A Tale of Two Suggestions: Action and Diagnosis Recommendations for Responding to Robot Failure

Siddhartha Banerjee[†], Matthew Gombolay[†], and Sonia Chernova[†]

Abstract— **Robots operating without close human supervision might need to rely on a remote call center of operators for assistance in the event of a failure. In this work, we investigate the effects of providing decision support through diagnosis suggestions, as feedback, and action recommendations, as feedforward, to the human operators. We conduct a 10-condition user study involving 200 participants on Amazon Mechanical Turk to evaluate the effects of providing noisy and noise-free diagnosis suggestions and/or action recommendations to operators. We find that although action recommendations (feedforward) have a greater effect on successful error resolution than diagnosis information (feedback), the feedback likely helps ameliorate the deleterious effects of noise. Therefore, we find that error recovery interfaces should display both diagnosis and action recommendations for maximum effectiveness.**

I. Introduction

As autonomy improves, robots are increasingly operating without close expert supervision. Robots making hospital deliveries, taking inventory at grocery stores, or organizing a warehouse operate largely independently but occasionally encounter an error. In such cases, it is unlikely that a local robotics expert will be available, and the robot will instead rely on remote *call center of operators* for assistance [1]. The operators are likely to have available an interface that is designed to help resolve the robots' failure. This work investigates the effects of decision support in such an interface.

Prevalent guidelines for designing the user experience (UX) for a failure resolution system suggest that in the face of failure, operators should be provided with *feedback* information—to be made better aware of the failure state—and with *feedforward* information—to better enable decision-making [2]. However, it is unclear from the prior literature on UX [3], [4] which of feedback or feedforward information could be more useful to robot failure resolution.

The importance of knowing the relative benefits of feedforward and feedback is highlighted by recent work, which has recommended that interfaces designed for error recovery not overwhelm the information processing capabilities of operators, lest operators themselves make mistakes [5]. Despite their usefulness, both feedback, such as through automated diagnoses, and feedforward, such as through action recommendations, can potentially add too much information to an interface. It is unclear from prior work if such is the case.

In addition to receiving too much information, an operator's information processing capabilities can be taxed in dealing with inaccuracies with decision aids: for instance, feedback provided through automated fault diagnosis systems or feedforward provided through action recommendation models can be imperfect, resulting in robots deployed with inaccurate decision aids. Prior work has shown that in the face of incorrect suggestions, humans are prone to both follow the recommendations [6] and to ignore them [7]. It is therefore important to determine how inaccurate decision aids might affect the resolution of robot failures.

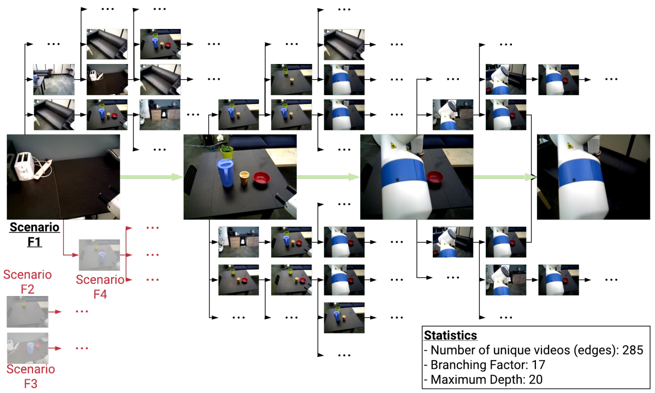


Fig. 1: Storyboard for interactive failure recovery. Participants start in one of four failure scenarios and attempt to resolve the error by selecting one of 17 actions, which the robot then executes in an accompanying video. We evaluate the action sequence taken by participants under different interface conditions, and how it compares to the shortest possible error recovery (green arrows)[1].

In this work, we contribute a 10-condition study evaluating the effects of providing noisy and noise-free diagnosis suggestions (feedback) and/or action recommendations (feedforward) as decision aids to humans. We perform our analysis within an interactive user experience, in which users select robot recovery actions in response to observed error states. The interactive experience occurs in the context of a dynamically generated story graph in which nodes are faulty or fault-free robot states. The story nodes are connected by edges corresponding to one of 17 actions, captured as 285 videos of a physical Fetch robot, and which are selected by the participants (Fig. 1). The resulting highly realistic evaluation framework enables us to examine how elements of UX design impact a user's ability to effectively recover from robot errors. Our findings show that although action recommendations (feedforward) have a greater effect on successful error resolution than diagnosis information (feedback), the feedback likely helps ameliorate the deleterious effects of noise. Therefore, we find that error recovery interfaces should display both diagnosis and action recommendations for maximum effectiveness.

This work was supported in part by an Early Career Faculty grant from NASA's Space Technology Research Grants Program and by a NASA Early Career Fellowship (80HQTR19NOA01-19ECF-B1).

[†]Georgia Institute of Technology, Atlanta, USA. Author contact: siddhartha.banerjee@gatech.edu

[1] A video showing the study design is at https://youtu.be/drCHgwkpaqA

II. Related Work

Robust robot execution is difficult to achieve. To address this challenge, prior research has proposed techniques for adjustable autonomy in order to enable humans to assist during difficult tasks [8]. A previous ethnographic study of a robot in various work environments found that even when collocated, humans tasked with intervening on behalf of the robot wanted assistance addressing two fundamental questions, "What's wrong?" and "How do I fix it?" [9]. The finding is consistent with prevalent design guidelines that in the face of failure, operators should be provided with *feedback* information—to be made better aware of the failure state—and with *feedforward* information—to better enable decision-making [2].

Prior work has introduced decision aids to help operators capture high level problem diagnosis [8], [10], assist with action selection or planning [11], [12], [13], or both [14], [15]. Furthermore, failure recovery (i.e., troubleshooting) is often an iterative process during which failure hypotheses and/or recovery actions are pruned through execution of diagnostic test actions [16], [17]. Automated troubleshooting aids are usually equipped with the capability to suggest potential problems and to recommend actions to fix them [18], [14], [15], [19]. However, automated diagnosis or action recommender systems are not perfect [20], [21]. As a result, robots must often be deployed with imperfect decision aids.

Prior work has shown that in the face of errors in robot decision support systems, humans are prone to both *overtrust* its recommendations—following them to their detriment [6]—as well as mistrust its recommendations—ignoring them to their detriment [7]. It is therefore unclear what the effects of imperfect decision support systems might be in the case of robot failure recovery, a scenario not evaluated in the aforementioned studies. We bridge that gap.

