# Extreme Scenario Generation Based on Adversarial Attack

Haoxin Ma
Department of Automation, Tsinghua University, Beijing, China
mahx21@mails.tsinghua.edu.cn

Jianming Hu*
Department of Automation, Tsinghua University, Beijing, China
hujm@mail.tsinghua.edu.cn

## ABSTRACT

The transportation field requires a large number of simulation scenarios for testing. At present, there is relatively little research on the generation of extreme scenarios. In this paper, we give the definition of extreme scenarios, which are prone to problems, and divide them into two categories: the extreme scenarios based on primitive value and the extreme scenarios based on primitive coupling. This paper focuses on the second which considers the coupling effect of different primitives in the scenarios, using the methods of adversarial attack: FGSM, FGSM-target, BIM, ILCM, PGD and strategically-timed attack. Using vehicle agent for test, the first five methods prove the feasibility and effectiveness of extreme scenario generation, and the sixth method simplifies the generation process.

## CCS CONCEPTS

• **Computing methodologies**; • **Artificial intelligence**; • **Distributed artificial intelligence**; • **Intelligent agents**;

## KEYWORDS

Extreme scenario generation, Primitive coupling, Vehicle agent, Adversarial attack

**ACM Reference Format:**
Haoxin Ma and Jianming Hu*. 2021. Extreme Scenario Generation Based on Adversarial Attack. In *The 5th International Conference on Computer Science and Application Engineering (CSAE 2021), October 19–21, 2021, Sanya, China.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3487075.3487186

## 1 INTRODUCTION

In the field of intelligent transportation, no matter what direction the research is, there is basically a major premise: the setting of traffic scenarios is required. For example, vehicle driving strategy research, vehicle pedestrian target detection, signal light control, etc., all need to be established on the basis of a certain traffic scenario, and the scenarios can be used for various tests. Therefore, the construction and generation of traffic scenarios are indispensable. There are many test scenarios in the field of automatic driving. The automatic driving car model is tested in real scenarios to find its problems. It can better show the car's automatic driving ability in various real scenarios, but it requires a large number of mileages as data support to produce more credible results [1]. For example, as mentioned in literature [2, 3], in order to ensure the safety of autonomous vehicles, at least 240 million kilometers need to be tested without any traffic accidents. In addition, the use of real scenario testing has certain limitations and risks of endangering safety. If an accident occurs, it will cause considerable losses and even endanger human lives. With the continuous improvement of requirements for automatic driving, the construction of test scenarios requires more consideration of the complexity of the car system, weather conditions, driving strategies, driving conditions and other factors, undoubtedly increasing the difficulty of building test scenarios [4-7], but the goal is to further ensure the authenticity and completeness of the test scenarios.

On the whole, real-scenario testing has been difficult to meet test requirements. Virtual simulation scenarios can be configured according to the needs of testers, and are easy to repeat and reproduce. They can also ensure the safety of the testing process and reduce the cost of testing. Therefore, many studies now give up testing in real scenarios and choose to test in simulation environments and virtual scenarios [8].

At present, there is no authoritative definition of extreme scenarios in the traffic test scenarios, so we provide a definition and explanation. Extremes refer to problems that are prone to occur in the scenario, such as collisions, etc., and extreme scenarios mean that the primitives in the scenario are highly likely to conflict, eventually leading to a problem. The conflict here can be explained from two directions.

The first is based on the numerical extreme of scenario primitives. For example, if the speed of a vehicle in the scenario is too fast relative to other vehicles, then the speed of the car belongs to the numerical extreme of primitives. If the speed is too fast, it is easy to cause vehicle collisions.

The second is based on the coupling extreme of scenario primitives, on which this paper focuses. Unlike the first type, the primitive values in this scenario seem to be very reasonable, but such scenarios are also prone to cause problems. Instead of the numerical analysis of the primitive itself, we consider the coupling relationship between the primitives. We believe that there is a conflicting coupling relationship between the primitives in this type of scenario. For example, in the vehicle collision scenario, speed and position of two vehicles are reasonable before collision, but in the end the collision still occurs. The problem is a certain correlation between two vehicles, namely the coupling relationship mentioned above.

