Model-based Constrained Reinforcement Learning using Generalized Control Barrier Function

Haitong Ma[‡], Jianyu Chen[†,||], Shengbo Eben Li[*‡], Ziyu Lin[‡], Yang Guan[‡], Yangang Ren[‡], and Sifa Zheng[‡]

Abstract— **Model information can be used to predict future trajectories, so it has huge potential to avoid dangerous regions when applying 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 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 when applying reinforcement learning (RL) to real-world tasks [1]. For instance, the collision of autonomous vehicle control must be avoided in case of causing physical harm to humans [2]. A safety-critical reinforcement learning problem is generally formulated to a constrained reinforcement learning problem, which aims to maximize the total reward while satisfying the safety constraints [3], [4].

Multiple definitions of the safety constraints are integrated with constrained RL studies. The one-hot formulation is usually used for cost function, a similar signal with reward obtained from environment. The one-hot cost function is commonly transferred to a chance constraint [5]. Both average cost-based constrained and accumulative cost constraints are considered in different algorithms [6], [7]. Value at risk measures risk as the maximum possible cost with a pre-defined confidence level [8]. Conditional value at risk (CVaR) is further designed to address those cases whose probability is small, usually used in portfolio optimization [9]. Both of them are designed with long-horizon data-driven expectation, which is the inevitable choice for model-free RL. The drawback is that existing model-free RL can only learn a safe policy by inevitably experiencing constraints violations through trial-and-error, which imposes significant safety issues, especially during exploration [10].

[‡]School of Vehicle and Mobility, Tsinghua University. Email: {maht19@mails., lishbo@, linzy17@mails., guany17@mails., ryg18@mails., zsf@}tsinghua.edu.cn.
[†] Institute for Interdisciplinary Information Sciences, Tsinghua University. Email: jianyuchen@tsinghua.edu.cn.
[||]Shanghai Qizhi Institute.
[*]All correspondence should be sent to S. Li.

Some existing constrained RL methods deploy model information to obtain a constraint-satisfying policy. Most existing studies aim to find a constrained optimal policy while adopting constraints on *every time step* in the prediction horizon with model rollout [11], [12]. Some learning-based controllers share the similar idea with multi-step rollout with model and constraints on each time step [13]. The major problem of multi-step model rollout is that the prediction becomes inaccurate with the increase of rollout steps. Moreover, a multi-step rollout uses too much sampling information to finish the constrained optimization, and the sampling efficiency is significantly decreased.

In this paper, we propose a model-based constrained reinforcement learning approach with the generalized control barrier function. Intuitively, applying the control barrier function can handle state constraints by penalizing the trends of getting closer to the constraint boundary [1]. The proposed GCBF constraints are only considered within one or a few prediction steps, so the sampling efficiency increases, and the issue of prediction inaccuracy is avoided. We apply the approximate Lagrangian solution technique to compute the constrained policy gradient, and an adaptive mechanism is further added to automatically choose a appropriate parameters to improve the constraint-satisfying performance. 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 with proof. 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 constrained RL and generalized control barrier function. Section III introduces the proposed model-based constrained RL algorithms and the adaptive mechanism to choose GCBF's parameters. Section IV demonstrates the experiment results on the simulation platform and a real autonomous vehicle. Section V concludes the paper.

II. Preliminaries

A. Constrained Reinforcement Learning

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

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

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

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

where $i = 1, 2, \ldots, k$ is the constraint index. Each J_{C_i} is the expected cost, and d_i is the corresponding safety threshold. Some recent constrained RL studies are based on the actor-critic architecture integrated with the "constrained policy optimization" technique. The actor update progress is modified to find a constraint-satisfying policy, and the critic update is the same as existing state-value RL algorithms like trust-region policy optimization (TRPO) [14], [15].

B. Formulations of Inequality Constraints

Constraint formulations directly affect the safety performance, which is critical in constrained RL. An early constrained RL algorithm, i.e., the policy gradient projection (PGP), whose constraints formulation is based on average cost [6]:

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

where r_{C_i} is the corresponding constraint cost in a one-hot formulation, where a constraint-violation action gets a cost of one. The average reward design is not able to handle the unsafe action with a low probability. Chow et al. (2015) instead adopt constraints on conditional value at risk (CVaR) in the primal-dual optimization (PDO) [4]:

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

The confidential level ζ is a pre-defined hyperparameter, v is a balance coefficient between reward and cost. CVaR is about to address the actions in low probability but severer consequences. However, the balancing parameters design still accepts some constraints violations, which is not appropriate for the safety-critical problems. Later, the famous constrained policy optimization (CPO) algorithm is proposed, which firstly claims to guarantee safe exploration [3]. The constraints formulation is the accumulative constraint costs with a trust-region constraint to bound the constraint performance:

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

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 is

regarded as a commonly used baseline of model-free safe RL.

