Supplementary Material

1 Implementation Details

Table. 1: RL Agent, GA and Simulation Hyperparameters

Component	**Parameter**	**Value**
RL Agent (DQN)	Neural network layers	2
	Neurons per layer	24
	Learning rate	0.001
	Initial epsilon	0.99
	Epsilon decay	0.995
	Minimum epsilon	0.01
	Collision reward (**I**)	1
	Training episodes	800
GA	Population size	20
	Generations per trial	20
	Simulation Budget	400
	Mutation rate	0.2
	Crossover rate	0.2
	Mutation candidates	10
Simulation Setting	Positional parameter range	[-1,1]
	Weather parameter confined range	[0,0.15]

2 Details of Baseline Settings

- *Offline RL Fuzzer* baseline: the state encodes the environment, the marker position, and available NPC objects' start and endpoints (see Equation 1). In the simulation, the RL agent can alter a weather condition or marker location by ± 0.1 or direct an NPC object in one of four directions: up, down, left, or right (refer to Equation 2). The configuration for the offline RL model—including its exploration strategy (initial epsilon and its decay) and learning rate—parallels that of our surrogate training protocol (see Table 1).

$$\begin{aligned} S_{offline} = (&dust, fog, \ldots, snow, \\ &P_{marker,x}, P_{marker,y}, \\ &P_{obj1,x}, P_{obj1,y}, \\ &D_{obj1,x}, D_{obj1,y}, \\ &\cdots, \\ &P_{objn,x}, P_{objn,y}, \\ &D_{objn,x}, D_{objn,y}) \end{aligned} \tag{1}$$

$$\begin{aligned} A_{offline} := \{&dust \pm 0.1, \ldots, snow \pm 0.1, \\ &P_{marker,x} \pm 0.1, P_{marker,y} \pm 0.1, \\ &U_{i1}, D_{i2}, L_{i3}, R_{i4}, S_{i5}, \\ &U_{d1}, D_{d2}, L_{d3}, R_{d4}, S_{d5}, \\ &\cdots, \\ &U_{in}, D_{in}, L_{in}, R_{in}, S_{in}, \\ &U_{dn}, D_{dn}, L_{dn}, R_{dn}, S_{dn}\} \end{aligned} \tag{2}$$

where P_{marker} indicates the position of the marker, P_{obj} and D_{obj} indicate the initial position and destination of the NPC-object, n is the number of NPC-object in the scenario, i, and d indicate the adjustment for the initial position and the destination respectively.

- *Online RL Testing* baseline 1: This baseline allows the RL agent to control both weather, daytime, and NPC objects online. Each simulation is started with a randomly initialized test scenario. The RL's state and action sets adhere to Equations 3 and 4, and the reward function mirrors that of our surrogate environment. During each simulation, an action will be chosen from the action space online. If the action is to move the NPC object, a random NPC object will be selected to move.

$$\begin{aligned} S_{online} = (&dust, fog, \ldots, snow, \\ &P_{obj,x} - P_{marker,x}, P_{obj,y} - P_{marker,y}, \\ &P_{uav,x} - P_{marker,x}, P_{uav,y} - P_{marker,y}) \end{aligned} \tag{3}$$

$$A_{online} := \{dust \pm 0.01, \ldots, snow \pm 0.01, \\ U, D, L, R, S\} \tag{4}$$

where P_{obj}, P_{uav}, and P_{marker} indicate the position of NPC-object, UAV and marker.

3 Detailed Violation Distribution

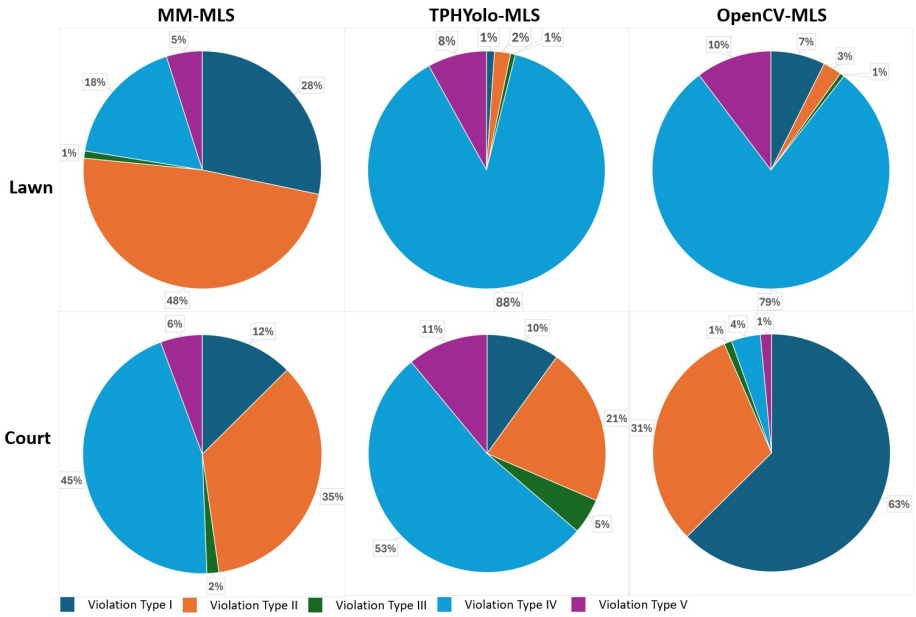

Fig. 10: Distribution of violation types (%) across three landing systems in two maps

