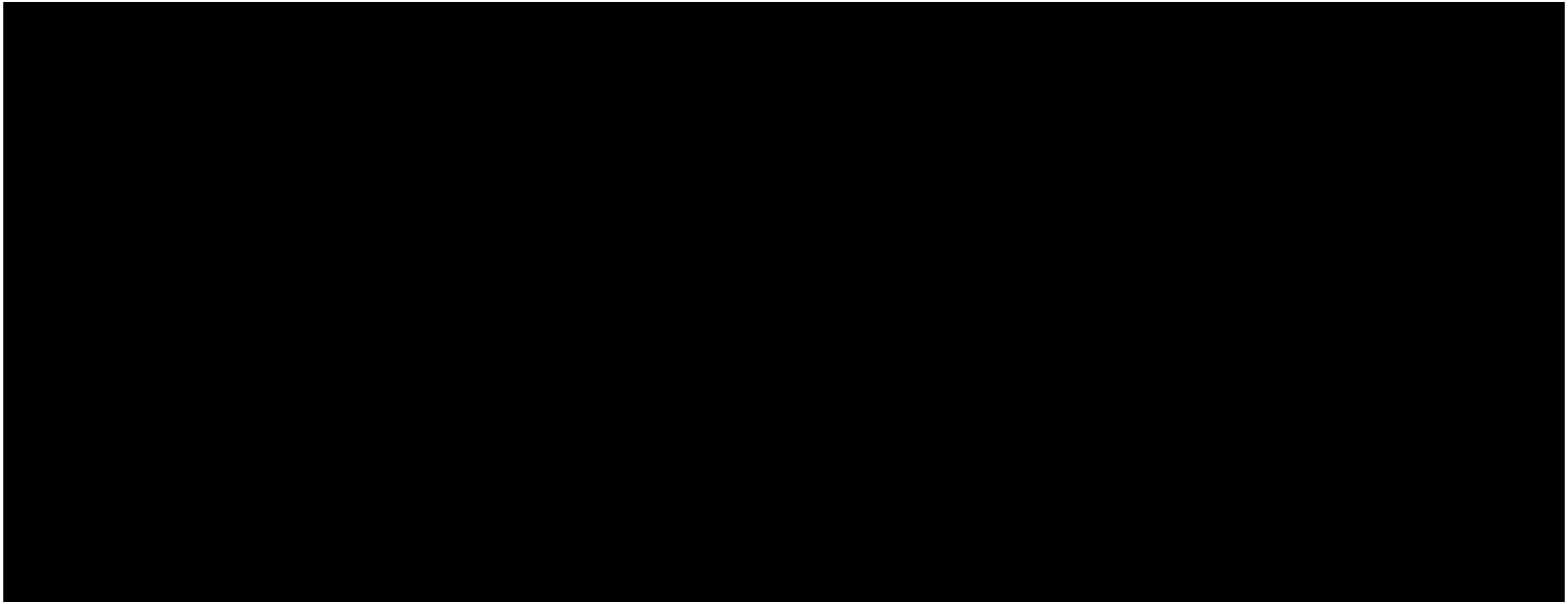
Deep R Learning for Continual Area Sweeping

Rishi Shah*, Yuqian Jiang*, Justin Hart, Peter Stone

The University of Texas at Austin
Department of Computer Science
College of Natural Sciences

