Bridging the Sim-to-Real Gap: A Hierarchical Deep Reinforcement Learning Architecture for Autonomous Navigation on Embedded Edge Hardware

Mauro A. Lopez

December 2025

The Problem: Why TinyML Navigation?

- Context: Navigation in unstructured environments.
- The Conflict:
  - Classical Methods (PID/APF): Fast but brittle; stuck in Local Minima.
  - Deep Learning (DRL): Adaptive but computationally expensive (GPUs).
- Research Question:
  - Can we run a complex Deep Q-Network on a $5 microcontroller (ESP32)?

AI Theoretical Foundation

- Markov Decision Process (MDP):
  - State (S): 3 Ultrasonic distances (Normalized 0.0 - 1.0).
  - Action (A): Discrete {Forward, Left, Right}.
  - Reward (R): +100 (Goal), -50 (Crash), -0.1 (Time penalty).
- The Bellman Equation (Q-Learning):

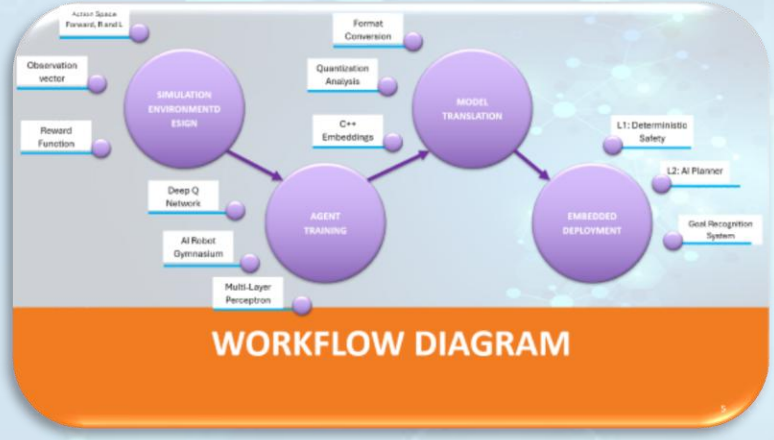
$$Q(s,a) = r + \gamma * \max Q(s',a')$$

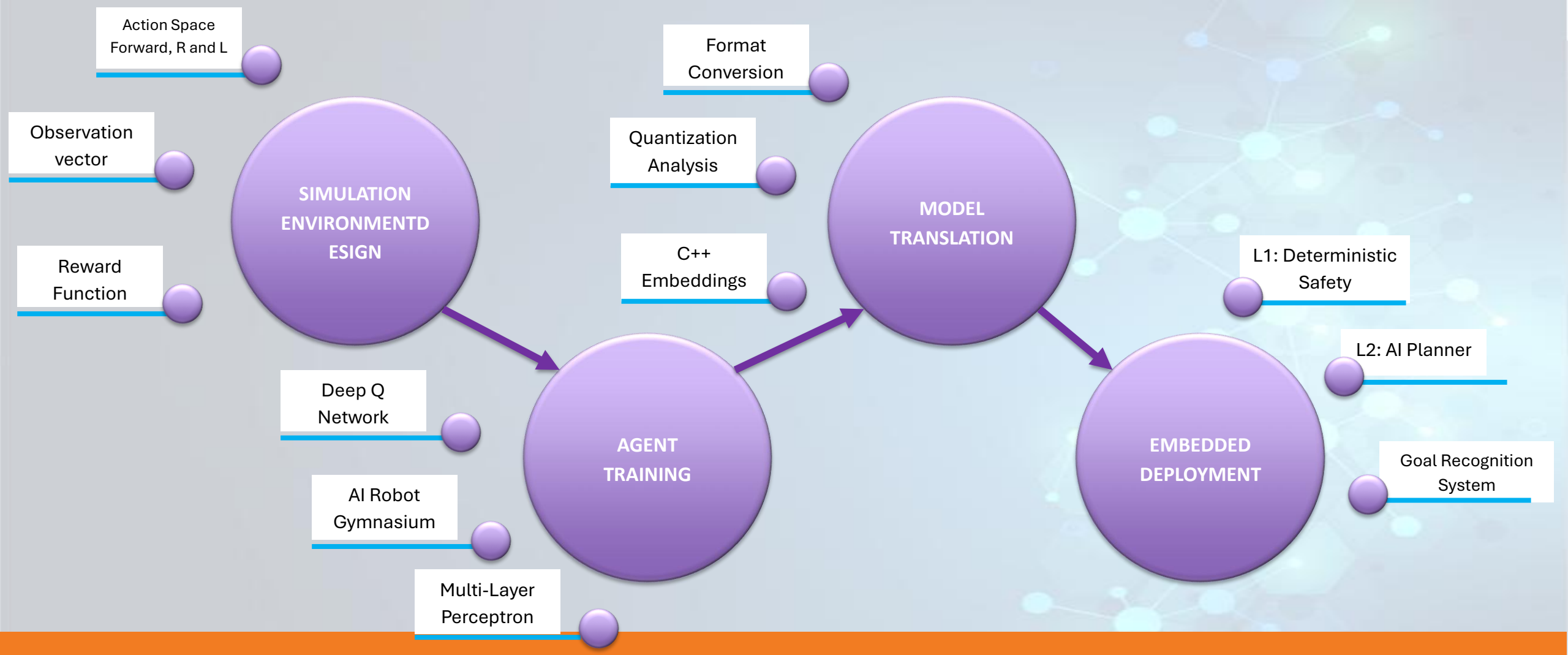
- Loss Function (Temporal Difference):

