Sequential Decision-making in the Partially Observable World

Hanna Kurniawati

hanna.kurniawati@anu.edu.au

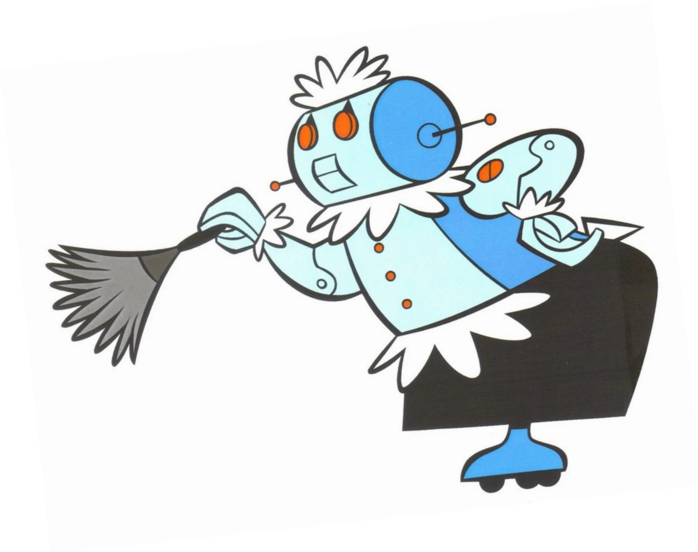
https://comp.anu.edu.au/people/hanna-kurniawati/

Australian National University | School of Computing

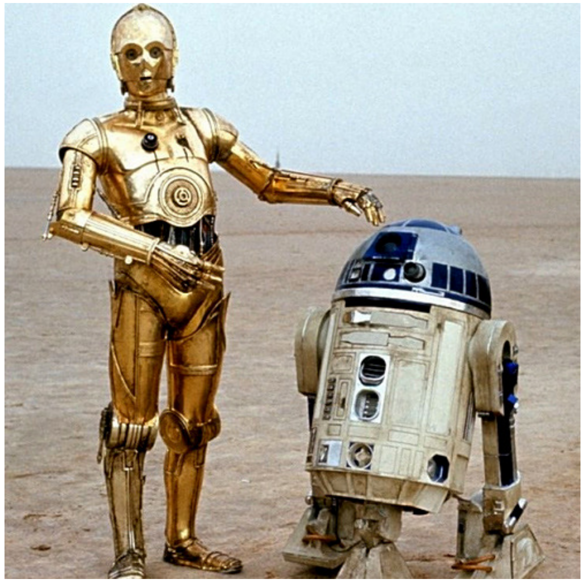
SMARTSAT Cooperative Research Centre

The dream…

Have robots do all the work we(I) don't really like



The housekeeper robot
Rosie (originally Rosey)



The chatty C3-PO and
the capable R2-D2

The creative carer robot
Doraemon

Pictures taken from their respective fandom pages

Manipulation

Dex-Net, CITRIS, UC Berkeley, since 2017
(https://www.youtube.com/watch?v=TwnO0aEFTrk)

https://www.pinterest.com.au/pin/205054589256148973/?d=t&mt=login

Navigation

Waymo fully autonomous car
In 2019, start operating without safety drivers in some public roads
(https://www.youtube.com/watch?v=TwnO0aEFTrk)

https://metro.tempo.co/read/1207214/semrawut-di-tanah-abang-parkir-liar-dan-pedagang-di-trotoar/full&view=ok

Flying

Alphabet Wing drone delivery, since 2019
(https://www.youtube.com/watch?v=TwnO0aEFTrk)

Photo taken from: https://www.alamy.com/stock-photo/propeller-helicopter.html

What’s the Difficulty?

Sequential + Uncertainty

What’s the Difficulty?



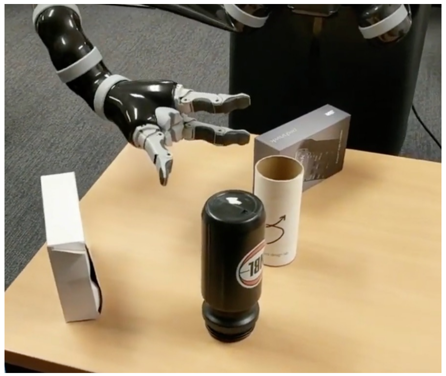
Same Problem: What strategy should the robot take now, to achieve good long-term outcomes, despite various types of uncertainty?

- General purpose robots require general purpose decision-making capabilities

- Even robots that are designed for specific tasks need to handle various types of uncertainty at various levels of decision-making

→ Require general purpose framework & method for handling uncertainty in decision-making

Partially Observable Markov Decision Processes (POMDPs)

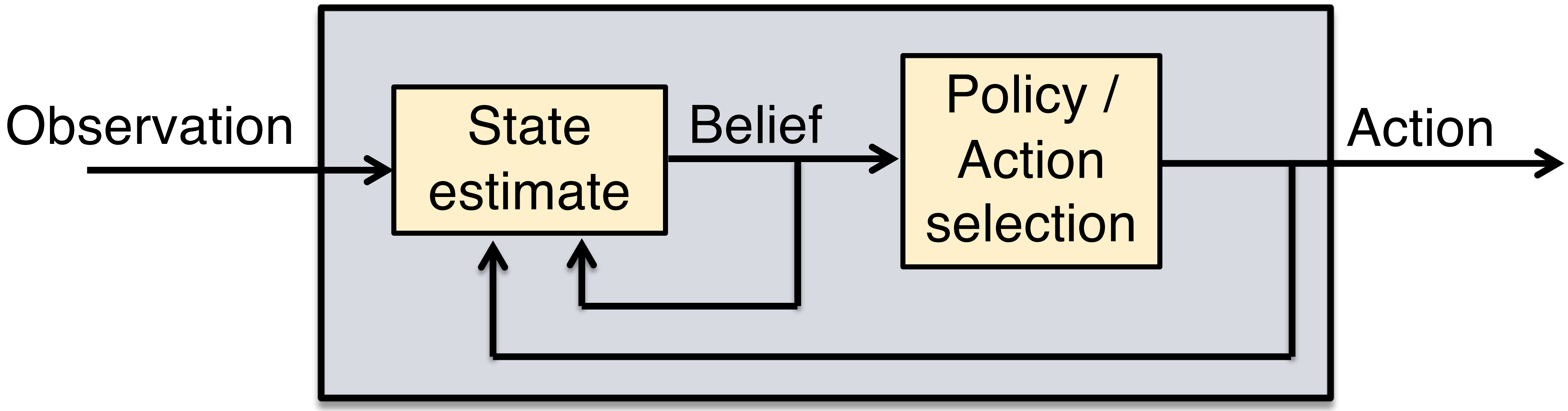


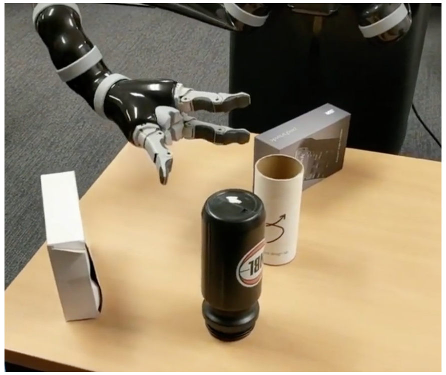
Z(s', a, o): Observation function P(o | s', a) ; o ∈ O
s': States, a: action, o: Observation

