AI

MDPI

Article

Trust-Aware Reflective Control for Fault-Resilient Dynamic Task Response in Human–Swarm Cooperation

Yibei Guo [1], Yijiang Pang [1], Joseph Lyons [2], Michael Lewis [3], Katia Sycara [4] and Rui Liu [1,*]

[1] Cognitive Robotics and AI Lab (CRAI), College of Aeronautics and Engineering, Kent State University, Kent, OH 44240, USA; yguo27@kent.edu (Y.G.); yijiang.pang@gmail.com (Y.P.)
[2] Air Force Research Laboratory, Wright-Patterson AFB, Dayton, OH 45433, USA; joseph.lyons.6@us.af.mil
[3] School of Computing and Information, University of Pittsburgh, Pittsburgh, PA 15260, USA; cmlewis@pitt.edu
[4] Robotics Institute, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213, USA; sycara@andrew.cmu.edu
* Correspondence: ruiliu.robotics@gmail.com

Abstract: Due to the complexity of real-world deployments, a robot swarm is required to dynamically respond to tasks such as tracking multiple vehicles and continuously searching for victims. Frequent task assignments eliminate the need for system calibration time, but they also introduce uncertainty from previous tasks, which can undermine swarm performance. Therefore, responding to dynamic tasks presents a significant challenge for a robot swarm compared to handling tasks one at a time. In human–human cooperation, trust plays a crucial role in understanding each other's performance expectations and adjusting one's behavior for better cooperation. Taking inspiration from human trust, this paper introduces a trust-aware reflective control method called "***Trust-R***". ***Trust-R***, based on a weighted mean subsequence reduced algorithm (WMSR) and human trust modeling, enables a swarm to self-reflect on its performance from a human perspective. It proactively corrects faulty behaviors at an early stage before human intervention, mitigating the negative influence of uncertainty accumulated from dynamic tasks. Three typical task scenarios {Scenario 1: flocking to the assigned destination; Scenario 2: a transition between destinations; and Scenario 3: emergent response} were designed in the real-gravity simulation environment, and a human user study with 145 volunteers was conducted. ***Trust-R*** significantly improves both swarm performance and trust in dynamic task scenarios, marking a pivotal step forward in integrating trust dynamics into swarm robotics.

Keywords: trust repairing; attention transfer; human–robot collaboration; fault recovery

check for updates

Citation: Guo, Y.; Pang, Y.; Lyons, J.; Lewis, M.; Sycara, K.; Liu, R. Trust-Aware Reflective Control for Fault-Resilient Dynamic Task Response in Human–Swarm Cooperation. *AI* **2024**, *5*, 446–464. https://doi.org/10.3390/ai5010022

Academic Editor: Demos T. Tsahalis

Received: 13 February 2024
Revised: 16 March 2024
Accepted: 19 March 2024
Published: 21 March 2024

1. Introduction

A robot swarm, comprising multiple homogeneous robots, autonomously coordinates itself through unified control laws to achieve collective behaviors such as aggregation, flocking, and navigation [1–3]. In contrast to a single robot, a robot swarm possesses an enhanced capability for task execution, owing to its multi-member characteristics and inherent resilience to failures. On the other hand, humans excel in tasks related to comprehension, plan adjustment, and risk management [4–6]. By integrating robot swarms with human operators, a collaborative human–swarm system can effectively perform complex and large-scale tasks, such as searching for victims in natural disasters [7,8], monitoring public areas [9,10], and tracking multiple ground targets [11,12]. These tasks are beyond the capabilities of single-robot systems due to the need for diversity in assistance, coverage of large areas, and tracking of numerous targets [13–16].

In real-world deployments, task requirements are often dynamic and manifold. A swarm is expected to execute tasks consecutively. For instance, in social security, the tracking of multiple suspicious vehicles must be continuous; during flood rescue operations, the search for new victims must commence immediately after the discovery of a victim; and in forest fire control, rapidly emerging fire-prone areas demand timely extinguishing. In all these scenarios, a swarm must perform sequential task execution without re-calibrating its motion status after each task. This dynamic task response affords a swarm team limited time for re-calibration and introduces cumulative uncertainty stemming from unstable physical systems, motion status, and environmental disturbances, ultimately diminishing team performance.

Although the capacity for dynamic task response holds significant importance in practical swarm deployments, the frequent occurrence of tasks presents challenges to the quality of human–swarm collaboration. Firstly, human cognitive abilities are limited, rendering it challenging for individuals to monitor and control multiple robots simultaneously as the swarm dynamically responds to tasks. The dynamic nature of robot behaviors and the constraints of human attention make it difficult for humans to discern and rectify faulty robot actions.

Secondly, real-world factors encountered during dynamic task responses, such as motor wear and tear, sensor failures, and disturbances caused by wind, introduce uncertainty into swarm executions. Maintaining high-quality cooperation with humans becomes a challenge for a robot swarm in the presence of various disturbances.

Thirdly, unlike a centralized swarm that exhibits consistent behaviors due to unified control laws, a decentralized swarm exhibits variations in behavior among different local regions due to distributed control laws. Distinguishing abnormal behaviors from normal ones is a daunting task for humans [17,18]. For instance, during the status adjustment phase, a decrease in speed may be considered normal to maintain connectivity, whereas, during the motion consensus phase, the same decrease in speed may be seen as abnormal due to the expectation of consistent motion patterns.

Given the aforementioned challenges, effectively aligning swarm behaviors with human expectations for productive collaboration is a complex endeavor.

Trust in human–human interactions reflects an attitude of cooperation, indicating an individual's willingness to rely on teammates [19]. Trust in human–swarm collaboration represents a human's belief in the capabilities and reliability of a robot team when performing a task. Greater levels of trust lead to increased willingness on the part of humans to allocate tasks and a reduced need for corrections in swarm behavior. Conversely, lower levels of trust prompt human interventions in both behavior correction and communication [20]. As trust profoundly influences interactions, trust modeling has the potential to enhance cooperation between humans and swarms.

Motivated by these advantages, this research introduces the ***Trust-R*** method, a trust-aware reflective control approach, to calibrate swarm behaviors during dynamic task responses under human supervision. Supported by a trust estimation, as illustrated in Figure 1, a weighted update algorithm enables a robot to selectively share information with trusted robot neighbors while limiting information exchange with distrusted neighbors. This approach mitigates the adverse impact of abnormal robots on the entire swarm, facilitating the repair of faulty robot behaviors and the calibration of human–swarm cooperation by reducing error accumulation during dynamic task responses. This paper makes three primary contributions:

- A trust-based control algorithm, denoted as ***Trust-R,*** has been devised. It relies on control laws and a trust model to plan swarm movements based on its comprehension of human expectations. This algorithm facilitates collaboration between a human and a swarm when responding to dynamic tasks.
- A reflection mechanism has been developed to calibrate cooperation between a swarm and a human during dynamic responses. Leveraging trust estimation, the ***Trust-R*** approach enables a swarm to self-diagnose its erroneous behaviors and proactively mitigate their effects by either rectifying or discontinuing the actions of a robot.
- A novel framework, termed "behavior-repair to trust-repair", has been introduced to sustain trust between a human and a robot swarm during their cooperation. This framework emphasizes the swarm's role in rectifying undesired behaviors to prevent the erosion of trust between the human supervisor and the robot team.

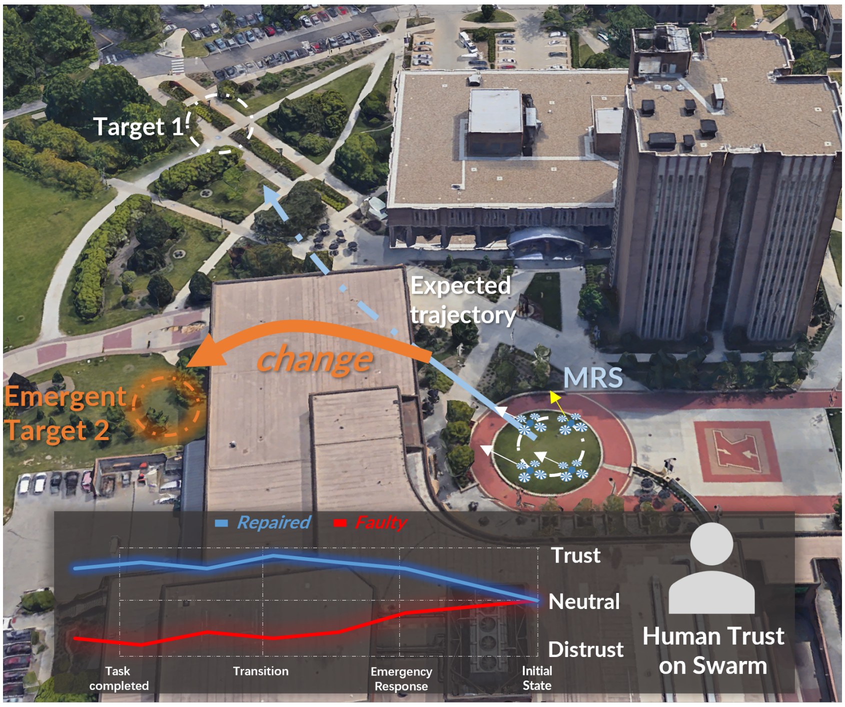

Figure 1. Illustration of the trust-aware reflective control of a swarm for dynamic task response. When a robot swarm exhibits faulty behaviors caused by faulty robots, the ***Trust-R*** repairs the performance of the swarm to regain human trust.

