On Using Social Signals to Enable Flexible Error-Aware HRI

Maia Stiber
mstiber@jhu.edu
Johns Hopkins University
Baltimore, Maryland, USA

Russell H. Taylor
rht@jhu.edu
Johns Hopkins University
Baltimore, Maryland, USA

Chien-Ming Huang
chienming.huang@jhu.edu
Johns Hopkins University
Baltimore, Maryland, USA

ABSTRACT

Prior error management techniques often do not possess the versatility to appropriately address robot errors across tasks and scenarios. Their fundamental framework involves explicit, manual error management and implicit domain-specific information driven error management, tailoring their response for specific interaction contexts. We present a framework for approaching error-aware systems by adding implicit social signals as another information channel to create more flexibility in application. To support this notion, we introduce a novel dataset (composed of three data collections) with a focus on understanding natural facial action unit (AU) responses to robot errors during physical-based human-robot interactions—varying across task, error, people, and scenario. Analysis of the dataset reveals that, through the lens of error detection, using AUs as input into error management affords flexibility to the system and has the potential to improve error detection response rate. In addition, we provide an example real-time interactive robot error management system using the error-aware framework.

CCS CONCEPTS

• **Computer systems organization** → **Robotics**.

KEYWORDS

dataset, robot error, social signals, error management, HRI

ACM Reference Format:
Maia Stiber, Russell H. Taylor, and Chien-Ming Huang. 2023. On Using Social Signals to Enable Flexible Error-Aware HRI. In *Proceedings of the 2023 ACM/IEEE International Conference on Human-Robot Interaction (HRI '23), March 13–16, 2023, Stockholm, Sweden.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3568162.3576990

1 INTRODUCTION

Robot errors are inescapable and perennial in complex human-robot collaboration; these errors are mostly unexpected and come from a multitude of sources ranging from mechanical malfunctions (*e.g.,* failure to close the gripper when attempting to grasp) to uncertain perceptions and reasoning (*e.g.,* errors in intent recognition) to shifts in the environment (*e.g.,* new physical constraints emerge). If not managed appropriately, their negative effects on task success, safety, and human trust and willingness to work together will snowball and require more time and effort to recover from [16, 24, 34].

HRI '23, March 13–16, 2023, Stockholm, Sweden

ACM ISBN 978-1-4503-9964-7/23/03.
https://doi.org/10.1145/3568162.3576990

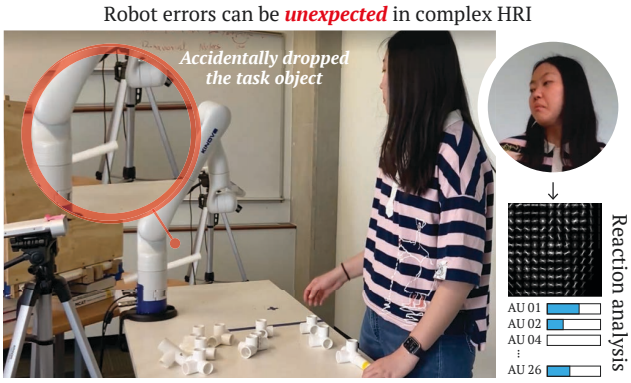

Figure 1: We present an error-aware robotic system framework that includes social signals as input. We illustrate the potential that social signals afford on improving the flexibility and timeliness of error detection via collecting, analyzing, and modeling a novel dataset—diverse in interaction contexts and tasks—of AU responses to robot errors.

Error detection is the critical first step in successful error management, followed by error identification, mitigation, and recovery. Automatic error detection methods proposed by prior work are largely domain specific (*e.g.,* [7]), requiring "hardcoding" rules to detect deviated task behaviors. Therefore, these methods are not adaptable to new, unexpected errors, scenarios, and tasks [19]. In this work, we ask the question: *how can we enable flexible methods for error detection that account for a range of unexpectedness and variability in people's behaviors and collaboration contexts?*

Social signals are a promising source of information for enabling flexible error detection. People exhibit a variety of social signals in response to robot errors due to their unexpectedness [13]; these signals include rich facial expressions, bodily gestures, and verbal responses. Most research exploring the relationship between social signals and robot errors has been situated in social contexts using a social robot (*e.g.,* [21, 28, 38]). More recently, studies [36, 37] have found that people exhibit similar social signals when encountering errors produced by a non-anthropomorphic robot manipulator in a non-social interaction context (*e.g.,* task demonstration).

As motivated by these prior studies, in this work, we aim to formalize the incorporation of social signals in error detection, and more broadly error management (Figure 1), and create resources—including a dataset, computational models, and a system—to foster the development of flexible, error-aware robotic systems for complex human-robot collaboration. Our work makes the following contributions to the HRI community:

- We introduce a new framework for error detection that augments *status quo* methods with social signal-based prediction (Section 3).
- We curate and present a dataset containing three physical human-robot interaction scenarios in which participants experienced different types of robot errors (Section 4). [1]
- We implement and release a flexible, real-time error-aware system to support a human-robot collaborative assembly scenario (Section 5). [2]

2 BACKGROUND

Technical and unexpected errors are unavoidable in deployed robotic systems; for example, it is difficult to decrease the mean time between failures below 72 to 216 hours in mobile robots deployed in museums [30]. Robot errors harm robot task performance, which negatively impacts user trust [34, 35]. The decrease in trust depends on the quantity and severity of the errors [29]. In addition, the situation and team members can dictate the impact those errors have on the collaboration [12, 33]. As a result, errors can create user resistance to collaborate [15].

2.1 Error Detection in HRI

Accurate and prompt error detection is critical for effective error mitigation and recovery, as user uncertainty of when or how errors might occur impacts recovery effectiveness [11, 26]. In part, this is because error management—consisting of error detection, classification, mitigation, and recovery—requires an understanding of the errors and their impact [16, 17]. So, a positive relationship can only be formed between the user and the robot if errors can be identified reliably, enabling appropriate recovery strategies [40].

Many error detection techniques in HRI, when specifically looking at physical interactions, are specially designed for each task and consider a set of predefined anticipated errors [19]. The resulting methods use task-specific or domain-specific information, including hierarchical action structure (*e.g.,* [25]), task structure (*e.g.,* [7]), characteristics of anticipated errors (*e.g.,* [9]), and past robot action to detect robot state anomalies [31]. None of these consider other factors, such as the teammate's values and task context, that affect whether robot actions are errors and so are not easily generalized across people, tasks, and errors [10, 33]. However, error detection adaptability is important because robot errors do not ascribe to our expectations of what could happen during a task [19].