Additionally, a recent survey calls into question a naïve recommendation to include both the feedback and feedforward decision aids because the authors find that failure recovery can be a cognitively demanding task for an operator and argue that the interfaces used must not overwhelm the information processing capabilities of a human [5]. They cite the prevailing design knowledge that humans are liable to themselves make errors if they are overwhelmed with too-much-information [4], [2]. Additional research in UX design has found that *feedforward* suggestions in widgets are especially suitable in applications that users might be unfamiliar with [3], but it is unclear whether the finding generalizes to robot error recovery, as well as when noise is present in feedforward output. We aim to identify the tradeoffs, if any, that might exist from providing either types of decision support to remote operators during robot failures.

Finally, while prior work has evaluated the interaction consequences of failure presentation, none evaluate the consequences on the robot's performance. Lee et al. [22] used online surveys to gauge participant evaluations of a robot's service based on the manner of robot communication during a failure; participants were not required to aid the robot based on the information presented to them. Similarly, Brooks et al. [23] introduced two types of decision support—human-support and task-support, which correspond to feedback and feedforward respectively—and found that people's reactions towards the robot were improved as a result of both types. However, their evaluations were conducted using survey responses to hypothetical scenarios and participants were not actually required to supervise a robot based on the information they received. In contrast, we allow people to supervise the execution of a robot during a failure, and we evaluate the robot's recovery outcome as a result of varying types of decision support.

III. Research Questions

In this work, we evaluate the relative benefits of both feedback and feedforward information by evaluating the following decision support for robot failures:

- **Diagnosis-based Suggestion**—the diagnosis of one or more faults (feedback). For example, "The [object] is not visible", (if a perception action fails) or "The robot has collided" (if the base of the robot is unable to move).
- **Action-based Recommendation**—the recommendation of actions to take to resolve the problem (feedforward). For example, "Navigate to [location]" (to perhaps check for a missing object), or "Move the robot back" (assuming the collision is at the front).

Despite several independent systems for diagnosis and recommendation having been developed (Sec. II), their relative benefit when used either independently or together remains unexplored. Furthermore, automated diagnosis or recommendation systems have their own limitations, leading to imperfect performance [20], [21]. As a result, it is important that we understand how the relative benefits of both techniques are affected by their accuracy.

In this work, we study how various types of decision support aids, under varying levels of performance noise, affect the user's ability to effectively recover from errors. Specifically, we formulate the following research questions:

RQ1 *How is a human operator's assistance of a robot affected by Action Recommendations (AX), Diagnosis Suggestions (DX), or both (DXAX)?* We formulate this first question to investigate the types of decision support that might be necessary in a failure resolution UX.

RQ2 *What is the effect of inaccuracies in the decision support provided to human operators?* The aim of this second question is to investigate trends in human operator performance as the reliability of decision support varies.

To answer **RQ1**-**RQ2**, we designed a large scale user study using Amazon Mechanical Turk, in which we varied two factors determining the manner of generating decision aids. The first factor determined the type of suggestions that participants received—no suggestions (BASELINE), Action Recommendations (AX), Diagnosis Suggestions (DX), or both (DXAX). The second factor determined the accuracy of the suggestions at three levels often achieved by fault diagnosis models in prior work [24]—100% accurate, 90% accurate, and 80% accurate. The resulting ten study conditions are enumerated in Table I.

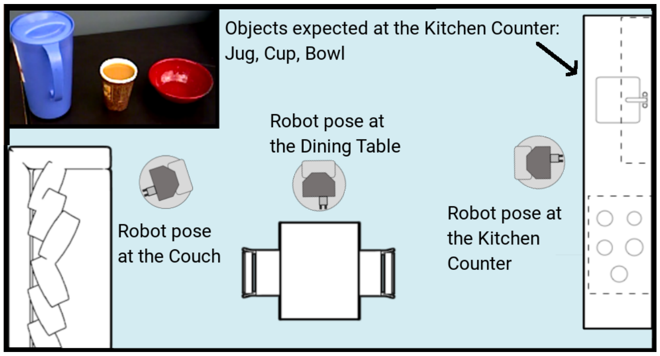

Objects expected at the Kitchen Counter: Jug, Cup, Bowl
Robot pose at the Dining Table
Robot pose at the Couch
Robot pose at the Kitchen Counter

Fig. 2: The robot is in a mock apartment with three locations. It can work with the Jug, Cup, and Bowl (left-to-right in inset).

IV. Domain

We situated our investigation in a mock apartment environment, with a Fetch mobile manipulator performing an object retrieval task (Fig. 2). We chose the apartment environment as one that would be familiar to study participants, and the retrieval task due to its intuitive nature and the diversity of potential errors [25], many of which are also common in other applications (e.g. occlusion of objects) [26].

We conducted our experiment online by simulating the experience of a participant remotely controlling the robot. Participants were told that they would be using a web interface (Fig. 3) to guide the robot through error recovery. In reality, in response to their actions, the interface would display videos that showed the robot executing the target behavior. All videos were pre-recorded for consistency and scalability of the experiment. As shown in Fig. 1, we recorded 285 videos representing a rich storyboard of potential recovery behaviors.

In addition to the video of the robot's action execution, the interface displayed a summary of the task objective and showed a history of participant actions alongside the results of those actions (i.e., success/failure). Using this information, at each step of the experiment participants could (1) indicate their diagnoses of problems with the robot from a set of 11 possible problems, D, and (2) select the next action to take to resolve the problem from a list of 17 possible actions, A.

A. Task Scenarios

The robot's environment consisted of three possible locations the robot could navigate to: *couch*, *dining table*, and *kitchen counter*. Three manipulable objects were present in the environment: *bowl*, *jug*, and *cup*. The robot was able to navigate to locations depending on localization, as well as recognize the three objects or pick up and place them, resulting in 13 possible object-action combinations. To construct an experiment storyboard, we modeled the state as the state of the robot, the objects, and the presence of any of the failures described below, and then applied deterministic action transitions based on predetermined action preconditions (e.g., executing *pickup(obj)* would cause the robot to pick up the *obj* if it was unoccluded in front of the robot).

