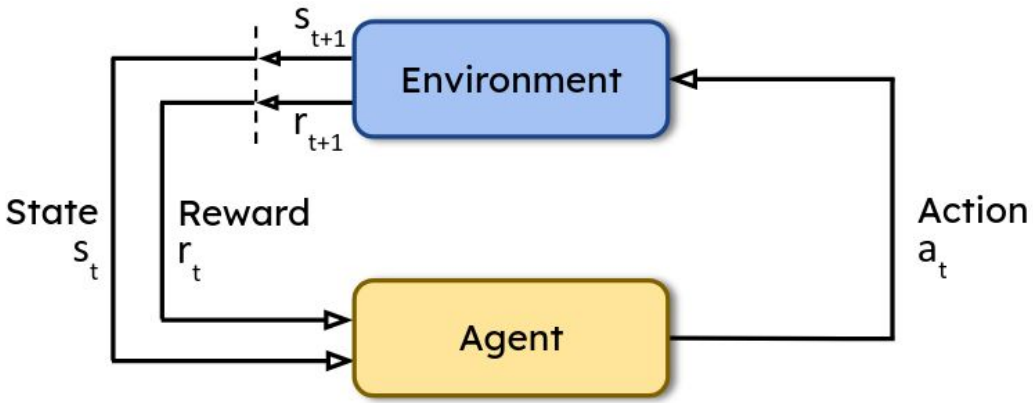
Lecture 15: Model Based Reinforcement Learning

B. Ravindran

Model-free RL

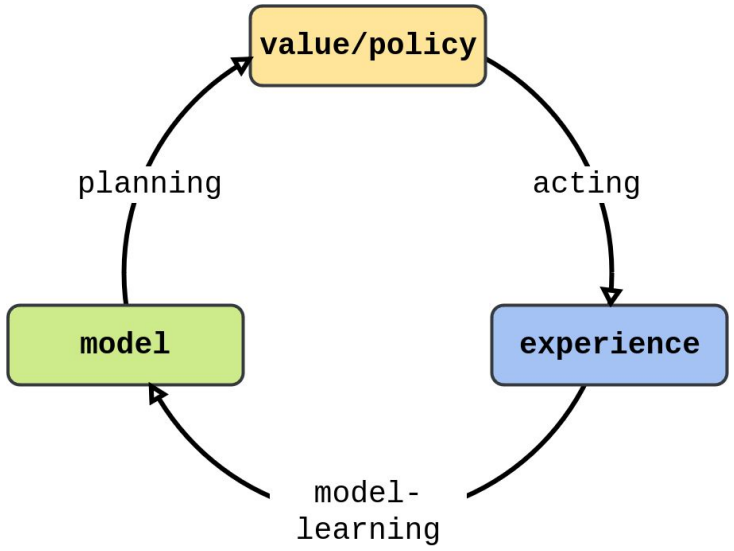
- No model of the environment
- Learn value function and/or policy directly from experience
- Experience collected through interaction with the environment



Model-based RL

- What if we can learn the dynamics of the environment?

- Learn a model of the environment dynamics
- Generate samples using the model
- Learn/plan using those samples



Model-based RL

- Advantages:
  - Can efficiently learn model by supervised learning methods
  - Can reason about model uncertainty
  - Much more sample-efficient than model-free methods
  - Transferability and generalization

- Disadvantages:
  - Additional source of approximation error in model learning
  - Poor model learning can lead to policies that perform suboptimally in the real environment

The Model

- Parameterized way of representing an MDP
- Suppose model is parameterized by η
- A model can represent state transitions and rewards as follows:

$$S_{t+1} \sim \mathcal{P}_\eta(S_{t+1} \mid S_t, A_t)$$
$$R_{t+1} = \mathcal{R}_\eta(R_{t+1} \mid S_t, A_t)$$

- We typically assume conditional independence between next states and rewards

$$\mathbb{P}\left[S_{t+1}, R_{t+1} \mid S_t, A_t\right] = \mathbb{P}\left[S_{t+1} \mid S_t, A_t\right] \mathbb{P}\left[R_{t+1} \mid S_t, A_t\right]$$

