International Journal of Social Robotics (2024) 16:1381–1403
https://doi.org/10.1007/s12369-023-01041-w

ORIGINAL RESEARCH

Simulation Evidence of Trust Calibration: Using POMDP with Signal Detection Theory to Adapt Agent Features for Optimised Task Outcome During Human-Agent Collaboration

Sarita Herse[1] · Jonathan Vitale[2,3] · Mary-Anne Williams[1]

Accepted: 3 August 2023 / Published online: 16 August 2023

Abstract
Appropriately calibrated human trust is essential for successful Human-Agent collaboration. Probabilistic frameworks using a partially observable Markov decision process (POMDP) have been previously employed to model the trust dynamics of human behavior, optimising the outcomes of a task completed with a collaborative recommender system. A POMDP model utilising signal detection theory to account for latent user trust is presented, with the model working to calibrate user trust via the implementation of three distinct agent features: disclaimer message, request for additional information, and no additional feature. A simulation experiment is run to investigate the efficacy of the proposed POMDP model compared against a random feature model and a control model. Evidence demonstrates that the proposed POMDP model can appropriately adapt agent features in-task based on human trust belief estimates in order to achieve trust calibration. Specifically, task accuracy is highest with the POMDP model, followed by the control and then the random model. This emphasises the importance of trust calibration, as agents that lack considered design to implement features in an appropriate way can be more detrimental to task outcome compared to an agent with no additional features.

Keywords Human-agent collaboration · Recommender system · Trust calibration · Partially observable Markov decision process · Signal detection theory

1 Introduction

Trust underscores the way users interact and work with collaborative technologies. For example, in March 2018 a semi-autonomous Tesla vehicle veered into a roadside barrier and crashed, resulting in the death of the driver. Investigation found that the Autopilot function was engaged at the time of the accident and that both visual and audible safety warnings were displayed to the driver before the accident occurred [16]. In this scenario, the driver trusted the collaborative system too much—over-relying on the vehicle's ability to navigate varied road conditions using the Autopilot system. Further, the driver under-relied on the vehicle's inbuilt warning system—lacking trust to acknowledge the warning signals as a legitimate call to action. To avoid events like this, it is necessary to design for socially intelligent systems that are able to manage appropriate levels of user trust [46].

Trust calibration is one strategy to mitigate negative events caused by over-reliance and under-utilisation of technology [15]. Trust calibration is timely for human-agent interaction given the adoption of agents within collaborative team environments in industries like education [37, 51, 67] healthcare [20, 24, 39], and defense [59, 60]. Notably, trust calibration models have been employed in Human-Robot Interaction (HRI) contexts to improve team performance [14, 15, 59, 60], as well as agent-agent [53] and human-agent interaction [1, 48].

Sarita Herse and Jonathan Vitale have contributed equally to this work.

✉ Sarita Herse
s.herse@unsw.edu.au

Jonathan Vitale
Jonathan.Vitale@uts.edu.au

Mary-Anne Williams
Mary-Anne.Williams@unsw.edu.au

1 School of Management and Governance, University of New South Wales, Sydney, NSW 2052, Australia

2 School of Computer Science, University of Technology Sydney, 15 Broadway, Ultimo, NSW 2007, Australia

3 School of Computer Science, University of New England, Armidale, NSW 2350, Australia

The current work contributes to the development of trust calibration models, with the authors proposing a collaborative system that can a) estimate a specific users level of trust to it and b) adapt its interface and interaction style to nudge the users trust towards an optimum level that is both task and user-specific. In this work, these adaptations are changes implemented on the agent interface and are referred to as "features". The proposed system learns about the users decision making, behaviours, and perception of the system over time and adjusts its features accordingly in order to keep user trust close to the user-specific optimum level. In short, the system employs a model of trust calibration that utilises a battery of its own features to influence user trust in real-time, ensuring optimal performance on a collaborative task.

The proposed trust calibration methodology utilises the relationship between user perception of a collaborative agent and their trust to it to improve computational calibration models. Given its success estimating [53] and calibrating [15] trust, a computational partially observable markov decision processes (POMDP) framework is employed to adapt agent features in-task with the goal of optimising collaborative task outcome. The current work extends upon previously proposed models of trust calibration [15, 41, 45] by adopting signal detection theory (SDT) modelling to estimate user trust in-task in an unobtrusive way [22, 23].

Simulation evidence is provided to support trust calibration for Human-Agent Collaboration (HAC) with a collaborative recommender system. Training and test data for the simulation experiment is modelled off human subject data from a previously conducted online human experiment involving a recommender system assisting with a diagnostics task [23]. Results of the simulation experiment suggest that users collaborating with an agent that had implemented the proposed trust calibration methodology achieve the highest task accuracy overall and that this accuracy level is significantly higher than accuracy levels resulting from collaboration with agents that did not have intelligent feature adaption.

Taken together, the overall aim of this work is to investigate the mechanisms governing user trust and its relationship with decision making and collaborative task performance in the context of trust calibration during HAC. In particular, the current work aims to provide:

1. Probabilistic dynamic modelling of human-trust beahviour in a decision-automation context.
2. Explicit modelling of the coupling between humans and trust.
3. Analysis of human partner behaviour using SDT modelling with focus on the effect of agent feature inclusion on user trust in-task.

This paper begins with background literature on trust calibration models. The human experiment is summarised, followed by details of simulation parameter estimation using SDT and the simulation experiment. Results of the simulation experiment are presented and the paper is concluded with a discussion, including limitations of the current work and insight for future work.

2 Background

There has been a recent interest in the investigation of trust calibration for HAC. Research in this area has focused on the estimation of user trust via observation of user behaviour [18, 36, 43, 61] as well as utilising estimates of trust to guide agent behaviour [15, 61, 64]. The end goal of this work is to develop trust-seeking adaptive agents that are able to sense deviations in user trust and adapt their behaviours in response to improve task performance [65]—trust calibration in short. It is important for trust calibration models to distinguish dynamic trust from more long-term, stable notions of trust [23]. Dynamic trust is closely linked to reliance and is based on the current situational context. It is impacted by contextual changes in real-time, such as the implementation of agent features and the difficulty of a specific stimulus within a task, with these changes influencing user trust, perception, and decision making [23].

A Markov decision process (MDP) is a formal, non-deterministic model for planning subject to stochastic behavior. Under an MDP model, a system chooses to perform an action under full knowledge of the environment it is operating in [11]. Tasks can be modelled as an MDP when completed by a single agent [9]. At each time step, the agent takes an action to change the state of the overall environment, where the outcome of the action is a probability distribution over the system states. The agent then receives an action-dependent reward. In doing so, the agent's objective is to maximise the expected total reward over time, with the outcome of continued actions leading to the creation of a probability distribution over the agent states.

However, HAC means there is no longer a single agent involved with the task. When humans are introduced into the task, both the human and the agent can take actions to change the environment. Both the human and the agent must be considered and, given a model of human behaviour, the agent needs to plan its own actions accordingly. In HAC, the agent not only needs to infer a user's mental state, but it also needs to make decisions and take actions based on its inferences [1]. These actions further affect user perception and trust to the agent, impacting user decision making in-task, and therefore must be incorporated into the model. A key challenge here is that user trust is a latent variable—it is hidden and not directly observable [26, 63]. Notably, the

cognitive states of a user, like user trust to the agent, can only be estimated or inferred via behavioural observations and self-report measures [1]. This lack of direct observability of user trust creates an issue when considering MDP models for trust calibration during HAC. Thus, classic MDP models are limited within this context and instead must be adapted to account for the covert and dynamic nature of human trust.

2.1 Partially Observable Markov Decision Process

The POMDP framework provides a solution for planning under uncertainty by incorporating features of a MDP while accounting for partially observable states over time [1]. In comparison to a state that is completely hidden, a partially observable state affords the provision of some information about the underlying state—though, not enough to guarantee that it will be known with certainty. This renders POMDP models as powerful since they allow an agent to reason about actions to take in-task in order to gather knowledge that will be important for decision making later on. In the present work, the state the agent will need to predict is the current level of dynamic trust experienced by the human partner. This prediction allows for trust calibration in order to reach an optimal level of trust to maximise the outcome of collaborative task performance. It is not feasible for human users to self-report trust at every time step during collaboration. However, trust can be indirectly measured, with user trust being previously tied to decision making and reliance [10, 13, 49, 50, 56]. Indirect measures allow for trust to be inferred through human behaviour, decision making and reliance during HAC. This inference allows trust to be considered as partially observable as opposed to merely a hidden or unobservable state [34, 47, 62]. In particular, POMDPs have been used to estimate trust in agent-agent interactions [53] and to automatically generate robot explanations to improve team performance [60] in a HRI context.

2.1.1 POMDP for Trust Calibration

POMDP models for trust calibration have been considered in previous work by Chen and colleagues [15] who utilise a POMDP computational model to integrate user trust (as a latent variable) into robot decision making. Similar to the model proposed in the current work, their model provided a principled approach for a robot to infer the trust of a human teammate through interaction, reasoning about the effect of its own actions on human trust and selecting actions to maximise team performance over the long term. Both simulation and laboratory experiments were run to validate their trust calibration model. However, Chen and colleagues [15] focused on robot *actions* to mediate user trust—with the robot manipulating low-risk objects to initially build trust and intentionally failing in order to modulate user trust to achieve the best team performance. This behaviour change parallels the work by Min [41] who developed a POMDP model that engages specific robot behaviours to leverage human trust—actively modulating it for seamless HAC. These behaviours are similar to the adaptive features employed by Okamura & Yamada [45], highlighting the ability for such in-task variations to the agent to be employed within a POMDP framework for trust calibration. The results from the above-mentioned studies demonstrate that POMDP can be utilised to calibrate user trust in the context of HAC and that this can be achieved by changing the behaviour of the collaborative agent. Notably, while previous work in this area has demonstrated the efficacy of a computational POMDP model for trust calibration, the contributions of the current work is directed toward trust calibration with intuitive, non-obtrusive variation to the agent interface as well as the extension of this methodology across different classes of collaborative agent.

Furthermore, previous investigation into trust calibration using POMDP models have included various approaches to interpreting trust during collaboration. Prior work has considered the hidden state to be a combination of trust and workload [2–5, 38], both of which are considered dichotomously as either "high" or "low", while the observation is made using compliance and response time data. Chen et al. [15] use a similar model to Xu and Dudek [65], modelling human trust evolution as a linear Gaussian system—relating human trust causally to robot task performance. In comparison, the present work considers the hidden state as a users dynamic trust, with this implemented as a set of three discrete levels of trust. Further, rather than using compliance or performance, the observation of trust is considered as the parameter c, representing response bias within a SDT model (see Sect. 5 for further discussion).

Taken together, computational POMDP models close the loop between a users dynamic trust, that must be assessed in real-time, and the agents decision making process in order to maximise outcomes of collaboration [29]. This model reflects efficient trust calibration: granting an agent the ability to influence human trust systematically to reduce and increase trust in states of over-reliance and under-utilisation, respectively.

3 POMDP Framework

A POMDP is an extension of an MDP that partially accounts for hidden states through observable outcomes that are related. This can be applied to HAC whereby the latent variable of trust is accounted for via outcomes of user decision making that are outwardly observable. Formally, a POMDP is defined by a tuple [55]: $(S, A, T, R, \Omega, O, \gamma)$, where:

- S is a finite set of states $\{s_1, \ldots, s_N\}$

- A is a finite set of actions $\{a_1, \ldots, a_M\}$
- T is a set of conditional transition probabilities between states
- R: $S \times A \rightarrow \mathbb{R}$ is a reward function
- Ω is a set of observations $\{\omega_1, \ldots, \omega_K\}$
- O is a set of conditional observation probabilities
- $\gamma \in [0,1)$ is the discount factor

When it is possible to only partially observe the states, the agent must infer the current state from its immediate observation $\omega \in \Omega$ and action $a \in A$, and this information is represented as a belief b over all the possible states

$$b = [b(s_1), \ldots, b(s_N)]$$

where $b(s_i)$ is the probability that the agent is in state s_i and $\sum_{i=1}^{N} b(s_i) = 1$.

At each time period, the environment is in some unobservable state $s \in S$. The agent takes an action $a \in A$, which causes the environment to transition to state s' with probability $T(s' \mid s, a)$. At the same time, the agent receives an observation $\omega \in \Omega$ which depends on the new state of the environment s' as well as the action just taken a, with probability $O(\omega \mid s', a)$. Finally, the agent receives a reward r from the reward function $R(s, a)$ computing the reward for taking action a in state s. This process repeats over the course of the collaboration.

