Blame-Free Motion Planning in Hybrid Traffic

Sanggu Park, Edward Andert, and Aviral Shrivastava
Arizona State University

Abstract—**Despite the potential of autonomous vehicles (AV) to improve traffic efficiency and safety, many studies have shown that traffic accidents in a hybrid traffic environment (where both AV and human-driven vehicles (HVs) are present) are inevitable because of the unpredictability of HVs. Given that eliminating accidents is impossible, an achievable goal is to design AVs in a way so that they will not be blamed for any accident in which they are involved in. In this paper, we propose BlaFT Rules – or Blame-Free hybrid Traffic motion planning Rules. An AV following BlaFT Rules is designed to be cooperative with HVs as well as other AVs, and will not be blamed for accidents in a structured road environment. We provide proofs that no accidents will happen if all AVs are using a BlaFT Rules conforming motion planner, and that an AV using BlaFT Rules will be blame-free even if it is involved in a collision in hybrid traffic. We implemented a motion planning algorithm that conforms to BlaFT Rules called BlaFT. We instantiated scores of BlaFT controlled AVs and HVs in an urban roadscape loop in the SUMO simulator and show that over time that as the percentage of BlaFT vehicles increases, the traffic becomes safer even with HVs involved. Adding BlaFT vehicles increases the efficiency of traffic as a whole by up to 34% over HVs alone.**

I. Introduction

With recent advances in machine learning, sensor accuracy, and edge-computing, Autonomous Vehicles (AVs) are getting closer to becoming a reality. Improvements in safety, traffic efficiency, and accessibility are being touted as benefits of the technology. However, before we reach the often-imagined world of all AVs, there is likely to be a long transition period where autonomous vehicles have to operate along-with human-driven vehicles (HVs). These HVs introduce complications to the environment that an AV operates in, namely non-determinism in their driving patterns and even the need to occasionally avoid accidents that can be entirely caused by the negligence of the human drivers. Ideally, we would like to build AVs that can avoid all accidents. However, several studies have shown that it is not possible for an AV to avoid all accidents in hybrid traffic where AVs and HVs co-exist due to the nature of these HVs [1]. Let us consider a simple example situation where an AV is traveling on a multi-lane street. There is a vehicle immediately ahead of our ego AV, and another vehicle immediately to the left and right of the ego AV, and finally, a vehicle following close behind the ego AV as well. The vehicles to the left of the AV begins encroaching into the ego AVs lane such that the ego vehicle must choose between hitting the vehicle to the right of it or waiting for this encroaching vehicle to strike it (see Fig. 1). It can be seen here that the unpredictable nature of HVs can cause a crash despite an AVs best attempts to avoid one. Because accidents are unavoidable in a hybrid environment, Shalev *et al.* argued that AVs should do their best to avoid the unavoidable accidents to be exempted from the blame [2]. This idea of a blame-free AV has been explored in various ways. As Gliess *et. al* argue that AVs cannot take blame in an accident as they have no concept of punishment, and there was an atmosphere of skepticism at the time that AV manufacturers should be responsible for the accident [3].

This work is partially supported by National Science Foundation under Grant Numbers CPS 1645578 and CCF 1723476 and the NSF/Intel joint research center for Computer Assisted Programming for Heterogeneous Architectures (CAPA).

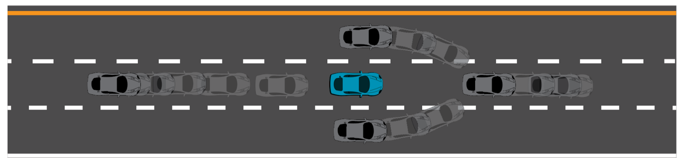


Fig. 1: Surrounding human driven vehicles (in grey) can crash into an AV (in blue) and the AV cannot avoid all accidents. The best an AV can do is not to be blamed for any accident it is involved in.

However, in March 2022, Mercedes-Benz broke the hesitation within AV industry and announced that they will accept legal responsibility for their autonomous driving system by volunteering the company to take the crash blame [4]. This move by Mercedes is a major break from previous incidents where the manufacturer pits the blame for its faulty autonomous system entirely on the driver [5]–[7]. Therefore building an AV software that will not be blamed for an accident will be very valuable to manufacturers taking on the liability for collisions of their AVs as will not have to suffer any legal issues as long as their code itself is blame free and another agent causes the crash. But first we must define what blame means.

To restate the US Law Second torts, § 454 on the matter of blame in a collision: *injurers are liable for accident damages if any of the two conditions are satisfied. First, of course, the injurer must have acted negligently – that is, he must have exercised less than "due care" (negligence), or secondly, the injurer's negligence must have caused the accident (causation)*. Based on this, [8] concludes that to avoid blame for an accident in which a vehicle is involved, it must prove that i) the vehicle did not cause the accident (causation), AND ii) it did its "due diligence" to avoid the accident (negligence). Geistfeld *et. al* argue that the US Law of Torts can and should be easily applied to AVs with minor modifications to federal law [9]. In this paper, we seek to further the idea of blame-free AVs by codifying the US Law of Torts blame definition into a general set of rules for motion planning algorithms. The intention is for any motion planning algorithm to be able to follow these rules, and as a proof of concept we create a motion planner that conforms to our set of rules. Thus, this paper makes the following contributions:

- We propose a novel set of motion planning rules: an autonomous vehicle that conforms to our '**B**lame-**F**ree hybrid **T**raffic Rules' (**BlaFT Rules**) fully observes the blame-free conditions in US Law of Torts.
- We propose a '**Bla**me-**F**ree hybrid **T**raffic motion planning algorithm' (**BlaFT**) that conforms to our **BlaFT Rules**. **BlaFT** is also a framework aimed to integrate existing sensing and collision avoidance maneuver (CAM) techniques which share I/O types.
- We provide safety proof that shows there is no accident in all-**BlaFT** traffic.
- We provide blame proof that shows an AV with our **BlaFT** is never be blamed for accident in hybrid traffic.

Furthermore, to demonstrate the benefits of **BlaFT Rules** on safety and efficiency aspects, we implemented hybrid traffic simulation in an urban roadscape loop in the Simulated Urban Mobility (SUMO) traffic simulator [10]. We implemented our **BlaFT** within the SUMO simulator. In the simulation, we intentionally designed HVs to drive in unsafe ways to cause accidents. Adjusting the ratio of **BlaFT** conforming AVs to HVs from 0% to 100%, we observe i) as **BlaFT**AV-to-HV ratio increases, the number of collisions linearly decreases, reaching no collisions when all the vehicles are **BlaFT** vehicles; and ii) adding **BlaFT** AVs actually increases the efficiency of the traffic by eliminating the chaotic behavior of HVs by up to 34% as compared to HV-only traffic.

II. Related Work

A. Safety Proofs for Autonomous Driving

As AVs are becoming more prevalent and real-world traffic transforms from HV-only traffic to hybrid traffic, issues of whether AVs and HVs can drive together in the same road has been raised [11], [12]. To make hybrid driving possible, Nyholm *et al.* and Roald *et al.* suggest that AVs need to predict the other vehicle's behaviors by estimating the externally measurable physical states, regardless of whether the other vehicle is an HV or AV [12]. On the other side, since HVs cannot distinguish if another vehicle is an AV or HV, an AV's driving should be indistinguishable from an HV to prevent disturbance to current traffic safety. And Matthias *et al.* shows in simulation that AVs drive in human-like manner and they reduced 90% of accidents with a relatively small response time [13]. However, despite the fact that traffic will be 'safer' with more AVs, Riedmaier *et al.* points out that public commercialization of AV is still a long way off, because whether AV is always safe in traffic has not been proved yet [14].

To address this challenge, Shalev *et al.* and Hilscher *et al.* suggest safety proofs that show in what condition AVs are free from the collision [1], [2]. First, Shalev *et al.* propose Responsibility-Sensitive-Safety (RSS) rules state that AVs should drive carefully by making sure AVs observe a conservative longitudinal and lateral safe distances to other vehicles and decide proper Right-Of-Way (ROW) among multiple vehicles [2]. And the authors prove by induction that the AV longitudinal headway distance rule they propose guarantees the safety in lane following case [2]. However, Shalev *et al.'s* approach and proofs ignored the limited lane width and ROW decisions in current traffic, and that can cause their driving to be blamed for accidents during lane changing and merging. Meanwhile, to prove lateral safety, Hilscher *et al.* provide formal expressions for fully autonomous driving in multi-lane traffic [1]. Distinguishing the driver's behaviors to be lane-following (LF) and lane-changing (LC), the authors prove that lateral safety between LF and LC is possible whenever LC maintains the reserved space in the target lane to be empty [1]. However, it still admits that lateral safety is only possible when both LF and LC participating in LC process are equipped with the same motion planner, and they explicitly understand the other's driving (or only for AV-only traffic). In other words, these disprove that both AVs and HVs are unsafe because of unidentifiable ROW decisions during the space-sharing conflict and the use of heterogeneous motion planning methods.

