International Conference on Intelligent Robotics and Systems (IROS) ‘25

Low-Power Neuromorphic Learning for Micro-Robotic Controls in the Wild

Chae Young Lee, Sara Achour, Zerina Kapetanovic

Stanford University

Background: Micro-Robotics

- **Small form factor:** explore areas inaccessible to larger robots and humans (e.g. under canopy, earthquake aftermath)
- **Low-cost hardware:** valuable when at risk of losing robots (e.g. remote field survey, search & rescue) or need in a large quantity

Tamimi, Rubenstein's Group at Harvard, EarthSense

Can micro-robots intelligently navigate?

Typical brain of these robots:

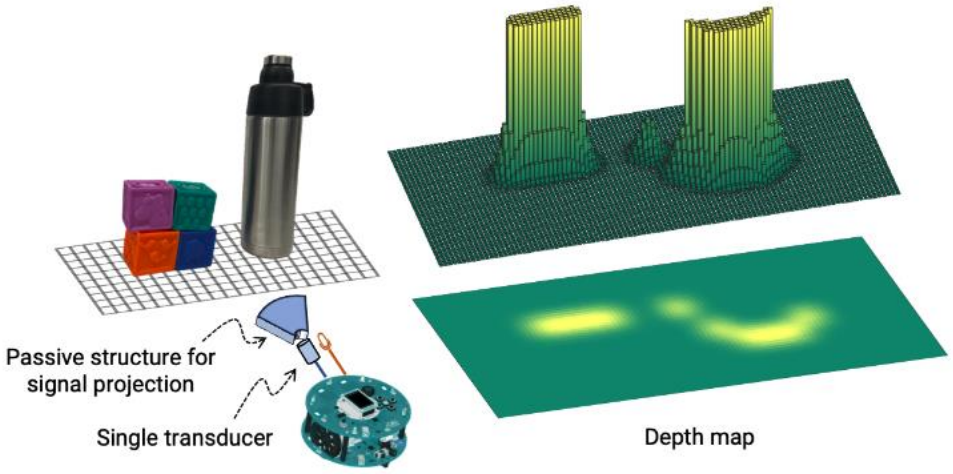
- ARM Cortex M-series core
- Flat memory: flash <1MB, RAM <500KB
- Clock <200MHz
- Power consumption <5mW