## 2 PROCESS DESIGN

This paper takes the vehicle agent as an example to study the generation of two types of extreme scenarios, and uses the agent's reward as an evaluation index for the extreme situations of the generated extreme scenarios.


*CSAE 2021, October 19–21, 2021, Sanya, China*

ACM ISBN 978-1-4503-8985-3/21/10…$15.00
https://doi.org/10.1145/3487075.3487186

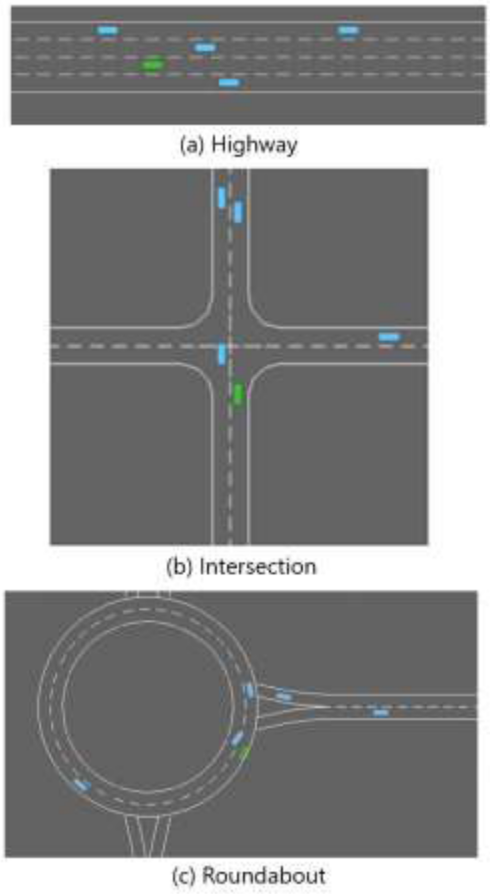


**Figure 1: Three Typical Scenarios in Highway-env.**

## 2.1 Simulation Platform

This paper uses the highway-env[1] simulation platform for process design and algorithm implementation. Figure 1 shows the three typical scenarios provided by the platform-highway, intersection, and roundabout. The green vehicle is the vehicle agent we control. In these three scenarios, the self-vehicle agent can perceive the state of 15 surrounding vehicles at most. The parameters of three scenarios are shown in Table 1. IDM in Table 1 is a model that comes with the simulation platform, which makes the vehicle accelerate and steer on the road according to relatively simple rules.

In the highway scenario, the self-vehicle agent drives on four lanes. In terms of state, the self-vehicle agent can obtain 7 state variables of 15 nearby vehicles- the presence or absence, lateral position difference, longitudinal position difference, lateral speed difference, longitudinal speed difference, vehicle steering cosine and sine value. In terms of action, the self-vehicle agent has five action options-keep the lane, turn left, turn right, accelerate, and decelerate. In terms of reward, the goal of the self-vehicle agent is to achieve high speeds, avoid collisions with neighboring vehicles, and try to keep driving on the right side of the road, so the rewards in this environment can be defined as shown in Formula (1), whose functions are to punish the self-vehicle agent's collision behavior, reward the self- vehicle agent to drive in the right lane, and encourage the self-vehicle agent to drive at a higher speed.

$$\begin{aligned} R_{collision} &= -1 \\ R_{right} &= 0.1 \\ R_{speed} &= 0.4 \times \frac{v - v_{min}}{v_{max} - v_{min}} \end{aligned} \quad \#(1)$$

The intersection is a two-lane design, with the ultimate goal of turning left through the intersection without collision. The observation of the state is similar to the acquisition of the state value of the highway. In terms of action, the environment is set to a two-way single lane so that no lane change action needs to be set. The final goal is to turn left through the intersection, so the overall action only needs to consider speed changes. Therefore, there are three choices for the actions of the self-vehicle agent in the intersection environment: decelerate, remain unchanged, and accelerate. In terms of reward, the goal of the self-vehicle agent is to turn left as quickly as possible to reach the destination through the intersection within a certain period of time without collision. Based on this, the reward in this environment can be defined as shown in Formula (2), whose functions are rewarding penalties for collision behaviors, encouraging self-vehicle agent to drive at high speed, and encouraging self-vehicle agent to reach the target location within the simulation time.

