iPrism: Characterize and Mitigate Risk by Quantifying Change in Escape Routes

Shengkun Cui[†], Saurabh Jha[‡], Ziheng Chen[†], Zbigniew T. Kalbarczyk[†], and Ravishankar K. Iyer[†]
[†]University of Illinois at Urbana-Champaign, Urbana-Champaign, IL 61801, USA
[‡]IBM Research, Yorktown Heights, NY 10598, USA

***Abstract*—This paper addresses the challenge of ensuring the safety of autonomous vehicles (AVs, also called ego actors) in real-world scenarios where AVs are constantly interacting with other actors. To address this challenge, we introduce iPrism which incorporates a new risk metric – the Safety-Threat Indicator (STI). Inspired by how experienced human drivers proactively mitigate hazardous situations, STI quantifies actor-related risks by measuring the changes in escape routes available to the ego actor. To actively mitigate the risk quantified by STI and avert accidents, iPrism also incorporates a reinforcement learning (RL) algorithm (referred to as the Safety-hazard Mitigation Controller (SMC)) that learns and implements optimal risk mitigation policies. Our evaluation of the success of the SMC is based on over 4800 NHTSA-based safety-critical scenarios. The results show that (i) STI provides up to 4.9× longer lead-time-for-mitigating-accidents compared to widely-used safety and planner-centric metrics, (ii) SMC significantly reduces accidents by 37% to 98% compared to a baseline *Learning-by-Cheating (LBC)* agent, and (iii) in comparison with available state-of-the-art safety hazard mitigation agents, SMC prevents up to 72.7% of accidents that the selected agents are unable to avoid.**

All code, model weights, and evaluation scenarios and pipelines used in this paper are available at: https://zenodo.org/doi/10.5281/zenodo.10279653.

***Index Terms*—Autonomous Vehicles; Autonomous Driving Safety; Risk Assessment; Safety-hazard Mitigation**

I. Introduction

While AV safety has improved over the years [1], recent AV accidents involving Tesla [2], Waymo [3] and GM Cruise [4] suggest that state-of-the-art ADSes are far from being "as safe as human drivers" [5]. For example, GM Cruise has stopped operating on public roads after a severe accident involving a pedestrian [4]; Tesla, with advanced driver-assistance system (ADAS), was involved in 273 accidents, accounting for 70% of all ADAS accidents in 2022 [6]. Clearly, the safety of AVs in real-world environments continues to be a major societal challenge.

We address this challenge inspired by how experienced human drivers proactively mitigate hazardous situations by dynamically assessing risky actors while keeping track of available "escape routes" to prevent possible accidents [7]. Instilling this concept, this paper develops iPrism, a dynamic risk assessment and mitigation framework. IPrism's risk assessment is based on a new metric, *Safety-Threat Indicator* (STI), that quantifies the risk posed on the AV (ego actor) by other actors[1]. The risk is quantified in terms of the differential increase or decrease in the available escape routes for the ego actor due to other actors, considered singly or collectively. Actors with increased risk are potentially safety threatening to the ego actor and can lead to accidents if their risk is not mitigated. To minimize the increased risk and avoid accidents, iPrism employs a reinforcement learning (RL)-based *safety-hazard mitigation controller* (SMC) that executes the optimal mitigation actions (e.g., braking and acceleration) in a timely manner to proactively reduce STI.

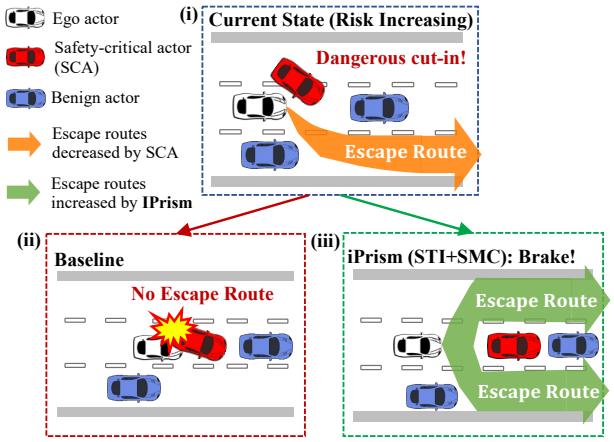


Figure 1: (i) shows the ego actor and other actors in a street scenario, with available escape route(s). In this scenario, a safety-critical actor dangerously cuts in front of the ego actor, eliminating the available escape routes (in orange) and increasing the risk. (ii) illustrates the failure of the ADS to address this safety threat, resulting in an accident and the elimination of escape routes. (iii) shows that the proposed method, iPrism, employing an RL-based mitigation controller, proactively brakes to increase escape routes (in green) and successfully avoids the accident. Braking is the most effective action in this case, as merging to the bottom lane eliminates the escape routes otherwise available on the top lane.

Evaluation Scenarios. A significant challenge in evaluation arises from the fact that most real-world datasets lack a substantial number of risky scenarios, limiting comprehensive assessments. To address this issue, we developed five multi-actor safety-critical typologies based on the National Highway Traffic Safety Administration's (NHTSA) pre-crash scenario typology report [8]. The five typologies, including front accident, dangerous cut-in, slowdown, and rear-end, are described in detail in §IV-B1. Collectively, these five scenario typologies account for approximately 80% of accidents in the United States. Utilizing these typologies, we created 4810 safety-critical scenarios, forming the core of our evaluation. These 4810 scenarios, now publicly available[2], can serve as a rigorous benchmark for future safety-focused research. An

[1]An actor is an on-road vehicle other than the AV.

[2]https://zenodo.org/doi/10.5281/zenodo.10279653

illustrative example of a safety-critical driving scenario where iPrism implements a mitigation action is presented in Fig. 1.

Results. We evaluate (i) the value of the STI metric in assessing risks, across the 4810 generated scenarios; (ii) the ability of the RL-based SMC to mitigate the posed risks in a timely manner to avoid accidents; (iii) iPrism's STI's efficacy in extracting safety-critical scenarios from a real-world dataset [9]. Our results show that iPrism significantly outperforms state-of-the-art methods.

(a) On the average, across all scenarios, STI achieves a 3.69s lead-time-for-mitigating-accident (LTFMA), while time-to-collision (TTC) [10]–[12], distance-to-closest-in-path-actor (Dist. CIPA) [13], and planning-KL-divergence (PKL) [14] achieve 0.83s, 1.38s, and 0.75s, respectively. Hence, STI achieves 4.4× improvement over TTC, 2.7× improvement over Dist. CIPA, and 4.9× improvement over PKL.

(b) On the average, iPrism's RL-based SMC controller mitigates 67.8% of the accidents that the often used Learning-by-Cheating (LBC) agent [15] fails to mitigate. In comparison, the TTC-based automatic collision avoidance (ACA) controller [11] mitigates only 30.7% of those accidents.

We evaluated (i) RIP [16][3] and (ii) RIP+iPrism using our NHTSA-based safety-critical scenarios. Based on the analysis of scenarios in real-world datasets (e.g., Argoverse [9]), we believe our NHTSA-guided pre-crash typologies are Out-of-Distribution (OOD) cases as they accentuate conditions that are not sufficiently well present in the real-world datasets [9], [17] commonly used in AV training.

What is OOD? Several studies [18]–[20] have shown that a machine-learning (ML) model degrades significantly when it encounters a significant shift from distribution and corresponding observations that are used in its training. This is because the ML model is unable or fails to incorporate the distribution shift when making its inference. In general, the distribution shift can result in an entirely new distribution. Practically, from a safety perspective, OOD distribution/data usually refer to input data that significantly differ from those used to train the model [19], [21]–[26] and can potentially lead to disastrous outcomes in safety-, mission-, and reliability-critical systems. More formally, [20] defines OOD as: given the training input distribution $P_{train}(\mathbf{x})$, the target input distribution $P_{target}(\mathbf{x})$ is OOD if $P_{train}(\mathbf{x}) \neq P_{target}(\mathbf{x})$. Due to the high complexity and dynamicity of the real world, a deployed ML model should handle OOD cases safely to avoid catastrophic failures.

Our mitigation controller helps RIP mitigate 72.7% of accidents which RIP could not handle, demonstrating iPrism's improvement and compatibility with a state-of-the-art safety-focused controller.

Finally, we perform a risk assessment on the real-world dataset, *Argoverse* [9], to help identify safety-critical scenarios that can be used for continuous testing and validation of AV safety. The actors' STI in the real-world driving dataset displays a long-tailed distribution, i.e., biased towards low-risk scenarios (over 90 percent of the values are $\leq$ 0.02), whereas actors in our NHTSA-based scenarios predominantly have high STI, i.e., STI values consistently exceed 0.25 for all cases, and in accident cases the values consistently reach 1.0. Our NHTSA-based scenarios are out of distribution for models trained solely on real-world datasets or those with limited safety-critical scenarios. This is evidenced by empirical results showing poor performance of LBC and RIP on our NHTSA-based scenarios. It is likely that training RIP on our NHTSA-based scenarios can improve their accident-prevention rate.

[3]We compare with RIP because RIP is designed to handle at least some out-of-distribution scenarios.

Putting iPrism in perspective. Existing research quantifies risk in terms of (a) time-to-collision (TTC), (b) distance to closest-in-path-actor (Dist. CIPA) [10]–[13], [27]–[33], or (c) the influence of actor(s) on the ego actor's trajectory distribution or planning decision [14], [34], [35], referred to as planner-centric metrics. Although these techniques identify risky actors and avoid accidents, they fail to prevent accidents in NHTSA safety-critical scenarios [8]. These metrics suffer from two problems: (a) they do not account for out-of-collision-path actors (which also impose risk towards the ego actors), and (b) they do not actively quantify and keep track of escape routes to mitigate accidents. Thus, as we observe from our empirical results, mitigation techniques using these metrics are unable to handle many safety-critical scenarios. By dynamically monitoring escape routes for the ego actor as an integral part of risk assessment, we are able to provide significantly better mitigation efficacy than any of the methods mentioned above.

II. Terminology

This section defines basic terminologies used in the paper.

State of an actor. An actor's *state* is defined by its position, velocity, acceleration, heading, and turning angle at a given point in time.

Ego actor. An *ego actor* refers to the vehicle driven by an autonomous driving system/agent.

Trajectory of an actor. A *trajectory* of an actor is defined as a time-ordered sequence of states representing the actor's dynamic evolution within an environment.

Safely navigable trajectory. A trajectory associated with the ego actor is said to be *safely navigable* if it does not intersect any of the other actors' trajectories.

Escape routes. *Escape routes* are the set of all safely navigable future trajectories.

Risk. In the context of autonomous driving and safety, *risk* refers to the degree of uncertainty and/or potential for accidents or harm inherent in the operational decisions of the ego agent. It encompasses the likelihood of unforeseen events and the possible consequences they may have for the safety of passengers, pedestrians, and other vehicles. In this paper, the only consequence we are concerned with is collision, i.e., we ignore other concerns such as violating rules of the road.

Safety hazard. A *safety hazard* occurs when the number of escape routes reduces to zero, indicating the absence of available mitigation strategies and an imminent safety violation.

Safety violation. A *safety violation* occurs when the AV collides with another actor.

