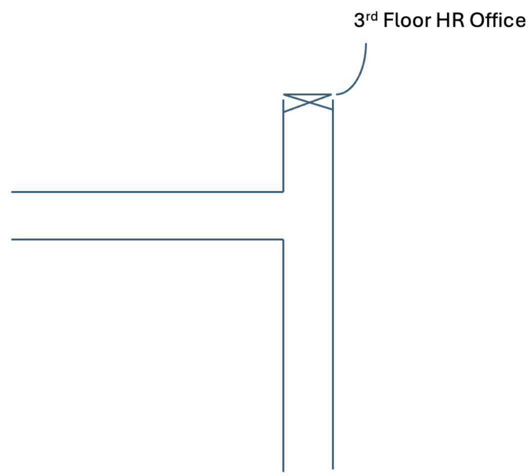
The main idea/ catch of our method over the other traditional methods is the long-horizon predictions, whereas the traditional method does short-horizon predictions.

The Advantage we have to show is that our method stops way before they get into the trap. For eg: The planners predict the dead end only in like 0.5- 1m after going into the region, and our method will be able to predict the dead end region 2-5 meters ahead and trigger the recovery mechanism.

Scenarios :

Scene 1 - T-junctions scenario (Mapless Environments)

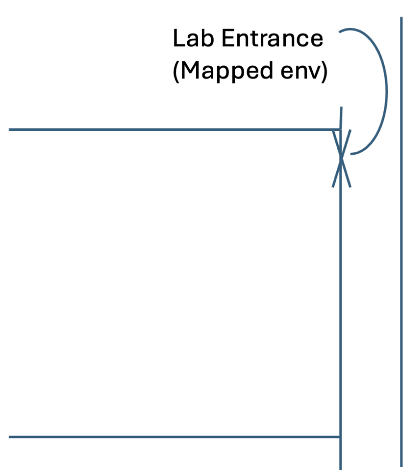


Test the baselines first - We want to see the traditional planner going inside the trap and then releasing that is a dead-end.

Our Method - Sees the dead end at the intersection at the start and should take a Left

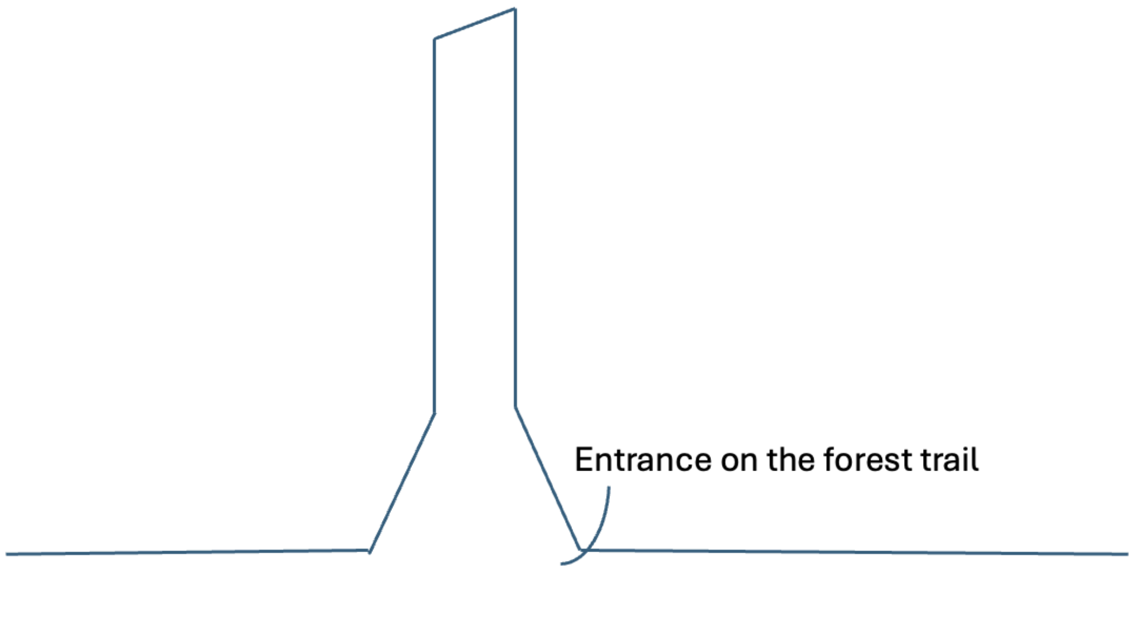
Scene 2 - Corridor-to-Room (Mapped Environment)

