RE-EXAMINING THE ROLE OF TIME IN HUMAN-ALGORITHM INTERACTION

BY

JOON SUNG PARK

THESIS

Submitted in partial fulfillment of the requirements for the degree of Master of Science in Computer Science in the Graduate College of the University of Illinois at Urbana-Champaign, 2020

Urbana, Illinois

Adviser:

Professor Karrie Karahalios
ABSTRACT

How fast we interact with technology has been one of the most salient topics for both the users and designers of computing systems. Experimental results since the days preceding personal computing era characterized short response time as an agent for user satisfaction, productivity, and engagement, and we have invested a significant amount of engineering effort to make our computers faster ever since. However, the role of speed in human-computer interaction is much richer and more multi-faceted than what today’s culture of moving fast makes it out to be; where faster interaction offers us efficiency, slower interaction offers us a chance for reflection, serendipity, and a moment for deliberate thinking. As we transition into the days of advanced algorithms and AI where even the most consequential and often problematic judgments are made on our behalf, I see an opportunity to revisit our belief in speed, and examine what slowing down can do to empower users in the interaction between humans and algorithms.

To this end, this thesis explores concrete, measurable benefits of slower interaction in improving users’ assessment of an algorithm’s accuracy in human-algorithm interaction. Specifically, I report a series of online and in-person between-subject user studies in which I isolate the impact of an algorithm’s speed on how users incorporate the algorithm’s advice when making judgments in the context of simple visual recognition tasks. I find that the participants followed good quality algorithms more and bad quality algorithms somewhat less if the response time of the algorithm is slower. Furthermore, qualitative analysis of the in-person study interviews reveals that the waiting was not time wasted, but was often used to reflect on, and think deliberately about the task and the estimation process of themselves and the algorithm, and to compare and reevaluate the two processes. Based on these findings, I outline design implications for future algorithmic systems.
ACKNOWLEDGMENTS

This thesis is dedicated to the wonderful individuals who made my academic journey possible and meaningful during the past two years.

- First and foremost, my adviser Professor Karrie Karahalios. A caring and thoughtful adviser can make all the difference in a graduate student’s life, and I could not have found a better person to call an adviser than Karrie. Beyond teaching me how to do research and write papers, she raised me to be a member of the broader academic community. Thank you one more time for being there – you are more than an adviser.

- My collaborators, Professor Alex Kirlik and Rick Barber, who helped design the experiments for this thesis and ran around the town with me to recruit participants.

- My friends and mentors, Danaë Metaxa and Motahhare Eslami, who guided me in the trenches. I would not be where I am without you.

- More mentors, Professors James Landay and Jeff Hancock, who took a chance and supported me as I got started in the business of doing research.

- Even more mentors, Professors Aditya Parameswaran, Christian Sandvig, and Niloufar Salehi, who provided continuous inspiration for my work.

- Friends and co-authors whom I was fortunate to work with: Mufan Luo, Tiffany Hsu, Sijia Xiao, Kristen Vaccaro, Min Kyung Lee, Ronald Robertson, Professor Christo Wilson, Liza Sivriver, and the members of the Social Spaces Group, Social Media Lab, ixd-research Group, Human-Computer Interaction Groups at the University of Illinois at Urbana-Champaign (UIUC) and Stanford University, and the DataSpread and Blackbox team.

- And a special thanks to Mary Wootters, who sat with me at Bytes Cafe, heard my story, and introduced me to the community of HCI researchers.

And finally, thank you to my parents for their unwavering support. And thank you Moeko for always being there (and reading my draft more than once). The love and support from all of you is what makes things possible.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>CHAPTER 1</th>
<th>INTRODUCTION</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Why Is Time So Important?</td>
<td>2</td>
</tr>
<tr>
<td>1.2</td>
<td>Scope</td>
<td>4</td>
</tr>
<tr>
<td>1.3</td>
<td>Contributions</td>
<td>5</td>
</tr>
<tr>
<td>1.4</td>
<td>Arc of This Thesis</td>
<td>6</td>
</tr>
<tr>
<td>CHAPTER 2</td>
<td>RELATED WORK</td>
<td>7</td>
</tr>
<tr>
<td>2.1</td>
<td>The Impact of System Response Time on User Interaction</td>
<td>7</td>
</tr>
<tr>
<td>2.2</td>
<td>The Slow Movement in Technology</td>
<td>8</td>
</tr>
<tr>
<td>2.3</td>
<td>Challenges in Human-Algorithm Interaction</td>
<td>10</td>
</tr>
<tr>
<td>CHAPTER 3</td>
<td>METHOD</td>
<td>12</td>
</tr>
<tr>
<td>3.1</td>
<td>Study 1 (Online)</td>
<td>13</td>
</tr>
<tr>
<td>3.2</td>
<td>Study 2 (Online)</td>
<td>15</td>
</tr>
<tr>
<td>3.3</td>
<td>Study 3 (In-Person)</td>
<td>15</td>
</tr>
<tr>
<td>3.4</td>
<td>Measures</td>
<td>16</td>
</tr>
<tr>
<td>3.5</td>
<td>Analysis</td>
<td>18</td>
</tr>
<tr>
<td>CHAPTER 4</td>
<td>RESULTS</td>
<td>19</td>
</tr>
<tr>
<td>4.1</td>
<td>Users are Better at Assessing the Accuracy of the Slower Algorithm</td>
<td>19</td>
</tr>
<tr>
<td>4.2</td>
<td>There is an Optimal Response Time for an Algorithm</td>
<td>22</td>
</tr>
<tr>
<td>4.3</td>
<td>Waiting Time is a Chance for Reflection</td>
<td>22</td>
</tr>
<tr>
<td>4.4</td>
<td>There were Benefits to Waiting and Reflection</td>
<td>23</td>
</tr>
<tr>
<td>4.5</td>
<td>Slowness May Lead to Frustration for Some Users</td>
<td>25</td>
</tr>
<tr>
<td>CHAPTER 5</td>
<td>DISCUSSION</td>
<td>26</td>
</tr>
<tr>
<td>5.1</td>
<td>Design Implications</td>
<td>26</td>
</tr>
<tr>
<td>5.2</td>
<td>Limitations and Future Work</td>
<td>28</td>
</tr>
<tr>
<td>CHAPTER 6</td>
<td>CONCLUSION</td>
<td>31</td>
</tr>
<tr>
<td>REFERENCES</td>
<td></td>
<td>32</td>
</tr>
</tbody>
</table>
CHAPTER 1: INTRODUCTION

Imagine that you are searching on a search engine. How long would it take for it to return you the results? If you are on Google at the time of this writing, it would most likely take less than a second, and this would be easy to confirm because the search engine’s response time – the amount of time it took to get you the search results – appears above all its results on the returned page. Even in its early days, Google famously invested heavily to keep the response time of its service under one second [1]. Today, it advocates that all websites appearing on its search results page do the same and keep the response time as low as they can in order to satisfy the information consumers [2].

Of course it’s not just Google that cares about speed; virtually all major internet platforms from social media sites to online marketplaces invest heavily to return their contents as fast as they can, sometimes even at the cost of content accuracy [3, 4]. Similarly, when new hardware products are launched, whether it’s a smartphone or a laptop, what is advertised is how fast the products and the embedded chips are. The culture of moving fast is as much a part of our technological landscape as the power of innovation that these technologies bring.