Learning The Model

- Learnt using experiences collected from the environment
- Learning the model is a supervised learning problem:

$$
\begin{aligned}
S_1, A_1 &\rightarrow R_2, S_2 \\
S_2, A_2 &\rightarrow R_3, S_3 \\
&\vdots \\
S_{T-1}, A_{T-1} &\rightarrow R_T, S_T
\end{aligned}
$$

Benefits of Model-based RL

- Models can be used to improve:
  - Sample Efficiency
  - Exploration
  - Asymptotic Performance
  - Transfer
  - Safety and Explainability

Planning Using The Model

- Learnt using experiences collected from the environment
- Learning the model is a supervised learning problem:

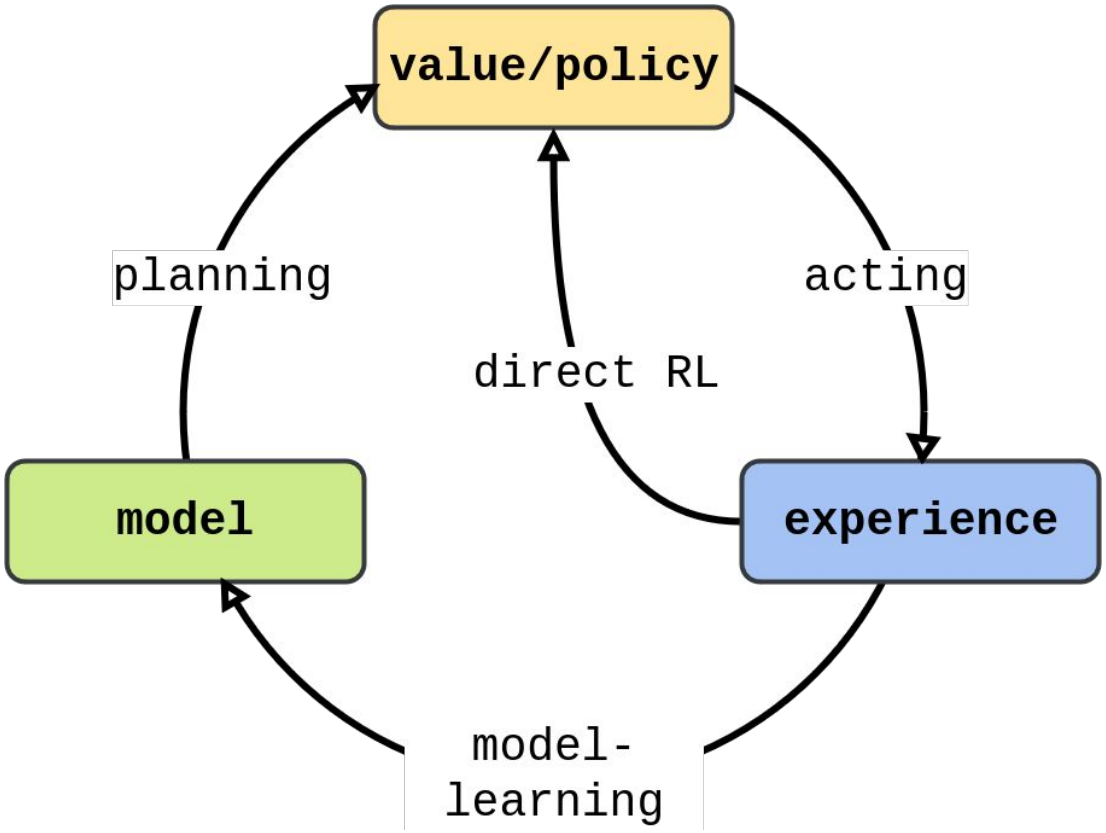
$$
\begin{aligned}
S_1, A_1 &\rightarrow R_2, S_2 \\
S_2, A_2 &\rightarrow R_3, S_3 \\
&\vdots \\
S_{T-1}, A_{T-1} &\rightarrow R_T, S_T
\end{aligned}
$$