T(s, a, s'): Transition function P(s' | s, a)
s, s': States, a: action

R: Reward function

Partially Observable Markov Decision Processes (POMDPs)





Belief: Distribution over states
Solving a POMDP: Computing the best policy –maps beliefs to the best action

Best policy

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

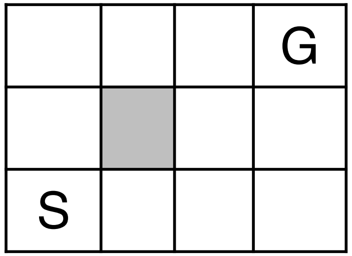
$$V^*(b) = \max_{a \in A} \left(\underbrace{\sum_{s \in S} R(s,a) b(s)}_{\text{Expected immediate reward}} + \gamma \underbrace{\sum_{o \in O} P(o|b,a) V^*(b')}_{\text{Expected total future reward}} \right)$$

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

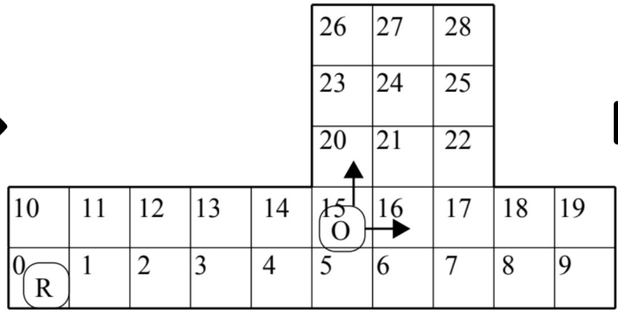
γ: discount factor (0,1)

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

Not All Gloom & Doom …

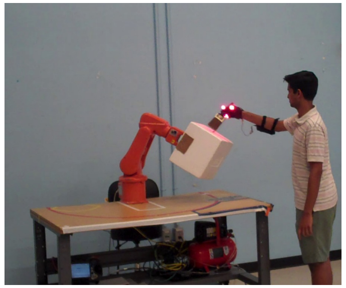


Before’03: ISI = 12, days

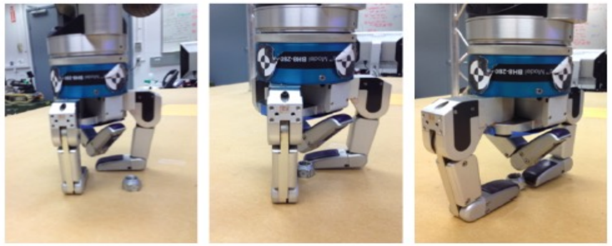


PBVI (Pineau’03): Tag ISI = 870, 50h

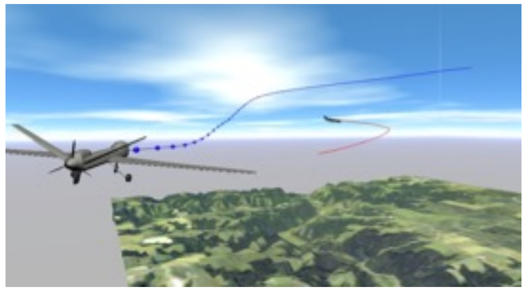
SARSOP (Kurniawati, et.al.’08,
RSS’21 Test of Time Award)
Tag in ~ 6sec
Up to ISI ~ 100K, 2h



Nikolaidis, et.al.



Horowitz & Burdick



Temizer, et.al.
Study of next-gen TCAS (improve safety of TCAS by 20X)

Bandyopadhyay, et.al.

Wang, et.al.
Learn interaction model of bees with reduced data
(ICAPS’15 best student paper)

What's the Trick?

- 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

The Problems & Some of Our Solutions

Sampling-based & Learning-based

- **Large state space**
  Kurniawati, et.al. (RSS'08), RSS'21 Test of Time Award
- **Large action space** – up to 12-D cont. action space
  Seiler, et.al. (ICRA'15, best paper award finalist), Wang, et.al. (ICAPS'18), Hoerger, et.al. (WAFR'22, IJRR'23), Hoerger, et.al. (AAAI'24 oral, to appear)
- **Large observation space**
  Kurniawati, et.al. (RSS'11, Auro'12), Hoerger & Kurniawati (ICRA'21)
- **Dynamically changing model**
  Kurniawati & Patrikalakis (WAFR'12), Kurniawati & Yadav (ISRR'13), Chen & Kurniawati (NeurIPS'23)
- **Long planning horizon**
  Kurniawati, et.al. (ISRR'09, IJRR'11), Liang & Kurniawati (IROS'23), Kim, et.al. (NeurIPS'23)
- **Complex dynamics**
  Hoerger, et.al. (WAFR'16), Hoerger et.al. (ISRR'19, IJRR'22)
- **When the POMDP model is not available**
  Collins & Kurniawati (WAFR'20, IJRR'22)

Of course, we're not the only one working in this domain

Summary and Insights of the Advances

H. Kurniawati. Partially Observable Markov Decision Processes and Robotics. *Annual Review of Control, Robotics, and Autonomous Systems.* Vol. 5. No. 1. 2022.

The Problems & Some of Our Solutions

Sampling-based & Learning-based

- **Large state space**
  Kurniawati, et.al. (RSS'08), RSS'21 Test of Time Award
- **Large action space** – up to 12-D cont. action space
  Seiler, et.al. (ICRA'15, best paper award finalist), Wang, et.al. (ICAPS'18), Hoerger, et.al. (WAFR'22, IJRR'23), Hoerger, et.al. (AAAI'24 oral, to appear)
- **Large observation space**
  Kurniawati, et.al. (RSS'11, Auro'12), Hoerger & Kurniawati (ICRA'21)
- **Dynamically changing model**
  Kurniawati & Patrikalakis (WAFR'12), Kurniawati & Yadav (ISRR'13), Chen & Kurniawati (NeurIPS'23)
- **Long planning horizon**
  Kurniawati, et.al. (ISRR'09, IJRR'11), Liang & Kurniawati (IROS'23), Kim, et.al. (NeurIPS'23)
- **Complex dynamics**
  Hoerger, et.al. (WAFR'16), Hoerger et.al. (ISRR'19, IJRR'22)
- **When the POMDP model is not available**
  Collins & Kurniawati (WAFR'20, IJRR'22)

Similar Idea: Primitive Computation

- Primitive computation: Functions that are called many times by the solver
  - Improving primitive components → substantial improvement in overall solving capability
  - Holds for non-learning based, learning based, and their combinations

- **Complex dynamics:** Hoerger et.al. (ISRR'19, IJRR'22) – Sampling-based
- **When the POMDP model is not available:** Collins & Kurniawati (WAFR'20, IJRR'22) – End-to-end learning

