DeepMind

Behavior Priors for data efficient Reinforcement Learning

Dhruva Tirumala

Contact: dhruvat [at] google.com

dhruva.bukkapatnam.20 [at] ucl.ac.uk

DeepMind Research Engineer,
UCL PhD candidate

A quick introduction

- **Who am I?**
  - Research engineer at DeepMind since 2016
  - Primarily focused on Reinforcement Learning (RL) for continuous control
  - Since 2019:
    - Joined the UCL-DeepMind PhD program
    - Co-advised by Dr. Nicolas Heess (at DeepMind) and Prof. Danail Stoyanov (at UCL)

A quick introduction

- **Who am I?**
  - Research engineer at DeepMind since 2016
  - Primarily focused on Reinforcement Learning (RL) for continuous control
  - Since 2019:
    - Joined the UCL–DeepMind PhD program
    - Co-advised by Dr. Nicolas Heess (at DeepMind) and Prof. Danail Stoyanov (at UCL)

Introduction

- **What am I going to talk about?**
  - A set of ideas that culminated in a recent submission to the Journal of Machine Learning Research
  - *Narrow view:*
    - Specific experiments that illustrate certain ideas
  - *Broad view:*
    - One way of thinking about RL through a lens of probabilistic modeling.
- **What is the purpose of this talk?**
  - Hopefully spawns interesting discussion and collaboration*!
  - *Subject to constraints from my DeepMind hat :)

Disclaimer

- **Assumptions:**
  - Some background in Probabilistic modeling; Reinforcement Learning and Machine Learning.
  - Allows us to explore these ideas deeper
  - Additional introductory material at end!

- Feel free to interrupt with questions and points of clarification!
- Ultimately these ideas are best digested in text form :)
  - *"Behavior Priors for Efficient Reinforcement Learning"*
  - https://arxiv.org/abs/2010.14274

“If I have seen further it is by standing on the shoulders of Giants”

– Isaac Newton

Alexandre Galashov

Leonard Hasenclever

Razvan Pascanu

Jonathan Schwarz

Guillaume Desjardins

Wojtek Czarnecki

Arun Ahuja

“If I have seen further it is by standing on the shoulders of Giants”

- Isaac Newton

Yee Whye Teh

Nicolas Heess

“If I have seen further it is by standing on the shoulders of Giants”

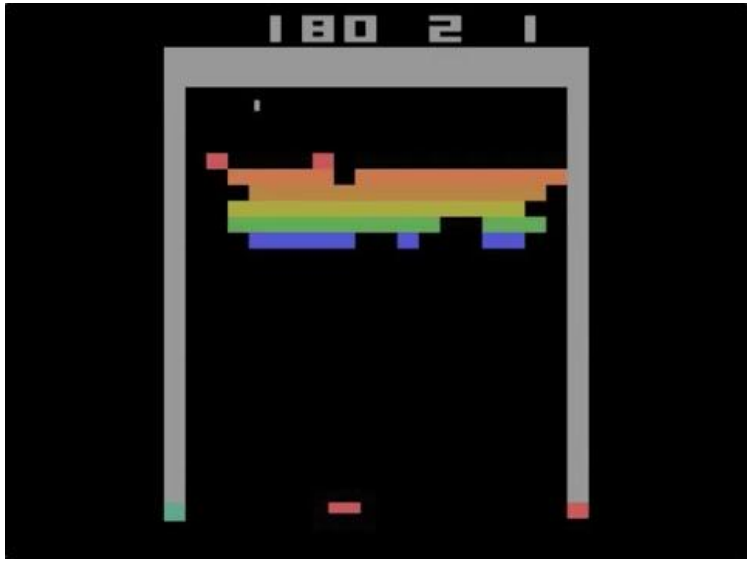
– Isaac Newton

Outline

- **The control challenge**
- Method: ‘Priors’ over behavior
- Experiments
- Discussion

Motivation

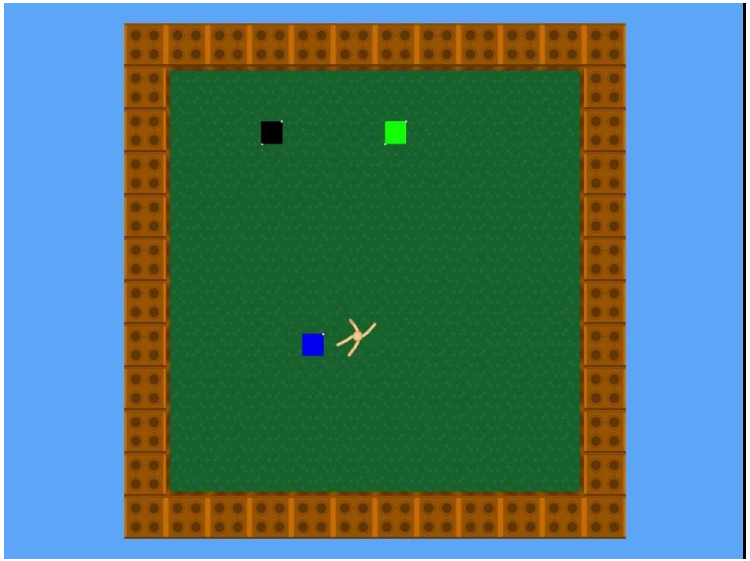
Let’s revisit one of RL’s most recent successes:



DQN on Breakout 100 episodes in training

Motivation - Control

Things look a bit different in control and robotics...



Spider (Ant) with a ‘random’ Gaussian policy

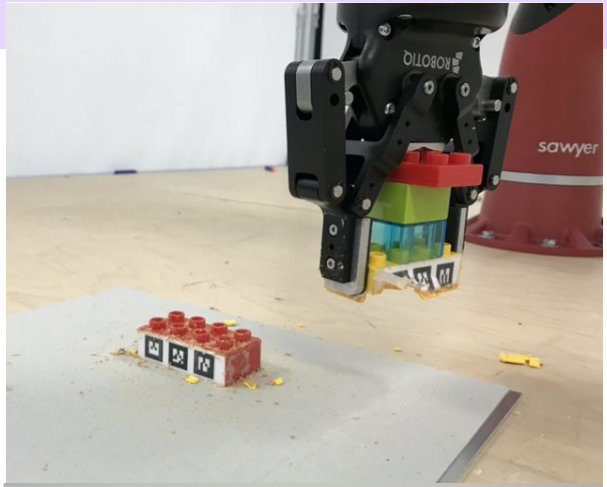
Humanoid with a Gaussian policy

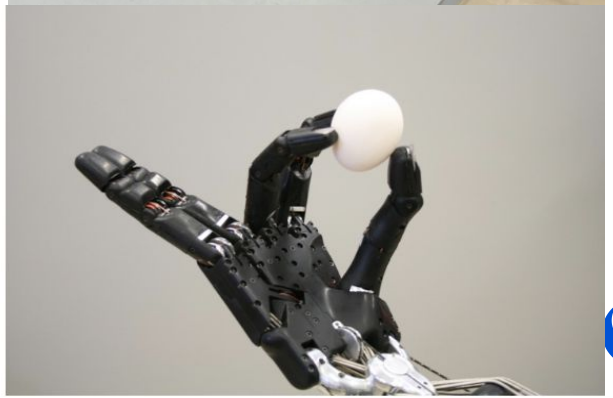
The control challenge

- Continuous control has a number of challenges:
  - **Dimensionality**
  - **Perception**
  - **Diversity of tasks**
  - **Multiple timescales**
  - **Real world constraints:**
    - Data efficiency; Wear and tear
    - Finite interaction and/or computation budgets

9 action dim

56 action dim





Given a limited computational or interaction budget, it is necessary to restrict the action search space, to be able to control high-dimensional systems in practice

Much prior work focuses exactly on this

- These challenges have long been studied!
- This has been the focus of:
  - RL and control
  - Transfer learning
  - Meta-learning

Policy Transfer via Modularity

Ignasi Clavera*[1], David Held*[1] and Pieter Abbeel[1,2,3]

samples required for learning. This is especially true for deep policy gradient methods; such methods have demonstrated impressive results in simulation, but their use in the real world is limited by their large sample complexity [17], [13],

Learning Attractor Landscapes for Learning Motor Primitives

Auke Jan Ijspeert[1,3*], Jun Nakanishi[2], and Stefan Schaal[1,2]

goal state. While reinforcement learning offers a theoretical framework to learn such control policies from scratch, its applicability to higher dimensional continuous state-action spaces remains rather limited to date. Instead of learning from scratch, in this paper we

Efficient Off-Policy Meta-Reinforcement Learning via Probabilistic Context Variables

Kate Rakelly[1*] Aurick Zhou[1*] Deirdre Quillen[1] Chelsea Finn[1] Sergey Levine[1]

few trials, during training, they require massive amounts of data drawn from a large set of distinct tasks, exacerbating the problem of sample efficiency that plagues RL algorithms.

Much prior work focuses exactly on this

- These challenges have long been studied!
- This has been the focus of:
  - RL and control
  - Transfer learning
  - Meta-learning
  - **Hierarchical Reinforcement Learning (HRL)**
- Model-free RL for control has largely focused on learning and transferring ‘skills’ or ‘behaviors’

Recent Advances in Hierarchical Reinforcement Learning

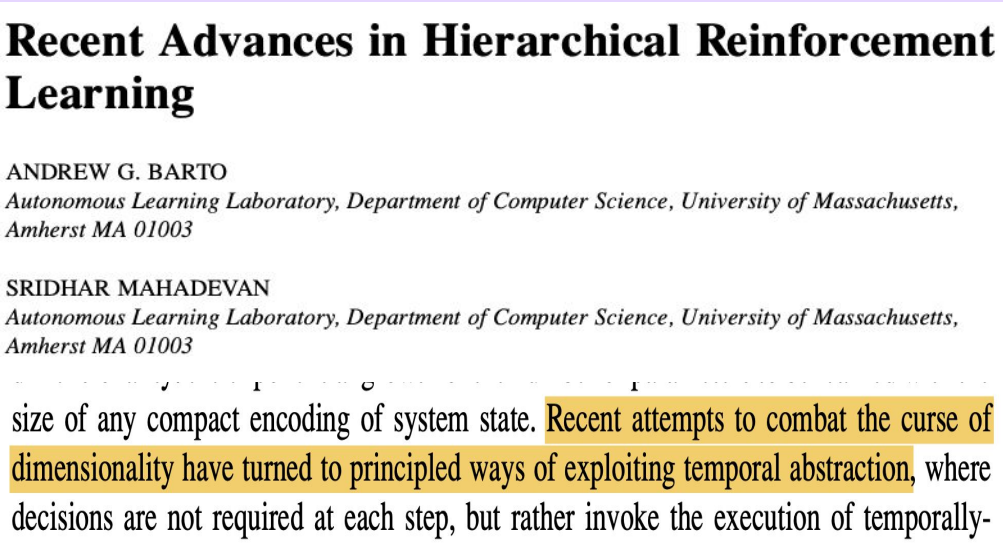
ANDREW G. BARTO
Autonomous Learning Laboratory, Department of Computer Science, University of Massachusetts, Amherst MA 01003