2. Related Work

Previous research has explored trust in human–robot interactions. In efforts to optimize task scheduling and alleviate human workloads, studies such as [21,22] have investigated time-series trust models for human–robot collaboration. These models were established based on trust factors encompassing factors such as prior trust levels and current robot performance. In an attempt to enhance the quality and adaptability of robot motion planning, ref. [23] proposed a trust-based real-time switching framework for human–robot systems. This framework dynamically switched between autonomous and manual motion planning modes based on trust value and operator availability. However, these approaches may not be suitable for real-world scenarios where low-trust robots frequently encounter inevitable disturbances. Such methods could inadvertently burden human operators by shifting the responsibilities of lower-trust robots onto them. In contrast, our approach in this paper leverages human trust as a behavioral expectation for a robot swarm's self-correction without placing additional demands on human attention. By utilizing trust, we align human expectations with robot behavior to facilitate fault-resilient human–robot cooperation.

The concept of trust repair mechanisms has been explored in prior work. Studies such as [24,25] have evaluated methods for repairing trust in human–robot cooperation through actions such as robot apologies for mistakes or promises of improved performance. Additionally, ref. [26] investigated the impact of explanation mechanisms on trust repair, aiding robots in regaining human trust by providing explanations for their decision-making processes. However, these studies often assessed trust after tasks were completed, failing to capture in-process changes in trust. This limitation made it challenging to identify critical factors influencing cooperation quality during robot task execution. In contrast, our paper focuses on the cognitive framework of repairing human trust by addressing faulty robot behaviors. Through trust modeling that helps robots understand human expectations, our ***Trust-R*** approach enables a swarm to self-correct its faulty behaviors promptly, rectifying these behaviors before human intervention becomes necessary. This proactive fault correction mechanism reduces the cognitive load on humans and fosters trust repair.

Previous research has also delved into fault detection and tolerance in robot swarms [27,28]. Inspired by the synchronized flashing behavior observed in fireflies, ref. [29] developed a general abnormality detection method to identify non-operational robots in a swarm by analyzing the different flashing frequencies of each robot's onboard light-emitting diodes. However, this method required faulty robots to proactively report their fault status to the swarm for correction, leaving unreported issues unaddressed. Refs. [30,31] developed behavior-based approaches to distinguish normal from abnormal robots in a swarm, treating persistent and abundant behaviors as normal and rare behaviors as abnormal. Nevertheless, these methods have limitations when it comes to robots diagnosing higher-level faults without a holistic view, as well as coping with variations in tolerance to abnormal behaviors over time and in specific situations. In contrast, our paper investigates a fault detection method for robot swarms by developing a robot's understanding of human trust. Through self-reflection of robot behaviors based on its estimation of human trust, ***Trust-R*** helps a swarm proactively correct its faulty behaviors.

Some previous work has addressed failures caused by faulty robots within swarms. For instance, ref. [32] used the transmission of position data between robots to identify and isolate faulty robots. Refs. [33,34] proposed a decentralized fault-tolerant rendezvous algorithm to enable fault-free robots to achieve rendezvous even in the presence of faulty robots. However, these studies assumed that the motion patterns of faulty robots significantly differed from those of others and had a limited impact on overall performance. Ref. [35] defined faulty robots as those not located in the desired position, while [36] defined them as robots with incorrect heading directions. These methods detected and corrected faulty behaviors by comparing observed behavior with ideal robot behaviors. Nonetheless, these studies did not account for real-world issues, such as environmental disturbances, motor degradation, or sensor failures, which introduce uncertainty and significantly affect swarm performance. In contrast, our ***Trust-R*** approach aids robots in self-reflecting on their behaviors from the perspective of human trust, enabling them to identify and correct swarm-wide faults while mitigating the negative influence of individual faulty robots. This proactive fault correction mechanism makes human–swarm cooperation more applicable to real-world deployments.

Previous research has addressed challenges in dynamic task response. Ref. [37] employed multiple cooperating robots, with some stationary robots serving as position references at each movement step to reduce accumulated position errors. Refs. [38,39] introduced dynamic task allocation, decomposing tasks into a sequence of sub-tasks to limit the effects of accumulated position errors resulting from hardware information noise over time. However, these studies assumed that a robot would stop and wait for others to take over its work once it reached a predefined threshold of position error, which is not suitable for general dynamic task responses that require continual task execution without interruptions. In contrast, ***Trust-R*** reduces accumulated uncertainty during dynamic task response by enabling robots to self-diagnose and correct their faulty behaviors early in the process.

Our previous work, as presented in [40,41], demonstrated the effectiveness of a decentralized trust-aware behavior reflection method in correcting faulty swarm behaviors. It also investigated the ***Trust-R*** approach, which restored performance and human trust in the swarm to an appropriate level by correcting undesirable behaviors. This paper extends our previous research by exploring the effectiveness of ***Trust-R*** in dynamic task response, where sequential task assignments lead to the accumulation of uncertainties that can undermine swarm performance. ***Trust-R*** assists in correcting these accumulated errors in real time to support dynamic task response in swarms. Additionally, this paper adopts real-gravity environmental settings and robot models, further aligning the ***Trust-R*** implementation with real-world scenarios.

3. Materials and Methods

3.1. Distrusted Flocking in Dynamic Task Response

3.1.1. Illustrative Scenario for Swarm Correction

The chosen task scenario involves the execution of patrol tasks in the realm of social security. A robot swarm is deployed to carry out area inspections as directed by human supervisors. The desired behaviors for the robots within the swarm encompass consistent speed and heading direction, along with functional connectivity and formation maintenance.

Let us consider a robot swarm comprising n holonomic robots, each characterized by their respective positions denoted as $X_i \in \mathbb{R}^3$, where $X_i = (\boldsymbol{x}_{i,\mathrm{h}}, \boldsymbol{x}_{i,\mathrm{v}}, \theta_i)$. Each robot is assigned a unique identifier (UID) represented as $i \in \{1, 2, \ldots, n\}$. The communication network is represented as $\boldsymbol{G} = (\boldsymbol{\mathcal{V}}, \mathcal{E})$, where each node $v \in \mathcal{V}$ corresponds to a robot. Robot i exclusively communicates with its immediate neighbors, denoted as $j \in N_i$, with N_i being the set of all neighbors within a specified communication radius R. If robot j is a neighbor of robot i, then there exists an edge $(v_i, v_j) \in \mathcal{E}$. The connectivity graph adheres to the principles of being connected and undirected, meaning that if $(v_i, v_j) \in \mathcal{E}$, then $(v_j, v_i) \in \mathcal{E}$ as well.

Each robot i is controlled via linear velocity $\boldsymbol{u}_i^v$ and angular velocity $\boldsymbol{u}_i^w$, which are generated by the robot's motors. The state of each robot is characterized by $\boldsymbol{x}_i$ and θ_i, representing the horizontal and vertical positions as well as the orientation state, respectively. For a detailed description of the dynamic model for each robot, please refer to our prior publication [40].

At each time step t, a robot i updates its motion status by computing an average of its neighbors' motion statuses using Equation (1).

$$\boldsymbol{u}_i[t+1] = \frac{1}{N_i + 1}(\boldsymbol{u}_i[t] + \sum_{j \in N_i} \boldsymbol{u}_j[t]) \tag{1}$$

As evident from the aforementioned distributed update method, faulty robots can transmit unreliable motion information to their neighbors, thereby misleading the motions of their neighbors.

Definition 1. *A "Faulty robot" is defined as a robot that exhibits undesired behaviors resulting from the propagation of faulty data from a failed robot, environmental disturbances, or other correctable causes. In contrast, a "Failed robot" refers to a robot whose undesired behaviors are not correctable.*

Definition 2. *During swarm deployments, where the influence of faulty robots is apparent, the swarm may display abnormal behaviors, such as partial disconnection or heading deviation. Such deviations erode human trust in the swarm's performance, leading to the categorization of the swarm as a "Distrusted swarm".*

Definition 3. *In the context of this study, "Influential Factors" refer to real-world factors such as degraded robot motors, sensor and mechanical system uncertainties, environmental disturbances (e.g., wind or rain), or other elements that can induce abnormal robot behaviors and compromise robot performance. These influential factors contribute to the occurrence of "robot faults", which are characterized by abnormal robot behaviors, such as degraded performance or anomalous motions.*

3.1.2. Dynamic Task Response with Accumulated Uncertainty

In practical scenarios characterized by the complexity of real-world tasks and environmental conditions, robot swarms are often tasked with continuous task execution. This includes transitioning from one task to another seamlessly and adapting to emergent tasks even in the middle of an ongoing one. To ensure that the robot swarm effectively navigates toward dynamic task destinations, the robots within the swarm are categorized into two types: leader robots and follower robots. In a conventional hierarchical swarm control setup [42], only leader robots have access to control information transmitted from the base station, such as dynamic target coordinates and cruising speeds. In contrast, sensor data, encompassing velocity and position information, can be exchanged among all types of robots within the swarm.

