Safe Policy Improvement via Diffusion Model for Offline Reinforcement Learning

Xiaohan Yang[1], Jun Li[1], Jiang Liu[2], Mengting Sun[1], Baozhu Chen[1]*

[1] School of Computer and Information, Anhui Polytechnic University, China
[2] Anhui CAS Terahertz Technology Co., Ltd, China
yangxh2233@163.com, edmondlee@ahpu.edu.cn, lj15683843420@gmail.com,
clownfish384@163.com, 1848795229@qq.com

Abstract

Offline reinforcement learning is a paradigm that learns policies without real-time interaction with the environment. Using offline datasets avoids the potential training dangers and overhead of real-time interaction sampling. However, one key challenge of offline reinforcement learning is that there is no guarantee the learned policy is safe because the high-reward actions in the offline dataset do not satisfy safe constraints. In this paper, we propose a novel algorithm that improves policy safety for offline reinforcement learning named SPI. The core idea of SPI is using diffusion model to expand the original action space by guiding the generation of high reward actions satisfying safety constraints in similar data distributions. First, we remap the offline dataset to enhance the cost property. Second, we integrate the diffusion model, action value function, and safety value function into a framework that promotes the action to converge towards a distribution that is compliant with safety constraints. Extensive experiments under the DSRL benchmark demonstrate that SPI outperforms the relevant baselines in most tasks.

Keywords: Offline reinforcement learning, Safe policy, Diffusion model, Learning from experience

1 Introduction

Offline reinforcement learning (Offline RL) is a paradigm that learns policies using offline dataset without interacting with the environment, and its optimization goal is to gain an policy that maximizes the expected cumulative reward [1-2]. Thanks to the shift from real-time interactive online sampling to multi-batch dataset offline sampling, training agents to learn policies has become more cost-effective and efficient. This shift has driven extensive research into Offline RL in some research areas where sampling is costly and time-consuming, such as: autonomous driving [3-4], recommendation systems [5-6], competitive games [7-8], and robot control [9-11]. However, when using offline dataset for policy learning, Offline RL agents trained by maximizing the expected cumulative rewards do not take safe constraints into account in decision making. For instance, in navigation tasks, a robot dog, in order to gain high rewards that are negatively correlated with duration, would not take any evasive actions against obstacles on the moving path to rush straight ahead as quickly as possible. When the robot dog is deployed, it will cause some risks due to the neglect of safety, which cannot be accepted in real word applications.

The safe offline reinforcement learning (Safe Offline RL) is a good solution, which integrates safety constraints into the optimization objectives of Offline RL. By employing constrained Markov decision processes (CMDPS), Safe Offline RL can ensure that both reward and safety attributes are taken into account when evaluating policies. However, the datasets used may come from multiple behavioral policies that may have different goal tasks [12], and their complex composition make it difficult to ensure that high reward actions can satisfy the safety constraints. Previous works concentrated on how to find the optimal policy while satisfying the safety constraints. However, existing works face certain difficulties. (1) One approach is to impose constraints directly on the policy itself [13-15]. For example, introducing Lagrange multipliers into existing Offline RL algorithms to impose penalties for safety constraint violations; this transforms the original constrained minimization problem into an unconstrained optimization problem, penalizing state-action pairs that do not meet the constraints. However, when dealing with high dimensional complex data, the Lagrange multiplier method can easily converge to local optimal. (2) Another approach involves applying regularization to the value function to make its safety assessments more conservative [16-17]. By modifying the Bellman update equation, a pessimistic approximation of the lower bound is established, which helps reduce the impact of overestimation. However, the trade-off between conservatism and activism poses a complexity within the algorithm. Policies that are excessively conservative or overly aggressive may fail to identify state-action pairs that offer low costs and high returns.

To address the aforementioned issue, we propose the **s**afe **p**olicy **i**mprovement via diffusion model for offline reinforcement learning (**SPI**), as shown in Figure 1. Inspired by the diffusion Q-learning algorithm [18-19], we propose to use diffusion concepts to expand the

*Corresponding Author: Jun Li; Email: edmondlee@ahpu.edu.cn

DOI: https://doi.org/10.70003/160792642025122607004

original offline dataset with a focus on cost, ensuring that SPI converges to a safe, high-reward policy through three processes. Specifically, we first remap the cost features of the offline dataset to avoid the problems associated with sparse data. Next, we apply the diffusion model to denoise and fit the processed data distribution and learn action value function and the safe value function. Additionally, we incorporate the value of two value functions into the optimization objective of the noisy diffusion process, ensuring that the generated action closely aligns with the policies distribution in the offline dataset while meeting safety constraints. The main contributions of this work are as follows:

1. SPI facilitates the evaluation of safety policies by providing data support through the mapping and processing of data within the offline dataset. Considering the temporal effects on the safety value function of action-state pairs, we remapped the cost attribute in the dataset to avoid events such as cost-effectiveness disappears.

2. We couple the diffusion model with action value function and safety value function to equip SPI with the ability to balance fitting distributions and generating safety actions. This enables SPI to learn complex Gaussian distributions and develop new, diverse safety actions.

3. In numerous safety-related offline reinforcement learning tasks, SPI demonstrates a better overall performance than baseline algorithms in both high reward and safety.

