Task Assignment in a Stochastic Environment Using Hierarchical Reinforcement Learning

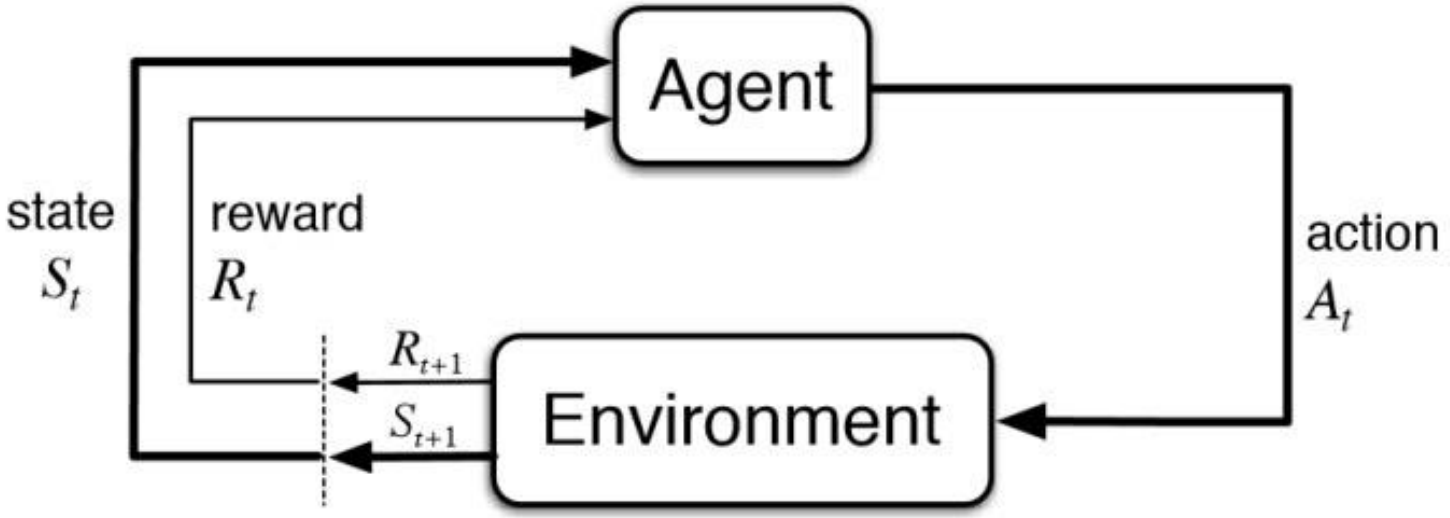
Intern: Colin Acker

Mentor: Scott Nivison

AIR FORCE RESEARCH LABORATORY
SCHOLARS PROGRAM

Reinforcement Learning

- Machine learning models
- Goal is to learn some policy $\pi(s)$, that maximizes return
  - Policy acts as controller
- Learn optimal policy $\pi^*(s)$
  - Random actions
  - Learn which actions give most reward