3.1 Planning

The goal of POMDP planning is to compute the optimal policy $\pi^*(b)$ that can select the optimal action $\hat{a}$ so the agent can maximise the expectation of the cumulative reward function $V^{\pi}(b)$:

$$\pi^*(b) = \hat{a} = \arg\max_{\pi} V^{\pi}(b) \tag{1}$$

and

$$V^{\pi}(b) = \sum_{t=0}^{\infty} \gamma^t r(b_t, a_t) \tag{2}$$

where $r(b_t, a_t)$ is the expected reward from the POMDP reward function over the belief state distribution:

$$r(b, a) = \sum_{s \in S} b(s) R(s, a) \tag{3}$$

Notably, the discount factor γ determines how much immediate rewards are favoured over more distant rewards, with $\gamma = 0$ signifying a policy directed to identifying an action that will result in the largest expected immediate reward, while $\gamma \rightarrow 1$ denotes a policy directed towards maximising the expected sum of future rewards. The value function $V^{\pi}(b)$ can be bounded to a fixed horizon H, meaning that Eq. 2 will become:

$$V^{\pi}(b) = \sum_{t=0}^{H} \gamma^t r(b_t, a_t) \tag{4}$$

The optimal value function V^* that employs the optimal policy π^* can be estimated by using the Bellman optimality equation [8]:

$$V^*(b) = \max_{a \in A} \left[r(b, a) + \gamma \sum_{\omega \in \Omega} Pr(\omega \mid b, a) V^*(\tau(b, a, \omega)) \right] \tag{5}$$

where $\tau(b, a, \omega)$ is the belief update function given the belief $b(s)$ on current state s, the taken action a and the gathered observation ω. The belief update function is defined as:

$$\tau(b, a, \omega) = b'(s') = \eta O(\omega \mid s', a) \sum_{s \in S} T(s' \mid s, a) b(s) \tag{6}$$

where $\eta = 1/\Pr(\omega \mid b, a)$ is a normalising constant with

$$\Pr(\omega \mid b, a) = \sum_{s' \in S} O(\omega \mid s', a) \sum_{s \in S} T(s' \mid s, a) b(s) \tag{7}$$

3.2 Priors Estimation

To implement a POMDP model it is necessary to know the conditional probabilities for T and O. In the current work, data from a training set of human participants is used to estimate those distributions. Specifically, the occurrences of each event are counted and normalised to obtain the priors.

Given N states and M actions, matrix T_{counts} with dimensions $N \times N \times M$ is used. Whenever a transition from state s_i to state s_j after taking action a_x occurs, 1 is added to the corresponding cell $T_{counts}^{(j,i,x)}$ at indexes (j, i, x). Similarly, given K possible observations, matrix O_{counts} with dimensions $K \times N \times M$ is used. For each observation ω_z gathered by the training subjects after taking action a_x and transitioning to state s_j, 1 is added to the corresponding cell $O_{counts}^{(z,j,x)}$ at indexes (z, j, x). The counts of both the matrix T_{counts} and O_{counts} are then normalised to represent probability distributions stored in the frequency matrix T_{freq} and O_{freq} respectively.

To implement the POMDP model, the reward function R must also be estimated. This is initially done by modelling a

conditional probability distribution over the set of the considered reward values $\Lambda = \{\lambda_1, \ldots, \lambda_L\}$ when taking an action a in state s. Similar to the previous estimations, given L possible reward values, matrix R_{counts} with dimensions $L \times N \times M$ is used. A count is added every time an observation of reward λ_z is received in the training data after taking an action a_x in state s_i in the corresponding cell $R_{counts}^{(z,x,i)}$ at indexes (z, x, i). The counts of this matrix are then normalised to represent probability distributions over the considered reward values. These frequencies are stored in the matrix R_{freq}. The reward function $R(s, a)$ can then be modelled as a weighted reward based on the probability of each reward value $\lambda \in \Lambda$ to occur when taking the considered action a in the given state s:

$$R(s, a) = \sum_{\lambda \in \Lambda} R_{freq}(\lambda, s, a)\lambda \tag{8}$$

It is possible that the training data does not cover all the possible events modelled by the counts matrix. For this reason, it is recommended to initialise each cell of the counts matrix with the value 1, as per the Lidstone estimate [35].

3.3 Online Learning

Although the priors estimated from training data can be a good starting point to inform the decision process of the POMDP model, better results in the test stage can be obtained by adapting those priors with new information gathered while performing the task with test subjects.

The reward λ obtained at each step after choosing an action a can be used to update the reward function $R(s, a)$. In reinforcement learning, this objective can be achieved by using the Q-learning algorithm. In this algorithm the function Q is considered to calculate the quality of a state-action combination:

$$Q : S \times A \rightarrow \mathbb{R}$$

This function can be updated after receiving a reward λ_t at time t by using the following equation:

$$\begin{aligned} &Q^{new}(s_t, a_t) \\ &\quad = Q(s_t, a_t) + \alpha\big(\lambda_t + \gamma_q \max_a Q(s_{t+1}, a) - Q(s_t, a_t)\big) \end{aligned} \tag{9}$$

where α is the learning rate ($0 < \alpha \leq 1$) and γ_q is a discount factor for the future reward expected from the new state.

Instead, the reward function $R(s, a)$ is used for the POMDP model considered in the present work, with it being updated at each step after receiving a reward. Unfortunately, the state is not directly observable in a POMDP framework. Instead, a probability distribution over states (beliefs) is utilised. Given an action a and a belief over states b, the formula in Eq. 9 can be adapted to update the expected reward function $R(s, a)$ for the considered action a and each state s. This is achieved by computing the reward value for each state s as a proportion of the received reward λ weighted by the probability of being in such stats (i.e. $b(s)$):

$$\begin{aligned} &R^{new}(s_t, a_t) \\ &\quad = R(s_t, a_t) + \alpha b(s_t)\big(\lambda_t + \gamma_r R^{future}(b') - R(s_t, a_t)\big) \end{aligned} \tag{10}$$

with b being the belief at time t, b' being the new updated belief at time $t + 1$ and $R^{future}(b')$ computed as:

$$R^{future}(b') = \sum_{s \in S} b'(s) \max_a R(s, a) \tag{11}$$

The frequency matrix T_{freq} and O_{freq} used to model the conditional probability distributions for T and O respectively, can also be updated without directly observing the state. To update the frequency matrix T_{freq} after choosing an action a_x and updating the belief distribution $b = [b(s_1), \ldots, b(s_N)]$ to the new belief distribution $b' = [b'(s_1), \ldots, b'(s_N)]$ the counts of the corresponding count cells in the matrix T_{counts} can be incremented given the result of the dot product between the two distributions b and b':

$$\begin{aligned} T_{counts}^{(*,*,x)} = T_{counts}^{(*,*,x)} &+ [b(s_1), \ldots, b(s_N)] \\ &\times [b'(s_1), \ldots, b'(s_N)] \end{aligned} \tag{12}$$

where the symbol $*$ denotes every cell of the matrix in that dimension, and x is the index of the selected action a_x. The counts matrix T_{counts} can then be normalised to obtain the updated frequency matrix T_{freq} that can be used as the conditional probability distribution for T.

Similarly, given the current observation ω_z after choosing the action a_x when in the belief b, the counts matrix O_{counts} can be updated with the following formula:

$$\begin{aligned} &O_{counts}^{(z,*,x)} \\ &\quad = O_{counts}^{(z,*,x)} + [b(s_1), \ldots, b(s_N)] \end{aligned} \tag{13}$$

with z and x being the indexes of the gathered observation ω_z and selected action a_x respectively. Then, the counts matrix O_{counts} can be normalised to obtain the updated frequency matrix O_{freq} used to model the probability distribution O.

4 Human Experiment Data

Real human data is required for simulation experiments in order to model, train and assess the proposed computational models. As such, data from an online human experiment involving a recommender system assisting with a radiology diagnostics task [23] was used. The purpose of this section

is to provide context behind the data used to model, train and assess the proposed computational model—which is the primary focus of the current work.

An online experiment was run to investigate whether stimulus difficulty and the implementation of agent features by a collaborative recommender system interact to influence user perception, trust and decision making. In this context, agent features were changes to the human-agent interface and interaction style and included: presentation of a disclaimer message, a request for more information from the user, and no additional feature. Signal detection theory was utilised to interpret decision making, with this applied to assess decision making on the task, as well as with the collaborative agent.

4.1 Design

A 3 × 2 simultaneous within-subjects design was implemented as an online experimental study. The first independent variable was agent feature with three levels: No Additional Feature, Disclaimer, and Request for More Information from user. After a request for more information, the user was made to report the presence of the following features: white spots, clouding, exposure, clarity, contrast, and other. The provision of additional information did not influence the final agent recommendation.

The second independent variable was trial difficulty with two levels: Easy and Hard. The dependent variable relevant to the current simulation experiment was decision making on the agent recommendation, operationalised via user sensitivity d' and bias c which were calculated using a SDT model applied to the agent (see Sect. 5.1.2 for calculation of SDT parameters).

4.2 Materials

The collaborative agent was a virtual recommender system tasked with assisting a human user to correctly classify the presence or absence of viral pneumonia in a set of X-ray images. The agent was a Softbank Pepper social robot. It was initially presented to the human user via an introduction video whereby the agent interacted using speech and gesture. Presentation of the agent during the X-ray task was achieved through static images paired with written text.

4.2.1 Collaborative Agent

A Softbank Pepper robot was used to visually represent the virtual collaborative agent "Assisto" as seen in Fig. 1. Google Speech Wavenet-C English (Australian) at 85% original speaking rate was used for speech generation. Participants were presented the collaborative agent via a two minute video where Assisto introduced themselves as an AI agent that had been specifically designed to help participants complete the X-ray classification task by providing them with a systems report containing a recommended decision. This video depicted the Pepper robot introducing itself using natural language, gesture, and animated lights. The tablet on the chest of the Pepper robot was not utilised for the study.

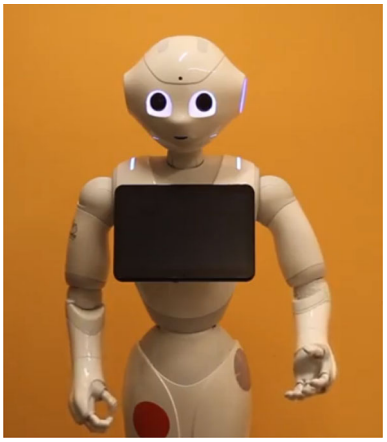

Fig. 1 Still image capture of Softbank Pepper robot "Assissto" during introduction video

4.2.2 Agent Features and Recommendation In-Task

The presentation of agent features and recommendations made to the participant in-task were achieved using static images of the Pepper robot and text to communicate information from the agent to the user. Additionally, the agent feature images included three separate icons – with each representing a specific feature condition. Examples of the *disclaimer* and *more information* features are seen in Fig. 2 and an example of a "Yes" recommendation from the agent is seen in Fig. 3.

4.2.3 X-ray Classification Task

An X-ray classification task was selected given that SDT can be applied to diagnostic accuracy [44]. Further, detection of viral pneumonia within X-ray images can be applied online as a real world application of a Yes-No task [22]. In psychophysics, a Yes-No task is a signal detection task where participants undergo a series of trials in which they must judge the presence or absence of a signal [7] – in this case, viral pneumonia.

Task Justification

While the real-world task of identifying abnormalities in X-rays (i.e. radiology diagnostics) is completed by trained professionals, the X-ray classification task in the present work was designed specifically to be completed by untrained participants in an online environment. Online studies can be limited in their ability to maintain user attention [12]. In an

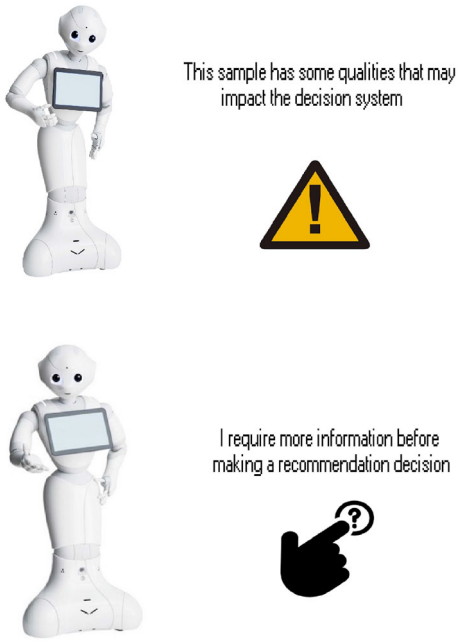

Fig. 2 Example of the agent features presented to users: disclaimer feature (top) and more information feature (bottom)

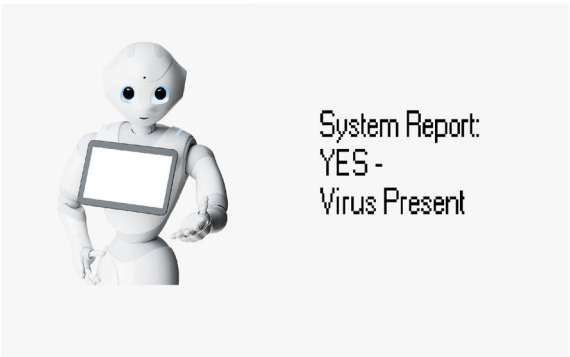

Fig. 3 Example of a "Yes" recommendation provided to the user

attempt to combat this, X-ray images depicting viral pneumonia were selected as stimuli to be used in an online Yes-No task in order to heighten the perceived importance of the study, given the study was run during the COVID-19 pandemic. Classification of predefined X-ray stimuli allows for the selection of the images into separate difficulty conditions and validation of the task itself (see *Data set classification* and *Stimuli Selection and Presentation* for further details).

