1 Violation Distribution

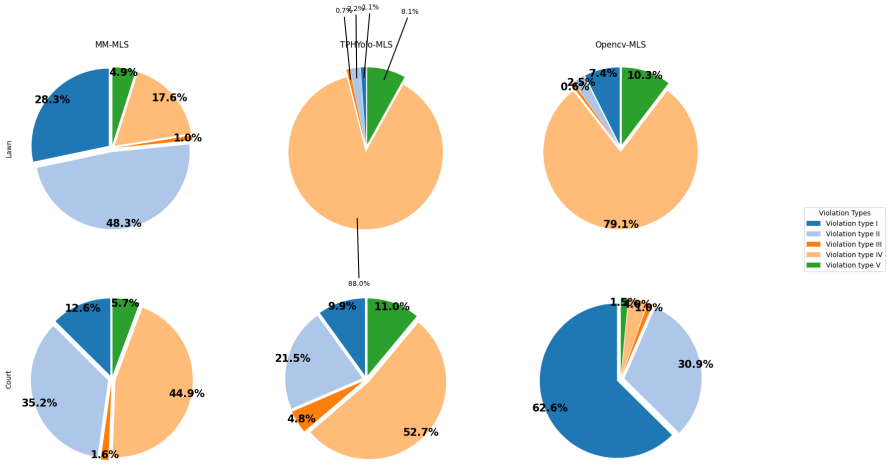


Figure 1: Distribution of violation Types across three landing systems

2 Static Analysis of Methods

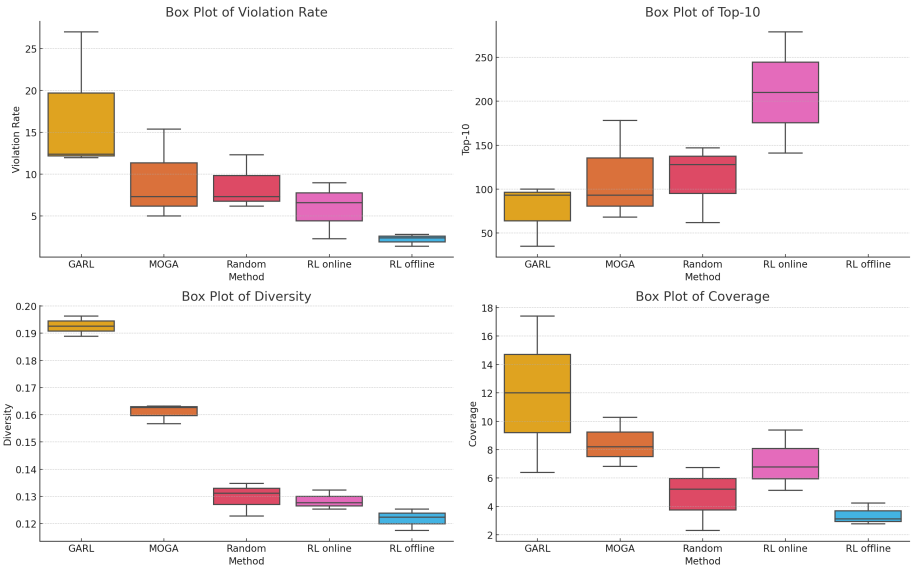


Figure 2: Box pilot of different metrics across different methods in *Lawn* map

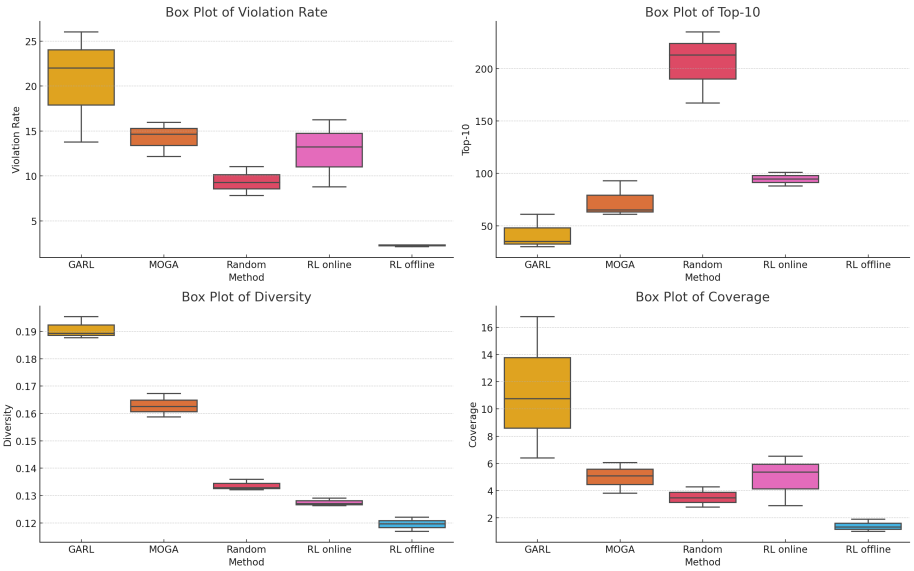


Figure 3: Box pilot of different metrics across different methods in *Court* map

3 Variation Operations

$$y_{ij}^{'} = m_{k \in K \arg\max, \sum_{y \in Y_{ij}} |m_k - y|} \tag{1}$$

Algorithm 1 The GA chromosome-based suite of variation operators

Input: Parents P_t, Offspring O_t, crossover threshold $threshold_c$, mutation threshold $threshold_m$, number of mutation candidates m
Output: P_{t+1}, O_{t+1}
$P_{t+1} \leftarrow \emptyset$, $O_{t+1} \leftarrow \emptyset$
$R_t \leftarrow P_t \cup O_t$
for i in range(0,$|P_t|$) **do**
  sort and select parent chromosome $x_i \in R_t$
  $P_{t+1} \leftarrow P_{t+1} \cup \{x_i\}$
end for
for each pair of chromosomes $(x_i, x_j) \in P_{t+1}$ **do**
  generate $r \sim U(0,1)$
  if $r > threshold_c$ **then**
