

A Theoretical Analysis

A.1 Feasibility-depend Advantage Function

In [13], the feasibility-dependent advantage function relates to $A_r^{\pi_{\theta_k}}(s, a)$ and $A_h^{\pi_{\theta_k}}(s, a)$. Since the LFR of the AV is pre-calculated and independent of the policy π , we can substitute $A_h^{\pi_{\theta_k}}(s, a)$ with $A_h^*(s, a)$. Thus, the combined advantage function is defined as:

$$\overline{A}^{\pi_{\theta_k}}(s, a) = A_r^{\pi_{\theta_k}}(s, a) + \lambda_{\xi}(s) \cdot A_h^*(s, a), \quad (10)$$

where $\lambda_{\xi}(s)$ acts as an indicator function. For feasible states, $\lambda_{\xi}(s) \rightarrow 0$ is finite, and for unfeasible states, $\lambda_{\xi}(s) \rightarrow +\infty$. Consequently, the feasibility-dependent advantage function simplifies to:

$$\overline{A}^{\pi_{\theta_k}}(s, a) = A_r^{\pi_{\theta_k}}(s, a) \cdot \mathbb{I}_{s \in \mathcal{S}_f^*} + A_h^*(s, a) \cdot \mathbb{I}_{s \notin \mathcal{S}_f^*} \quad (11)$$

Aligning with Definition 2 and the viewpoint transformation function $g(\cdot)$, we reinterpret \mathcal{S}_f^* using the optimal feasible state-value function V_h^* , which is relative to the AV's state:

$$\overline{A}^{\pi_{\theta_k}}(s, a) = A_r^{\pi_{\theta_k}}(s, a) \cdot \mathbb{I}_{V_h^*(g(s)) \leq 0} + A_h^*(s, a) \cdot \mathbb{I}_{V_h^*(g(s)) > 0} \quad (12)$$

Given that the PPO-based method calculates the advantage solely from the trajectories stored in the buffer, we propose imposing stricter constraints on $A_r^{\pi_{\theta_k}}(s, a)$. Specifically, if the AV's next state falls outside the LFR, the optimization should prioritize minimizing feasibility violations over maximizing adversarial rewards. Consequently, we formulate the final advantage function in Eq. (6).

A.2 Proof of Lemma 1

According to Eq. (2), $Q_h^*(s^{\text{AV}}, a^{\text{AV}})$ is solely dependent on $h(\cdot)$, which in turn is influenced by $s_t^{\text{AV}}, t \in \mathbb{N}$. The action a_t^{AV} influences only the subsequent state s_{t+1}^{AV} in conjunction with the environment transition function. Therefore, in a deterministic environment, this relationship can be described as follows:

$$Q_h^*(s^{\text{AV}}, a^{\text{AV}}) := \min_{\pi^{\text{AV}}} \max_{t \in \mathbb{N}} \{h(s_0^{\text{AV}}), h(s_{t+1}^{\text{AV}})\}, \\ s_0^{\text{AV}} = s^{\text{AV}}, a_0^{\text{AV}} = a^{\text{AV}}, a_{t+1}^{\text{AV}} \sim \pi^{\text{AV}}(\cdot | s_{t+1}^{\text{AV}}), s_{t+1}^{\text{AV}} = P(s_t^{\text{AV}}, a_t^{\text{AV}}), \quad (13)$$

where $P(\cdot)$ denotes the deterministic transition dynamics. However, during the training of our CBV method, the policy varies significantly, impacting s_{t+1}^{AV} through both the AV's action a_t^{AV} and the CBV's action a_t . Given that other BVs adhere to a consistent rule-based policy, we can reasonably assume that the environment, excluding the CBV and AV, is deterministic. Thus, we can redefine the equation as follows:

$$Q_h^*(s^{\text{AV}}, a^{\text{AV}}, s, a) := \min_{\pi^{\text{AV}}} \max_{t \in \mathbb{N}} \{h(s_0^{\text{AV}}), h(s_{t+1}^{\text{AV}})\}, \\ s_0^{\text{AV}} = s^{\text{AV}}, a_0^{\text{AV}} = a^{\text{AV}}, s_0 = s, a_0 = a, s_{t+1}^{\text{AV}} = P(s_t^{\text{AV}}, a_t^{\text{AV}}, s_t, a_t), s_{t+1} = g(s_{t+1}^{\text{AV}}) \\ a_{t+1}^{\text{AV}} \sim \pi^{\text{AV}}(\cdot | s_{t+1}^{\text{AV}}), a_{t+1} \sim \pi^{\text{CBV}}(\cdot | s_{t+1}), \quad (14)$$

