A Reward Function

Based on the idea of parameterized policies, our goal is to find actions that are likely to yield more rewards:

$$\max_{\theta} J(\theta) = \max_{\theta} E_{\tau\sim\pi_\theta} R(\tau) = \max_{\theta} \sum_{\tau} P(\tau;\theta) R(\tau) \tag{1}$$

For $R(\tau)$, we design a piecewise function:

$$R_1(\tau) = \log(d_{travel}/d_{total}) \quad (d_{travel} < d_{total}) \tag{2}$$

$$R_1(\tau) = c_1(d_{travel} - d_{total}) \quad (d_{travel} >= d_{total}) \tag{3}$$

where d_{total} is the distance to the endpoint, d_{travel} means the distance you have traveled, and c_1 is a constant. $R_2(\tau)$ detects whether the vehicle will depart from the lane. If it does, $R_2(\tau) = -c_2$. $R_3(\tau)$ determines whether the vehicle has collided. $R_2(\tau) = -c_3$ if a collision occurs. We set $c_1 = 1, c_2 = c_3 = 10$. $R(\tau)$ is ultimately the sum of rewards R_1 , R_2 and R_3. I also recommend using this function $R_1(\tau) = -\frac{\|d_{total} - d_{travel}\|_1}{c_1}$ for $R_1(\tau)$. You can freely design reward functions and set up task scenarios.

B CCE-MASAC Parameters

parameter	Value
seed	666 / 1000 / 1666
the number of agents	$n \geq 1$
actor learning rate	5e-5
critic learning rate	1e-4
discounted factor	0.99
gradient clip norm	5e-1
buffer batch size	32
hidden dimension	128
epoch	10
episode	300
soft update coefficient	5e-3
η (temperature coefficient) learning rate	5e-5
initial η log coefficient	5e-3
density learning rate	1e-4
kld weight	2.5e-4
density model coefficient	1e-1
η_1	5e-2
$\frac{1}{\eta_2}$	5e-2

Table 1: CCE-MASAC parameters example table

C Proof of Laplace reparameterization trick

We use the **Inverse Transform Sampling** method to generate random samples from the Laplace distribution. The probability density function (PDF) of the Laplace distribution is:

$$f(x|\mu, b) = \frac{1}{2b} \exp\left(-\frac{|x-\mu|}{b}\right) \qquad (b > 0) \tag{4}$$

The cumulative distribution function (CDF) of the Laplace distribution is handled in two parts. We set x is an instance and when $x < \mu$:

$$F(x|\mu, b) = \frac{1}{2} \exp\left(\frac{x-\mu}{b}\right) \qquad (b > 0) \tag{5}$$

and $x \geq \mu$:

$$F(x|\mu, b) = 1 - \frac{1}{2} \exp\left(-\frac{x-\mu}{b}\right) \qquad (b > 0) \tag{6}$$

We set $u \sim U(0, 1)$, and then Laplace-distributed samples are generated using the inverse transform of the Laplace distribution's CDF. For $u < 0.5$, we have

$$F(x|\mu, b) = u \tag{7}$$

Then we get:

$$x = \mu + b \ln(2u) \tag{8}$$

In the same way, for $u \geq 0.5$, we also have

$$1 - F(x|\mu, b) = u \tag{9}$$

Then we get

$$x = \mu - b \ln(2(1-u)) \tag{10}$$

To simplify the expression, we assume $u \sim U(-1, 1)$, yielding the following equation:

$$x = \mu - b \operatorname{sgn}(u) \ln(1 - |u|) \tag{11}$$

D Algorithm

E Experimental Details

The training process for the world model utilizes 160GB of GPU memory across 8 RTX 4090s and typically runs around 36 hours. Fine-tuning a scene requires 1 RTX 3090 and 3-4 hours of running. The pretraining data is the training dataset of the Argoverse 2 dataset. These four representative scenarios are selected in the validation dataset. The performance of the generative world model can be seen in Table 2.