<

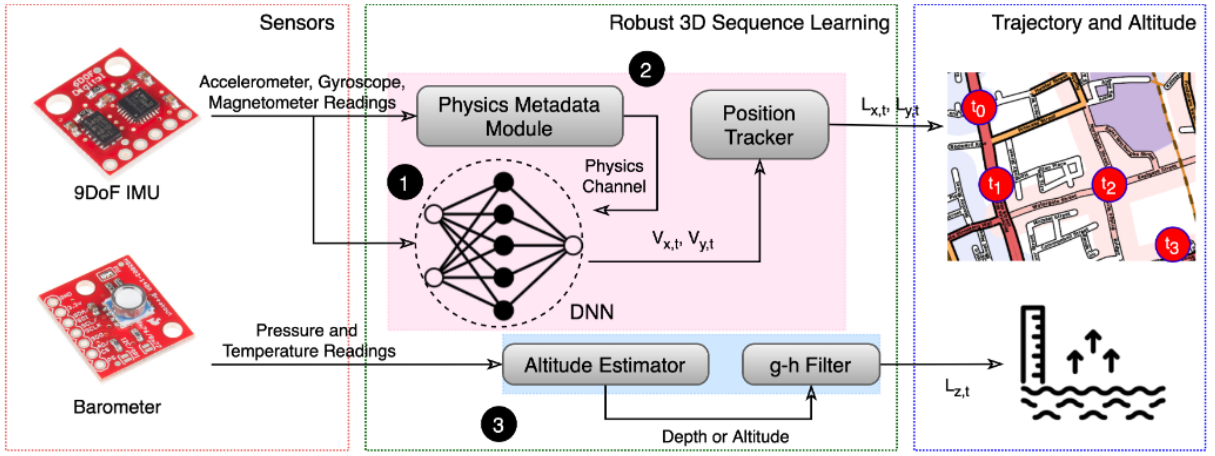
<

Today’s micro-robotic navigation

- Rely on cheap low-power sensors
- Run lightweight processing pipeline onboard



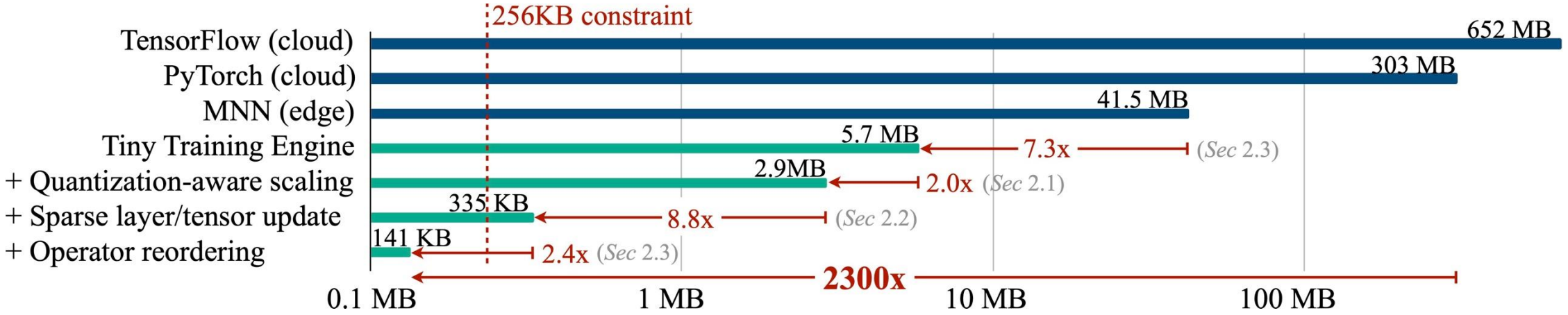
SpiDR (Bai et al., MobiSys ‘22)



TinyOdom (Saha et al., IMWUT ‘22)

Towards lightweight intelligence

- Deep neural networks (DNNs) are widely used in navigation (e.g. object avoidance, path planning, SLAM) but require **substantial hardware resources**.
- Adapting DNNs for resource-limited hardware is an active area of research.



Lin et al. "On-device training under 256kb memory." NeuRIPS, 2022.

Hyperdimensional Comptuing

- Brain-inspired computing paradigm that represents information using hypervectors.
- Known to be lightweight, error-resilient, hardware-friendly.

Model	MNIST Acc (%)	Flash (KB)	RAM (KB)	Latency (s)
HyperCam*	93.60	63.00	22.25	0.26
HyperCam**	90.36	52.62	22.25	0.08
xgBoost	76.86	365.55	77.09	0.01
MobileNet V3	98.69	1640.00	302.87	3.29
MCUNet V3	98.97	1340.00	502.91	46.71

Lee et al. ”HyperCam: Low-Power Onboard Computer Vision for IoT Cameras.” MobiCom, 2025.

HD-Based Q-Learning

Uses an HD encoder to map input state into hypervector and an HD-based regression to predict Q-value $Q(\mathbf{h}_t, \mathbf{a})$.