Additionally, based on Definition 1 and Eq. (14), we identify two cases:

Case1: if $h(s^{\text{AV}'}) \geq h(s^{\text{AV}})$, then:

$$Q_h^*(s^{\text{AV}}, a^{\text{AV}}, s, a) = \min_{\pi^{\text{AV}}} \max_{t \in \mathbb{N}} \{h(s_0^{\text{AV}}), h(s_{t+1}^{\text{AV}})\} = \min_{\pi^{\text{AV}}} \max_{t \in \mathbb{N}} \{h(s_{t+1}^{\text{AV}})\} = V_h^*(s^{\text{AV}'}) \quad (15)$$

Case2: if $h(s^{\text{AV}'}) < h(s^{\text{AV}})$, then:

$$Q_h^*(s^{\text{AV}}, a^{\text{AV}}, s, a) = \min_{\pi^{\text{AV}}} \max_{t \in \mathbb{N}} \{h(s_0^{\text{AV}}), h(s_{t+1}^{\text{AV}})\} = \max\{h(s^{\text{AV}}), V_h^*(s^{\text{AV}'})\} \quad (16)$$

Building upon Eqs. (15) and (16), we conclude in Eq. (17) that $Q_h^*(s^{AV}, a^{AV}, s, a)$ primarily depends on the states of the AV. The actions of both the AV and the CBV mainly influence state transitions.

$$Q_h^*(s^{AV}, a^{AV}, s, a) = Q_h^*(s^{AV}, s^{AV'}) = \begin{cases} V_h^*(s^{AV'}) & h(s^{AV'}) \geq h(s^{AV}) \\ \max\{h(s^{AV}), V_h^*(s^{AV'})\} & h(s^{AV'}) < h(s^{AV}) \end{cases} \quad (17)$$

Thus, through the viewpoint transformation function $g(\cdot)$ and a deterministic AV policy π^{AV} , we derive the optimal feasible action-value function, taking into account both the state and action of the CBV:

$$\begin{aligned} Q_h^*(s^{AV}, a^{AV}) &= Q_h^*(s^{AV}, a^{AV}, s, a) \\ &= Q_h^*(s^{AV}, s^{AV'}) \\ &= Q_h^*(g(s), g(s')) \\ &= \begin{cases} V_h^*(g(s')) & h(g(s')) \geq h(g(s)) \\ \max\{h(g(s)), V_h^*(g(s'))\} & h(g(s')) < h(g(s)) \end{cases} \end{aligned} \quad (18)$$

B Experiment Details

Building on foundational concepts from [3, 4], we apply our *FREA* method to critical background vehicles (CBVs) within traffic flows. We first outline the mechanisms for specifying and withdrawing CBVs, as detailed in Appendix B.1. We then discuss the safe RL-based setting implemented in our *FREA* method, described in Appendix B.2. Finally, we provide implementation details of the baseline methods in Appendix B.3.

B.1 Specifying and Withdrawal of CBVs

To appropriately select CBVs and exclude unsuitable BVs, we established criteria to filter out ineligible candidates:

- Case1: The BV is located in the opposing lane relative to the AV.
- Case2: The distance between the BV and AV exceeds 25 meters.
- Case3: The BV is positioned behind the AV with a relative yaw angle greater than 90 degrees, indicating no interaction.
- Case4: The BV has previously served as a CBV and has reached its goal in the scenario.

Based on the predefined criteria, our system assesses the scenario at each simulation step in Carla. If the number of active CBVs drops below a predefined threshold, the nearest candidate to the AV is automatically selected as a new CBV. To prevent disruptions in normal traffic flow during training, we implemented a withdrawal mechanism for CBVs that specifies conditions for their removal, whether they complete their tasks or not.

- Case1: The CBV achieves its objective (it is then terminated and reverts to a standard BV).
- Case2: The BV is positioned behind the AV with a relative yaw angle greater than 90 degrees (it is truncated and reverts to a standard BV).
- Case3: The CBV obstructs traffic flow or exceeds the maximum allowed duration (it is truncated and reverts to a standard BV).
- Case4: The CBV collides with any BV or the AV (it is terminated and removed from the simulation).

This selection and withdrawal mechanism effectively manages the CBV training process. However, considering the potential limitations of these rules, developing more intelligent strategies remains a future research direction.