$$L = E[\,(Target - Prediction)^2\,]$$

Methodology: Sim-to-Real Pipeline

- System Architecture (See Block Diagram):
- 1. Simulation (Digital Twin):
  - OpenAI Gymnasium + Ray Casting.
- 2. Training:
  - Stable-Baselines3 (DQN Agent).
- 3. Model Translation Pipeline:
  - PyTorch -> ONNX -> TensorFlow Lite -> C++ Byte Array.
- 4. Deployment:
  - ESP32 Inference Engine (32ms per loop).





WORKFLOW DIAGRAM

Challenges & Engineering Solutions

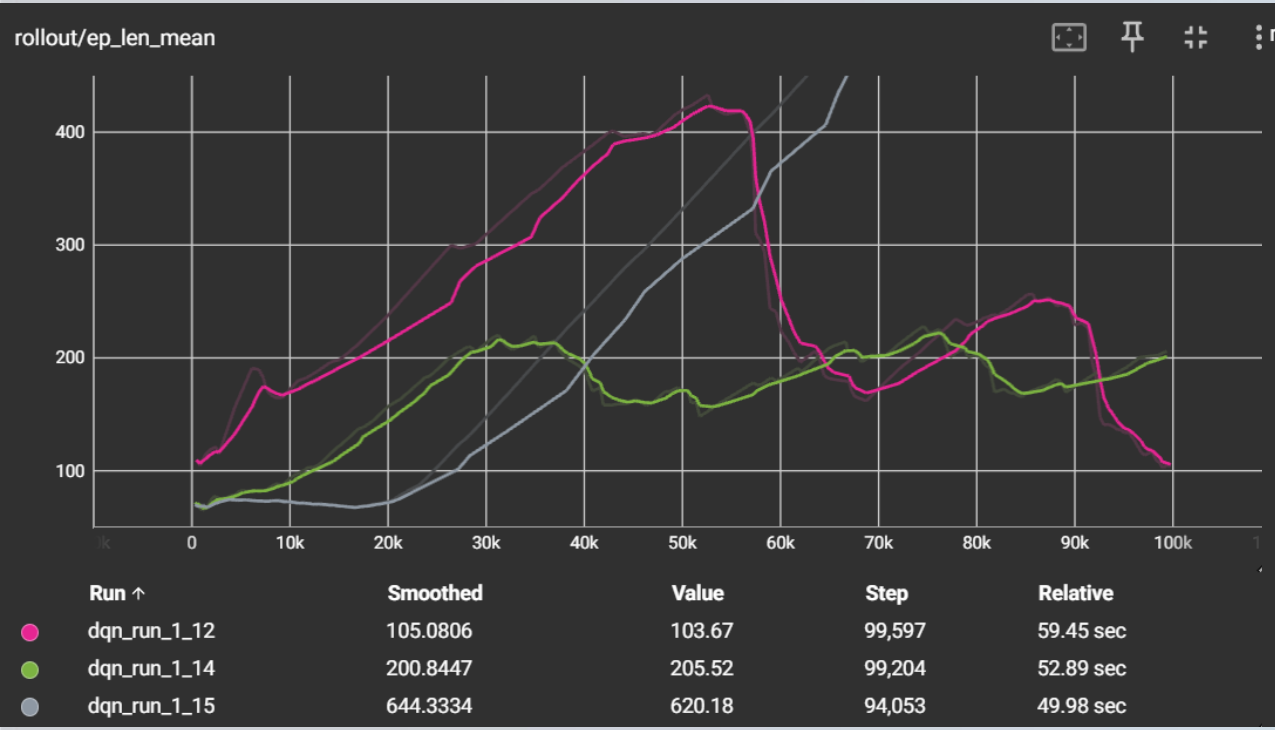
- Challenge 1: Quantization Collapse
  - Problem: Int8 quantization (15KB) lost precision.
  - Symptom: Robot spun in circles (output constant 'Turn Right').
  - Solution: Enforced Float32 precision (60KB).
- Challenge 2: The Mirror Brain
  - Problem: Sim wiring [Right, Front, Left] != Real wiring.
  - Symptom: Robot turned INTO walls.
  - Solution: Software remapping of input vector.

