Trust-aware Control for Intelligent Transportation Systems

Mingxi Cheng, Junyao Zhang, Shahin Nazarian, Jyotirmoy Deshmukh, Paul Bogdan

Abstract—Many intelligent transportation systems are multi-agent systems, i.e., both the traffic participants and the subsystems within the transportation infrastructure can be modeled as interacting agents. The use of AI-based methods to achieve coordination among the different agents systems can provide greater safety over transportation systems containing only human-operated vehicles, and also improve the system efficiency in terms of traffic throughput, sensing range, and enabling collaborative tasks. However, increased autonomy makes the transportation infrastructure vulnerable to compromised vehicular agents or infrastructure. This paper proposes a new framework by embedding the trust authority into transportation infrastructure to systematically quantify the trustworthiness of agents using an epistemic logic known as subjective logic. In this paper, we make the following novel contributions: (i) We propose a framework for using the quantified trustworthiness of agents to enable trust-aware coordination and control. (ii) We demonstrate how to synthesize trust-aware controllers using an approach based on reinforcement learning. (iii) We comprehensively analyze an autonomous intersection management (AIM) case study and develop a trust-aware version called AIM-Trust that leads to lower accident rates in scenarios consisting of a mixture of trusted and untrusted agents.

I. INTRODUCTION

Intelligent transportation systems are effectively multi-agent systems (MAS), where both participants in the traffic as well as components of the transportation infrastructure can be modeled as agents that interact with each other [1]. For example, autonomous intersection management [2] consists of autonomous or semi-autonomous vehicles that interact with an intersection manager, while various systems such as adaptive platoons [3], cooperative highway merging [4], [5], and cooperative collision avoidance [6], [7] have interacting vehicles that can be modeled as MAS. In the basic versions of all such systems, the central focus is on the control algorithms required to achieve the desired coordination objective. However, increased level of autonomy renders such systems vulnerable to agents whose functional behavior does not respect the assumptions made by the coordination protocols. Agents can become compromised because they could be the subjects of a malicious attack or simply because they have defective sensors, actuators or control software, and thus threatens the entire system. The central question we investigate in this paper is: How can we guarantee safety and performance of a multi-agent transportation system, when participating agents are compromised?

While there has been significant emphasis on reasoning about the security and privacy of MAS applications [8]–[10], using ideas from control theory and cyber-security. These approaches tackle important problems such as secure state estimation, attack detection and mitigation, and system resilience. The view of most security-based approaches is binary: either an agent is safe or compromised, and the mitigation strategies are also thus restricted. We argue that a transportation system must remain operational even in the presence of compromised agents, and in order to do so, we need a way to quantify the level of trustworthiness of its constituent agents. Notions of trustworthiness have been studied for vehicular ad hoc networks (e.g., [11]–[13]). However, a formal definition and analysis of trustworthiness and how it can be systematically used to perform trust-aware control in MAS has received limited attention. In previous work we proposed a general framework to tackle this problem but with limited details [14]. In this work, we provide trust evaluation and trust-aware control in intelligent transportation systems through a detailed case study of autonomous intersection control.

What makes an agent trustworthy? While this is a nuanced question, we propose a mathematically precise definition of trustworthiness that relies on two key principles: (i) trusted or reliable agents obey the control actions suggested by a central (or distributed) coordinator. (ii) trusted agents do not provide false information.

We assume that a decision-making component in the transportation infrastructure has an associated trust authority (TA) that can evaluate an agent’s trustworthiness through two kinds of observations: direct observations and indirect observations gleaned from other sources (e.g. other vehicles, other components of the infrastructure such as road-side units). We call the latter local trust authorities (LTAs). Each observation represents evidence that enhances the TA’s belief or disbelief in each agent (depending on whether the evidence respectively indicates a desired or undesired behavior). For agents that the TA has no opportunity to observe, the TA has no belief or disbelief in the agent, but instead has uncertainty about its trustworthiness. These ideas are rigorously developed in an epistemic logic called subjective logic that we employ for trust quantification and analysis.

We provide a conceptual depiction of a trust-aware MAS in Fig. 1. The system consists of a number of distributed agents, each with their own sensors, actuators and local control algorithms. The coordination among the agents is achieved through either a centralized or a distributed control algorithm. Traditionally, the only input to such a controller is the global specification of desired behavior (encoding both mission objectives for the MAS and safety constraints). In our framework, we provide a trustworthiness score for each agent that is assumed to be stored on a secure cloud/edge-based server. This score is computed by a TA from observations of agents (reported through the agents’ sensing and communication modules). The trust-aware control algorithm uses the global specification describing desired coordination behavior and trustworthiness scores to give control actions, which the agents can choose to act on through their decision-making, planning and actuation modules.

The authors are with the Department of Electrical and Computer Engineering, University of Southern California, Los Angeles, CA, 90089, USA (corresponding e-mail: pbogdan@usc.edu).
In this paper, we perform a detailed analysis of our trust quantification and trust-aware control framework on the autonomous intersection management (AIM) system. AIM is a visionary system proposed by Dresner and Stone [15] to reduce the traffic load of the current transportation infrastructure and the resulting delays [16]. It improves the intersection efficiency under the assumption that vehicles communicate with AIM and strictly follow the AIM controller’s instructions. Compared to conventional intersection control mechanisms such as stop signs and traffic signals, AIM policies provide consistently high traffic throughput while ensuring the safety of the traffic participants [16].