B.2 Safe RL-based Setting

In this framework, each CBV acts as an RL agent tasked with attacking the AV while maintaining the AV’s feasibility. Previous works [5, 12] primarily focused on minimizing the distance between CBV and AV, often leading to unavoidable collisions, as discussed in Section 2. To mitigate this, we replace the collision reward with the goal-based adversarial reward, which encourages the CBV to reach a potential collision point with the AV. The configuration details are as follows:

State. The state of each CBV is represented by an array of dimensions $(V + 2) \times F$, where V represents the number of nearby vehicles and F the features of these vehicles, the AV, and the goal point. Recorded features include relative x and y positions, object extent along the X and Y axes, relative yaw angle, and absolute speed. For the goal point, speed is replaced by relative distance. The features of the AV and the goal point are positioned in the first and second rows of the array, respectively.

Action. Following the guidelines in [25], we define a continuous action space with specific constraints to prevent unreasonable actions. The acceleration range is set between -3 and 3 , and the steering angle is limited to a maximum absolute value of 0.3 .

Reward. As previously discussed, we have replaced the collision reward with a goal-based reward. This adjustment encourages the CBV to navigate toward a potential collision point while avoiding collisions with other BVs. In practice, we set the local target for the AV as the potential collision point, which forms the overall reward function:

$$R_t = d(\text{CBV}_{t-1}, \text{Goal}_{t-1}) - d(\text{CBV}_t, \text{Goal}_t) + 15 * r_t^{\text{collision}} + 15 * r_t^{\text{finish}}, \quad (19)$$

where $d(\text{CBV}_t, \text{Goal}_t)$ denotes the Euclidean distance between the CBV and its goal point at time t , the collision reward $r_t^{\text{collision}}$ is set to -1 if the CBV collides with any BVs at time t , and the goal-reaching reward r_t^{finish} is set to 1 if the CBV is within 2 meters radius of the goal at time t .

B.3 Implementation Details of Baselines.

B.3.1 Algorithms.

To evaluate the performance of *FREA*, we propose three CBV methods as baselines for a comprehensive quantitative comparison:

Standard. This method utilizes a rule-based autopilot policy implemented in the Carla Simulator [24] to generate realistic urban traffic flows.

PPO. This method employs an adversarial CBV policy based on PPO that aims to reach a potential collision point with the AV, utilizing the adversarial reward outlined in Eq. (19).

FPPO-RS. This method integrates a feasibility penalty term into the PPO-based adversarial policy, which penalizes violations of the AV’s feasibility constraints in the adversarial reward function. The modified adversarial reward function used during training is specified as follows:

$$R_t^{\text{fea}} = R_t - \frac{\text{clip}(V_h^*(g(s_t)), 0, f_{\max}) \cdot p_{\max}}{f_{\max}}, \quad (20)$$

where f_{\max} represents the upper bound for feasible value clipping, and p_{\max} is the upper bound for penalty rewards.

B.3.2 Hyperparameters.

Table 3 shows the hyperparameters of baseline methods.

Training Curves about CBV methods. To ensure robustness in our training process, we aggregated results from three different random seeds, as shown in Figure 6. This illustration confirms that all three CBV methods converge well in various scenarios. Furthermore, the PPO method achieves the highest adversarial reward, reflecting its focus on adversarial objectives. Conversely, FPPO-RS and

Table 3: Detailed hyperparameters of *FREA* and baselines.

Parameter	Value
PPO, FPPO-RS, <i>FREA</i> shared.	
Optimizer	Adam ($\epsilon = 1e - 5$)
Approximation function	Multi-layer Perceptron
Number of hidden layers	2
Number of hidden units per layer	256
Nonlinearity of hidden layer	RELU
Nonlinearity of output layer	linear
Critic learning rate	Linear annealing $3e - 4 \rightarrow 0$
Reward discount factor (γ)	0.98
GAE parameters (λ)	0.98
entropy parameters	0.01
Batch size	256
horizon length	2048
update repeat times:	4
Max episode length (N)	2000
Actor learning rate	Linear annealing $3e - 4 \rightarrow 0$
Clip ratio	0.2
FPPO-RS	
feasible value clip upper bound f_{max}	8
penalty reward upper bound p_{max}	1

452 *FREA* need to balance adversariality and AV’s feasibility, resulting in a lower adversarial reward.
 453 Given our focus on creating reasonable adversarial scenarios, achieving appropriate adversariality is
 454 more important than merely pursuing high adversarial rewards.