In each experiment, the robot started in one of four errors, F1–4, described below. The common task objective (i.e., terminal condition of the experiment) was for the robot to be located near the *couch* while holding the *cup*. In the absence of errors, this objective could be achieved by the robot navigating to the *kitchen counter*, picking up the *cup* (Fig. 2), and taking it to the *couch*.

To facilitate our goal of evaluating decision support suggestions for robot failures, we injected an error into each participant trial in order to study the participant's recovery behavior. Each failure scenario, and corresponding start state, represents a type of error commonly encountered in robot task execution [26]:

F1: *Mismatch between design and the environment* where the objects are actually at a location different from the one specified in the nominal task objective. Participants started with a view of an empty *kitchen counter* and needed to find the objects, which were on the *dining table*. Min. recovery steps: 3.

F2: *Non-fatal cause of a failure* where the robot was mislocalized so that navigation commands were remapped, which then triggers a failure while trying to find the objects. E.g. the command "Navigate to [locationA]" sent the robot to [locationB] instead. As a symptom of the remapping, participants started with a view of an empty *dining table*. Min. recovery steps: 4.

F3: *Environment occlusion*, with the *jug* occluding the *cup* on the *kitchen counter*. The scenario also showcases aliased faults, because visually this failure is similar to F1. Min. recovery steps: 4.

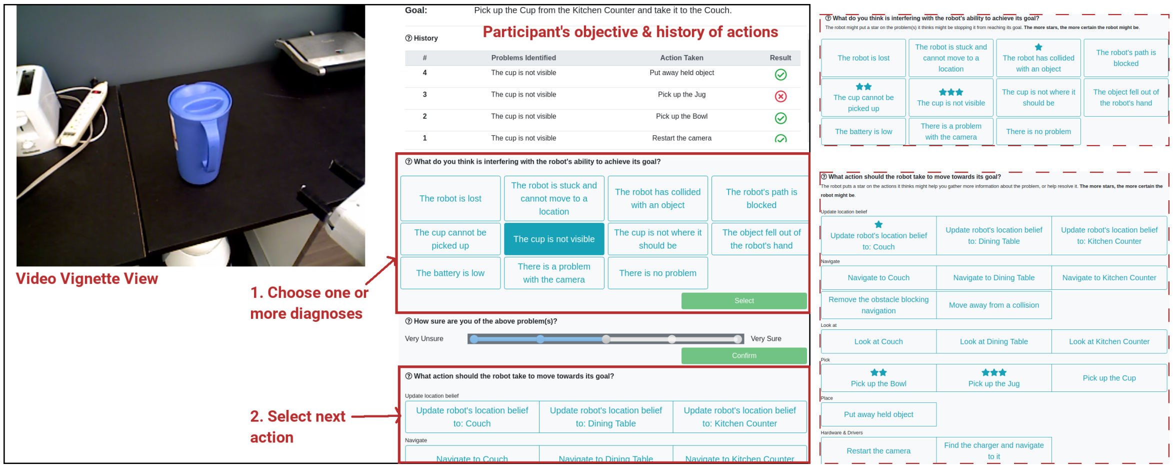
F4: *Multiple concurrent faults*. The task was misspecified (F1), the *jug* occluded the *cup* (F3), and the *bowl* was placed on top of the *cup* requiring it to also be moved out of the way (an additional fault). Min. recovery steps: 7.

B. Suggestions

Depending on the study condition, participants were shown three diagnosis suggestions and/or three action recommendations. Stars were used to recommend diagnosis/actions to participants in the user interface (UI) shown in Fig. 3b. Instructional text in the UI notified participants that the number of stars was a proxy for model certainty. We provided a ranked list of suggestions without numerical values as a result of recommendations from prior work [7].

Diagnoses: There were 11 total diagnoses, such that $D = D_{none} \bigcup D_{fault} \bigcup D_{distractors}$, where D_{none} indicated no fault in the current robot execution, $|D_{fault}| = 4$ corresponded to four failures from the scenarios above (unknown to participants ahead of time), and $|D_{distractors}| = 6$ represented distractor diagnoses that never occurred in the experiments (e.g., *There is a problem with the camera*). Based on the pilot studies, all diagnoses were assigned easy-to-understand labels, e.g., the term "gripper" was substituted with "hand" because the latter is more accessible to the general public. Each robot state was associated with one or more diagnoses in the set $D_{none} \bigcup D_{fault}$ and we used a lookup table to suggest diagnoses to participants (suggestions were padded to three by random sampling of D).

Actions: There were 17 total actions, such that $A = A_{domain} \bigcup A_{distractor}$, where $|A_{domain}| = 13$ corresponded



Goal: Pick up the Cup from the Kitchen Counter and take it to the Couch.

ⓘ History

#	Problems Identified	Action Taken	Result
4	The cup is not visible	Put away held object	✓
3	The cup is not visible	Pick up the Jug	✗
2	The cup is not visible	Pick up the Bowl	✓
1	The cup is not visible	Restart the camera	✓

ⓘ What do you think is interfering with the robot's ability to achieve its goal?

ⓘ How sure are you of the above problem(s)?

ⓘ What action should the robot take to move towards its goal?

The robot might put a star on the problem(s) it thinks might be stopping it from reaching its goal. **The more stars, the more certain the robot might be**.

The robot puts a star on the actions it thinks might help you gather more information about the problem, or help resolve it. **The more stars, the more certain the robot might be**.

Participant's objective & history of actions
Video Vignette View
1. Choose one or more diagnoses
2. Select next action
The robot is lost
The robot is stuck and cannot move to a location
The robot has collided with an object
The robot's path is blocked
The cup cannot be picked up
The cup is not visible
The cup is not where it should be
The object fell out of the robot's hand
The battery is low
There is a problem with the camera
There is no problem
Select
Very Unsure
Very Sure
Confirm
Update location belief
Update robot's location belief to: Couch
Update robot's location belief to: Dining Table
Update robot's location belief to: Kitchen Counter
Navigate
Navigate to Couch
Navigate to Dining Table
Navigate to Kitchen Counter
Remove the obstacle blocking navigation
Move away from a collision
Look at
Look at Couch
Look at Dining Table
Look at Kitchen Counter
Pick
Pick up the Bowl
Pick up the Jug
Pick up the Cup
Place
Put away held object
Hardware & Drivers
Restart the camera
Find the charger and navigate to it

(a) (b)

Fig. 3: (a) The web UI for participants in the BASELINE condition. The red annotations are for illustration purposes only. (b) Examples of starred suggestions for diagnoses (top) and for actions (bottom).

