Deep Reinforcement Learning for Safe Local Planning of a Ground Vehicle in Unknown Rough Terrain

By Shirel Josef and Amir Degani

Published in 2020

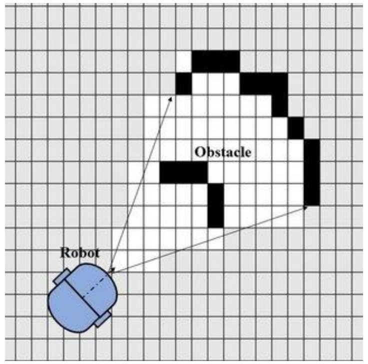
Guillaume Genois, 20248507

IFT 6757

Université de Montréal

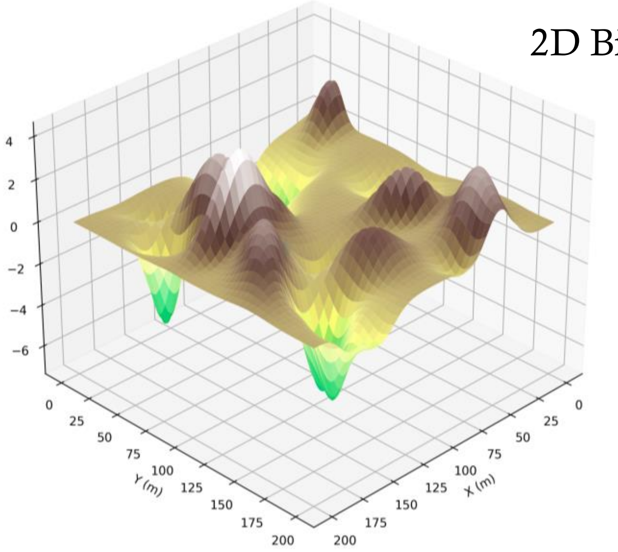
ROUGH TERRAIN NAVIGATION

- Applications: Exploration, search and rescue, agriculture
- High level of uncertainty
- Continuous obstacles

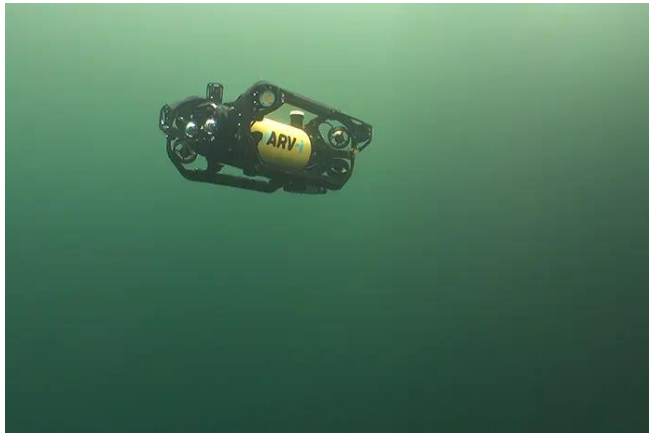


2D Binary Surface

3D Continuous Surface



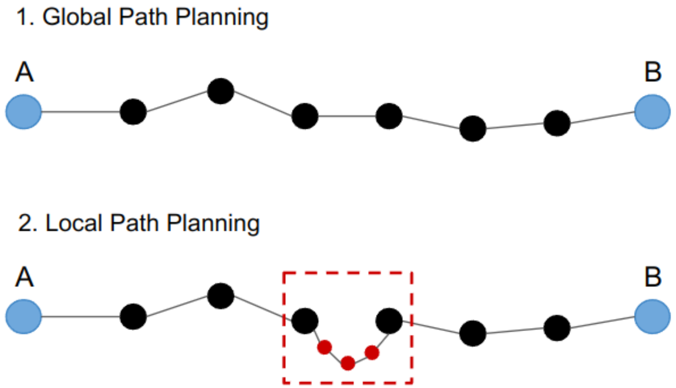
Weed Cutter AGV

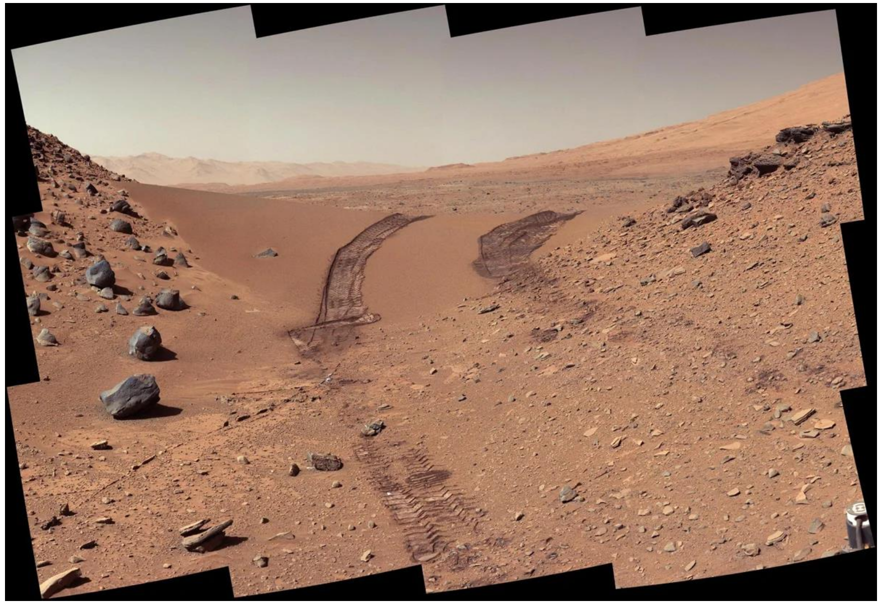


Automated Underwater Vehicles (AUV)

LOCAL PLANNING IN UNKNOWN TERRAIN

- Classic method requires prior knowledge of the terrain
- Global planning not possible
- Require multiple dynamic corrections with new data
- Planning path toward target based on kinematic constraints and surface geometry





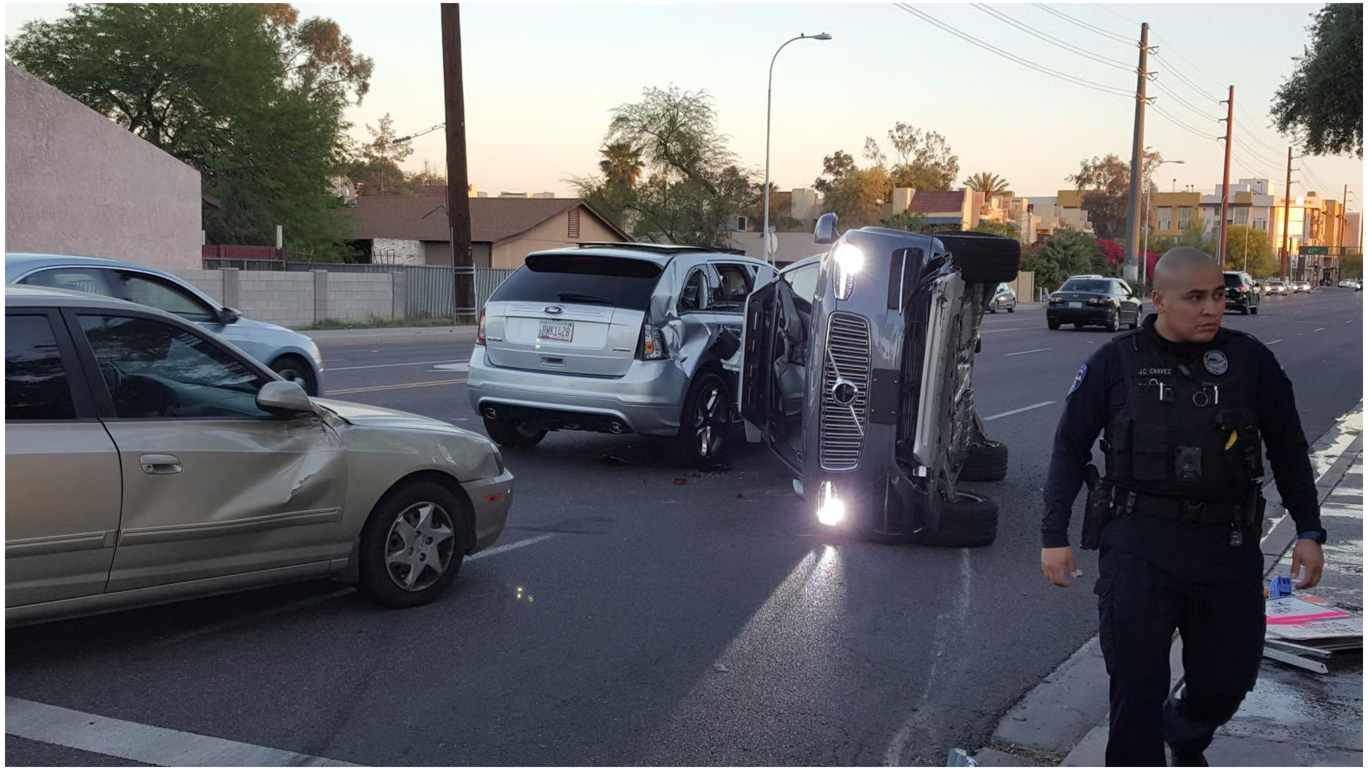
Curiosity Rover on Mars

Up to 20 minutes for signal to get to Earth.