SRIDHAR MAHADEVAN
Autonomous Learning Laboratory, Department of Computer Science, University of Massachusetts, Amherst MA 01003

size of any compact encoding of system state. Recent attempts to combat the curse of dimensionality have turned to principled ways of exploiting temporal abstraction, where decisions are not required at each step, but rather invoke the execution of temporally-

Much prior work focuses exactly on this

- These challenges have long been studied!
- This has been the focus of:
  - RL and control
  - Transfer learning
  - Meta-learning
  - **Hierarchical Reinforcement Learning (HRL)**
- Model-free RL for control has largely focused on learning and transferring ‘skills’ or ‘behaviors’
- **Let’s try to devise a more unified perspective**
  - Through the lens of probabilistic modeling



Recent Advances in Hierarchical Reinforcement Learning

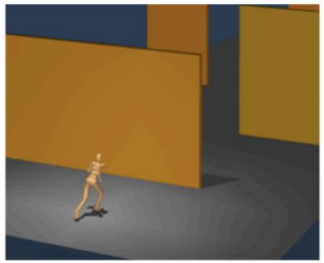
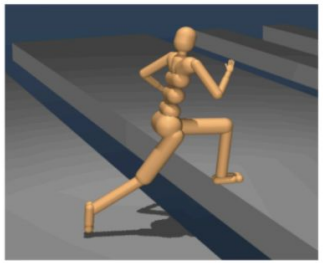
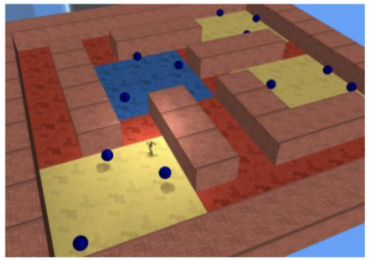
ANDREW G. BARTO
Autonomous Learning Laboratory, Department of Computer Science, University of Massachusetts, Amherst MA 01003

SRIDHAR MAHADEVAN
Autonomous Learning Laboratory, Department of Computer Science, University of Massachusetts, Amherst MA 01003

size of any compact encoding of system state. Recent attempts to combat the curse of dimensionality have turned to principled ways of exploiting temporal abstraction, where decisions are not required at each step, but rather invoke the execution of temporally-

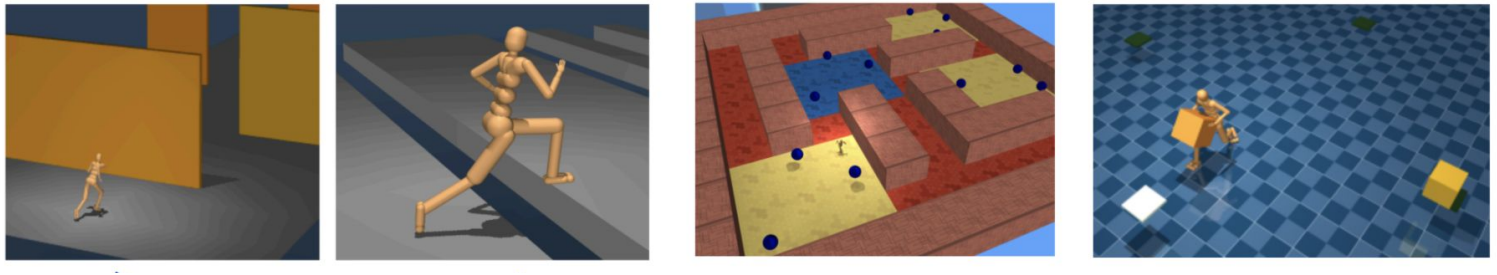
‘Behaviors’ as probabilistic models

Consider the humanoid solving many tasks


‘Behaviors’ as probabilistic models

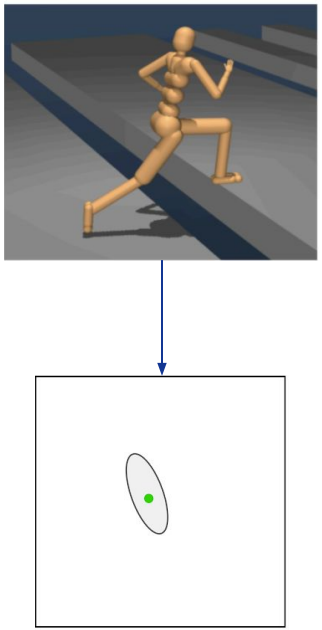
The solution to each task espouses a distribution over trajectories



