Your Neighbor: RDL2
Robust Decision-Making & Learning Lab

Hanna Kurniawati

hanna.kurniawati@anu.edu.au

http://users.cecs.anu.edu.au/~hannakur/

Australian National University | RESEARCH SCHOOL OF COMPUTER SCIENCE

What we do

Algorithms for robust decision making:

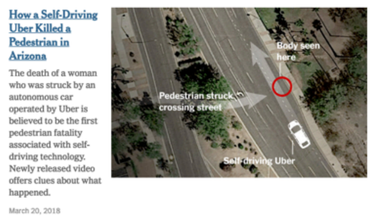
- Large uncertainty
- Complex system dynamics (including multi agents & human intention)

Robust Manipulation Planning

Task: Make me a cup of coffee
1st time seeing the coffee maker

How should robots use tools and manipulate objects to accomplish specific tasks when its understanding about the tools, objects, and its environment are limited to none?

Assuring Autonomous Systems



New York Times (https://www.nytimes.com/), Troy Griggs and Daisuke Wakabayashi

The Daily Dot (https://www.dailydot.com/) Molly McHugh

How to automatically find scenarios that can cause catastrophic failures (before it happens)?

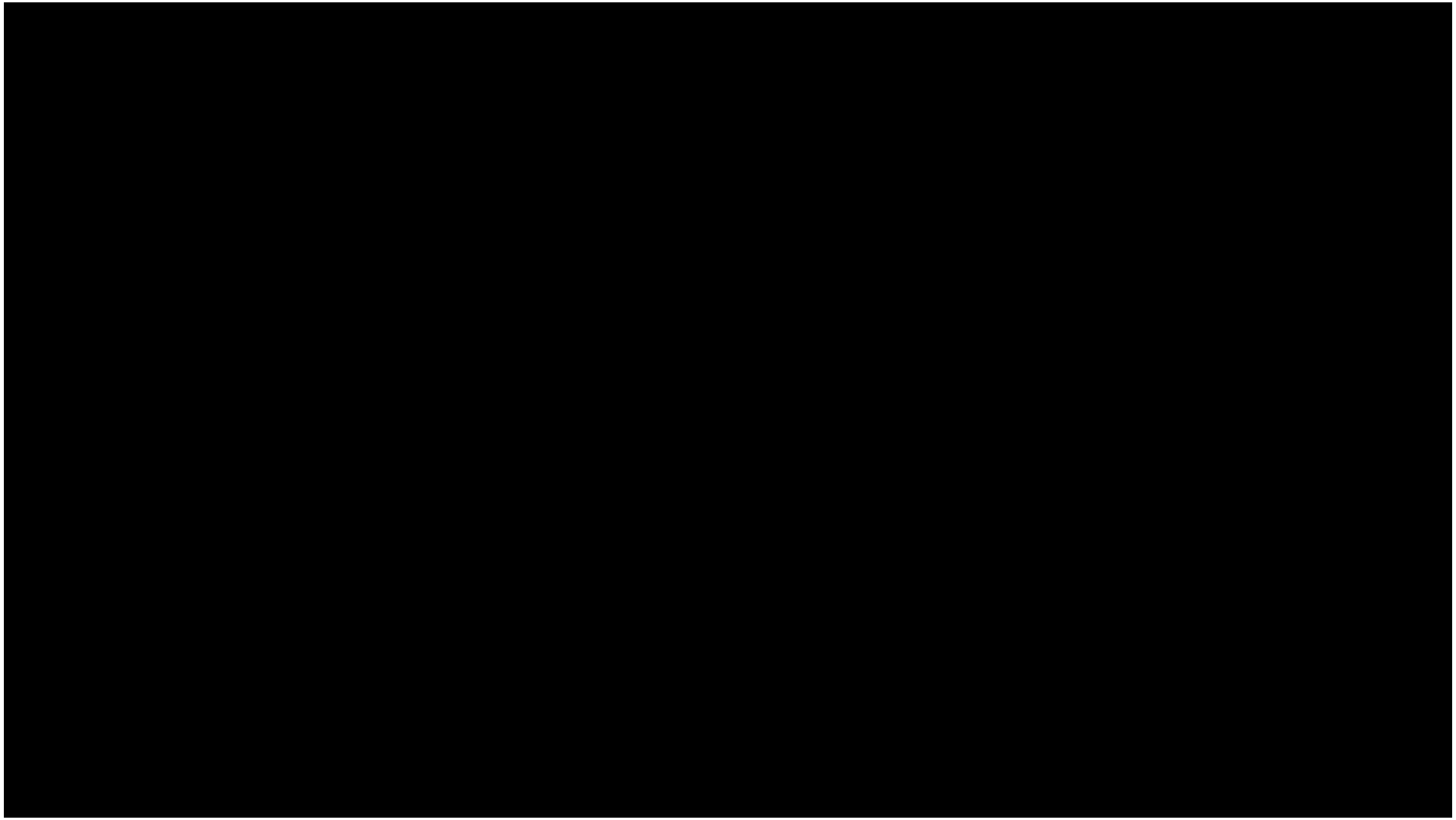
CSIT N323: Robust Decision-making & Learning Lab (RDL2)

The dream: Where it all begins…

Make me a cup of coffee, 1st time seeing the coffee maker

Enable robots to manipulate objects to accomplish specific tasks when its understanding about the system (e.g., objects, available tools, and its environment) are limited to none

Slowly, slowly catchy monkey (hopefully)



Details in ICAPS'19

What's needed?

Usually:

Sense → Plan → Act

1. Use sensors to collect lots of data & learn best model from data
2. Use deterministic planning … he model is faithful
3. Execute the plan

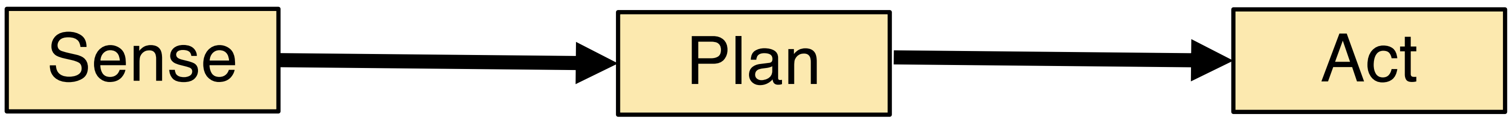
Just an approach to solve the problem

What's the problem?