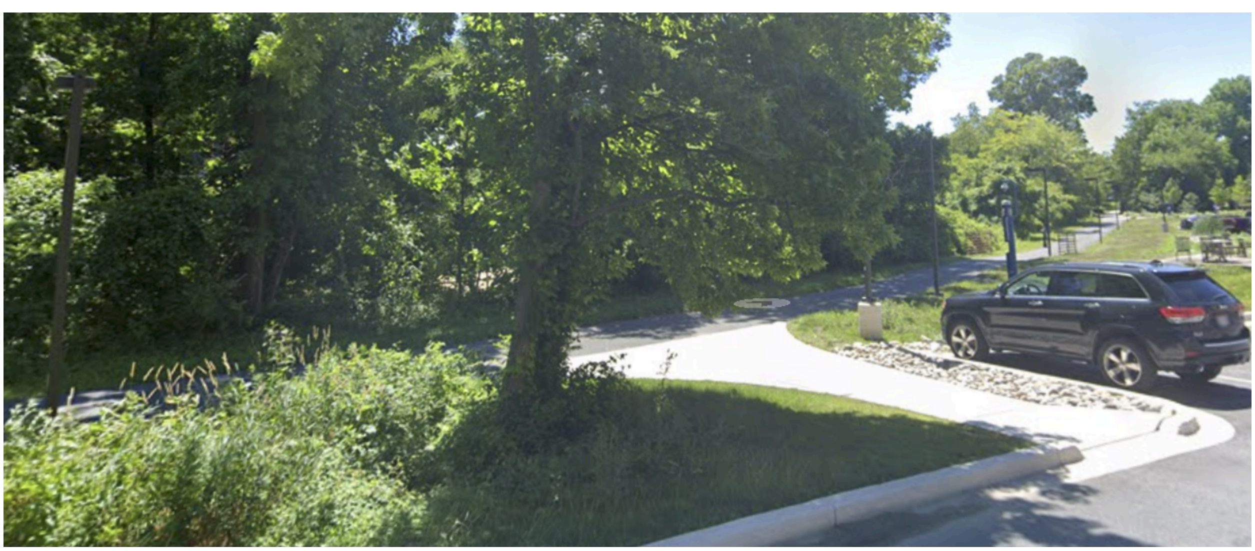
Create a static map of a corridor with a closed doorway leading into a room. In this map, the doorway is permanently closed. And give the goal inside the room and open the door at times when the robot is trying to go inside, I want to show that even though there is a map, the robot is able to understand the semantics of the environment and act accordingly.



Scene 3 - Outdoor Scene - A long outdoor pathway that eventually terminates in a dead-end (Obstructed by a Dense vegetation/tree) - Mapless Navigation

- There is a small opening opp to the parking lot GG1, which leads to the river and the path is not traversable. And our method helps the robot to slow down and eventually decide to turn around much earlier than the baselines.





Ablation Study:

1. Run with LiDAR Only
2. Dr.Nav without the cross-attention model

Metrics:

1. Success rate - Dead-end identifications
2. Time to goal - Time to reach the goal
3. Pre-emptive Avoidance Distance (PAD) - The distance between the robot and the dead-end's back wall at the moment the robot decides to turn around or change its path.
4. Negative Progress Ratio (NPR) - The total distance the robot traveled *into* a dead-end and then had to travel *back out* of to resume the correct path.
5. Recovery Smoothness - How smooth can it recover without any jitter

Comparisons Methods:

1. Traditional path planning method
    a. DWA
    b. MPPI
    c. Nav2- DWB
2. Semantic Understanding
    a. Gemini
    b. GPT 5
    c. CLIP