A typical model-based policy optimization (MBPO) for constrained RL proposed by Duan et al. (2019) adopts multi-step rollout to confine policy update, where the constraints are separately posed on *each rollout step* [11]:

$$J_{C_i}(\pi_k) = \mathbb{E}_{a \sim \pi_k} \{r_{C_i}(s_{t+i}, a)\} \le d_i \tag{6}$$

where $i \in 1, 2, \ldots N$, and $\forall s_t$ in the safe state set. Each policy update needs an N-steps model rollout. The comparison between four typical algorithms is listed in TABLE. I. CPO and MBPO are chosen as the baselines of our proposed algorithms.

TABLE I: Constraint Formulations of Typical Algorithms

Algorithms	Constraints formulation
PGP	Average cost constraint
PDO	Conditional value at risk
CPO	Accumualtive cost constraints & trust region
MBPO	Model-based statewise constraint & trust region

In summary, the cost-based constraints usually adopted in model-free constrained RL are learned with experiencing the constraint violations, which causes significant safety issues. The model-based approaches pose constraints based on the multi-step rollout, which causes problems with the low sampling efficiency and inaccuracy prediction in the future rollout steps. All of these issues block the performance improvement of existing constrained RL.

C. Generalized Control Barrier Function

Aforementioned methods all directly adopts constraints formulation with $h(\cdot) \le 0$. On the contrary, control barrier function (CBF) adopts a more concise formulation. Control barrier function is proposed to address safety with dynamic systems, also called the safety barrier certificate [16], [17]. We define a safe state set concerning real-world safety requirements:

$$\mathcal{C} = \{s \mid h(s) \le 0\} \tag{7}$$

Consider a general discrete-time dynamical system:

$$s_{t+1} = f(s_t, a_t) \tag{8}$$

Definition 1 (Control barrier function)**.** *The discrete-time control barrier function (CBF) for a constraint $h(s_t) \le 0$ is*

$$h(s_{t+1}) \le (1 - \alpha) h(s_t) \tag{9}$$

where α is the conservativeness coefficient.

For a constrained set $\mathcal{C}$ with a CBF constraint is satisfied for all states, the set can be guaranteed safe with respect to the system (8). Intuitively, control barrier function can be explained by confining the trend of getting closer to the constraint boundary shown in Fig. 1. A larger α indicates that the constraints are less conservative.

A major drawback of the original formulation is that it cannot be applied on high relative-degree dynamic systems [18], [19]. The relative-degree is defined as which order derivative of constraints is relevant with the control input, i.e.,
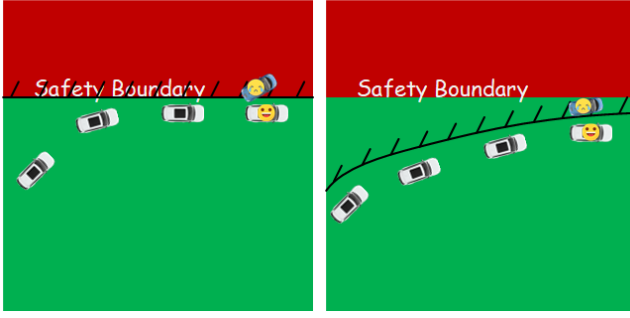
(a) Traditional pointwise constraints   (b) Control barrier function.

Fig. 1: Intuitive explanation of control barrier functions.

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

$$\frac{dh\left(s_{t+m}\right)}{ds_{t+m}}\frac{df(s_{t+i-1},a_{t+i-1})}{da_t} = 0 \quad (10)$$

for $\forall i \in \{0,1,\dots m-1\}, \forall s_t \in \mathbb{R}^n$, with respect to system ()8), $m \in \{2,3,\dots n\}$. 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 arbitrary relative-degree m.