Learning Agents Research Group
The University of Texas at Austin

Motivation



Motivation

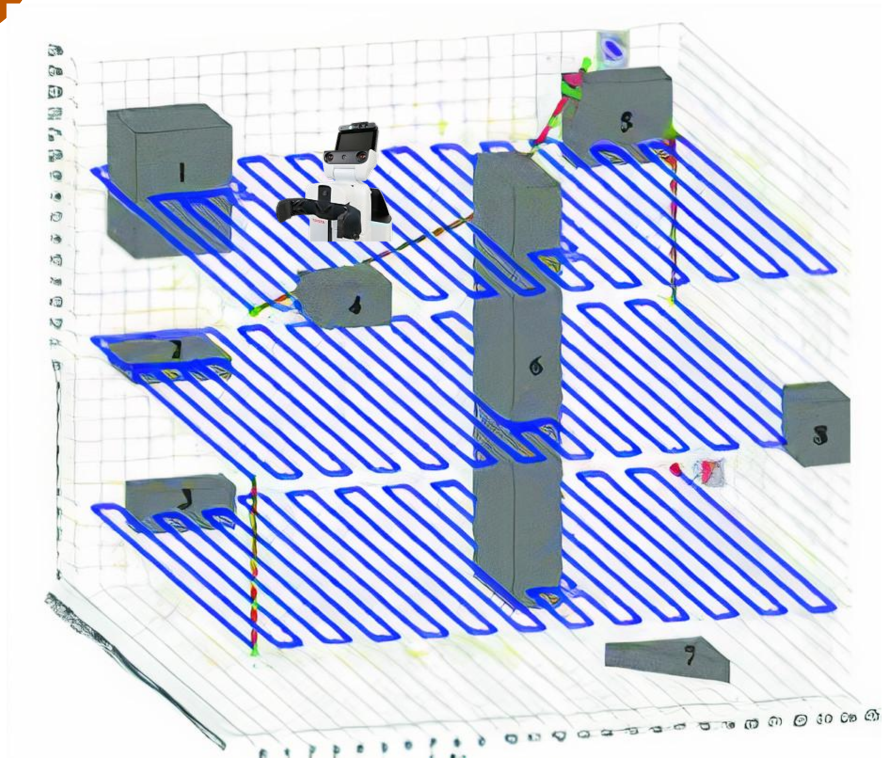
- Ongoing stream of tasks
  - Service robot

- Long term task
  - Cleaning robot
  - Surveillance robot

Motivation

- Ongoing stream of tasks
  - Service robot
- Long term task
  - Cleaning robot
  - Surveillance robot
- Efficiently build up background knowledge
  - Semantic map

Coverage Path Planning



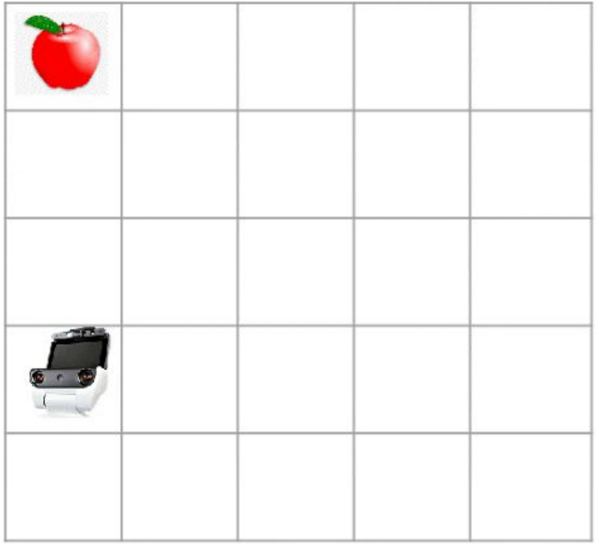
Sun et. al. 2018

Coverage Path Planning (Choset 2001)

- Complete coverage of the building
- Highly wasteful
    - Food delivery robots don't care about restrooms

Efficient Coverage

Detections per Second (**DPS**): Average events detected per second



Continual Area Sweeping

How can we continually patrol an area?

Continual Area Sweeping

How can we continually patrol an area in a **non-uniform** way in order to efficiently use travel time?

Prior Work: ADT-Greedy

- ADT-Greedy (Ahmadi and Stone 2005)
  - Introduces Continual Area Sweeping problem
  - Uses metric tailored towards first-responders (e.g. first-aid robot)

Prior Work: ADT–Greedy

- ADT-Greedy (Ahmadi and Stone 2005)
  - Introduces Continual Area Sweeping problem
  - Uses metric tailored towards first-responders (e.g. first-aid robot)

- Limitations
  - Events assumed to follow binomial distribution
  - Appearance assumed to be linear in time
  - Events never disappear

Prior Work: ADT-Greedy

- ADT-Greedy (Ahmadi and Stone 2005)
    - Introduces Continual Area Sweeping problem
    - Uses metric tailored towards first-responders (e.g. first-aid robot)

