



Human-Robot Interaction in Search and Rescue: Modeling Victim Behaviour and Trust

How can trust between victims and autonomous rescue robots be computationally modeled and incorporated into a search and rescue simulation to improve evacuation outcomes?

Maria Grouev ¹

Supervisors: A. Jamshidnejad¹, S. Schoonebeek¹

¹EEMCS, Delft University of Technology, The Netherlands

A Thesis Submitted to EEMCS Faculty Delft University of Technology,
In Partial Fulfilment of the Requirements
For the Bachelor of Computer Science and Engineering
June 21, 2026

Name of the student: Maria Grouev
Final project course: CSE3000 Research Project
Thesis committee: A. Jamshidnejad, S. Schoonebeek, L. Cavalcante Siebert

An electronic version of this thesis is available at <http://repository.tudelft.nl/>.

Abstract

Autonomous robots increasingly support Search and Rescue (SaR), yet victim trust in these robots remains largely unaddressed in simulation, where robots typically neither estimate nor respond to it. This paper implements a transparent, bidirectional trust model in a SaR simulation with cognitively realistic victims based on the GT-BDI framework. Victims maintain a true internal trust state driven by observable signals—proximity and perceived danger—while the robot estimates this trust from proximity and switches between approaching the victim to build trust and leading it to the exit. An ablation study over 100 matched seeds compared trust-aware against trust-blind robots. Trust-aware behavior saved significantly more victims (+0.74 on average, $p < 0.001$), reducing victims stranded at timeout rather than lowering mortality. This gain stemmed not from higher trust, but from better robot positioning. These results suggest that how a robot responds to trust can meaningfully improve evacuation outcomes, offering a practical and interpretable path to better robot-guided rescue.

1 Introduction

When disaster strikes, the speed and effectiveness of emergency first response is critical [1]. Victim survival rate when rescued within 24 hours of disaster stands at 74%, whereas after 72 hours the survival rate of extracted victims drops sharply to a mere 22% [1]. Thus, the Search and Rescue (SaR) process is within society’s best interest to develop and improve upon, as it concerns real human lives and safety. The addition of autonomous drones or robots can greatly improve SaR: through increasing speed and efficiency [2], as well as by offering safety by replacing real human first responders when in very dangerous or unknown situations. Although substantial research has been conducted on robotics in SaR (in particular, the exploration phase), the integration of human-robot interaction, and more specifically victim trust in rescue robots, remains largely unaddressed in literature [3].

The dynamics between victims and SaR robots are a less studied field, which raises the question: How can trust between victims and autonomous rescue robots be computationally modeled and incorporated into a search and rescue simulation to improve evacuation outcomes? In order to answer this broad question, several sub questions are explored:

1. **Modeling:** *How can victim trust in a rescue robot be modeled as a dynamic variable, and what factors drive its formation, decay, and recovery?*
2. **Perception:** *How can a rescue robot estimate a victim’s current trust level from observable behavior?*
3. **Robot strategy:** *What robot behaviors or strategies most effectively build and sustain victim trust during a rescue interaction?*
4. **Evaluation:** *How does the level of victim trust in rescue robots affect evacuation time and survival rate?*

This paper investigates the above questions, and is structured as follows. Section 2 reviews previous research, with subsection 2.1 focusing on victim modeling and subsection 2.2 on trust modeling, while subsection 2.3 outlines the current gaps in knowledge. Section 3 presents the methodology, which includes both the victim implementation in subsection 3.1, and the trust dynamics and final rescue mission implementation in subsection 3.2. Results are presented and discussed in Section 4, whose setup is detailed in subsection 4.1, the raw results drawn up in subsection 4.2, with a discussion on them in subsection 4.3. Section 5 includes a discussion on the Ethics of the study (subsection 5.1) and the reproducibility (subsection 5.2). Finally, a conclusion is drawn and presented in Section 6 subsection 6.1, with future recommendations explored in subsection 6.2.

2 Literature Review

2.1 Victim Modeling

What makes modeling SaR scenarios computationally difficult often comes down to the complex nature, behaviors, and decision-making of the human victims involved. Ideally, victims would behave in predictable, logical ways, so as to increase the efficiency of saving them. People, however, are complex and difficult to model mathematically or computationally. Victim modeling has been attempted several times with artificial intelligence [4], but in SaR scenarios, it is important that decision making is transparent and interpretable due to potential life-altering consequences. This transparency criterion makes black-box AI models less desirable for modeling human-robot interaction (HRI) in SaR.

Other more transparent methods have been used, but most fail to account for the illogical tendencies of humans. Several game theory (GT) based frameworks are built on the assumptions that humans act rationally, in their best interest, in order to maximize a certain utility they could gain. In the scope of SaR, this can be seen as the clearcut desire to escape a dangerous situation. However, in reality this rational desire is often undercut by “illogical”, emotion-driven decisions such as staying inside a burning building, putting yourself at harm, in order to search for a family member. If the model is to be realistic, it is important to include this emotional aspect that is very prevalent in high-intensity, emotional situations precisely like SaR. To capture this emotional dimension, [3] makes use of the BDI (Belief-Desire-Intention) framework.

The BDI framework models the cognitive processes and decision making of human (and autonomous) agents. Based on their environment, observations are drawn which are used to form a certain *belief* about the world. Additionally, an agent (or victim) has various *desires* that it would like to see fulfilled, which are more closely tied to its core, innate goals. Based on the combined beliefs and desires, an agent decides on an *intention*, or a plan of action to complete, which hopefully brings the agent closer to their desire. After completing this intention, the state of the environment changes, which can then influence the beliefs and desires of the agent, and a cycle is formed.

This framework has been used in combination with game theory in [3] to model victims in SaR simulations. It is pre-

sented as a more realistic model, as it both takes into account the emotional decision making on an individual level (represented in the BDI model), as well as the rational desires of humans as a cohesive group (due to the game theory aspect).

Game theory (GT) models the conflicts between victims and represents group-level decision making, treating humans as rational agents who act to maximize their utility. In [3], when two victims attempt to move into the same space, GT determines the "winner" who occupies it. Victims are assigned personality types with varying degrees of selfishness, where more selfish types win conflicts against less selfish ones. This models group dynamics, adding realism beyond the individual decision making of BDI. Although every victim has its own desire, in real SaR scenarios group dynamics and collaboration significantly impact the outcome, which GT captures here.