III. Methodology

Our methodology draws inspiration from the techniques of skilled human drivers in handling risky situations. These

drivers proactively evaluate potential hazards, keeping a mental track of various 'escape routes' to evade potential accidents. The viability of these escape routes is influenced by factors such as unpredictable behavior of other road users, unfamiliar road layouts, and the availability of safe alternatives. A reduction in the number of these escape routes can serve as a measure of the safety risk and the likelihood of accidents. In the following section, we will detail our approach that employs the concept of escape routes to prevent accidents.

Overview. Here we describe our methodology for estimating and mitigating risk. (i) We use the concept and enumeration of escape routes to define a new metric for risk assessment, namely the *safety-threat indicator (STI)*, defined in §III-A below. (ii) We calculate the STI values for all actors individually and collectively to determine the risk envelope. The STI metric quantifies the combined risk for the ego actor, which can lead to accidents if the risk is not mitigated. (iii) To minimize risk with the intent of avoiding accidents, we propose a reinforcement learning (RL) based safety-hazard mitigation controller (SMC) to learn the optimal mitigation policy that provides mitigation actions to avoid an accident.

A. STI for Quantifying Risk Posed by the Actors

STI uses counterfactual reasoning to quantify the increase in the number of escape routes available to the ego actor by answering the following counterfactual query: *What would be the increase in the number of escape routes for the ego actor if a specific actor were not present?* (i) Using this counterfactual query, STI quantifies the impact of an actor and all actors collectively on the ego actor's safety. (ii) STI guides the RL-based safety mitigation controller (SMC) to apply mitigation actions (such as braking) to improve safety by increasing the ego actor's available escape routes. A formulation of STI based on reachability analysis is presented next.

STI formulation. We represent the number of escape routes available to the ego actor as the set of all safely navigable future trajectories from the current time t to time $t+k$ in the future. We first assume a world with N actors excluding the ego actor. Let $x_t^{(i)}$ be the state of the actor i at time t and $X_{t:t+k}^{(i)}$ be the actor's trajectory from time t to $t+k$; $\mathcal{X}_{t:t+k} = \{X_{t:t+k}^{(1)}, ..., X_{t:t+k}^{(N)}\}$ denotes the trajectory set of all actors except the ego actor in that period. Let $\mathbf{P}_{ideal}^{*}$ be an *oracle trajectory generator* that generates the set of all safely navigable future trajectories $\mathcal{T}_{t:t+k}$ from t to $t+k$ in the future given $\mathcal{X}_{t:t+k}$, the trajectories of all other actors; the ego actor $x_t^{(ego)}$ state at time t; and the drivable areas $\mathcal{M}$. $\mathbf{P}_{ideal}^{*}$ is mathematically specified as

$$\mathcal{T}_{t:t+k} = \mathbf{P}_{ideal}^{*}(\mathcal{M}, \mathcal{X}_{t:t+k}, x_t^{ego}) \tag{1}$$

The number of trajectories in $\mathcal{T}_{t:t+k}$, denoted $|\mathcal{T}_{t:t+k}|$, quantifies the ego actor's available safely navigable future trajectories (escape routes) given all other actors. The STI (the risk imposed by) of an actor on the ego actor is the amount by which the available escape routes is reduced because of that actor. The available escape routes in the absence of actor i is $\mathcal{T}_{t:t+k}^{/i}$, and is given by:

$$\mathcal{T}_{t:t+k}^{/i} = \mathbf{P}_{ideal}^{*}(\mathcal{M}, \mathcal{X}_{t:t+k}^{/i}, x_t^{ego}) \tag{2}$$

Similarly, the available escape routes in the absence of all actors is $\mathcal{T}_{t:t+k}^{\varnothing}$ and is given by:

$$\mathcal{T}_{t:t+k}^{\varnothing} = \mathbf{P}_{ideal}^{*}(\mathcal{M}, \varnothing, x_t^{ego}); \quad \text{where } \mathcal{X}_{t:t+k} = \varnothing \tag{3}$$

Based on (1–3), the STI for a specific actor i, $STI^{(i)}$ at time t, is shown in (4):

$$STI_t^{(i)} = \frac{|\mathcal{T}_{t:t+k}^{/i}| - |\mathcal{T}_{t:t+k}|}{|\mathcal{T}_{t:t+k}^{\varnothing}|} \tag{4}$$

Equation (4) formulates the counterfactual query that estimates the risk associated with actor i. $|\mathcal{T}_{t:t+k}^{\varnothing}|$ normalizes STI to a range $[0, 1]$, enabling comparison of STI across different driving scenarios. Thus, an STI value of 0 indicates that actor i has no impact on the ego actor's escape routes, and an STI of value 1 indicates that actor i fully eliminates the ego actor's escape routes. The STI for all actors combined, $STI^{(combined)}$, is obtained by a counterfactual that involves removing all actors, and is normalized to $[0, 1]$, as shown in (5).

$$STI_t^{(combined)} = \frac{|\mathcal{T}_{t:t+k}^{\varnothing}| - |\mathcal{T}_{t:t+k}|}{|\mathcal{T}_{t:t+k}^{\varnothing}|} \tag{5}$$

Combining Equations (1–4), we define function f_{STI_t} that calculates STI at time t as

$$STI_t^{(i)} = f_{STI_t}(\mathcal{M}, \mathcal{X}_{t:t+k}^{/i}, \mathcal{X}_{t:t+k}, x_t^{ego}) \tag{6}$$

and $STI_t^{(combined)}$ is defined similarly by combining equations (1–3) and (5).

STI implementation. We compute the available escape routes (i.e., the set of all safely navigable future trajectories) $\mathcal{T}_{t:t+k}$ as the reach-tube of the ego actor from the current time t up to time $t+k$ via reachability analysis [36]. Following the definition used in control and dynamic systems [36]–[41], "a *reach-set* is the set of states occupied by trajectories at exactly some specific time, and the *reach-tube* is the set of states traversed by those same trajectories over all times prior to and including the specified time (a time interval)" [40]. A more formal definition of reach-tube is given in Chapter 2.3.1 of [41].

To efficiently compute a reach-tube that provides a tight bound of all possible trajectories, which in theory can be infinite, a common practice in the dynamic system literature is sampling [37] of initial states or control inputs. Sampling allows one to construct the reach-tube (approximating the real one) from a finite number of trajectories resulting from these samples. Fan et al. [37] shows that with a sufficient number of samples, the real reach-tube can be accurately approximated with trajectories generated from those samples.

In summary, to compute the reach-tube for the ego actor from t to $t+k$ we first discretize the time span into time slices of size Δt. Then, for each time slice Δt, the ego actor state is propagated forward through use of a Bicycle Model [42][4]. The model's control values include acceleration (a) and turning angle (ϕ), sampled from the range of possible control values, until $t+k$ is reached. If an end-point is reached at $t+k$,

[4]Bicycle model is a widely used kinematic model that represents vehicle dynamics in autonomous driving motion planning and reachability analysis [42]–[44].

a trajectory (ego actor's state over time) without a collision or an impediment is a safely navigable future trajectory, i.e., an escape route. The trajectories generated by applying the maximum and minimum control values form the boundaries of the reach-tube, which is a collection of all possible escape routes. For example, the minimum (ϕ_{min}) and maximum (ϕ_{max}) turning angles constrain the leftmost and rightmost positions that the ego actor can reach in a time interval from t to $t+k$. To ensure the boundaries are included, we always include the maximum and minimum control values as part of the control samples at every Δt. Additional control values are sampled between the maximum and minimum control values to ensure that we can detect collisions and remove unreachable states (not part of the safe escape route) between the boundaries, as specified in Algorithm 1 below. Finally, to evaluate STI using Equations (4) and (5) in practice, $|\mathcal{T}_{t:t+k}|$ is obtained by calculating the volume of $\mathcal{T}_{t:t+k}$. The volume of the reach-tube $|\mathcal{T}_{t:t+k}|$ represents the reach-tube's state-space occupancy [45] of the ego actor's escape routes, and a reach-tube with a larger volume indicates that the ego actor can potentially reach a larger portion of the state-space (drivable area) safely from t to $t+k$, i.e., more escape routes are available to the ego actor.

Algorithm 1 describes the evaluation of escape routes $\mathcal{T}_{t:t+k}$ using reach-tube analysis. $x_t^{ego} = [x_t, y_t, \theta_t, v_t]$ is the vehicle state that consists of position (x_t, y_t), velocity v, and heading θ_t. The tuple $u = (a_t, \phi_t)$ is the control input to the Bicycle Model consisting of acceleration (a) and turning angle (ϕ). a_{min}, a_{max}, ϕ_{min}, and ϕ_{max} are the minimum and maximum control values. We use N as the sample size for sampling the control input at each time slice Δt.

In algorithm 1, the number of ego actor's trajectories can grow exponentially because of the sampling of N control values at each time slice Δt. Therefore, we apply the following optimizations to accelerate the computation of $\mathcal{T}_{t:t+k}$ for practical evaluation:[5]

1) A state $x_t'^{ego}$ generated in step 2 of algorithm 1 is ignored if the L2-norm to an existing state x_t^{ego} (already visited in the initCondDict of algorithm 1) is less than some threshold ϵ.
2) To compute the reach-tube, it's sufficient to calculate its boundary without determining all trajectories in the tube. Therefore, we enumerate all (a, ϕ) control value combinations from the sets $\{0, a_{max}\}$ and $\{\phi_{min}, 0, \phi_{max}\}$ to calculate the reach-tube boundary, rather than uniformly sample control values.

The parameter values of the bicycle model are set following [46].

B. RL-based Safety-hazard Mitigation Controller (SMC)

The role of the SMC, shown in Fig. 2, is to execute the optimal mitigation policy ψ^*, i.e., the optimal sequence of actions that proactively reduce STI. We define a cost function C in Equation (7) that consists of three terms: (1) the future risk (as STI), (2) path completion towards the destination $\mathbf{G}$, (3) a penalty term to deter aggressive SMC interventions. The cost function is minimized to determine the optimal mitigation policy ψ^*. To learn ψ^*, we apply reinforcement learning (RL) that effectively minimizes the cost function by converting C into the reward formulation as defined in Equation (8).

As the future state $x_{t+1}^{(ego)}$ of the ego actor depends on its current state $x_t^{(ego)}$ and the action a_t taken at time t, we define $h(\cdot)$ which outputs $x_{t+1}^{(ego)}$ given a_t and $x_t^{(ego)}$, according to the

[5]We conducted experiments before and after applying the optimization, and the overall results are marginally different.

Algorithm 1: Reach: Compute escape routes using reach-tubes.