In real-world scenarios, driving is fraught with uncertainties arising from fallible as well as compromised human drivers, defective sensors or actuators, noisy sensing environments, and vehicle-to-infrastructure (V2X) communication. While existing AIM work focuses on extending to scenarios with human-driven or semi-autonomous vehicles [17], [18], it does not account for the possibility of such systemic or adversarial uncertainties. Thus, untrustworthiness arising out of uncertainty can invalidate the benefits of AIM. A key contribution in this paper is to showcase AIM as a detailed case study to highlight our algorithms for trust quantification and trust-aware control for MAS. We show how we can modify AIM to obtain AIM-Trust that uses trustworthiness scores of traffic participants to generate trust-aware control actions. We develop a reinforcement learning (RL) framework, where the action space for the RL problem is the allocation of space-time buffers for cars to navigate the intersection modulated by their trustworthiness, and the reward space is defined in terms of collision freedom and intersection throughput. Our formulation allows AIM-Trust to explore a trade-off between performance (throughput) and safety (collision avoidance). The main contributions are as follows:

- We propose a method to quantify trustworthiness scores for individual agents in a MAS.
- We provide a general and abstract framework for incorporating trust in generating control actions that guarantee the satisfaction of a global MAS specification.
- We showcase our trust quantification and trust-aware control methodology through a detailed case study of Autonomous Intersection Management (AIM).
- We demonstrate how to embed the trustworthiness scores in a RL-based policy synthesis procedure.
- Our empirical results show higher safety (at least 18.18% and up to 89.28% collision reduction) and efficiency (15.53% throughput improvement on average) of intersection management under the AIM-Trust controller compared to classical AIM that does not account for trustworthiness.

II. PRELIMINARIES

Multi-agent System Model. A MAS is a collection of dynamic agents interacting with each other and with a controller (which could be centralized or distributed). Each agent \( A \) can be described as a tuple \((\text{id}, X, U, Y, T)\), where \( \text{id} \) is a unique positive integer, \( X \) is a set of (internal) states of the agent, \( U \) is a set of inputs to the agent, \( Y \) is a set of observations provided by the agent, and \( T \) is a nondeterministic transition relation, subset of \( X \times U \times X \times Y \). For every internal state \( x \in X \), the agent reads an input \( u \in U \) and nondeterministically transitions to some state \( x' \in X \) providing output \( y \in Y \). We use the set \( U \) to model commands originated from the controller (projected on this agent) and sensor inputs for the agent, while the set \( Y \) models the information shared by the agent about its own state with other agents and the controller. We remark that agents can also be modeled as having a stochastic transition relation, where \( T \) describes a probability distribution of the next state and output conditioned on the current state and input.

We assume that the set of agents interacting with the controller is dynamic and in any episode (defined as a finite period of time), agents can enter or leave the episode. An episode captures a time slice of the operation of the MAS. The controller itself is also modeled as a tuple \((Q, \Sigma, \Gamma, \Delta)\), where \( Q \) are the internal controller states, \( \Sigma \) is the set of inputs read by the controller, \( \Gamma \) are the command actions published by the controller to all agents that are within the episode. The controller policy \( \Delta \) is a function from \( Q \times \Sigma \) to \( Q \times \Gamma \) that maps each controller state and input to its next state and publishes output actions for the agents.

Global System Specification. We assume that the global state space of the MAS is the product of the state spaces of the controller and the agents involved in an episode. Essentially, at time \( t \), the controller observes \( \sigma(t) \) (which is a collection of \( y(t) \) values of the available agents in the episode) and publishes a (possibly empty) control action \( \gamma(t+1) \), and each agent processes the control action \( \gamma(t) \) (projected to its \( \text{id}, \text{i.e.}, u(t) \)) and outputs \( y(t+1) \). Both the controller and the agents also move to their respective next (internal) states upon executing internal actions. We assume that a global system specification is a property defined on the space of the internal state trajectories of the agents. Such a property can be easily specified in a multi-agent extension to any standard temporal logic such as Signal Temporal Logic (SLT) [19].

Background on Subjective Logic. To enable trustworthiness evaluations, we utilize a probabilistic epistemic logic known as subjective logic (SL). SL is used in social systems to quantify opinions and model trust relationships among humans. In general, SL is suitable for modeling and analyzing situations involving uncertainty and relatively unreliable sources [20] and provides representations for opinions, observations, and
trust relationships. In our MAS context, each agent $A$ is associated with an opinion and a corresponding trust value, which are evaluated by the TA following specific rules.

**Definition 2.1 (Opinion [20]):** Let $r$ be a quantity indicating the magnitude of positive evidence obtained by the TA while observing the behavior an agent $A$, and let $s$ be an analogous quantity indicating the magnitude of negative evidence. The binomial opinion of an $A$ according to the TA is the set $W_A = \{ b_A, d_A, u_A, a_A \}$, which consists of belief mass ($b_A$), disbelief mass ($d_A$), uncertainty mass ($u_A$), base rate ($a_A$), where $b_A = \frac{s}{r + s + t + u}$, $d_A = \frac{t}{r + s + t + u}$, $u_A = \frac{r}{r + s + t + u}$, $\omega = 2$ is a default non-informative prior weight, satisfying the condition $b_A + d_A + u_A = 1$.

Since we may have LTAs helping TA and they can observe agents and form opinions about a specific agent individually, we sometimes need to combine LTAs’ opinions. Suppose our LTAs are trustworthy, then we can use cumulative fusion operator to combine opinions as follows:

**Definition 2.2 (Cumulative Fusion [20]):** Suppose we have two LTAs $A$ and $B$, and they form opinions of agent $A$ as $W_A = \{ b_A^A, d_A^A, u_A^A, a_A^A \}$ and $W_B = \{ b_A^B, d_A^B, u_A^B, a_A^B \}$. The cumulative fusion, i.e., the combined opinion, $W_A^C$ of these two opinions is calculated as follows: For $u_A^C \neq 0$ or $u_B^C \neq 0$:

$$
\begin{align*}
\begin{cases}
    b_A^C = (b_A^A u_B^B + b_A^B u_A^B)/(u_A^A + u_B^B - u_A^A u_B^B), \\
    d_A^C = (d_A^A u_B^B + d_A^B u_A^B)/(u_A^A + u_B^B - u_A^A u_B^B), \\
    u_A^C = (u_A^A u_B^B - u_A^A u_B^B)/(u_A^A + u_B^B - u_A^A u_B^B), \\
    a_A^C = (a_A^A u_B^B + a_A^B u_B^B)/(a_A^A + a_A^B - a_A^A a_A^B). \\
\end{cases}
\end{align*}
$$

(1)

III. TRUST INFRASTRUCTURE

We consider a MAS consisting of a mixture of trustworthy and untrustworthy agents and define a TA as an augmentation of the controller $C$. Subject to an agent $A$, the TA observes its behavior $Y$ and extracts knowledge (also known as opinion in SL) from observations (also known as evidence) and evaluates $A$’s trustworthiness as follows:

**Definition 3.1 (Trustworthiness):** Given a TA and a specified agent $A$ in the MAS, the trustworthiness of $A$ assessed by TA is defined as $p_A = b_A T_A + u_A A_A \ast T_A$ [21], where $b_A$, $u_A$, and $a_A$ are belief mass, uncertainty mass, and base rate, respectively.

