Appendix of "Multi-Constraint Deep Reinforcement Learning for Smooth Action Control"

A Environment Description

A.1 State Space

As shown in Tabel 1, we classify the state space features into two categories, i.e., the features in the world coordinate system and the features in the ego coordinate system. The world coordinate system information is provided by the CARLA simulator output, and the origin of the world coordinate system is determined at the center of the map. The ego coordinate system is based on the center of the ego vehicle as the origin, the x-axis direction is the forward direction of the ego vehicle, and the y-axis direction is the lateral movement direction of the ego vehicle. The ego coordinate system contributes to represent the relative features of the ego vehicle to other objects.

Category	Name	Dimension	Range
Ego	lateral distance	1	$[0, 10]\,(m)$
	yaw rate	2	$[-\pi, \pi]\,(rad/s)$
	steering angle	1	$[-\pi, \pi]\,(rad)$
	velocity	2	$[-100, 100]\,(m/s)$
	ratio of v_{la} and v_{lo}	2	$[-\pi, \pi]\,(rad)$
	acceleration	2	$[-\pi, \pi]\,(m/s^2)$
	position change	3	$[-10, 10]\,(m\,or\,rad)$
	way points	12*3	$[-10, 10]\,(m\,or\,rad)$
World	position	2	$[-10, 10]\,(m)$

Table 1: The features of the 50-dimensional state space.

A.2 Action Space

Tabel 2 is about the continuous action space in our autonomous driving control tasks. The advantage of continuous action space is that fine action control can be achieved. However, the action space is greatly increased, bringing the challenge of unsmooth action and searching the optimal policy with more difficulty.

Name	Dimension	Range	Detail
longitudinal control	1	$[-1, 1]$	1 means throttle maximum, -1 means brake maximum
lateral control	1	$[-1, 1]$	1 means right steering maximum, -1 means left steering maximum

Table 2: The continuous action space for 2-degree-of-freedom autonomous driving lateral-longitudinal cooperative control.

A.3 Reward

There is about the introduction of task reward and safety reward. The purpose of the task reward R^t is to encourage the vehicle to drive itself and complete the autonomous driving scenario as quickly as possible. And the role of the safety reward R_s is to punish the ego vehicle for risky behaviors to reduce dangerous actions. Task reward R_t^t has two parts

$$R^t = \alpha_0 R^{VLo} + \alpha_1 R^{Stop}, \tag{1}$$

and safety reward also has two parts

$$R^s = \alpha_2 R^{Coll} + \alpha_3 R^{LaD}. \tag{2}$$

Figure 1 illustrates the relationship of sub-rewards in the task reward and safety reward on specific metrics. Table 3

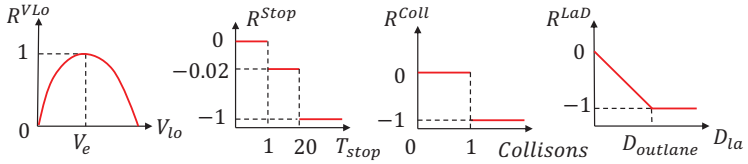


Figure 1: The first is the speed reward R_{VLo}, which is the quadratic function of the longitudinal velocity of the ego vehicle V_{lo}, and V_e is the desired speed we set. The second is the parking reward R_{Stop} related to parking time steps T_{stop}, which is used to penalize illegal parking behavior. The third is the collision reward R_{Coll} used to penalize collision with other objects. The fourth is the reward of keeping lane R_{LaD} related to distance between the ego vehicle and the center of lane D_{la}, and $D_{outlane}$ is threshold for judging the ego vehicle out of lane.

shows the weights of each sub-rewards for the task reward, safety reward, action magnitude reward and comfort reward.

α_0	α_1	α_2	α_3	α_4	α_5	α_6	α_7	α_8	α_9
1.0	5.0	15.	0.5	0.1	0.1	0.5	0.5	0.5	0.5

Table 3: The weights of each sub-reward. Note that α_0 and α_1 are for task, α_2 and α_3 are for safety, α_4 and α_5 are for action magnitude, α_6 α_7 α_8 and α_9 are for comfort.

A.4 Environment Setting

Table 4 shows three environments we designed.