Inputs : $\mathcal{M}$, $\mathcal{X}_{t:t+k}$, x_t^{ego}
Control Constants: $[a_{min}, a_{max}]$, $[\phi_{min}, \phi_{max}]$
Constants : Δt, k, N
Output : $\mathcal{T}_{t:t+k}$

/* initialization */
initCondDict $\leftarrow$ EmptyDict
initCondDict$[t] \leftarrow \{x_{t+\Delta t}^{ego}\}$

/* compute reach-tube */
for *time-slice* Δt *from* t *to* $t+k$ **do**
  for *sample* $x_t^{ego} \in initCondSet[t]$ **do**
    while *sample less than* N **do**
      1. Uniformly sample $u = (a, \phi)$ from $[a_{min}, a_{max}] \times [\phi_{min}, \phi_{max}]$
      2. Compute $x_{t+\Delta t}^{ego}$ by applying the Bicycle Model with (a, ϕ) and Δt
      3. **if** $x_{t+\Delta t}^{ego}$ *do not collide with* $X_{t:t+\Delta t} \in \mathcal{X}_{t:t+\Delta t}$ *and within the boundary of* $\mathcal{M}$ **then**
        add $x_{t+\Delta t}^{ego}$ to set initCondDict$[t+\Delta t]$
      end
    end
  end
  $t \leftarrow t + \Delta t$
end

/* generate $\mathcal{T}_{t:t+k}$ */
$\mathcal{T}_{t:t+k} \leftarrow$ BoundedReachTube(initCondDict)
return $\mathcal{T}_{t:t+k}$

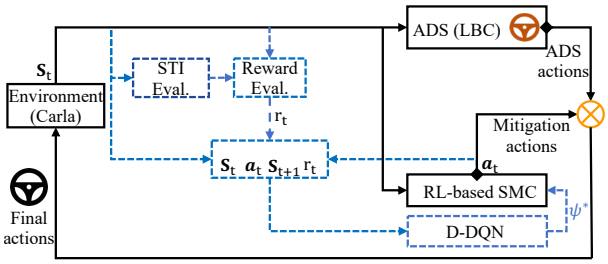


Figure 2: Overview of the proposed mitigation decision framework containing the RL-based SMC and the ego actor; blue dashed lines and boxes indicates RL training using D-DQN [47]; the black solid lines and boxes apply to both RL training and inference. $\otimes$ is a generic operator that augments the ADS actions with the mitigation actions.

vehicle dynamics and other constrains (e.g., drivable lanes).

$$\begin{aligned}&\arg\min_{a_t\sim\psi} C(\mathcal{M}, \mathbf{G}, x_t^{(ego)}, \hat{\mathcal{X}}_{t+1:t+k+1}, a_t), \forall t\\ &C = \alpha_0 f_{STI_{t+1}}(\mathcal{M}, \hat{\mathcal{X}}^{/i}_{t+1:t+k+1}, \hat{\mathcal{X}}_{t+1:t+k+1}, x^{ego}_{t+1}) + \\ &\qquad \alpha_1 d(\mathbf{G}, x_t^{(ego)}) + \alpha_2 \mathbb{1}\left[a_t \neq \text{No-Op}\right]\\ &\text{where } x_{t+1}^{(ego)} = h(x_t^{(ego)}, a_t)\end{aligned} \tag{7}$$

α_0, α_1, and α_2 are adjustable weight terms, and d is the remaining-cost to reach the destination. Since the ground-truth value of $\mathcal{X}$ for future time steps are unknown in online settings during SMC training and inference, we use the predicted value denote $\hat{\mathcal{X}}$ instead of $\mathcal{X}$. While it may be necessary to model the future trajectories of other actors as reach-tubes to calculate STI, in practice, such modeling is expensive in terms of both computational time and resources. Therefore, we made a simplified assumption that the trajectories of other actors can be estimated correctly in the near term, so predicting a single trajectory for each non-ego actor will suffice. Predicting $\hat{\mathcal{X}}$ is described in more details in §IV-C. The sections below describe the SMC's training using RL, the model architecture and implementation, and the inference process.

SMC Training (Learning ψ^*). RL is defined by states, actions, and a reward model, as described below.

(i) The state $\mathbf{S}_t$ of the RL at time t encompasses the state of the ego actor, the states of other actors, and the driving environment (e.g., map, lanes, static obstacles), as sensed by the ego actor's sensors (e.g., camera, radar, LiDAR, GPS, IMU).

(ii) The action a_t at time t is a mitigation action or a "no operation." Potential mitigation actions include braking (BR), acceleration (ACC), and lane changes to the left (LCL) and right (LCR) for risk mitigation and accident avoidance. In addition, the choice of "no operation" (No-Op) is included as an additional action when mitigation action is not needed. In this study, we demonstrate the value of STI as a risk assessment metric to guide the mitigation actions that use braking and acceleration.

(iii) The reward model r_t at time t is defined as

$$r_t = \alpha_0 \left(1 - STI^{(combined)}\right) + \alpha_1 r_{pc} + \alpha_2 p_{am}. \tag{8}$$

$\alpha_0, \alpha_1, \alpha_2$ are hyperparameters that control the trade-off between reward terms: $\left(1 - STI^{(combined)}\right)$, r_{pc}, and p_{am}. $STI^{(combined)}$, r_{pc}, and p_{am} are the STI values of all actors collectively (equation (5)), the reward for path-completion, and the penalty for activation of mitigation, i.e., $p_{am} = \mathbb{1}\left[a_t \neq \text{No-Op}\right]$, respectively at time t. The first term penalizes SMC actions that increase $STI^{(combined)}$ on the ego actor. The second term rewards SMC actions that drive towards the destination. The third term (negative) penalizes SMC's frequent activation of mitigation actions. Additional reward terms can be added to r_t as needed.

The training phase of the RL starts with random exploration, followed by a shift towards exploitation, where actions are chosen to maximize expected cumulative reward. At the end of the training phase, D-DQN yields the optimal mitigation policy ψ^*. We use the *learning-by-cheating agent* (LBC) proposed in [15] as the autonomous agent to plan, manage, and execute the actions of the ego actor in the CARLA [48] simulation environment. CARLA simulates the driving environment, including the map, other actors, and static objects. At any given time, the RL-agent observes the states of actors, the ego actor, and the driving environment, and executes an action that maximizes the expectation of the cumulative reward at that step. The D-DQN RL algorithm [47] is used to learn the expected cumulative reward for state-action pairs, as indicated in Fig. 2.

SMC Implementation and Architecture. The SMC is implemented with Deep Q-learning [49], in which a convolutional neural network (CNN) with parameter θ, denoted V_θ, is trained to approximate the $Q^{\psi^*}(\mathbf{S}_t, a_t)$ values (as a vector) for all actions given the state observation $\mathbf{S}_t$ (Equation (9)).

$$Q^{\psi^*}(a_t, \mathbf{S}_t; \theta) = V_\theta(\mathbf{S}_t)_{a_t} \tag{9}$$

We use the camera frames from the three front- and side-facing cameras provided by the CARLA Simulator [48] as $\mathbf{S}_t$. V_θ adapts the CNN architecture of the *Sensorimotor agent* from Chen et al. [15] as the backbone feature extractor to extract relevant information from the camera frames, i.e., $\mathbf{S}_t$. The output head of the *Sensorimotor* CNN architecture is modified to the same size as the number of actions to predict the Q values for all actions in one shot. As mentioned, the parameter of V_θ, θ, is learned using the D-DQN [47] training algorithm. At inference time, the action with the maximum Q value is chosen (Equation (10)).

SMC Inference and Deployment. During deployment, the SMC executes the learned mitigation policy to infer the mitigation action given the state observation, as shown in Fig. 2. The SMC's action a_t at each decision time step t is determined by the policy ψ^* that maximizes the expected cumulative reward. For example, given a state observation $\mathbf{S}_t$, i.e., the camera frames, at time t, "braking" is chosen if it maximizes the expected cumulative reward given $\mathbf{S}_t$ as shown in equation (10), in which Q^{ψ^*} is the expected cumulative reward under ψ^*. The mitigation action a_t at time t augments (in our implementation, overwrites) the ADS action and is then realized by the ego actor's actuator (e.g., throttle, brake, steering), except for the "No-Op" action, which involves no mitigation action.

$$a = \arg\max_{a\in\mathbb{A}} Q^{\psi^*}(a, \mathbf{S}_t), \quad a_t = \begin{cases} a, & \text{if } a \text{!= No-Op} \\ \text{No-Op}, & \text{otherwise} \end{cases} \tag{10}$$

IV. Experimental Setup

A. Autonomous Driving Agent

iPrism works in conjunction with an existing autonomous driving system/agent (ADS) by monitoring safety-threatening (risky) actors and providing mitigation actions to reduce risks when necessary, while the existing ADS makes normal driving decisions. We use a well-recognized ADS: the *Learning-by-cheating agent* proposed by Chen et al. [15] (*LBC agent*, hereafter) as the baseline agent for evaluating the STI's characteristics and iPrism's mitigation efficacy.

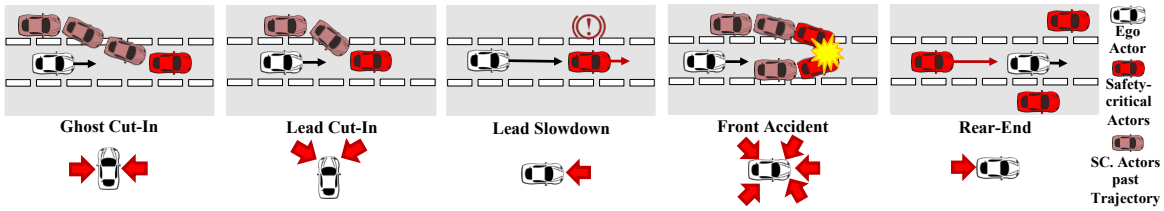


Figure 3: Overview of the scenario typologies. Red arrows indicate the directions from which the safety-threats are approaching. SC. Actors stands for safety-critical actors.

B. Driving Scenarios and Datasets

This section introduces safety critical scenarios derived based on NHTSA pre-crash scenario typology report and used in evaluation of iPrism.

1) Simulated safety-critical scenarios including OOD scenarios: The NHTSA mandates thorough evaluation of autonomous driving techniques in safety-critical scenarios [50]. However, as shown in §V-D, existing real-world datasets, such as [9], lack safety-critical scenarios because they are collected in a controlled environment with human drivers who obey traffic rules and avoid dangerous scenarios. To comprehensively evaluate our approach on safety-critical scenarios, we (i) selected the top-ranking safety-critical scenario typologies in terms of fatality rate from the NHTSA pre-crash scenario typology report [51], and (ii) used those typologies as high-level scenario descriptions and generated a set of safety-critical scenarios with the CARLA Simulator [48].

A safety-critical scenario typology provides a high-level description of a safety-critical scenario. We chose five multi-actor *safety-critical scenario typologies*. Together these five typologies account for ∼80% of the accidents in the United States as outlined in the NHTSA's pre-crash scenario typology report [8]. Each typology represents a safety-threat from a different direction relative to the ego actor. The typologies are described below, and Fig. 3 illustrates them and their respective threat directions.

(a) *Ghost cut-in:* An actor approaches from behind the ego actor in the adjacent lane and cuts into the ego actor's lane abruptly once it catches up with the ego actor. This scenario typology represents a safety-threat approaching from the side.

(b) *Lead cut-in:* An actor driving in front of the ego actor in the adjacent lane cuts into the ego actor's lane as the ego actor approaches it. This typology represents a safety-threat approaching from the front and the side.

(c) *Lead slowdown:* An actor driving in front of the ego actor in the same lane slowly stops in front of the ego actor. This typology represents a safety-threat approaching from the front.