$$\begin{aligned} R_{collision} &= -5 \\ R_{speed} &= \frac{v - v_{min}}{v_{max} - v_{min}} \\ R_{arrived} &= 1 \end{aligned} \quad \#(2)$$

The roundabout is also a two-lane design, with the ultimate goal of passing the roundabout without collision. In terms of state observation, 4 state variables can be obtained for the lateral position

[1] https://github.com/eleurent/highway-env

**Table 1: Parameters of Three Scenarios**

| | Highway | Intersection | Roundabout |
|---|---|---|---|
| Simulation time (s) | 40 | 13 | 11 |
| Number of lanes | 4 | 2 | 2 |
| Total number of vehicles | 15 | 15 | 15 |
| Model of other vehicles | IDM | IDM | IDM |
| Speed of vehicles (m/s) | 0~40 | 0~20 | 0~20 |
| Acceleration of vehicles ($m/s^2$) | 4 | 3 | 3 |

difference, longitudinal position difference, lateral speed difference, and longitudinal speed difference of 15 vehicles near the self-vehicle. In terms of action, similar to the highway environment, the self-vehicle agent also has five actions to choose from: keeping the lane, turning left, turning right, accelerating, and decelerating. In terms of reward, the ultimate goal of the self-vehicle agent is to quickly pass through the roundabout without collision and change lanes as little as possible. Based on this, the reward in this environment can be defined as shown in Formula (3), whose functions are to punish the collision behavior, encourage the self-vehicle agent to drive at high speed, and punish the agent's lane changing behavior.

$$\begin{aligned} R_{collision} &= -1 \\ R_{speed} &= 0.2 \times \frac{v - v_{min}}{v_{max} - v_{min}} \\ R_{change} &= -0.05 \end{aligned} \quad \#(3)$$

## 2.2 Coupling Extreme

To generate extreme scenarios based on primitive coupling, it is necessary to find a kind of data that can reflect the coupling relationship of primitives without affecting the rationality of primitive values, and use this type of data to complete the generation of such extreme scenarios.

We use the idea of adversarial attack. Adversarial attack on original scenarios can produce corresponding primitive disturbances. The primitive disturbances are added to the corresponding primitives. Since the ratio of primitive disturbances to the value of the primitive itself is small, from the perspective of primitive values, the scenario after adding disturbance is almost the same as the original normal scenario, but when we test on the scenario after adding disturbance, if the reward the agent obtains is significantly smaller than the reward obtained in the original ordinary scenario, it means that the disturbance has an impact on the scenario, causing the scenario to have an extreme situation. This effect is not numerical, but for the coupling relationship between primitives. Therefore, obtaining the primitive disturbances and adding them to the original scenarios, we can get extreme scenarios based on primitive coupling. Figure 2 shows the specific generation process of such extreme scenarios.

As shown in Figure 2, we firstly determine the original scenarios-highway, intersection and roundabout for experiments. Then for these three scenarios, try a variety of training algorithms to obtain the agent model. Random strategy means that the agent randomly chooses actions at every step, which gives the lower limit of the model. The planning algorithm uses the Monte Carlo tree search algorithm [9], which gives the upper limit of the model. Based on the premise of knowing the global characteristics, it searches for the optimal result, which is more approximate to the optimal strategy, but actually, the planning algorithm cannot be used to make decisions, because the scenario is not completely knowable. What we consider is the agent trained by the reinforcement learning algorithm. The four methods of DQN [10], Double-DQN [11], Dueling-DQN [12] and Attention-DQN [13] are used for training. Random strategy and planning algorithm are used as the baselines, helping choose the best agent to enter the next step. After training the best agent model, we use the adversarial attack algorithm to generate disturbances, specifically using the six algorithms of FGSM [14], FGSM-target [15], BIM [16], ILCM [16], PGD [17] and strategically-timed attack [18]. The content of these six algorithms will be introduced in detail