Excluding h/w design, the problem is:

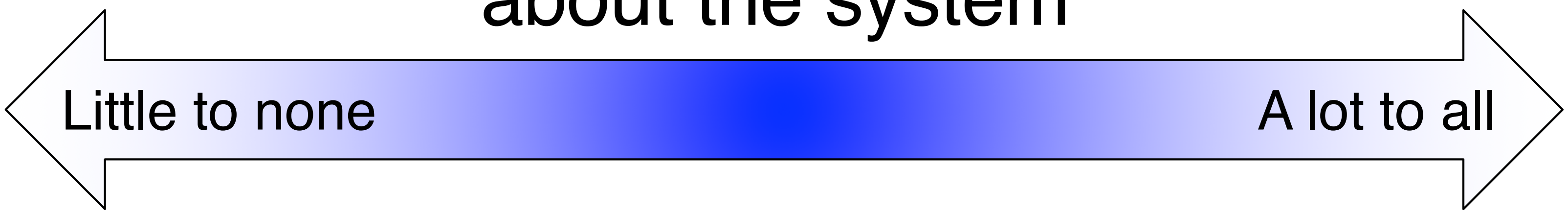
What should robots do now, so that they can get good long term returns (e.g., accomplish a task), despite various types of uncertainty

The common approach…



To a large extent is based on a dichotomy of planning vs learning (control vs SID):

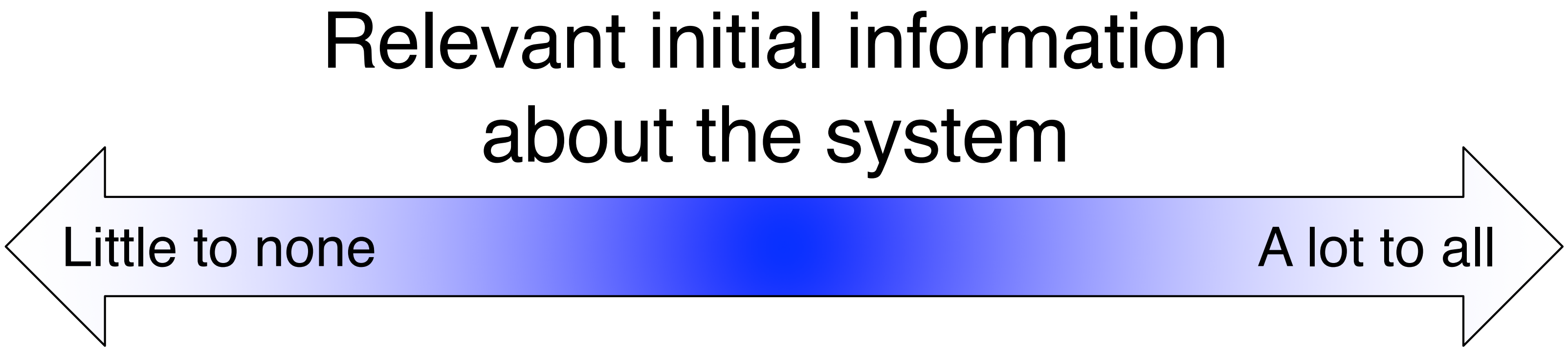
Relevant initial information about the system



Imitation / Reinforcement Learning

Deterministic planning

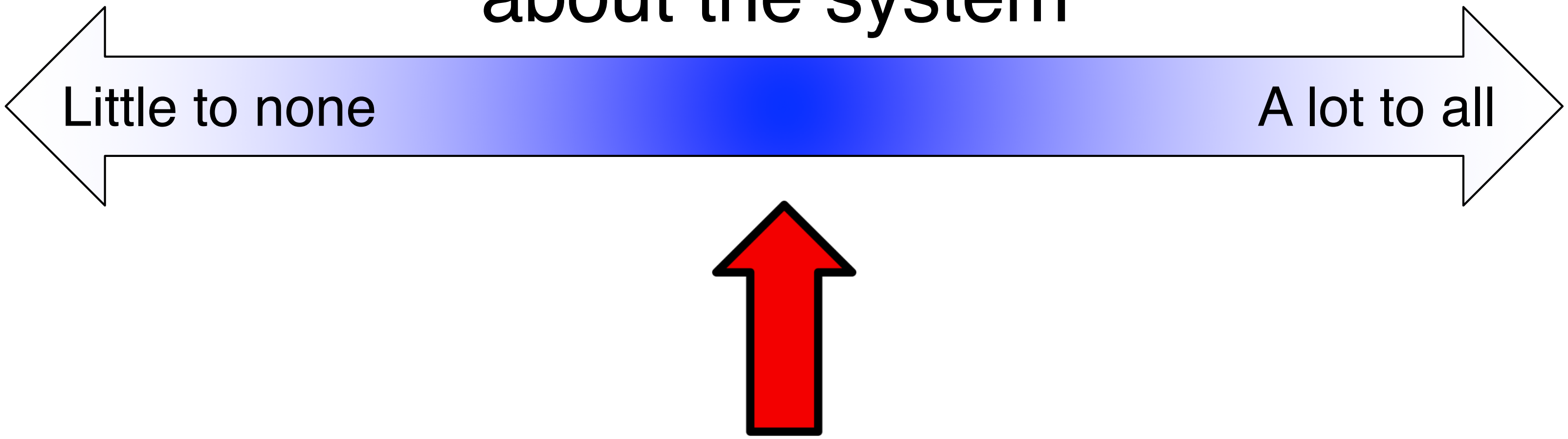
But, the world is usually not that binary…



- Know **something** (not all) about the system **to some extent** (not exact)