2.2 Use of Socials Signals in HRI

Through user's natural response to human-robot interaction, information about robot actions, tasks, and users' mental models of a robot can be imparted to the robot. Social signals are already used to aid collaboration in HRI through conveying perceived task difficulty (*e.g.,* [1]), need for help (*e.g.,* [39]), and engagement breakdown (*e.g.,* [4]). Social signals are also dependable indicators of errors as people react to robot errors socially due to their unexpectedness [13, 27, 38]. Users display more social signals during situations with errors than ones without [8]. Common instinctive responses to robot errors are gaze [2, 20, 21], facial expressions [37, 38], verbalizations [21],

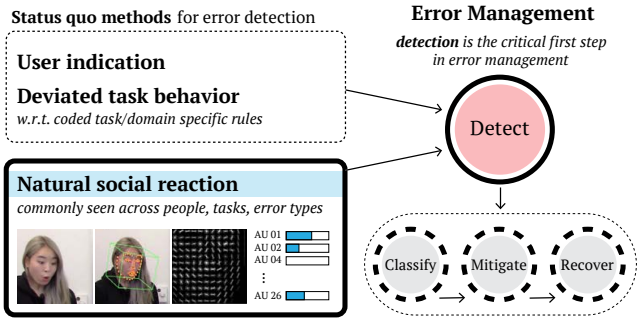

Figure 2: Our proposed error-aware framework contains three layers of input (explicit user indicator, implicit domain-specific indicator, and *implicit social signals indicator*).

and body movements [7, 13, 21]. Error severity and type were also found to influence participants' responses [21, 22, 36]. However, much prior work was contextualized in social scenarios interacting with humanoid robots, which may elicit different responses than interactions with non-anthropomorphic robots [20].

2.2.1 Robot Error Detection Using Social Signals. Two studies utilized social signals to detect robot errors automatically in HRI. One study showed that—with a collection of head, body, gaze, AU, and verbal tracking—it is possible to automatically detect and classify certain social errors during conversational failures with a social robot. In addition, different social signals were indicative of errors in different conversational scenarios [21]. The other study focused on physical-based interactions, exploring the feasibility of using facial AUs to automatically detect errors in real-time during Programming by Demonstration (PbD) scenarios. In addition, they suggested that error detection using social signals might be generalizable to identify errors of different types and in different tasks [37]—though the exploration was done between two different PbD tasks. Research in this area is still in early stages, especially for physical human-robot interactions. There has been little investigation into how to incorporate social signals into a full error-aware robotic system and what the advantages are by integrating it into the system.

3 TOWARDS FLEXIBLE, ERROR-AWARE HRI: A CONCEPTUAL FRAMEWORK

Adapting from teamwork principles in aviation [16, 17], we define error management as having four main aspects: (1) error detection, (2) classification, (3) mitigation, and (4) recovery. We propose that implicit human input—social signals—should be considered when designing error-aware robotic systems that address either error management as a whole or an aspect of it.

The general framework to error management [18] consists of explicit handling (*e.g.,* human manual reporting [14]) and/or implicit domain-specific information (*e.g.,* task tracking [7] and anomaly tracking [32]). While such *status quo* framework has the potential to provide robust error management in particular tasks and expected error types, it is too rigid to handle cross-context error management and unanticipated error types [19].

We propose modifying the error-aware robot system framework to consist of *three* layers of input: explicit markers, implicit domain-specific markers, and *implicit social signals markers*, and assert that

[1] https://github.com/intuitivecomputing/Response-to-Errors-in-HRI-Dataset
[2] https://github.com/intuitivecomputing/Error-Aware-Robotic-System

Table 1: Summary of dataset.

	HRC-A	HRC-C	PbD	
			Grocery	Case Study
Total duration (min)	151.11	257.50	25.05	23.47
Total duration reaction to error (min)	1.73	5.72	3.03	2.61
# of participants	12	33	23	5
# of pre-programmed errors per participant	2	2	1	3
Error type experienced	physical conceptual	conceptual	physical	physical conceptual generalization
# of cameras	2	1	2	2

this framework has the potential to allow flexibility across task and error type (Figure 2). Social signals in response to errors are commonly seen across people, task, and error types in non-social tasks and with non-social robots. In addition, social signals have the capacity to allow for earlier error management, as we will show through the analysis of our dataset. We note that this additional layer is meant to augment the *status quo* methods and is not necessarily robust enough alone, because not everyone exhibits social reaction [27] and people may overreact (novelty effect) [37] to robot actions. Yet, the rich information provided by social signals allows for new opportunities to capture unexpectedness and variability in complex human-robot collaboration. In our work, we demonstrate the potential of this framework and develop a real-time autonomous error detection system using facial AUs.

4 SOCIAL RESPONSE TO ERRORS IN HRI: A DATASET

To gain a more generalized understanding of humans' natural responses to robot errors in physical-based interactions and to examine the potential flexibility benefit of adding social signals to an error-aware framework, we curated a dataset from three HRI studies. We collected natural reactions to unexpected robot errors during physical-based, dyadic, human-robot interactions. Prior works have introduced datasets that contain natural user responses to robot errors (*e.g.,* [21, 27]); however, none—to the best of our knowledge—are open-source and focus on robot errors in physical human-robot interaction.

Our dataset was collated from three studies varying in task, error type, participants, robots, and scenario to enable analysis over a diverse array of interactions; see Figure 3 for a visual overview of the scenarios and Table 1 for a summary. Each data collection had participants do a practice task before the actual one and the resulting data was obtained from the actual trial. The data consists of calculated facial AUs per timestep (a third of a second) using OpenFace [3] on videos of the interactions. Each "entry" consists of the AUs from one error sequence: pre-error, during error, reaction to error, and post-error interactions. For all data, ground truths were determined by two independent coders, indicated by timesteps for *perceived error start* (when the coder is sure that the error is happening), *user reaction start* (first instance, post-error, where the participant's face visibly begins to move), and *user reaction end* (when the user's behavior returned back to their normal behavior). We evaluated the intercoder agreement using Intraclass Correlation Coefficient (ICC), as the measures were on a continuous scale, calculated using a two-way mixed, average measures, absolute agreement model. ICC values between 0.75 and 0.90 have good reliability and above 0.90 is excellent [23]. Based on these values, all evaluated coded metrics had good and excellent coded reliability.

Next, we present details of each data collection scenario including information about the tasks, errors, and data collection process.