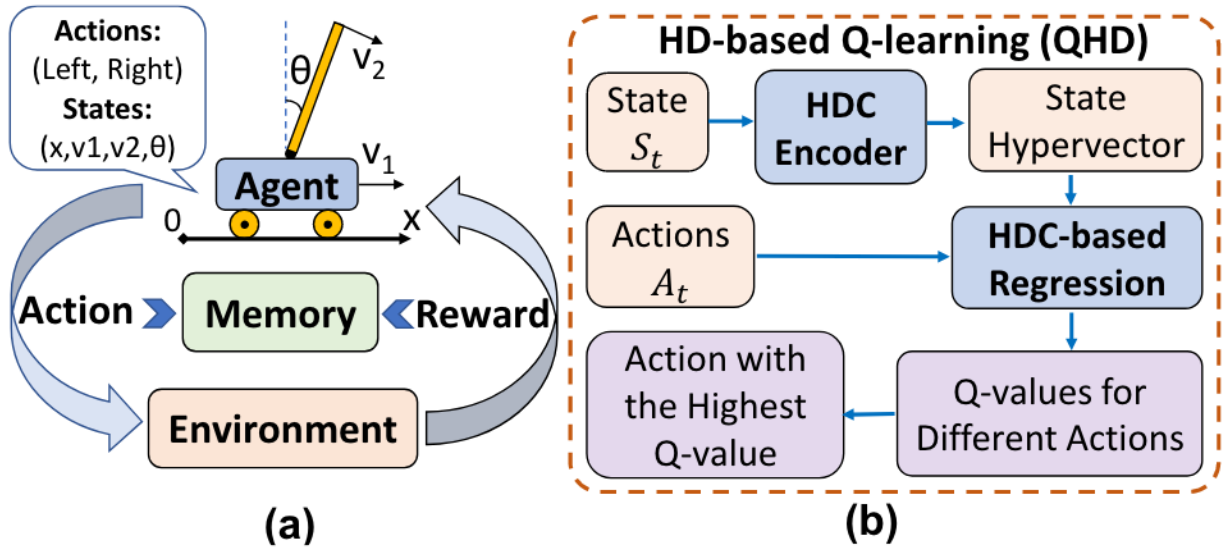


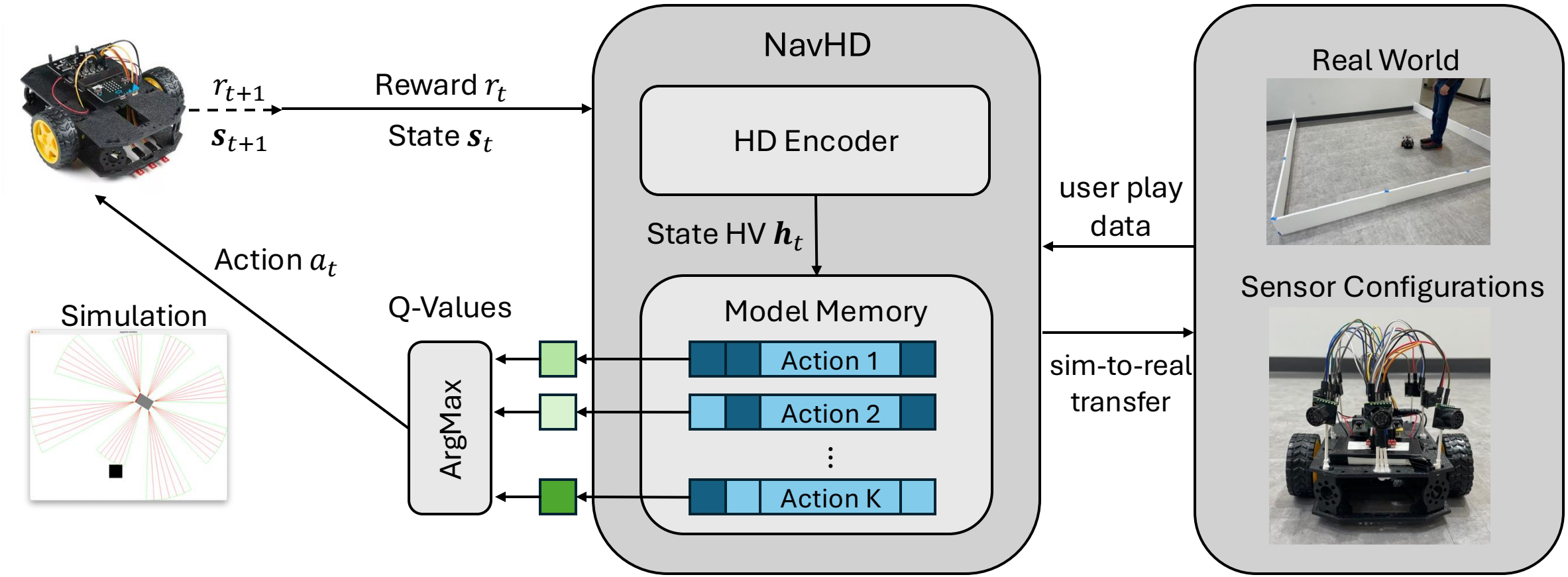
Figure 1: Overview of QHD reinforcement learning.

Ni et al. "Efficient Off-Policy Reinforcement Learning via Brain-Inspired Computing." GLSVLSI, 2023.

NavHD

- A differentiable HDC-based sense-actuate model designed for training on common navigation tasks.
- Outperforms DNN and previous HD methods in Q-learning and zero-shot real-world experiments.
- Achieves highest resource efficiency: 10.2 KB for model storage, 900 clock cycles per inference, 1.1 mJ of energy per frame.

NavHD


r_{t+1}
s_{t+1}
Reward r_t
State s_t
NavHD
HD Encoder
State HV h_t
Model Memory
Action 1
Action 2
Action K
Action a_t
Q-Values
ArgMax
Simulation
user play data
sim-to-real transfer
Real World
Sensor Configurations

Encoder

Given encoder $\phi: \mathbb{R}^d \rightarrow \mathbb{R}^n$,

input sensors $\mathbf{s}_t \in \mathbb{R}^d$ in d directions is encoded into $\mathbf{h}_t = \phi(\mathbf{s}_t)$.

Using a learned scaling matrix $\mathbf{B} \in \mathbb{R}^{n \times d}$ with $B_{ij} \sim N(0, \sigma^2)$ and a learned offset vector $\mathbf{b} \in \mathbb{R}^n$ with $b_i \sim \text{Uniform}(0, 2\pi)$, $\mathbf{h}_t$ is

$$\phi(\mathbf{s}_t) = \cos(\mathbf{B} \cdot \mathbf{s}_t + \mathbf{b}) \odot \sin(\mathbf{B} \cdot \mathbf{s}_t)$$

The resulting $\mathbf{h}_t$ is bounded within [-1, 1].