I'll try to argue…

Relevant initial information
about the system

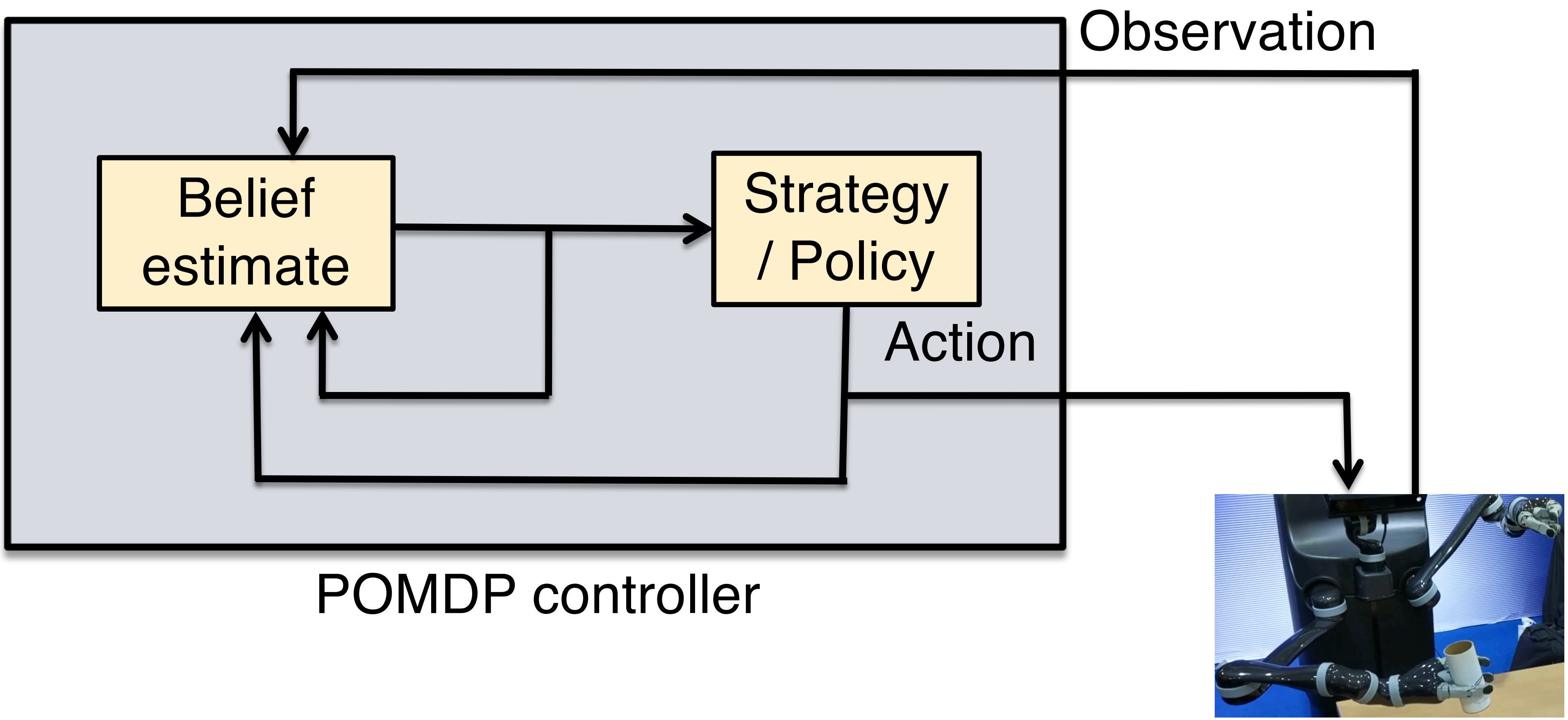


Learning (at least RL) ↔ Planning

The problem

What should robots do now, so that they can get good long term returns (e.g., accomplish a task), despite various types of uncertainty

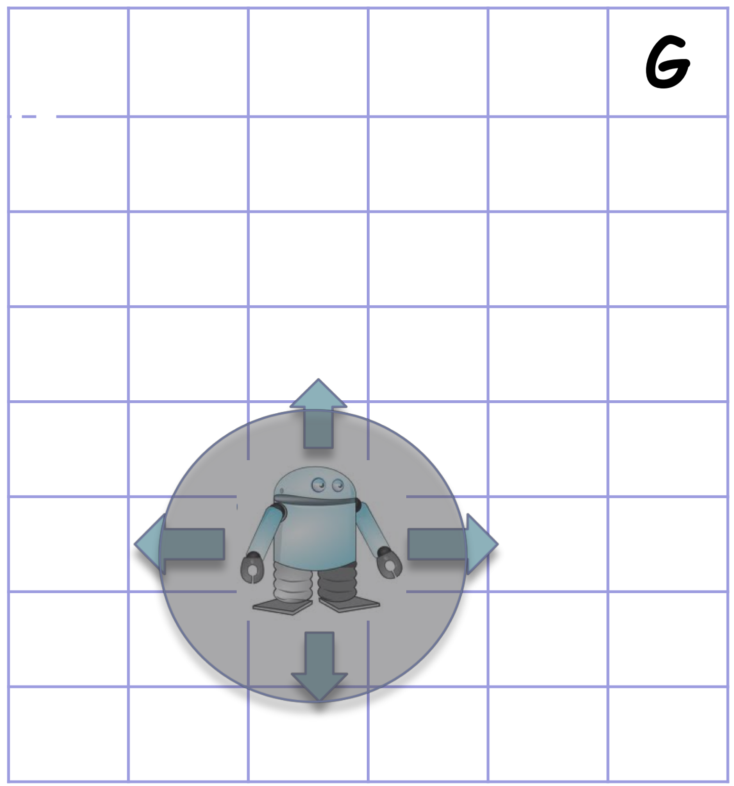
Partially Observable Markov Decision Processes (POMDPs)



Decide the best strategy (often called policy) based on distributions over states (often called beliefs)

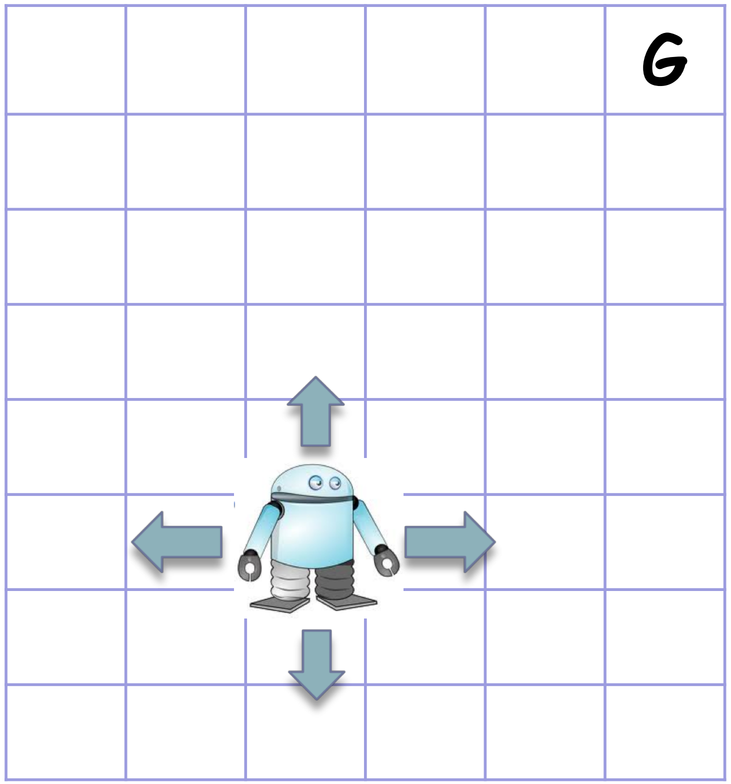
Framing the Problem: POMDP Model

- Main components:
  - State space (S)
  - Action space (A)
  - Observation space (O)