While [3] models victim cognition and group dynamics in detail, and even includes the victims' trust in the evacuation robots, the rescue side of the interaction is simplified. The robots move in fixed or random patterns, and neither estimate nor adapt to a victim's trust. How this trust itself is modeled, and how a robot might respond to it, is the focus of the following subsection.

2.2 Trust Modeling

In a general sense, trust refers to an individual's belief that an agent will aid them, at the cost of accepting vulnerability to that agent's actions [5]. In SaR specifically, this translates to the victim's willingness to believe that the SaR robot will act in their best interest, while simultaneously accepting the vulnerability of placing their safety in the hands of the robot.

Because trust in SaR context deals with human safety, how much a victim trusts the robot could be the difference between life and death. This is not to say, that victims who always blindly trust robots will succeed, but rather an appropriate amount of trust is necessary. If there is not enough trust in the robot, this can result in *disuse*, where the victim refuses to rely on a robot that is in fact capable of guiding them to safety, while excessive trust leads to *misuse*, which refers to failures that occur when the human over-relies on the robot [5]. Optimal safety requires a sweet spot, where the victim's trust matches the robot's actual capabilities.

Importantly, trust is *dynamic* and *asymmetric*: it is lost quickly and regained slowly. [6] showed that a single robot error can result in a large, immediate drop in human trust, whereas to regain the trust requires time and consistency. These findings later motivate the asymmetric update rule used in subsection 3.1, where the parameter for trust loss is much more severe than trust gain.

The factors that shape trust have been studied extensively. Hancock et al. [7] group trust into 3 categories: human-related factors (e.g., personality traits); robot-related factors (e.g., performance); and environmental factors (e.g., danger) based on empirical HRI studies. This categorization is directly reflected in the trust model presented in Section 3.1, where the victim's personality determines a human-related sensitivity parameter, and a combination of the perceived danger of the environment and the robot's ability to lead the victim to safety comprise the input signal.

Beyond identifying factors that influence trust, several works attempt to model trust computationally, so that a robot can estimate it and respond to it during operation [8]. These models vary widely in data requirements and transparency. At the simplest end, trust is calculated with a closed-form update equation, as later presented in subsection 3.1. This representation of trust is fully interpretable and requires no training data, which in the context of SaR where transparency is prioritized and will not often have access to a history of interactions with a victim, is strongly preferred. More sophisticated approaches also exist, such as OPTIMO [9], which treats trust as a hidden state to be inferred, or a Dynamic Bayesian Network trained on users' interaction data, or the trust-POMDP [10], in which the robot plans its actions around its predicted effects on trust. At the far end of this spectrum, data-driven classifiers detect trust from physiological signals such as facial expressions or heart rate [8], which unfortunately would not work in SaR where visibility may be low and victims cannot be expected to be wearing any monitoring equipment. Essentially, as you move along the spectrum, you generate a more sophisticated model of trust at the cost of more required data and a lack of transparency, particularly with black-box models. The model developed in this work therefore belongs to the simplest family: an algebraic update equation. It is less expressive than the probabilistic and data-driven alternatives, but in return it is fully transparent, and only requires information a rescue robot can realistically observe.

A concrete example of such an algebraic model comes from [3], where each victim's trust evolves through a saturated first-order difference equation: each time step the trust value is incremented or decremented, then clipped between 0 and a per-victim maximum reflecting that individual's propensity to trust. The sign and magnitude of the update depend on the most recent interaction: a helpful robot raises trust gradually, whereas a faulty robot drops it sharply, scaled to the severity of the mistake. This formulation is fully transparent, requires no training data, and naturally captures the asymmetry described above. Its input signal, however, derives from the robot's internal *failure class* – ground-truth information the simulation can access but a victim realistically cannot. The model adopted in this work (subsection 3.1) keeps the same algebraic, saturated structure but replaces the input with a signal a victim could plausibly observe, such as the perceived danger of their surroundings or their proximity to the robot.

2.3 Research Gap and Contributions

Despite significant research on trust in HRI, there still remain gaps in the knowledge, specifically within the application of SaR. The aforementioned computational trust models were developed with a certain setting in mind: cooperative, calm humans engaged in a shared task with the robot toward a common goal. None of them target a disaster victim, whose cognition is highly panic impaired. While [6] investigate victim-like trust behavior, no dynamic trust model is implemented in a simulation to evaluate rescue strategies.

The other side of this gap lies on the rescue side of the interaction. As established in subsection 2.1, existing SaR simulations with realistic victim cognition, such as the GT-BDI

framework of [3], do model the victims’ trust in the robots, but only in one direction. The robots do not estimate or respond to victim trust, and instead move in a fixed or random pattern. The trust relationship is unidirectional, only being modeled on the victim’s side, while the robots behavior remains trust-blind. To date, no SaR simulation couples a cognitively realistic victim with a robot that both estimates and responds to the victim’s trust.

This paper addresses the intersection of these gaps: it implements a transparent, dynamic trust model in a SaR simulation with cognitively realistic victims, and does so *bidirectionally*, where victims maintain a true internal trust state (subsection 3.1) while the robot estimates the victim’s trust and accordingly changes its guidance strategy (subsection 3.2). This uses trust as a hidden state the robot must estimate, but with simple, transparent equations instead of a black box. Crucially, the contribution is not a new victim model: the victim cognition builds directly on the GT-BDI framework, and the trust update adapts its algebraic formulation with observable input. The novelty lies in closing the loop, by making the robot’s behavior trust-aware, and in doing so within a transparent, simulation-based setting where the effect of victim trust on evacuation time and survival rate can be measured directly.

3 Methodology

To answer the research question, it was necessary to implement a simulation that would run a SaR scenario. This simulation consists of a 2-dimensional testbed with various actors and events that play out over a number of discrete time steps. The details of the setup of this simulation are explained in my colleague’s paper [11]. This next section presents this paper’s individual contributions, specifically the rescue protocol and HRI interaction within.

3.1 Victim Modeling

In order to observe the human robot interaction between victims and agents, it was necessary to accurately model the victims as close as possible to a person in such a disaster scenario, while also taking into account limitations such as computational power, time, and economic constraints. As [3] established, it was important to find a balance between the rationality of humans yearning to escape a dangerous situation, and the irrational emotional decision making at play in emergencies. The victim model closely followed the GT-BDI model from [3].