- Limitations
    - Events assumed to follow binomial distribution
    - Appearance assumed to be linear in time
    - Events never disappear

- Acceptable Scenario: **Dust Cleaning**

Our Work: DPS-Max

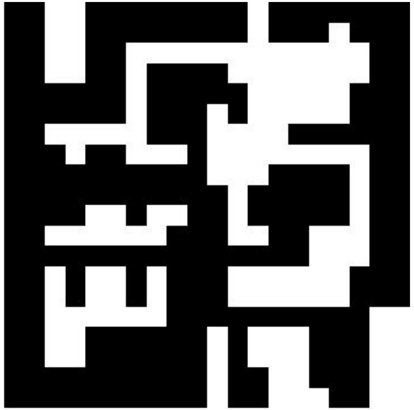
DPS-Max:

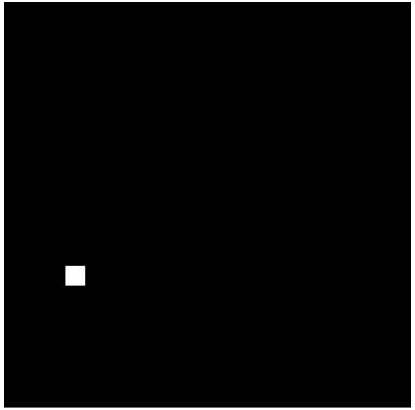
- No prior assumptions
- We provably maximize DPS (Detections per Second) by employing a semi-MDP formulation
- Novel deep R-learning approach to solve problem

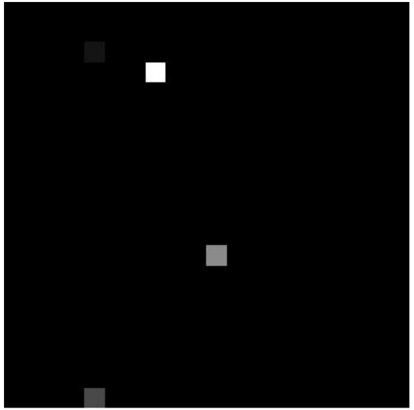
Semi-MDP

State Space:

- 2D Navigational Costmap
- Robot Position
- Events Trace







Semi-MDP

Action Space:

- Any location in the map
    - Motion is deferred to the robot’s path planner

Semi-MDP

Action Space:

- Any location in the map
  - Motion is deferred to the robot's path planner

- Actions take different amounts of time
  - This is what gives us a **Semi-**MDP

Average Reward Setting

Usual discounted reward setting:

$$\mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k R(s_k, a_k, s_{k+1})\right]$$

Average Reward Setting

Usual discounted reward setting:

$$\mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k R(s_k, a_k, s_{k+1})\right]$$

Our goal is to maximize DPS (Detections per Second) – we can't express that in this setting!

Average Reward Setting

Usual discounted reward setting:

$$\mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k R(s_k, a_k, s_{k+1})\right]$$

Average reward setting:

$$\liminf_{n\to\infty} \frac{1}{n} \mathbb{E}\left[\sum_{k=0}^{n-1} R(s_k, a_k, s_{k+1})\right]$$

Reward Construction

Proposition 1. *Take* $\{(s_n, a_n)\}_{n\geq 0} \subset \mathcal{S} \times \mathcal{A}$ *to be a trajectory generated from a policy* π. *Let* $\{\phi_n\}_{n\geq 0} \subset \mathbb{R}$ *a sequence, and* $\{t_n\}_{n\geq 0} \subset \mathbb{R}$ *an increasing sequence denoting the associated environmental time. Construct* R *in the following way:*

$$R(s_0, a_0, s_1) := 0$$

$$R(s_n, a_n, s_{n+1}) := (n+1)\frac{\phi_{n+1}}{t_{n+1}} - n\frac{\phi_n}{t_n}$$

Then $\rho^\pi(s_0) = \liminf_{n\to\infty} \frac{\mathbb{E}\,\phi(s_n)}{t_n}$

Deep R Learning

- Off-policy RL is desired
    - Greater sample efficiency

Deep R Learning

- Off-policy RL is desired
  - Greater sample efficiency

- R Learning (Schwartz 1993)
  - Classical modification to Q Learning for the average reward setting

