Progress Report

Multi-Robot Target Tracking Project

Arpit Aggarwal

Introduction

1. Robot-target tracking is the task of localizing the target by use of mobile robots

2. Task is to determine an optimal robot trajectory to minimize the uncertainty in target position

Method

1. Information given to us:

   initial robot position, initial estimated target position (mean, variance)

2. Create uncertainty map of the target position at time 't' using:

   Extended Kalman Filter or Bayesian Histograms

3. Then use the heatmap as the input to the greedy algorithm/RL that would give the next step for the robot for time 't+1'

Bayesian Histograms

Probability of grid cells at time t:

$$\mathbb{P}(v = q^* | p, \widehat{z}) = f(z_v | \widehat{z}, \sigma_s^2) = \frac{1}{\sqrt{2\pi\sigma_s^2}} exp(\frac{-1}{2\sigma_s^2}(z_v - \widehat{z})^2)$$

Heatmap calculated using probabilities from time 1 to t as:

$$\mathbb{P}(v = q^* | P_{1:T}, \widehat{z}_{1:T}) = \prod f(z(p_k, v) | \widehat{z}_k, \sigma_s^2)$$

Extended Kalman Filter(EKF)

EKF Equations:

Prediction:

$$\hat{o}^-(k) = \hat{o}(k-1),$$

$$\hat{\Sigma}^-(k) = \hat{\Sigma}(k-1) + R(k).$$

Update:

$$K(k) = \hat{\Sigma}^-(k)H^T(k)(H(k)\hat{\Sigma}^-(k)H^T(k) + Q(k))^{-1},$$

$$\hat{o}(k) = \hat{o}^-(k) + K(k)(z(k) - h(\hat{o}^-(k)))$$

$$\hat{\Sigma}(k) = (I - K(k)H(k))\hat{\Sigma}(k)^-$$

where $R(k)$ and $Q(k)$ are the covariance matrices of the noise from target's motion model and robot's measurement, respectively. $h(\hat{o}^-(k)) := \|p(k) - \hat{o}^-(k)\|_2$. $z(k)$ denotes the noisy distance measurement from the robot. $H(k)$ is the Jacobean of $h(\hat{o}^-(k))$.

Greedy Algorithm

1. After generating the uncertainty map at time ‘t’, next step is to use the greedy algorithm to move the robot for time ‘t+1’

2. Assumed a circular region around the robot with $r = 2$ and took 12 actions

Greedy Algorithm (contd.)

1. Determined optimal action for the robot using the determinant of the Fisher Information Matrix(FIM) which is shown as follows:

$$L = \frac{1}{2\sigma^2} * v_i^2 * v_j^2 * sin^2\alpha$$

where,

v(i) = the distance between robot position at time step ‘t’ and estimated target mean

v(j) = the distance between the future robot position and estimated target mean

alpha = angle between v(i) and v(j)

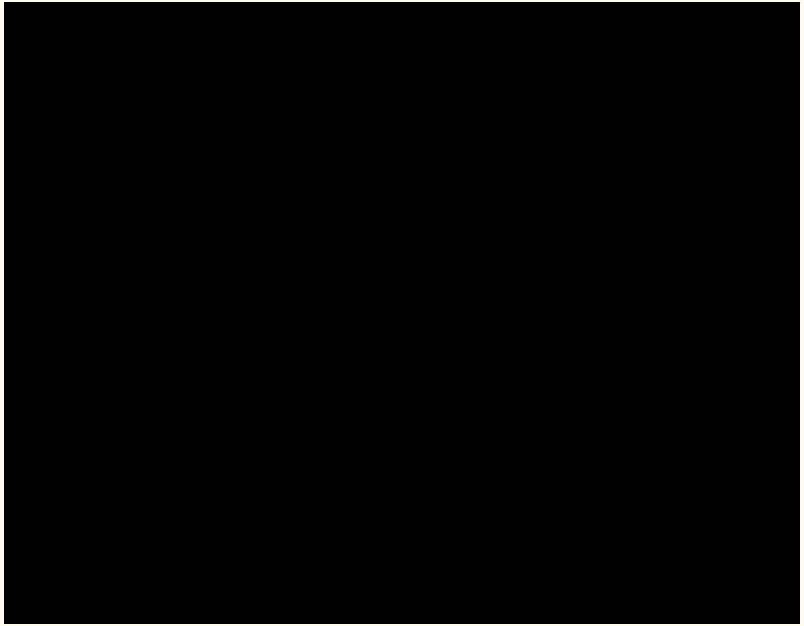
Analysis (robots=1, targets=1, greedy algo)

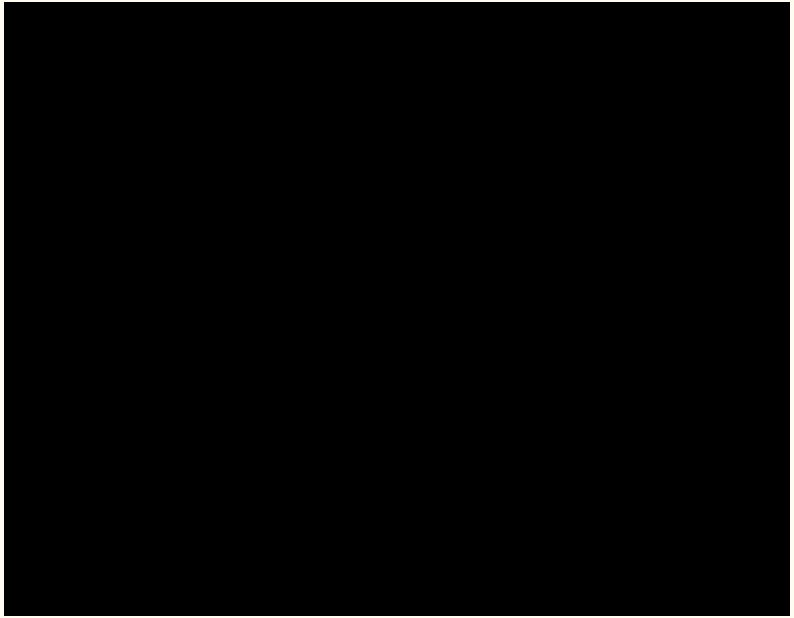
1. Considered an 2D environment with dimensions 20 x 20 (unit)
2. Robot uses range sensor to track target, equation given as:

$$\hat{d}_i = e + d_i, e \sim N(0, \sigma_s^2)$$

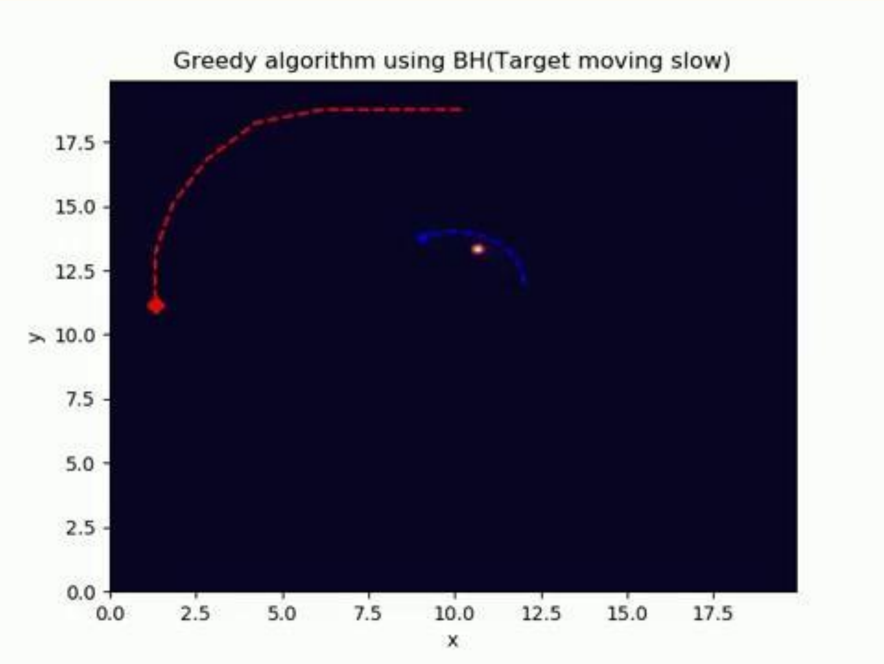
3. Assumed the target to be within range of the sensor
4. Compared EKF and Bayesian Histograms for target moving slow (omega=100 rad/s) and target moving fast (omega=33 rad/s) cases.

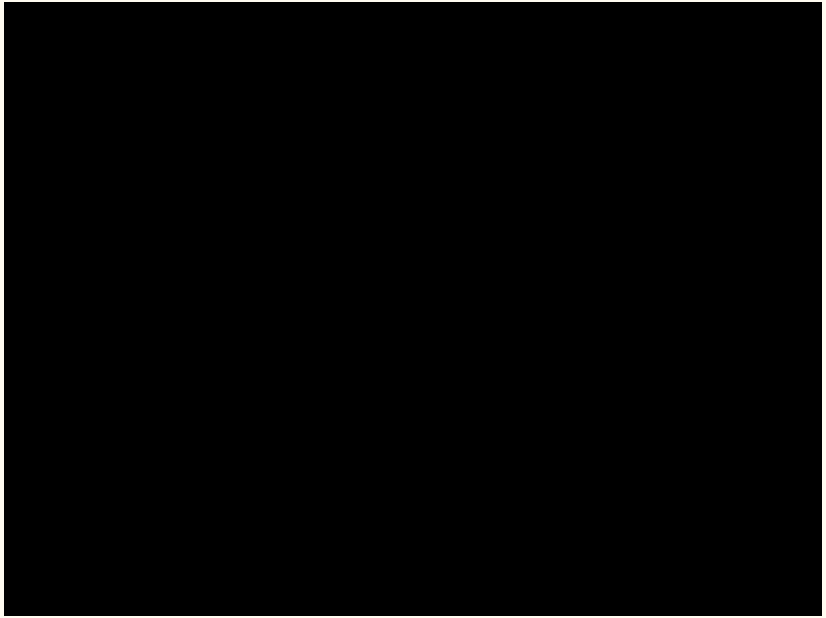
Bayesian Histograms with Stationary robot



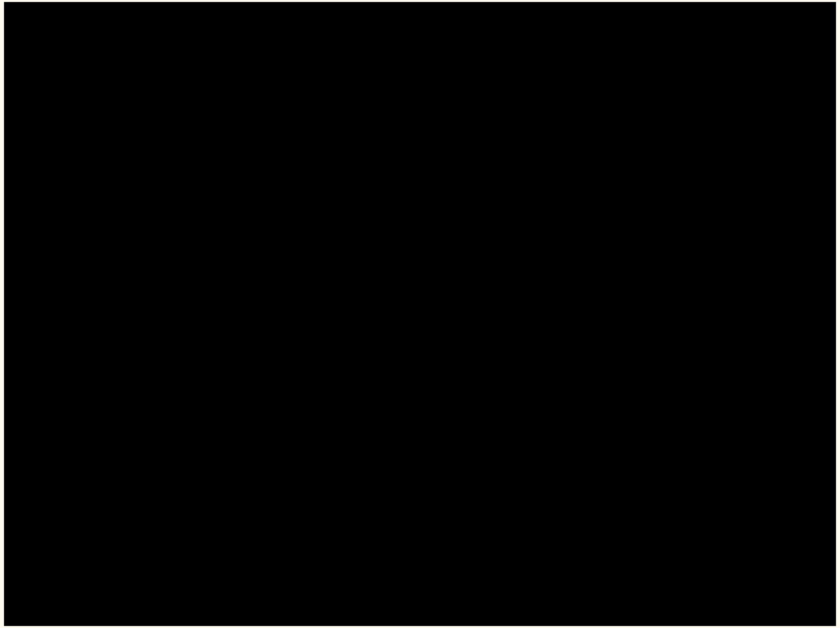


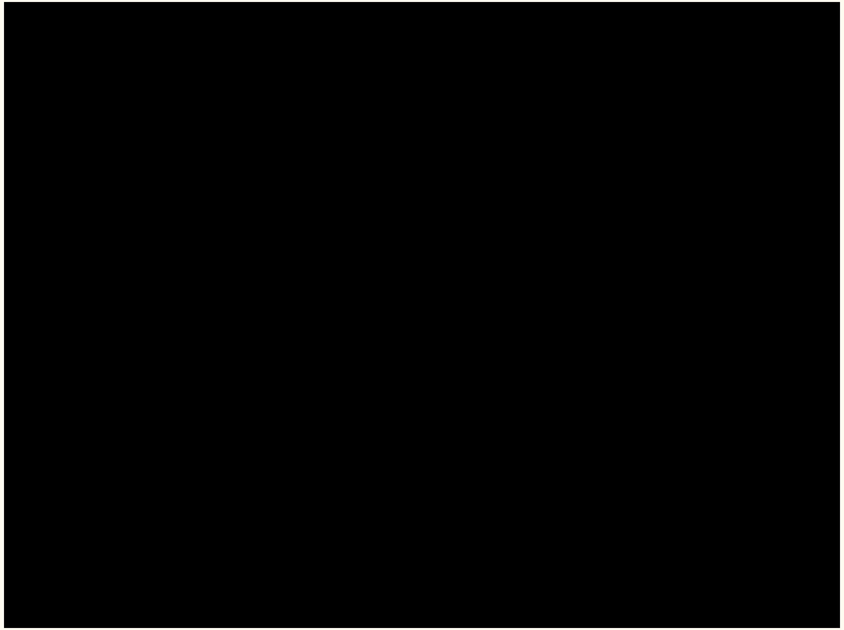
Bayesian Histograms with Greedy Algorithm