Particularly, the personality system and GT based conflict resolution are directly inherited from [3], which categorizes victims along a selfishness scale and uses these personality types to resolve conflicts over contested cells. There are however two differences: firstly, only three personalities were implemented instead of 4 – specifically, the selfless type from [3] was dropped, as it was very similar to the altruist type and was not essential for studying trust-driven rescue; secondly, a trust-sensitivity parameter was attached to each type, which is original to this work and links personality to the trust dynamics introduced in subsection 2.2. The three personality types and their behaviors are summarized in table 1.

Table 1: Victim personality types and their behaviours.*When two victims of the same type conflict, the winner is chosen at random.
†Higher values mean the victim’s trust responds more strongly to robot interactions; see Equation 1.

Personality	Behaviour	Conflict behaviour*	Trust sens.†
Altruist	Seeks out and groups with nearby victims, then heads for the exit	Cedes the contested cell to egoist and selfish types	1.2
Egoist	As altruist	Cedes to selfish types and becomes selfish after the encounter; wins against altruists	1.0
Selfish	Ignores others and moves directly toward the exit	Wins against altruists and egoists	0.6

Additionally, victims can be in one of three age groups, whose behaviors are summarized in table 2. Because the simulation is modeled in discrete time steps, speed is represented as a probability that a victim moves in one tick. The adult speed is taken as a baseline of 0.9, and the elderly and child speeds are scaled to preserve their real-world ratios to the adult speed: $1.04/1.45 \approx 0.72$ gives $0.9 \times 0.72 \approx 0.65$ for the elderly, and $0.84/1.45 \approx 0.58$ gives $0.9 \times 0.58 \approx 0.525$ for children (Table 2). The adult baseline is set to 0.9 rather than 1 so that their speed can still be subject to a panic multiplier described below.

In disaster scenarios, a victim’s speed is influenced by panic and fear. According to [12], speeds during evacuations can increase by 50-60%, which is modeled in the simulation through a panic multiplier of 1.5 (and is capped at 1 as a probability). However, victims can also often freeze during disaster due to panic. For simplicity’s sake, only the increase in speed was modeled rather than the freeze as well, which limits how realistic this model is. Note that because adult and elderly panicked speeds both approach the cap of 1, the panic multiplier compresses the speed difference between age groups – a simplification, since in reality panic does not equalize mobility.

Table 2: Walking speeds by age group and their corresponding simulation move probabilities.

Age Group	Average real walking speed	Simulation walking speed	Panicked speed
Adult	1.45 m/s [13]	0.9	1
Elderly	1.04 m/s [13]	0.65	0.975
Children	0.84 m/s [14]	0.525	0.7875

A limitation of such a discrete categorization of victim types is of course a decrease in realism. In the real world, a young teenager might have a walking speed between the child and adult categories modeled here, but the economic

and computational cost of modeling every possible walking speed or personality type outweighs the extra realism that could be achieved.

During one tick, a victim acts according to Algorithm 1. The key steps are the trust update and trust-gated intention formation, described below. Otherwise, the algorithm also includes danger perception (see Table 4), a stochastic speed gate (the implementation of Table 2), and the end-of-tick scan, which returns the current perceived state matrices.

Algorithm 1 Victim GETACTION

```

1: UPDATETRUST() ▷ Eq. 1; also resets  $\tau$  if robot or victim
   stuck  $\geq 10$  ticks
2: inDanger  $\leftarrow$  PERCEIVEDANGER()
3:  $p_{\text{move}} \leftarrow \min(1, p_{\text{base}} \cdot (\alpha_{\text{panic}} \text{ if inDanger else } 1))$ 
4: if  $U(0, 1) > p_{\text{move}}$  then
5:   return WAIT
6: end if
7: if  $\tau \geq \tau_{\text{threshold}}$  and robot is visible then
8:   return STEPTOWARDS(robot position)
9: end if
10: if inDanger or personality = SELFISH then
11:   return MOVETOWARDEXITSAFELY()
12: else
13:   return WANDERORSEEK()
14: end if

```

Trust update. Using the perception scanned in the *previous* tick, the victim updates its trust in the nearest visible robot according to Equation 1, whose parameters are summarized in Table 3. The input $\sigma(k)$ combines two signals: a proximity term rewarding the robot for staying close by, and a danger term that rewards improving local danger but penalizes worsening danger three times as steeply. This 3:1 asymmetry encodes the fast-erosion/slow-recovery property established in subsection 2.2. The danger term is calculated as a safety score following the rules in Table 4. Two further conditions erode trust when the robot fails to make progress: if the robot has not moved for 10 ticks, its trust is reset to 0, and if the victim itself is stuck for 10 ticks while following, trust is dropped to 0.2. Both reflect a victim losing faith in a robot that is no longer effectively guiding them.

$$\tau_i(k+1) = \max\{0, \min\{\tau_i(k) + \delta_i \sigma(k), \tau_i^{\max}\}\} \quad (1)$$

Intention formation. If trust exceeds $\tau_{\text{threshold}} = 0.4$ and a robot is visible, the victim defers entirely to it and steps toward it, overriding personality, danger, and grouping. The threshold was set just below the victim’s initial trust $\tau_{\text{init}} = 0.5$, so victims begin in a trusting, following state. This choice was motivated by [15]’s findings that victims tend to overtrust robots in SaR scenarios. Victims defer entirely to the robot’s lead in order to reflect the view that, in trusting the robot, the victim places its safety in the robot’s hands; a finer model in which personality still impacts following is left to future work. When trust drops below the threshold, the victim’s personality type and local danger dictate the intention;

Table 3: Trust-model parameters for the victim (Equation 1) and the robot’s estimate (Equation 2). Subscript i denotes per-victim quantities; k is the discrete tick.

