collected and quantified the completed tasks of novices, our goal was to identify the novices who had followed the Super Turker criteria. To accomplish this, we took the transparency data available for each task that novices completed (i.e., TurkerView's expected hourly wage, Turkopticon's fairness score, the HIT reward) and used the flowchart presented in section 3.2.2 to determine whether the HIT followed the criteria. We name HITs that meet the established criteria as Super HITs. After this step, each novice had a list of Super HITs and non-Super HITs associated with them. We considered that novices with over 70% of their HITs labeled as Super HITs to be workers who followed the Super Turker criteria. For our study, the threshold of 70% was selected based on prior research that has established this amount as an adequate threshold to measure whether people are following new behavioral patterns [23, 55]. After this step, we had a list of novices in our experimental group who had followed (or not) the criteria.

4.1.3 Measuring Workers' Hourly Wages. After compiling the list of novices who followed the criteria and those who did not, our goal was to calculate each novice's hourly wage and determine whether the workers following the criteria increased their wages. To calculate the hourly wages of a worker we need to measure: (a) the income that a worker earned; and (b) the amount of time they worked to earn that income.

A. Total Earnings. A worker's earnings does not come solely from HIT compensation (i.e., the salary that MTurk states they will pay when HITs are completed). Workers might also earn HIT bonuses, i.e., additional rewards from requesters whose amount is usually unknown before performing the HIT. To record both reward and bonuses adequately, we logged the "Daily Income" from workers' dashboard which already considers both values directly. Using the values specified in workers' dashboard helps us to consider circumstances where workers' labor might have been rejected, i.e., no payment for completing a HIT. However, there are some issues with tracking workers' wages using the dashboard: not all requesters pay workers as soon as they submit their work. In these cases, the payment can be delayed for several days. To overcome this problem, we checked workers' dashboard three days after the end of the whole experiment to give requesters time to make their corresponding payments. For all participants in our study we calculated their daily wages ($Income_D$) through this method for the twelve days they participated in our study. Using this method, we were able to calculate workers' total earnings.

B. Total Working Time. To calculate how much time a worker required to complete a task and earn specific wages, our extension logged: $\text{Time}_{\text{start}}$: the exact moment when a worker accepts a HIT; and Time_{end}: the moment when a worker returns or submits a HIT. Notice that workers might take breaks [59] and spend time searching for HITs, calculating the working time as simply $\text{Time}_{\text{end}} - \text{Time}_{\text{start}}$ is not appropriate. To overcome this problem, we adopt an approach similar to [30], where we consider that a worker is doing series of tasks continuously if their time interval is less than B minutes (as shown in Figure 6.I) and consider they have started a new series of tasks if the time interval is larger than B minutes (as shown in Figure 6.II). When calculating the hourly wage, we set the between time (B) between the task series as 12 minutes because in normal industries employees must be paid for any break [59]. The tasks done within the same work series S share the same time interval $\text{Time}_{S,d}$, where d represents the date when that particular work series began. We measure the time interval for series S as follows:

$$\text{Time}_{S,d} = \max_{s \in S}\{\text{Time}_{\text{end},s,d}\} - \min_{s \in S}\{\text{Time}_{\text{start},s,d}\}. \tag{1}$$

A worker's total number of work hours in day D, ($WorkingHour_D$), is the sum of the working time of all the series they did on day D:

$$WorkingHour_D = \Sigma_{d \in D}\text{Time}_{S,d} \tag{2}$$

For a given worker w, her overall hourly wage for day D is:

$$w_D = \frac{\text{Income}_D}{WorkingHour_D}. \tag{3}$$

After this step, we had the hourly wages earned for all novices; and for novices in the experimental group, we labeled them according to whether or not they followed the criteria.

4.2 Field Experiment: Findings

Our experiment ran for 12 days during late October 2018. A total of 100 unique novice workers participated in our study and were randomized across our two conditions. Participants visited (i.e., previewed or accepted) a total of 25,899 HITs during the two week process. The recorded HITs belonged to 2,568 unique HIT groups posted by 1,394 unique requesters. Novices visited 261 HITs on average, and visited 102 HITs in the median.

4.2.1 Novices, Super Turker Criteria, And Hourly Wages of HITs. In the previous section, we established a common and concise criteria constructed from surveying Super Turkers. However, it was unclear whether following the criteria would guide novices to perform individual HITs that actually paid them more per hour.

Part of the criteria was to select tasks that TurkerView predicted would likely earn workers a particular hourly wage. However, it was yet unclear whether TurkerView offered available information on these individual HITs. To better understand the ecosystem in which our novices operated and contextualize the availability of transparency information, we analyzed the details of all the HITs novices completed. In our field experiment, our participants worked for 1,394 different requesters. Although, our statistics demonstrated that only 772 of these requesters had been reviewed on both TurkerView and Turkopticon, 500 requesters were not reviewed on TurkerView, 430 requesters were not on Turkopticon, and 318 requesters had not been reviewed on either TurkerView or Turkopticon. These results demonstrated that for 36% of the requesters for whom novices worked with, there was no information about them on transparency tools. For requesters whose transparency information was missing, the Super Turker criteria recommends analyzing the reward of the HIT, i.e., the default metric on MTurk, and only performing HITs that paid more than $0.23.