Action Prediction

Given model memory $\mathbf{C} = \langle \mathbf{c}_1, \mathbf{c_2}, \ldots, \mathbf{c}_k \rangle$ consisting of k action hypervectors,

$$\mathrm{sim}(\mathbf{h}_t, \mathbf{c}_k) = \mathbf{h}_t^T \cdot \mathbf{c}_k$$

Next action is determined by

$$\hat{y}_t = \mathrm{argmax}_k \mathrm{sim}(\mathbf{h}_t, \mathbf{c}_k)$$

NavHD Characteristics

- **Differentiable:** Including encoder and similarity function. Can be trained using backpropagation.
- **Exchangeable**: All parameters are i.i.d. random variables at initialization. Loss function is based on hypervector distances, which are symmetric. Gradient-based training preserves symmetry.

NavHD Q-Learning

Estimate Q-values using $Q(\mathbf{s}_t, a) = \mathrm{sim}(\phi(\mathbf{s}_t), \mathbf{c}_a)$. Using Bellman equation, the target Q-value is:

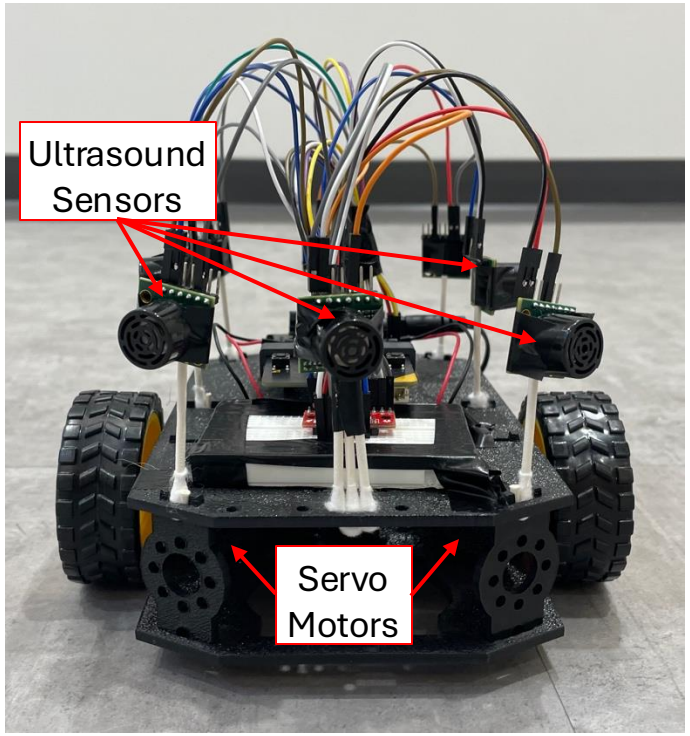
$$y_t = r_t + \gamma(1 - \mathrm{done}_t) \max_{a'} Q'(\mathbf{s}_{t+1}, a')$$

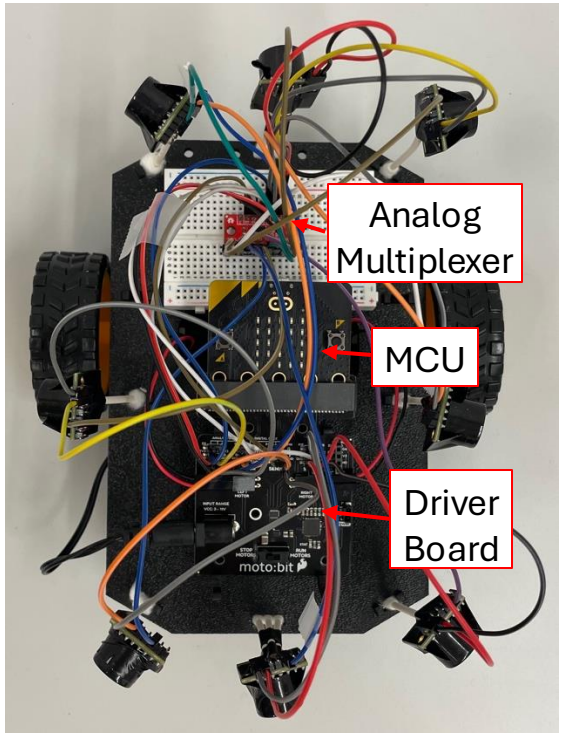
Then loss is computed as

$$l_t = \mathrm{HuberLoss}(Q(\mathbf{s}_t, a_t; \mathbf{C}, \mathbf{B}, \mathbf{b}), y_t)$$

NavHD uses Double DQN and experience replay to enhance training stability.