Besides the centralized trust authority TA, there might exist distributed local trust authorities (LTAs) that help the TA to enlarge observation range. Since in some cases, a single centralized trust manager cannot cover all observation areas (e.g., in transportation systems, imagine the TA as the department of motor vehicles and the TA cannot directly observe all roads, hence, LTAs like road side units serve as helpers and enlarge the observation range of TA).

We envision a cloud-based (or edge-based) architecture as shown in Fig. 2 where centralized trust authority TA manages trust histories of agents in a MAS in a hash table $\mathcal{H}$ on cloud. TA regularly sends updates to pull records from cloud. LTAs report to TA but not directly talk to the cloud. In each trust authority, an evidence measurement framework is embedded to evaluate the behavior of agents to be positive $r$ or negative $s$ evidence. Then the evidences are used to calculate the opinion as defined in Definition 2.1. Following our transportation system example, road side units capture the undesired behaviors ($s$) of vehicle $A$ and report to TA; the TA then updates the opinion of $A$ using cumulative fusion operator defined in Definition 2.2 and push this record to the cloud and update $\mathcal{H}$ to decrease the trustworthiness score of $A$. In the end, the trustworthiness measurement comes down to control algorithms in $Cs$ as shown in Fig. 1. The control algorithm that takes trust as input is trust-aware, and makes decisions with considerations of agents’ historical behaviors.

**Trust-aware Control in MAS.** In the context of MAS, such as adaptive cruise control system [22], multi-agent autonomous traffic management [23], and air/drone traffic control system [24] the safety and behavior of one or a subset of agents affects the efficiency and safety of the whole system. Such systems are usually vulnerable to attacks that insert malicious agents into the system for various purposes. In these cases, a subjective measurement is a must to identify malicious and untrustworthy agents. Therefore, we propose to augment the controller’s input space $\Sigma$ to include the trustworthiness ($p$) of agents calculated by the TA and enhance the controller policy $\Delta$ to be trust-aware.

![Fig. 2: A four-way intersection. Color-shaded areas represent the space-time buffers for each vehicle. Trustworthy vehicles have tight buffer since they are expected to obey the instructions with small error. Untrustworthy vehicles have large buffer because it is highly likely that they will act differently than instructed. Dark red area represents a collision warning in simulated trajectories.](image)

**IV. CASE STUDY IN AUTONOMOUS INTERSECTION MANAGEMENT**

**Autonomous Intersection Management (AIM).** The intersection traffic in AIM is a simplified version of real-world intersection traffic. Fig. 2 illustrates a four-way intersection example with three lanes in each road leading to an intersection area $I$ (marked by white dotted rectangle). A vehicle agent $A \in V$ on the road traveling towards but not already in the

---

1 For example, if the agent violates (resp. satisfies) a behavioral specification, $s$ (resp. $r$) would quantify the degree of violation (resp. satisfaction). For example if the global specifications is an STL formula $\phi$, $s$ (resp. $r$) is the (magnitude of) the robust satisfaction value of $\phi$ by the agent’s behavior.
interconnection \( I \) is considered to be on the AIM map \( M \). Any \( A \in M \) communicates with the intersection manager (IM) \( C \) by sending a request \( y(t) \) which consists of the vehicle identification number, vehicle size, predicted arrival time, velocity, acceleration, arrival and destination lanes, and then receives an instruction \( u(t) \). The IM calculates the trajectory of \( A \), makes a grant or reject decision \( \gamma(t) \) and sends the decision to \( A \). \( C \) rejects a request if there exist conflicts in the simulated trajectories. If \( C \) approves the request, \( A \) is responsible for obeying the instructions to enter and drive through \( I \). If \( C \) rejects the request, \( A \) has to resend the request and await further instructions.

Important assumptions in AIM are as follows: (i) For all \( A \in V \), they follow the instructions of \( C \) strictly within an error tolerance. This restriction guarantees safety by simulating trajectories and rejecting conflicting requests. (ii) Each \( A \in V \) is associated with a static buffer size with a minimum of value 1. The buffer size indicates the time-space reservation of \( A \). Trajectories are defined as conflicting if buffers of at least two vehicles overlap (marked as dark red in Fig. 2 where shaded area represents the buffer of each vehicle). The larger the buffer size, the higher the safety, and the lower the efficiency or throughput. Note that in conventional AIM [15], [17], all vehicles' buffer sizes are set to 1 and this preserves collision-freedom because of assumption (i). However, assumption (i) can be invalid as compromised agents can act recklessly and disobey instructions, which can lead to collisions in \( I \). The small static buffer size in assumption (ii) intensifies this situation.

**AIM-Trust.** Since in the real world, many vehicles can be malicious and violate AIM assumptions, we associate every IM with a TA based on our SL-based trust evaluation framework. The TA uses the trustworthiness table \( H \) to obtain the trustworthiness for each vehicle in \( M \). It uses these values to make better trajectory approval decisions. This framework is called AIM-Trust. We assume that TA-augmented IMs (TA-IMs) communicate with each other and they all persistently maintain \( H \) through appropriate synchronization mechanisms to maintain data coherence. Each TA-IM maintains a (coherent) local copy with part of \( H \) for efficiency and scalability. In addition, road side units (RSUs) serve as LTAs and provide coverage in places between the intersections to track trustworthiness of vehicles. AIM-Trust has two collision avoidance mechanisms: (i) a vehicle surveillance system to identify untrustworthy behavior of vehicles (within \( M \)), and (ii) an intelligent trust-based buffer adjusting mechanism to help decrease collision risk while maintaining high throughput. Fig. 2 and Algorithm 1 show the operation details. Note that if there is no untrustworthy vehicle in \( M \), AIM-Trust will reduce to original AIM algorithm with fixed buffer size 1 to ensure efficiency.