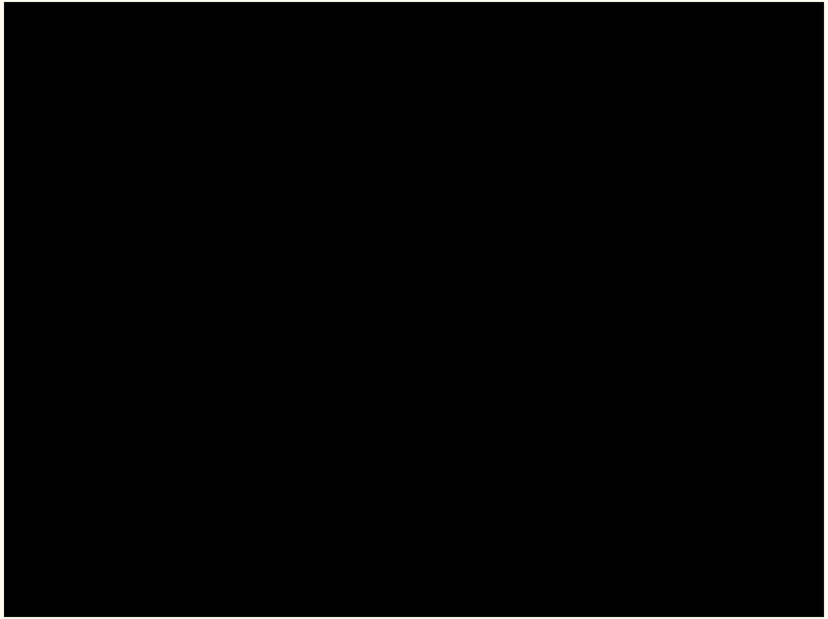


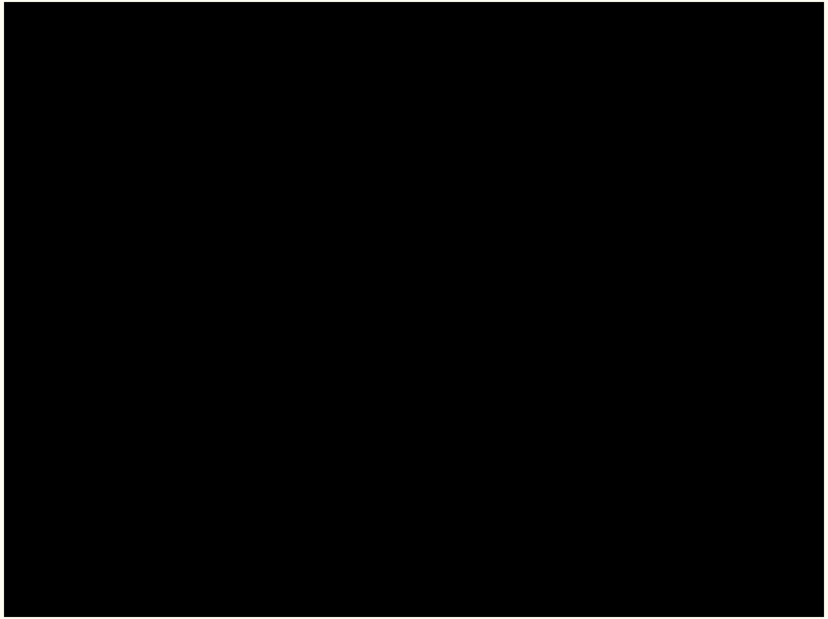
EKF with Stationary robot



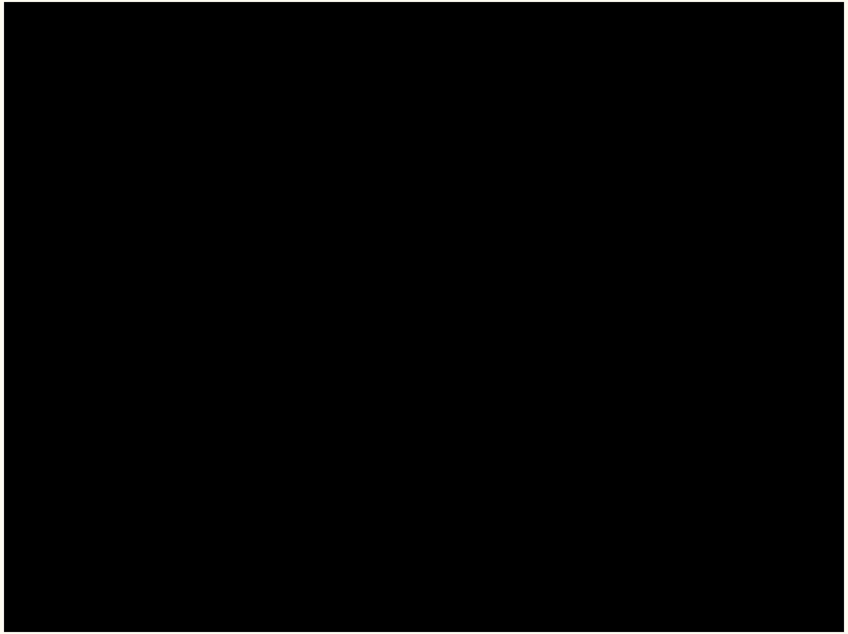


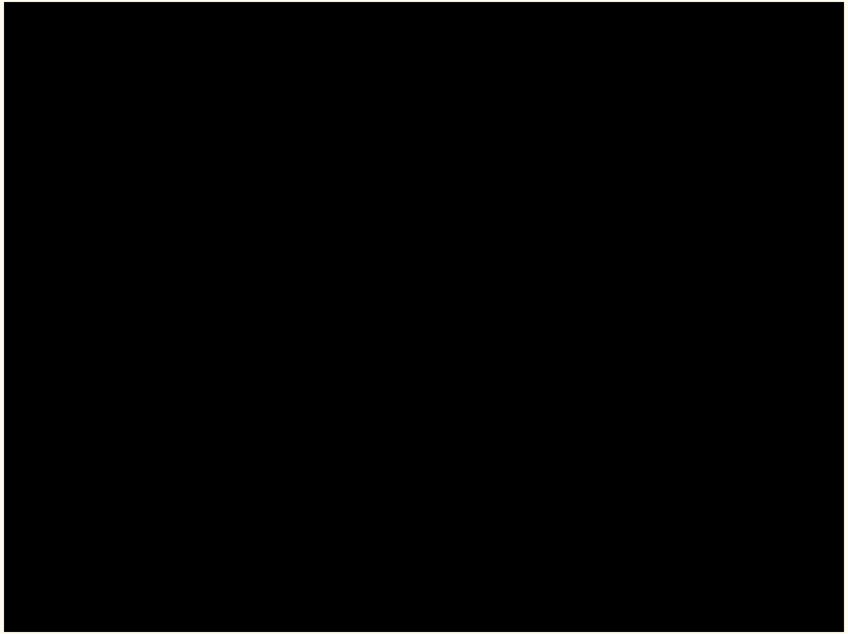
EKF with Greedy Algorithm (sensors=1, targets=1)





EKF with Greedy Algorithm (sensors=1, targets=2)





RL for generating optimal action

1. Greedy algorithm might perform well for single target case. However, for more than one target case, RL needs to be used for generating optimal action.

2. First step was generating EKF heatmaps which would be used as an input to the RL algorithm.

3. Setup of Actor Critic model and deciding the input and output to the model.

RL for generating optimal action (contd)

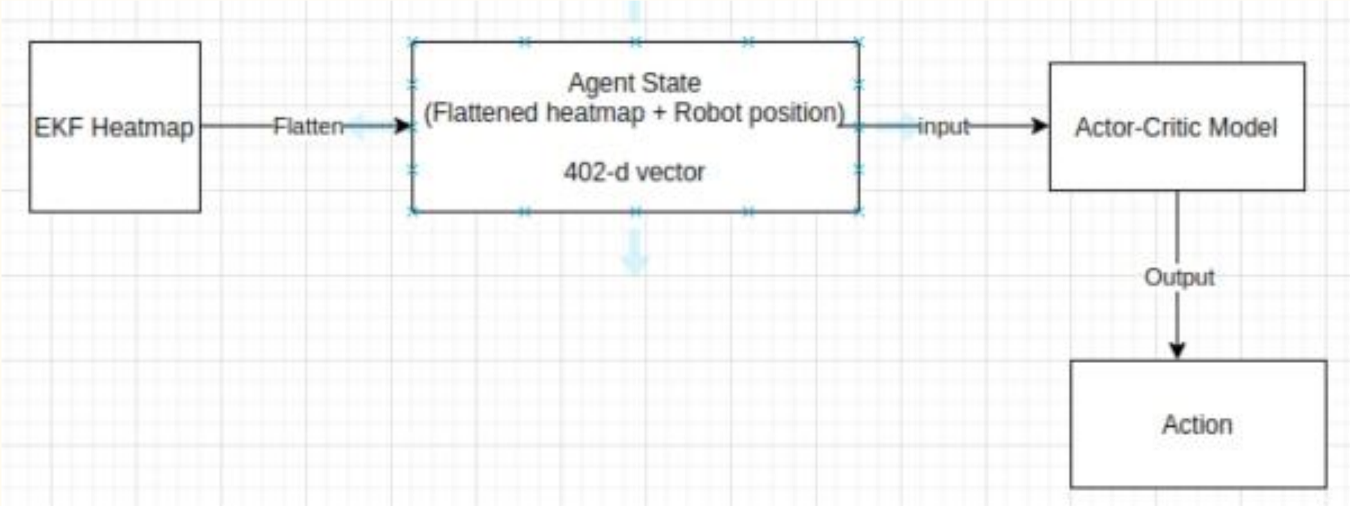
1. **Case 1:** Robots $= 1$ and targets $= 1$

2. **Case 2:** Robots $= 1$ and targets > 1

a. Targets $= 2$, b. Targets $= 4$

3. **Case 3:** Robots > 1 and targets > 1

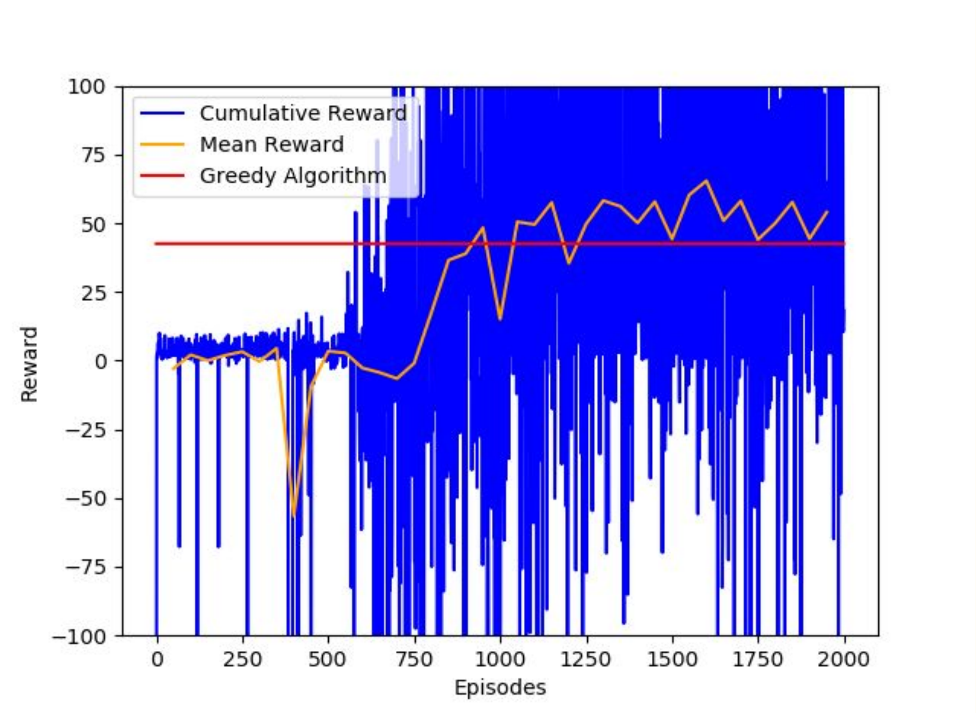
Actor Critic Model (setup, sensors=1)