POMDPs with complex dynamics

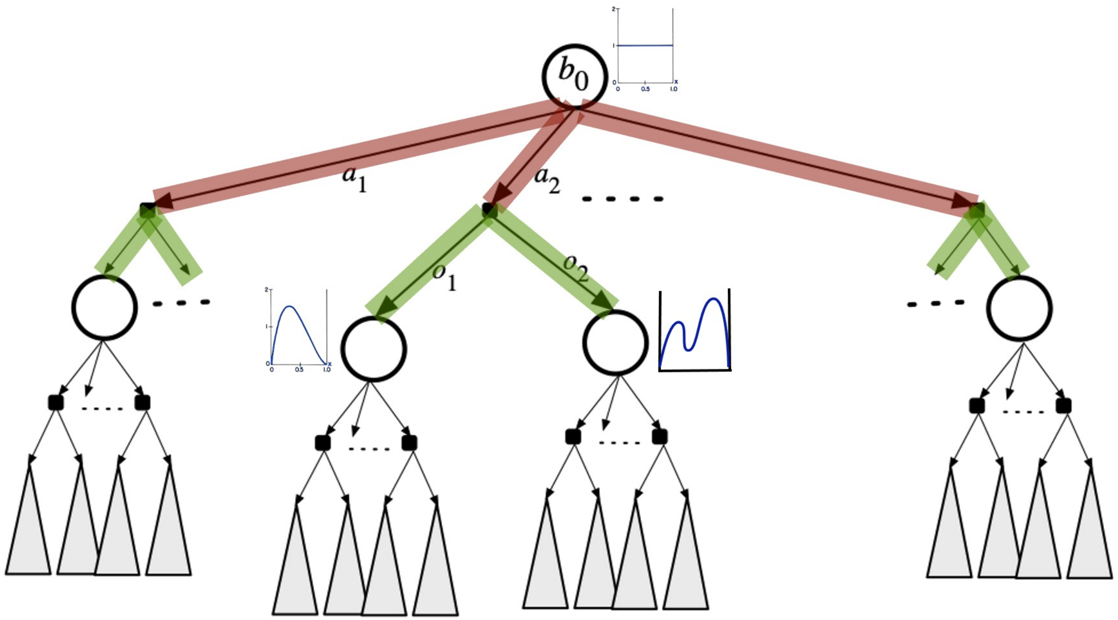
Problems where the state dynamics is governed by non-linear functions that admit no closed form solution and are computationally expensive to solve

Why is this a problem?

Marcus Hoerger

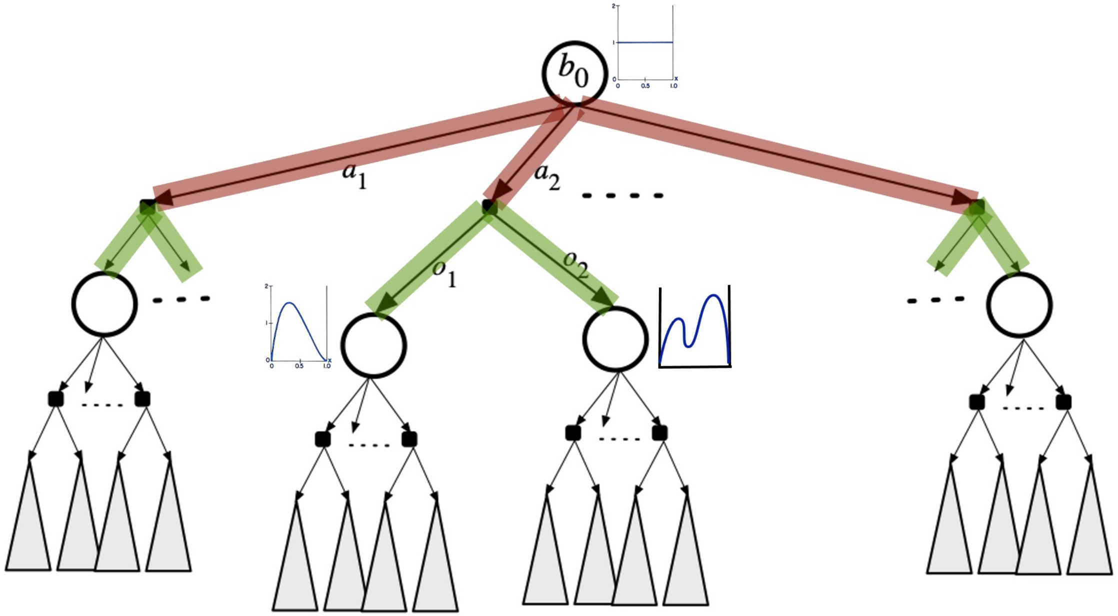
Marcus Hoerger, Hanna Kurniawati and Alberto Elfes. Multilevel Monte-Carlo for Solving POMDPs Online. IJRR 2022. An earlier version appears in ISRR'19.

Typical (On-line) POMDP Solving



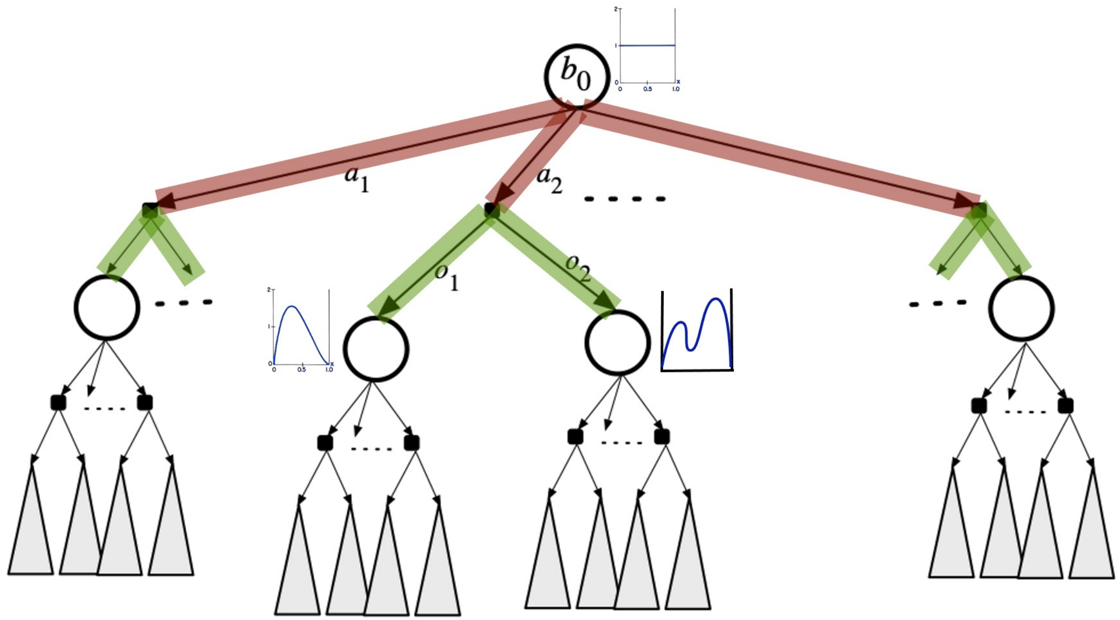
- Sample beliefs reachable from a given initial belief
  - Given a belief, sample a state and select an action to use
  - Given the sampled state and selected action, use generative model to sample the next state, observation & reward
- Perform backup at each sampled belief

