The Apple Does Fall Far from the Tree: User Separation of a System from its Developers in Human-Automation Trust Repair

Theodore Jensen

Department of Computer
Science & Engineering
University of Connecticut
Storrs, CT, USA
theodore.jensen@uconn.edu

Yusuf Albayram

Department of Computer
Science & Engineering
University of Connecticut
Storrs, CT, USA
yusuf.albayram@uconn.edu

**Mohammad Maifi Hasan
Khan**

Department of Computer
Science & Engineering
University of Connecticut
Storrs, CT, USA
maifi.khan@uconn.edu

Md Abdullah Al Fahim

Department of Computer
Science & Engineering
University of Connecticut
Storrs, CT, USA
md.fahim@uconn.edu

Ross Buck

Department of Communication
University of Connecticut
Storrs, CT, USA
ross.buck@uconn.edu

Emil Coman

Health Disparities Institute
University of Connecticut
Health Center
Hartford, CT, USA
coman@uchc.edu

ABSTRACT

To promote safe and effective human-computer interactions, researchers have begun studying mechanisms for “trust repair” in response to automated system errors. The extent to which users distinguish between a system and the system’s developers may be an important factor in the efficacy of trust repair messages. To investigate this, we conducted a 2 (reliability) x 3 (blame) between-group, factorial study. Participants interacted with a high or low reliability automated system that attributed blame for errors internally (“I was not able...”), pseudo-externally (“The developers were not able...”), or externally (“A third-party algorithm that I used was not able...”). We found that pseudo-external blame and internal blame influenced subjective trust differently, suggesting that the system and its developers represent distinct trustees. We discuss the implications of our findings for the design and study of human-automation trust repair.

Author Keywords

Trust repair; human-automation trust; blame; attribution theory.

CCS Concepts

•Human-centered computing → HCI theory, concepts and models; Empirical studies in HCI; Collaborative interaction;

INTRODUCTION

Trust has been studied in various disciplines as a fundamental factor in human relationships [1, 4, 36]. Additionally, researchers have investigated the notion of a human’s “trust” in machines, computers, robots, and automation in general [12, 19]. Given the growing prevalence of automated systems in various domains (e.g., autonomous drone systems, self-driving cars, home assistants), understanding the factors that break and repair a user’s trust is increasingly important for system design.

For instance, consider an autonomous drone system used in a surveillance task. If the system misidentifies an image, leaving you exposed to a threat, you may feel that it violated your trust. In a comparatively low-risk scenario, if you use a home assistant system (e.g., Amazon Echo) to manage your calendar and end up missing an important appointment, the experience will likely affect your willingness to rely on the automation in the future. In both cases, the system has the opportunity to repair trust, that is, to respond to the trust violation in an effort to improve future outcomes. We pose the question—*are the developers of an automated system implicated in its mistakes? Or is the system itself deemed a responsible actor that can repair broken trust?*

While users likely correctly understand that these machines are products of human design, evidence suggests that humans respond to computers socially [30]. Perhaps, then, the user is engaging in a trusting relationship the system itself. If so,

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
DIS '19, June 24–28, 2019, San Diego, CA, USA.
Copyright © 2019 Association of Computing Machinery.
ACM ISBN 978-1-4503-5850-7/19/06 ...\$15.00.
<http://dx.doi.org/10.1145/3322276.3322349>

users may be prone to poor “trust calibration” [19], that is, the extent to which their perceptions (subjective trust) and, ultimately, their reliance decisions (behavioral trust) are appropriate relative to a system’s actual ability to perform in a given context (reliability). Discounting a fundamental design flaw as a non-repeatable offense that the system can correct may lead to misuse of the system in unsafe circumstances. On the other hand, diagnosing a rare system malfunction as a serious miscalculation by developers could lead to disuse of the system when it could actually provide an advantage. These negative outcomes represent “overtrust” and “undertrust,” respectively [31]. The current study seeks to elucidate whether a system-developer separation influences trust calibration by investigating the effects of blame attribution to a system or to its developers.

We recruited 147 participants on Amazon Mechanical Turk (MTurk) to play an online game where they collaborated with an Automated Target Detection (ATD) system in 5 rounds of an image classification task. In a 2 (reliability) x 3 (blame) between-group study, participants interacted with a high or low reliability system. After each round, the system displayed a text message acknowledging its errors in identifying images in the previous round, attributing blame either internally (“I was not able...”), pseudo-externally (“The developers were not able...”), or externally (“A third-party algorithm that I used was not able...”). Participants chose how many images to allocate to the ATD system (i.e., behavioral trust) and were compensated based on combined manual and automated performance. After gameplay, participants responded to a survey on subjective trust.

We found that reliability influenced both behavioral and subjective trust, while blame influenced subjective trust. Specifically, internal blame led to more positive impressions of the system than pseudo-external blame, suggesting that a system’s errors are not considered the same as the developers’ errors.

In the paper, we first review related areas of work in both human-human and human-computer interaction. Then, we describe our study methodology and hypotheses. Lastly, we present statistical analyses of gameplay and survey data and discuss the implications of our results.

RELATED WORK

In the broad literature on trust, Mayer, Davis, and Schoorman’s [25] definition is one of the most widely accepted:

The willingness of a party to be vulnerable to the actions of another party based on the expectation that the other will perform a particular action important to the trustor, irrespective of the ability to monitor or control that party.

Despite its original application to trust between humans, this definition readily applies to interactions with automation. For instance, if we trust the Amazon Echo, we expect that it will maintain an accurate, updated schedule of our activities. As we are unable to see inside the black box of the system, our trust involves an acceptance of the risk that it may not end up helping us. If we trust an autonomous vehicle, we expect that it will accurately recognize and respond to obstacles in the driving path. Again, because we cannot fully comprehend the

inner workings of the system, our trust exposes us to the potential consequences of an unreliable system. This notion that a non-human may be “trusted” is supported by the Computers are Social Actors (CASA) paradigm.

Computers are Social Actors & Human-Computer Trust

As its namesake suggests, CASA research has found that, in interactions with computers, users can “*be induced to elicit a wide range of social behaviors, even though [they] know that the machines do not actually possess feelings*” [30]. For instance, experiments have shown that people use politeness toward computers, as well as social rules regarding praise of others and praise of self (see [30] and [34] for more examples). Although research on trust in automation tends to tailor the construct to the trustees’ non-human attributes [19, 22], CASA predicts that social rules of trust reserved for other humans are extended to automation.