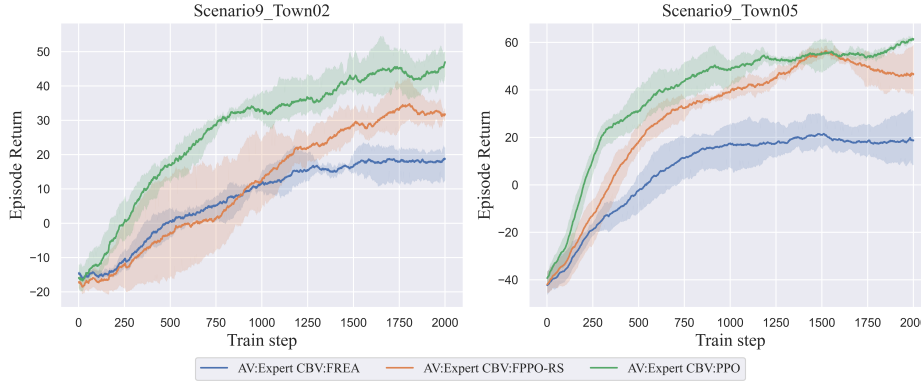


Figure 6: Episode return of different CBV methods.

455 C Largest Feasible Region: Training and Application

456 C.1 Training Details about LFR

457 **Offline Datasets.** As highlighted in [15], extensive coverage of the state space in datasets is crucial
 458 for determining the LFR of AVs using offline RL. Following this guideline, we employed the Expert
 459 [26] and Behavior [24] agents as surrogate AVs, collecting $100k$ instances of interaction data under
 460 standard traffic conditions for each. Additionally, to introduce randomness, we employed the PPO
 461 method as CBV with the Expert as AV, gathering another $100k$ instances of interaction data. This re-
 462 sulted in a comprehensive dataset of $300k$ interaction data for LFR training. As depicted in Figure 7,
 463 the offline dataset extensively covers most of the potential state space, satisfying the requirements
 464 for training an optimal feasible value function.

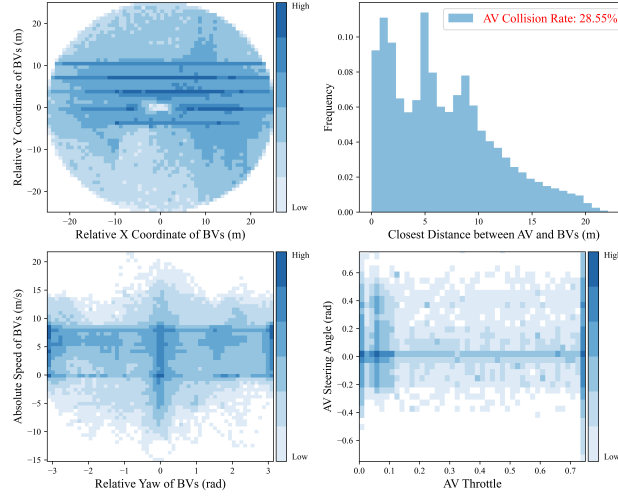


Figure 7: The offline data distribution.

465 **Constrain Function Setting.** As outlined in Eq. (9), the hyperparameters M and d_{th} , along with the
 466 minimum distance between vehicles' bounding boxes, are crucial for defining $h(s^{AV})$. Specifically,
 467 we employed the method described in [31] to calculate the minimum distance, which is always non-
 468 negative. To ensure a balance between positive and negative samples, we experimented with various
 469 settings for M and d_{th} .

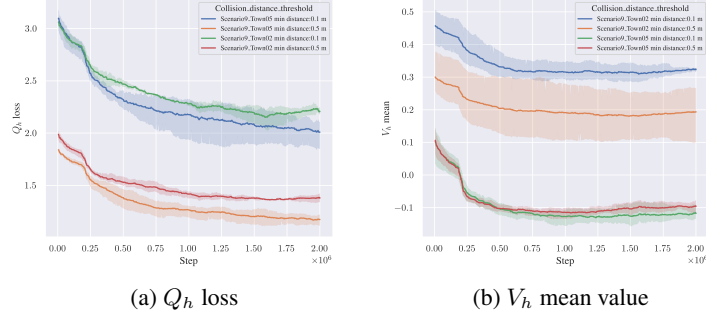


Figure 8: Learning curves in LFR training

470 In our experiment, we set d_{th} at 0.1 meters and 0.5 meters in separate trials to find the optimal
 471 parameter combination. Drawing on the findings from [15], we determined that optimal performance
 472 is achieved when the mean of the feasible value function is close to zero. Through empirical testing,
 473 we established $d_{th} = 0.1m$, $M = 18$ and $d_{th} = 0.5m$, $M = 12$ as the optimal settings. Given that
 474 these parameters are used to train across multiple towns, it is crucial to balance their optimality for
 475 various environments. Figure 8 illustrates the learning curves for these settings, confirming that the
 476 chosen parameters effectively maintain the mean values of the feasible value function in different
 477 towns close to zero.