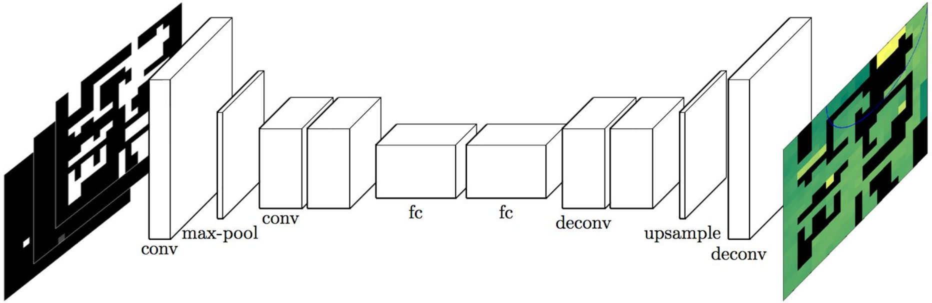
- We want to use deep function approximators
  - Experience Replay
  - Modify Double DQN

Deep R Learning

Algorithm 1 Deep R-Learning

Initialize empty experience replay buffer $\mathcal{D}$.
Initialize network Q with random weights $\boldsymbol{\theta} = \boldsymbol{\theta}^{-}$.
Initialize $\rho = 0$.
for t = 1, . . . , M **do**
    Select an action a_t according to an action selection mechanism like ϵ-greedy.
    Execute a_t and store the resulting transition (s_t, a_t, r_t, s_{t+1}) in $\mathcal{D}$.
    Randomly sample a batch of transitions $\{(s_j, a_j, r_j, s_{j+1})\}$ from $\mathcal{D}$.
    Let $q_{max} = Q\left(s_{j+1}, \text{argmax}_a\, Q\left(s_{j+1}, a; \boldsymbol{\theta}\right); \boldsymbol{\theta}^{-}\right)$.
    Let $y_j = r_j - \rho + q_{max}$.
    Take a gradient descent step on $L\big(y_j, Q\left(s_j, a_j; \boldsymbol{\theta}\right)\big)$.
    Let $\Delta_j = y_j - Q\left(s_j, a_j; \boldsymbol{\theta}\right)$
    Let $\Delta = \text{avg}\{\Delta_j \text{ s.t. } |Q(s_j, a_j) - q_{max}| < \delta\}$
    if Δ is well-defined **then**
        $\rho = \rho + \alpha\Delta$ for learning rate α
    end if
end for

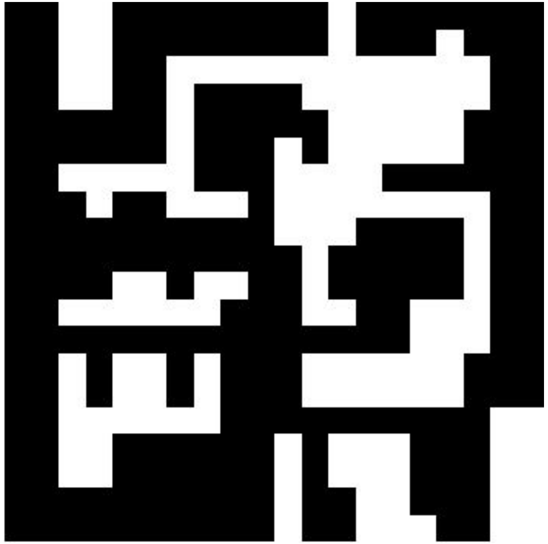
Deep R Learning



- Huge action space
  - # actions: height x width of map
  - Value based methods traditionally struggle in this context

- Architecture circumvents the issue by exploiting the topology of our action space