Might such an attribution of trust occur because of the machine’s ultimately human creators? Along these lines, “institutional trust” has been measured as a way to capture a trustor’s perceptions of the people behind a system [13, 21, 26]. Similarly, Hoff and Bashir [12] propose that developers are indirectly the trustees, and that trust in automation may be viewed as “*a specific type of interpersonal trust in which the trustee is one step removed from the trustor.*” However, in CASA studies, participants have generally denied responding socially to the computer and remarked that they were not thinking of any human during their interaction. Thus, researchers have suggested that social responses to computers occur due to “mindlessness,” as an automatic response where the presence of social cues in a machine’s interface leads people to use scripts otherwise reserved for human-human interaction [29].

Sundar and Nass [39] tested the mindlessness mechanism by investigating whether social responses occur because the computer is perceived as a *medium* between user and developer. In their experiment, one group of participants interacted with a machine consistently referred to as “Computer” while the other group’s machine was referred to as “Programmer.” Authors predicted that, if the computer were simply considered a medium between user and developer, there would be no difference between experimental conditions. However, those in the “Computer” condition perceived their interaction more positively than those in the “Programmer” condition. Authors suggest that mindlessness prevented participants in the computer condition from considering the developers. Computers appear to be treated as distinct *sources*. We extend this work to the context of trust repair to observe whether an automated system is treated as a distinct *trustee*, and not merely an extension of its developers.

Trust Repair

Referring to trust as the “glue” that holds relationships together, Lewicki and Brinsfield [20] note the importance of trust repair following a violation of trust:

...it is essential that this glue be rebonded if a broken relationship is to have any hope of returning to a productive or fruitful state.

The authors go on to note various strategies for trust repair that are certainly not uncommon in day-to-day human-human interactions, such as apologies and denials [20].

Recently, de Visser et al. [8] called for human-computer trust researchers to consider systems with the capability of “*building and actively repairing trust*.” In their framework, errors by a system may be viewed as costly *relationship acts*, while positive interactions such as good performance are considered beneficial relationship acts. *Relationship regulation acts* are needed to maintain “*optimal relationship equilibrium*” and consist of repair acts and dampening acts, which should follow costly and beneficial relationship acts, respectively [8]. Relationship regulation by the system can help to maintain a user’s appropriately calibrated trust.

In this vein, researchers have studied how users perceive apologies by machines. Tzeng [41] found that, although apologies led to more positive impressions of a computer program in a computer-assisted guessing game, this did not reduce blame for poor game performance. Robinette, Howard, and Wagner [35] found that apologies and promises made immediately after a robot’s mistake were less effective than those made at the time users had to make their next reliance decision. Moreover, self-blame has been observed to lead to greater perceived trustworthiness of virtual agents [3].

In one study of human-human trust repair, Kim, Ferrin, Cooper, and Dirks [16] found that the effectiveness of apologies varies with the type of trust violation. Internal apologies (i.e., trustee assumes full responsibility for violation) were more effective than external apologies (i.e., trustee assumes partial responsibility and blames another person) when the violation was a matter of competence, that is, the trustee’s lack of knowledge. However, the opposite was true for an integrity violation, when the trustee knowingly violated trust. Authors suggest that this is a result of the “diagnosticity” of each type of violation—a highly competent individual may demonstrate low competence in some situation, whereas an individual with high integrity is unlikely to demonstrate low integrity. Admitting to low integrity via an internal apology is more diagnostic and, therefore, more hurtful to trust than blaming someone else. Quinn, Pak, and de Visser [8, 33] found tentative support for Kim et al.’s findings in the context of human-automation interaction.

We build upon these findings and apply attribution theory to observe whether system developers are considered an external target of blame with respect to an automated system.

Attribution Theory in Human-Computer Interaction

Attribution theory explores how, in response to a negative event, people attempt to identify and make sense of the event’s causes [42]. De Visser et al. [8] suggest that the work of Tomlinson and Mayer [40] on the role of attributional processes in trust repair can be applied to the human-machine context.

In Tomlinson and Mayer’s model, the locus of causality, controllability, and stability associated with a trust-violating event moderate how perceptions of trustworthiness are affected [40]. *Locus of causality* refers to whether the event was caused by

the trustee (i.e., internal) or another actor (i.e., external). *Controllability* is the degree to which an actor had control over the situation. *Stability* reflects the likelihood that the cause will reoccur in the future [42]. Trustworthiness perceptions, also referred to as trusting beliefs, consist of the perceived ability, integrity, and benevolence of the trustee. *Ability* or *competence* consists of the trustee’s skills in a particular domain. *Integrity* reflects that the trustee adheres to a set of principles that are acceptable to the trustor. *Benevolence* is the desire of the trustee to do good for the trustor [25].

Social accounts such as apologies and blame can repair trust by managing the trustor’s attributions of a trust violation and, in turn, their trustworthiness perceptions [40]. For example, perceptions of the trustee’s ability may be less likely to be affected if the trustee convinces the trustor that the event was caused by some external actor or circumstance.

We investigate whether developers are considered an external locus of causality following errors by an automated system. This proposition implies that, to an extent, the automation acts independently from its developers’ control—that computers have agency. In fact, in an interview-based study of 29 computer science majors, Friedman found that 79% of participants judged computers to have decision-making capabilities and 45% judged computers to have intentions, although nearly all participants agreed that a computer’s “decision-making” and “intentions” were different from a human’s [9]. The study also found that participants who did not blame a computer for errors gave reasons that diminished its agency. Those who did blame the computer often reasoned about its participation in the events leading up to the error.

Researchers have further investigated this idea of computers as “scapegoats” to which blame can be attributed. For example, a self-serving bias has been identified where people are less willing to blame computers that are similar to themselves [28]. Moreover, the greater degree of autonomy of an agent [38] as well as mere perceptions of a teammate as human or AI [27] have been observed to influence the assignment of blame. Prior work has found that minor linguistic manipulations in a virtual driving assistant’s messages (e.g., using “you” instead of “we”) can influence the degree to which a driver attributes responsibility to the system, as well as their perceptions of the system in general [15].

While there are numerous approaches for supporting user trust in automation (e.g., transparency [17], agent anthropomorphism [7]), observing how blame of developers compares to blame of the system can give insight into *where* the user is placing their trust. This consideration is relevant to any design that aims to promote good trust calibration.

METHODOLOGY

To investigate the extent to which users distinguish between system and developers, as well as how this varies with system reliability, we designed a 2 (reliability: high, low) x 3 (blame: internal, pseudo-external, external) between-group, factorial study with the following hypotheses.

Internal	“I am sorry that X images assigned to me were counted as misidentifications. I was not able to process those images.”
Pseudo-External	“I am sorry that X images assigned to me were counted as misidentifications. The developers were not able to account for processing those images.”
External	“I am sorry that X images assigned to me were counted as misidentifications. A third-party algorithm that I used was not able to process those images.”