4.1 Scenario 1: Human-Robot Collaborative Assembly (HRC-A)

The first set of data was collected in a human-robot collaboration scenario, where participants actively completed part of the task alongside the robot—a Kinova Gen3 arm—run by a completely autonomous robotic system (as described in Section 5). Data was collected using two synchronized static cameras in a real-time system, recording video and calculating AUs for each timestep, constructed on the Platform for Situated Intelligence (\psi) [6]. The intercoder reliability was 0.986 for *perceived error start*, 0.891 for *user reaction start*, 0.805 for *user reaction end*, and 0.998 for *explicit error reporting reaction time*.

4.1.1 Task Context. The task had participants and the robot collaboratively build PVC pipe tables. The participants were supplied with joints and an image of the structure they needed to build. The participants could request a pipe by saying *"Give me a [yellow or green] pipe"*, and the robot provided the requested pipe. It was up to the participant to assemble the structure. During the actual task, two pre-programmed errors happened (see section below for details). The participants were able to verbally and explicitly report errors using the phrase *"Report error"* any time during the task.

4.1.2 Error Manipulation. Two pre-determined robot errors occurred, unexpected for the participant, during the actual task—a physical error (failing to grasp a pipe) and a conceptual error (picking an incorrect pipe). Temporal error placement within the task was determined to allow a minimum of nine pipe requests before an error occurred; this choice was to let the participants develop a mental model of the robot and reduce the novelty effect of working with the robot. The ninth yellow pipe request triggered the conceptual error and the tenth green pipe request triggered the physical one. Therefore, the errors materialized at different times depending on how the user built the structure.

4.1.3 Participants. A total of 12 (8 male and 4 female) participants were recruited with ages ranging from 18 to 63 ($M = 26, SD = 13.14$). They had a medium amount of experience with technology ($M = 4.09, SD = 0.67$, a 7-point scale with seven being very experienced) and little experience with robots ($M = 2.25, SD = 1.14$).

4.2 Scenario 2: Human-Robot Collaborative Cooking (HRC-C)

Similar to the previous scenario, this scenario involved human-robot collaboration, but used a UR5 robot arm—ran autonomously—and did not use the same data collection system [5]. Instead, the data was videoed through a camcorder and later fed through OpenFace. The intercoder reliability was 0.983 for *perceived error start*, 0.826 for *user reaction start*, and 0.822 for *user reaction end*.

Human-Robot Collaborative Assembly (**HRC-A**) | Human-Robot Collaborative Cooking (**HRC-C**) | Programming by Demonstration (**PbD**)

Figure 3: Visual summary of the dataset, comprised of three data collections.

4.2.1 Task Context. The study, contextualized as a cooking task, contained one between-subjects factor: recipe delivery method (verbal delivery by robot vs. textual delivery). For the verbal condition, we obtained 115.69 min of AUs (2.60 min containing reactions to errors) and for the textual one 141.81 min (3.12 min containing reactions to errors). Participants were given a recipe (method depending on the factor) and received ingredients from the robot for them to "cook"; the task consisted of two recipes.

4.2.2 Recipe Delivery Method Manipulation. For verbal recipe delivery, the robot provided step-by-step instructions verbally and then gave the participants the respective ingredients. There was also a written recipe provided on the side. For textual recipe delivery, participants were only given a step-by-step written recipe and could request ingredients from the robot by saying "Please give me [name of the needed ingredient]". We note that the verbal condition added social-based interaction to an otherwise purely physical human-robot collaboration; therefore, another way to look at this study is social vs. non-social instructional delivery.

4.2.3 Error Manipulation. Regardless of the recipe delivery method, participants experienced the same two pre-programmed *conceptual* robot errors (*i.e.,* giving incorrect ingredients). There was one error per recipe, which occurred in the same place within the task for each participant. Note that even in the verbal condition, which introduces a social interaction to the physical collaboration, the error was related to the physical task and not a conversational failure.

4.2.4 Participants. The dataset includes 33 participants (15 female and 18 male) with ages ranging from 18 to 60 ($M = 26.21, SD = 9.50$). Seventeen participants were in the verbal delivery condition and 16 in the textual one. The participants were medium experienced with both technology ($M = 3.00, SD = 1.27$ on a 5-point scale with one as expert and five as beginner) and robots ($M = 3.33, SD = 1.31$).

4.3 Scenario 3: Programming by Demonstration (PbD)

The last scenario was a kinesthetic Programming by Demonstration with a Kinova Gen3 robot contextualized as a grocery unpacking task. A follow-up study, situated as a pipe and joint sorting task, was conducted to explore three types of errors: physical, conceptual, and generalization. We obtained this data from prior work [37]. The intercoder reliability was 0.996 for *perceived error start*, 0.907 for *user reaction start*, and 0.975 for *user reaction end*. The grocery and sorting tasks had participants "program" the robot (did not affect the robot's behavior as it was operated via Wizard of Oz) and then observe the robot's task execution. All errors were pre-programmed into the robot's execution. The dataset includes 28 participants (17 female and 11 male) with an age range of 18 to 39 ($M = 23.72, SD = 4.43$). Participants were experienced with technology ($M = 4.30, SD = 0.76$ on a 5-point scale with one as beginner) and less experienced with robots ($M = 2.65, SD = 1.11$).

4.4 Dataset Analysis

We analyze the dataset to understand what social signals are exhibited in response to errors across these diverse interaction contexts.

4.4.1 Metrics. The values calculated below were derived from the ground truth annotations of the dataset videos (performed by two independent coders) and AU intensity outputs from OpenFace.

- **Human Reaction Time** (seconds). This measure computes how quickly a participant reacted to an error; the difference between *user reaction start* and *perceived error start*.
- **Human Reaction Duration** (seconds). This metric is the length of visible reaction to an error, calculated as the difference between *user reaction start* and *user reaction end*.
- **Percent Visible Error Reaction**. This measure quantifies the fraction of the error instances where participants reacted over the number of error instances.
- **Significant AU Predictors**. Calculated using the Welch's t-test, this metric quantifies which AUs' intensities collected were significant in determining whether a timestep was an error or non-error one. See Appendix for detailed analyses.[3]

The dataset is comprised of 126 error instances of which 79.53% contained visible reactions to errors. Participants' average *human reaction time* to the errors was 3.00s ($SD = 3.30$) and *human reaction duration* was 7.28s ($SD = 6.20$). Looking closer at the specific AUs exhibited during the interactions, all AUs outputted by OpenFace except AU_5 (upper lid raiser) and AU_25 (lips part) are *significant AU predictors*. The error timesteps had higher intensities for AU_6 (cheek raiser), AU_7 (lid tightener), AU_10 (upper lid raiser), AU_12 (lip corner puller), AU_14 (dimpler), AU_15 (lip corner depressor), AU_20 (lip stretcher), and AU_45 (blink) than non-error ones. The error timesteps had lower intensities for AU_1 (inner brow raise),

[3]https://intuitivecomputing.github.io/publications/2023-hri-stiber-supp.pdf

AU_2 (outer brow raiser), AU_17 (chin raiser), AU_23 (lip tightener), and AU_26 (jaw drop) than the non-error ones.