Hardware Implementation





Component	Idle (mW)	Full Throttle (mW)
Micro:bit v2	39.0	39.0
Carrier Board	43.9	48.4
Motors	0.0	2307.0
Ultrasound Sensors	40.0	40.0
Multiplexer	0.2	0.2
Total	**123.1**	**2434.6**

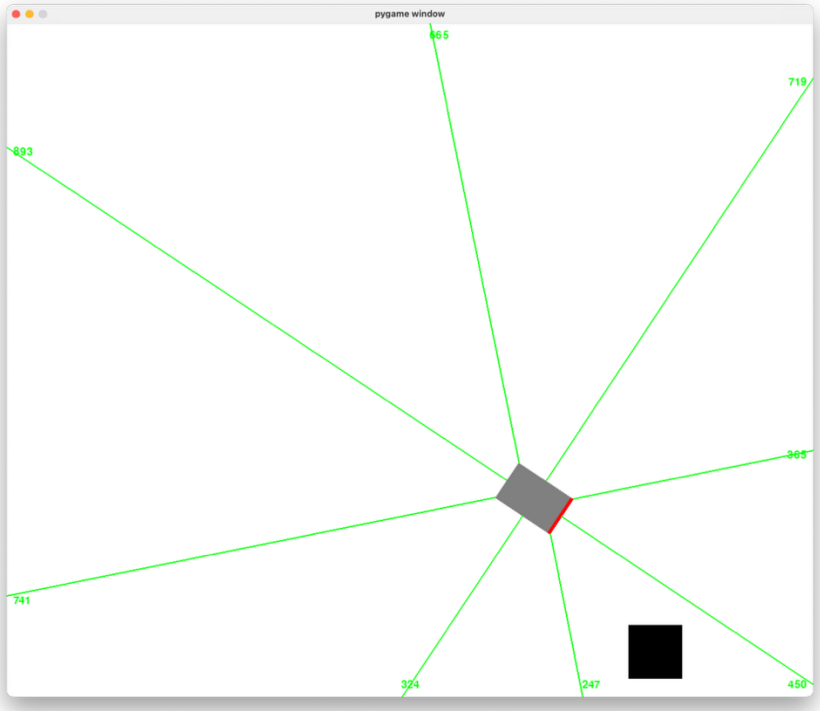
TABLE II: System power consumption.

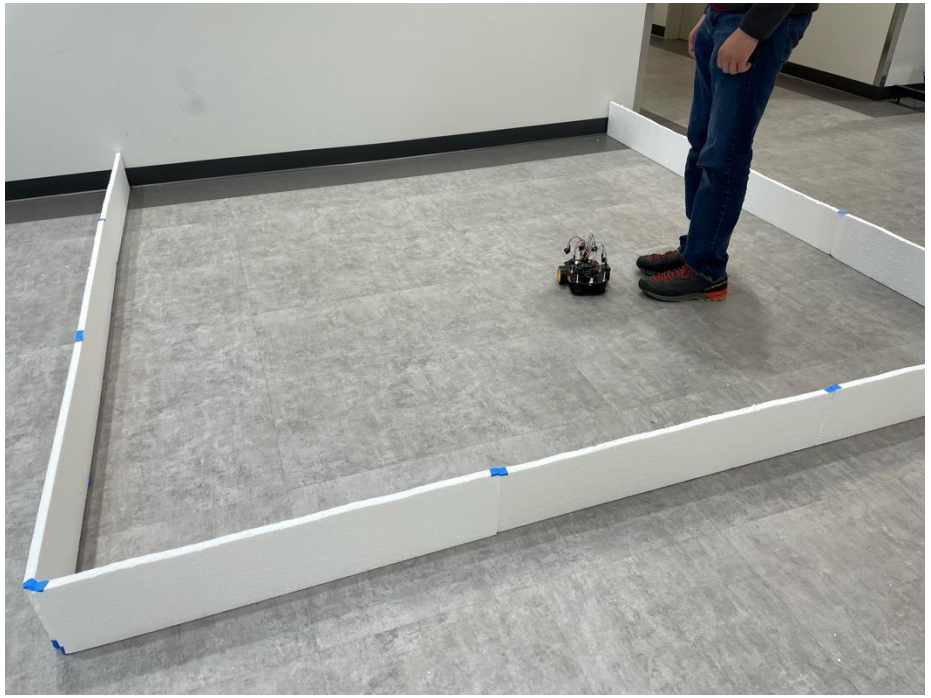
Evaluation

Object Avoidance Task

Input: observation (point cloud)