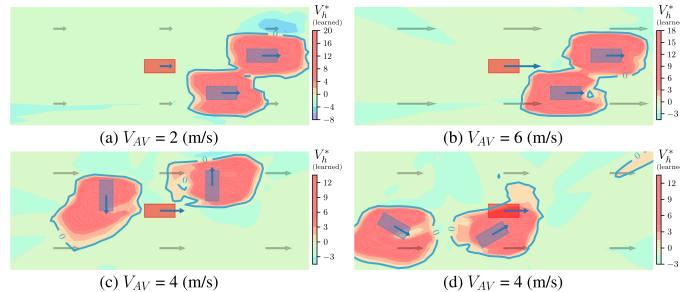


Figure 9: LFR visualization with 0.5m d_{th} under various traffic scenarios. Red: AV. Blue: BV

478 The visualization results for $d_{th} = 0.1m, M = 18$ are presented in Figure 2, while those for
479 $d_{th} = 0.5m, M = 12$ are shown in Figure 9. The former parameter set proved to be more effective
480 in identical scenarios, leading us to select $d_{th} = 0.1m, M = 18$ for subsequent CBV training.
481 Detailed hyperparameter settings for the feasible value function can be found in Table 4.

Table 4: Detailed hyperparameters of feasible value function training.

Parameter	Value
Optimizer	Adam ($\epsilon = 1e - 5$)
Approximation function	Multi-layer Perceptron
Number of hidden layers	2
Number of hidden units per layer	64
Nonlinearity of hidden layer	RELU
Nonlinearity of output layer	linear
learning rate	Linear annealing $3e - 4 \rightarrow 0$
Batch size	1024
Training steps	2e6
Discount factor γ	0.98
Expectile τ	0.9
Soft update	5e-3
d_{th}	0.1
M	18

482 C.2 Application Details of LFR

483 During the feasibility-guided adversarial policy training for CBV, we encountered a significant chal-
484 lenge: the optimal feasible value function produces a single scalar value at each timestep. In scenar-
485 ios with multiple CBVs, it is difficult to identify which CBV renders the AV’s operation unfeasible.
486 To address this issue, we introduced a “pseudo” state for each CBV at every timestep. This “pseudo”
487 state captures information relevant to both the AV and CBV from the perspective of the AV. Cru-
488 cially, this “pseudo” state is utilized solely to assess the threat posed by each CBV to the AV at
489 each timestep. By employing this method, we eliminate the need for the viewpoint transformation
490 function $g(\cdot)$, previously described in Eqs. (6) and (8). Instead, the “pseudo” state is directly derived
491 from the information available at each timestep, simplifying the process and enhancing the accuracy
492 of LFR.

493 D Detail Explanation about Feasibility Metric

494 In Section 4.3, we introduce four key metrics to evaluate the feasibility of AV along collision tra-
495 jectories induced by various CBV methods. Among these, the Infeasible Ratio (IR) and Infeasible
496 Distance (ID) are novel metrics proposed in this paper. Their definitions and implications are de-
497 tailed as follow:

498 **IR: Infeasible Ratio.** This metric quantifies the proportion of infeasible states for the AV along
499 the collision trajectory induced by a specific CBV method. As the CBV approaches the AV, the
500 likelihood of infeasibility naturally increases. The IR assesses the severity of collision risk by ex-
501 amining the percentage of these infeasible states, serving as a measure of the overall aggressiveness
502 of the CBV collision trajectory. Since these trajectories inevitably lead to collisions, they must in-
503 clude segments where infeasible states occur. Therefore, evaluating the relative performance of IR
504 is crucial.