4.4.2 HRC-A. Specifically for HRC-A, we see that the *percent visible error reaction* was 74.0% and that 83.33% of the 12 participants reacted to errors. Their average *human reaction time* was 4.76s (SD = 4.16) ranging from -3.33s to 11.33s. The negative reaction times are due to participants who reacted before *perceived error start*, which is possible because these errors caused gradual deviations in the robot's trajectory. The *human reaction duration* was 3.63s (SD = 2.87) with nine *significant AU predictors*: AU_1, AU_4, AU_9, AU_14, AU_15, AU_20, AU_23, AU_25, and AU_45.

4.4.3 HRC-C. In the HRC-C scenario, 91% of the participants reacted to errors of which the *percent visible error reaction* was 73% of the error instances. The *human reaction time*, on average, was 3.98s (SD = 4.13) and the average *human reaction duration* was 6.31s, SD = 6.84. There were 16 *significant AU predictors*, all but AU_15.

When separating the data into the two study conditions, there is a statistically significant difference between participants' reaction times during the verbal ($M = 5.49s, SD = 4.08$) and textual condition ($M = 2.47s, SD = 3.67$), $t(45.47) = 2.69, p = .010$. There was, however, no difference in reaction duration between the verbal ($M = 4.92s, SD = 8.49$) and textual conditions ($M = 7.49s, SD = 4.42$), $t(35.14) = -1.29, p = .20$. In addition, analyzing the difference in reactions to errors during the two conditions revealed 11 AUs are significant (AU_2, AU_5, AU_6, AU_7, AU_9, AU_10, AU_12, AU_14, AU_17, AU_25, and AU_26).

4.4.4 PbD. The *percent visible error reaction* was 94.73% and the participants' *human reaction time* to errors was 0.24s (SD = 1.64) and reaction duration was 10.94s (SD = 9.01). For the reactions, we found that there were 16 *significant AU predictors* (all but AU_4).

5 AN INTERACTIVE ERROR-AWARE ROBOT SYSTEM: AN IMPLEMENTATION EXAMPLE

From dataset analysis, we showed that social signals in response to robot errors are fairly widespread and prompt across errors, tasks, and scenarios. Therefore, we postulate that social signals are a pivotal input source for flexible, timely error detection in support of successful error management. In this section, we illustrate how natural social reactions to robot errors may be used for error detection in HRI. We first present computational modeling of people's AUs for error detection and classification of interaction context. We then describe an example implementation of an interactive robot system (Figure 4)—built on Microsoft's Platform for Situation Intelligence (\psi) [6]—that incorporates all three portions of our proposed error-aware framework: explicit input, implicit domain-specific input, and implicit social signal input. Currently, the system is configured for the study described in Section 4.1, but it can be modified to accommodate other HRI tasks.

5.1 Modeling Action Units for Error Detection

In modeling AUs for error detection, we explored two models; both models were informed by prior work [37] that investigated how social signals in human-robot interactions might be modeled to detect robot errors. The first model consisted of two parts; the first part was a binary classifier, which classified each AU timestep as an error or no-error timestep, whereas the second part was a sliding window that collated classified timesteps to determine if an error indeed happened and when. The outputs were *detected error timestep* (current timestep of the recent error detected) and *estimated error start* (earliest timestep for current detected error that is classified as an error). Different from prior work [37], we chose to use a deeper and larger neural network for classification (input: 17, hidden: 64, 128 and 64, output: 2). The first model was trained on the entirety of the dataset (three scenarios) and evaluated using leave-one-out cross validation. The second model was structurally the same as the first one, except that it was trained only on *PbD* data and tested on the data obtained from the other two scenarios.

The purpose for employing these two models is two-fold: (1) understanding error detection using AUs—if data quantity impacts performance—via model comparison and (2) evaluating the potential generalizability of our approach across tasks and scenarios.

5.1.1 Performance Measures. We used the following metrics to assess the model performance in detecting robot errors:

- **Algorithmic Detection Delay** (seconds). This metric measures the model's delay in error detection; the mean difference between the *detected error timestep* and *perceived error start*.
- **Reaction Time Difference** (seconds). This metric represents time difference between the error detection and the user's response to the error; the average difference between *algorithmic error detection* and *user reaction start*.
- **Internal Decision Delay** (seconds). This metric computes the average difference between the algorithm's *detected error timestep* and *estimated error start*.
- **False Positive Rate**. This quantifies the average number of false positives detected per trial. A false positive is defined as when the algorithm's detected error timesteps do not coincide with the coded participant's reaction to the error.
- **False Negative Percentage**. This calculates the fraction of error instances that were not detected by the model.

In addition to evaluating model performance, we used the data from *HRC-A* to examine the potential gain in time to detect possible robot errors using social signals. Particularly, we used the following measure to quantify the potential:

- **Social Signal Potential** (seconds). This metric represents the difference between *user reaction start* and when the participant verbally reported the error, *explicit error reporting reaction time* (seconds)—defined as the difference between the verbal error reporting and *perceived error start*.

5.1.2 Model Performance Analysis. We compared model 1's (more training data) performance with model 2's (less training data) to explore the influence of data size on error detection. For evaluating our approach's generalizability, we contrasted model 2's leave-one-out cross validation performance on *PbD* with its test performance on the other two HRC scenarios. Table 2 details the two models' performances for detecting errors using our dataset.