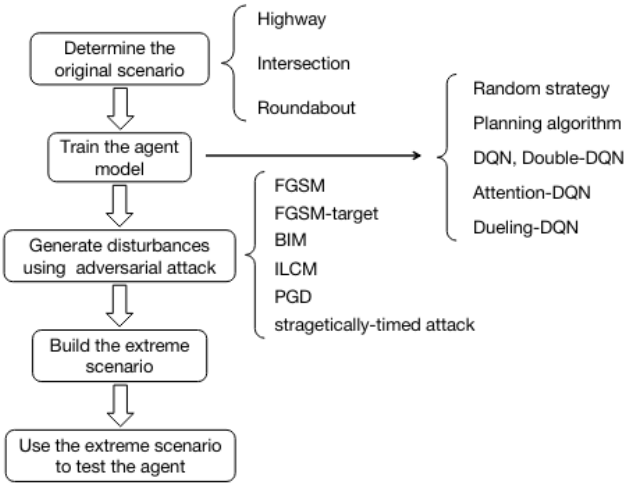


**Figure 2: Generation Process of Extreme Scenarios Based on Primitive Coupling.**

in Section 3. Then we add the disturbances to the original scenario to obtain the extreme scenario, and finally test the agent in the extreme scenario to evaluate the extreme situation.

# 3 ALGORITHM IMPLEMENTATION

After training the agent model with the reinforcement learning algorithm in the original scenario, we use the adversarial attack algorithm to generate disturbances, specifically using the six algorithms of FGSM, FGSM-target, BIM, ILCM, PGD and strategically-timed attack. These algorithms attacked image classification when they were proposed, in order to make the trained model classify incorrectly, but they cannot be directly used on the agent trained by the reinforcement learning algorithm and need to be adjusted.

## 3.1 FGSM

FGSM uses Formula (4) to generate disturbances, and only uses the direction of the gradient after calculating the gradient. $x$ is the input, and it is pixel value in the image field, but primitive state value in the extreme scenarios based on primitive coupling. $y$ is the output, and it is a specific correct category in the image field, but the correct action selected in the study of extreme scenario generation based on primitive coupling. $J$ is the loss function of the output to the input in the neural network, and $\varepsilon$ is the set disturbance ratio, which can limit the generated disturbance range and should not be set too large.

$$X' = x + \varepsilon * sign\left(\nabla_x J\left(x, y\right)\right) \#(4)$$

The output of FGSM uses probability, but the DQN algorithm we use to train the agent model directly outputs the value corresponding to the action, so we cannot directly use FGSM. Instead, we consider using the Softmax function to convert the value into probability, and use Formula (5) to determine $J$, which represents the cross entropy between the output probability of the strategy network and the

point distribution of the strategy with the highest probability.

$$\begin{aligned} P_{target} &= onehot\left(argmax\left(\pi\left(s\right)\right)\right) \\ P_{disturb} &= \pi\left(s+\Delta s\right) \\ J &= -\sum_{i=1}^{n} P_{target}\log\left(P_{disturb}\right) \end{aligned} \quad \#(5)$$

### 3.2 FGSM-Target

FGSM-target is a variant of the FGSM algorithm. The gradient direction used by the FGSM algorithm to calculate the disturbance is obtained based on the correct decision. The gradient obtained by the correct decision is added to the original state as an incremental disturbance, which can make the agent less likely to make the correct decision. In contrast to the FGSM-target algorithm, the gradient direction it uses is based on the decision with the lowest probability value, that is, the most unlikely decision. The disturbance obtained by multiplying this gradient by the ratio is the decrement disturbance of the original state, which can make the agent go in the wrong direction of decision-making. The specific expression is shown in Formula (6). $y_{t\arg et}$ is the most unlikely decision.

$$X^{'} = x - \varepsilon * sign\left(\nabla_x J\left(x, y_{target}\right)\right) \#(6)$$

### 3.3 BIM

BIM uses an iterative method to find the disturbance of each input value. The iterative method means that each input increases or decreases $\alpha$ based on the disturbance of the previous step, and then performs cropping to ensure that the new input value is in the $\varepsilon$ neighborhood of the original $x$ Compared with FGSM, this method may find smaller disturbances, and the worst is the same as FGSM. The specific expression is shown in Formula (7).