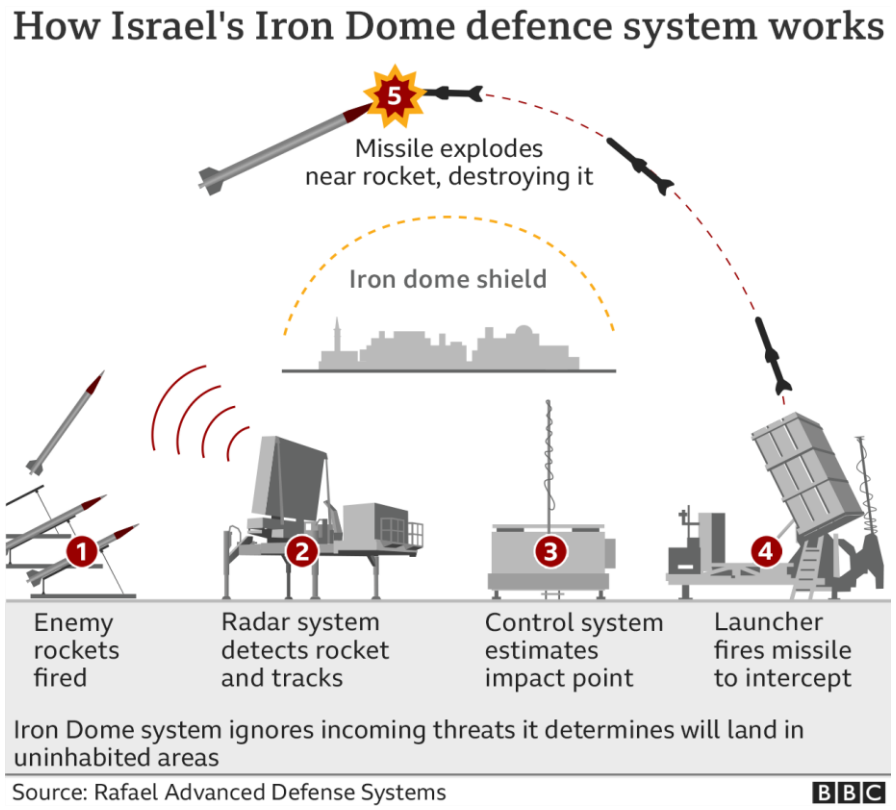


The Task Allocation Problem

- What is the Task Allocation Problem?
  - Great Interest
  - Lots of applications
  - Task planner is responsible for a list of tasks
  - Must choose UAVs based on environment
- What are some previous solutions?
  - Genetic algorithms
  - Particle swarm optimization algorithm
  - Deep Q Learning

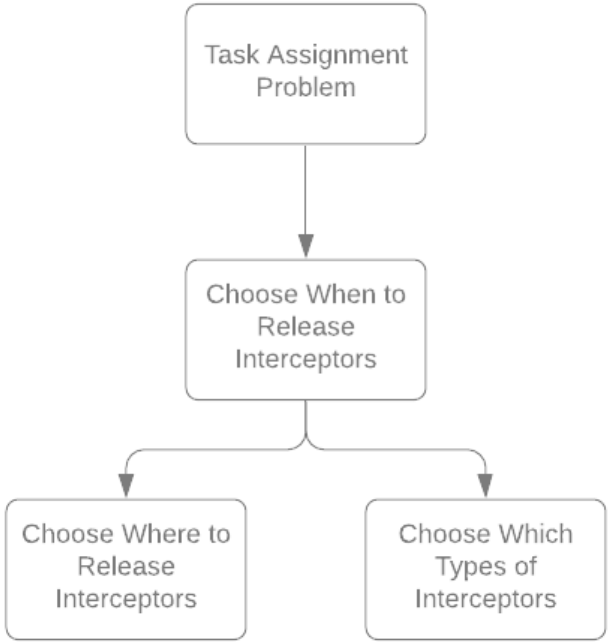
Israeli Iron Dome

- Turn the Task Allocation Problem into a missile defense problem
- Use reinforcement learning to solve step 2



Hierarchical Reinforcement Learning (HRL)

- HRL decomposes an RL problem into a hierarchy of subproblems
- Reduces computational complexity (smaller state space)
- Must choose most effective time to release interceptors given types chosen, and the location of each interceptor



Environment Formulation

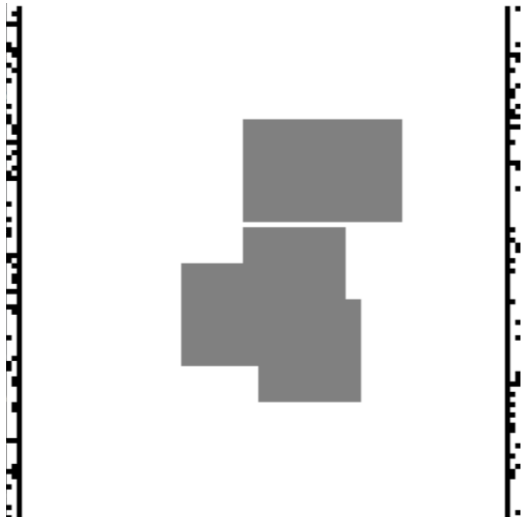
- Choose tasks

- 1-6 incoming missile trying to destroy our assets
  - A* pathfinding
- 3 different types of missiles and interceptors