    generate crossover point $s \sim U(0, Len(x_i))$
    $x_i^{'}, x_j^{'} \leftarrow NuclearGeneCrossover(x_i, x_j, s)$
    $O_{t+1} \leftarrow O_{t+1} \cup \{x_i^{'}, x_j^{'}\}$
  else
    $O_{t+1} \leftarrow O_{t+1} \cup \{x_i, x_j\}$
  end if
end for
for each chromosome $x_i \in O_{t+1}$ **do**
  for each nuclear gene $y_i \in x_i$ **do**
    for each property $y_{ij} \in y_i$ **do**
      generate $r \sim U(0,1)$
      if $r > threshold_m$ **then**
        $M \leftarrow \emptyset$
        for i in range(0,m) **do**
          generate $c \sim$ Property Range
          $M \leftarrow M \cup \{c\}$
        end for
        $y_{ij}^{'} \leftarrow PropertyMutation(y_{ij}, M)$
        $y_{ij} = y_{ij}^{'}$
      end if
    end for
  end for
end for
return P_{t+1}, O_{t+1}

4 3D-trajectory coverage

Table 1: Updated 3D-Trajectory Coverage Results for Different Methods

Method	**Court**	**Lawn**
GARL	**11.24%**	**11.94%**
Multi-Obj GA	4.92%	8.43%
Random	3.51%	4.92%
Offline RL Fuzzer	1.41%	3.51%
Online RL	4.92%	7.03%
Surrogate trained RL with random scenario	8.43%	8.43%