The TA-IM in AIM-Trust discriminates incoming vehicles into three bins: unprocessed, approved, and safe (see Fig. 2). A vehicle with its request unapproved by TA-IM is unprocessed. Once its request is approved, the vehicle is approved and under the surveillance of TA-IM for a few time steps. If the vehicle behaves well, then it becomes safe and the surveillance ends. However, if the vehicle violates the approved trajectory with an intolerable error, then the vehicle goes back to unprocessed bin and TA-IM starts processing its requests all over again. Detailed state transition process can be found in Algorithm 1. Compared to classical AIM where IM stops interacting with a vehicle once the request is approved, AIM-Trust includes a trust-based approve-observe process to decide whether to revoke and remake the approval decision.

### Algorithm 1: AIM-Trust algorithm

Vehicle \( A \) sends a request to TA-IM \( C \), which responds based on simulated trajectories.

| Pre-process | Post-process |
|-------------|--------------|
| Pre-process \( y(t) \) for new reservation. \( \triangleright \) Same as AIM | Send the decision to \( A \). \( \triangleright \) Same as AIM |
| Trustworthiness \( p^A \leftarrow trust\_calculator(id^A) \) | \( \xi^A \leftarrow unprocessed \) |
| Vehicle status \( \xi^A \leftarrow unprocessed \) | |
| State Transition | |
| Buffer size \( a^A \leftarrow buffer\_calculator(id^A, e^A, o^A, p^A) \) | Eq. (2) |
| Decision \( \triangleright \) AIM\_control\_policy\( (a^A) \) \( \triangleright \) Same as AIM | Eq. (2) |
| Post-process | Go to Pre-process; |
| If Decision \( \triangleright \) Approve then | |
| \( \xi^A \leftarrow approved, run surveillance(id^A) \) | |
| if surveillance\( (id^A) = malicious \) then | |
| Go to Pre-process; | |
| \( \xi^A \leftarrow safe, p^A \leftarrow trust\_calculator(id^A) \) once \( A \) exits \( M \). | |

**Trust Calculation.** In AIM-Trust, TA-IM maintains a trustworthiness \( t \) / opinion table \( H \) and updates \( H \) by either communicating with RSUs or considering new evidences via cumulative fusion operator \( \odot \) as shown in Algorithm line 3 and 13. The detailed procedure is shown in Algorithm 2. The first trustworthiness update happens when \( A \) enters \( M \) and sends requests to manager \( C \):

\[
W_A^c = \begin{cases} 
W_A^{UN}, & \text{if vehicle } A \text{ is unknown}, \\
W_A^{p}, & \text{if TA-IM(s) } C \text{ knows vehicle } A, \\
W_A^{LTA}, & \text{if road side unit knows vehicle } A, \\
W_A^{LAT\odot C}, & \text{if both LTA(s) and } C \text{ know vehicle } A. 
\end{cases}
\]

(2)

**Case I.** “A is unknown” represents that the vehicle does not have a record in \( H \).

**Case II.** “C knows A” represents that \( H \) has an entry of \( A \) and \( A \) has not been picked up by RSUs after previous record update.

**Case III.** Vehicle \( A \) enters \( M \) and \( C \) receives LTA’s report about \( A \). Since RSUs are only activated when undesired behavior happens, we expect \( W_A^{LTA} \) to be a negative opinion.

**Case IV.** RSU reports about \( A \), which already has a record in \( H \), hence, we use a cumulative fusion operator \( \odot \) to merge these two opinions together.

This first updates of \( W_A^{c} \) and \( p^A \) are now completed and then used by buffer adjustment agent as shown in Algorithm

\(^2\)Note that in AIM-Trust, we denote a TA-augmented IM as \( C \) for simplicity, while it is in fact the combination of a TA and an IM.

\(^3\)We assume \( W_A^{UN} = \{1, 0, 0, 0, 0, 0, 0\} \) to represent the maximum belief based on autoepistemic logic [25] (the vehicle is not reported to be untrustworthy, so it is trustworthy).
Algorithm 2: trust_calculator(id\(i\))

1. if \(\xi_{id}^t = \text{unprocessed}\) then \(\triangleright A\) enters \(\mathcal{M}\)
2. \(W_{A}^E \leftarrow \text{Eq. 2}\) \(\triangleright y(t)\) approved and surveillance(id\(i\)) = malicious
3. else if \(\xi_{id}^t = \text{approved}\) then \(\triangleright y(t)\) approved and surveillance(id\(i\)) \neq \text{malicious}
4. \(W_{A}^E \leftarrow \text{Def. 2.1}\) \(\triangleright\) Evidence is collected before \(I\)
5. \(W_{A}^E \leftarrow W_{A}^E\) \(\triangleright\) Evidence is collected in \(I\)
6. else if \(\xi_{id}^t = \text{safe}\) then \(\triangleright y(t)\) approved and surveillance(id\(i\)) \# malicious
7. \(W_{A}^E \leftarrow \text{Def. 2.1}\) \(\triangleright\) Evidence is collected in \(I\)
8. \(W_{A}^E \leftarrow W_{A}^E\) \(\triangleright\) \(A\) exits \(\mathcal{M}\)
9. return \(\rho_{id}^t \leftarrow \rho_{id}^t = b_{A}^t + u_{A}^t \ast \sigma_{A}^t\) \(\triangleright\) Definition 3.1

Evidence Measurement Framework. Exploiting the STL formalism, we define a set of rules to specify a driving behavior to be desired or undesired, i.e., quantify positive \(s\) or negative \((\bar{s})\) evidence for trust estimation. Desired / undesired behavior contribute to positive / negative evidence; hence, they contribute to increasing / decreasing of trustworthiness of an agent. Before the target driver approaches \(\mathcal{M}\), RSUs that have observed the target vehicle assess the (undesired) behavior and generate (negative) evidence. When the target vehicle arrives at the intersection, AIM-Trust uses a self-embedded behavior measurement system to quantify the target’s behavior based on trajectory and collision status.