505 **ID: Infeasible Distance.** This metric records the distance between the CBV and the AV when the AV
506 first enters the infeasible region along the collision trajectory. Since CBVs may initiate their attacks
507 from various starting points, the initial locations of collision trajectories can vary significantly. This
508 variability means the IR metric alone may not fully capture the inevitability of a collision. The
509 Infeasible Distance (ID) is therefore introduced to record the distance between the AV and CBV at

the first instance when the AV’s feasibility value exceeds zero. Intuitively, a larger ID indicates a more adversarial CBV approach, while a shorter ID suggests that the CBV allows the AV to remain feasible for longer, indicating a potentially avoidable collision.

E Evaluation Metric of the AV

To fairly evaluate the performance of AV methods across different adversarial scenarios, we adopt the evaluation metrics from SafeBench [25], focusing on two main categories: *Safety level* and *Functionality level*. It is important to note that while the *Etiquette level* is discussed in the SafeBench paper, it has not been implemented in the actual code. This exclusion is likely due to the prioritization of safety and functionality in safety-critical scenarios, where etiquette is less essential. To avoid confusion, our evaluations strictly follow the implementation details specified in the SafeBench code, rather than the descriptions in the paper.

Within the *Safety level* and *Functionality level*, we define several specific metrics that contribute to an *overall score*, which is defined as a weighted sum of all evaluation metrics.

Safety Level. This level evaluates the safety performance of AV methods using two primary metrics: *collision rate (CR)* and *average distance driven out of road (OR)*. We define τ as the scenario trajectory collected through interaction. The number of collisions in a scenario is represented by $c(\tau)$, and the distance driven out of the road is denoted as $d(\tau)$. Therefore, the metrics are calculated as follows: $CR = \mathbb{E}_\tau[c(\tau)]$, and $OR = \mathbb{E}_\tau[d(\tau)]$.

Functionality Level. This level measures the functional capabilities of AV agents in completing designated routes within testing scenarios. It employs three metrics: *route-following stability (RF)*, *average percentage of uncompleted route (UC)*, and *average time spent to complete the route (TS)*. *RF* quantifies the average distance between the AV and the reference route during testing, expressed as $RF = 1 - \mathbb{E}_\tau[\min\{\frac{x(\tau)}{x_{max}}, 1\}]$, where x_{max} represents the maximum allowable deviation. *UC* reflects the complement of the average completion percentage of the route, calculated as $UC = 1 - \mathbb{E}_\tau[p(\tau)]$, where $p(\tau)$ is the percentage of route completion of each testing scenario. *TS* is defined as the average time required to complete a route, computed only for fully completed routes: $TS = \mathbb{E}_\tau[t(\tau)|p(\tau) = 100\%]$, where $t(\tau)$ denotes the time cost of each testing scenario.

Overall Score. The overall quality of AV methods is quantified by an *overall score (OS)*, which aggregates the five metrics using a weighted sum formula: $OS = \sum_{i=1}^5 w^i \times g(m^i)$, where each m^i is a specific metric, w^i is its weight, and $g(m^i)$ adjusts the metric based on its desirability:

$$g(m^i) = \begin{cases} \frac{m^i}{m_{max}^i}, & m^i \text{ is the higher the better} \\ 1 - \frac{m^i}{m_{max}^i}, & m^i \text{ is the lower the better} \end{cases}, \quad (21)$$

where m_{max}^i is a constant representing the maximum allowed value for each metric m^i . Further details about w^i and m_{max}^i are provided in Table 5.

Table 5: Detailed parameters of evaluation metrics.

Metric	Weight w^i	Maximum allowed value m_{max}^i
<i>CR</i>	0.4	1
<i>OR</i>	0.1	10 (m)
<i>RF</i>	0.1	5 (m)
<i>UC</i>	0.3	1
<i>TS</i>	0.1	30 (s)

F Scenario Analysis

F.1 Successful Scenarios

To demonstrate that the *FREA* method can generate AV-feasible adversarial events across various traffic scenarios, we present additional visualizations from different towns and intersections, as

546 shown in Figure 10. Specifically, Figure 10(a) illustrates scenarios where the AV is preparing to
 547 make a right turn, while the CBV exhibits adversarial behavior by overtaking and executing an early
 548 right turn. Figure 10(b) displays situations where the CBV makes a U-turn from its lane, leading
 549 to a potential collision with the AV and creating adversarial scenarios. Figure 10(c) highlight the
 550 adversarial behavior of the CBV within an intersection, where CBV pre-empts the AV while passing
 551 through. These scenarios highlight the *FREA* method’s adaptability to different traffic infrastructures
 552 and its effectiveness in generating safety-critical scenarios with reasonable adversariality.