Additionally, even when TurkerView's expected hourly wage was available, it is calculated for average workers (i.e., TurkerView considers the average time it takes all workers to complete tasks for a particular requester and based on this calculates the expected hourly wage given what that requester typically pays.) However, it is unclear whether this average length of time would also pertain to novices. It could potentially take novices more time to complete certain tasks, and hence they would earn less.

In order to investigate whether novices following the Super Turker criteria performed tasks that actually paid them more per hour, we took all the HITs that novices completed and calculated the real hourly wage that novices received for the HITs. This was based on how much time it took them to perform the task, and the actual amount of money they received for finishing the task. Next, we identified the tasks that were Super HITs and those that were not, and we compared the hourly wage between these two groups.

We first focused on inspecting the HITs that lacked transparency information. Across conditions, novices in our study accepted and submitted 9,503 HITs. Of these, there were 979 HITs from requesters who lacked data on Turkopticon and TurkerView. From this group there were 232 Super HITs (i.e., HITS that rewarded more than $0.23) and 747 HITs that rewarded less than $0.23 (non Super Hits). We conducted a Mann-Whitney U test to compare the real hourly pay between these two groups given that the sample sizes of these two kinds of HITs were different and they presented a large standard deviation. The Mann-Whitney U test showed that there was a significant difference in the median hourly wage between Super HITs and non-Super HITs when workers can only access the information about the rewards (U=68667, p<0.0001). The median hourly wage that Super HITs paid was $3.97, while non-Super HITs paid $2.76. Thus, following the SuperTurker criteria can guide novices in their decision-making to identify individual HITs that pay them higher hourly wages, even when transparency data about the requesters posting the HITs is missing.

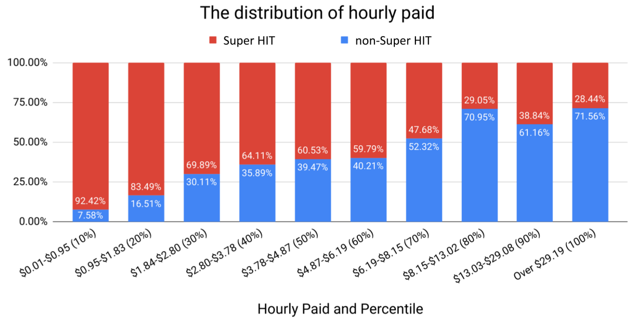


Figure 7: Distribution of the hourly wage of all HITs that novices selected that had transparency data available. Super HITs tended to provide a higher hourly wage.

Next, we compared the real hourly wages of Super HITs and non-Super HITs that had available transparency data. Figure 7 showcases the hourly wage that workers earned for Super HITs (N=4,047, 42.6% of the submitted tasks) and for the HITs that did not follow the criteria (N=5,456, 57.4% of the submitted tasks). We eliminated the outliers and computed a t-test to compare the difference between these two groups. The t-test showed that there was a significant difference in the actual hourly wage that workers received for completing Super HITs and non-Super HITs ($t(5806)=11.151$, $p < 0.0001$). This highlights, that novices who follow the Super Turker criteria are able to distinguish which individual HITs, when performed, will pay them higher wages per hour.

4.2.2 SEARCHING TIME AND THE SUPER TURKER CRITERIA. Our previous analysis uncovered that individual Super HITs had a higher hourly wage than non-Super HITs. In general, the median hourly wage of all Super HITs was $7.04/h , while the median hourly wage of all non-Super HITs was $3.27. Note this hourly wage only considers the time that workers spent in completing a task, not the uncompensated time that workers need to expend to search for HITs. Crowd markets have placed these costs, that were traditionally absorbed by companies, onto workers [28]. Such costs include the time that workers spend searching for work. In a micro job atmosphere, these costs to workers pose a serious issue.

In this analysis, we were interested in studying whether the quest for Super HITs could possibly lead novices to spend a greater amount of time searching for work, hence reducing their hourly wages (considering that the hourly wage involves the time workers spend searching for work in addition to the time completing work.) In this setting, we considered that the searching time includes searching for HITs in the HIT pool, previewing HITs, and accepting HITs but not yet submitting them. We identified that novices for a

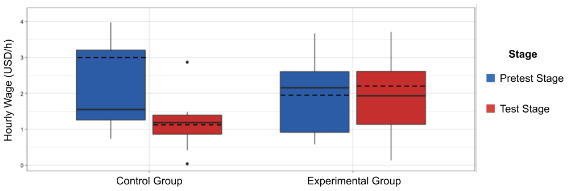


Figure 8: Box plot comparing the hourly wage of novices in both groups. The solid line denotes the median value and the dashed line denotes the mean value. Novices following the Super Turker criteria increased their wages more.

given task spent a mean and median searching time of 168 seconds (SD=388) and 20 seconds, respectively. Additionally, the searching time for Super HITs and normal HITs was different. The median searching time for normal HITs was 9 seconds, while the median searching time for Super HITs was 61 seconds. Next we calculated novices' total working time as time spent searching for work plus time spent completing tasks. Through our analysis, we identified that the median hourly wage for Super HITs, when considering searching and working time together, was $4.12/h; while for non-Super HITs this wage was $2.13/h. Therefore, it appeared that while there was an overhead cost imposed upon workers for searching for Super HITs (i.e. avoiding non-Super HITs), the overhead cost was worth it for increasing the hourly wage.

