Model-based Safe Reinforcement Learning using Generalized Control Barrier Function

Haitong Ma[‡], Jianyu Chen[†], Shengbo Eben Li[*‡], Ziyu Lin[‡], and Sifa Zheng[‡]

***Abstract*— Model information can be used to predict future trajectories, so it has huge potential to avoid dangerous region when implementing reinforcement learning (RL) on real-world tasks, like autonomous driving. However, existing studies mostly use model-free constrained RL, which causes inevitable constraint violations. This paper proposes a model-based feasibility enhancement technique of constrained RL, which enhances the feasibility of policy using generalized control barrier function (GCBF) defined on the distance to constraint boundary. By using the model information, the policy can be optimized safely without violating actual safety constraints, and the sample efficiency is increased. The major difficulty of infeasibility in solving the constrained policy gradient is handled by an adaptive coefficient mechanism. We evaluate the proposed method in both simulations and real vehicle experiments in a complex autonomous driving collision avoidance task. The proposed method achieves up to four times fewer constraint violations and converges 3.36 times faster than baseline constrained RL approaches.**

I. Introduction

Safety is critical for implementing reinforcement learning (RL) on real-world facilities [1]. For instance, in the field of autonomous vehicle control, the collision must be avoided in case of causing physical harm to humans [2]. The pursuit of reducing constraints violation during training arises in both real-world training and sim-to-real transfer, i.e., the safe exploration problem. A safety-critical sequential decision problem is generally formulated to a constrained Markov decision process (CMDP), where some cost-based constraints are defined separately from maximizing the reward function [3]. There are three key elements in solving CMDP, including the constraints definition, solution techniques of the constrained optimization, and methods to handle infeasibility issues.

Firstly, multiple definitions of the cost-based constraints can be integrated with CMDP framework. Chance constraint is the most popular choice, where a one-hot design of cost signal is commonly used [4]. Conditional value at risk (CVaR) is designed to address those cases whose probability is small, but the cost incurred could still be significant, which is an improved version of basic chance constraint usually used in portfolio optimization [5]. Both of them are designed with long-horizon data-driven expectation, which is the inevitable choice for model-free RL. Although the low sampling efficiency can be handled with learning a cost value, the approximation error still poses a great challenge on learning a constraint-satisfying policy, which ends in the rising issue of constraints violations [6]. Numerous model-free algorithms are designed to solve this problem by focusing on the latter two components, i.e., developing solution techniques and designing special mechanisms to remedy this unexpected infeasibility issue. Uchibe et, al. (2007) adopts policy gradient projection and handle infeasibility issue by the restoration projection direction [7]. Chow et, al. (2018) adaptively adjusts the balance coefficient with reward and cost to achieve feasible solutions [8]. The well-known constrained policy optimization (CPO) utilizes a trust-region constraint to prevent reckless policy updates and also adopts rewards shaping for cost hazards [9]. However, the trial-and-error nature and inherent instability of data-driven approaches still block their constraint-obeying performance to be satisfying enough [6].

Model information is powerful assistance for planning and confining policy updates in reinforcement learning to predict future policy performance[10] [11]. Lyapunov stability is firstly considered to give the stability guarantee for a safe model-based learner while learning dynamics [12]. For feasible regions defined by inequality safety constraints, multiple learning-based controllers, i.e., the model-based constrained RL approaches, is proposed based on multi-step model rollout [13][14][15]. However, directly using the inequality safety constraint leads to high sensitiveness to disturbance and causes many unexpected constraints violations. The multi-step rollout also decreases sampling efficiency. The model information still has the potential for feasibility enhancement.

Control barrier function (CBF) is a powerful approach to ensure safety is a rather small prediction horizon using a concise constraint based on the current distance to the constraint boundary [16]. Continuous-time CBF is proposed as an inequality constraint on derivatives of constraints function lying only on the boundary of constrained set, developing from the safe barrier certificates [17][18]. Soon it transfers to the discrete-time version posing constraints on the adjacent time steps discussed by Agrawal et al. (2017) [19]. A significant drawback of standard CBF is the inability to handle high-order constraints. Nguyen et al. (2016) discusses the continuous-time high relative-degree constraint and augments new states by Lie derivatives [20]. The corresponding discrete-time version can refer to our previous work, which poses constraints on two nonadjacent steps [21].

[‡]School of Vehicle and Mobility, Tsinghua University. Email: {maht19@mails., lishbo@, linzy17@mails., zsf@}tsinghua.edu.cn.

[†]Institute for Interdisciplinary Information Sciences, Tsinghua University. Email: jianyuchen@tsinghua.edu.cn.

[*]All correspondence should be sent to S. Li.

In summary, Existing constrained RL studies suffer from the following problems: (1) Directly explore the dangerous area leads to dangerous constraint violations, where model information has potential to avoid violations, but existing studies have not fully tapped it; (2) The efficiency of existing model-based approaches is usually not satisfying due to multi-step rollout. We propose a model-based policy optimization approach with the generalized control barrier function (GCBF-MBPO), which can handle state constraints by penalizing the trends of getting closer to the constraint boundary. The model information is better utilized for preventing the agents from stepping into the actual dangerous zone. The main contribution of this paper is summarized as follows:

(1) We have fully dug the model's information for constrained RL by penalizing the trends getting closer to the constraint boundary. A constraint-satisfying policy can be learned without violating actual safety constraints. The constraints violations during training are up to 73.83% lower than baseline constrained RL approaches.

(2) The constraints formulation has the theoretically smallest required steps in each iteration without learning the cost approximation. The sampling efficiency improves by 3.36 times compared to baseline model-based constrained RL.

The paper is organized as follows. Section II is the preliminaries about the key components of solving CMDP and generalized control barrier function. Section III introduces the proposed model-based safe RL architecture and the adaptive conservativeness mechanism to handle the potential infeasible update. Section IV demonstrates the experiment results on the simulation platform and real autonomous vehicle. Section V concludes the paper.