This almost relentless focus on speed of our interaction with technology is well-founded in human-computer interaction literature. Since the early days of personal computing, the speed of interaction was a theme that emerged as an important component for allowing computing platform’s seamless integration into user’ workflow. Studies have shown that shorter response time correlates with higher user satisfaction, productivity, and engagement, and this in turn motivated an immense effort for reducing any forms of system latency in our technology [5, 6]. The impact of this can be seen in a wide range of applications that we use today, anywhere from simple text editors and spreadsheets where our inputs get recorded in a blink of an eye, to more complex algorithmic systems like the search engines where we search through millions of documents in less than a second.

However, there is an important shift that is happening in our day to day technology from personal computing and internet era that may prompt a reexamination of our belief that faster is better in technology [7]. Today, algorithmic systems are becoming ubiquitous; powered by the recent advances in machine learning and AI, algorithms play an increasingly important role in advising and influencing our actions anywhere from how we search for information [8] to how we make decisions like whom to send to jail [9] or whom to date [10]. And unlike the more traditional applications, these algorithmic, or AI-infused applications, have the potential to produce complex and broad set of possible outputs that are ever changing as more data gets added to train their models. For the designers of human-
1.1 WHY IS TIME SO IMPORTANT?

The questions posed in this thesis are about the time it takes for us to interact with algorithmic systems. But needless to say, time is but one of several long-held tenets of designing seamless interaction with technology. So why is the element of time so important in our interaction with technology, and why is it the center of this thesis?

One reason is that time is arguably one of the most ubiquitous element to designing technology. Regardless of whether it’s computer software presented with graphical user interface or command line interface, or hardware like a smartphone or laptop, the speed at which the technology can respond to its users is fundamental to shaping the user experience. It is therefore no surprise that we teach every computer science students the big O notation for calculating how fast an algorithm can run, and that a great portion of the engineering academy puts the mission for faster and more efficient technology at its heart (and of course it is important to note that in certain use cases, timely interaction is a key to creating technology that is meaningful; it’s hard to imagine a GPS be useful if it took even minutes to tell us whether to turn left or right at the next intersection when we are driving, or a recommender system that takes hours to suggest which restaurant to dine in for lunch). So understanding when we can, or should pull back on, our assumption that time is a dimension that needs to be reduced at all cost can have an impact that resonates across
At the same time, time is also closely associated with our modes of thinking. Daniel Kahneman’s seminal work presents a theory of human minds that separates our way of thinking into two distinct modes [12]. One is that of fast thinking, or System 1 thinking, that specializes in modeling familiar situations and making short-term, intuitive decisions or predictions. System 1 is powerful because it is relatively accurate and quick in situations that has been seen before. However, this mode of thinking is prone to biases and systematic errors under specified circumstances, and can be rendered helpless in truly novel situations. To solve this, slow thinking, or System 2 thinking, is presented as our control for making up the failings of System 1 thinking. In contrast to fast thinking, slow thinking is a deliberate and much more reflective way of thinking whose job is to catch our worst impulses and biases, and generate long-term plans that can even navigate previously unseen situations (and note that slowness in Kahneman’s work is meant to be taken, in large part, literally. Although what distinguishes slow thinking from fast thinking is the deliberate and reflective attitude that we take, Kahneman stresses that there is a ”natural speed” to slow thinking that is indeed slower than our fast thinking [12]).

There could be a parallel here between the two modes of thinking in our minds – fast and slow – and the two types of decision-makers in our today’s society – algorithms and humans. Much like our fast thinking, algorithms have proven themselves to be powerful tools that can summarize previous instances and quickly output accurate suggestions for our future actions. But they are also bias-ridden, and not a month goes by without a major discovery that suggests that deployed algorithmic systems discriminated against vulnerable or underprivileged communities [13] or inappropriately handled previously unseen scenarios [14]. On the other hand, though often slower and more expensive to employ, human minds are capable of understanding the context from which to interpret novel instances [15] and untangle potentially biased algorithmic outputs [16, 17]. As the days of automation progresses, can we imagine the role of humans as the guardians that keep the algorithms’ biases and failings at bay, much like what our slow thinking does for our fast thinking?

Asking this question may give us a framework to think about the roles of humans and algorithms in much discussed human-in-the-loop or human-AI collaborative systems. This conversation has already started. Human-computer interaction scholars have tinkered with what slowness in technology would mean for our ability to reflect and be conscious in our digital environment [18, 19, 4]. And in the last half a years, calls for a slow movement in technology to fight against biases and discrimination in technology have been advocated by developers of some of the most prominent algorithmic systems [1, 16]. But there’s much left to be done and more concrete results to be had to show the benefits of slowness in the
context of human-algorithm interaction.

1.2 SCOPE

It is also worth noting here the scope of this thesis; designing an algorithmic system is a multi-faceted problem that needs investigation from more angles than what this thesis can cover, and a complete case for a slow algorithm would need to respect challenges from these various angles and consider trade-offs if necessary. For example, how would slowness affect the usability and satisfaction of the algorithmic system? If the longer response time can be allowed as a design feature, then what can the system developers do to take advantage of that extra time to provide even more value to the users? Is it ethical to arbitrarily slow down the system and potentially mislead the users on what the system is capable of? And perhaps most fundamentally, are there actually any concrete benefits of slowing down our interaction with algorithmic system that we can measure?

The first three of these questions have been directly, or indirectly touched upon by prior works. Ever since Lars Hallnäs and Johan Redström first discussed the merits of slow technology in their seminal article [18], scholars in human-computer interaction and design have explored the topic of reevaluating the values of waiting time as moments of reflection, mental rest, and a catalyst for serendipity [20]. For example, multiple workshops at major conference venues explored the pros and cons of slowness in modern times [21, 22], while Jaime Teevan et al. presented the concept of “slow search” to suggest that search engines can take extra time to look at a more comprehensive set of information to return more useful information to the users [4]. Meanwhile, Eytan Adar et al. proposed that “benevolent deception” could be acceptable in design for user interaction if it means benefitting the users and the developers [23].

While more future work is needed in each of these directions, there is a particular lack of prior work that confirms the concrete, measurable benefits to slowing down our interaction with algorithmic systems. To this end, this thesis focuses on the last question, on finding measurable benefits of slower interaction with algorithmic systems. More precisely, I explore whether slowing down the response time of an algorithm can help users better assess the algorithm’s accuracy when performing a simple visual recognition task, thereby allowing them to recognize when to rely on the tool and when to discount it.

Whether a slow algorithm can help improve users’ assessment of the algorithm’s accuracy has not been studied in prior literature. However, this topic is central to successful interaction between humans and algorithms. Failure for users to clearly understand when technology can be relied upon can lead to “poor partnerships between people and automation [that]
will become increasingly costly and catastrophic” [24]. Even in the recent years, we have seen this failure leading to disastrous outcomes in the human-algorithm cooperation in the context of self-driving cars [25] and airplane autopilot [26]. It is, therefore, no surprise that this topic of enabling users to better understand algorithm’s accuracy and reliability has been at the core of the efforts to integrate algorithmic systems into broader social contexts [27, 28]. To date, studies in this regard has focused on making the model learned by the algorithms interpretable and helping users understand why certain inputs map to certain outputs [29, 30, 31]. But going beyond the commonly focused topic of making these models interpretable by exploring how the design elements like the speed of interaction affects the user’s ability to assess the algorithm’s accuracy could provide a valuable perspective to the existing literature.