Table 1. Feedback messages. Based on a participant’s blame condition, these messages were identical across reliability groups. “X” represents the number of images that the automation was unable to identify in the previous round.

First, we expected a main effect of reliability on participants’ trust:

H_1 : High reliability leads to greater trust in the system than low reliability.

Moreover, in line with CASA, we anticipated that blame of the system would have a different effect than blame of the developers:

H_2 : Pseudo-external blame (i.e., blame of developers) will influence trust differently than internal blame (i.e., blame of the system itself).

We designed an online game where participants were scored and compensated based on their performance with an automated system of high or low reliability. After each round, the system displayed a feedback message acknowledging errors in the previous round. Feedback messages in each blame condition are shown in Table 1. After gameplay, participants responded to a survey on their perceptions of the system. The details of our methods are presented below.

Game Design

In the “Target Identification Task,” participants had to classify 20 images of vehicles as “Dangerous” or “Not Dangerous” in each of 5, two-minute rounds with the help of the Automated Target Detection (ATD) system. We refer to the ATD system as “the automation” or “the system” in the rest of the paper.

The game represented a task where two drones (one automated, one manually controlled) monitor a large area for criminal activity. After clicking on a map marker, the manual drone icon moved toward that location. Upon arriving, an image of a vehicle was shown in the “Vehicle Identification Panel.” Participants then used “Zoom In,” “Zoom Out,” and “Rotate” buttons to manually inspect n images, where $0 \leq n \leq 20$. Non-dangerous vehicles had only text on top of them. Dangerous vehicles had numbers in addition to text. Correct and incorrect manual identifications were accompanied by a bell and buzzer sound, respectively. A timer counting down from 2 minutes was shown in the upper right part of the game interface, which is shown in Figure 1.

Figure 1. Target Identification Task interface. On the left, the map with a marker for each image is shown. A check mark is displayed at locations of correctly classified images, an “X” for incorrectly identified manual images, and a “?” for automation images that could not be processed. Accuracy for both parties is updated in real-time above the map. On the right, the Vehicle Identification Panel shows the current manual image. Participants answer “Is this vehicle dangerous?” with the “Yes” or “No” button after using the “Zoom In,” “Zoom Out,” and “Rotate” buttons to inspect the image. The timer at the upper right counts down from 2 minutes.

The automation ostensibly worked in parallel on $(20 - n)$ images (it did not actually process images, per se—it spent a fixed amount of time and had accuracy determined by the reliability condition). In the first round, n was set to 10. In later rounds, participants were able to choose how many images to allocate to the automation beforehand. This was used as a measure of behavioral trust, or reliance on the automation.

A similar drone-based monitoring game was used by Satterfield, Baldwin, de Visser, and Shaw [37] to study trust. Our design differs in that the participant and automation control only one UAV each, rather than multiple assets. Moreover, rather than intervening during gameplay, our participants allocate control to the automation before a given round. Lastly, while Satterfield et al. give participants extensive practice before their single experimental session, we decided not to use a practice round in order to observe participants’ trust development as they grew familiar with the system.

Scoring

For each round, participants were given a 100-point “Round Score” crediting their collective speed and accuracy with the automation. Speed was ostensibly important in order to identify criminals before they could harm people; accuracy so that resources were not inefficiently devoted to stopping innocent people. The Round Score was calculated by averaging a participant’s “Time Score” and “Accuracy Score.” The Time Score credited those with more time remaining in the 2-minute round. The Accuracy Score was based on the correctness of the user’s and the automation’s combined performance. The number of correct identifications, $n_{correct}$, for the automation in a given round was calculated as follows:

$$n_{correct} = \text{floor}(r * A_k)$$

where r was 0.6 and 0.9 for low and high reliability groups, respectively, and A_k represents the number of images allocated to the automation in the k^{th} round. For example, if 13 images

were allocated to the low reliability automation before a round, there would be 7 correct identifications and 6 images unable to be processed by the automation (i.e., “counted as misidentifications”). As a result, the automation improved speed, but could compromise accuracy. We pilot tested various scaling functions for the Time and Accuracy scores in order to produce Round Scores that discouraged full allocation to the automation or fully manual identification. The reliability condition thus allowed us to observe how participants calibrated their trust to an appropriate level as they grew familiar with the system’s capabilities [19]. The accuracy in each reliability condition was chosen based on prior work identifying 70% as the level at which automation is considered reliable [43].

In order to motivate good performance and appropriate trust calibration in the game, we compensated participants for their cumulative Round Scores at a rate of 100 points = \$0.40.

Feedback Page

After each round, participants were shown a feedback page containing three elements shown in Figure 2: 1) feedback message, 2) score information, and 3) allocation decision. The feedback messages for each group are shown in Table 1. All messages were displayed with a typewriter effect. Raw scoring information about the time and accuracy of both manual and automated identification was next displayed in a table, underneath which a participant’s Round Score was shown along with its associated compensation amount. A colored gauge indicated where the score fell out of 100 possible points. Lastly, participants were asked to choose how many images to allocate to the automation for the following round using a slider. A dialog box asked for confirmation of this choice before the next round of gameplay was started. Each panel had a time-delayed “Next” button to prevent participants from quickly advancing through the page.

Post-Gameplay Survey

The survey was hosted on our university’s Qualtrics server. To measure perceptions of the automation in the game, as opposed to the overall game interface or the computer they were using to complete the study, we explicitly noted that the phrase “the system” throughout the survey would refer to the ATD system that helped participants identify images during gameplay. Participants first were required to correctly answer a multiple choice question ensuring this understanding before they advanced to the survey.

The survey began with demographic questions on gender, age, race, education level, experience operating drones or UAV’s, military service, and frequency of video game playing. Manipulation check questions were also included to verify the effect of our independent variables and framing of the messages.

We measured participants’ attributions of responsibility for their performance in the game using 2 items adapted from Moon and Nass [28]. Participants used a 10-point slider from “You” to “Automation” indicating who was more responsible for 1) overall performance in the game and 2) Round Scores.