Notably, the choice of the task itself is not important for answering the considered research question under the proposed methodology as long as the following assumptions apply: *(a)* the task is a Yes-No task; *(b)* there are labeled samples available; and *(c)* the considered samples can be classified by a layperson at above a 50% chance rate but without achieving a ceiling effect. These assumptions have been confirmed for the selected data set [22], providing evidence to assert the validity of this task in the present work.

Breakdown of X-ray Stimuli							
Easy Stimuli				Hard Stimuli			
Virus Absent		Virus Present		Virus Absent		Virus Present	
Correct Rec	Incorrect Rec	Correct Rec	Incorrect Rec	Correct Rec	Incorrect Rec	Correct Rec	Incorrect Rec
3	2	3	2	3	2	3	2

Fig. 4 Breakdown of stimuli selection

Data set classification

X-ray images of a pair of lungs with or without viral pneumonia were sourced from an Open Source data set [27, 28], see Fig. 5 for examples. These images were pre-classified as *viral pneumonia present* and *viral pneumonia absent*. However, these images needed to be further classified by difficulty level before the appropriate subset could be selected for the experimental task.

64 X-ray images (32 with viral pneumonia present and 32 with viral pneumonia absent) were presented to 48 participants ($Male$ = 30, $Female$ = 17, $Non\text{-}Binary$ = 1; M_{age} = 39.83, SD_{age} = 13.26) via Amazon Mechanical Turk. After presentation, participants answered *Yes/No* to whether they thought the X-ray image showed signs of viral pneumonia. Determining difficulty level of the X-ray images was achieved using a T-test to compare overall participant accuracy against chance rate. Images were defined as *significant correct* (p <0.05, positive mean difference), *significant incorrect* (p <0.05, negative mean difference), and *non-significant chance* (p >0.05).

Thirty-six images were classified as significant correct, meaning participants were able to correctly identify the absence or presence of viral pneumonia in each image above chance level. A further six images were classified as significant incorrect, meaning participants performed worse than chance level when classifying the images. Finally, twenty-two images were classified as non-significant chance, from which it can be inferred that participants performed at chance level. This classification of X-ray images allows for control over task difficulty.

Stimuli Selection and Presentation

Twenty distinct X-ray images were selected for use. This included ten *significant correct* images (five with virus present and five with virus absent) and ten *non-significant chance* images (five with virus present and five with virus absent). These were defined as the *Easy* and *Hard* task difficulty conditions, respectively. A summary of this breakdown is seen in Fig. 4.

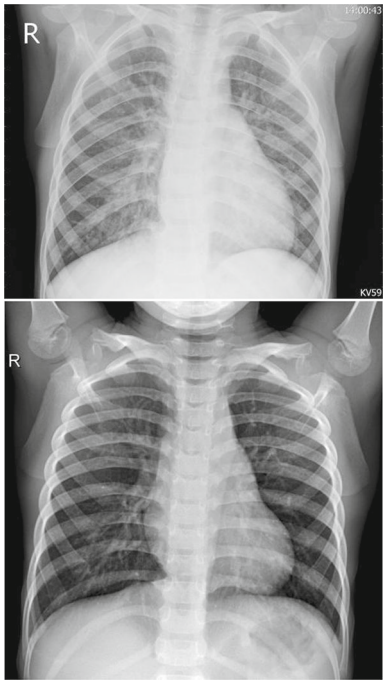

Fig. 5 Example X-ray images. Top image, Virus Present; Bottom image, Virus Absent

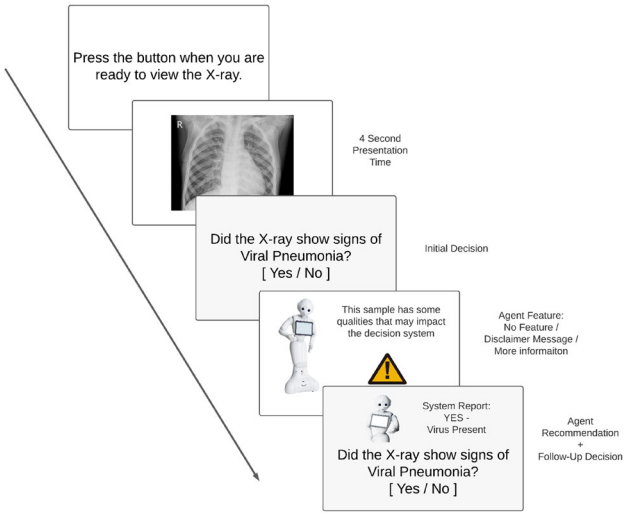

Fig. 6 Order presentation of a single trial

This composition of X-ray stimuli was selected to ensure that all novice participants, even those unskilled or untrained in the detection of abnormalities in X-ray images, would be able to successfully complete the task above chance level. Each X-ray image was presented for four seconds per trial. This length of presentation time was selected in order to maximise the total number of stimuli presented throughout the task while minimising potential impact to task performance, based off a recommender study with the same X-ray data set [22].

Experimental Task & Agent Performance

The twenty X-ray images were presented three times throughout the experimental task, once for each agent feature condition, resulting in a total of sixty trials. The order of the sixty trials was randomised for each participant, in line with a simultaneous within-subject design [25]. Participants received no penalty for incorrect responses during the X-ray classification task and were blind to task difficulty, the outcome of their decisions, and the true accuracy of the agent.

The agent performed at a 60% accuracy rate. This performance level was selected to prevent ceiling effects that may have otherwise been experienced with a higher accuracy rate [23]. The 60% accuracy rate was pre-generated and specific to each of the three feature conditions. Prior to experimentation, stimuli across each agent feature condition were randomly categorised in order to achieve a 60% split (i.e. the agent performed at 60% for each of the three conditions). This split was predetermined and, therefore, every participant experienced the same recommendation for each stimuli.

Trial breakdown

The following details a step-by-step breakdown of a single trial in the X-ray classification task. A visual representation of this is presented in Fig. 6. For each trial, participants were prompted with a screen with instructions: *Press the button below when you are ready to view the X-ray*. Participants were presented with an X-ray image for 4 s and then automatically displayed a new screen where they answered *Yes/No* to the following question: *Did the X-ray show signs of Viral Pneumonia?*

Participants were then presented with one of the three agent feature conditions. This included a banner with an image of the collaborative agent and one of the following three additional messages: *No additional feature* "A recommendation decision has been made!", *Disclaimer* "This sample has some qualities that may impact the decision system.", or *More Information* "I require more information before making a recommendation decision." The latter was followed by the instruction *Please select the features of the X-ray that may have influenced your decision* and a list of X-ray features for participants to select from including: white spots, clouding, exposure, clarity, contrast, and other. Importantly, this selection did not influence agent recommendation.

After presentation of the feature screen, participants clicked through to the final decision screen. Depending on the recommendation given, the banner on the final decision screen either displayed "System Report: YES-Virus Present" or "System Report: NO-Virus Absent" alongside an image of the collaborative agent. On the same page, the participants' initial answer was displayed and participants were asked to answer *Yes/No* to the same question: *Did the X-ray show signs of Viral Pneumonia?*

4.2.4 Trust Questionnaire

The 14-item sub-scale of the Trust Perception Scale-HRI [52] was used to assess trust once the participant was introduced to the agent but before starting the X-ray classification task (Pre-Trust) and after the participant had completed the X-ray classification task with the agent (Post-Trust). The Pre-Trust measure is relevant to the current work as it is used for the estimation of the regression parameters from c to s (see *State transitions conditional probabilities* in Sect. 6.2.3 for details).

4.3 Participants

Amazon Mechanical Turk was used to recruit 153 participants ($Male = 93$, $Female = 60$, $Non - Binary = 0$; $M_{age} = 42$, $SD_{age} = 11.08$) from Australia, Canada, the United Kingdom, and the United States. Participants were informed the experiment would take 25–30 min and were reimbursed USD$3 for their time. The inclusion criteria required participants to be proficient in English and at least 18 years old. All participants provided informed consent in accordance with human research ethical standards prior to experimentation.

Participants were excluded if it was determined that they did not complete the X-ray classification task seriously. This was achieved by considering participant decision making, whereby a one-tailed z-test with α=0.05 was run to compare the proportion of dichotomous decision making against a hypothesis level of 95% [23].

Following these criteria resulted in the data of 35 participants being removed from analysis under the assumption that they did not complete the X-ray classification task seriously. Therefore, final analyses were conducted on sample size N= 118 ($Male = 77$, $Female = 41$, $Non - Binary = 0$; $M_{age} = 41$, $SD_{age} = 10.86$).

4.4 Procedure

Participants first answered demographics questions. This was followed by information on the X-ray classification task and three example trials without assistance to ensure understanding of the task. Participants then met the collaborative assistant via the introduction video and answered five multiple choice attention check questions about the video to confirm their understanding of the role and ability of the collaborative agent during the task. Incorrect answers on the attention check questions resulted in a ten second time penalty before moving onto the next question.

Participants attempted six more example trials, this time—with the help of the agent, in order the ensure understanding of the task with the collaborative agent. This was followed with the Pre-Trust questionnaire. Participants then completed the X-ray classification task with the collaborative agent, followed by the Post-Trust questionnaire. Participants finished the session with a set of follow up questions regarding participant perception of the task and stimuli presented.

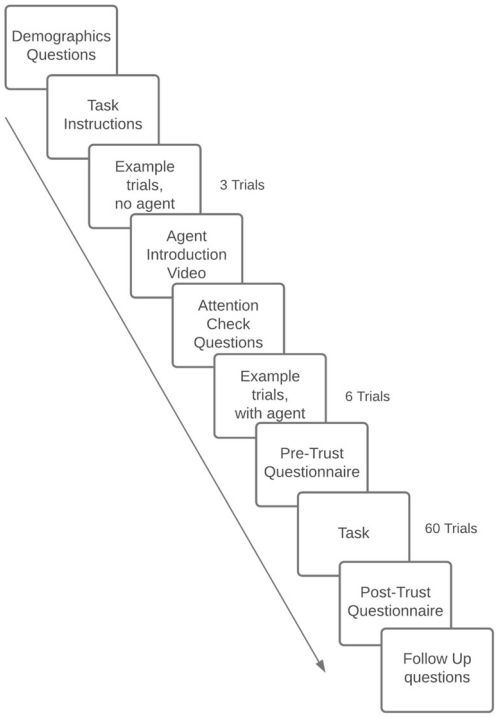

Fig. 7 Order presentation of the experiment procedure

4.5 Summary of Key Findings

SDT was applied to the collected data from the human experiment to interpret human behaviour and trust. Results from the statistical analyses demonstrated that decision change occurred more for the hard stimuli, with participants choosing to change their initial decision across all features to follow the agent recommendation $F(1, 117) = 7.03$, p =0.009, with an increase in decision change of 3%, 95% CA [0.01, 0.04] from the easy stimuli ($M = 0.23$, $SD = 0.18$) to the hard stimuli ($M = 0.26$, $SD = 0.18$).

Furthermore, agent features can be utilised to mediate user decision making and trust in-task, $F(2, 234) = 6.03$, p =0.003; with decision change significantly increasing by 3% from Disclaimer ($M = 0.23$, $SD = 0.20$) to No Additional Feature ($M = 0.26$, $SD = 0.21$), 95% CI [0.01, 0.06]; p =0.013, as well as significantly increasing by 4% units from the More Information ($M = 0.22$, $SD = 0.18$) to the No Additional Feature condition ($M = 0.26$, $SD = 0.21$), 95% CI [0.01, 0.07], p =0.001. Though there was a slight 1% decrease in decision change from the Disclaimer condition ($M = 0.23$, $SD = 0.20$) to the More Information condition ($M = 0.22$, $SD = 0.18$), this change was not statistically significant 95% CI [−0.03, 0.03], $p = 1.00$. This result suggests that the agent

is able to influence user trust, via decision change, by implementing or removing certain features. Specifically, the agent in the present work could reduce user trust by implementing the considered features or conversely, increase user trust by removing implemented features.