1.3 CONTRIBUTIONS

The core of the contributions made by this thesis is a series of online (study 1, n=140; study 2, n=200) and in-person (study 3, n=32) between-subject user studies in which I isolate the impact of an algorithm’s speed on how users incorporate the algorithm’s advice while performing a simple visual recognition task with the help of an algorithm. In particular, I asked the participants to estimate the number of jelly beans in a jar with the help of an algorithm’s suggestion. I find evidence that users are better at assessing the accuracy of an algorithm’s advice if the speed of the algorithm is slow. More specifically, I find that:

- People adhere to a slower algorithm more if the output accuracy is good.
- People adhere to a slower algorithm somewhat less if the output accuracy is bad.

In addition to this, my work contributes to furthering our understanding of a user’s mental model during the waiting time by qualitatively analyzing the content of the interviews that took place at the end of the in-person study. I find that for many of our participants, the waiting time was not time wasted; the time was often used to reflect on the problem at hand and the estimation process of themselves and the algorithm, and to introspectively compare and reevaluate the two processes. Some specifically appreciated the waiting time for giving them a chance to rethink their own estimation before being primed by the estimation of the algorithm, helping them avoid blindly following, or blindly dismissing the algorithm’s outputs.

However, it is important to not overstate the generalizability of the results presented here. As algorithmic and AI systems become more advanced, the context of interaction would also
become much more diverse than the experimental setup presented here (in fact, one could argue that the setup here is a relatively simple one), and it would be wrong to imply that users can gain comparable benefits from slower interaction regardless of the task they are tackling using the system. But the message I hope to convey in this thesis is not to suggest that we should slow down all our interaction with algorithmic systems. Rather, it is to pose the speed of interaction as an element of design that can benefit the users, and needs to be experimented with even if it is against our long-held belief that faster is always better.

1.4 ARC OF THIS THESIS

The remainder of this thesis will be structured as follows: I first cover the classical literature that studied how the response time of a system affects its users, as well as the recent interest in adopting the slow movement in technology. In Chapter 3 and 4, I present the experiments and their results that serve as the backbone of this thesis. Finally in Chapter 5, I discuss my work’s design implications for future algorithmic systems, as well as the limitations and potential future work.
CHAPTER 2: RELATED WORK

In this chapter, I review two distinct bodies of literature: one that is about the element of time in human-computer interaction, and other that is about the recent progress in human-algorithm interaction. In section 2.1, I summarize how in the recent years, the response time of a system was characterized as an element to be eliminated for greater user engagement and satisfaction despite the much more nuanced nature of its impact on users. In section 2.2, I cover the slow movement in technology that has gained greater attention as a counterargument for technology’s focus on speed and efficiency. Finally in section 2.3, I describe current challenges in human-algorithm interaction to set the stage for arguing how the findings in this thesis related to slowness can contribute to the greater effort to improve our interactions with algorithms and AI.

2.1 THE IMPACT OF SYSTEM RESPONSE TIME ON USER INTERACTION

The pace of interaction with our technology has been an important, and widely studied area in human-computer interaction since even before the days of personal computing [32]. But in the recent years, with the rise of algorithmic platforms such as search engines and social media news feeds, there has been renewed interests in response time in regards to algorithmic systems. The efforts have been focused mostly on the users’ satisfaction, productivity, and engagement, concluding that fast interaction is almost always preferred. For example, Google conducted online experiments in which the response time of search outputs was intentionally delayed by 100 to 400 milliseconds and saw a significant drop in the number of searches per user [3]. Similarly, Bing experimented by adding an intentional server delay of 50 to 2,000 milliseconds and observed a decrease in not only the number of searches, but also in users’ engagement with the search results [3]. These findings led to the technology platforms heavily optimizing for speed even at the cost of the output quality of the algorithms; search engines search through a previously cached, incomprehensive set of available documents even at the cost of returning less relevant information [4], while social media news feeds (e.g., Facebook’s Newsfeed) prioritize on showing fast loading content [33].

However, earlier studies that explored the impact of response time on the users send a much more nuanced message to the designers of computing systems. In 1968, Robert B. Miller summarized 17 unique situations and tasks such as simple data entry and page navigation that can arise while using computers of his time, and qualitatively presented guidelines for an acceptable response time in each case [32]. He proposed that the context
of the interaction is integral to the process of defining the appropriate response time. For example, if a user is simply pressing down on a key to enter a character in a command-line interface, the character should show up on the screen with almost no delay. However, if the user is engaged in a much more complex process like restructuring multiple columns of tabular data, the user may be willing to wait significantly longer than two seconds.

Following up on Miller’s guidelines, for the next decade and a half, scholars experimentally tested the acceptable range of response times for various tasks, and expanded our understanding of what needs to be considered when deciding on acceptable wait times. The results suggest that faster does not necessarily mean better; while a short response time of under one second is preferred for user satisfaction and productivity in most tasks discussed by Miller [34, 35], it also leads to an increase in error rates for certain tasks if users pick up the pace of a rapid interaction sequences, lowering the overall quality of user’s work [36, 37].

My view presented in this thesis is inspired, in part, by these earlier studies on the role of system response time in human-computer interaction. In our pursuit of users’ satisfaction and engagement, our modern day interpretation of the ideal response time has characterized waiting time as a negative element that must be eliminated, losing much of the nuance that was represented in the classical literature in this area. But some of these earlier results suggesting that users are willing to wait longer if they are involved in a more complex task using a machine, and that they are more prone to making errors in some cases if the interaction were too fast seem particularly poignant today. As we employ more algorithmic systems to make increasingly complex and high-stakes decisions where errors could lead to potentially catastrophic outcomes, our mindset in regards to response time should be reexamined and updated.

2.2 THE SLOW MOVEMENT IN TECHNOLOGY

The movement in technology that is challenging the recent focus on speed and moving fast is the slow movement that began in 1986 with an activist in Italy protesting against the opening of a fast-food chain restaurant, advocating for a slower, traditional and mindful way of eating [38]. Over the years, the slow movement has also had its influence on thinkers in technology, as evident from Hallnäs and Redström’s influential work from 2001 “Slow Technology – Designing for Reflection” that first presented us with a vision of designing technological artifacts that are focused not on efficiency and performance, but rather on reflection and mental rest by creating technological artifacts that are meant to be consumed slowly, over a long period of time [18].

Since then, the agenda of slow technology has been gradually applied to various causes
like supporting better social connections through online messaging with temporal delay [39] and experiences of anticipation through a printer that prints nostalgic images from the user’s photo library each month [19]. But recently, there has been a growing interest in applying the framework of slow technology to how we interact with algorithms. Search engines today, for example, not only retrieve simple facts and related documents, but also return answers to complex questions like where one should take vacation, or have dinner. In cases like these, a “slow search” as proposed by Jaime Teevan et al. can take extra time to look at a more comprehensive set of information available and return more relevant and useful information to the users [4, 1]. And importantly for the users, human-computer interaction scholars have suggested that the waiting time can be used to encourage reflection and letting the mind wander to increase the chance of serendipitous discovery [21], or to slowly think about the decisions being offered by the algorithm and ponder on its potential biases or flaws [16].