UNMANNED GROUND VEHICLE CHALLENGES

- Ensure safety
- Start position to goal position
- Dodge obstacles and pitfalls
- Move in ascending ground

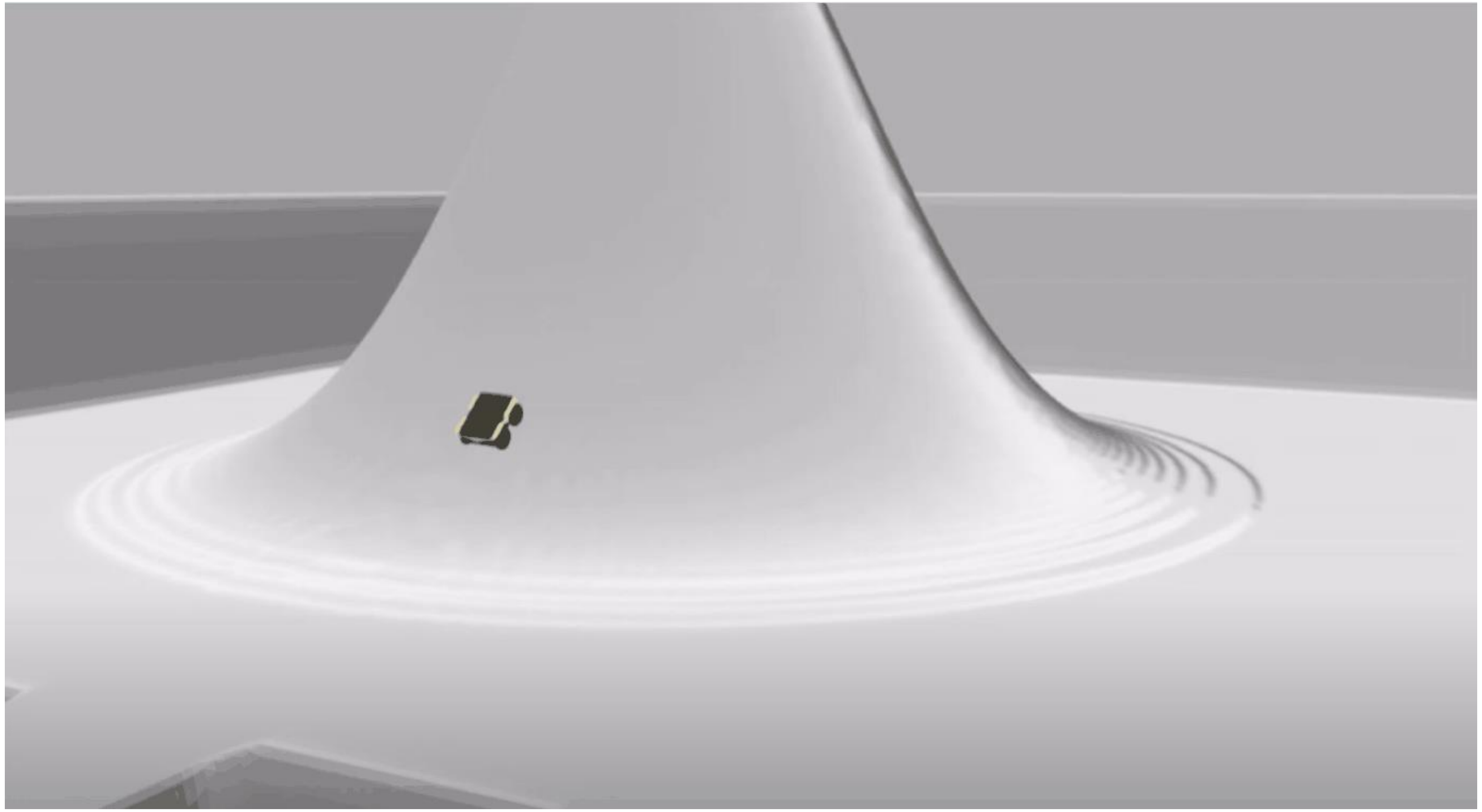
Self-delivery robot in pitfall



Uber self-driving car flipped

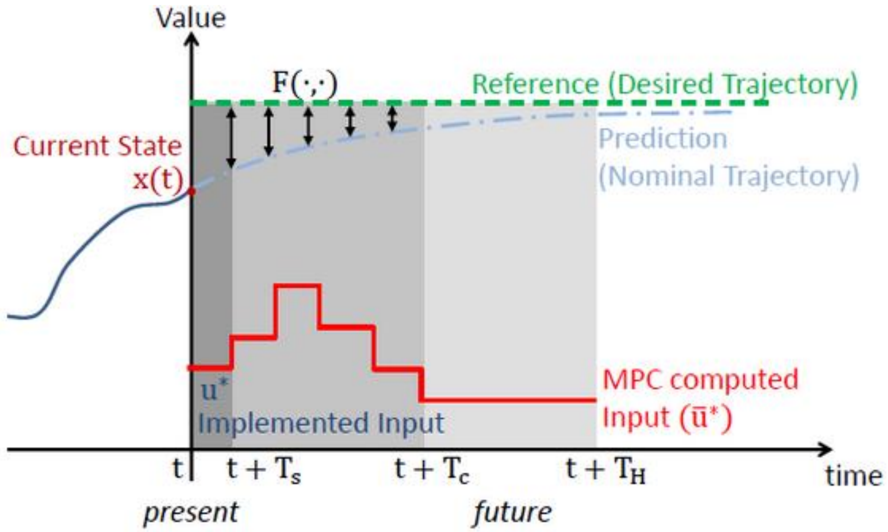
* No Duckiebot was harmed during the making of this presentation.

UNMANNED GROUND VEHICLE CHALLENGES



MARKOV DECISION PROCESS (MDP)

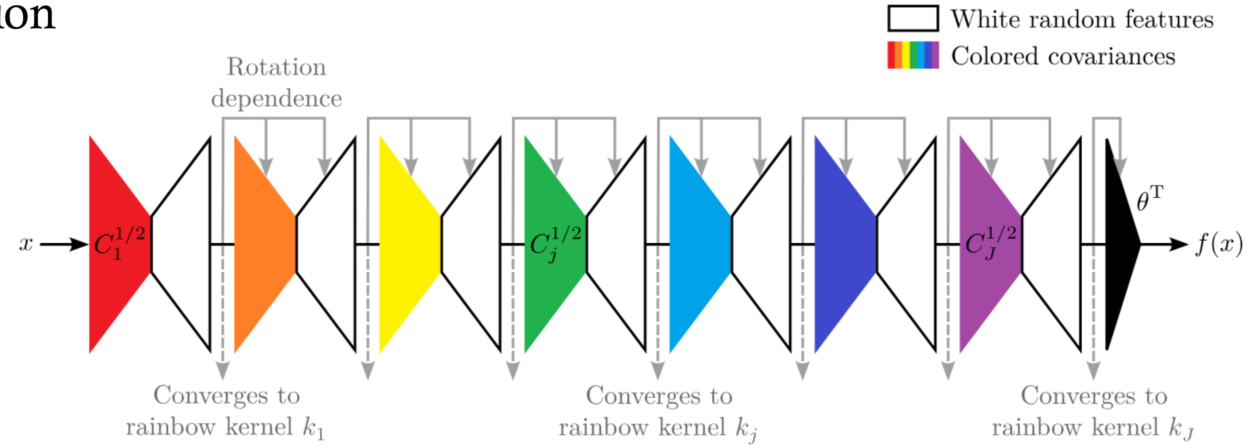
- Agent interacts with environment to maximize cumulative reward
- State (s_t), Action (a_t), Reward (r_{t+1})
- Discounted return: $G_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1}$
- Action-value function $q(s, a)$:
  expected return from state-action pair
- Goal: Learn optimal policy $\pi^*(a|s)$ that maximizes expected return



DEEP Q-NETWORK & RAINBOW

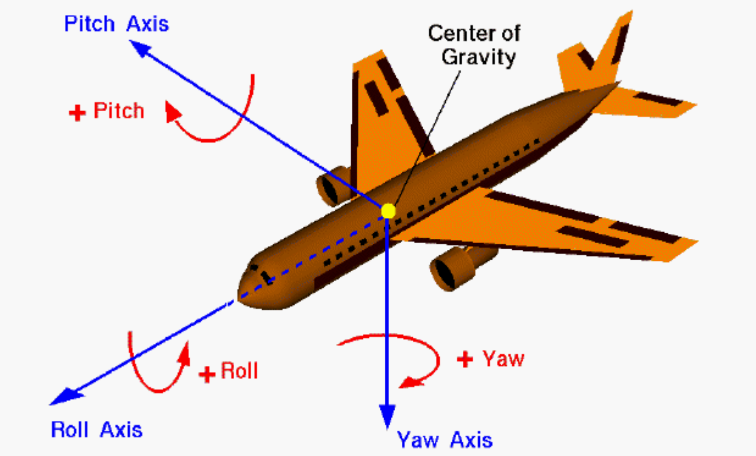
- Uses deep neural networks to approximate optimal action-value function $\hat{q}^*(s, a)$