Trust is complicated and distinguishing between distinct components of trust is not an easy task. Use of SDT applied to the agent afforded the ability to infer trial-specific notions of user trust in-task, highlighting the dynamic nature of trust during HAC. This dynamic trust may be closely linked to reliance and is impacted by contextual changes in real-time, such as the implementation of agent features and the difficulty of a specific stimulus. These contextual changes are enough to influence user trust and, as a result, user decision making. Taken together, in-task trust ($\tau_{dynamic}$), is dynamic and based on the current situational context and thus, is distinct from long-term, stable notions of trust τ_{stable}. It is important that these two components of trust are considered separately for implementation within trust calibration models in order to mediate issues associated extreme levels of trust that might otherwise result in over-reliance and under-utilisation.

Taken together, the results of the online human study emphasised the complexity of user trust in HAC, highlighting the importance of considering the individuals perception of task context, including task difficulty and agent feature, in the wider perspective of trust calibration. Specifically, SDT models should be considered as a tool to detect in-task changes to task performance and dynamic user trust during HAC tasks and thus, should be considered within trust calibration frameworks.

5 Simulation Parameters Using Signal Detection Theory

SDT can be applied to binary decision making such as yes-no tasks, as long as participant responses can be compared to the presence or absence of the target stimulus. Thus, tasks involving recommender systems can be interpreted using SDT [54, 66].

In a SDT model, the participant's perception of a stimulus is assumed to be distributed along a psychological continuum. Decisions are made against a background of uncertainty, where the participants' aim is to tease out the decision *signal* from background *noise*. Both signal and noise are represented probabilistically within each participant (see Fig. 8). The extent to which these two distributions overlap can be estimated based on the participants' responses and whether or not the signal is present. The participant bases their decision relative to their own internal criterion β, where a signal will be reported *present* when the internal signal is stronger than β and *absent* when the internal signal is weaker than β [6]. Importantly, for every individual, an optimal operating point β^* exists where task performance is maximised [54].

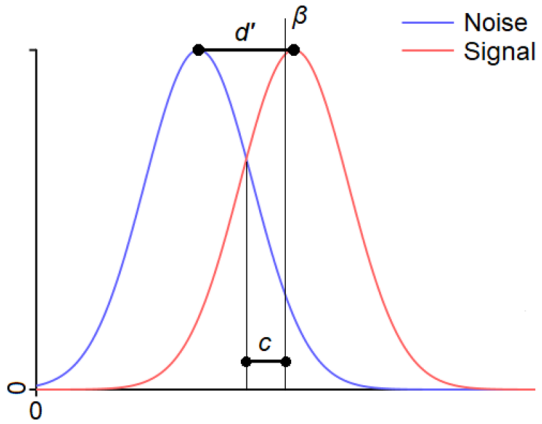

Fig. 8 SDT demonstrating signal and noise distributions as well as sensitivity d', response bias c, and the criterion β

Sensitivity d' measures the distance between the signal and noise means in standard deviation units [57]. It is independent of where β is placed. Thus, d' is a measure of performance that is independent of subject bias [6]. Response bias is the general tendency to respond *yes* or *no* as determined by the location of the criterion. It is estimated from the difference between the participant's β and an unbiased participants β [6]. Sheridan [54] highlights the link between the criterion β and the human agent's level of trust, suggesting that the criterion β indicates how the subject calibrates trust during interactions with an automated system over a set of repeated trials. While β has historically been the most popular measure of response bias [57], c offers an analogue of β, unaffected by changes in d'. As such, c is used to assess response bias. Response bias, c, is negative when the participant is more likely to report the signal is present and vice versa. The absolute value of c provides an indication of the strength of the subject's bias.

SDT can be applied to the context of a recommender system providing a recommendation on a Yes-No task (in this case, a radiology diagnostics task) in two distinct ways. First, the discrimination task can involve the goal of correctly classifying each X-ray image. In this interpretation, the stimuli are the X-ray images: signal refers to a stimulus X-ray with viral pneumonia present and noise refers to a stimulus X-ray without signs of viral pneumonia. The second application of SDT is applied by considering the discrimination task as the detection of a correct recommendation from the agent. Within this interpretation, the stimulus is the combination of the perceived X-ray image alongside the recommendation provided by the agent. Here, the signal is a correct recommendation from the agent for a particular X-ray image, whereas noise is considered an incorrect recommendation of the agent for that particular X-ray image.

User trust c can be appropriately evaluated using the SDT model [22]. In particular, the online experiment detailed in Sect. 4 considered dynamic trust during the discrimination task as the identification of a correct agent recommendation given a specific X-ray stimulus. Furthermore, evidence was provided to demonstrate that the implementation of distinct agent features during collaboration influenced dynamic trust during HAC. As such, the SDT model implemented throughout the following computational simulations considers the discrimination task as the correct classification of the agent recommendations in combination with the presented X-ray images.

In a SDT model, each signal and noise stimulus is modelled as an internal response distributed along a psychological continuum (with psychological continuum being user specific). This internal response is sampled from two Gaussian distributions, one for the signal stimuli and the other for noise stimuli, with both having mean $\mu = 0$ and standard deviation $\sigma = 1$. Thus, each stimulus is modelled as an internal response distributed along a psychological continuum within the range of $-\infty$ and $+\infty$, as depicted in Fig. 9.

Here, Ψ_{signal} and Ψ_{noise} represent the stimulus psychological continuum which defines the distribution of signal and noise stimuli, respectively. Finally, ψ^y represents the internal response elicited by a stimulus y sampled accordingly to the following equation:

$$\psi^y \sim \mathcal{N}(0, 1)$$

Where $\sim$ denotes the probabilistic sampling process and $\mathcal{N}$ denotes a Gaussian distribution. A signal sample placed further toward the right tail of the Gaussian distribution will, on average, be easier for participants to correctly detect as a signal. Conversely, the further to the left tail a signal sample lies on the Gaussian distribution, the harder it will be for participants to correctly detect it as a signal. The opposite is true for the noise stimuli. Specifically, the further a noise sample lies to the right tail of the Gaussian distribution, the harder will be for participants to correctly detect it as noise and the further a noise sample lies on the left tail of the Gaussian distribution, the easier will be for participants, to correctly detect it as noise (see Fig. 9 for example).

At this stage, the internal response of an individual participant has not been applied. In fact, when modelling the decision making process of the participant according to the SDT model, the position of the means for both the noise and signal Gaussian distributions on the participant's psychological continuum $\hat{\Psi}$ will be translated along the x-axis relative to the participants' sensitivity on the considered task (see Fig. 9). This is further discussed with reference to parameters from the SDT model in the following section.

5.1 Modelling the Participants' Decision Making

Part of the computational model to execute the simulation experiments will require additional simulation of the decision making process of the participant population that completed the task in the online human experiment (Sect. 4). The decision making process of participants was modelled using the SDT model [57] for the human experiment. The key parameters of SDT to determine a decision making outcome are sensitivity d' and bias c of the participant for the considered task. As previously mentioned, the internal response generated by the stimulus on the psychological continuum $\hat{\Psi}$ of each participant is dependent not only on the sampled position of the stimulus along the psychological continua Ψ_{signal} or Ψ_{noise}, but also on participant sensitivity d' for the considered task. When the participant i perceives a given stimulus ψ^y, a mapping function $\mathcal{M}$ takes the sampled internal response for the stimulus ψ^y and the participant's sensitivity d'_i to return a new internal response $\hat{\psi}_i^y$ of the stimulus ψ^y, denoting the position of the internal response for that specific stimulus onto the participant's i psychological continuum $\hat{\Psi}_i$.

For a given stimulus ψ^y and participant i, this mapping function is defined as:

$$\hat{\psi}_i^y = \mathcal{M}(\psi^y, d'_i) = \begin{cases} \psi^y + \frac{d'_i}{2} & \text{if } \psi^y \text{ sampled from } \Psi_{signal} \\ \psi^y - \frac{d'_i}{2} & \text{if } \psi^y \text{ sampled from } \Psi_{noise} \end{cases} \quad (14)$$

Therefore, a single stimulus sampled from Ψ_{signal} or Ψ_{noise} can manifest in different positions along the psychological continua of each participant depending on their d' (see Fig. 9).

5.1.1 Participants' Psychological Continua and Their Parameters

When applying SDT to model participant decision making for the detection of a correct (signal) or incorrect (noise) recommendation from a collaborative agent given a specific stimulus, the psychological continuum $\hat{\Psi}$ of the SDT model represents the overall distribution of all possible stimuli that can be perceived by participants throughout the task. As previously stated, these stimuli are considered as a combination of the presented X-ray image and the agent recommendation provided. There can be infinite X-ray images, with a possible correct or incorrect recommendation from the agent for each. As such, there are infinite signal and noise stimuli along this psychological continuum.

Additionally, the considered human experiment (Sect. 4) investigated the ability for independent agent features (provision of a disclaimer, a request for more information, and the

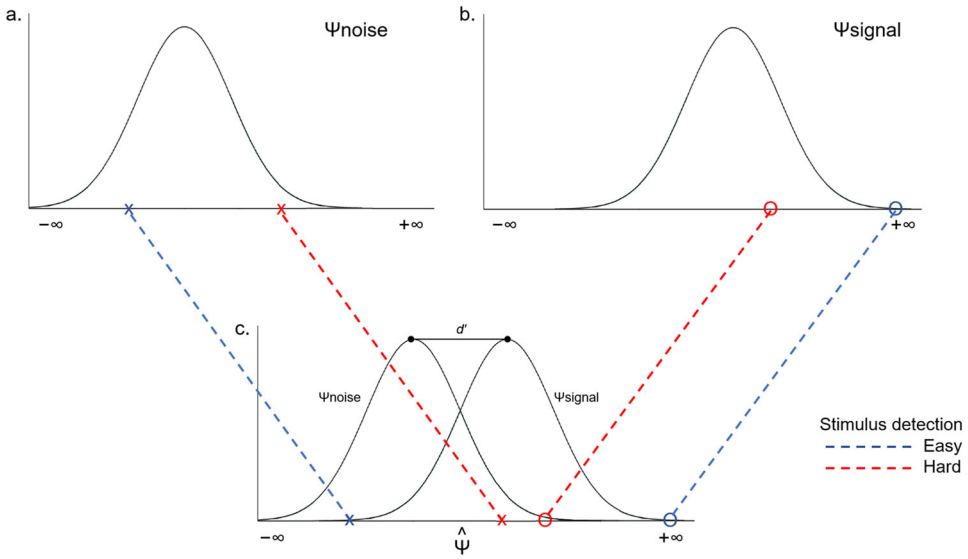

Fig. 9 A visual representation of four distinct stimuli is presented: one easy and one hard stimuli for both the noise and signal Gaussian distributions (a. and b., respectively). These two distributions are applied to each a specific participant as a representation of their internal response (c.) where the distance between distributions represents the specific sensitivity d' of each individual participant given the applied SDT model

no additional feature) to influence the users' perception of trust to the agent. This was assessed through changes to the SDT parameters d' and c. Taken together, each participant had a distinct SDT model comprised of parameters d' and c for each experimental condition. In other words, the decision making process for each participant can be modelled with the parameters outlined in in Table 1. All the parameters were estimated from the data gathered during the human experiment. Thus, for each participant i during each experimental condition x, there is a distinct parameter ${d'}_i^x$ leading to the perception of a distinct internal response as per the set of parameters Θ_i^x that can be estimated from the data gathered in each experimental condition of the human experiment.

5.1.2 Calculation of SDT Parameters

SDT parameters d' and c are calculated using the following two equations [57]:

$$d' = \Phi^{-1}(H) - \Phi^{-1}(F)$$
$$c = -\frac{\Phi^{-1}(H) + \Phi^{-1}(F)}{2}$$

Where Φ^{-1} represents the manthematical function converting probabilities into z-scores, H is used to indicate the hit rate (ie. the total number of hits divided by the total number of signal trials) and F represents the false-alarm rate (i.e. the total number of false alarms divided by the total number of noise trials). Figure 10 outlines each the signal and response combination required to results in a hit, false alarm, miss, and correct rejection for the current application of the SDT model.