Definition 3 (Generalized Control Barrier Function). *For a constraint with relative degree m, the generalized control barrier function is*

$$h\left(s_{t+m}\right) \leq (1-\alpha)^m h\left(s_t\right), \forall k \in \mathbb{Z}_+ \quad (11)$$

The intuitive explanation is that the high-order derivatives are "flatten" on the time axis. In order to track the input, the constraint is posed between two nonadjacent steps. Details about discrete-time control barrier function are provided in our previous work [20].

III. ALGORITHM DETAILS

This section introduces how to confine policy updates by GCBF, including the problem formulation, the approximate update rules, and an adaptive conservativeness mechanism to correct the parameters in control barrier function.

A. Model-based Policy Optimization with GCBF

1) Problem formulation: A reinforcement learning algorithm is to optimize the expected returns. The critic and actor need to be updated during the policy optimization. Defining the return G as $\sum_{j=t}^{t+m} \gamma^{j-t} r\left(s_j, \pi\left(s_j;\theta\right)\right) + \gamma^{m+1} V\left(s_{t+m+1};w\right)$, the actor update stage is a constrained optimization with the GCBF constraints, where the optimization problem is:

$$\min_{\Delta\theta} \ J_r(\theta) = \mathbb{E}_{s\sim\mathcal{C},a\sim\pi(\theta)}\{G\}$$
$$\text{s.t. } J_{C_i}(\theta) = \mathbb{E}_{a\sim\pi(\theta)}\left[h_i\left(s_{t+m}\right)\right] \quad (12)$$
$$\leq (1-\alpha)^m h_i\left(s_t\right)$$

Note that the $J_{C_i}(\theta)$ is calculated by m-steps rollout with models. 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 need m-steps information to finish a policy update. The following section will demonstrate that the efficiency improvement.

Proposition 1 (Least Required Sampling Steps). *For a constraint with relative-degree m, the model-based constrained policy optimization should rollout at least m steps.*

The proof is provided in Appendix. The critic update rule is similar to the unconstrained version, where the critic loss is defined as

$$L(w) = \mathbb{E}_{s_t\sim\mathcal{C}}\left\{\frac{1}{2}\left(G - V\left(s_t;w\right)\right)^2\right\} \quad (13)$$

and the gradient of critic is

$$\frac{dL}{dw} = \mathbb{E}_{s_t\sim\mathcal{C}}\left\{\left(G - V\left(s_t;w\right)\right)\frac{dV\left(s_t;w\right)}{dw}\right\} \quad (14)$$

The critic update has not changed compared to the unconstrained version.

2) Approximate Solution for Constrained Policy Gradient: The gradient $\Delta\theta$ to update actor must satisfy (12). We implement the approximate solution technique by linearized objective and constraints added with a distance constraint.

$$\min_{\Delta\theta} \ g^T\Delta\theta$$
$$\text{s.t. } z + C^T\Delta\theta \leq 0 \quad (15)$$
$$\overline{D_p}\left(\theta;\theta_k\right) \approx \frac{1}{2}\Delta\theta^T H\Delta\theta \leq \delta$$

where $g = \frac{dJ_r}{d\theta}/\left\|\frac{dJ_r}{d\theta}\right\|^2$, $z_i = \left(J_{C_i}|_{\theta_k} - (1-\lambda)^m h\left(s_k\right)\right)$, $C_i = \frac{dJ_{C_i}}{d\theta}/\left\|\frac{dJ_{C_i}}{d\theta}\right\|^2$. With $C \doteq [c_1, c_2,\dots,c_M]$ and $z \doteq [z_1, z_2,\dots,z_M]$, the the analytical solution of (15) can be analytically solved by Lagrange multiplier method. The Lagrange function are

$$L(\Delta\theta,\lambda,v) = g^T\Delta\theta + \lambda\left(\frac{1}{2}\Delta\theta^T H\Delta\theta - \delta\right) + v(z + C^T\Delta\theta) \quad (16)$$

where λ, ν is the dual variable. The analytical optimal solution is

$$\Delta\theta^* = \frac{H^{-1}\left(g + Cv^*\right)}{\lambda^*} \quad (17)$$

where λ^*, ν^* is the optimal dual solution obtained by analytical solution (single-dimension constraint) or solvers (multi-dimension constraints). If the problem does not have a feasible solution, the policy update rule changes to a retrieval mechanism:

$$\theta_{k+1} = \theta_k - \sqrt{\frac{2\delta}{b^T H^{-1} b}} H^{-1} b \quad (18)$$

The pseudocode is shown in Algorithm 1.