II. Preliminaries

A. Constrained Markov Decision Process

Constrained reinforcement learning (RL) indicates the general problem of training an RL agent with constraints, usually with the intention of satisfying constraints throughout exploration in training and at test time.

$$\pi^* = \text{argmin}_{\pi \in \Pi_C} J_r(\pi) \tag{1}$$

where $J_r(\pi)$ is the expected return. The feasible policy set Π_C is determined by cost-based constraints:

$$\Pi_{\mathrm{C}} = \{\pi : J_{C_i}(\pi) \leq d_i\} \tag{2}$$

where $i = 1, 2, \ldots, k$ is the constraint index. Each J_{C_i} is the expected cost, and d_i is a pre-defined threshold.Recently, numerous efforts to improve constrained RL is based on the actor-critic architecture integrated with so called "constrained policy optimization" technique, which is also called constrained actor-critic (CAC). The critic update is the same as existing state-value algorithms like trust-region policy optimization (TRPO), and the actor update progress is constrained to find a constraint-satisfying policy. We here concisely introduce four algorithms based on CAC architecture.

B. Constraints Formulation and Methods to Handle Infeasibility Issue

A learning-based optimization procedure easily leads to the infeasibility issue, which a policy update outside Π_c happens due to some unexpected error. Therefore, feasibility is critical in safety-critical sequential decision problems. The aforementioned 3 key points of CMDP all help to address and handle the infeasibility issue. The first CAC-like algorithm is the policy gradient projection (PGP) [7]. The constraints formulation is an average reward constraint:

$$\lim_{T \to \infty} \left[\mathbb{E}_{s \sim d(s), a \sim \pi_k} \left(\frac{1}{T} \sum_{t=1}^{T} r_{c_i} \right) \right] \leq d_i \tag{3}$$

where r_{C_i} is the corresponding surrogate cost. Rosen gradient projection is used to solved constrained actor update, and the feasibility issue is considered by the restoration projection direction d for infeasible actions. The policy projection method is only appropriate for finite-number constraints, and cannot handle the infeasible initial policy. Chow et, al. (2015) instead adopt primal-dual optimization (PDO) method with constraints on conditional value at risk (CVaR) [8]:

$$\min_{v \in \mathbb{R}} \left\{ v + \frac{1}{1-\zeta} \mathbb{E}_{s \sim d(s), a \sim \pi_k} \left[\left(r_{C_i} - v \right)^+ \right] \right\} \leq d_i \tag{4}$$

where v is a balance coefficient between reward and cost, and PDO adjusts it during exploration to handle infeasibility issue. The confidential level ζ is a pre-defined hyperparameter. PDO cannot guarantee the constraint satisfaction during training. Later, the famous constrained policy optimization (CPO) algorithm is proposed, which firstly claims to guarantee safe exploration [9]. The constraints formulation is degenerated to a basic one-hot chance-constraint added with a distance constraint:

$$\overline{D_p}\left(\pi_k, \pi_{k+1}\right) \approx \frac{1}{2} \Delta\theta^T H \Delta\theta < \delta \tag{5}$$

where $\overline{D_p}$ is a distance measurement. In practice, $\overline{D_p}$ is replaced with the KL divergence with second-order Taylor approximation, H is the Fisher information matrix. CPO also designs a retrieval mechanism, which aims to only decrease constraints value to compensate the infeasibility caused by approximation error.

$$\theta_{k+1} = \theta_k - \sqrt{\frac{2\delta}{b^T H^{-1} b}} H^{-1} b \tag{6}$$

where b is the derivative of cost with respect to the policy parameters, i.e., $\partial J_{C_i} / \partial \theta$. CPO is regarded as a commonly used baseline of model-free safe RL. A typical model-based policy optimization (MBPO) for constrained RL is proposed by Duan et, al. (2019) adopts multi-step rollout to confine policy update, where the constraints are separately posed on *each* rollout step [13]:

$$J_{C_i}\left(\pi_k\right) = \mathbb{E}_{a \sim \pi_k} \left\{ r_{C_i}\left(s_{t+i}, a\right) \right\} \leq d_i \tag{7}$$

where $i \in 1, 2, \ldots N$, and $\forall s_t$ in the feasible state set, i.e., the feasible region. Each policy update needs to a N-steps model rollout. The comparison between three typical algorithms is listed in TABLE. I.

TABLE I: Typical CAC algorithms for CMDP

Algorithms	Constraints formulation	Solution technique	Method to handle Infeasibility
PGP	Linearized chance constraint	Rosen projection	restoration projection
PDO	Conditional value at risk	Dual ascent	Adaptive trade-off coefficients
CPO	Linearized chance & distance constraint	Approximate Lagrangian	Retrieval policy update & reward shaping
MBPO	Model-based statewise constraint & distance constraint	Approximate Lagrangian	Retrieval policy update

C. Generalized Control Barrier Function

We define a feasible state set with respect to real-world safety requirements:

$$\mathcal{C} = \{s \mid h(s) \leq 0\} \tag{8}$$

Aforementioned methods all directly adopts constraints formulation with $h(\cdot) \leq 0$. In contrary, control barrier function (CBF) adopts a more concise formulation to address the feasibility theoretically. Consider a general discrete-time dynamical system is

$$s_{t+1} = f(s_t, a_t) \tag{9}$$

The theory of CBF is based on control forward invariance:

Definition 1. *The set $\mathcal{C}$ is controlled forward invariant if $\forall s_t \in \mathcal{C}$ there exists a policy such that $s_{t+i} \in \mathcal{C}$ or $i \in \{1, 2, \ldots, \infty\}$ with respect to system (9). The discrete-time CBF for a constraint $h(s_t) \leq 0$ is*

$$h(s_{t+1}) \leq (1-\alpha) h(s_t) \tag{10}$$

where α is the conservative coefficient. Intuitively, a larger α indicates that the constraints is less conservative.