Data Size Influence. Comparing error detection performance between the two models, with respect to *PbD*, shows that model 1 is about the same in detection (model 1: 4.31 ± 4.11, model 2: 4.27 ± 3.06); had a 0.27 higher *false positive rate* (model 1: 0.75 ± 0.75,

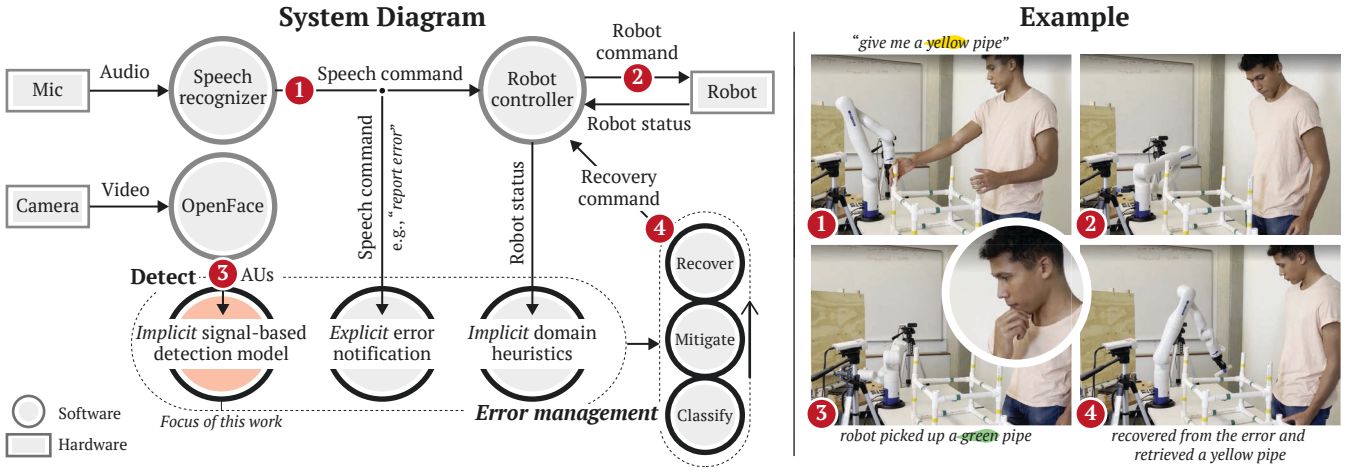

Figure 4: Example autonomous error-aware robotic system diagram. Left diagram illustrates the components of the example system and flow of information. (1) The system records speech commands, executes speech recognition, and sends the command to the robot controller. (2) The robot controller tells the robot to execute a sequence. (3) The perception component records video, processes it through OpenFace, and sends the AUs to the error management component. An error would be detected using a trained model, verbally mitigated using acknowledgement, and repaired via a command to the robot controller. (4) The robot controller receives the repair command and sends a command to the robot to run a pre-programmed repair sequence.

Table 2: Summary of the performance of the two models on the dataset. The grayed rows in the table are the breakdown of the dataset by data collection scenario. Model 1 was trained on the complete dataset and evaluated using leave-one-out cross validation. Model 2 was trained on PbD, evaluated on the two HRC scenarios, and leave-one-out cross validated on *PbD*.

		Algorithmic Detection Delay	Reaction Time Difference	Internal Decision Delay	False Positive Rate	False Negative Percentage
Model 1	**Complete Dataset**	5.54 ± 4.91	2.88 ± 4.09	2.61 ± 0.85	1.10 ± 1.07	30.70%
	HRC-A	5.00 ± 6.46	-1.61 ± 4.85	2.72 ± 0.71	1.17 ± 1.47	61.11%
	HRC-C	6.60 ± 5.10	2.72 ± 3.23	2.71 ± 0.65	1.36 ± 1.15	25.00%
	PbD	4.31 ± 4.11	4.04 ± 4.35	2.49 ± 1.09	0.75 ± 0.75	5.55%
Model 2	**HRC**	6.78 ± 4.82	3.06 ± 5.07	2.51 ± 0.48	1.21 ± 1.12	45.45%
	HRC-A	6.17 ± 4.80	2.13 ± 4.17	2.60 ± 0.52	1.90 ± 1.60	44.44%
	HRC-C	7.00 ± 4.90	3.36 ± 5.39	2.48 ± 0.48	0.96 ± 0.79	45.83%
	PbD (Cross Validation)	4.27 ± 3.06	4.02 ± 3.10	2.87 ± 0.51	0.48 ± 0.70	28.95%

model 2: 0.48 ± 0.70); and was 23.40 lower in *false negative percentage* (model 1: 5.55%, model 2: 28.95%) than model 2. For detection performance on *HRC-A*, when compared to model 2, model 1 was about 1.17s faster (model 1: 5.00 ± 6.46, model 2: 6.17 ± 4.80); had a 0.73 lower *false positive rate* (model 1: 1.17 ± 1.47, model 2: 1.90 ± 1.60); and was 16.67 higher in *false negative percentage* (model 1: 61.11%, model 2: 44.44%). Additionally, for *HRC-C*, when compared to model 2, model 1's performance was quite similar in detection (model 1: 6.60 ± 5.10, model 2: 7.00 ± 4.90); had a 0.4 higher *false positive rate* (model 1: 1.36 ± 1.15, model 2: 0.96 ± 0.79); and was 20.83 lower in *false negative percentage* (model 1: 25.00%, model 2: 45.83%).

Generalizability. Model 2's performance on *HRC-A*, when compared to *PbD*, was 1.90s slower in detection (note the *human reaction time* was 4.52s slower); 1.42 greater *false positive rate*; and 15.59 higher in *false negative percentage*. When comparing its performance on *HRC-C* with *PbD*, we show that error detection was 2.73s slower (*human reaction time* was 3.74s slower); had a 0.48 greater *false positive rate*; and 16.88 higher in *false negative percentage*.

5.1.3 Social Signal Potential Analysis. Participants' average *explicit error reporting reaction time* was 12.41s (*SD* = 3.86) after the *perceived error start*, though it ranged from 7.33s to 25s. On the other hand, the *social signal potential* was 7.56s (*SD* = 4.71), ranging from 0.67s to 4.71s.

5.1.4 Integration with Error Management. Since modeling social signals for error detection proved to be feasible, our example robot system currently includes a simplified version of error management (with a focus on detection): error detection, mitigation through verbal acknowledgment, and pre-programmed recovery. It implements the three layers of explicit and implicit error input. We note that the acknowledgment strategy can be swapped out for code, for example, that automatically creates explanations or apologies.

Explicit Error Detection. The explicit indicator is the verbal command *"Report error"* spoken by the participant. This reporting can be done anytime during the interaction. Once reported, the robot acknowledges the error, sends a message to the robot controller that an error occurred, and executes a recovery sequence.

