Collaborative-Online-Learning-Enabled Distributionally Robust Motion Control for Multi-Robot Systems: Supplementary Material ⋆

Chao Ning [a,b], Han Wang [a,b], Longyan Li [a,b], Yang Shi [c]

[a] *Department of Automation, Shanghai Jiao Tong University, Shanghai, China*

[b] *Key Laboratory of System Control and Information Processing, Ministry of Education of China, Shanghai, China*

[c] *Department of Mechanical Engineering, University of Victoria, Victoria, Canada*

5 Simulation results

We conduct simulations to demonstrate the effectiveness and superiority of the COOL-DRMC. We consider three scenarios: (i) two quadrotors navigating in a 3D environment with two dynamic obstacles whose motions are subject to multimodal distributions, (ii) two nonlinear vehicles navigating in an intersection-like scenario with two dynamic obstacles whose motions are subject to multimodal distributions, and (iii) three car-like robots navigating in a 2D environment with several static obstacles and dynamic obstacles whose motions are subject to time-varying distributions. Furthermore, we discuss the impact of ambiguity set compression on control performance and computation time. All the simulations are implemented in MATLAB R2023b. The YALMIP toolbox is utilized for mathematical modeling. The IPOPT 3.12.9 and MOSEK 10.1 solver are adopted to solve the optimization problem (11) and (31), respectively. The computational experiments are performed on a personal computer with 2.10 GHz Inter Core i7-13700 CPU and 32 GB RAM.

5.1 Simulation 1 with multimodal distributions

Consider two quadrotors navigating in a 3D environment with the following dynamics Mistler *et al.* (2001).

$$\begin{aligned} \ddot{x} &= g\theta \qquad \ddot{y} = -g\phi \qquad \ddot{z} = \tfrac{1}{m_Q}u_1 \\ \ddot{\phi} &= \tfrac{l_Q}{I_{xx}}u_2 \qquad \ddot{\theta} = \tfrac{l_Q}{I_{yy}}u_3 \qquad \ddot{\psi} = \tfrac{l_Q}{I_{zz}}u_4 \end{aligned} \tag{S.1}$$

where $g = 9.81kg/m^2$ is the gravitational acceleration, $m_Q = 0.68kg$ is the mass of the quadrotor, $l_Q = 0.23m$ is the distance between the CoM of the quadrotor and the rotor, and $I_{xx} = 0.0075kg \cdot m^2$, $I_{yy} = 0.0075kg \cdot m^2$ and $I_{zz} = 0.013kg \cdot m^2$ represent the area moments of inertia about the principal axes in the body frame. The states are $\left(x, y, z, \dot{x}, \dot{y}, \dot{z}, \phi, \theta, \psi, \dot{\phi}, \dot{\theta}, \dot{\psi}\right) \in \mathbb{R}^{12}$, and the outputs are taken as the CoM of the quadrotor (x, y, z). The objective is to control the quadrotors to track their reference trajectories while navigating around two randomly moving obstacles (grey) as shown in Fig. S.1. The initial states of Quadrotor 1 and 2 are $x_1(0) = (2, 2, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0)^T$ and $x_2(0) = (22, 2, 11, 0, 0, 0, 0, 0, 0, 0, 0, 0)^T$, respectively. The motions of the Obstacle 1 and 2 are sampled from the Gaussian mixture distributions with three components, denoted as $p_{1,t} = \sum_{i=1}^{3} \gamma_{1,t}^i \mathcal{N}(\mu_{1,t}^i, \Sigma_{1,t}^i)$ and $p_{2,t} = \sum_{i=1}^{3} \gamma_{2,t}^i \mathcal{N}(\mu_{2,t}^i, \Sigma_{2,t}^i)$ respectively. The parameters of the distribution are provided in Table S.1.

The total simulation time is set to $T = 100s$, and the system (S.1) is discretized using the sampling time $T_s = 0.1s$. The prediction horizon is set to $K = 10$. Increasing K improves trajectory tracking accuracy but also increases computational complexity while decreas-

Table S.1
The parameters of motion distributions of Obstacle 1 and 2 in simulation 1 ($p_{1,t}$ and $p_{2,t}$).