The Solution: Hierarchical Control

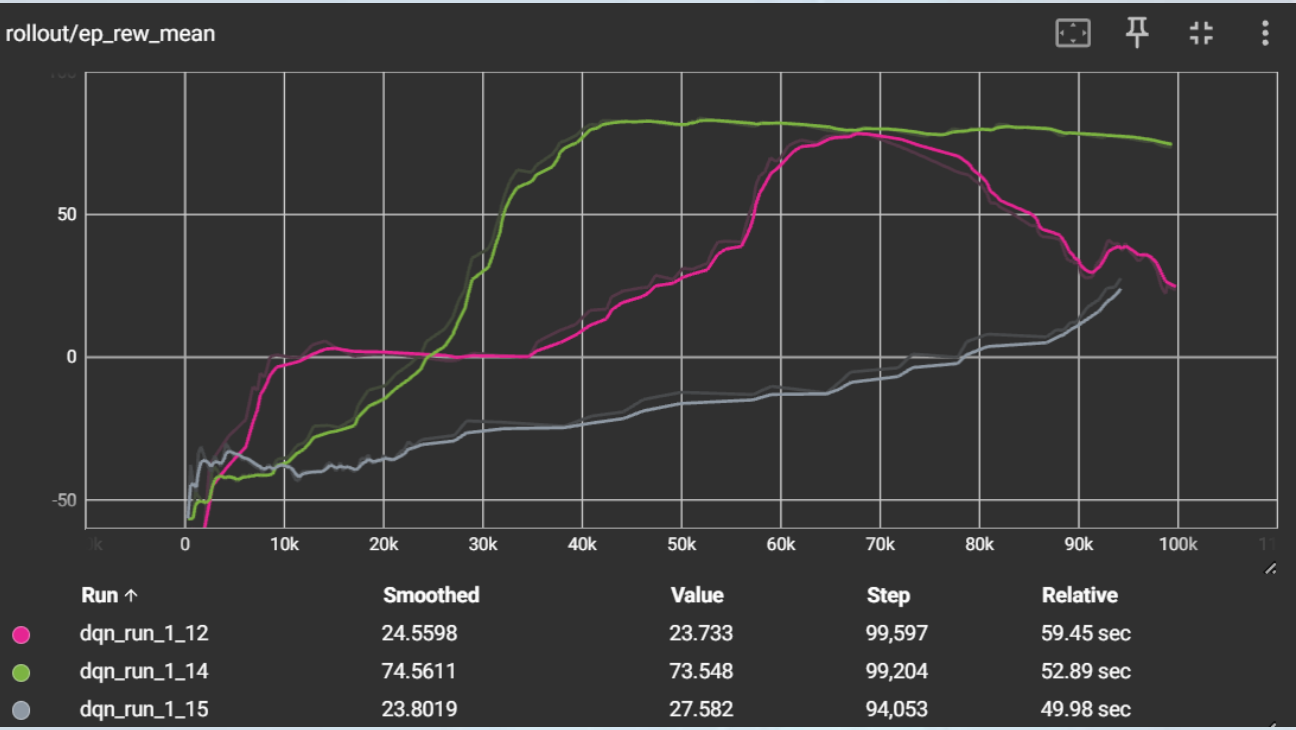
- Hierarchical Control Architecture (Hybrid AI):
- Layer 1: The Reflex (Safety)
  - If Distance < 15cm: Force Turn (Deterministic).
- Layer 2: The Planner (Intelligence)
  - If Safe: Query Neural Network (Probabilistic).
- Layer 3: The Mission (Success)
  - If IR Sensor == Black: STOP Motors.