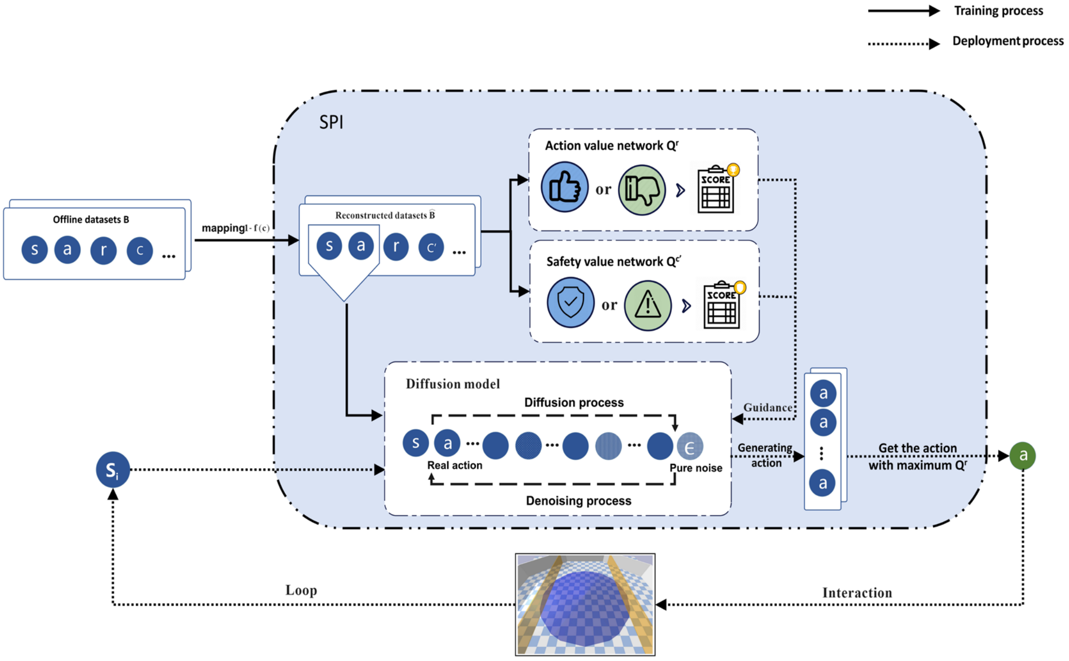


Figure 1. The framework of SPI

2 Related Work

2.1 Offline Reinforcement Learning

The distributional shift, which is a significant issue in Offline RL, refers to the discrepancy between the distributions of training data and actual data, leading an offline-trained model to overestimate the Q-value of unknown state-action pairs. The BCQ [20] algorithm considers constraining the deviation of the actions of the current policy from the actions in the utilized dataset, but the generated policy is constrained by the expressive capacity of its behavior policy. The BEAR [21] proposes support-set matching, which reduces reliance on the behavior policy. However, the implementation of support-set matching introduces the complexity of tuning various hyperparameters. The TD3+BC [22] algorithm has a simple architecture, adding an action cloning regularization term to the value function and normalizing states, but it requires further adjustment of the action cloning constraint. The IQL [23] algorithm reconstructs its policy and value function used a SARSA-style approach and employs the advantage weighted regression (AWR) method for policy extraction purposes, limiting the update of the Q-value function entirely within the offline dataset avoids the constraints associated with behavior cloning. However, this approach is also susceptible to the influence of noise present in the data set. The RAMBO [24] algorithm achieves conservatism by modifying the learned model dynamic equation, using an adversarial approach not only ensures the pessimism of the algorithm but also enhances its robustness.

2.2 Safe Offline Reinforcement Learning

Safe offline reinforcement learning is based on Offline RL to find policies that satisfy safety constraints. This requires finding safe policies while simultaneously avoiding extrapolation errors, ensuring that the learned policies are both effective and compliant with safety requirements. Previous works have classified safe constraint levels into three categories: hard, probabilistic, and soft constraints. The degrees of the constraints range from strong to weak. For all times $i \in \{0, ..., I\}$ and constraint indexes $j \in \{1, ..., n_c\}$, respectively, hard constraints ($c_i^j \le 0$) mean that the agent cannot break any rules during training. The TREBI [15] algorithm addresses the constrained policy optimization problem from the trajectory distribution perspective. Probabilistic constraints ($P(c_i^j \le 0) \ge p^j$), as their name suggests, mean that the agent is less likely to break the rules. The CODAC [25] algorithm learns conservative reward distributions by penalizing the predicted quantiles of outlier behaviors. The VOCE [26] algorithm uses probabilistic inference and pessimistic estimation methods to reconstruct an offline safe reinforcement learning problem. The COptiDICE [14] algorithm optimizes policies by correcting their steady-state distributions. Soft constraints ($c_i^j \le \lambda_j$) look for a feasible solution that makes all violations of the objective function as small as possible. When an action is out of distribution, the CPQ [17] algorithm increases its cost value while protecting the state-action pairs that satisfy the safety requirements and have similar distributions more.

Our algorithm belongs to the category of soft constraint algorithms. Unlike the aforementioned algorithms which reduce the cost of the policy to an acceptable range through constraints, SPI remaps the dataset and utilizes safety value functions and action value functions to guide the diffusion model in generating safety policies.

3 Preliminaries

3.1 Problem Settings

For safe offline reinforcement learning, CMDPS ($\mathcal{S}$, $\mathcal{A}$, $\mathcal{P}$, r, c, γ) are used for modeling. $\mathcal{S}$ is a finite set of states, and $\mathcal{A}$ is a finite set of actions. $\mathcal{P}(s_{i+1} \mid s_i, a_i) \rightarrow [0,1]$ is a state transition probability matrix, $r_i(s_i, a_i) \rightarrow \mathbb{R}$ is a reward function that describe the rewards associated with the current state-action pair, $c_i(s_i, a_i) \rightarrow \mathbb{R}$ is a cost function that express the risk factors associated with the current state-action pair, and γ is a discount factor: $\gamma \in [0,1]$. In a CMDPs trajectory $\tau = \{s_0, a_0, r_0, c_0, \ldots, s_i, a_i, r_i, c_i, \ldots\}$, expected cumulative reward $G^r = \Sigma_{i=0}^{\infty} \gamma^i r_i$ and expected cumulative cost $G^c = \Sigma_{i=0}^{\infty} \gamma^i c_i$ are the important metrics used to evaluate policy performance. Safe offline reinforcement learning aims to use an offline dataset $B(s_i, a_i, r_i, c_i, s_{i+1})$ to train a policy $\pi\ (a \mid s)$ that maximizes the expected cumulative reward under the constraints of safety, such as expected cumulative cost less than the cost threshold b. The Q-value function or action value function $Q^r(s_i, \mathbf{a}_i) = \mathbb{E}_{\tau\sim\pi}[G_r | s_i, a_i]$ is typically used to evaluate the action a taken by policy π in the current state s. The optimal policy is represented as π^*.

$$\begin{aligned} \pi^* &= \arg\max_{\pi} G^r \\ \text{s.t.} G^c &\le b. \end{aligned} \tag{1}$$

3.2 DDPM

The DDPM trains a noise predictor by gradually adding noise to the original image data until the original image data $\mathbf{x}_0 \sim q(\mathbf{x}_0)$ is completely transformed into a standard normal distribution $\mathcal{N}(\mathbf{0}, \mathbf{I})$. This noise predictor is then used to generate images by denoising based on the standard normal distribution $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$.