5 System Architecture and Components

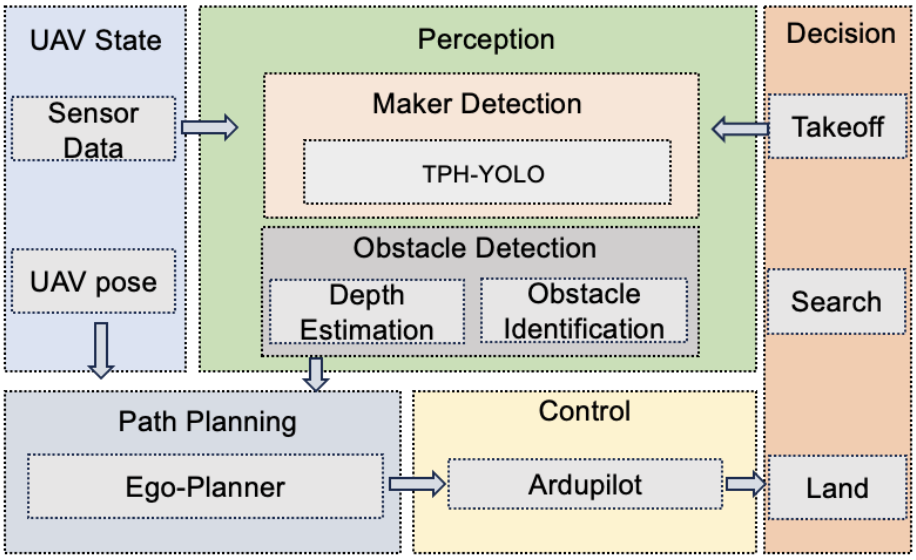


Figure 4: The system architecture of MM-MLS

Landing Systems	**UAV State**	**Detection**		**Planning**	**Control**		**Decision**
		OpenCV [2]	TPHYolo [5]	Ego-Planner [4]	Airsim [3]	Ardupilot [1]	
OpenCV-MLS	✓	✓			✓		✓
TPHYolo-MLS	✓		✓		✓		✓
MM-MLS	✓		✓	✓		✓	✓

Table 2: The component for each landing system

6 High-level overview of GARL

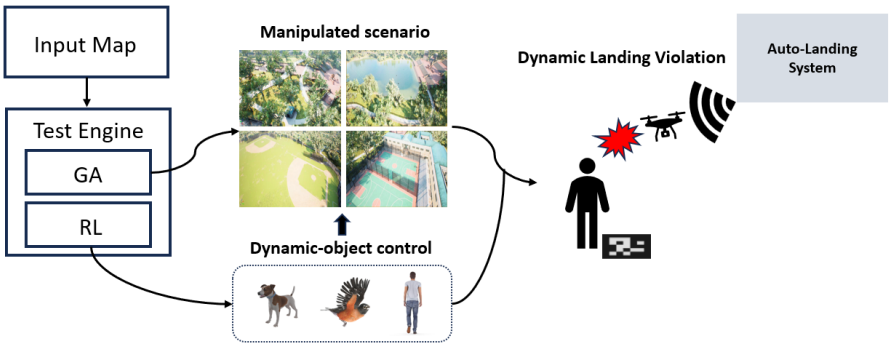
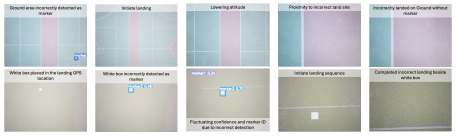
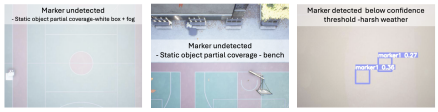


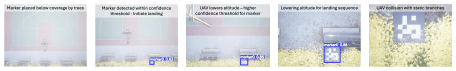
Figure 5: A high-level overview of GARL framework, including the simulator and the search engine.



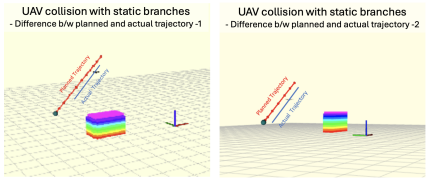
(a) Examples of Violation Type I: UAV lands on a false positive location due to wrong detection



(b) Examples of Violation Type II: No marker is detected

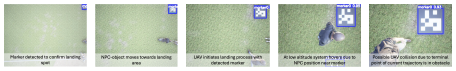


(c) Examples of Violation Type III: Static Object Collision

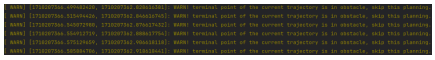


(d) Examples of trajectory difference visualized from Rviz

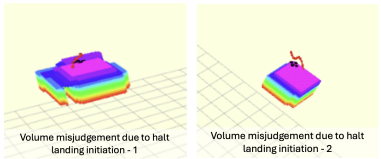
(e) Examples of Violation Type IV: Dynamic Object Collision



(f) Examples of Violation Type V: Planner Crash



(g) Example of crash message

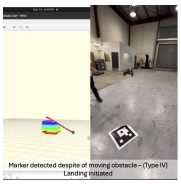


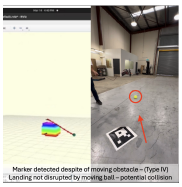
(h) Examples of the UAV stack in obstacle visualized from Rviz