4.2.3 Novices Discovering The Super Turker Criteria Independently. Next, we examined whether novices had already adopted the decision-making criteria of the Super Turkers before instructing them to follow it. If this were the case, it might be unnecessary to teach novices how to effectively use transparency tools to decide which tasks might pay more. For this purpose, we analyzed the meta-data of all the HITs that novices decided to perform in the pretest stage (i.e., we studied the HITs novices accepted and submitted). From this, we inspected the number of HITs that were Super HITs. We identified that a minority of novice workers in our study (25%) were already unwittingly utilizing the Super Turker criteria.

4.2.4 The Super Turker Criteria And Novices' Hourly Wages. In this analysis, we focused on comparing the actual hourly wages that were received by novices following the Super Turker criteria compared to what control group novices received. However, 32 control group participants and 19 from the experimental group uninstalled our plugin before the test stage started. This was likely due to the lengthy nature of our study. For the remaining 49 participants, we identified that 7 control group participants were already using the Super Turker criteria from the beginning (i.e., during pretest stage), and 12 of the participants in the experimental group never followed the identified criteria. Given that we were interested in studying the change of wages that occurred after utilizing the Super Turker criteria, we discarded the above participants' data. In the end, 21 experimental group participants and 11 control group participants were remaining for our analysis.

From these participants, we studied how much their hourly wage changed between the pretest and test stages. For all novices, we calculated the change of their hourly wage as the median hourly wage they received in the test stage minus their median hourly wage in the pretest stage. Fig 8 presents an overview of how much the hourly wage of novices changed in both the control and the experimental group. During the pretest stage: the control group had a median hourly wage of $1.55 and mean of $3.00; while the experimental group had a median hourly wage of $2.16 and mean of $1.95. In the test stage, the median and mean hourly wage that control group novices earned reduced to $1.19 and $1.17 respectively. Meanwhile, the median and mean hourly wage of novices in the experiment group increased slightly to $1.93 and $2.20 respectively.

We computed a Mann-Whitney U test to examine whether the difference in observed wages between the control and experimental groups was significant. We used the Mann-Whitney U test given that both the control and experimental groups were independent, had different variances, and presented small sample sizes. Moreover, The Mann-Whitney U test does not compare mean scores but median scores of two samples. Thus, it is much more robust against outliers and heavy tail distributions. The Mann-Whitney U test showed that there were significant differences in how much the median hourly wages of novices varied between the control and experimental groups ($U = 48, p = 0.006$).

Next, we investigated whether the change in wages that each condition presented between pretest and test stages was significant. A Wilcoxon signed-rank test highlighted that the change in wages in the experiment group was not significantly different (Z=115, p=1.00). However, the change in wages that the control group presented was significantly different (Z=61, p=0.04). This finding might hint that the general pool of HITs during the test stage had worse remuneration than the pretest; hence, this explains why we witnessed a decrease in control group wages since there was no intervention. To better understand this ecosystem, we examined the hourly wages of all HITs in the pretest and test stages. In general, the mean and median hourly pay of all HITs in the pretest stage was $11.68/h (SD=44.92) and $4.73/h. The mean and median hourly wage in the test stage was $14.877 (SD=92.66) and $4.59/h. Between stages, there was no significant difference in how much HITs paid novices per hour (U=5,280,800, p=0.34).

Given that the hourly wage of the HITs appeared to be in general the same in the pretest and test stages, it remained unclear why the control group had decreased their wages so drastically in the test stage. We decided to investigate further. We studied the amount of time workers spent searching for work in the pretest and the test stages. A change in searching time could denote that workers might have experienced a harder time finding tasks to perform, even if there was no change in the rewards that they received for the tasks. In general, workers could have seen a drop in their earnings simply due to not enough available tasks for that week. For this purpose, we calculated for each stage the average time that all workers spent searching for tasks. We discovered that there was a significant difference between the amount of time that workers spent searching for HITs in pretest vs the test stages ($U = 4,109,600, p < 0.001$). Participants spent 158 seconds on average (median=16, SD=379) to find a HIT in the pretest stage; and 262 seconds (median=64, SD=436) to find a HIT in the test stage. This increment could be attributed to the fact that identifying Super HITs is more time consuming than searching for normal HITs.

To understand more deeply what was taking place, we compared the difference in searching time for the Super HITs and non-Super

HITs. In the pretest stage, the median searching time for Super HITs was 46 seconds; while it took 9 seconds for non-Super HITs. In the test stage, the median searching time that the control group took to identify Super HITs was 131 seconds; while it took 121 seconds for non-Super HITs. Note that we analyzed all HITs in the pretest stage; however, we only analyzed the control group HITs in the test stage. This was done to avoid the influence of our intervention (i.e., the experimental group in the test stages cannot represent the regular searching time due to the fact that we informed them to search according to the criteria). We identified that the searching time for both Super HITs and non-Super HITs increased from pretest stage to test stage. This result suggests that, in general, MTurk likely had fewer HITs available for novices in the test stage. This was why, on average, the novices searching time increased. However, despite this adversity, novices following the Super Turker criteria maintained their wage level, while the wages in the control group decreased significantly.