The diffusion proceθss (forward process) entails the incorporation of the corresponding variances β into the original probability distribution $q(\mathbf{x}_0)$ or the previous probability distribution $q(\mathbf{x}_t)$ in accordance with the timestep t, thereby yielding a conditional data probability distribution $q(\mathbf{x}_t | \mathbf{x}_{t-1})$ at the timestep t.

$$q(\mathrm{x}_t \mid \mathrm{x}_{t-1}) := \mathcal{N}(\mathrm{x}_t; \sqrt{1-\beta_t}\mathrm{x}_{t-1}, \beta_t \mathrm{I}) \tag{2}$$

The denoising process (reserve process) involves the predicted conditional probability distribution $p_\theta\ (\mathbf{x}_{t-1} | \mathbf{x}_t)$ of the sample variables produced by the trained noise predictor $\mu_\theta(\mathbf{x}_t, t)$ at the timestep t, again operating as a Markov decision process.

$$p_\theta(\mathbf{x}_{t-1} | \mathbf{x}_t) := \mathcal{N}\ (\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t,t), \textstyle\sum_\theta(\mathbf{x}_t,t)) \tag{3}$$

4 Safe Policy Improvement

SPI achieves safe policy improvement through three steps: safe character improvement, safety distribution capturing, and safe policy extraction.

4.1 Safe Character Improvement

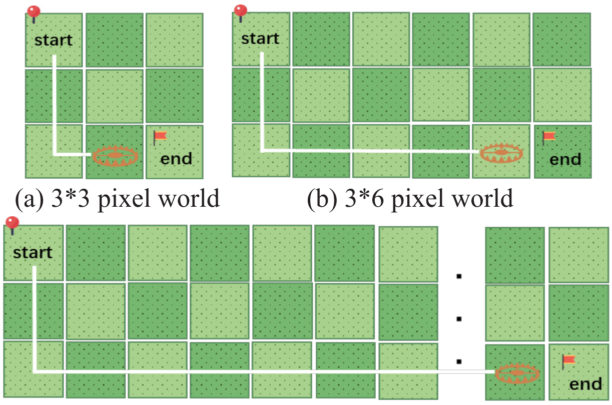
Safe Offline RL uses CMDPS to represent trajectories, quantifying potential cost factors into deterministic values using cost features. To improve data quality, we first consider reconstructing the dataset. This reconstructing is intended to enhance the cost characteristics and encourage reconstructed data to meet the needs of algorithm convergence. We have defined c' to measure the safety of the actions taken by policy π in the current state.

$$c' = 1 - f(c) = \begin{cases} 0, & \text{if unsafe} \\ else, & \text{if safe} \end{cases} \tag{4}$$

where $f(c) \rightarrow [0, 1]$, and through reconstruction, we obtained a feature-enhanced dataset $\hat{B}(s_i, a_i, r_i, c'_i, s_{i+1})$ from dataset $B(s_i, a_i, r_i, c_i, s_{i+1})$. Analogous to the action value function $Q^r(s_i, a_i)$, we have developed a safety value function to evaluate the safety of the actions taken by policy π in the current state.

$$Q^{c'}(s_i, a_i) = \mathbb{E}_{\tau\sim\pi}[\sum_{i=0}^{\infty} \gamma^i c'_i] \tag{5}$$

As shown in Eq. (5), when calculating the safety value function of the action-state pairs, the discount factor is also included in the judgment to ensure that the state-action pairs have excellent safety even in long-period trajectories. When the discount factor γ is small, the policy relies on the immediate cost, while the policy is more far-sighted when the discount factor γ tends to one. However, when the data is sparse, it may trigger the occurrence of vicious events such as cost-effectiveness disappears.



(a) 3*3 pixel world (b) 3*6 pixel world

(c) 3*n pixel world

Figure 2. Pixel world

As illustrated in Figure 2, the agent always starts at the initial position (0,0) with the goal of reaching the end safely. If it falls into a trap it will get the cost of 1, and if it reaches the end it will get the reward of 1. The rest is grass, with no reward nor cost.

When constraining using c, except for the trap position where the cost is 1, all other squares have a cost of 0. The policy is derived by minimizing the total cost. When we consider the use of white trajectory to measure the safety value function of the state-action pairs of the initial position, we get $1{*}\gamma^3$ in 3*3 pixel world as Figure 2(a), $1{*}\gamma^6$ in 3*6 pixel world as Figure 2(b). If considering a larger pixel world like 3*n($n\to\infty$) pixel world in Figure 2(c), the exponent of the discount factor is so large that the power operation ultimately offsets to the effect of cost since the discount factor is a decimal number.

To address this, we introduce c', which solves the problem of vanishing influence through mapping. All other squares have c' of 1 except trap, where c' is 0. The policy is derived by maximizing c'. Similarly, we use the white trajectory to measure the safety value function of the state-action pairs of the initial position, and we get $1{*}\gamma^5 + 1{*}\gamma^4 + 1{*}\gamma^3 + 1{*}\gamma^2 + 1{*}\gamma^1 + 1{*}\gamma^0$ in 3*6 pixel world. For larger pixel worlds, we can get larger values.

4.2 Safety Distribution Capturing

The diffusion model exhibits strong data modeling capabilities and policy representation abilities. However, naively applying it directly to the reconstructed dataset would only yield policies similar to those in the original dataset, akin to imitation learning. To enable the model to capture the distribution of safe data, we integrate the diffusion model with two value functions. This allows the diffusion model to be guided by both value functions during training, facilitating the generation of diverse policies. We conceptualize this as safety distribution capturer D_ω, which is trained using data (s_i, a_i, r_i, c'_i, s_{i+1}) sampled from the reconstructed offline dataset $\hat{\mathcal{B}}$.

$$\begin{aligned}\omega \leftarrow arg\min\nolimits_{\omega}[\eta KL_{\omega}((s_i,a_i)\,|\,(s_i,a_i^g))\\ -\mu Q_{\theta}^{r}(s_i,a_i^g)-\phi Q_{\lambda}^{c'}(s_i,a_i^g)]\\ \text{s.t. } (s_i,a_i)\sim\hat{\mathcal{B}},(s_i,a_i^g)\sim\mathcal{D}_{\omega}\end{aligned} \quad \textbf{(6)}$$

We used a combination of the KL divergence term and the value function term to form the safety distribution capturer D_ω. The KL divergence term ensures that the generated policies are sufficiently similar to the distribution in the original dataset, thereby avoiding extrapolation errors. The value function term facilitates the safety distribution capturer in selecting actions that have high values for both Q^r and $Q^{c'}$ in the current state. To generate policies as diverse as possible, we define the parameter relationship between the KL divergence term and the two value functions as $\eta + \mu + \phi = 1$ and $\eta \in [0,1]$, $\mu \in [0,1]$, $\phi \in [0,1]$.