(d) *Front accident:* Two actors driving in front of the ego actor in two different lanes collide because of a merging conflict. This typology represents a safety-threat approaching from all possible directions because of uncertainties about other actors' behavior during and after the accident. In practice, while this typology can lead to accidents involving the ego actor, in this study, the baseline agent [15] avoided all accidents in scenarios of this typology.

(e) *Rear-end:* Multiple actors are driving in front of and behind the ego actor in multiple lanes. An actor approaches the ego actor in the same lane and hits the ego actor from behind. This typology represents a safety-threat approaching from the back.

Table I: Number of safety-critical scenario instances and list of hyperparameters per scenario typology.

Scenario Typology	# of Scenario Instances	List of Hyperparameters	# of Accidents of Baseline Agent (LBC)
Ghost Cut-in	1000	distance_same_lane, distance_lane_change, speed_lane_change	519
Lead Cut-in	1000	event_trigger_distance, distance_lane_change, speed_lane_change	170
Lead Slowdown	1000	npc_vehicle_location, npc_vehicle_speed, event_trigger_distance	118
Front Accident	810	distance_lane_change, distance_same_lane, event_trigger_distance	0
Rear-end	1000	npc_vehicle_1_speed, npc_vehicle_2_speed, npc_vehicle_1_location	770

A *safety-critical scenario* instantiates a scenario typology by specifying hyperparameters. For example, in a *lead cut-in* scenario, based on the typology "an actor cuts in front of the ego actor," the hyperparameters are "cut-in angle," "cut-in speed," and "event-triggering distance to the ego actor." Safety criticality in a scenario varies with its hyperparameter values. For example, higher cut-in speeds reduce reaction time, which, in turn, increases criticality compared to that for lower speeds. We varied the hyperparameters uniformly for each typology to simulate 1000 *safety-critical scenarios* per scenario typology using CARLA [48] except for the front-accident typology, for which only 810 of the simulated scenarios were valid. The remaining 190 scenarios for the front-accident typology were discarded, as they did not contain an accident between two non-ego actors. Table I summarizes the number of safety-critical scenarios, the hyperparameters for each scenario type, and the total number of accidents encountered by the LBC agent. There were a total of 4810 *safety-critical scenarios* across the five scenario typologies. Our methodology focuses on creating safety-critical scenarios that are out-of-distribution compared to most driving datasets. This distinction arises because typical driving datasets are gathered under human supervision and are generally accident-free, unlike the scenarios we generate. For each typology, we select one scenario among all the scenarios generated from that typology for training the RL-based SMC; the SMC is then evaluated on all scenarios of this typology. The front-accident typology is excluded for SMC training and evaluation as none of the front-accident scenarios resulted in an accident,

as shown in §V-A. For the other four typologies, the scenario with the highest average STI before the accident is chosen per typology for SMC training.

2) Real-world pre-recorded scenarios: To demonstrate that STI can identify interesting, safety-critical scenarios in actual autonomous driving datasets, we applied STI to the Argoverse dataset [9]. Four notable safety-critical cases are presented in §V-D. In addition, we used the Argoverse dataset to assess bias towards less hazardous scenarios in the real-world datasets. Bias arises as real-world data are often gathered in controlled settings in which human drivers are adhering to traffic rules and avoiding dangerous situations.

C. Baselines for Evaluating STI's Efficacy

We compared STI with methods identifying important and potentially hazardous actors. Baselines included (i) *time to collision (TTC)* [11], [12], [33], (ii) *distance to closest in-path actor (Dist. CIPA)* [13], and *planner KL-divergence (PKL)* [14]. We chose TTC and Dist. CIPA because they are widely utilized in automated collision warning and automated collision avoidance (ACA) systems [11], [13], [27], [29]. We selected PKL because its objective of characterizing the importance of an actor closely aligns with the STI. These metrics are described below.

Time to collision (TTC) estimates the time it will take for the ego actor to collide with another actor that is in its path. TTC characterizes risk; a lower value of TTC signifies a higher chance of an accident due to less available time for mitigation. TTC is defined as the ratio of the distance d between the ego actor and an actor in-path[6] to the relative speed s_r between the ego actor and that actor, i.e., $TTC = \frac{d}{s_r}$.

Distance to closest in-path actor (Dist. CIPA) measures the distance from the ego actor to the closest actor that is in the ego actor's path, which acts as a proximity indicator. A lower Dist. CIPA value indicates a higher risk as the ego actor gets closer to an obstacle.

Planner KL-divergence (PKL) estimates the planning uncertainty by determining how differently the ego actor would plan if it saw only imperfect detection of actors versus perfect, ground-truth detection of actors in the scenario. A higher PKL value for an actor means that the actor has more influence on the ego actor's planning decisions. PKL indirectly characterizes the risk for an actor because missing a highly influential actor in trajectory planning may lead to accidents.

We use the ground-truth actor trajectories as $\mathcal{X}_{t:t+k}$ for STI evaluation and characterization in sections V-A, V-B and V-D and the predicted actor trajectories using the constant-velocity-and-turn-rate (CVTR) model as $\hat{\mathcal{X}}_{t:t+k}$ (the predicted value of $\mathcal{X}_{t:t+k}$) in SMC training and evaluation in §V-C.

D. Baselines for Evaluating RL-based SMC's Efficacy

We compared our SMC-enhanced agent, LBC+SMC w/ STI (i.e., LBC + iPrism), against the following:

1) LBC: The original *LBC agent* as the baseline for comparison. We used the weights provided by the authors of [15] "as is."
2) LBC+TTC-based ACA: The *LBC agent* with the time-to-collision (TTC) based automatic collision avoidance (ACA) controller. TTC-based ACA is a standard dedicated safety controller with which modern vehicles are equipped; it is used in [11], [13] for accident mitigation.
3) LBC+SMC w/o STI: The *LBC agent* with an SMC without STI in the reward formulation (refer to (8)). This serves as an ablation study of how STI contributes to a more effective mitigation policy.
4) RIP-WCM: We include the *robust imitative planning agent (RIP agent)* by Filos et al. [16] with the worst case model (WCM) configuration as an additional comparable method. RIP represents the state-of-the-art learning-based approach for improving AV safety with its ability to handle out-of-distribution (OOD) scenarios. Since the code and data used are open-source but weights of the models used by the authors are not publicly available, we trained the model as described in the paper using the code and training data provided by the author to obtain a functional RIP-WCM agent. Since the NHTSA-based scenarios used in our evaluation are not part of the training dataset for the RIP agent, therefore these scenarios are OOD by definition for the RIP agent [16].

Finally, to demonstrate the ADS-agnostic nature of SMC and its ability to improve the safety of various ADSes, we employed iPrism in conjunction with the RIP-WCM agent [16]. We call the combined system RIP+iPrism. Note that we do not compare LBC+iPrism with LBC + SMC w/ PKL [14] because the primary purpose of PKL is to improve the perception subsystem instead of mitigating the safety-threat.

E. Hardware & Software Platform

We used a platform with an AMD Ryzen Threadripper 3990X CPU and 128 GB of RAM for STI evaluation (sections V-A, V-B and V-D), and a platform with an AMD Ryzen 9 3950X CPU, 32 GB of RAM, and an NVIDIA RTX 3090 GPU for *simulated safety-critical scenarios* simulation (§IV-B1) and SMC evaluation (§V-C). All software implementations were done in Python 3 and tested on Ubuntu 20.04.

V. Evaluation and Results

This section introduces research questions and discusses results from experimental evaluation of iPrism,

Question 1: Can STI provide advanced detection capabilities to help mitigate accidents? Refer to §V-A.

Question 2: Does STI correctly characterize risk? How does it compare to other risk metrics? Refer to §V-B.

Question 3: Does reducing STI reduce the likelihood of accidents? How does a safety-hazard mitigation controller (SMC) perform with respect to other baselines? Refer to §V-C.

Question 4: Can STI be used to identify safety-critical scenarios from real-world datasets? Refer to §V-D.

A. Lead Time for Mitigating Accidents

We evaluate the effectiveness of various risk metrics in providing timely warnings for accident prevention, and propose a heuristic, *Lead-Time-for-Mitigating-Accident (LTFMA)*, against which all risk metrics are compared. (Refer to Table II.) LTFMA is determined by counting the consecutive time

[6] *In-path actors* are actors whose trajectories intersect with that of the ego actor.

Table II: Comparative analysis of Lead-Time-for-Mitigating-Accident (LTFMA) in seconds across various risk metrics. PKL-All: trained on all scenarios. PKL-Holdout: trained on all scenarios except the *ghost cut-in* and the *lead cut-in* scenarios.

Metric	Ghost Cut-In Mean (SD)	Lead Cut-In Mean (SD)	Lead Slowdown Mean (SD)	Rear-End Mean (SD)	All Scenarios Average
TTC	0.00 (0.00)	0.00 (0.00)	3.30 (0.89)	0.02 (0.17)	0.83
Dist. CIPA	0.00 (0.00)	0.00 (0.00)	**5.50 (0.89)**	0.02 (0.17)	1.38
PKL-All	0.75 (0.30)	1.01 (0.76)	1.22 (0.62)	0.01 (0.12)	0.75
PKL-Holdout	0.14 (0.21)	3.36 (4.18)	1.23 (0.69)	0.01 (0.12)	1.19
STI (ours)	**2.94 (0.33)**	**8.37 (0.70)**	2.22 (0.23)	**1.23 (0.11)**	**3.69**

We used *LBC agent* [15] as the ADS to control the ego actor to obtain these results.
In the front accident scenario, the ego actor's ADS (*LBC agent*) avoided the accident, resulting in no LTFMA metric to report.
SD stands for *standard deviation*.

steps with nonzero risk before an accident, thereby gauging the predictive capability of the metrics. It is defined as

$$LTFMA = \sum_{i=1}^{t_{\text{accident}}} \left(\mathbb{1}[\text{risk}(i) \neq 0)] \prod_{j=i+1}^{t_{\text{accident}}} \mathbb{1}[\text{risk}(j) \neq 0] \right).$$

To demonstrate the limitations of PKL in terms of training data requirements, we created two variants of PKL: PKL-All and PKL-Holdout. PKL-All was trained on all five scenario typologies, while PKL-Holdout was trained on all scenario typologies except the two cut-in typologies.

(a) On average, across all scenarios, iPrism's risk assessment component achieved a 3.69s LTFMA, while TTC, Dist. CIPA, and PKL achieved 0.83s, 1.38s, and 0.75s, respectively. In comparison, iPrism's risk assessment component achieved a 4.4× improvement over TTC, 2.7× improvement over Dist. CIPA, and 4.9× improvement over PKL.
(b) The LTFMA was at least 1.2 seconds (see the STI value for rear-end scenario in Table II) when STI was used, surpassing the 0.85–1.09-second reaction time range necessary for mitigating accidents, as identified in prior research [5]. Thus, LTFMA is a more effective than other risk metrics in monitoring hazardous driving situations.
(c) PKL is sensitive to training data, as evidenced by a significant decrease in LTFMA, from 0.75 seconds to 0.14 seconds, for the ghost cut-in scenario, and a significant increase in LTFMA, from 1.01 seconds to 3.36 seconds, for the lead cut-in scenario.