5 DISCUSSION

Our experiments demonstrate the potential of using the criteria of Super Turkers to guide novices on how to use transparency tools to find work on MTurk. The majority of novices following the Super Turker criteria increased their wages, even while novices in the control condition decreased their salaries (likely due to the limited tasks available that week on MTurk.) Our study provides insights into the impact these highly effective workers can have on novices, as well as demonstrating the feasibility of connecting transparency tools with educational opportunities to increase workers' wages. In this section we discuss our results, highlighting especially the challenges and opportunities we envision in the research area.

Super Turker Criteria. Our study uncovered one of the most common and concise set of criteria that Super Turkers adopted to handle transparency to find fair work. It was computationally derived from surveys of Super Turkers and helped novices decide which MTurk tasks to perform by utilizing transparency tools as a means to earn higher wages. Our model was based on previous studies that demonstrated how "shepherding" novices could help them to improve their labor [21].

An interesting observation on the particular Super Turker criteria that we uncovered, is that it considered the circumstance that when the hourly wage metric was missing, it was best to look at Turkopticon's fairness value instead of inspecting other metrics related to how well the requester paid. Fairness on Turkopticon regards whether the requester will reject or accept a worker's labor. Crowd markets have traditionally contained power imbalances where workers have limited power in comparison with requesters and platform owners [28]. Part of the imbalances arise because most crowd markets are very concentrated: almost 99% of all tasks are posted by 10% of the requesters, who do not have to negotiate with workers about whether or not they will accept or reject their labor. If workers and their work are rejected by a requester, workers generally suffer from a lack of accountability in response to their complaints or attempts at restitution from requesters and platform owners. This can translate into wasted time, loss of a paycheck, and little opportunity to raise awareness about possible exploitation [72]. According to Pew research, 30% of all gig workers have experienced non-payment at least once [28]. The US Freelancer Union reports a much larger number, where allegedly 70% of current freelancers have at least one client who has not paid them [33]. This could explain why the criteria recommended inspecting Turkopticon's fairness score. This step in the criteria, in particular, attempted to ascertain which situations needed to be avoided by Turkers in order to prevent non-payment or rejected labor.

The Super Turker criteria we identified considered that if the Turkopticon fairness score and the TurkerView hourly wage were missing, the best course of action was to look directly at the HIT's reward. One reason for this strategic behavior by Super Turkers is that for every minute they spend analyzing a HIT's transparency metrics, they are losing out on the chance to earn money (ultimately reducing their hourly wages). Notice also that lower reward HITs could still provide high hourly wages, especially because the distribution of the hourly wage is a near normal distribution given a specific reward amount [30]. In this setting, Super Turkers might be betting that HITs that are paying more than $0.23 will likely provide higher hourly wages. Moreover, following this behavior will likely reduce the unpaid time spent searching for labor.

Field Experiment. While transparency is now available to Turkers via different plugins, workers still receive less than the minimum wage [30, 73]. We believe that part of the problem is that workers likely cannot interpret or evaluate the value of their work and its relationship with the transparent information they are now observing. Several workers might also not necessarily have the analytical skills to interpret all of the transparency metrics that tools, like Turker View, provide [41]. Our aim with this research was to identify a practical way to use transparency in MTurk and study if that could help crowd workers to increase their salaries. Our field experiment highlighted that guiding novices in their decision-making by following the Super Turker criteria lead them to earn wages that were higher than what they would earn working on their own. Our results emphasize that transparency is required but it is not sufficient. Utilizing transparency skillfully can transform salaries on MTurk. Our method helped novices to earn more. We identified that the difference in hourly wage between the control group and the experimental group was significant.

Observing the decrease in wages of the control group during the test stage, we recognize that workers' salaries are dependent on task availability. Thus, we suspect the hourly wage decline may be attributed to the fluctuation of the task pool. The composition of the tasks available to workers was different each day during the time period we ran our experiment. The fact that tasks are different from day to day in crowd work is documented [44]. MTurk does not guarantee that there are enough well-paid tasks every day and this issue poses a difficulty for crowd workers[4]. Through our analysis we identified that novices in the control group spent a greater amount of time searching for HITs during the test stage. The time spent searching for work is time where workers are not paid. Hence, novices in the control condition saw their hourly wages reduced as they spend more unpaid time searching for work. However, even during a phase where there might have been limited high paying tasks, our experimental group was able to increase their wages.

As we think in practice about how to deploy these strategies at scale [50], it can be important to consider how enforcing strategies that make use of commercial tools, such as Turker View, could potentially create further social divisions on MTurk [80] (especially, as only workers who could afford to pay for such tools could follow the strategy). Moving forward, we plan to explore strategies with only opensource platforms and that are aware of workers' privacy and autonomy concerns[28, 42, 54].

Difficulties in promoting the Super Turker Criteria. Our field experiment also helped us to identify first hand the difficulties in getting novices to follow the decision-making criteria from Super Turkers (even though following the criteria had the added incentive of potentially higher wages). Half of the novices in our experimental group did not follow the instructions from Super Turkers. One of the reasons for this might be that novices simply did not have the qualifications to perform the HITs that satisfied the criteria that Super Turkers recommended. Therefore, even if they wanted to follow the criteria they might not have been able to access and do the related HITs. In order to protect workers' privacy, our plugin did not track worker qualifications or the qualifications that tasks imposed on workers, we believe there is likely value in exploring interfaces that help crowd workers gain the qualifications they need to access higher paying jobs [44]. However, it is interesting to note that 75% of the Super Turkers in our study lacked master Turker qualifications. It is unclear which type of qualifications novices should strive for in order to successfully perform higher paying tasks. Future work could explore how qualifications affect the type of labor and wages novices can access.