$$\begin{aligned} X_0^{adv} &= X \\ X_{N+1}^{adv} &= Clip_{X,\varepsilon}\left\{X_{N+1}^{adv} + \alpha sign\left(\nabla_X J\left(X_N^{adv}, y_{true}\right)\right)\right\} \end{aligned} \#(7)$$

### 3.4 ILCM

Analogous to the changes of FGSM-target, ILCM algorithm adjust the BIM algorithm, replacing the original output with the most unlikely output. The specific expression is shown in Formula (8).

$$\begin{aligned} X_0^{adv} &= X \\ X_{N+1}^{adv} &= Clip_{X,\varepsilon}\left\{X_{N+1}^{adv} - \alpha sign\left(\nabla_X J\left(X_N^{adv}, y_{target}\right)\right)\right\} \end{aligned} \#(8)$$

### 3.5 PGD

PGD is similar to BIM. The difference is that BIM initializes disturbances to 0, while PGD initializes disturbances to randomly select values between $(-\varepsilon, \varepsilon)$ Both of them need to go through multiple iterations to find the most suitable gradient to generate the final disturbance. The specific expression is shown in Formula (9). $g_t$ represents the loss $L$ in relation to the gradient of the input $X$ at $t$ moment. $\frac{g_t}{g_t}$ provides the direction of the gradient and $\varepsilon$ is the set disturbance ratio. $\prod_{X+S}$ limits the range of disturbance. Since only a small step is taken each time, the local linear assumption is basically established. After many steps, the optimal solution can be reached.

$$\begin{aligned} g_t &= \nabla_{X_t}\left(L\left(f_\theta\left(X_t\right), y\right)\right) \\ X_{t+1} &= \prod_{X+S}\left(X_t + \varepsilon\left(\frac{g_t}{g_t}\right)\right) \end{aligned} \#(9)$$

### 3.6 Strategically-timed Attack

The idea of strategically-timed attack is to add disturbance to the observed state at the right time, so as to reduce the reward. If taking a certain action in a state can significantly increase the reward, it means that the agent is inclined to choose this action, and perturb this state to make the agent not to take the action. The advantage of this method is that it does not need to add disturbance to the observed state at each moment, and achieves better results under the premise of less time. The specific expression is shown in Formula (10). $b_t$ represents whether to add disturbance. We use a threshold-based method to select the moment, and set a threshold for the difference between the maximum and minimum Q values in the DQN network. When the difference exceeds the threshold, this moment is selected to add disturbance, that is, the selected moment needs to meet Formula (11). $t$ is the moment, $a$ is the action, and $\gamma$ is the set threshold.

$$\begin{aligned} \min_{b_1, b_2, \ldots, b_L, \delta_1, \delta_2, \ldots, \delta_L} \quad & R\left(\overline{s_1}, \ldots, \overline{s_L}\right) \\ s.t. \quad & \overline{s_t} = s_t + b_t\delta_t \quad for\ all\ t = 1, \ldots, L \\ & b_t \in \{0,\ 1\} \quad for\ all\ t = 1, \ldots, L \\ & \sum_t b_t \le \Gamma \end{aligned} \#(10)$$

$$\max_a Q(s_t, a) - \min_a Q(s_t, a) > \gamma \#(11)$$

## 4 EXPERIMENTAL RESULTS

For the extreme scenario generation based on primitive coupling, it is determined to use highways, intersections and roundabouts as the original scenarios. Then for these three scenarios, try six training algorithms to obtain the agent model. Random strategy and planning algorithm are used as the baselines. DQN, Double-DQN, Dueling-DQN and Attention-DQN are used for finding the best agent model. Table 2 shows the test results of the trained agent using these methods. The numbers in the table represent the reward of the trained agent model tested in the corresponding scenario.

According to Table 2, by comparison, in the first two scenarios, Attention-DQN shows a better model training effect, while in the third scenario, Dueling-DQN shows a better model training effect. Therefore, in the next steps, we use the model trained by Attention-DQN for highway and intersection and the model trained by Dueling-DQN for roundabout.