Implicit Error Detection. Implicit error detection is a combination of domain-specific and social signal input. The domain-specific inputs for the system are if the robot is moving and command count. They are treated as filtering for automatic detection using the implicit social signal (AU) input. The AUs are fed into an error detection algorithm, same as model 2 in Section 5.1.2, which can be easily swapped for another. If the algorithm detects an error, it can only report it if the robot presently is moving, as a robot can only make an error when it is moving for our use case, and the command count is above a threshold. The current domain-specific inputs used are tailored for our particular task and should be changed for others. The reason why we used it this way is that AUs as input is a sometimes ambiguous information source, reactions do not discern between positive or negative (*i.e.,* errors) robot actions causing false positives [37]. In addition, if an error is detected using AUs, the robot queries the participant on whether an error actually happened as an additional false positive screening mechanism. An affirmative response by the participant causes the robot to acknowledge the error, notify the robot controller that an error occurred, and execute a pre-programmed recovery sequence. The system's automatic error detection using AUs can also be turned off so that there is only explicit error detection, done for the study described in Section 4.1.

5.2 Error Context Classification

Another use for social signals, other than error detection, is error context classification. As shown in Section 4.2, the *HRC-C* dataset contained two conditions in which the same errors occurred: social (verbal) and non-social (textual). In addition, through behavioral analysis of the same dataset, there was a statistically significant difference in AU intensities between the reactions to the errors for the two error contexts, prompting us to explore the possibility of using AUs to classify error context.

To this end, we developed an algorithm that consisted of a three-layer binary classifier (input: 17, hidden: 64, 128 and 64, output: 2), classifying one timestep at a time, and a function that combines a series of classifications for a single output. The input was a window of timesteps of 17 AU intensities. The output of the algorithm is a single classification of whether the error occurred in the social or non-social context. The model was trained on the AU intensities for the ground truth marked error timesteps from *HRC-C*. We evaluated the algorithm using leave-one-out cross validation. The algorithm was fed in an error detection window (timesteps between *estimated error start* and *detected error timestep*) outputted from the machine learning algorithm described in Section 5.1 for *HRC-C*. The algorithm's classification accuracy was 92.31%. We note that this algorithm can be run in real-time and integrated into this system.

5.3 Other System Components

The example system (shown in Figure 4) has two other integrated components besides error management (described in Section 5.1.4)—(1) a robot controller to allow it to autonomously run the robot and respond to voice commands and (2) a perception system that takes and processes sensor input (*e.g.,* extracting AUs) for robot operations and necessary error management.

5.3.1 Robot Controller. Much of the implementation in this component is robot and task dependent: in our case, Kinova Gen3 and assembly task. Our system enables the robot to respond to verbal commands, input for the robot controller. Upon a valid command, the system sends a message to the controller to run the corresponding action. Once an action finishes executing, it messages back to the system that the robot is no longer moving and readies for the next command. Also, the controller can pause and stop the robot's execution of an action. The error management component uses this mechanism to interrupt the robot's current action for necessary mitigation when an error is detected.

5.3.2 Perception System. The perception system takes and synchronizes input from two cameras and a microphone in real-time using \psi. Additionally, it sends verbal commands to the robot controller and processes the video feed through OpenFace to extract AUs. Once the AUs per timestep are calculated, this component sends the AUs, as well as whether the robot is moving (received from the robot controller), to the error management component.

6 DISCUSSION

In this work, we propose an error-aware framework (including social signals as an implicit input) to provide error management techniques the flexibility to address unexpected errors in various interaction scenarios. To demonstrate the utility of social signals, we shared, analyzed, and modelled a natural-response-to-error dataset showing that AUs are prevalent and have the potential to be used for error detection across tasks, errors, and scenarios. We additionally showed that modeling social signals has the potential to detect errors before a user's explicit error reporting.

6.1 Social Signals as Error-Aware Input

Our dataset shows that regardless of the scenario, error, and task, people exhibit social signals to unexpected robot errors during physical-based human-robot interaction. About 80% of the error instances had visible responses with a response time of 3.00s. Even during active collaboration (HRC-A), where participants were also working and not necessarily tracking the robot's execution other than to retrieve an object the robot had placed, participants reacted to the robot's errors promptly ($M = 4.18s, SD = 4.13$). We found that participants often checked the instructions before reacting contributing to the reaction time.

However, when looking at the three scenarios, the natural responses elicited from errors were different depending on the interaction contexts, according to the significant AU predictors (Section 4.4). The response diversity is especially exemplified by the two interaction conditions of *HRC-C*: verbal (social) and textual (non-social). Despite experiencing the same error, participants' AU reactions to it were distinctive between the two contexts: 11 AUs were statistically significant in distinguishing between the two. In Section 5.2, we further showed that it is possible to classify the context in which the error occurred given a window of "error" AUs.

6.1.1 Data Size Influence on Error Detection. In spite of the divergent human responses to errors, use of social signals as input for error detection results in timely detection. Using a larger set of responses as a training set (Model 1) appears promising to significantly reduce the number of false negatives, especially in reference to *HRC-C* and *PbD* (*HRC-C*: decrease of 20.83%, *PbD*: decrease of 23.40%). However, a smaller dataset (Model 2) does seem to have relatively comparable performance with respect to detection timeliness, within 1.24s for across the entire dataset. Increasing the data size improves reliability, but the detection timeliness is a function of the overall system design and not simply the modeling algorithm.

6.2 Flexible Error-Aware Framework Potential

Adding social signals as another implicit input in the error-aware framework provides timeliness and flexibility in error detection.

6.2.1 Generalizability of Social Signal Modeling for Error Detection. Our results showed that it is possible to detect errors in one scenario (HRC) using a social signal model trained on a different scenario (PbD). Model 2 was trained on *PbD*; when compared to the cross validation performance of detection delay on PbD, it showed a longer delay on *HRC-A* and *HRC-C*. The most notable difference between cross-validated performance on *PbD* and test performance on the HRC scenario was that the model had a larger false negative percentage under domain shift. Overall, our results are similar to that of a prior work [37], also seeking to examine the potential generalizability of modeling social signals for error detection. The challenge faced by domain shift in social signal-based error detection, however, may be managed by explicit indicators and implicit domain-specific input; together, the three layers of input promise to afford reliable and flexible error detection.

6.2.2 Error Detection Timeliness. Social signals also have the potential to improve the timeliness of error detection following our error-aware framework. As illustrated in Section 5.1.3, reactions to errors are 7.56s faster than having participants manually reporting errors. Depending on the error, participants could in fact react before *perceived error start* if the error developed slowly.