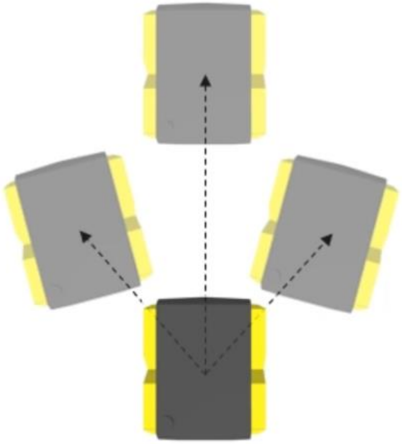
- Rainbow integrates multiple DQN improvements:
  - Improved action-value function estimation
  - Increased sample efficiency
  - Better handling of sparse rewards
  - More robust exploration
  - Achieved state-of-the-art performance

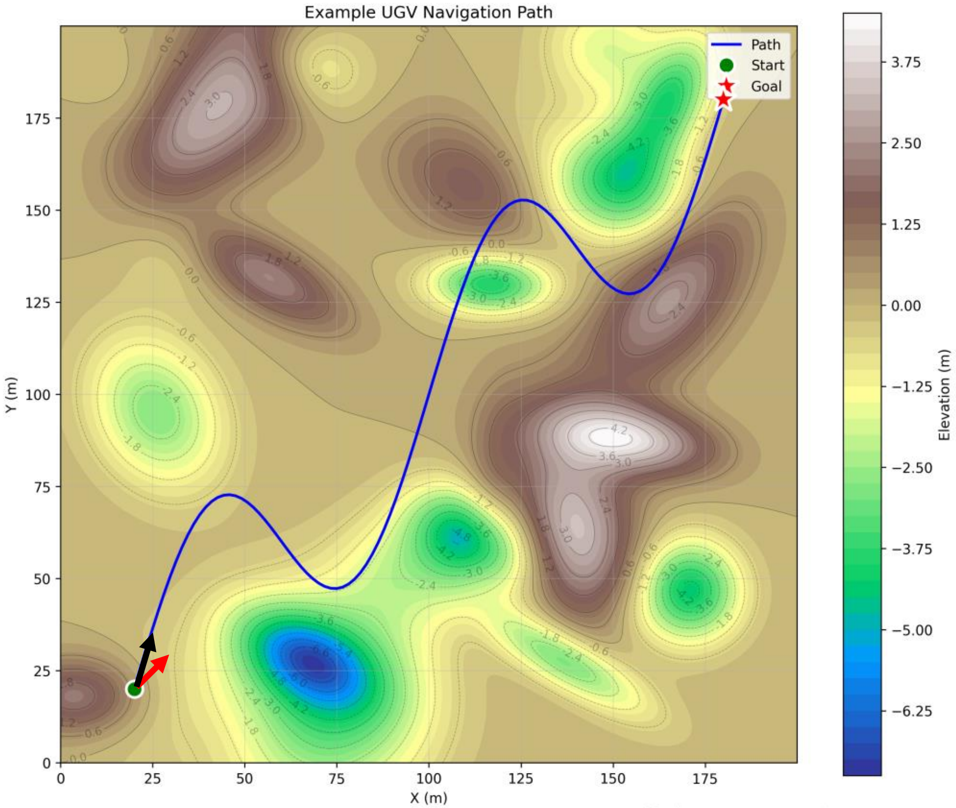


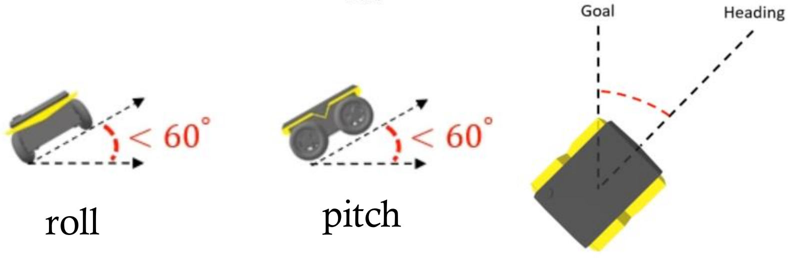
PROBLEM FORMULATION

- UGV position: (x, y, yaw) configuration
- Three actions: forward, right, left
- Safety constraints: $roll < 60°$, $pitch < 60°$
- Goal: construct safe path from start to goal position
- Only angle to goal relative to heading is known





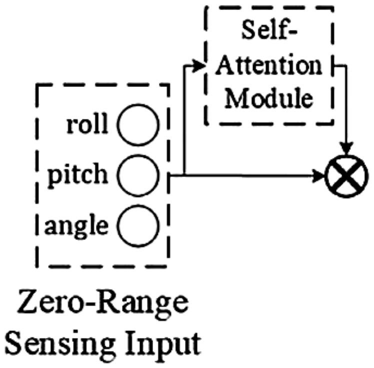


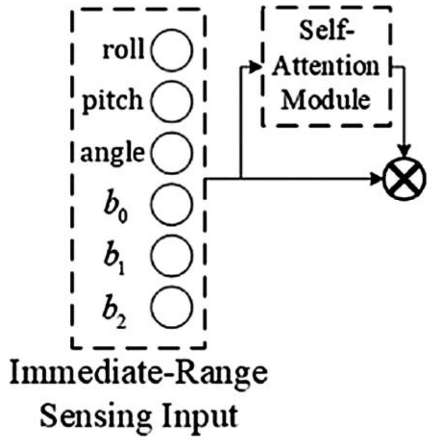


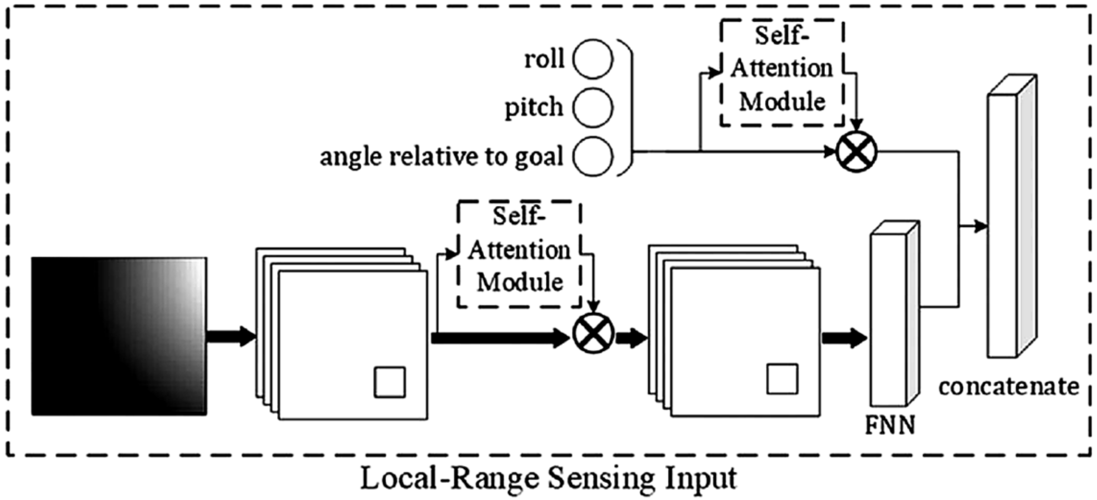
X-RANGE SENSING INPUT

1. Zero-range: IMU only (roll, pitch, angle to goal)
2. Immediate-range: one-step look-ahead binary traversability
3. Local-range: $3.2m \times 3.2m$ elevation map + IMU data

- Different inputs tested for various sensing capabilities

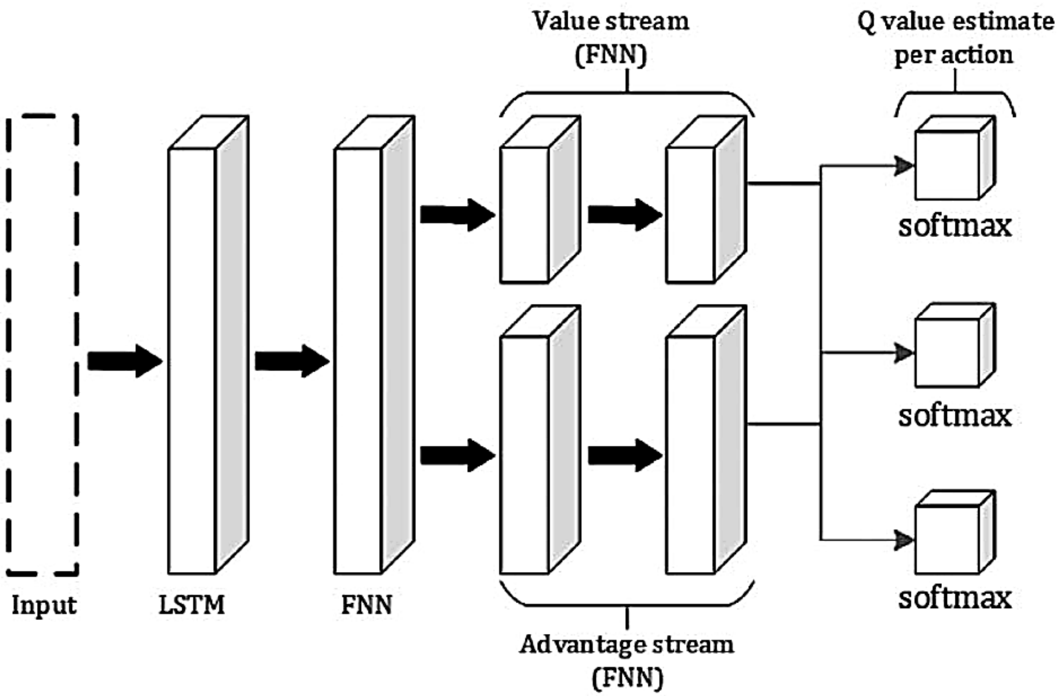
1.