Proposition 1 (discrete-time control barrier function)**.** *The set $\mathcal{C}$ is controlled forward invariant along the trajectories if and only if there exists a policy satisfying*

$$(h(s_{t+1}) - h(s_t)) + \alpha h(s_t) \leq 0, \forall s_t \in \mathcal{C} \tag{11}$$

with a scalar $\alpha \in (0, 1]$.

Proof. $h(s_{t+i}) \leq (1-\alpha)^i h(s_t) \leq 0, \cdot \forall i \in \mathbb{Z}_+$. A detailed proof is presented in Agrawal's study [19]. □

A major drawback of the original formulation is the implicit condition that the relative-degree of constraints and input must be 1, which causes that standard CBF cannot be implemented on all dynamic systems. The relative-degree is defined as which order derivative of constraints is relevant with the control input, i.e.,

Definition 2 (High relative-degree constraints)**.** *The constraint has relative-degree m with respect to control input if*

$$\frac{dh(s_{t+m})}{ds_{t+m}} \frac{df(s_{t+i-1}, a_{t+i-1})}{da_t} = 0 \tag{12}$$

for $\forall i \in \{0, 1, \ldots m-1\}, \forall s_t \in \mathbb{R}^n$, with respect to system 9, $m \in \{2, 3, \ldots n\}$. If the above equality does not hold, the constraint has relative-degree 1.

In our previous work, we propose the generalized control barrier function to handle high relative-degree constraints is to pose constraints on the nonadjacent steps for a constraint function with relative-degree m.[21]

Definition 3. *For a constraint with relative degree m, the generalized control barrier function is*

$$h(s_{t+m}) \leq (1-\alpha)^m h(s_t), \forall k \in \mathbb{Z}_+ \tag{13}$$

The intuitive explanation is that the high-order derivatives is "flatten" on the time axis. In order to track the input, the constraint is posed between two nonadjacent steps. Detailed proof of feasibility guarantee with generalized control barrier function is provided in our previous work [21].

III. Algorithm Details

This section introduces how to confine policy update by GCBF including the problem formulation, the approximate update rules, and an adaptive conservativeness mechanism in case of an infeasible policy update.

A. Model-based Policy Optimization with GCBF

1) Problem formulation: Different with MBPO which utilizes multi-step infomation for one-step actor-critic update, the proposed GCBF-MBPO reduces to a small number of the required information steps which is the relative-degree of constraint. The critic and actor need to be updated during the policy optimization. Defining the return as $\sum_{j=t}^{t+m} \gamma^{j-t} r(s_j, \pi(s_j; \theta)) + \gamma^m V(s_{t+m+1}; w)$, the critic loss is

$$L(w) = \mathbb{E}_{s_t \sim \mathcal{C}} \left\{ \frac{1}{2} (G - V(s_t; w))^2 \right\} \tag{14}$$

and the gradient of critic is

$$\frac{dL}{dw} = \mathbb{E}_{s_t \sim \mathcal{C}} \left\{ (G - V(s_t; w)) \frac{dV(s_t; w)}{dw} \right\} \tag{15}$$

The critic update has not changed compared to the unconstrained version. The actor update stage considers the GCBF constraints, where the loss and constraints are listed as:

$$\begin{aligned} J(\theta) &= \mathbb{E}_{s \sim \mathcal{C}, a \sim \pi(\theta)}\{G\} \\ J_{C_i}(\theta) &= \mathbb{E}_{a \sim \pi(\theta)}[h_i(s_{t+m})] \leq (1-\alpha)^m h_i(s_t) \end{aligned} \tag{16}$$

Note that here the $J_{C_i}(\theta)$ is calculated by m-steps rollout with models.

Proposition 2. *For a constraint with relative-degree m, the model-based constrained policy optimization should rollout at least m steps.*

Proof of Prop. 2. Assume a constraint $J_{C_i}(\theta)$ is defined with an expectation of q-steps rollout smaller than m, the

gradient of constraints with respect to actor parameters are

$$\begin{aligned}\frac{\mathrm{d}J_{C_i}}{\mathrm{d}\theta} &= \mathop{\mathbb{E}}_{s_t\sim\mathcal{C}}\left\{\frac{\mathrm{d}h_{C_i}\left(s_{t+q}\right)}{\mathrm{d}\theta}\right\} \\ &= \mathop{\mathbb{E}}_{s_t\sim\mathcal{C}}\left\{\sum_{j=t}^{t+q}\frac{\partial h_{C_i}\left(s_{t+q}\right)}{\partial s_{t+q}}\left[\phi_{j-t}+\psi_{j-t}\right]\right\}\end{aligned} \tag{17}$$

where

$$\phi_{i+1} = \begin{cases} 0 & , i=-1 \\ \frac{\partial f(s_{t+i},a_{t+i})}{\partial s_{t+i}}\phi_i + \frac{\partial f(s_{t+i},a_{t+i})}{\partial a_{t+i}}\psi_i & , \text{ else} \end{cases}$$

$$\psi_{i+1} = \frac{\partial\pi\left(s_{t+i};\theta\right)}{\partial s_{t+i}}\phi_i + \frac{\partial\pi\left(s_{t+i};\theta\right)}{\partial\theta}$$

According to Definition 2, Each iterative item of ψ_i is equal to zero, and $\frac{\mathrm{d}J_{C_i}}{\mathrm{d}\theta}=0$. Therefore, if the rollout step is less than m, the input fails to affect constraints cost, and the constraints costs can not be optimized. □

2) Approximate Solution for Constrianed Policy Gradient: The gradient $\Delta\theta$ to update actor must satisfy (16). We implement the approximate solution technique by linearized objective and constraints added with a distance constraints.