Suggestion Type	Acc: 80%	Acc: 90%	Acc: 100%
No suggestions	-	-	BASELINE*
AX	AX_{80}	AX_{90}	AX_{100}
DX	DX_{80}	DX_{90}	DX_{100}
DX & AX	DXAX_{80}	DXAX_{90}	DXAX_{100}

**Baseline does not provide feedback, so has no associated accuracy*

TABLE I: Study Conditions.

to the 13 object-action combinations defined with the domain (above), and $|A_{distractor}| = 4$ represented distractor actions that did not cause any visible changes in the participant videos (e.g., *Restart the camera*). Participants could select any of the 17 actions at all times. When presenting action recommendations, we included the optimal action (i.e., the action on the trajectory with the least number of actions to the goal) as the highest priority, and then randomly chose from the remaining executable actions at the state (i.e., actions that would succeed) in order to present a total of three recommendations.

Modulating Accuracy: We provided the participant with three incorrect suggestions if the study condition required it. For diagnosis suggestions, we used three random choices among all the diagnoses that were not applicable in the robot state. For action recommendations, we made three random choices among all actions that could successfully execute in the state, taking care to not select the optimal action. In the DXAX conditions, action recommendations were corrupted when diagnosis suggestions were corrupted.

V. Experiment Procedure

We conducted a 4x3 between-subjects fractional factorial experiment, with a total of 10 conditions (Table I).

A. Protocol

We recruited 200 participants through Amazon Mechanical Turk, with 20 participants per condition. The study was designed to take 20 minutes and participants were compensated $4 for their time. After providing basic demographic information, participants were introduced to the robot system through an instructional web page containing an accompanying video (available at https://youtu.be/0jYuxLTKlyM) to familiarize them with the domain. They were then asked five yes/no knowledge review questions to test their understanding of the task. Participant data was discarded if participants failed the review questions more than five times or refreshed the browser during the experiment.

Participants who passed the knowledge review were presented with the UI introduced in Sec. IV and allowed up to 20 actions to assist the robot. Within the 20 actions, participants in the 90% accuracy conditions received inaccurate AX and/or DX suggestions on the 2nd and 12th actions (if they took at least 12 actions), and participants in the 80% accuracy conditions received inaccurate suggestions on the 2nd, 5th, 12th, and 15th actions. Once a participant resolved the error or exhausted their budget of 20 actions, they were directed to a post-study usability questionnaire.

B. Metrics & Hypotheses

We evaluate the following performance metrics:

1. *Failure resolution rate* (**FRR**): The failure scenario is considered resolved if the participant accomplishes the robot's goal within the budget of 20 actions. FRR captures the likelihood of a participant resolving a failure scenario.
2. *Rate of optimal action selection* (**RAX**): Each state has an optimal action that leads to the goal in the shortest number of actions (Sec. IV). RAX examines the propensity of participants to select the optimal action and is a measure of operator reliance on decision support [28].
3. *Rate of correct diagnosis selection* (**RDX**): Each state corresponds to a set of correct fault diagnoses (Sec. IV-B). RDX examines the propensity of participants to select at least one of those diagnoses and is a measure of operator reliance on decision support [28].
4. *Compliance with AX suggestions* (**CAX**): When provided with action recommendations (in the AX or DXAX conditions), CAX captures participants' likelihood of following those suggestions and is a measure of operator compliance with decision support [28].
5. *Compliance with DX suggestions* (**CDX**): When provided with diagnosis suggestions (in the DX or DXAX condi-

Metrics (data type)	**Assumed Model**	**Parameter Priors**	**ROPE**
FRR (binary)	$metric_i \sim Bernoulli(p_i)$ $logit(p_i) = \beta_0 + \mathbf{X_{control,i}}\beta_{control}+$ $\mathbf{X_{condition,i}}\beta_{condition}$	$\beta. \sim \mathcal{N}(0, 10)$	[-0.055, 0.055]
RAX, RDX, CAX, CDX (binary)	$metric_{ij} \sim Bernoulli(p_{ij})$ $logit(p_{ij}) = \beta_0 + \beta_{0,i} + \mathbf{X_{control,i}}\beta_{control}+$ $\mathbf{X_{condition,i}}\beta_{condition} + X_{state_{ij}}\beta_{state}$	$\beta. \sim \mathcal{N}(0, 10)$ $\beta_{0,i} \sim \mathcal{N}(0, \sigma_i)$ $\sigma_i \sim HalfStudent(3, 0, 10)$	[-0.055, 0.055]
SUS (interval from 0–100)	$metric_i \sim SkewNormal(\mu_i, \sigma, \alpha)$ $\mu_i = \beta_0 + \mathbf{X_{control,i}}\beta_{control}+$ $\mathbf{X_{condition,i}}\beta_{condition}$	$\beta. \sim \mathcal{N}(0, 10)$ $\sigma \sim HalfStudent(3, 0, 22)$ $\alpha \sim \mathcal{N}(0, 4)$	[-2.3, 2.3] $(0.1 * SD[metric])$

TABLE II: The assumed Generalized Linear Mixed Models for each of the metrics in the analyses. In the models, i indexes a participant, and j is the jth action taken by participant i. ROPE is set based on recommendations by Kruschke [27].

tions), CDX captures participants' likelihood of following those suggestions and is a measure of operator compliance with decision support [28].

6. *System Usability Scale* (**SUS**): The SUS is a 10-item Likert scale used the measure the usability of the UX [29] and was administered to participants in the post-study questionnaire. In our study, the reliability rating, Cronbach's α, for items in this instrument was 0.94.

The hypotheses associated with each of the above metrics are enumerated in Table III. Each metric is associated with two hypotheses pertaining to each of the two research questions that motivate this work.