2.


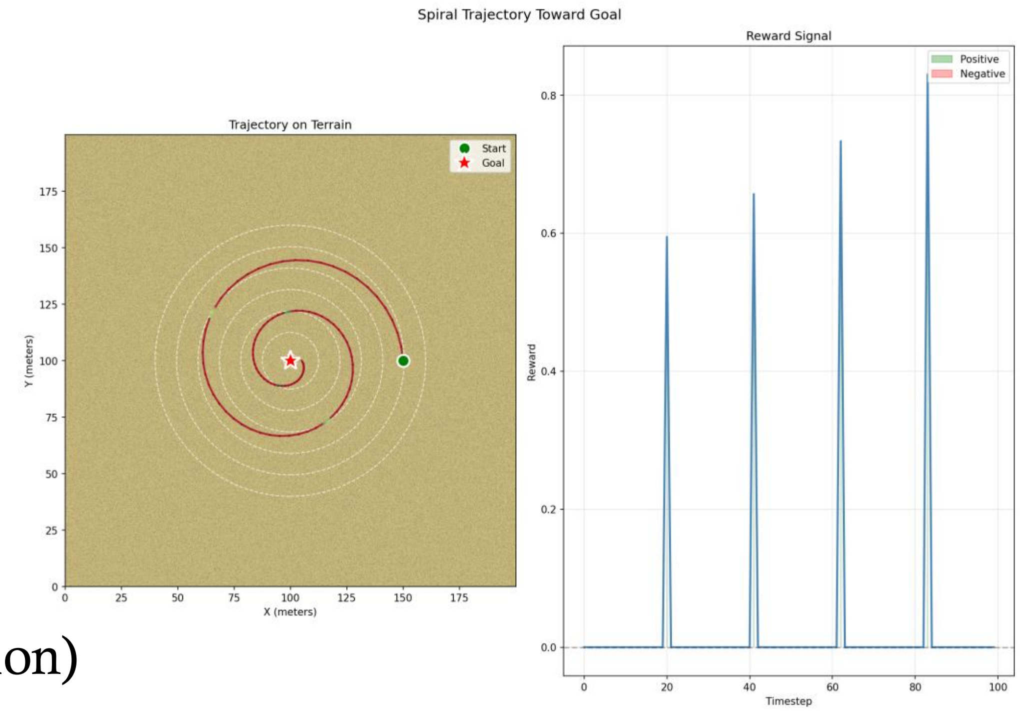
3.


NETWORK ARCHITECTURE

- Based on Rainbow agent architecture
- LSTM layer to estimate underlying system state
- Fully-connected layer after LSTM
- Split into advantage and value streams (dueling networks)
- Outputs distribution of q-value estimates for each action
- ε-greedy exploration (Noisy Networks omitted)



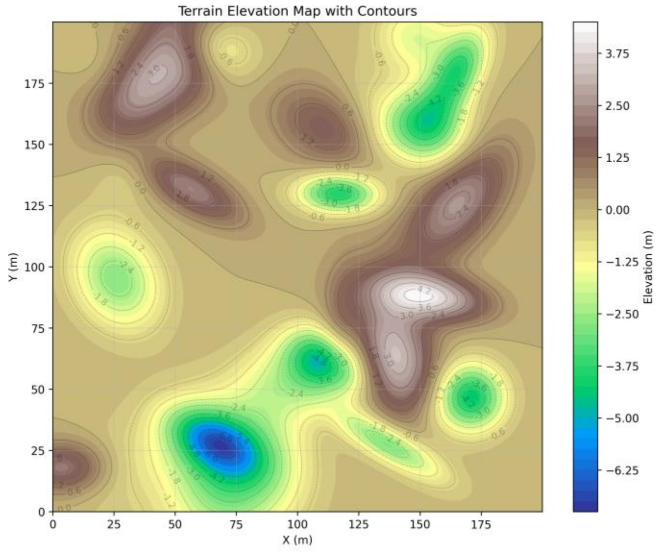
REWARD FUNCTION



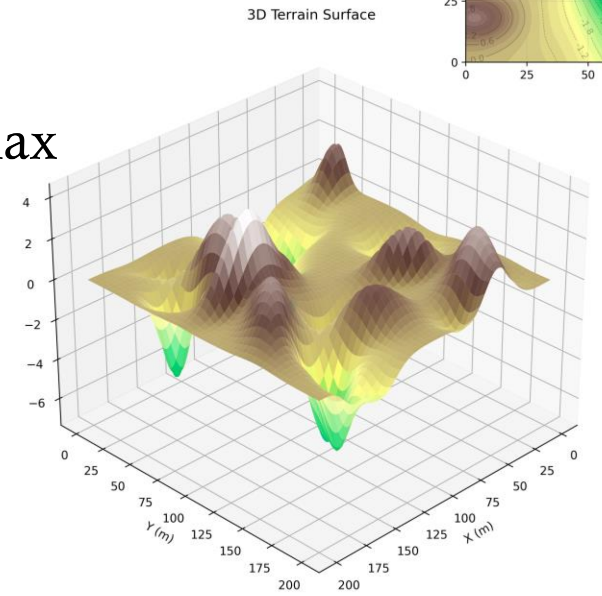
- Inspired by round shooting target structure
- Goal surrounded by n concentric rings
- Positive reward (R_{ring}) for entering inner ring
- Negative reward for exiting ring (prevents exploitation)
- Rings closer to goal have higher reward (controlled by α)
- R_{goal} received when reaching ε-environment around goal
- Episode terminates if roll/pitch exceeds safety threshold

$$r_t = \begin{cases} \alpha \cdot R_{\text{ring}}, & \text{enters a ring} \\ -\alpha \cdot R_{ring}, & \text{exits a ring} \\ R_{\text{goal}}, & \text{enters } X_\varepsilon \\ 0, & \text{elsewhere} \end{cases}$$

MODEL TRAINING AND EVALUATION



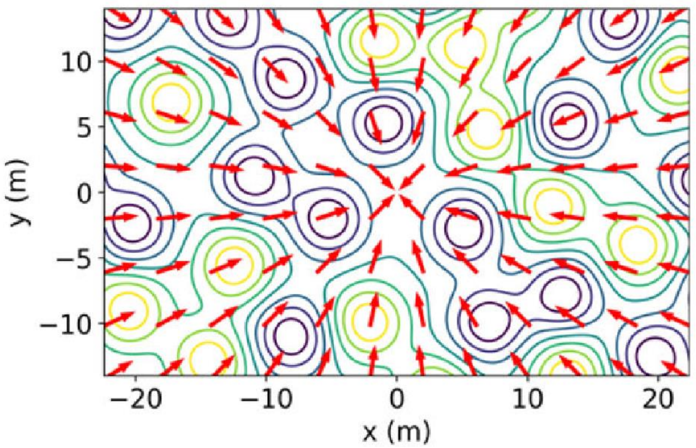
- 200 × 200 m² terrain with 0.025 × 0.025 m² cells
- Random Gaussian hills and valleys for continuous terrain
- Random start/goal positions each episode
- Max 1,500 timesteps per episode, 100,000 episodes max
- Evaluated every 10,000 episodes on 100 test positions
- Best parameters chosen



BASELINE APPROACHES

- Baseline-1: Potential field + bearing-only navigation
  - Attraction to goal + repulsion from high gradients
  - Compares to zero-range sensing

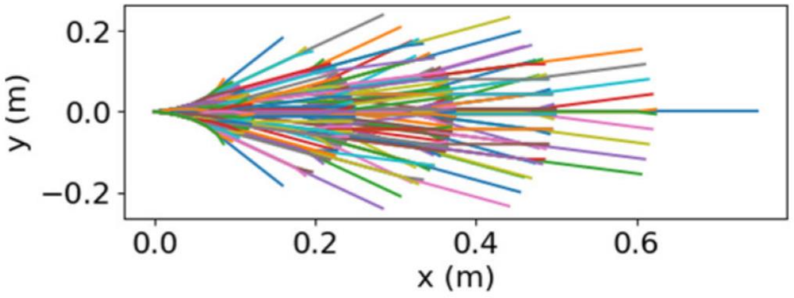
$$U(x, y, yaw) = (\text{heading}(yaw))^2 + \alpha \cdot \|\nabla T(x, y)\| \cdot (\cos(4 \cdot (\theta_G - yaw)))$$



Baseline-1 Attraction forces to goal (center)

- Baseline-2: Ego-graph with candidate paths
  - Depth-1: compares to immediate-range sensing
  - Depth-5: compares to local-range sensing
  - Evaluates path cost, executes first action only

$$\text{cost}(path) = \sum_{q \in path} \text{heading}(yaw_q) + \alpha \cdot \|\nabla T(x_q, y_q)\|$$