- 100x100 grid with attrition zones

- Objective is to minimize number of assets left

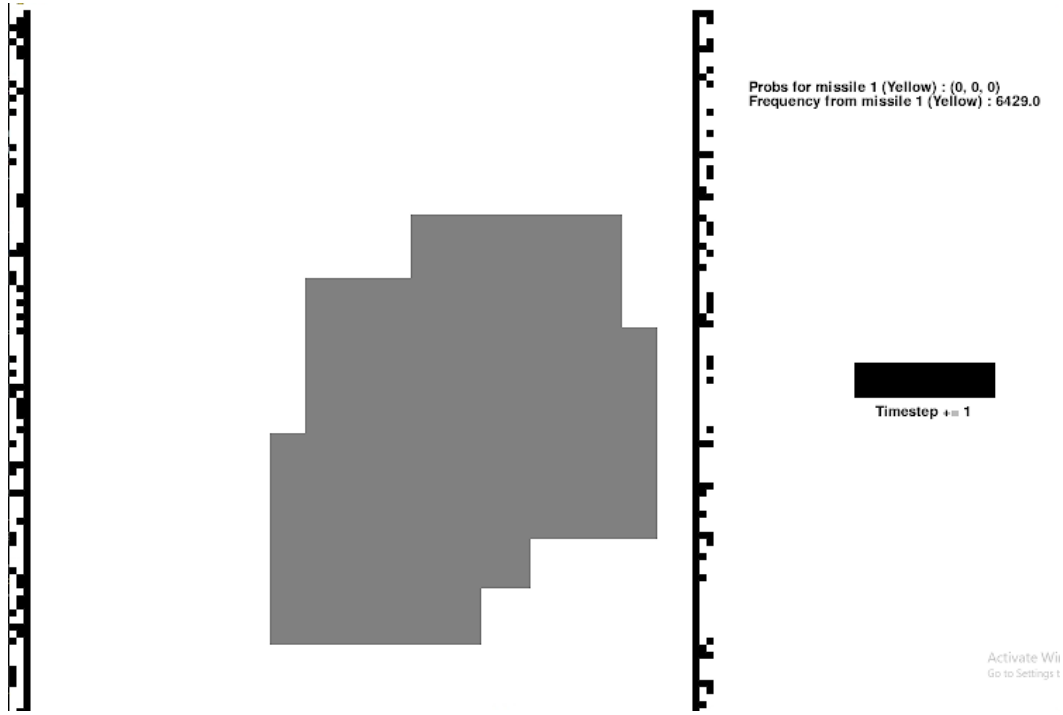
- How do we know what the type of each incoming missile is?
  - Radar frequency sampling



A* Pathfinding Algorithm

- Attrition zones

- Picks point with the lowest movement cost



Radar

- Missile attributes

- Frequency sampled from normal distribution
  - Standard deviation is defined as:

$$\sigma = \frac{50 * M_d + 500}{3}$$

  - Mean is defined as:

$$\mu = 2000 * M_t$$

- The further the missile, the more uncertain we are

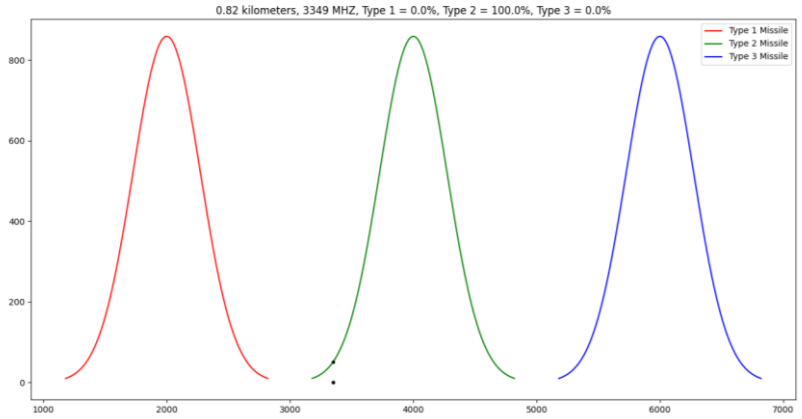
- With every timestep, radar updates probabilities accordingly
  - Probabilities are defined as such:

$$P(t) = \frac{f_t(x)}{\sum_{t=t_1}^{t_3} f_t(x)}$$

Radar Visualization

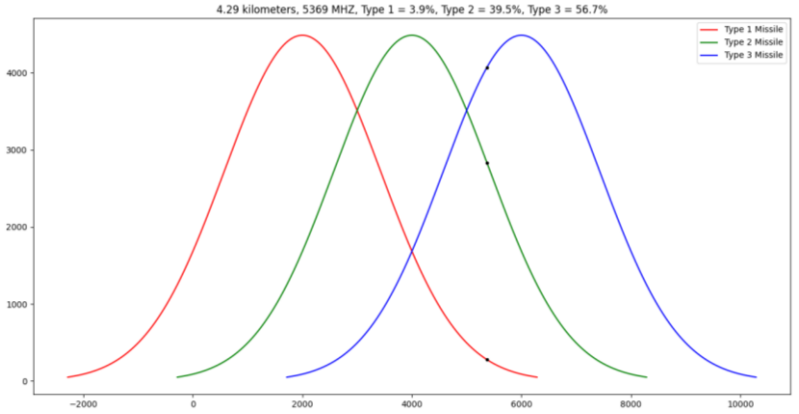
Close Distance

Sampled from type 2 interceptor



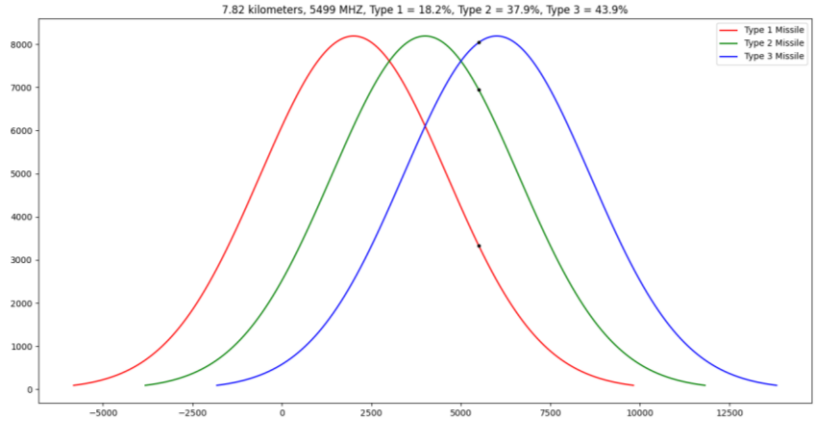
Medium Distance

Sampled from type 3 interceptor



Long Distance

Sampled from type 1 interceptor



Lever Neural Network

- LNN
  - NN with waiting node and lever node
- Output / Binary action space $A = \{0, 1\}$
- Input / State space $S = \{x_1, y_1, \cdots, x_6, y_{6,}\, t\}$
- Rewards
- Why
  - Large state and action spaces
  - Useful when trying to estimate timesteps
  - Less actions = easier to learn

Proximal Policy Optimization

- Most popular model-free RL algorithm
- Actor-critic method
- Output is a distribution over actions
  - Why LNN is better?