B. Blame Attribution for Collisions Involving Autonomous Vehicle(s)

Since an accident in hybrid traffic is always possible, the best AV (or AV manufacturer) can do for the unpredictable accident is to avoid blame [15]. To be blame-free, the current law requires the victim to prove that they did not cause an accident (causation) AND did their best to avoid or mitigate the crash damage (due diligence) so that no one else can make a counterargument [8]. In HV-only traffic, the driver's blame is subjectively determined by the investigations and reasoning in court because it is impossible to prove an individual human's decisions and driving behavior. On the other hand, when it comes to AV's blame for accidents, a wide social consensus has been studied and confirmed that AV manufacturers should be blamed for accidents when the autonomous agent is driving the vehicle [4], [16]–[18]. Unlike the mysterious nature of the typical human driver's thought process, AV's decision making should be transparent (traceable and reversible) so that it contributes to passengers' increased trust on AV system and promote clear blame discrimination [19]–[21]. Nevertheless, to the best of our knowledge, no AV blame criteria currently exists, so it is hard to decide if AV manufacturers should be blamed and bear the cost for an unpredictable accident.

C. Challenges in Existing Motion Planning Methods

Macroscopically, there are two approaches to AV motion planning: end-to-end machine learning and modular-based approach [22]. First, end-to-end machine learning methods have the benefit of having the flexibly to deal with any scenario [21], [23], however, it suffers from a 'black-box' problem so that AV's exact decision process is untraceable and cannot be estimated in a stochastic way [21]–[23]. Thus, a motion planner should adopt the modular-based approach to clarify blame. However, if the modular-based methods don't take care of every traffic scenario, the missing scenarios might incur an accident [14]. Plus, as Laurene *et al.* criticize, each modular-based method is developed independently so they focus on different kinds of scenarios (e.g., car following, lane changing, emergency, etc.), and there are gaps in use cases such as communication methods, I/O types, and driving appetite [24]–[27]. Therefore, these limitations make it hard to assess the safety of the existing modular-based approaches with a unified standard [23], [24].

On the other hand, to enable 'safer' autonomous driving in hybrid traffic, Markkula *et. al.* and Wang *et. al.* emphasizes successful and implicit interaction between AV and HV for space-sharing conflict [12], [26], [28]. To achieve this, recent studies have widely introduced game-theoretic methods to anticipate the other driver's vague intentions and make AVs adapt to them [27], [29]–[31]. But this approach cannot provide any safety or blame guarantee due to the inherently probabilistic nature of reward functions. Furthermore, collision avoidance and mitigation (CAM) algorithms with time-to-collision metrics have been introduced to assess risk and prevent the accident in time [32]–[35]. However, not just because hybrid traffic is not safe but also because safety concepts are irrelevantly defined from other works, the spatio-temporal collision decision points differ from each other or are even unclear. Thus, we want to conclude that existing methods fail to provide blame solution for autonomous driving. Rather, we a need blame-free motion planning method that is applicable in all scenarios.

III. **BlaFT** Rules

In this section, we define safety envelopes, 'safe', 'crash', and 'blame-free' states that are used to set up **BlaFT Rules**.

A. Safety Envelopes

To start with, we define two levels of braking for a vehicle. The first one is $a_{dec,max}$ – which is the 'maximum braking' of the vehicle, and it can be referred to by the manufacturer's vehicle specification. And the second one is $a_{dec,res}$ – which is the 'response braking' that

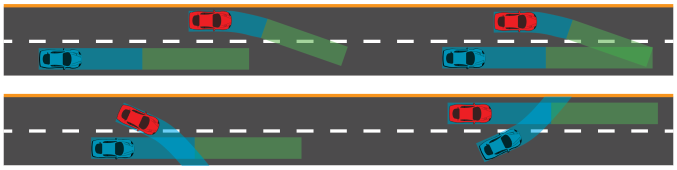


Fig. 2: Vehicles are in $safe$ state (upper case), if vehicles' REs (green shade) do not overlap, or the REs overlap but CEs (blue shade) do not. This is because both vehicles can avoid accident by proper braking. However, if the CEs overlap, vehicles are in a $crash$ state (bottom case) and the accident will be unavoidable even with the maximum braking.

the vehicle seeks to apply if it senses danger. To be more specific about the latter one, it is a target braking threshold that HVs in a hybrid environment expect other vehicles to apply typically. Next, we define the vehicle's stopping distances w.r.t. the $a_{dec,max}$ as d_c using Eq. (1) and $a_{dec,res}$ as d_r using Eq. (2). (Note that these equations consider 'sense-to-actuation time' (ρ).) We assume the worst case is that the vehicle accelerates at the maximum rate during ρ. Thus we can calculate the stopping distance for both terms as follows:

$$d_c = v \cdot \rho + \frac{1}{2} a_{acc,max} \cdot \rho^2 + \frac{(v + \rho \cdot a_{acc,max})^2}{2a_{dec,max}} \quad (1)$$

$$d_r = v \cdot \rho + \frac{1}{2} a_{acc,max} \cdot \rho^2 + \frac{(v + \rho \cdot a_{acc,max})^2}{2a_{dec,res}} \quad (2)$$

Crash and Response Envelope: Corresponding to these two braking and two stopping distances, we define two trajectory envelopes for a vehicle. The *Crash Envelope (CE)* – of length d_c, and the *Response Envelope (RE)* – of length d_r, and the width equal to the width of the vehicle w. Using the lane-based coordinate system from [2], we define CE and RE as:

$$CE = \{t(Y) + \alpha \cdot w(Y) \cdot t^{\perp}(Y) | Y \in [Y_{tail}, Y_{head} + d_c], \alpha \in [\pm 1/2]\} \quad (3)$$

$$RE = \{t(Y) + \alpha \cdot w(Y) \cdot t^{\perp}(Y) | Y \in [Y_{tail}, Y_{head} + d_r], \alpha \in [\pm 1/2]\} \quad (4)$$

where the Y-axis is the curve of the center line of the trajectory of the vehicle, starting from Y_{tail} to Y_{head}. $t(Y)$ is the trajectory of the vehicle in the direction of the center line of the lane, and w is the width of the vehicle. The parameters $\alpha \in [\pm 1/2]$ allow all the points around the trajectory line within the width of the vehicle to be included in the envelope.

One key idea in the driving algorithm of the AV is that at each time-step (of ρ time), if the AV senses another vehicle's estimated trajectory to overlap with the RE, then the AV will update its motion plan to avoid the overlap. If another vehicle's estimated trajectory overlaps with the CE of the AV, then the accident may be unavoidable. Fig. 2 shows depictions of the possible RE and CE overlap scenarios.

B. Safe and Crash States

From the perspective of an AV, now we can define its state w.r.t. another vehicle at the moment in time as a *crash state*, *safe state*, or *blame-free state*. First, an AV is in a *crash state* with another vehicle if the (Crash Envelope) CE of the other vehicle overlaps with the CE of the AV. As depicted in Fig. 2, this is because not just the colliding vehicles' trajectories are overlapped but also they are not able to stop before the collision point even with '$a_{dec,max}$.' (We discussed this with Eq. (1) and (3).) On the contrary, if the vehicles are not in a *crash state*, we define them as a *safe state*. This is because the AV can avoid the accident by merely slowing down with less than the maximum deceleration rate while following the route. Thus:

$$crash \equiv \forall c : c \neq ego \wedge \langle CE_c \cap CE_{ego} \rangle \quad (5)$$

$$safe \equiv \neg crash \quad (6)$$

C. Blame-free State

To determine whether an AV is to be blamed or not, we follow Kahan *et al.*'s two blame conditions, causation, and negligence [8]. As per Kahan *et al.*, a vehicle will be blamed for an accident if it caused the accident (causation) **OR** if it could have avoided the accident (negligence) [8]. Thus:

$$blame\text{-}free \equiv safe \vee \big(crash \wedge \neg causation \wedge \neg negligent\big) \quad (7)$$