There certainly have been many doubts and critiques toward the idea of simply slowing down technology [22]. Long response times can be frustrating for all stakeholders of the system, and simply waiting a long time may not result in new useful insight for the user [40, 41]. However, the recent trend of overly focusing on fast interaction that is in part driven by behavioral advertising that benefits from high user engagement and satisfaction [1, 42] leaves room for revisiting the question of how fast we should interact with algorithms. A successful design of human-computer interaction goes well beyond allowing for users’ productivity, efficiency, and engagement, and have to take into consideration different contexts in which algorithms are deployed. For algorithmic systems like GPS navigators that are used often and needs to deliver instructions to drivers in real time, it might be preferred, or even necessary, for the speed of the interaction to be fast. But when designing for algorithmic systems that are not time-sensitive, and that are responsible for giving us consequential advice like who will go to jail or who needs serious medical attention, we should consider slowing down our interaction with these systems if it means we are better at noticing its potential.

And although prior work has not directly investigated whether slowness could yield concrete benefits to the users such as helping them better assess the errors of the algorithm, literature on human minds and psychology provides interesting hints that poses slowness as a possible candidate for improving human-algorithm interaction. For example, despite his work being not often associated with the slow movement in technology and human-computer interaction literature, Daniel Kahneman’s extensive work on slow and fast thinking provides an important insight in this regard [12]. In it, Kahneman summarizes a human mind into two components: fast thinking (System 1) relies on repeated past experience to focus on instinctive and short-term decision-making for familiar situations much like an algorithm,
while slow thinking (System 2) relies on reflections and deliberate thinking to focuses on more novel and longer-term decision-makings. In this narrative of human mind, the role of slow thinking is to remedy the failings of fast thinking that is more susceptible to biases and mistakes in novel situations through reflection and deliberate thinking. Can users of the algorithmic system be encouraged to provide the slow thinking capacity in settings where humans and algorithms cooperate?

Ultimately, the contribution of slow technology is not advocating for unreasonable and meaningless delays in our digital lives, but offering us a framework to think about the optimal amount of time needed to process the information and decisions presented to us much the same way the early pioneers of human-computer interaction approached system response time. It is also in this context that I place the contributions of my thesis.

2.3 CHALLENGES IN HUMAN-ALGORITHM INTERACTION

In the prior sections of this chapter, I summarized how the existing works have studied system response time and argued for slowness in the broader context of technology. In this section, I describe the current literature on designing human-algorithm interaction to set the stage for the focus in this thesis, which is to help users better assess the accuracy of an algorithmic system.

To start this section, I highlight that there are unique challenges to designing the interaction with an algorithmic or AI system when compared to other types of technology what human-computer interaction literature is accustomed to working with. Qian Yang et al. summarize these challenges as follows: 1) uncertainty around what the system is capable of achieving and the fact that its capacity might continuously change if it gets more training data, and 2) complexity of its possible output space that ranges from complex numeric values to complete sentences, for example by a voice assistant [11]. These new challenges mean that traditional methods of designing interaction often fail, and the task of assessing the power of the algorithmic system, which is critical to successful interaction between humans and machines, often falls on the users during their interaction.

However, users are often not well equipped to assess the power of an algorithmic system, and are prone to blindly trusting its outputs [43]. And to make the matter more urgent, many of the algorithmic systems that are currently deployed are often flawed in some ways that are not clearly communicated to their users. For example, the number of cases where algorithmic systems, handicapped by training datasets that are often flawed and biased, returning inappropriate responses like marking a defendant more likely to commit crimes in the future based on his race [9] or judging that patients with asthma has a better chance
of surviving pneumonia [44] are growing. Rather than blindly following, or rejecting the suggestion made by an algorithm, it is important to encourage the users to become the judge of when to follow the algorithm.

Because of this, significant efforts have been spent on improving users’ ability to assess the accuracy of the algorithm [27, 28]. Numerous studies, for example, have explored the relationship between an algorithm’s interpretability, transparency, and users’ assessment of the algorithm’s performance [29, 30, 31, 45]. Meanwhile, Poursabzi-Sangdeh et al. found that transparency towards an algorithm’s attributes affect the users’ abilities to detect the algorithm’s mistakes [46], while the recent human-computer interaction approach to this challenge has focused on studying how to communicate the accuracy of an algorithm to the users. In a controlled experiment where the participants were given an estimate of the algorithm’s accuracy, Yin et al. found that people’s trust in an algorithm correlates with the stated accuracy of the algorithm [45].

Our work contributes to this growing body of literature that explores users’ interaction with algorithms by revisiting the effect of the waiting time on the users. In particular, we study whether the slower interaction with an algorithmic system could benefit the users’ ability to assess the accuracy of the algorithm.

**Research Question (RQ).** *Would a slow algorithm improve users’ assessments of the algorithm’s accuracy?*
CHAPTER 3: METHOD

I conducted two online (study 1, n=140; study 2, n=200) and one in-person (study 3, n=32) between-subject user studies in which participants were assigned with a simple visual challenge of estimating the number of jelly beans in a jar with the help of an algorithm. In all three studies, participants were presented with five images of jelly bean jars one at a time, and were asked to make an initial estimation of how many jelly beans were in each jar. After each time they recorded their initial estimation, the participants were given advice from an algorithm of varying response times and accuracy about what the correct number of jelly beans in the jar might be. The participants then had a chance to change their answer and record their final estimation.

It is worth noting that quantitatively measuring how closely a participant follows an algorithm’s estimation is challenging. To this end, the task of estimating the number of jelly beans was carefully chosen based on the prior literature. Although the task could be considered simple and somewhat artificial, early studies in psychology that studied people’s tendency to conform to the other’s opinion [47], or that studied the wisdom of crowds [48] have frequently employed this particular task because it allows for an easy comprehension on the part of the subjects, quantitative standards of measurement, central point from which to measure dispersion, and sufficient range for expression of opinion so that no one might hold a more extreme opinion or judgment than is provided for in the scale of measurement [47]. And importantly for this study, for many of the participants, the task was not so mechanical in the sense that it was not immediately clear how difficult it would be for a computer algorithm to make an accurate estimation.

Given this, despite its simplicity, the jelly bean task gives us a convincing experimental conditions to study how people’s adherence to an algorithm’s suggestion changes by observing how much the participants would change their initial estimation on the number of jelly beans to adhere to the algorithm’s advice given the varying response times and algorithm’s accuracy. If a slower response time improves users’ assessments of the algorithm’s accuracy, we should observe that the participants’ confidence in the slow algorithm’s output will strongly correlate with its accuracy.

**Hypothesis 1** (H1). Given an accurate advice from the algorithm, participants will adhere to the algorithm’s advice more and exhibit higher confidence in the algorithm’s accuracy if the response time of the algorithm is slower.