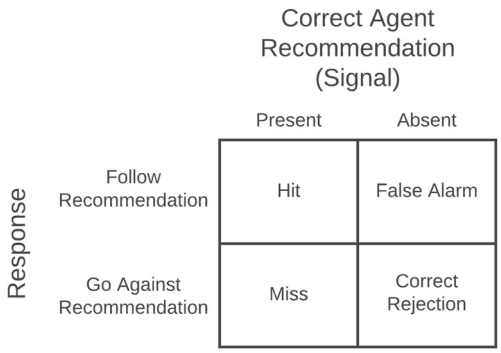

Fig. 10 Outcome breakdown of SDT applied to user decision making with a collaborative agent recommender system

Table 1 The experimental condition refers to a combination of stimulus difficulty (i.e. e easy and h hard) and agent feature (i.e. d disclaimer, m more information, and $\emptyset$ no additional feature)

Experimental condition	Model parameters
Easy-no feature	$\Theta_i^{e\emptyset} = \{d_i'^{e\emptyset}, c_i^{e\emptyset}\}$
Easy-disclaimer	$\Theta_i^{ed} = \{d_i'^{ed}, c_i^{ed}\}$
Easy-more information	$\Theta_i^{em} = \{d_i'^{em}, c_i^{em}\}$
Hard-no feature	$\Theta_i^{h\emptyset} = \{d_i'^{h\emptyset}, c_i^{h\emptyset}\}$
Hard-disclaimer	$\Theta_i^{hd} = \{d_i'^{hd}, c_i^{hd}\}$
Hard-more information	$\Theta_i^{hm} = \{d_i'^{hm}, c_i^{hm}\}$

5.1.3 Simulating the Decision Making Process

For each experimental condition x and each participant i there are a set of parameters Θ_i^x defining a psychological continuum $\hat{\Psi}_i^x$ for the participant i under the experimental condition x. During each task evaluation, a stimulus ψ^y can be sampled from either the signal psychological continuum

Ψ_{signal} or from the noise psychological continuum Ψ_{noise}. As such, the stimulus ψ^y is perceived by participant i as $\hat{\psi}_i^y$ as per the mapping equation (see Eq. 15), with the parameter ${d'}_i^x$ corresponding to the current experimental condition x the participant i is in. Therefore, the decision making process $\mathcal{D}$ can be modelled by using the following formula:

$$\mathcal{D}(\psi^y, \Theta_i^x) = \begin{cases} signal & \text{if } \mathcal{M}(\psi^y, d_i'^x) \geq c_i^x \\ noise & \text{otherwise} \end{cases} \tag{15}$$

For this considered model, an evaluation of the stimulus as signal means that the participant assessed the recommendation of the agent as correct and they accepted it. On the contrary, an evaluation of the stimulus as noise means that the participant assessed the recommendation of the agent as incorrect and they rejected it, thus choosing the alternative classification for the X-ray image.

5.2 Estimation of Stimuli Samples

As previously stated, there are infinite stimuli that can be sampled from Ψ_{signal} and Ψ_{noise}. However, only a subset of stimuli are able to be sampled from the considered population when performing an experiment with human participants. To ensure that the stimuli sampled during the computational simulations represent a valid subset of stimuli similar to those presented to participants in the human experiment (presented in Sect. 4), a region of the psychological continua Ψ_{signal} and Ψ_{noise} must be defined that is reflective of the experimental task.

Given a set of stimuli S_{exp} that represent the stimuli (i.e. X-ray image and given recommendation) used in the human experiment, the outcome parameters gathered from the experimental task E_{exp} given the considered population of human participants for the control conditions can be defined as:

$$\begin{aligned} E_{exp}(S_{exp}) &= \Theta_{exp}^{\emptyset} \\ &= \{(\Theta_1^{e\emptyset}, \Theta_1^{h\emptyset}), \ldots, (\Theta_n^{e\emptyset}, \Theta_n^{h\emptyset})\} \end{aligned} \tag{16}$$

With n refers to the number of participants in the experimental study and $\emptyset$ represents the control (i.e. no feature) experimental condition.

Similarly, given a set of simulated stimuli S_{sim} sampled from the psychological continua Ψ_{signal} and Ψ_{noise}, the outcome parameters gathered from the simulated experimental task E_{sim} given the simulated population of participants can be defined as:

$$\begin{aligned} E_{sim}(S_{sim}) &= \Theta_{sim}^{\emptyset} \\ &= \{(\tilde{\Theta}_1^{e\emptyset}, \tilde{\Theta}_1^{h\emptyset}), \ldots, (\tilde{\Theta}_n^{e\emptyset}, \tilde{\Theta}_n^{h\emptyset})\} \end{aligned} \tag{17}$$

The optimal sample of simulated stimuli S_{sim}^* is one that can produce a similar set of experimental outcome parameters, namely $\Theta_{sim}^{\emptyset} \approx \Theta_{exp}^{\emptyset}$. This sample lies within a specific region of the psychological continua Ψ_{signal} and Ψ_{noise}.

A grid-search algorithm was implemented to find an optimal region. This was done by introducing a set of assumptions and modelling sub-regions of the continua with appropriate hyper-parameters in order to ensure the similarity $\Theta_{sim}^{\emptyset} \approx \Theta_{exp}^{\emptyset}$. The considered assumptions and hyper-parameters were:

- The optimal regions in Ψ_{signal} and Ψ_{noise} are distributed around their zeros with the range being equally divided above and below zero, i.e. the optimal regions lies between $-j$ and $+j$, with $-j < 0 < +j$;
- Within each optimal region $[-j, +j]$ there is a portion of that region where "hard" stimuli (i.e. stimuli that are hard to classify) are sampled from. This hard-region is defined as a ratio ρ of hard stimuli with range $0 < \rho < 1$;
- The optimal regions for both Ψ_{signal} and Ψ_{noise} are equal in size, i.e. $j_{signal} = j_{noise}$;
- The ratios for both the optimal regions of Ψ_{signal} and Ψ_{noise} are the same, i.e. $\rho_{signal} = \rho_{noise}$;

Therefore, the hyper-parameters to optimise using a grid-search algorithm were j and ρ. The following values for j and ρ were considered for the grid-search algorithm:

$$j = \{0.1, 0.2, \ldots, 1\}, \rho = \{0.1, 0.2, \ldots, 0.9\}$$

The values for j were selected considering the assumption that a stimulus above or below 1 standard deviation in the psychological continua is too far from the mean of the Gaussian distribution and thus, unlikely to reflect the stimuli presented during the human experiment (for example, very noisy X-ray images).

In order to implement the grid-search algorithm, a distribution of stimuli was sampled for each combination of hyper-parameters j and ρ. For each iteration of the grid-search algorithm, labels were assigned to the generated samples according to the same distribution considered during the human experimental task described in Sect. 4. Specifically, 50% of the samples were assigned to the label "pneumonia present" and 50% to the label "no pneumonia present". Within each of those two classes, 60% of the samples were assigned to a correct recommendation (signal) and 40% to an incorrect one (noise). A stimulus ψ^y was considered valid if it was sampled within the considered interval $[-j, +j]$ and its difficulty was labelled given the following

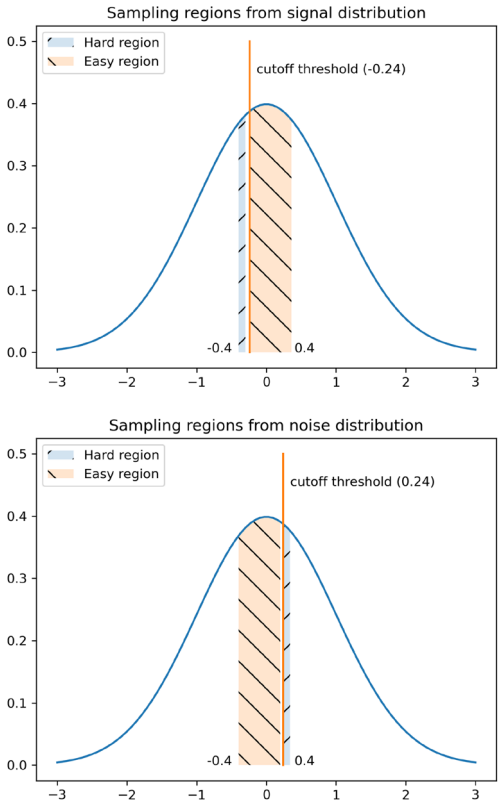

Fig. 11 Sampling regions of stimuli from the signal and noise Gaussian distributions

rules:

$$\text{For } \psi^y \sim \Psi_{signal} \begin{cases} hard & \text{if } \psi^y < -j + (2 * j) * \rho \\ easy & \text{otherwise} \end{cases}$$

$$\text{For } \psi^y \sim \Psi_{noise} \begin{cases} hard & \text{if } \psi^y > j - (2 * j) * \rho \\ easy & \text{otherwise} \end{cases}$$

The grid-search algorithm led to the estimation of the optimal parameters $j^* = 0.4$ and $\rho^* = 0.2$ (see Sect. 6.1). The intervals identified from the estimated parameters and application of the above rules are visually depicted in Fig. 11.

6 Simulation Experiment

The simulation experiment was run with the simulated participants and samples as described in Sects. 5.1 and 5.2 respectively. Simulated participants and samples were similar to what would be expected in the human experiment (see Sect. 4). Specifically, for each experimental subject i, task difficulty level l and experimental condition x, the experimental outcome parameters $\Theta_i^{l,x}$ were computed using the formula described in Sect. 5.1.2. These parameters were used to adapt the decision making process of the participants throughout the simulations based on the presented agent's feature. Importantly, these parameters were unknown by the agent and the POMDP model.

6.1 Estimation of the Hyper-Parameters j and ρ

To sample stimuli resembling those presented in the human experiment, the hyper-parameters j and ρ must be estimated from a set of simulated experimental subjects modelled on the human experiment data.

During the simulation, each simulated participant i first maps the simulated stimuli in their psychological continuum by using the mapping function $\mathcal{M}$ from Eq. 15 with their parameters $\Theta_i^{e\emptyset} = \{d_i'^{e\emptyset}, c_i^{e\emptyset}\}$, for easy stimuli, and $\Theta_i^{h\emptyset} = \{d_i'^{h\emptyset}, c_i^{h\emptyset}\}$, for hard stimuli, computed from the collected human data within the control (no feature) condition by using the formula provided in Sect. 5.1.2. Their decision making process is then simulated using Eq. 15.

After "perceiving" and "evaluating" each simulated stimulus sample, a set of simulation outcomes (hit, miss, false alarm, correct reject) is determined for each simulated participant, which are then computed into the relevant parameters d' and c as described in Sect. 5.1.2. These simulation outcomes are used to determine an estimation for the control (no feature) condition parameters achieved during the simulation: $\tilde{\Theta}^{e\emptyset} = \{\tilde{d'}^{e\emptyset}, \tilde{c}^{e\emptyset}\}$ and $\tilde{\Theta}^{h\emptyset} = \{\tilde{d'}^{h\emptyset}, \tilde{c}^{h\emptyset}\}$. The similarity $\Theta_{sim}^{\emptyset} \approx \Theta_{exp}^{\emptyset}$ of these parameters is then evaluated using the loss function L:

$$L = \sum_{i=1}^{n} \frac{1}{n} \sqrt{(d_i'^{e\emptyset} - \tilde{d'}_i^{e\emptyset} + d_i'^{h\emptyset} - \tilde{d'}_i^{h\emptyset} + c_i^{e\emptyset} - \tilde{c}_i^{e\emptyset} + c_i^{h\emptyset} - \tilde{c}_i^{h\emptyset})^2} \tag{18}$$

The optimal hyper-parameters j^* and ρ^* used during the computational simulations are those that minimise L. In the present work, the optimal values are $j^* = 0.4$ and $\rho^* = 0.2$, leading to the sub-regions depicted in Fig. 11.

6.2 Proposed Computational Models

Three types of models were compared using the computational simulations: control model, random model, and POMDP model. Each model represents a function that will predict the next action of the agent.

6.2.1 Control Model

The control model sees the agent present a consistent interface across all trials. Specifically, the simulated agent does not provide any additional feature to the simulated participants. As such, the mental model of the simulated participants is always realised by considering the parameters $\Theta^{e\emptyset}$ and $\Theta^{h\emptyset}$

for the easy and hard stimuli, respectively, as estimated from the collected human data.

6.2.2 Random Model

The random model selects the next feature (i.e. no feature, disclaimer, or more information) randomly from a uniform distribution. Therefore, the mental model of the simulated participant depends on the feature and difficulty level of the presented stimulus as per the parameters Θ estimated from the collected human data. This model reflects the human experiment (see section 4), where participants observed randomised agent features sampled from a uniform distribution. In the present work, the random sequence was the same for all participants to better compare the outcome of the simulated computational models. Since this is a computational simulation experiment, participant fatigue and order effects are not relevant. As such, there is no requirement to counterbalance the order of presentation for the simulated stimuli.

6.2.3 POMDP Model

This model makes use of a POMDP framework to determine the next feature to present to the simulated participants. In this case, the mental model of the simulated participant depends on both the presented feature and the difficulty level of the presented stimulus as per the parameters Θ estimated from the collected human data. These parameters are used by the simulation to determine the final decision of the simulated participant. However, these parameters (including the difficulty level) are not known by the POMDP models used to infer the state of each simulated participant.