- Apply model-free RL to samples
  - e.g., MC-control, SARSA, Q-Learning

Planning Using The Model

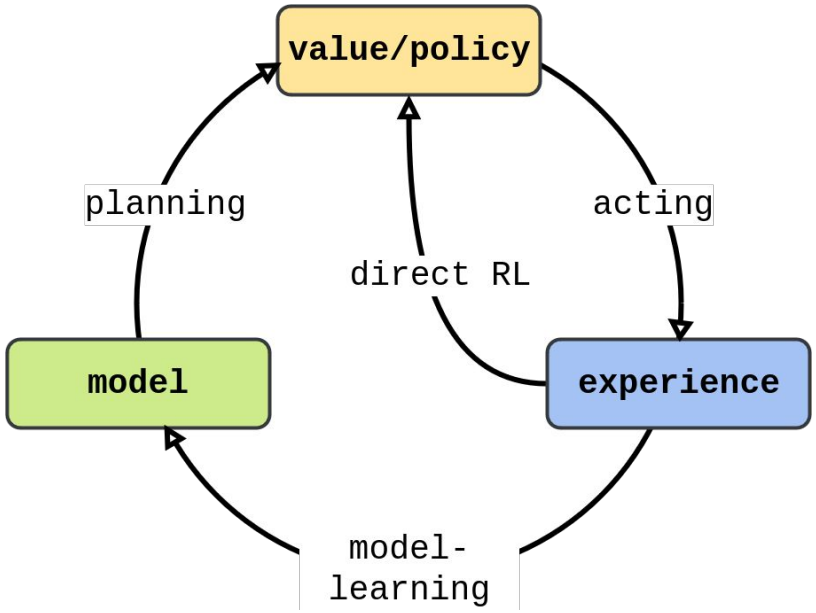
1. Interact with the environment
2. Learn the model
3. Use the model to generate experiences
4. Use the simulated experiences to train your RL algorithm of choice
5. Repeat steps 1 to 4 till convergence

Dyna: Integrating Learning & Planning



Dyna: Integrating Learning & Planning

- Sample-based Planning:
  - Learn a model from real experience
  - Plan using simulated experience
- Dyna:
  - Learn a model from real experience
  - Learn and plan from real and simulated experience



Dyna-Q

Initialize $Q(s,a)$ and $Model(s,a)$ for all $s \in \mathcal{S}$ and $a \in \mathcal{A}(s)$

Loop forever:

(a) $S \leftarrow$ current (nonterminal) state

(b) $A \leftarrow \varepsilon$-greedy(S,Q)

(c) Take action A; observe resultant reward, R, and state, S'

(d) $Q(S,A) \leftarrow Q(S,A) + \alpha\big[R + \gamma \max_a Q(S',a) - Q(S,A)\big]$

(e) $Model(S,A) \leftarrow R, S'$ (assuming deterministic environment)

(f) Loop repeat n times:

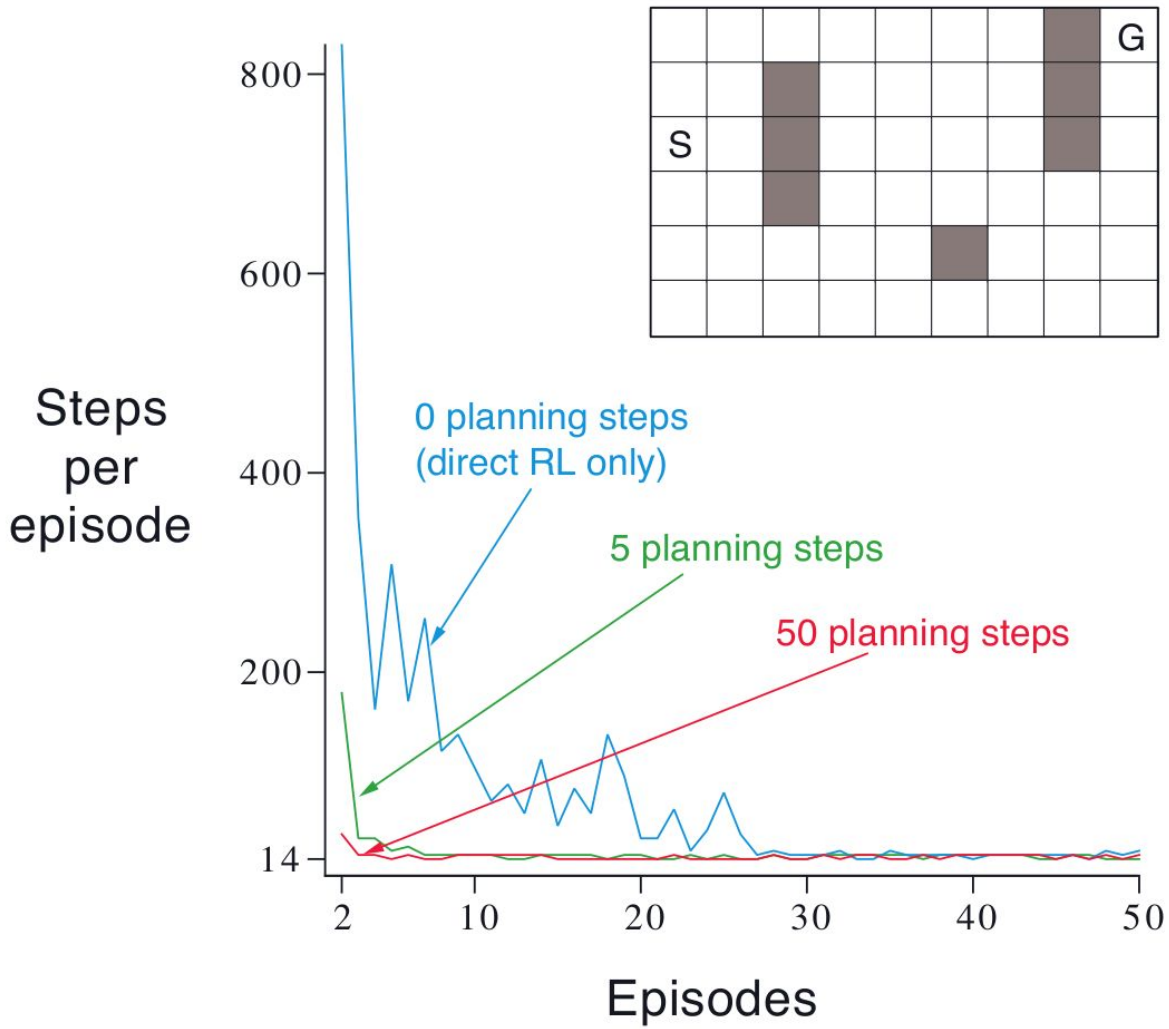
$S \leftarrow$ random previously observed state

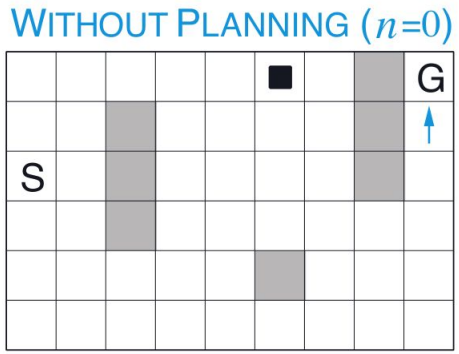
$A \leftarrow$ random action previously taken in S