The Problem with Complex Dynamics



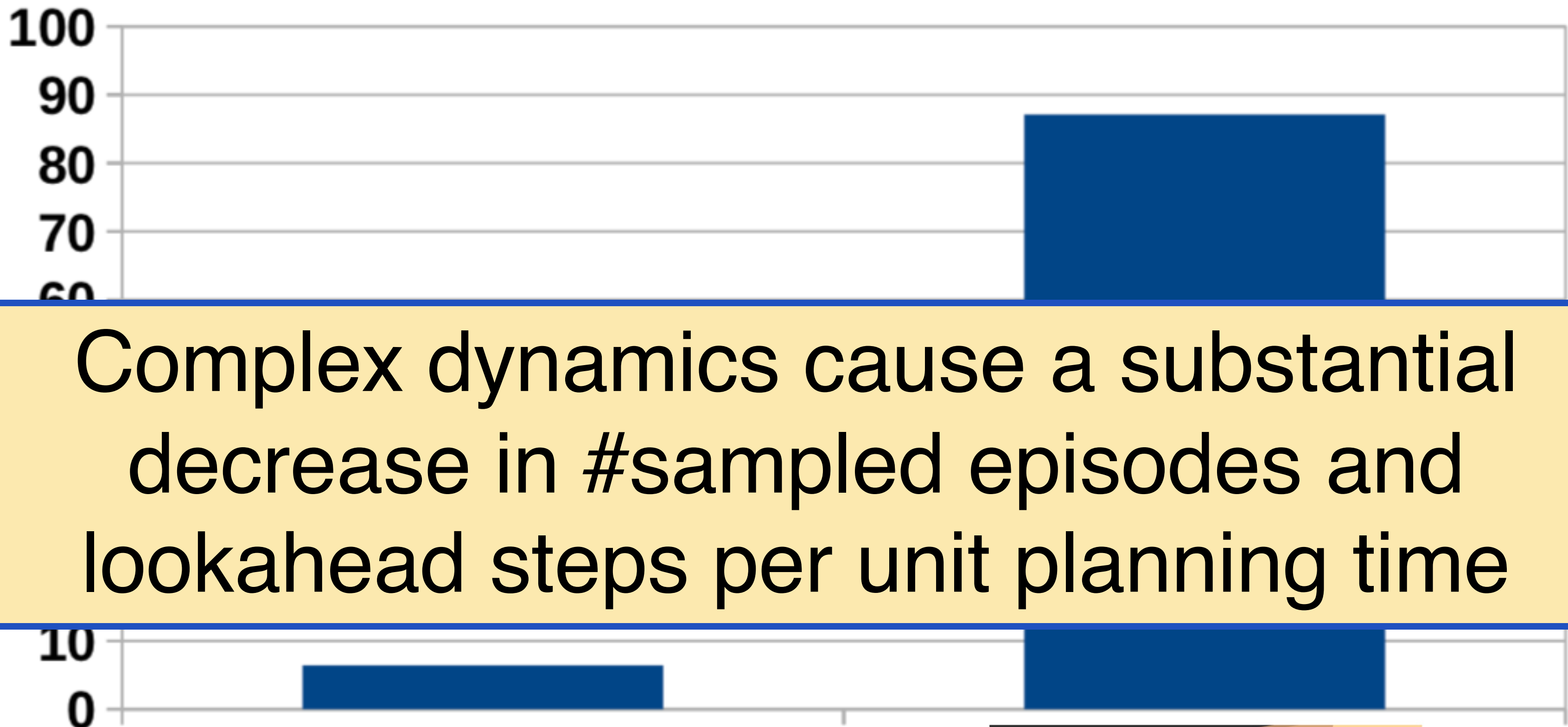
- Sample beliefs reachable from a given initial belief
  - Given a belief, sample a state and select an action to use
  - Given the sampled state and selected action, use **generative model** to sample the next state, observation & reward
- Perform backup at each sampled belief

The Problem with Complex Dynamics

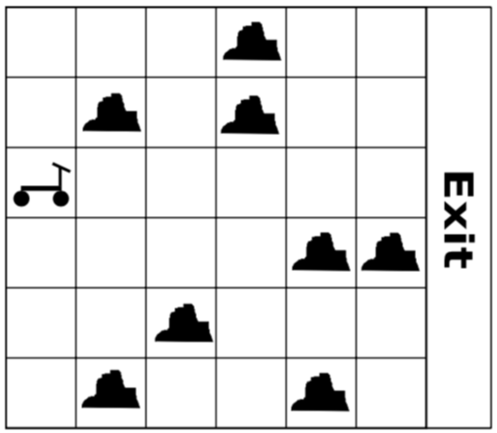


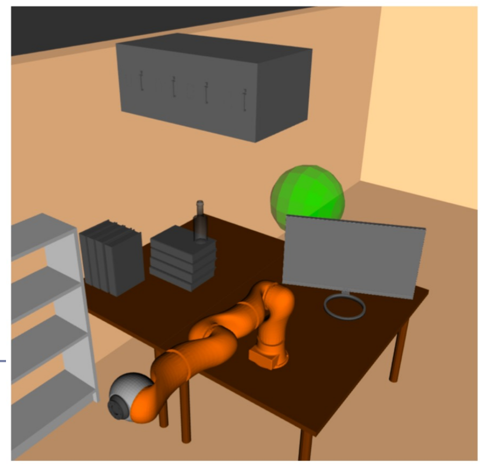
- The generative model is essentially a simulator for a single-step forward
- Need to solve the dynamics equation to sample next state
- To generate good policy, requires tens of thousands to millions of calls

Dynamics Computation / Total Time (%)



Complex dynamics cause a substantial decrease in #sampled episodes and lookahead steps per unit planning time





Approach: Simplified Dynamics

- Linearize
  - Belief-roadmap [Prentice et al., IJRR'10], LQG-MP [Berg, et al., IJRR'11], FIRM [Mohammadi et al., IJRR'14], HFR [Sun, et al., TRO'15]
  - Linearization doesn't always work well, esp. when we operate near constraints (obstacles, torque limits, etc.). Combine linearized and original dynamics

    [Marcus Hoerger, Hanna Kurniawati, Tirthankar Bandyopadhyay and Alberto Elfes. Linearization in Motion Planning under Uncertainty. *WAFR.* 2016.]

- Multi-level POMDP Planning

  [Marcus Hoerger, Hanna Kurniawati and Alberto Elfes. Multilevel Monte-Carlo for Solving POMDPs Online. *IJRR 2022.* An earlier version appears in ISRR'19.]

The Idea

- Have multiple levels of fidelity in solving the dynamic equation
  - Use the original complex dynamics (expensive to compute) sparingly
  - Use the lower-level fidelity dynamics (cheaper to compute) more often
- Multi-level Monte Carlo provides a framing for this idea

Marcus Hoerger, Hanna Kurniawati and Alberto Elfes. Multilevel Monte-Carlo for Solving POMDPs Online. *IJRR 2022.* An earlier version appears in ISRR'19.