Evidence Evaluation at Road Side Units. Suppose an RSU observes vehicle \(A\)’s trajectory and velocity, and the desired behavior is defined by a set of rules, e.g., driving within one lane with negligible deviation and under the designated speed limit. Hence, we define these properties formally as follows: Subject to \(A\), suppose the true trajectory observed by RSU is \(\sigma_{tr}\), the requested (or approved) trajectory is \(\gamma_{tr}\), the reported trajectory is \(\hat{\gamma}_{tr}\), and the negligible deviation or error is \(\epsilon_{tr}\). Similarly, the observed speed of the vehicle is \(\sigma_{sp}\), and the designated speed is \(\gamma_{sp}\), the reported speed is \(\hat{\gamma}_{sp}\), and the negligible error is \(\epsilon_{sp}\). We quantify \(r\) and \(s\) as follows:

\[
\begin{align*}
\text{If } & (\sigma_{tr}, t) \models \varphi \land (\sigma_{sp}, t) \models \psi; \\
\text{Then, } & (r = r + \beta_1, \text{if } (\sigma_{tr}, t) \models \varphi \land (\sigma_{sp}, t) \models \psi; \\
& s = s + \beta_2, \text{otherwise.} \\
\end{align*}
\]

Note that in a MAS where all agents are honest, the agent reported \(y\) is the same as controller observed \(\sigma\).

\[
\varphi \equiv G_{[t_1, t_2]}((\sigma_{tr}(t) - \gamma_{tr}(t)) \leq \epsilon_{tr} \land (\sigma_{sp}(t) - \gamma_{sp}(t)) \leq \epsilon_{sp}) \tag{4}
\]

\[
\psi \equiv G_{[t_1, t_2]}((\sigma_{sp}(t) - \gamma_{sp}(t)) \leq \epsilon_{sp} \land (\sigma_{sp}(t) - \gamma_{sp}(t)) \leq \epsilon_{sp}) \tag{5}
\]

These equations indicate that the true (observed) trajectory / speed of a vehicle should not deviate from (i) the requested trajectory / speed and (ii) the reported trajectory / speed by more than \(\epsilon_{tr} / \epsilon_{sp}\) in time interval \([t_1, t_2 + t + t_2]\), where \(t_1, t_2, \epsilon_{tr}, \epsilon_{sp}, \beta_1\), and \(\beta_2\) are hyper parameters. \(\beta_1, \beta_2 \in \mathbb{Z}^+\) are positive integers correlated with values of \(\varphi(t) - y(t)\) and \(\psi(t) - y(t)\), which indicates that the more proper / improper the behavior is, the bigger the reward / penalty is.

Evidence Evaluation at TA-IM. When vehicles are under the surveillance of AIME-Trust before entering \(I\) (Algorithm 1 line 11), the evidence measurements in Eq. 3 are used to quantify the positive and negative evidence. If negative evidence is observed, it means the vehicle violates the approved trajectory by an intolerable error. For AIME-Trust, when TA-IM approves \(A\)’s requested trajectory, \(y = \gamma\). Once vehicles enter \(I\), a new set of rules are used to take into account the collision status of vehicles and collisions in \(I\): \(r = r + \beta_1\), if the vehicle follows the approved trajectory and no collision happens; otherwise \(s = s + \beta_2\).

RL-based Buffer Adjustment Agent. AIM-Trust operates under the assumption that there may exist untrustworthy vehicles who would not follow instructions from TA-IM. Under such scenarios, the agents can not execute potential evasive maneuver to avoid the collisions, thereby these malicious agents with fixed and small buffer size will threat others agent, even the whole system. Then, how do we determine optimal buffer size for agents with different trust values? In order to assess this, and to have a buffer allocation policy, we use reinforcement learning (RL) to explore the unknown environment and figure out the appropriate buffer sizes. In this section, we define the RL formulation (deep Q-learning [26]) including definitions of states, actions, and rewards. In deep Q-learning, the neural network is approximating a Q-learning table, where each entry in the table is updated by \(q(s_i, a_i) \leftarrow q(s_i, a_i) + \alpha [r_{i+1} + \gamma \max_a q(s_{i+1}, a) - q(s_i, a_i)]\) [27], where \(s_i\) is state, \(a_i\) is action, \(r_{i+1}\) is reward, \(\alpha\) is learning rate, and \(\gamma\) is discounting factor.

State–State Transition–Action Spaces. We model a four-way intersection with three lanes in each direction as shown in Fig. 2. To simulate the real world scenarios, we explicitly allow vehicles on each lane to either go straight, turn left or right. We define our states as \(s_i = (id_i^t, e_i^t, o_i^t, p_i^t, ..., id_i^{t}, e_i^{t}, o_i^{t}, p_i^{t})^T\), where \((id_i^t, e_i^t, o_i^t, p_i^t)\) are the vehicle identification number, starting point, requested destination, and trustworthiness of vehicle \(i \in [1, n]\) at time \(t\). In each training time step \(t\), vehicles pass through \(I\) and fully exit \(\mathcal{M}\). Within one episode, there are in total \(\tau\) steps, which represent that the \(n\) vehicles pass through \(\tau\) intersections. \(\forall i, t, (e_i^{t}, o_i^t)\) are randomly generated by the simulator, while \(p_i^{t}\) is continuously updated based on Algorithm 2. The state transition equations for the environment are defined as: \((id_i^{t+1} \leftarrow id_i^t; p_i^{t+1} \leftarrow trust_calculator(id_i^t); e_i^{t+1}, o_i^{t+1} \leftarrow Random(e_i^{t}, o_i^t))\), where \(Random(\cdot)\) is the random starting point and destination generator in simulator. The action is defined as \(a_t = \) 5In SL, \(\beta_1\) and \(\beta_2\) usually takes value of 1.
(a_1^n, a_2^n, ..., a_i^n)^T$, where $a_i^t$ is the buffer size of vehicle $i$ at time $t$. Neural network makes prediction by assessing vehicles’ positions, trustworthiness, and requests.