4.2.1 KL Divergence Term

To avoid extrapolation errors, we use KL divergence to measure the discrepancy between the captured data distribution and the data distribution in the offline dataset. According to the Markov property, it can be inferred:

$$\begin{aligned}(\mathrm{s}_0,\mathrm{a}_t) &= \sqrt{\alpha_t}(\mathrm{s}_0,\mathrm{a}_{t-1})+\sqrt{1-\alpha_t}\,\mathrm{y}_{t-1}\\ &= \sqrt{\alpha_t\alpha_{t-1}}(\mathrm{s}_0,\mathrm{a}_{t-2})+\sqrt{1-\alpha_t\alpha_{t-1}}\,\bar{\mathrm{y}}_{t-2}\\ &= \ldots\\ &= \sqrt{\bar{\alpha}_t}(\mathrm{s}_0,\mathrm{a}_0)+\sqrt{1-\bar{\alpha}_t}\,\mathrm{y}\end{aligned} \quad \textbf{(7)}$$

Where $\alpha_t = 1-\beta_t$, $\bar{\alpha}_t = \Pi_{j=1}^{T}\alpha_j$, $\mathbf{y}_t$ is a standard gaussian distribution, and $\bar{\mathbf{y}}_t$ is the result after combining the two Gaussian distributions. Furthermore, we use {1, 2, 3, …, t} to express the diffusion steps.

In the forward process or diffusion process, we sample state-action pairs and continuously add gaussian noise until the distribution of the state-action pairs becomes pure noise. For the reverse process or denoising process, we use a parameterized neural network p_ω for fitting.

$$\begin{aligned}&p_{\omega}((\mathbf{s}_0,\mathbf{a}_{t-1})|\,(\mathbf{s}_0,\mathbf{a}_t)) =\\ &\mathcal{N}(\boldsymbol{\mu}_{\omega}((\mathbf{s}_0,\mathbf{a}_t),t),\textstyle\sum_{\omega}((\mathbf{s}_0,\mathbf{a}_t),t))\end{aligned} \quad \textbf{(8)}$$

Specifically, we use noise model ϵ_ω to predict the added noise distribution, and then gradually denoise from pure gaussian noise, ultimately fitting it back to the state-action pairs. Optimization is conducted by using gradient descent with the evidence lower bound (ELBO) in variational inference. Where it represents the added noise labels, while denotes the noise generated by the neural network. The fitting between the two ensures the consistency of the generated state-action pair (s_i, a_i^g) with (s_i, a_i) in the dataset.

$$KL_{\omega}((s_i,a_i)\,|\,(s_i,a_i^g)) = \\ \|\epsilon-\epsilon_{\omega}(\sqrt{\overline{\alpha}_t}(\mathrm{s}_i,\mathrm{a}_i)+\sqrt{1-\overline{\alpha}_t}\epsilon,t)\|^2 \tag{9}$$

4.2.2 Value Function Term

We utilize two value functions to guide the diffusion model in capturing a safe and high reward distribution. Here, Q_{θ}^{r} encourages sampling in action space with high rewards, while $Q_{\lambda}^{c'}$ promotes sampling in action space with high safety. Together, both Q_{θ}^{r} and $Q_{\lambda}^{c'}$ functions influence the diffusion model.

$$\theta \leftarrow arg\min_{\theta} \\ \|r_i+\gamma Q_{\theta}^{r}(s_{i+1},a_{i+1})-Q_{\theta}^{r}(s_i,a_i)\|^2 \tag{10}$$

$$\lambda \leftarrow arg\min_{\lambda} \\ \|c_i'+\gamma Q_{\lambda}^{c'}(s_{i+1},a_{i+1})-Q_{\lambda}^{c'}(s_i,a_i)\|^2 \tag{11}$$

4.3 Safe Policy Extraction

After safe character improvement and safety distribution capturing, SPI has the capability to model safe data. Ultimately, we use the state value function to filter the generated actions, extracting safe policies.

$$\pi(s) = arg\max_{a_i^g} Q_{\theta}^{r}(s,a_i^g) \\ \text{s.t. } \{a_i^g \sim D_{\omega}(s)\}_{i=1}^{n} \tag{12}$$

In the actual experiments, to mitigate the impact of random noise and outliers, we duplicate the state s 50 times and input it into D_{ω}. Then, we perform a maximum selection on the 500 action values Q_{θ}^{r} generated by D_{ω}.

The entire process is illustrated in Algorithm 1.

Algorithm 1. Safe policy improvement via diffusion model

```
Require: safety distribution capturer $D_\omega$, offline dataset
$\mathcal{B}$, parameter $\eta$, $\mu$, $\phi$;
Initialize: noise model $\epsilon_\omega$, action value network $Q_\theta^r$,
safety value network $Q_\lambda^{c'}$
for each iteration do:
    # Safe Character Improvement
    Reconstructing dataset $\mathcal{B}$ yields $\hat{\mathcal{B}}$ by Equation (4)
    # Safety Dritribution Capturing
    Random batch sample $(s_i, a_i, s_{i+1}, r_i, c_i') \sim \hat{\mathcal{B}}$
    Using $(s_i, a_i, s_{i+1}, r_i)$ trains $Q_\theta^r$ by Equation (10)
    Using $(s_i, a_i, s_{i+1}, c'_i)$ trains $Q_\lambda^{c'}$ by Equation (5) and by Equation (11).
    Using trains $\epsilon_\omega$ by Equation (9).
    Updating noise model $\epsilon_\omega$, action value network $Q_\theta^r$ safety value network $Q_\lambda^{c'}$.
    Train $D_\omega$ by Equation (6).
    Updating safety distribution capturer $D_\omega$.
    # Safety Policy Extraction
    Extracting the safe policy by Equation (12)
end for
```