Simultaneously, faulty robots tend to accumulate motion uncertainties, manifesting as shifts in location, deviations in heading direction, fluctuations in speed, and inconsistencies in mapping within the swarm due to updates in consensus policies. In this paper, focusing on the speed and heading direction requirements for flocking behaviors, the accumulated uncertainty is quantified as the accumulated speed $\delta \boldsymbol{u}$ and accumulated location shift $\delta \boldsymbol{x}$ of the swarm relative to the expected swarm status upon receiving a new task assignment. Under this assumption, the velocity $\boldsymbol{u}_i$ of leader robots is updated using Equation (2):

$$\boldsymbol{u}_i[t+1] = \frac{1}{N_i+1}(\boldsymbol{u}_i[t] + \sum_{j \in N_i} \boldsymbol{u}_j[t]) + \boldsymbol{u}_i^{\gamma}[t] \tag{2}$$

where $\boldsymbol{u}_i^{\gamma}$ is the navigational feedback and accumulated uncertainty given by:

$$\begin{aligned}\boldsymbol{u}_i^{\gamma}[t] &:= f_i^{\gamma}(\boldsymbol{x}_i[t], \boldsymbol{x}_{\gamma}, \delta \boldsymbol{x}, \boldsymbol{u}_i[t], \boldsymbol{u}_{\gamma}, \delta \boldsymbol{u}) \\ &= -c_1^{\gamma}(\boldsymbol{x}_i[t] - \boldsymbol{x}_{\gamma} + \delta \boldsymbol{x}) - c_2^{\gamma}(\boldsymbol{u}_i[t] - \boldsymbol{u}_{\gamma} + \delta \boldsymbol{u}), c^{\gamma} > 0,\end{aligned}$$

and the $\gamma -$ robot $(\boldsymbol{x}_{\gamma}, \boldsymbol{u}_{\gamma})$ is the virtual leader that leads the swarm to follow its trajectory. Here, $\boldsymbol{x}_{\gamma}$ and $\boldsymbol{u}_{\gamma}$ represent the destination and cruising speeds of the virtual leader (γ), which guides the swarm to follow a predetermined trajectory [43,44]. The parameters c^{γ} represent the gain components associated with the virtual leader's control. In the context of normal task execution, as time progresses ($t \to \infty$), the behaviors of individual robots are expected to converge toward values such that $|\boldsymbol{x}_i[t] - \boldsymbol{x}_{\gamma}| \to 0$ and $|\boldsymbol{u}_i[t] - \boldsymbol{u}_{\gamma}| \to 0$ [45]. However, when the swarm responds to a dynamic task, the expected swarm status upon receiving the new task assignment can be influenced by $\delta \boldsymbol{x}$ and $\delta \boldsymbol{u}$. As demonstrated by the dynamic task update method mentioned earlier, faulty robots have the potential to introduce uncertainty into the swarm's behavior, thereby deteriorating the swarm's task performance. To mitigate this uncertainty and ensure swarm performance, our method is designed to actively suppress the negative influences represented by $\delta \boldsymbol{x}$ and $\delta \boldsymbol{u}$ in real time, ultimately leading to the correction of swarm behaviors.

3.2. Trust-Aware Reflective Control for Dynamic Task Response

The architecture of the ***Trust-R*** system is illustrated in Figure 2. With the ***Trust-R*** approach, an understanding of human-expected behaviors is cultivated to govern the quality of communication between the robot and its neighboring robots, with the aim of minimizing the adverse impact of a faulty robot on the entire swarm.

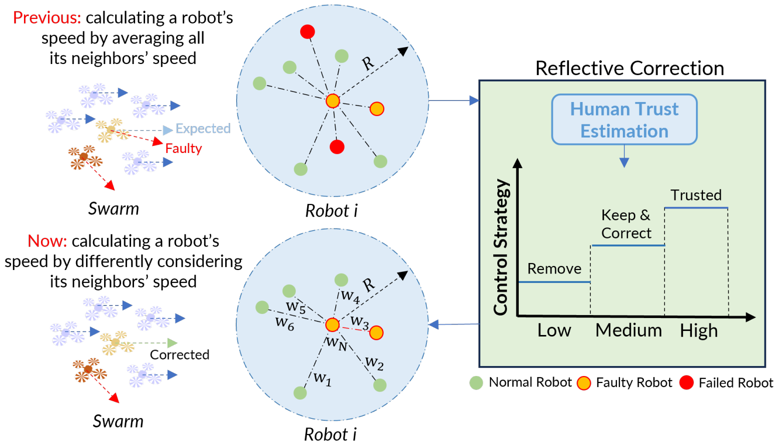

Figure 2. Trust-aware reflective control for dynamic task response in human–supervisory swarm deployment. The weighted communication quality method enhances information exchange and connectivity between trustworthy robots and decreases information sharing between faulty robots.

3.2.1. Human Supervision

Within the realm of human–swarm cooperation, the human assumes the role of an operator responsible for monitoring and directing the task execution of the robot swarm. As the operator, the human continuously monitors the real-time status of all robots and possesses knowledge of the desired swarm behavior, including parameters such as minimum velocity, heading direction, and formation criteria. The operator is capable of discerning the current performance from the expected performance and assigns performance scores to individual robots and the entire swarm. Subsequently, these performance scores are employed to establish thresholds that determine the trust level of the behaviors exhibited by the swarm. The swarm, informed of these thresholds, then undertakes self-correction based on its comprehension of human trust. In this paper, the scores assigned by the human operator to each robot serve as fundamental parameters for adjusting the overall performance of the robot swarm.

3.2.2. Trust-Aware Connectivity

In response to the trust signal conveyed by a human operator, each robot formulates its communication strategy with neighboring robots. Typically, each robot within the swarm calculates its speed by averaging the speeds of all its neighbors in order to reach a consensus. However, in instances where faulty robots are present within the swarm, their detrimental impact on swarm performance poses a risk to the successful execution of assigned tasks. For example, a robot in the swarm may exhibit unexpected linear velocity and angular velocity due to a motor malfunction. Since the swarm strives to reach a consensus, where robots exchange status information with their neighbors, the presence of faulty robots can disrupt the swarm's performance. To address this challenge, this paper employs a weighted connection approach, drawing inspiration from the weighted mean subsequence reduced algorithm [46]. This method enhances connectivity and communication between trustworthy robots while reducing the level of information-sharing among faulty robots. Incorporating the WMSR algorithm, ***Trust-R*** dynamically adjusts communication weights among robots based on trust assessments. Each robot i exclusively engages in communication with its direct neighbors, denoted as $j \in N_i$, where N_i represents the set of all neighbors of robot i within the communication radius. Specifically, for robot i, the velocity u_i is updated with a weighted reference to its neighbors, incorporating trust-derived weights:

$$u_i[t+1] = w_i[t]u_i[t] + \sum_{j \in N_i} w_j[t]u_j[t] \tag{3}$$

Here, $w_i[t]$ and $w_j[t]$ are weights aligned with trust levels of robot i and its neighbors at time t, respectively. The calculation method of the weights is detailed in Section 3.2.4. This integration of WMSR into ***Trust-R*** effectively diminishes the influence of less trustworthy robots, ensuring the swarm's robustness and alignment with human expectations. The major parameters of the robot in the swarm used are illustrated in Figure 3.

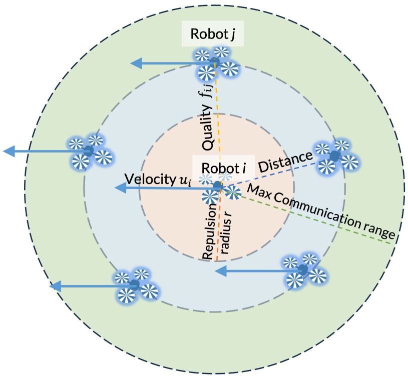

Figure 3. Major robot movement parameters used in the trust-aware reflective control framework.

3.2.3. Trust-Aware Communication Quality Assessment

For robot i, the complete communication graph is denoted as $\boldsymbol{\mathcal{E}} = (i,j) \mid j \in N_i$. Within this context, the estimated trust levels of the two robots i,j are employed to establish the communication quality, denoted as $f_{ij} \in [0,1]$, which serves as a metric for assessing the reliability of exchanged information. The trust-aware communication quality undergoes dynamic updates to adapt to changes in the communication graph, as described by Equation (4). The parameter ρ signifies the optimal communication distance between two robots i and j, where communication within this distance is deemed to possess the highest quality. Additionally, the parameter η is introduced as a weighting factor designed to mitigate the impact of faulty robots on their neighboring robots.

$$f_{ij} = \begin{cases} 0 & ||\boldsymbol{x}_i - \boldsymbol{x}_j|| \geq R \\ \frac{1}{2}(g_i + g_j)\eta & ||\boldsymbol{x}_i - \boldsymbol{x}_j|| \leq \rho \\ \frac{(g_i+g_j)\eta}{2} \exp \frac{-\gamma(||\boldsymbol{x}_i - \boldsymbol{x}_j|| - \rho)}{R-\rho} & otherwise \end{cases} \tag{4}$$

where g_i is the trust level of robot i. The communication quality evaluation method outlined above implies that within the communication range, the reliability of communication is determined by averaging the trust values of the two robots involved, i,j. In cases where both robots are trusted, their communication is considered the most reliable. However, if one of the robots is deemed faulty, the most reliable communication within that connection is attributed to the trusted robot.