B. Characterizing Risk of Safety-critical Scenario Typologies

Fig. 4 presents the risk characterization for each scenario typology, in terms of STI $^{\text{(combined)}}$ (Fig. 4 a–e), PKL (Fig. 4 f–j) and TTC (Fig. 4 k–o). It includes line plots of the mean and standard deviation (shaded region) of these risk metrics over time t for each scenario typology. To understand the relationship between risk and safety, we plotted the metrics separately for scenarios that are safe vs. those that lead to accidents per typology. The results on Dist. CIPA are omitted due to space constraints and because the trends are similar to those for TTC.

Safety-threat Indicator (STI) (Fig. 4(a)–(e)). The STI $^{\text{(combined)}}$ generally increases as the driving scenario becomes hazardous (i.e., the number of escape routes decreases). Although STI $^{\text{(combined)}}$ is not strictly monotonic, it usually increases and peaks at the moment of an accident, thereby demonstrating its capability for detection and prediction of imminent accidents. The *ego actor* (driven by *LBC agent* [15]) manages to avoid accidents in the scenarios labeled as safe. In these safe scenarios, the risk increases when the situation becomes hazardous. However, the *ego actor* avoids accidents by taking mitigation actions on its own. The mitigation action taken by the *ego actor* results in reduced STI $^{\text{(combined)}}$ in these safe scenarios. Therefore, any agent that aims to prevent accidents must reduce STI $^{\text{(combined)}}$ as a strategy to mitigate risk.

Planning KL-divergence (PKL) (Fig. 4(f)–(j)). PKL is typically higher in accident scenarios than in safe ones. However, its values in accident scenarios are not consistent; they fluctuate significantly, sometimes decreasing even when an accident is imminent, as illustrated in Fig. 4(g) and (h). Meanwhile, PKL can also be higher than usual in some safe driving scenarios despite the absence of immediate danger (see Fig. 4(g)). Unlike STI $^{\text{(combined)}}$, PKL varies across different scenarios and does not reliably indicate the likelihood of an accident. For example, in Fig. 4(g) at time step 40, PKL is high, but there is no danger to the ego actor, and many scenarios with such high PKLs do not lead to an accident.

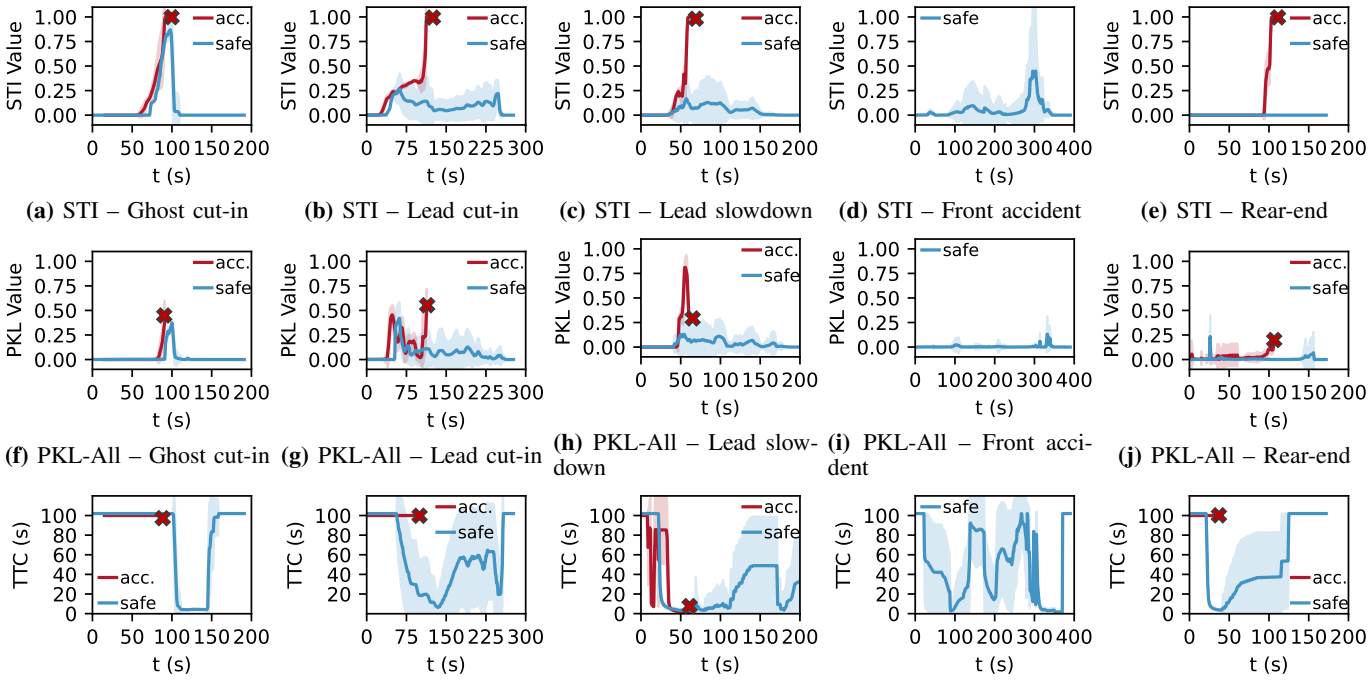
Time-to-collision (TTC) (Fig. 4(k)–(o)). TTC is expected to decrease as the probability of an accident increases. However, in all scenario typologies except the lead slowdown typology, TTC does not show a decrease. In the lead slowdown typology (see Fig. 4(m)), TTC decreases for both safe and accident scenarios, rendering them indistinguishable. This limitation of TTC exists because it considers only one in-path actor at any given time, while ignoring other actors in the scenario that might also contribute to the risk.

The key takeaways are:

(a) STI $^{\text{(combined)}}$ is statistically different for safe and accident scenarios.
(b) STI $^{\text{(combined)}}$ is mathematically well-defined and almost always monotonically increases before an accident, making STI $^{\text{(combined)}}$ an effective metric for monitoring and mitigating hazardous situations.
(c) An STI $^{\text{(combined)}}$ of zero indicates no risk of an accident, while a score of one indicates that an accident is likely, as there are no escape routes.
(d) Unlike other metrics, STI $^{\text{(combined)}}$ considers risk imposed by all the actors on the road.

C. Efficacy of STI-driven Safety-hazard Mitigation Controller

We evaluated the effectiveness of iPrism for accident mitigation. Table III offers a comparative analysis of agents' accident prevention rates across different scenario typologies.



(a) STI – Ghost cut-in **(b)** STI – Lead cut-in **(c)** STI – Lead slowdown **(d)** STI – Front accident **(e)** STI – Rear-end

(f) PKL-All – Ghost cut-in **(g)** PKL-All – Lead cut-in **(h)** PKL-All – Lead slowdown **(i)** PKL-All – Front accident **(j)** PKL-All – Rear-end

(k) TTC – Ghost cut-in **(l)** TTC – Lead cut-in **(m)** TTC – Lead slowdown **(n)** TTC – Front accident **(o)** TTC – Rear-end

Figure 4: Characterizing risk metrics — $\text{STI}^{(\text{combined})}$, PKL, and TTC — across five scenario typologies. STI and PKL: higher is riskier. TTC: lower is riskier. Red crosses indicate when scenarios end because of accidents. We used *LBC agent* [15] as the ADS to control the ego actor to obtain these results. Please note that STI on the y-label axis refers to $\text{STI}^{(\text{combined})}$.

Table III: Comparative analysis of agents' accident prevention rates across scenarios.

Agent	Reasons for Comparison	Ghost cut-in				Lead cut-in				Lead slowdown			
		CA↑ (%)	TCR↓ (%)	CA↑ (#)	TAS (#)	CA↑ (%)	TCR↓ (%)	CA↑ (#)	TAS (#)	CA↑ (%)	TCR↓ (%)	CA↑ (#)	TAS (#)
LBC+SMC w/ STI (LBC+iPrism)	To show improvement over baseline agent.	49%	26.7%	252	519	98%	0.3%	167	170	87%	1.5%	103	118
LBC+SMC w/o STI	To show that STI is important (ablation study).	1%	51.6%	3	519	2%	16.7%	3	170	86%	1.6%	102	118
LBC+TTC-based (ACA)	To show improvement w.r.t. ACA techniques.	0%	51.9%	0	519	0%	17.0%	0	170	92%	1.0%	108	118
RIP+SMC w/ STI (RIP+iPrism)	To show generalization with other agents.	86%	6.5%	413	478	61%	26.5%	406	671	71%	12.9%	311	440

CA# stands for collision avoided (higher is better ↑); **CA**% stands for the percentage of accident scenarios (shown in TAS) prevented by the mitigation strategy. $CA(\%) = (CA(\#)/TAS(\#)) \times 100$; **TCR** stands for total collision rate (lower is better ↓); **ACA** stands for automatic collision avoidance.
1000 scenario instances were executed for each scenario and for each baseline agent.
TAS is the number of total accident scenarios, i.e., the total number of driving scenario instances that led to accidents. Thus, it captures the number of accidents experienced by the LBC and RIP agents. For example, LBC and RIP had collisions in 519 (out of 1000) and 478 (out of 1000) of the driving scenarios for the ghost cut-in typology, respectively.
TCR is the total accident rate for that controller. $TCR(\%) = ((TAS(\#) - CA(\#))/1000) \times 100$.

Table IV: Comparative analysis of initial mitigation activation timing between LBC+TTC-based ACA and LBC+SMC w/ STI (LBC+iPrism). The first two rows show the average time (in seconds) into the scenario when the mitigation actions are performed by each agent, per typology; lower is better.

Agent	Ghost cut-in Avg. time (s)	Lead cut-in Avg. time (s)	Lead slowdown Avg. time (s)
LBC+SMC w/ STI (LBC+iPrism)	9.63	5.01	3.86
LBC+TTC-based ACA	10.20	8.74	5.18
Lead Time in Mitigation (s)	**0.57**	**3.73**	**1.32**

ACA stands for automatic collision avoidance.
Lead time to mitigation is calculated as the time differences between LBC+iPrism and LBC+TTC-based ACA when the first mitigation action is performed.

We excluded results for the front-accident typology since they included no accidents that involved the ego actor.

Outperforms the baseline agent (LBC). The driving scenarios presented in this paper were created to test for safety, so the ego actor driven by the baseline agent (in this case LBC) was expected to collide frequently with other actors. LBC collided in 519 (out of 1000), 170 (out of 1000), and 118 (out of 1000) scenarios across ghost cut-in, lead cut-in, and lead slowdown driving scenarios (as indicated in the TAS column of Table III). As shown in Table III, iPrism was able to prevent 49%–98% of accidents, depending on the scenario typology. It did so by actively trying to reduce the risk. For example, Fig. 5 shows that the $\text{STI}^{(\text{combined})}$ values for the ghost cut-in scenario typology for LBC + iPrism (labeled "iPrism" in the figure) are lower than those for LBC.