**Hypothesis 2** (H2). Given an inaccurate advice from the algorithm, participants will adhere to the algorithm’s advice less and exhibit less confidence in the algorithm’s accuracy.
3.1 STUDY 1 (ONLINE)

3.1.1 Participants.

A total of 140 participants on Amazon Mechanical Turk (MTurk) completed Study 1 that took 14.3 minutes on average to finish. Participants consented to participate once at the beginning of the study, and once at the end of the study when they were debriefed. They had to be at least 18 years old, living in the US, and have completed at least 100 Human Intelligence task (HITs – MTurk’s task unit) with at least a 95% HIT approval rate. The
mean age score was 4.60 (SD=0.96; 3=“18-24 years old,” 4=“25-34 years old”), and 45 of them identified themselves as female. In addition, 59 of the participants held a bachelor’s degree, 19 held a higher degree, and the rest a high school diploma or some high school-level education. The sample was 76.43% Caucasian, 8.05% Hispanic, 6.45% Asian, 5.00% African American, and 2.01% Native American, and 0.07% other. After pilot studies, I expected the participants to take roughly 12 minutes or less. To this end, the participants were initially paid $1.50 for their time through the standard payment system of MTurk. My post-study analysis revealed, however, that the participants in Study 1 took longer than my expectation. Therefore, following the recent practice of using MTurk’s bonus system that allows requesters to pay the workers extra money after the initial payment, I paid every participant in this study extra $0.30 (for an example, see [49]). This ensured that participants were paid at least the US Federal minimum wage of $7.25 per hour.

3.1.2 Procedure.

When the participants accepted the task on MTurk platform, they were redirected to a custom built website for this study and randomly placed into one of 14 categories: a combination of seven different response times (1, 5, 15, 30, 45, 60, and 75 seconds) and two algorithm accuracy (high accuracy in which the algorithm’s advice were off by only 2% from the correct answer, and low accuracy in which the algorithm overestimated the correct answer by 100%). There were 10 participants in each category. Similar to a procedure used in a study that explored human’s perception of algorithms [50], I started the study by providing the participants the following definition of algorithms: “Algorithms are processes or sets of rules that a computer follows in calculations or other problem-solving operations” [50]. The participants were then given a brief explanation that machine vision algorithms are actively researched type of algorithms which focus on understanding the contents of videos or images, and that a group of university researchers have developed a version of a machine vision algorithm named ObjectRecognizer that can count the number of jelly beans in a container from a photo of the container.

In the study, the participants were shown five images one at a time, each of unique and transparent jars with 520, 450, 660, 730, and 590 jelly beans in this order of appearance. After each image was shown, the participants were asked to record how many jelly beans they thought were in the container. They were then presented with a button on the website to start running the algorithm to get its estimation with the following explanation: “Now, you will run our machine vision algorithm, ObjectRecognizer, in real-time. Once you receive its suggested answer, you are welcome to change your final answer as much, or as little
as you want.” When the participant pressed the button, all participants were shown a commonly used loading GIF until the algorithm returned its estimation. The amount of time participants had to wait before the algorithm returned its estimation, and the accuracy of its estimation, were based on categories the participants were placed in. No further instructions were given during the waiting time. Finally, the study ended with a short survey that included a manipulation check and a short survey about the participants’ confidence level in the algorithm’s output accuracy in a 7 point Likert scale, and demographics.

3.2 STUDY 2 (ONLINE)

3.2.1 Participants.

A total of 200 participants on Amazon Mechanical Turk (MTurk) completed Study 2 that took 13.9 minutes on average to complete. Participants in Study 2 were recruited and paid through the same procedure as Study 1; the participants were initially paid $1.50 through the standard payment system of MTurk, and later received a bonus of $0.30 to ensure that they were compensated at least the US Federal minimum wage for their time. The mean age score was 4.60 (SD=1.05; 3=“18-24 years old,” 4=“25-34 years old”), and 74 of them identified themselves as female. Also, 94 of the participants held a bachelor’s degree, 21 held a higher degree, and the rest a high school diploma or some high school-level education. The sample was 75.0% Caucasian, 7.50% Hispanic, 5.0% Asian, 8.50% African American, and 2.0% Native American, and 1.50% other.

3.2.2 Procedure.

Study 2’s procedure was identical to that of Study 1. However, informed by the results from Study 1, I narrowed down our participant categories to four: a combination of two different response times (1 second and 45 seconds) and two algorithm accuracy, which were the same as the ones stated in Study 1. Each category had 50 participants.

3.3 STUDY 3 (IN-PERSON)

3.3.1 Participants.

I recruited a total of 32 participants around a university town in the Midwest region of the United States through flyers and an online newsletter for an in-person study that took
around 45 minutes to complete. Participants consented to participate once at the beginning of the study, and once at the end of the study when they were debriefed. The participants were asked to come in to the lab and were paid $10 for their time. The mean age score was 4.13 (SD=1.58; 3=“18-24 years old,” 4=“25-34 years old”), and 22 of them identified themselves as female. And 90 of the participants held a bachelor’s degree, 6 held a higher degree, and the rest a high school diploma or some high school-level education. The sample was 59.3% Caucasian, 12.5% Hispanic, 21.9% Asian, and 6.25% African American.

3.3.2 Procedure.

Study 3 is an in-person replication of Study 2. I invited participants who responded to our flyers and online newsletter to our lab. The participants were then randomly placed in one of the four categories used in Study 2, with each category having 8 participants. The participants were then directed to the same custom website used in Study 2 on a laptop that was provided by the researcher who conducted all in-person sessions. Rest of the procedure follows that of Study 2. At the end of the study, however, the researcher conducted a 10 to 15 minutes exit interview with the participants to explore the participants’ mental model while waiting for the algorithm to return its estimation.

3.4 MEASURES

Below are the measures I used to test my hypotheses in all three studies. As mentioned above, Study 3 included an exit interview in addition to these measures.

3.4.1 Adherence to the algorithm’s advice.

To measure how closely the participants followed the advice from the algorithm, I calculated how much the participants changed their initial estimation of the number of jelly beans towards the algorithm’s suggestion. In my analysis, I only focus on the first iteration of estimating the number of jelly beans out of the five due to the learning effect that occurs as the iterations continue.

Additionally, I highlight the following observations in my results to help justify this measure:

• *Consistent distribution:* For all three studies, participants were randomly assigned into one of the categories in the study. My results show that the distribution of the average
initial estimation during the first iteration were not significantly different between different categories.

- **Similar starting point:** Participants in all three studies started from relatively similar initial estimations with most of them underestimating the number of jelly beans, on average by around 200. In order to further restrict the variance of the initial condition that the participants started from, I also analyzed my results with only the participants whose first initial estimation was within one standard deviation away from the mean, and found our findings to replicate.

- **A benefit from the good algorithm:** That a lot of my participants underestimated the number of jelly beans by around 200 on average meant the participants who received advice from a good algorithm (off by only 2% from the correct answer) almost always benefited from adhering closely to the algorithm and outputted a more accurate final estimation.

- **A harm from the bad algorithm:** This also meant that for the participants who received advice from a bad algorithm (overestimated the correct answer by 100%), they were almost always better off not listening to the algorithm’s suggestion as this would have led them to make a more inaccurate estimation.