If we were to look at the performance of model 1 (as defined in Section 5.1) on *HRC-A* data, we were able to detect error 5.80s (SD = 4.30) faster using social signals than the explicit response. For model 2, the error detection was 10.56s (SD = 13.08) faster than *explicit error reporting reaction time*. However, it is important to note that not all participants visually reacted to errors so maintaining the other two input streams is important.

6.3 Limitations

One limitation for using social signals as input in an error-aware framework is that such use requires humans to pay attention to the robot and task, allowing them to reflexively respond in a timely manner. Also, factors that affect the speed and observability of the reaction could include task difficulty in human-robot collaboration, amount that a person values robot performance and task outcomes, and a person's instinctive reaction mode (could be no response).

6.3.1 Dataset. Although our dataset provides an illustration of AU responses to errors in a diverse array of interactions, it does not necessarily contribute a comprehensive image of all reactions, task types, and error types. Currently, the dataset only contains AUs, but there are other ways in which humans naturally respond to errors (*e.g.*, verbal, gaze, head movements) [27, 38]. In fact, for many of the *HRC* error instances, gaze would be an insightful indicator as many participants seemed to follow a similar pattern post-error. In addition, for the data collected in *HRC-C* scenario, it is missing some small fraction of the AU responses to errors. The loss in AUs corresponds to when the robot passed in front of the camera, obscuring the view of a participant's face. Another limitation is that the tasks used, while representative of many interaction modes in human-robot interaction, were all highly structured interactions, where potentially domain-specific task tracking could detect errors with high accuracy. Though, it is important to note that using domain-specific information would not be flexible enough to "cross-detect" errors in the other tasks. In addition, because the dataset was annotated using coders watching the video, the error annotations might not reflect subtler reactions that are not visible to the eye but are quantifiable and detectable using OpenFace. Therefore, the reaction times and durations to errors could be dependent on reaction magnitude.

6.3.2 Interactive Error-Aware Robot System. As with many robot systems designed to interact with people closely, our error-aware system is highly dependent on appropriate camera placement for the task. We tried to combat this issue by having two cameras as input, but we found out placing the cameras farther apart and at a greater angle reduced the continuity of AU detection between the two cameras. In addition, we currently use the domain-specific input only as a technique to filter the false positives generated from the AU modeling machine learning. However, we could improve error detection performance by modeling that input source and fusing it with the social signal model for error detection.

6.4 Implications for HRI and Future Work

Our exploration showed that including social signals as input in the error-aware framework potentially improves the flexibility and timeliness of error detection. We are scratching the surface of social signal-based detection. Future implementation of the framework may consider other behavioral channels and finer grained information from facial expressions or other social signals. In addition, future research should assess the benefits of the error-aware framework on human-robot teaming in terms of trust and willingness to work with the robot after an error has been managed. We should also consider what it means for a system that employs the error-aware framework to be successful in terms of the collaboration and with what metrics success should be determined. For modeling social signals for error detection in the physical human-robot interaction domain, we should develop alternative ways to model AUs in response to robot errors that involve treating them temporally as opposed to within isolated timesteps. Research should also explore the use of social signals as input for error classification—whether that be error contexts, severity, or type—providing more information to help with error mitigation and recovery.

ACKNOWLEDGMENT

This work was supported by the Johns Hopkins Institute for Assured Autonomy and the National Science Foundation award #2143704.

REFERENCES

[1] Henny Admoni, Thomas Weng, Bradley Hayes, and Brian Scassellati. 2016. Robot nonverbal behavior improves task performance in difficult collaborations. In *2016 11th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*.

[2] Reuben M. Aronson and Henny Admoni. 2018. Gaze for Error Detection During Human-Robot Shared Manipulation. In *RSS Workshop: Towards a Framework for Joint Action*.

[3] Tadas Baltrušaitis, Peter Robinson, and Louis-Philippe Morency. 2016. Openface: an open source facial behavior analysis toolkit. In *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*.

[4] Atef Ben-Youssef, Chloé Clavel, and Slim Essid. 2019. Early detection of user engagement breakdown in spontaneous human-humanoid interaction. *IEEE Transactions on Affective Computing* (2019).

[5] Ulas Karli Berk, Shiye Cao, and Chien-Ming Huang. 2023. "What If It Is Wrong": Effects of Power Dynamics and Trust Repair Strategy on Trust and Compliance in HRI. In *Proceedings of the ACM/IEEE International Conference on Human-Robot Interaction*.

[6] Dan Bohus, Sean Andrist, Ashley Feniello, Nick Saw, Mihai Jalobeanu, Patrick Sweeney, Anne Loomis Thompson, and Eric Horvitz. 2021. Platform for Situated Intelligence. arXiv:2103.15975 [cs.AI]

[7] Riccardo Bovo, Nicola Binetti, Duncan P. Brumby, and Simon Julier. 2020. Detecting Errors in Pick and Place Procedures. In *ACM Conference on Intelligent User Interfaces*.

[8] Dito Eka Cahya, Rahul Ramakrishnan, and Manuel Giuliani. 2019. Static and Temporal Differences in Social Signals Between Error-Free and Erroneous Situations in Human-Robot Collaboration. In *International Conference on Social Robotics*. 189–199.

[9] Greg Chance, Antonella Camilleri, Benjamin Winstone, Praminda Caleb-Solly, and Sanja Dogramadzi. 2016. An assistive robot to support dressing - strategies for planning and error handling. In *2016 6th IEEE International Conference on Biomedical Robotics and Biomechatronics (BioRob)*.

[10] Filipa Correia, Carla Guerra, Samuel Mascarenhas, Francisco S Melo, and Ana Paiva. 2018. Exploring the impact of fault justification in human-robot trust. In *Proceedings of the 17th international conference on autonomous agents and multiagent systems*.

[11] Timon Gehr, Matthew Mirman, Dana Drachsler-Cohen, Petar Tsankov, Swarat Chaudhuri, and Martin Vechev. 2018. Ai2: Safety and robustness certification of neural networks with abstract interpretation. In *2018 IEEE symposium on security and privacy (SP)*.

[12] Romi Gideoni, Shanee Honig, and Tal Oron-Gilad. 2022. Is it personal? The impact of personally relevant robotic failures (PeRFs) on humans' trust, likeability, and willingness to use the robot. *arXiv preprint arXiv:2201.05322* (2022).

[13] Manuel Giuliani, Nicole Mirnig, Gerald Stollnberger, Susanne Stadler, Roland Buchner, and Manfred Tscheligi. 2015. Systematic Analysis of Video Data from Different Human-Robot Interaction Studies: A Categorisation of Social Signals During Error Situations. *Frontiers in Psychology* 6 (2015).