Outperforms other safety agents (RIP and LBC+TTC-based ACA). We explicitly compared iPrism against RIP and LBC+TTC-based ACA (a controller that uses TTC for

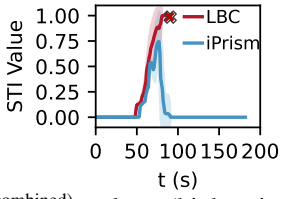


Figure 5: STI $^{(\text{combined})}$ values (higher is riskier) on ghost cut-in scenario typology. Red line: Original LBC agent. Blue line: iPrism-enabled LBC agent. Red cross indicates when the scenarios end because of accidents.

automatic collision avoidance), which are built and advertised to handle hazardous scenarios.

(a) RIP had collisions in 47.8% (478 out of 1000), 67.1% (671 out of 1000), and 44% (440 out of 1000) scenarios for ghost cut-in, lead cut-in, and lead slowdown driving scenarios, respectively[7]. Despite being designed for out-of-distribution scenarios, RIP underperforms compared to the baseline agent (LBC). RIP, as described in [16], selects the most pessimistic trajectory by using likelihood values from ensembles of imitation learning-based models. However, these likelihood values often do not correspond to the actual risks of the trajectories, especially in safety-critical scenarios [11], leading to failure to avoid accidents.

(b) LBC+TTC-based ACA helps reduce the total collision rate significantly compared to LBC and RIP for lead slowdown scenarios. However, it underperforms relative to iPrism on the ghost cut-in and lead cut-in scenario typologies because TTC-based ACA fails to account for out-of-path risky actors approaching from the side. The total collision rates (TCRs) for LBC+TTC-based ACA are 51.9%, 17.0%, and 1% for the ghost cut-in, lead cut-in, and lead slowdown driving scenario typologies, respectively. In comparison, the TCR for our LBC + iPrism is much lower than, or comparable to, that of LBC+TTC-based ACA. The TCRs for LBC+iPrism are 26.7%, 0.3%, and 1.5% for the ghost cut-in, lead cut-in, and lead slowdown driving scenario typologies, respectively.

Table IV shows the mitigation action activation time for TTC-based ACA and iPrism. As evidenced from the data, iPrism mitigates the risk more proactively than ACA. Specifically, iPrism achieves an improvement over TTC-based ACA of 0.57s, 3.73s, and 1.32s on ghost cut-in, lead cut-in, and lead slowdown, respectively. The worst-case lead time in mitigation in Table IV is 0.57 seconds for the ghost cut-in scenario; it still reduced the number of accidents by 49%.

STI is important for safety hazard mitigation. We conducted an ablation study in which iPrism's SMC was trained without STI in the RL reward formulation (labeled as LBC+SMC w/o STI). This agent matched the performance of LBC+SMC w/ STI (or iPrism) for the lead slowdown scenario. However, it did poorly in the ghost cut-in and lead cut-in scenarios, for which the timing of the mitigation is more critically important. The LBC+SMC w/o STI prevented only 1% and 2% of the accidents in the ghost cut-in and lead cut-in scenarios, respectively. The reason was that without STI, the SMC tends to activate at a suboptimal moment, either too early or too late, making it less effective in accident prevention.

Generalizable and compatible with other agents. To demonstrate that our proposed technique can work in conjunction with other agents, we applied iPrism trained on LBC to the RIP agent. Our results show that iPrism could generalize to the different agent and significantly reduce the number of accidents in each scenario typology. Collision avoidance (CA) rates of 86%, 61%, and 71% were achieved for the ghost cut-in, lead cut-in, and lead slowdown scenario typologies. We additionally evaluate RIP on a scenario typology that combines the Ghost cut-in typology with the roundabout scenario. The roundabout scenario typology was used by authors of [16] to demonstrate its efficacy. RIP collided in 84.3% (843 out of 1000[8]). In comparison, RIP+iPrism collided in 68.6% of scenarios, i.e., iPrism mitigates 18.6% of the accident caused by the RIP agent.

Extension to other mitigation actions. Rear-end driving typologies often cannot be addressed through braking alone, and most agents struggle with these hazardous situations. For instance, LBC experiences collisions in 77% (770 out of 1000) of such scenarios. The issue lies in the fact that braking by the *ego actor* in these scenarios often exacerbates the risk by rapidly increasing the relative velocity between the rear actor w.r.t. the *ego actor*, necessitating actions beyond braking for accident avoidance. Therefore, we enhanced iPrism's SMC to incorporate acceleration in addition to braking. Current safety mitigation techniques like ACA or RIP are ineffective in these scenarios. Nevertheless, iPrism successfully avoided accidents in 37% of the cases (282 out of 770). The inability of iPrism to prevent accidents in numerous instances can be attributed to the high velocity or acceleration of the trailing vehicle, making accidents inevitable even with maximum acceleration. In other words, an accident is unavoidable even with an oracle mitigation agent with acceleration as the only option.

Overall our results show:

(a) iPrism demonstrates superior performance in mitigating accidents compared to other techniques or agents across various scenario typologies.

(b) iPrism is designed for seamless integration with diverse agents, regardless of their specific architectural designs or algorithmic foundations.

(c) Despite its effectiveness, iPrism does not mitigate all accident scenarios, as certain accidents are unavoidable due to the dynamics of the actors (i.e., even an expert human driver or oracle would find it difficult to mitigate the accident in those scenarios) or the limited action space. We assert that including more action space will boost the performance of our iPrism, thereby enabling it to prevent more accidents.

D. Identifying Safety-critical Scenarios in the Real-world

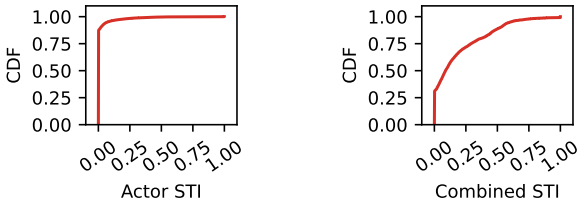
Since safety-critical scenarios are known to be rare in real-world datasets [14], [34], it is desirable to identify these sce-

[7]Refer to the TAS column of the RIP+SMC w/ STI (RIP+iPrism) row in Table III.

[8]1000 scenarios are generated following the methodology described in §IV-B1.

narios for testing and validation. We used STI to characterize and identify risky scenarios in real-world autonomous driving datasets, specifically the *Argoverse* dataset [9], because, as discussed in §V-B, PKL, TTC, and Dist. CIPA are not robust metrics for characterizing risk.

Characterizing STI. We measured the STI of actor(s) individually and collectively for all driving scenarios and time steps of the Argoverse dataset using equation (4) and equation (5), respectively. Our evaluation results show that the 50^{th}, 75^{th}, 90^{th}, and 99^{th} percentiles of the actor STI are 0.0, 0.0, 0.020, and 0.33, respectively, while the 50^{th}, 75^{th}, 90^{th}, and 99^{th} percentiles of the STI $^{\text{(combined)}}$ are 0.09, 0.29, 0.52, and 0.93, respectively, as depicted in Fig. 6. The STI of an individual actor and combined STI across all actors is zero for 90% and 50% of the time, respectively. High STI values are rare and fall into the long tail of the STI distribution. Like other studies [14], [34], our analysis highlights the limitations of current state-of-the-art real-world datasets in capturing the distribution of rare, safety-critical driving scenarios. Consequently, many scenario typologies required by NHTSA for AV/ADAS assessment are absent in these datasets, making these NHTSA typologies out-of-distribution.



(a) CDF of Actor STI **(b)** CDF of Combined STI

Figure 6: STI characterization of Argoverse dataset.

Identifying safety-critical scenarios from real-world data. Fig. 7 shows four distinct driving scenarios that were identified as risky using STI.

Case (a): Pedestrian crossing (Fig. 7(a)). Here a pedestrian crossing the street forces the ego actor to stop and yield, making it the most safety-threatening actor with a STI of 0.72.

Case (b): Oversized actor (Fig. 7(b)). In this scenario, an oversized actor in the adjacent lane is partially occupying the ego lane but has no intention to merge into the ego lane. The oversized actor dominates the STI with a value of 0.69 out of 1.0, making it the most safety-threatening actor.

Case (c): Cluttered environment (Fig. 7(c)). This cluttered scenario features actors exiting and entering the drivable lane. The actor behind the ego actor has a STI of 0, as it exits the lane, while the actor in red, entering the lane, has a STI of 0.35. The STI also highlights safety-critical actors like the badly parked one (in the orange box), which partially blocks the ego lane and poses a risk to the ego actor's safety.

Case (d): Actor pulling out (Fig. 7(d)). In this scenario, the *ego actor* is traveling in the bottom lane while two other actors are traveling in the top lane (red box). The STI values of these two actors are nonzero because each of them occupies part of the top lane into which the *ego actor* might maneuver if they were absent. In addition, the actor that is pulling out from its parking spot into the *ego lane* also results in a nonzero

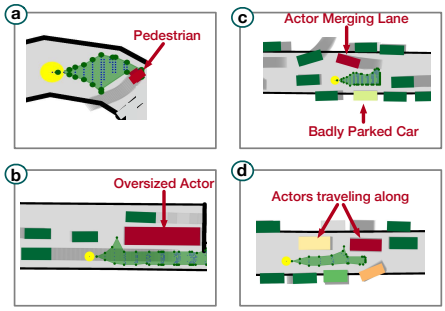


Figure 7: Safety-critical driving scenarios from *Argoverse* dataset (best viewed in color). The yellow circle denotes the ego actor, with its reach-tube shaded green and the road surface in gray. Driving scenario descriptions: (a) pedestrian crossing, (b) oversized actor blocking an adjacent lane, (c) cluttered with actors entering and exiting, (d) actor pulling out from a parking space. Actor (rectangle) color intensity ranges from green (less risky) to red (more risky), indicating risk relative to the scenario's most risky actor (in red).

STI value, as its future trajectory will constrain the available escape routes of the *ego actor*.

We observe the following across all these scenarios:

1) The closest actor to the ego actor is a heuristic also used by others, such as Waymo [34], but it may not always be the most risky actor, because the closest actor does not necessarily block the largest number of escape routes, as illustrated in Fig. 7(b), Fig. 7(c), and Fig. 7(d).
2) The closest in-path actor might not pose the highest risk, an intuition encapsulated in TTC and Dist. CIPA, as the risk also depends on how much an actor is reducing the number of escape routes. For example, in driving scenario Fig. 7(b) no actor has a trajectory intersecting with that of the ego actor; however, an inherent risk is posed by the oversized actor because it blocks escape routes for the ego actor.
3) The total risk to the ego actor increases with an increasing number of blocked escape routes. Since each actor in the typology can block multiple escape routes, one must consider the combined risk ($\text{STI}^{\text{(combined)}}$) posed by all actors at any given time step. For example, the scenarios in Fig. 7(c) and Fig. 7(d) have higher combined risk than those in Fig. 7(a) and Fig. 7(b).

E. Execution Overheads

STI evaluation overheads. On the average, it took 0.61 seconds to evaluate STI on our evaluation platform. There are multiple optimization opportunities that would accelerate the STI evaluation, such as: (i) use of a high-performance programming language (e.g., C++) and libraries instead of Python, (ii) propagation of reachable states in parallel, and (iii) tuning of hyperparameters for better performance.