After training the best agent model, FGSM, FGSM-target, BIM, ILCM and PGD are used to generate disturbances, and then the disturbances are added to the original scenarios to obtain extreme scenarios. Finally, the trained agent model was tested in extreme scenarios, and the results are shown in Table 3. We hope that the addition of disturbance does not affect the numerical presentation of the original state, so the disturbance ratio $\varepsilon$ is set to be small.

Compared with the reward obtained in the original scenarios, the more the reward in the extreme scenarios is reduced, the better the extreme situation of the extreme scenarios. It can be seen from the table that the extreme situations for the extreme scenarios of highways and intersections are better. BIM is the best for highways, and FGSM-target is the best for intersections. In addition, choosing $\varepsilon$ that is smaller than random attack, the extreme situation obtained in these two scenarios is better than that of random attack, apart from PGD algorithm, indicating that these algorithms can more

**Table 2: Test Results of Trained Agent Using Six Methods**

| Scenario | Random strategy | Planning algorithm | DQN | Double-DQN | Dueling-DQN | Attention-DQN |
|---|---|---|---|---|---|---|
| Highway | 13.42 | 35.25 | 29.31 | 29.98 | 30.55 | **31.08** |
| Intersection | 2.58 | 10.78 | 7.95 | 8.13 | 8.86 | **9.03** |
| Roundabout | 4.65 | 11.0 | 9.86 | 10.02 | **10.35** | 10.27 |

**Table 3: Test Results of Extreme Scenarios Generated by Adversarial Attack Algorithms**

| Attack method | $\varepsilon$ | Reward | | |
|---|---|---|---|---|
| | | Highway | Intersection | Roundabout |
| No attack | | 33.08 | 9.03 | 10.35 |
| Random attack | 0.02 | 28.68 | 5.56 | 9.65 |
| FGSM | 0.01 | 27.69 | 4.93 | **9.12** |
| FGSM-target | 0.01 | 27.33 | **4.75** | 9.63 |
| BIM | 0.01 | **26.29** | 5.36 | 9.78 |
| ILCM | 0.01 | 28.25 | 5.25 | 9.76 |
| PGD | 0.01 | 30.05 | 5.86 | 9.62 |

easily generate extreme scenarios with better extreme effects. However, the extreme scenarios corresponding to the roundabouts also have certain extreme effects, but the extreme situation is relatively poor. In general, these experimental results prove the validity of these algorithms. Under the same perturbation ratio and attack algorithm settings, the test rewards of the agent trained in the early stage in the extreme scenarios corresponding to highways and intersections are reduced more. This is because the algorithm used to train the agent in these two scenarios is the DQN algorithm introducing the attention mechanism. Compared to Dueling-DQN, it is more sensitive to the input state, and the addition of disturbances will more easily affect it. Moreover, for the extreme scenarios of roundabouts, FGSM perform the best, and other algorithms have similar effects to random attack. From this perspective, under the premise of these three scenarios and two training algorithms, the applicability of FGSM is more extensive than other algorithms.

Figure 3 shows the influence on the extreme situation of the generated extreme scenario when the $\varepsilon$ value changes. In each scenario, the $\varepsilon$ value is modified from 0.002 to 0.05 with a step length of 0.002. It can be seen from the figure that the extreme scenarios generated by adding the disturbances for the three original scenarios have certain extreme effects. In addition, as the $\varepsilon$ value increases, the extreme situations of extreme scenarios are getting better and better in the overall trend.

In the extreme scenarios corresponding to highway, the overall downward trend of these five algorithms is similar, and it is not praised to judge which method is the best. Relatively speaking, FGSM-target exhibits a better extreme situation. In the extreme scenarios corresponding to intersection, it can be seen from Figure 3(b) that when $\varepsilon$ takes 0.02 and less, the extreme situations brought by these methods are similar, and when $\varepsilon$ is greater than 0.02, the effect of FGSM-target is the best. In the extreme scenarios corresponding to roundabout, the downward trend of FGSM and ILCM is obvious, but the other three algorithms have general effects, and even the curve of BIM fluctuates sharply. The extreme situation is the best when $\varepsilon$ takes 0.03. My understanding for this point is that the perturbation direction generated by BIM is not suitable for the agent in this scenario, which brings uncertainty and causes obvious fluctuations.