5 Experiments

Environments and Dataset

We leverage the DSRL [27] dataset, which was specifically designed for offline safe reinforcement learning. The performance of the tested algorithms is evaluated by assessing baseline tasks with different difficulty levels. In the context of applying safe offline reinforcement learning algorithms, we select common safe environments such as the Bullet Safety Gym [28] and Safety Gymnasium [29], which encompass a variety of applied intelligent agents including drones, ants, and hoppers. The implemented tasks can be divided into three categories. The Circle task requires the agent to move in a clockwise direction within the defined circle, and the agent receives cost penalties if it deviates from the specified safe boundary. In the running task scenario, the agent is trained to move between the safety boundaries on either side. In addition to the penalty for exceeding the safety boundaries, the agent also receives a penalty if its speed becomes too high during its movement. The velocity task demands that the agent walks smoothly while satisfying the imposed velocity constraints. The used experimental environment and agent types are shown in Figure 3.

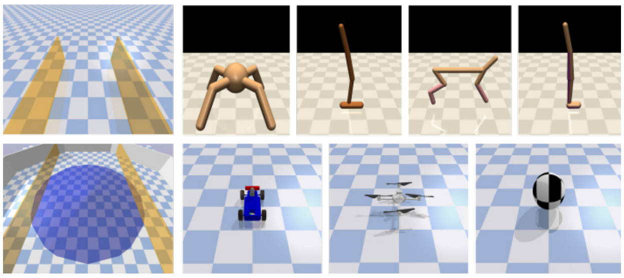


Figure 3. Experimental environments and agents

Evaluation Indicators

The primary objective within the safe offline reinforcement learning domain is to identify the optimal policy that meets to the constraints. The key performance metrics used for evaluation are the maximization of rewards and the minimization of costs. Additionally, to mitigate the impact of random errors, the collected data undergoes a normalization process, whereby the results are averaged across 20 episodes. Moreover, we use bold to indicate the highest safety and reward performance trained under the same task.

Baselines

We used five different baselines for comparison:

CPQ [17] algorithm involves constraining the value function by penalizing the action value function of out-of-distribution actions to satisfy safety constraints.

COptiDICE [14] algorithm regularizes the policy by utilizing the difference between the steady-state distribution of the optimal policy and the offline dataset distribution to optimize the problem.

BEARL and **BCQL** [30] algorithm belong to the same category as COptiDICE, and they incorporate Lagrange constraints in offline reinforcement learning algorithms to regularizes the policy.

BC [31] algorithm represents behavior cloning, and its performance can be viewed as the behavioral policy performance derived from an offline dataset. By comparing our proposed algorithm with the BC algorithm, we can assess whether our approach has learned a better policy from the offline dataset.

5.1 Comparison Experiment

A performance comparison between SPI and other baselines across 8 different experimental tasks is shown in Table 1. Owing to its powerful data fitting capabilities and ability to balance various tradeoffs, SPI algorithm shows respectable performance across various tasks. By examining the average metric, our method exhibits a balanced performance in terms of reward and cost, which aligns with our anticipated desirable outcome as depicted in Figure 4 and Figure 5. Moreover, in some tasks where SPI does not yield the highest rewards, the performance of SPI is still comparable to the optimal algorithms. Specifically, under the offlineAntVelocityGymnasium-v1 task, SPI is approximately 1 percentage point lower than both BCQL and BC algorithms in terms of maximum rewards, but the cost metrics for BCQL and BC are 328 and 573 percentage points higher than SPI, respectively. For the OfflineWalker2dVelocityGymnasium-v1 task, the CPQ algorithm, while having a cost value similar to SPI, demonstrates a significant discrepancy in the reward metric, being 78 percentage points lower than SPI. Additionally, both BEARL and BCQL match SPI's performance on this specific task but do not outperform SPI in other tasks. For example, in certain tasks such as OfflineHopperVelocityGymnasium-v1, the performance is unacceptably low, indicating potential shortcomings in the algorithm's robustness. In contrast, the baseline algorithms struggle to strike a balance between their rewards and costs. Specifically, some baseline algorithms (**CPQ**, **COptiDICE**, and **BEARL**) are overly conservative, becoming excessively focused on safety and unwilling to execute policies with higher rewards. Conversely, BCQL algorithm is too aggressive, violating the cost constraints while attempting to pursue high-reward policies. Additionally, the performance of the BC algorithm is heavily depended on the quality of the employed dataset; therefore, it has relatively low robustness. SPI exhibits superior performance across all datasets, due to its capacity to capture safety characteristics and effectively articulate policies.

Table 1. Comparison experiment results

Task	Ours		CPQ		COptiDICE		BEARL		BCQL		BC	
	reward↑	cost↓	reward↑	cost↓	reward↑	cost↓	reward↑	cost↓	reward↑	cost↓	reward↑	cost↓
OfflineBallCircle-v0	**0.95**	5.65	0.66	**0.00**	0.69	3.87	0.64	2.64	0.72	2.46	0.63	2.57
OfflineCarCircle-v0	**0.71**	0.19	0.70	**0.00**	0.42	2.37	0.66	0.34	0.56	1.22	0.40	3.94
OfflineCarRun-v0	**0.97**	**0.00**	0.93	**0.00**	0.95	**0.00**	0.42	**0.00**	0.96	**0.00**	0.96	**0.00**
OfflineDroneCircle-v0	**0.64**	6.16	-0.26	**0.00**	0.27	1.74	-0.26	**0.00**	0.48	0.35	0.60	4.90
OfflineAntVelocityGymnasium-v1	0.98	0.57	-1.01	0.08	0.97	2.73	-0.51	**0.00**	**0.99**	3.85	**0.99**	6.30
OfflineHalfCheetahVelocityGymnasium-v1	**0.97**	**0.00**	-0.22	**0.00**	0.69	**0.00**	-0.27	0.12	0.96	9.89	0.93	1.93
OfflineHopperVelocityGymnasium-v1	**0.89**	4.86	0.03	**0.00**	0.40	1.22	0.17	9.44	0.83	9.37	0.70	3.37
OfflineWalker2dVelocityGymnasium-v1	0.79	0.01	0.01	**0.00**	0.13	2.43	0.79	**0.00**	**0.80**	**0.00**	0.77	0.02
Average	**0.86**	2.18	0.11	**0.01**	0.57	1.80	0.21	1.57	0.79	3.39	0.75	2.88