Algorithm 1: GCBF-MBPO

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

1 **for** $k = 1, 2, \ldots$ **do**
2   Sample a set of trajectories
      $\mathcal{D} = \{\tau\} \sim \pi_k = \pi(\theta_k)$
3   From samples predicts g, b, H, c
4   **if** *appximate update is feasible* **then**
5   Solve dual problem and update theta with (17)
6   **else**
7   Compute recovery policy with (18)
8   Update critic with (14)

B. Adaptive Conservativeness Mechanism

Intuitively, a more conservative choice of α in CBF may lead to more retrieval updates and affects the constraint-satisfying performance. To find a proper conservativeness coefficient, we propose an adaptive updating rule of α, which adjusts the value according to the severity of violations of the GCBF constraint. We predict the constraints violation ξ from the trajectory $\mathcal{T}$:

$$\xi = \mathbb{E}_{\mathcal{T}} \sum_i [J_{C_i}(\pi) - d_i]^+ \tag{19}$$

If the constraints violation exceeds a pre-defined threshold, the conservativeness coefficient is adjusted to releases the constraints with a learning rate β. We name the modified version with adaptive conservativeness coefficient as adaptive α GCBF-MBPO, shown in Algorithm 2.

Algorithm 2: Adaptive α GCBF-MBPO

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

1 **for** $k = 1, 2, \ldots$ **do**
2   Sample a set of trajectories
      $\mathcal{D} = \{\tau\} \sim \pi_k = \pi(\theta_k)$
3   From samples predicts g, b, H, c, ξ
4   **if** *appximate update is feasible* **then**
5   Solve dual problem and update theta with (17)
6   **else**
7   Compute recovery policy with (18)
8   Update critic with (14)
9   **if** $\xi > \xi_c$ **then**
10         $\alpha \leftarrow \alpha + \beta\xi$

IV. EXPERIMENTAL RESULTS

Autonomous driving is a complex safety-critical sequential decision-making problem with multi-objective orientation, which poses great challenges to decision and control systems [21][22]. The intersection is a complex scenario for autonomous driving, where collision avoidance is the major

safety concern [23] [24]. This section evaluates 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 apply our proposed algorithm to a real autonomous vehicle to verify the collision avoidance ability. The surrounding vehicles are generated virtually by a digital twin system for the safety consideration shown in Fig. 2.

Fig. 2: 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. 3, and the random traffic flow is generated by SUMO. The prediction on surrounding vehicles are assumed by approximately following the path generated by the SUMO planner.
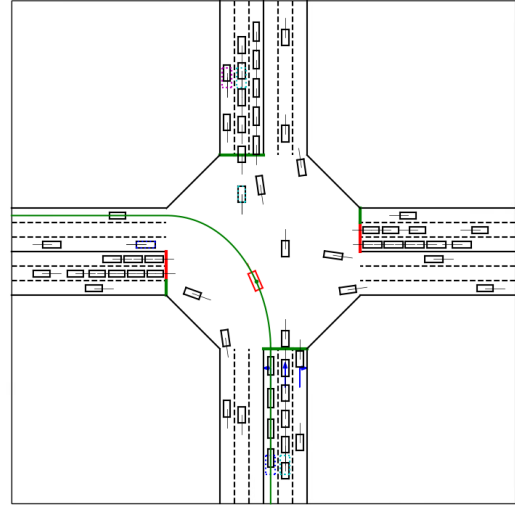

Fig. 3: The intersection for autonomous driving control task. We wrap the scenario as a safety-gym third party environment, the code repo is on https://github.com/mahaitongdae/safe_exp_env.

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

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]
Surrounding 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]

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:

$$r(s, a) = 0.05 \left(v - v_{\text{target}}\right)^2 + 0.8\Delta y^2 + 30\Delta\phi^2 \\ + 0.02r_y^2 + 5\delta^2 + 0.05a_{\text{Acc}}^2 \quad (20)$$

The model of ego vehicle uses a numerically stable dynamic bicycle model [25]. As for the surrounding vehicles, a simple kinematics model 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.,

$$x_i' = x_i + v_i \cos(\phi_i) T \\ y_i' = y_i + v_i \sin(\phi_i) T \\ \phi_i' = \begin{cases} \phi_i & \text{if going straight} \\ \phi_i + \frac{v_i}{R^*}T & \text{if turning} \end{cases} \quad (21)$$

where R^* is an estimated radius depending on the intersection's size demonstrated in Fig. 4. For instance, in the simulation scenario, the intersection's size is 50 m, and the turning radius of the right turn is 20 m, while the left turn is 30 m. Both ego and surroundings model are not perfect, but the results section will show a considerable reduction of constraints violation.