Symbol	Value	Description
<i>Victim trust (Eq. 1)</i>		
$\tau_i(k)$	—	Trust of victim i in the robot at tick k ; the state variable updated by Eq. 1
τ_{init}	0.5	Initial trust assigned to every victim at the start of the simulation
$\tau_i^{\max}$	1.0	Per-victim maximum (saturation ceiling) on trust
$\tau_{\text{threshold}}$	0.4	Follow threshold; above it the victim defers to the robot
δ_i	1.2 / 1.0 / 0.6	Personality-based trust sensitivity (altruist/egoist/selfish); scales the update magnitude (Table 1)
$\sigma(k)$	—	Composite input signal driving the victim update; sum of the proximity and danger terms below
w_{prox}	0.05	Weight on the proximity term; rewards the robot closing distance to the victim
w_{gain}	0.10	Weight on improving local danger (trust recovery)
w_{loss}	0.30	Weight on worsening local danger (trust erosion); $3 \times w_{\text{gain}}$, encoding the fast-loss/slow-gain asymmetry
<i>Robot estimate (Eq. 2)</i>		
$\hat{\tau}(k)$	—	Robot’s estimate of the victim’s trust at tick k
$\hat{\tau}_{\text{init}}$	0.0	Initial estimate; conservative “assume no trust until shown” prior
$\Delta d(k)$	—	Change in robot–victim distance, $d(k-1) - d(k)$; positive when the victim moves closer
σ_{robot}	0.1	Update rate of the robot’s estimate

a selfish or in-danger victim moves greedily exitward, while a safe egoist or altruist seeks others to group with.

Table 4: Cell safety scores as computed by `_cell_safety_score`. A victim perceives itself as in danger when its current cell scores ≤ 2 .

Score	Fire level	Vulnerability	Description	In danger
−1	—	—	Impassable (wall or out of bounds)	—
0	Burning/Burnt	any	Cell is on fire	✓
1	< Burning	High risk	Structurally critical	✓
2	Flammable	< High risk	Flammable, not yet burning	✓
3	Safe	Vulnerable	Structurally at risk	×
4	Safe	Safe	No hazard	×

All together, when several victims are placed in the simulation upon generation, they are randomly assigned some combination of personality type and speed, while any local danger further affects their actions. It’s assumed that all victims are aware that the exit is in the top left corner (0,0), however, in

their states of panic and disarray they are not fully capable of reaching the exit, or don't remember how to. So, without the help of a rescuing agent, the victims try greedily to move up and to the left in order to reach the exit, but will often fail if they are in a room or a hallway that does not favor simply going up and left. This realistically models disoriented victims trying to escape disaster scenarios, who may either be unfamiliar with their surroundings (if they are not frequent visitors of the building) or whose visibility may be impaired by factors like smoke and dim light. So, it is up to the robot to establish a connection with the victim, gain their trust, and intelligently lead them to the exit.

3.2 Agent trust modeling

Human-robot dynamics are complex in the real world. In dangerous or life-threatening situations, it can be hard to see outside of the adrenaline-fueled fight for survival, especially when a human is lost in an area with no knowledge of how to escape. We rely on the autonomous agent to lead the human to safety. However, every person does not always automatically trust a robot or drone. It is important that the agents can accurately model the trust and act accordingly in order to gain trust. The rescue agent therefore operates in two distinct modes depending on its estimate of how much the victim trusts it.

The rescue agent is assigned to a specific victim by the supervisor and goes through two phases. In the **search phase**, the agent explores the map and searches for victims. Upon finding a victim (which for scan purposes means being within 3 cells of one), it notifies the supervisor and transitions to the **rescue phase**, initiating the rescue protocol as described in this subsection.

During the rescue phase, the agent switches between two behavioral modes as supported by Chen et al. [16]. [16] offers that the rescue robot should always remain in motion, either actively leading the victim towards the exit, or moving closer towards the victim so as to establish deeper trust. This simulation follows this ideology. The rescue agent switches between the two behavioral modes based on its trust estimate $\hat{\tau}$, updated each tick as:

$$\hat{\tau}(k+1) = \max(0, \min(1, \hat{\tau}(k) + \sigma_{\text{robot}} \cdot \Delta d(k))) \quad (2)$$

This estimate is driven by the *change* in distance $\Delta d(k)$ (Table 3) rather than the absolute distance, because trust in the victim model is built by the robot *closing* distance (the proximity term of $\sigma(k)$), not by the gap itself. A victim may be far yet approaching, or close yet retreating; only the change in distance captures whether the interaction is currently building or eroding trust, so Δd aligns the robot's estimate with the signal that actually drives the victim's trust. When the victim is stationary ($\Delta d = 0$) but remaining near a following robot, the estimate would otherwise freeze; a small fixed nudge of $\Delta d = 0.01$ is therefore applied, so sustained co-location still gradually reinforces the estimate. This nudge is a minor modeling choice rather than a tuned parameter.

The robot's estimate (Eq. 2) is deliberately simpler than the victim's trust equation 1, in two regards. First, the input signal is a singular (observable) signal – proximity – as the robot

in this simulation does not have access to more complex signals, such as physiological ones. Secondly, although a more sophisticated state estimator could be used (that additionally tries to infer victim personality or sensitivity δ_i), such a state estimator is beyond the scope of this paper, therefore proximity is treated as a sufficient input signal.

Furthermore, an asymmetry is introduced: while the robot's estimate is initialized at $\hat{\tau} = 0$, the victim's true trust starts at $\tau_{\text{init}} = 0.5$. This was intentionally implemented as an "assume no trust until shown" prior on the robot's side. Because the robot cannot observe the victim's true internal trust state, it errs on the side of safety and assumes none.

Mode 1 - Approach ($\hat{\tau} < 0.4$): the robot moves towards the victim in order to build trust through proximity. A sweet-spot strategy is used: if the robot is already within the victim's perception radius, it moves exitward along the BFS path rather than crowding the victim. In this way, the robot can directionally hint the route to the exit, while still remaining visible to the victim and thus gaining its trust.

Mode 2 - Lead ($\hat{\tau} \geq 0.4$): the robot follows the shortest path to the exit derived from a Breadth First Search (BFS) map, where each cell stores the minimum number of steps needed to reach $(0, 0)$. The BFS is rebuilt from the robot's current perception every tick, so newly scanned areas are immediately incorporated. The robot is prevented from outpacing the victim as well – if the victim falls more than 4 cells behind (outside of the robot's scan radius), the robot waits. In this way, it keeps the victim within sight at all times, as supported by [16].

Once the victim reaches the 3×3 exit zone at the origin, it is marked as rescued, removed from the simulation, and the rescue agent reports completion to the supervisor and returns to idle. Alternatively, victims are marked as dead if they pass through a cell that is actively burning, and they are marked as leftover if they fail to be rescued before the simulation times out yet remain alive.