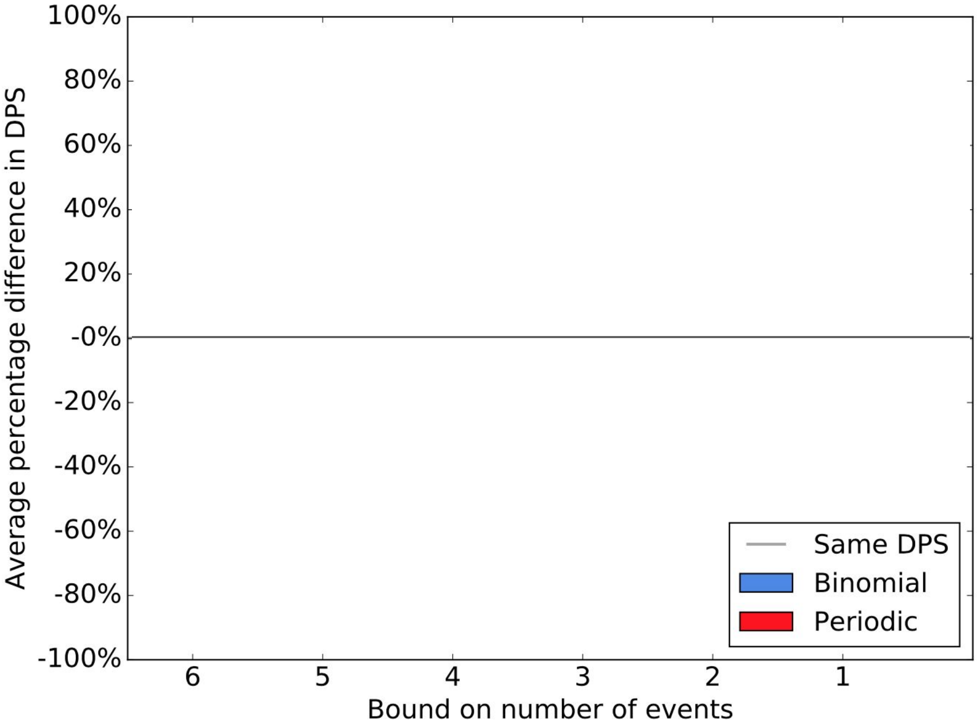
Gridworld Experiments



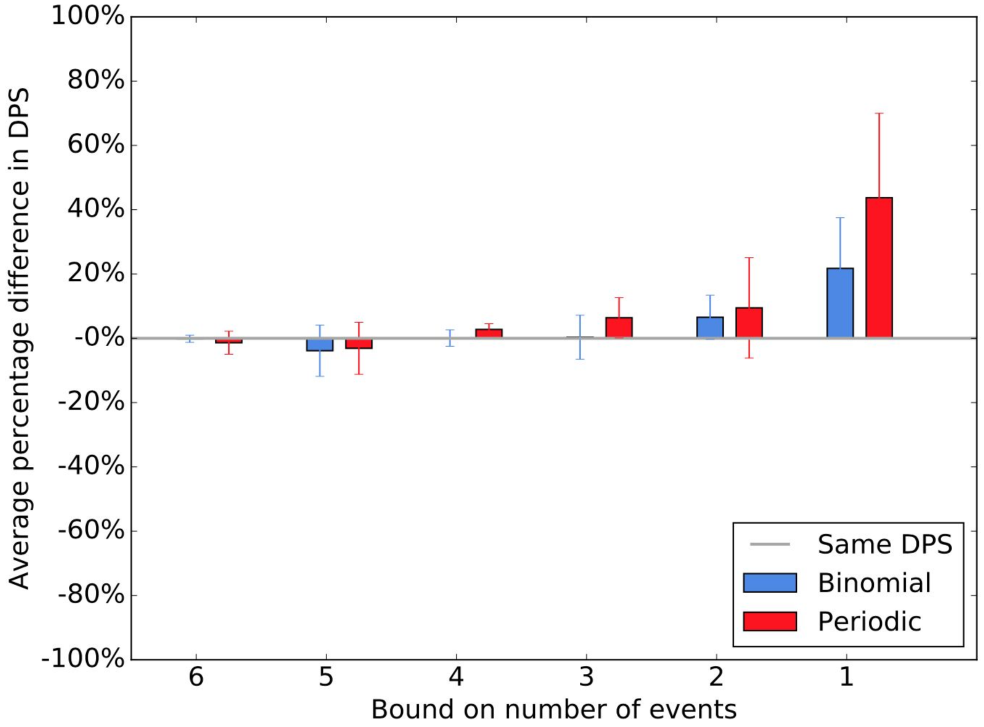
- Initial experiments on 20x20 gridworld to compare with ADT-Greedy