ℓ	i	$\gamma_{\ell,t}^i$	$\mu_{\ell,t}^i$	$\Sigma_{\ell,t}^i$
1	1	0.3	$(0, 0, 0.1)^T$	$diag(0.01, 0.01, 0.01)$
	2	0.4	$(0, 0, 0.2)^T$	$diag(0.01, 0.01, 0.01)$
	3	0.3	$(0, 0, 0.3)^T$	$diag(0.01, 0.01, 0.01)$
2	1	0.3	$(0, -0.2, 0)^T$	$diag(0.01, 0.09, 0.09)$
	2	0.4	$(0.2, -0.2, 0)^T$	$diag(0.01, 0.01, 0.01)$
	3	0.3	$(0.2, 0, 0)^T$	$diag(0.01, 0.01, 0.01)$

⋆ This paper was not presented at any IFAC meeting. (Corresponding authors: C. Ning and Y. Shi)

Email addresses: chao.ning@sjtu.edu.cn (Chao Ning), h.wang@sjtu.edu.cn (Han Wang), lilongyan@sjtu.edu.cn (Longyan Li), yshi@uvic.ca (Yang Shi).

ing K can enhance the response speed of the controller. In our case study, $K = 10$ is appropriate in our case study. The state and input weighting matrices are $Q = diag(1, 1, 1, 0, ..., 0)$ and $R = 0.01diag(1, 1, 1, 1)$, respectively. We set $\alpha = 0.95$. The relative increase in Q and R will respectively cause the controller to emphasize tracking accuracy and the smoothness of the control trajectory. In our case study, we choose a relatively large Q to achieve a more precise control trajectory. The robotic motion control methods analyzed in this subsection are listed below.

- W-DRMC: Wasserstein Distribution Robust Motion Control with radius $\theta = 0.01, 0.03, 0.05$ Hakobyan and Yang (2021).
- SH-W-DRMC: Safe Halfspace based W-DRMC with radius $\theta = 0.01, 0.03, 0.05$ Safaoui and Summers (2023).
- COOL-DRMC: The proposed method with the maximum number of basic ambiguity set $M_{\ell,t,k} = 1, 10, 100$.
- COOL-DRMC-WUP: The proposed method Without Uncertainty Propagation. The separating hyperplanes $\mathcal{H}^{obs}_{\ell,t,k}$ over the entire prediction horizon are computed based on the ambiguity set of $\varphi_\ell(1|t)$.

The trajectories generated by different motion control methods are shown in Fig. S.1. We observe that, compared with other methods, COOL-DRMC exhibits a closer proximity to the reference trajectory when maneuvering around obstacles.

Table S.2 shows the average cost and computation time

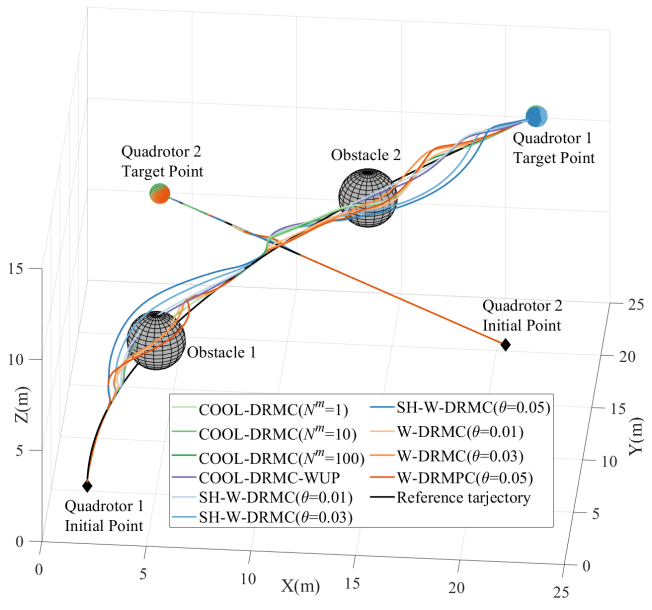


Fig. S.1. Trajectories of the quadrotors with different methods in simulation 1 (The positions of the obstacles and the quadrotors correspond to the positions at $T = 100s$).

Table S.2
Average cost and computation time of quadrotor 1 with different methods.