Experimental Results

- Training Analysis (See Graphs):
- 1. The Failure (Pink):
  - Catastrophic Forgetting due to high Learning Rate.
- 2. The Coward (Lime):
  - Reward Hacking (spinning to avoid death).
- 3. The Success (Green):
  - Steady S-Curve Convergence.
  - Reward Plateau at +80.
  - Key: Extended Exploration (40k steps) + Lower LR.

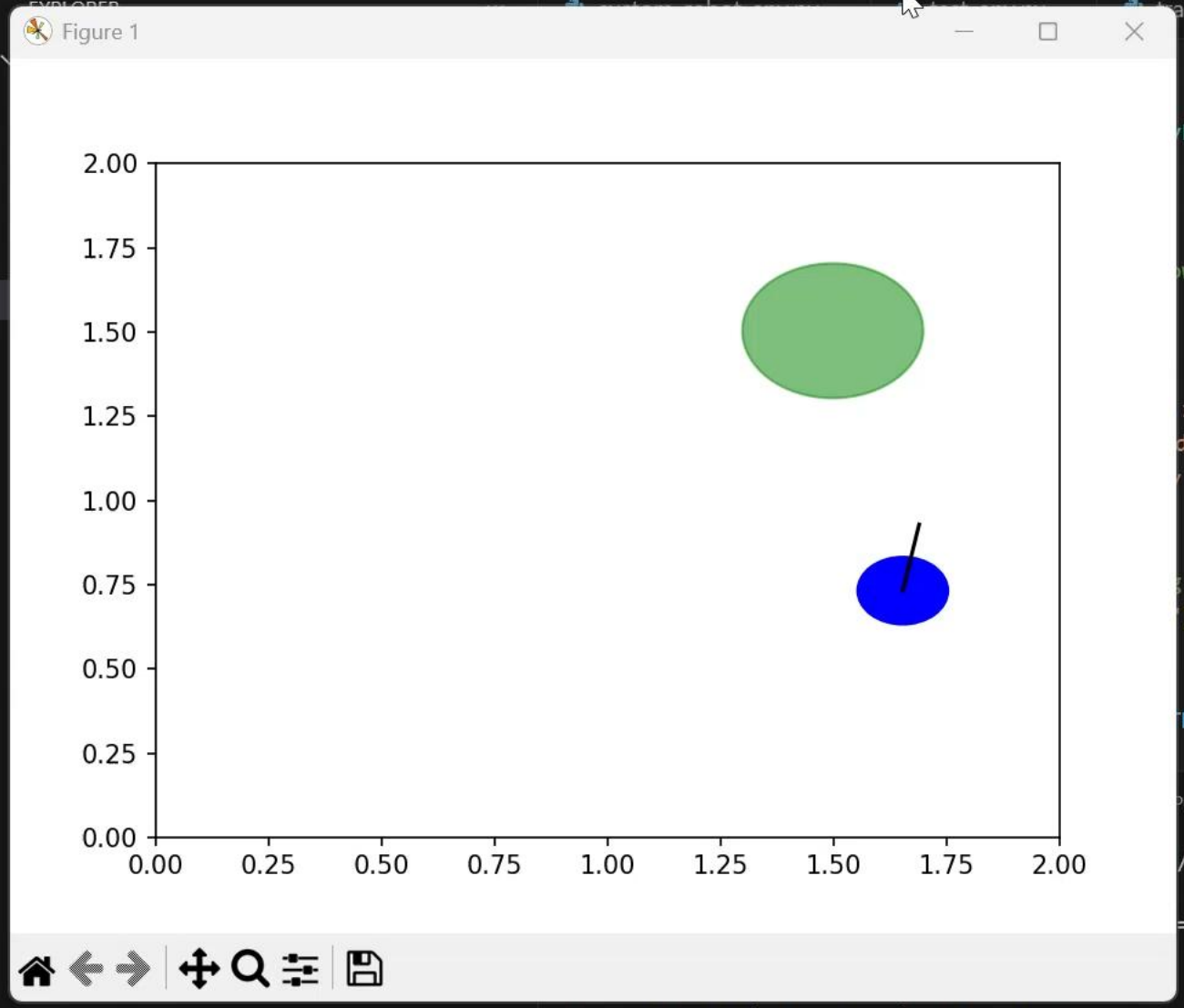


Run ↑	Smoothed	Value	Step	Relative
dqn_run_1_12	105.0806	103.67	99,597	59.45 sec
dqn_run_1_14	200.8447	205.52	99,204	52.89 sec
dqn_run_1_15	644.3334	620.18	94,053	49.98 sec



Run ↑	Smoothed	Value	Step	Relative
dqn_run_1_12	24.5598	23.733	99,597	59.45 sec
dqn_run_1_14	74.5611	73.548	99,204	52.89 sec
dqn_run_1_15	23.8019	27.582	94,053	49.98 sec