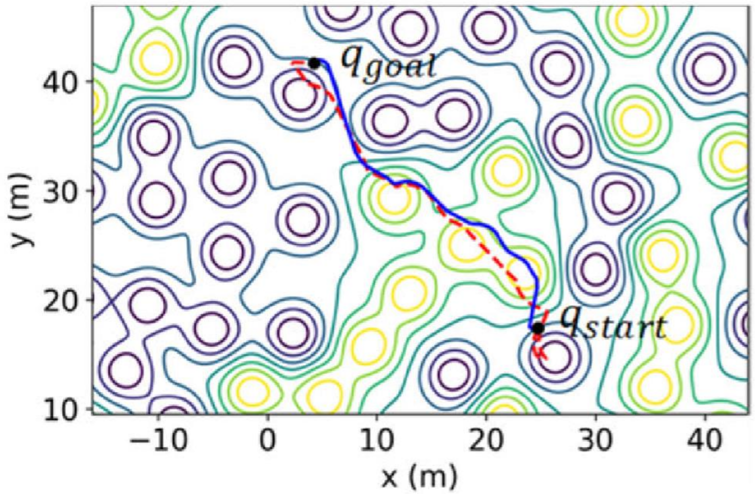
Baseline-2 Eco graph of depth 5

RESULTS

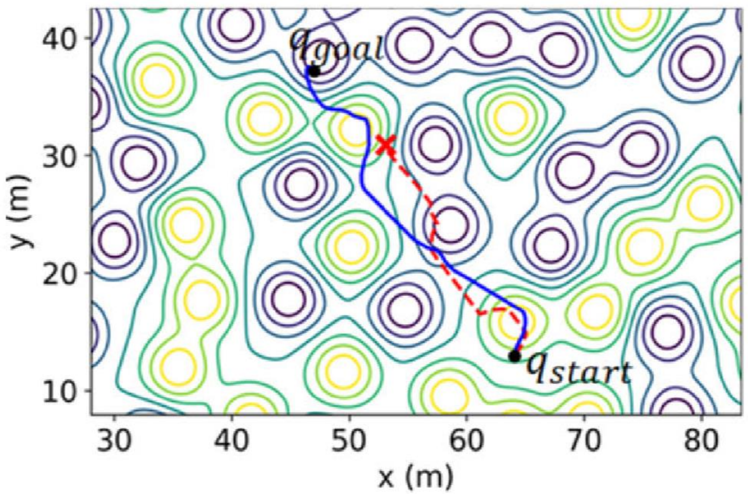
- DRL methods show significant improvement over baselines
- Similar planning time for low-dimensional inputs
- DRL exhibits more robust maneuvers and terrain understanding

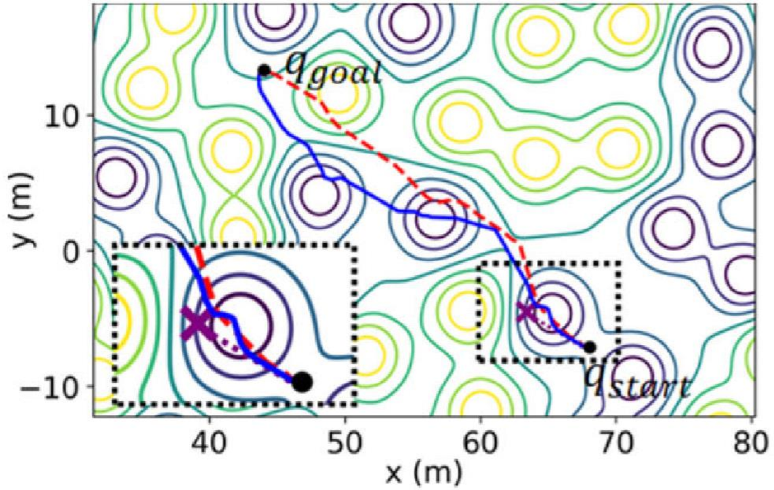
TABLE I
PERFORMANCE OVER TEST PATHS

Approach	*Success (%)*	*Avg Planning Time (sec)*
Baseline-1	26	0.15
DRL Zero-range Sensing	**48**	0.24
Baseline-2 Depth-1	61	0.73
DRL Immediate Range Sensing	**69**	0.34
Baseline-2 Depth-5	69	130
DRL Local Range (3m box centered)	**77**	1.51
DRL Local Range (3m box ahead)	**82**	1.72

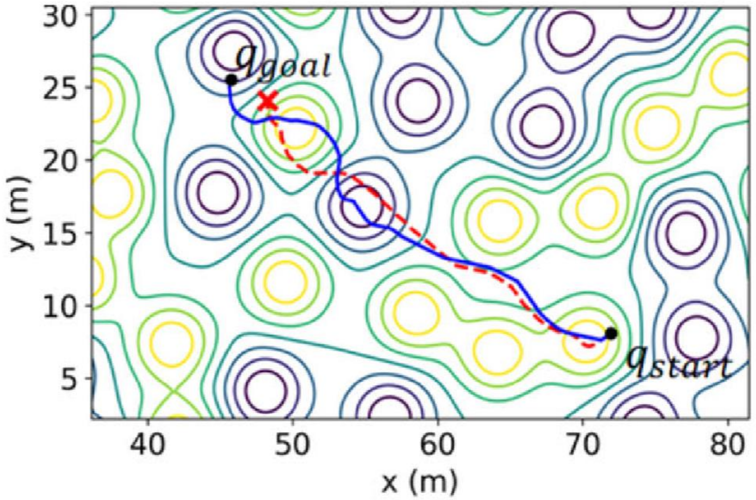


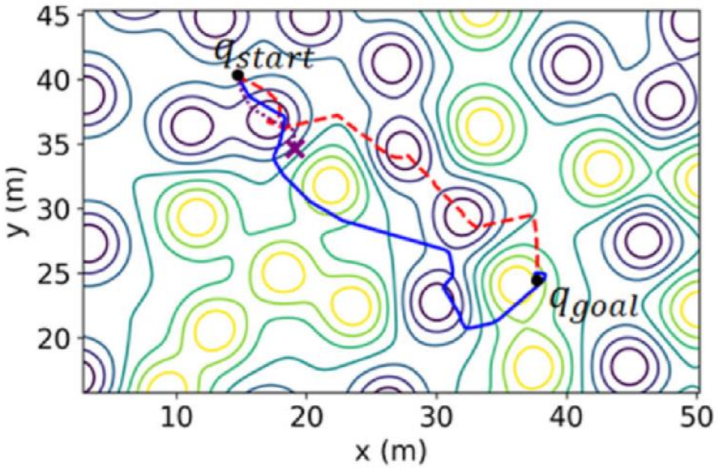
Zero Range Sensing
and Baseline-1



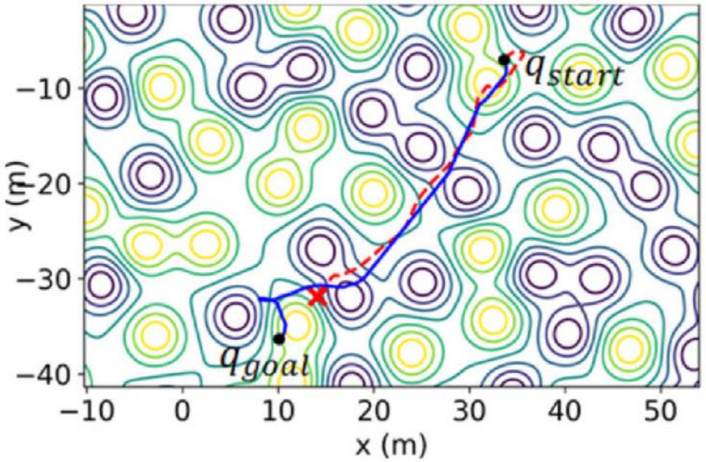


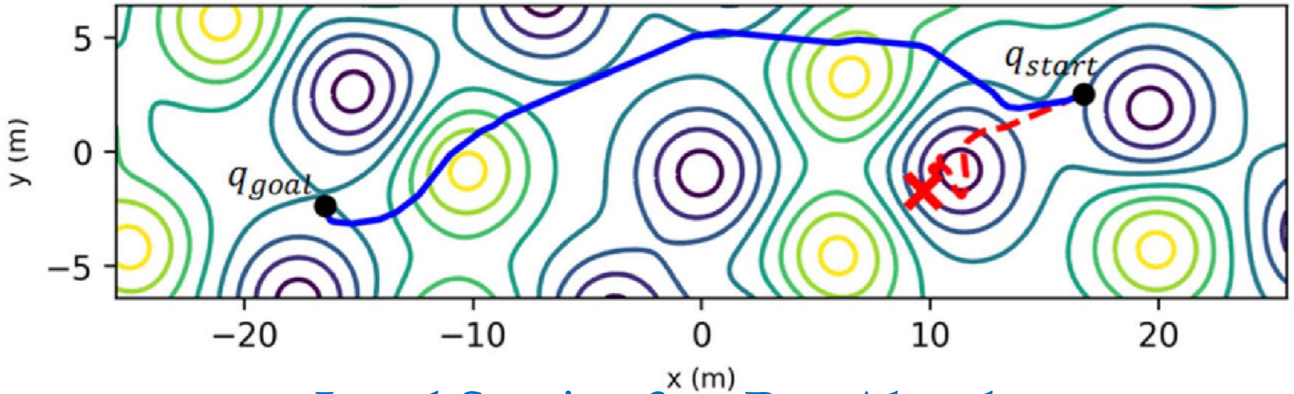
Immediate Range Sensing
and Baseline-2 (d=1)
and Zero Range Sensing





Local Sensing 3 m Box Centered
and Baseline-2 (d=5)
and Immediate Range Sensing





Local Sensing 3 m Box Ahead
and Local Sensing 3 m Box Centered

FAILURES

Two main failure modes (18% total failures):

1. Local minimum in broad high gradient areas (13%)
2. Pathological starting poses (5%)

Can be addressed with:

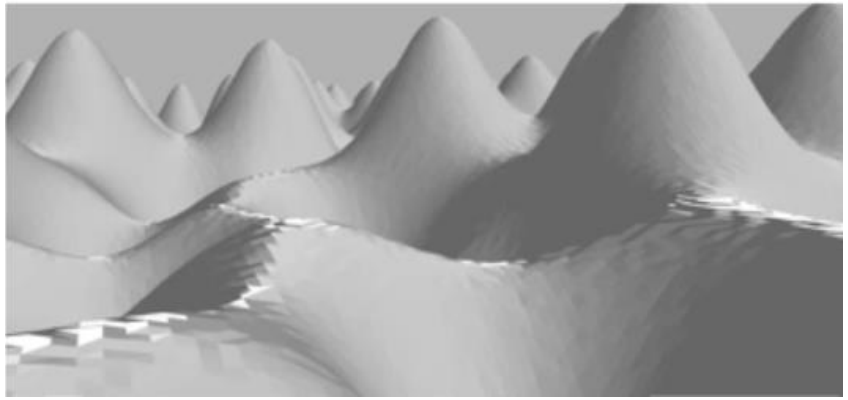
- Increasing the sensing range
- Global path planner
- Changing starting poses' distribution

TABLE I
PERFORMANCE OVER TEST PATHS

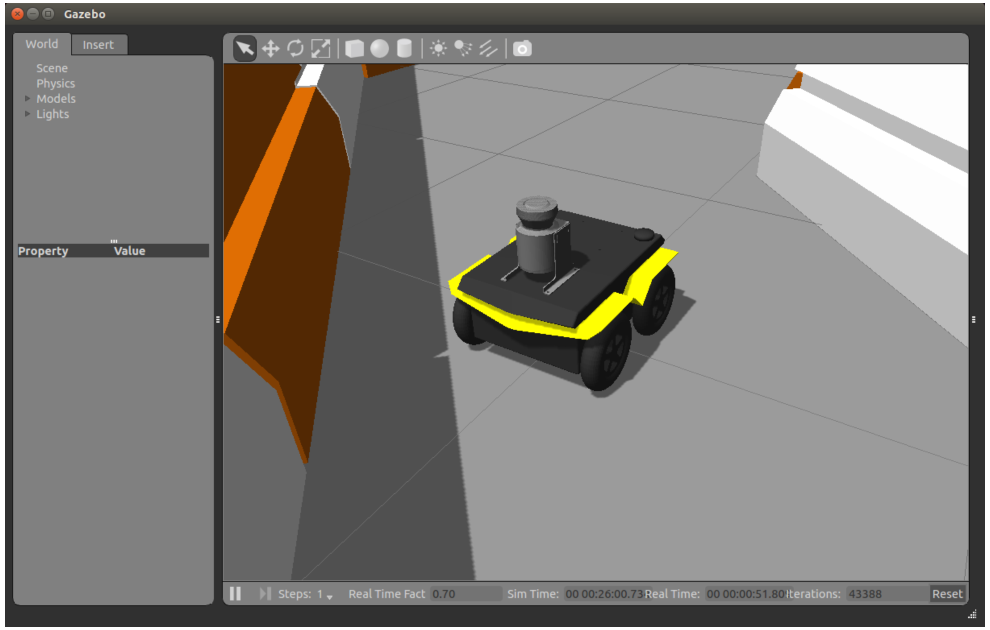
Approach	*Success (%)*	*Avg Planning Time (sec)*
Baseline-1	26	0.15
DRL Zero-range Sensing	**48**	0.24
Baseline-2 Depth-1	61	0.73
DRL Immediate Range Sensing	**69**	0.34
Baseline-2 Depth-5	69	130
DRL Local Range (3m box centered)	77	1.51
DRL Local Range (3m box ahead)	**82**	1.72

DYNAMIC VALIDATION

- Terrain modeled in Gazebo
- Vehicle simulated with a Clearpath Robotics Jackal
- Incoporate vehicle dynamics, sliding and slipping



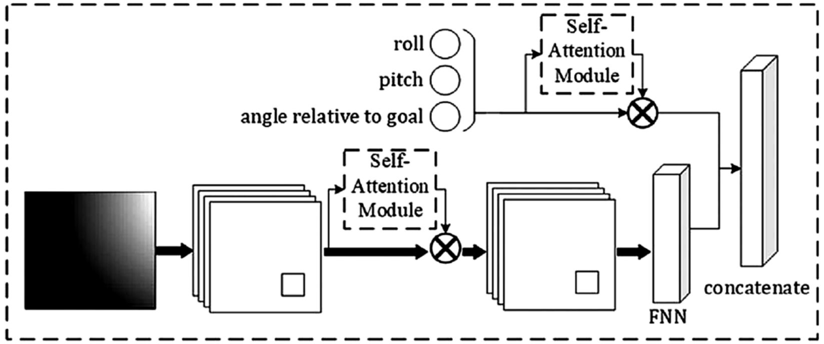
Gazebo Terrain

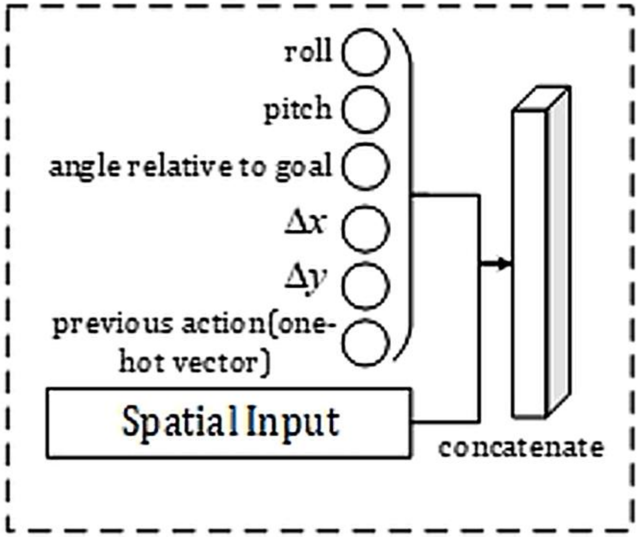


Clearpath Robotics Jackal in Gazebo

ARCHITECTURE EXTENSION AND TRAINING

- Added geometric transformation between timesteps as input
- Action encoded as one-hot vector
- Implicitly learns to compensate for wheel slip and sliding





Extended Local-Range Sensing Input

Training:

- Random friction value selected each episode
- Different start position each episode for exploration

RESULTS

- Successfully navigates training terrain with dynamics
- Generalizes to test terrain with different obstacles
- Baseline-2 depth-5 fails in dynamically challenging areas
- Extended architecture handles different friction levels

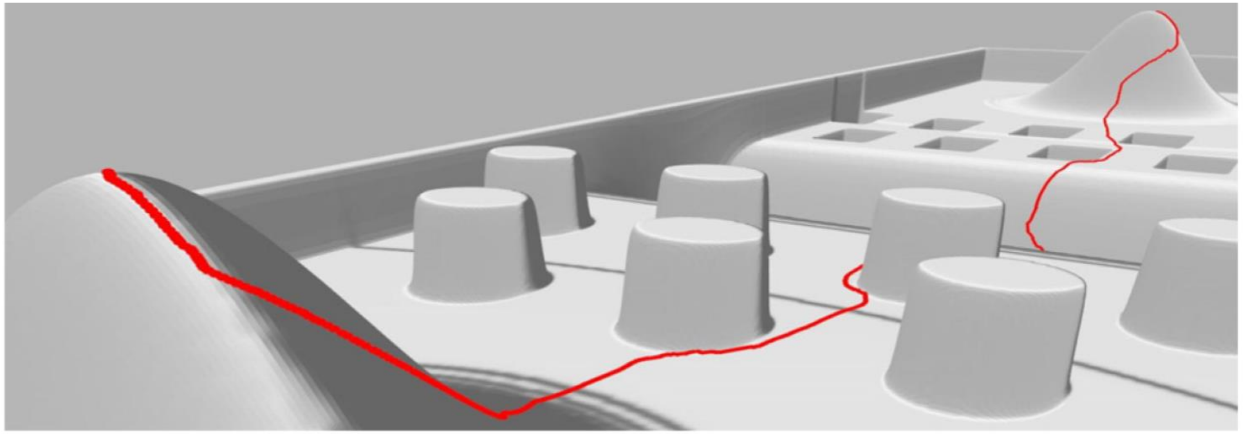
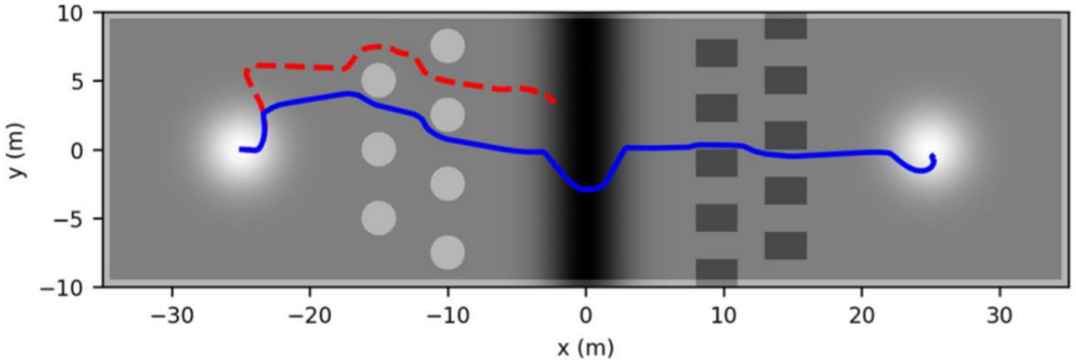