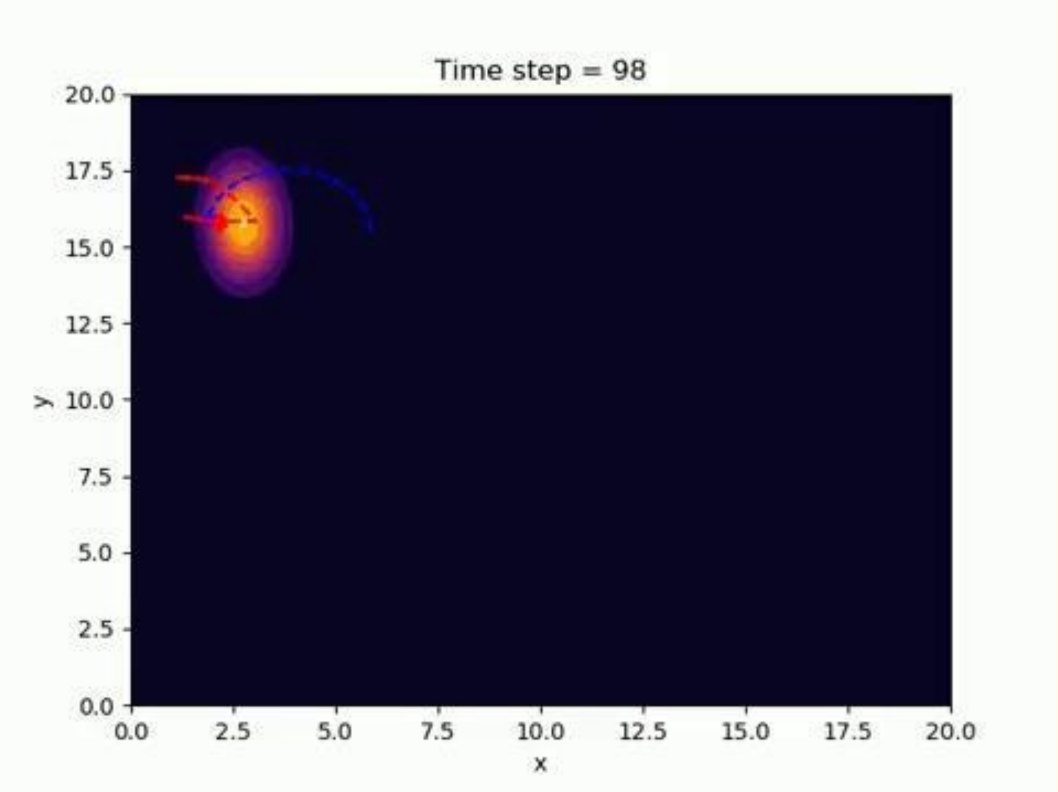
Actor-Critic Model (specifics, sensors=1)

1. **Agent State:** 402-d vector, Flattened vector + robot position
2. **Reward:** -log(det(covariance_matrix))
3. **Action Space:** angle = -pi to pi, step size = 0 to 1

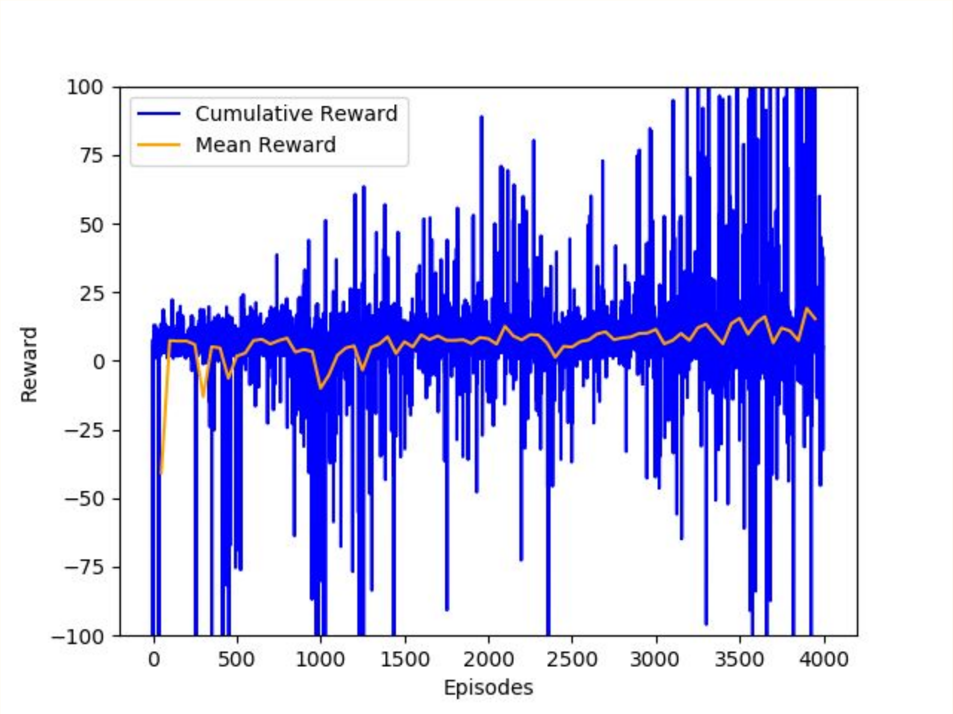
Case 1: Actor-Critic Model Training



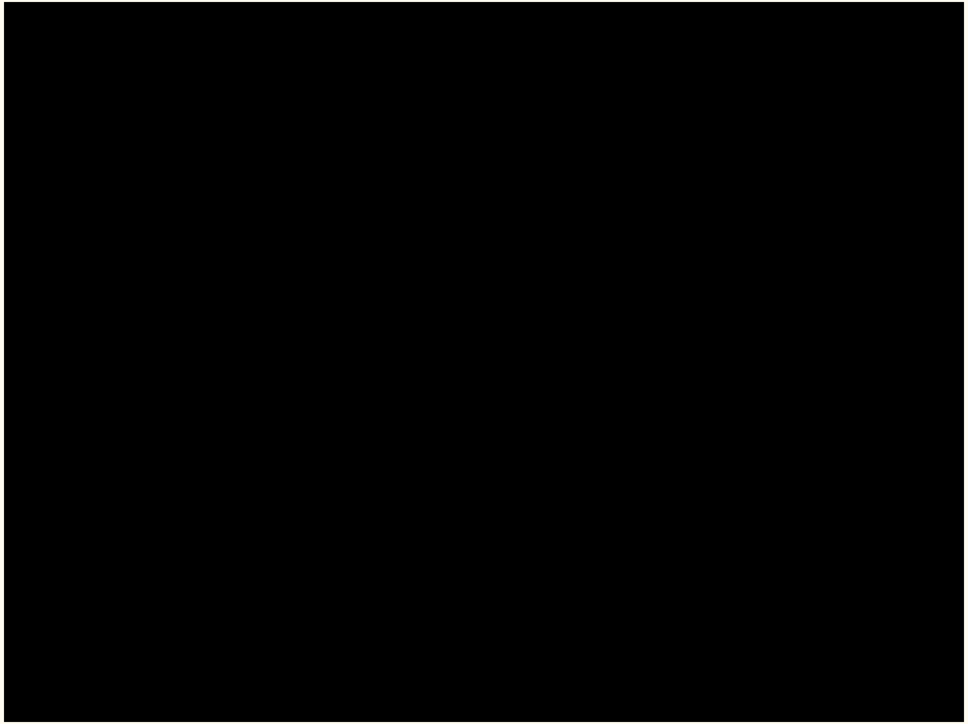
Case 1: Actor-Critic Model Evaluation



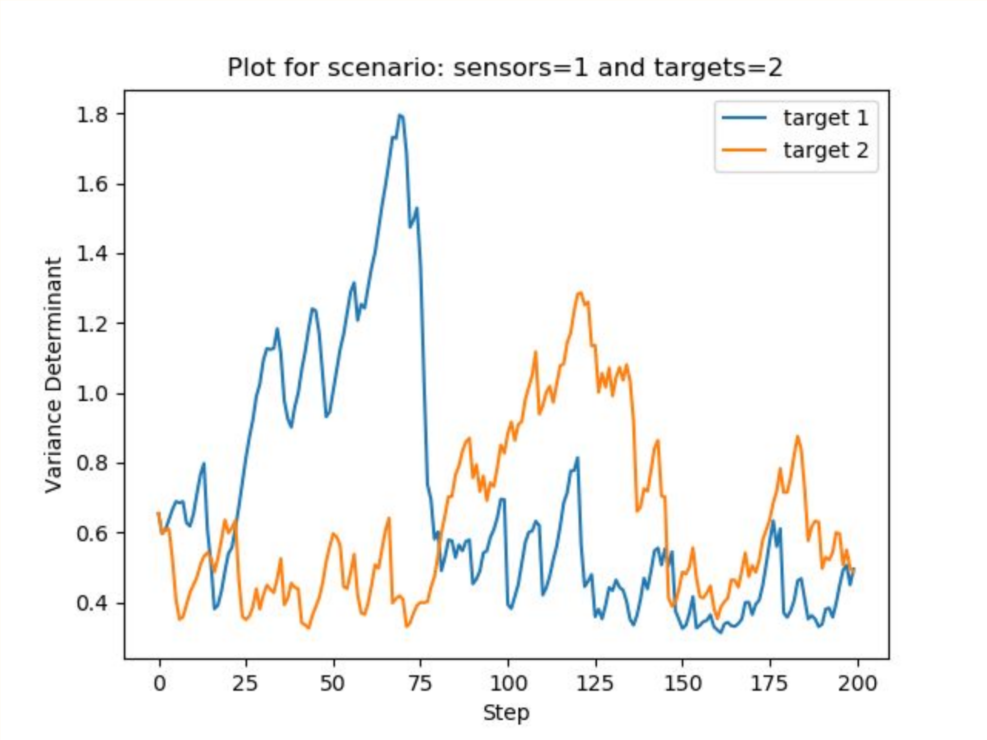
Case 2: Actor-Critic Model Training (targets=2)



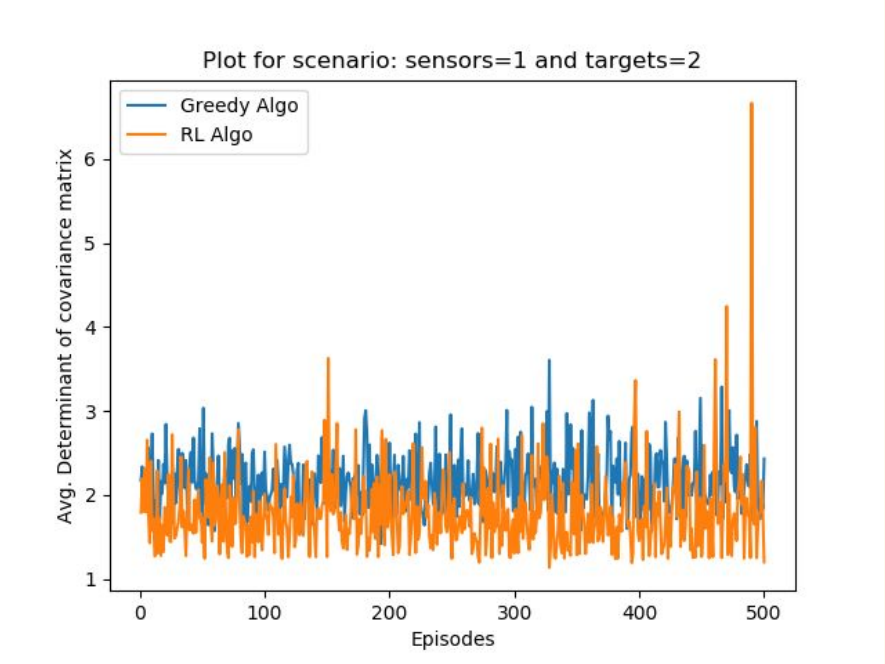
Case 2: Actor-Critic Model Evaluation (targets=2)

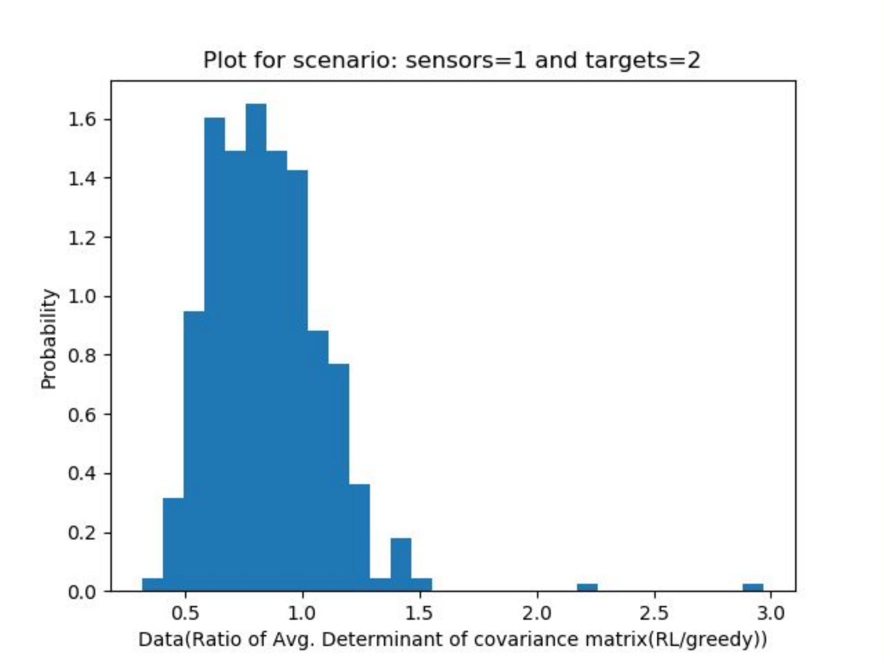


Case 2: Det vs Step Plot (targets=2)