To complement our behavioral measure of trust, we measured trusting perceptions, or subjective trust, in the survey. Researchers have measured trust in automation in various

Figure 2. Feedback page. The feedback message is shown at the upper left. The upper right scoring panel contains a table with the correctness and time elapsed for manual and automated identification. The “Total” column shows the longer of the two times and combined correctness, which contribute to Time Score and Accuracy Score, respectively. The Round Score is shown with a colored gauge and its associated bonus compensation amount (see Compensation subsection in Methodology). In the lower panel, participants choose how many images to allocate to the automation for the next round. Adjusting either the manual or automation slider simultaneously adjusts both values.

ways. Jian, Bisantz, and Drury’s scale [14] is one of the most widely cited and has been applied to the study of apology messages [32]. In general, human-automation trust instruments [5, 23] incorporate dimensions similar to Mayer et al.’s trustworthiness characteristics: ability, integrity, and benevolence [25]. For example, Madsen and Gregor’s scale consists of five constructs that reflect these elements: perceived reliability, perceived technical competence, perceived understandability, faith, and personal attachment [23]. Researchers investigating trust in e-Commerce [26] and online recommendation agents [2] have directly adapted the trusting beliefs from Mayer and Davis [24]. However, such an approach has not been taken in the trust in automation literature. For this reason, we measured perceptions of trust in the ATD system with 1) trusting belief items from McKnight, Choudhury, and Kacmar [26] (originally from Mayer and Davis [24]) and 2) Jian et al.’s trust in automation scale [14], both rated on 7-point scales. Subjective trust items are included in the Appendix.

We also included 4 attention check questions throughout the survey.

Study Procedure

The study was posted as a Human Intelligence Task (HIT) on MTurk, available to workers 18 years or older, living in the United States, and having completed at least 1000 HIT's with an approval rating of 95%. When participants accepted the job on MTurk, a link directed them to the game website. The first page of the game displayed an information sheet, the end of which asked participants whether or not they consented for participation in the study.

Participants who gave consent were then brought to an instruction page explaining the task that the game represented, the importance of speed and accuracy in scoring, and game controls. At the end of this page, participants were required to correctly answer a series of questions to ensure their understanding of these aspects. One question ensured they were using audio to hear the sound effects. If a participant incorrectly answered a question, they remained on the instruction page until they answered all correctly. No feedback was given as to which answer was incorrect.

Participants were then shown the following introduction message: "Welcome! I will use these messages to communicate with you about my performance." After advancing, they began the first round of the game. We told participants beforehand that certain behaviors would prevent them from completing the study, including inactivity for a round where there were manual images to identify, refreshing the webpage, or clicking back to return to the previous webpage.

Data collected from gameplay were anonymous and linked to survey responses with a participant's randomized ID. Those who completed all rounds of gameplay and the survey submitted their random, Qualtrics-generated ID into MTurk to be compensated with \$2 for completion of the HIT. We utilized MTurk's bonus payment feature for the aforementioned bonus compensation based on cumulative Round Scores. The study was approved by our university's IRB.

EVALUATION

A total of 264 participants completed the study. We first conducted a series of screening procedures to ensure that our sample consisted of attentive MTurk users. We removed data for 13 participants who incorrectly answered at least 1 attention check question in the survey. Additionally, we removed data of 7 participants who allocated all 20 images to the automation in each of the final 4 rounds, a strategy indicating a clear lack of motivation and regard for the scoring mechanism. There were no participants who used fully manual identification in the final 4 rounds. For the remaining 244 participants, we next looked to our manipulation check questions.

Manipulation Checks

Because our focus was on the locus of blame in the messages, we wanted to ensure that participants recognized 1) that the source of the messages was the computer itself, and 2) the target of blame that aligned with their group.

We first asked participants what entity was communicating with the messages, with options "The computer," "The system developers," "Not sure," and "Other." Since all of those

	$n_{initial}$	Fail. Source	Fail. Blame	n_{final}
Low-Int.	38	4 (10.5%)	17 (44.7%)	21
Low-Pseudo.	42	12 (28.6%)	13 (31.0%)	19
Low-Ext.	43	10 (23.3%)	5 (11.6%)	29
High-Int.	41	6 (14.6%)	14 (34.1%)	25
High-Pseudo.	40	16 (40.0%)	9 (22.5%)	22
High-Ext.	40	7 (17.5%)	2 (5.0%)	31

Table 2. Group sizes and manipulation check failure rate. The percentages of participants within each group who failed the source and blame manipulation checks are shown. Some participants failed both manipulation checks.

who chose "Other" mentioned either the "the system," "the automation," or "ATD," they were coded as correct answers along with those who answered "The computer."

Next, we asked what entity was mentioned in the message as the cause of system errors, with options "The computer," "The system developers," "A third-party algorithm," "Not sure," and "Other." For those who chose "Other," who all were in an internal blame condition, if the participant mentioned that it was the system that was unable to process images, these were coded as correct answers.

Initial group sizes, the number of participants in each group who failed source and blame manipulation checks, and group sizes after screening out incorrect answers are shown in Table 2. The high rate of source failure in pseudo-external conditions may have resulted from explicit mention of the system developers. This may have prompted participants to think about the developers' role in creating the system's messages despite the use of "I." Likewise, the high rate of blame failure in internal blame conditions may have resulted from the target of blame being given implicitly (i.e., "I was not able to...") rather than explicitly as in the other conditions (e.g., "The developers were not able to...").

All subsequent analyses are conducted on the remaining 147 participants, with group sizes shown as n_{final} in the last column of Table 2.

To test the efficacy of our reliability manipulation, we asked participants to report how many images they would expect the automation to correctly identify out of 100 using a slider. Those in both the low (Mean = 56.1, Median = 60.0, SD = 11.1) and high (Mean = 82.7, Median = 85.0, SD = 14.4) reliability conditions accurately assessed the automation's reliability¹. A Mann-Whitney U-test confirmed that the difference in perceived reliability between groups was significant ($U = 5,164.00$, $p < 0.001$) and that our reliability manipulation was effective.

Sample Demographics

Of the 147 remaining participants, 83 (56.5%) were male and 64 (43.5%) female. The average age was 34.9 years (Median = 33.0, SD = 9.5). There were 115 (78.2%) white/Caucasian, 12 (8.2%) African American, 10 (6.8%) Asian, 7 (4.8%) Hispanic,

¹ While the mean of high reliability ratings is slightly below 90, this number is relatively accurate since allocating 11 or more images to the automation corresponded to 2 errors.

and 3 other participants. Of these, 85 (57.8%) reported having at least a 4-year college degree, 144 (98.0%) having never operated a military or commercial drone or UAV, and 10 (6.8%) having served in the military. Lastly, 57 (38.8%) participants reported playing games on their computer or mobile device daily, 57 (38.8%) a few times a week, and the remaining 33 (22.4%) a few times a month or less.

A Chi-Square test revealed that there were no significant differences between groups in terms of gender ($\chi^2(5) = 2.78, p = 0.73$). Using Fisher Exact Tests to handle cells with less than 5 participants, we found no significant differences between groups in terms of drone or UAV experience ($p = 0.43$), military service ($p = 0.75$), or race ($p = 0.48$). Moreover, using Kruskal-Wallis Tests, we found no significant differences in terms of age ($\chi^2(5) = 3.31, p = 0.65$), education level ($\chi^2(5) = 5.23, p = 0.39$), or gaming frequency ($\chi^2(5) = 7.41, p = 0.19$). We concluded that the groups are demographically similar and can be compared based on our manipulations.