Method		Average Cost	Computation Time(s)
W-DRMC	$\theta = 0.01$	8.6658	3.0535
	$\theta = 0.03$	12.1944	3.4039
	$\theta = 0.05$	17.1208	3.6547
SH-W-DRMC	$\theta = 0.01$	27.6748	0.0069
	$\theta = 0.03$	44.2133	0.0070
	$\theta = 0.05$	64.6716	0.0071
COOL-DRMC-WUP		22.0459	0.0328
COOL-DRMC	$M_{\ell,t,k} = 1$	7.6468	0.0601
	$M_{\ell,t,k} = 10$	7.5875	0.1286
	$M_{\ell,t,k} = 100$	7.4860	0.8267

of different methods. It can be observed that the average cost of the proposed method is significantly smaller than those of all other methods. The COOL-DRMC method with $M_{\ell,t,k} = 10$ reduces the cost by an average of 12.44% compared with the W-DRMC with $\theta = 0.01$ by using the fined-grained distribution information in ambiguity sets. Furthermore, compared with W-DRMC, COOL-DRMC reduces the average computation time by one to two orders of magnitude depending on the value of $M_{\ell,t,k}$ through the utilization of separating hyperplanes. Note that while COOL-DRMC-WUP and SH-W-DRMC exhibit a shorter computation time compared to COOL-DRMC, the quadrotors controlled by COOL-DRMC-WUP and SH-W-DRMC are confined within the same halfspace throughout the entire prediction horizon due to the absence of uncertainty propagation, which results in control lag and a cost increase of over 188%.

Additionally, based on the ambiguity set compression, COOL-DRMC can balance the control performance and computation time by adjusting $M_{\ell,t,k}$. Specifically, when $M_{\ell,t,k}$ is large ($M_{\ell,t,k} = 100$), the ambiguity set is scarcely compressed, thus boosting control performance but incurring a substantial computational burden. Conversely, as $M_{\ell,t,k}$ decreases ($M_{\ell,t,k} = 1$), the ambiguity set is gradually compressed, leading to a drastic reduction in computational burden at the expense of slightly diminished control performance.

5.2 Simulation 2 using nonlinear dynamics

Consider two vehicles navigating in a 2D environment with the following nonlinear dynamics Pepy *et al.* (2006).

$$\begin{bmatrix} p_{x,i}(t+1) \\ p_{y,i}(t+1) \\ \theta_i(t+1) \\ v_i(t+1) \end{bmatrix} = \begin{bmatrix} p_{x,i}(t) + T_s \cdot v_i(t) \cos\theta_i(t) \\ p_{y,i}(t) + T_s \cdot v_i(t) \sin\theta_i(t) \\ \theta_i(t) + T_s \cdot \frac{v_i(t)}{l} \tan\phi_i(t) \\ v_i(t) + T_s \cdot a_i(t) \end{bmatrix} \quad \text{(S.2)}$$

where $x_i(t) = (p_{x,i}(t), p_{y,i}(t), \theta_i(t), v_i(t))$ are the po-

Table S.3
The parameters of motion distributions of Obstacle 1 and 2 in simulation 2 ($p_{1,t}$ and $p_{2,t}$).

ℓ	i	$\gamma^i_{\ell,t}$	$\mu^i_{\ell,t}$	$\Sigma^i_{\ell,t}$
1	1	0.25	$(0, 0.4)^T$	$diag(0.05^2, 0.05^2)$
	2	0.5	$(-0.2, -0.2)^T$	$diag(0.05^2, 0.05^2)$
	3	0.25	$(0.4, 0)^T$	$diag(0.05^2, 0.05^2)$
2	1	0.25	$(0, -0.4)^T$	$diag(0.05^2, 0.05^2)$
	2	0.5	$(-0.2, 0.2)^T$	$diag(0.05^2, 0.05^2)$
	3	0.25	$(0.4, 0)^T$	$diag(0.05^2, 0.05^2)$

sition, orientation, and velocity of the vehicle, respectively. $l := 0.25m$ is the length, and $T_s = 0.1$ is the sampling time. The system inputs are the steering angle $\phi_i(t) \in [-\pi/6, \pi/6]$ and the acceleration $a_i(t) \in [-5, 5]$. The control horizon is set to $K = 10$. The objective is to control the vehicles to track their reference trajectories through an intersection-like scenario while navigating around two randomly moving obstacles. The initial states of Vehicles 1 and 2 are $x_1(0) = (6, 0.5, \pi/2, 0)^T$ and $x_2(0) = (2, 4, 0, 0)^T$, respectively. The motions of the Obstacles 1 and 2 are sampled from the Gaussian mixture distributions with three components, denoted as $p_{1,t} = \sum_{i=1}^{3} \gamma^i_{1,t} \mathcal{N}(\mu^i_{1,t}, \Sigma^i_{1,t})$ and $p_{2,t} = \sum_{i=1}^{3} \gamma^i_{2,t} \mathcal{N}(\mu^i_{2,t}, \Sigma^i_{2,t})$ respectively. The parameters of the distribution are provided in Table S.3. The total simulation time is set to $T = 100s$. The state and input weighting matrices are $Q = diag(10, 10, 0, 0)$ and $R = 0.01diag(1, 0.1)$, respectively. We set $\alpha = 0.95$.