Multi-level Monte Carlo (MLMC)

To compute E[X] of a random variable X, use

$$E[X] = E[X_0] + \sum_{l-1}^{L} E[X_l - X_{l-1}]$$

Where $X_0, X_1, \dots, X_L$ with $X_L = X$, X_0 the cheapest approximation of X, and $X_1, \dots, X_{L-1}$ are all cheaper approximations of X

X_{L-1} and X_L must be correlated

M. B. Giles. "Multilevel monte carlo methods." *Acta numerica* 24 (2015): 259-328.

Specifically,

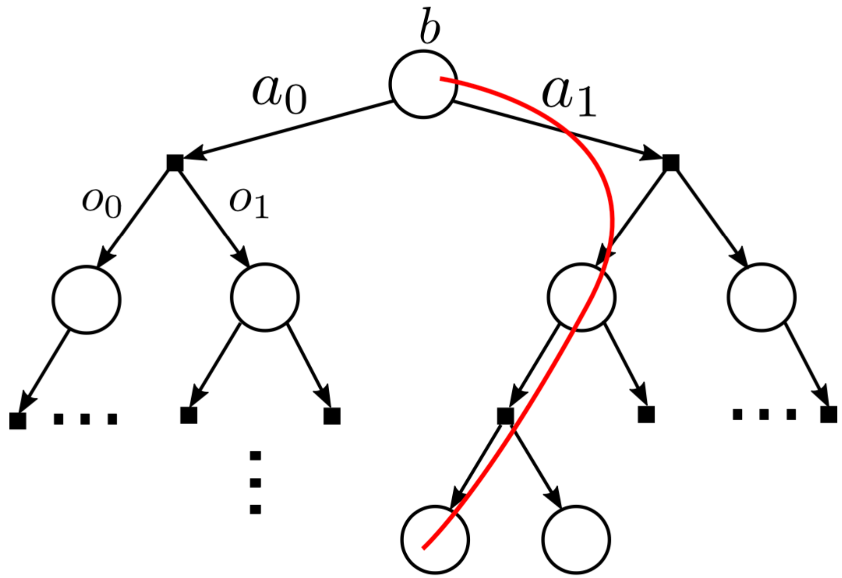
While planning time not over

Sample CoarseEpisode()

Sample fidelity level l

Sample CorrelatedEpisodes(l, $l-1$)

Use MLMC to improve value function estimates



$$\hat{Q}(b,a) = \boxed{\frac{1}{N_0^{b,a}} \sum_{i=1}^{N_0^{b,a}} V(h_0^i)} + \sum_{l=1}^{L} \frac{1}{N_l^{b,a}} \sum_{i=1}^{N_l^{b,a}} \left(V(h_l^i) - V(h_{l-1}^i)\right)$$

Specifically,

While planning time not over

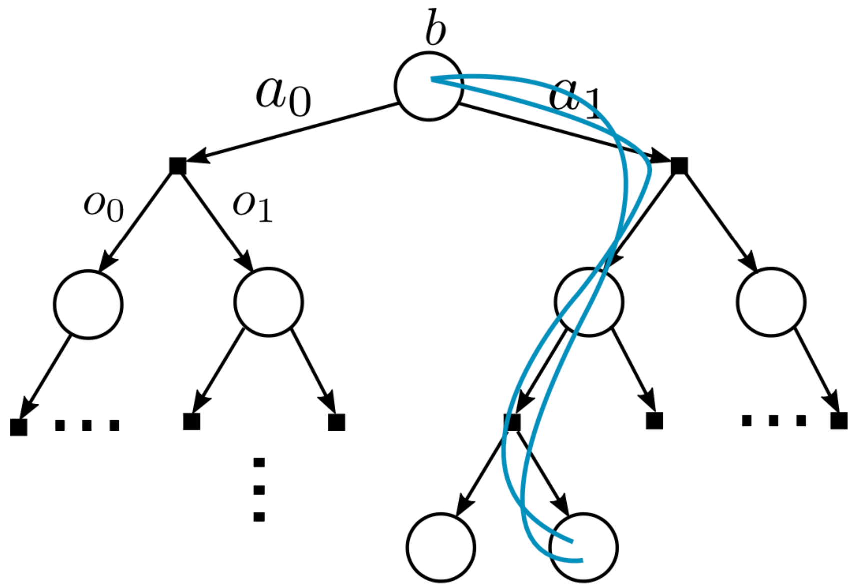
Sample CoarseEpisode()

Sample fidelity level l

Sample CorrelatedEpisodes(l, $l-1$)

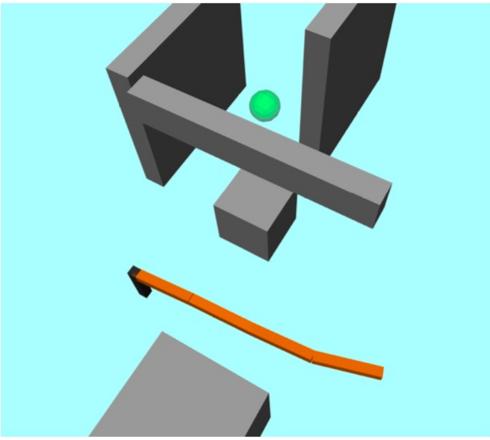
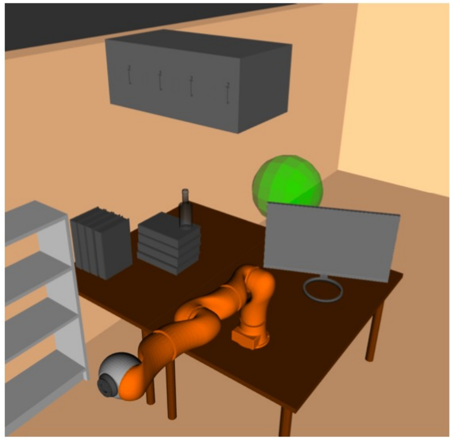
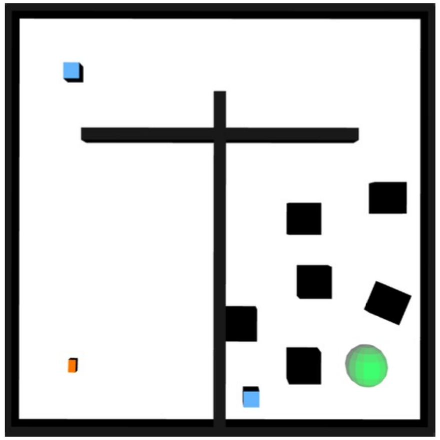
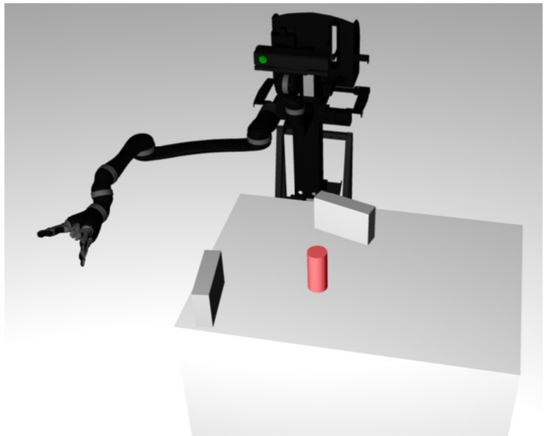
Use MLMC to improve value function estimates