The instruction page and 5 rounds of gameplay took an average of 11.1 minutes (Median = 10.2, SD = 4.2) while the survey took 9.8 minutes (Median = 9.3, SD = 3.9)².

Gameplay Behavior

For each group, the average number of images allocated to automation in each of the 5 rounds of gameplay is shown in Figure 3. The first, second, and third plots show internal, pseudo-external, and external blame conditions, respectively. Low and high reliability groups are shown as separate lines in each plot.

In general, Figure 3 gives an idea of how participants in each group calibrated their trust in the ATD system over time. The effects of reliability are relatively clear. Blame conditions also appear to have influenced participants' willingness to allocate to the automation, especially entering Round 2 after initial exposure to the feedback message. For example, in the low reliability condition, pseudo-external participants reduced their allocation by an average of nearly 3 images ($M = -2.89, SD = 2.33$) prior to Round 2 while internal ($M = -0.52, SD = 3.64$) and external ($M = -0.72, SD = 3.70$) participants reduced their allocation by an average of less than 1 image. These results and Figure 3 should be interpreted with caution given the large standard deviations of the metrics.

Analysis of Variance

To investigate our hypotheses, we conducted a two-way, 2 (reliability) x 3 (blame) Analysis of Variance (ANOVA) for various dependent variables.

First, we combined the 2 attributional items for overall performance and Round Scores into a single Attribution of Responsibility measure because they were highly correlated (Spearman's $\rho = 0.79, p < 0.001$). Higher values indicate that a participant considered the automation as more responsible than them for outcomes in the game.

²Reported survey time accounts for other items that are not mentioned in this paper due to space constraints.

Figure 3. Group mean allocation to ATD. Means for each group's allocation to automation over the 5 rounds of gameplay are shown. Error bars are excluded for ease of viewing, although the figure should be interpreted with caution given the large standard deviations of group allocation. The ANOVA section gives more detailed insight into significant group differences.

Next, behavioral trust was operationalized as Total Allocation (TA) and First Calibration (FC):

$$TA = \sum_{k=1}^5 A_k$$

$$FC = A_2 - 10$$

where A_k indicates the number of images allocated to the automation for the k^{th} round of gameplay. TA gives a sense of overall trust in the system and FC gives an impression of participants' immediate reactions to Round 1 gameplay and feedback messages. FC values are negative for participants who allocated fewer than 10 images for Round 2 and positive for those who increased their allocation above 10.

Lastly, subjective trust measures consisted of Jian et al.'s trust in automation scale (12 items, Cronbach's $\alpha = 0.88$) and the perceived trustworthiness characteristics: ability (4 items, $\alpha = 0.95$), integrity (4 items, $\alpha = 0.88$), and benevolence (3 items, $\alpha = 0.71$). All of the subjective trust scales demonstrated acceptable reliability [10] and are included in the Appendix.

The correlations between dependent variables are shown in Table 3. All post-hoc pairwise comparisons between blame conditions were done using Tukey's HSD. We report effect size as partial eta-squared, η_p^2 , which represents the proportion of variance explained by a predictor relative to the error term in a given model [6].

Since the attributional measure was the only dependent variable to yield an insignificant ANOVA model, we focus on trust measures. The details of our analyses follow.

Total Allocation

There was a significant main effect of reliability on Total Allocation ($F(1, 141) = 30.36, p < 0.001, \eta_p^2 = 0.177$). Participants in the high reliability condition ($M = 51.06, SD = 9.28$) allocated more images to the automation overall than those in the low reliability condition ($M = 41.10, SD = 11.95$).

	1	2	3	4	5	6
1– Total Allocation	—					
2– First Calibration	0.71**	—				
3– Trust in Automation	0.45**	0.47**	—			
4– Ability	0.38**	0.47**	0.79**	—		
5– Integrity	0.09	0.18*	0.50**	0.55**	—	
6– Benevolence	0.29**	0.29**	0.51**	0.53**	0.65**	—
7– Attribution of Responsibility	0.44**	0.18*	0.11	0.16*	0.02	0.20*

Table 3. Correlation between dependent variables. Spearman's ρ between each pair of variables are shown. ** $p < 0.01$, * $p < 0.05$.

First Calibration

There was also a significant main effect of reliability on First Calibration ($F(1, 141) = 57.86, p < 0.001, \eta_p^2 = 0.291$). While participants in the high reliability condition ($M = 3.08, SD = 3.54$) increased their allocation to automation following Round 1, those in the low reliability condition ($M = -1.26, SD = 3.47$) decreased their allocation.

Trust in Automation

The ANOVA on Trust in Automation revealed a significant main effect of reliability, ($F(1, 141) = 45.82, p < 0.001, \eta_p^2 = 0.245$). High reliability participants ($M = 5.16, SD = 0.89$) trusted the system more than low reliability participants ($M = 4.13, SD = 0.86$).

Next, we conducted a Multivariate Analysis of Variance (MANOVA) for the perceived trustworthiness characteristics. As this test revealed significant main effects of reliability ($F(3, 139) = 24.29, p < 0.001, \text{Wilks' } \lambda = 0.656, \eta_p^2 = 0.344$) and blame ($F(6, 278) = 4.01, p = 0.001, \text{Wilks' } \lambda = 0.847, \eta_p^2 = 0.080$), we conducted an ANOVA for each characteristic, using a Bonferroni-adjusted significance level of $\alpha = 0.05/3 = 0.0167$.

Ability

There was a significant main effect of reliability ($F(1, 141) = 65.89, p < 0.001, \eta_p^2 = 0.318$)³. High reliability participants ($M = 5.20, SD = 1.03$) had significantly greater perceptions of the system's ability than low reliability participants ($M = 3.53, SD = 1.35$).

Integrity

There were significant main effects of reliability ($F(1, 141) = 7.05, p = 0.009, \eta_p^2 = 0.048$) and blame ($F(2, 141) = 9.97, p < 0.001, \eta_p^2 = 0.124$) on integrity. High reliability participants ($M = 4.88, SD = 1.18$) had significantly greater perceptions of the system's integrity than low reliability participants ($M = 4.30, SD = 1.23$). Moreover, post-hoc analysis revealed that internal blame participants ($M = 5.17, SD = 1.20$) had significantly greater perceptions of integrity than both pseudo-external ($M = 4.07, SD = 1.03$) ($p < 0.001$) and external participants ($M = 4.55, SD = 1.23$) ($p = 0.016$).