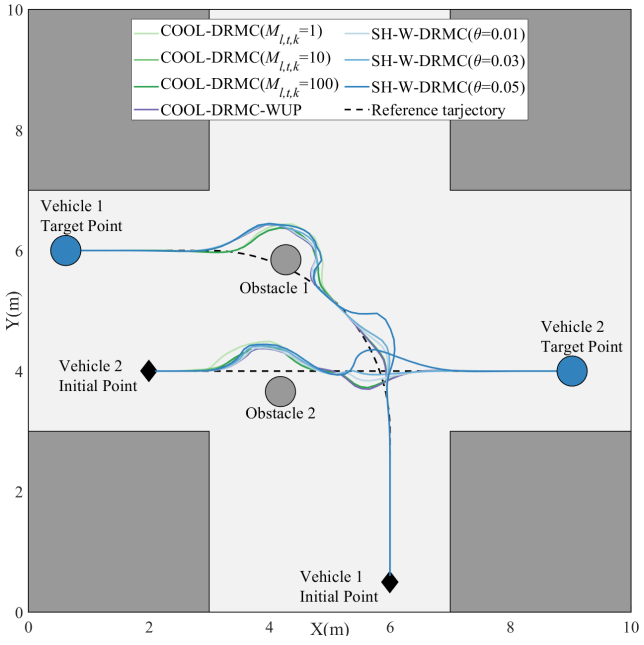


Fig. S.2. Trajectories of the vehicles with different methods (The positions of the obstacles and the vehicles correspond to the positions at $T = 100s$).

Table S.4
Average cost and computation time of vehicle team with different methods.

Method		Average Cost	Computation Time(s)
SH-W-DRMC	$\theta = 0.001$	10.5347	0.0875
	$\theta = 0.003$	10.5064	0.0730
	$\theta = 0.005$	13.5858	0.0758
COOL-DRMC-WUP		10.0661	0.1164
COOL-DRMC	$M_{\ell,t,k} = 1$	5.6015	0.1667
	$M_{\ell,t,k} = 10$	3.8281	0.3493
	$M_{\ell,t,k} = 100$	3.6395	1.4477

The robotic motion control methods analyzed in this subsection, except for W-DRMC, are the same as those in simulation 1.

The trajectories generated by different motion control methods are shown in Fig. S.2. The average cost and computation time of different methods are shown in Table S.4. It can be observed that COOL-DRMC is still capable of avoiding collisions with moving obstacles even for nonlinear vehicle dynamics. Additionally, the COOL-DRMC method with $M_{\ell,t,k}$ reduces the cost by an average of 63.66% compared with the W-DRMC with $\theta = 0.001$. Similar to the results of Simulation 1, COOL-DRMC can flexibly balance the trade-off between average cost and computation time by adjusting $M_{\ell,t,k}$. It is important to note that in this simulation, we do not compare the proposed method with W-DRMC. This is because W-DRMC does not employ the separating hyperplane technique to convexify the distributionally robust collision avoidance constraint, which, in combination with the nonlinear model, imposes a significant computational burden on the control optimization problem. Specifically, the average computation time of W-DRMC exceeds 100 seconds at each sampling time, rendering it entirely unsuitable for real-time robotic control. In summary, the simulation results demonstrate that the proposed method can ensure safety and achieve excellent control performance even for nonlinear models.