4 Results

4.1 Setup

To isolate the effect of trust-aware behavior, an ablation test compared the robot designed in subsection 3.2 against a robot without trust adaptation, one that always leads regardless of its estimated trust of the victim. This aimed to observe how the trust-aware behavior (specifically, the adaptive switching between Approach mode and Lead mode based on trust estimation) affects both the final evacuation outcomes and individual trust-related metrics within the run.

First, a configuration setup had to be decided upon. For the sake of realism, a simple setup with only 1 robot and victim was avoided, as it would lose the victim realism developed in subsection 3.1. The GT-BDI victim modeling manifests itself in larger groups of victims, displaying realistic group dynamics with varying individual desires and actions. Thus, a large map of size 45×45 cells was chosen, with 10 victims dispersed across it. It was also important to choose an optimal number of robots, so before beginning the ablation studies a test was run comparing the impact of the number of robots on the evacuation outcomes. These tests ran as follows: first, a

random seed was generated. Then, for each quantity of robots $n = (0, 2, 4, 6, 8)$, a map was generated from this seed and n robots were placed at the entrance. The simulation was run for this map and robot configuration, recording victims saved, dead, and leftover at the end of a 300 step simulation. The seed was then incremented and the process repeated. Each robot quantity was run on the same 100 consecutive seeds (the first randomly generated) for matched results. Note that the same seed always generates the same map setup, including traversability, fire spread, and victim generation and placement. However, once the robots are set in action, the simulations become unique to each robot count, as changing the number of robots affects victim movement. This is not a problem; importantly, the initial variance in map generation is accounted for by matching seeds across test arms.

The results of this test are described in detail in the following subsection 4.2. Ultimately 4 robots were selected, as the point at which the marginal benefit of adding more robots had substantially diminished (Subsection 4.2). This isolates the trust effect at a realistic level without over saturating the map with robots. Other minor configuration details are displayed in Appendix A.1.

With the configuration set, the ablation test was run as follows: two arms were established, one in which the robot behaves as in subsection 3.2, and another in which it is always in lead mode regardless of estimated trust. As before, each arm ran on the same 100 consecutive seeds (starting from a randomly chosen seed and incremented thereafter) for matched results, accounting for variance caused by the wide variety of map generation possibilities. The recorded results consisted of both evacuation outcomes – total victims saved, dead, and leftover after the 300 step simulation – as well as trust-specific metrics as defined in table 5.

4.2 Results

The results of the initial test comparing the effect of the quantity of robots on the evacuation outcomes are plotted in Figure 1. The raw means, standard deviations, and confidence intervals for victims saved, dead, and leftover per robot count are displayed in Tables 9, 10, and 11 in Appendix A.2. A quadratic regression was fit across the five robot-counts per seed, and these per-seed fits were then averaged, resulting in the following equation (also plotted in Figure 1):

$$\hat{y} = 1.899 + 0.440r - 0.031r^2 \quad (3)$$

where r is the number of robots and $\hat{y}$ is the predicted number of victims saved.

Importantly, the quadratic coefficient’s negative sign ($-0.031, p < 0.001$) indicates diminishing returns; that is, as the number of robots increases, the marginal benefit decreases. The low p value further indicates this effect is statistically significant rather than noise.

These results motivate the choice of 4 robots in the final ablation studies (as described in subsection 4.1). According to equation 3, the marginal gain (of average victims saved) decreases with each additional pair: approximately $+0.76$ for $0 \rightarrow 2$, $+0.51$ for $2 \rightarrow 4$, $+0.26$ for $4 \rightarrow 6$, and $+0.01$ for $6 \rightarrow 8$. This plateau is consistent with [3]’s claim that at a certain point, increasing the number of robots does not

Table 5: Trust-related metrics recorded during the ablation study. Outcome metrics (victims saved, dead, and leftover) are reported separately. All per-victim metrics are averaged over the victims in a run, then over seeds; estimation metrics (mean estimation error, mean signed error, and the over- and under-trust fractions) are instead pooled over all active-rescue ticks before averaging over seeds.

Metric	What it measures
Final trust	The victim’s true trust τ at the moment it leaves the simulation (rescued, dead, or timed out).
Average trust	The victim’s true trust τ averaged over every tick of the run, capturing the trust level sustained throughout rather than only at the end.
Trust drops	The number of ticks on which a victim’s trust decreased relative to the previous tick, measuring how frequently trust eroded over a run.
Follow fraction	The fraction of a victim’s active ticks spent following the robot, out of all ticks it was alive.
Mean estimation error	The mean absolute gap $ \hat{\tau} - \tau $ between the robot’s estimated trust and the victim’s true trust, over all ticks of active rescue; measures how accurately the robot reads trust.
Mean signed error	The mean signed gap $\hat{\tau} - \tau$ over active-rescue ticks. Positive values indicate the robot <i>over-trusts</i> (estimate above truth); negative values indicate it <i>under-trusts</i> .
Over-trust fraction	The fraction of active-rescue ticks on which the robot over-trusted the victim ($\hat{\tau} > \tau$).
Under-trust fraction	The fraction of active-rescue ticks on which the robot under-trusted the victim ($\hat{\tau} < \tau$).

marginally impact the evacuation outcomes. Additionally, although the mean saved victims at six robots is marginally higher than at four (3.35 vs 3.08), their confidence intervals overlap substantially (table 9), so the difference is not statistically distinguishable. Thus, 4 robots were used in the ablation studies.

The results of the ablation studies are presented in table 6, which shows the evacuation outcomes, and table 7, which shows the specific trust metrics. All comparisons in both tables between the aware arm and blind arm are paired across the 100 shared seeds (subsection 4.1), and reported p -values come from a paired t-test.

In regard to the evacuation outcome, the trust-aware robot saved on average significantly more victims ($+0.74$, 95% CI $[0.51, 0.96]$, $p < 0.001$). Importantly, this increase corresponds to a decrease in the amount of victims left (-0.66 , 95% CI $[-0.89, -0.42]$, $p < 0.001$) as opposed to any change in the count of victims dead (-0.08 , 95% CI $[-0.27, 0.10]$, $p > 0.05$). Furthermore, because the confidence interval of the deaths count includes 0, the difference is not statistically distinguishable from no difference.