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

$$\left(x^\# - x_j^*\right)^2 + \left(y^\# - y_j^*\right)^2 \geq d_{\text{safe}}^2 \\ \left(x^\# - x_{\text{road}}\right)^2 + \left(y^\# - y_{\text{road}}\right)^2 \geq d_{r_{\text{safe}}}^2 \quad (22)$$

where (x^*, y^*) is the center of circles, and the subscripts $j \in 1, 2, \ldots 8$ represents the index of surrounding vehicles. The up-scripts $\#, * \in \{f, r\}$ represents the front or rear safety circle as shown in Fig. 5. The road margin is also considered similar to the two-circles safety distance constraints, where the nearest point to the road margin is represented by (x_{road}, y_{road}).
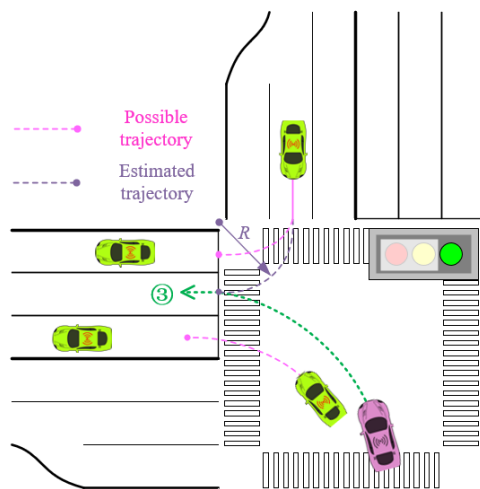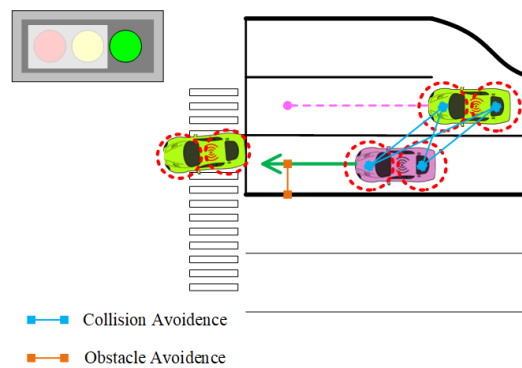

Fig. 4: Predicting surrounding vehicles.

Fig. 5: 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.

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 $\mathcal{T}$:

$$\mathbb{E}_{\mathcal{T}} \sum_{j, \#, *} \left[d_{\text{safe}}^2 - \left(x^\# - x_j^*\right)^2 - \left(y^\# - y_j^*\right)^2\right]^+ \quad (23)$$

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

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$\rightarrow$ 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

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.
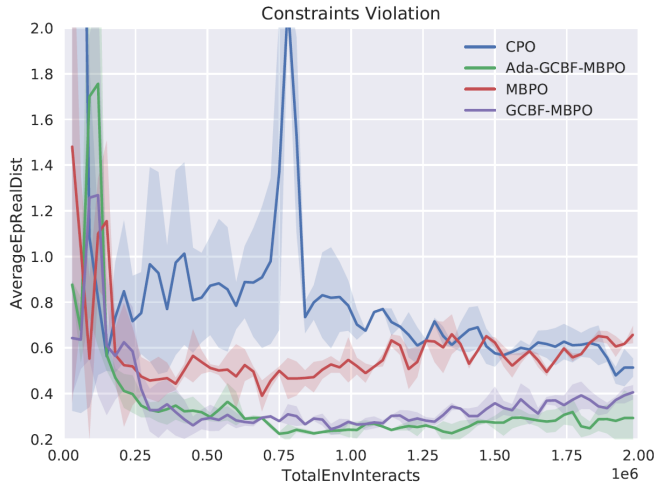

Fig. 6: Average episode constraints violation distance with different algorithms.