### 3.4.2 Confidence in the algorithm’s accuracy.

Complementing the above measure, I also measured the participants’ level of confidence during the exit survey in 7 point Likert scale with the following question adopted from a previous research [51]: “How confident were you in the ObjectRecognizer algorithm’s estimate?” Although self-reported measures are not as strong as behavioral measures, in my results, I find this measure to correlate with the behavioral measure described above about how closely the participants adhered to the algorithm’s advice.

### 3.4.3 Definition of algorithm manipulation check.

In previous research, the participants were asked the following open-ended question [50]: “In your own words, please briefly explain what you think algorithms are.” This question functioned as a manipulation check to ensure that all participants had a reasonable understanding of what an algorithm is. This work also makes use of this question; the answers to this question confirmed that the participants understood algorithms as autonomous decision-makers.
3.5 ANALYSIS

I used the one-sided Mann-Whitney U test in order to test my hypotheses that 1) the participants will adhere more to the advice of a slow algorithm given highly accurate output, and that 2) the participants will adhere less to the advice of a slow algorithm given inaccurate output. To analyze the main themes discussed by the participants during the exit interview in Study 3, a collaborator and I labeled the interview transcription using line-by-line open coding. We revised our labeling through a collaborative and iterative process, and then used axial coding to extract the relationship between themes.
CHAPTER 4: RESULTS

I summarize my findings from the three studies here. In subsection 4.1 and 4.2, I focus on the quantitative measures described above to explore how the response time of the ObjectRecognizer algorithm affected the participants' process of estimation, and their confidence in the algorithm's output accuracy. In subsection 4.3, 4.4, and 4.5, I take a qualitative approach and thematically analyze the contents of the exit interview in Study 3 to help elucidate what is driving the results in the earlier subsections.

4.1 USERS ARE BETTER AT ASSESSING THE ACCURACY OF THE SLOWER ALGORITHM

4.1.1 For an accurate algorithm, users trust the slower algorithm more (H1).

My results confirm H1. In all three studies, when given a good (2% error rate) advice from the algorithm, participants who received advice from a somewhat slow algorithm with a response time of 45 seconds changed their initial estimation to a number much closer to the algorithm's estimation than the participants who received advice from a fast algorithm with a response time of 1 second (Study 1, Z=84.0, p=0.0056; Study 2, Z=1548.0, p=0.02; Study 3, Z=52.0, p=0.02). Additionally, I also notice that given an accurate algorithm, participants in the slow algorithm group were a little more confident in the output of the algorithm than participants in fast algorithm group for all three studies.

4.1.2 For an inaccurate algorithm, users trust the slower algorithm somewhat less (H2).

However, if the accuracy of the algorithm's output is low (100% overestimation), I see the opposite trend. Across all three studies, I find some evidence that weakly supports H2; when given an inaccurate advice from the algorithm, participants who received advice from a slow algorithm with a response time of 45 seconds changed their initial estimation less than the participants who received advice from a fast algorithm with a response time of 1 second (Study 1, Z=43.0, p=0.31; Study 2, Z=1025.0, p=0.06; Study 3, Z=21.0, p=0.13). Similarly, given an inaccurate algorithm, participants in the slow algorithm group were a little less confident in the output of the algorithm than participants in the fast algorithm group.

Given the number of participants I was able to recruit, I do not claim statistical significance for our findings in the bad algorithm’s case. However, the results here indicate that a slow
Figure 4.1: Figures summarizing the participants’ degree of adherence to the advice given by the algorithm. When given an accurate algorithm, the participants changed their initial response more towards the algorithm’s suggestion if the algorithm were slower. But when given an inaccurate algorithm, the participants changed their initial response more towards the algorithm’s suggestion if the algorithm were faster.
Figure 4.2: Figures summarizing the participants’ degree of confidence in the algorithm. When given an accurate algorithm, the participants were more confident in the algorithm if it were slower. But when given an inaccurate algorithm, the participants were more confident in the algorithm if it were faster.
algorithm did cause the participants to not blindly trust its suggestion, but rather encouraged them to better recognize an accurate algorithm. Thus I answer my overarching research question: in the context of my study, a slow algorithm improves users’ assessments of the algorithm’s accuracy.

4.2 THERE IS AN OPTIMAL RESPONSE TIME FOR AN ALGORITHM

Despite my findings presented above, it was not the case that the algorithm could be indefinitely slower to be beneficial to the participants. Instead, the results from Study 1 show that participants were most trusting of the good algorithm when its response time was approximately 45 seconds, and least trusting of the bad algorithm when its response time was approximately 30 seconds. Previous literature has shown that in the context of simple tasks involving computers such as data entry, there is an optimal system response time for decreasing the user’s error rates, and that the response time should be neither too long nor too short [36, 37, 5]. My result seems to indicate that this trend extends into more algorithmic tasks like the one presented in the three studies above; there is an optimal response time for the algorithm that provides cognitive benefits to the participants who are making a decision with this algorithm.

4.3 WAITING TIME IS A CHANCE FOR REFLECTION

Having observed that participants were better at assessing the accuracy of a slow algorithm in my results presented above, I moved on to explore the mental model of my participants as they were waiting for the algorithm to return its answer by thematically analyzing the exit interview in Study 3. After the participants completed their estimation tasks and the demographics survey, I asked the participants to openly describe what they were thinking about as they were waiting for the algorithm to return its estimation of the number of jelly beans. A longer waiting time corresponded with a higher likelihood of the participants reflecting, and deliberately thinking about the estimation task at hand. While almost all participants in the slow algorithm group (87.5%) had noted that they actively thought about the task at some point while they were waiting, only some of the participants in the fast algorithm group (37.5%) had noted the same. This is an expected result considering that the participants in the fast algorithm group barely had any time to think at all as illustrated in the following quote: “It wasn’t very long to wait. So I didn’t have time to think about it a whole lot” (P22).
However, for the 20 participants who reported to have used the waiting time to reflect on the process of estimation, I see a clear thematic pattern arise. Of these participants, 10 participants reported to have reflected on their own process of estimating the number of jelly beans to improve their answers. For example, one participant mentioned: “as I was waiting, I was like still trying to look at the jar to see like maybe if I change my response based on like taking more time to inspect the jar and try to like guess how tall it was” (P14). Another 10 participants reported to have reflected on the process of the algorithm, speculating what and how it would estimate: “I think it would distinguish the jelly beans by, like, the pixel colors, but I’m not sure if it actually does that. I don’t know. I was trying to think of ways that the algorithms did it” (P9). Seven participants tried to compare their own process of estimation with that of the algorithm: “… the first time, I was more thinking about the algorithm and how it was [estimating]... but overtime I thought differently. I’m like, okay, now I still want to know what numbers coming up, but how does that fit with the numbers that I’m estimating? Are there patterns in that” (P9)?

4.4 THERE WERE BENEFITS TO WAITING AND REFLECTION

We have seen now that slowness encourages reflection, and more active thinking on the user’s part. But an important question that this thesis is trying to answer is whether there are benefits to this reflection, and if there are, what causes that those benefits. In our interview, we observe that the reflection induced new insights about the task from the users, and prevented them from making snap judgements about whether to rely on the algorithm.