4 Statistical Analysis

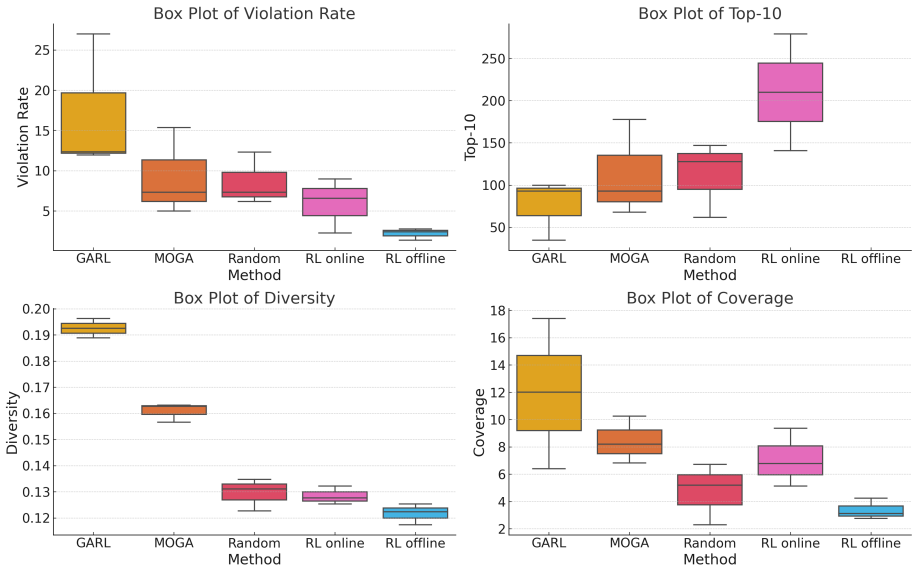

Fig. 11: Box plot of different metrics across different methods in *Lawn* map

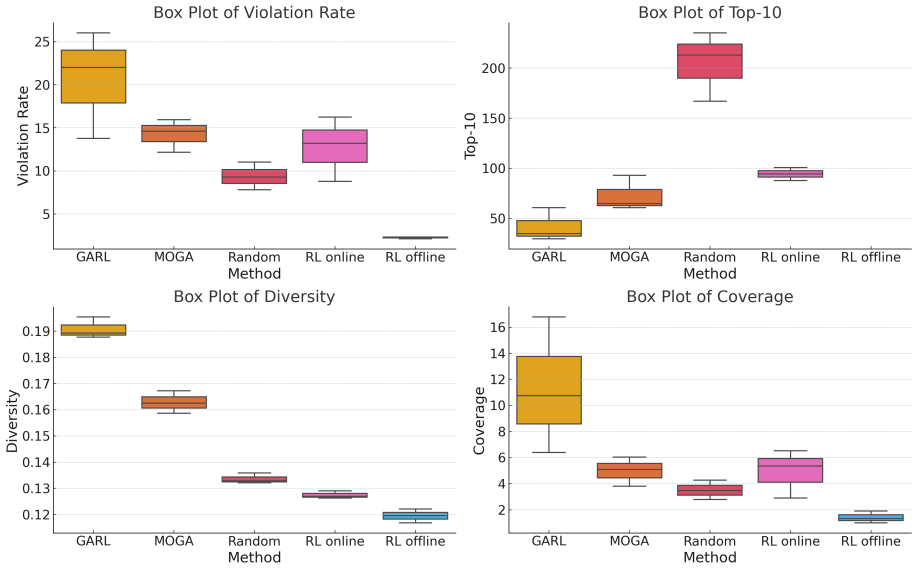

Fig. 12: Box plot of different metrics across different methods in *Court* map

5 RL Training Curve

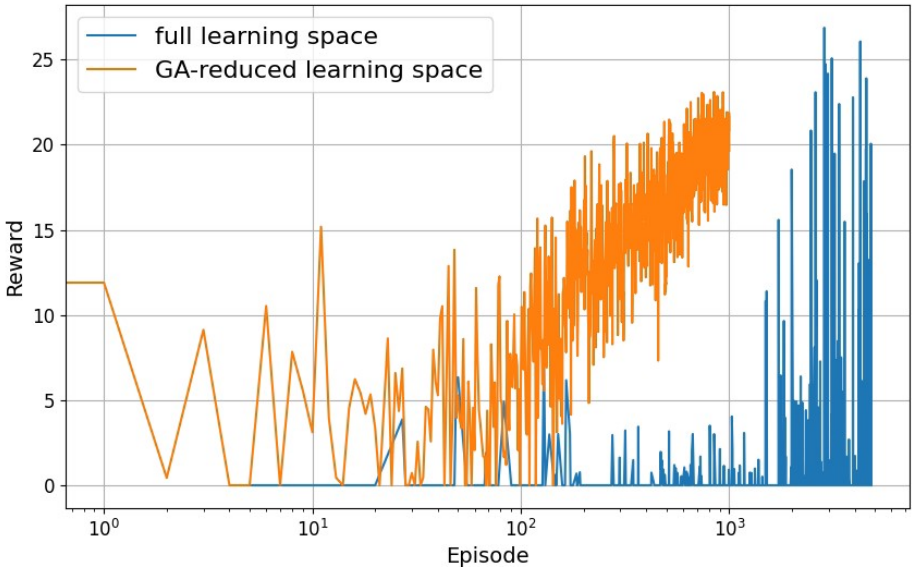

Fig. 13: Comparison of Reward for Different Learning Space in Surrogate Training