The exact 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 converge 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 are 3.36 times faster than MBPO.
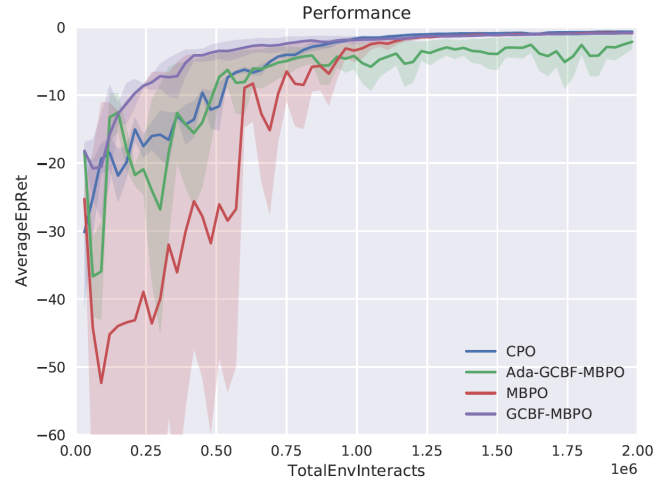

Fig. 7: Average episode return with different algorithms.

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: Real Vehicle Test

Limited by the autonomous driving test regulations, we instead choose a two-lane intersection to perform the real 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 surrounding virtual vehicles. 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. 8. Parallel structure is designed in the on-board PC. The autonomous driving architecture is based on the integrated decision and control (IDC) framework, which use both the value and policy network trained by RL to achieve interpretable and efficient decision and control.[26]

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. 9(a). We demonstrate the experiment from three perspectives, including real-world and virtual environments, as shown in Fig 2. The arrows represent the surrounding vehicle trajectories, and the indexes are the order to pass the intersection. The results are demonstrated in Fig. 9(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.
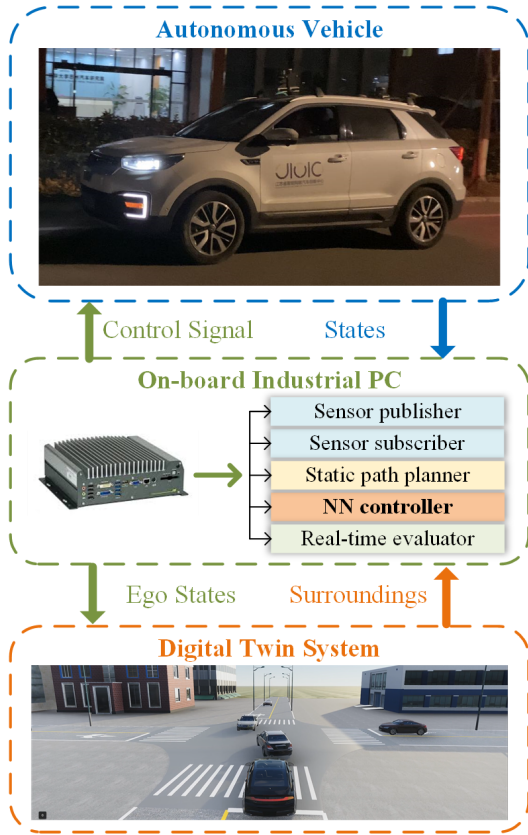
Fig. 8: Hardware and software architecture of autonomous vehicles.

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 constrained policy optimization technique with the generalized control barrier function. 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 a proof for reducing each policy update's required sampling steps. We further designed an adaptive conservativeness coefficient to handle the infeasibility issue. We evaluate the proposed framework on a collision avoidance task on 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 can improve constraint-satisfying performance by model information, the constraints violations still happened due to the approximate solution technique. In the future, we will develop proper solution techniques like augmented Lagrangian to improve the feasibility performance further. We also use a static prediction model for surrounding vehicles, and the realistic behavior of surrounding vehicles will introduce critical uncertainty factor to safety consideration. In the future, we consider handling the surroundings uncertainty by adopting the worst-case safety constraint.

APPENDIX

Proof of Prop. 1. 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

$$
\begin{aligned}
\frac{\mathrm{d}J_{C_i}}{\mathrm{d}\theta} &= \mathop{\mathbb{E}}_{s_t \sim \mathcal{C}} \left\{ \frac{\mathrm{d}h_{C_i}(s_{t+q})}{\mathrm{d}\theta} \right\} \\
&= \mathop{\mathbb{E}}_{s_t \sim \mathcal{C}} \left\{ \sum_{j=t}^{t+q} \frac{\partial h_{C_i}(s_{t+q})}{\partial s_{t+q}} [\phi_{j-t} + \psi_{j-t}] \right\}
\end{aligned}
\tag{24}
$$

where

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

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

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. □

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] 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.