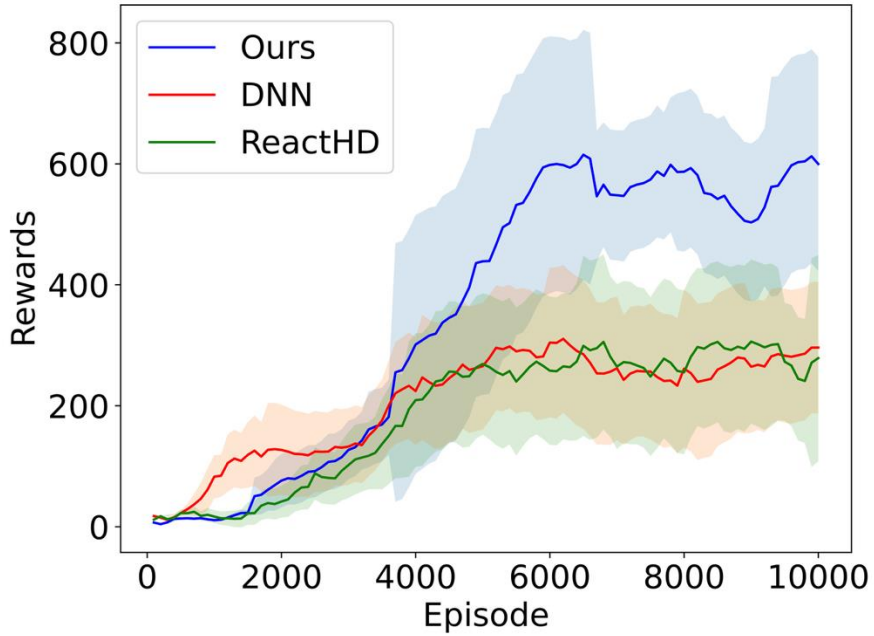
Output: action (4-class forward differential drive)



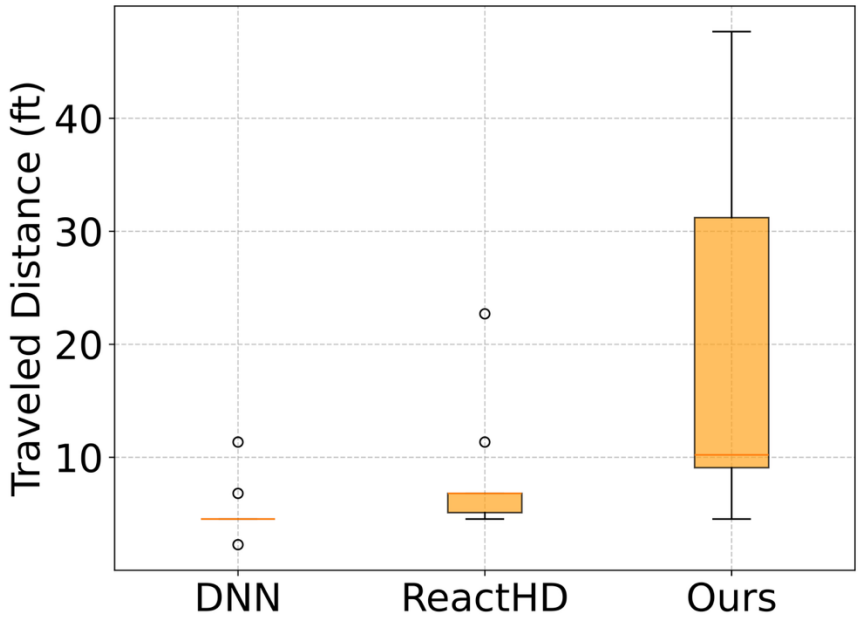


Learning Quality

- DNN: four-layer with hidden size 256, 128, 64.
- ReactHD: existing HD-based RL work. n=5,000.



(a) Hardware-optimized models in simulation



(b) Real world experiments

Resource Usage

TABLE IV: Resource usage on the target hardware.