The motivation behind developing the trust-aware communication quality is twofold. Firstly, it aims to promote information-sharing with trusted robots by assigning higher upper limits to their communication quality. Conversely, it discourages information-sharing with untrusted robots by setting lower upper limits on their communication quality. Secondly, to encourage the formation of a cohesive swarm with robots positioned close to each other, the communication quality diminishes as the distance between robots increases.

3.2.4. Trust-Aware Behavior Correction

The process of proactively correcting faulty behaviors within a swarm is a two-step procedure. Initially, it involves the correction of faulty robots by limiting the exchange of unreliable information originating from these robots and relying on trusted robots for behavior correction. Failed robots are segregated from the group of trusted robots to prevent the dissemination of unreliable motion information. Subsequently, the connectivity control mechanism introduced in Section 3.3 is employed to reduce the separation between robots and their "normal" neighbors. This adjustment results in each robot modifying its behavior, including heading direction and speed, based on a higher proportion of trusted motion information.

$$w_k[t] = \frac{\hat{f}_k[t]}{\hat{f}_i[t] + \sum_{j \in N_i} \hat{f}_j[t]}, k \in [i, N_i] \tag{5}$$

The calculation of weights for updating each robot's status is determined by Equations (3) and (5). The outcome of the weighted connection mechanism is depicted on the right side of Figure 2 and can be expressed using Equation (6).

$$\boldsymbol{u}_i[t+1] = \frac{\hat{f}_k[t]}{\hat{f}_i[t] + \sum_{j \in N_i} \hat{f}_j[t]} (\boldsymbol{u}_i[t] + \sum_{j \in N_i} \boldsymbol{u}_j[t]) \tag{6}$$

In the process of updating the status of robot i, the weights w_k are computed by normalizing all the communication quality values within the communication range, as described in Equation (5). Specifically, when $k = i$, the trust level of the robot itself is used, denoted as $\hat{f}_k = g_i$. Conversely, when $k = j \in N_i$, the communication quality between robots i and j is utilized, denoted as $\hat{f}_k = f_{ij}$. It is important to note that $\hat{f}_i = g_i$ holds true for all values of k.

With the trust-weighted update, the control input $\boldsymbol{u}_v^i$ and $\boldsymbol{u}_w^i$ for robot motors are changed to $\boldsymbol{u}_{v,trust}^i$ and $\boldsymbol{u}_{w,trust}^i$. The gains K_v and K_w are parameters for adjusting the motor output.

$$\boldsymbol{u}_{i,trust}^{\mathrm{v}} = (K_{\mathrm{v}} + K_{\mathrm{v},trust})(\mathbf{v}_i + \boldsymbol{q}_{\boldsymbol{N}_i})^T \boldsymbol{b}_i \tag{7}$$

$$\boldsymbol{u}_{i,trust}^{w} = (K_w + K_{w,trust})\big(\gamma_i + \phi(\boldsymbol{b}_i, \boldsymbol{q}_{\mathrm{N}_i})\big) \tag{8}$$

Let $\boldsymbol{u}_i[t+1]$ represent the current speed of a robot exhibiting abnormal behaviors at time $t+1$, while the expected speed calculated by referencing its trusted neighbors is denoted as $\boldsymbol{u}_{i,trust}[t+1]$. To adjust the control output of the robot's motors, we can determine the additional trust gains $K_{\mathrm{v},trust}$ and $K_{w,trust}$. These gains are updated according to the disparity between the robot's actual speed and the speed trusted by the human operator.

$$K_{\mathrm{v},trust}[t+1] = \frac{\boldsymbol{u}_{i,trust}^{\mathrm{v}}[t] - \boldsymbol{u}_i^{\mathrm{v}}[t]}{\boldsymbol{u}_i^{\mathrm{v}}[t]} \tag{9}$$

$$K_{w,trust}[t+1] = \frac{\boldsymbol{u}_{i,trust}^{w}[t] - \boldsymbol{u}_i^{w}[t]}{\boldsymbol{u}_i^{w}[t]} \tag{10}$$

In order to avoid collision, the safe distance (repulsion radius) for separating robots is set to r.

3.3. Trust-Aware Connectivity Maintenance for Motion Consensus

To further rectify faulty swarm behaviors, the connectivity between a faulty robot and the other trusted robots is enhanced using Equation (6). In this context, $\boldsymbol{L}$ represents the Laplacian matrix of the graph, and $\boldsymbol{\lambda}_2$ denotes the algebraic connectivity, with e_2 corresponding to the associated eigenvector. This enhancement aims to reduce the separation between a faulty yet correctable robot and its neighboring trusted robot. By bringing these robots closer together, more reliable information becomes available, facilitating the correction of the faulty robot's behavior. Here, $\boldsymbol{x}_{i,\psi}$ represents the position component in the direction ψ (either horizontal or vertical) for robot i. The computation of $\frac{\boldsymbol{\alpha L}(\boldsymbol{x})}{\boldsymbol{\alpha x}_{i,\psi}}$ is achieved by assessing the difference in reliability values, f_{ij}, between adjacent time steps, as demonstrated in Equation (6).

$$\boldsymbol{u}_i = \nabla_{i,\psi}\boldsymbol{\lambda}_2 \tag{11}$$

$$= \frac{\boldsymbol{\alpha\lambda}_2(\boldsymbol{L})}{\boldsymbol{\alpha x}_{i,\psi}} = \frac{\boldsymbol{\alpha\lambda}_2(\boldsymbol{L})}{\boldsymbol{\alpha L}(\boldsymbol{x})}\frac{\boldsymbol{\alpha L}(\boldsymbol{x})}{\boldsymbol{\alpha x}_{i,\psi}} = \textbf{Trace}\left\{\left[\frac{e_2 e_2^T}{e_2^T e_2}\right]^T \left[\frac{\boldsymbol{\alpha L}(\boldsymbol{x})}{\boldsymbol{\alpha x}_{i,\psi}}\right]\right\} \tag{12}$$

Theorem 1. *The* ***Trust-R*** *method promotes a relatively shorter distance between a robot and other trusted robots, while simultaneously advocating for a relatively greater distance between a robot and other distrusted robots. This adjustment gradually diminishes to zero once a consensus is attained within the flocking behavior.*

Proof. When using the trust-aware communication quality to adjust the distance of robot i to other robots, the adjustment along a direction ψ is

$$\boldsymbol{u}_i = \textbf{Trace}\left\{\left[\frac{e_2 e_2^T}{e_2^T e_2}\right]^T \left[\frac{\boldsymbol{\alpha L}(\boldsymbol{x})}{\boldsymbol{\alpha x}_{i,\psi}}\right]\right\}$$

$$= \textbf{Trace}\left\{\left[\frac{e_2 e_2^T}{e_2^T e_2}\right]^T \left[\frac{\boldsymbol{\alpha}[\boldsymbol{L}]_{ij}}{\boldsymbol{\alpha x}_{i,\psi}}\right]\right\}$$

For the off-diagonal elements in the Laplacian matrix, $\boldsymbol{L}$, $\frac{\alpha[\boldsymbol{L}]_{ij}}{\alpha \boldsymbol{x}_{i,\psi}}$ is solved by

$$\sum_{K} -\frac{\alpha f_{ij}}{\alpha x_{k,\psi}} \boldsymbol{u}_{k,\psi} = \frac{\alpha f_{ij}}{\alpha x_{i,\psi}} (\boldsymbol{u}_{j,\psi} - \boldsymbol{u}_{i,\psi})$$

For the diagonal elements in $\boldsymbol{L}$, $\frac{\alpha[\boldsymbol{L}]_{ij}}{\alpha \boldsymbol{x}_{i,\psi}}$ is solved by

$$\sum_{k} \left(\sum_{j} \frac{\alpha f_{ij}}{\alpha \boldsymbol{x}_{k,\psi}} \right) \boldsymbol{u}_{k,\psi} = \sum_{j} \frac{\alpha f_{ij}}{\alpha \boldsymbol{x}_{i,\psi}} (\boldsymbol{u}_{i,\psi} - \boldsymbol{u}_{j,\psi})$$

Since

$$\frac{\alpha f_{ij}}{\alpha \boldsymbol{x}_{i,\psi}} = -\frac{\gamma\eta(g_i + g_j)(\boldsymbol{x}_{i,\psi} - \boldsymbol{x}_{j,\psi})}{2(R-\rho)||\boldsymbol{x}_i - \boldsymbol{x}_j||} \exp \frac{-\gamma(||\boldsymbol{x}_i - \boldsymbol{x}_j|| - \rho)}{R-\rho}$$

$\frac{\alpha f_{ij}}{\alpha x_{i,\psi}}$ is constrained by the distance between the robots, which is always smaller than the communication radius R. When considering a desired flocking direction q_0, the degree of adjustment u_i between two robots i and j is directly proportional to their average trust score, calculated as $\frac{(g_i+g_j)\eta}{2}$. A higher trust score results in a more significant adjustment. Consequently, the ***Trust-R*** method encourages robots to maintain a relatively shorter distance from other trusted robots while advocating for greater separation from abnormal robots. As a result, the swarm gradually distances itself from the abnormal, faulty robots.