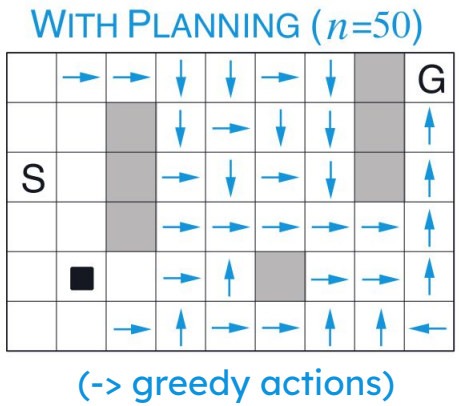
$R, S' \leftarrow Model(S,A)$

$Q(S,A) \leftarrow Q(S,A) + \alpha\big[R + \gamma \max_a Q(S',a) - Q(S,A)\big]$

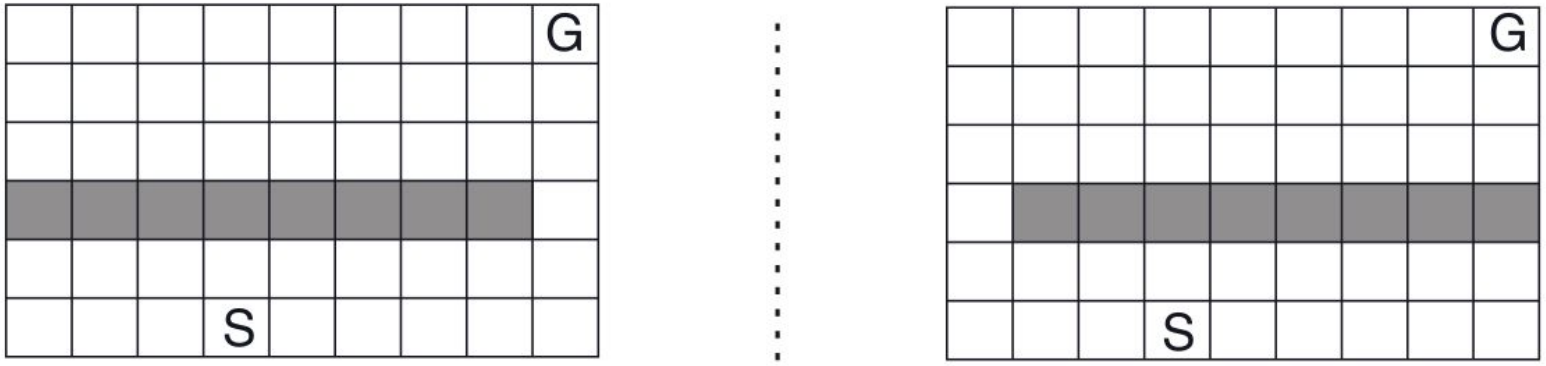
Dyna-Q Learning







Dyna-Q +



- Maze changes dynamically at a certain timestep t
- At t, a shortcut to G will open as shown (right)
- Will a Dyna-Q agent be able to find the optimal solution?
- Agent starts at S and needs to reach G