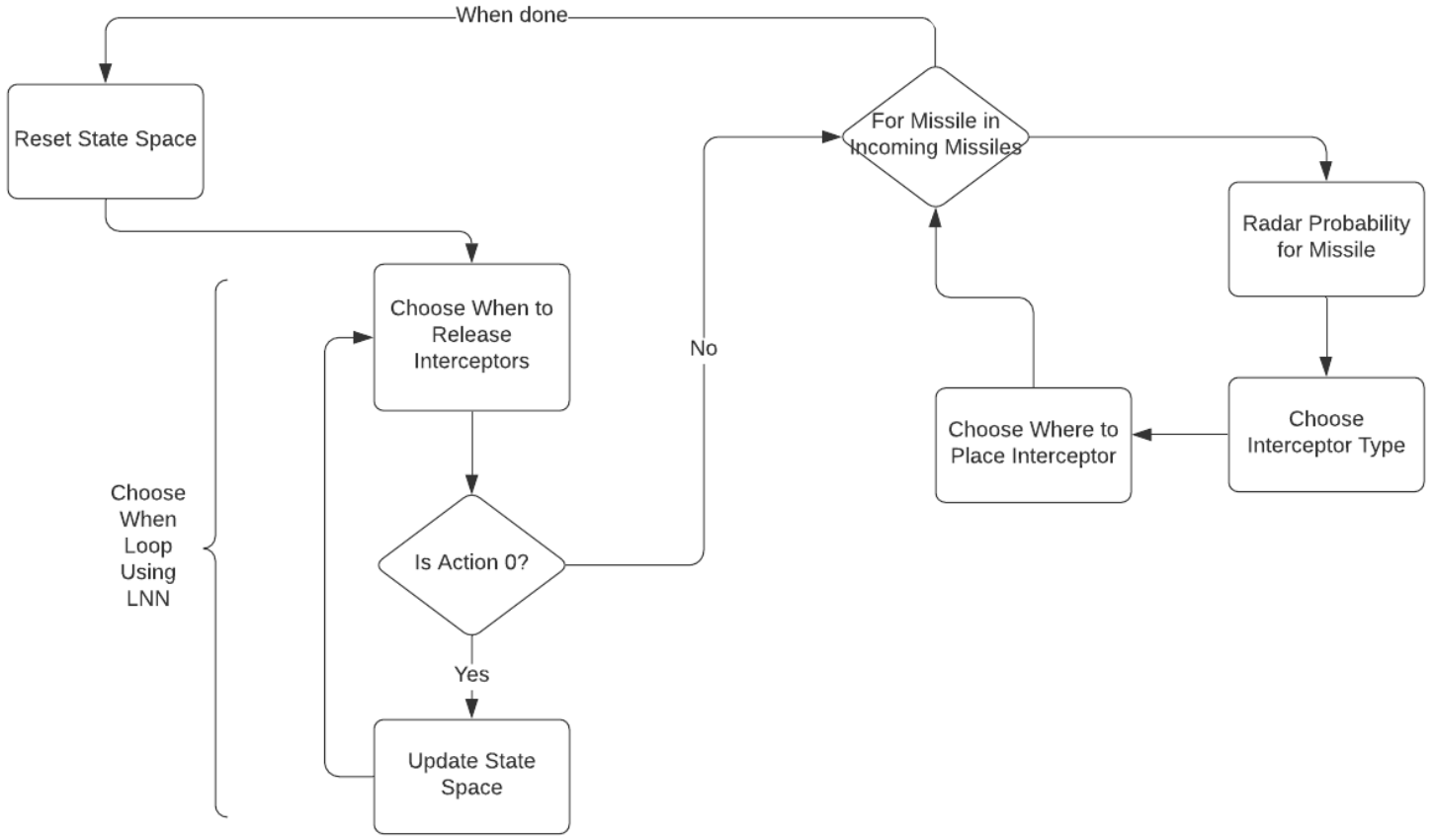
LNN vs NN

- LNN and NN both using PPO

- LNN output vs NN output

Comparison Table	LNN	NN
Average Reward Convergence	~ -220	~ -888
Average Timesteps Convergence	~ 62	~ 44

Flowchart


When done
Reset State Space
For Missile in Incoming Missiles
Radar Probability for Missile
Choose When to Release Interceptors
No
Choose Interceptor Type
Choose Where to Place Interceptor
Choose When Loop Using LNN
Is Action 0?
Yes
Update State Space