Correlated samples:
Same sequence of actions



$$\hat{Q}(b,a) = \frac{1}{N_0^{b,a}} \sum_{i=1}^{N_0^{b,a}} V(h_0^i) + \sum_{l=1}^{L} \frac{1}{N_l^{b,a}} \sum_{i=1}^{N_l^{b,a}} \left(V(h_l^i) - V(h_{l-1}^i) \right)$$

Simulation Results

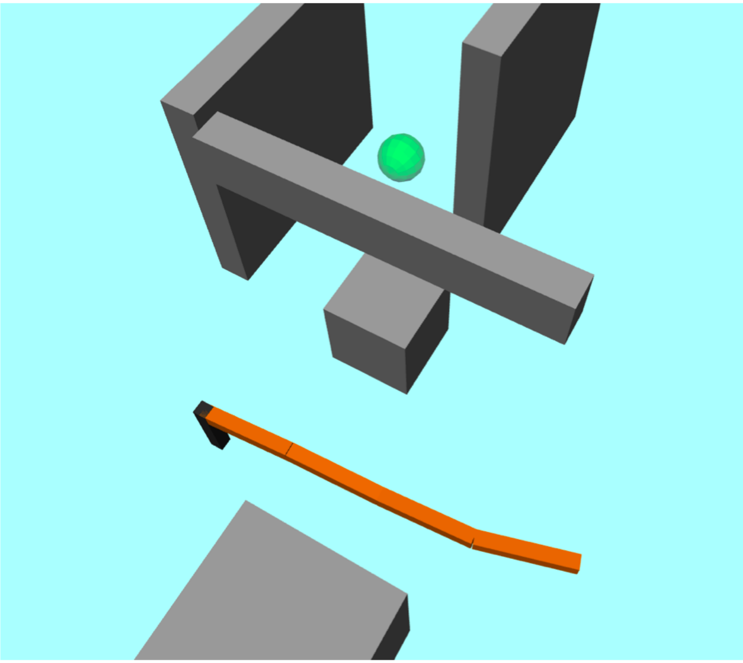





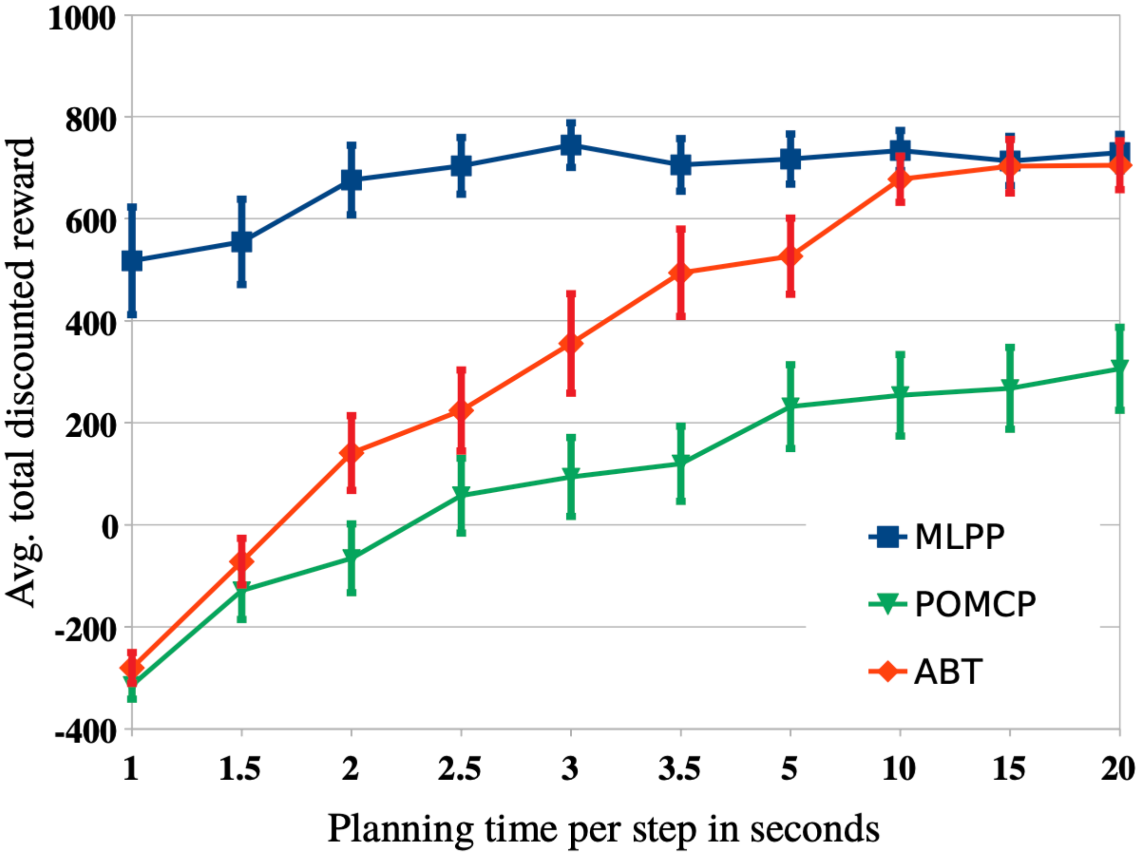
Average total discounted reward

	ABT	POMCP	MLPP	$t_{planning}$/ $step$
4DOF-Factory	-216.2 +/- 13.9	-323.5 +/- 4.7	**584.4 +/- 47.4**	1 second
KukaOffice	438.4 +/- 45.6	407.5 +/- 3.7	**693.2 +/- 44.8**	5 seconds
CarNavigation	-80.8 +/- 4.6	-111.6 +/- 4.4	**285.3 +/- 49.1**	1 second
MovoGrasp	386.9 +/- 48.8	204.8 +/- 34.1	**584.9 +/- 18.5**	1 second

More results are available in the paper

Simulation Results





More results are available in the paper

Similar Idea: Primitive Computation

- Primitive computation: Functions that are called many times by the solver
  - Improving primitive components → substantial improvement in overall solving capability
  - Holds for non-learning based, learning based, and their combinations

- **Complex dynamics:** Hoerger et.al. (ISRR'19, IJRR'22) – Sampling-based
- **When the POMDP model is not available:** Collins & Kurniawati (WAFR'20, IJRR'22) – End-to-end learning

What if we don't have the POMDP models?

Meaning, no transition, observation, & reward functions

Nicholas Collins

Nicholas Collins and Hanna Kurniawati. Locally-Connected Interrelated Network: A Forward Propagation Primitive. *IJRR 2022.* An earlier version appears in WAFR'20.