Another reason why workers might not have followed the Super Turker criteria is that we posted and shared the criteria as requesters. Workers might have viewed the Super Turker instructions as part of the task they were doing for us and not as something that was meant to really support them. As a result, their motivation for following the criteria might have been lacking. Previous education research has shown that students who use reciprocal teaching strategies (i.e., strategies where students share with each other advice) tend to have better performance than students working on their own or even students who are working directly with a professor [70]. Similarly, recent crowdsourcing research has highlighted that increasing the interactions between workers can help novices to more easily develop their skills and grow [12, 19, 75]. In future work, we will examine different interfaces for sharing Super Turker criteria with workers to increase the adoption of the criteria.

Design Implications and Future Work. We observed that novices could improve their wages by imitating how experienced workers used transparency tools. We believe there is value for researchers and practitioners to build systems that teach novices the transparent information that is pertinent to achieving their goals. The objectives of these systems are not only to recommend to novices the tasks they should perform, but also to help them understand and learn what kind of tasks pay fairly (potentially extrapolating such knowledge to other crowd markets and online spaces). We also believe there is value in creating on-the-job tutorials that can guide workers on the type of labor they should perform. Designing such tutorials is time-consuming but crucial for empowering workers, who often have limited time and resources, to earn higher wages. Future work could explore designing data driven tutorials that are generated in part based on the patterns of effective workers. There might also be value in exploring educational material that has been generated for audiences with time constraints [22].

Additionally, to build more trust and participation on crowd markets, it might also be worth to explore transparent interfaces that can inform the different actor of a marketplace just how much each actor is being fair and respectful of others' values [11]. Future work could explore other ways of recruiting Super Turkers and eliciting information from them, e.g., via video recordings or interviews[60]. Such studies could explore how using different mechanisms for eliciting information relates to the type of information that is obtained from Super Turkers. Related, it might be interesting to specifically investigate other types of Super Turker criteria that might exist. For instance, investigate how Super Turkers use transparency tools when multitasking[80]. Future work could also explore what happens in other crowd platforms (e.g., Uber, Upwork, or Citizen Science platforms) when novices adopt the strategies from high earning participants or the strategies from accounts who are contributing the most [2, 14, 52, 64, 69].

Limitations. We conducted a real world experiment which is not simple given the limited availability of HITs and lack of information provided by MTurk about workers' hourly wages [30]. The issue of limited and inconsistent HIT availability has been documented in other research and we experienced, firsthand, the possible implications this has on workers' wages and on conducting "clean" research [4]. Notice also that we recruited Super Turkers who were willing to engage in surveys on MTurk (missing those who do not do surveys). Additionally, our algorithm focused on computing a criteria that was commonly used and simple to implement. Therefore, our criteria did not represent all Super Turker behavior. For instance, some Super Turkers might only do HIT batches that pay $0.01 cent and can be completed in less than 10 seconds, resulting in an hourly wage of approximately $36/hour. Nonetheless, given that our goal was to identify one of the strategies that Super Turkers adopted, and study how it plays out when used by novices, we considered our approach to be computationally appropriate and representative. Notice that our study focused on breadth instead of depth to start to shed needed light on how Super Turkers use transparency and how this plays out when adopted by novices. Future work could conduct longitudinal studies inspecting the amount of time it takes novices to adopt on their own some of the Super Turker strategies vs guiding them to adopt the strategies from the start. Having task interruptions and multitasking are dependent on people's work style and preferences [53, 80]. Future work could explore how strategies with multitasking differ from strategies with only monotasking.

ACKNOWLEDGMENTS

Special thanks to Amy Ruckes for the immense feedback and iterations on this work. Thanks to Caroline Anderson, Pankaj Ajit for helping us to start exploring this area. This work was partially supported by NSF grant FW-HTF-19541.

REFERENCES

[1] 2017. [Guide] - Welcome to the world of Mechanical Turk. Retrieved June 26, 2019 from https://forum.turkerview.com/threads/welcome-to-the-world-of-mechanical-turk.112/

[2] aaai Fall Symosium 2019. [n.d.]. Solving AI's last-mile problem with crowd-augmented expert work. ([n. d.]).

[3] Janine Berg. 2015. Income security in the on-demand economy: Findings and policy lessons from a survey of crowdworkers. *Comp. Lab. L. & Pol'y J.* 37 (2015), 543.

[4] Janine Berg, Marianne Furrer, Ellie Harmon, Uma Rani, and M Six Silberman. 2018. Digital labour platforms and the future of work: Towards decent work in the online world. *Geneva: International Labour Organization* (2018).

[5] Birgitta Bergvall-Kåreborn and Debra Howcroft. 2014. Amazon Mechanical Turk and the commodification of labour. *New Technology, Work and Employment* 29, 3 (2014), 213–223.

[6] Jeffrey P Bigham, Chandrika Jayant, Hanjie Ji, Greg Little, Andrew Miller, Robert C Miller, Robin Miller, Aubrey Tatarowicz, Brandyn White, Samual White, et al. 2010. VizWiz: nearly real-time answers to visual questions. In *Proceedings of the 23nd annual ACM symposium on User interface software and technology.* ACM, 333–342.