$$\tau = (s_0, a_0, s_1, a_1, s_2, \ldots)$$

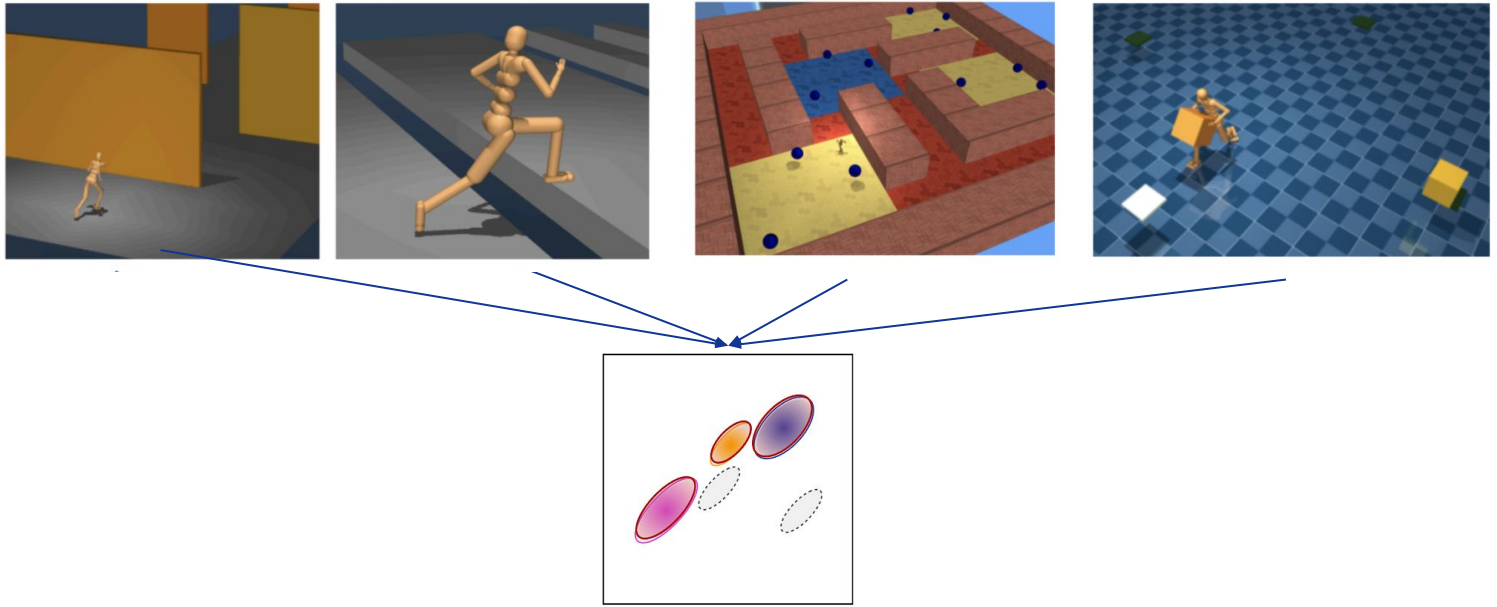
‘Behaviors’ as probabilistic models

(Overly) simplified view of a task in trajectory space



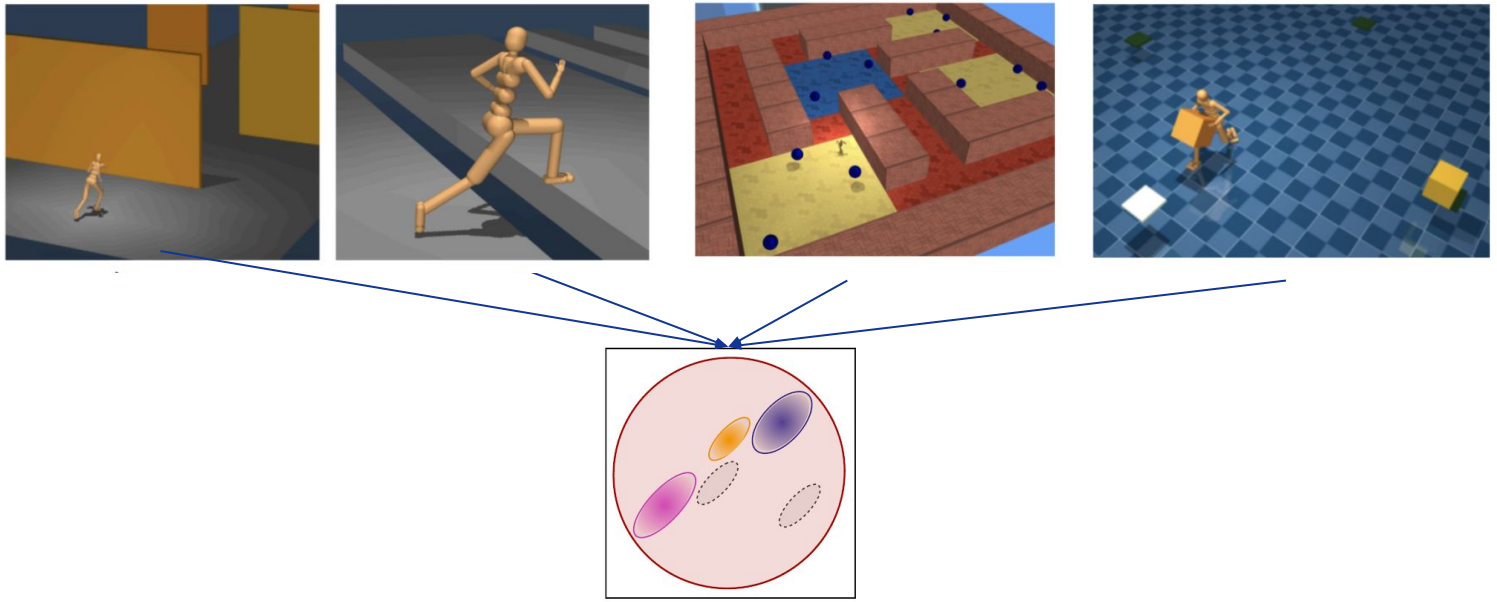
‘Behaviors’ as probabilistic models

The solutions to many tasks may or may not have overlap in that space



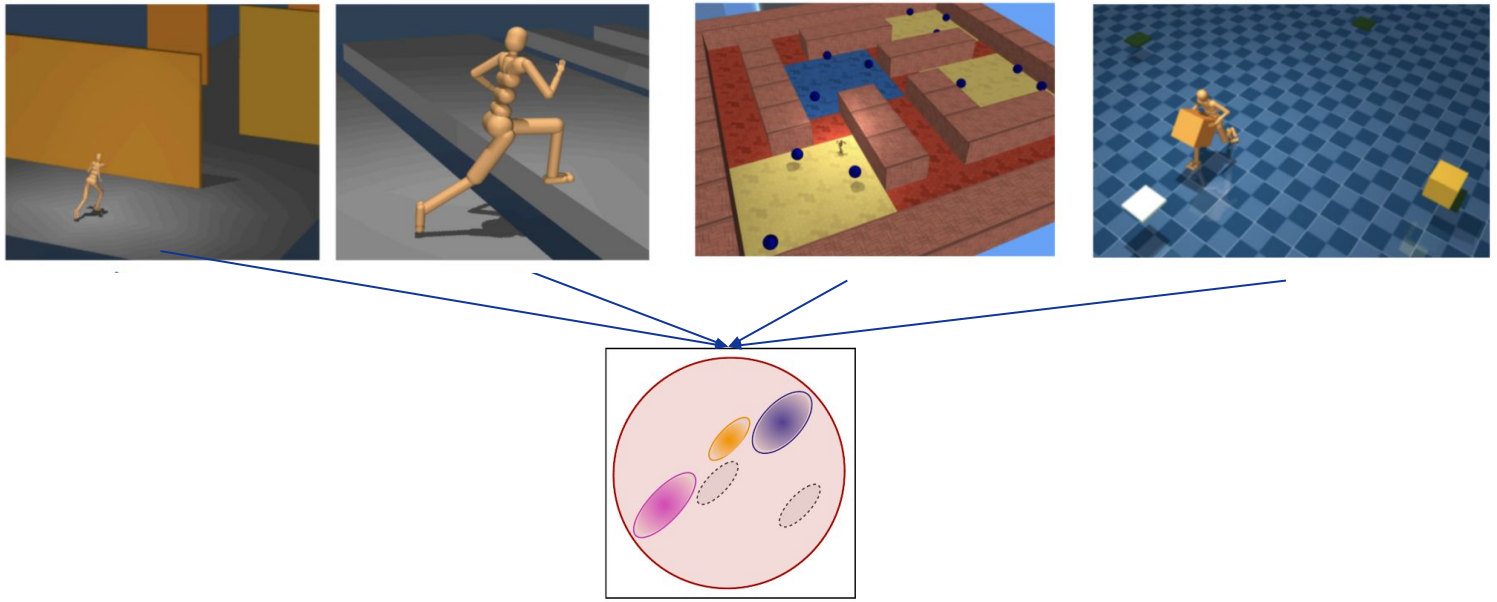
‘Behaviors’ as probabilistic models

When we refer to ‘skills’ or ‘behaviors’, we’re really talking about the trajectory space shared across tasks.



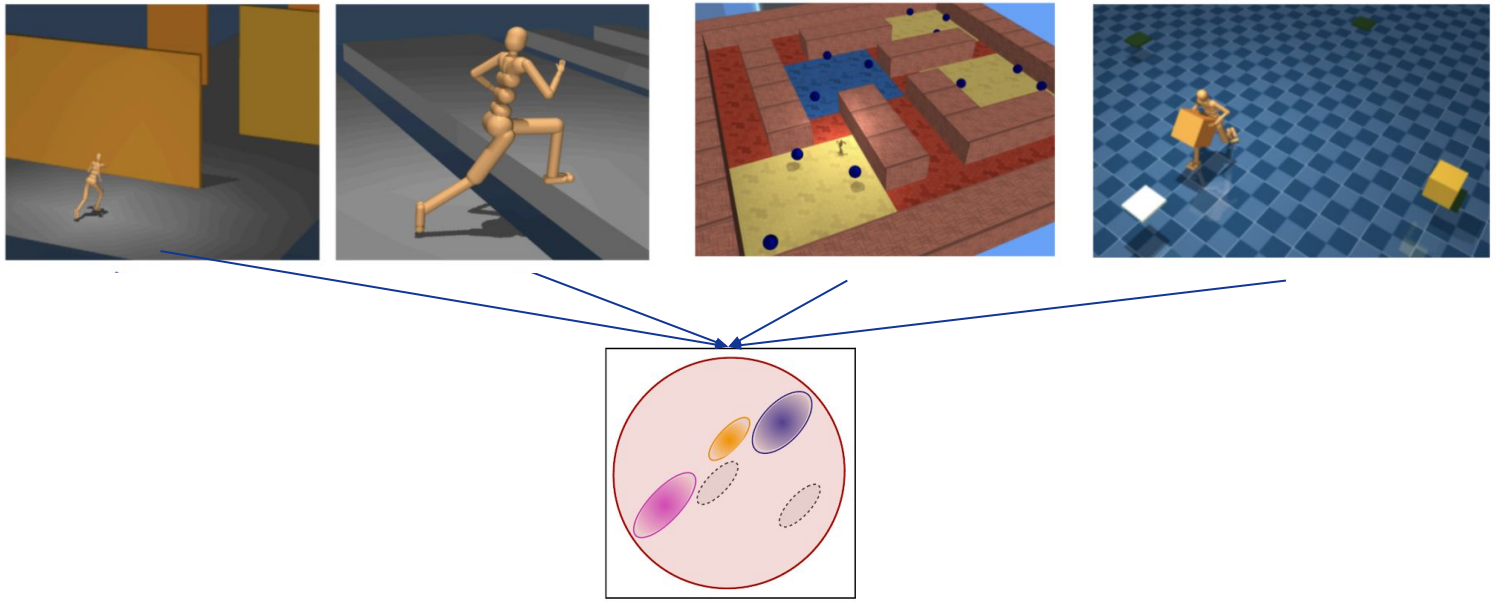
‘Behaviors’ as probabilistic models

Q: How can we learn these behaviors?



‘Behaviors’ as probabilistic models

Q: How can we incorporate it into the RL problem?



Outline

- The control challenge
- **Method: ‘Priors’ over behavior**
- Experiments
- Discussion

Setup

Markov Decision Process (MDP)

$$P : S \times A \times S \to \mathbb{R}_+ \qquad P_0 : S \to \mathbb{R}_+$$

$$\pi(\tau) = P_0(s_0) \prod_{t=0}^{\infty} P(s_{t+1}|s_t, a_t)\pi(a_t|x_t).$$

$$\tau = (s_0, a_0, s_1, a_1, \dots) \quad x_t = (s_0, a_0, \dots s_t)$$

Setup

Markov Decision Process (MDP)

$$P : S \times A \times S \to \mathbb{R}_+ \qquad P_0 : S \to \mathbb{R}_+$$

$$\pi(\tau) = P_0(s_0) \prod_{t=0}^{\infty} P(s_{t+1}|s_t, a_t)\pi(a_t|x_t).$$

$$\tau = (s_0, a_0, s_1, a_1, \dots) \quad x_t = (s_0, a_0, \dots s_t)$$

KL-Regularized RL

- We can begin with the KL-Regularized RL objective:

$$\mathcal{L} = \mathbb{E}_{\pi}[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) - \alpha\gamma^t \mathrm{KL}[\pi(a_t|x_t)||\pi_0(a_t|x_t)]].$$

Maximize Rewards

Minimize KL to some reference behavior π_0

$\pi(\tau)$ **Policy**

$\pi_0(\tau)$ **'Reference' behavior**

High level idea

$$\mathcal{L} = \mathbb{E}_{\pi}[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) - \alpha\gamma^t \mathrm{KL}[\pi(a_t|x_t)||\pi_0(a_t|x_t)]].$$

$\pi_0(\tau)$ = Uniform distribution => Entropy Regularized RL

High level idea

$$\mathcal{L} = \mathbb{E}_{\pi}[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) - \alpha\gamma^t \mathrm{KL}[\pi(a_t|x_t)||\pi_0(a_t|x_t)]].$$

$\pi_0(\tau)$ = EM style optimization => MPO (as one example)

*Maximum a-posteriori policy optimization (Abdolmaleki et. al)

High level idea

$$\mathcal{L} = \mathbb{E}_{\pi}[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) - \alpha\gamma^t \mathrm{KL}[\pi(a_t|x_t)||\pi_0(a_t|x_t)]].$$

$\pi_0(\tau)$ = Behavior Prior?

A step back: DISTRAL

- The multi-task DISTRAL setup

$$\mathcal{L} = \sum_{w} p(w) \mathbb{E}_{\pi_w} \left[\sum_{t} \gamma^t r_w(s_t, a_t) - \gamma^t \mathrm{KL}[\pi_w(a_t|x_t) || \pi_0(a_t|x_t)] \right]$$

Distribution over tasks

Task specific rewards

Task-agnostic prior

*DISTRAL: Robust Multitask Reinforcement Learning: Teh et al.

A step back: DISTRAL

- Given a set of optimal task-specific policies, optimal prior is:

$$\pi_0^* = \sum_w p(w)\pi_w(a_t|x_t)$$

Weighted mixture of task specific policies

A step back: DISTRAL

- Given a specific task w, and and a reference behavior $\pi_0(\tau)$, objective minimized by:

$$\pi_w^*(a|x_t) = \pi_0(a_t|x_t)\exp(Q_w^*(x_t,a) - V_w^*(x_t))$$

$$V_w^*(x_t) = \max_{\pi_w \sim \Pi} \mathbb{E}_{x_t \sim d_{\pi_w,t}} V_w^\pi(x_t)$$

$$Q_w^*(x_t,a) = r(s_t,a) + \gamma \mathbb{E}_{P(x_{t+1}|x_t,a)}[V_w^*(x_{t+1})]$$

Prior reweighted by (soft-)Q function

See also :"Reinforcement Learning with Deep Energy-Based Policies"*: Haarnooja et al. and *Modeling Purposeful Adaptive Behavior with the Principle of Maximum Causal Entropy: Ziebart*

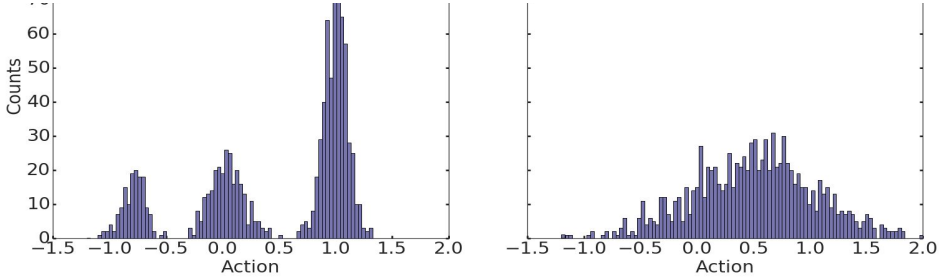
A step back: DISTRAL

$$\mathcal{L} = \sum_w p(w) \mathbb{E}_{\pi_w} \left[\sum_t \gamma^t r_w(s_t, a_t) - \gamma^t \mathrm{KL}[\pi_w(a_t|x_t) || \pi_0(a_t|x_t)] \right]$$

- Intuitions:
  - Prior is a **mixture** over task-specific policies that we can **specialize** to new tasks.

A step back: DISTRAL

$$\mathcal{L} = \sum_w p(w) \mathbb{E}_{\pi_w} \left[\sum_t \gamma^t r_w(s_t, a_t) - \gamma^t \mathrm{KL}[\pi_w(a_t|x_t) || \pi_0(a_t|x_t)] \right]$$



- Intuitions:
  - Prior is a **mixture** over task-specific policies that we can **specialize** to new tasks.
  - KL direction means prior is **mode-covering.**
    - Note that if Π_0 had access to **all** information, the solution is trivial.