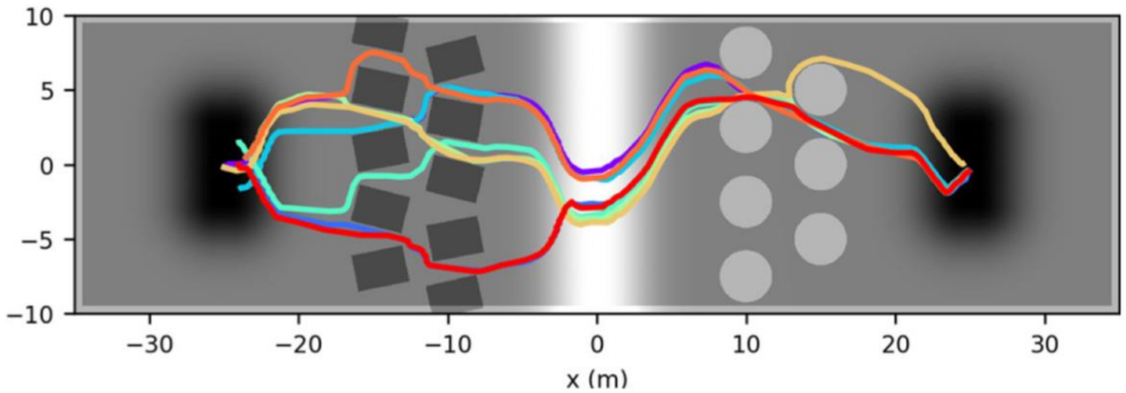
TABLE II
PERFORMANCE UNDER DIFFERENT FRICTION SETTINGS

	Approach	*Original Architecture*	*Extended Architecture*
High friction setting	***Success***	10/10	10/10
	Avg Timesteps	397.5	**381.1**
Medium friction setting	***Success***	4/10	**10/10**
	Avg Timesteps	1960.5	**628.8**
Low friction setting	***Success***	0/10	**10/10**
	Avg Timesteps	-	**1046.7**

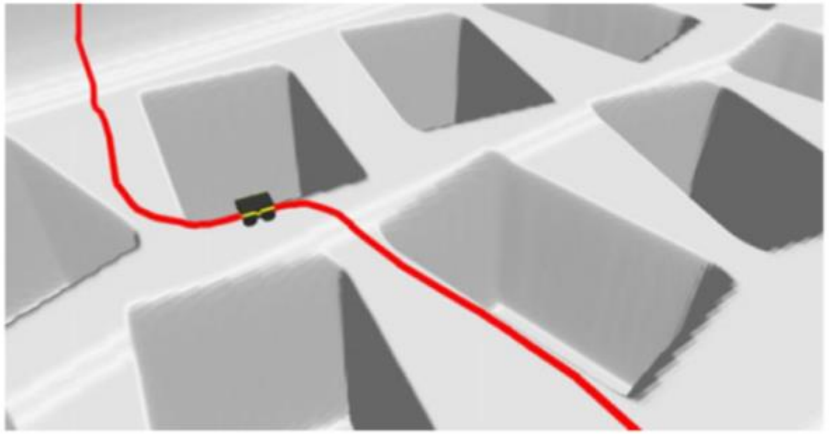
DYNAMIC VALIDATION - RESULTS

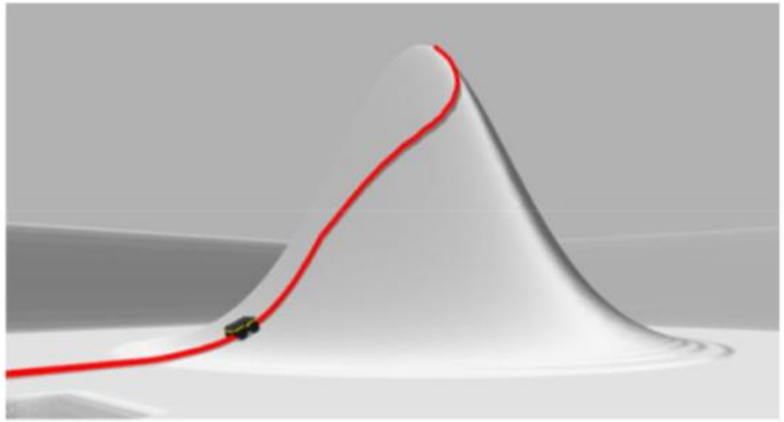


DRL Method and Baseline-2 (d=5)



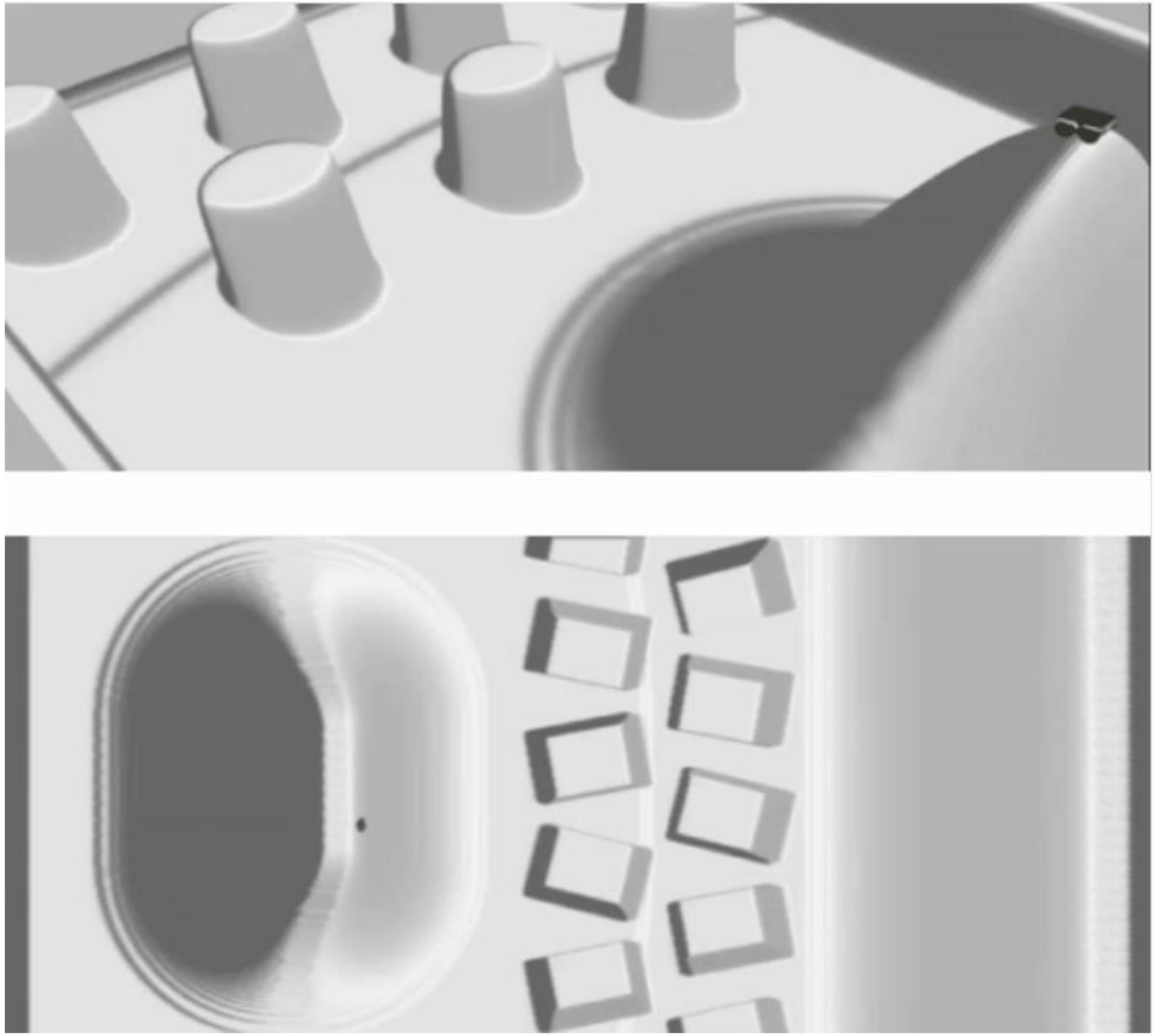
Path examples generated by DRL method





DRL method on binary and continuous obstacles

- Circumnavigates boulders
- Stays clear of hole perimeters to avoid falling
- Climbs hills with spiral trajectory to avoid high pitch

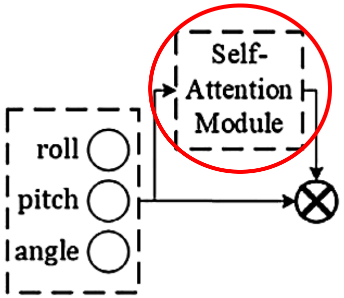


SELF-ATTENTION MODULES

- Attention mechanism models dependencies between sources
- Computes compatibility score between query and key-value pairs
- Self-attention: query and key-value from same source
- Increases explainability of learned policy