[4] 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.

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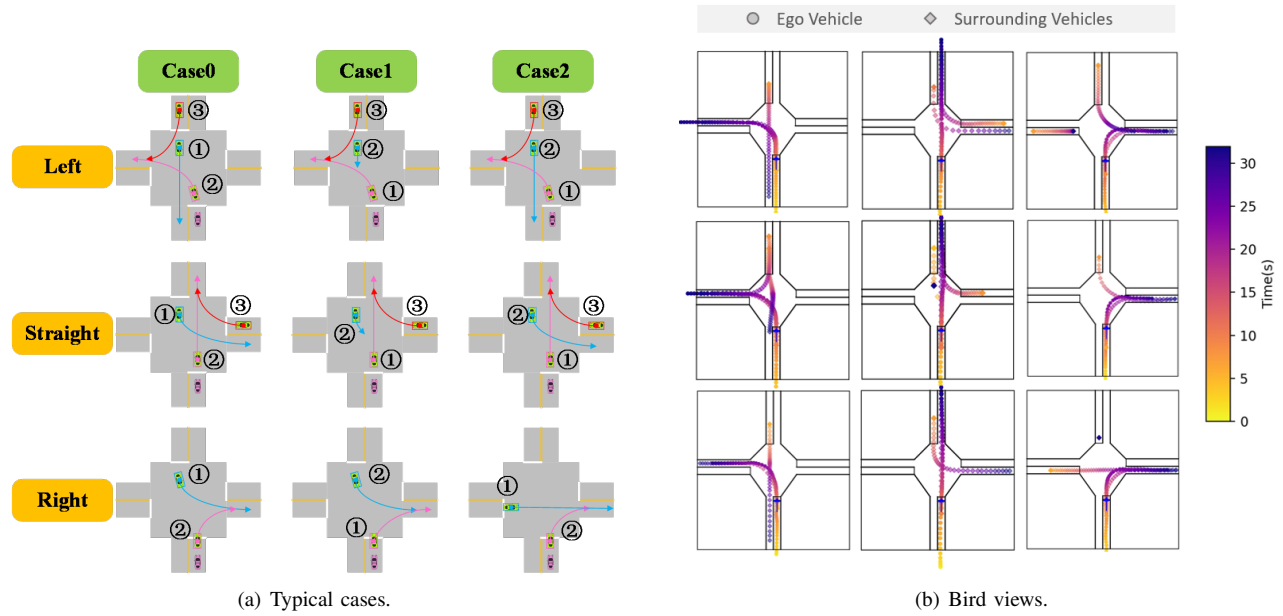
(a) Typical cases.

(b) Bird views.

Fig. 9: Autonomous vehicle experiments. A short movie is provided to demonstrate the avoidance behaviors on `https://youtu.be/WCL2kei0Va0` or `https://b23.tv/k22nVZ`. We select 3 typical cases to demonstrate 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.

[6] 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.

[7] C. Tessler, D. J. Mankowitz, and S. Mannor, "Reward constrained policy optimization," *arXiv preprint arXiv:1805.11074*, 2018.

[8] P. Jorion, *Value at risk: the new benchmark for managing financial risk*. The McGraw-Hill Companies, Inc., 2007.

[9] 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.

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

[11] 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.

[12] 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.

[13] 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.

[14] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz, "Trust region policy optimization," in *International conference on machine learning*. PMLR, 2015, pp. 1889–1897.

[15] Z. Lin, J. Duan, S. E. Li, J. Li, H. Ma, Q. Sun, J. Chen, and B. Cheng, "Solving finite-horizon hjb for optimal control of continuous-time systems," in *2021 International Conference on Computer, Control and Robotics (ICCCR)*, 2021, pp. 116–122.

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

[17] 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.

[18] 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.

[19] 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.

[20] 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.

[21] 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.

[22] 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.

[23] 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.

[24] 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.

[25] 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.

[26] Y. Guan, Y. Ren, S. E. Li, H. Ma, J. Duan, and B. Cheng, "Integrated decision and control: Towards interpretable and efficient driving intelligence," *arXiv preprint arXiv:2103.10290*, 2021.