$$\begin{aligned}&\min_{\Delta\theta}\ g^T\Delta\theta \\ &\text{s.t. } z + C^T\Delta\theta \le 0 \\ &\qquad \overline{D_p}\left(\theta;\theta_k\right) \approx \frac{1}{2}\Delta\theta^T H\Delta\theta \le \delta\end{aligned} \tag{18}$$

where $g = \frac{dJ}{d\theta}/\left\|\frac{dJ}{d\theta}\right\|^2, \cdot z_i = \left(J_{C_i}|_{\theta_k} - (1-\lambda)^m h\left(s_k\right)\right)$, $C_i = \frac{dJ_{C_i}}{d\theta}/\left\|\frac{dJ_{C_i}}{d\theta}\right\|^2$. With $C \doteq [c_1, c_2, \ldots, c_M]$ and $z \doteq [z_1, z_2, \ldots, z_M]$, the the analytical solution of (18) can be analytically solved by Lagrange multiplier method. The Lagrange function and analytical solution are

$$\begin{aligned}&L(\Delta\theta,\lambda,v) \\ &\quad = g^T\theta + \lambda\left(\frac{1}{2}\Delta\theta^T H\Delta\theta - \delta\right) + v(z + C^T\Delta\theta) \\ &\Delta\theta^* = \frac{H^{-1}\left(g+Cv^*\right)}{\lambda^*}\end{aligned} \tag{19}$$

where λ, ν is the dual variable, and λ^*, ν^* is the optimal dual solution which can be obtained by analytical solution (single-dimension constraint) or solvers (multi-dimension constraints).The original MBPO uses a multi-step rollout, for example, 10-steps setting in the original paper, as a constrained prediction horizon, while we only needs 3-steps information to finish a policy update. The following section will demonstrates that the efficiency improvement. The pseudocode is shown in Algorithm 1:

B. Adaptive Coefficient to Handle Feasibility Issue

As described in Prop. 1, the conservativeness coefficient α exists but we do not know the exact value. Intuitively, a more aggressive choice of α may lead to more reckless actions and fails to guarantee the set invariance. In order to find a proper conservativeness coefficient, we propose an adaptive updating rule of α, which adjusts the value according to the severity of constraints violation. We predict the constraints violation ξ from the trajectory $\mathcal{T}$:

$$\xi = \mathbb{E}_{\mathcal{T}}\sum_i\left[J_{C_i}(\pi) - d_i\right]^+ \tag{20}$$

if the constraints violation exceeds a pre-defined threshold, the conservativeness coefficient is adjusted to releases the constraints. We name the improved version with adaptive conservativeness coefficient as adaptive α GCBF-MBPO, which is shown in Algorithm 2:

Algorithm 1: GCBF-MBPO

Input: Feasible policy $\pi(\theta_0)$, constraint relative degree m, conservativeness coefficient α

for $k = 1, 2, \ldots$ **do**
- Sample a set of trajectories $\mathcal{D} = \{\tau\} \sim \pi_k = \pi\left(\theta_k\right)$
- From samples predicts g, b, H, c
- **if** *appximate update is feasible* **then**
  - Solve dual problem and update theta with (19)
- **else**
  - Compute recovery policy with (6)
- update critic with (15)

Algorithm 2: Adaptive α GCBF-MBPO

Input: Feasible policy $\pi(\theta_0)$, constraint relative degree m, conservativeness coefficient α, violation tolerance ξ_c

for $k = 1, 2, \ldots$ **do**
- Sample a set of trajectories $\mathcal{D} = \{\tau\} \sim \pi_k = \pi\left(\theta_k\right)$
- From samples predicts g, b, H, c, ξ
- **if** *appximate update is feasible* **then**
  - Solve dual problem and update theta with (19)
- **else**
  - Compute recovery policy with (6)
- update critic with (15)
- **if** $\xi > \xi_c$ **then**
  - $\alpha \leftarrow \alpha + \beta\xi$

IV. Experimental Results

Autonomous driving is a complex safety-critical sequential decision-making problem whose multi-objective orientation poses great challenges to decision and control systems [22][23]. Intersection is a complex scenarios for autonomous driving, which poses great challenges for vehicle control [24] [25]. In this section, we evaluate the proposed algorithms on a large-scale autonomous driving task in a two-way six-lane intersection to show the constraints violations reduction and efficiency improvements. We also implement our proposed algorithm on a real autonomous vehicle to verify the collision avoidance ability. The surrounding vehicles is generated

virtually by a digital twin system for the safety consideration.

Fig. 1: The autonomous vehicle collision avoidance with a digital twin system.

A. Experiment 1: Simulation

1) Problem Description: The autonomous driving task requires the agent to track the pre-defined reference path to pass the intersection without colliding into other vehicles or road margins. The intersection is demonstrated in Fig. 2, and the random traffic flow is generated by SUMO.

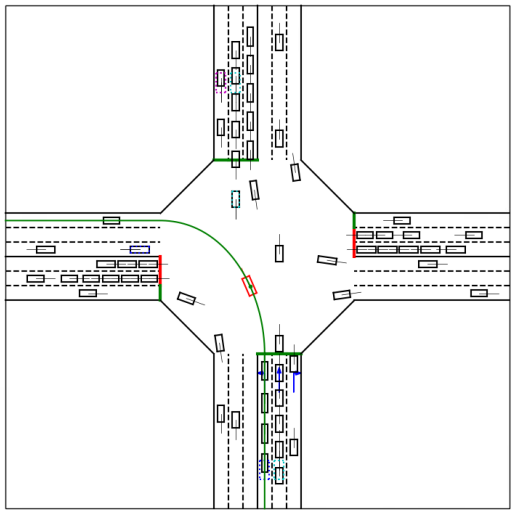


Fig. 2: The intersection for autonomous driving control task. A safety-gym third party environments repo refers to https://github.com/mahaitongdae/safe_exp_env.