[7] Marko Bohanec. 2009. Decision making: A computer-science and information-technology viewpoint. *Interdisciplinary Description of Complex Systems: INDECS* 7, 2 (2009), 22–37.

[8] John Bohannon. 2011. Social science for pennies.

[9] Chris Callison-Burch. 2014. Crowd-workers: Aggregating information across turkers to help them find higher paying work. In *Second AAAI Conference on Human Computation and Crowdsourcing.*

[10] Krista Casler, Lydia Bickel, and Elizabeth Hackett. 2013. Separate but equal? A comparison of participants and data gathered via Amazon's MTurk, social media, and face-to-face behavioral testing. *Computers in human behavior* 29, 6 (2013), 2156–2160.

[11] Chun-Wei Chiang, Eber Betanzos, and Saiph Savage. 2018. Exploring blockchain for trustful collaborations between immigrants and governments. In *Extended Abstracts of the 2018 CHI Conference on Human Factors in Computing Systems.* 1–6.

[12] Chun-Wei Chiang, Anna Kasunic, and Saiph Savage. 2018. Crowd Coach: Peer Coaching for Crowd Workers' Skill Growth. *Proceedings of the ACM on Human-Computer Interaction* 2, CSCW (2018), 37.

[13] ChrisTurk. 2018. *TurkerView.* Retrieved November 4, 2018 from https://turkerview.com/

[14] Kevin Crowston, Erica Mitchell, and Carsten Østerlund. 2019. Coordinating advanced crowd work: Extending citizen science. *Citizen Science: Theory and Practice* 4, 1 (2019).

[15] Ellen Cushing. 2012. Dawn of the digital sweatshop. *East Bay Express* 1 (2012).

[16] Valerio De Stefano. 2015. The rise of the just-in-time workforce: On-demand work, crowdwork, and labor protection in the gig-economy. *Comp. Lab. L. & Pol'y J.* 37 (2015), 471.

[17] 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 conference on computer vision and pattern recognition.* Ieee, 248–255.

[18] Dimiter M Dimitrov and Phillip D Rumrill Jr. 2003. Pretest-posttest designs and measurement of change. *Work* 20, 2 (2003), 159–165.

[19] Mira Dontcheva, Robert R Morris, Joel R Brandt, and Elizabeth M Gerber. 2014. Combining crowdsourcing and learning to improve engagement and performance. In *Proceedings of the 32nd Annual ACM Conference on Human Factors in Computing Systems.* ACM, 3379–3388.

[20] 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.* ACM, 2623–2634.

[21] Steven Dow, Anand Kulkarni, Brie Bunge, Truc Nguyen, Scott Klemmer, and Björn Hartmann. 2011. Shepherding the crowd: managing and providing feedback to crowd workers. In *CHI'11 Extended Abstracts on Human Factors in Computing Systems.* ACM, 1669–1674.

[22] Lizbeth Escobedo, David H Nguyen, LouAnne Boyd, Sen Hirano, Alejandro Rangel, Daniel Garcia-Rosas, Monica Tentori, and Gillian Hayes. 2012. MOSOCO: a mobile assistive tool to support children with autism practicing social skills in real-life situations. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems.* 2589–2598.

[23] BJ Fogg. 2009. A Behavior Model for Persuasive Design. In *Proceedings of the 4th International Conference on Persuasive Technology (Persuasive '09).* ACM, New York, NY, USA, Article 40, 7 pages. https://doi.org/10.1145/1541948.1541999

[24] Ernest H Forman. 1990. Multi criteria decision making and the analytic hierarchy process. In *Readings in multiple criteria decision aid.* Springer, 295–318.

[25] Ujwal Gadiraju, Alessandro Checco, Neha Gupta, and Gianluca Demartini. 2017. Modus operandi of crowd workers: The invisible role of microtask work environments. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 1, 3 (2017), 1–29.

[26] Ujwal Gadiraju and Gianluca Demartini. 2019. Understanding Worker Moods and Reactions to Rejection in Crowdsourcing. In *Proceedings of the 30th ACM Conference on Hypertext and Social Media.* 211–220.

[27] Ujwal Gadiraju, Besnik Fetahu, and Ricardo Kawase. 2015. Training workers for improving performance in crowdsourcing microtasks. In *Design for Teaching and Learning in a Networked World.* Springer, 100–114.

[28] Mary Gray and Siddharth Suri. 2019. Ghost Work: How to Stop Silicon Valley from Building a New Global Underclass.

[29] 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.* ACM, 321–329.

[30] 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.* ACM, 449.

[31] Ellie Harmon and M Six Silberman. 2018. Rating working conditions on digital labor platforms. *Computer Supported Cooperative Work (CSCW)* 27, 3-6 (2018), 1275–1324.

[32] Paul Hitlin. 2016. Research in the crowdsourcing age, a case study. *Pew Research Center* 11 (2016).

[33] Sara Horowitz. 2015. The Costs of Nonpayment. Retrieved June 26, 2019 from https://blog.freelancersunion.org/2015/12/10/costs-nonpayment/

[34] John J Horton. 2011. The condition of the Turking class: Are online employers fair and honest? *Economics Letters* 111, 1 (2011), 10–12.