Work Plan -
I think we can start Scene 2 and continue to get the results for Scenes 1 and 3. I have created a table so that we can share the work

Tasks	Gershom	Pon Ashwin	Vignesh
Static map creation	✔️	✔️	
Nav2 DWB planner	✔️	✔️	
DWA Planner	✔️	✔️	
MPPI Planner Testing	✔️		
MPPI Code Debug		✔️	
Goal Generator Code Debug			✔️
Model Fine-tuning if required	✔️		✔️
Ablation study	✔️	✔️	
Paper Corrections			✔️

Detailed steps about the Scenes:

Scene 2: Lab Hallway (Dynamic Re-entry) - Static map

a. **Goal:** Demonstrate that DR. Nav can identify new traversable paths that are marked as "Blocked" on a static map.
a. **The Tasks:**

i. Generate a map of the lab hallway with the door fully closed. The costmap should show a solid wall at the entrance.
ii. **Baseline Setup**: Deploy Nav2 DWB, DWA, and MPPI.And set the navigation goal inside the lab. Open the lab door completely. Observe how the global planner fails or refuses to create a path because the "static map" is blocked.
iii. If it does move, show it "hovering" outside the door, unable to enter because the costmap inflation from the old map still thinks there is a wall there.
iv. **Our method** - As the robot approaches the door, the RGB-LiDAR fusion should recognize the Open Doorway features.
    1. Verify the Bayesian Log-Odds update the semantic costmap from "Blocked" (from the map) to "Open" (from live sensors).

b. **The Data Metrics:**
    i. Time-to-Goal - The baselines will likely have a "Time-to-Goal" of infinity (failure), while DR. Nav succeeds.
    ii. Re-classification Latency -Time from when the camera first sees the open door to when the robot changes its path to go inside.
    iii. Success Rate: Count how many times each planner successfully enters the room.
    iv. Semantic Map Visualization - Capture a screenshot showing the Grey/Black wall on the static map being overridden by your Green/Open semantic prediction.

Scene 1: Indoor (Unmapped) – The Recovery Process

a. **Goal:** Specifically to address Reviewers 1 & 2's demand to see the Recovery Mechanism in action.
b. **Tasks:**
    i. Use the waypoint generator in a T-junction
    ii. Drive the robot into the dead-end until the EDE (Expected Dead-End Exposure) is high.
    iii. Force a recovery**.** The robot must stop, look up the nearest stored recovery point, and set it as the new Nav2 goal.
    iv. Record Recovery Success Rate (did it get back to the hallway safely?) and Path Smoothness (was it a smooth U-turn or a jerky spin?). Take a couple of screenshots and videos.
c. **Data metrics**
    i. Path Smoothness: Measure the curvature/jerkiness of the U-turn.
    ii. Compare our "Point-Return" vs. standard "Spin-in-Place" recoveries.

Scene 3: Outdoor (Unmapped) – Proactive Avoidance

a. **Goal:** Prove we detect a dead-end visually before LiDAR sees a closing/end.
b. **Tasks**:
    i. **Baseline Setup**: Deploy Nav2 DWB, DWA, and MPPI. Set the global goal 10m past the end of the path.
    ii. **Debugging**: If the robot doesn't turn, check the EDE threshold and the Bayesian update rate.
c. **Data metrics**

i. Pre-emptive Avoidance Distance (PAD)- Distance from the robot to the wall when it first stops or begins its turn.
ii. Negative Progress, Total distance traveled *into* the dead-end before retreating.
iii. Outcome: Baselines should drive until they are ~1m from the end (LiDAR range). Our method should react at ~5m+ (Camera range).

Specifics for the Algorithms:

1. MPPI: Please ensure the critics are set to default. We need the "Standard" version for testing.
2. Run one test "Without Cross-Attention" of our method(Dr.Nav) (just RGB+LiDAR concatenation) for our ablation study.
3. Logging: For every run, we need a .bag or .mcap file containing:
   a. /odom (for distance travel)
   b. /cmd_vel (to show smoothness)
   c. Our custom /ede_score or /semantic_costmap topic.