In the POMDP condition, every simulated participant starts with the same POMDP model using priors estimated from a subset of the collected human data as described in Sect. 3.2. This subset is defined by the training set of the current fold and is used to train and test the model during that iteration (see cross-fold validation methodology in Sect. 6.2). However, every POMDP model evolves distinctively for each participant based on their ongoing decision making process and outcomes and the online learning process described in Sect. 3.3. The sequence of presented features for each participant is then defined by the ongoing training of the POMDP model, which aims to improve its policy to select the best feature at each time point for the considered subject.

Implementation of POMDP model

In the present work, the POMDP model is implemented as follows

States S refers to trust level with the considered interval range [0,1]. In this study were considered three equally sized discrete levels within this range: Low, Moderate, and High.

Actions A refers to the implementation of an agent feature during collaboration. There are three considered actions as per the features considered in the human experiment: No Feature Implemented, Disclaimer Feature Implemented and Request for More Information Implemented.

Observations Ω refers to the SDT parameter observation measured at every time step using the decisions made on the samples presented during the training batches so far (see Sect. 5.2 for an explanation of the samples presentation). This is the measure observed on behalf of user trust. Specifically, previous work highlights a relationship between user perception of an agent, decision making, and trust during HAC [17, 21, 22, 30, 31, 58]. SDT parameter c is used as c offers an analogue of β, with β considered in indicator of user trust that is unaffected by changes in d' [54]. The considered interval range of c [−2.32, 2.32] was considered for this study and divided into five equally sized discrete levels: Very Low, Low, Moderate, High, Very High. This interval range defines 98% of the probability covered by the Gaussian distribution along the psychological continuum.

State transitions conditional probabilities T defines the probability of the next state being s' given the previous state s and selected action a. The prior for this transition matrix T is estimated by using the training human data for each fold generated by the cross fold validation process (see Sect. 6.3). Specifically, linear regressions between self reported user trust and SDT parameter c measured during the control conditions of human experiment were used to estimate the parameters to regress an underlying state s from a given observation of the measured c. The estimated states s and s' were then used to model the conditional probability priors as explained in Sect. 3.2. This regression was used given a demonstrated linear relationship between user trust and SDT c [22]. The regression parameters were only computed for the control conditions "no feature easy" $c^{e\emptyset}$ and "no feature hard" $c^{h\emptyset}$ in order to consider the impact of task difficulty on user trust. However, for the process of priors estimation, these regression parameters were considered to predict an underlying state s from a measure c for all the considered experimental conditions. Although this choice may lead to priors unable to accurately predict the underlying state in each considered condition, those priors are further updated during the test stage as explained in Sect. 3.3. This estimation is only performed by using the data gathered from the training population (i.e. 90%) of each fold and it is solely used to generate a common prior for the matrix T to use as a starting point for all the simulated test participants.

Observations conditional probabilities O is the probability matrix for the observations. The prior for this matrix is computed with a process similar to that used to estimate the prior for the matrix T and described in Sect. 3.2, namely by using the linear regression parameters mapping values of c to trust

levels s'. This prior was computed by only using the data gathered from the training population (i.e. 90%) of each fold. *Reward values* Λ is the set of rewards considered for the POMDP model. These rewards are calculated by computing the difference in accuracy $\Delta\mathcal{A}$ obtained by comparing the measured accuracy level $\mathcal{A}$ achieved in the previous training batch with the measured accuracy level $\mathcal{A}'$ achieved in the current training batch:

$$\Delta\mathcal{A} = \mathcal{A}' - \mathcal{A} \tag{19}$$

Accuracy is measured as the number of hits and correct rejections over the total number of presented samples in the training batch. The reward value is allocated based on the function $\mathcal{R}(\Delta\mathcal{A})$ described below:

$$\mathcal{R}(\Delta\mathcal{A}) = \begin{cases} -3 & \text{if } \Delta\mathcal{A} < -0.1 \\ -2 & \text{if } -0.1 \leq \Delta\mathcal{A} < -0.05 \\ -1 & \text{if } -0.05 \leq \Delta\mathcal{A} < -0.01 \\ 0 & \text{if } -0.01 \leq \Delta\mathcal{A} < 0.01 \\ +1 & \text{if } 0.01 \leq \Delta\mathcal{A} < 0.05 \\ +2 & \text{if } 0.05 \leq \Delta\mathcal{A} < 0.1 \\ +3 & \text{if } \Delta\mathcal{A} \geq 0.1 \end{cases} \tag{20}$$

Value function discount factor $\gamma \in [0, 1)$ is the discount factor for the value function. The present study considers $\gamma = 0.95$ and a horizon $H = 1$, meaning that the POMDP model only looks one step ahead from the interaction when making a prediction.

Online learning discount factor $\gamma_r \in [0, 1]$ is the discount factor for the learning process of the reward function. The present study considers $\gamma_r = 1$.

Learning rate $\alpha \in (0, 1]$ is the learning rate for the update process of the reward function. The present study considers $\alpha = 1$.

6.3 Method

A 10-fold cross-validation method was implemented to train the POMDP model and determine the next best agent feature to display. The application of k-fold cross-validation methodologies in applied machine learning is commonly used to compare and select the best model for a given predictive modelling problem as cross-validation methodologies are simple to understand, easy to implement and result in skill estimates that generally have a lower bias than other methods [40]. Cross-validation is a technique used to evaluate predictive models by partitioning the original sample into a training phase to train the model, and a test phase to evaluate it. In k-fold cross-validation, the original sample is randomly partitioned into k equal size sub-samples. Of the k sub-samples, a single sub-sample is retained as the validation data for testing the model, with the remaining $k - 1$ sub-samples used as training data. The cross-validation process is then repeated k times (i.e. the number of folds), with each of the k sub-samples used exactly once as the validation data. The results from the folds can then be averaged (or otherwise combined) to produce a single estimation. The advantage of this method is that all observations are used for both training and validation, and each observation is used for validation exactly once.

6.3.1 Stimuli Presentation and Learning Process

Throughout the experimental task, the presented simulated stimuli are divided into training batches and test batches. The stimuli in the training batches are stimuli for which the label (i.e. pneumonia or not pneumonia) is known by the agent, whereas the test batches are stimuli for which the label is not known by the agent. However, the label is known by the computational model to compute the final task accuracy used to run the comparative analyses. The test batches and training batches each contain 10 stimuli samples and these are alternated throughout the experimental task. Every epoch includes a test batch and a training batch, thus each epoch is composed of a total of 20 stimuli samples. During each epoch, the same agent feature is presented to the simulated participants for all the epoch's stimuli samples. When presented with the test stimuli from the test batch, the simulated participant is asked to classify a set of simulated stimuli and the decisions made on these samples are used to estimate the participant's task accuracy for the current epoch.

The learning process occurs after the presentation of stimuli in the test batch. The training batch sees the presentation of stimuli with labels known by the agent implementing the POMDP model. These stimuli are used to estimate the current parameter c of the participant, with c used as an indirect observation ω of the participants hidden trust state [22]. The training samples are also used to estimate the participant accuracy $\mathcal{A}$ at time t. This is compared to accuracy $\mathcal{A}$ over the training samples at time $t - 1$ to determine a reward for the current time t (see the reward function described in Sect. 6.2.3). Given the observation c^t computed by considering the participant's outcomes on the training samples presented up to time t and the action a^{t-1} at time $t - 1$, the belief state and reward function of the participant's POMDP model are updated to predict the new action a^{t+1} to present to the participant during the next epoch (i.e. at time $t + 1$). The conditional probabilities T and O are also updated accordingly to the process described in Sect. 3.3.

As mentioned in Sect. 6.2, in the case of the control model the selected action will always result in the selection of no additional features; whereas for the random model, the

selected action will always be randomly selected to be the same for all the simulated participants.

For each cross-fold validation process a set of stimuli with known and unknown labels were sampled to generate the set of training and test batches to present during the considered epochs. Each simulated participant evaluated the same set of training and test stimuli throughout the experimental task with the three considered models (control, random and POMDP). The test stimuli were sampled according to the parameters used in the human experimental study. Namely, the probability of sampling an image with label "pneumonia present" was 50% and the probability of the agent providing a correct recommendation for each sampled image was 60%. The training stimuli considered a correct recommendation level from the agent of 50% to prevent biases when estimating c as an indirect observation ω of the underlying trust level s. The probability of a hard or easy X-ray sample in training and test stimuli is indirectly modelled by the estimated ratio parameter ρ^*. For the simulations, a total of 100 epochs were considered. Therefore, a total of 1000 samples were used for the training batches and 1000 samples for the test batches.

7 Results

The results from each validation set of the cross fold validation process were gathered and combined together. Here, accuracy refers to the ability for participants to correctly classify the test stimuli into their correct label (i.e. "pneumonia present" or "no pneumonia present"). Paired-samples t-tests with Bonferroni adjusted $\alpha = 0.025$ to control for inflated Type I error were run for average and cumulative accuracy to determine whether there were statistically significant mean differences between the three computational models. Finally, a one-way repeated measures ANOVA was run to assess final task accuracy across all three computational models. Data are mean ± standard deviation, unless otherwise stated.

7.1 Average Accuracy by Epoch

Average accuracy refers to the accuracy achieved by the whole participant population for each proposed model within the test batch of each epoch. The average accuracies by epoch for the three conditions are visually represented in Fig. 12.[1] Average accuracy was higher for the POMDP model (0.63 ± 0.09) compared to the Control model (0.58 ± 0.09), a statistically significant difference of 0.05 (97.5% CI [0.04, 0.05]), $t(99) = 32.89$, $p < 0.001$, d = 3.29—a small effect size.

[1] Due to space limitations in the bar chart, the figure displays the results of every 10 epochs. However, the statistical analyses consider all 100 epochs.

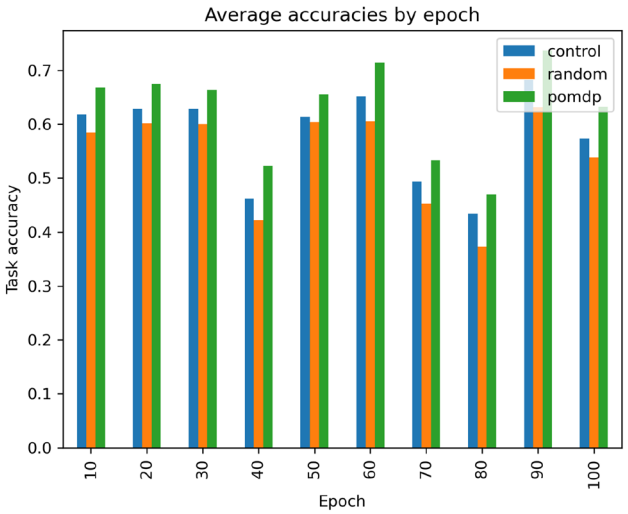

Fig. 12 Average accuracy of each model by epoch, with each column representing every 10 epochs. Statistics were computed on all 100 epochs

Accuracy was also higher for the POMDP model when compared against the Random model (0.55 ± 0.09), a statistically significant difference of 0.08 (97.5% CI [0.07, 0.09]), $t(99) = 30.56$, $p < 0.001$, $d = 3.06$—a small effect size. Finally, accuracy was higher for the Control model when compared against the Random model, a statistically significant difference of 0.03 (97.5% CI [0.03, 0.04]), $t(99) = 18.21$, $p < 0.001$, $d = 1.82$—a small effect size.

7.2 Cumulative Accuracy by epoch

Cumulative accuracy refers to the development of task accuracy over each test batch of each epoch for the whole participant population and thus, represents how task accuracy evolves for each model over time. At each epoch, the current cumulative accuracy was computed as the average accuracy achieved by the participant population when considering the outcomes over all the presented test stimuli for each model. Cumulative accuracy was higher for POMDP model (0.61 ± 0.01) compared to Control model (0.57 ± 0.01), a statistically significant increase of 0.04 (97.5% CI [0.04, 0.04]), $t(99) = 35.15$, $p < 0.001$, $d = 3.52$—a small effect size. Cumulative accuracy was also higher for the POMDP model when compared against the Random model (0.54 ± 0.01), a statistically significant increase of 0.07 (97.5% CI [0.07, 0.07]), $t(99) = 60.66$, $p < 0.001$, $d = 6.07$—a medium effect size. Finally, cumulative accuracy was higher for the Control model compared to the Random model, a statistically significant increase of 0.03 (97.5% CI [0.03, 0.03]), $t(99) = 70.76$, $p < 0.001$, $d = 7.08$—a moderate effect size. These differences are highlighted in Fig. 13.