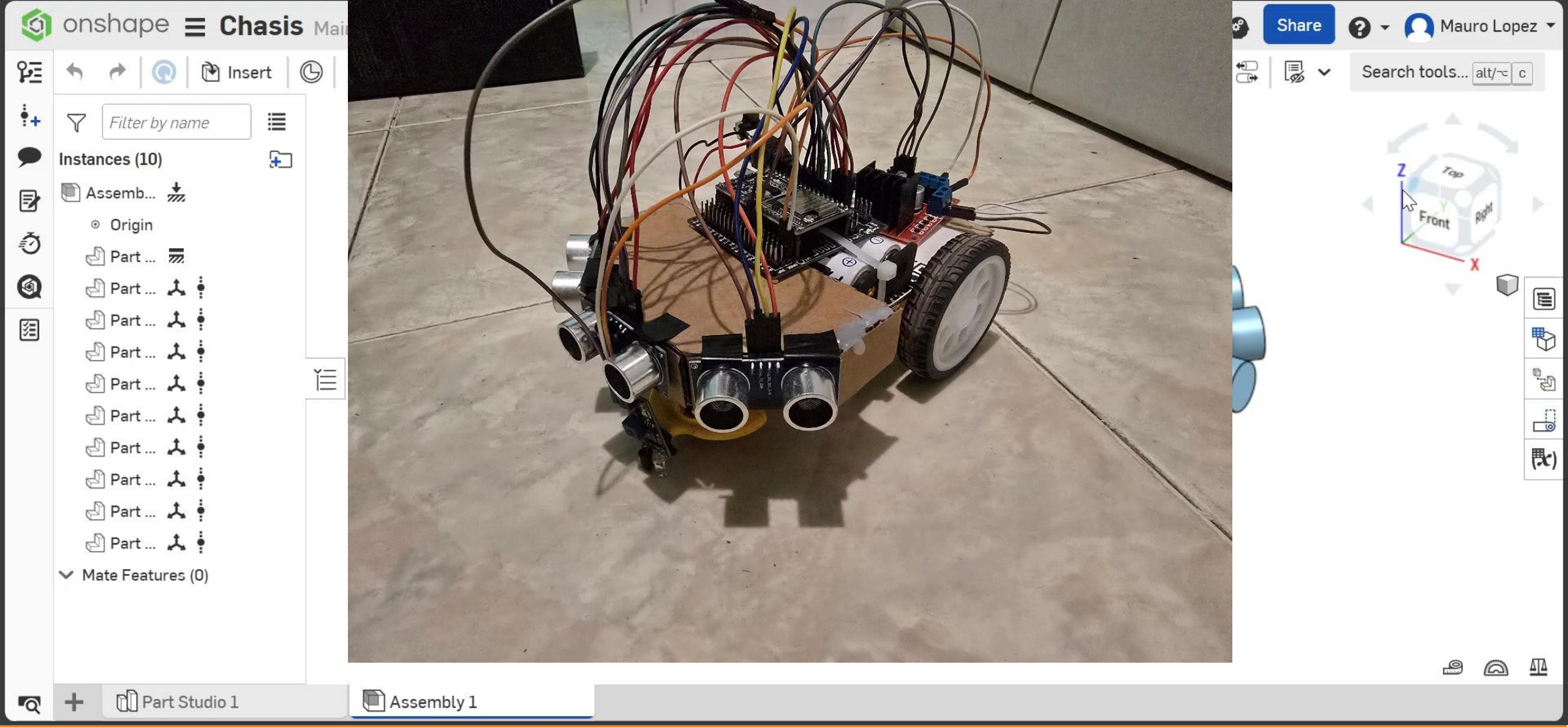
TENSORBOARD




```
File Edit Selection View Go Run ···  ai_final_project

enjoy_agent.py

Env

w that you are in a safe folder!)

zip"):
d at {MODEL_PATH}.zip")
first?")

ON so we can see it
)

TH}...")

PORTS  POSTMAN CONSOLE  Python

/util/port.cc:153] oneDNN custom operations are on. You may see slightly diff
round-off errors from different computation orders. To turn them off, set th
=0`.

Press Ctrl+C in the terminal to stop.
End of Episode (Goal or Crash)

OUTLINE
TIMELINE

Ln 42, Col 12  Spaces: 4  UTF-8  CRLF  { } Python  3.10.19 (ai_robot)

Buscar  ESP  10:43 p. m.  25/11/2025
```

Hardware conections

Component	Pin Name	ESP32 Pin
Left Sensor (-45°)	Trig Echo	GPIO 13 GPIO 12
Front Sensor (0°)	Trig Echo	GPIO 14 GPIO 27
Right Sensor (+45°)	Trig Echo	GPIO 26 GPIO 25
L298N Motor Driver	IN1 (Left Motor) IN2 (Left Motor) IN3 (Right Motor) IN4 (Right Motor) ENA / ENB	GPIO 33 GPIO 32 GPIO 18 GPIO 19 Jumpers on 5V