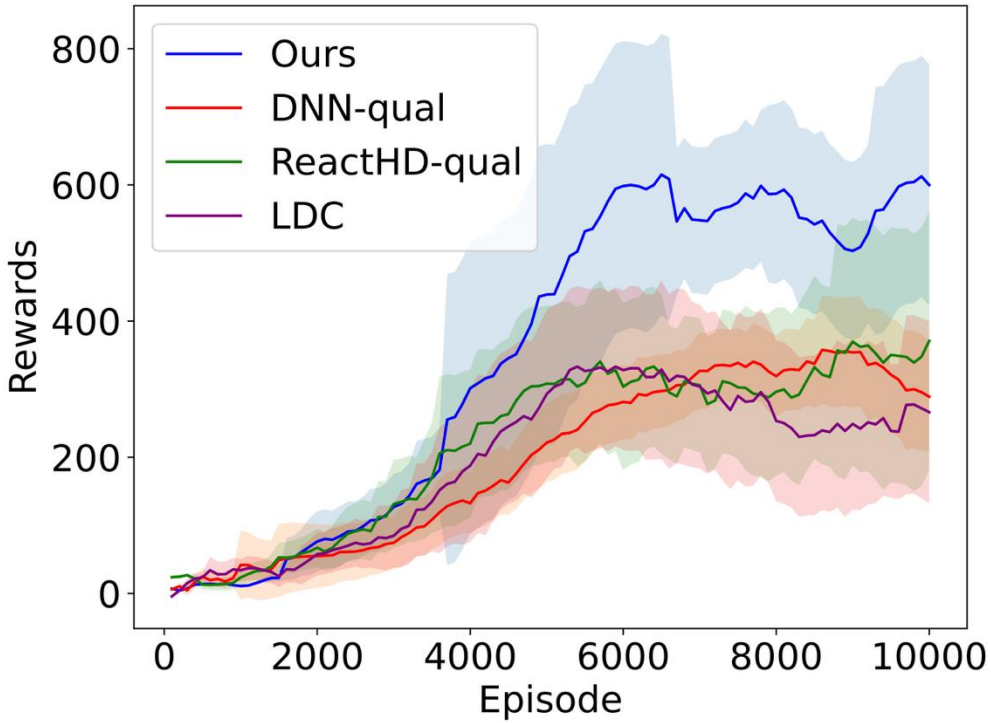
Model	Memory (kB)	Clock Cycle (k)	Energy (mJ)
DNN	263.8	2.9	3.4
ReactHD	19.5	8.4	9.9
NavHD	**10.2**	**0.9**	**1.1**

900 clock cyles in our MCU (64 MHz) takes ~14 us.

CPU-Only Experiments

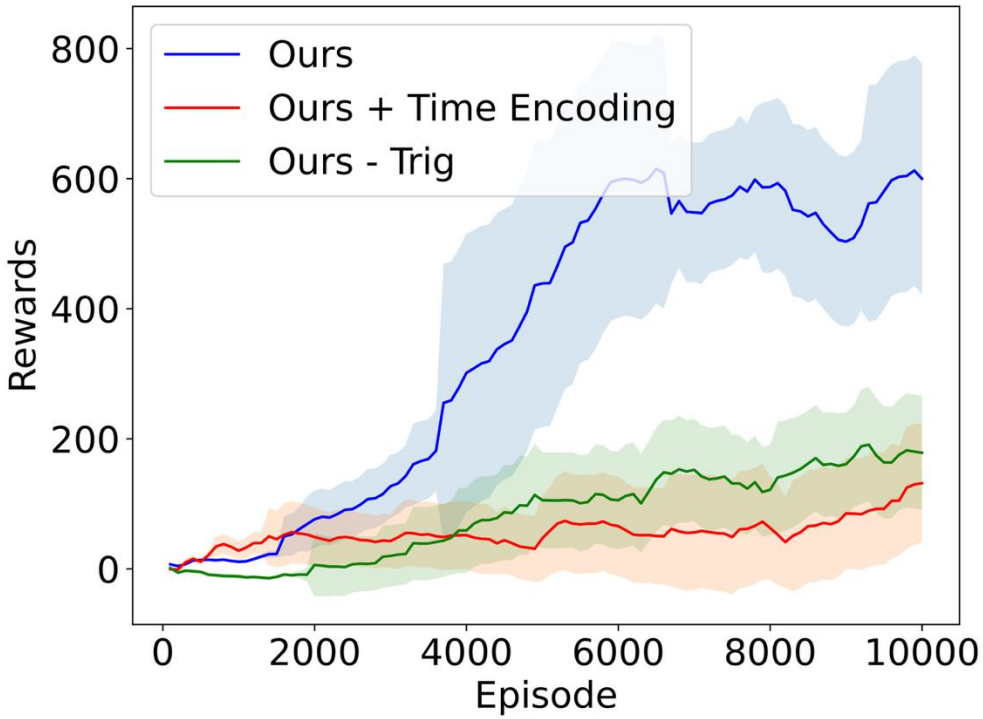
Models that don't fit in embedded hardware.