C. Bayesian Data Analysis

We draw our conclusions from a Bayesian analysis performed on Generalized Linear Mixed Models over the data. The Bayesian analysis allows us to quantify both the likelihood for the *existence* of an effect as well as a practical estimate of the *significance* of that effect. To perform the analysis, we assume that all our metrics are generated under a structural model with the following explanatory variables:

- $\mathbf{X_{condition,i}} = X_{Type,i} + X_{Acc,i}$: A suggestion type factor ($Type \in \{$BASELINE, AX, DX, DXAX$\}$) and an accuracy factor ($Acc \in \{80\%, 90\%, 100\%\}$). We encode the Type factor as Helmert contrasts and report the effects of AX, DX, and DXAX levels vs. the BASELINE level. We encode the Acc factor as orthogonal polynomials in order to investigate linear or quadratic trends in the effects of the factor. Our model does not include interaction effects between Type and Acc in order to preserve model identifiability.
- $\mathbf{X_{control,i}}$: The demographics of a participant and the failure scenario (F1–4) they were assigned.
- $X_{state,ij}, \beta_{0,i}$ (for RAX, RDX, CAX, & CDX): The state of the robot and a random intercept effect of the participant.

Depending on the nature of a metric's data (i.e. binary, count, etc.), we fit recommended probability distributions [30] using Maximum Likelihood Estimation (MLE). The chosen distribution is the one that had the lowest AIC of fit. The probability distribution and the model we used in the analysis of each metric are listed in Table II.

In order to perform Bayesian analysis, we formulate a null hypothesis of minimal effect based on the nature of the data: this is called the Region of Practical Equivalence (ROPE). For instance, we can set the ROPE to be [-0.055, 0.055] for binary data, which then implies that effects resulting in a likely change of less than 0.055 (within the ROPE interval) are considered insignificant (we cannot reject the null hypothesis). The ROPE for each analysis are in Table II.

We begin the analysis by initializing model parameters with weak priors (e.g. $\mathcal{N}(0, 10)$). We then sample parameters for models that might explain the observed data using Hamiltonian Monte-Carlo sampling [30], [32]. We use 4 chains with a burn-in of 1000 iterations, before sampling for 1000 iterations to get the the posterior distribution of the parameters. We verify the diagnostics of the convergence of the samples using established methods involving Leave-One Out Cross-Validation [33]. Note that the posterior distributions of the parameters imply a posterior distribution on the metrics' values. Our inferences on the effects of interest are performed using the posterior distributions.

Using the guidelines presented in [34], we report:

1. A Probability of Direction [pd], which quantifies the likelihood of the *existence* of an effect.
2. The Median and the 89% High Density Credible Intervals (CI) of effect sizes. Effect sizes are classified from Median estimates using the thresholds in prior work [31].
3. The degree of overlap of the full posterior distribution of the metric with the ROPE. The value is used to reject (or not) the null hypothesis on the *significance* of the effect.

Due to limitations of space, we report only a subset of the data analyzed in this paper. Interested readers can find the complete analysis, including model diagnostics and the effects of non-condition factors at https://bit.ly/2UjPPtE.

VI. Results

Table III summarizes our research hypotheses and key results of the study, which we discuss in detail in this section.

Demographics: Our experiment consisted of 200 participants (age group mode 26–30 years, 36.5% / 63% / 0.5% female/male/unspecified gender). The majority of participants (166/200) interacted with a robot at most three times a year.

Failure Resolution Rate (FRR): Fig. 4a shows the proportion of participants that resolved the fault for a given condition. Across conditions, the FRR ranged from 0.6 (DX$_{90}$) to 1.0 (AX$_{100}$).

On evaluating $\mathbf{H1_{FRR}}$, we find that the resolution rate with action suggestions (AX) compared to BASELINE has a 97.2% [pd] of being positive (Median = 1.89, 89% CI [0.29,

Metric	Hypotheses	Metric better with AX	Metric better with DX	Metric better with DXAX	Metric trend with Acc. is Linear	Metric trend with Acc. is Quadratic
FRR	$\mathbf{H1_{FRR}}$: FRR increases with suggestions than without. $\mathbf{H2_{FRR}}$: FRR increases with suggestion accuracy.	97.2% [pd] ***		96.2% [pd] ***		U-shape 97.0% [pd] *
RAX	$\mathbf{H1_{RAX}}$: RAX increases with suggestions than without. $\mathbf{H2_{RAX}}$: RAX increases with suggestion accuracy.	97.0% [pd] **		99.0% [pd] **		
RDX	$\mathbf{H1_{RDX}}$: RDX increases with suggestions than without. $\mathbf{H2_{RDX}}$: RDX increases with suggestion accuracy.		97.8% [pd] *	98.7% [pd] **	Positive slope 99.3% [pd] *	
CAX	$\mathbf{H1_{CAX}}$: CAX improves with DX suggestions. $\mathbf{H2_{CAX}}$: CAX increases with suggestion accuracy.	N/A‡				
CDX	$\mathbf{H1_{CDX}}$: CDX improves with AX suggestions. $\mathbf{H2_{CDX}}$: CDX increases with suggestion accuracy.		N/A‡		Positive slope 100% [pd] *	
SUS	$\mathbf{H1_{SUS}}$: SUS increases with suggestions than without. $\mathbf{H2_{SUS}}$: SUS increases with suggestion accuracy.	98.9% [pd] n.s.	95.2% [pd] n.s.			

‡ *CAX (CDX) does not apply when AX (DX) is not present.*

TABLE III: Metrics, hypotheses, and the main effects results from the data analysis (Sec. V-C). In the results columns, we report in the table if [pd] >95%. We show an effect size if the overlap in ROPE is <2.5%. Effect sizes are indicated by the asterisks: *** for a large effect (Std.Median >.8), ** for a medium effect (Std.Median >.5). and * for a small effect (Std.Median >.2) [31].

3.37]) and can be considered large (Std.Median = 1.04) and significant (0.73% in ROPE) [ROPE (full)]. We also find that the resolution rate with both suggestions (DXAX) compared to BASELINE has a 96.2% [pd] of being positive (Median = 1.57, 89% CI [0.16, 2.97]) and can be considered large (Std.Median = 0.87) and significant (1.15% in ROPE) [ROPE (full)]. As seen in Fig. 5a, the results indicate that adding action suggestions (AX and DXAX) greatly increases the probability of the participants resolving the robot's faults.

On evaluating $\mathbf{H2_{FRR}}$, we find that the noise level has a quadratic relationship to the probability of fault resolution with a 97.0% [pd] positive effect size (convex-shape) (Median = 0.77, 89% CI [0.12, 1.40]), which can be considered small (Std.Median = 0.42) and significant (1.75% in ROPE) [ROPE (full)]. Therefore, as seen in Fig. 5b, the data suggests that as the accuracy of suggestions increases, there is a U-shaped relationship to participant performance.