The states include both states of ego vehicle, tracking error, and surrounding vehicles. All surroundings are filtered to 8 involved vehicles according to the distance to ego vehicle and each vehicle's goal lane. If the number of involved vehicles is less than 8, certain virtual vehicles are augmented with a distant location. The dimension of state space sums up to be 41, and the action includes desired acceleration and steering angle of the ego vehicle. Details are listed in TABLE II.

The reward function is formulated to track a static trajectory randomly selected to reach each destination lane:

$$\begin{aligned} r(s,a) = 0.05\left(v - v_{\text{target}}\right)^2 + 0.8\Delta y^2 + 30\Delta\phi^2 \\ +0.02r_y^2 + 5\delta^2 + 0.05a_{\text{Acc}}^2 \end{aligned} \tag{21}$$

TABLE II: State and Control Input

Ego vehicle state	Speed	(v_x, v_y)	[m/s]
	Yaw rate	r_y	[rad/s]
	Position	(x, y)	[m]
	Heading angle	ψ	[rad]
Tracking states	position error	$(\Delta x, \Delta y)$	[m]
	Heading angle error	$\Delta\psi$	[rad]
Surronding vehicle states	Position	(x_j, y_j)	[m]
	Velocity	v_j	[m/s]
	Heading angle	ψ_j	[rad]
Input	Steering anlge	δ	[rad]
	Acceleration	a_{Acc}	[m/s2]

The dynamic model is utilized to predict trajectory of ego and surrounding vehicles. The model of ego vehicle uses a numerically stable bicycle model [26]. As for the surrounding vehicles, a simple kinematics model with the uniform recurrence assumption is adopted. The target for each surrounding vehicle can be obtained from SUMO, which tells whether a vehicle prepares to go straight, turn left, or right. The states for position information are predicted with uniform recurrence driven by current speed, and the yaw angle is predicted by the constant-speed rotation, i.e.,

$$\begin{aligned} rx_i' &= x_i + v_i\cos\left(\phi_i\right)T \\ y_i' &= y_i + v_i\sin\left(\phi_i\right)T \\ \phi_i' &= \begin{cases} \phi_i & \text{if going straight} \\ \phi_i + \frac{v_i}{R^*}T & \text{if turning} \end{cases} \end{aligned} \tag{22}$$

where R^* is an estimated radius depending on size of the intersection demonstrated in Fig. 3. For instance, in the simulation scenario, the size of the intersection is 50 m, and the turning radius of right turn is 20 m, while the left turn is 30 m. The prediction model is not so perfect, but the results still show a considerable reduction of constraints violation.

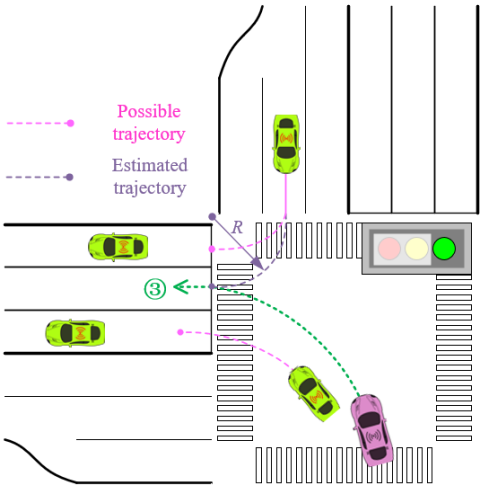


Fig. 3: Predicting surrounding vehicles.

The safety constraints include the collision avoidance and road margin. A two-circles safe distance constraint are implemented between ego vehicle and each vehicle:

$$\begin{aligned} \left(x^{\#} - x_j^*\right)^2 + \left(y^{\#} - y_j^*\right)^2 \geq d_{\text{safe}}^2 \\ \left(x^{\#} - x_{\text{road}}\right)^2 + \left(y^{\#} - y_{\text{road}}\right)^2 \geq d_{r_{\text{safe}}}^2 \end{aligned} \tag{23}$$

where (x^*, y^*) is the center of circles, and the subscripts $j \in 1, 2, \dots 8$ represents the index of surrounding vehicles. The up-scripts $\#, * \in \{f, r\}$ represents the front or rear safety circle as shown in Fig. 4. The road margin is also considered similar with the two-circles safety distance constraints, where the nearest point with respect to the road margin is represented by (x_{road}, y_{road}).

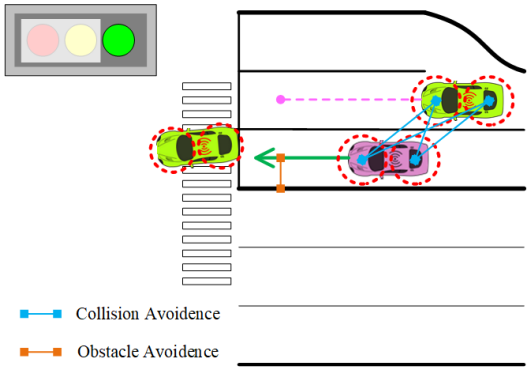


Fig. 4: Demonstration of state constraints.

2) Training Results: We compare our adaptive α GCBF-MBPO (Ada-GCBF-MBPO) and the original version (GCBF-MBPO) with model-based policy optimization with original constraints (MBPO) and model-free constrained policy optimization (CPO). The number of environment interactions is limited to 2 million. The hyperparameters are listed in TABLE III.