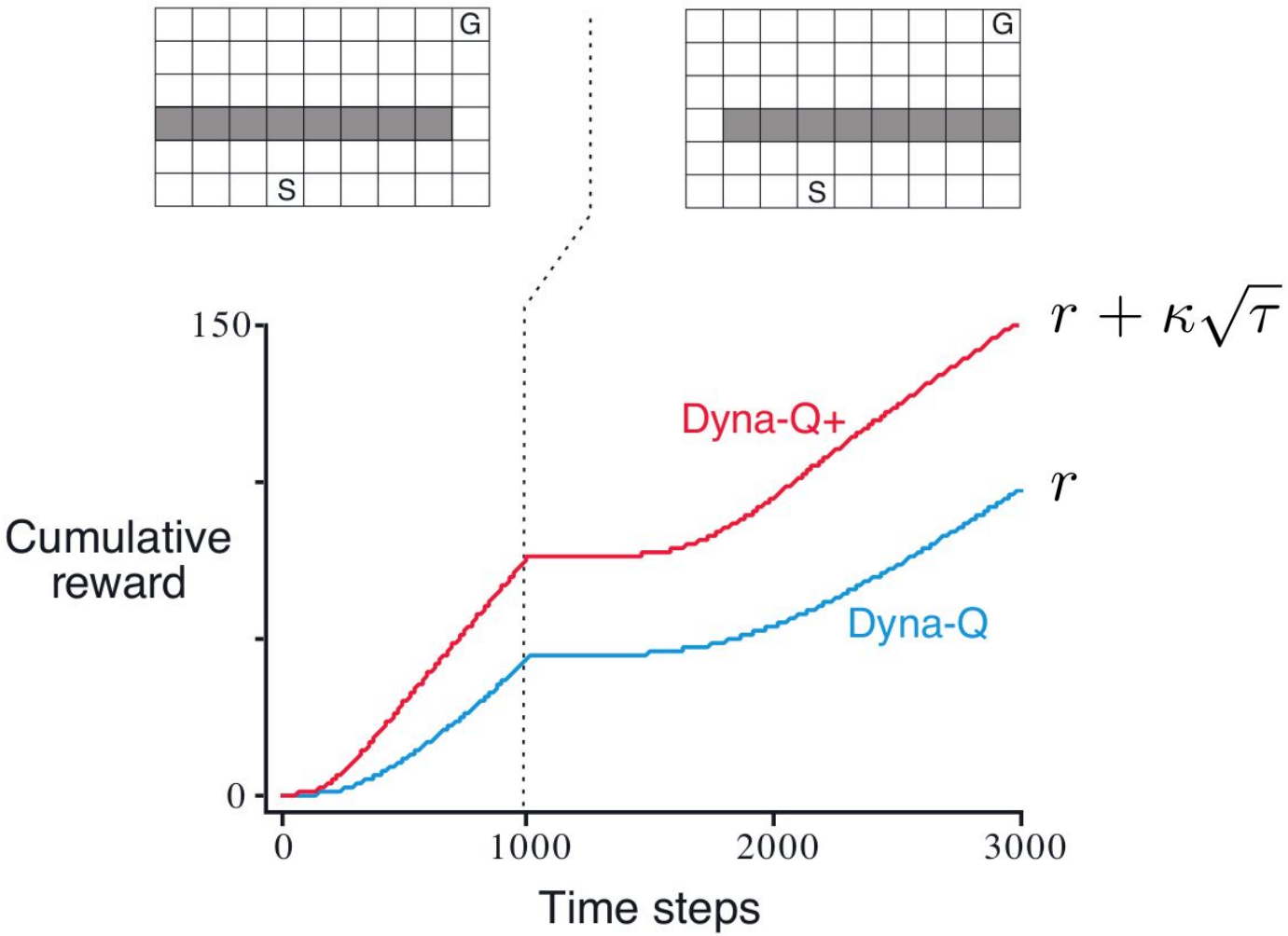
Dyna-Q +

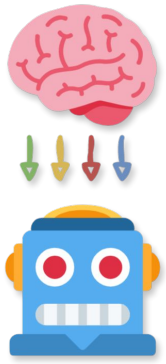
- Solution: Dyna-Q+
- Uses an “exploration bonus”
- Keeps track of time since each state-action pair was tried in a real interaction with environment

$$r + \kappa\sqrt{\tau}$$

- An extra reward is added for transitions caused by state-action pairs related to how long ago they were tried: the longer unvisited, the more reward for visiting

Dyna-Q +


G
S
G
S
150
0
Cumulative reward
0
1000
2000
3000
Time steps
Dyna-Q+
Dyna-Q
$r + \kappa\sqrt{\tau}$
r



MBRL Applications & Developments

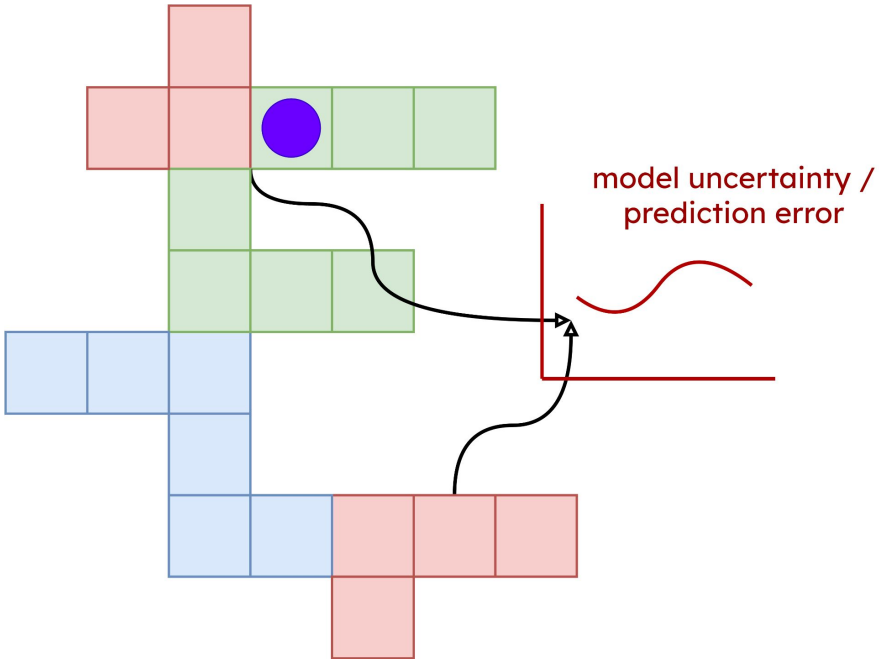
MBRL - Exploration

- Redecide on their exploratory decision at every timestep
- May make an exploratory decision in a state, but decide to undo it at the next timestep. eg. Epsilon-Greedy
- Model-based RL can be used for potentially better exploration strategies that do not cause such “jitter”