Solving (PO)MDP w/o (Z), T & R via End-To-End Learning

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

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

In VIN & QMDP-Net

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

- Forward propagation (e.g., transition function) is assumed to be independent of states…

Locally-Connected Interrelated Network (LCI-Net): The Idea

- Exploit locality structure for forward propagation

- This is a “primitive” component (used repeatedly)
- If we can make them more efficient, we’ll gain a lot! ☺

Nicholas Collins and Hanna Kurniawati. Locally-Connected Interrelated Network: A Forward Propagation Primitive. *IJRR 2022.* An earlier version appears in WAFR’20.

Forward Propagation Primitive in MDP

- LCI-Net: Represents Transition as T(s, a, ds)

$$V^*(s) = \max_{a \in A}\left(R(s,a) + \gamma \underbrace{\sum_{s\prime \in S} T(s'|s,a)V^*(s')}_{\text{Represented by LCI-Net}}\right)$$

Forward Propagation Primitive in POMDP

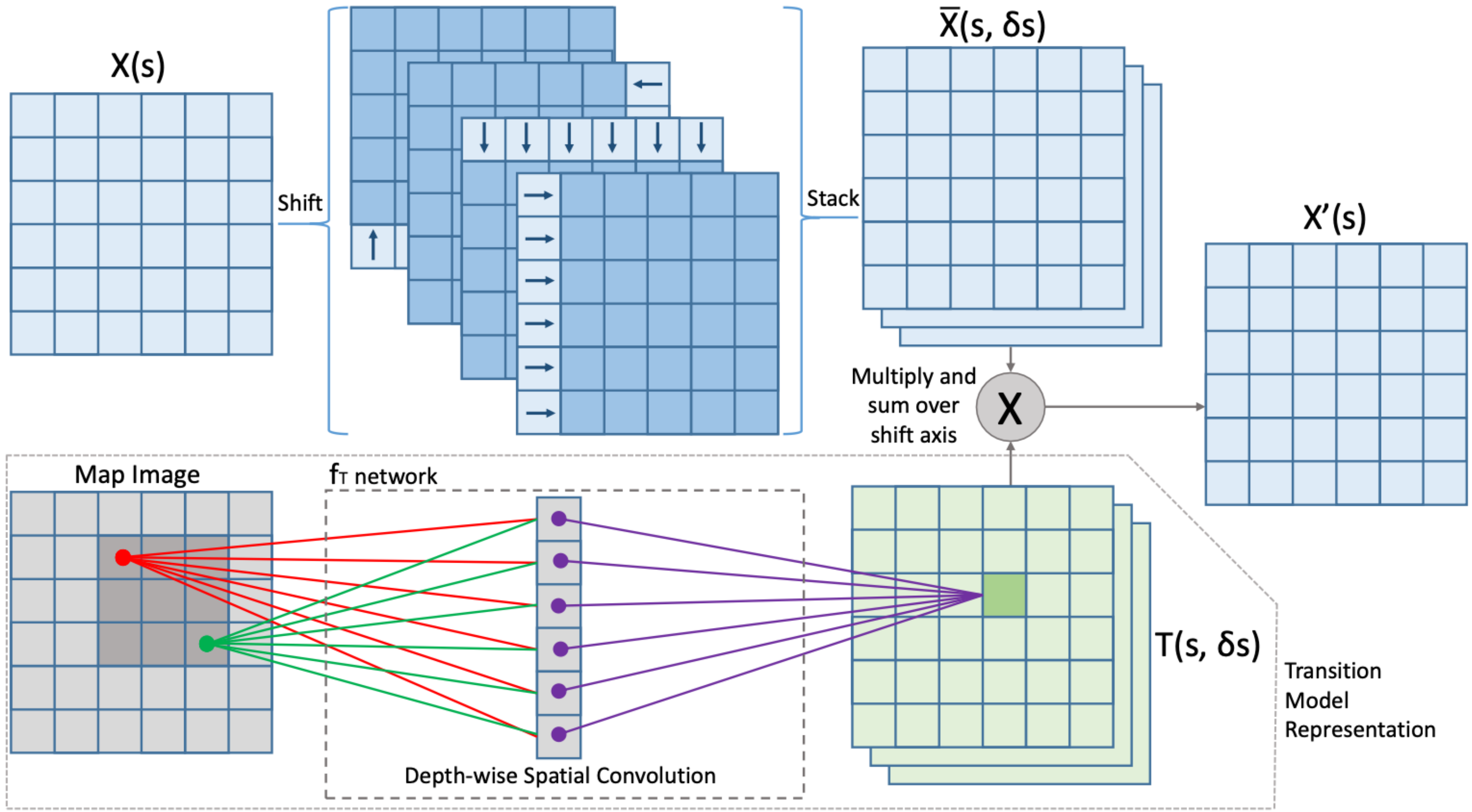
$$V^*(b) = \max_{a \in A}\left(\sum_{s \in S} R(s,a)b(s) + \gamma \sum_{o \in O} P(o|b,a)V^*(b')\right)$$

$$\sum_{o \in O}\sum_{s' \in S}\sum_{s \in S} b(s)Z(s',a,o)T(s,a,s')V^*(b')$$

Represented by LCI-Net

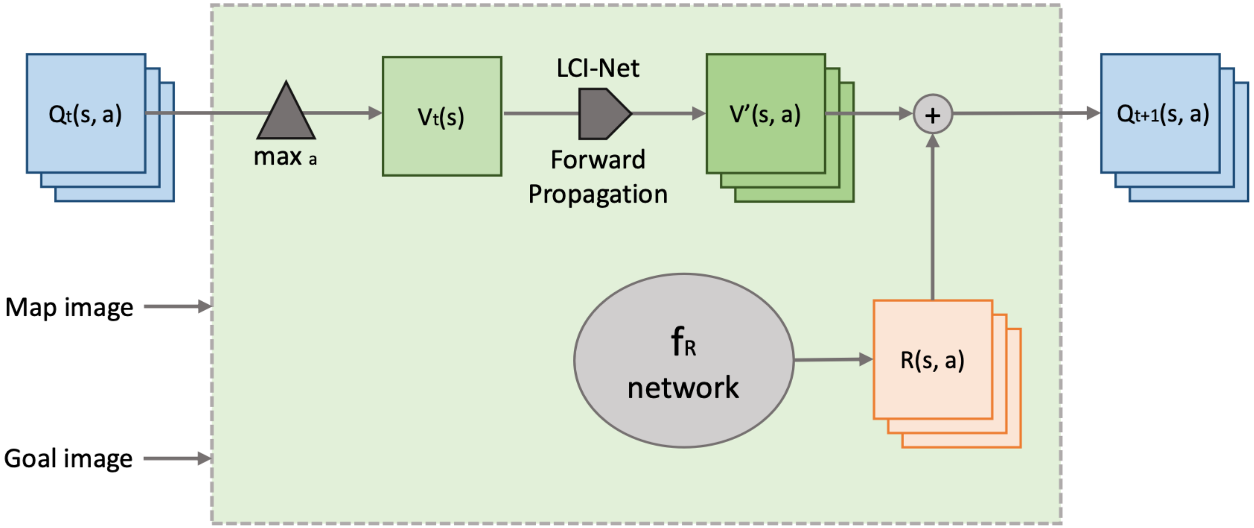
$$b'(s) = \tau(b,a,o)(s') = \eta Z(s',a,o)\sum_{s \in S} T(s,a,s')b(s)$$