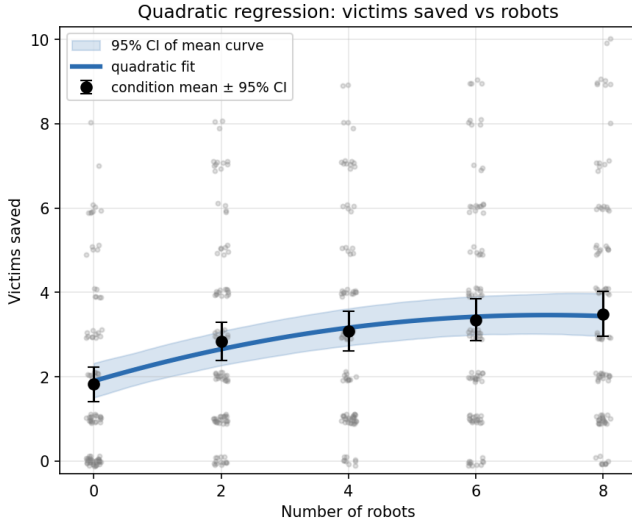


Figure 1: Mean victims saved as a function of the number of rescue robots, across 100 matched seeds. The fitted quadratic (Eq. 3) shows diminishing returns: the marginal benefit of each additional robot decreases, approaching zero beyond roughly four robots.

Table 6: Paired comparison of outcome metrics between the trust-aware and trust-blind conditions. Differences are trust-aware minus trust-blind, with 95% confidence intervals and paired t -test p -values.

Metric	Aware	blind	Diff	95% CI	p
Victims saved	3.08	2.34	+0.74	[0.51, 0.96]	<0.001
Victims dead	2.05	2.13	-0.08	[-0.27, 0.10]	0.402
Victims left	4.87	5.53	-0.66	[-0.89, -0.42]	<0.001

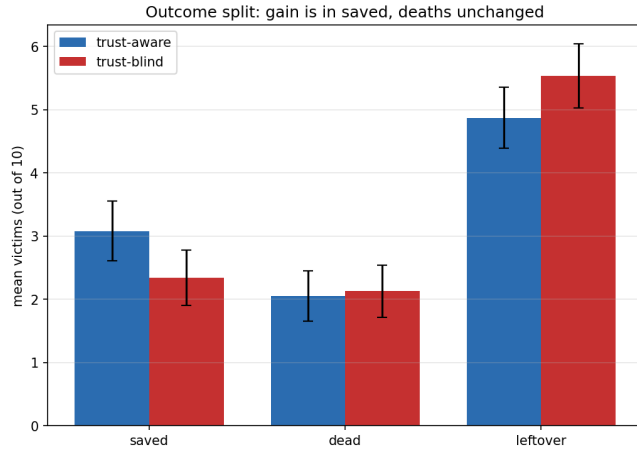


Figure 2: Mean evacuation outcomes (victims saved, dead, and left behind, out of 10) for the trust-aware and trust-blind conditions, with 95% confidence intervals. The trust-aware robot saves significantly more victims and strands fewer, while the number of deaths is statistically indistinguishable between conditions, indicating the benefit lies in evacuation completion rather than reduced mortality.

Despite the difference in evacuation outcome, trust levels did not differ substantially between the two arms. Both final

trust (+0.004) and average trust (+0.014) have minimal differences between the trust-aware arm and the trust-blind arm, with p values well above 0.05. The follow fraction was also somewhat similar between the two arms, with a difference of -0.030 , and a p value on the border, with $p = 0.051$.

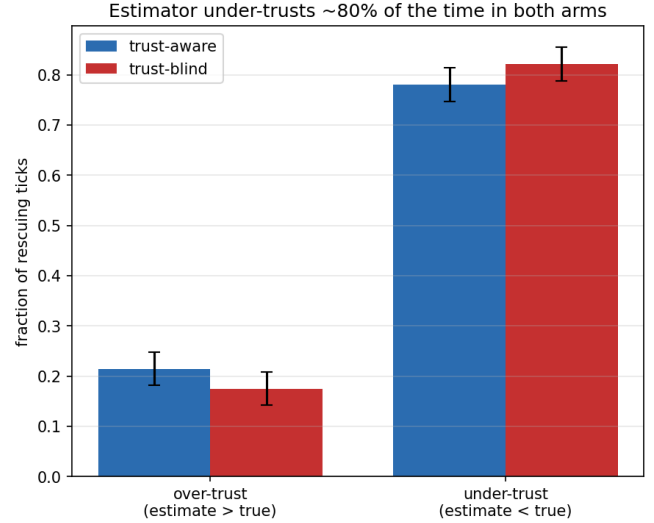


Figure 3: Fraction of active-rescue ticks on which the robot over-trusted (estimate above the victim’s true trust) and under-trusted (estimate below true trust), for the trust-aware and trust-blind conditions, with 95% confidence intervals. In both conditions the robot under-trusts on roughly 80% of ticks and over-trusts on roughly 20%, indicating a consistent under-trust bias that is a property of the estimator itself rather than of the robot’s guidance policy.

Interestingly, the difference in trust drops per arm was significantly different, where the aware arm received on average 0.837 more trust drops per victim per simulation. The $p < 0.001$ indicates a strong statistical difference. Furthermore, both the overtrust fraction and undertrust fraction differ by roughly 0.04 between arms. Their p values (0.049 and 0.044 respectively) border 0.05, so although there may be a marginal difference between arms, this claim is shaky. More interestingly, both arms consistently undertrust – 78% of the time in the Aware arm, and 82% of the time in the blind arm. This tendency to undertrust is consistent with the mean signed error of -0.26 for each arm, whose negative sign also points towards a tendency to undertrust.

Table 7: Paired comparison of trust metrics between the trust-aware and trust-blind conditions, with 95% confidence intervals and paired t -test p -values.

Metric	Aware	blind	95% CI	p
Final trust	0.503	0.500	[-0.017, 0.025]	0.724
Avg. trust	0.511	0.497	[-0.010, 0.039]	0.278
Trust drops	1.074	0.237	[0.626, 1.062]	<0.001
Follow fraction	0.435	0.465	[-0.060, -0.000]	0.051
Mean est. error	0.351	0.355	[-0.025, 0.018]	0.749
Mean signed error	-0.257	-0.266	[-0.027, 0.044]	0.651
Overtrust fraction	0.215	0.175	[0.001, 0.079]	0.049
Undertrust fraction	0.780	0.822	[-0.081, -0.002]	0.044

4.3 Discussion

Trust-aware robots significantly improve the evacuation outcome in SaR, by increasing the average count of victims saved by +0.74. In fact, the increase in saved victims by adding trust-aware behavior on the robot side is more than the increase achieved by doubling the robot count: 4 trust-blind robots save on average a mere 2.34 victims, while 2 trust-aware robots save on average 2.84 victims. It is however important to note that, the robots improve *evacuation completion*, not *mortality*. While the saved count does significantly increase, the count of dead victims remains the same.