Table 2. Ablation experiment results

Task	SPI		SPI except $Q^{c'}$		SPI except Q^{r}	
	reward↑	cost↓	reward↑	cost↓	reward↑	cost↓
OfflineBallCircle-v0	**0.95**	5.65	0.94	5.88	0.04	**0.00**
OfflineCarCircle-v0	0.71	0.19	**0.93**	9.08	-0.01	**0.00**
OfflineCarRun-v0	**0.97**	**0.00**	0.96	0.18	0.95	**0.00**
OfflineDroneCircle-v0	**0.64**	6.16	0.56	8.22	-0.26	**2.89**
OfflineAntVelocityGymnasium-v1	0.98	0.57	**1.04**	19.70	-1.01	**0.00**
OfflineHalfCheetahVelocityGymnasium-v1	0.97	**0.00**	**1.01**	2.99	-0.11	**0.00**
OfflineHopperVelocityGymnasium-v1	**0.89**	4.86	0.17	8.26	0.06	**0.00**
OfflineWalker2dVelocityGymnasium-v1	**0.79**	0.01	0.17	6.10	-0.01	**0.00**
Average	**0.86**	2.18	0.72	7.55	-0.04	**0.36**

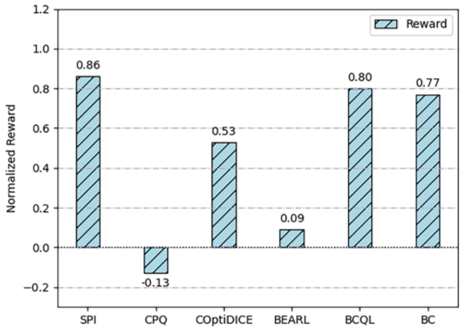


Figure 4. Average reward of comparison experiment

5.2 Ablation Results

To evaluate the effectiveness of the proposed algorithmic design, we decompose the SPI algorithm into several functional components and modify the model by selectively removing or replacing certain modules. The complete SPI algorithm is composed of KL divergence term and value function term. The combination of KL divergence term and Q^r function focuses on producing high reward policies, while the combination of KL divergence term and $Q^{c'}$ function ensures the generation of policies that satisfy safety constraints.

In this ablation study, we aim to address the following research questions: 1) Do the individual modules within the SPI algorithm function as intended and contribute their expected outcomes? 2) What is the feasibility of addressing the mixed problem using a single constraint, specifically by focusing exclusively on the safety constraint while disregarding the original data distribution during the learning process?

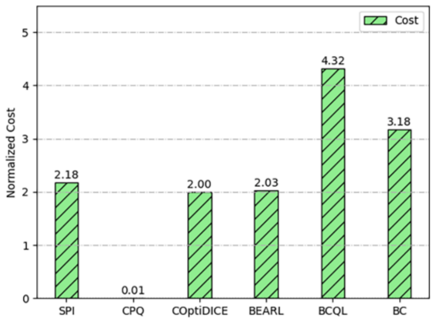


Figure 5. Average cost of comparison experiments

On the basis of the results of the ablation study, as summarized in Table 2, we analyze the findings as follows:

1. The results clearly demonstrate that the full SPI algorithm outperforms the versions where certain modules have been removed or replaced. The performance of the Q^r function and the $Q^{c'}$ function confirms their effectiveness in exploring and satisfying the safety constraint, respectively. The discrepancies among the metrics observed across the different variants indicate that, in most environments, the regions with high rewards, low costs, and similarity to the original data distribution do not completely overlap.

2. The stark difference in returns between the KL divergence term + $Q^{c'}$ and the SPI versions underscore that relying on a single constraint cannot achieve the same level of performance as that of the multi-constraint approach. Solely guiding the algorithm toward safety during training causes KL divergence term + $Q^{c'}$ variant to struggle in fitting the out-of-distribution action data during the evaluation, leading it to converge to a local optimum.

Moreover, through a comprehensive analysis of the performance of various ablation modules across 8 tasks, we find that SPI achieves more robust performance while pursuing stably excellent high expected returns and low costs. As shown in Figure 6 and Figure 7, the overall performance of the SPI algorithm is superior to that of its simplified versions, which suggests that when the individual components of SPI work together, they are able to navigate multiple objectives more effectively than the reduced versions are able to.

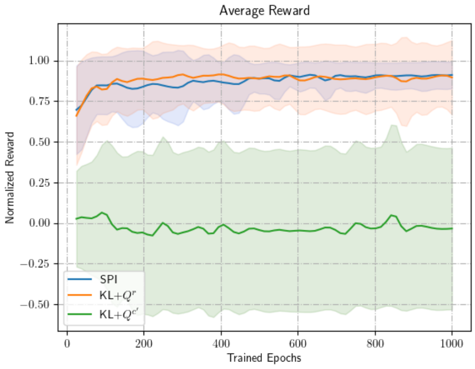


Figure 6. Average reward of ablation experiments

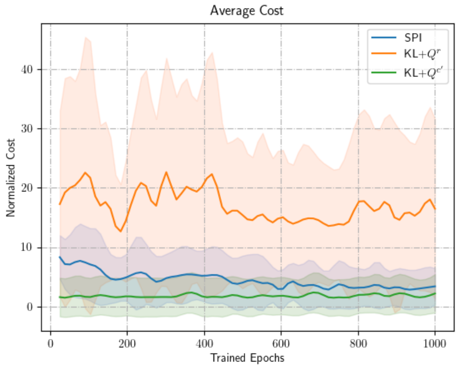


Figure 7. Average cost of ablation experiments

5.3 Parameter Sensitivity

To address diverse policy needs, we leverage random functions and conduct three sets of random sampling experiments under the constraint that the sum of the three hyperparameters must be 1. η, μ, and ϕ are used to regulate the relationship between the KL divergence term and the two value functions. The performance of SPI is shown in Figure 8 and Figure 9. Regardless of whether the policy is inclined towards more conservative or more aggressive configurations, our SPI algorithm is able to converge quickly. This ability to adapt to different parameter settings demonstrates the robustness and flexibility of the SPI approach. By adjusting the balance between the constituent objectives, the algorithm can cater to a wide range of policy needs, from high-cost to low-cost scenarios.