**Reward Function.** The goal of our RL agent is to operate the intersection with lowest collision rate and high throughput. It is known that throughput is sensitive to buffer size, i.e., large buffer size harms throughput. We take safety as our primary consideration and improve the throughput under the premise of safety. Therefore, the reward function reads:

$$r_i^t = \begin{cases} 
1 + \lambda(b_{th} - a_i^t), & \text{if no collision}, \\
-(\tau - 1) + [1 + \lambda(b_{th} - a_i^t)], & \text{otherwise},
\end{cases}$$

where $b_{th}$ is a hyper parameter indicating a reasonable upper bound buffer size, $\lambda$ is a hyper parameter balances the collision and throughput. The vehicle is removed once it collides and not blocking $\mathcal{I}$, and is put back in $\mathcal{M}$ in the next step. A training episode contains $\tau$ steps, and an episode ends once the maximum $\tau$ step size is reached. The formulation of $r_i^t$ indicates that we want the buffer size to be as small as possible to increase the throughput while penalizing collisions.

**V. EXPERIMENTAL RESULTS FOR AIM**

**Experiment Setup.** We consider in an RL training episode, $n = 10$ vehicles pass through $\tau = 10$ intersections and we monitor the collisions occurring within these $n\tau = 100$ passages as $c$. For each intersection (or step) $t$ in an episode, $n$ vehicles enter and leave the intersection following randomly picked start points, $[e_i^1, ..., e_i^n]$, and destinations, $[a_i^1, ..., a_i^n]$. We generate 1 set of starting points and destinations as training set, and generate 10 independent sets as a test set. We consider two performance metrics: collision rate $\frac{n}{n\tau}$, and throughput $\frac{n}{\tau(T - 2)}$, where $T$ is the time (in seconds) elapsed in one episode. All simulations are done in the AIM simulator [28], and we provide video demonstrations in [29].

**Baselines.** We first compare our proposed AIM-Trust with 3 AIM family baselines: the original AIM algorithm with fixed buffer size 1, namely AIM-1; the modified AIM algorithm with fixed averaging buffer size, namely AIM-Fix (we manually select the fixed buffer size for this baseline to force it perform similarly as AIM-Trust in terms of collision rate, then compare the throughput with AIM-Trust); and a variation of AIM-Trust without considering the trust factor, $p$, in the state space, namely AIM-RL.

In addition, we compare AIM-Trust with traffic light-based intersection control methods which are not in AIM family. We follow [30] to construct a deep reinforcement learning (DRL)-based traffic light cycle control method as $C$ to operate the intersection. We denote this method as TIM-RL, i.e., traffic-light intersection management based on RL. Since TIM methods focus on operating the intersection with high efficiency without considering untrustworthy agents, they have no collision avoidance mechanism. To make the baseline more competitive in the scenarios involving untrustworthy vehicles, we propose two enhanced TIM methods: (i) TIM-Trust, which includes trustworthiness $p_i^t$ in TIM-RL’s state space, and (ii) TIM-Trust 2.0, which includes trustworthiness and has collision penalties in reward function. Furthermore, we include a fixed cycle traffic light (no RL agent involved) to replicate conventional traffic light control method for comparison, which is denoted as TIM-Fix.

**Collision Results.** In this section, we first present collision comparison results in AIM family as shown in Fig. 3b. We vary the percentage of untrusted vehicles from 20% to 100%. Since RL training embeds the randomness from initialization naturally, we train AIM-Trust 10 times and report the mean-variance results to show the stability. Fig. 3b shows that AIM-Trust decreases the collision numbers drastically compared to AIM-1 (see Table 1 for detailed numerical results). Since AIM cannot deal with the violation of assumption (i) (as described in Section IV), the small and fixed buffer size leads to high collision rate. The more untrustworthy vehicles in the system, the more the collisions. AIM-Trust’s adjustable buffer size helps to decrease the collision rate and maintains stable low collision rate even when all vehicles are untrustworthy.

In order to examine the effectiveness of the trustworthiness, we make a baseline AIM-RL by taking out the trust factor from AIM-Trust. Except the trust factor, AIM-RL is exactly the same as AIM-Trust with adjustable buffer size to decrease the collision rate. In addition, we control the training process of AIM-RL and AIM-Trust to be the same to ensure fair comparison. The experimental results in Fig. 3b and Table 1 demonstrate that the trustworthiness of a vehicle is key to infer the appropriate buffer size. To investigate the convergence and robustness of the AIM-Trust agent, we consider the collision performance of AIM-Trust in test set and training set is consistent as shown in Fig. 3b, which indicates that pre-trained AIM-Trust performs well in unseen traffic scenarios.

Next, we show the performance of AIM-Trust compared with non-AIM methods, TIM-Fix, TIM-RL, TIM-Trust, and TIM-Trust 2.0, in Figure 4 and Table 1. Compared with conventional traffic light-based intersection control methods, AIM-Trust is advantageous since it considers the uncertainty and trustworthiness of vehicles, and decreases collision rate by foreseeing the potential trajectories of trustworthy and untrustworthy vehicles. To demonstrate the significance and advantage of proposed trustworthiness of agents, we enhance TIM-RL and reveal that trust-based methods, TIM-Trust and TIM-Trust 2.0, beat TIM-RL in terms of collision rate in all scenarios. Experimental results confirm that in a MAS, when there exist untrustworthy agents, trustworthiness is important for control algorithms to infer the involved uncertainty.