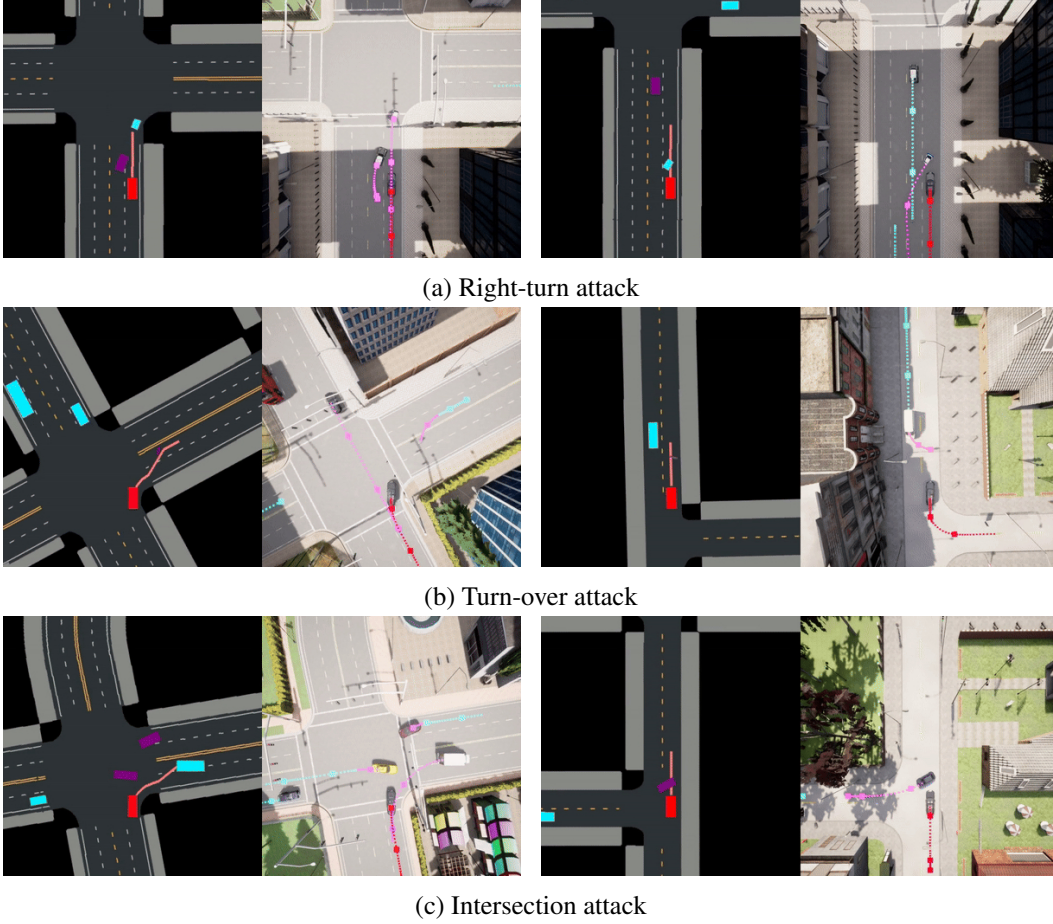
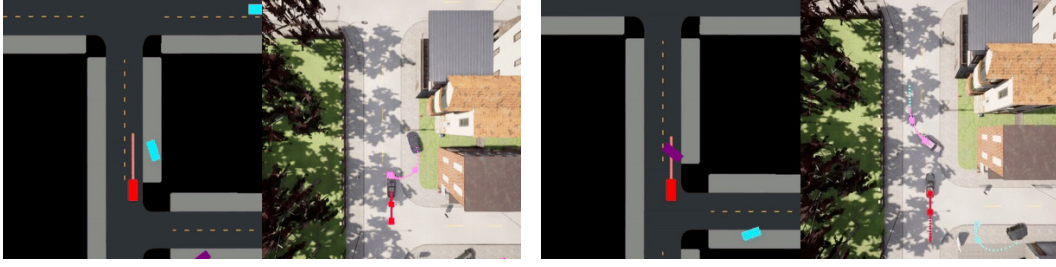


Figure 10: Successful scenarios. Red: AV (Expert). Blue: BV. Purple: CBV (*FREA*).