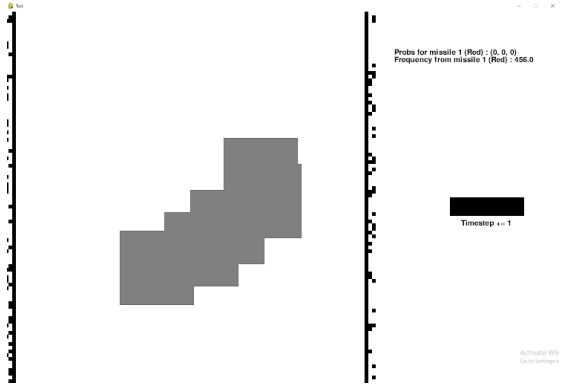
Choosing When to Release Interceptors

- Uncertainty

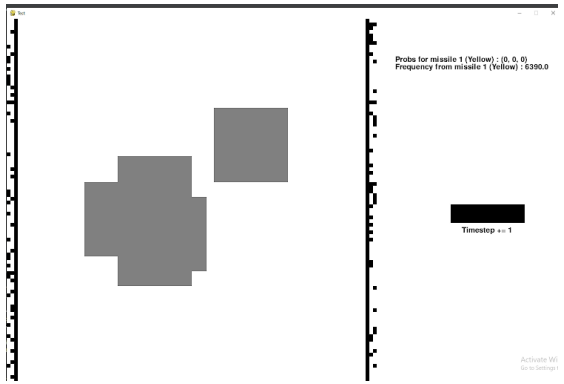
- Wrong interceptor

- Use Lever Neural Network (LNN) to choose when

- If we release interceptors too early



- If we release interceptors too late, we can pick the right type of interceptor, but the missile reaches it's target asset

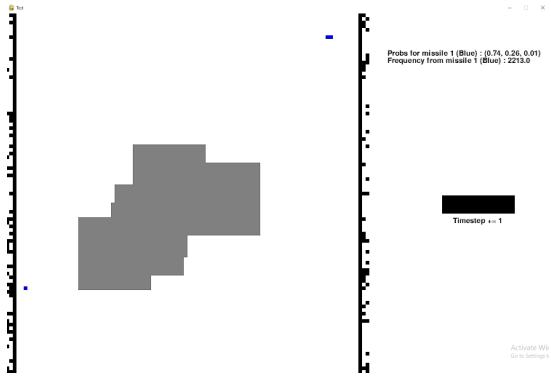


- Use LNN to choose optimal point of release

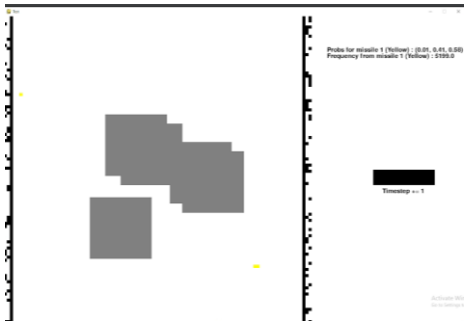
Choosing Where to Release Interceptors

- LNN assumes optimal interceptor placement
- Choose silo
  - We are assuming we know the missile target
- Output / action space $A = \{0,\ 1,\ 2\}$
  - Silos $\mathrm{L} = \{(5,\ 25), (5,\ 50), (5,\ 75)\}$
- Input / state space $S = \{x, y\}$
- Reward is defined as:

$$r = \sqrt{(L[a][0] - S[0])^2 + (L[a][1] - S[1])^2}$$



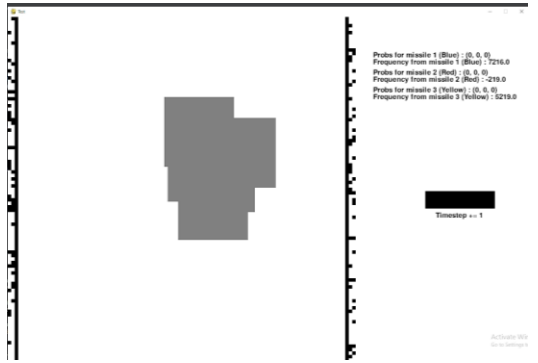
- Untrained network



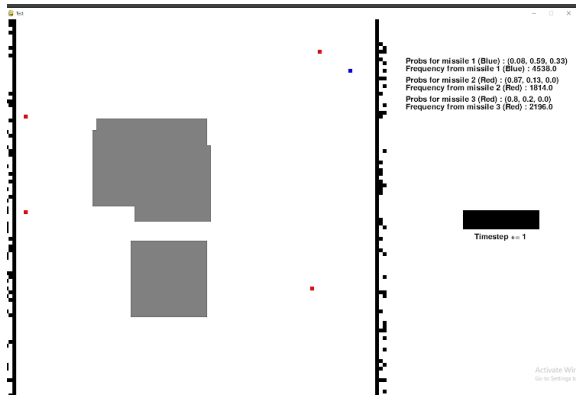
- Trained network

Choosing Which Types of Interceptors

- Choose types based on what the radar predicts
- Output / action space $A = \{0, 1, 2\}$
- Input / state space $S = \{P_m(1), P_m(2), P_m(3)\}$
- Rewards



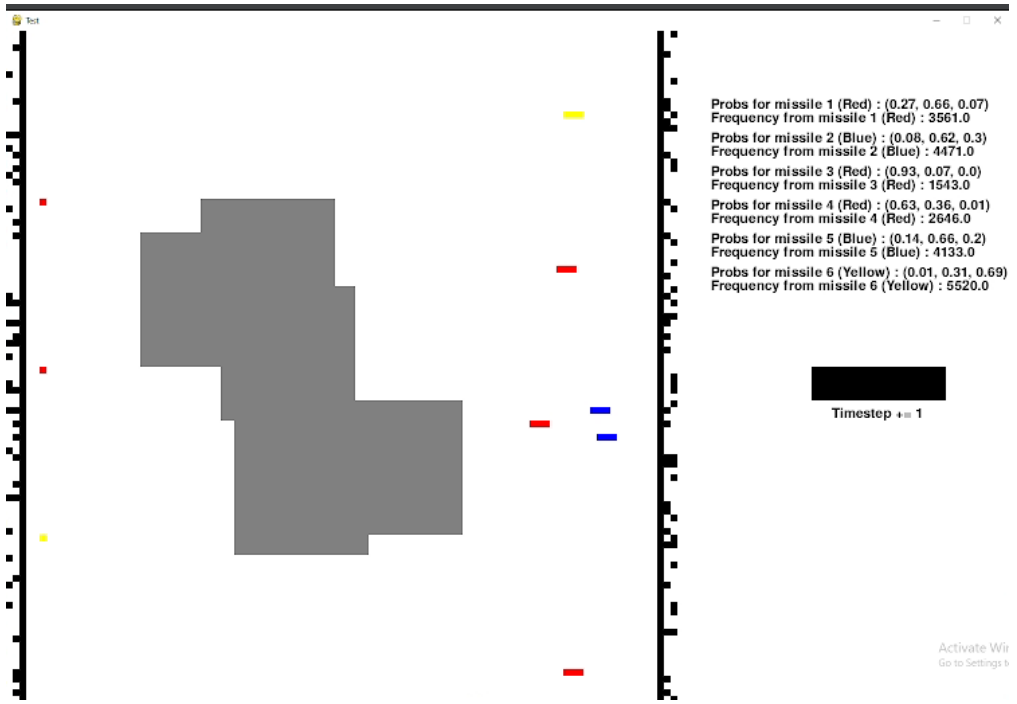
- Untrained network
- Used 6 interceptors for 3 missiles