Once the robots achieve consensus in their heading direction, the values of u_i and u_j will become equal within a finite time. Consequently, $\frac{\alpha f_{ij}}{\alpha x_{i,\psi}}$ will reach 0, effectively ceasing the adjustment process when consensus is attained. □

3.4. Experiment

To assess the effectiveness of ***Trust-R*** in assisting the swarm in self-diagnosing and proactively mitigating the influence of faulty behaviors, three sequential task scenarios were developed. These scenarios were employed to compare the accumulated error and human trust ratings of the swarm both before and after implementing the ***Trust-R*** method.

3.4.1. Environment Design

The simulation environment was constructed using the ***CRAImrs*** framework, which is based on the Gazebo simulation software, task-specific swarm control laws, and trust models [47,48]. This environment is capable of simulating multi-robot teamwork while accounting for faulty robot behaviors resulting from real-world factors.

The simulated world has a map size of 50 × 50 m, as depicted in Figure 4. Within this environment, three green areas represent the targets for three distinct tasks. A quadrotor UAV model is incorporated into the simulation, and it can be controlled through velocity components in three-axis directions. The UAV model provides status information such as linear velocity, angular velocity, altitude, and absolute coordinates in the simulation world via an API. The experiment involves six robots, including one robot with motor issues, simulating a faulty robot. Each robot has a velocity set at 5.0 m/s, and a repulsion radius of 2 m is established to prevent collisions. The communication radius for all robots is 7.5 m.

In the communication quality assessment method (Equation (4)), g and η are used to define upper limits on communication quality, while γ determines the sensitivity of quality to mutual distance. Based on previous research [40], the g values are set as (1, 0.5, 0) for trusted robots, faulty robots, and failed robots, respectively. The η values are (1, 1, 0.4, 0.3, 0.2, 0.2), and the γ values are (0.1, 0.5, 1, 3, 5, 7) for communications between trusted-trusted robots, trusted-faulty robots, trusted-failed robots, faulty-faulty robots, faulty-failed robots, and failed-failed robots.

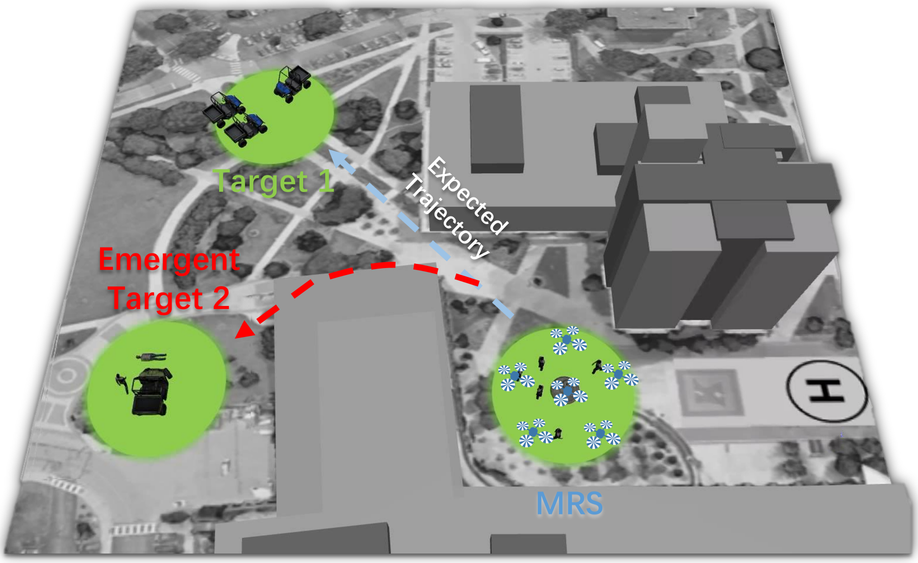

Figure 4. The simulation environment. In each task, the robot swarm will dynamically flock to an assigned target area marked in green on the map.

3.4.2. Task Scenario Design

Three typical tasks covering essential elements of dynamic task response scenarios were designed: ***Scenario 1: flocking to the assigned destination; Scenario 2: a transition between destinations; and Scenario 3: emergent response.*** In the first scenario, the UAV swarm's task is to monitor a target. The swarm is initially positioned at the "base station", and its objective is to flock to "target 1". During this flocking operation, one of the UAVs experiences a motor issue, affecting the entire swarm's performance. The application of ***Trust-R*** is tested in this scenario to determine whether it can help restore trust between the human operator and the UAV swarm by improving the swarm's performance. In the second scenario, the UAV swarm is programmed to follow the final state achieved in Scenario 1 and then transition to "target 2". This represents a situation where the UAV swarm needs to switch destinations, moving from "target 1" to "target 2". The scenario assesses how the accumulated error during the transition between destinations affects human trust in the swarm and whether ***Trust-R*** can enhance the UAV swarm's performance in dynamic situations. In the third scenario, similar to Scenario 2, the UAV swarm follows the final state achieved in Scenario 1 and then transitions to "target 2." However, in the middle of the flight, the leader UAV receives an order to change the target to an "emergent target" ("target 3"). The entire UAV swarm is expected to shift from the normal target to the emergent target, altering the swarm's state from cruising response to emergent response in terms of direction and velocity.

Each of the three task scenarios was simulated under four different conditions to thoroughly evaluate the effectiveness of ***Trust-R*** in various scenarios and with different levels of influential factors. The two different levels of influential factors, pertaining to motor issues, have restricted maximum speeds of 40% and 70%, respectively, to test the effectiveness of ***Trust-R*** with different faulty levels. Moreover, the variation in motor fault allows for more opportunities for the human to perceive the fault if the faults were not readily salient and apparent to the participants. The four simulated conditions are faulty condition suffering one failed robot with 40% maximum speed, faulty condition suffering one failed robot with 70% maximum speed, repaired condition suffering one failed robot with 40% maximum speed, and repaired condition suffering one failed robot with 70% maximum speed.

3.4.3. Human User Study

A human user study was conducted with the participation of 145 volunteers. The complete questionnaire can be accessed via the link https://kent.qualtrics.com/jfe/form/SV_0OImxiQRWOqxSId (accessed on 20 March 2024). The study was conducted on the crowd-sourcing platform Amazon Mechanical Turk, as detailed by Buhrmester et al. (2016) [49]. Volunteers who were proficient in English were recruited and compensated with a payment for their active involvement. To ensure data quality, strict eligibility criteria were enforced, requiring participants to be Amazon Turk Masters with an answer Approval Rate exceeding 80%.

The user study comprised two primary segments: a tutorial and an actual survey. The tutorial encompassed instructional video materials pertaining to a specific task, along with responses to associated queries. Subsequently, the survey phase featured twelve distinct sections, including three task scenarios, each encompassing four simulated conditions. While the three task scenarios followed a sequential order, the presentation of the four conditions within each scenario was randomized to mitigate any potential bias from prior knowledge. In each section of the study, participants were tasked with monitoring the progress of a designated task and assessing the swarm's motion behaviors. These behaviors included attributes such as flocking speed, heading direction, and the spatial relations of robots (connectivity and formation). Participants were then presented with a video depicting a UAV swarm engaged in a task and asked to determine whether any faults occurred in the video. Subsequently, a series of questions concerning the participants' judgments regarding the swarm and specific UAVs were presented. For illustrative purposes, two sample questions pertaining to the normal performance of a UAV swarm from the tutorial section are provided below. Following the viewing of a video showcasing the normal performance of a UAV swarm, participants were introduced to these two questions:

Question 1: *According to your observation of their motion behaviors and performance, do you think a fault occurs in the video?* ***Answer 1:*** *a. Yes, a fault occurs. b. No, it looks normal.*

Question 2: *Determine to which extent you trust the whole robot swarm.* ***Answer 2:*** *a. Completely Distrust. b. Distrust. c. Neutral. d. Trust. e. Completely Trust.*

4. Results

The data we have collected comprises 50 instances of valid data, with the collected variables falling under the category of ordinal categorical data. To evaluate the influence of various factors on human trust in UAV swarm interactions, we have employed the Mann–Whitney U test as our analytical method. This non-parametric statistical test is ideal for our analysis because it compares differences between two independent groups when the dependent variable, in this case, ordinal trust levels, is not normally distributed. For the sake of convenience, a mapping relationship has been established between human trust levels and numerical values, which are as follows: *Completely Distrust: 1, Distrust: 2, Neutral: 3, Trust: 4, and Completely Trust: 5.*

In summary, Table 1 presents the trust levels of human participants in both the faulty and repaired conditions for UAV swarms across all scenarios. Additionally, Table 2 compiles the final values of the flocking heading direction and distance to the destination for all scenarios. The results of testing the effects of ***Trust-R*** on human trust reveal substantial disparities between the faulty and repaired conditions, as depicted in Figure 5. The mean trust levels in the faulty and repaired conditions were 2.4 and 4.0, respectively. Participants were more likely to bestow higher levels of trust upon the repaired condition when compared to the faulty condition ($U = 18560, \rho = 0.21$). Notably, in Scenario 1, participants did not exhibit a significantly elevated level of trust in the repaired conditions as opposed to the faulty conditions. However, as the accumulated error propagated negative effects from faulty UAVs to the entire swarm in Scenarios Two and Three, participants were more inclined to place their trust in the repaired conditions rather than the faulty conditions. Detailed results from the three designed scenarios are presented as follows.

Table 1. Dynamic scenario conditions * Mann–Whitney U.