[35] John Joseph Horton and Lydia B Chilton. 2010. The labor economics of paid crowdsourcing. In *Proceedings of the 11th ACM conference on Electronic commerce.* ACM, 209–218.

[36] Ting-Hao Kenneth Huang, Joseph Chee Chang, and Jeffrey P Bigham. 2018. Evorus: A Crowd-powered Conversational Assistant Built to Automate Itself Over Time. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems.* ACM, 295.

[37] International Labour Office (ILO). 2016. Non-standard employment around the world: Understanding challenges, shaping prospects.

[38] Panagiotis G Ipeirotis. 2010. Mechanical Turk, low wages, and the market for lemons. *A Computer Scientist in a Business School* 27 (2010).

[39] Lilly C Irani and M Silberman. 2013. Turkopticon: Interrupting worker invisibility in amazon mechanical turk. In *Proceedings of the SIGCHI conference on human factors in computing systems.* ACM, 611–620.

[40] Lilly C Irani and M Silberman. 2016. Stories we tell about labor: Turkopticon and the trouble with design. In *Proceedings of the 2016 CHI conference on human factors in computing systems.* ACM, 4573–4586.

[41] Mohammad Hossein Jarrahi and Will Sutherland. 2019. Algorithmic Management and Algorithmic Competencies: Understanding and Appropriating Algorithms in Gig work. In *International Conference on Information.* Springer, 578–589.

[42] Mohammad Hossein Jarrahi, Will Sutherland, Sarah Beth Nelson, and Steve Sawyer. 2019. Platformic Management, Boundary Resources for Gig Work, and Worker Autonomy. *Computer Supported Cooperative Work (CSCW)* (2019), 1–37.

[43] Tao Jiang and Ming Li. 1995. On the approximation of shortest common supersequences and longest common subsequences. *SIAM J. Comput.* 24, 5 (1995), 1122–1139.

[44] Toni Kaplan, Susumu Saito, Kotaro Hara, and Jeffrey P Bigham. 2018. Striving to Earn More: A Survey of Work Strategies and Tool Use Among Crowd Workers.. In *HCOMP.* 70–78.

[45] N Kasperczyk and K Knickel. 1996. The analytic hierarchy process (AHP). *Retrieved from* (1996).

[46] Otto Kässi and Vili Lehdonvirta. 2018. Online labour index: Measuring the online gig economy for policy and research. *Technological forecasting and social change* 137 (2018), 241–248.

[47] Anna Kasunic, Chun-Wei Chiang, Geoff Kaufman, and Saiph Savage. 2019. Crowd Work on a CV? Understanding How AMT Fits into Turkers' Career Goals and Professional Profiles. *arXiv preprint arXiv:1902.05361* (2019).

[48] Miranda Katz. 2017. Amazon Mechanical Turk Workers Have Had Enough. https://www.wired.com/story/amazons-turker-crowd-has-had-enough/

[49] Nicolas Kaufmann, Thimo Schulze, and Daniel Veit. 2011. More than fun and money. Worker Motivation in Crowdsourcing-A Study on Mechanical Turk.. In *AMCIS*, Vol. 11. Detroit, Michigan, USA, 1–11.

[50] PM Krafft, Meg Young, Michael Katell, Karen Huang, and Ghislain Bugingo. 2019. Defining AI in Policy versus Practice. *arXiv preprint arXiv:1912.11095* (2019).

[51] Siou Chew Kuek, Cecilia Paradi-Guilford, Toks Fayomi, Saori Imaizumi, Panos Ipeirotis, Patricia Pina, and Manpreet Singh. 2015. The global opportunity in online outsourcing. (2015).

[52] Airi Lampinen and Coye Cheshire. 2016. Hosting via Airbnb: Motivations and financial assurances in monetized network hospitality. In *Proceedings of the 2016 CHI conference on human factors in computing systems.* 1669–1680.

[53] Laura Lascau, Sandy JJ Gould, Anna L Cox, Elizaveta Karmannaya, and Duncan P Brumby. 2019. Monotasking or Multitasking: Designing for Crowdworkers' Preferences. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems.* 1–14.

[54] Claudia López, Rosta Farzan, and Yu-Ru Lin. 2017. Behind the myths of citizen participation: Identifying sustainability factors of hyper-local information systems. *ACM Transactions on Internet Technology (TOIT)* 18, 1 (2017), 1–28.

[55] S Lynch, A Blase, A Wimms, L Erikli, A Benjafield, C Kelly, and L Willes. 2015. Retrospective descriptive study of CPAP adherence associated with use of the

ResMed myAir application. *Available at:) ResMed Science Center, ResMed Ltd, Sydney (Australia)* (2015).

[56] David Martin, Benjamin V Hanrahan, Jacki O'Neill, and Neha Gupta. 2014. Being a turker. In *Proceedings of the 17th ACM conference on Computer supported cooperative work & social computing*. ACM, 224–235.

[57] Brian 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*. ACM, 2271–2282.

[58] IG Metall. 2016. Frankfurt paper on platform-based work—Proposals for platform operators, clients, policy makers, workers, and worker organizations. *IG Metall, Frankfurt* (2016).