Causation is determined by Right-of-Way (ROW). Thus, we define that a vehicle without ROW is the cause of an accident. ROW is determined from the structured road rules, such as a stop sign indicating that the ego vehicle would not have the ROW when merging. (Please refer to Fig. 3.)

$$causation_{ego} \equiv \neg ROW_{ego} \quad (8)$$

Next, we define a collision to be not negligent whenever the '$crash$' condition appears with no prior warning. In other words, if the collision happened and it was predictable for a driver, then the driver was 'negligent' for the accident. In terms of the envelopes we define, this means the **RE** area of the other vehicle did not pass through AV's **RE** in at least the previous time step. Therefore the AV never had a chance to react to the potential collision and avoid it.

$$\begin{aligned} negligent_{ego} \equiv \forall c : c \neq ego \\ \wedge \Big\langle \big(RE_{ego} \cap \neg CE_{ego}\big) \cap RE_c \Big\rangle \end{aligned} \quad (9)$$

By substituting Eq. (8) and (9) into Eq. (7), we derive an Eq. (10). It concludes that an AV is $blame\text{-}free$ whenever it is not engaged in the accident($safe$) or proves it had the ROW against the other car 'c' for the accident ($crash \wedge ROW_{ego}$).

$$\begin{aligned} blame\text{-}free \equiv safe \vee \Big(& crash \wedge ROW_{ego} \\ & \wedge \neg \big\langle \big(RE_{ego} \cap \neg CE_{ego}\big) \cap RE_c \big\rangle \Big) \\ \leftrightarrow safe \vee \big(& crash \wedge ROW_{ego}\big) \end{aligned} \quad (10)$$

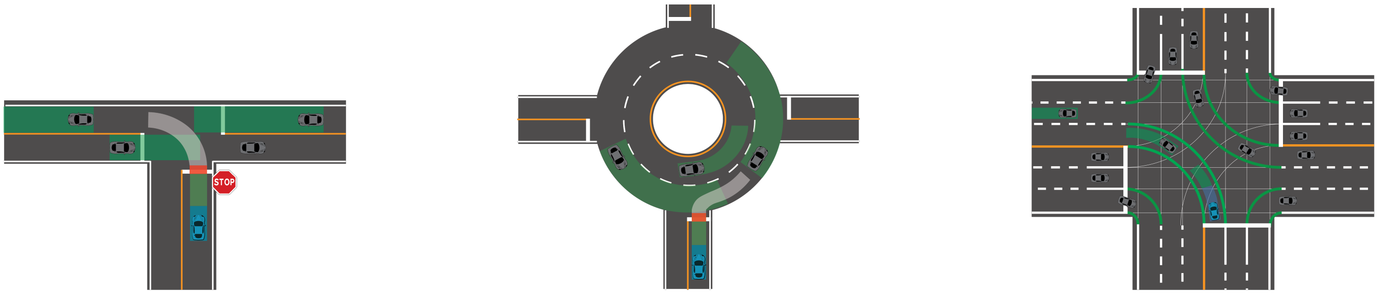
IV. BLAFT ALGORITHM

Fig. 4 outlines **BlaFT** – A '**Bla**me-**F**ree' motion planning algorithm which works in hybrid **T**raffic. **BlaFT** is a motion planning algorithm and runs at a high frequency and conforms to **BlaFT Rules**. And a relatively infrequent routing algorithm runs concurrently on the top of this motion planning algorithm and provides the context for the motion planning algorithm – specifically, it provides it with the set of waypoints (WP) to follow towards the destination (dst).

A. Behavior Decision

An AV may want to change the lane for various reasons, such as following its route to the destination, optimizing the time of travel/fuel efficiency, or even avoiding accidents. Thus, as the first step of the algorithm, **BlaFT** decides whether it plans to continue on the current lane (Lane Following or LF), or it plans to change its lane (Lane Changing or LC).

Before the decision, **BlaFT** creates a Behavior Line (BL) as a behavior criterion. As expressed in an Eq. (11), BL is the vertical line which is 'd_{blink}' distant alongside the planned trajectory ($t(Y)$)



(a) Merging before the stop sign (b) Merging into roundabout (c) Entering into intersection area

Fig. 3: **BlaFT Rules** in multi-road scenarios. Due to the clear ROWs among the roads, Fig. 3a and 3b show that **BlaFT** AV (blue car) not just maintains the safe distance but also yields the 'ROW' against the vehicles (grey cars) on the adjacent roads. In the intersection (Fig. 3c), traffic lights temporally separate the overlapping routes, and **BlaFT** does not need to predict the vehicles in other routes as it is expected the road rules are followed. In this manner, **BlaFT Rules** are observed by the ROWs among the roads and safe distances.

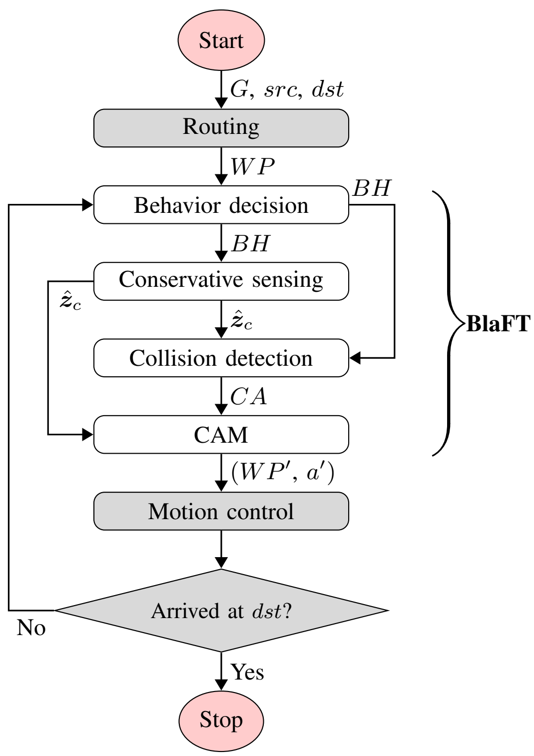


Fig. 4: Flowchart of **BlaFT** driving. **BlaFT** consists of four parts: behavior decision, conservative sensing, collision detection, and collision avoidance and mitigation (CAM).

and the wide as much as the ego vehicle's width (w_{ego}). 'd_{blink}' is the distance within which **BlaFT** has to turn on the blinker to warn other vehicles. And it is the minimum blinking time (t_{blink}) multiplied by the velocity limit (v_{limit}) so that it enables the AV to warn others safely. (We assume 't_{blink}' and 'v_{limit}' are preset by the traffic rules.)

$$BL = \{t(Y) + \alpha \cdot w_{ego} \cdot t^{\perp}(Y) | Y = Y_0 + d_{blink}, \alpha \in [\pm 1/2]\} \quad (11)$$

If there is at least one lane (L_i) that overlaps with ego AV's BL (BL_{ego}) but does not overlap with ego AV (ego) at all, then **BlaFT** considers the current behavior as LC. Otherwise, the behavior becomes LF. As a result, as shown in Fig. 5, ego AV's behavior is

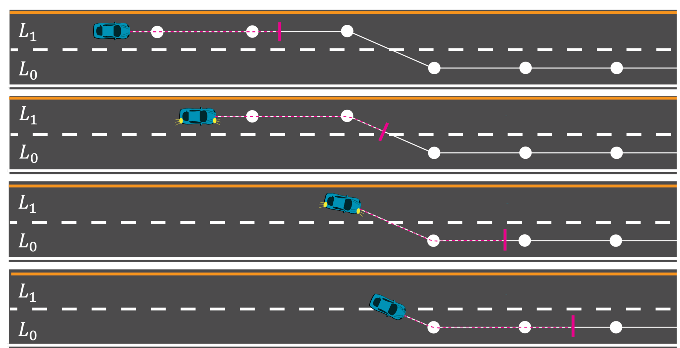


Fig. 5: Demonstration of behavior decision. Ego AV (blue car) draws BL (pink line), which is 'd_{blink}' distant alongside the planned trajectory (white line). If ego and BL overlap with the lane L_1 only, then AV is in LF mode (top). However, if BL starts to overlap with L_0 but ego does not, AV is in LC mode (middle). After entering the target lane, then ego overlaps with L_0 and recovers back to the LF mode (bottom).

switched from LF to LC as its BL starts to overlap with the target lane. And it is continued until the moment ego AV starts to overlap with the target one. After the moment, the behavior is recovered back from LC to LF.

$$BH = \begin{cases} LC, & \exists L_i \in L : \langle BL_{ego} \cap L_i \rangle \wedge \neg \langle ego \cap L_i \rangle \\ LF, & otherwise \end{cases} \quad (12)$$

B. Conservative Sensing

As a part of AV motion planning, sensing techniques (e.g., object detection) are developed in various ways but not in a unified way. Therefore, we propose the 'minimum requirements of sensing information' for AV to be blame-free, enabling the multiple existing techniques to be integrated into **BlaFT** framework.