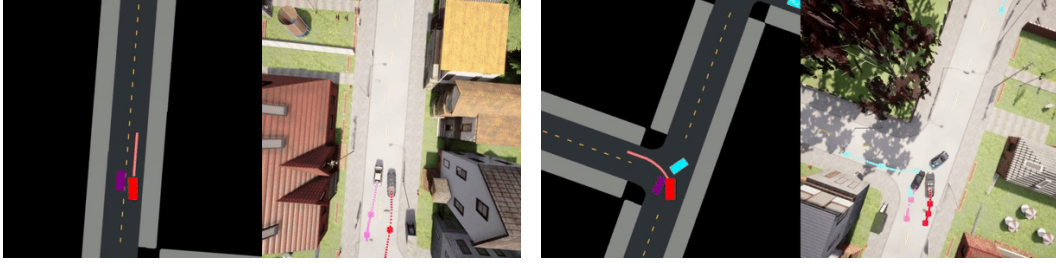
553 F.2 Failure Scenarios

554 While the *FREA* method makes progress in generating adversarial yet reasonable safety-critical
 555 scenarios, particularly on the reasonableness of adversarial attacks, it also has several limitations, as
 556 illustrated by the failure cases depicted in Figure 11.

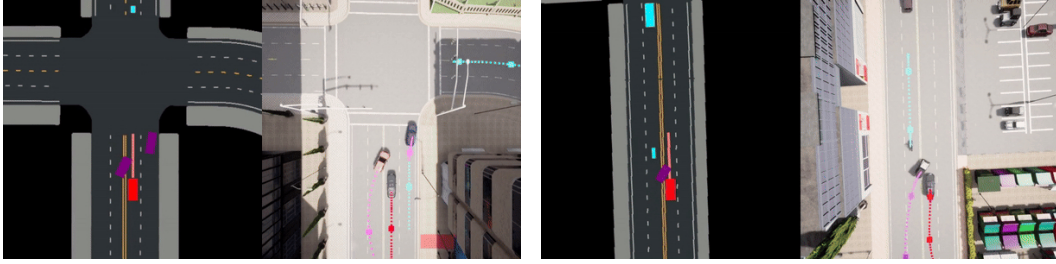
557 As illustrated in Figure 11(a), after completing its attack, the CBV is released as a normal vehicle
 558 on a narrow road. However, its turning radius exceeds the width of the road, forcing it onto the
 559 pavement and resulting in an unreasonable scenario. In Figure 11(b), after an unsuccessful initial
 560 attack due to the AV’s obstacle avoidance maneuvers, the CBV attempts a second attack through a
 561 reverse maneuver. Although technically feasible, this approach deviates from typical driving objec-
 562 tives, compromising the scenario’s reasonableness. Figure 11(c) shows a scenario where the CBV
 563 crosses a solid yellow line to initiate an attack. Since the state information in *FREA* lacks map de-
 564 tails, violations of traffic rules are foreseeable. Future studies will focus on enhancing adherence
 565 to traffic regulations. Finally, Figure 11(d) depicts a traffic jam scenario at intersections caused by
 566 unreasonable attacks.



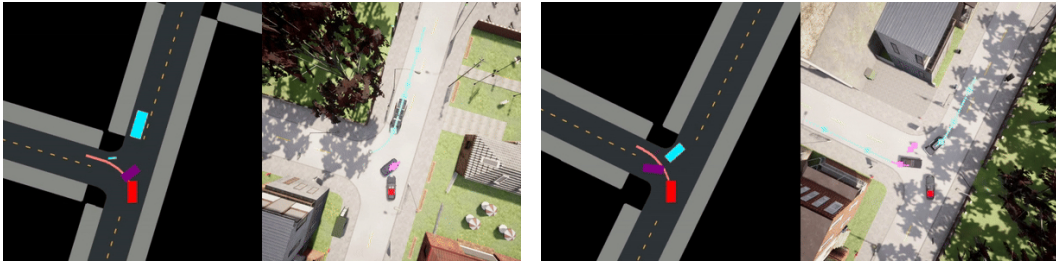
(a) Turning around on the pavement



(b) Reverse attack



(c) Traffic rules violation



(d) Traffic congestion

Figure 11: Failure scenarios: Red: AV (Expert). Blue: BV. Purple: CBV (FREA).

567 In conclusion, this paper acknowledges specific limitations in generating adversarial yet reasonable
 568 safety-critical scenarios. These limitations primarily include the lack of traffic regulation compliance
 569 and the challenge of aligning CBV behaviors during attacks with their driving objectives. Address-
 570 ing these issues is crucial for developing more reasonable adversarial scenarios.