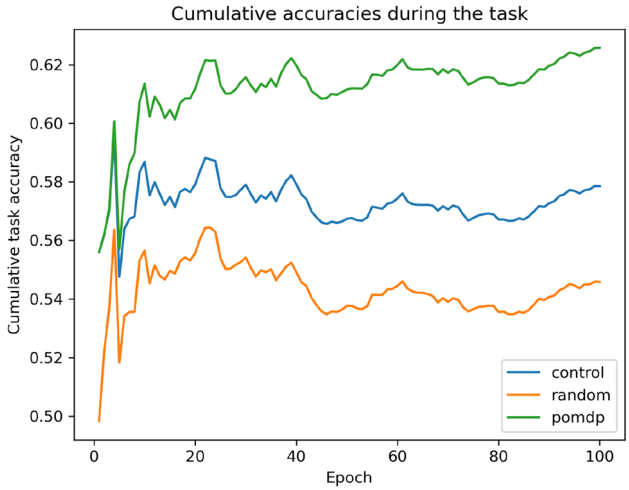

Fig. 13 Cumulative accuracy of each model throughout the simulation experiment

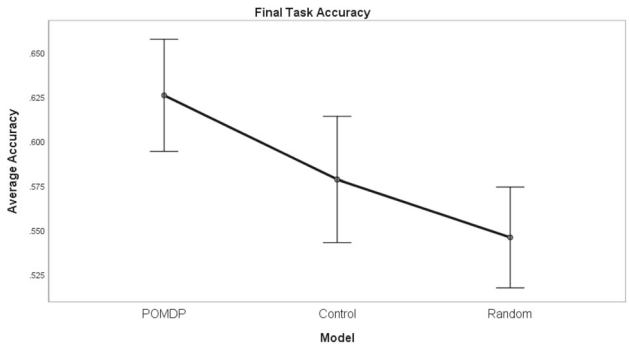

Fig. 14 Average final task accuracy across all participants for each computational model is presented, including 95% confidence intervals

7.3 Final Task Accuracy by Participants

Final task accuracy refers to the accuracy achieved by each participant at epoch 100 with each model. This represents the final accuracy of the participant population at the end of the simulated HAC task. A one-way repeated measures ANOVA was conducted to determine whether there were statistically significant differences in final task accuracy between the three simulated collaborative system models. The assumption of sphericity was violated, as assessed by Mauchly's test of sphericity, $\chi^2(2) = 31.49$, $p < 0.001$. Therefore, a Greenhouse-Geisser correction was applied ($\epsilon = 0.808$). The inclusion of the models elicited statistically significant changes in final task accuracy, $F(1.62, 189.06) = 27.81$, $p < 0.001$, partial $\eta^2 = .19$. Accuracy was lowest for the Random model ($M = 0.55$, $SD = 0.16$). Pairing participants with the Control model improved accuracy ($M = 0.58$, $SD = 0.19$). However, pairing participants with the POMDP model had the highest accuracy outcome ($M = 0.63$, $SD = 0.17$). Means of the final task accuracy of each model is depicted with 95% confidence intervals in Fig. 14. Post-hoc analysis with a Bonferroni adjustment confirms the significance of these differences, whereby final task accuracy improved statistically significantly from the Random model to the POMDP model ($M = 0.08$, 95% CI [0.06, 0.10], $p < 0.001$), the Control model to the POMDP model ($M = 0.05$, 95% CI [0.02, 0.08], $p = .001$), and from the Random model to the Control model ($M = 0.03$, 95% CI [0.01, 0.06], $p = .015$).

8 Discussion

A simulation experiment was run to assess the impact of implementing three different decision making models on task accuracy during a HAC task. There was a control model which did not include the implementation of any additional agent features, a random model which saw the random presentation of agent features across the simulation, and a POMDP model which based agent decision making on the defined POMDP framework outlined in section 6.2.3. The participants, task, and stimuli were simulated (as detailed in Sect. 5), with parameters based off the human experiment (see Sect. 4). Evidence demonstrates that the proposed POMDP model can appropriately adapt agent features in-task based on human trust belief estimates in order to achieve trust calibration. Specifically, task accuracy is highest with the POMDP model, followed by the control and then the random model. This emphasises the importance of trust calibration, as agents that lack considered design to implement features in an appropriate way can be more detrimental to task outcome compared to an agent with no additional features.

8.1 POMDP Model is Most Accurate

The underlying result of the simulation experiment suggests that the POMDP model, which worked to identify the best agent feature to present to each participant at every time point during a HAC task, resulted in greater task accuracy overall when compared against the random and control models. In particular, this result was seen for the POMDP model across average accuracy, cumulative accuracy, and final task accuracy.

When assessing average and cumulative accuracy, the POMDP model achieved higher accuracy compared to the control model and the random model. This is further emphasised with the results for final task accuracy, which demonstrated that the POMDP model had the best performance overall, followed by the control model and then the random model. This provides evidence to suggest that trust calibration through the implementation of agent features via a POMDP model leads to improved task performance. Furthermore, the result highlights the fact that the random, non-considered implementation of agent features can lead to worse task outcomes during HAC compared to a control

model (i.e. an agent that presents no additional features over the course of collaboration). This underscores the importance of a considered, user-specific approach when implementing agent features in attempt to improve the outcome of HAC. In particular, this was achieved in the present work using a POMDP model incorporated with an SDT model to infer user trust in order to select agent features in-task to calibrate user trust during collaboration.

8.1.1 Trust Calibration Can be Achieved for an Agent with Intuitive Changes To The Interface

In this work, there is the assumption that there is an optimal position of the threshold criterion c^* for each participant that, when achieved, leads to optimal task accuracy. There is a relationship between optimal trust level τ^* and the parameter c^*. For every participant, the POMDP model presents the feature that is predicted to result in the best collaborative decision making between the human and the agent. Notably, the model is not directed towards maximising user trust, rather its goal is to maximise task outcome by identifying an optimal trust level τ^* via a trust calibration process that can implement an optimal value for the parameter c^* leading to the highest task accuracy for each participant. Thus, every POMDP model is user-specific, with the POMDP model for each participant converging to a unique interface in order to optimise task outcome for them. This provides evidence to support the importance of personalisation when designing for collaborative agents, a concept generally acknowledged in the HAC literature [32, 33, 42, 45], reinforcing the consideration of the user as well as the technology and environment when considering humans and agents working together [19]. Rather than design an interface that works, on average, for a group of people or specific population, a collaborative agent should have the capacity to select the most ideal interface for each individual user. In particular, this interface may contain no additional features, one specific feature, or even a combination of features integrated together.

8.1.2 Calibrating Dynamic Trust in Task

Trust calibration in the present work is applied to dynamic trust. In this context, dynamic trust refers to the trust inferred in-task and is assessed at each time step over the course of the interaction. This work sought to investigate whether the outcome of HAC could be improved by discretely influencing dynamic trust at each of these time steps via the implementation of agent features determined by the POMDP model. The outcome of the simulation experiment demonstrate that this is possible, highlighting the importance of intelligently implementing agent features to do so. Taken together, regardless of the stable, conscious perception of trust an individual has toward a collaborative agent, it is possible to calibrate their dynamic trust in-task through the intelligent implementation of agent features in order to optimise the outcome of collaboration.

8.2 Application of SDT Model within Trust Calibration Framework for HAC

The application of SDT modelling within a trust calibration framework is novel. SDT modelling can be used to determine participant sensitivity d' and bias c (used to infer user trust) when assessing user decision making with a collaborative recommender system. Further, the application of the SDT model in the present work is also novel. The model used considers the users ability to detect signal and noise, referring to a correct and incorrect recommendation from the agent, respectively. This framing differs to more classic applications of SDT where the signal is often considered as the presence of a target within a stimulus (i.e. the presence of viral pneumonia within an X-ray image). The POMDP model, which used parameters modelled from SDT applied to detection of correct/incorrect agent recommendations, resulted in the highest task accuracy of all considered computational models across all accuracy measures. As such, the novel use of SDT within the identified trust calibration framework results in improved task outcome and should be considered for future work on trust calibration with collaborative recommender systems.

8.3 Performing this Methodology in a Different Application Domain

The methodology outlined in the present work has been designed to be implemented in different application domains. There are four underlying assumptions that must be met in order to do so:

i. The HAC task must be a binary decision making task (e.g a Yes-No task).
ii. The data set used must include a portion of samples with known labels.
iii. A collaborative agent recommender system must provide a recommendation for one of the two presented classes (a requirement in order for a participant to accept/reject the recommendation of the agent).
iv. Adaptive features included in the methodology vary the interaction between collaborative agent and human.

If these four assumptions are met, the following steps can be followed in order to assess the impact on task accuracy of including certain adaptive features within a collaborative recommender system context:

1. Run a human study with the separate features as conditions that are presented together in random order. See Sect. 4 for an example.
2. Apply SDT modelling to the outcome of participant decision making. Specifically, the SDT model of interest is the one applied to the participants appraisal of the agent recommendations. See Sect. 5 for details.
3. Estimate the following parameters for the simulation experiment: Θ (Sect. 5.1), POMDP priors (T, O and R, Sects. 3.2 and 6.2.3), hyper-parameters j and ρ (Sects. 5.2 and 6.1).
4. The POMDP model is applied to the simulation experiment with simulated participants in order to assess the model's ability to calibrate user trust over the task. This is compared against a random and control computational model.
5. The POMDP model performance is evaluated using comparative analyses: results in a significantly higher task accuracy.
    i. If the POMDP leads to a significantly higher task accuracy, the features and parameters included within the model can be considered for implementation in a real world application in order to assess the efficacy and generalisability of the model. This trust calibration assessment is completed with human participants in a laboratory or "in the wild" using a similar methodology (i.e. implementation of training and test batches).
    ii. If the POMDP does not lead to a significantly higher task accuracy, experimenters are encouraged to go back to the first step to run additional human studies investigating different agent features.

9 Limitations and Future Work

The model considered in the present work has been developed for a recommender system which is a decision domain. In this context, the agent only makes a recommendation, with the final action still being taken by the human. With decision domains, the humans' interaction with the agent is characterised by trust and reliance when presented with a recommendation. However, action domains should be distinguished from decision domains [4]. In an action domain context, the agent will be given the opportunity to take action on a task unless the human collaborator intervenes. The human collaborator is required to continuously monitor the agent, with non-action interpreted as reliance and action resulting in intervention to take over control, with examples of this including power plant operation, aircraft autopilot, and self-driving cars. Thus, while the current work provides evidence for the use of POMDP frameworks for decision domains in HAC, the evidence provided is limited in its ability to generalise across all HAC applications.

Additionally, the use of data from the human experiment detailed in Sect. 4 resulted in a limited simulation experiment. While participant behaviour and stimuli were simulated for the current work, the specific agent features used in the previous human experiment were kept consistent. Each feature had a specific influence on user perception and trust in task, with this influence dependent on task difficulty. Moving forward, it would be valuable to introduce additional features with distinct influences on user trust, with both simulated features and features modelled off real-world data offering unique insight. In particular, varying the number of included features with distinct influence, as well as combinations of included features, and assessing the resulting impact on task accuracy to determine a ceiling effect for feature inclusion would be a welcomed contribution to the trust calibration literature for HAC.

Furthermore, the present work had pre-set parameters for agent accuracy and proportion of task difficulty, as defined by the human experiment. Future work will benefit from re-running the simulation experiment varying agent accuracy between 0 and 100% and proportion of task difficulty between 0 and 1. The outcome of these simulations would result in an accuracy matrix that would be a beneficial reference for the implementation of trust calibration within HAC. Finally, while further simulation studies would help to improve the current understanding of POMDP models for trust calibration in HAC, this assessment would benefit from the addition of supplementary evidence from human studies conducted both in a laboratory context and in the wild.

10 Conclusion

The results of the presented simulation experiment provide evidence for the use of a POMDP model for trust calibration during dyadic HAC between a human and a collaborative agent recommneder system. The application of SDT modelling in this framework is novel, offering an innovative way of inferring user trust during HAC to be incorporated in the POMDP model. Step-by-step instructions are provided to apply the experimental methodology to different application domains, offering a beneficial contribution to industry regarding the identification and assessment of potential agent features across a variety of agents and tasks. Taken together, the outcomes of this research are positive and much needed additions to the understanding and development of trust calibration frameworks within HAC contexts.

Author Contributions Sarita Herse and Jonathan Vitale contributed equally to the ideation, development, experimentation, and write-up of this work. Mary-Anne Williams assisted with final review of this work.

Funding Open Access funding enabled and organized by CAUL and its Member Institutions. This work was supported by an Australian Government Research Training Program Scholarship.

Availability of data, materials, and code Relevant datasets, materials, and code generated for the current study are available from the corresponding author on reasonable request.

Declarations

Conflict of interest The authors have no competing interests to declare that are relevant to the content of this article.

Ethics approval Approval for this study was obtained from the Human Research Ethics Committee at the University of Technology Sydney (Ethial approval number: ETH20-5517).

Consent to participate Informed consent was obtained from all individual participants included in the study.

Consent for publication All participants provided consent for their anonymised data to be used in research publications.