This step uses the sensed data that the AV has collected to project the worst-case future trajectory of the other vehicles on the road, considering the error margin of the sensor(s) [36]. The sensing information consists of the location (x_c, y_c), the heading angle θ_c, and the velocity v_c of each of the vehicles it observes in it's sensing range. Furthermore, when admitting the road only allows 2-dimensional maneuver, we define that ego AV's safety-related area can be limited to its related lane (L_i) and its adjacent one(s) ($L_{i\pm1}$). Thus, we avoid unnecessary sensing demands which are not related to safety at all. To clearly specify the sensing target following AV's behavior, we define the vehicle as a sensing target in the second line in Eq. (13).

$$\begin{gathered} \boldsymbol{z}_c = \begin{pmatrix} x_c & y_c & \theta_c & v_c \end{pmatrix}^T, \\ \forall c, \forall L_i \in L : c \neq ego \wedge \langle (BL_{ego} \cup ego) \cap L_i \rangle \\ \wedge \langle c \cap (L_i \cup L_{i \pm 1}) \rangle \end{gathered} \tag{13}$$

To calculate other vehicles' near-future trajectories, **BlaFT** uses the Constant Turn Rate and Velocity (CTRV) model [37]. In the CTRV model, the differentials of turning angle (θ) and of velocity (v) are assumed to be constants 'ω' and '0', respectively, where 'ω' is the maximum steering angular velocity. Based on the constants, **BlaFT** predicts other vehicle's near-future information ($\boldsymbol{z}'_c$) as an Eq. (14).

$$\boldsymbol{z}'_c = \begin{pmatrix} x'_c \\ y'_c \\ \theta'_c \\ v'_c \end{pmatrix} = \begin{pmatrix} x_c + \frac{v}{\omega} \cdot \sin(\theta_c + \omega \Delta t) - \frac{v}{\omega} \cdot \sin \theta_c \\ y_c - \frac{v}{\omega} \cdot \cos(\theta_c + \omega \Delta t) + \frac{v}{\omega} \cdot \sin \theta_c \\ \theta_c + \omega \Delta t \\ v_c \end{pmatrix} \tag{14}$$

Lastly, **BlaFT** uses the worst-case sensing information, which is within the range of $\boldsymbol{z}_c$'s over the error margins ($\epsilon_{\boldsymbol{z}}$) of the values. The worst-case sensing information ($\hat{\boldsymbol{z}}_c$) is set to minimize the distance between the vehicle c's position and the ego vehicle's position at the time of '$t + \Delta t$' (Eq. (15), (16)). Then, **BlaFT** can draw the worst-case trajectory of all the other vehicles in the sensing range of the AV.

$$\hat{\boldsymbol{z}}_c = \begin{pmatrix} \hat{x}_c \\ \hat{y}_c \\ \hat{\theta}_c \\ \hat{v}_c \end{pmatrix} = \underset{\hat{\boldsymbol{z}}_c}{\operatorname{argmin}} \left([D'_{c,ego} | \hat{\boldsymbol{z}}_c \in [\boldsymbol{z}_c \pm \epsilon_{\boldsymbol{z}}] \right) \tag{15}$$

$$D'_{c,ego} = \sqrt{(\hat{x}'_c - x'_{ego})^2 + (\hat{y}'_c - y'_{ego})^2} \tag{16}$$

Algorithm 1 Collision Detection

Input: WP, BH, $\boldsymbol{z}_{ego}$, $\hat{\boldsymbol{z}}_c$
Output: CA

1: $CA = \emptyset$
2: $RE_{ego} = create\_RE(WP, \boldsymbol{z}_{ego})$
3: **for all** $c \in C$ **do**
4:   **if** $Y_c > Y_{ego}$ **then**
5:     $\hat{RE}_c = create\_RE(\hat{\boldsymbol{z}}_c)$
6:     $CA = CA \cup (RE_{ego} \cap \hat{RE}_c)$
7: **if** $BH = LC$ **then**
8:   $LE_{ego} = create\_LE(WP, \boldsymbol{z}_{ego}, t_{blink})$
9:   **for all** $c \in C$ **do**
10:     **if** $(L_c \neq L_{ego}) \wedge (Y_c < Y_{ego})$ **then**
11:       $\hat{RE}_c = create\_RE(\hat{\boldsymbol{z}}_c, W_L(Y_c), t_{enter})$
12:       **if** $LE_{ego} \cap \hat{RE}_c \neq \emptyset$ **then**
13:         $CA = CA \cup (RE_{ego} \cap LE_{ego})$
14:         **break**
15: **return** CA

C. Collision Detection

Once **BlaFT** derives the worst-case trajectories of the other vehicles, it determines the potential collision area. **BlaFT** draws the envelopes of itself and of the sensed vehicles and searches for the overlap between them. Then, the union of the overlapped areas is considered to be the potential collision area. And the way **BlaFT** draws the envelopes depends on whether it is in LF or LC mode. The process is described in algorithm 1 and Fig. 6, 7.

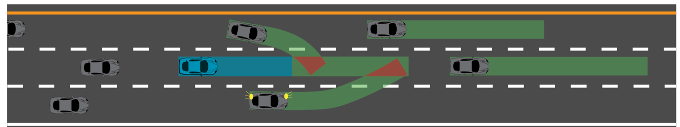


Fig. 6: Collision detection in lane-following mode. Ego AV (blue car) draws CE_{ego} (blue shade), RE_{ego} (green shade), and the worst case $\hat{RE}_c$ of other vehicle c (grey car). If ego AV detects the overlap(s) between them (red shades), the union of the overlaps becomes the collision area.

First, if AV is in LF mode, the default collision area (CA) is set to be an empty set and **BlaFT** draws 'RE_{ego}' (line 1-2). Then, **BlaFT** creates the worst-case response envelopes '$\hat{RE}_c$' for all other vehicles $c \in C$, which are longitudinally in front (line 3-6). Now, CA is the union of the overlaps between RE_{ego} and $\hat{RE}_c$ (line 7).

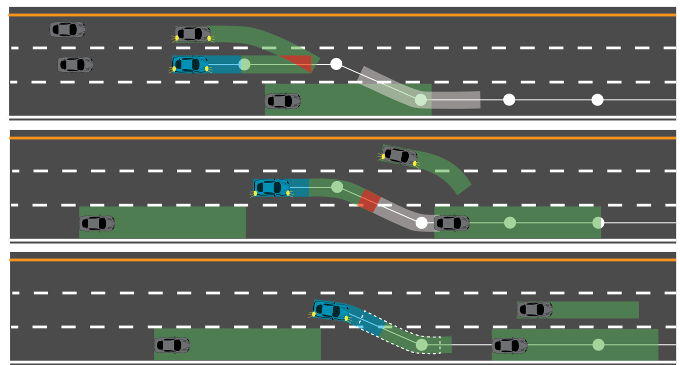


Fig. 7: Collision detection in lane-changing mode. While conducting the collision detection same as LF (top), ego AV additionally draws LE_{ego} on the target lane (white shade). If ego AV detects the overlap between LE_{ego} and $\hat{RE}_c$, $RE_{ego} \cap LE_{ego}$ is considered to be the collision (top, middle). Otherwise, if no overlap exists (bottom), AV neglects the area of LE_{ego} (dashed white box).