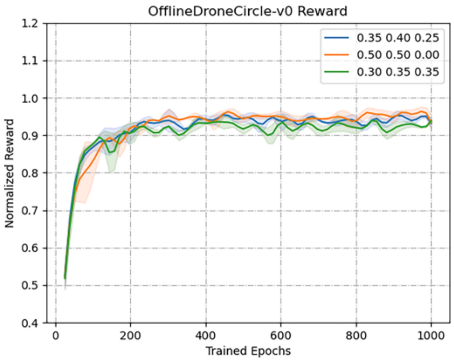


Figure 8. Average reward of random parameters

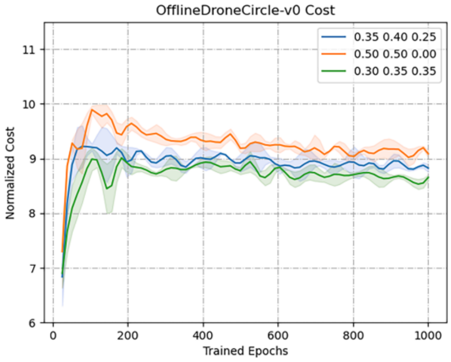


Figure 9. Average cost of random parameters

6 Conclusion

We have proposed an algorithm to make safe policy improvement via diffusion model for offline reinforcement learning, named SPI**.** To capture the data distribution within the offline dataset and generate safe policies, we integrate a diffusion model into the offline reinforcement learning framework. By guiding the diffusion process with two value functions, we enable SPI to progressively approach safe policies. Additionally, we remap the cost within the dataset to evaluate policies to avoid cost-effectiveness disappears, encouraging the policies to achieve high levels of safety. The SPI nearly attains optimal performance in individual metrics, including overall performance, SPI markedly surpasses the baseline algorithms.

In future work, we aim to extend our algorithm to the domain of real-time budget constraints, enabling dynamic adjustment of constraint parameters in response to feedback during deployment, while still satisfying safety requirements.

Acknowledgements

This work is supported by the Anhui Future Technology Research Institute Industry Guidance Fund Project (Grant No. 2023cyyd04), the Key Project of Natural Science Research of Higher Education Institution of Anhui Province of China (Grant No. 2022AH050970, 2022AH050972), and the Project of Research of Anhui Polytechnic University (Grant No. Xjky2022150).

References

[1] R. F. Prudencio, M. R. O. A. Mximo, E. L. Colombini, A Survey on Offline Reinforcement Learning: Taxonomy, Review, and Open Problems, *IEEE Transactions on Neural Networks and Learning Systems*, Vol. 35, No. 8, pp. 10237-10257, August, 2024.
https://doi.org/10.1109/TNNLS.2023.3250269

[2] Y. Singh, R. Kumar, S. Kabdal, P. Upadhyay, YouTube Video Summarizer using NLP: A Review, *International Journal of Performability Engineering*, Vol. 19, No. 12, pp. 817-823, December, 2023.
https://doi.org/10.23940/ijpe.23.12.p6.817823

[3] J. Nan, W. Deng, R. Zhang, Y. Wang, J. Ding, A Long-Term Actor Network for Human-Like Car-Following Trajectory Planning Guided by Offline Sample-Based Deep Inverse Reinforcement Learning, *IEEE Transactions on Automation Science and Engineering*, Vol. 21, No. 4, pp. 7094-7106, October, 2024.
https://doi.org/10.1109/TASE.2023.3337230

[4] X. Fang, Q. Zhang, Y. Gao, D. Zhao, Offline Reinforcement Learning for Autonomous Driving with Real World Driving Data, *25th IEEE International Conference on Intelligent Transportation Systems*, Macau, China, 2022, pp. 3417-3422.
https://doi.org/10.1109/ITSC55140.2022.9922100

[5] C. Gao, K. Huang, J. Chen, Y. Zhang, B. Li, P. Jiang, S. Wang, Z. Zhang, X. He, Alleviating Matthew Effect of Offline Reinforcement Learning in Interactive Recommendation, *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, Taipei, China, 2023, pp. 238-248.
https://doi.org/10.1145/3539618.3591636

[6] T. Xiao, D. Wang, A General Offline Reinforcement Learning Framework for Interactive Recommendation, *Thirty-Fifth AAAI Conference on Artificial Intelligence*, virtual, 2021, pp. 4512-4520.
https://doi.org/10.1609/aaai.v35i5.16579

[7] W. Xiong, H. Zhong, C. Shi, C. Shen, L. Wang, T. Zhang, Nearly Minimax Optimal Offline Reinforcement Learning with Linear Function Approximation: Single-Agent MDP and Markov Game, *The Eleventh International Conference on Learning Representations*, Kigali, Rwanda, 2023. pp.1-32.

[8] Q. Cui, S. S. Du, Provably Efficient Offline Multi-agent Reinforcement Learning via Strategy-wise Bonus, *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems*, New Orleans, LA, USA, 2022, pp.11739-11751.

[9] A. X. Lee, C. Devin, J. T. Springenberg, Y. Zhou, T. Lampe, A. Abdolmaleki, K. Bousmalis, How to Spend Your Robot Time: Bridging Kickstarting and Offline Reinforcement

Learning for Vision-based Robotic Manipulation, *International Conference on Intelligent Robots and Systems*, Kyoto, Japan, 2022, pp. 2468-2475. https://doi.org/10.1109/IROS47612.2022.9981126

[10] J. Jin, D. Graves, C. Haigh, J. Luo, M. Jgersand, Offline Learning of Counterfactual Predictions for Real-World Robotic Reinforcement Learning, *2022 International Conference on Robotics and Automation*, Philadelphia, PA, USA, 2022, pp. 3616-3623. https://doi.org/10.1109/ICRA46639.2022.9811963

[11] G. Zhou, L. Ke, S. S. Srinivasa, A. Gupta, A. Rajeswaran, V. Kumar, Real World Offline Reinforcement Learning with Realistic Data Source, *IEEE International Conference on Robotics and Automation*, London, UK, 2023, pp. 7176-7183. https://doi.org/10.1109/ICRA48891.2023.10161474

[12] N. Y. Siegel, J. T. Springenberg, F. Berkenkamp, A. Abdolmaleki, M. Neunert, T. Lampe, R. Hafner, N. Heess and M. A. Riedmiller, Keep Doing What Worked: Behavior Modelling Priors for Offline Reinforcement Learning, *8th International Conference on Learning Representations*, Addis Ababa, Ethiopia, 2020. pp. 1-21.