Framing the Problem: POMDP Model

- Main components:
  - State space (S) Not known
  - Action space (A)
  - Observation space (O)
  - Transition function (T)
  - Observation function (Z)
  - Reward function (R)



“Best” policy

- Maps each belief to an action that satisfies the following objective function

$$V^*(b) = \max_{a \in A} \left(\boxed{\sum_{s \in S} R(s,a) b(s)} + \gamma \boxed{\sum_{o \in O} P(o|b,a) V^*(b')} \right)$$

Expected immediate reward

Expected total future reward

b': next belief after the system at belief b performs action a and observes o

γ: discount factor, (0,1)

How to get the POMDP model?

- Spaces are easy, how about the functions?
- We could embed uncertainty about the POMDP model in the POMDP itself

In ~~PO~~MDP

MDP Model

- State space (S_{MDP})
- Action space (A_{MDP})
- Transition function (T_{MDP})
- Reward function (R_{MDP})

Not known

Construct a POMDP

- Where the states are MDP states X parameters of the T_{MDP} & R_{MDP}
- Essentially, partial observability on which MDP model is the right model
- A, T, R follows from the particular MDP model
- O & Z are observations & observation function about which MDP model is correct

But, how to get the POMDP model?

- POMDP is MDP in the belief space
- So, the same concept applies
  - Off course, much more complicated

In ~~PO~~MDP

MDP Model

- State space (S_{MDP})
- Action space (A_{MDP})
- Transition function (T_{MDP})
- Reward function (R_{MDP})

Not known

Reinforcement learning (RL)

Bayesian RL

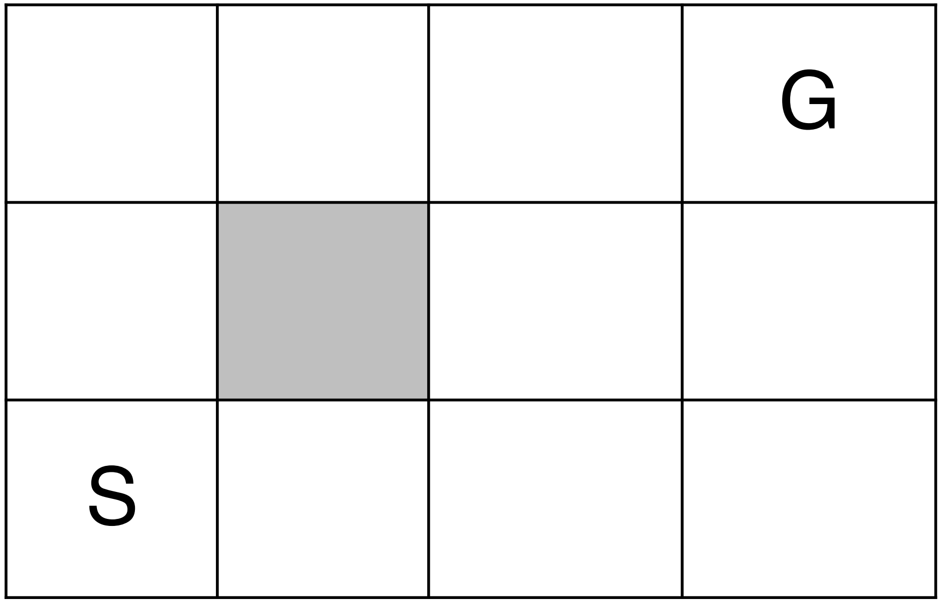
Construct a POMDP

- Where the states are MDP states X parameters of the T_{MDP} & R_{MDP}
- Essentially, partial observability on which MDP model is the right model
- A, T, R follows from the particular MDP model
- O & Z are observations & observation function about which MDP model is correct

So, everything is planning…

Just need to solve that huge POMDP problem

Reality Check



Computationally intractable [Papadimitriou & Tsikilis'87, Madani, et.al.'99].

Not all gloom & doom

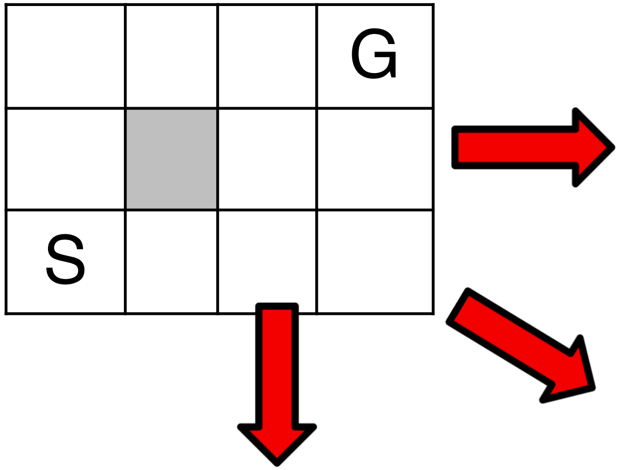
- Close to optimal solution is often good enough
  → Sampling
- There's many useful "structures" even in seemingly unstructured problems
  → Perhaps not environmental structures, but uncertainty structures (e.g., correlation, dependencies / independencies, etc.)
  → Inherent properties of the problems (e.g., continuity of motion in robotics)
  → Significantly reduce sampling domain, converge to good solutions faster

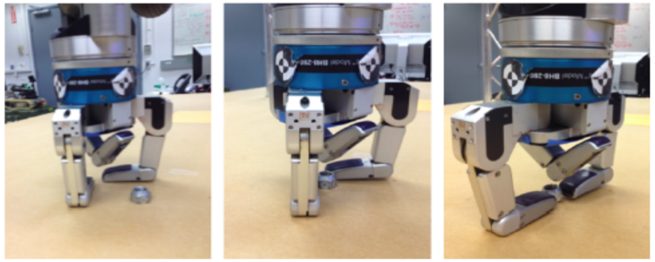
Scaling up POMDP solving capability

- Large state space [Kurniawati, et.al. (RSS'08)]
- Large observation space [Kurniawati, et.al. (RSS'11, Auro'12 invited)]
- Long planning horizon [Kurniawati, et.al. (ISRR'09, IJRR'11 invited)]
- Model may change [Kurniawati & Patrikalakis (WAFR'12), Kurniawati & Yadav (ISRR'13)]
- Large action space [Seiler, et.al. (ICRA'15, best paper award finalist), Wang, et.al. (ICAPS'18)]
- Complex transition functions [Hoerger, et.al. (WAFR'16), Hoerger et.al. (submitted to RSS'19)]