TABLE III: Algorithms Hyperparameters

Algorithms	Value
shared	
Optimizer	Conjugate gradient optimizer
Damping coefficient	0.1
Backtracking coefficient	0.8
Max backtracking iterations	10
Approximation function	Multi-layer perceptron
Number of hidden layers	2
Number of hidden units per layer	256
Nonlinearity of hidder layer	ELU
Nonlinearity of output layer	tanh
Critic learning rate	Linear Annealing 8e-5→ 8e-6
Discounted factor	0.99
GCBF-MBPO	
Conservativeness coefficient	0.3
Constraints relative-degree	3
Adaptive α GCBF-MBPO	
Initial α	0.1
Violation tolerance	0.3
α learning rate	1e-3
MBPO	
Constrained rollout steps	10

The average episode returns and episode constraints violation distance are chosen to evaluate the performance of algorithms. The average episode returns are defined with the expectation of episode returns and the feasibility performance, i.e., the constraints violation distance is calculated by for a trajectory $\mathcal{T}$:

$$\mathbb{E}_{\mathcal{T}} \sum_{j,\#,*} \left[d_{\text{safe}}^2 - \left(x^{\#} - x_j^*\right)^2 + \left(y^{\#} - y_j^*\right)^2 \right]^+ \tag{24}$$

where $[\cdot]^+$ represents the positive part, i.e., the violation level of the inequality constraints. The smaller constraints violation distance is, the better feasibility performance algorithm shows. The performance during the training procedure is shown in Fig. 4 and Fig. 5. Results show that the original version of GCBF-MBPO has already decreased the constraints violation by a considerable decent. The performance is not that stable, where lower constraints violations exist in the middle stages of training. The adaptive α mechanism can automatically handle the performance-feasibility balance and keep lower constraint violations throughout the training process.

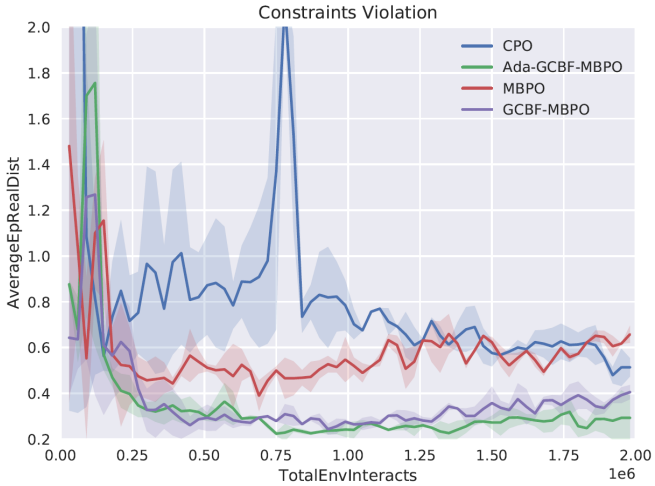


Fig. 5: Average episode constraints violation distance with different algorithms.

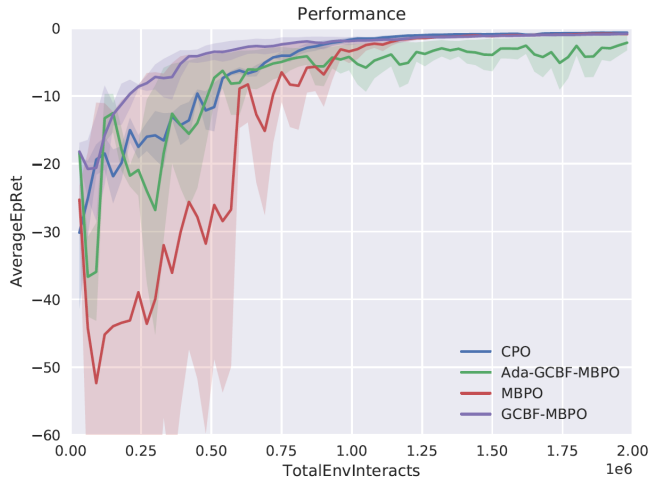


Fig. 6: Average episode return with different algorithms.

The detailed numbers of performance and constraints violation distance are shown in TABLE IV, which demonstrates that GCBF-MBPO can reduce the constraints violation during training from 24.14% to 73.83%, while the performance only changes in a reasonable range. Furthermore, it is easy to see the two GCBF-MBPO converges much faster than MBPO algorithms with respect to total environment interactions. We take the total environment interactions when the average episode return reaches several thresholds (-20,

-10, -5), the average environment interactions of two GCBF-MBPO is 3.36 times faster than MBPO.

TABLE IV: Algorithms Performance

Algorithms	Average Episode Constraints violation	Average Episode Return
Adaptive α GCBF-MBPO	0.169	-1.052
GCBF-MBPO	0.374	-0.769
MBPO	0.493	-0.785
CPO	0.646	-0.735

B. Experiment 2: Autonomous Vehicle

Limited by the regulations of autonomous driving test, we instead choose a two-lane intersection to demonstrate the vehicle experiment.

1) Hardware and Software Architectures: The autonomous vehicle is a Chang-An CS55 equipped with an on-board industrial PC as the controller. A digital twin-system is adopted to simulate virtual surrounding vehicles, and the information of the ego vehicle is also sent back to project the real vehicle in the virtual environment. The details of hardware and software architecture are shown in Fig. 7. Parallel structure is designed in the on-board PC, including neural-network-based controller and planner.