Generalizing this intuition

- Can we generalize this intuition to the original objective?
- Key idea: **Constrain** the processing capacity of Π_0 thereby forcing it to **generalize.**

$$\mathcal{L} = \mathbb{E}_{\pi}[\sum_{t} \gamma^t r(s_t, a_t)] - \gamma^t \mathrm{KL}[\pi(a_t|x_t)||\pi_0(a_t|x_t^D)].$$

$$\pi(\cdot|x_t) = \pi(\cdot|x_t^G, x_t^D)$$

Generalizing this intuition

$$\mathcal{L} = \mathbb{E}_\pi[\sum_t \gamma^t r(s_t, a_t)] - \gamma^t \mathrm{KL}[\pi(a_t|x_t)||\pi_0(a_t|x_t^D)].$$

DISTRAL → $x_t^G = w \quad x_t^D = x_t$

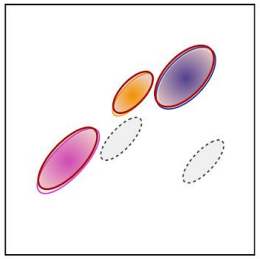
Generalizing this intuition

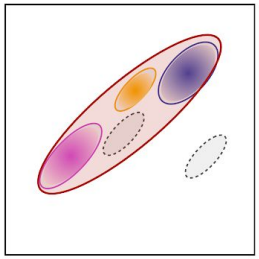
$$\mathcal{L} = \mathbb{E}_{\pi}[\sum_{t} \gamma^{t} r(s_t, a_t)] - \gamma^{t} \mathrm{KL}[\pi(a_t|x_t)||\pi_0(a_t|x_t^D)].$$

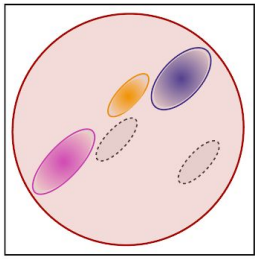
$$\longrightarrow x_t^D = (a_0, a_1, \ldots a_t)$$

Generalized Information asymmetry

- Key intuitions:
  - **Constrain** the processing capacity of Π_0 thereby forcing it to **generalize.**
  - Prior also acts like a **shaping reward** : prefer solutions that have support in the prior.
  - Behaviors are just **statistical patterns** that can be captured in a reusable way.
  - Different choices for the **information constraints**, **model capacity and architecture** can lead to different *kinds* of priors.







Outline

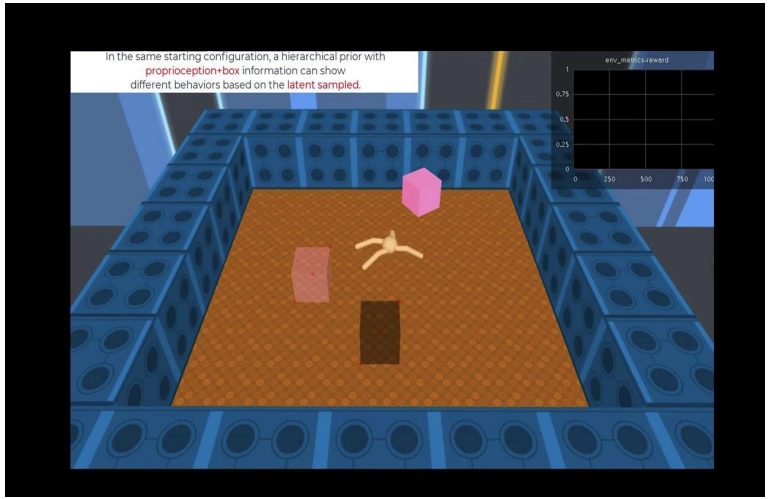
- The control challenge
- Method: ‘Priors’ over behavior
- **Experiments**
- Discussion

Information constraints

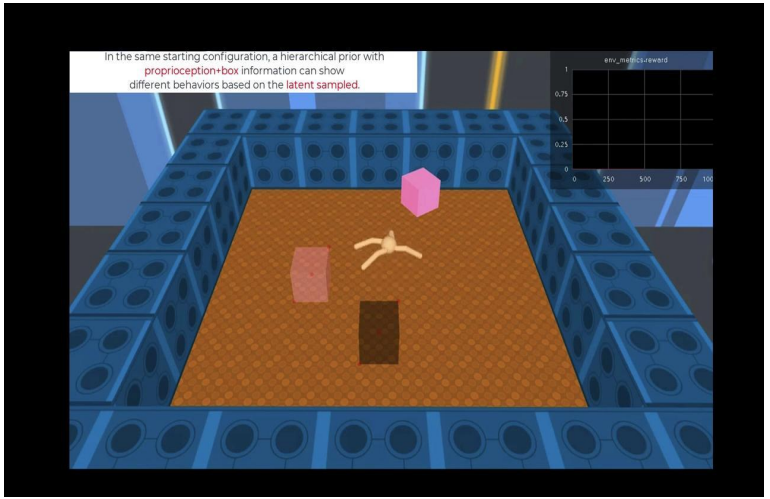
- What we would like to show
  - **Information constraints** can lead to more general behavior priors.
- Desiderata:
  - Multi-task setup (similar to DISTRAL)
  - Tasks with many choices of information
  - Common set of shared behaviors across the tasks

Locomotion and Manipulation

- Illustrative example: Move a box to a target (**Manipulation**) and go to another target (**Locomotion**)
- Setup:
  - Multi-task scenario where each ‘task’ involves a specific configuration of box and targets.
  - The *prior* is shared across all of these ‘tasks’ but crucially only has access to a subset of the information.



Task setup



Task Solution

Locomotion and Manipulation

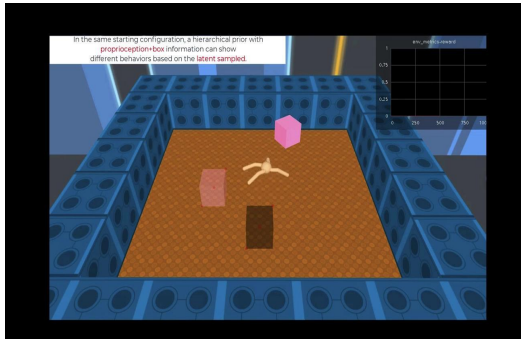
- Task: Move a box to a target (Manipulation) and go to another target (Locomotion)
- Information set:
  - Proprioception
  - Box Locations
  - Target locations
  - Identity of target to move to.

Many choices for information factorization

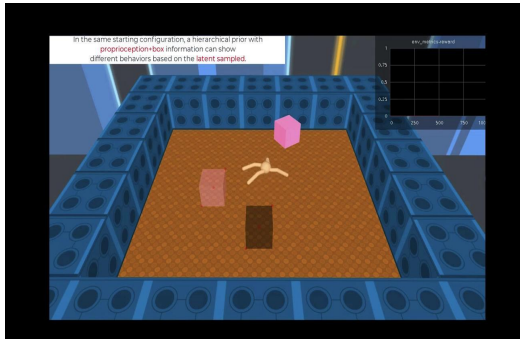
Locomotion and Manipulation

- Task: Move a box to a target (Manipulation) and go to another target (Locomotion)
- Information set:
  - Proprioception
  - Box Locations
  - Target locations
  - Identity of target to move to.

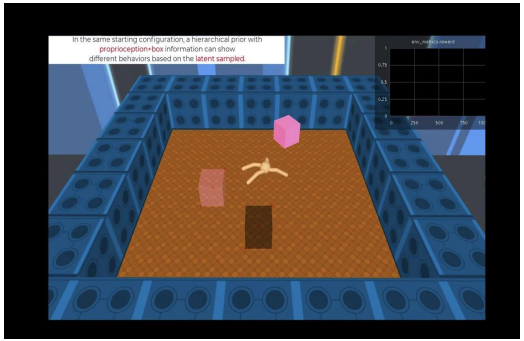
→ **Many choices for information factorization**



Prior with just proprioception

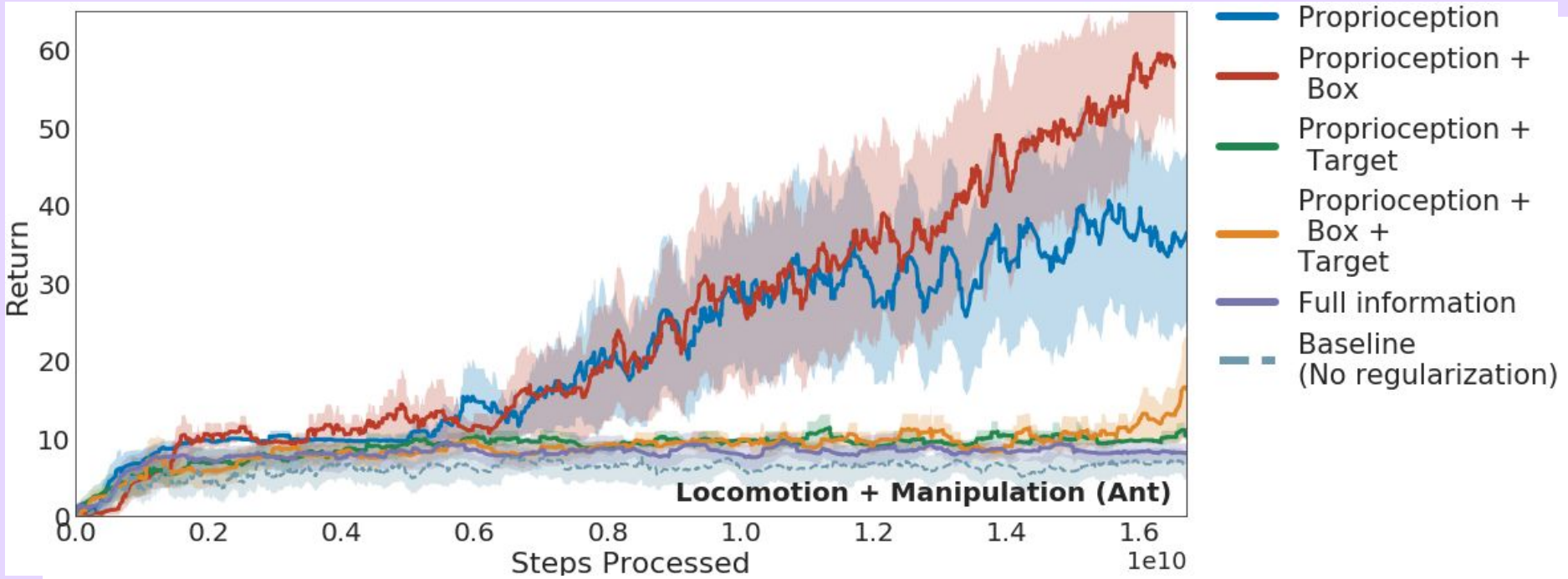


Prior with proprioception and box



Prior with proprioception and targets

Does this help?