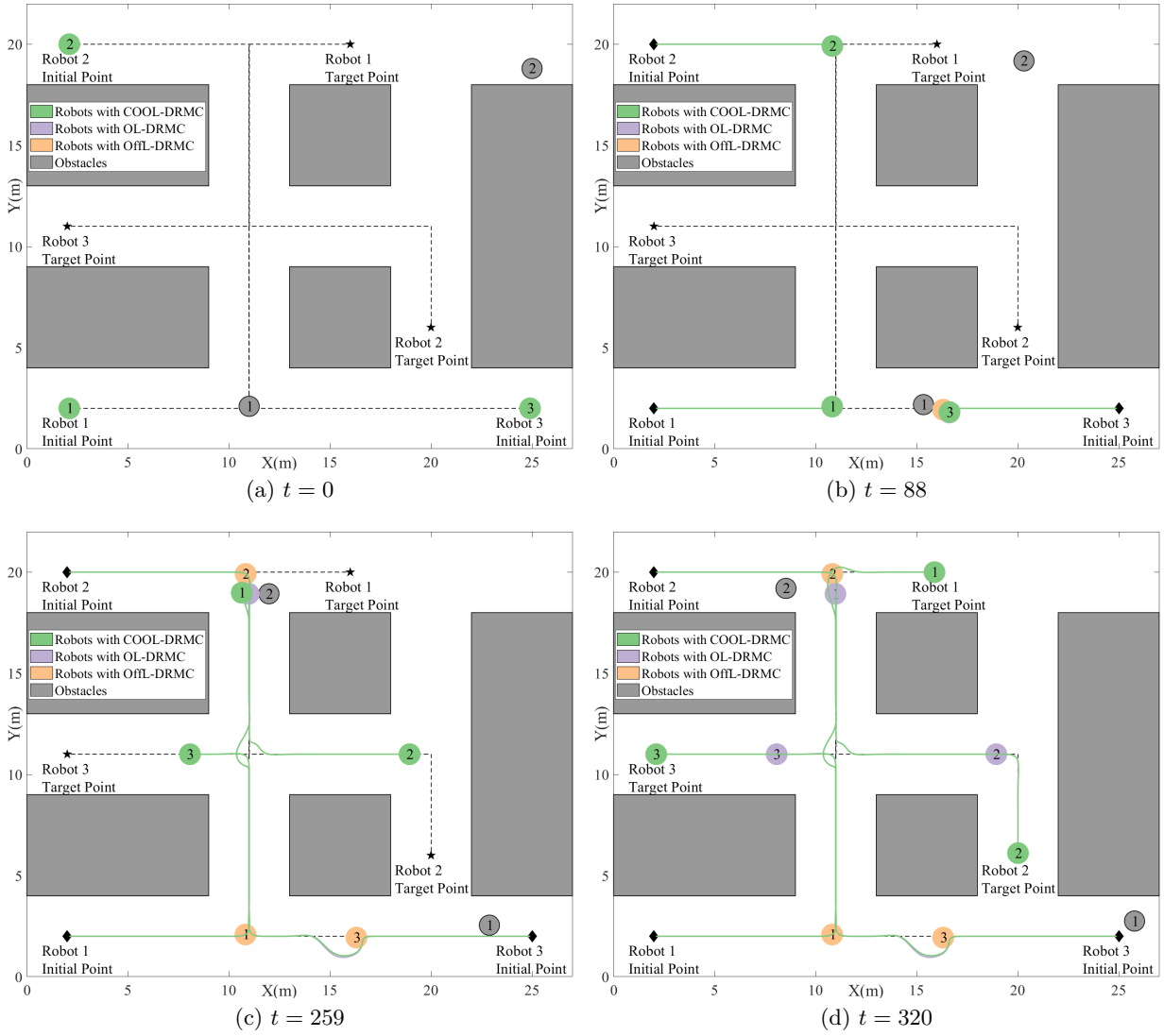
5.3 Simulation 3 with time-varying distributions

Consider three car-like robots navigating in a 2D environment with the following double integrator dynamics.

$$x(t+1) = \begin{bmatrix} 1 & 0 & T_s & 0 \\ 0 & 1 & 0 & T_s \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} x(t) + \begin{bmatrix} \frac{T_s^2}{2} & 0 \\ 0 & \frac{T_s^2}{2} \\ T_s & 0 \\ 0 & T_s \end{bmatrix} u(t) \quad \text{(S.3)}$$

where (p^x, p^y, v^x, v^y) is the state of a robot, consisting of its CoM and velocity vector. The input $u = (a^x, a^y)$ is chosen as the acceleration vector. Similarly, the sampling time T_s is selected as $0.1s$.