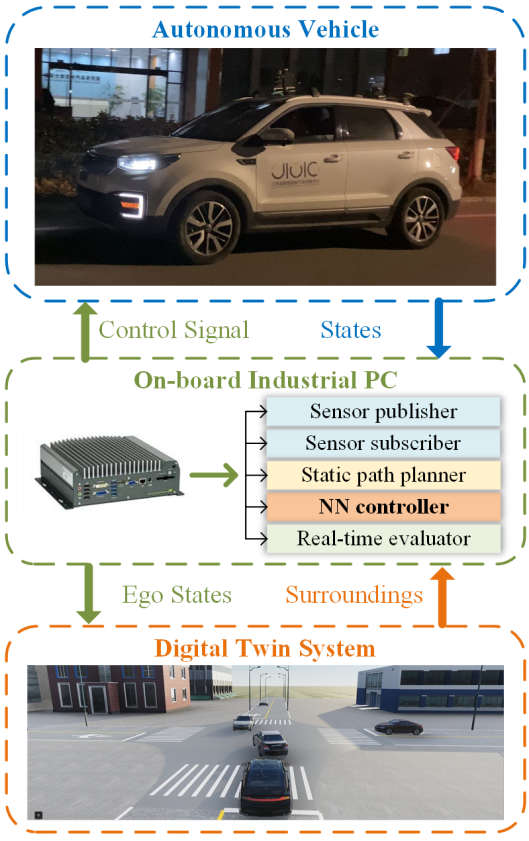


Fig. 7: Hardware and software architecture of autonomous vehicles.

2) Experiment Results: We select nine typical cases of surrounding vehicles with 3 cases for each destination to test the collision avoidance performance, shown in Fig. 8(a). We demonstrate the experiment from three perspectives, including real-world and virtual environments, as shown in Fig 1. The arrows represent the surrounding vehicle trajectories, and the indexes are the order to pass the intersection. The results are demonstrated in Fig. 8(b), which includes the time sequences to show the collision avoidance behaviors. Results show that trained policy learns multiple approaches for avoiding collision, including deceleration, accelerating, pulling up and wait, deviating the reference to bypass the vehicles, listed in TABLE V.

TABLE V: Collision Avoidance Behaviors

Destinations	Decelerating	Pulling up	Accelerating	Turning
Left	case 1,2	case 0	-	case1
Straight	case 0,1,2	case 2	-	case 0,1
Right	case 2	-	case 0	case 1

V. Conclusion

In this paper, we proposed a model-based feasibility enhancement technique of constrained RL. The policy optimization was confined using generalized control barrier function, and the model information was utilized to penalize actions that drive agents closer to the constraint boundary. By the proposed approach, learning a constraint-satisfying policy did not need to violate real-world safety constraints. Compared to the baseline model-based constrained policy optimization technique, the efficiency was improved to the maximum with proof for reducing the required sampling steps of each policy update. We further designed an adaptive conservativeness coefficient to handle the infeasibility issue. We evaluate the proposed framework on a collision avoidance task on both simulation scenarios and a real autonomous vehicle. Compared to baseline constrained RL, the constraints violation during training decreased by up to 73.83%, and the efficiency increased 3.36 times. We verified the algorithm functions on the actual autonomous driving vehicles, and the results showed that the policy learned multiple modals of behaviors to avoid collisions.

Although the proposed approach is able to enhance feasibility by model information, the constraints violation still happened due to mostly the approximate solution technique. In the future, we will develop proper solution techniques like augmented Lagrangian to improve the feasibility performance further.

Acknowledgment

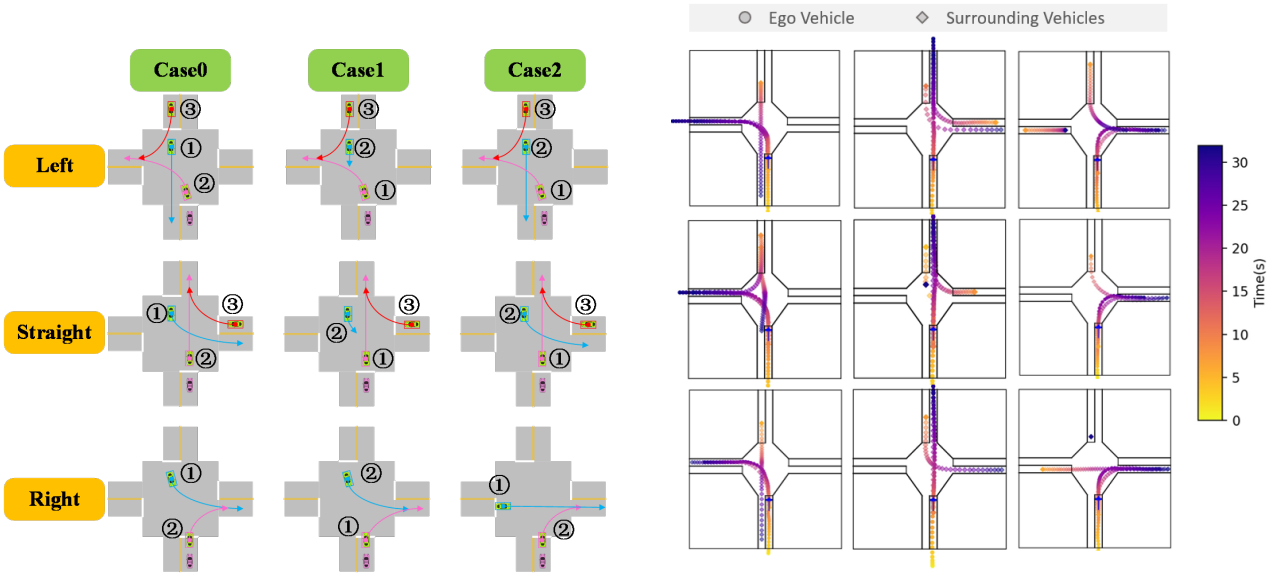
This study is supported by National Key R&D Program of China with 2020YFB1600200. This study is also supported by Tsinghua University-Toyota Joint Research Center for AI Technology of Automated Vehicle. The authors would like to thank Mr. Yang Guan, Mr. Yangang Ren, Mr. Wei Xu, and Prof. Bo Cheng for their valuable suggestions in the autonomous vehicle experiments.