³Levene's test revealed that the variance in ability beliefs across groups was not equal. However, the p-value for this effect was substantially below our threshold, suggesting that the result is robust in the face of heterogeneous variances.

Benevolence

There were significant main effects of reliability ($F(1, 141) = 15.58, p < 0.001, \eta_p^2 = 0.100$) and blame ($F(2, 141) = 5.68, p = 0.004, \eta_p^2 = 0.075$) on benevolence. High reliability participants ($M = 4.51, SD = 1.14$) had significantly greater perceptions of the system's benevolence than low reliability participants ($M = 3.69, SD = 1.25$). Additionally, internal blame participants ($M = 4.57, SD = 1.17$) had perceptions of benevolence that were significantly greater than pseudo-external participants ($M = 3.75, SD = 1.12$) ($p = 0.003$), and marginally significantly greater than external participants ($M = 4.04, SD = 1.31$) ($p = 0.050$).

DISCUSSION

Overall, supporting H_1 , we found a consistent main effect of reliability, wherein high reliability participants demonstrated greater behavioral trust and reported greater subjective trust than low reliability participants. Also, although blame did not significantly affect behavior, we found partial support for H_2 in that pseudo-external blame caused lower perceptions of the system's integrity and benevolence than internal blame, suggesting that the system's mistakes were not inherently attributed to its developers. Instead, it appears that framing errors as the developers' responsibility led to more negative perceptions of the system's trustworthiness.

We discuss this apparent separation of system and developers in the context of the Target Identification Task, as well as the implications of this finding and some possible directions for future trust repair research.

The Target Identification Task as an Environment of Human-Automation Collaboration and Trust

The online game we developed appears a ripe context for the study of human-automation trust. Applying Mayer et al.'s definition of trust, participants were not able to "control" the ATD system during the game. They could only take a leap of faith before each round in deciding how many images to allocate to it. This represented a "willingness... to be vulnerable to the actions" of the system, with the "expectation" that the system would assist in achieving good performance [25].

To create the perception of vulnerability and risk that is necessary for trust, the scoring function in the game penalized lower accuracy that could result from using the automation. To create a need for reliance on the automation, it penalized slow speed that could result from manual identification. The

main effect of reliability lends to the effectiveness of this scoring mechanism. In support of H_1 , high reliability participants chose to allocate more images to the automation than low reliability participants over the course of gameplay. This effect carried over into subjective reports of the system's ability, integrity, and benevolence.

Contextual factors of this particular study, such as MTurk users' incentive to quickly complete HIT's, may influence the validity of allocation as a behavioral measure of trust. While we controlled for this with bonus compensation and attention checks, and corroborated with subjective trust measures, in-lab studies may reduce this particular bias of an MTurk sample.

We believe our findings generalize to other instances of human collaboration with or use of automated systems, yet recognize that some aspects of the game may not apply to practical systems. For instance, in order to quantify behavioral trust, we allowed participants a spectrum of control over the automation. In practical systems, reliance may be a binary decision where a user decides whether or not to use automation. We encourage studies that build on our findings in contexts where reliance takes the latter form. Also, the ATD system performed with consistent accuracy in both reliability conditions. As systems become increasingly autonomous, future studies should consider what type of relationship regulation acts can maintain appropriately calibrated trust if system behavior and reliability are dynamic.

Conceptual Separation of Developers and the System

The crux of our experiment lies in that internal and pseudo-external blame have the same technical implications for the system. The messages "I was unable to process those images" and "The developers were not able to account for processing those images" should theoretically have the same effect on trust, since the system's inability is ultimately a result of its developers not being able to give it some functionality. Despite this, we found support for H_2 in that there were differences in subjective trust between internal and pseudo-external blame conditions.

Specifically, internal blame led to significantly greater integrity perceptions than pseudo-external and external blame. The difference between the latter two groups was not significant. Internal blame also led to significantly greater benevolence perceptions than pseudo-external blame. Although we did not find significant results in terms of behavioral trust, group means for First Calibration indicate that these reduced integrity and benevolence perceptions may have initially impacted allocation decisions. Understanding the situations in which subjective trust (e.g., perceived integrity and benevolence) influences reliance behavior is an important area of future research.

These findings coincide with those where self-blame was associated with more positive perceptions of a robot [11] and virtual agent [3] than blame of other parties. Yet further, the fact that mentioning developers led to differences in perception suggests that developers are considered an *other* relative to the system. This is in line with Sundar and Nass' finding that computers are distinct sources and not mere media be-

tween user and developer [39], and suggests that trust in an automated system is not exactly the same as trust in its developers. In other words, errors that are attributed to developers appear to influence trust differently than errors attributed to the system itself. In our study, this manifested in perceptions of the system's trustworthiness.

The Ability, Integrity, and Benevolence of Non-Human Trustees

The trust in automation literature generally conceptualizes and measures subjective trust in ways specifically crafted for such non-human trustees. For example, Lee and Moray [18] refer to performance, process, and purpose as bases for trust. These are essentially machine-oriented synonyms for Mayer et al.'s ability, integrity, and benevolence [25]. Taking a CASA approach, we directly applied Mayer et al.'s human-human trustworthiness characteristics to the automated system in our study.

It is important to note that each of these trusting belief items in our study referred directly to "the system" as the trustee. We intentionally did not mention the developers. While some items may have appeared illogical (e.g., "The system is concerned about my well-being, not just its own," "The system is sincere and genuine"), we found that responses in each group were approximately normally distributed and moderately correlated with Jian et al.'s trust scale. It appears that participants were able to think about the concern and sincerity of the automated system.

In general, the trusting beliefs seem to paint a more thorough picture of the trustworthiness of automated systems. Not only did low reliability predictably reduce perceptions of ability, integrity, and benevolence, but manipulations in message content influenced integrity and benevolence perceptions. In fact, the effect of blame on perceived integrity ($\eta_p^2 = 0.124$) was nearly three times the size of the effect of reliability ($\eta_p^2 = 0.048$). This effect of blame on perceptions of the system was not captured by Jian et al.'s scale.

In some sense, blame of developers may have reduced integrity perceptions because it seemed hypocritical on the part of the system (integrity items refer to the system being "truthful" and "genuine"). It may also be that internal blame is the most conventional, and that participants were not used to a system blaming another entity. Yet, this study may also provide evidence that the "integrity" of an automated system reflects the qualities given to it by the developers, much like the integrity of a human consists of their "set of values" [25]. Recall Kim et al.'s suggestion that admitting to an ability violation is less diagnostic, while admitting to an integrity violation reflects on more central aspects of the trustee's character and implies that they cannot be trusted in the future [16]. Internal blame by a computer may signal an isolated, ability-based violation of trust, whereas blame of the developers may reflect deeper-seated problems with the system's "integrity."