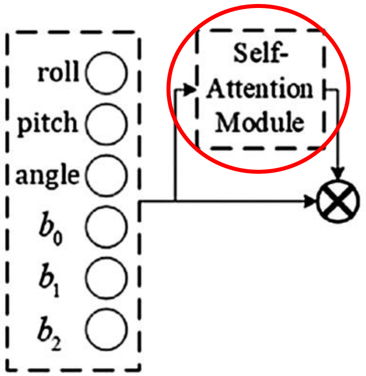
Concat attention:

$$g(x_i, q) = w^T \tanh(W(x_i;\ q))$$



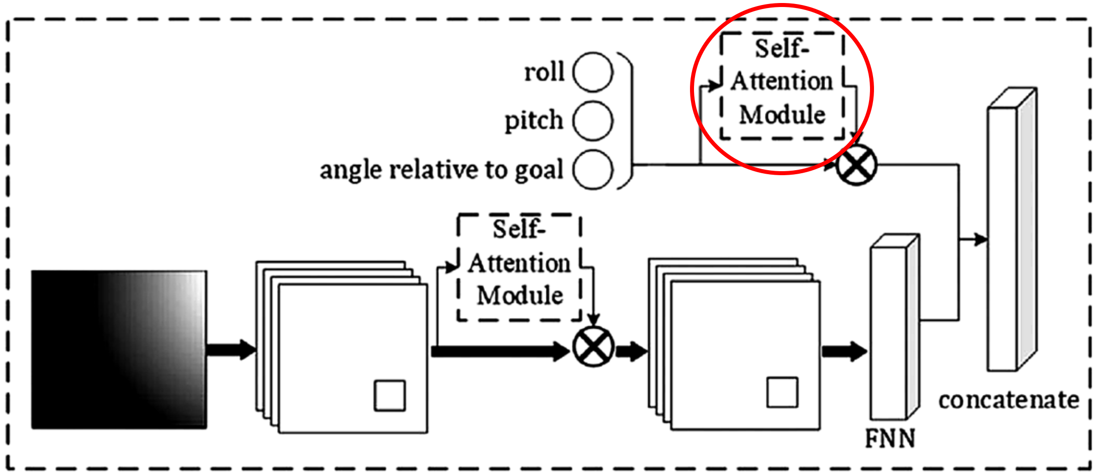
1. Zero-Range Sensing Input

2.



Immediate-Range Sensing Input

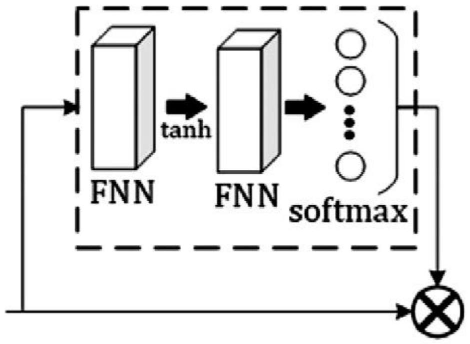
3.



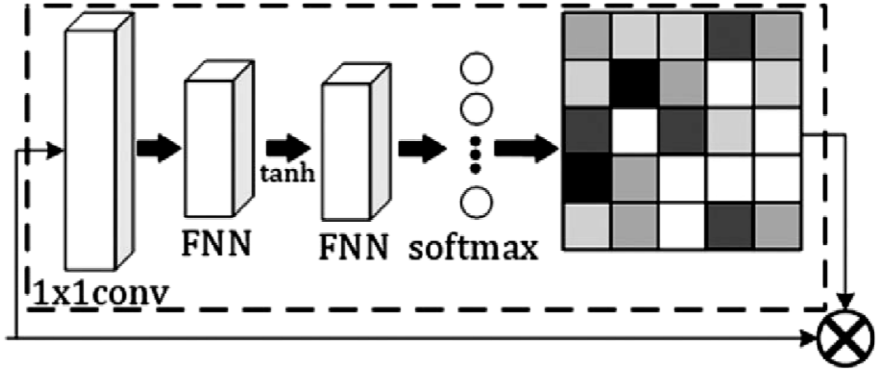
Local-Range Sensing Input

DEEP REINFORCEMENT LEARNING

- Non-spatial input: FC layer → tanh → FC layer → sigmoid
- Spatial input: 1×1 convolution added before softmax
- Element-wise multiplication of weights with input
- Visualize attention through softmax outputs from modules



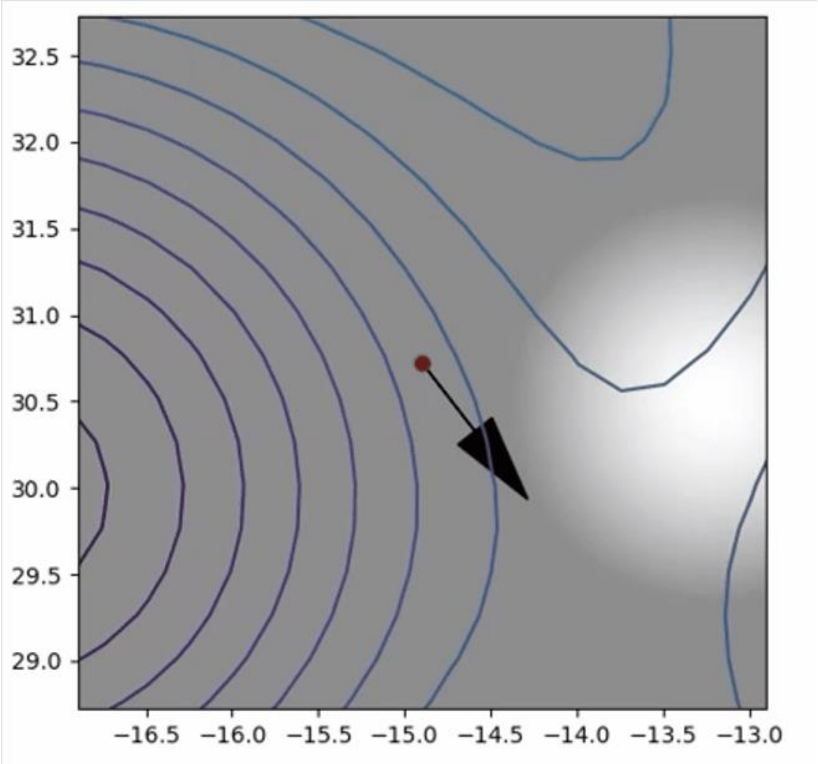
Non-spatial input



Spatial input

ATTENTION VISUALIZATION

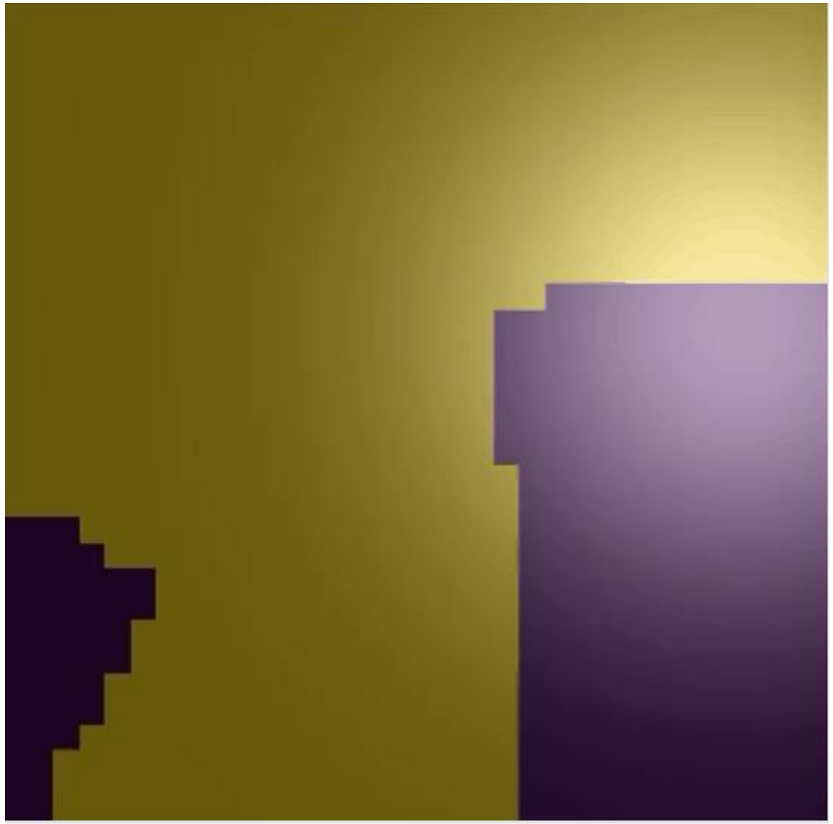
- Height map attention focused on front-left/right areas
- Attention on nearby high gradient hazardous areas
- Shifts as UGV proceeds through terrain
- Provides insight on relative importance of sensed inputs
- Helps build trust by showing what agent focuses on



Most attended area is focused on nearby high gradients

DYNAMIC VALIDATION

- Attention module also works with discrete obstacles
- Focuses on hazardous areas to avoid (e.g., nearby holes)
- Spatial input: focuses on nearby terrain features
- Non-spatial: consistent behavior across input types
- Used for searching safe areas, not for ascending/descending



Most attended area is focused on a nearby hole

CONCLUSION

- DRL approach for local rough terrain navigation
- Improved planning time and success rate vs. traditional methods
- Captures vehicle dynamics and terrain interaction
- Self-attention provides policy explainability
- Future: integrate global planner, sim-to-real transfer