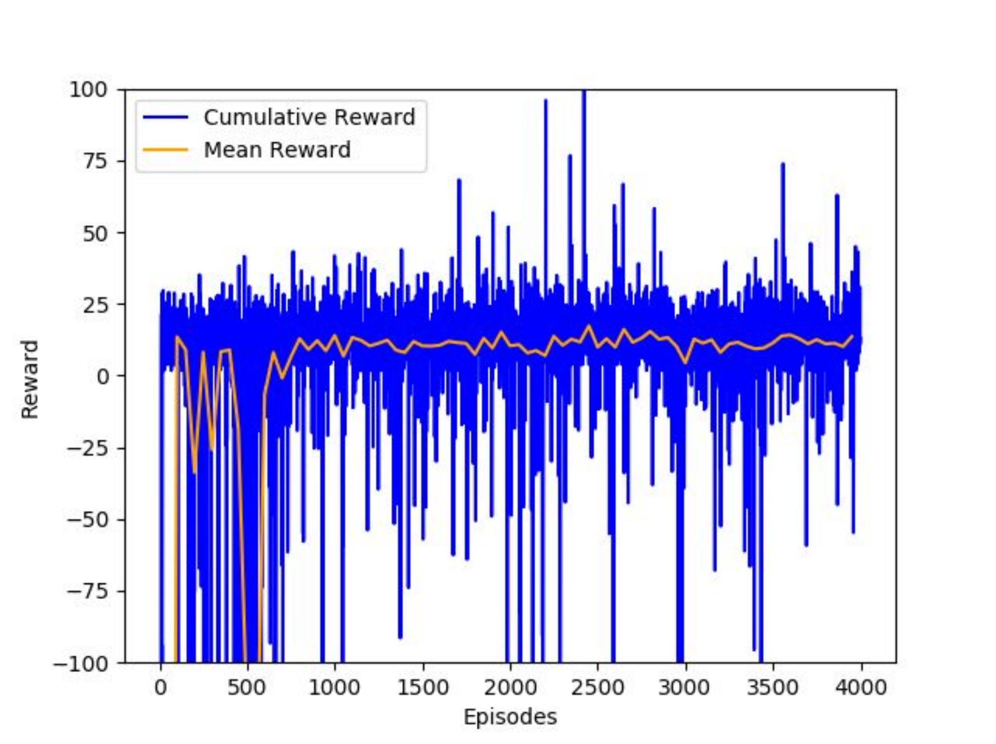


Case 2: RL vs Greedy Algorithms (targets=2)

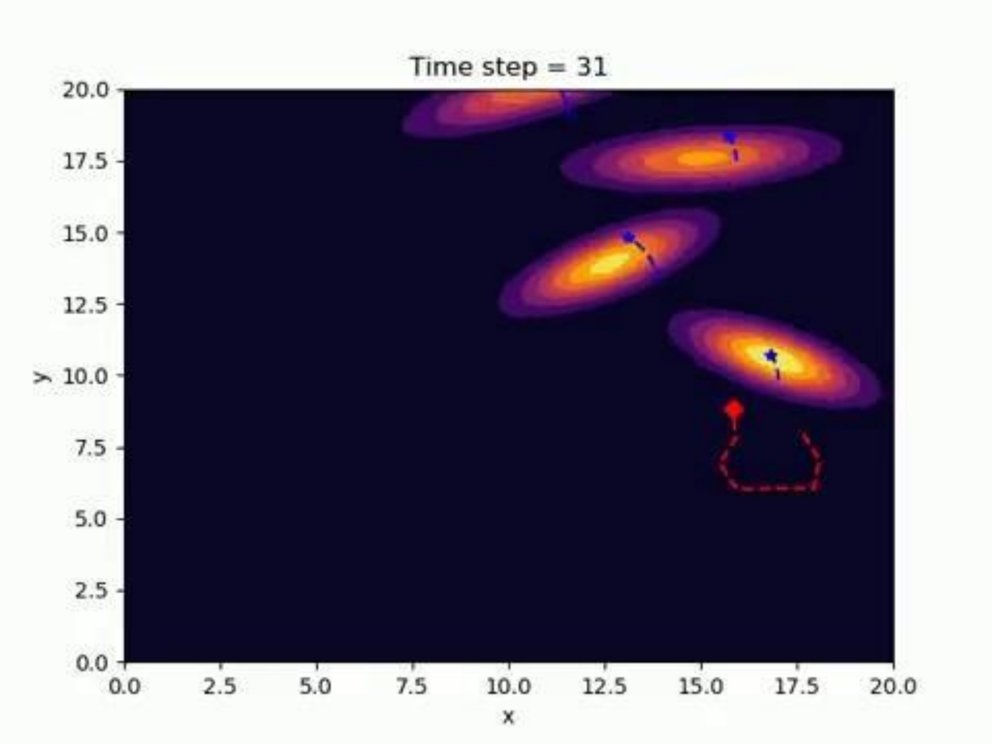




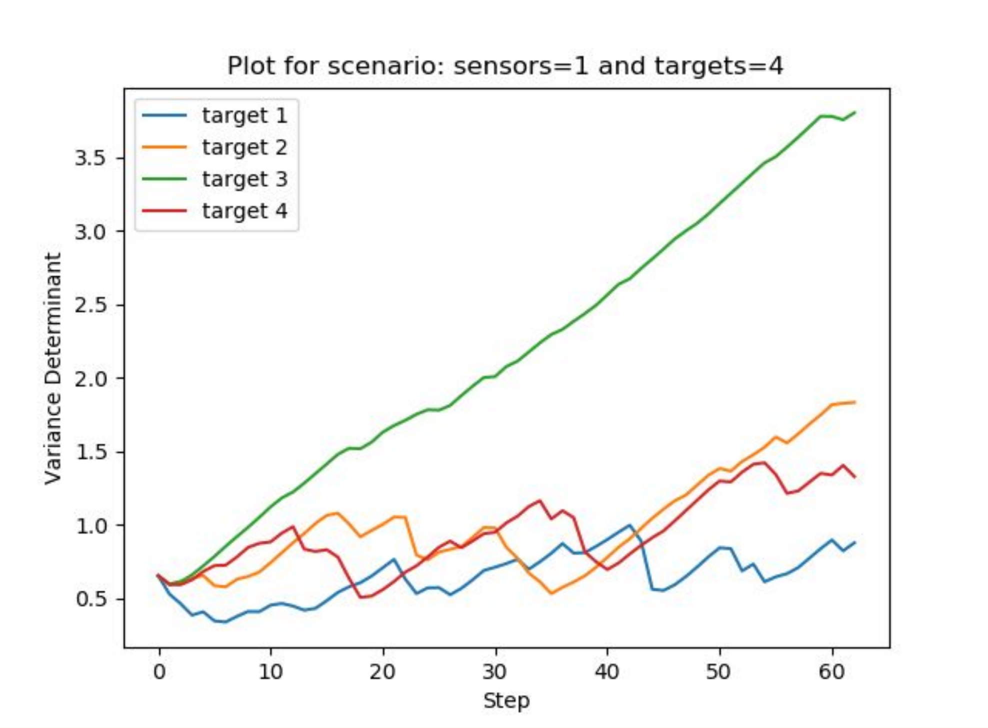
Case 2: Actor-Critic Model Training (targets=4)



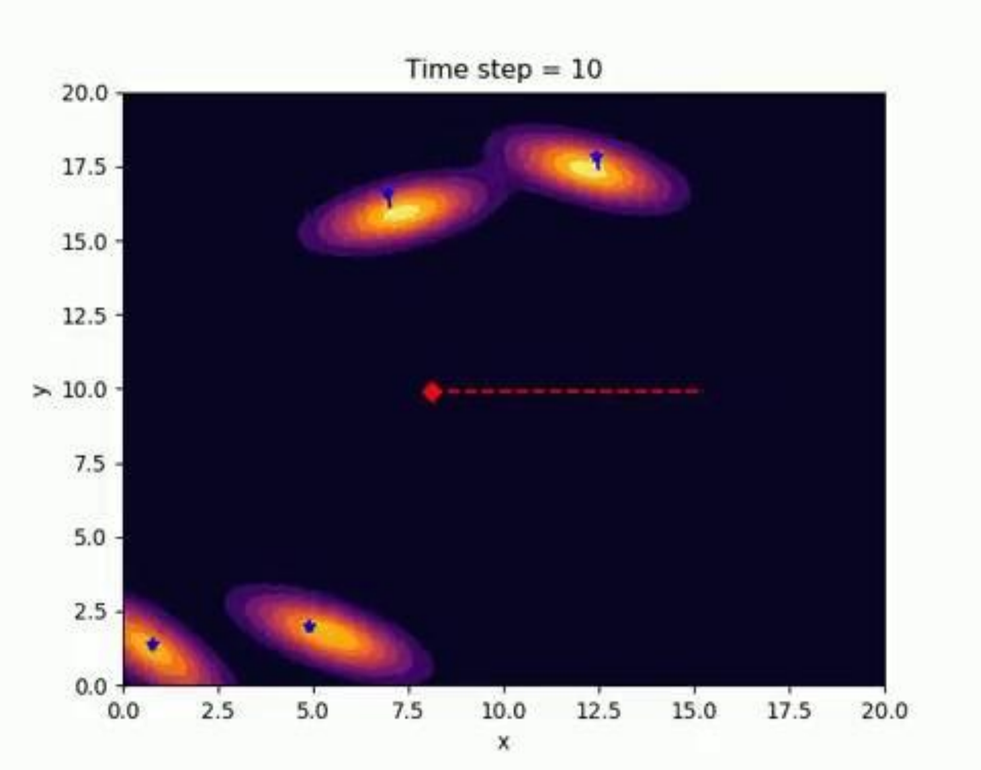
Case 2: Actor-Critic Model Evaluation (targets=4) Best case



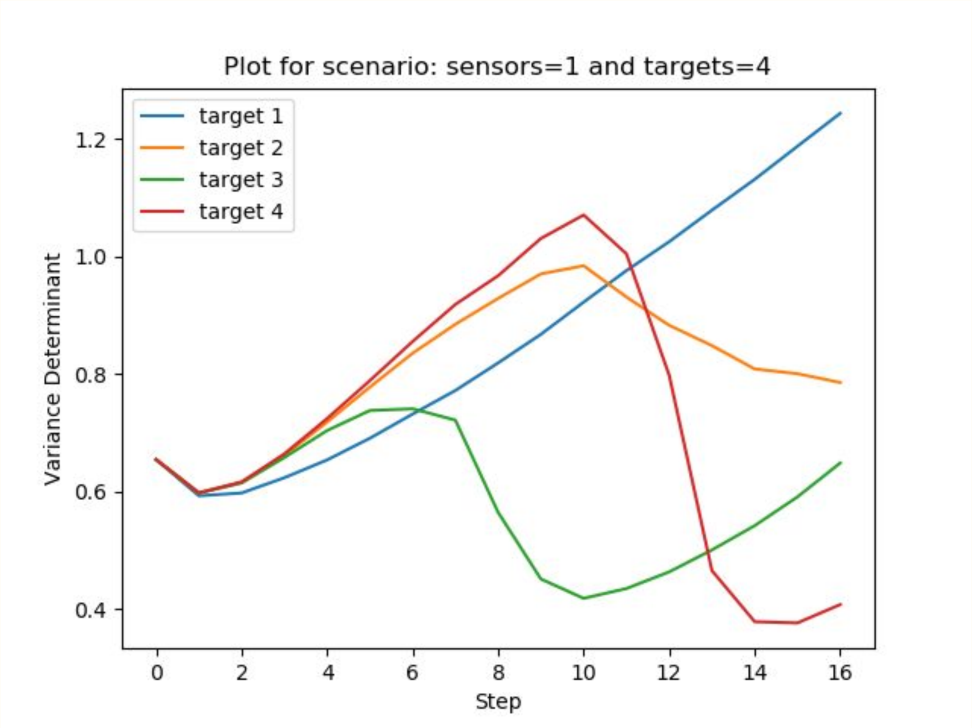
Case 2: Det vs Step Plot (targets=4)



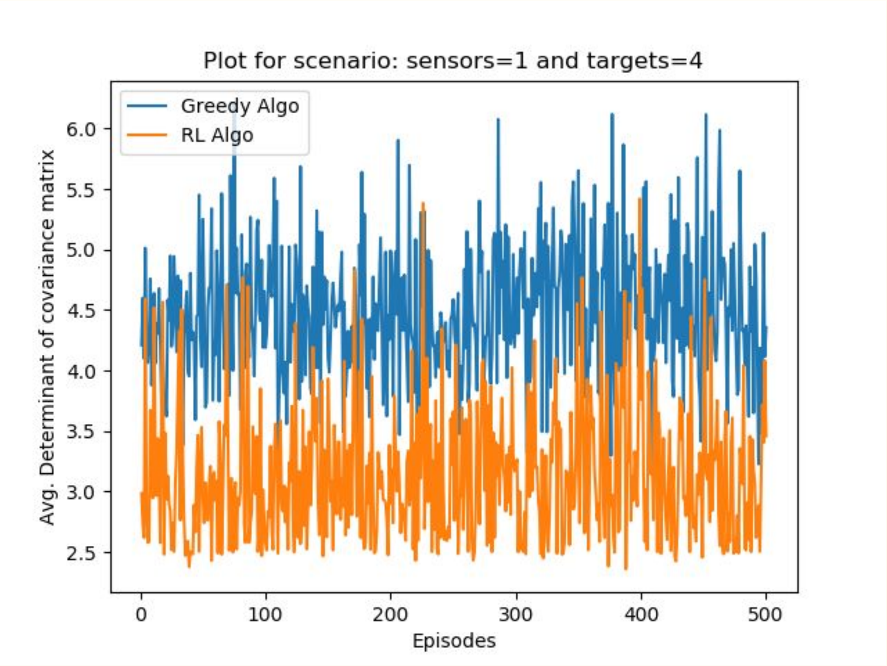
Case 2: Actor-Critic Model Evaluation (targets=4)
Worst case