References

[1] S. E. Li, *Reinforcement Learning and Control*. Tsinghua University Lecture Notes, 2020. [Online]. Available: http://www.idlab-tsinghua.com/thulab/labweb/publications.html

[2] D. Amodei, C. Olah, J. Steinhardt, P. Christiano, J. Schulman, and D. Mané, "Concrete problems in AI safety," *arXiv preprint arXiv:1606.06565*, 2016.

[3] E. Altman, *Constrained Markov decision processes*. CRC Press, 1999, vol. 7.



(a) Typical cases.

(b) Bird views.

Fig. 8: Autonomous vehicle experiments. A short movie is provided to demonstrate the avoidance behaviors on https://youtu.be/5oKOV_drY4o. We select 3 typical cases to demonstrates the autonomous driving vehicle is able to learn avoiding collision by pulling up, decelerating, accelerating and turning. Three perspectives are recorded including autonomous vehicle, steering wheel and digital twin system.

[4] J. García and F. Fernández, "A comprehensive survey on safe reinforcement learning," *Journal of Machine Learning Research*, vol. 16, pp. 1437–1480, 2015.

[5] R. T. Rockafellar and S. Uryasev, "Conditional value-at-risk for general loss distributions," *Journal of Banking and Finance*, vol. 26, no. 7, pp. 1443–1471, 2002.

[6] A. Ray, J. Achiam, and D. Amodei, "Benchmarking safe exploration in deep reinforcement learning," *arXiv preprint arXiv:1910.01708*, 2019.

[7] E. Uchibe and K. Doya, "Constrained reinforcement learning from intrinsic and extrinsic rewards," in *2007 IEEE 6th International Conference on Development and Learning*. IEEE, 2007, pp. 163–168.

[8] Y. Chow, M. Ghavamzadeh, L. Janson, and M. Pavone, "Risk-constrained reinforcement learning with percentile risk criteria," *Journal of Machine Learning Research*, vol. 18, pp. 1–51, 2018.

[9] J. Achiam, D. Held, A. Tamar, and P. Abbeel, "Constrained policy optimization," in *Proceedings of the 34th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, D. Precup and Y. W. Teh, Eds., vol. 70. International Convention Centre, Sydney, Australia: PMLR, 06–11 Aug 2017, pp. 22–31.

[10] Y. Guan, J. Duan, S. E. Li, J. Li, J. Chen, and B. Cheng, "Mixed policy gradient," *arXiv preprint arXiv:2102.11513*, 2021.

[11] Z. Lin, J. Duan, S. E. Li, H. Ma, Y. Yin, and B. Cheng, "Continuous-time finite-horizon adp for automated vehicle controller design with high efficiency," in *2020 3rd International Conference on Unmanned Systems (ICUS)*, 2020, pp. 978–984.

[12] F. Berkenkamp, M. Turchetta, A. P. Schoellig, and A. Krause, "Safe model-based reinforcement learning with stability guarantees," in *NIPS*, 2017, pp. 908–919.

[13] J. Duan, Z. Liu, S. E. Li, Q. Sun, Z. Jia, and B. Cheng, "Deep adaptive dynamic programming for nonaffine nonlinear optimal control problem with state constraints," *arXiv preprint arXiv:1911.11397*, 2019.

[14] T. Koller, F. Berkenkamp, M. Turchetta, and A. Krause, "Learning-based model predictive control for safe exploration," in *2018 IEEE Conference on Decision and Control (CDC)*. IEEE, 2018, pp. 6059–6066.

[15] M. Memarzadeh and M. Pozzi, "Model-free reinforcement learning with model-based safe exploration: Optimizing adaptive recovery process of infrastructure systems," *Structural Safety*, vol. 80, pp. 46–55, 2019.

[16] F. Blanchini and L. Weng, "Set invariance in control," *Automatica*, vol. 35, no. 11, pp. 1747–1767, 1999.

[17] S. Prajna, "Barrier certificates for nonlinear model validation," *Automatica*, vol. 42, no. 1, pp. 117–126, 2006.

[18] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, "Control barrier functions: Theory and applications," in *2019 18th European Control Conference (ECC)*, 2019, pp. 3420–3431.

[19] A. Agrawal and K. Sreenath, "Discrete control barrier functions for safety-critical control of discrete systems with application to bipedal robot navigation." in *Robotics: Science and Systems*, 2017.

[20] Q. Nguyen and K. Sreenath, "Exponential control barrier functions for enforcing high relative-degree safety-critical constraints," in *2016 American Control Conference (ACC)*, 2016, pp. 322–328.

[21] H. Ma, X. Zhang, S. E. Li, Z. Lin, Y. Lyu, and S. Zheng, "Feasibility enhancement of constrained receding horizon control using generalized control barrier function," *arXiv preprint arXiv:2102.13304*, 2021.

[22] S. Li, K. Li, R. Rajamani, and J. Wang, "Model predictive multi-objective vehicular adaptive cruise control," *IEEE Transactions on Control Systems Technology*, vol. 19, no. 3, pp. 556–566, 2011.

[23] S. E. Li, Z. Jia, K. Li, and B. Cheng, "Fast online computation of a model predictive controller and its application to fuel economy-oriented adaptive cruise control," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 3, pp. 1199–1209, 2015.

[24] Y. Guan, Y. Ren, S. E. Li, Q. Sun, L. Luo, and K. Li, "Centralized cooperation for connected and automated vehicles at intersections by proximal policy optimization," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 11, pp. 12 597–12 608, 2020.

[25] Y. Ren, J. Duan, S. E. Li, Y. Guan, and Q. Sun, "Improving generalization of reinforcement learning with minimax distributional soft actor-critic," in *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, 2020, pp. 1–6.

[26] Q. Ge, S. E. Li, Q. Sun, and S. Zheng, "Numerically stable dynamic bicycle model for discrete-time control," *arXiv preprint arXiv:2011.09612*, 2020.