While further studies are needed to confirm the effects of the system-developer conceptual divide, designers should consider how the framing of error messages can impact perceptions of

their systems. If framing errors as caused by the computer itself signals a lack of system ability, and not integrity, users may regard the violation as impermanent, expecting that the system's future behavior in similar circumstances will be different. This may lead to misuse [31] of the system in a situation where the error does repeat itself. Systems such as the Amazon Echo that are designed for particularly anthropomorphic interactions (e.g., reference by a human name, use of speech) may be prone to such overtrust if users tend toward perceiving them as independent actors rather than programmed machines.

On the other hand, if framing system errors as caused by the developers signals a lack of system integrity, users may regard the violation as a more permanent flaw. This may lead to disuse [31] of the system when it can actually help. Figure 3 demonstrates this initial tendency toward undertrust, where in the low reliability condition, pseudo-external participants tended to decrease Round 2 allocation (i.e., First Calibration) more than the other blame conditions. This may have been due to reduced integrity and benevolence perceptions associated with blame of the developers, and is in line with the idea that trust in machines is initially based on faith, and later on perceptions of dependability and predictability that develop as the relationship progresses [12, 22]. Roughly equal allocation across blame conditions toward the end of the game suggests that experience with the system eventually informed participants' expectations more than the messages.

Nonetheless, reference to a system's developers may make its status as a machine more salient. This may be utilized by designers as a dampening act [8] to temper user's expectations of system performance and prevent overtrust. Alternatively, making the system seem less like a machine with internal blame may be utilized as a repair act [8] to avoid undertrust and disuse of systems. The relationship between perceived "machineness" and trust across contexts and users is an important area of future work. As systems become increasingly autonomous, understanding how users consider them as products of human design versus independent actors will only grow in importance. It is critical that trust repair mechanisms consider how this separation impacts calibration of trust.

CONCLUSION

In this study, we sought to observe whether users distinguish between an automated system and its developers when evaluating trust. We designed a game in which participants collaborated with automation in an image classification task. A high or low reliability system attributed blame for its errors internally, pseudo-externally, or externally. As expected, we found a main effect of reliability on both behavioral trust and trusting perceptions. Moreover, we found that internal blame and blame of the developers was perceived differently. These findings suggest that, when it comes to trust, the apple *does* fall far from the tree⁴—automated systems are not treated merely

⁴The expression "the apple *doesn't* fall far from the tree" means that a child has similar qualities to their parents (from <https://dictionary.cambridge.org/us/dictionary/english/apple-doesn-t-fall-far-from-the-tree>). In our case, the automated system is analogous to the child, and the developers to the parents.

as reflections of their developers, but as distinct social actors. This notion is critical for designers to ensure that users are able to accurately gauge the trustworthiness of systems, and for fostering a future of healthy human-machine interactions.

Acknowledgments

This material is based upon work supported by the Air Force Office of Scientific Research under award number FA9550-15-1-0490.

APPENDIX

Survey Items

Perceived Trustworthiness Characteristics (Trusting Beliefs)

Adapted from [24, 26]. Rated on a 7-point Likert scale from "Strongly Disagree" to "Strongly Agree."

Ability

- The system is competent and effective in identifying vehicles.
- The system performs its role of identifying vehicles very well.
- Overall, the system is a capable and proficient means for identifying vehicles.
- In general, the system is very knowledgeable about identifying vehicles.

Integrity

- The system is truthful in its dealings with me.
- I would characterize the system as honest.
- The system keeps its commitments.
- The system is sincere and genuine.

Benevolence

- I believe that the system acts in my best interest.
- When I require help, the system does its best to help me.
- The system is concerned about my well-being, not just its own.

Trust in Automation

Adapted from [14]. Rated on a 7-point scale from "Not At All" to "Extremely" with the prompt "Please rate intensity of your feeling of trust, or your impression of the system while operating it."

- The system is deceptive
- The system behaves in an underhanded (concealed) manner
- I am suspicious of the system's intent, action, or outputs
- I am wary of the system
- The system's actions will have a harmful or injurious outcome
- I am confident in the system
- The system provides security
- The system has integrity
- The system is dependable
- The system is reliable
- I can trust the system
- I am familiar with the system