SMC training and inference overheads. SMC is trained for 100 episodes per scenario typology, where each episode takes, on average, 344 seconds to complete. Furthermore, the average inference time of SMC is 0.012 seconds, making it suitable for operation alongside the original ADS, as the planning period of an ADS is typically 0.1–0.3 seconds [52].

Table V: Related work.

Related Work	Relevance	Methodology	Limitations
Kinematics-based risk metrics (e.g., TTC, Dist. CIPA) [10]–[13], [27]–[33]	Assess risk	Estimate collision likelihood using kinematics.	(i) Fail to consider out-of-path actors (e.g., [11], [13]), or (ii) Need extensive kinematic models and constraints to calculate collision probability (e.g., [28], [31], [32]).
Planner-centric metrics [14], [34], [35]	Improve perception & planning for safety	Quantify actor's influence on the AV's planning decision distribution to improve perception and planning.	(i) The magnitude of change in the AV's planning decision correlates poorly with accident probability, or (ii) Assume access to the planner or perception, or (iii) Modifies the planner or perception.
Rule-based collision mitigation techniques [11], [13]	Mitigate collision	Thresholding of kinematics based metric such as TTC and Dist. CIPA.	(i) Activate after threshold violations have occurred (reactive) resulting in less time available for mitigation, or (ii) Consider primarily in-path actors.
Learning-based collision mitigation techniques [15], [16], [53]	Improve planning	Improve safety by learning from human demonstrations and online adaptation to OOD scenarios.	(i) Require a large amount of data on safety-critical scenarios, or (ii) Requires expert demonstration or intervention (e.g., [16]).

VI. Related Work

Incorporating safety into autonomous systems is a critical aspect, typically integrated within the planning module of the ADS. However, despite these efforts, there are still notable challenges and areas for improvement. This section outlines the major research directions addressing the safety issues: (a) risk assessment [10]–[13], [27]–[33], [54], [55], which focuses on characterizing risk to identify potential safety hazards; (b) learning to plan [15], [53], [56], which learns to make safe planning decisions by interacting the environment or from expert demonstration; (c) out-of-distribution (OOD) detection [16], [24], [57], which is crucial for handling scenarios that fall outside the typical range of expected conditions, (d) mitigation [11], [13], [30], [58], which involves strategies to reduce or manage risks; and (e) validation & verification [37], [44], [46], [59]–[63], which ensures correctness of the AV controller.

We describe and identify limitations of related work that directly address risk and safety hazard mitigation in Table V. In comparison, inspired by how experienced human drivers proactively mitigate hazardous situations, this paper introduces a new metric called STI that uses the concept of escape routes to characterize risk by considering all the actors in the risk envelope. The evaluation of STI does not require any pre-training. None of the previous approaches use the idea of escape routes as a way to characterize risk. Our paper also proposes a safety-hazard mitigation controller (SMC) that actively reduces risk to avoid accidents

VII. Conclusion

We propose iPrism, an accident mitigation framework, that encompasses (i) a new risk metric – a Safety-Threat Indicator (STI) that quantifies the change in ego actor's escape routes due the presence of other actors, and (ii) an RL-based Safety-hazard Mitigation controller (SMC) that proactively lowers the STI to improve ego actor's safety.

A significant challenge in evaluation is that real-world datasets lack risky scenarios, limiting comprehensive assessments. Hence, we developed five multi-actor safety-critical typologies based on the NHTSA pre-crash report [8] that account for approximately 80% of accidents in the United States. Sampling these typologies, we created 4810 safety-critical scenarios that can serve as a rigorous benchmark for future safety-focused research.

There are limitations of our current implementation: (i) The RL-based SMC has been demonstrated on braking and acceleration, in the present examples, excluding complex maneuvers like lane changes. Executing these complex maneuvers requires closer integration of the RL-based SMC with the ADS to avoid potential conflicting decisions between the ADS and the SMC. Multi-agent RL provides the framework for continuously resolving these conflicts, and our future research will explore these directions. (ii) STI currently evaluates all escape routes equally in terms of accident avoidance when assessing risk. It is indeed true that the ego actor can use any of these routes as they are deemed safe. However, some escape routes may pose greater danger if chosen by the ego actor. While SMC is trained to select the most appropriate escape route through reinforcement learning, STI lacks this capability. Therefore, our future research will focus on integrating the potential consequences of choosing specific escape paths into the assessment of future risk by STI.

Despite limitation, our evaluation of the RL-based SMC on NHTSA-based safety-critical scenarios shows: (i) SMC significantly enhances safety, reducing accident occurrences by 37% to 98% compared to a baseline learning-by-cheating agent, and (ii) achieving up to 72.7% accident prevention when compared to the leading safety hazard mitigation agents.

Finally, iPrism evaluates and mitigates risks imposed by other actors, i.e., actor-related risks. The assessment and handling of non-actor-related risks, such as those caused by weather, road conditions, sensor occlusion, and faulty software or hardware systems, are orthogonal to the scope of this work. That said, assessing and managing risk holistically, including both actor-related and non-actor-related risks, is an important and open research problem that needs attention from the safety community.

VIII. Acknowledgement

We thank Professor Zubair Baig (shepherd) and anonymous reviewers for their valuable feedback. We thank A. Patke, H. Qiu, M. Barletta, H. Sreejith, J. Applequist, and K. Atchley for providing insightful comments on the earlier drafts of this paper. This work is supported by the National Science Foundation (NSF) under Grant No. 2029049 and by the IBM-ILLINOIS Discovery Accelerator Institute (IIDAI). Any opinions, findings, conclusions, or recommendations expressed in this work are those of the authors and do not necessarily reflect the views of NSF or IBM.

References

[1] B. Templeton, "Waymo and cruise have both hit 1m miles with no driver, but waymo publishes detailed safety data," Mar 2023. [Online]. Available: https://www.forbes.com/sites/bradtempleton/2023/02/28/waymo-and-cruise-have-both-hit-1m-miles-with-no-driver-but-waymo-published-detailed-safety-data/?sh=2cb60e221421

[2] S. Alvarez, "Research group demos why Tesla Autopilot could crash into a stationary vehicle," https://www.teslarati.com/tesla-research-group-autopilot-crash-demo/, June 2018.

[3] A. J. Hawkins, "Waymo's driverless cars were involved in two crashes and 18 'minor contact events' over 1 million miles," https://www.theverge.com/2023/2/28/23617278/waymo-self-driving-driverless-crashes-av, 2023.

[4] Y. Lu, Oct 2023. [Online]. Available: https://www.nytimes.com/2023/10/26/technology/cruise-driverless-taxi-united-states.html

[5] S. S. Banerjee, S. Jha, J. Cyriac, Z. T. Kalbarczyk, and R. K. Iyer, "Hands off the wheel in autonomous vehicles?: A systems perspective on over a million miles of field data," in *2018 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, 2018, pp. 586–597.

[6] R. MITCHELL, "Most driver-assist crashes involved teslas, new data show. but questions abound," Jun 2022. [Online]. Available: https://www.latimes.com/business/story/2022-06-15/tesla-autopilot-crash-report-nhtsa

[7] H. New, "Chapter 8: Defensive driving," Jul 2021. [Online]. Available: https://dmv.ny.gov/about-dmv/chapter-8-defensive-driving

[8] National Highway Traffic Safety Administration (NHTSA), "Pre-crash scenario typology for crash avoidance research DOT HS 810 767," 2007.

[9] M.-F. Chang, J. Lambert, P. Sangkloy, J. Singh, S. Bak, A. Hartnett, D. Wang, P. Carr, S. Lucey, D. Ramanan, and J. Hays, "Argoverse: 3d tracking and forecasting with rich maps," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019, Dataset: CC BY-NC-SA 4.0 License, tools and code: MIT License.

[10] A. Tarko, *Chapter 17. Surrogate Measures of Safety*, 04 2018, pp. 383–405.

[11] G. Li, Y. Yang, T. Zhang, X. Qu, D. Cao, B. Cheng, and K. Li, "Risk assessment based collision avoidance decision-making for autonomous vehicles in multi-scenarios," *Transportation Research Part C: Emerging Technologies*, vol. 122, p. 102820, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0968090X20307257

[12] L. Westhofen, C. Neurohr, T. Koopmann, M. Butz, B. Schütt, F. Utesch, B. Neurohr, C. Gutenkunst, and E. Böde, "Criticality metrics for automated driving: A review and suitability analysis of the state of the art," *Archives of Computational Methods in Engineering*, Aug 2022. [Online]. Available: https://doi.org/10.1007/s11831-022-09788-7

[13] H. Bae, G. Lee, J. Yang, G. Shin, G. Choi, and Y. Lim, "Estimation of the closest in-path vehicle by low-channel lidar and camera sensor fusion for autonomous vehicles," *Sensors*, vol. 21, no. 9, 2021. [Online]. Available: https://www.mdpi.com/1424-8220/21/9/3124

[14] J. Philion, A. Kar, and S. Fidler, "Learning to evaluate perception models using planner-centric metrics," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 14 052–14 061.

[15] D. Chen, B. Zhou, V. Koltun, and P. Krähenbühl, "Learning by cheating," in *Proceedings of the Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, L. P. Kaelbling, D. Kragic, and K. Sugiura, Eds., vol. 100. PMLR, 30 Oct–01 Nov 2020, pp. 66–75. [Online]. Available: http://proceedings.mlr.press/v100/chen20a.html

[16] A. Filos, P. Tigkas, R. Mcallister, N. Rhinehart, S. Levine, and Y. Gal, "Can autonomous vehicles identify, recover from, and adapt to distribution shifts?" in *Proceedings of the 37th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, H. D. III and A. Singh, Eds., vol. 119. PMLR, 13–18 Jul 2020, pp. 3145–3153. [Online]. Available: https://proceedings.mlr.press/v119/filos20a.html

[17] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nuscenes: A multimodal dataset for autonomous driving," *arXiv preprint arXiv:1903.11027*, 2019.

[18] Y. Ovadia, E. Fertig, J. Ren, Z. Nado, D. Sculley, S. Nowozin, J. Dillon, B. Lakshminarayanan, and J. Snoek, "Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift," in *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds., vol. 32. Curran Associates, Inc., 2019. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2019/file/8558cb408c1d76621371888657d2eb1d-Paper.pdf

[19] D. Amodei, C. Olah, J. Steinhardt, P. Christiano, J. Schulman, and D. Mané, "Concrete problems in ai safety," 2016.

[20] M. Sugiyama and M. Kawanabe, *Machine Learning in Non-Stationary Environments: Introduction to Covariate Shift Adaptation*. The MIT Press, 03 2012. [Online]. Available: https://doi.org/10.7551/mitpress/9780262017091.001.0001

[21] J. Quionero-Candela, M. Sugiyama, A. Schwaighofer, and N. D. Lawrence, *Dataset Shift in Machine Learning*. The MIT Press, 2009.

[22] J. Leike, M. Martic, V. Krakovna, P. A. Ortega, T. Everitt, A. Lefrancq, L. Orseau, and S. Legg, "Ai safety gridworlds," 2017.