4.4.1 Reflection led to new insights about the task, but not about the algorithm.

Some participants have noted that they gained more insight about how to estimate the number of jelly beans during the waiting time. One particularly convincing insight that four of the participants who reported to have used the waiting time for reflection mentioned was them noticing that there was a door knob in the background of the photos of the jelly bean that could be used to better speculate how big the container would be in real life: “I kind of consciously realized that the picture, I could get a sense of how tall something was because it showed it in relation to the door with the handle” (P19). However, when asked about whether the waiting time helped them better understand the algorithm’s estimation process, the answer was, perhaps unsurprisingly, negative with 4 out of 10 participants who reported to have reflected on the process of the algorithm specifically mentioning that they do not understand how the algorithm works, and the others left guesses they were not confident in:
“[the waiting] made me feel like I understood the algorithm a little bit more, but it still is kind of like a black box. I wouldn’t know” (P9).

4.4.2 The waiting time gave participants time to reflect before seeing the algorithm’s answer.

According to some of the participants, however, a convincing benefit to a longer waiting time came from the fact that the participants in the slow group had a chance to think over their estimations before seeing and being influenced by the algorithm’s estimation. Here are quotes from two different participants, both of whom received an inaccurate advice that overestimated the correct answer by 100%. Whereas the participant of the first quote received advice from a fast algorithm, the participant of the second quote received advice from a slow algorithm.

I think once I saw the algorithm’s answer, I was more inclined to be like, that’s probably right. Whereas if maybe I had more time to think about my own answer, I would have felt more comfortable with mine and less inclined to just blindly adjust my answer compared to the algorithm’s answer... Because once I, once I made my guess and then I instantly see the algorithm’s then it’s like, oh, okay. (P20)

Expressing a similar sentiment, a participant who was given an advice from a slow and inaccurate algorithm mentioned:

While I was waiting for the algorithm’s prediction, I kind of just was like thinking over my answer and... I decided like, okay, mine is more accurate before seeing the prediction and then after seeing the prediction, I think that time allowed me to I guess like reaffirm my prediction. (P27)

For P20, seeing the algorithm’s estimation right away made the participant much more likely to make a snap judgement to blindly trust the algorithm when it presented an estimation that was likely too high. On the other hand, for P27, the waiting time gave the participant an opportunity to reassess the accuracy of the participant’s own estimation, and helped the participant be less influenced by the algorithm’s bad estimation. P20 likened this effect to having an answer sheet right next to you when you are solving a problem set to study for a test; knowing that the answers to the problems are right there, you would exert much less effort to solve the problems and rush into check whether your preliminary answer
matches with what is in the answer sheet. This would be harmful to the participants’ ability to come up with a better answer on their own, causing them to be less able to judge the accuracy of an algorithm.

4.5 SLOWNESS MAY LEAD TO FRUSTRATION FOR SOME USERS

Within the context of my study, 29 of the participants in both the fast and slow algorithm groups found the respective response time to be acceptable. But it would be misleading to not point out that four of the 16 participants in the slow algorithm group made remarks on the slowness of the algorithm and had hoped for the algorithm to be a little faster: “I think it was like kind of long, but it wasn’t like too long. Um, but for me, I feel like I’m relatively impatient and so I just like, wanted to know [the algorithm’s estimation] right away” (P12). This response, in part, seems to depend on the participants’ existing expectations that stem from previous experience with technology and algorithms in general. A couple of participants in the slow algorithm group suspected that the algorithm might have crashed or was programmed inefficiently. Similarly, a participant in the fast algorithm group was surprised by how fast the algorithm returned the estimation based on the participant’s previous experience running programs: “I’ve been doing a lot of like Matlab homework and I kind of equated that to that... I think analyzing images is harder for computers than it is for us. Especially with all the different colors of the jelly beans and that sort of thing. So that was kind of impressive how well it did and how quickly it went” (P7).
CHAPTER 5: DISCUSSION

The results from the experiments presented in this thesis provide an evidence that slowness could indeed provide concrete benefits to the users in at least some contexts of human-algorithm interaction in the form of improving users’ assessment of the algorithm’s accuracy. As discussed in the earlier chapters, this is a salient form of benefit when users have to interact with modern day algorithmic systems where the responsibility often falls on the users to determine whether they should rely on, or ignore the outputs of the system. To this end, there are concrete design implications that we can extract from these results that may inform the designers of future algorithmic or AI systems.

However, it is just as important to note both the goal and the limitations of the results presented here. As noted in the introduction, the message of this thesis is not to suggest that users would gain benefits by slowing down the pace of interaction with an algorithmic system regardless of the context in which the interaction is taking place. Instead, I hope my experiments undid our mostly unquestioning belief that faster is better in human-algorithm interaction and convey that the speed of interaction is a worthwhile element of design to explore to benefit the users. Therefore, the limitations and the possible future work stemming from this thesis are particularly important part of the contribution I pose.

In this chapter, I discuss the immediate design implications as well as the limitations and suggestions for future work.

5.1 DESIGN IMPLICATIONS

An important design implication of the results presented here is to start re-imagining our relationship with technology in contemporary judgment and decision-making scenarios. Ever since Douglas Engelbart presented his 1968 demo of his user interface, the goal of human-computer interaction has been to augment human intelligence rather than to undermine or replace it [52]. Even though intelligent algorithms today make judgments and decisions that are seemingly as good as, or even better than, those of humans, it would be unwise for us to fully delegate all decision-making tasks to machines and be subjected to their potential biases and flaws. Rather than blindly accepting or rejecting the decisions made by algorithms, perhaps we can use waiting time as a time to reflect on and assess the algorithm in the process of decision-making.

But in what instances would the results presented here be most applicable? One way to unpack this question could be through the lens of two attributes of algorithmic systems that
are central to the struggles of designing human-algorithm interaction as discussed by Qian Yang et al [11]. The two attributes include 1) capability uncertainty that speaks to the fact that many algorithmic systems’ accuracy and capability are uncertain and potentially ever-changing as more data gets added, and 2) output complexity that refers to the fact that the forms of outputs from algorithmic systems can be both diverse (ex. from a numeric values suggesting the likelihood of an event, or a GPS system that suggests a route to a destination) and context dependent (ex. a voice assistant that produces natural sentences on the fly depending on the context of interaction).

If we try to place the ObjectRecognizer algorithm that suggests that number of jelly beans along these two attributes, the ObjectRecognizer could be considered to have bounded-capabilities (the training data was not updated during the experiment) and relatively simple in terms of its output (the output was given in the form of an integer that represented the estimated number of jelly beans). Yang et al. classify such algorithms as Level one: probabilistic systems, which includes algorithmic systems such as face-detection systems in camera apps, adaptive menus that are used to rank which option the user is likely to choose, clinical diagnostic systems, and text toxicity detectors that classify a phrase as toxic or non-toxic. The design challenges and the way the interaction between the user and the algorithm is choreographed are quite similar for these algorithms, and in these contexts, I suspect that slowness could provide benefits akin to what was presented in this thesis.