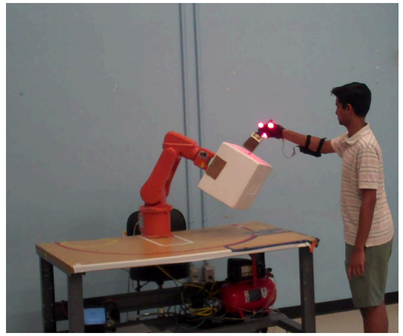
Implementation: http://rdl.cecs.anu.edu.au/software

Some Progress

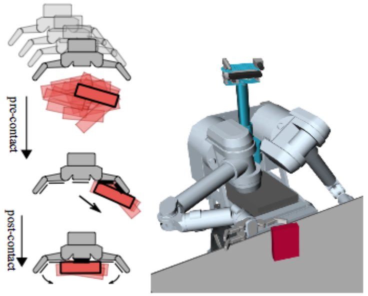




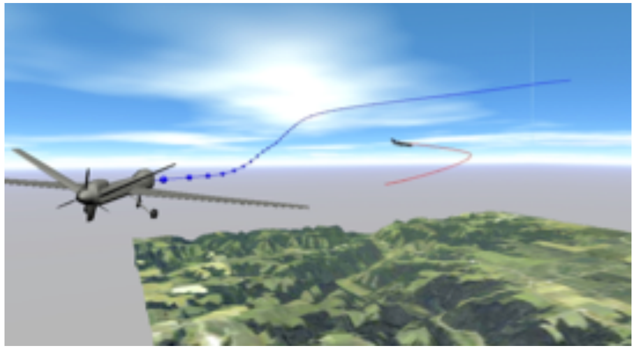
Horowitz &
Burdick (Joel Burdick's group)



Nikolaidis, et.al. (Julie Shah's group)



Koval, et.al. (Sid Srinivasan's group)



Temizer, et.al. (Lincoln Lab TR'09)
Improve safety of TCAS by 20X

Bandyopadhyay, et.al. (early work leading to nuTonomy)

Wang, et.al. (ICAPS'15)
Learn interaction model of bees with reduced data

Hopefully, satisfied users of our POMDP solvers

But …

- Still not enough to consider uncertain model in general
  - To model initially unknown transition of a simple grid navigation where
    - A robot can move in 8 wind direction
    - Assuming transition is the same everywhere
    - The probability value is discretized into 5 bins
  - we'll need to multiply the number of states by ~390K
    - Also observations

Machine Learning Solutions

- Computing a good policy is viewed as the problem of finding a **mapping** that fits the **data**
  - Mapping from which space to which space?
    - Model-based
    - Model-free
  - Where does the data come from?
    - Someone / something provides examples
    - Trying on a simulator / the system
  - Use optimization (e.g., policy search) to find a mapping that “best” fits the data
  - More recently, frame as a deep learning problem

Embedding & Solving MDP w/o T & R in Neural Net

$$V^*(s) = \max_{a \in A} \left(R(s,a) + \gamma \sum_{s' \in S} T(s'|s,a) V^*(s') \right)$$

Convolution, T as the kernel (learned weight)

Sum, R as CNN (learn mapping from images to a map of real number)

max-pool

Iteration: RNN, 1 iteration = 1 layer
Train end-to-end, imitation learning

VIN (Tamar, et.al. NIPS'16)

POMDP?

- Propagate belief (Bayes filter)
  - Jonchowski, et.al.: Histogram (NIPS'16), particle (RSS'18)
- Planning:
  - Straightforward extension of MDP Value Iteration use QMDP planner

QMDP-Net (Karkus, et.al. NIPS'17)

- Modify the planning architecture to embed better POMDP planner:? [Student Project]

So, everything is learning…

Just need to get those data somehow

Reducing data requirements

- Turns out, non learning-techniques (including planning) helps …
  - POMDP planning [ICAPS'15, best student paper]
  - Computer graphics + sampling [ICRA'19]
  - Local structures [submitted to CoRL'19]

POMDP planning to accentuate data

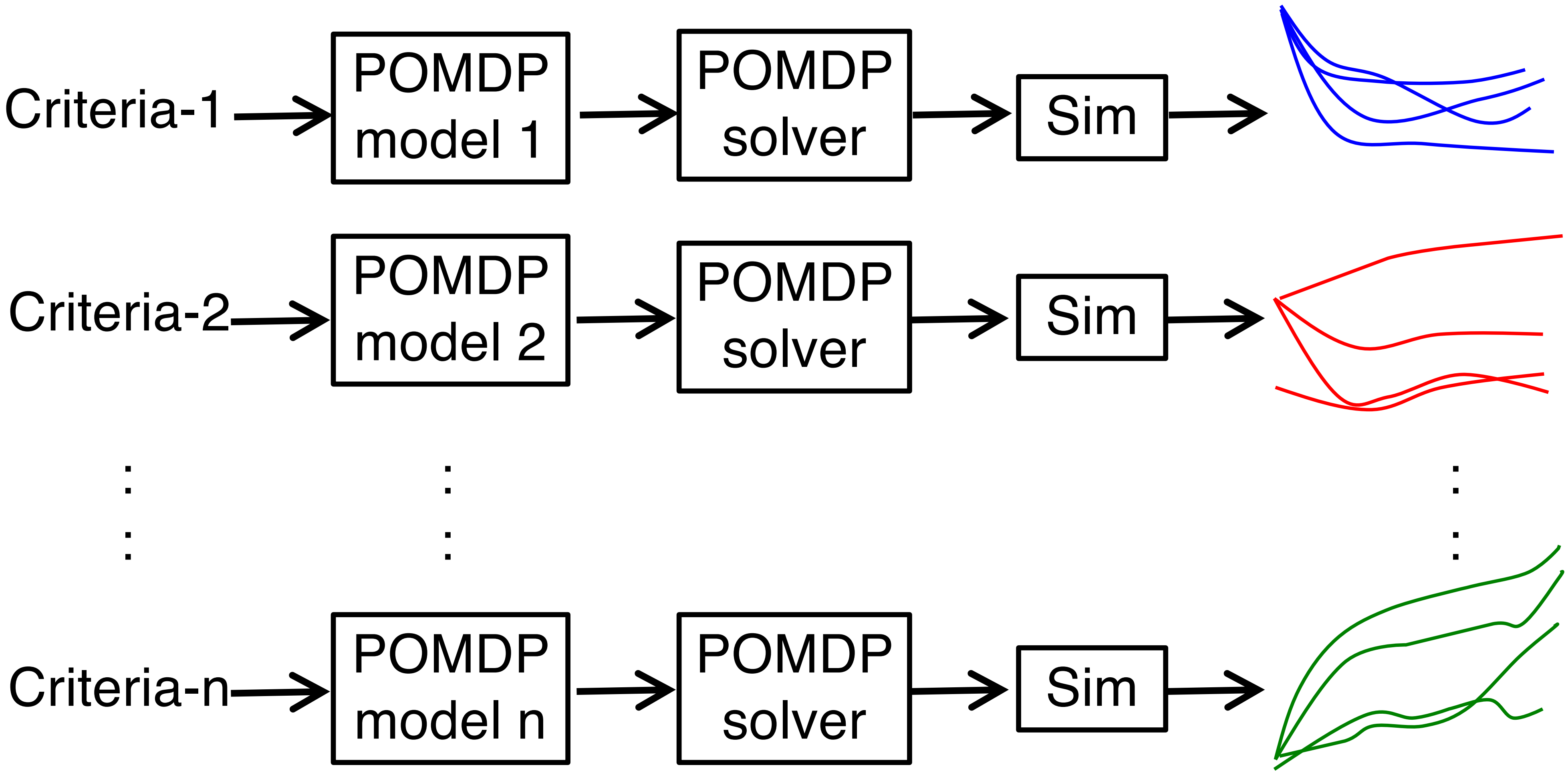
How do they avoid mid-air collision?