However, if **BlaFT** is in LC mode, **BlaFT** additionally draws a Lane-change Envelope (LE) (lines 7-8). LE is essentially the RE in the target lane that the ego vehicle will draw at the moment it starts to enter the new lane (Eq. (17)). And that is the reason why the range of Y is from Y^{lc} to $Y^{lc} + d_r$, where Y^{lc} is the Y-axis position of the vehicle when it changes lane.

$$LE_{ego} = \{t(Y) + \alpha \cdot w(Y) \cdot t^{\perp}(Y) | Y \in [Y^{lc}, Y^{lc} + d_r], \alpha \in [\pm 1/2]\} \tag{17}$$

The collision detection in LC mode is depicted in Fig. 7. To detect the potential collision in the target lane with the LE_{ego}, **BlaFT** draws the Response Envelopes ($\hat{RE}_c$) of all the other surrounding vehicles (except the rear vehicles in the same lane) assuming that their response time to be $\rho_c + t_{enter}$ (line 10-12). This essentially has the effect of elongating the $\hat{RE}_c$ to cover the area that the other vehicle could be in, even if it did not notice the ego vehicle while ego vehicle was entering, and the vehicle accelerated to it's maximum. Furthermore, in order to prevent the ego vehicle from even partially entering the target lane if it is not safe, **BlaFT** widens the $\hat{RE}_c$ of the other vehicles in the target lane to cover the whole width of the target lane ($W_L(Y)$), instead of the vehicle width ($W_c(Y)$) (line 12). Once the envelopes are drawn, **BlaFT** searches for the overlap between LE_{ego} and $\hat{RE}_c$. Once **BlaFT** finds out the overlap, then

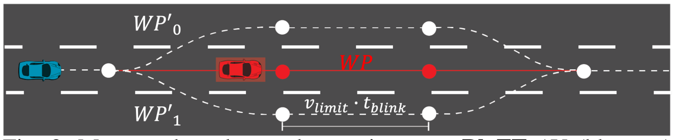


Fig. 8: Map graph update and rerouting step. **BlaFT** AV (blue car) updates the graph G to G' by removing the waypoints (red circles) close to the CA (red rectangle). Note that the interval of the removed waypoints is longer than the blinking distance '$v_{limit} \cdot t_{blink}$' to allow the AV to activate the blinker for enough time before recovering to the intended lane. In this example, the AV finds two sets of WP' when re-routing from its current position to dst.

BlaFT considers the overlap between RE_{ego} and LE_{ego} as the CA and stops the searching (lines 13-14). If there is no overlap, then LE_{ego} becomes an empty set and there is no CA in the target lane area.

Algorithm 2 Collision Avoidance and Mitigation

Input: CA, $\boldsymbol{z}_{ego}$, $\hat{\boldsymbol{z}}_c$, WP, G
Output: $response$

```
1:  response = (WP, −a_dec,max)
2:  if CA = ∅ then
3:      WP' = WP
4:  else if CA ≠ ∅ then
5:      G' = remove_colliding_nodes(G, CA, d_blink)
6:      WP' = rerouting(G', z_ego, dst)
7:  for all WP' do
8:      for all a' ∈ [−a_dec,max, a_acc,max] do
9:          z'_ego = update_velocity(z_ego, a')
10:         if (v'_ego ≥ v_limit) ∨ (a'_lat ≥ a_lat,max) then
11:             continue
12:         BH' = decide_behavior(WP', G', z'_ego, d_blink)
13:         CA' = detect_collision(WP', BH', z'_ego, ẑ_c)
14:         if CA' = ∅ then
15:             response = (WP', a')
16:             return response
17: return response
```

D. Collision Avoidance and Mitigation

When **BlaFT** detects a potential collision area, it tries to come up with a pair of alternative waypoints (WP') and acceleration rate (a') so as to avoid the collision. If the collision area is unavoidable, **BlaFT** applies maximum braking to minimize the impact of the collision. The overall process is outlined in algorithm 2.

To start with, the default response is set to be a pair of the current waypoints (WP) and maximum braking rate ($a_{dec,max}$) (line 1). If ego AV detects no collision area (line 2), it maintains the current trajectory (line 3). In contrast, if the collision area is detected, it updates the road graph (G) by removing the nodes (lines 4-5). The removed ones are close to the collision area, as shown in Fig. 8. To enable AV to take sufficient time to take over the collision area, the distance between the removed ones should be longer than '$d_{blink} = v_{limit} \cdot t_{blink}$.' Then, it finds out all possible future waypoints (WP') to reach the destination (dst) (line 6).

Subsequently, **BlaFT** applies all possible acceleration rates $a \in [-a_{dec,max}, a_{acc,max}]$ (searching from $-a_{dec,max}$ to pursue the best effort to mitigate the unpredicted damage) to find out the collision area (CA') by the response alongside each alternative waypoints (WP') (line 7-9). Using the future velocity ($v'_{ego} = v_{ego} + a \cdot \Delta t$), **BlaFT** checks if the future velocity and the centrifugal acceleration (a'_{lat}) exceeds the maximum constrained values (line 10). The centrifugal acceleration is calculated by using the maximum curvature of the trajectory (κ) and the future velocity (v') as an Eq. (18). If it violates the condition 18, then it goes to the next iteration (line 11).

$$a'_{lat} = \kappa \cdot v'^2 < a_{lat,max} \tag{18}$$

If the condition is met, **BlaFT** predicts its future behavior (BH') and future collision area (CA') (lines 12-13). And it uses the same methods proposed in previous sections. If no collision area is detected, then **BlaFT** returns a tuple of the set of waypoints and the acceleration rate as a response (lines 14-16). Otherwise, it continues the search.

In spite of searching, cannot return a safe response, then there is no option for collision avoidance. In this case, **BlaFT** returns the default response which consists of the current waypoints and the maximum deceleration rate '$a_{dec,max}$' to mitigate the damage (line 17).

V. Proofs

In this section, we first prove that in a situation where all vehicles are known to be running **BlaFT**, there will not be any collisions (safety proof). Then, we also prove that in a hybrid traffic scenario, a vehicle driving with **BlaFT** will not be blamed for any accident that it is involved in (blame proof). These proofs are done by following the collision scenarios that AV might face. As depicted in Fig. 1, we classify the scenarios into three: collision from the different lane(s), collision from a rear vehicle in the same lane, and collision with a vehicle in front in the same lane. Our proof is based on the two following assumptions: i) the trajectories of the other vehicles given by **BlaFT** are on the road maneuver following the continuous in 2-dimensional space, so that they do not jump abruptly from one position to another; ii) other vehicle's driving specifications, e.g., acceleration rate or response time, are safely estimated by **BlaFT**.

A. Safety Proof

Theorem 1. *If all vehicles on a structured road are driven by **BlaFT**, and they start from a $\boldsymbol{safe}$ state, then there will be no accident.*

Proof. We will prove this by mathematical induction at the time moments defined in multiples of ρ (the response time of *ego* vehicle). We will show that at any timestep k, the Collision Envelope of the *ego* vehicle CE_{ego} does not overlap with the CE_c of any other vehicle c. In fact:

$$\forall c : c \neq ego \wedge \neg\langle CE_{ego} \cap CE_c \rangle \tag{19}$$

<u>*Base Case* ($t = 0$)</u>: We assume that the base case is true by the assumption in the theorem – that we start from a $\boldsymbol{safe}$ state. This implies that the CE_{ego} does not overlap with the CE_c of other vehicles '*c*.'

<u>*Inductive Hypothesis* ($t = k$)</u>: At some time step $t = k$, we assume the Crash Envelope of the '*ego*' vehicle CE_{ego} does not overlap with the one of any other vehicle '*c*' CE_c at time step k, i.e.,

$$\forall k \geq 0, \forall c : c \neq ego \wedge \neg\langle CE^k_{ego} \cap CE^k_c \rangle \tag{20}$$

<u>*Inductive Step* ($t = k + 1$)</u>: Now, we prove that CE^{k+1}_{ego} does not overlap with CE^{k+1}_c of any other vehicle c. We divide the proof by the behavior modes – i.e., whether the ego vehicle is in the lane following (LF) mode, or in the lane changing (LC) mode.