Algorithm 1 Multi-agent Soft Actor-Critic for capturing CCE (CCE-MASAC)

Input: Offline dataset $\mathcal{D}^o$, the number of total agents I;
Initialize the actors $\{\pi_{\theta_i}\}_{i=1}^I$, the world model $\mathcal{M}_\theta$, the critics $\{Q_{\phi_i}\}_{i=1}^I$, and the target critics $\{Q_{\hat{\phi}_i}\}_{i=1}^I$;
for $n = 1, 2, \ldots, N$ **do**
  Retrieve the n^{th} scenario (including trajectories $\{\tau_{n,i}\}_{i=1}^I$ and map ζ_n) from $\mathcal{D}^o$;
  Update the world model $\mathcal{M}_\theta$ with the objective (4);
end for
for $n = 1, 2, \ldots, N$ **do**
  Retrieve the n^{th} scenario (including trajectories $\{\tau_{n,i}\}_{i=1}^I$ and map ζ_n) from $\mathcal{D}^o$;
  for $i = 1, 2, \ldots, I$ **do**
    Update the critics Q_{ϕ_i} and $Q_{\hat{\phi}_i}$ with the objective (12);
    Update the actor π_{θ_i} with the objective (14);
  end for
end for

Table 2: world model performance

dataset	minFDE (K=6)	minADE (K=6)	MR (K=6)	brier-minFDE (K=6)
val	1.2529	0.7203	0.1578	1.8637

F Experimental Table Results

Table 3 shows all agents' CCE-gap of all algorithms in 4 scenarios.

Table 3: The mean±std of CCE-gap averaged over 300 episodes. The best average performance is highlighted in bold.

Method	Agent 1	Agent 2	Agent 3	Agent 4	Agent 1	Agent 2	Agent 3	Agent 4	Agent 5
	Scenario 1				**Scenario 2**				
QCNet	10.56±0.60	33.51±0.88	39.88±0.59	27.94±0.66	12.48±0.57	17.14±0.74	27.95±0.81	79.97±.71	
GameFormer	19.53±0.78	32.48±3.97	44.94±3.13	40.78±2.14	10.42±6.44	18.80±2.87	25.23±3.45	89.21±1.38	
MAPPO	11.96±1.64	28.15±7.55	35.33±10.22	26.73±7.24	15.36±7.11	14.15±4.25	21.32±9.03	71.16±5.29	
MASAC	9.13±0.79	7.80±2.61	13.02±3.06	7.66±2.64	6.96±1.17	8.06±2.26	8.69±3.27	3.04±2.19	
CCE-MASAC	**5.43±2.26**	**4.17±2.48**	**8.05±3.14**	**5.63±2.98**	**4.30±1.40**	**4.94±3.64**	**6.22±3.58**	**2.47±1.66**	
	Scenario 3				**Scenario 4**				
QCNet	148.33±3.00	85.81±2.09	5.62±0.31	6.91±0.34	23.69±0.90	2.61±0.58	3.60±0.44	4.99±0.39	8.43±0.5
GameFormer	140.88±7.85	59.15±6.85	6.73±4.55	15.33±9.36	32.72±1.18	8.28±1.80	3.23±1.97	7.75±2.49	7.59±3.04
MAPPO	146.85±13.54	74.49±14.95	7.52±0.96	2.92±0.59	32.12±1.94	2.58±2.3	5.14±2.02	5.77±1.51	9.00±0.60
MASAC	28.34±21.01	16.83±4.29	3.16±0.54	2.06±0.50	11.92±2.93	1.31±0.28	2.25±0.35	2.32±0.28	2.92±0.65
CCE-MASAC	**20.70±16.09**	**8.39±3.91**	**1.97±1.07**	**1.36±0.54**	**6.59±3.96**	**1.05±0.25**	**1.43±0.40**	**1.94±0.29**	**1.41±0.78**

G Experimental Figure Results

Our evaluation environment includes 4-5 agents in our paper. We additionally verify the scalability of CCE-MASAC when the number of agents increases (6, 8, and 10) in Figure 1.

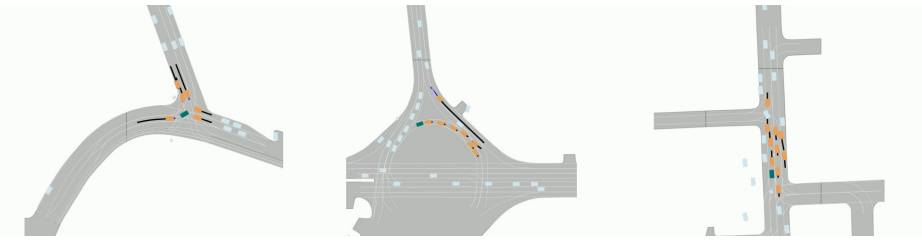

Fig. 1: visualization of additional different scenarios