How do bees avoid collision?

- Current view:
  - Animal behavior optimizes certain criteria
  - The question is what criteria is being optimized

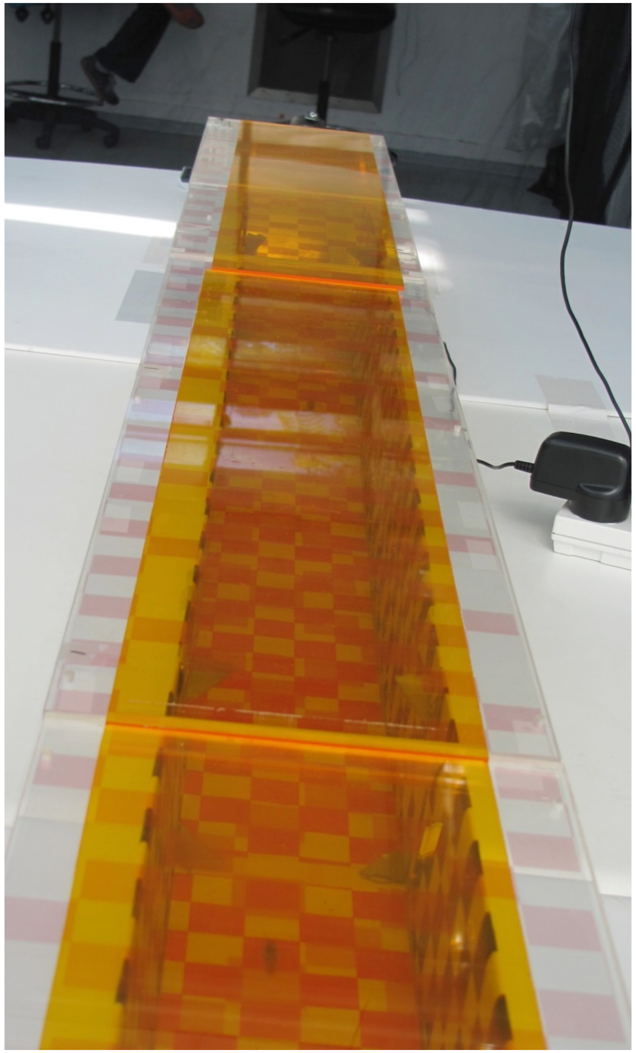
Wang, et.al. ICAPS' 15 (Outstanding Student Paper Award)

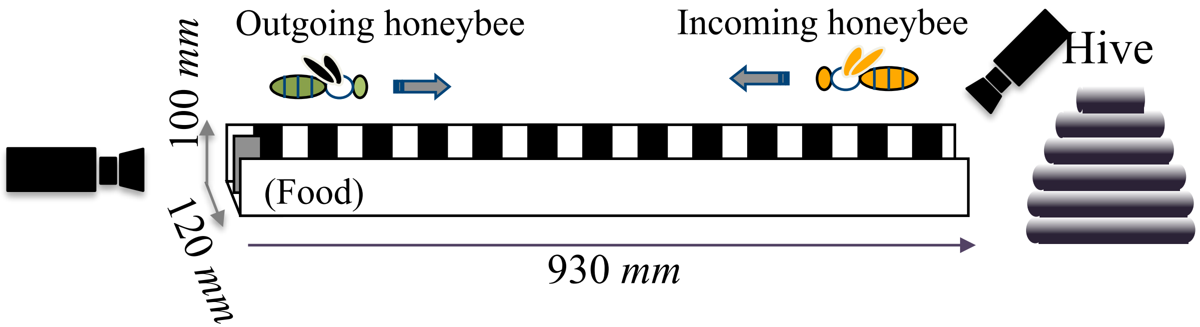
A Hypothesis Ranking System



Rank the criteria based on how similar the simulated trajectory is to the (limited) experimental data

A Hypothesis Ranking System
(from 100 real data)





Correctly rank phototaxis behavior + horizontal centering at the top of the bees' behaviour

Reducing Data Requirements

- Turns out, non learning-techniques (including planning) helps …
  ✓POMDP planning [ICAPS'15, best student paper]
  - Computer graphics + sampling [ICRA'19]
  - Local structures [submitted to CoRL'19]

Robot Object Fetching

- Household objects usually have logos

☺ Trademarks database contains
  - Lots of logo images (designer images)
  - Classification based on brand and type (e.g., food)

☹ Images from camera on robots are of much lower quality than designer images

- Randomization-based Data Synthesizer for Logos (RDSL): Use computer graphics rendering + domain randomization

Song & Kurniawati. To appear in ICRA'19

Randomization-based Data Synthesizer for Logos (RDSL)

SSD Mobile Net (an off-the-shelf CNN logo detector) trained with **only** the synthetic images RDSL generates