To begin with, **BlaFT** in LC-mode does not allow the overlap between its LE and RE, whenever its LE overlaps with other vehicle's

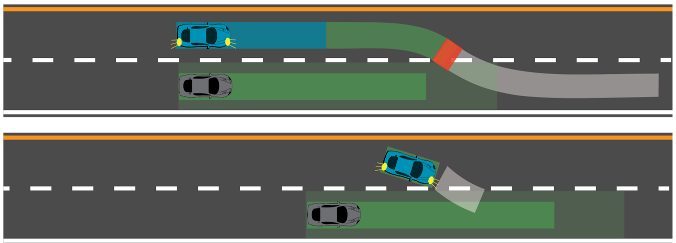


Fig. 9: **BlaFT** (blue car) in LC-mode. If LE (white shade) overlaps with the other(grey car)'s estimated RE (shallow green shade), it does not overcome the LE. Due to this, the **BlaFT** in LC-mode does not allow the REs (thick green shades) to overlap.

estimated RE (Fig. 9). And **BlaFT** does not enter the target lane until the overlap disappears. This prevents the overlap between the RE of ego vehicle and real RE of the other vehicle in the target lane (Eq. (21)).

$$\begin{aligned}&\forall c: c \neq ego \land \neg\langle RE_{ego}^{k+1} \cap LE_{ego}^{k+1}\rangle \\ &\rightarrow \forall c: c \neq ego \land \neg\langle RE_{ego}^{k+1} \cap RE_{c}^{k+1}\rangle\end{aligned} \tag{21}$$

Since the ego vehicle's RE and the other vehicle's RE never overlap with each other, their CEs also do not overlap. Thus, the lane-changing process satisfies Eq. (22).

$$\begin{aligned}&\forall c: c \neq ego \land \neg\langle CE_{ego}^{k+1} \cap CE_{c}^{k+1}\rangle \\ &\leftrightarrow safe_{k+1}\end{aligned} \tag{22}$$

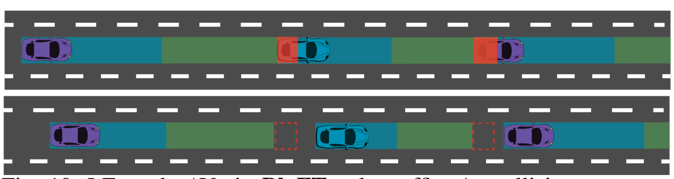


Fig. 10: LF-mode AVs in **BlaFT**-only traffic. As collision areas are detected within $RE \cap \neg CE$ (red shades, upper), AVs are able to avoid them by braking (dashed boxes, lower).

However, if **BlaFT** is in LF-mode in **BlaFT**-only traffic, all possible opponents can be in three categories: opponent from the adjacent lane(s), an opponent from behind, and from the front. First, if one or more opponents come from the adjacent lanes, following the previous discussion, they do not allow the RE overlap. Second, the last remaining scenarios are the opponents are in the same lane as depicted in Fig. 10. In this case, **BlaFT** is able to avoid the RE overlap with the front car by maintaining enough distance.

$$\begin{aligned}&\forall c: c \neq ego \land \big\langle \big(RE_{ego}^{k} \cap \neg CE_{ego}^{k}\big) \cap RE_{c}^{k}\big\rangle \\ &\leftrightarrow \forall c: c \neq ego \land \neg\langle RE_{ego}^{k+1} \cap \hat{RE}_{c}^{k+1}\rangle \\ &\rightarrow \forall c: c \neq ego \land \neg\langle CE_{ego}^{k+1} \cap CE_{c}^{k+1}\rangle \\ &\leftrightarrow safe_{k+1}\end{aligned} \tag{23}$$

As a consequence of the induction, **BlaFT** satisfies '$safe$' state at every time step in **BlaFT**-only traffic.

□

Lemma 1. *If a vehicle suddenly appears in '$RE_{ego} \cap \neg CE_{ego}$' of a **BlaFT**, then **BlaFT** will be able to avoid crashing with it.*

Proof. The maximum length of overlapped area between the RE of ego vehicle and RE of the other vehicle is the distance that AV maneuvers during the time of 'ρ', while accelerating with '$a_{acc,max}$.'

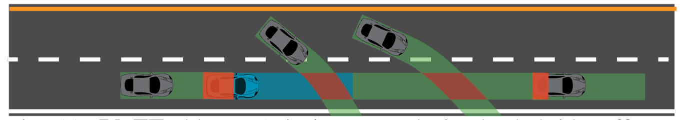


Fig. 11: **BlaFT** (blue car) is in LF-mode in the hybrid traffic. It is safe from the front car (the rightmost red rectangle) by lemma 1. When it comes to the other vehicles from adjacent lane and the behind (the remaining red ones), **BlaFT** has the right-of-way. For this reason, if the accident came from the adjacent lane(s) or behind, **BlaFT** remains $blame\text{-}free$.

And it is less than the crash distance 'd_c' (see Eq. (24)). Thus, if the collision area was outside the CE at '$t = k$', then ego vehicle is possible to avoid the accident by braking between 0 to maximum at '$t = k + 1$'.

$$0 \leq v \cdot \rho + \frac{1}{2} a_{acc,max} \cdot \rho^2 < d_c \tag{24}$$

□

B. Blame Proof

Theorem 2. *If **BlaFT** starts the driving from a $\boldsymbol{blame\text{-}free}$ state, then **BlaFT** will always maintain $\boldsymbol{blame\text{-}free}$ state.*

Proof. We will show that at any time k, ego vehicle is in $safe$ state or $crash \land ROW_{ego}$ state. In fact:

$$\forall c: c \neq ego \land safe \lor \big(crash \land ROW_{ego}\big) \tag{25}$$

<u>*Base Case* $(t = 0)$:</u> We assume that the base case is true by the assumption in the theorem – that we start from a $\boldsymbol{blame\text{-}free}$ state. This implies that the ego AV is in the state either $safe$ or $crash \land ROW_{ego}$.
<u>*Inductive Hypothesis* $(t = k)$:</u> At some time step $t = k$, we assume CE_{ego} does not overlap with CE_c or CE_{ego} overlaps with CE_c while ego possesses the right-of-way 'ROW_{ego}.' Thus:

$$\begin{aligned}&\forall k \geq 0, \forall c: c \neq ego \land \neg\langle CE_{ego}^{k} \land CE_{c}^{k}\rangle \\ &\lor \Big(\langle CE_{ego}^{k} \land CE_{c}^{k}\rangle \land ROW_{ego}^{k}\Big)\end{aligned} \tag{26}$$

<u>*Inductive Step* $(t = k + 1)$:</u> Now, we prove the Eq. (26) at the time step $t = k + 1$. We divide the proof by the modes of **BlaFT** – i.e., whether the ego vehicle is in the lane following (LF) mode, or in the lane changing (LC) mode.

First, same as the safety proof, **BlaFT** in LC-mode satisfies the state '$safe_{k+1}$.' When **BlaFT** is in LF-mode, there are three possible cases that its RE overlap: from the front or behind in the same lane, or from the adjacent lanes (Fig. 11). If **BlaFT**'s $RE_{ego} \cap \neg CE_{ego}$ overlaps with either the front one or the one from the adjacent lane(s), **BlaFT** is $safe_{k+1}$ by the Lemma 1. However, in the state $CE_{ego} \cap RE_c$ from behind or adjacent lane(s), **BlaFT** always has a right-of-way (ROW_{ego}). And all these facts are summarized in Eq. (27). Following Eq. (27), **BlaFT** in LF-mode is ***blame-free*** at every time step.

$$\begin{aligned}&\forall c: c \neq ego \land safe_k \lor \big(\langle CE_{ego}^{k} \cap RE_{c}^{k}\rangle \land ROW_{ego}\big) \\ &\rightarrow \forall c: c \neq ego \land safe_{k+1} \lor \big(crash_{k+1} \land ROW_{ego}\big) \\ &\leftrightarrow blame\text{-}free_{k+1}\end{aligned} \tag{27}$$

As a consequence of the induction, **BlaFT** satisfies '$blame\text{-}free$' state at every time step in hybrid traffic. □

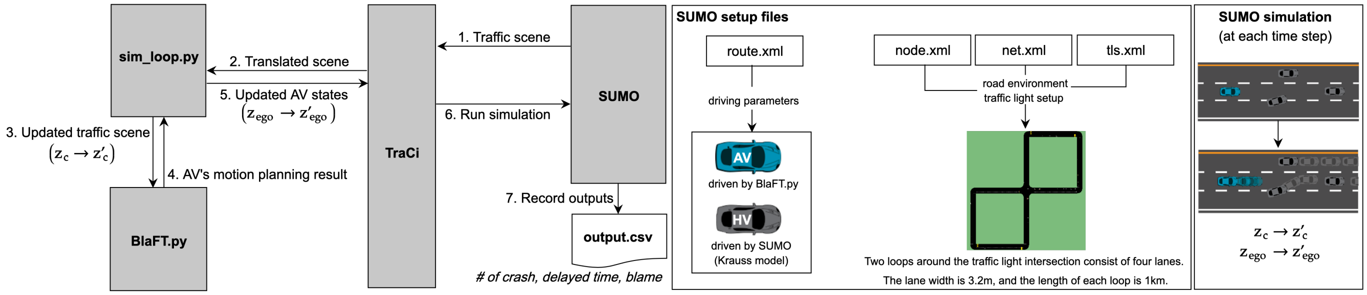


Fig. 12: **BlaFT** SUMO simulation set-up. Physical specifications of road environment and vehicles are designated by SUMO setup files. Then, by interfacing the python codes to SUMO simulator using 'TraCI', we enabled the AVs to collect the traffic scene information and react to the surrounding environment. At the same time vehicles react to the environment at the current time step and the whole traffic scene is updated and goes for the next time step, the results are accumulated on the output file.