Interestingly, this improvement in evacuation outcome is not a direct result of increased trust on either side. Average trust, final trust, and following fraction remain relatively similar across the two arms. This poses the question: why does the evacuation outcome then improve? This likely has to do with better robot positioning. Although not directly measured in this experiment, the pattern is consistent with the trust-aware robot maintaining better relative position rather than higher victim trust. The trust-blind robot is likely to outpace the victim and abandon or lose it, while the trust-aware robot keeps pace with the victim by switching between lead and approach mode. This mode-switching helps the trust-aware robot act as a better guide. These findings are consistent with [16], who frame evacuation guidance as a trade-off between leading the victim toward the exit and remaining within their viewpoint, and show that balancing the two improves evacuation time.

Although the aforementioned trust metrics do not change, the number of trust drops does significantly change between arms. The increased trust drops in the Aware arm (+0.837) may seem negative at first, but the metric in fact reflects overall interaction. The trust-blind robot always leads and does not stop to wait for the victim, so it does not linger in the dangerous situations where trust is likely to drop. The trust-aware robot, by contrast, enters approach mode and stays by the victim through danger, where trust initially drops – so it makes sense that the victim interacts with and loses trust more with the trust-aware robot.

Another interesting metric to evaluate is the overtrust/undertrust ratio. Although between arms there is not a very significant difference, both arms do consistently undertrust. This tendency is slightly undesired, as it increases inefficiency: the victim already trusts the robot, but time is wasted in approach mode nonetheless. However, this is preferable over consistent overtrust. An undertrust bias is the safe failure direction – although it is more inefficient, it is essentially harmless. An overtrust bias risks stranding and abandoning a non-following victim. In fact, an overtrusting robot is likely to enter lead mode prematurely and more often, effectively acting like a trust-blind robot, which we have shown performs worse.

There are still some limitations – because victims are initiated with a trust level higher than the threshold (subsection 3.1), the results track trust maintenance as opposed to trust building, and likely understate the benefits trust-awareness could pose for skeptical, untrusting victims. Another limitation lies in the simplifications made in the model, as explained in subsection 3.1.

5 Responsible Research

5.1 Ethics

Search and Rescue is a field deeply entrenched in ethics, and throughout this project several ethical considerations were not only taken into consideration but served as primary driving motivations behind decisions. In particular, there is a great need for transparency in SaR. When using “intelligent” robots to help rescue victims, it is of utmost importance that human first responders are fully aware of why the robot makes certain decisions, which necessitates the white box model of trust and guidance proposed.

Additionally, the project as a whole is well-suited to the early stages of SaR research, because the research was conducted in a virtual simulation. As such, there was room to test the robot in all stages of development, without worrying about harming real live people as it was developed.

5.2 Reproducibility

In order to ensure reproducibility of this paper’s findings, the entire process was outlined transparently, and all decisions were motivated and explained. The entire repository for the simulation itself can be found in the github link in Appendix A.3 for reproducibility purposes. Furthermore, Section 4.1 details clearly how the results were extracted and calculated. Because seeds and configuration settings were included, the entire experiment can be reproduced using the currently pushed testbed at time of writing.

6 Conclusions

6.1 Conclusion

This paper investigated whether victim trust can be computationally modeled to improve evacuation outcomes in Search and Rescues, and whether trust-aware robot behavior can further help. Building on the GT-BDI victim model of [3], it contributes a transparent, bidirectional trust loop: victims maintain a true internal trust state, while the rescue robot estimates that trust from proximity and switches between leading the victim and approaching to rebuild trust, combining the victim-side and robot-side trust modeling that prior work addressed only separately.

Trust-aware robots significantly improved the evacuation outcomes, saving on average an extra 0.74 victims, a larger gain than doubling the number of robots. This improvement was in evacuation *completion* rather than *mortality*; the additional rescued victims were not saved from death, but from the survivors still stranded in the simulation after timeout. Furthermore, this outcome was not a direct result of increased victim trust as initially assumed; rather, the trust-aware robot positioned itself better and kept pace with the victim, guiding it to the exit more efficiently. The robot also consistently under-estimated victim trust, which, while inefficient, is the safe failure direction for a safety-critical system, since it errs towards staying with the victim as opposed to abandonment. Together, these findings suggest that how a rescue robot interprets and responds to trust can matter as much as the size of the rescue fleet, and that transparent, dynamic trust models offer a practical and interpretable means of improving robot-guided evacuation.

6.2 Future Recommendations

These findings open several directions for further work, the most immediate of which follows from a limitation of the present setup. Because victims initialize with a trust level above the follow threshold (subsection 3.1), the paper studies the maintenance of trust more than the initial build up. As such, a first point worth exploring would be to run tests with a lower initial trust on the victim's end. It would be interesting to both study trust formation, as well as to see if the under-trust in the model continues even with a more balanced initiation.

Secondly, it would be very interesting to test with different map sizes. The size and layout of the map is very important, because on a practical level it can reveal insights on construction layouts for building with evacuation in mind. Such a simulation as developed in this paper can be used on a larger scale to help build with Search and Rescue in mind. Beyond testing with a larger scale, it would also be helpful to pre-categorize maps according to difficulty (e.g., based on fire, vulnerability) and run tests comparing evacuation outcomes with different map difficulties. This has a dual benefit: practically, it further informs construction planning with safety in mind, and methodologically, it would help separate the effect of trust-aware behavior from the noise introduced by differing map layouts.