Name	Town07_V1	Town07_V2	Town03_V1
Dimension(S)	50	50	50
Dimension(A)	2	2	2
Decision interval(Δt)	0.2s	0.2s	0.2s
Episode horizon(H)	200	1000	1000
Waypoints number	12	12	12
Waypoints interval	3m	3m	3m
Outlane	3m	3m	3m
Initial speed	0m/s	0m/s	0m/s
Desired speed	12m/s	12m/s	12m/s

Table 4: The configuration of our three environments. Dimension(S) is the dimension of state space. Dimension(A) is the dimension of action space. The waypoints number and the waypoints interval are two parameters of determining the navigation route of the ego vehicle.

B Algorithm

The pseudo-code of multi-constraint proximal policy optimization (MCPPO) is depicted in Algorithm 1.

Algorithm 1 MCPPO algorithm

Input: Environment Env, total training sample size N, sample Buffer B, number of updates in one iteration M
Ouput: Policy net parameters θ^{π}
Initialization: Policy net parameters θ^{π}, value net parameters $\theta^{\boldsymbol{V}}$, Lagrange multipliers $\boldsymbol{\lambda}$

$n = 0$
while $n \leq N$ **do**
  while B is full **do** {sample environment}
    $a_t \sim \pi(\cdot|s_t)$
    $s_{t+1}, r_t, \boldsymbol{c_t} = Env.step(a_t)$
    Save $[s_t, a_t, r_t, \boldsymbol{c_t}]$ to buffer B
  end while
  Calculate advantage vector $\widehat{\boldsymbol{A}}_t$ by Eq.(17).
  Update Lagrange multipliers $\boldsymbol{\lambda}$ by Eq.(20).
  for $i = 0, \ldots, M$ **do**
    Update reward value net parameters $\theta^{\boldsymbol{V^0}}$ by Eq.(18).
    Update cost value nets parameters $\theta^{\boldsymbol{V^{1:|C|}}}$ by Eq.(18).
    Update policy net parameters θ^{π} by Eq.(14).
  end for
  $n+ = |B|$
  Clear B
end while

C Experiments

C.1 Hyperparameters

Tabel 5 shows the hyperparameters of training mainstream DRL algorithms (D3QN, SAC, and PPO) and our MCPPO for our autonomous driving control tasks.

C.2 Training Curve

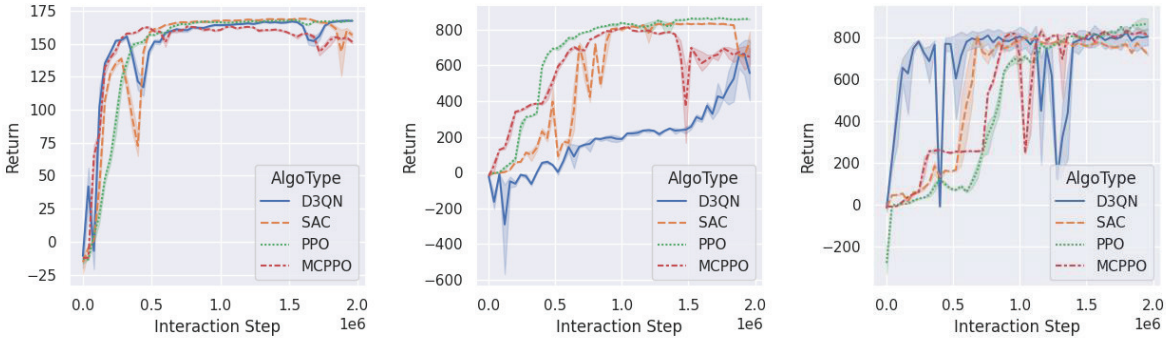


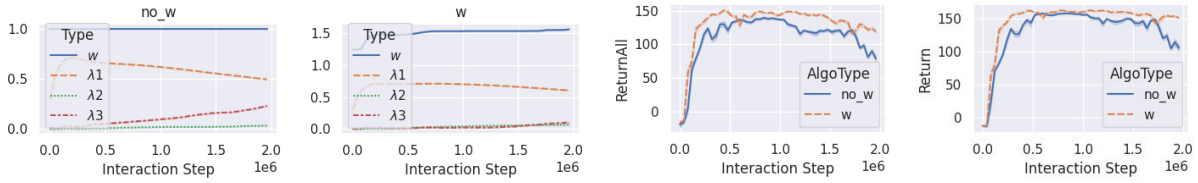
Figure 2: The average return performance of mainstream DRL algorithms and our MCPPO during training. The left is in Town07_V1, the middle is in Town07_V2 and the right is in Town03_V1.