TABLE I: Naturalistic driving parameters

Types	ρ $[s]$	$a_{dec,res}$ $[m/s^2]$	$a_{acc,max}$ $[m/s^2]$	$a_{dec,max}$ $[m/s^2]$	$l \times w$ $[m \times m]$
AV	0.1	2.0 ~7.0	1.8 (4.1)	7.0	5.0 × 1.8
HV (worst-case)	0.2 (0.5)	3.6 (4.1)			

VI. Experiments

In this section, we empirically evaluate how **BlaFT** affects traffic in terms of 'safety' and 'efficiency.' To do this, we have implemented **BlaFT** within SUMO simulator. The overall simulation set-ups and procedures are demonstrated in Fig. 12. First, we created a road environment conducive to infinite driving. Then, to simulate real-world traffic flow in the simulator, every vehicle in the simulation followed the naturalistic driving parameters (see table I). We enabled AVs in the simulation to conduct its motion planning by **BlaFT** algorithm, and AV's states are updated by simulation loop. The updated states are transferred to SUMO simulator by 'TraCI', which interfaces the python codes and the simulator. On the other hand, 'Krauss car-following model' is applied by SUMO simulator to control HVs to drive in a human-like way [38], [39]. At the last step of the procedures, the whole traffic scene is updated at every time step by applying new states of the vehicles.

We have modified the HV's velocity to follow a normal distribution that averaged the speed limit v_{limit} such that they occasionally make mistakes and cause a collision, e.g., $v_{HV} \sim \mathcal{N}(v_{limit}, 1)$. To make **BlaFT** safe from the misunderstanding of other's true specifications, it assumes the other vehicle's 'ρ', '$a_{dec,res}$', and '$a_{acc,max}$' to be the worst-case ones as the parameters in parentheses in table I. We also set that AV's $a_{dec,res}$ varies in the range $2.0 \sim 7.0m/s^2$ for each experiment since AV can freely choose the deceleration rate below the maximum deceleration rate as $a_{dec,res}$.

We then perform a simulation with a set ratio of HVs to AVs that are running **BlaFT** (e.g., 0:30, 5:25, etc.). Our simulation takes 30 vehicles, and we set the speed limit v_{limit} to either $12m/s$ or $25m/s$ to assume the typical city and highway driving for each. To check the average results, experiments of 30-minute duration are done 10 times for each data point. To check the safety and efficiency, we recorded two statistics using the existing analysis tools in SUMO, which are the number of collisions, and the delayed time of each vehicle. Furthermore, we automatically checked which vehicle is to be blamed whenever the collision happened, using an embedded collision tool in SUMO.

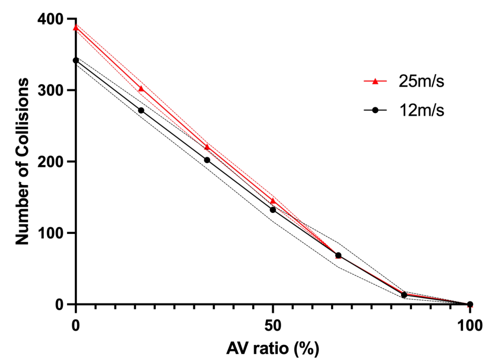


Fig. 13: The number of collisions by AV ratio.

A. Safety Evaluation

To evaluate the safety of **BlaFT**, we checked the number of collisions as AV-ratio differs. Whenever vehicles engaged in accident, they are teleported automatically to different random points on the track by SUMO so that they can continue driving from there. Therefore there at always 30 vehicles present in the simulation.

Fig. 13, shows the the number of collisions decreased linearly, as ratio of the number of **BlaFT** versus the number of HVs was increased. When the ratio became 100%, the number of collision reduced to zero. And this was common in both of the velocities tested ($12m/s$, $25m/s$). The linear decrease of the number of collisions came from the fact that **BlaFT** maintains the response distance and reacts properly to any sudden danger. In fact, the collisions go downs slightly more than linearly near the end of the graph. This is an extra benefit of the **BlaFT** vehicles driving less aggressively and therefore keeping the HV vehicles behind them slightly safer as well by having predictable acceleration and movement. The safety (e.g. collision count) does not seem significantly depend on the required deceleration rate or the length of RE of **BlaFT**. It is important to note that vehicles driven by **BlaFT** are never at fault for an accident according to any USA crash standards in this simulation especially the US Law of Torts, no matter the ratio.

B. Efficiency Evaluation

The efficiency is evaluated by the average delayed time as AV-ratio differs. The delayed time is the amount of time loss while braking to avoid the crash compared to the pure arrival time with target velocity. The result is depicted in a Fig. 14. First, as **BlaFT** ratio is increased, the average delayed time decreased with both of the maximum velocities. This is due to the safe driving of **BlaFT**, since

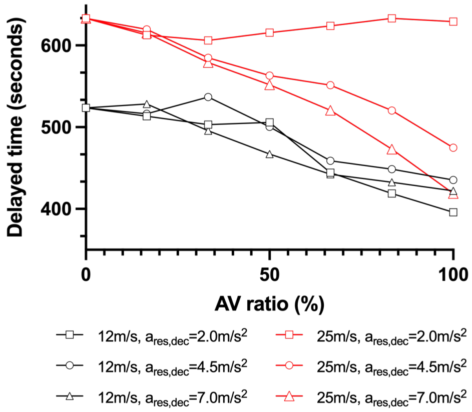


Fig. 14: Average delayed time by AV ratio.

it reduces the accidents which cause the multiple vehicles' delayed time to increase.

Furthermore, when it comes to the slower traffic where the speed limit is $12m/s$, it shows that there was no significant impact of the required braking rate or the length of RE on the average delay. However, in the faster traffic where its speed limit is $25m/s$, **BlaFT** resulted in the larger delayed times compared to the slower traffic. Also, when the response deceleration rate was at a minimum ($2.0m/s^2$), the delay did not change significantly compared to HV-only traffic. However, with other response deceleration rates ($4.5m/s^2$, $7.0m/s^2$), they showed 25% and 34% decrease in delayed time.

These results comes from the fact that the delayed time is affected by the 'length of RE.' First, in the slow traffic, **BlaFT** maintains the shorter RE compared to the ones in the fast traffic. However, as the traffic gets faster and the area of RE gets wider, **BlaFT** has more chances of RE overlap and this leads to the frequent braking. However, **BlaFT** can reduce the length of RE by increasing the response deceleration rate up to maximum deceleration rate. Then, **BlaFT** is able to maintain a shorter RE even in the faster traffic. In fact, the whole traffic has more chances of being efficient if **BlaFT** uses a more rapid braking to avoid the collision area and thus maintains a shorter RE.

VII. Conclusion

In this paper, we proposed '**Bla**me-**F**ree hybrid **T**raffic **Rules** (**BlaFT Rules**)' and our motion planning algorithm implementation of these rules for AVs cooperating with human drivers (**BlaFT**). Additionally, we have provided a set of equations so that the rules can be used to apply blame when there is a crash and sensor data available. We prove both mathematically and empirically via simulation that an AV using **BlaFT** (conforming to **BlaFT Rules**) will neither be the cause of an accident nor be blamed for an accident that does involve it. However, there is a limitation that **BlaFT** maintains 'over-conservative' safe distances because it does not consider the cooperation among the multiple drivers but instead is focused on being blame-free. Thus, we leave the question to future researchers how much blame that AVs have to endure while enhancing the cooperation among heterogeneous drivers to make overall traffic more efficient. Furthermore, in the future we would like to further explore parameters of **BlaFT** such as the Response Envelope deceleration rate and other heuristics for choosing a more optimal path that still conforms to the safety and blame constraints but will be less likely to cause traffic delays.

References

[1] Martin Hilscher, Sven Linker, Ernst-Rüdiger Olderog, and Anders P Ravn. An abstract model for proving safety of multi-lane traffic manoeuvres. In International Conference on Formal Engineering Methods, pages 404–419. Springer, 2011.