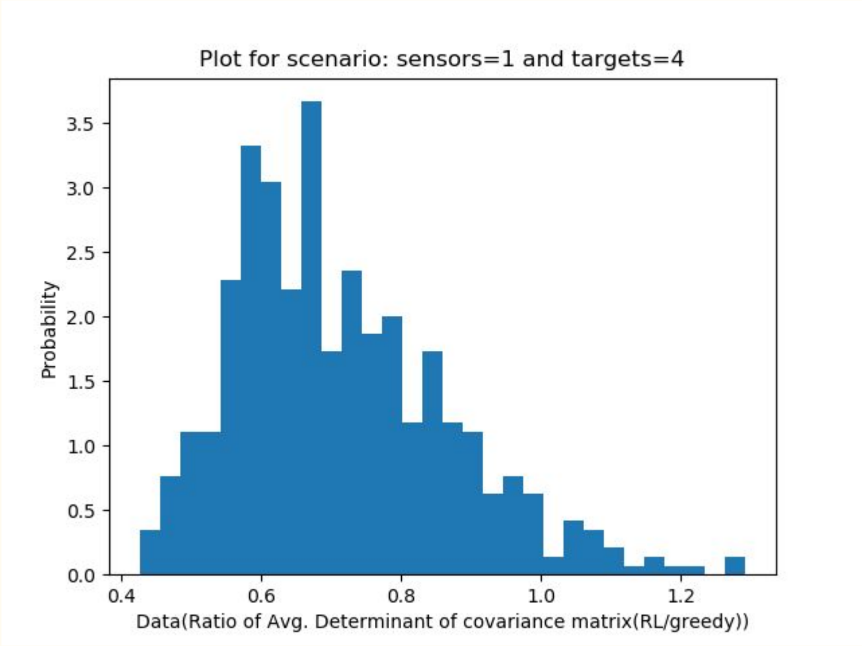


Case 2: Det vs Step Plot (targets=4)

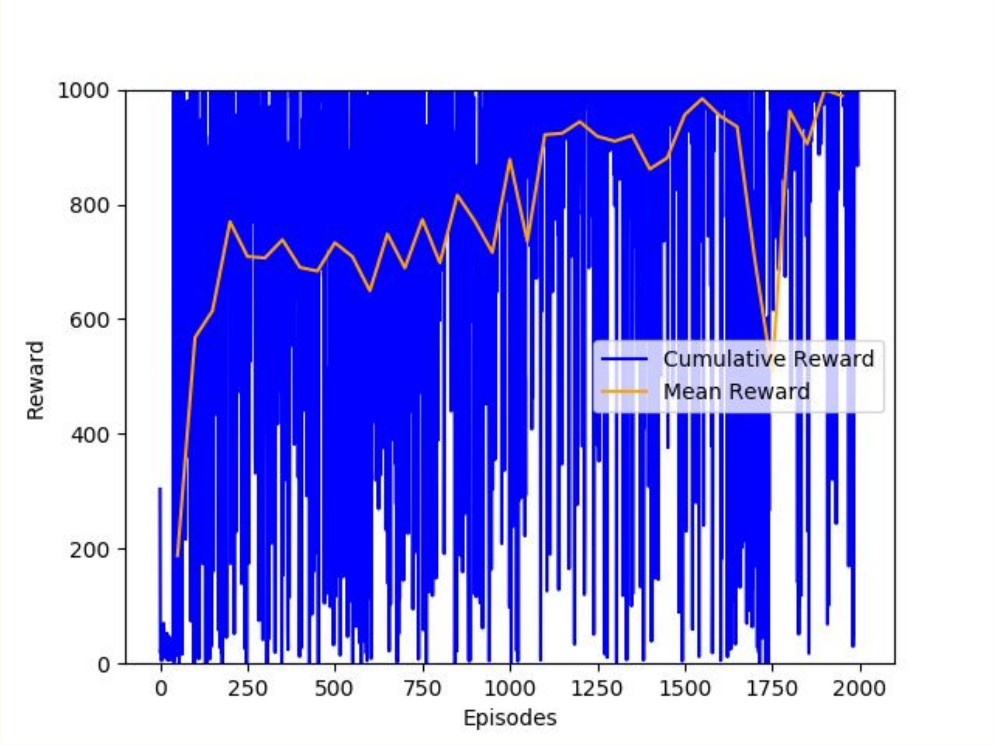


Case 2: RL vs Greedy Algorithms (targets=4)

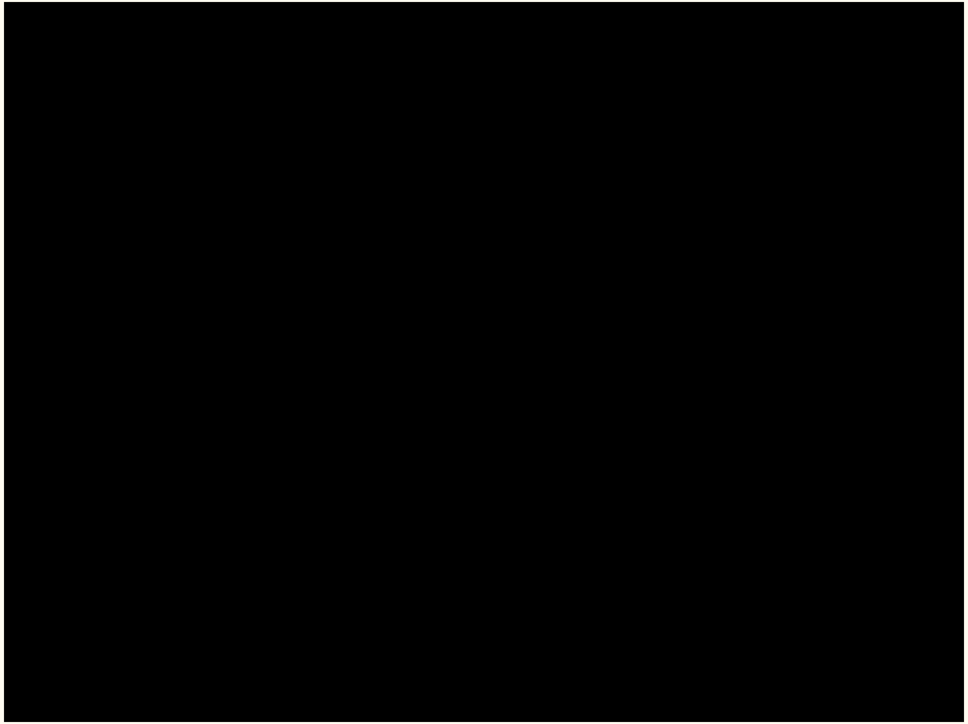




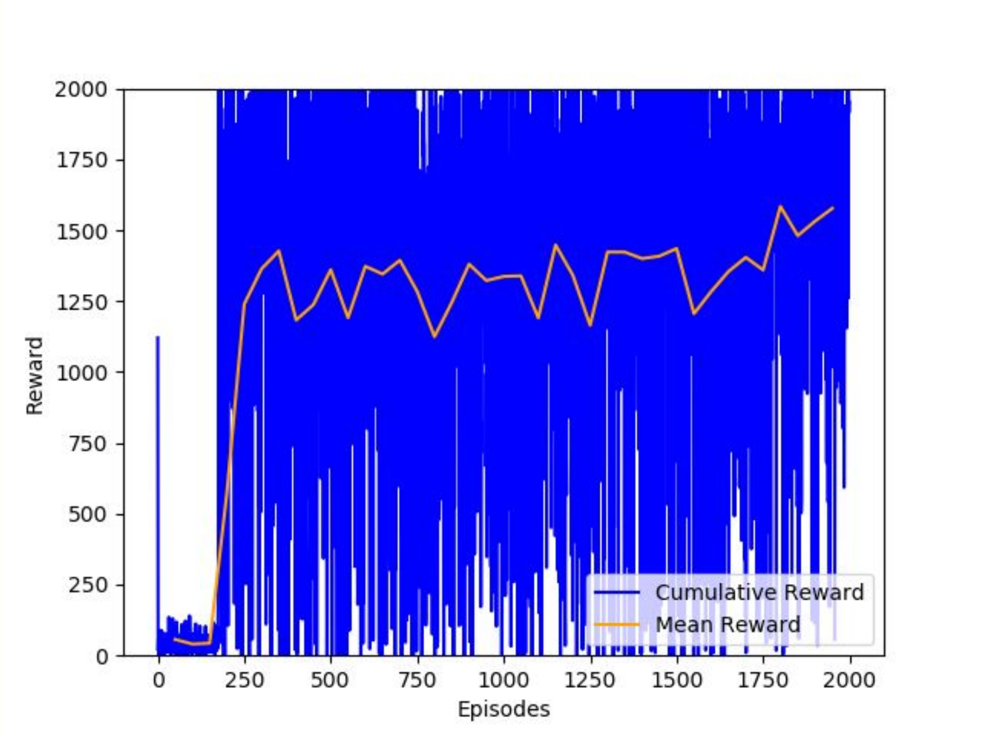
Case 3: Actor-Critic Model Training (sensors=2, targets=2)



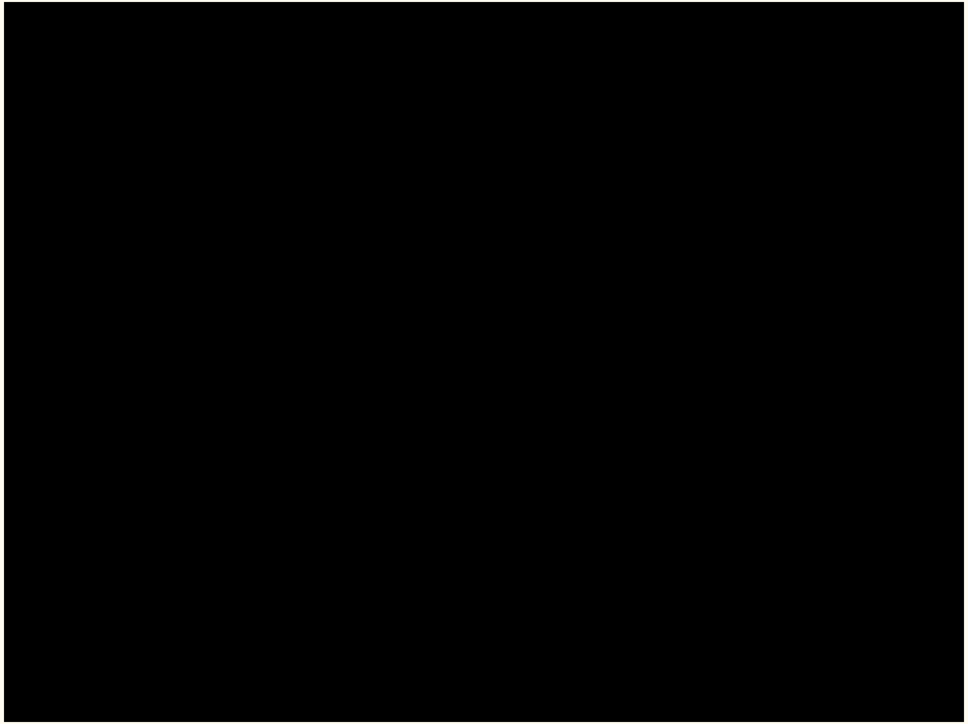
Case 3: Actor-Critic Model Evaluation(sensors=2, targets=2)



Case 3: Actor-Critic Model Training (sensors=2, targets=4)



Case 3: Actor-Critic Model Evaluation(sensors=2, targets=4)

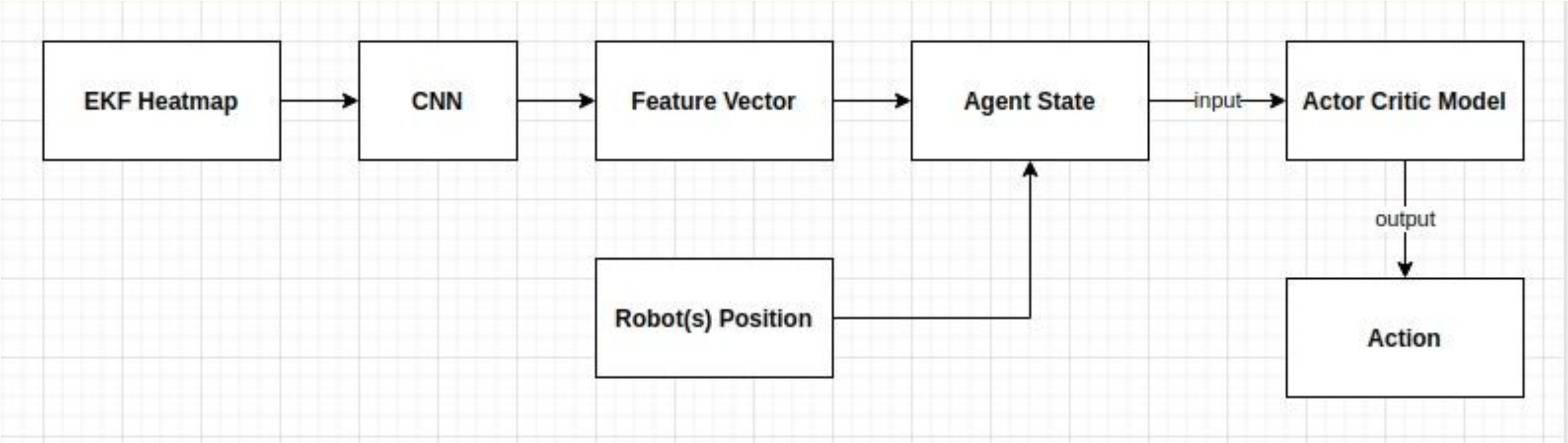


Case 3: Actor-Critic Model with CNN(weights fixed)