Locomotion and Manipulation

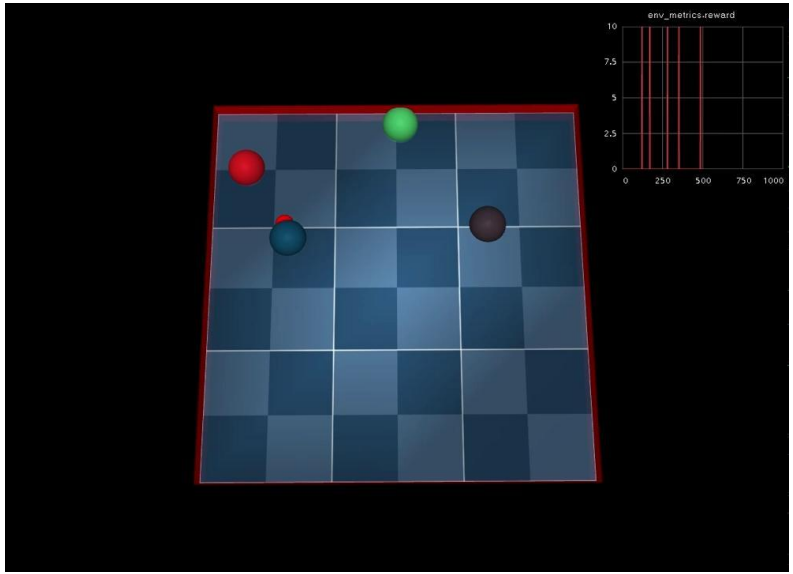
- Moving towards targets and boxes is quite goal-directed behavior.
- More generally control involves repetitive, cyclical patterns.
  - Gaussian policies can capture this temporally correlated behavior.
  - No explicit need for hierarchy

Extending: Task with more temporal structure

- **Goal**: Extend these ideas to tasks with more **temporal structure**
- Gaits and movement patterns involve temporal structure but do not require explicit memory-dependent models
- Can we design a **simple** experiment with this in mind?

The continuous gridworld

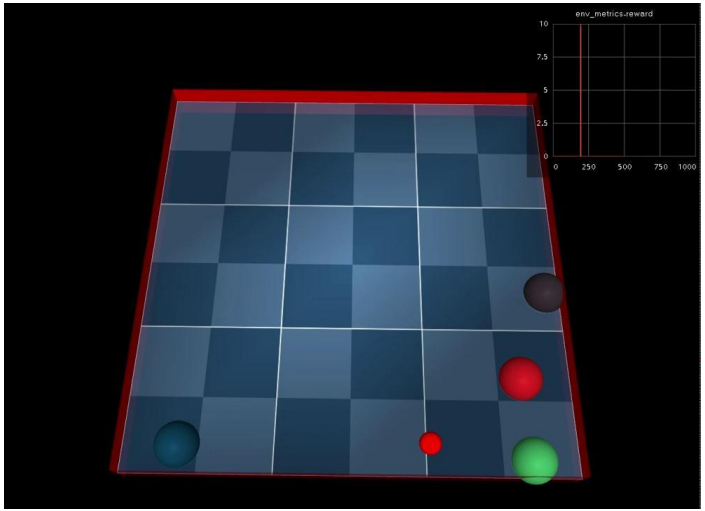
- Sequential Navigation task with Pointmass:
  - 2-DoF body => effectively a continuous gridworld.
  - The goal is to reach different targets
  - At each point of time, the agent is given its location, target locations and **the next target** to visit.



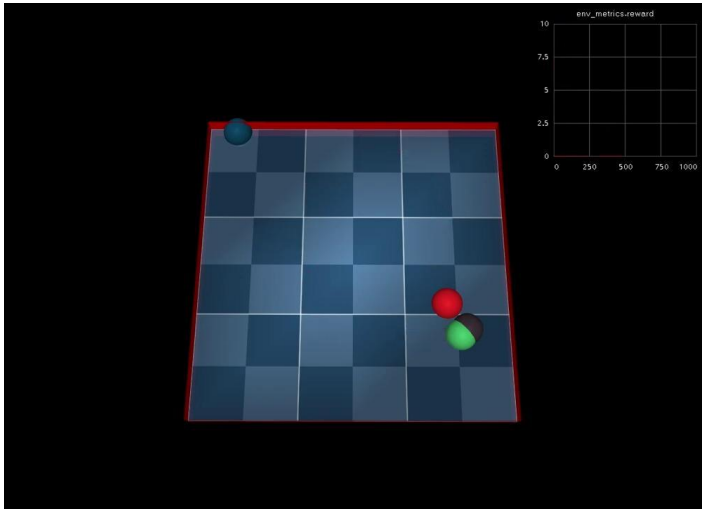
The ‘temporal’ correlation bit

- How can introduce temporally correlated structure here?
  - What if the **next target** choice is **non-uniform?**
  - When at a given target the agent should not be able to predict this.
    - **Idea**: Targets generated under **2nd order Markovian** dynamics
      - Given the **last 2** targets visited, the next target becomes **more** predictable.
      - E.g. If the last two targets in sequence was **blue->green,** the next target is black with **probability say 0.9** and and red with probability **0.1.**
        - **Crucially** if you only know the **last target** was **green** then the next target could be blue, red or black **uniformly at random.**
        - We can do this by appropriately shifting the probabilities for the other combinations (red->green; black->green).
    - Why do we want this?
      - A prior that can **remember** the previous targets may be able to capture these correlations.
      - Allows us to try different **architectural constraints.**
        - Prior with memory (**LSTM)**
        - Prior from before **(MLP).**

Sequential Navigation Results



LSTM prior



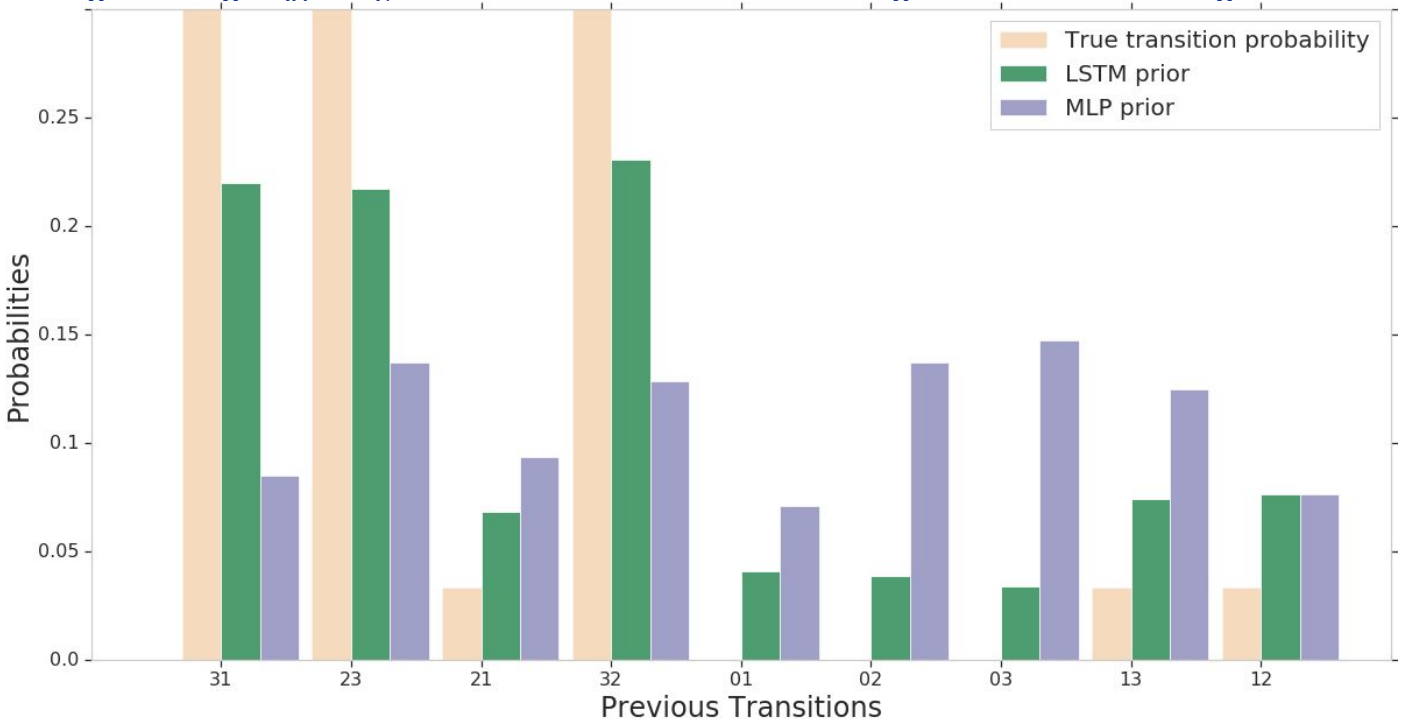
MLP prior

Sequential Navigation Results (Under the hood)

- We can generate trajectories from the **priors** (since they are just policies really) and **study statistics**
    - For a **given target** (green), what is the visitation distribution **given the last two targets**?

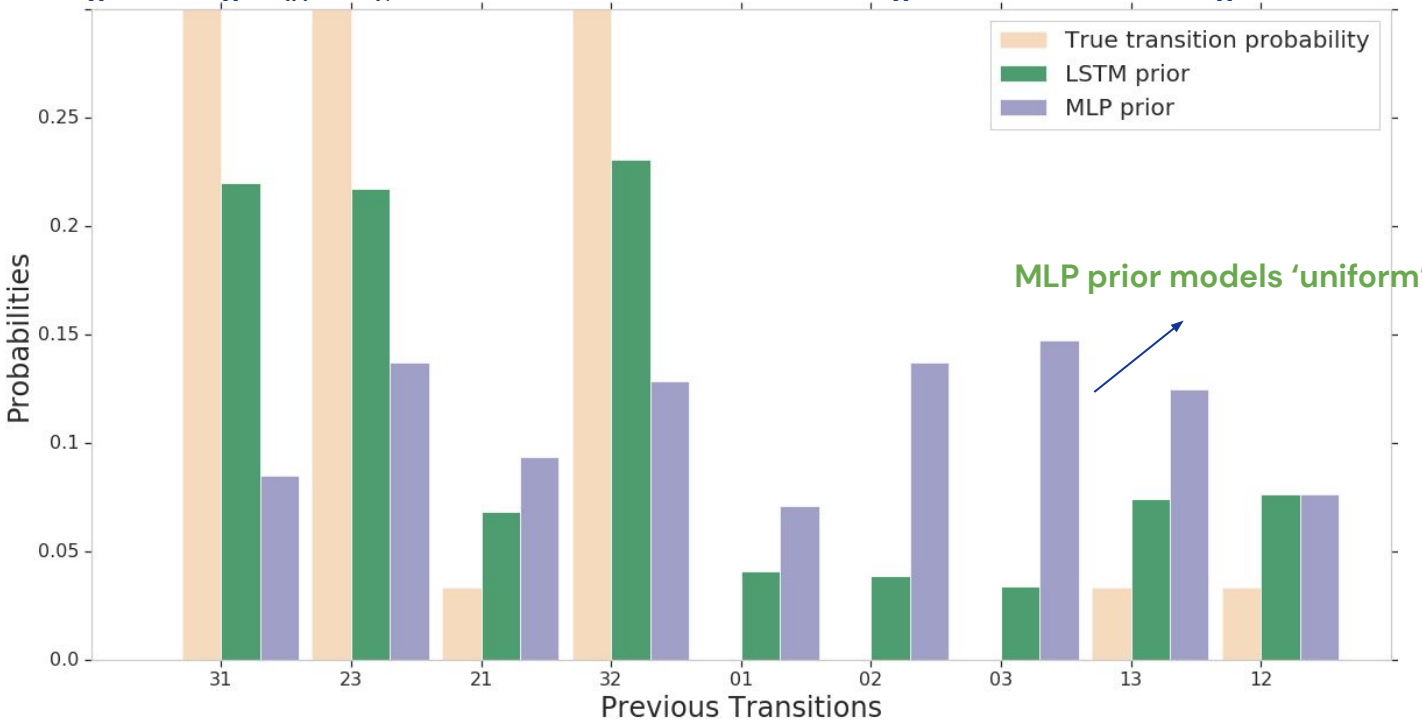
Sequential Navigation Results (Under the hood)

- We can generate trajectories from the **priors** (since they are just policies really) and **study statistics**
  - For a **given target** (green), what is the visitation distribution **given the last two targets**?