**Throughput Results.** In addition to collision rate, we compare the throughput between AIM-Trust and AIM-Fix. For a fair comparison, we let buffer size of AIM-Fix to be [9, 9.5, 11, 13, 21.5] under 20% to 100% untrusted vehicles such that AIM-Fix has similar (slightly larger) collision rates as AIM-Trust. Then, under similar collision rate, we compare the throughput. As shown in Fig. 3b and Table 1, AIM-Trust’s throughput improvement demonstrates that the RL-based buffer adjustment not only decreases the collision rate, but also benefits the throughput. Compared to AIM-RL, AIM-Trust on average achieves higher throughput in all cases. Note that the collision rate of AIM-Fix is higher than AIM-Trust and based on throughput calculation, high collision rate actually gives advantages. With different collision rate, the comparison of throughput is unfair since the sacrifice of safety generates better performance in throughput (note that in simulation we remove the collision vehicle immediately from the map and does not effect the traffic flow). To compare with AIM-RL fairly, we can compare with 80% untrusted
we consider collision and throughput via balancing factor $\lambda$ and note in AIM-Trust and AIM-RL, the buffer size ranges from 0 to 16 in cases with 20% to 60% untrusted vehicles, while it ranges from 5 to 21 in cases with more untrusted vehicles (since the upper bound of 16 is not enough for RL agents to learn a good collision avoidance strategy). This change of action space causes the discontinuity of trends in terms of collisions and throughput from 60% and 80% cases. Through these minor modifications, AIM-Trust also learns to avoid collision completely, but with higher training. On average, the resulting buffer sizes of AIM-Trust compared to AIM-1. However, it cannot guarantee collision-free due to the safety and throughput dilemma. In AIM-Trust, compared to AIM-Fix and AIM-RL, vehicle as AIM-Trust and AIM-RL achieve similar collision rate in this case; and the throughput results show AIM-Trust achieves higher throughput. In other words, AIM-Trust with both lower collision rate and higher throughput indicates that AIM-Trust is much better than TIM-Fix and TIM-RL; and the trust factor we defined in this work is the major contributor of this out-performance.

**Collision-free AIM-Trust.** AIM-Trust with the throughput consideration provides significant collision rate reduction compared to AIM-1. However, it cannot guarantee collision-free due to the safety and throughput dilemma. In AIM-Trust, we consider collision and throughput via balancing factor $\lambda$ in the reward function Eq. [6]. To demonstrate that AIM-Trust can deduce appropriate buffer sizes based on trust, we relax the throughput requirement and modify the reward function of AIM-Trust to be $r_t^i = 1$ if no collision and $r_t^i = 40$ otherwise. We also let RL agent choose buffer size in range 0 to 26 (we denote this new version as AIM-Trust 2.0). Through these minor modifications, AIM-Trust 2.0 focuses on collision avoidance and learns to achieve collision-free in training. On average, the resulting buffer sizes of AIM-Trust 2.0 in cases with 20% to 100% untrustworthy vehicles are [14.4, 14, 14.4, 20, 24]. With same reward function, AIM-RL 2.0 (i.e., AIM-Trust 2.0 without trust factor in state space) also learns to avoid collision completely, but with higher buffer sizes [14, 15, 16, 20, 26] that lead to lower throughput.

| UV     | 20% | 40% | 60% | 80% | 100% |
|--------|-----|-----|-----|-----|------|
| AIM-RL | 51.35% | 50.00% | 64.44% | 18.18% | 71.69% |
| AIM-1  | 64.00% | 82.65% | 79.37% | 83.84% | 89.28% |
| TIM-RL | 44.46% | 51.48% | 60.94% | 78.11% | 79.85% |
| TIM-Fix| 58.70% | 56.51% | 64.33% | 78.11% | 81.55% |
| TIM-Tru| 42.98% | 52.31% | 57.32% | 71.16% | 78.48% |
| TIM-Tru 2.0 | 25.14% | 36.50% | 54.20% | 70.61% | 76.86% |
| AIM-Fix | 16.94% | 17.25% | 30.13% | 3.61% | 9.77% |

**Trust Results.** Here, we present the trustworthiness quantification of vehicles in our experiments. Fig. 5 shows the trustworthiness results of 10 vehicles in one of our experiments. Each column corresponds to a vehicle, which may or may not be trustworthy. Each row represents a full episode of training with 10 time steps (i.e., vehicles passing
through 10 intersections). For example, the first row contains the trustworthiness evaluations of all 10 vehicles passing through 10 intersections, and 20% of them are untrustworthy. Red areas indicate that a vehicle does not follow the approved trajectory in simulation, and this causes trustworthiness (blue line) decrements. These results show that our trustworthiness evaluation accurately captures the undesired behavior of vehicles, and is significantly helpful when used in control algorithms.

VI. CONCLUSION

AIM is designed to provide a collision-free intersection management with high throughput. However, in the application environment where there exists untrustworthy even malicious vehicles that do not follow the instructions, conventional AIM leads to a large amount of collisions (up to 28%). This example reveals the need for trustworthiness measure in MAS we proposed in this work. We design a trust evaluation framework and propose to use evaluated trustworthiness in control algorithms. We demonstrate in a case study how to embed trustworthiness in intersection management by designing AIM-Trust. To evaluate the effectiveness of the trust factor, we explicitly compare our AIM-Trust with baselines, and the experimental results show that the trust factor reduces the collision rate in all cases. In addition, in trustworthiness results, we directly see that trust scores accurately reflect the behavior quality of vehicles. For future work, we would like to refine our trust framework to be more comprehensive and applicable to a broader range of MAS control algorithms.

VII. ACKNOWLEDGMENT

The authors gratefully acknowledge the support by the National Science Foundation under the Career Award CPS/CNS-1453860, the Career Award SHF-2048094, the NSF awards 1453860, the Career Award SHF-2048094, the NSF awards 1932620, SHF-1910088, CPS/CNS-2039087, CMMI 1936624. This research was supported by the Berkeley Wireless Research Center (BWRC). The authors gratefully acknowledge the support by the National Science Foundation under the Career Award CPS/CNS-1936624, the DARPA Young Faculty Award and DARPA Director's Special Research Initiation Award, and the DARPA Young Faculty Award, D3000006022, for the development of the AIM-Trust system. The use of the AIM-Trust system was approved by the institutional review boards at UC Berkeley. The authors acknowledge the contributions of students in the AIM-Trust team, including Michael Zeng, Tran Anh Luu, Ali Motlagh, and Ben Allen, for their significant contributions to the development of the AIM-Trust system.