Name	D3QN	SAC	PPO	MCPPO
Optimizer	Adam	Adam	Adam	Adam
Learning rate	1e-4	[1e-4]*3	1e-4	[1e-4]*3
Interaction steps per iteration	4096	4096	4096	4096
Number of parallel environments	4	4	4	4
Buffer size	1e6	1e6	4096	4096
Batch size	256	256	256	256
Number of policy reuse	2	2	8	8
Reward scaling	1	32	1	1
Policy type	argmax	normal	normal	normal
Policy net size	[256,256]	[256,256]	[256,256]	[256,256]
Value net size	[256,256]	[256,256]	[256,256]	[256,256]*4
Gradient crop	4.0	4.0	4.0	4.0
γ	0.99	0.99	0.99	0.99, [0.95]*3
Explore rate	0.25	1	1	1
Target entropy	-	log2	-	-
$\boldsymbol{\lambda^{GAE}}$	-	-	0.97	[0.97]*4
PPO clip	-	-	0.2	0.2
Entropy coefficient	-	-	0.01	0.01
$\boldsymbol{d}$	-	-	-	[0.02, 0.05, 0.12]
Initial $\boldsymbol{\lambda}$	-	-	-	[0]*3
$\boldsymbol{K_p}$	-	-	-	[0.25]*3
$\boldsymbol{K_i}$	-	-	-	[0.01]*3
$\boldsymbol{K_d}$	-	-	-	[4]*3

Table 5: The hyperparameters of mainstream DRL algorithms and our MCPPO for our autonomous driving control tasks.

Figure 2 shows that the average return performance on mainstream DRL algorithms and our MCPPO during training. The convergence returns of D3QN, PPO, SAC and MCPPO are close. The return curves of MCPPO first increase and then slightly decrease. The cost weights are 0 at the beginning of the training phase, which makes the policy optimization start with a focus on maximizing the return value. With the increment of cost constraint violations, it makes a slight loss of reward return that policy optimization gradually shifts toward satisfying the constraint.

C.3 Training of MCPPO



(a) The curve of reward and costs weights over the training process. (b) The curve of return and return G_{CRL} over the training process

Figure 3: Compare the performance of MCPPO with or without reward weight adaptation in Town07_V1. Note that no_w means without no reward weighting adaptation.

Figure 3(a) shows the MCPPO curve of the reward and costs weights during the training, with or without reward weight adaptation. Figure 3(b) shows the better performance of MCPPO with reward weight adaptation. It is worth noting that reward weight adaptation can suppress the return decay caused by constraints. It is because that increases

of the constraint weight cause the policies to gradually ignore the importance of the reward, which triggers the reward decay and makes the policy tend to satisfy the local optimal policies of the constraint.

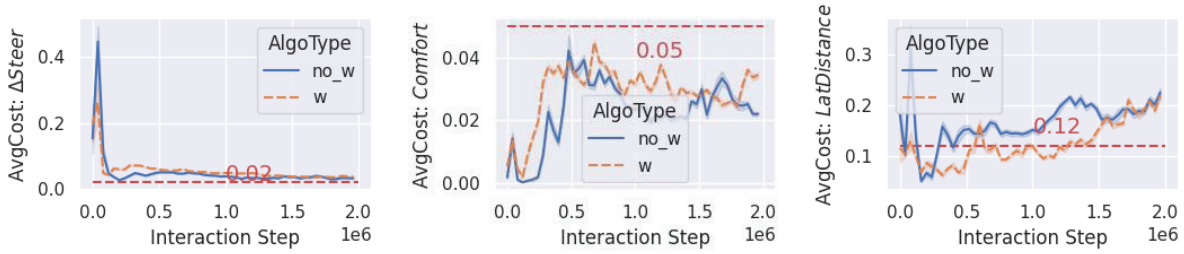


Figure 4: The three costs performance of our MCPPO during training in Town07_V1. no_w means no reward weighting adaptation.

Figure 4 shows the thresholds and the cost curves of action magnitude cost, comfort cost and safety cost. All costs tend to be close to the direction of satisfying the constraints. The costs of the action magnitude and safety fully satisfy their constraints after training. And the comfort cost is close to the threshold of the constraint. This further illustrates that our MCPPO can effectively solve the multi-constraint DRL problems and the effectiveness of the constraints we set.