Reducing Data Requirements

- Turns out, non learning-techniques (including planning) helps …
  ✓POMDP planning [ICAPS’15, best student paper]
  ✓Computer graphics + sampling [ICRA’19]
  - Local structures [submitted to CoRL’19]

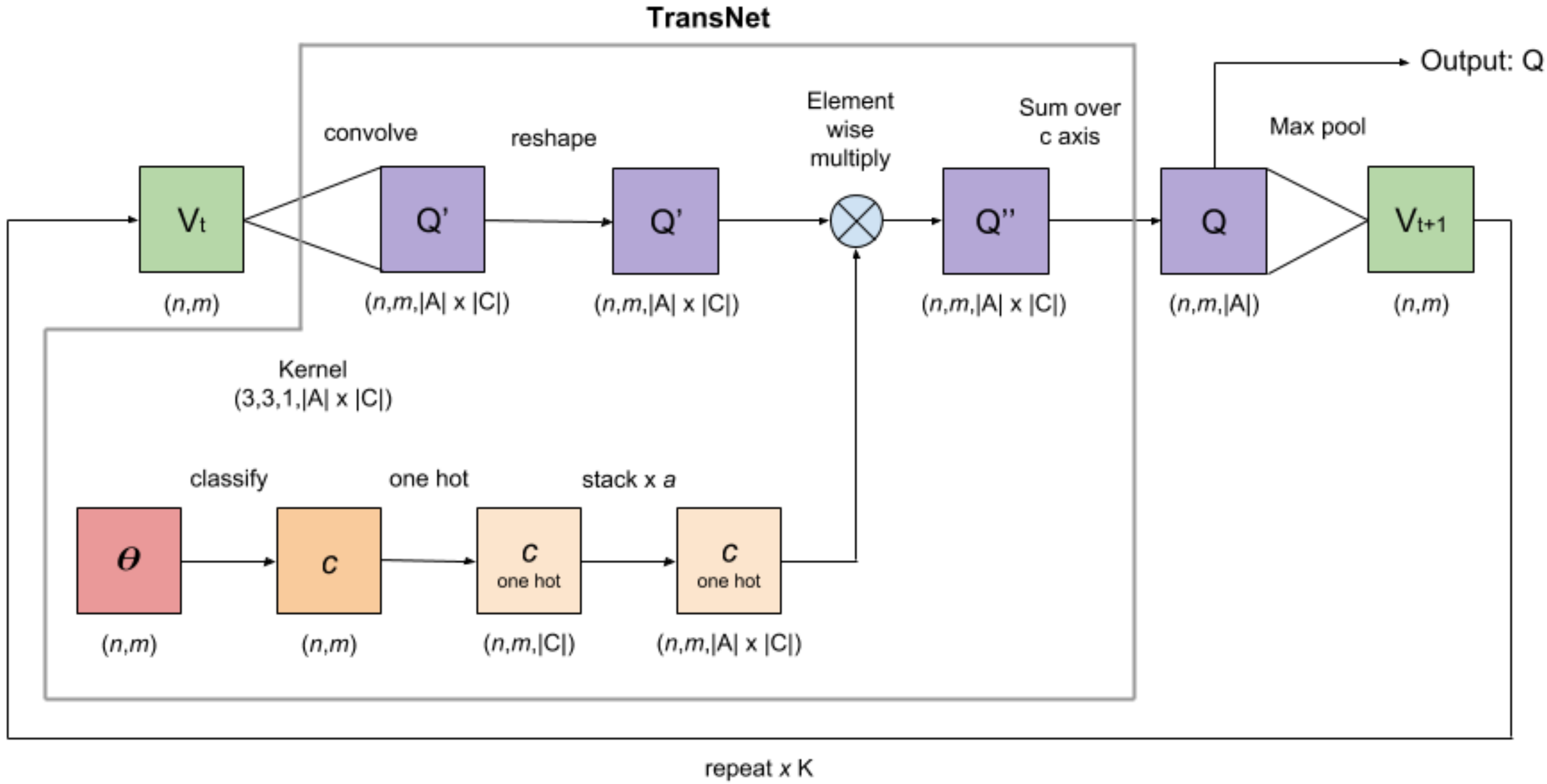
Caveat in VIN, QMDP-Net

$$V^*(s) = \max_{a \in A}\left(R(s,a) + \gamma \underbrace{\sum_{s' \in S} T(s'|s,a)V^*(s')}_{\text{Convolution, T as the kernel (learned weight)}}\right)$$

- That T is assumed to be the independent of states…
  - Makes the #learned weight small
  - Reduce data requirement

TransNet

- T depends on local geometry (and action)



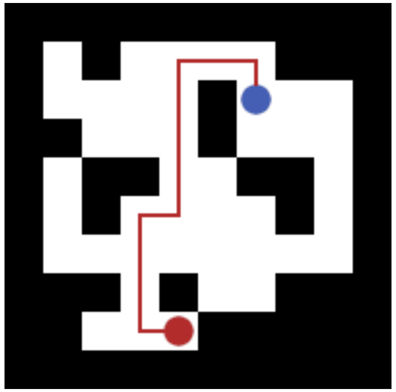
Collins & Kurniawati. Submitted to CoRL'19

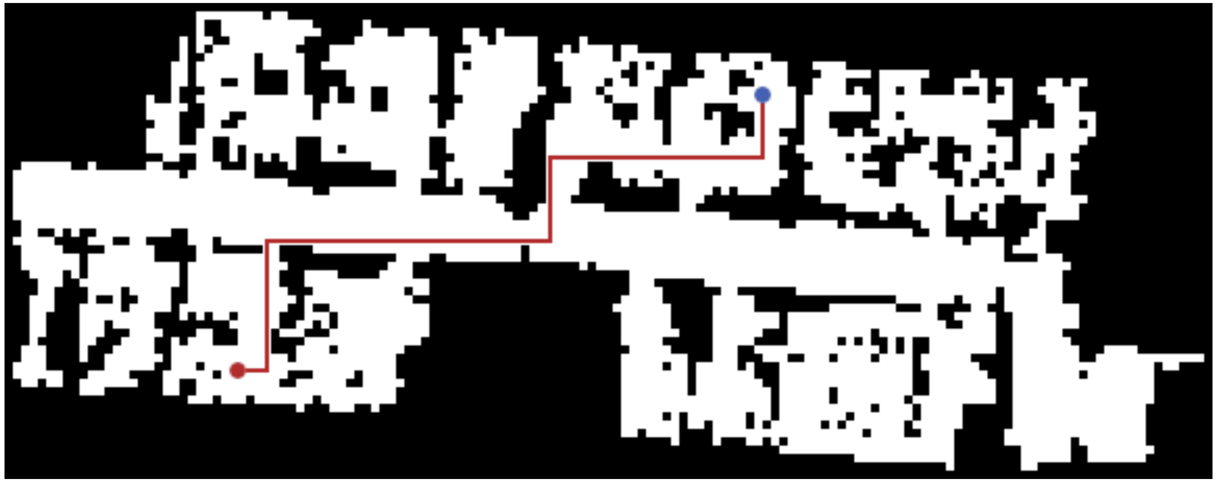
Results

- 10X10 navigation in a grid world
- Input: Image of the environment (contain obstacles) & init. belief
- Obstacles are generated uniformly at random
- Train until convergence