The final results of using strategically-timed attack to select some moments to add disturbances for experiments are shown in Table 4. It shows the changes of the extreme situation and the number of selected moments under different $\varepsilon$ values of the two attack algorithms. Because in the previous experiments, the extreme situations of extreme scenarios generated by FGSM and FGSM-target are better overall, this part uses these two algorithms to verify the feasibility of strategically-timed attack. $\gamma$ represents the set threshold. Through multiple tests, we find $\gamma$ that satisfies the two conditions that only a part of the moment needs to be selected to add disturbances and the extreme situation does not weaken more. In Table 4, the number before the arrow represents the reward for adding disturbance at all moments, and the number after the arrow represents the reward for adding disturbance at some selected moments. The numbers in parentheses indicate the number of selected moments in all moments. Judging from the overall results, although the strategically-timed attack selects part of the moment, the extreme situation is not greatly weakened. It can be seen that in the attack on the agent model, there are indeed some moments when the attack is invalid, which means that the generation of extreme scenarios can be simplified.

Specifically, for highway, under the same threshold and disturbance ratio settings, the extreme scenarios generated by FGSM-target are weakened less, and at the same time, it chooses to add disturbances less time. However, for intersection, the comparison results between FGSM-target and FGSM are different. From the perspective of the degree of weakening of extreme situation, FGSM is better, but the number of moments when disturbance is added is still less using FGSM-target. In the roundabout scenario, the situation

**Table 4: Experimental Results of Strategically-timed Attack**

| Attack method | $\varepsilon$ | Reward | | |
|---|---|---|---|---|
| | | Highway($\gamma$=0.2) | Intersection($\gamma$=0.2) | Roundabout($\gamma$=0.04) |
| No attack | | 33.08 | 9.03 | 10.35 |
| Random attack | 0.02 | 28.68 | 5.56 | 9.65 |
| FGSM | 0.02 | 22.43→25.97(29/40) | 4.02→4.15(6/13) | 9.36→10.13(5/11) |
| FGSM | 0.03 | 14.68→19.71(24/40) | 3.16→3.26(5/13) | 9.04→9.74(4/11) |
| FGSM-target | 0.02 | 21.36→22.89(28/40) | 3.01→3.18(6/13) | 9.73→10.06(6/11) |
| FGSM-target | 0.03 | 14.98→16.24(20/40) | 2.66→2.94(4/13) | 9.34→9.72(5/11) |

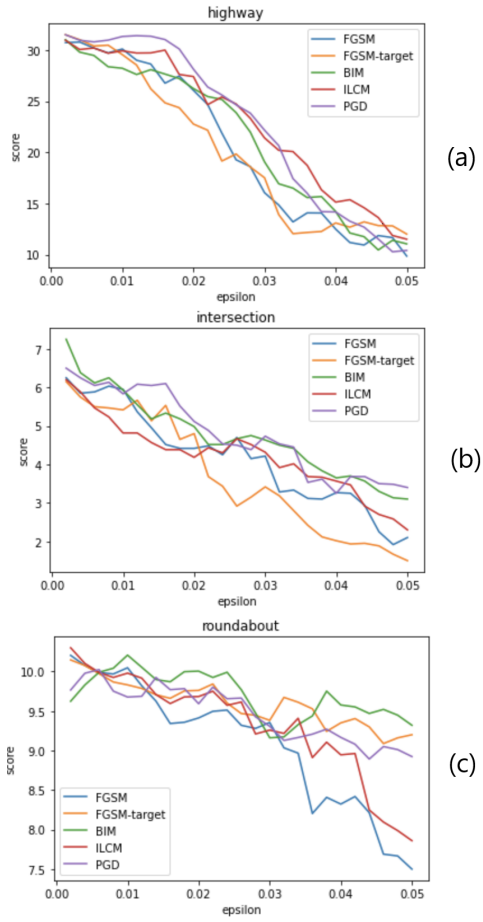


**Figure 3: Extreme Situation with Changes in $\varepsilon$.**

changes again. From the perspective of the degree of weakening of the extreme situation, FGSM-target is better, but the number of moments when disturbance is added is less using FGSM. Therefore, in general, the method of strategically-timed attack is feasible. As to which method to choose to generate disturbance and the number of moments to choose to add disturbance, experiments need to be performed to observe.