The simulation configuration is illustrated in Fig. S.3a. The initial states of Robot 1, 2 and 3 are $x_1(0) = (2, 2, 0, 0)^T$, $x_2(0) = (2, 20, 0, 0)^T$ and $x_3(0) = (25, 2, 0, 0)^T$, respectively. The circular robots with radius $r_r = 0.5$ aim to track the given reference trajectories in a cluttered 2D environment with some static and dynamic obstacles where the static obstacles are modeled as rectangles representing topographic barriers and the dynamic obstacles are approximated by the circles with radius $r_o = 0.5$ representing human or other robots that cannot be communicated with. The motions in each direction of the Obstacle 1 and 2 are sampled from time-varying Gaussian distributions. Specifically, the means of the motion distribution of Obstacle 1 and 2 are $(0.5, 0)^T$ and $(-0.52, 0)^T$ respectively, and the standard deviation of all distributions used to generate historical data is 0.01, while the standard deviation increases to 0.3 for the real-time data generation during the runtime of robotic motion control. The control input for robots is limited to lie in $\mathcal{U} := \{u \in \mathbb{R}^2 | \|u\|_\infty \leq 4\}$ and its state is restricted to $\mathcal{X} := \{x \in \mathbb{R}^4 | (0, 0, -2, -2) \leq x \leq (22, 27, 2, 2)\}$. The reference trajectories of each robot are generated through the upper-level planner. The prediction horizon is set to $K = 10$ and the state and input weighting matrices are chosen as $Q = diag(1, 1, 0, 0)$ and $R = 0.1diag(1, 1, 1, 1)$. The maximum number of base ambiguity sets is selected as $M_{\ell,t,k} = 10$ and we set $\alpha = 0.95$. The simulation is independently repeated 500 times. The following robotic motion control methods are analyzed in this subsection to demonstrate the superiority of the proposed collaborative online learning.



(a) $t = 0$ (b) $t = 88$

(c) $t = 259$ (d) $t = 320$

Fig. S.3. Trajectories of robots with different control methods in simulation 2. (Rectangles represent static obstacles and circles represent the robots and obstacles. Numbers on the circles denote the indices of robots or obstacles and the different colors of the circles indicate the robots controlled by different methods as shown in the legend.)

- OffL-DRMC: Offline DRO-based Motion Control. Without online learning, the ambiguity sets are constructed only based on historical data.
- OL-DRMC: Online-Learning-enabled DRO-based Motion Control. Without collaborative learning, each robot updates the ambiguity sets solely based on its own observed data.
- COOL-DRMC: The proposed method. The ambiguity sets are updated through the collaborative online learning method.

Fig. S.3 shows the simulation results from one of the 500 independent simulations. For OffL-DRMC, the estimated distributions of random motions are based exclusively on historical data. Unfortunately, the motions of the obstacles become violent because of the increased standard deviation and the deviation between estimated and real-time distributions leads to the collision between obstacle 1 and robot 3 controlled by the OffL-DRMC at time $t = 88$ as shown in Fig. S.3b. In contrast, COOL-DRMC and OL-DRMC can learn the changes in the distribution and keep a safe distance from the obstacle.

For obstacle 2, it cannot be observed by robot 1 until it is approached due to the occlusion of static obstacles. For the OL-DRMC, only a few observed data of obstacle 2 are available when the controller attempts to avoid obstacle 2. The lack of data leads to the inability of robot 1 to learn the changed distribution in time, which causes the collision between obstacle 2 and robot 1 at time $t = 259$ as shown in Fig. S.3c. Fortunately, before robot 1 approaches obstacle 2, robot 2 has learned the changed distribution using sufficient observed data streams thanks to its advantageous position. Utilizing the collaborative online learning, robot 1 with COOL-DRMC learns the distribution of obstacle 2 based on the learning structure received from robot 2 instead of relying on the initial learning structure when robot 1 initially observes obstacle 2. Therefore, COOL-DRMC controls robot 1 to avoid obstacle 2 safely. Finally, as shown in Fig. S.3d, the robot team controlled by COOL-DRMC successfully reached their target points, outperforming the other two methods.

Table S.5 shows the collision-free rate of different methods with 500 independent simulations. Without online learning, OffL-DRMC has a 35.4% rate of collision with obstacle 1. In contrast, COOL-DRMC and OL-DRMC demonstrate a collision-free rate of over 99% in navigating around obstacle 1. Although OL-DRMC avoids collisions with obstacle 1 with a high probability, robot 1 controlled by OL-DRMC has a 30.8% rate of collision with obstacle 2 due to the absence of real-time data, as described above. Therefore, the collision-free rate throughout the total navigation of OL-DRMC is only 68.8%. Thanks to the proposed collaborative online learning, the collision-free rate throughout the total navigation of COOL-DRMC is 97.6%.