Rate of optimal action selection (RAX): Fig. 4b shows the proportion of optimal actions taken by participants in each study condition. Across conditions, the Median RAX ranged from 0.50 (DX_{100}, DX_{80}) to 0.76 (AX_{100}).

On evaluating $\mathbf{H1_{RAX}}$, we find that the optimal action rate with action suggestions (AX) compared to BASELINE has a 97.0% [pd] of being positive (Median = 1.03, 89% CI [0.16, 1.95]) and can be considered medium (Std.Median = 0.57) and significant (1.98% in ROPE) [ROPE (full)]. We also find that the optimal action rate with both suggestions (DXAX) compared to BASELINE has a 99.0% [pd] of being positive (Median = 1.20, 89% CI [0.39, 2.09]) and can be considered medium (Std.Median = 0.66) and significant (0.50% in ROPE) [ROPE (full)]. As seen in Fig. 5c, the results indicate that adding action suggestions (AX and DXAX) greatly increases the likelihood that participants take the desired actions to resolve a failure.

On evaluating $\mathbf{H2_{RAX}}$, we find no significant effect of the optimal action rate with the accuracy of the suggestions.

Rate of correct diagnosis selection (RDX): Fig. 4c shows the proportion of of correct diagnoses made by participants in each study condition. Across conditions, the Median RDX ranged from 0.59 (DX_{80}) to 0.86 (DX_{100}).

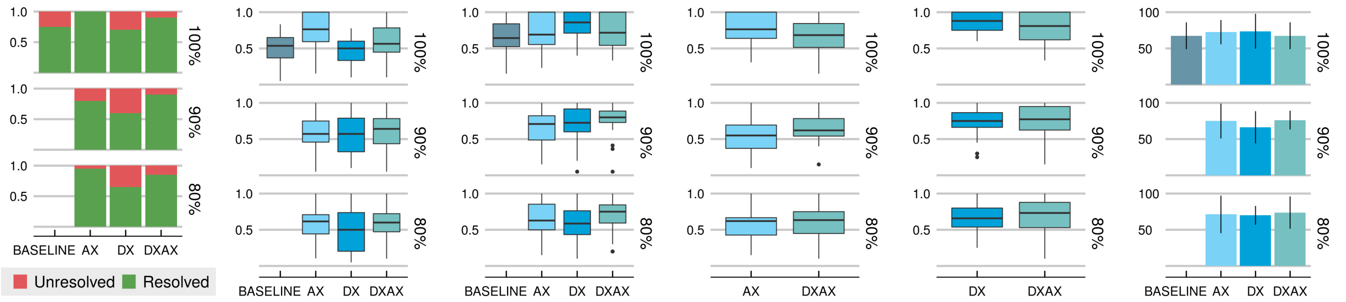
On evaluating $\mathbf{H1_{RDX}}$, we find that the correct diagnosis rate with diagnosis suggestions (DX) compared to BASELINE has a 97.8% [pd] of being positive (Median = 0.82, 89% CI [0.15, 1.51]) and can be considered small (Std.Median = 0.45) and significant (1.60% in ROPE) [ROPE (full)]. We also find that the correct diagnosis rate with both suggestions (DXAX) compared to BASELINE has a 98.7% [pd] of being positive (Median = 0.92, 89% CI [0.22, 1.60]) and can be considered medium (Std.Median = 0.51) and significant (0.98% in ROPE) [ROPE (full)]. As seen in Fig. 5d, the results indicate that adding diagnosis suggestions (DX and DXAX) results in more correct diagnoses.

On evaluating $\mathbf{H2_{RDX}}$, we find a 99.3% [pd] positive effect (positive slope) of accuracy in suggestions to correct diagnosis rate (Median = 0.48, 89% CI [0.16, 0.79]), which can be considered small (Std.Median = 0.26) and significant (1.25% in ROPE) [ROPE (full)]. As shown in Fig. 5e, there is a linear improvement in the rate of correct diagnoses from participants as the accuracy of suggestions improves.

Compliance with AX suggestions (CAX): Fig. 4d shows the proportion of participants that complied with action suggestions in the conditions that received AX suggestions (AX & DXAX). Across conditions, the Median CAX ranged from 0.55 (AX_{90}) to 0.76 (AX_{100}). On evaluating $\mathbf{H1_{CAX}}$ and $\mathbf{H2_{CAX}}$, we find no significant effects of either diagnosis suggestions (DXAX vs. AX) or the accuracy of the suggestions.

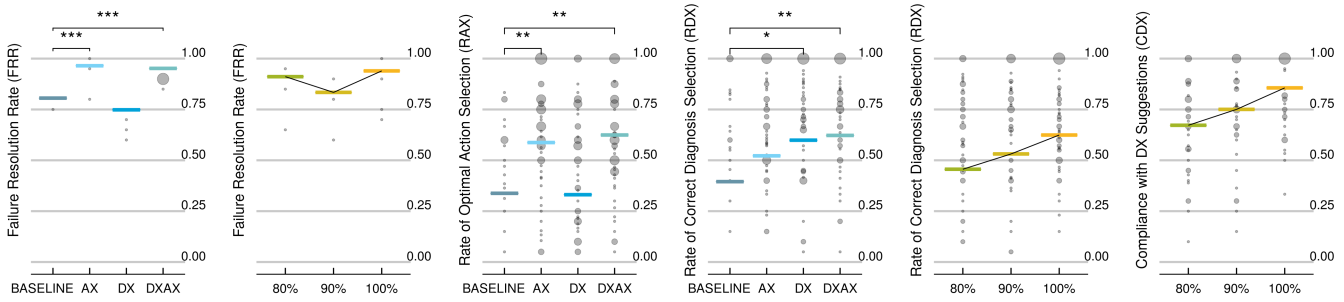
Compliance with DX suggestions (CDX): Fig. 4e shows the proportion of participants that complied with diagnosis suggestions in the conditions that received DX suggestions (DX & DXAX). Across conditions, the Median CDX ranged from 0.66 (DX_{80}) to 0.88 (DX_{100}).

On evaluating $\mathbf{H1_{CDX}}$, we find no significant effects of action suggestions (DXAX vs. AX) on the compliance rate. On evaluating $\mathbf{H2_{CDX}}$, we find that there is a 100% [pd]



(a) FRR (b) RAX (c) RDX (d) CAX (e) CDX (f) SUS

Fig. 4: Study data for each of the metrics defined in Table III.