Swarm Status	Median Trust Level		
	Faulty	**Repaired**	ρ *
Scenario 1	Neutral/3.34	Trust/3.53	0.46
Scenario 2	Distrust/1.83	Trust/4.18	0.04
Scenario 3	Distrust/2.0	Trust/4.22	0.09

Table 2. Values of flocking heading direction and final distance to target.

	Value 1			Value 2		
	Designed	**Faulty**	**Repaired**	**Designed**	**Faulty**	**Repaired**
Scenario 1	−4	−25	−19	0.0	6.6	5.6
Scenario 2	43	3	46	0.0	20.6	2.0
Scenario 3	0	−19	−4	0.0	9.8	3.2

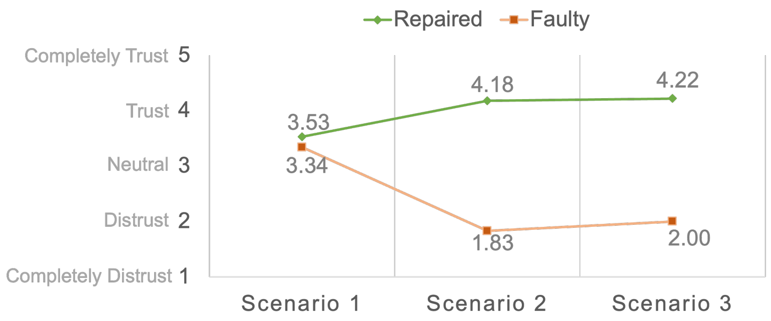

Figure 5. The average human trust level of the faulty condition and repaired condition for dynamic task response.

4.1. Scenario I: Flocking to the Assigned Destination

Figure 6 presents the outcomes of the experiment conducted within Scenario 1. Notably, participants exhibited similar tendencies to report faults in both the faulty and repaired conditions ($U = 4550, \rho = 0.46$). The mean trust levels for the faulty and repaired conditions were 3.34 and 3.53, respectively, with participants not indicating significantly higher trust levels in the repaired condition than in the faulty condition ($U = 4746, \rho = 0.47$).

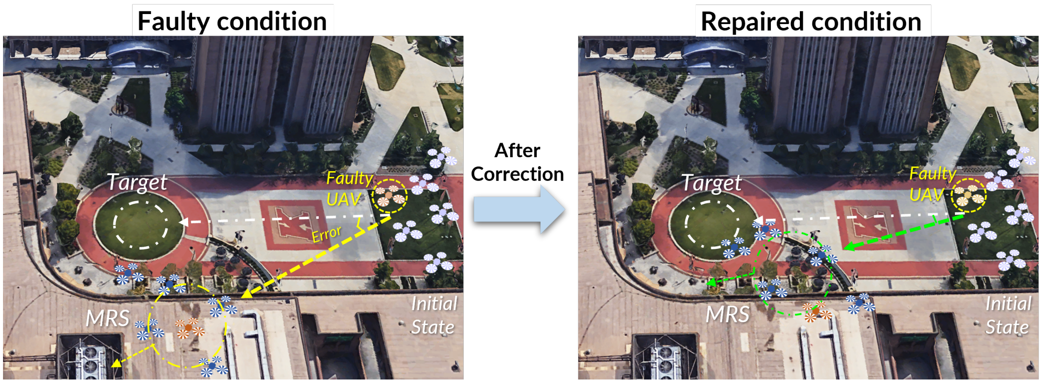

Figure 6. Experiment result for Case Study I: flocking to the assigned destination. In the faulty condition, the UAV swarm performs the task in the presence of a faulty robot; while applying the ***Trust-R***, the UAV swarm will restrict the influence of the faulty robot, which improves the performance of the UAV swarm.

In the context of Scenario 1, the likelihood of participants reporting faults in the faulty conditions varied with the severity of the faulty issues ($U = 600, \rho = 0.24$). Participants were more inclined to report faults when confronted with more substantial levels of faults. Concerning the faulty conditions, participants exhibited lower trust levels as the severity of faulty issues increased ($U = 479, \rho = 0.19$). Similarly, in the repaired condition, participants displayed diminished trust levels with increasing severity of faulty issues ($U = 440, \rho = 0.18$). Regarding the faulty condition, participants conveyed an average confidence level of 3.6 points in their ability to identify the faulty robots within the swarm. This confidence level was categorized as follows: (Very difficult: 1, Difficult: 2, Neutral: 3, Easy: 4, and Very easy: 5). In the faulty condition, 52% of participants who reported faults were able to correctly identify the faulty UAV within the swarm but made misidentifications among the normal UAVs. In contrast, only 8% of the participants correctly identified the faulty robot without any errors.

4.2. Scenario II: A Transition between Destinations

Figure 7 shows the result of the experiment under Scenario 2. In Scenario 2, a significant distinction was observed between the likelihood of participants reporting faults in the faulty condition compared to the repaired condition. Participants were notably more inclined to report a fault in the faulty conditions as opposed to the repaired conditions ($U = 400, \rho = 0.04$). The mean trust levels for the faulty and repaired conditions were 1.83 and 4.18, respectively, with participants demonstrating a stronger tendency to report higher trust levels in the repaired condition than in the faulty condition ($U = 316, \rho = 0.03$).

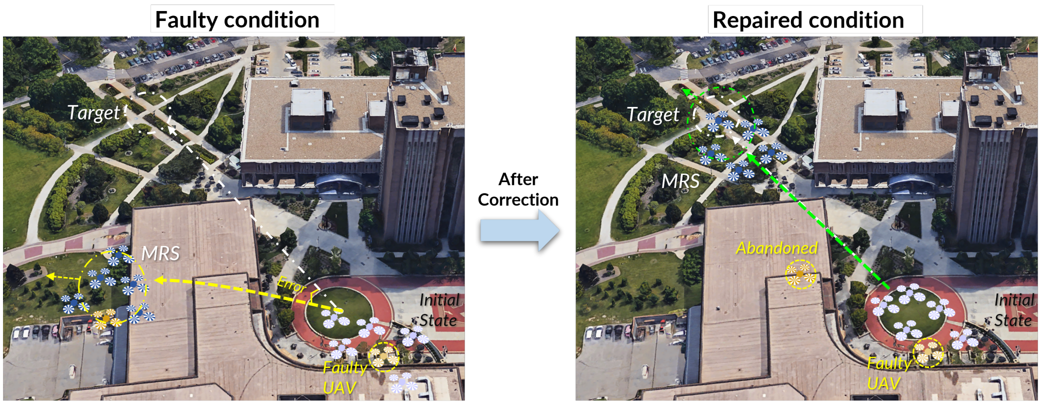

Figure 7. Experiment results for Case Study II: a transition between destinations. In the faulty condition, the UAV swarm performs a task in the presence of a faulty robot; while applying the ***Trust-R***, the UAV swarm will restrict the influence of the faulty robot, which improves the performance of the UAV swarm.

The probability of participants reporting faults in the faulty conditions remained relatively consistent across different levels of faulty issues ($U = 1250, \rho = 0.5$). Participants exhibited a similar disposition to report faults irrespective of the severity of faulty issues in Scenario 2. Furthermore, within the faulty condition, participants displayed uniform trust levels across varying levels of faulty issues ($U = 1226, \rho = 0.49$). Similarly, in the repaired condition, participants exhibited consistent trust levels across different levels of faulty issues ($U = 1220, \rho = 0.49$). For the faulty condition, participants expressed an average confidence level of 3.9 points in their ability to identify faulty robots within the swarm. In the faulty condition, 95% of participants who reported faults were able to correctly identify the faulty UAV within the swarm, albeit with misidentifications among the normal UAVs. Moreover, 23% of the participants correctly identified the faulty robot without any errors.

4.3. Scenario III: Emergent Response

Figure 8 shows the results of the experiment under Scenario 3. The likelihood of participants reporting faults in both the faulty and repaired conditions exhibited a significant disparity in Scenario 3. Participants displayed a greater propensity to report faults in the faulty conditions compared to the repaired conditions ($U = 850, \rho = 0.09$). The mean trust levels in the faulty and repaired conditions were 2 and 4.22, respectively. Furthermore, participants were more inclined to attribute higher trust levels to the repaired condition as opposed to the faulty condition ($U = 480, \rho = 0.05$).

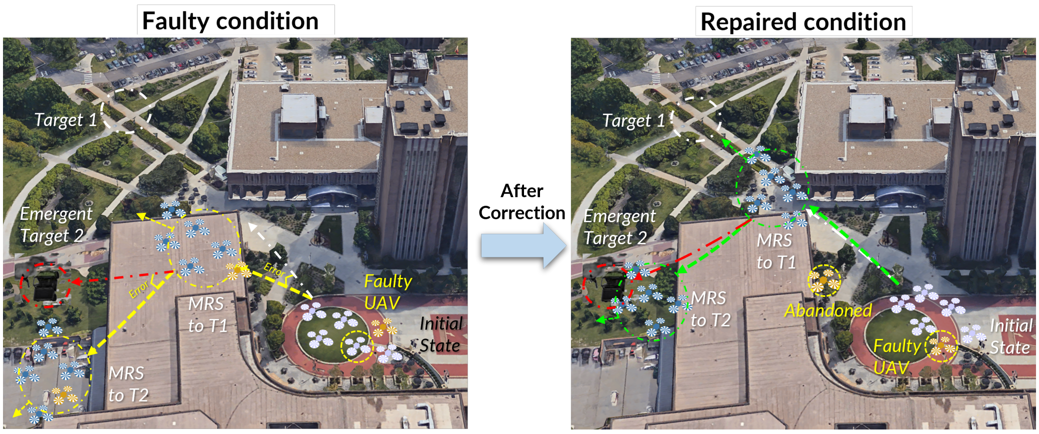

Figure 8. Experiment results for Case Study III: emergent response. In the faulty condition, the UAV swarm performs a task in the presence of a faulty robot; while applying the ***Trust-R***, the UAV swarm will restrict the influence of the faulty robot, which improves the performance of the UAV swarm.