[2] Shai Shalev-Shwartz, Shaked Shammah, and Amnon Shashua. On a formal model of safe and scalable self-driving cars. arXiv preprint arXiv:1708.06374, 2017.

[3] Sabine Gless, Emily Silverman, and Thomas Weigend. If robots cause harm, who is to blame? self-driving cars and criminal liability. New Criminal Law Review, 19(3):412–436, 2016.

[4] Steven Loveday. Mercedes to accept liability when autonomous drive pilot is engaged. International journal of automotive technology, 2022.

[5] Puneet Kohli and Anjali Chadha. Enabling pedestrian safety using computer vision techniques: A case study of the 2018 uber inc. self-driving car crash. In Future of Information and Communication Conference, pages 261–279. Springer, 2019.

[6] Joanne M Bennett et al. Attribution of blame of crash causation across varying levels of vehicle automation. Safety Science, 132:104968, 2020.

[7] Elin Pöllänen et al. Who is to blame for crashes involving autonomous vehicles? exploring blame attribution across the road transport system. Ergonomics, 63(5):525–537, 2020.

[8] Marcel Kahan. Causation and incentives to take care under the negligence rule. The Journal of Legal Studies, 18(2):427–447, 1989.

[9] Mark A Geistfeld. A roadmap for autonomous vehicles: State tort liability, automobile insurance, and federal safety regulation. Calif. L. Rev., 105:1611, 2017.

[10] Pablo Alvarez Lopez, Michael Behrisch, Laura Bieker-Walz, Jakob Erdmann, Yun-Pang Flötteröd, Robert Hilbrich, Leonhard Lücken, Johannes Rummel, Peter Wagner, and Evamarie Wießner. Microscopic traffic simulation using sumo. In The 21st IEEE International Conference on Intelligent Transportation Systems. IEEE, 2018.

[11] Roald J Van Loon and Marieke H Martens. Automated driving and its effect on the safety ecosystem: How do compatibility issues affect the transition period? Procedia Manufacturing, 3:3280–3285, 2015.

[12] Sven Nyholm and Jilles Smids. Automated cars meet human drivers: responsible human-robot coordination and the ethics of mixed traffic. Ethics and Information Technology, 22(4):335–344, 2020.

[13] Matthias Althoff and Robert Lösch. Can automated road vehicles harmonize with traffic flow while guaranteeing a safe distance? In 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC), pages 485–491. IEEE, 2016.

[14] Stefan Riedmaier et al. Survey on scenario-based safety assessment of automated vehicles. IEEE access, 8:87456–87477, 2020.

[15] Ivó Coca-Vila. Self-driving cars in dilemmatic situations: An approach based on the theory of justification in criminal law. Criminal Law and Philosophy, 12(1):59–82, 2018.

[16] Alexander Hevelke and Julian Nida-Rümelin. Responsibility for crashes of autonomous vehicles: An ethical analysis. Science and engineering ethics, 21(3):619–630, 2015.

[17] Andreia Martinho et al. Ethical issues in focus by the autonomous vehicles industry. Transport reviews, 41(5):556–577, 2021.

[18] Ryan M McManus and Abraham M Rutchick. Autonomous vehicles and the attribution of moral responsibility. Social psychological and personality science, 10(3):345–352, 2019.

[19] Luiz Marcio Cysneiros et al. Software transparency as a key requirement for self-driving cars. In 2018 IEEE 26th international requirements engineering conference (RE), pages 382–387. IEEE, 2018.

[20] Tobias Schneider et al. Explain yourself! transparency for positive ux in autonomous driving. In Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems, pages 1–12, 2021.

[21] Amarildo Likmeta et al. Combining reinforcement learning with rule-based controllers for transparent and general decision-making in autonomous driving. Robotics and Autonomous Systems, 131:103568, 2020.

[22] Daniel Coelho and Miguel Oliveira. A review of end-to-end autonomous driving in urban environments. IEEE Access, 10:75296–75311, 2022.

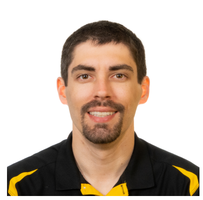
[23] Florin Leon and Marius Gavrilescu. A review of tracking and trajectory prediction methods for autonomous driving. Mathematics, 9(6):660, 2021.

[24] Laurene Claussmann et al. A review of motion planning for highway autonomous driving. IEEE Transactions on Intelligent Transportation Systems, 21(5):1826–1848, 2019.

[25] Can Zhao et al. A comparative study of state-of-the-art driving strategies for autonomous vehicles. Accident Analysis & Prevention, 150:105937, 2021.
[26] Wenshuo Wang et al. Social interactions for autonomous driving: A review and perspectives. Foundations and Trends® in Robotics, 10(3-4):198–376, 2022.
[27] Wilko Schwarting and Others. Social behavior for autonomous vehicles. Proceedings of the National Academy of Sciences, 116(50):24972–24978, 2019.
[28] Gustav Markkula et al. Defining interactions: A conceptual framework for understanding interactive behaviour in human and automated road traffic. Theoretical Issues in Ergonomics Science, 21(6):728–752, 2020.
[29] Hongtao Yu, H Eric Tseng, and Reza Langari. A human-like game theory-based controller for automatic lane changing. Transportation Research Part C: Emerging Technologies, 88:140–158, 2018.
[30] C Fox et al. When should the chicken cross the road? Game theory for autonomous vehicle-human interactions, 2018.
[31] Mingyu Wang et al. Game-theoretic planning for risk-aware interactive agents. In 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 6998–7005. IEEE, 2020.
[32] Andreas Tamke, Thao Dang, and Gabi Breuel. A flexible method for criticality assessment in driver assistance systems. In 2011 IEEE Intelligent Vehicles Symposium (IV), pages 697–702. IEEE, 2011.
[33] Janghee Park, Dongchan Kim, and Kunsoo Huh. Emergency collision avoidance by steering in critical situations. International journal of automotive technology, 22(1):173–184, 2021.
[34] Ishan Tyagi. Threat assessment for avoiding collsions with perpendicular vehicles at intersections. In 2021 IEEE International Conference on Electro Information Technology (EIT), pages 184–187. IEEE, 2021.
[35] Guofa Li et al. Risk assessment based collision avoidance decision-making for autonomous vehicles in multi-scenarios. Transportation research part C: emerging technologies, 122:102820, 2021.
[36] Edward Andert and Aviral Shrivastava. Accurate cooperative sensor fusion by parameterized covariance generation for sensing and localization pipelines in cavs. In 2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC), pages 3595–3602. IEEE, 2022.
[37] Robin Schubert, Eric Richter, and Gerd Wanielik. Comparison and evaluation of advanced motion models for vehicle tracking. In 2008 11th international conference on information fusion, pages 1–6. IEEE, 2008.
[38] Jie Song et al. Research on car-following model based on sumo. In The 7th IEEE/International Conference on Advanced Infocomm Technology, pages 47–55. IEEE, 2014.
[39] Jeffrey Wishart et al. Driving safety performance assessment metrics for ads-equipped vehicles. SAE Technical Paper, 2(2020-01-1206), 2020.

VIII. Biography Section

Sanggu Park obtained a Bachelor's degree in Applied Physics from Korea Military Academy in 2015 and a Master's degree in Computer Engineering from Arizona State University in 2022. His research was on safe motion planning for autonomous vehicles in a hybrid traffic environment where humans and autonomous vehicles must interact safely. He currently works as a signal officer in Republic of Korea Army.



Edward Andert is a Ph.D. student in Computer Engineering at Arizona State University. His research focuses on the use of cooperative sensor fusion of data from autonomous vehicles to detect and prevent errors from turning into crashes on the road. Prior to starting the Ph.D., Edward has a Master's degree in Computer Engineering from ASU. Edward previously worked for Intel and Mobileye as an autonomous vehicle sensor integration engineer.

Dr. Aviral Shrivastava is a professor in the School of Computing and Augmented Intelligence at Arizona State University, where he has established and heads the Make Programming Simple (MPS) Lab (https://labs.engineering.asu.edu/mps-lab/). His research lies in the broad area of "Software for Embedded and Cyber-Physical Systems." More specifically, Dr. Shrivastava is interested in making programming simpler for i) heterogeneous, many-core and accelerated computing, ii) low-power and error-resilient computing, and of iii) time-sensitive applications. Dr. Shrivastava received his Ph.D. and Masters in Information and Computer Science from the University of California, Irvine, and a bachelor's in Computer Science and Engineering from the Indian Institute of Technology, Delhi.