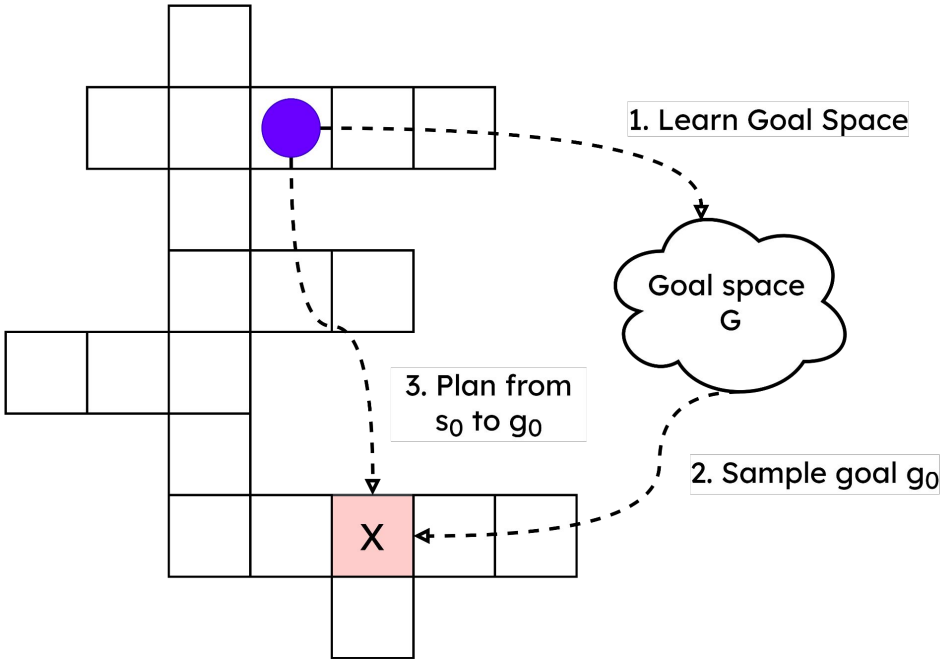
- Intrinsic motivation: Model-based exploration directed at novel or highly uncertain states
  - **Knowledge-based:** Every state gets associated with an intrinsic reward based on local characteristics
  - **Competence-based:** Identify goal states that capture directions of variation in the domain and try to reach them

MBRL - Exploration

- **Knowledge-based:** Every state gets associated with an intrinsic reward based on local characteristics
- **Competence-based:** Identify goal states that capture directions of variation in the domain and try to reach them



Knowledge-based intrinsic motivation



Competence-based intrinsic motivation

MBRL - Exploration

- **Knowledge-based**
- Prioritizes states for exploration when they provide new information about the MDP
- Commonly uses a specific intrinsic reward, which is then propagated together with the extrinsic reward

$$r_t(s, a, s') = r^e(s, a, s') + \eta \cdot r^i(s, a, s')$$

- Novelty eg. Bayesian Exploration Bonus (Kolter and Ng, 2009)

$$r^i(s, a, s') \propto 1/(1 + n(s, a))$$

- Recency eg. Dyna-Q+
- Intrinsic rewards may also help overcome non-stationarity
- A combination of multiple intrinsic rewards may also be used

MBRL - Exploration

- **Competence-based IM**
- Performs exploration based on “learning progress”
- We may have visited a state often, which would make it uninteresting for knowledge-based IM
- If we still get better/faster at actually reaching this state, i.e., we still make learning progress

- Competence-based IM aims to improve learning progress
- Generates an automatic “curriculum” of tasks, guided by learning progress

MBRL - Exploration

- **Competence-based IM**

- A competence-based IM strategy could be as follows
  1. Learn how to select a set of states with high potential learning progress
  2. Sample from the set eg. sample a state with highest potential for learning progress
  3. Attempt to reach the sampled state .eg. Goal-conditioned Value Functions

- Apart from IM, hierarchical methods can also be used for model-based exploration

MBRL - Performance

- It maybe be argued that model-based RL leads to worse asymptotic performance than model-free methods
- However, empirical success of AlphaZero, MuZero etc. suggests otherwise