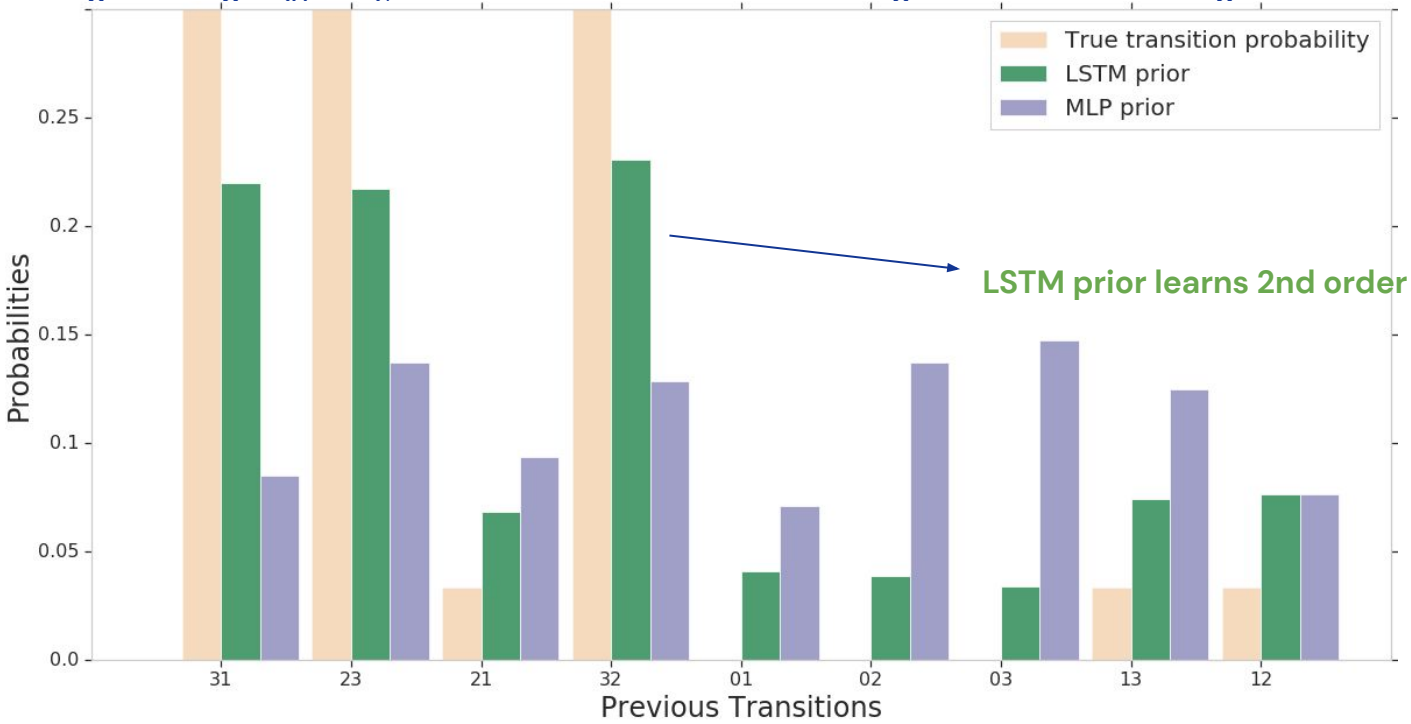
Sequential Navigation Results (Under the hood)

- We can generate trajectories from the **priors** (since they are just policies really) and **study statistics**
  - For a **given target** (green), what is the visitation distribution **given the last two targets**?



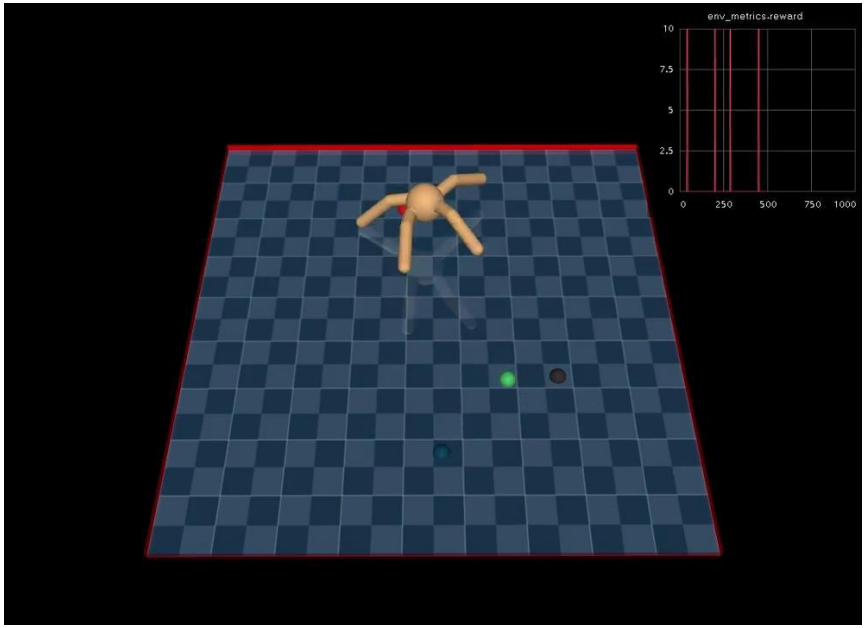
Sequential Navigation Results (Under the hood)

- We can generate trajectories from the **priors** (since they are just policies really) and **study statistics**
  - For a **given target** (green), what is the visitation distribution **given the last two targets**?

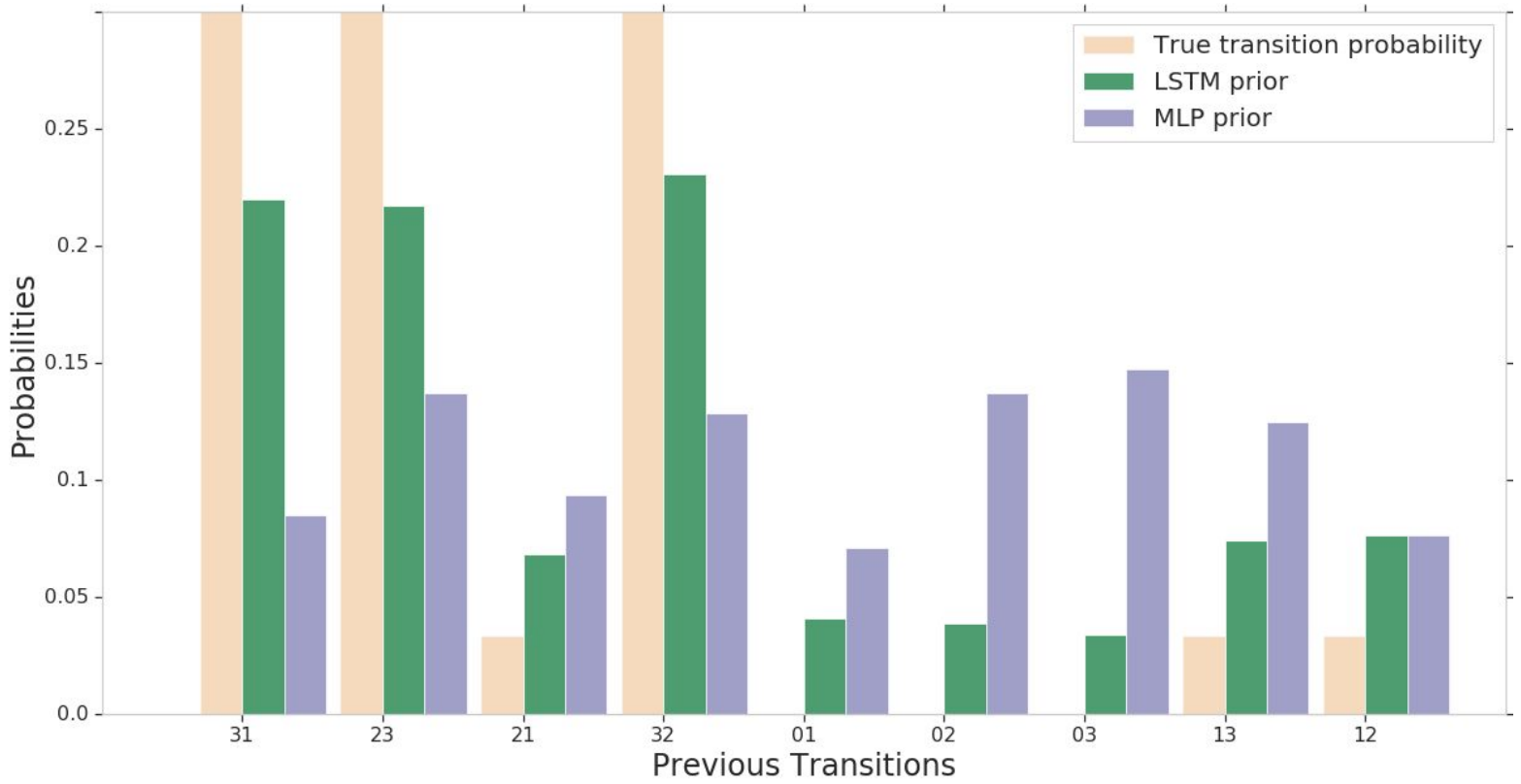


Sequential Navigation (Ant)

- We can even extend this to a more complicated body: Ant
  - The statistics would now need to really extend over time and include body movements!