REFERENCES

- [1] Bernard Barber. 1983. *The logic and limits of trust*. Rutgers University Press.
- [2] Izak Benbasat and Weiquan Wang. 2005. Trust in and adoption of online recommendation agents. *Journal of the Association for Information Systems* 6, 3 (2005), 4.
- [3] Victoria Buchholz, Philipp Kulms, and Stefan Kopp. 2017. It's (Not) Your Fault! Blame and Trust Repair in Human-Agent Cooperation. (2017).
- [4] Ross Buck. 2014. *Emotion: A biosocial synthesis*. Cambridge University Press.
- [5] Shih-Yi Chien, Zhaleh Semnani-Azad, Michael Lewis, and Katia Sycara. 2014. Towards the development of an inter-cultural scale to measure trust in automation. In *International Conference on Cross-Cultural Design*. Springer, 35–46.
- [6] Jacob Cohen. 1973. Eta-squared and partial eta-squared in fixed factor ANOVA designs. *Educational and Psychological Measurement* 33, 1 (1973), 107–112.
- [7] Ewart J de Visser, Samuel S Monfort, Ryan McKendrick, Melissa AB Smith, Patrick E McKnight, Frank Krueger, and Raja Parasuraman. 2016. Almost human: Anthropomorphism increases trust resilience in cognitive agents. *Journal of Experimental Psychology: Applied* 22, 3 (2016), 331.
- [8] Ewart J de Visser, Richard Pak, and Tyler H Shaw. 2018. From 'automation' to 'autonomy': the importance of trust repair in human-machine interaction. *Ergonomics* (2018), 1–19.
- [9] Batya Friedman. 1995. "It's the computer's fault": reasoning about computers as moral agents. In *Conference Companion on Human Factors in Computing Systems*. ACM, 226–227.
- [10] Joseph A Gliem and Rosemary R Gliem. 2003. Calculating, interpreting, and reporting Cronbach's alpha reliability coefficient for Likert-type scales. Midwest Research-to-Practice Conference in Adult, Continuing, and Community.
- [11] Victoria Groom, Jimmy Chen, Theresa Johnson, F Arda Kara, and Clifford Nass. 2010. Critic, compatriot, or chump?: Responses to robot blame attribution. In *Proceedings of the 5th ACM/IEEE International Conference on Human-Robot Interaction*. IEEE Press, 211–218.
- [12] Kevin Anthony Hoff and Masooda Bashir. 2015. Trust in automation: Integrating empirical evidence on factors that influence trust. *Human Factors* 57, 3 (2015), 407–434.
- [13] Theodore Jensen, Yusuf Albayram, Mohammad Maifi Hasan Khan, Ross Buck, Emil Coman, and Md Abdullah Al Fahim. 2018. Initial Trustworthiness Perceptions of a Drone System based on Performance and Process Information. In *Proceedings of the 6th International Conference on Human-Agent Interaction*. ACM, 229–237.
- [14] Jiun-Yin Jian, Ann M Bisantz, and Colin G Drury. 2000. Foundations for an empirically determined scale of trust in automated systems. *International Journal of Cognitive Ergonomics* 4, 1 (2000), 53–71.
- [15] Ing-Marie Jonsson, Clifford Nass, Jack Endo, Ben Reaves, Helen Harris, Janice Le Ta, Nicholas Chan, and Sean Knapp. 2004. Don't blame me I am only the driver: impact of blame attribution on attitudes and attention to driving task. In *CHI'04 Extended Abstracts on Human Factors in Computing Systems*. ACM, 1219–1222.
- [16] Peter H Kim, Donald L Ferrin, Cecily D Cooper, and Kurt T Dirks. 2004. Removing the shadow of suspicion: the effects of apology versus denial for repairing competence-versus integrity-based trust violations. *Journal of Applied Psychology* 89, 1 (2004), 104.
- [17] René F Kizilcec. 2016. How much information?: Effects of transparency on trust in an algorithmic interface. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. ACM, 2390–2395.
- [18] John Lee and Neville Moray. 1992. Trust, control strategies and allocation of function in human-machine systems. *Ergonomics* 35, 10 (1992), 1243–1270.
- [19] John D Lee and Katrina A See. 2004. Trust in automation: Designing for appropriate reliance. *Human Factors* 46, 1 (2004), 50–80.
- [20] Roy J Lewicki and Chad Brinsfield. 2017. Trust repair. *Annual Review of Organizational Psychology and Organizational Behavior* 4 (2017), 287–313.
- [21] Xin Li, Traci J Hess, and Joseph S Valacich. 2008. Why do we trust new technology? A study of initial trust formation with organizational information systems. *The Journal of Strategic Information Systems* 17, 1 (2008), 39–71.
- [22] Poornima Madhavan and Douglas A Wiegmann. 2007. Similarities and differences between human-human and human-automation trust: an integrative review. *Theoretical Issues in Ergonomics Science* 8, 4 (2007), 277–301.
- [23] Maria Madsen and Shirley Gregor. 2000. Measuring human-computer trust. In *11th Australasian Conference on Information Systems*, Vol. 53. Citeseer, 6–8.
- [24] Roger C Mayer and James H Davis. 1999. The effect of the performance appraisal system on trust for management: A field quasi-experiment. *Journal of Applied Psychology* 84, 1 (1999), 123.
- [25] Roger C Mayer, James H Davis, and F David Schoorman. 1995. An integrative model of organizational trust. *Academy of Management Review* 20, 3 (1995), 709–734.
- [26] D Harrison McKnight, Vivek Choudhury, and Charles Kacmar. 2002. Developing and validating trust measures for e-commerce: An integrative typology. *Information Systems Research* 13, 3 (2002), 334–359.

- [27] Tim R Merritt, Kian Boon Tan, Christopher Ong, Aswin Thomas, Teong Leong Chuah, and Kevin McGee. 2011. Are artificial team-mates scapegoats in computer games. In *Proceedings of the ACM 2011 Conference on Computer Supported Cooperative Work*. ACM, 685–688.
- [28] Youngme Moon and Clifford Nass. 1998. Are computers scapegoats? Attributions of responsibility in human-computer interaction. *International Journal of Human-Computer Studies* 49, 1 (1998), 79–94.
- [29] Clifford Nass and Youngme Moon. 2000. Machines and mindlessness: Social responses to computers. *Journal of Social Issues* 56, 1 (2000), 81–103.
- [30] Clifford Nass, Jonathan Steuer, and Ellen R Tauber. 1994. Computers are social actors. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 72–78.
- [31] Raja Parasuraman and Victor Riley. 1997. Humans and automation: Use, misuse, disuse, abuse. *Human Factors* 39, 2 (1997), 230–253.
- [32] S Joon Park, Craig M MacDonald, and Michael Khoo. 2012. Do you care if a computer says sorry?: User experience design through affective messages. In *Proceedings of the Designing Interactive Systems Conference*. ACM, 731–740.
- [33] Daniel B Quinn, Richard Pak, and Ewart J de Visser. 2017. Testing the Efficacy of Human-Human Trust Repair Strategies with Machines. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, Vol. 61. SAGE Publications Sage CA: Los Angeles, CA, 1794–1798.
- [34] Byron Reeves and Clifford Ivar Nass. 1996. *The media equation: How people treat computers, television, and new media like real people and places*. Cambridge University Press.
- [35] Paul Robinette, Ayanna M Howard, and Alan R Wagner. 2015. Timing is key for robot trust repair. In *International Conference on Social Robotics*. Springer, 574–583.
- [36] Julian B Rotter. 1980. Interpersonal trust, trustworthiness, and gullibility. *American Psychologist* 35, 1 (1980), 1.
- [37] Kelly Satterfield, Carryl Baldwin, Ewart de Visser, and Tyler Shaw. 2017. The Influence of Risky Conditions in Trust in Autonomous Systems. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, Vol. 61. SAGE Publications Sage CA: Los Angeles, CA, 324–328.
- [38] Alexander Serenko. 2007. Are interface agents scapegoats? Attributions of responsibility in human-agent interaction. *Interacting with Computers* 19, 2 (2007), 293–303.
- [39] S Shyam Sundar and Clifford Nass. 2000. Source orientation in human-computer interaction: Programmer, networker, or independent social actor. *Communication Research* 27, 6 (2000), 683–703.
- [40] Edward C Tomlinson and Roger C Mayer. 2009. The role of causal attribution dimensions in trust repair. *Academy of Management Review* 34, 1 (2009), 85–104.
- [41] Jeng-Yi Tzeng. 2004. Toward a more civilized design: studying the effects of computers that apologize. *International Journal of Human-Computer Studies* 61, 3 (2004), 319–345.
- [42] Bernard Weiner. 1985. An attributional theory of achievement motivation and emotion. *Psychological Review* 92, 4 (1985), 549–573.
- [43] Christopher D Wickens and Stephen R Dixon. 2007. The benefits of imperfect diagnostic automation: A synthesis of the literature. *Theoretical Issues in Ergonomics Science* 8, 3 (2007), 201–212.