In Scenario 3, concerning various levels of faulty issues, the probability of participants reporting faults in the faulty conditions was relatively similar ($U = 1050, \rho = 0.42$). Participants did not manifest a discernible variance in fault reporting across different faulty levels in Scenario 2. Both in the faulty and repaired conditions, participants exhibited uniform trust levels across varying faulty levels ($U = 1060, \rho = 0.42$ and $U = 1071, \rho = 0.43$ for faulty and repaired conditions, respectively). In the faulty condition, participants exhibited an average confidence score of 3.9 in identifying faulty robots within the swarm. In this condition, 90% of participants correctly identified the faulty UAV among the swarm when they reported a fault, while 22% of the participants identified the faulty robot without any misclassification errors when normal UAVs were erroneously identified.

5. Discussion

This study has delved into the potential application of ***Trust-R*** within the context of robot swarms under human supervision. In such scenarios, information exchange among neighboring robots plays a pivotal role in the control loop of each individual agent. The utilization of a weighted connection mechanism facilitates the translation of human trust into the information-sharing process, thereby imposing constraints on the dissemination of untrustworthy information, ultimately leading to performance enhancements for the entire team. In essence, human trust serves as an estimation of the capabilities of individual robots or the swarm as a whole. The implementation of a trust mechanism has demonstrated its capacity to bolster the performance of Unmanned Aerial Vehicles (UAVs) and foster improved collaboration between humans and robots. ***Trust-R*** serves as a crucial intermediary, bridging the gap between robot performance and human oversight, thus facilitating more dependable predictions of successful task completion.

The research indicates that the application of ***Trust-R*** enhances the performance of swarms and gains higher human trust. It is important to see trust-aware mechanisms in the context of human–robot collaboration. ***Trust-R*** works by making robot actions closer to what humans think should happen and fixing problems fast, improving the overall functionality and reliability of the system, which is vital for tasks requiring quick and accurate execution, such as disaster response or surveillance. ***Trust-R***'s objective focuses on improving task efficiency and better understanding and integrating trust dynamics within the human–swarm cooperation scenario. By embedding trust as a core component of the control algorithm, ***Trust-R*** enhances the build-up of better trust and belief between humans and groups of robots. Trust-R can be adopted in practical applications for various needs, such as environmental monitoring and search and rescue operations. Using ***Trust-R***, these systems can perform their jobs better and are more trustable, making them better for use in the real world. Additionally, the principles underlying ***Trust-R*** can inspire new models for human–swarm interactions, underscoring the importance of trust and the ability to rectify mistakes for effective collaboration.

6. Conclusions

This paper presents the methodology for enhancing human–swarm cooperation and reports the outcomes of an experimental investigation in which the application of ***Trust-R*** was employed to mitigate the impact of malfunctioning UAVs within the swarm. In Scenario 1, participants were unable to discern a significant difference between the faulty and repaired conditions ($\rho_1 = 0.46$). Their levels of trust in both conditions were similar, measuring at 3.34 and 3.53, respectively. However, in Scenarios 2 and 3, participants could readily distinguish between the faulty and repaired conditions ($\rho_2 = 0.04$, $\rho_3 = 0.09$). Following the implementation of ***Trust-R***, participants exhibited increased trust in the repaired condition compared to the faulty condition. In both normal and emergent scenarios, ***Trust-R*** substantially mitigated the adverse influence of malfunctioning UAVs on the swarm, achieving comparable effectiveness in guiding a human–supervised swarm toward dynamic task responses.

Looking ahead, our future research endeavors will encompass a comprehensive consideration of additional fault factors arising from system instability and environmental disturbances. This will enable the development of a precise trust-based methodology aimed at enhancing the performance of UAV swarms engaged in dynamic tasks. Furthermore, we intend to measure the trust levels assessed by human operators and the associated thresholds, shedding light on the intricate relationship between fault factors and human judgment. Simultaneously, challenges emerge when operating in environments replete with obstacles or overseeing swarms comprising a multitude of UAVs, primarily stemming from the inherent limitations of human cognitive capacities when it comes to multitasking and attending to multiple robots. Therefore, forthcoming research endeavors will be geared toward adapting robots to accommodate these constraints inherent in human cognition during human–swarm collaboration scenarios.

Author Contributions: Conceptualization, Y.G., Y.P., J.L., M.L., K.S. and R.L.; methodology, Y.G., Y.P., J.L., M.L., K.S. and R.L.; software, Y.G., Y.P. and R.L.; validation, Y.G., Y.P., J.L., M.L., K.S. and R.L.; formal analysis, Y.G., Y.P., J.L., M.L., K.S. and R.L.; investigation, Y.G., Y.P., J.L., M.L., K.S. and R.L.; resources, Y.G., Y.P., J.L., M.L., K.S. and R.L.; data curation, Y.G., Y.P., J.L., M.L., K.S. and R.L.; writing—original draft preparation, Y.G., Y.P., J.L., M.L., K.S. and R.L.; writing—review and editing, Y.G., Y.P., J.L., M.L., K.S. and R.L.; visualization, Y.G., Y.P., J.L., M.L., K.S. and R.L.; supervision, R.L.; project administration, R.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: This study was conducted in accordance with the Declaration of Helsinki and approved by the Kent State Institutional Review Board (IRB) (FWA 00001853, expires 2 September 2026). This article does not contain any studies with animals performed by any of the authors.

Informed Consent Statement: Informed consent was obtained from all subjects involved in this study. Human volunteers answered questionnaires online with consent for their participation; no actual humans were involved in the physical experiment. Written informed consent was obtained from the volunteers to publish this paper.

Data Availability Statement: The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of this study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

WMSR	Weighted Mean Subsequence Reduced
UAV	Unmanned Aerial Vehicle

References

1. Reynolds, C.W. Flocks, herds and schools: A distributed behavioral model. In Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques, Anaheim, CA, USA, 27–31 July 1987.
2. Jadbabaie, A.; Lin, J.; Morse, A.S. Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Trans. Autom. Control* **2003**, *48*, 988–1001. [CrossRef]
3. Amirkhani, A.; Barshooi, A.H. Consensus in multi-agent systems: A review. *Artif. Intell. Rev.* **2021**, *55*, 3897–3935 [CrossRef]
4. Wang, X.; Wang, Y. Co-design of Control and Scheduling for Human-Swarm Collaboration Systems Based on Mutual Trust. In *Trends in Control and Decision-Making for Human-Robot Collaboration Systems*; Springer: Cham, Switzerland, 2017; pp. 387–413.
5. Muir, B.M. *Operators' Trust in and Use of Automatic Controllers in a Supervisory Process Control Task*; National Library of Canada: Ottawa, ON, Canada, 2002.
6. Zanone, R.O.; Velni, J.M. Trust-based Performance Optimization for Human-Swarm Collaboration. *IFAC-PapersOnLine* **2023**, *56*, 571–576. [CrossRef]
7. McGuire, K.N.; De Wagter, C.; Tuyls, K.; Kappen, H.J.; de Croon, G.C.H.E. Minimal navigation solution for a swarm of tiny flying robots to explore an unknown environment. *Sci. Robot.* **2019**, *4*, eaaw9710. [CrossRef]
8. Soares, P.P.; de Souza, L.B.; Mendonca, M.; Palacios, R.H.C.; de Almeida, J.P.L.S. Group of Robots Inspired by Swarm Robotics Exploring Unknown Environments. In Proceedings of the 2018 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), Rio de Janeiro, Brazil, 8–13 July 2018; IEEE: Piscataway, NJ, USA, 2018.
9. Carpentiero, M.; Gugliermetti, L.; Sabatini, M.; Palmerini, G.B. A swarm of wheeled and aerial robots for environmental monitoring. In Proceedings of the 2017 IEEE 14th International Conference on Networking, Sensing and Control (ICNSC), Calabria, Italy, 16–18 May 2017; IEEE: Piscataway, NJ, USA, 2017.
10. Duarte, M.; Gomes, J.; Costa, V.; Rodrigues, T.; Silva, F.; Lobo, V.; Monteiro, M.; Oliveira, S.M.; Christensen, A.L. Application of swarm robotics systems to marine environmental monitoring. In Proceedings of the OCEANS 2016, Shanghai, China, 10–13 April 2016; IEEE: Piscataway, NJ, USA, 2016.
11. Wise, R.; Rysdyk, R. UAV coordination for autonomous target tracking. In Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit, Keystone, CO, USA, 21–24 August 2006.
12. Dobrokhodov, V.N.; Kaminer, I.I.; Jones, K.D.; Ghabcheloo, R. Vision-based tracking and motion estimation for moving targets using small UAVs. In Proceedings of the 2006 American Control Conference, Minneapolis, MN, USA, 14–16 June 2006; IEEE: Piscataway, NJ, USA, 2006.
13. Robla-Gómez, S.; Becerra, V.M.; Llata, J.R.; González-Sarabia, E.; Torre-Ferrero, C.; Pérez-Oria, J. Working together: A review on safe human-robot collaboration in industrial environments. *IEEE Access* **2017**, *5*, 26754–26773. [CrossRef]
14. Bauer, A.; Wollherr, D.; Buss, M. Human-robot collaboration: A survey. *Int. J. Humanoid Robot.* **2008**, *5*, 47–66. [CrossRef]
15. Crandall, J.; Goodrich, M.; Olsen, D.; Nielsen, C. Validating human-robot interaction schemes in multitasking environments. *IEEE Trans. Syst. Man Cybern. Part A Syst. Hum.* **2005**, *35*, 438–449. [CrossRef]
16. Dahiya, A.; Aroyo, A.M.; Dautenhahn, K.; Smith, S.L. A survey of multi-agent Human-Robot Interaction systems. *Robot. Auton. Syst.* **2023**, *161*, 104335. [CrossRef]
17. Walker, P.; Lewis, M.; Sycara, K. Characterizing human perception of emergent swarm behaviors. In Proceedings of the 2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Budapest, Hungary, 9–12 October 2016; IEEE: Piscataway, NJ, USA, 2016.
18. Capiola, A.; Hamdan, I.A.; Lyons, J.B.; Lewis, M.; Alarcon, G.M.; Sycara, K. The effect of asset degradation on trust in swarms: A reexamination of system-wide trust in human-swarm interaction. *Hum. Factors* **2022**, *66*, 1475–1489. [CrossRef]
19. Jones, K. Trust as an affective attitude. *Ethics* **1996**, *107*, 4–25. [CrossRef]
20. Lee, J.; Moray, N. Trust, control strategies and allocation of function in human-machine systems. *Ergonomics* **1992**, *35*, 1243–1270. [CrossRef] [PubMed]
21. Sadrfaridpour, B.; Saeidi, H.; Burke, J.; Madathil, K.; Wang, Y. Modeling and control of trust in human-robot collaborative manufacturing. In *Robust Intelligence and Trust in Autonomous Systems*; Springer: Boston, MA, USA, 2016; pp. 115–141.
22. Wang, X.; Shi, Z.; Zhang, F.; Wang, Y. Dynamic real-time scheduling for human-agent collaboration systems based on mutual trust. *Cyber-Phys. Syst.* **2015**, *1*, 76–90. [CrossRef]
23. Wang, Y.; Humphrey, L.R.; Liao, Z.; Zheng, H. Trust-based multi-robot symbolic motion planning with a human-in-the-loop. *ACM Trans. Interact. Intell. Syst. (TiiS)* **2018**, *8*, 31. [CrossRef]