Table S.5
Collision-free rate of different methods with 500 simulations.

Method	Collision-free rate		
	With obstacle 1	With obstacle 2	Total
OffL-DRMC	64.6%	69.2%	46.4%
OL-DRMC	99.4%	69.2%	68.8%
COOL-DRMC	99.6%	98%	97.6%

5.4 Discussions on ambiguity set compression

In this section, we discuss the influence of the maximum number of base ambiguity sets $M_{\ell,t,k}$ on the control performance and computation time. We consider a robot with the dynamics as (S.3) circumvents a dynamic obstacle whose motions in each direction are sampled from a Gaussian mixture distribution with three components.

In Fig. S.4, the average cost and computation time with $M_{\ell,t,k} = 1 \sim 80$ are shown. The average cost decreases rapidly with an increase of $M_{\ell,t,k}$ when $M_{\ell,t,k} < 10$ and remains stable as $M_{\ell,t,k} > 10$. Instead, the average computation time consistently ascends as $M_{\ell,t,k}$ increases until $M_{\ell,t,k} > 66$, especially when $M_{\ell,t,k} < 40$. Therefore, it is a wise choice to choose $M_{\ell,t,k} = 10$ to balance the average cost and computation time. Additionally, the user can choose other values of $M_{\ell,t,k}$ according to the computational capability, the real-time requirement, and control performance.

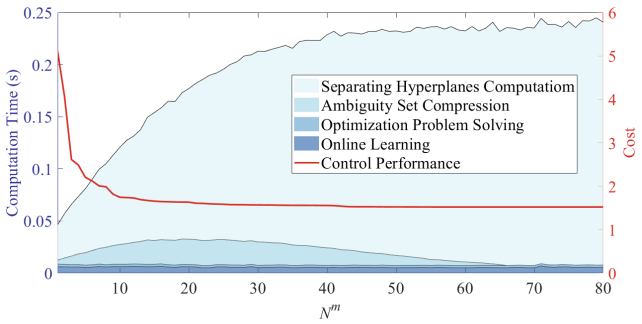


Fig. S.4. The average cost and the average computation time with $M_{\ell,t,k} = 1 \sim 80$.

The computation time is mostly spent on calculating the robot-obstacle separating hyperplanes over the entire prediction horizon, and therefore $M_{\ell,t,k}$ determines the maximum number of constraints in the optimal problem (11) and in turn determines the computation time. For the simulation configuration with prediction horizon $K = 10$ and the number of components of distribution $m_{\ell,t} = 3$, the maximum number of base ambiguity sets for the non-compressed ambiguity set is $m_{\ell,t,K} = \binom{m_{\ell,t}-1}{K+m_{\ell,t}-1} = 66$. Thus the ambiguity set compression time reduces to 0 and the computation time is no longer increased when $M_{\ell,t,k} > 66$. Note that if the hardware computational capability is sufficient and better control

performance is desired, $M_{\ell,t,k}$ can be increased. Conversely, to improve computational speed, $M_{\ell,t,k}$ can be decreased. Empirically, we show that $M_{\ell,t,k} = 10$ provides a reasonable compromise in most cases.

References

Hakobyan, A. and Insoon Yang (2021). 'Wasserstein distributionally robust motion control for collision avoidance using conditional value-at-risk'. *IEEE Transactions on Robotics* **38**(2), 939–957.

Mistler, V., Abdelaziz Benallegue and NK M'sirdi (2001). Exact linearization and noninteracting control of a 4 rotors helicopter via dynamic feedback. In 'Proceedings 10th IEEE International Workshop on Robot and Human Interactive Communication'. IEEE. pp. 586–593.

Pepy, R., Alain Lambert and Hugues Mounier (2006). Path planning using a dynamic vehicle model. In '2006 2nd International Conference on Information & Communication Technologies'. Vol. 1. IEEE. pp. 781–786.

Safaoui, S. and Tyler H Summers (2023). 'Distributionally robust cvar-based safety filtering for motion planning in uncertain environments'. *arXiv preprint arXiv:2309.08821*.