[13] H. Le, C. Voloshin, Y. Yue, Batch Policy Learning under Constraints, *Proceedings of the 36th International Conference on Machine Learning*, California, USA, 2019, pp. 3703-3712.

[14] J. Lee, C. Paduraru, D. J. Mankowitz, N. Heess, D. Precup, K. Kim, A. Guez, COptiDICE: Offline Constrained Reinforcement Learning via Stationary Distribution Correction Estimation, *The Tenth International Conference on Learning Representations*, Virtual, 2022. pp.1-24.

[15] Q. Lin, B. Tang, Z. Wu, C. Yu, S. Mao, Q. Xie, X. Wang, D. Wang, Safe offline reinforcement learning with real-time budget constraints, *International Conference on Machine Learning*, Honolulu, Hawaii, USA, 2023, pp. 21127-21152.

[16] J. J. Choi, F. Castaeda, C. J. Tomlin, K. Sreenath, Reinforcement Learning for Safety-Critical Control under Model Uncertainty, using Control Lyapunov Functions and Control Barrier Functions, *Robotics: Science and Systems XVI*, Virtual Event / Corvalis, Oregon, USA, 2020. pp. 1-9.

[17] H. Xu, X. Zhan, X. Zhu, Constraints Penalized Q-learning for Safe Offline Reinforcement Learning, *Thirty-Sixth AAAI Conference on Artificial Intelligence*, virtual, 2022, pp. 8753-8760. https://doi.org/10.1609/aaai.v36i8.20855

[18] Z. Wang, J. J. Hunt, M. Zhou, Diffusion Policies as an Expressive Policy Class for Offline Reinforcement Learning, *The Eleventh International Conference on Learning Representations*, Kigali, Rwanda, 2023. pp. 1-17.

[19] S. Jhingran, M. K. Goyal, N. Rakesh, DQLC: A Novel Algorithm to Enhance Performance of Applications in Cloud Environment, *International Journal of Performability Engineering*, Vol. 19, No. 12, pp. 771-778, December, 2023. https://doi.org/10.23940/ijpe.23.12.p1.771778

[20] S. Fujimoto, D. Meger, D. Precup, Off-Policy Deep Reinforcement Learning without Exploration, *Proceedings of the 36th International Conference on Machine Learning*, California, USA, 2019, pp. 2052-2062.

[21] A. Kumar, J. Fu, M. Soh, G. Tucker, S. Levine, Stabilizing Off-Policy Q-Learning via Bootstrapping Error Reduction, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems*, Vancouver, BC, Canada, 2019, pp. 11761-11771.

[22] S. Fujimoto, S. S. Gu, A Minimalist Approach to Offline Reinforcement Learning, *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems*, Virtual, 2021, pp. 20132-20145.

[23] I. Kostrikov, A. Nair, S. Levine, Offline Reinforcement Learning with Implicit Q-Learning, *The Tenth International Conference on Learning Representations*, Virtual, 2022. pp. 1-13.

[24] M. Rigter, B. Lacerda, N. Hawes, RAMBO-RL: Robust Adversarial Model-Based Offline Reinforcement Learning, *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems*, New Orleans, LA, USA, 2022. pp. 16082-16097.

[25] Y. J. Ma, D. Jayaraman, O. Bastani, Conservative Offline Distributional Reinforcement Learning, *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems*, Virtual, 2021, pp. 19235-19247.

[26] J. Guan, G. Chen, J. Ji, L. Yang, A. Zhou, Z. Li, C. Jiang, VOCE: Variational Optimization with Conservative Estimation for Offline Safe Reinforcement Learning, *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems*, New Orleans, LA, USA, 2023. pp. 33758-33780.

[27] Z. Liu, Z. Guo, H. Lin, Y. Yao, J. Zhu, Z. Cen, H. Hu, W. Yu, T. Zhang, J. Tan, D. Zhao, Datasets and Benchmarks for Offline Safe Reinforcement Learning, *Journal of Data-centric Machine Learning Research*, Vol. 1, pp. 1-29. July, 2024.

[28] S. Gronauer, *Bullet-Safety-Gym: A Framework for Constrained Reinforcement Learning*, Report, January, 2022.

[29] J. Ji, B. Zhang, J. Zhou, X. Pan, W. Huang, R. Sun, Y. Geng, Y. Zhong, J. Dai, Y. Yang, Safety Gymnasium: A Unified Safe Reinforcement Learning Benchmark, *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems*, New Orleans, LA, USA, 2023. pp. 18964-18993.

[30] A. Stooke, J. Achiam, P. Abbeel, Responsive Safety in Reinforcement Learning by PID Lagrangian Methods, *Proceedings of the 37th International Conference on Machine Learning*, Virtual Event, 2020, pp. 9133-9143.

[31] M. Bain, C. Sammut, A Framework for Behavioural Cloning, *Machine Intelligence*, Oxford, UK, 1995, pp. 103-129.

Biographies

Xiaohan Yang received her B.S. degree from Anhui Polytechnic University, in 2022. She is currently pursuing the master degree with the Anhui Polytechnic University of Software Engineering. Her recent research interests are offline reinforcement learning.

Jun Li is currently an associate professor of the School of Computer Science and Information at Anhui Polytechnic University, Wuhu, China. His research interests include deep reinforcement learning, autonomous driving, and human-machine interactions.

Jiang Liu earned his M.S. degree from the School of Computer and Information at Anhui University of Technology, China. He is currently an algorithm engineer at Anhui Zhongke Terahertz Technology Co., Ltd. His research interests primarily include image processing, reinforcement learning, and deep learning, among others.

Mengting Sun received her B.S. degree from Anhui Polytechnic University, in 2022. She is currently pursuing the master degree with the Anhui Polytechnic University of Computer Technology. Her recent research interests are Autonomous Driving decision methods.

Baozhu Chen received his B.S. degree from Anhui Polytechnic University, in 2023. He is currently pursuing the master degree with the Anhui Polytechnic University of Computer Technology. His recent research interests are Safety-Critical Scenario Generation for Autonomous Vehicles.