[59] U.S. Department of Labor. [n.d.]. Breaks and Meal Periods. https://www.dol.gov/general/topic/workhours/breaks

[60] Sunghyun Park, Philippa Shoemark, and Louis-Philippe Morency. 2014. Toward crowdsourcing micro-level behavior annotations: the challenges of interface, training, and generalization. In *Proceedings of the 19th international conference on Intelligent User Interfaces*. ACM, 37–46.

[61] John W Payne, James R Bettman, and Mary Frances Luce. 1996. When time is money: Decision behavior under opportunity-cost time pressure. *Organizational behavior and human decision processes* 66, 2 (1996), 131–152.

[62] Adam M Persky and Jennifer D Robinson. 2017. Moving from novice to expertise and its implications for instruction. *American journal of pharmaceutical education* 81, 9 (2017), 6065.

[63] Nagrale Pritam. 2017. An Ultimate Guide to Making Money with Amazon mTurk. Retrieved June 26, 2019 from https://moneyconnexion.com/mturk-guide-to-make-money.htm

[64] Jason Radford, Andy Pilny, Ashley Reichelmann, Brian Keegan, Brooke Foucault Welles, Jefferson Hoye, Katherine Ognyanova, Waleed Meleis, and David Lazer. 2016. Volunteer science: An online laboratory for experiments in social psychology. *Social Psychology Quarterly* 79, 4 (2016), 376–396.

[65] Joel Ross, Lilly Irani, M Silberman, Andrew Zaldivar, and Bill Tomlinson. 2010. Who are the crowdworkers?: shifting demographics in mechanical turk. In *CHI'10 extended abstracts on Human factors in computing systems*. ACM, 2863–2872.

[66] Susumu Saito, Chun-Wei Chiang, Saiph Savage, Teppei Nakano, Tetsunori Kobayashi, and Jeffrey Bigham. 2019. TurkScanner: Predicting the Hourly Wage of Microtasks. *arXiv preprint arXiv:1903.07032* (2019).

[67] Niloufar Salehi, Lilly C Irani, Michael S Bernstein, Ali Alkhatib, Eva Ogbe, Kristy Milland, et al. 2015. We are dynamo: Overcoming stalling and friction in collective action for crowd workers. In *Proceedings of the 33rd annual ACM conference on human factors in computing systems*. ACM, 1621–1630.

[68] Shruti Sannon and Dan Cosley. 2019. Privacy, Power, and Invisible Labor on Amazon Mechanical Turk. (2019).

[69] Saiph Savage, Andres Monroy-Hernandez, and Tobias Höllerer. 2016. Botivist: Calling volunteers to action using online bots. In *Proceedings of the 19th ACM Conference on Computer-Supported Cooperative Work & Social Computing*. 813–822.

[70] Rustam Shadiev, Wu-Yuin Hwang, Shih-Ching Yeh, Stephen JH Yang, Jing-Liang Wang, Lin Han, and Guo-Liang Hsu. 2014. Effects of unidirectional vs. reciprocal teaching strategies on web-based computer programming learning. *Journal of educational computing research* 50, 1 (2014), 67–95.

[71] M Six Silberman and IG Metall. 2009. Fifteen criteria for a fairer gig economy. *Democratization* 61, 4 (2009), 589–622.

[72] M Six Silberman, Bill Tomlinson, Rochelle LaPlante, Joel Ross, Lilly Irani, and Andrew Zaldivar. 2018. Responsible research with crowds: pay crowdworkers at least minimum wage. *Commun. ACM* 61, 3 (2018), 39–41.

[73] Joseph E Stiglitz. 2000. The contributions of the economics of information to twentieth century economics. *The quarterly journal of economics* 115, 4 (2000), 1441–1478.

[74] Marilyn Strathern. 2000. The tyranny of transparency. *British educational research journal* 26, 3 (2000), 309–321.

[75] Ryo Suzuki, Niloufar Salehi, Michelle S Lam, Juan C Marroquin, and Michael S Bernstein. 2016. Atelier: Repurposing expert crowdsourcing tasks as micro-internships. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. ACM, 2645–2656.

[76] Wade M Vagias. 2006. Likert-type scale response anchors. clemson international institute for tourism. *& Research Development, Department of Parks, Recreation and Tourism Management, Clemson University* (2006).

[77] Omkarprasad S Vaidya and Sushil Kumar. 2006. Analytic hierarchy process: An overview of applications. *European Journal of operational research* 169, 1 (2006), 1–29.

[78] Donna Vakharia and Matthew Lease. 2015. Beyond Mechanical Turk: An analysis of paid crowd work platforms. *Proceedings of the iConference* (2015), 1–17.

[79] Mark E Whiting, Grant Hugh, and Michael S Bernstein. 2019. Fair Work: Crowd Work Minimum Wage with One Line of Code. (2019).

[80] Alex C Williams, Gloria Mark, Kristy Milland, Edward Lank, and Edith Law. 2019. The Perpetual Work Life of Crowdworkers: How Tooling Practices Increase Fragmentation in Crowdwork. *Proceedings of the ACM on Human-Computer Interaction* 3, CSCW (2019), 1–28.

[81] Man-Ching Yuen, Irwin King, and Kwong-Sak Leung. 2012. Task recommendation in crowdsourcing systems. In *Proceedings of the first international workshop on crowdsourcing and data mining*. ACM, 22–26.