(a) FRR v. Type (b) FRR v. Acc. (c) RAX v. Type (d) RDX v. Type (e) RDX v. Acc. (f) CDX v. Acc.

Fig. 5: Predicted Median of the posterior of significant effects after Bayesian analysis. Asterisks indicate effect sizes (see Table III). Points in the figure represent data from the study; larger points indicate more data instances with the same value.

positive effect (positive slope) of accuracy in suggestions to the rate of compliance with suggestions (Median = 0.77, 89% CI [0.39, 1.11]), which can be considered small (Std.Median = 0.42) and significant (0.08% in ROPE) [ROPE (full)]. As shown in Fig. 5f, the compliance of participants with diagnosis suggestions improves as the accuracy improves.

System Usability Scale (SUS): Fig. 4f shows the responses of participants to the SUS questionnaire across the different conditions. Across conditions, the median SUS scores range from 66 (DX_{90}) to 76 (DXAX_{90}).

On evaluating $\mathbf{H1_{SUS}}$, we find that the score with action suggestions (AX) compared to BASELINE has a 98.9% [pd] of being positive (Median = 11.68, 89% CI [3.82, 19.0]), which can be considered medium (Std.Median = 0.56) but not significant (3.02% in ROPE) [ROPE (full)]. We also find that the SUS score with diagnosis suggestions (DX) compared to BASELINE has a 95.2% [pd] of being positive (Median = 8.16, 89% CI [1.17, 16.04]), which can be considered small (Std.Median = 0.38) but not significant (9.48% in ROPE) [ROPE (full)]. The results indicate that adding suggestions, AX or DX, but not both (DXAX), might result in greater usability.

On evaluating $\mathbf{H2_{SUS}}$, we find that the accuracy of suggestions does not affect the usability score.

VII. Discussion and Conclusions

In this section, we discuss the implications of the statistical results presented above. We frame the discussion in relation to our research questions, with potential guidelines for future UX development highlighted in bold.

RQ1—Type of Decision Support: **Error recovery systems should display both feedback and feedforward information for maximum effectiveness.** From the results of evaluating $\mathrm{H1_{RAX}}$ and $\mathrm{H1_{RDX}}$, we find that participants are more likely to select the correct failure resolution actions if feedforward action recommendations (AX) are provided, and more likely to select the correct problem diagnoses if feedback as diagnosis suggestions (DX) are provided. Operators perform both functions in most common error recovery scenarios, with diagnosis typically informing identification of subsequent actions [14], [19], [15]. As a result, most systems should display both feedforward and feedback information.

Feedforward information has a greater effect on successful error resolution than feedback information. Analysis of the failure resolution rate metric in the context of $\mathrm{H1_{FRR}}$ highlights that participant ability to successfully recover from errors was greatest in the presence of feedforward action recommendations (AX & DXAX conditions) than when presented with feedback diagnosis information alone (DX). The result, consistent with UX research [3], demonstrates that although diagnosis suggestions aid in greater understanding of the overall state of the system (as shown by RDX results), feedforward information, suggesting "what-to-do" is more effective in leading to the correct solution.

RQ2—Decision Support Accuracy: **If the feedforward information is noisy (as in most systems), supplementing feedforward with feedback information (even if also noisy) leads to effective recovery strategies.** Analysis of the failure resolution rate metric in the context of $\mathrm{H2_{FRR}}$ highlights that participant performance was significantly affected by accuracy levels. Specifically, we observe a U-shaped response in which participant performance is likely to drop significantly in the 90% accuracy conditions. The drop is likely evidence of overtrust in the system [6]. However, Fig. 4a and Fig. 4b provide an indication that the performance drop might not be present in the DXAX conditions, indicating that diagnosis information, and the situational awareness that users might gain from it, can help ameliorate overtrust in the faulty system.

Additional Findings: Evaluating $H2_{CDX}$, we find that the compliance of participants with the suggestions linearly improves with the accuracy of the suggestions. The finding is consistent with prior work, which has found that operator compliance is dependent on the reliability of the decision support [28], [7]. Additionally, we find that there is a linear improvement in participants choosing the correct diagnoses (RDX) as the accuracy of suggestions improves, showing that reliance on decision support can also be dependent on the reliability of the decision support [28].

Evaluating $H1_{SUS}$, we find that both feedforward (AX) suggestions and feedback (DX) suggestions might independently improve the usability of an error recovery UX over a baseline without decision support, but the same might not hold true when both are present (DXAX). While the result might indicate a potential problem of too-much-information, further investigation is needed because the independent effects of DX and AX were not considered to be of practical significance (based on overlap within the ROPE).

Final Conclusions: In summary, we find that users are most effective in guiding error recovery when both feedback-focused diagnosis information and feedforward-focused action suggestions are presented. Note that further studies are required to better understand the above effects. For instance, we evaluated non-experts for whom diagnosis suggestions might not have been as useful as for experts. Additionally, when inaccurate, our suggestions for diagnoses and actions were inaccurate at the same time, but an accurate diagnosis might have ameliorated inaccurate action recommendations, or vice-versa. Future work can study the additional factors.

References

[1] B. Mutlu and J. Forlizzi, “Robots in organizations,” in *HRI*. New York, New York, USA: ACM Press, 2008, p. 287.

[2] D. Norman, *The design of everyday things: Revised and expanded edition*. Basic books, 2013.

[3] S. Coppers, K. Luyten, D. Vanacken, D. Navarre, P. Palanque, and C. Gris, “Fortunettes: Feedforward about the future state of gui widgets,” *Proc. HCI*, vol. 3, no. EICS, pp. 1–20, 2019.

[4] B. Lafreniere, P. K. Chilana, A. Fourney, and M. A. Terry, “These aren’t the commands you’re looking for: Addressing false feedforward in feature-rich software,” in *UIST*, 2015, pp. 619–628.

[5] S. Honig and T. Oron-Gilad, “Understanding and resolving failures in human-robot interaction: Literature review and model development,” *Frontiers in psychology*, vol. 9, p. 861, 2018.

[6] M. Natarajan and M. Gombolay, “Effects of anthropomorphism and accountability on trust in human robot interaction,” in *HRI*, 2020, pp. 33–42.