References

- [1] Y.-Y. Chiu, H. Omura, H.-E. Chen, and S.-C. Chen, "Indicators for post-disaster search and rescue efficiency developed using progressive death tolls," *Sustainability*, vol. 12, no. 19, p. 8262, 2020. DOI: 10.3390/su12198262.
- [2] M. H. Mugo, "Robotics in disaster response: Enhancing search and rescue operations," *Research Inventon Journal of Biological and Applied Sciences*, vol. 3, no. 3, 2024, ISSN: 1115-6171. [Online]. Available: <https://rijournals.com/biological-and-applied-science/>.
- [3] L. Wilmes and A. Jamshidnejad, "GT-BDI model: A combined game-theoretic and BDI-based computational model for emergency evacuation with search and rescue robots," *IEEE Access*, 2025, Published 28 March 2025. DOI: 10.1109/ACCESS.2025.3555870.
- [4] M. L. M. Patricio and A. Jamshidnejad, "Dynamic mathematical models of theory of mind for socially assistive robots," *IEEE Access*, vol. 11, pp. 103 956–103 975, 2023. DOI: 10.1109/ACCESS.2023.3316603.
- [5] J. D. Lee and K. A. See, "Trust in automation: Designing for appropriate reliance," *Human Factors*, vol. 46, no. 1, pp. 50–80, 2004. DOI: 10.1518/hfes.46.1.50.30392.
- [6] P. Robinette, A. M. Howard, and A. R. Wagner, "Timing is key for robot trust repair," in *Social Robotics (ICSR 2015)*, ser. Lecture Notes in Computer Science, vol. 9388, Springer, 2015, pp. 574–583. DOI: 10.1007/978-3-319-25554-5_57.
- [7] P. A. Hancock, D. R. Billings, K. E. Schaefer, J. Y. C. Chen, E. J. de Visser, and R. Parasuraman, "A meta-analysis of factors affecting trust in human-robot interaction," *Human Factors*, vol. 53, no. 5, pp. 517–527, 2011. DOI: 10.1177/0018720811417254.
- [8] Z. R. Khavas, S. R. Ahmadzadeh, and P. Robinette, "Modeling trust in human-robot interaction: A survey," in *Social Robotics (ICSR 2020)*, ser. Lecture Notes in Computer Science, vol. 12483, Springer, 2020, pp. 529–541. DOI: 10.1007/978-3-030-62056-1_44.
- [9] A. Xu and G. Dudek, "OPTIMO: Online probabilistic trust inference model for asymmetric human-robot collaborations," in *Proceedings of the 10th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, 2015, pp. 221–228. DOI: 10.1145/2696454.2696492.
- [10] M. Chen, S. Nikolaidis, H. Soh, D. Hsu, and S. Srinivasa, "Planning with trust for human-robot collaboration," in *Proceedings of the ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, 2018, pp. 307–315. DOI: 10.48550/arXiv.1801.04099.
- [11] T. R. Schutgens, "Coordination-aware task allocation via a lightweight supervisory layer: An event-driven approach to multi-agent search and rescue," EEMCS Faculty, supervised by A. Jamshidnejad and S. Schoonebeek, Bachelor's Thesis, Delft University of Technology, Jun. 2026.
- [12] J. Yan, G. He, A. Basiri, C. Hancock, and S. K. Yeboah, "The impact of postures and moving directions in fire evacuation in a low-visibility environment," *Sensors*, vol. 24, no. 5, 2024. DOI: 10.3390/s24051378.
- [13] A. Forde and J. Daniel, "Pedestrian walking speed at un-signalized midblock crosswalk and its impact on urban street segment performance," *Journal of Traffic and Transportation Engineering (English Edition)*, vol. 1, no. 4, pp. 236–246, 2014. DOI: 10.1016/S2095-7564(15)30262-1.
- [14] A. R. Larusdottir and A. S. Dederichs, "Evacuation dynamics of children – walking speeds, flows through doors in daycare centers," in *Pedestrian and Evacuation Dynamics*, New York, NY: Springer, 2012, pp. 139–148. DOI: 10.1007/978-1-4419-9725-8_13.
- [15] P. Robinette, W. Li, R. Allen, A. M. Howard, and A. R. Wagner, "Overtrust of robots in emergency evacuation scenarios," in *Proceedings of the 11th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, Christchurch, New Zealand, Mar. 2016, pp. 101–108.
- [16] G. Chen, M. Meghjani, and M. B. Prasetyo, *Robot guided evacuation with viewpoint constraints*, 2024. DOI: 10.48550/arXiv.2409.19466. arXiv: 2409.19466 [cs.R0].

A Appendix

A.1 Test Map Configurations

Table 8: Map generation configuration parameters.

Category	Parameter	Value
General	Number of rooms	10
	Number of victims	10
Room	Vulnerability probability	0.4
	Vulnerability severity	0.2
	Min. length	5
	Max. length	10
	Min. width	5
	Max. width	10
Tunnel	Vulnerability probability	0.1
	Min. thickness	2
	Max. thickness	3
Fire	Spread rate	0.02
	Duration	100
	Initial fire points	1

A.2 Robot Count test results

Table 9: Victims saved as a function of the number of agents, summarised over simulation runs (mean, standard deviation, median, interquartile range, and 95% confidence interval).

Agents	Mean	SD	Median	IQR	CI95 lo	CI95 hi
0	1.82	2.06	1.0	3.00	1.42	2.22
2	2.84	2.31	2.0	3.00	2.39	3.29
4	3.08	2.41	3.0	3.25	2.61	3.55
6	3.35	2.52	3.0	4.00	2.86	3.84
8	3.49	2.69	3.0	4.00	2.96	4.02

Table 10: Victims dead as a function of the number of agents, summarised over simulation runs (mean, standard deviation, median, interquartile range, and 95% confidence interval).

Agents	Mean	SD	Median	IQR	CI95 lo	CI95 hi
0	2.06	2.09	2.0	3.0	1.65	2.47
2	1.96	1.95	2.0	3.0	1.58	2.34
4	2.03	2.02	2.0	3.0	1.63	2.43
6	2.00	1.96	2.0	3.0	1.61	2.39
8	2.20	2.09	2.0	3.0	1.79	2.61

Table 11: Victims left as a function of the number of agents, summarised over simulation runs (mean, standard deviation, median, interquartile range, and 95% confidence interval).

Agents	Mean	SD	Median	IQR	CI95 lo	CI95 hi
0	6.12	2.52	6.0	4.00	5.63	6.61
2	5.20	2.66	5.0	4.00	4.68	5.72
4	4.89	2.49	5.0	4.00	4.40	5.38
6	4.65	2.79	4.0	4.25	4.10	5.20
8	4.31	2.78	4.0	4.25	3.77	4.85

A.3 Repository Link

<https://github.com/mariagroove/BEP-SaR>