REFERENCES

[1] M. S. Shirazi and B. T. Morris, “Looking at intersections: a survey of intersection monitoring, behavior and safety analysis of recent studies,” IEEE Transactions on Intelligent Transportation Systems, vol. 18, no. 1, pp. 24–26, 2016.
[2] K. Dresner and P. Stone, “Multiagent traffic management: A reservation-based intersection control mechanism,” in Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems-Volume 2, 2004, pp. 530–537.
[3] V. Milanes, S. E. Shladover, J. Spring, C. Nowakowski, H. Kawazoe, and M. Nakamura, “Cooperative adaptive cruise control in real traffic situations,” IEEE Transactions on Intelligent transportation systems, vol. 15, no. 1, pp. 296–305, 2013.
[4] J. Rios-Torres and A. A. Malikopoulos, “Automated and cooperative vehicle merging at highway on-ramps,” IEEE Transactions on Intelligent Transportation Systems, vol. 18, no. 4, pp. 780–792, 2016.
[5] I. A. Ntousakis, I. K. Nikolos, and M. Papageorgiou, “Optimal vehicle trajectory planning in the context of cooperative merging on highways,” Transportation research part C: emerging technologies, vol. 71, pp. 464–488, 2019.
[6] J. Alonso-Mora, P. Beardsley, and R. Siegwart, “Cooperative collision avoidance for nonholonomic robots,” IEEE Transactions on Robotics, vol. 34, no. 2, pp. 404–420, 2018.
[7] F. Bin, F. XiaoFeng, and X. Shao, “Research on cooperative collision avoidance problem of multiple uav based on reinforcement learning,” in 2017 10th International Conference on Intelligent Computation Technology and Automation (ICICTA). IEEE, 2017, pp. 103–109.
[8] C.-C. Sun, C.-C. Liu, and X. Xie, “Cyber-physical system security of a port: the grid: State-of-the-art,” Electronics, vol. 5, no. 3, p. 40, 2016.
[9] R. Algluiyev, Y. Imamverdiyev, and L. Sukhostat, “Cyber-physical systems and their security issues,” Computers in Industry, vol. 100, pp. 212–223, 2018.
[10] D. Kamran, C. F. Lopez, M. Lauer, and C. Stiller, “Risk-aware high-level decisions for automated driving at occluded intersections with reinforcement learning,” in 2020 IEEE Intelligent Vehicles Symposium (IV). IEEE, 2020, pp. 1205–1212.
[11] B. Hurl, R. Cohen, K. Czarnecki, and S. Wislander, “TruPerceptr: Trust modelling for autonomous vehicle cooperative perception from synthetic data,” in 2020 IEEE Intelligent Vehicles Symposium (IV). IEEE, pp. 341–347.
[12] H. Lu, R. Lu, Z. Zhang, and J. Shao, “Replace: A reliable trust-based platoon service recommendation scheme in vanet,” IEEE Transactions on Vehicular Technology, vol. 66, no. 2, pp. 1786–1797, 2016.
[13] F. Li, D. Mikulski, J. R. Wagner, and Y. Wang, “Trust-based control and scheduling for ugv platoon under cyber attacks,” SAE Technical Paper, Tech. Rep., 2016.
[14] M. Cheng, C. Yin, J. Zhang, S. Nazarian, J. Deshmukh, and P. Bogdan, “A general trust framework for multi-agent systems,” in Proceedings of the 20th International Conference on Autonomous Agents and Multiagent Systems, 2021, pp. 332–340.
[15] K. Dresner and P. Stone, “A multiagent approach to autonomous intersection management,” Journal of artificial intelligence research, vol. 31, pp. 591–656, 2008.
[16] M. Hausknecht, T.-C. Au, and P. Stone, “Autonomous intersection management: Multi-intersection optimization,” in 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, 2011, pp. 4581–4586.
[17] G. Sharon and P. Stone, “A protocol for mixed autonomous and human-operated vehicles at intersections,” in International Conference on Autonomous Agents and Multiagent Systems. Springer, 2017, pp. 151–167.
[18] T.-C. Au, S. Zhang, and P. Stone, “Autonomous intersection management for semi-autonomous vehicles,” in Routledge Handbook of Transportation. Routledge, 2015, pp. 116–132.
[19] O. Maler and D. Nickovic, “Monitoring temporal properties of continuous signals,” in FORMATS/ETAPS, 2004.
[20] A. Jussang, Subjective logic. Springer, 2016.
[21] M. Cheng, S. Nazarian, and P. Bogdan, “There is hope after all: Quantifying opinion and trustworthiness in neural networks,” Frontiers in Artificial Intelligence, vol. 3, p. 54, 2020.
[22] S. Gong, A. Zhou, and S. Peeta, “Cooperative adaptive cruise control for a platoon of connected and autonomous vehicles considering dynamic information flow topology,” Transportation Research Record, vol. 2673, no. 10, pp. 185–198, 2018.
[23] T.-C. Au, N. Shahidi, and P. Stone, “Enforcing liveliness in autonomous traffic management,” in Twenty-Fifth AAAI Conference on Artificial Intelligence, 2011.
[24] E. K. Butler, A. A. Chandra, P. R. Chowdhary, S. M. Glissmann-Hochstein, T. D. Griffin, D. Jadav, S. Lee, and H. R. Strong Jr, “Drone air traffic control and flight plan management,” Dec. 26 2017, uS Patent 9,852,642.
[25] R. C. Moore, “Semantical considerations on nonmonotonic logic,” Artificial intelligence, vol. 25, no. 1, pp. 75–94, 1985.
[26] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski et al., “Human-level control through deep reinforcement learning,” nature, vol. 518, no. 7540, pp. 529–533, 2015.
[27] A. K. Dixit, J. J. Sherrerd et al., Optimization in economic theory. Oxford University Press on Demand, 1990.
[28] “Aim4 1.0-snapshot api,” http://www.cs.utexas.edu/~aim/aim4sim/aim4-release-1.0.3/aim4-root/target/site/apidocs/index.html, accessed: 2020-07-26.
[29] “Video demonstrations for aim-trust.” [Online]. Available: https://drive.google.com/drive/folders/1xYcmgs9UKMoZ5yg8BZre_GSGHd1XXo?usp=sharing.
[30] X. Liang, X. Du, G. Wang, and Z. Han, “Deep reinforcement learning for traffic light control in vehicular networks,” arXiv preprint arXiv:1803.11115, 2018.