#Trajectories	Agent	Success	Traj Length	Collision
2,000	Base	0.704	21.5	0.320
	TransNet	0.982	15.3	0.112
10,000	Base	0.950	15.1	0.139
	TransNet	0.998	14.1	0.1
50,000	Base	0.972	16.2	0.079
	TransNet	0.992	15.4	0.068

Results





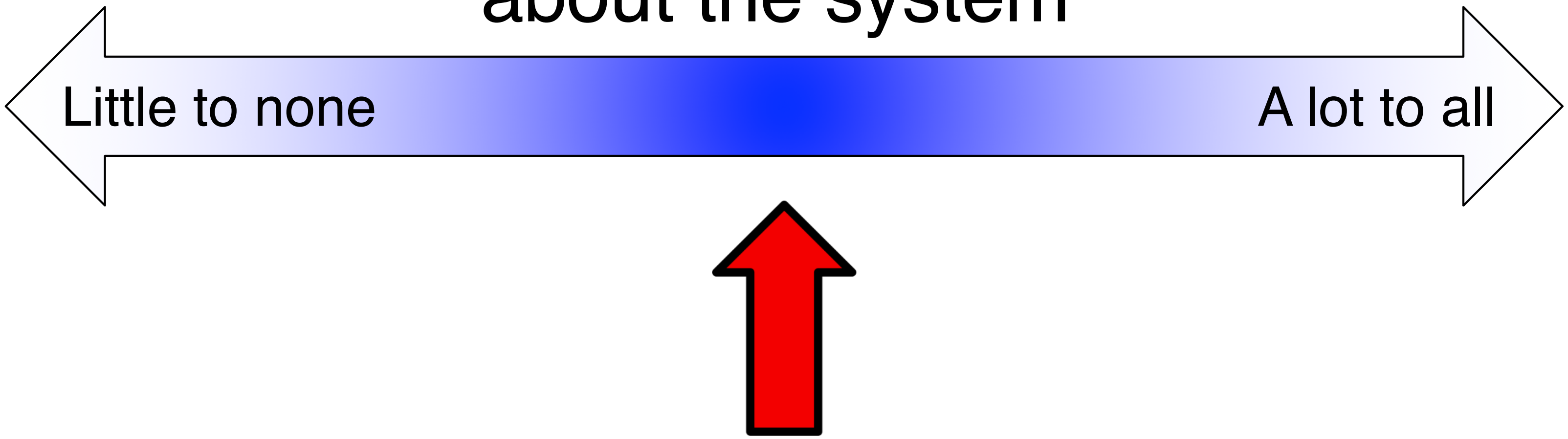
Domain	Agent	Success	Traj Length	Collision
Intel Labs 101x99 D	Base	0.400	100.0	0.066
	TransNet	0.960	94.3	0.012
Building 079 145x57 D	Base	0.560	70.8	0.225
	TransNet	0.780	65.2	0.048
Hospital 193x104 D	Base	0.140	85.1	0.286
	TransNet	0.840	91.2	0.039

Reducing Data Requirements

- Turns out, non learning-techniques (including planning) helps …
  - ✓POMDP planning [ICAPS'15, best student paper]
  - ✓Computer graphics + sampling [ICRA'19]
  - ✓Local structures [submitted to CoRL'19]

So, it seems…

Relevant initial information
about the system



Learning (at least RL) ↔ Planning

Perhaps…

- The Problem:

Robust Autonomy:
What should robots do now, so as to accomplish specific tasks well, despite various types of uncertainty

- Framework: MDP, RL (MDP w. missing component), POMDP, …

- Solution:

  - Planning, learning, & combination
  - The problem is hard, better take anything that can help solve

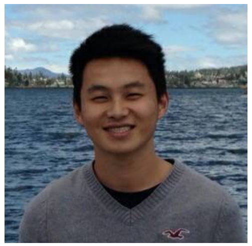
Acknowledgement

Team:





Now at Clarivate


Sponsors:

Australian National University | RESEARCH SCHOOL OF COMPUTER SCIENCE

DATA 61 CSIRO

Australian Government
Department of Defence
Science and Technology

woodside

Lloyd's Register Foundation

ORACLE

UNIVERSITY of York

KINOVA ROBOTICS

Office of Naval Research ONR Science & Technology

What we do

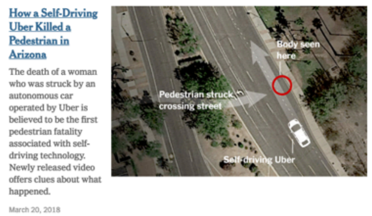
Scaling up algorithms for robust autonomy:
- Large uncertainty
- Complex system dynamics (including multi agents & human intention)

Robust Manipulation Planning

Task: Make me a cup of coffee
1st time seeing the coffee maker

How should robots use tools and manipulate objects to accomplish specific tasks when its understanding about the tools, objects, and its environment are limited to none?

Assuring Autonomous Systems


How a Self-Driving Uber Killed a Pedestrian in Arizona

The death of a woman who was struck by an autonomous car operated by Uber is believed to be the first pedestrian fatality associated with self-driving technology. Newly released video offers clues about what happened.

March 20, 2018

New York Times (https://www.nytimes.com/), Troy Griggs and Daisuke Wakabayashi


A robot vacuum tried to eat its sleeping owner's head

The fire department was called in to rescue her.

The Daily Dot (https://www.dailydot.com/) Molly McHugh


The 'suicidal robot' that drowned in a fountain didn't kill itself after all

Technical error as algorithm fails to detect uneven surface

How to automatically find scenarios that can cause catastrophic failures (before it happens)?

CSIT N323: Robust Decision-making & Learning Lab (RDL2)

Thank you

Q&A