24. Robinette, P.; Howard, A.M.; Wagner, A.R. Timing is key for robot trust repair. In *International Conference on Social Robotics*; Springer: Cham, Switzerland, 2015.
25. Schweitzer, M.E.; Hershey, J.C.; Bradlow, E.T. Promises and lies: Restoring violated trust. *Organ. Behav. Hum. Decis. Process.* **2006**, *101*, 1–19. [CrossRef]
26. Wang, N.; Pynadath, D.V.; Hill, S.G. Trust calibration within a human-robot team: Comparing automatically generated explanations. In Proceedings of the 2016 11th ACM/IEEE International Conference on Human-Robot Interaction (HRI), Christchurch, New Zealand, 7–10 March 2016; IEEE: Piscataway, NJ, USA, 2016.
27. Visinsky, M.L.; Cavallaro, J.R.; Walker, I.D. Robot fault detection and fault tolerance: A survey. *Reliab. Eng. Syst. Saf.* **1994**, *46*, 139–158. [CrossRef]
28. Khaldi, B.; Harrou, F.; Cherif, F.; Sun, Y. Monitoring a robot swarm using a data-driven fault detection approach. *Robot. Auton. Syst.* **2017**, *97*, 193–203. [CrossRef]
29. Christensen, A.L.; OGrady, R.; Dorigo, M. From fireflies to fault-tolerant swarms of robots. *IEEE Trans. Evol. Comput.* **2009**, *13*, 754–766. [CrossRef]
30. Tarapore, D.; Christensen, A.L.; Timmis, J. Generic, scalable and decentralized fault detection for robot swarms. *PLoS ONE* **2019**, *12*, e0182058. [CrossRef]
31. Tarapore, D.; Timmis, J.; Christensen, A.L. Fault detection in a swarm of physical robots based on behavioral outlier detection. *IEEE Trans. Robot.* **2019**, *35*, 1516–1522. [CrossRef]
32. Khadidos, A.; Crowder, R.M.; Chappell, P.H. Exogenous fault detection and recovery for swarm robotics. *IFAC-PapersOnLine* **2015**, *48*, 2405–2410. [CrossRef]
33. Park, H.; Hutchinson, S. A distributed robust convergence algorithm for multi-robot systems in the presence of faulty robots. In Proceedings of the 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hamburg, Germany, 28 September–2 October 2015; IEEE: Piscataway, NJ, USA, 2015.
34. Park, H.; Hutchinson, S. An efficient algorithm for fault-tolerant rendezvous of multi-robot systems with controllable sensing range. In Proceedings of the 2016 IEEE International Conference on Robotics and Automation (ICRA): Stockholm, Sweden, 16–21 May 2016; IEEE: Piscataway, NJ, USA, 2016.
35. Zhang, F.; Chen, W. Self-healing for mobile robot networks with motion synchronization. In Proceedings of the 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems, San Diego, CA, USA, 29 October–2 November 2007; IEEE: Piscataway, NJ, USA, 2007.
36. Saulnier, K.; Saldana, D.; Prorok, A.; Pappas, G.J.; Kumar, V. Resilient flocking for mobile robot teams. *IEEE Robot. Autom. Lett.* **2017**, *2*, 1039–1046. [CrossRef]
37. Rekleitis, I.M.; Dudek, G.; Milios, E.E. Multi-robot exploration of an unknown environment, efficiently reducing the odometry error. *Int. Jt. Conf. Artif. Intell.* **1997**, *15*, 1340–1345.
38. Pini, G.; Brutschy, A.; Scheidler, A.; Dorigo, M.; Birattari, M. Task partitioning in a robot swarm: Object retrieval as a sequence of subtasks with direct object transfer. *Artif. Life* **2014**, *20*, 291–317. [CrossRef]
39. Pini, G.; Brutschy, A.; Scheidler, A.; Dorigo, M.; Birattari, M. *Task Partitioning in a Robot Swarm: Retrieving Objects by Transferring Them Directly between Sequential Sub-Tasks*; IRIDIA, Université Libre de Bruxelles: Brussels, Belgium, 2012.
40. Liu, R.; Jia, F.; Luo, W.; Chandarana, M.; Nam, C.; Lewis, M.; Sycara, K.P. Trust-Aware Behavior Reflection for Robot Swarm Self-Healing. In Proceedings of the 18th International Conference on Autonomous Agents and Multi Agent Systems, Montreal, QC, Canada, 13–17 May 2019; International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, USA, 2019.
41. Liu, R.; Cai, Z.; Lewis, M.; Lyons, J.; Sycara, K. Trust Repair in Human-Swarm Teams+. In Proceedings of the 2019 28th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN), New Delhi, India, 14–18 October 2019; IEEE: Piscataway, NJ, USA, 2019.
42. Gupta, L.; Jain, R.; Vaszkun, G. Survey of important issues in UAV communication networks. *IEEE Commun. Surv. Tutor.* **2015**, *18*, 1123–1152. [CrossRef]
43. Olfati-Saber, R. Flocking for multi-agent dynamic systems: Algorithms and theory. *IEEE Trans. Autom. Control* **2006**, *51*, 401–420. [CrossRef]
44. La, H.M.; Sheng, W. Adaptive flocking control for dynamic target tracking in mobile sensor networks. In Proceedings of the 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, St. Louis, MO, USA, 10–15 October 2009; IEEE: Piscataway, NJ, USA, 2009.
45. Ren, W. Multi-vehicle consensus with a time-varying reference state. *Syst. Control Lett.* **2007**, *56*, 474–483. [CrossRef]
46. Saldana, D.; Prorok, A.; Sundaram, S.; Campos, M.F.M.; Kumar, V. Resilient consensus for time-varying networks of dynamic agents. In Proceedings of the 2017 American Control Conference (ACC), Seattle, WA, USA, 24–26 May 2017; IEEE: Piscataway, NJ, USA, 2017.
47. Quigley, M.; Conley, K.; Gerkey, B.; Faust, J.; Foote, T.; Leibs, J.; Wheeler, R.; Ng, A.Y. ROS: An open-source Robot Operating System. *Icra Workshop Open Source Softw.* **2009**, *3*.

48. Koenig, N.; Howard, A. Design and use paradigms for gazebo, an open-source multi-robot simulator. In Proceedings of the 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No. 04CH37566), Sendai, Japan; IEEE: Piscataway, NJ, USA, **2004**; Volume 3, pp. 2149–2154.
49. Buhrmester, M.; Kwang, T.; Gosling, S.D. Amazon's Mechanical Turk: A new source of inexpensive, yet high-quality data? *Perspect. Psychol. Sci.* **2016**, *6*, 3–5. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.