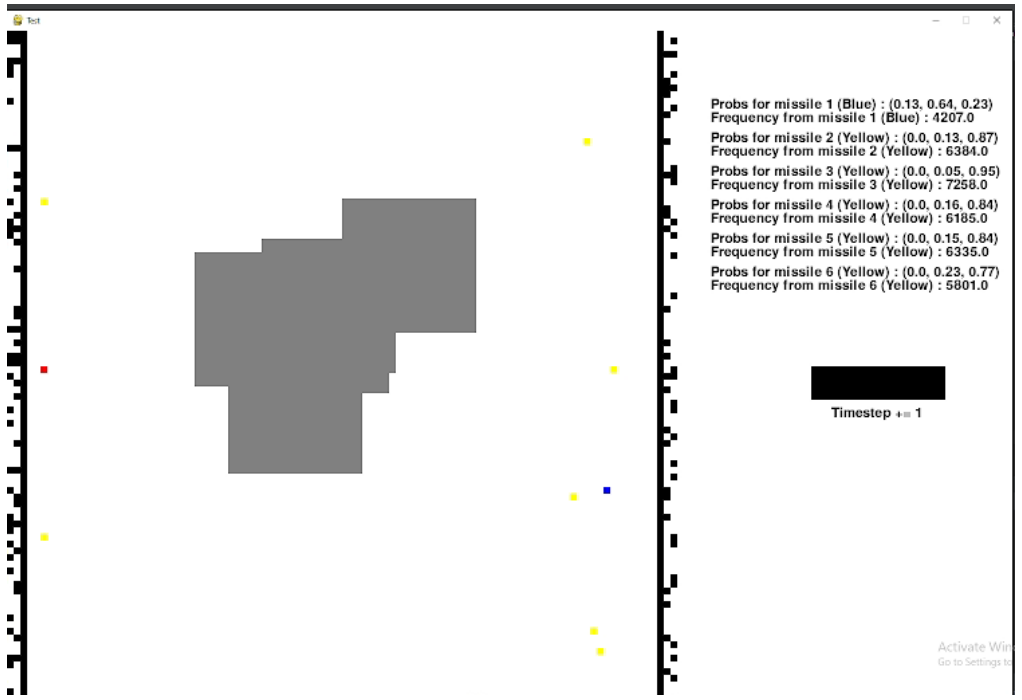
- Trained network

6 Missiles

Trained

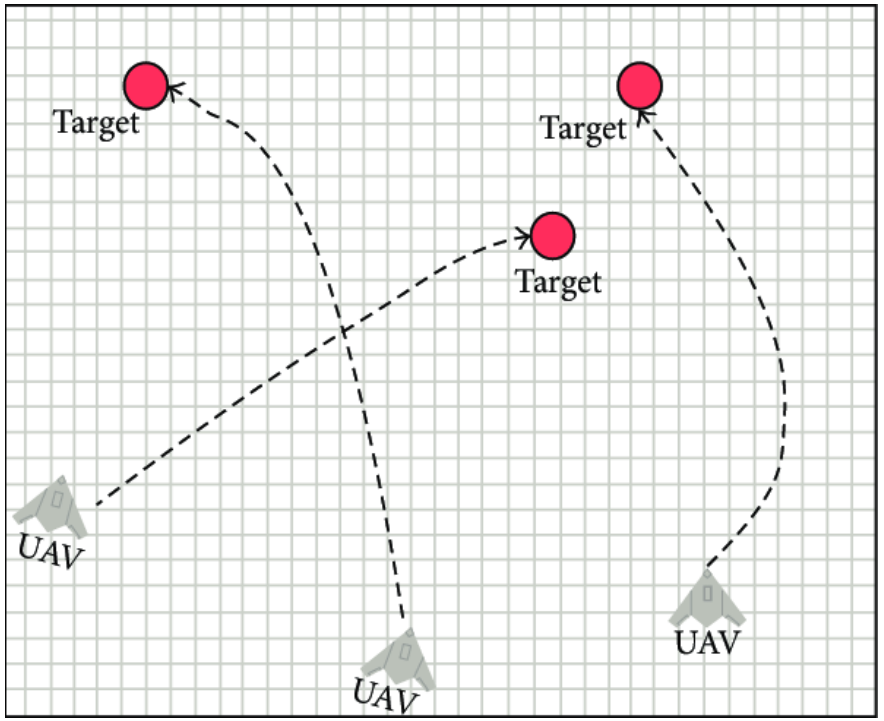


Untrained



In Summary

- Current task allocation algorithms are not optimal
- Simple to Complex
- Created novel LNN
- Solved TA with LNN and HRL



For the Future

- When choosing where, we assume we know missile's target
- Choosing types is too simple
- Game theory
- Turn environment into continuous

Questions