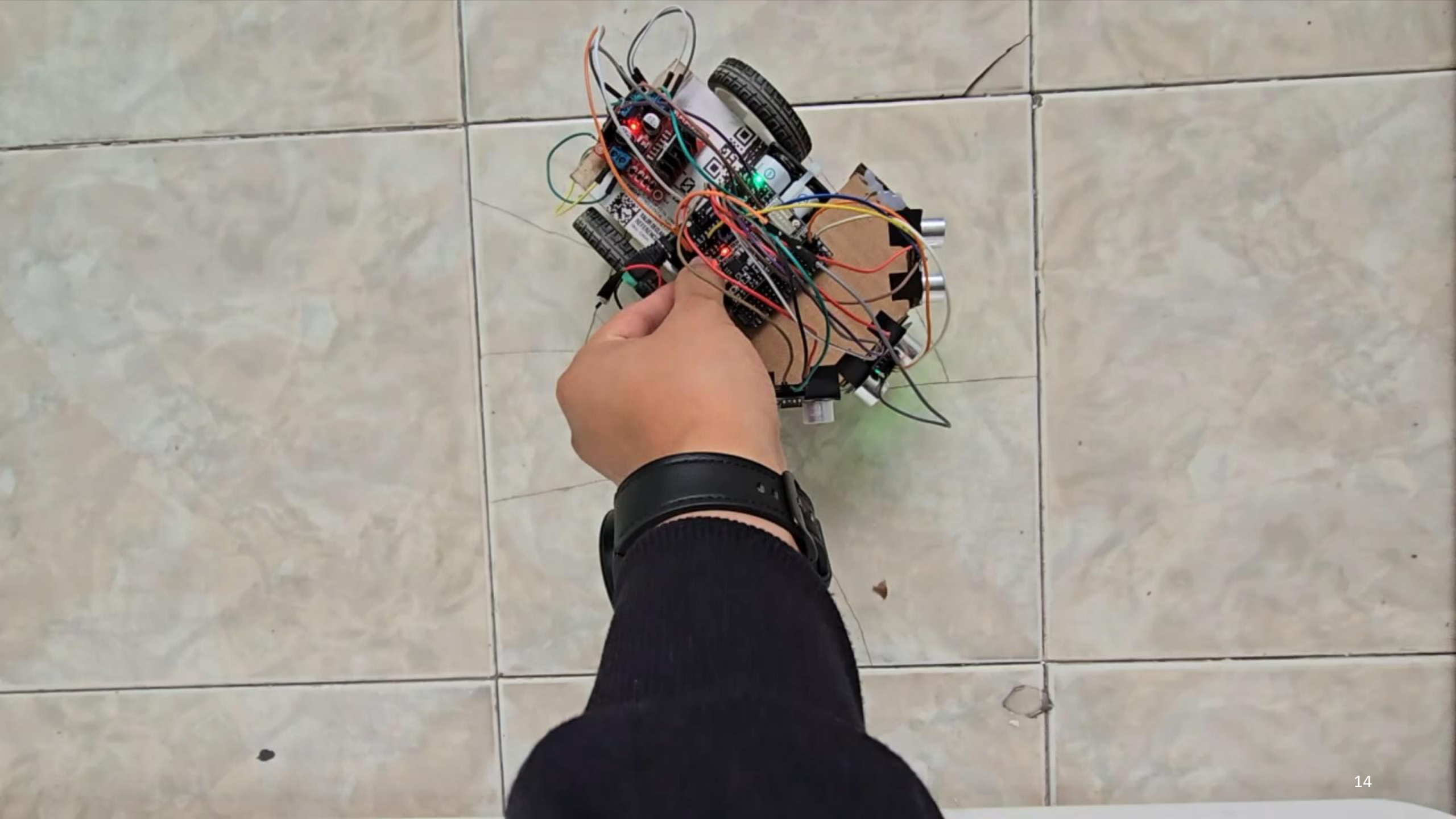


DESIGN AND CONSTRUCTION

Sim-to-Real Deployment Demo

- Real World Performance:
  - Success Rate: 80% (8/10 trials).
  - Inference Time: 32ms (Real-time capable).
  - RAM Usage: 11.5% (60KB used of 520KB).

Metric	Method A (Int8 Quantized)	Method B (Hybrid Float32)
Success Rate (n=10)	0%	80%
Avg. Inference Latency	~8 ms	~32 ms
Primary Failure Mode	Policy Collapse (Spinning)	Geometric Traps (Narrow Corners)
Model Size (Flash)	15 KB	60 KB
RAM Usage (Arena)	~4 KB	~60 KB



Conclusion & Future Work

- Conclusions:
  - Precision Matters: Int8 quantization destroys navigation policies.
  - Safety First: AI needs a deterministic safety wrapper.
  - Feasibility: ESP32 is capable of complex closed-loop control.
- Future Work:
  - Replace IR Goal Detector with ESP32-CAM (Computer Vision).
  - Train using PPO for continuous action space.