## 5 CONCLUSIONS

The research goal of this paper is to construct and generate extreme scenarios in traffic scenarios. First, define the traffic extreme scenarios: the primitives in the scenario are likely to conflict and eventually lead to problems. Under this definition, the extreme scenarios are divided into two categories: one is based on the primitive coupling and the other is based on primitive value. This paper focuses on the former, that is, how to generate the extreme scenarios based on primitive coupling, and propose a new idea of using adversarial attack methods to generate extreme scenarios. Using vehicle agent for testing, we adopt six adjusted algorithms of adversarial attack as the solutions to generating extreme scenarios. Experiments on FGSM, FGSM-target, BIM, ILCM and PGD prove the effectiveness of these algorithms of generating extreme scenarios, and experiments on strategically-timed attack prove that the process of generating extreme scenarios can be further simplified.

## REFERENCES

[1] Kalra N, Paddock S M (2016). Driving to safety: How many miles of driving would it take to demonstrate autonomous vehicle reliability? Transportation Research Part A: Policy and Practice, 94, 182-193.
[2] Christensen A, Cunningham A, Engelman J, *et al* (2015). Key considerations in the development of driving automation systems. 24th enhanced safety vehicles conference. Gothenburg, Sweden.
[3] Benmimoun M (2017). Effective evaluation of automated driving systems. SAE Technical Paper.
[4] Urmson C, Anhalt J, Bagnell D, *et al* (2008). Autonomous driving in urban environments: Boss and the urban challenge. Journal of Field Robotics, 25(8), 425-466.
[5] Saust F, Wille J M, Lichte B, *et al* (2011). Autonomous vehicle guidance on braunschweig's inner ring road within the stadtpilot project. 2011 IEEE Intelligent Vehicles Symposium (IV). IEEE, 169-174.
[6] Ardelt M, Coester C, Kaempchen N (2012). Highly automated driving on freeways in real traffic using a probabilistic framework. IEEE Transactions on Intelligent Transportation Systems, 13(4), 1576-1585.
[7] Anderson J M, Nidhi K, Stanley K D, *et al* (2014). Autonomous Vehicle Technology: A Guide for Policymakers[M]. New York: Rand Corporation.
[8] Masuda S (2017). Software testing design techniques used in automated vehicle simulations. 2017 IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW). IEEE, 300-303.
[9] Browne C B, Powley E, Whitehouse D, *et al* (2012). A survey of monte carlo tree search methods. IEEE Transactions on Computational Intelligence and AI in games, 4(1), 1-43.
[10] Mnih V, Kavukcuoglu K, Silver D, *et al* (2015). Human-level control through deep reinforcement learning. nature, 518(7540), 529-533.
[11] Van Hasselt H, Guez A, Silver D (2016). Deep reinforcement learning with double q-learning. Proceedings of the AAAI Conference on Artificial Intelligence: volume 30.
[12] Wang Z, Schaul T, Hessel M, *et al* (2016). Dueling network architectures for deep reinforcement learning. International conference on machine learning. PMLR, 1995-2003.
[13] Sorokin I, Seleznev A, Pavlov M, *et al* (2015). Deep attention recurrent Q-network. arXiv preprint arXiv:1512.01693.
[14] Goodfellow I J, Shlens J, Szegedy C (2014). Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572.

[15] Kurakin A, Goodfellow I, Bengio S (2016). Adversarial machine learning at scale. arXiv preprint arXiv:1611.01236.
[16] Kurakin A, Goodfellow I J, Bengio S (2016). Adversarial examples in the physical world. arXiv preprint arXiv:1607.02533.
[17] Madry A, Makelov A, Schmidt L, *et al* (2017). Towards deep learning models resistant to adversarial attacks. arXiv preprint arXiv:1706.06083.
[18] Lin Y C, Hong Z W, Liao Y H, *et al* (2017). Tactics of adversarial attack on deep reinforcement learning agents. arXiv preprint arXiv:1703.06748.