- **DNN-qual:** additional hidden layer of size 512.
- **ReactHD-qual:** uses hypervector length 10,000.
- **LDC:** state-of-the-art HD classifier + NavHD Q-learning



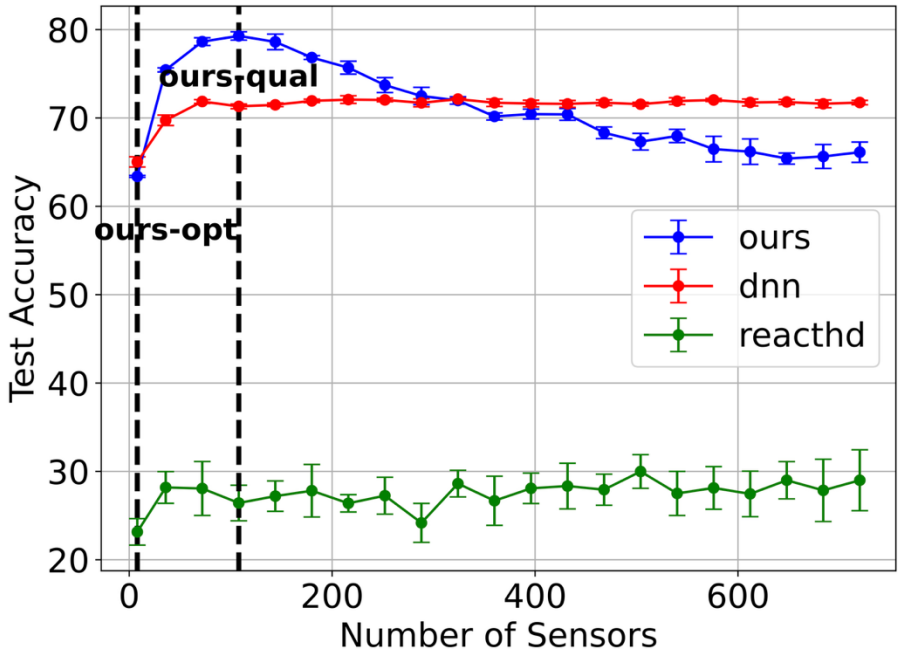
Encoder Ablation Study

- Explicit time encoding:
  $\phi(\mathbf{s_t}) = \mathbf{h}_t \oplus \rho^1(\mathbf{h}_{t-1}) \oplus \cdots \oplus \rho^L(\mathbf{h}_{t-L})$.
- Without activation function:
  $\phi(\mathbf{s_t}) = \mathbf{B} \cdot \mathbf{s}_t$

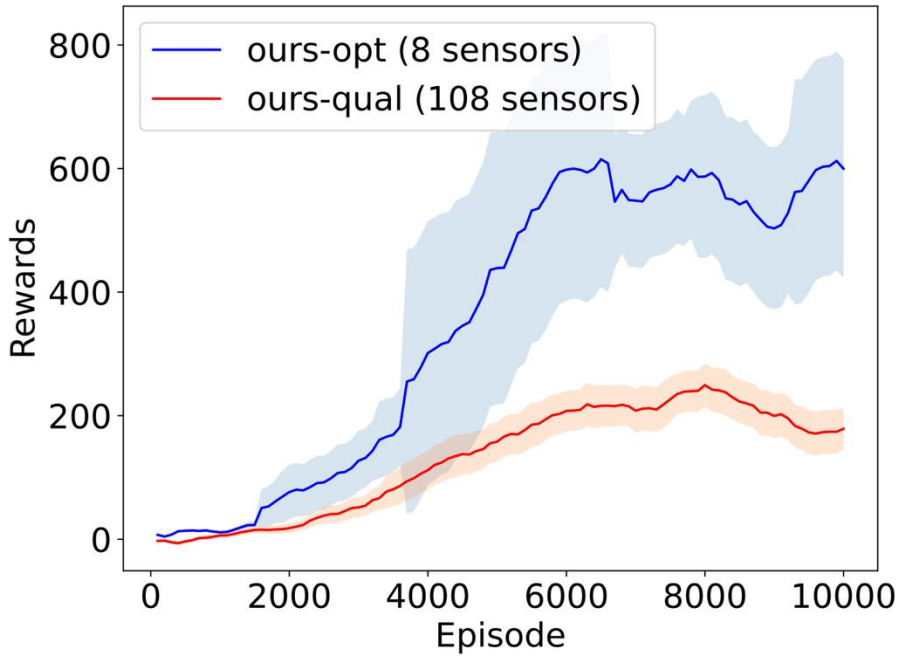


Sensor Count

Ours-qual (108 sensors), ours-opt (8 sensors).



(a) Imitation Learning



(b) Q-Learning

Conclusion & Takeaway

- NavHD outperforms both DNN and prior HD-based RL models in both accuracy and resource efficiency.
- NavHD uses ~10KB and can be deployed on extremely resource-limited hardware. If needed, model can be trained onboard.
- We found a sweet spot at 8 proximity sensors that gives us comparable accuracy to LiDAR while using ~100x less power.