References

1. Akash K, Polson K, Reid T et al (2019) Improving human-machine collaboration through transparency-based feedback-part I: human trust and workload model. IFAC-PapersOnLine 51(34):315–321
2. Akash K, Polson K, Reid T et al (2019) Improving human-machine collaboration through transparency-based feedback-part I: human trust and workload model. IFAC-PapersOnLine 51(34):315–321
3. Akash K, Reid T, Jain N (2019) Improving human-machine collaboration through transparency-based feedback-part II: control design and synthesis. IFAC-PapersOnLine 51(34):322–328
4. Akash K, Jain N, Misu T (2020) Toward adaptive trust calibration for level 2 driving automation. In: Proceedings of the 2020 international conference on multimodal interaction, pp 538–547
5. Akash K, McMahon G, Reid T et al (2020) Human trust-based feedback control: dynamically varying automation transparency to optimize human-machine interactions. IEEE Control Syst Mag 40(6):98–116
6. Anderson ND (2015) Teaching signal detection theory with pseudoscience. Front Psychol 6:762
7. Association AP (2020) Apa dictionary of psychology: yes-no task. https://dictionary.apa.org/yes-no-task
8. Bellman R (1954) The theory of dynamic programming. Bull Am Math Soc 60(6):503–515
9. Bellman R (1957) A Markovian decision process. J Math Mech 6(5):679–684
10. Benbasat I, Wang W (2005) Trust in and adoption of online recommendation agents. J Assoc Inf Syst 6(3):4
11. Carr S, Jansen N, Wimmer R et al (2018) Human-in-the-loop synthesis for partially observable Markov decision processes. In: 2018 Annual American control conference (ACC). IEEE, pp 762–769
12. Chandler J, Mueller P, Paolacci G (2014) Nonnaïveté among Amazon Mechanical Turk workers: consequences and solutions for behavioral researchers. Behav Res Methods 46(1):112–130
13. Chavaillaz A, Schwaninger A, Michel S et al (2018) Automation in visual inspection tasks: X-ray luggage screening supported by a system of direct, indirect or adaptable cueing with low and high system reliability. Ergonomics 61(10):1395–1408
14. Chen M, Nikolaidis S, Soh H et al (2018) Planning with trust for human-robot collaboration. In: Proceedings of the 2018 ACM/IEEE international conference on human-robot interaction, pp 307–315
15. Chen M, Nikolaidis S, Soh H et al (2020) Trust-aware decision making for human-robot collaboration: model learning and planning. ACM Trans Hum Robot Interact (THRI) 9(2):1–23
16. Dent S (2017) Tesla driver in fatal autopilot crash ignored safety warnings. https://www.engadget.com/2017/06/20/tesla-driver-in-fatal-autopilot-crash-ignored-safety-warnings
17. Grodzinsky FS, Miller KW, Wolf MJ (2011) Developing artificial agents worthy of trust:"Would you buy a used car from this artificial agent?". Ethics Inf Technol 13(1):17–27
18. Guo Y, Zhang C, Yang XJ (2020) Modeling trust dynamics in human-robot teaming: a Bayesian inference approach. In: Extended abstracts of the 2020 CHI conference on human factors in computing systems, pp 1–7
19. Hancock P, Kessler TT, Kaplan AD et al (2021) Evolving trust in robots: specification through sequential and comparative meta-analyses. Hum Factors 63(7):1196–1229
20. Hebesberger D, Koertner T, Gisinger C et al (2017) A long-term autonomous robot at a care hospital: A mixed methods study on social acceptance and experiences of staff and older adults. Int J Soc Robot 9(3):417–429
21. Herse S, Vitale J, Tonkin M et al (2018) Do you trust me, blindly? Factors influencing trust towards a robot recommender system. In: 2018 27th IEEE International Symposium on robot and human interactive communication (RO-MAN). IEEE, pp 7–14
22. Herse S, Vitale J, Johnston B et al (2021) Using trust to determine user decision making & task outcome during a human-agent collaborative task. In: Proceedings of the 2021 ACM/IEEE international conference on human-robot interaction, pp 73–82
23. Herse S, Vitale J, Williams MA (2023) Using agent features to influence user trust, decision making and task outcome during human-agent collaboration. Int J Hum Comput Interact 39(9):1740–1761
24. Jeong S, Logan DE, Goodwin MS et al (2015) A social robot to mitigate stress, anxiety, and pain in hospital pediatric care. In: Proceedings of the tenth annual ACM/IEEE international conference on human-robot interaction extended abstracts, pp 103–104
25. Jhangiani RS, Chiang I, Price PC (2015) Research methods in psychology, 2nd Canadian edn. BC Campus
26. Kaelbling LP, Littman ML, Cassandra AR (1998) Planning and acting in partially observable stochastic domains. Artif Intell 101(1–2):99–134
27. Kermany D, Zhang K, Goldbaum M (2018) Labeled optical coherence tomography (oct) and chest x-ray images for classification. Mendeley data 2
28. Kermany DS, Goldbaum M, Cai W et al (2018) Identifying medical diagnoses and treatable diseases by image-based deep learning. Cell 172(5):1122–1131
29. Khavas ZR, Ahmadzadeh SR, Robinette P (2020) Modeling trust in human-robot interaction: a survey. In: International conference on social robotics. Springer, pp 529–541

30. Kim Dj, Lim YK (2019) Co-performing agent: Design for building user-agent partnership in learning and adaptive services. In: Proceedings of the 2019 CHI conference on human factors in computing systems, pp 1–14
31. Kunze A, Summerskill SJ, Marshall R et al (2019) Automation transparency: implications of uncertainty communication for human-automation interaction and interfaces. Ergonomics 62(3):345–360
32. Lee MK, Forlizzi J, Kiesler S et al (2012) Personalization in HRI: a longitudinal field experiment. In: 2012 7th ACM/IEEE international conference on human-robot interaction (HRI). IEEE, pp 319–326
33. Leyzberg D, Spaulding S, Scassellati B (2014) Personalizing robot tutors to individuals' learning differences. In: 2014 9th ACM/IEEE international conference on human-robot interaction (HRI). IEEE, pp 423–430
34. Li M, Okamura AM (2003) Recognition of operator motions for real-time assistance using virtual fixtures. In: 11th Symposium on haptic interfaces for virtual environment and teleoperator systems, 2003. HAPTICS 2003. Proceedings. IEEE, pp 125–131
35. Lidstone GJ (1920) Note on the general case of the Bayes-Laplace formula for inductive or a posteriori probabilities. Trans Fac Actuar 8(182–192):13
36. Liu B (2020) A survey on trust modeling from a Bayesian perspective. Wirel Pers Commun 112(2):1205–1227
37. Luckin R, Holmes W, Griffiths M et al (2016) Intelligence unleashed: an argument for ai in education. UCL Knowledge Lab
38. McMahon G, Akash K, Reid T et al (2020) On modeling human trust in automation: Identifying distinct dynamics through clustering of Markovian models. IFAC-PapersOnLine 53(5):356–363
39. Meghdari A, Shariati A, Alemi M et al (2018) Arash: a social robot buddy to support children with cancer in a hospital environment. Proc Inst Mech Eng H 232(6):605–618
40. Michaelsen J (1987) Cross-validation in statistical climate forecast models. J Appl Meteorol Climatol 26(11):1589–1600
41. Min C (2018) Trust and intention in human-robot interaction: a pomdp framework
42. Mindell DA (2015) Our robots, ourselves: robotics and the myths of autonomy. Viking Adult
43. Nam C, Walker P, Li H et al (2019) Models of trust in human control of swarms with varied levels of autonomy. IEEE Trans Hum Mach Syst 50(3):194–204
44. Obuchowski NA (2003) Receiver operating characteristic curves and their use in radiology. Radiology 229(1):3–8
45. Okamura K, Yamada S (2020) Adaptive trust calibration for human-AI collaboration. PLoS ONE 15(2):e0229,132
46. Parasuraman R, Sheridan TB, Wickens CD (2000) A model for types and levels of human interaction with automation. IEEE Trans Syst Man Cybern Part A Syst Hum 30(3):286–297
47. Pineau J, Gordon G, Thrun S et al (2003) Point-based value iteration: an anytime algorithm for pomdps. In: IJCAI. Citeseer, pp 1025–1032
48. Pynadath DV, Wang N, Kamireddy S (2019) A Markovian method for predicting trust behavior in human-agent interaction. In: Proceedings of the 7th international conference on human-agent interaction, pp 171–178
49. Ricci F, Rokach L, Shapira B (2011) Introduction to recommender systems handbook. In: Recommender systems handbook. Springer, pp 1–35
50. Rousseau DM, Sitkin SB, Burt RS et al (1998) Not so different after all: a cross-discipline view of trust. Acad Manag Rev 23(3):393–404
51. Saadatzi MN, Pennington RC, Welch KC et al (2018) Effects of a robot peer on the acquisition and observational learning of sight words in young adults with autism spectrum disorder. J Spec Educ Technol 33(4):284–296
52. Schaefer KE (2016) Measuring trust in human robot interactions: development of the "trust perception scale-HRI". In: Robust intelligence and trust in autonomous systems. Springer, pp 191–218
53. Seymour R, Peterson GL (2009) A trust-based multiagent system. In: 2009 International conference on computational science and engineering. IEEE, pp 109–116
54. Sheridan TB (2019) Extending three existing models to analysis of trust in automation: signal detection, statistical parameter estimation, and model-based control. Hum Factors 61(7):1162–1170
55. Sigaud O, Buffet O (2013) Markov decision processes in artificial intelligence. John Wiley & Sons, Hoboken
56. Staffa M, Rossi S (2016) Recommender interfaces: the more human-like, the more humans like. In: Agah A, Cabibihan JJ, Howard AM et al (eds) Social robotics. Springer International Publishing, Cham, pp 200–210
57. Stanislaw H, Todorov N (1999) Calculation of signal detection theory measures. Behav Res Methods Instrum Comput 31(1):137–149
58. Tjøstheim TA, Johansson B, Balkenius C (2019) A computational model of trust-, pupil-, and motivation dynamics. In: Proceedings of the 7th international conference on human-agent interaction, pp 179–185
59. Wang N, Pynadath DV, Hill SG (2016) The impact of pomdp-generated explanations on trust and performance in human-robot teams. In: AAMAS, pp 997–1005
60. Wang N, Pynadath DV, Hill SG (2016) Trust calibration within a human-robot team: comparing automatically generated explanations. In: 2016 11th ACM/IEEE International conference on human-robot interaction (HRI). IEEE, pp 109–116
61. Wang Y, Humphrey LR, Liao Z et al (2018) Trust-based multi-robot symbolic motion planning with a human-in-the-loop. ACM Trans Interact Intell Syst (TiiS) 8(4):1–33
62. Wang Z, Peer A, Buss M (2009) An HMM approach to realistic haptic human-robot interaction. In: World haptics 2009-third joint EuroHaptics conference and symposium on haptic interfaces for virtual environment and teleoperator systems. IEEE, pp 374–379
63. Wongpiromsarn T, Frazzoli E (2012) Control of probabilistic systems under dynamic, partially known environments with temporal logic specifications. In: 2012 IEEE 51st IEEE conference on decision and control (CDC). IEEE, pp 7644–7651
64. Xu A, Dudek G (2012) Trust-driven interactive visual navigation for autonomous robots. In: 2012 IEEE International conference on robotics and automation. IEEE, pp 3922–3929
65. Xu A, Dudek G (2015) Optimo: online probabilistic trust inference model for asymmetric human-robot collaborations. In: 2015 10th ACM/IEEE International conference on human-robot interaction (HRI). IEEE, pp 221–228
66. Yeh M, Wickens CD (2001) Display signaling in augmented reality: effects of cue reliability and image realism on attention allocation and trust calibration. Hum Factors 43(3):355–365
67. Zorcec T, Robins B, Dautenhahn K (2018) Getting engaged: assisted play with a humanoid robot Kaspar for children with severe autism. In: International conference on telecommunications. Springer, pp 198–207

Sarita Herse holds a PhD in Social Robotics from the University of New South Wales (UNSW) and works as a Research Fellow for UNSW's Creative Robotics Lab and the National Facility for Human-Robot Interaction Research. Her research involves investigating the

intersection of humans and technology, with particular interest in Human-Agent Collaboration.

Jonathan Vitale holds a PhD in Information Technology from the University of Technology Sydney (UTS) and he serves as a lecturer at UTS and at the University of New England. His research covers topics concerning computational models of human cognition applied to AI and social robotics in public environments.

Mary-Anne Williams is the Michael J. Crouch Chair for Innovation at UNSW. She was previously Distinguished Research Professor at UTS and Director of the UTS Magic Lab. Williams focuses on Innovation and works on AI, Robotics and Law, holding a PhD in Computer Science and a Master of Laws.