[7] N. Du, K. Y. Huang, and X. J. Yang, “Not all information is equal: Effects of disclosing different types of likelihood information on trust, compliance and reliance, and task performance in human-automation teaming,” *Human Factors*, 2019.

[8] D. Crestani, K. Godary-Dejean, and L. Lapierre, “Enhancing fault tolerance of autonomous mobile robots,” *Robotics and Autonomous Systems*, vol. 68, pp. 140–155, jun 2015.

[9] A. Sauppé and B. Mutlu, “The Social Impact of a Robot Co-Worker in Industrial Settings,” in *CHI*. ACM Press, 2015, pp. 3613–3622.

[10] E. Rogers and R. R. Murphy, “Tele-assistance for semi-autonomous robots,” 1994.

[11] M. Ai-Chang, J. Bresina, L. Charest, A. Chase, J.-J. Hsu, A. Jonsson, B. Kanefsky, P. Morris, K. Rajan, J. Yglesias, *et al.*, “Mapgen: mixed-initiative planning and scheduling for the mars exploration rover mission,” *IEEE Intelligent Systems*, vol. 19, no. 1, pp. 8–12, 2004.

[12] A. Mohseni-Kabir, C. Rich, S. Chernova, C. L. Sidner, and D. Miller, “Interactive hierarchical task learning from a single demonstration,” in *HRI*, 2015, pp. 205–212.

[13] M. Gombolay, A. Bair, C. Huang, and J. Shah, “Computational design of mixed-initiative human–robot teaming that considers human factors: situational awareness, workload, and workflow preferences,” *IJRR*, vol. 36, no. 5-7, pp. 597–617, 2017.

[14] A. B. Beck, A. D. Schwartz, A. R. Fugl, M. Naumann, and B. Kahl, “Skill-based Exception Handling and Error Recovery for Collaborative Industrial Robots.” in *FinE-R@ IROS*, 2015, pp. 5–10.

[15] S. A. Patel and A. K. Kamrani, “Intelligent decision support system for diagnosis and maintenance of automated systems,” *Computers & industrial engineering*, vol. 30, no. 2, pp. 297–319, 1996.

[16] L. Parker and B. Kannan, “Adaptive Causal Models for Fault Diagnosis and Recovery in Multi-Robot Teams,” in *IROS*. IEEE, oct 2006, pp. 2703–2710.

[17] R. R. Murphy and D. Hershberger, “Handling sensing failures in autonomous mobile robots,” *IJRR*, vol. 18, no. 4, pp. 382–400, 1999.

[18] B. Nushi, E. Kamar, E. Horvitz, and D. Kossmann, “On human intellect and machine failures: Troubleshooting integrative machine learning systems,” in *AAAI*, 2017.

[19] J. S. Breese and D. Heckerman, “Decision-Theoretic Troubleshooting: A Framework for Repair and Experiment,” in *UAI*. Morgan Kaufmann Publishers Inc., 1996, pp. 124–132.

[20] J. Guiochet, M. Machin, and H. Waeselynck, “Safety-critical advanced robots: A survey,” *Robotics and Autonomous Systems*, vol. 94, pp. 43–52, aug 2017.

[21] H. A. Yanco, A. Norton, W. Ober, D. Shane, A. Skinner, and J. Vice, “Analysis of Human-robot Interaction at the DARPA Robotics Challenge Trials,” *Journal of Field Robotics*, vol. 32, no. 3, pp. 420–444, may 2015.

[22] M. K. Lee, S. Kiesler, J. Forlizzi, S. Srinivasa, and P. Rybski, “Gracefully mitigating breakdowns in robotic services,” in *HRI*. IEEE, mar 2010, pp. 203–210.

[23] D. J. Brooks, M. Begum, and H. A. Yanco, “Analysis of reactions towards failures and recovery strategies for autonomous robots,” in *RO-MAN*. IEEE, 2016, pp. 487–492.

[24] H. Wu, S. Luo, L. Chen, S. Duan, S. Chumkamon, D. Liu, Y. Guan, and J. Rojas, “Endowing Robots with Longer-term Autonomy by Recovering from External Disturbances in Manipulation through Grounded Anomaly Classification and Recovery Policies,” *arXiv: 1809.03979*, sep 2018.

[25] S. Banerjee, A. Daruna, D. Kent, W. Liu, J. Balloch, A. Jain, A. Krishnan, M. A. Rana, H. Ravichandar, B. Shah, N. Shrivatsav, and S. Chernova, “Taking recoveries to task: Recovery-driven development for recipe-based robot tasks,” *ISRR*, 2019.

[26] M. Vasic and A. Billard, “Safety issues in human-robot interactions,” in *ICRA*. IEEE, 2013, pp. 197–204.

[27] J. K. Kruschke, “Rejecting or accepting parameter values in bayesian estimation,” *Advances in Methods and Practices in Psychological Science*, vol. 1, no. 2, pp. 270–280, 2018.

[28] S. R. Dixon, C. D. Wickens, and J. S. McCarley, “On the independence of compliance and reliance: Are automation false alarms worse than misses?” *Human factors*, vol. 49, no. 4, pp. 564–572, 2007.

[29] J. Brooke, “Sus-a quick and dirty usability scale,” *Usability evaluation in industry*, vol. 189, no. 194, pp. 4–7, 1996.

[30] P.-C. Bürkner, “Advanced Bayesian multilevel modeling with the R package brms,” *The R Journal*, vol. 10, no. 1, pp. 395–411, 2018.

[31] J. Cohen, “Statistical power analysis for the social sciences,” 1988.

[32] B. Carpenter, A. Gelman, M. D. Hoffman, D. Lee, B. Goodrich, M. Betancourt, M. Brubaker, J. Guo, P. Li, and A. Riddell, “Stan: A probabilistic programming language,” *Journal of statistical software*, vol. 76, no. 1, 2017.

[33] A. Vehtari, A. Gelman, and J. Gabry, “Practical bayesian model evaluation using leave-one-out cross-validation and waic,” *Statistics and computing*, vol. 27, no. 5, pp. 1413–1432, 2017.

[34] D. Makowski, M. S. Ben-Shachar, S. Chen, and D. Lüdecke, “Indices of effect existence and significance in the bayesian framework,” *Frontiers in Psychology*, vol. 10, p. 2767, 2019.