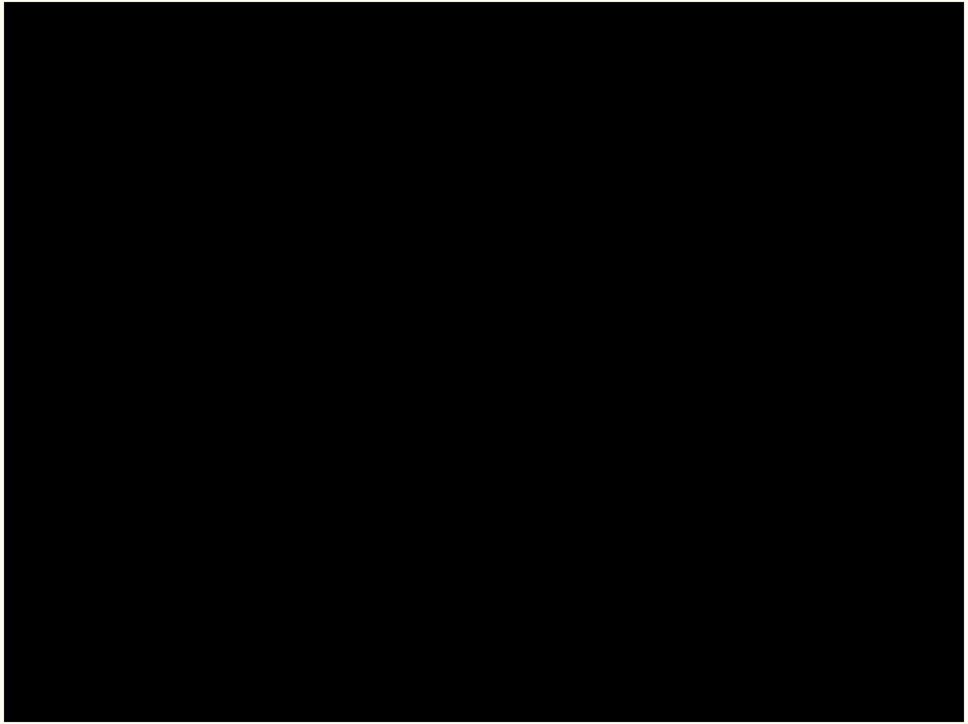
1. Heatmap passed through pretrained ResNet-34

2. Kept CNN weights fixed and the 128 feature vector is used along with robot position(s) as the input to actor-critic model

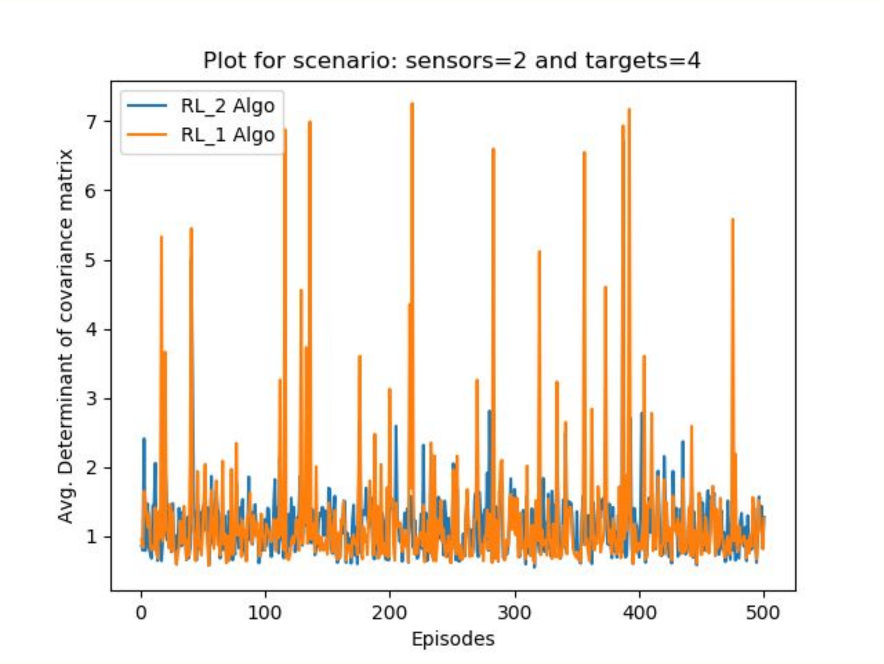
Case 3: Actor-Critic Model with CNN(weights fixed)

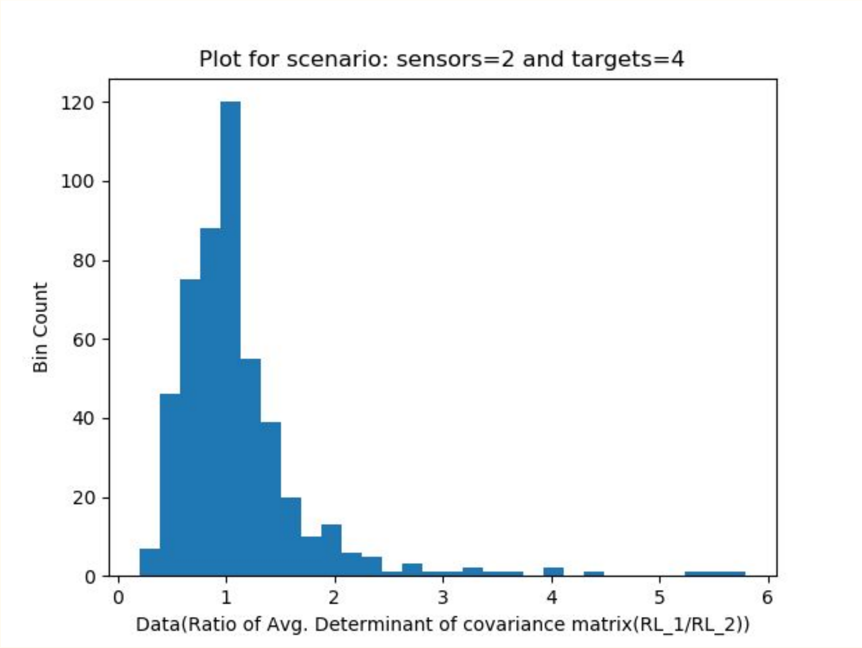


Case 3: Actor-Model Evaluation(sensors=2, targets=4)



Comparison of Actor-Critic Models



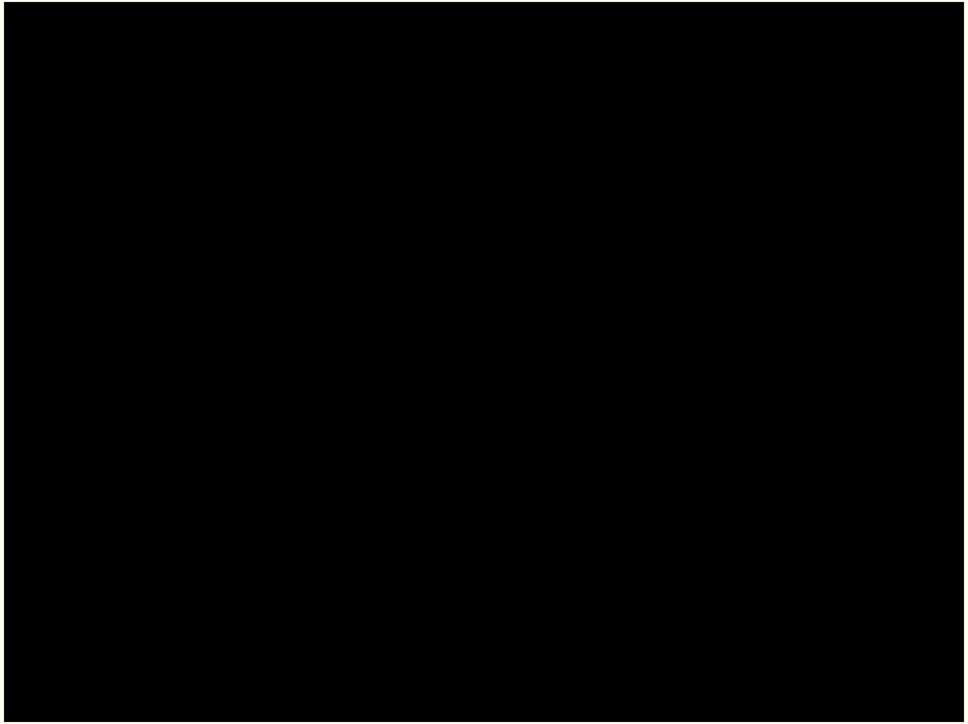


Case 3: Actor-Critic Model with CNN

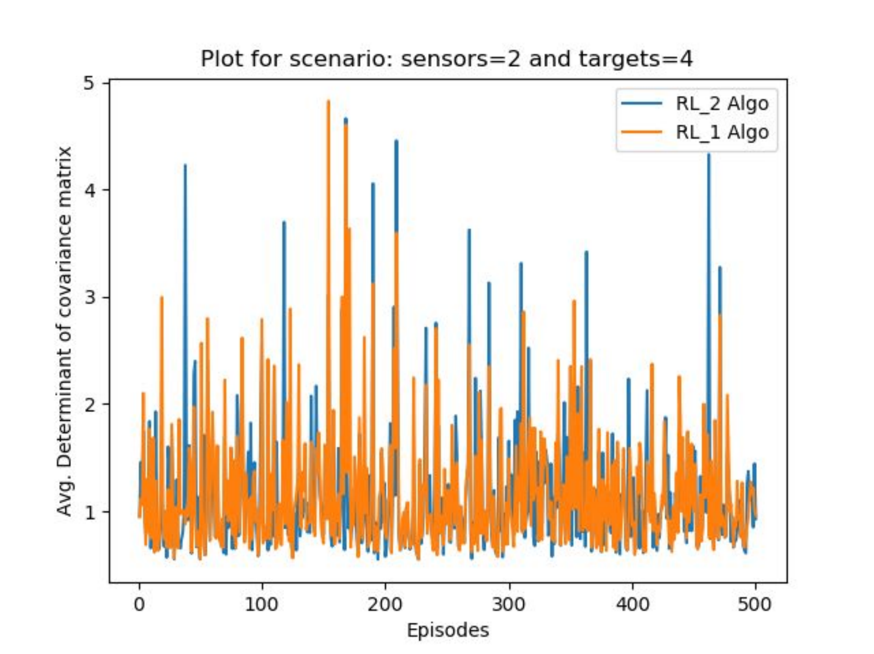
1. Heatmap passed through pretrained ResNet-18

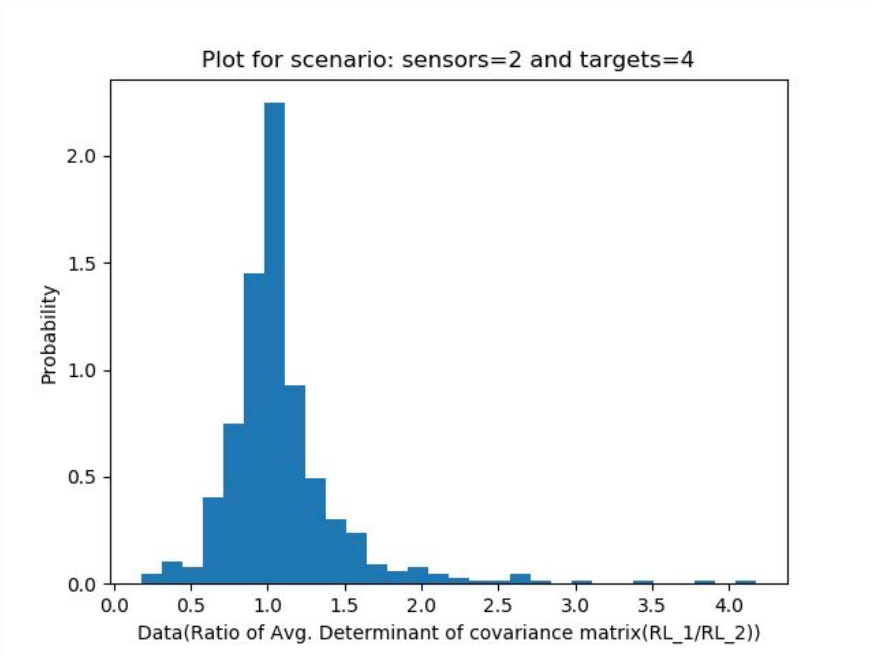
2. Kept CNN weights fixed and the 128 feature vector is used along with robot position(s) as the input to actor-critic model

Case 3: Actor-Model Evaluation(sensors=2, targets=4)