[14] Dylan F Glas, Satoru Satake, Florent Ferreri, Takayuki Kanda, Norihiro Hagita, and Hiroshi Ishiguro. 2012. The network robot system: enabling social human-robot interaction in public spaces. *Journal of Human-Robot Interaction* 1, 2 (2012), 5–32.

[15] Svyatoslav Guznov, J Lyons, Marc Pfahler, A Heironimus, Montana Woolley, Jeremy Friedman, and A Neimeier. 2020. Robot transparency and team orientation effects on human–robot teaming. *International Journal of Human–Computer Interaction* (2020).

[16] Robert L Helmreich. 2000. On error management: lessons from aviation. *Bmj* 320, 7237 (2000).

[17] Robert L Helmreich, Ashleigh C Merritt, and John A Wilhelm. 2017. The evolution of crew resource management training in commercial aviation. In *Human Error in Aviation*.

[18] Shanee Honig and Tal Oron-Gilad. 2018. Understanding and Resolving Failures in Human-Robot Interaction: Literature Review and Model Development. *Frontiers in Psychology* (June 2018).

[19] Shanee Honig and Tal Oron-Gilad. 2021. Expect the Unexpected: Leveraging the Human-Robot Ecosystem to Handle Unexpected Robot Failures. *Frontiers in Robotics and AI* 8 (2021).

[20] Dimosthenis Kontogiorgos, Andre Pereira, Boran Sahindal, Sanne van Waveren, and Joakim Gustafson. 2020. Behavioural Responses to Robot Conversational Failures. In *Proceedings of the 2020 ACM/IEEE International Conference on Human-Robot Interaction*. Association for Computing Machinery, 53–62.

[21] Dimosthenis Kontogiorgos, Minh Tran, Joakim Gustafson, and Mohammad Soleymani. 2021. A Systematic Cross-Corpus Analysis of Human Reactions to Robot Conversational Failures. In *Proceedings of the 2021 International Conference on Multimodal Interaction*.

[22] Dimosthenis Kontogiorgos, Sanne van Waveren, Olle Wallberg, Andre Pereira, Iolanda Leite, and Joakim Gustafson. 2020. Embodiment Effects in Interactions with Failing Robots. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery.

[23] Terry K Koo and Mae Y Li. 2016. A guideline of selecting and reporting intraclass correlation coefficients for reliability research. *Journal of chiropractic medicine* 15, 2 (2016), 155–163.

[24] John Lee and Neville Moray. 1992. Trust, control strategies and allocation of function in human-machine systems. *Ergonomics* 35, 10 (1992).

[25] Zongyu Li, Kay Hutchinson, and Homa Alemzadeh. 2022. Runtime Detection of Executional Errors in Robot-Assisted Surgery. In *IEEE International Conference on Robotics and Automation (ICRA)*.

[26] Liang Ma and Chen Wang. 2022. Safety Issues in Human-Machine Collaboration and Possible Countermeasures. In *International Conference on Human-Computer Interaction*.

[27] Nicole Mirnig, Manuel Giuliani, Gerald Stollnberger, Susanne Stadler, Roland Buchner, and Manfred Tscheligi. 2015. Impact of Robot Actions on Social Signals and Reaction Times in HRI Error Situations. In *International Conference on Social Robotics*. Springer, 461–471.

[28] Nicole Mirnig, Gerald Stollnberger, Markus Miksch, Susanne Stadler, Manuel Giuliani, and Manfred Tscheligi. 2017. To Err Is Robot: How Humans Assess and Act toward an Erroneous Social Robot. *Frontiers in Robotics and AI* 4 (2017).

[29] Bonnie M Muir and Neville Moray. 1996. Trust in automation. Part II. Experimental studies of trust and human intervention in a process control simulation. *Ergonomics* 39, 3 (1996).

[30] Illah R Nourbakhsh, Clayton Kunz, and Thomas Willeke. 2003. The mobot museum robot installations: A five year experiment. In *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)*, Vol. 4.

[31] James O'Keeffe, Danesh Tarapore, Alan G Millard, and Jon Timmis. 2018. Adaptive online fault diagnosis in autonomous robot swarms. *Frontiers in Robotics and AI* (2018).

[32] Daehyung Park, Yuuna Hoshi, and Charles C Kemp. 2018. A multimodal anomaly detector for robot-assisted feeding using an lstm-based variational autoencoder. *IEEE Robotics and Automation Letters* 3, 3 (2018), 1544–1551.

[33] Alessandra Rossi, Kerstin Dautenhahn, Kheng Lee Koay, and Michael L. Walters. 2017. Human Perceptions of the Severity of Domestic Robot Errors. In *Social Robotics*. Springer International Publishing, 647–656.

[34] Maha Salem, Gabriella Lakatos, Farshid Amirabdollahian, and Kerstin Dautenhahn. 2015. Would You Trust a (Faulty) Robot? Effects of Error, Task Type and Personality on Human-Robot Cooperation and Trust. In *Proceedings of the Tenth Annual ACM/IEEE International Conference on Human-Robot Interaction*. 141–148.

[35] Kristin E. Schaefer, Jessie Y. C. Chen, James L. Szalma, and P. A. Hancock. 2016. A Meta-Analysis of Factors Influencing the Development of Trust in Automation: Implications for Understanding Autonomy in Future Systems. *Human Factors* 58, 3 (2016).

[36] Maia Stiber and Chien-Ming Huang. 2020. Not All Errors Are Created Equal: Exploring Human Responses to Robot Errors with Varying Severity. In *Companion Publication of the 2020 International Conference on Multimodal Interaction*. 97–101.

[37] Maia Stiber, Russell H. Taylor, and Chien-Ming Huang. 2022. Modeling Human Response to Robot Errors for Timely Error Detection. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*.

[38] Pauline Trung, Manuel Giuliani, Markus Miksch, Gerald Stollnberger, Susanne Stadler, Nicole Mirnig, and Manfred Tscheligi. 2017. Head and Shoulders: Automatic Error Detection in Human-Robot Interaction. In *ACM International Conference on Multimodal Interaction*.

[39] Jason R Wilson, Phyo Thuta Aung, and Isabelle Boucher. 2021. Enabling a Social Robot to Process Social Cues to Detect when to Help a User. *arXiv preprint arXiv:2110.11075* (2021).

[40] Hiroyuki Yasuda and Mitsuharu Matsumoto. 2013. Psychological impact on human when a robot makes mistakes. In *2013 IEEE/SICE International Symposium on System Integration*.