- Events appear in some random cells
  - Binomially (like dust)
  - Periodically (like objects)

Gridworld Experiments



DPS Comparison



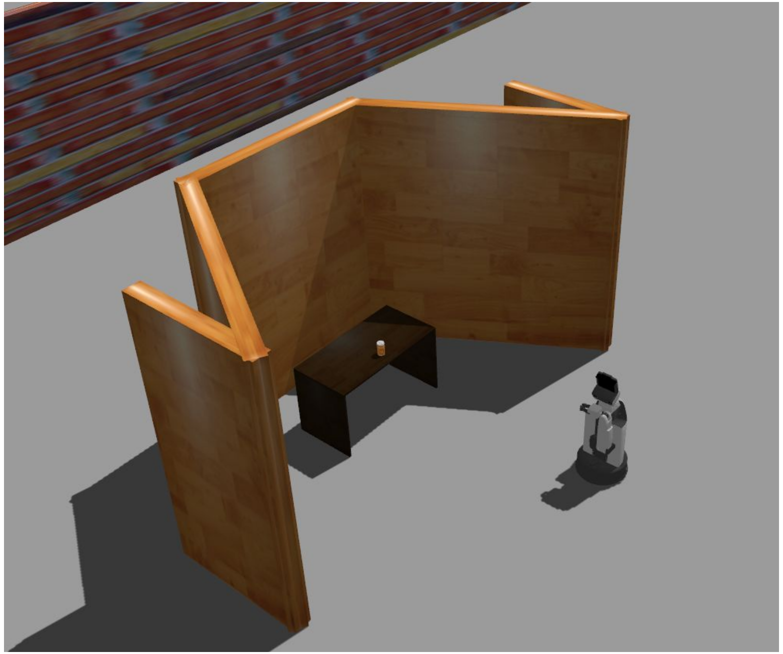
Higher beats baseline

Leveraging Extra Knowledge

- Littering experiment
  - Person moves around and sometimes drops trash
  - Location of person added to robot’s state

- DPS-Max leverages person information and learns that it is correlated with trash appearance
  - Outperforms baseline by utilizing this information

Gazebo Experiments



- Gazebo is a high-fidelity simulator
- Simulated robot in an apartment
- Realistic map size representing 900 m^2 area
  - With 10cm x 10cm grid cells

Service Robot Demonstration

Related Work

- Adversarial Coverage
  - Game-theoretic approaches (Gatti 2008, Basilico et. al. 2012, Bošanský et. al. 2011)
  - Focus is on adversarial two player games

Related Work

- Adversarial Coverage
  - Game-theoretic approaches (Gatti 2008, Basilico et. al. 2012, Bošanský et. al. 2011)
  - Focus is on adversarial two player games

- Coverage with Metrics
  - Known spatial distribution
    - Ergodic coverage, Information Surfing (Ayvali et. al. 2017, Ratto et. al. 2015)

Related Work

- Adversarial Coverage
  - Game-theoretic approaches (Gatti 2008, Basilico et. al. 2012, Bošanský et. al. 2011)
  - Focus is on adversarial two player games

- Coverage with Metrics
  - Known spatial distribution
    - Ergodic coverage, Information Surfing (Ayvali et. al. 2017, Ratto et. al. 2015)
  - Unknown/changing spatial distribution
    - Adaptive ergodic approaches (Mavrommati et. al. 2017)
    - ADT-Greedy (Ahmadi and Stone 2005)

Summary

- Continual area sweeping important for ongoing streams / long term tasks
  - Service robots, cleaning, surveillance, etc.

- Our novel algorithm DPS-Max outperforms and generalizes the baseline

- DPS-Max provably maximizes detections per second

Deep R Learning for Continual Area Sweeping

Rishi Shah*, Yuqian Jiang*, Justin Hart, Peter Stone

Paper: https://arxiv.org/abs/2006.00589