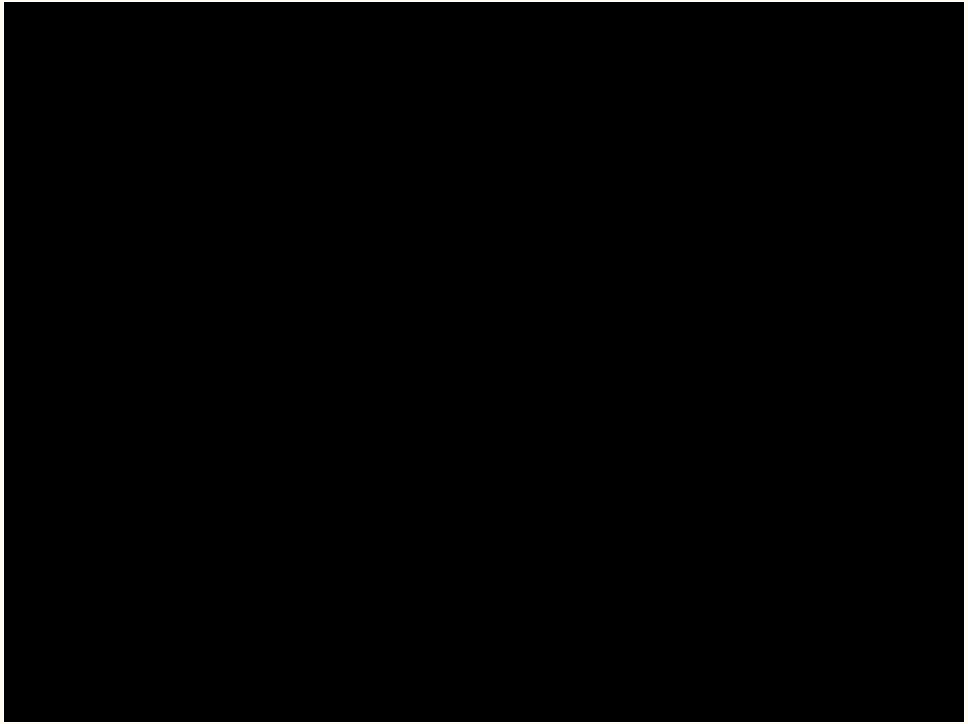


Comparison of Actor-Critic Models

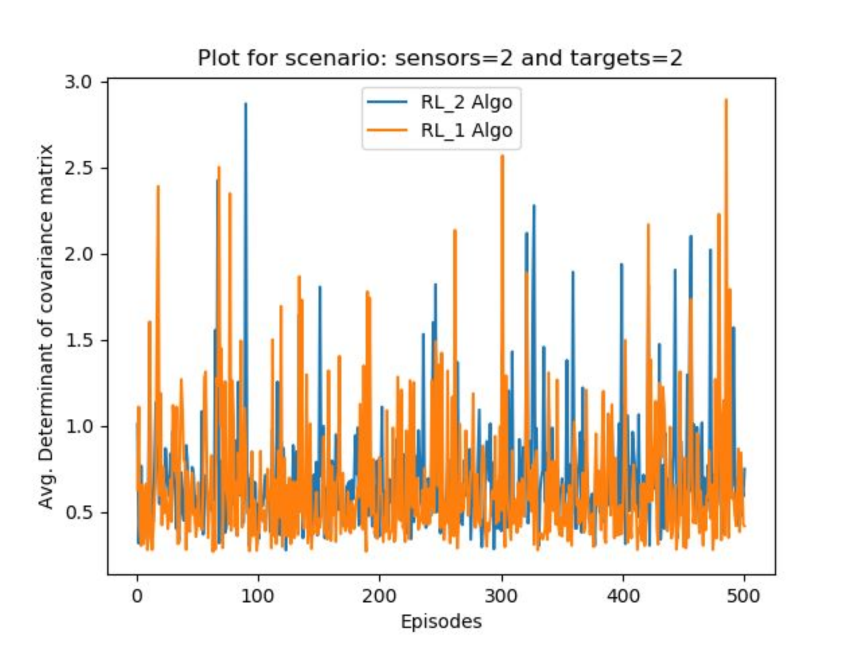


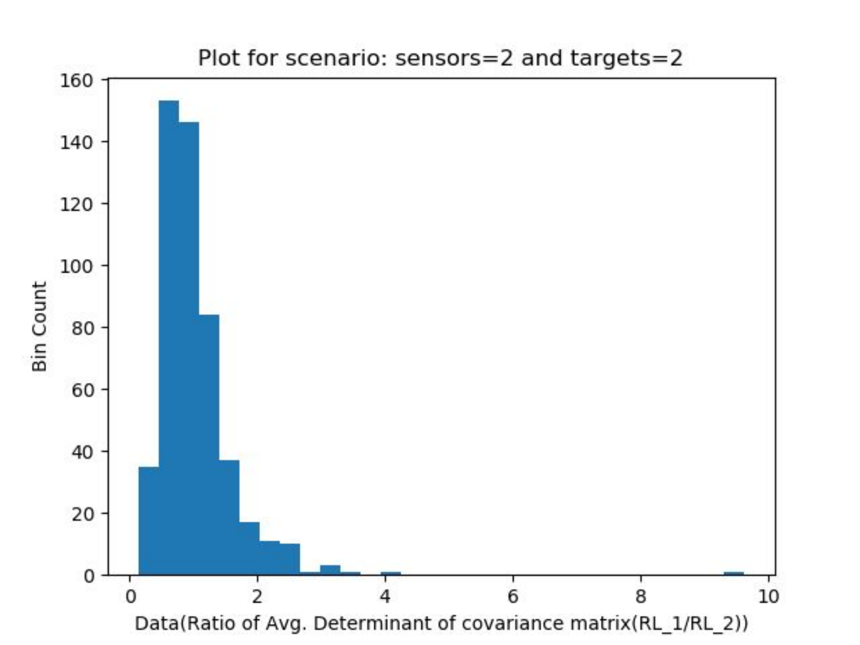


Case 3: Actor-Model Evaluation(sensors=2, targets=2)

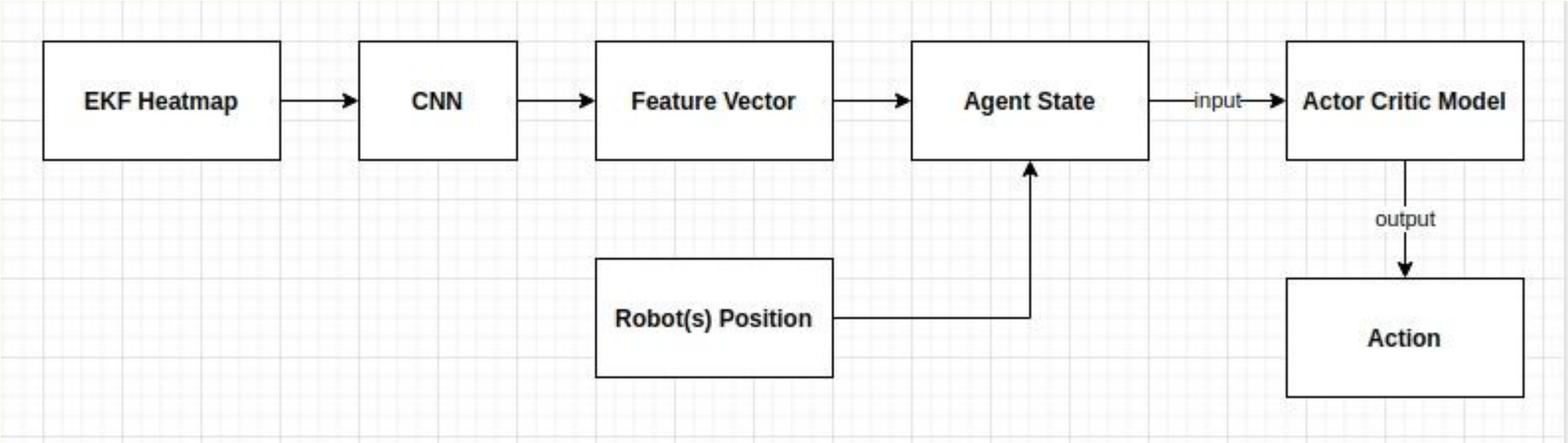


Comparison of Actor-Critic Models

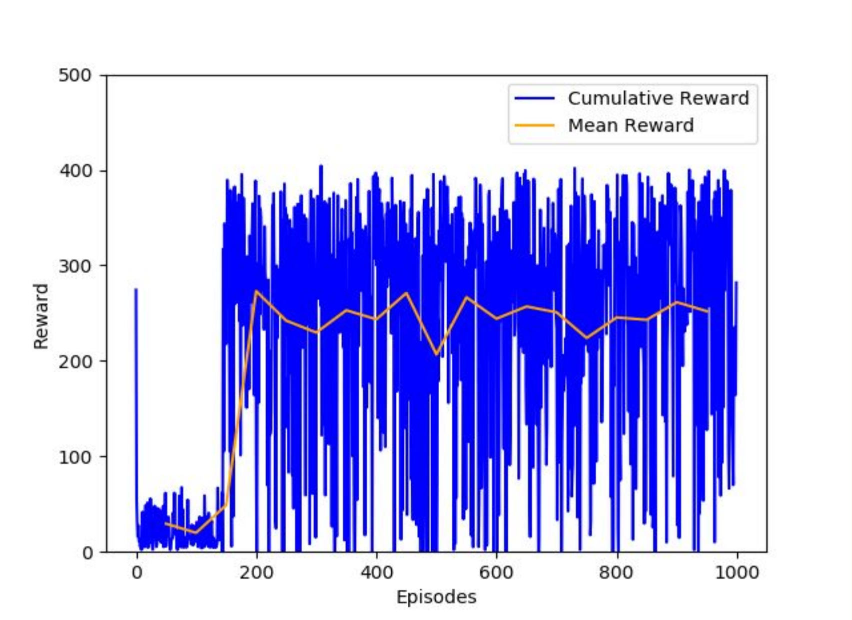


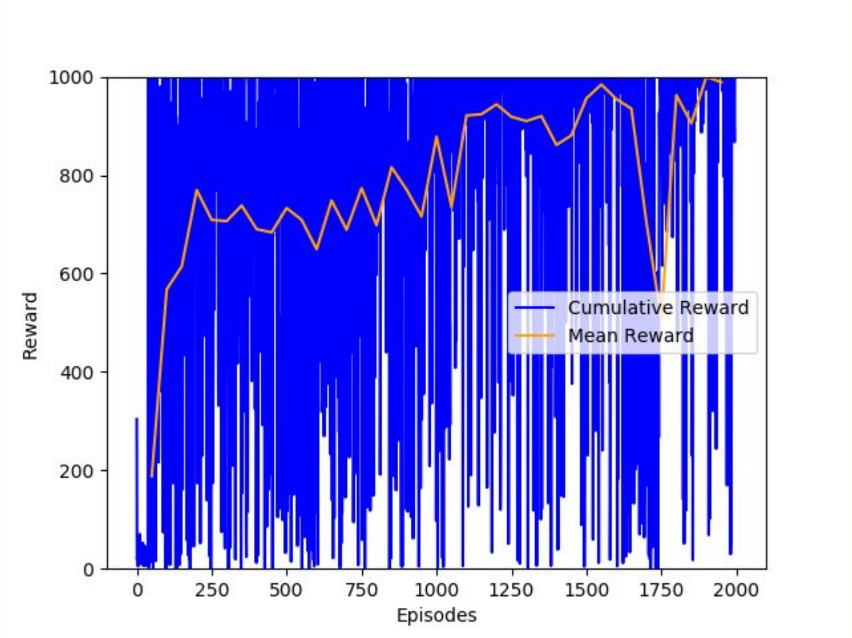


Case 3: Actor-Critic Model with CNN(weights not fixed)

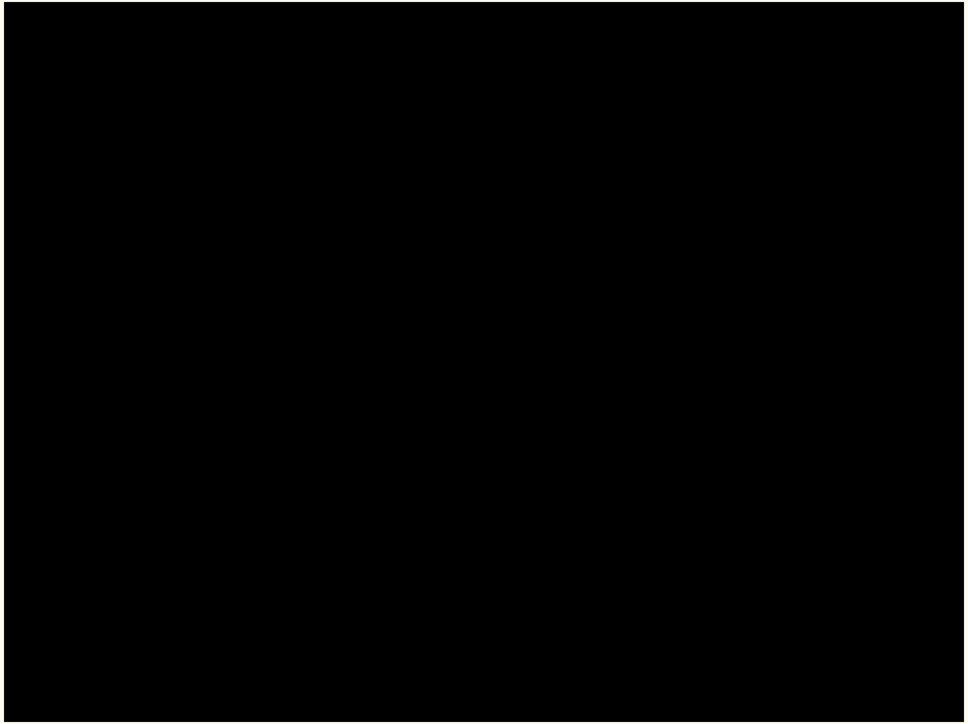


Case 3: Actor-Critic Model Training (sensors=2, targets=2)