Figure 6: Examples of violation from simulation

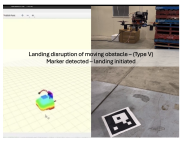
(a) False negative detection

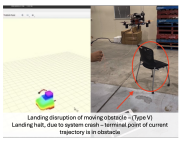
(b) False positive detection





(c) Real-world experiment for *Type IV*





(d) Real-world experiment for *Type V*

Figure 7: Example of real-world violation reproduction

7 Violation Example in Simulation

8 Real-world Testing Examples

9 Dynamic Object Trajectory Example

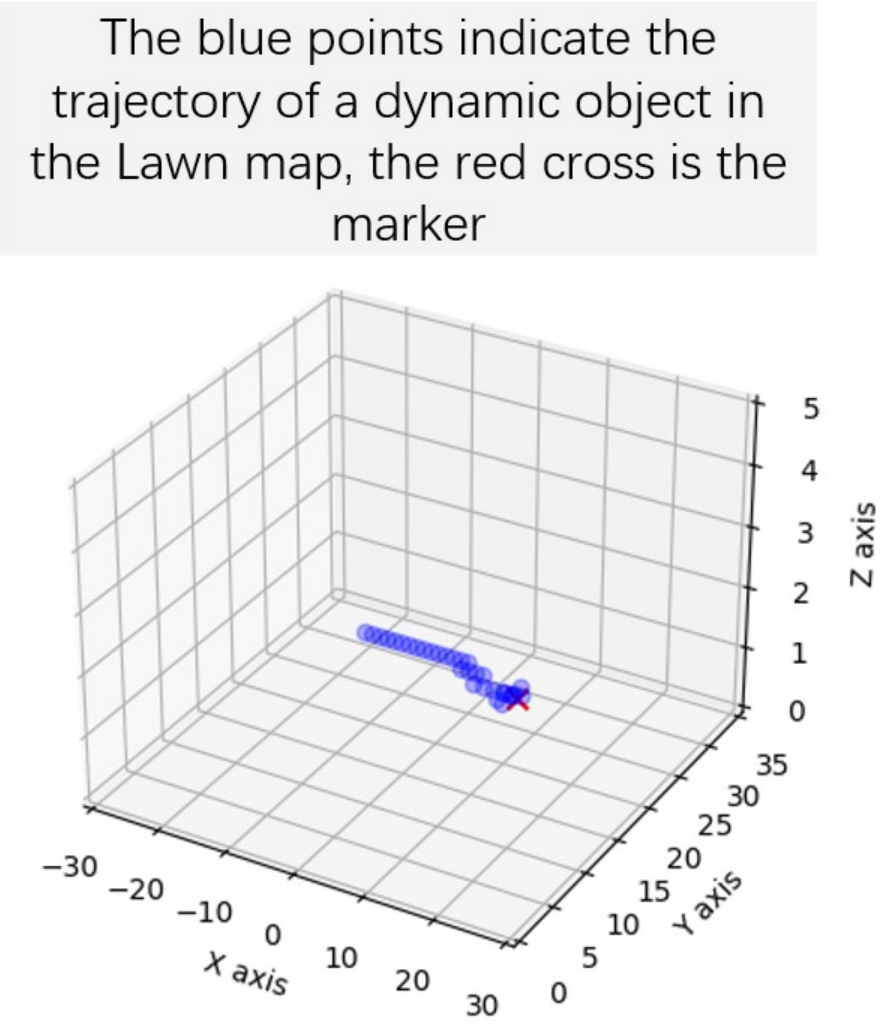


Figure 8: The trajectory of a person in the lawn map

References

[1] Ardupilot, an open source autopilot, https://github.com/ArduPilot/ardupilot.

[2] Bradski, G.: The OpenCV Library. Dr. Dobb's Journal of Software Tools (2000)

[3] Shah, S., Dey, D., Lovett, C., Kapoor, A.: Airsim: High-fidelity visual and physical simulation for autonomous vehicles. In: Field and Service Robotics (2017), https://arxiv.org/abs/1705.05065

[4] Zhou, X., Wang, Z., Ye, H., Xu, C., Gao, F.: Ego-planner: An esdf-free gradient-based local planner for quadrotors. IEEE Robotics and Automation Letters **6**(2), 478–485 (2020)

[5] Zhu, X., Lyu, S., Wang, X., Zhao, Q.: Tph-yolov5: Improved yolov5 based on transformer prediction head for object detection on drone-captured scenarios. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 2778–2788 (2021)