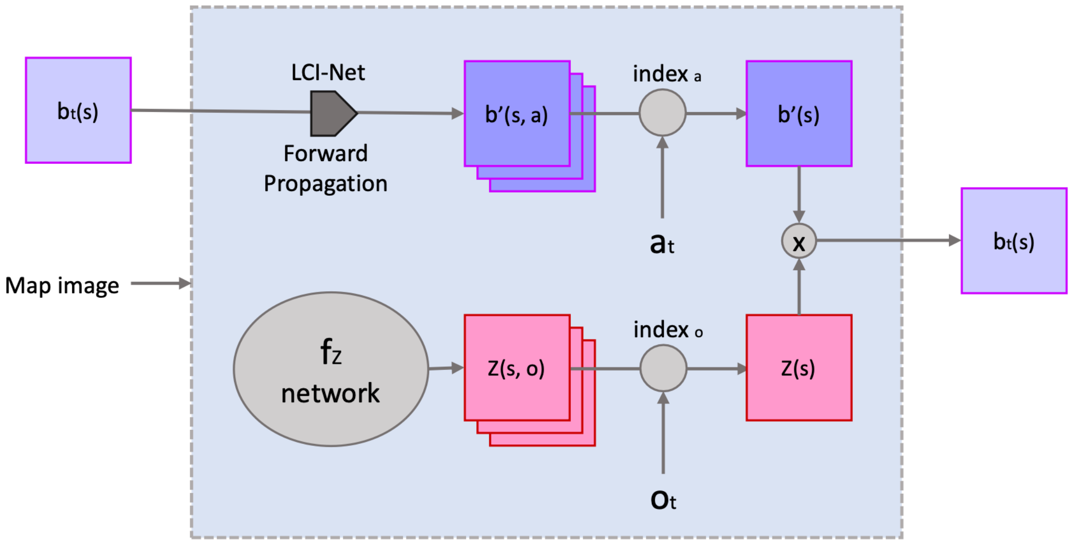
LCI-Net



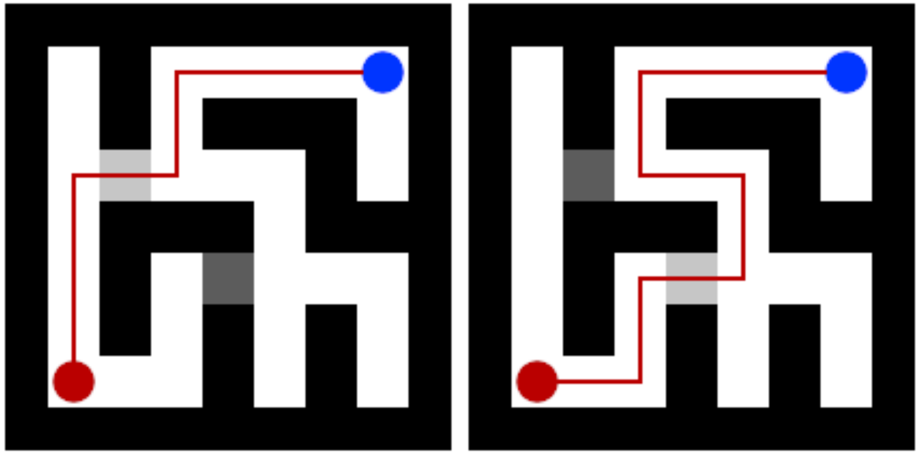
X(s)
Shift
Stack
X̄(s, δs)
X′(s)
Multiply and sum over shift axis
X
Map Image
f_T network
Depth-wise Spatial Convolution
T(s, δs)
Transition Model Representation

How LCI-Net is Used





Results: Dynamic Environment

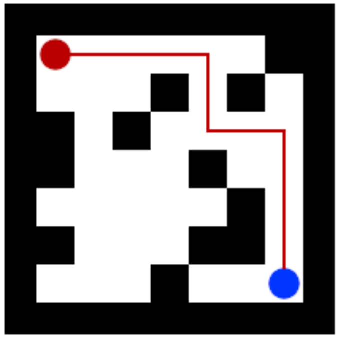


	% success		total training time (hours)	
	QMDP-Net	LCI-Net	QMDP-Net	LCI-Net
Dynmaze v1	89	98	5.50	4.75
Dynmaze v2	81	97	6.50	3.33

More results are available in the paper

Results: Generalization

	% success	
	VIN	LCI-Net
Building 79 Trained on 16X16	17	37
Intel Labs Trained on 16X16	6	23
Hospital Trained on 16X16	20	51



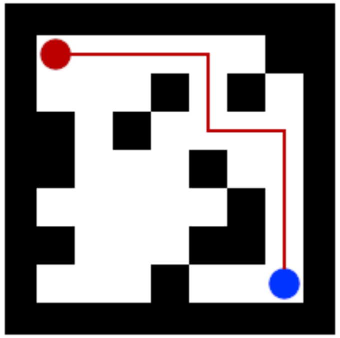
Training

Evaluation

More results are available in the paper

Results: Generalization

	% success	
	QMDP-Net	LCI-Net
Building 79 Trained on 10X10	11	57
Intel Labs Trained on 10X10	7	55
Hospital Trained on 10X10	9	43



Training

Evaluation

More results are available in the paper

Similar Idea: Primitive Computation

- Primitive computation: Functions that are called many times by the solver
  - Improving primitive components → substantial improvement in overall solving capability
  - Holds for non-learning based, learning based, and their combinations

- **Complex dynamics:** Hoerger et.al. (ISRR'19, IJRR'22) – Sampling-based
- **When the POMDP model is not available:** Collins & Kurniawati (WAFR'20, IJRR'22) – End-to-end learning

The Problems & Some of Our Solutions

Sampling-based & Learning-based

- **Large state space**
  Kurniawati, et.al. (RSS'08), RSS'21 Test of Time Award
- **Large action space** – up to 12-D cont. action space
  Seiler, et.al. (ICRA'15, best paper award finalist), Wang, et.al. (ICAPS'18), Hoerger, et.al. (WAFR'22, IJRR'23), Hoerger, et.al. (AAAI'24 oral, to appear)
- **Large observation space**
  Kurniawati, et.al. (RSS'11, Auro'12), Hoerger & Kurniawati (ICRA'21)
- **Dynamically changing model**
  Kurniawati & Patrikalakis (WAFR'12), Kurniawati & Yadav (ISRR'13), Chen & Kurniawati (NeurIPS'23)
- **Long planning horizon**
  Kurniawati, et.al. (ISRR'09, IJRR'11), Liang & Kurniawati (IROS'23), Kim, et.al. (NeurIPS'23)
- **Complex dynamics**
  Hoerger, et.al. (WAFR'16), Hoerger et.al. (ISRR'19, IJRR'22)
- **When the POMDP model is not available**
  Collins & Kurniawati (WAFR'20, IJRR'22)

Take Home Message

The POMDP has now become practical for realistic robotics problems (to some extent).

Improving primitive modules helps improves overall performance of sampling-based, learning-based, and those in-between

Thank you