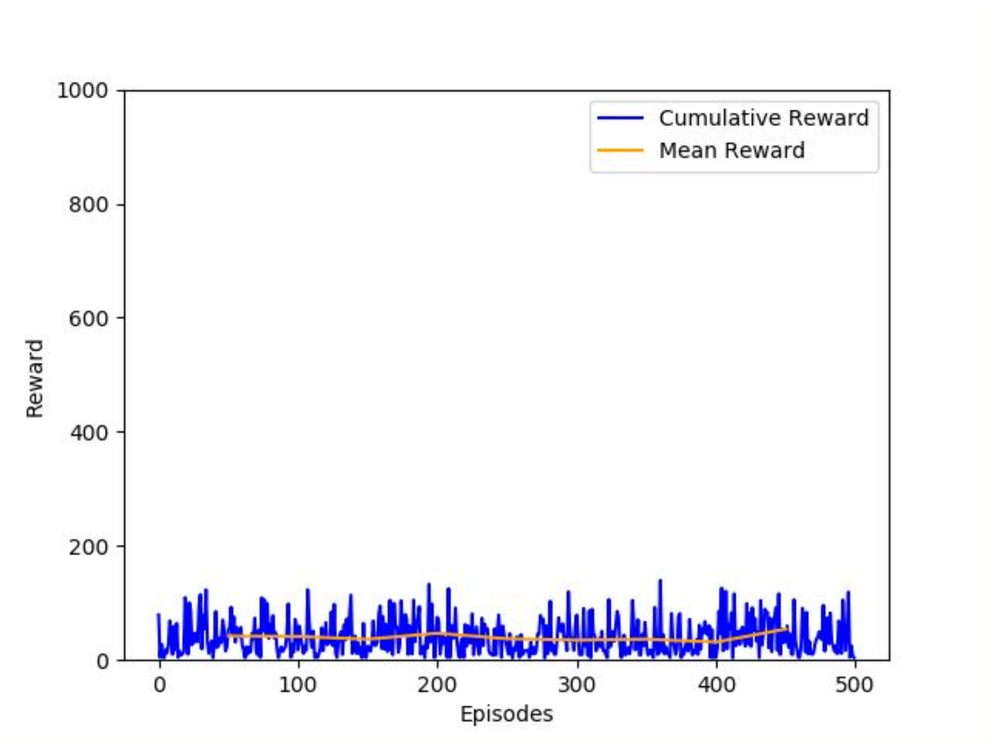




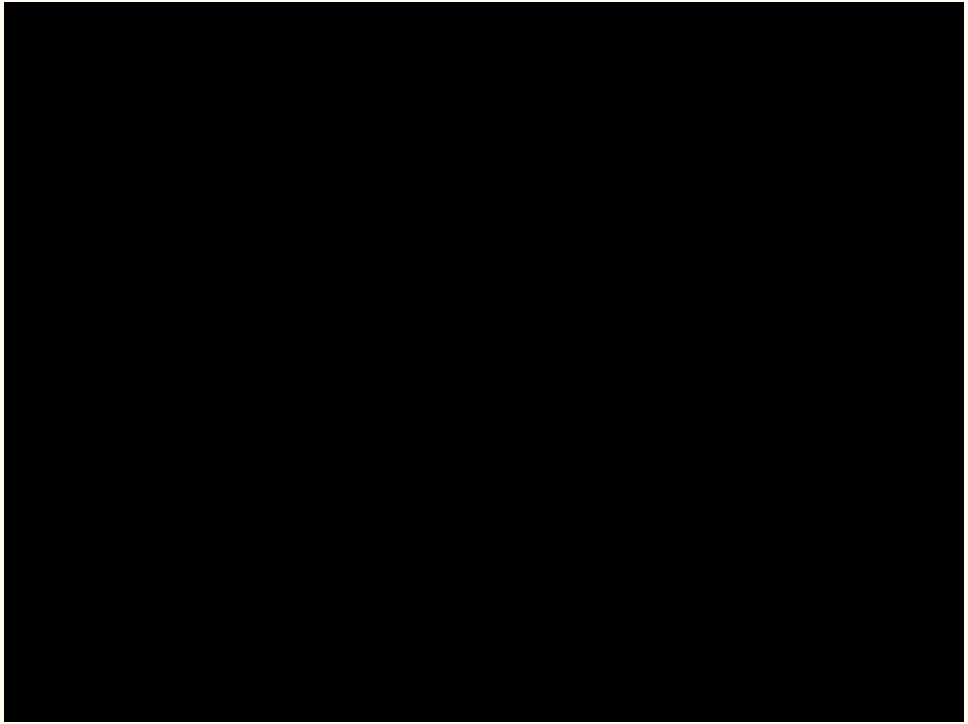
Case 3: Actor-Critic Model Evaluation (sensors=2, targets=2)



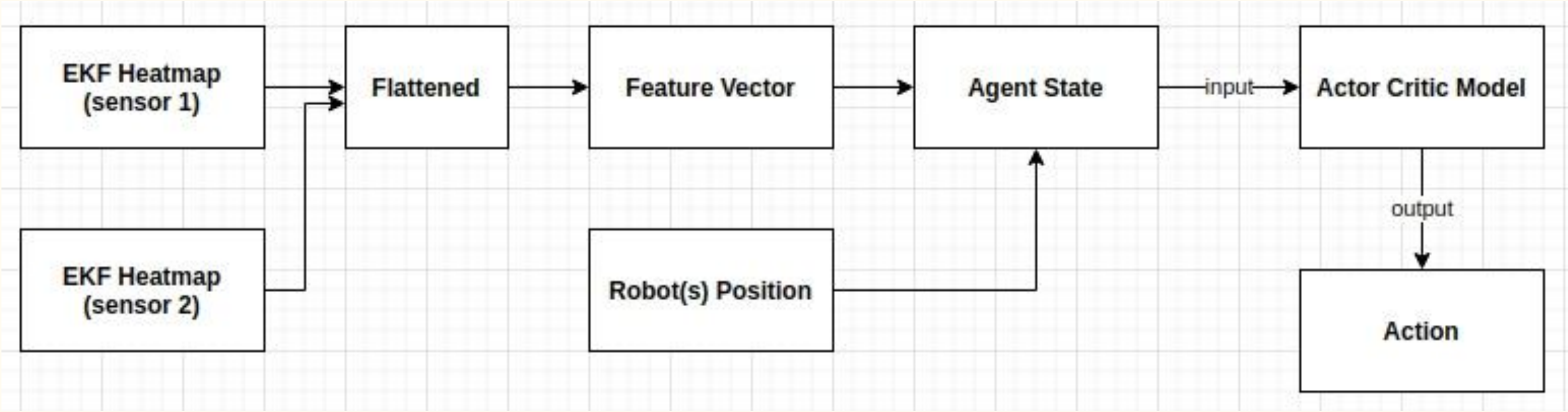
Case 3: Actor-Critic Model Training (sensors=2, targets=4)



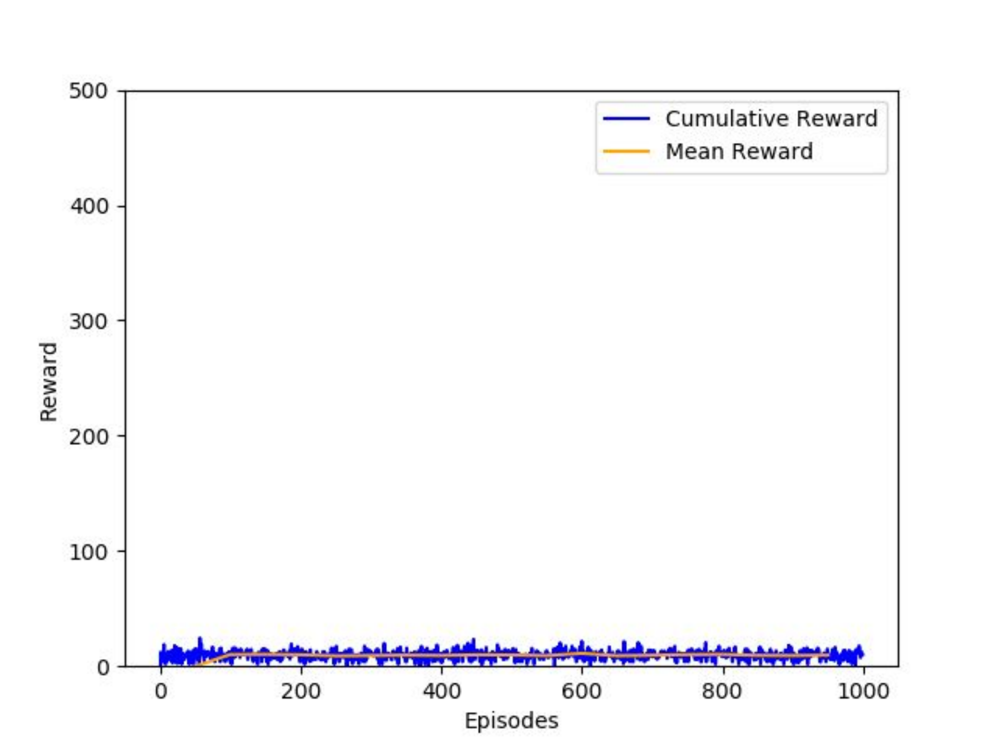
Case 3: Actor-Model Evaluation(sensors=2, targets=4)



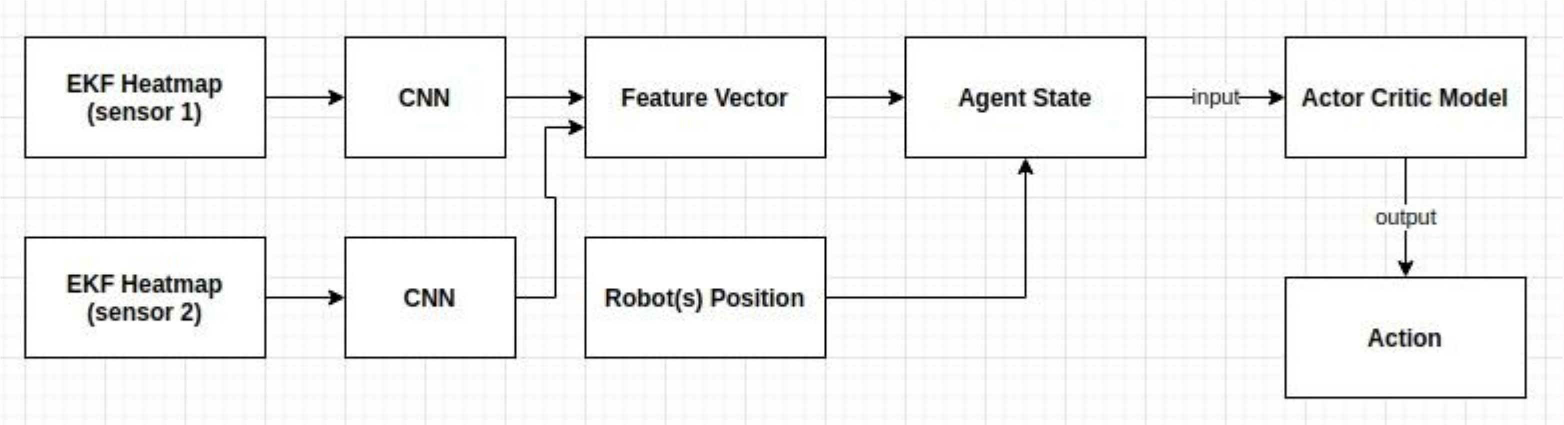
Case 3: Actor-Critic Model with CNN(weights not fixed)



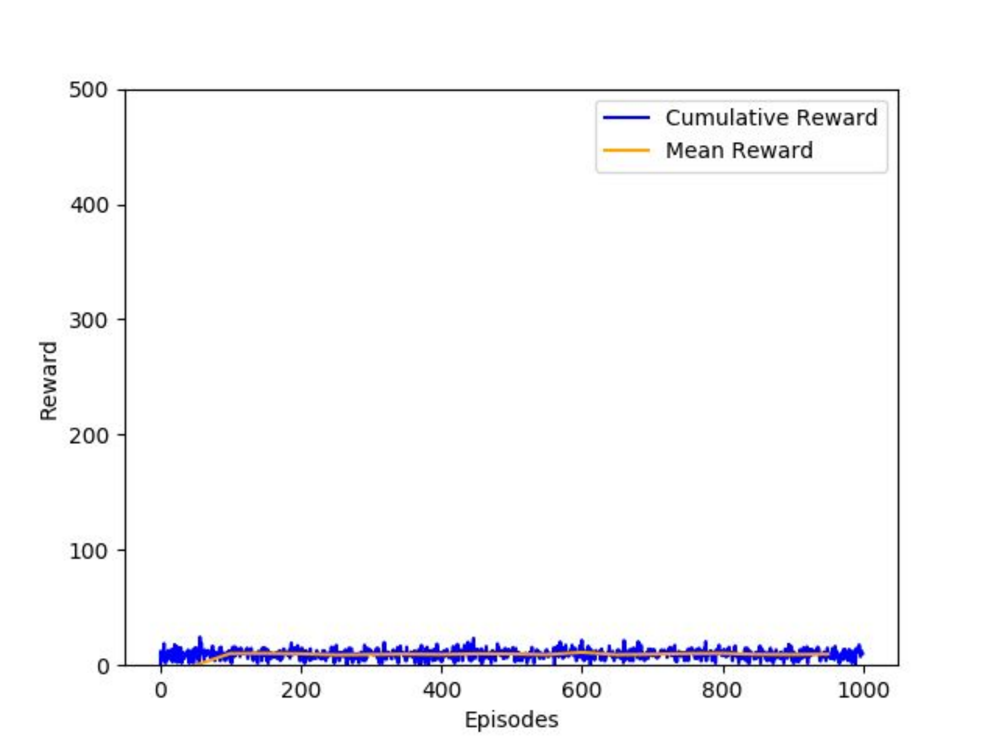
Case 3: Actor-Critic Model Training (sensors=2, targets=2)



Case 3: Actor-Critic Model with CNN(weights not fixed)



Case 3: Actor-Critic Model Training (sensors=2, targets=2)



Future Plan(if given more time)

Extend work on the heatmap generation for each sensor and compare the results where only one heatmap is generated for the multi-robot case

Instead of concatenating the robot position with the final feature vector obtained from the CNN, try using the image with 2 channels where the first channel encodes the heatmap information and second channel encodes the robot position as the input to the CNN