Sequential Navigation (Ant): Under the hood



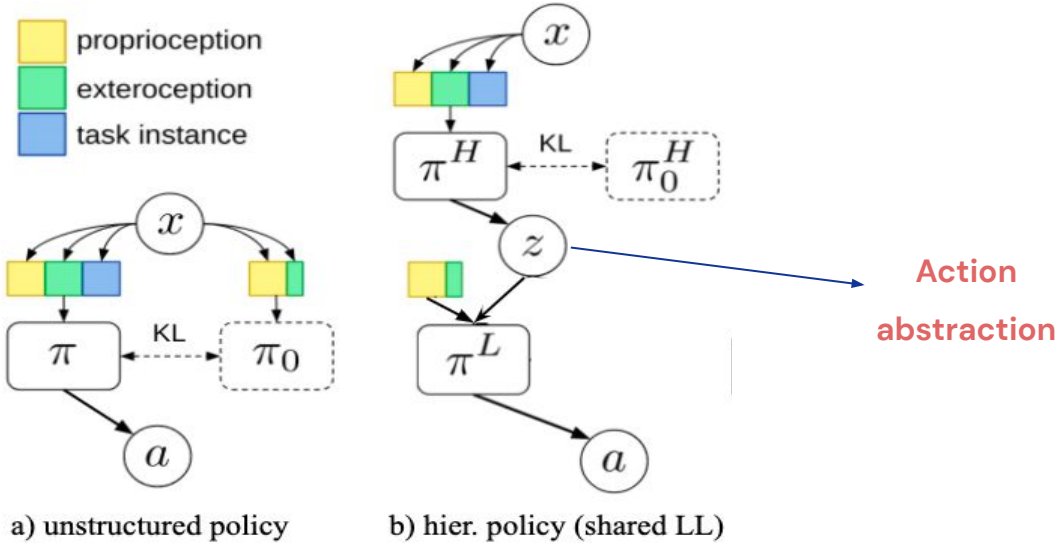
Why is this useful?

- So far everything we have shown is somewhat contrived
  - Specific setups with information asymmetry and temporal structure
- This criticism is fair but misses the higher level point
  - Behaviors are useful statistical patterns that we hope to reuse
  - How useful this is depends on **where and how** we want to use them
  - The models we choose to learn them, whether explicitly or implicitly, can matter.
  - In fact typically learning behaviors has been the purview of Hierarchical reinforcement learning (HRL).
  - All our results so far only looked at **flat, non-hierarchical priors.**
  - Ultimately this is just a slightly different perspective then that is rooted in probabilistic modeling of trajectories.

- Q: What does this perspective mean for hierarchy?

So where is hierarchy needed?

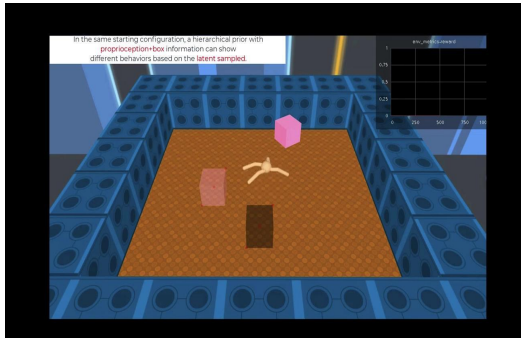
- Hierarchy in this view is a **modeling choice**
- Another view: Hierarchy as a latent-variable model



*Note: Computing the KL with a continuous latent variable model is intractable. We derive a lower bound and use that instead for our experiments.

Locomotion and Manipulation revisited

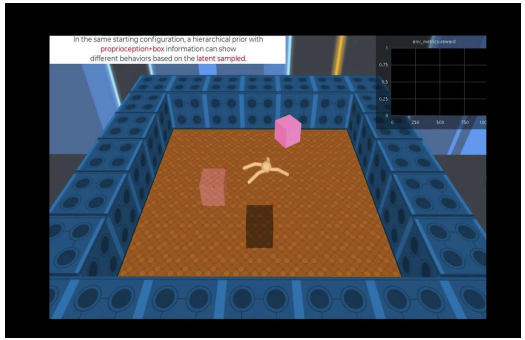
- Example task: Move a box to a target (Manipulation) and go to another target (Locomotion)
  - **Fixed configuration** of box, walker and target positions
  - Hierarchical priors allow more complex, multi-modal behaviors to be captured.



Hierarchical prior

Locomotion and Manipulation revisited

- For a fixed location a hierarchical model is more **expressive.**
  - In a given situation, the optimal may be to go left or right.
  - Unimodal Gaussians cannot capture this.
    - A latent variable model can.



Hierarchical prior

Quantitative analysis

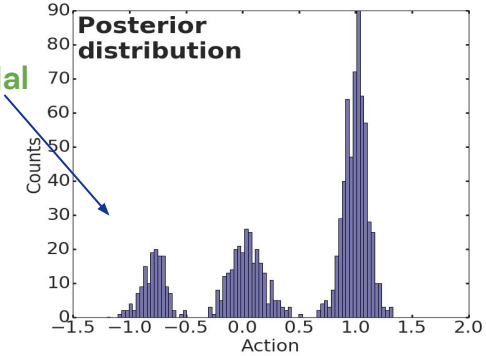
- We can also analyze this a bit more quantitatively.
- For a fixed configuration where the information to the prior is **fixed**, we can generate the distribution for different values of the **goal.**

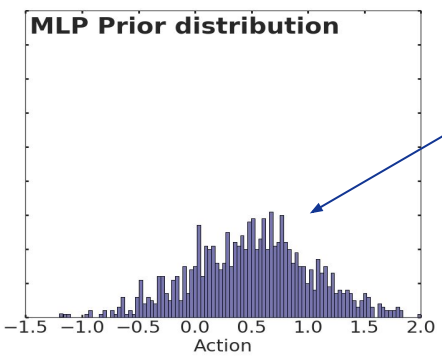
$$\mathcal{L} = \mathbb{E}_{\pi}[\sum_{t} \gamma^t r(s_t, a_t)] - \gamma^t \mathrm{KL}[\pi(a_t|x_t)||\pi_0(a_t|x_t^D)].$$

$$\pi(\cdot|x_t) = \pi(\cdot|x_t^G, x_t^D)$$

Action abstraction

Distribution under posterior is a multi-modal mixture of Gaussians

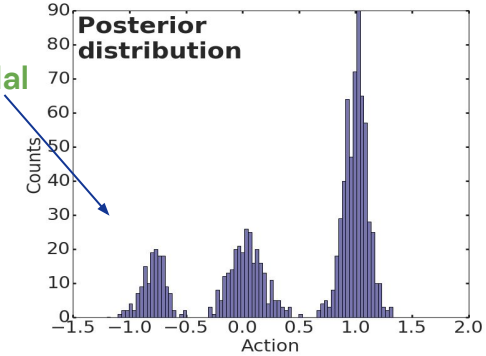


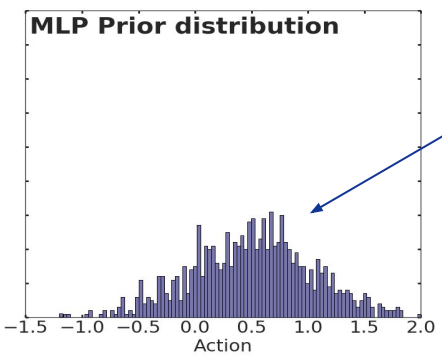


Prior without latent captures ‘mean’ behavior

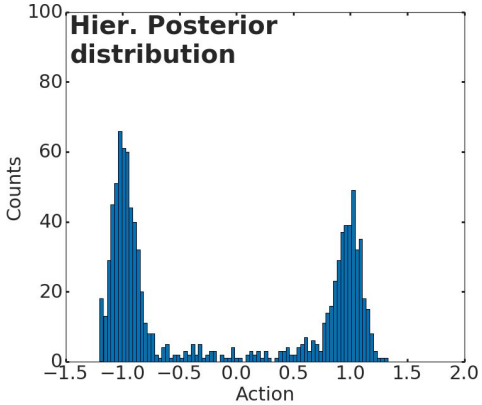
Action abstraction

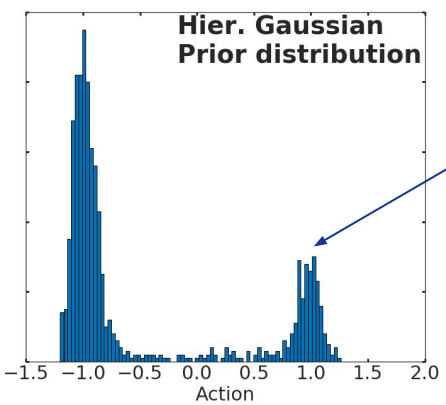
Distribution under posterior is a multi-modal mixture of Gaussians





Prior without latent captures ‘mean’ behavior





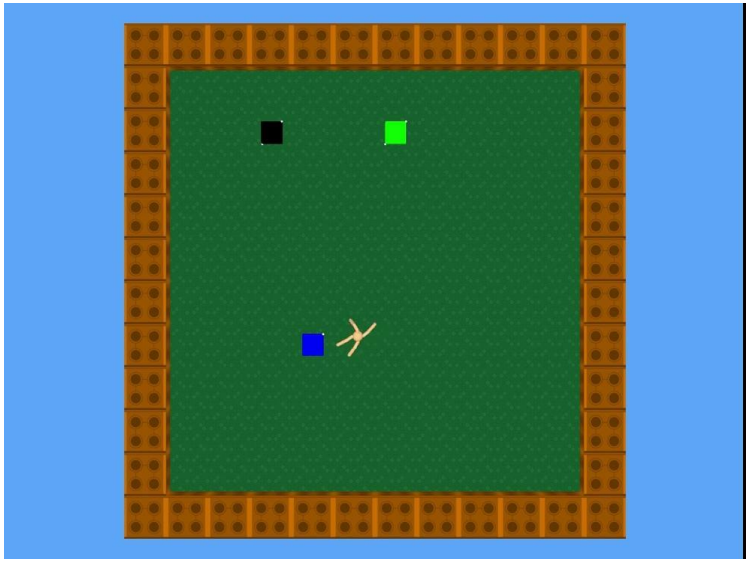
A hierarchical prior can capture this distribution!

Bits I skipped

- A lot more analysis and discussion in the paper that I left out for time.
- A short snippet
  - Movement priors with the humanoid (23-dof)
  - Transfer experiments
  - Experiments with other hierarchical model variants
    - Hierarchies with memory
    - Hierarchical models with separate higher and lower level components
  - Learning curves! (Quantitative analysis)
- Connection to Mutual information based objectives and other ideas in HRL (e.g. options)

Let us revisit the challenge from before

The exploration challenge for control…

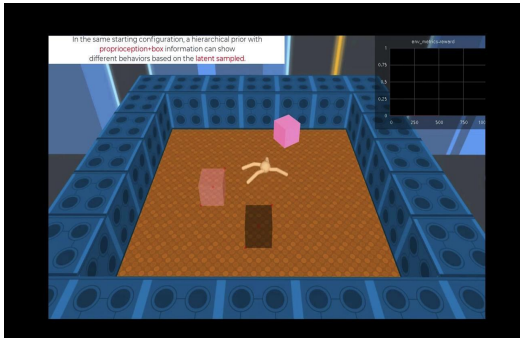


Ant with a ‘random’ Gaussian policy

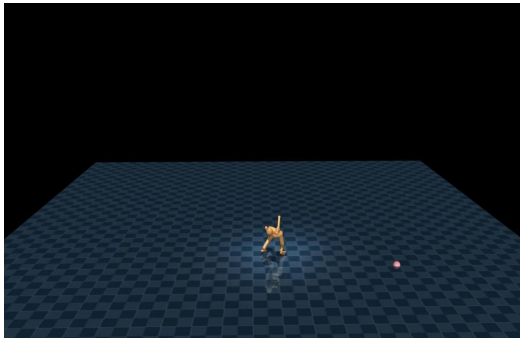
Humanoid with a Gaussian policy

The view with priors

Behavior Priors look a lot better...