5.1.1 When Slowing Down Is Not Applicable

However, it is also clear that for certain contexts of interaction, slowness might not be the right choice to consider at all. For example, algorithmic systems such as GPS navigators and disaster relief systems require immediacy to be useful. The same is true for other algorithms such as matching algorithms that match ridesharing drivers to passengers, which we interact with frequently in low-stakes environments. In these cases, slowness could cause frustration, inefficiency, and harm. Additionally, it must be recognized that slow interaction with computing systems can lower the efficiency and productivity of the users, and cause frustration for all stakeholders. This could affect not only the users of the systems, but also the business interests of the companies that are creating and maintaining these algorithms in use today, potentially hindering more widespread consideration for slower interaction. What needs to be highlighted here is the complex nature of designing an algorithmic system; the benefits of a design decision, including slowing down the interaction, has to be analyzed from multiple angles and take into consideration the contexts in which the interaction takes place. When the benefits of slowness stand out despite its drawback, we have a good reason
to consider slowing down our interaction with an algorithmic system.

5.2 LIMITATIONS AND FUTURE WORK

I do not conclude from this work that slowing down any algorithm would always result in the outcomes and benefits similar to what is presented in this thesis. Earlier scholars in human-computer interaction have shown us that finding the optimal speed for interaction is complicated and context-dependent in traditional computing systems [32]. What I tried to show is that this logic could apply to our interaction with algorithmic systems. To this end, I studied how users interact with algorithms of varying response times in an experimental algorithmic setup that estimates the number of jelly beans in a jar. In doing so, I showed that the slow response time of an algorithm can bring cognitive benefits to the users in certain contexts. I hope this can rekindle our community’s interest in exploring slowness as an interesting venue for future studies in the context of human-algorithm interaction.

I suspect that there will be two strands of future work stemming from this thesis. One is relatively straightforward and has to do with searching for more scenarios in which slowness can improve our interaction with algorithmic systems as it did in my experiments. The other is more open-ended and has to do with imagining ways to take advantage of the fact that longer waiting time might be acceptable to provide extra values to the users. For the remainder of this section, I will briefly outline potential future work in both of these directions.

5.2.1 What Algorithmic Applications May Benefit From Slowness?

What would be potential scenarios in which slower interaction with algorithmic systems could provide benefits to the users? I suggested in the previous section that the ObjectRecognizer used in this study can be described as an algorithmic system that have bounded-capabilities (that is, the capability of the system does not change while the user is interacting with the system) and produces relatively simple probabilistic outputs in the form of numbers or categorical suggestions. Applications like this, as well as numerous clinical diagnostic systems or text toxicity detectors are what Yang et al. would classify as a probabilistic algorithmic system and share similar user experience flow and design challenges, and might be a good starting place to test out the benefits of slowness [11]. For example, when a human doctor is trying to determine whether a patient needs immediate care with the help of an algorithm that outputs a numeric risk assessment, would a slower interaction with the algorithm help the doctor to make a better decision?
Future work also needs to verify whether the findings hold under different and more complex contexts of human-algorithm interaction. These may include scenarios in which the capacity of the algorithmic system is continuously changing as more data is added, or the output space of the system is much more complex than simple categories or integers. For example, newsfeed algorithm, search engines, or other forms of recommender system would pose as interesting scenarios to investigate. Imagine a scenario in which a user is browsing on their newsfeed on a major social media platform. What would a slower newsfeed look like, and would it help the user better judge the relevance of the contents appearing on the newsfeed instead of becoming easily allured by clickbait titles (there is also a broader discussion to be had about slow media that is beyond the scope of this thesis. Today, social, and online news media flood information consumers with catchy and controversial titles at a pace that is too fast for the consumers to meaningfully parse. As a response, the call for “slower, better news” is gradually getting its attention [53]. There likely is a role that literally slowing down the media platforms can have in this mission of slow media)?

5.2.2 How Can We Further Enrich the Waiting Time?

But we should focus not only on identifying scenarios in which slowing down our interaction with an algorithmic system may provide benefits, but also consider ways to actively enrich and transform the waiting time to benefit users by taking advantage of the fact that a system may take more time to return its result in certain contexts. Jaime Teevan et al.’s work on slow search takes a step towards this direction by suggesting that if a longer response time is allowed, that extra time could be used to curate more relevant, higher quality search results to the users [4]. Meanwhile, some have suggested that we can further enhance the thinking process of the users during the waiting time by designing the interaction with the notion of serendipity in mind [21, 54]. For example, instead of showing a blank page with a loading gif, can an algorithmic system show incremental changes to the algorithm’s output? Prior work in data visualization such as Rahman et al.’s incremental visualization has suggested that when visualizing a dataset, the dataset can be sampled to generate visualizations faster while improving the displayed estimates incrementally, until the displayed estimates converge to the exact visualization computed on the entire data [55]. How would such an idea generalize to contexts in human-algorithm interaction?

The works mentioned here represent some of the exciting efforts that have been put to enrich the waiting time, but many of the ideas explored need further investigation. I hope that by showing that simply slowing down the pace of interaction could provide benefits to the user, my thesis can inspire further efforts in this regard.
5.2.3 Going Beyond the Speed of Interaction

Finally, going beyond exploring the speed of interaction, we should start thinking about what other long-held tenets of human-computer interaction may need to be revisited. I hope it was clear that design is a means to an end and that making an algorithmic system fast or slow just to adhere to a convention likely is a pointless exercise. Indeed, what matters is the affect that our design decisions have on our users. One main reason why this thesis explored the role of slowness is because its potential benefits to the users were largely overshadowed by our convention for fast interaction with technology. Similarly, there could be other design techniques that are overlooked that may provide insights to the users of algorithmic systems. For example, if seamless and efficient interaction with technology has been the focus of design so far, what should be the focus going forward as we welcome the days of advanced algorithms and AI? At least some portions of our design tenets will have to be re-envisioned to ensure that our technology continues to provide values to the users instead of undermining them, and future work should further investigate this.
CHAPTER 6: CONCLUSION

My thesis presents empirical evidence that there can be benefits for users in slowing down the response time of algorithms. In the context of the experiments presented here, the waiting time was often used by the participants to reflect on the task and the estimation process of themselves and the algorithm, and to compare and reevaluate the two processes. This process of reflection often produced benefits to the participants’ ability to make decisions with an algorithm by helping them better understand the task, and by encouraging them to more carefully evaluate their answers. This resulted in a statistically significant difference between those who interacted with a fast algorithm and those who interacted with a slow algorithm in the measures we used to determine how well the participants evaluated the accuracy of the algorithm’s output. In our discussion, we laid out where these results would most likely be applicable, and where we would need further work.

But most importantly, I hope that this thesis was able to convey that time, and more specifically, slowness, is an element of design in human-algorithm interaction that is worth exploring. Without a doubt, slowing down our technology comes at a cost. Prior work discussed in this thesis has shown that a slow response time of computing systems, whether it’s the traditional computers explored by the early scholars in human-computer interaction [5] or the newer algorithms [4], could lead to lower satisfaction, productivity, and engagement. Even in my results, a few of the participants who interacted with the slow algorithm expressed some degree of frustration and hoped for a faster interaction. However, decisions in an ever growing number of areas such as the justice system, the employment market, and the medical field are being made by algorithms. These are deeply consequential decisions that could have profound impact on individuals and society. Perhaps then, users’ satisfaction, productivity, and engagement – some of the most widely used dimensions to evaluate our technology – might not be the right measures to optimize for. In such contexts, if slowing down our technology offers us an opportunity to make better, and more conscious decisions with algorithms, we need to go beyond our most accepted evaluation metrics and design tenets to explore and experiment with the potential of slow algorithms.
REFERENCES