[23] J. Henriksson, S. Ursing, M. Erdogan, F. Warg, A. Thorsén, J. Jaxing, O. Örsmark, and M. O. Toftås, "Out-of-distribution detection as support for autonomous driving safety lifecycle," in *Requirements Engineering: Foundation for Software Quality: 29th International Working Conference, REFSQ 2023, Barcelona, Spain, April 17–20, 2023, Proceedings*. Berlin, Heidelberg: Springer-Verlag, 2023, p. 233–242. [Online]. Available: https://doi.org/10.1007/978-3-031-29786-1_16

[24] J. Nitsch, M. Itkina, R. Senanayake, J. Nieto, M. Schmidt, R. Siegwart, M. J. Kochenderfer, and C. Cadena, "Out-of-distribution detection for automotive perception," in *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*. IEEE Press, 2021, p. 2938–2943. [Online]. Available: https://doi.org/10.1109/ITSC48978.2021.9564545

[25] D. Hendrycks, M. Mazeika, and T. Dietterich, "Deep anomaly detection with outlier exposure," *Proceedings of the International Conference on Learning Representations*, 2019.

[26] Y. Li, C. Wang, X. Xia, T. Liu, X. Miao, and B. An, "Out-of-distribution detection with an adaptive likelihood ratio on informative hierarchical VAE," in *Advances in Neural Information Processing Systems*, A. H. Oh, A. Agarwal, D. Belgrave, and K. Cho, Eds., 2022. [Online]. Available: https://openreview.net/forum?id=vMQ1V_z0TxU

[27] M. Maurer, *Forward Collision Warning and Avoidance*. London: Springer London, 2012, pp. 657–687. [Online]. Available: https://doi.org/10.1007/978-0-85729-085-4_25

[28] A. Tejada and M. J. E. Legius, "Towards a quantitative " safety " metric for autonomous vehicles," 2019.

[29] S. H. Haus, R. Sherony, and H. C. Gabler, "Estimated benefit of automated emergency braking systems for vehicle–pedestrian crashes in the united states," *Traffic Injury Prevention*, vol. 20, pp. S171–S176, 2019, peer-Reviewed Journal for the 26th International Technical Conference on the Enhanced Safety of Vehicles (ESV). [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1538958822012279

[30] Y. Wang, C. Wang, W. Zhao, and C. Xu, "Decision-making and planning method for autonomous vehicles based on motivation and risk assessment," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 1, pp. 107–120, 2021.

[31] N. David, L. Hon-Leung, N. Julia, and W. Yizhou, "An introduction to the safety force field," https://www.nvidia.com/content/dam/en-zz/Solutions/self-driving-cars/safety-force-field/an-introduction-to-the-safety-force-field-updated.pdf, 2019.

[32] S. Shalev-Shwartz, S. Shammah, and A. Shashua, "On a formal model of safe and scalable self-driving cars," *CoRR*, vol. abs/1708.06374, 2017. [Online]. Available: http://arxiv.org/abs/1708.06374

[33] J. B. Cicchino, "Effectiveness of forward collision warning and autonomous emergency braking systems in reducing front-to-rear crash rates," *Accident Analysis and Prevention*, vol. 99, pp. 142–152, 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0001457516304006

[34] E. Tolstaya, R. Mahjourian, C. Downey, B. Vadarajan, B. Sapp, and D. Anguelov, "Identifying driver interactions via conditional behavior prediction," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 3473–3479.

[35] B. Ivanovic and M. Pavone, "Injecting planning-awareness into prediction and detection evaluation," in *2022 IEEE Intelligent Vehicles Symposium (IV)*, 2022, pp. 821–828.

[36] A. B. Kurzhanski and P. Varaiya, "Ellipsoidal techniques for reachability analysis," in *Hybrid Systems: Computation and Control*, N. Lynch and B. H. Krogh, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2000, pp. 202–214.

[37] C. Fan, B. Qi, S. Mitra, and M. Viswanathan, "Dryvr: Data-driven verification and compositional reasoning for automotive systems," in *Computer Aided Verification*, R. Majumdar and V. Kunčak, Eds. Cham: Springer International Publishing, 2017, pp. 441–461.

[38] A. B. Kurzhanski and T. F. Filippova, *On the Theory of Trajectory Tubes — A Mathematical Formalism for Uncertain Dynamics, Viability and Control*. Boston, MA: Birkhäuser Boston, 1993, pp. 122–188. [Online]. Available: https://doi.org/10.1007/978-1-4612-0349-0_4

[39] I. M. Mitchell, "Scalable calculation of reach sets and tubes for nonlinear systems with terminal integrators: a mixed implicit explicit formulation," in *Proceedings of the 14th International Conference on Hybrid Systems: Computation and Control*, ser. HSCC '11. New York, NY, USA: Association for Computing Machinery, 2011, p. 103–112. [Online]. Available: https://doi.org/10.1145/1967701.1967718

[40] I. M. Mitchell, "Comparing forward and backward reachability as tools for safety analysis," in *Proceedings of the 10th International Conference on Hybrid Systems: Computation and Control*, ser. HSCC'07. Berlin, Heidelberg: Springer-Verlag, 2007, p. 428–443.

[41] A. B. Kurzhanski and P. Varaiya, *Dynamics and control of trajectory tubes; Theory and computation*, 2014th ed., ser. Systems & Control: Foundations & Applications. Springer International Publishing, Jan. 2014.

[42] J. Kong, M. Pfeiffer, G. Schildbach, and F. Borrelli, "Kinematic and dynamic vehicle models for autonomous driving control design," in *2015 IEEE Intelligent Vehicles Symposium (IV)*, 2015, pp. 1094–1099.

[43] P. Polack, F. Altché, B. d'Andréa Novel, and A. de La Fortelle, "The kinematic bicycle model: A consistent model for planning feasible trajectories for autonomous vehicles?" in *2017 IEEE Intelligent Vehicles Symposium (IV)*, 2017, pp. 812–818.

[44] Y. Li, H. Zhu, K. Braught, K. Shen, and S. Mitra, "Verse: A python library for reasoning about multi-agent hybrid system scenarios," in *Computer Aided Verification*, C. Enea and A. Lal, Eds. Cham: Springer Nature Switzerland, 2023, pp. 351–364.

[45] D. Sun and S. Mitra, "Neureach: Learning reachability functions from simulations," in *Tools and Algorithms for the Construction and Analysis of Systems*, D. Fisman and G. Rosu, Eds. Cham: Springer International Publishing, 2022, pp. 322–337.

[46] S. Jha, S. Banerjee, T. Tsai, S. K. S. Hari, M. B. Sullivan, Z. T. Kalbarczyk, S. W. Keckler, and R. K. Iyer, "ML-based fault injection for autonomous vehicles: A case for Bayesian Fault Injection," in *2019 49th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, June 2019, pp. 112–124.

[47] H. van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double q-learning," *CoRR*, vol. abs/1509.06461, 2015. [Online]. Available: http://arxiv.org/abs/1509.06461

[48] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," in *Proceedings of the 1st Annual Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, S. Levine, V. Vanhoucke, and K. Goldberg, Eds., vol. 78. PMLR, 13–15 Nov 2017, pp. 1–16. [Online]. Available: http://proceedings.mlr.press/v78/dosovitskiy17a.html

[49] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing atari with deep reinforcement learning," *arXiv preprint arXiv:1312.5602*, 2013.

[50] NHTSA, "A framework for automated driving system testable cases and scenarios," United States. National Highway Traffic Safety Administration, Tech. Rep., 2018.

[51] W. G. Najm, J. D. Smith, M. Yanagisawa *et al.*, "Pre-crash scenario typology for crash avoidance research," United States. National Highway Traffic Safety Administration, Tech. Rep., 2007.

[52] H. Fan, F. Zhu, C. Liu, L. Zhang, L. Zhuang, D. Li, W. Zhu, J. Hu, H. Li, and Q. Kong, "Baidu apollo em motion planner," 2018.

[53] N. Rhinehart, R. McAllister, and S. Levine, "Deep imitative models for flexible inference, planning, and control," in *International Conference on Learning Representations*, 2020. [Online]. Available: https://openreview.net/forum?id=Skl4mRNYDr

[54] X. Zhou, B. Ahmed, J. H. Aylor, P. Asare, and H. Alemzadeh, "Hybrid knowledge and data driven synthesis of runtime monitors for cyber-physical systems," *IEEE Transactions on Dependable and Secure Computing*, pp. 1–18, 2023.

[55] H. Wang, B. Lu, J. Li, T. Liu, Y. Xing, C. Lv, D. Cao, J. Li, J. Zhang, and E. Hashemi, "Risk assessment and mitigation in local path planning for autonomous vehicles with lstm based predictive model," *IEEE Transactions on Automation Science and Engineering*, vol. 19, no. 4, pp. 2738–2749, 2022.

[56] F. Codevilla, M. Miiller, A. López, V. Koltun, and A. Dosovitskiy, "End-to-end driving via conditional imitation learning," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE Press, 2018, p. 1–9. [Online]. Available: https://doi.org/10.1109/ICRA.2018.8460487

[57] M. Yuhas, Y. Feng, D. J. X. Ng, Z. Rahiminasab, and A. Easwaran, "Embedded out-of-distribution detection on an autonomous robot platform," in *Proceedings of the Workshop on Design Automation for CPS and IoT*, ser. Destion '21. New York, NY, USA: Association for Computing Machinery, 2021, p. 13–18. [Online]. Available: https://doi.org/10.1145/3445034.3460509

[58] K. Lee and D. Kum, "Collision avoidance/mitigation system: Motion planning of autonomous vehicle via predictive occupancy map," *IEEE Access*, vol. 7, pp. 52 846–52 857, 2019.

[59] S. Jha, S. Cui, S. Banerjee, J. Cyriac, T. Tsai, Z. Kalbarczyk, and R. K. Iyer, "ML-driven malware that targets AV safety," in *2020 50th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*. IEEE, 2020, pp. 113–124.

[60] S. Jha, S. Cui, T. Tsai, S. K. S. Hari, M. B. Sullivan, Z. T. Kalbarczyk, S. W. Keckler, and R. K. Iyer, "Exploiting temporal data diversity for detecting safety-critical faults in av compute systems," in *2022 52nd Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, 2022, pp. 88–100.

[61] C. Dawson, B. Lowenkamp, D. Goff, and C. Fan, "Learning safe, generalizable perception-based hybrid control with certificates," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 1904–1911, 2022.

[62] D. J. Fremont, E. Kim, Y. V. Pant, S. A. Seshia, A. Acharya, X. Bruso, P. Wells, S. Lemke, Q. Lu, and S. Mehta, "Formal scenario-based testing of autonomous vehicles: From simulation to the real world," in *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*. IEEE Press, 2020, p. 1–8. [Online]. Available: https://doi.org/10.1109/ITSC45102.2020.9294368

[63] T. Dreossi, D. J. Fremont, S. Ghosh, E. Kim, H. Ravanbakhsh, M. Vazquez-Chanlatte, and S. A. Seshia, "VerifAI: A toolkit for the formal design and analysis of artificial intelligence-based systems," in *31st International Conference on Computer Aided Verification (CAV)*, Jul. 2019.