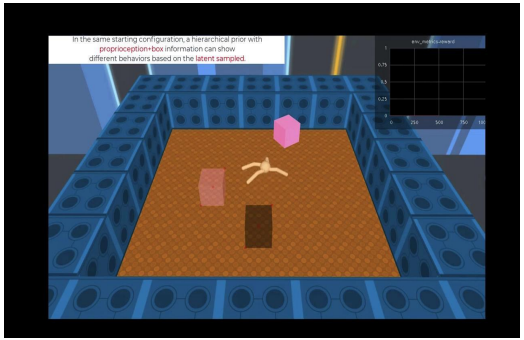
Prior with just proprioception



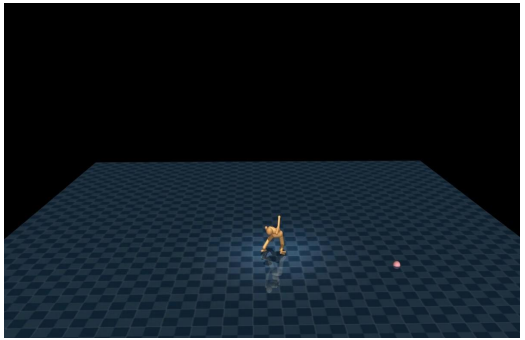
Humanoid proprioceptive prior

The view with priors

Behavior Priors look a lot better... and we can experiment with better data and models

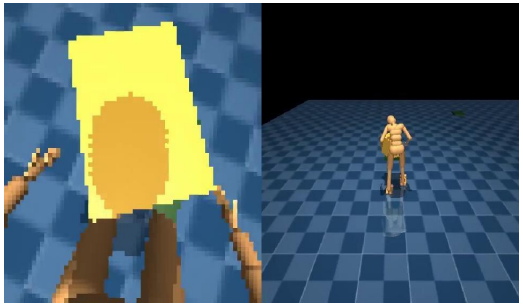


Prior with just proprioception



Humanoid proprioceptive prior

Humanoid with Neural Probabilistic motor primitive (NPMP) prior



*Neural Probabilistic Motor Primitives (Merel et. al.)

Discussion and future research

- Questions?
- Some ideas from KL-regularized RL seem to quite useful in control and robotics:
  - Algorithms like MPO and SAC have gained prominence
  - Offline-RL (Batch-RL) uses concepts very similar to the ‘behavior priors’.
  - Many interesting ideas in HRL like Neural Probabilistic Motor Primitives (NPMP) fit in nicely with this framework.
- Focus here on model-free methods
- Research Questions
  - Can we extract behaviors entirely offline? What representations would be interesting to explore?
  - How can we incorporate models into the priors perspective?
  - What tradeoffs does this introduce? Perhaps temporal consistency in model space adds benefits?
  - …

DeepMind

Thank You!

Outline

- The control challenge
- Method: ‘Priors’ over behavior
- Experiments
- **Overarching view: Related work**
- Discussion

Connection to Information bottleneck

- Consider the following objective instead:

$$\mathcal{L}_{\mathcal{I}} = \mathbb{E}_{\pi}[\sum_{t} \gamma^{t} r(s_t, a_t) - \alpha \gamma^{t} \mathrm{I}[x_t^G; a_t | x_t^D]]$$

Connection to Information bottleneck

- Consider the following objective instead:

$$\mathcal{L}_{\mathcal{I}} = \mathbb{E}_{\pi}[\sum_{t} \gamma^t r(s_t, a_t) - \alpha\gamma^t \mathrm{I}[x_t^G; a_t | x_t^D]]$$

$$\mathrm{I}(X;Y) \equiv \mathrm{H}(X) - \mathrm{H}(X|Y)$$

- Mutual information quantifies the amount of information obtained about one random variable when **observing** another.

Connection to Information bottleneck

- Consider the following objective instead:

$$\mathcal{L}_{\mathcal{I}} = \mathbb{E}_{\pi}[\sum_{t} \gamma^t r(s_t, a_t) - \alpha\gamma^t \mathrm{I}[x_t^G; a_t | x_t^D]]$$

- This objective **penalize** the dependence on actions that contain **information hidden** from the prior.
  - In other words, in this view we prefer actions that are more general than the particular task context.

Connection to Information bottleneck

- Consider the following objective instead:

$$\mathcal{L}_{\mathcal{I}} \geq \mathbb{E}_{\pi}[\sum_{t} \gamma^{t} r(s_t, a_t) - \alpha \gamma^{t} \mathrm{KL}[\pi(a_t|x_t)||\pi_0(a_t|x_t)]]$$

- It turns out this is a lower bound to the objective we are optimizing for!
  - This connects back to the intuition from before.
    - We want to learn policies that are more general than the particular context we're currently in.

Connection to HRL

- Hierarchy as a latent variable model.
- There are more details to this formulation that were skipped:
  - Latent variables only in the prior
  - Latent variables only in posterior
  - Lower bound when latent variables in both
  - Options framework as a probabilistic model
- In the paper though!

Discussion

- Questions?
- The ideas here are quite useful in control and robotics:
  - Algorithms like MPO and SAC have gained prominence
  - Offline-RL (Batch-RL) uses concepts very similar to the ‘behavior priors’.
  - Many interesting ideas in HRL like Neural Probabilistic Motor Primitives (NPMP) fit in nicely with this framework.
- The idea behind these approaches is to build and reuse knowledge.
- However some parts of the field are moving in a different direction:
  - ‘Pure’ methods with fewer inductive biases seem to work better in many domains given an abundance of data.
  - Transformers are now working as well or better than CNNs for vision stacks
  - GPT-3 etc. use large sources of data with the underlying principle seems to be - how far can we get with large models and a ton of data?

DeepMind

Some RL concepts

The reinforcement learning paradigm

Objective: $r_1 + r_2 + r_3 + r_4 + \ldots$

*Slide credit: Nicolas Heess

The reinforcement learning paradigm

Specify the **objective** – not the **solution**

*Slide credit: Nicolas Heess

The *deep* reinforcement learning paradigm

Objective: $r_1 + r_2 + r_3 + r_4 + \ldots$

*Slide credit: Nicolas Heess

RL terminology

Let’s delve into this in a bit more detail and introduce some terminology:

RL terminology

Let's delve into this in a bit more detail and introduce some terminology:

1) Trajectory:

$$\tau = (s_0, a_0, s_1, a_1, s_2, \ldots)$$

RL terminology

Let's delve into this in a bit more detail and introduce some terminology:

1) Trajectory:

$$\tau = (s_0, a_0, s_1, a_1, s_2, \ldots)$$

2) Policy – an action selection rule (deterministic or stochastic) :

$$\pi(a_t|s_t)$$

RL terminology

Let's delve into this in a bit more detail and introduce some terminology:

1) Trajectory:

$$\tau = (s_0, a_0, s_1, a_1, s_2, \ldots)$$

2) Policy - an action selection rule (deterministic or stochastic) :

$$\pi(a_t|s_t)$$

3) Discounted returns and the RL Objective:

$$R(\tau) = \sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \qquad \gamma \in [0, 1)$$

$$\mathcal{J}(\pi) = \mathbb{E}_{\pi}[R(\tau)]$$

RL terminology

Let's delve into this in a bit more detail and introduce some terminology:

1) Trajectory:

$$\tau = (s_0, a_0, s_1, a_1, s_2, \ldots)$$

2) Policy - an action selection rule (deterministic or stochastic) :

$$\pi(a_t|s_t)$$

3) Discounted returns and the RL Objective:

$$R(\tau) = \sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \qquad \gamma \in [0, 1)$$

$$\mathcal{J}(\pi) = \mathbb{E}_{\pi}[R(\tau)]$$

4) Value functions

$$V^{\pi}(s_t) = \mathbb{E}_{\pi}[R(\tau)|s_t]$$

$$Q^{\pi}(s_t, a_t) = \mathbb{E}_{\pi}[R(\tau)|s_t, a_t]$$

RL terminology

Let's delve into this in a bit more detail and introduce some terminology:

1) Trajectory:

$$\tau = (s_0, a_0, s_1, a_1, s_2, \ldots)$$

2) Policy - an action selection rule (deterministic or stochastic) :

$$\pi(a_t|s_t)$$

3) Discounted returns and the RL Objective:

$$R(\tau) = \sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \qquad \gamma \in [0, 1)$$

$$\mathcal{J}(\pi) = \mathbb{E}_{\pi}[R(\tau)]$$

4) Value functions

$$V^{\pi}(s_t) = \mathbb{E}_{\pi}[R(\tau)|s_t]$$

$$Q^{\pi}(s_t, a_t) = \mathbb{E}_{\pi}[R(\tau)|s_t, a_t]$$

5) Exploration

RL terminology

Let's delve into this in a bit more detail and introduce some terminology:

1) Trajectory:

$$\tau = (s_0, a_0, s_1, a_1, s_2, \ldots)$$

2) **Policy - an action selection rule (deterministic or stochastic) :**

$$\pi(a_t|s_t)$$

3) Discounted returns and the RL Objective:

$$R(\tau) = \sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \qquad \gamma \in [0, 1)$$

$$\mathcal{J}(\pi) = \mathbb{E}_{\pi}[R(\tau)]$$

4) **Value functions**

$$V^{\pi}(s_t) = \mathbb{E}_{\pi}[R(\tau)|s_t]$$

$$Q^{\pi}(s_t, a_t) = \mathbb{E}_{\pi}[R(\tau)|s_t, a_t]$$

5) Exploration

RL terminology

Let's delve into this in a bit more detail and introduce some terminology:

1) Trajectory:

$$\tau = (s_0, a_0, s_1, a_1, s_2, \ldots)$$

2) **Policy - an action selection rule (deterministic or stochastic) :**

$$\pi(a_t|s_t)$$

3) Discounted returns and the RL Objective:

$$R(\tau) = \sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \qquad \gamma \in [0, 1)$$

$$\mathcal{J}(\pi) = \mathbb{E}_{\pi}[R(\tau)]$$

4) **Value functions**

$$V^{\pi}(s_t) = \mathbb{E}_{\pi}[R(\tau)|s_t]$$

$$Q^{\pi}(s_t, a_t) = \mathbb{E}_{\pi}[R(\tau)|s_t, a_t]$$

5) Exploration

Useful resources to get started with RL

1. Richard Sutton and Andrew Barto's Reinforcement Learning textbook
2. David silver's course on UCL Reinforcement Learning
3. RL course at UC Berkeley
4. Emma Brunskill's theory focused course at CMU
5. DeepMind's ACME framework for RL
6. Stable baselines: A collection of baseline RL agents

DeepMind

Additional details

Experimental Details

- Method can be incorporated into **any** RL algorithm
- We focus on SVG-0 with RETRACE (RSO)
- Setup:
  - 32 actors
  - 1 learner
  - Replay buffer
  - 5 seeds per run
  - 2 * 5 seeds for transfer
- Focus: Qualitative results (w/ some quantitative analysis)
- Tasks: Sparse reward
  - Less hand-engineering