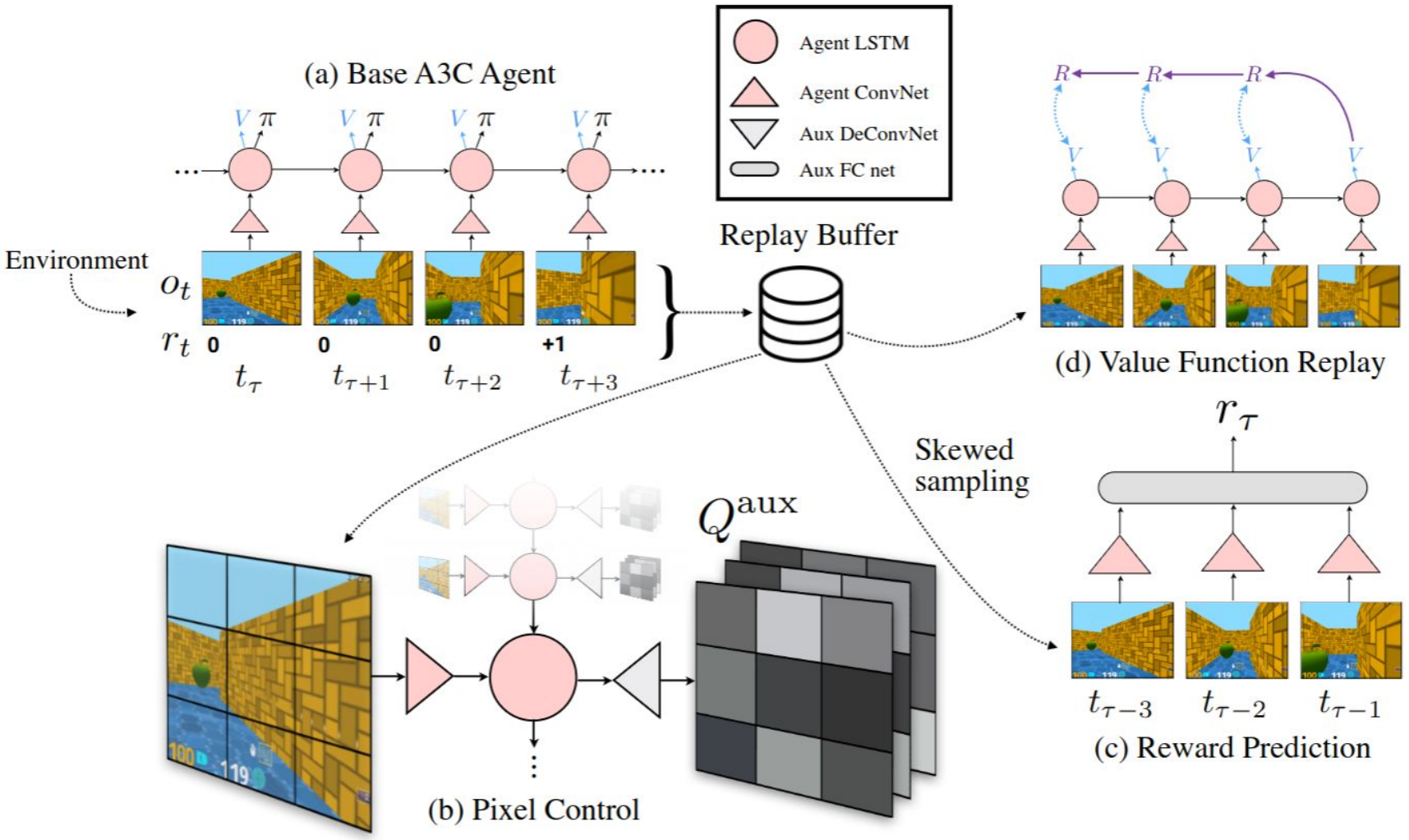
- With a perfect (or good) model, model-based RL may actually lead to better (empirical) asymptotic performance

MBRL - UNREAL

- Can we use models to learn better representations for better performance?

- UNREAL: Reinforcement Learning with Unsupervised Auxiliary Tasks
  - Model prediction used for computing auxiliary objectives
  - All objectives share a common representation

MBRL - UNREAL


(a) Base A3C Agent
Agent LSTM
Agent ConvNet
Aux DeConvNet
Aux FC net
Environment
o_t
r_t
0
0
0
+1
t_τ
$t_{\tau+1}$
$t_{\tau+2}$
$t_{\tau+3}$
Replay Buffer
(d) Value Function Replay
Skewed sampling
r_τ
$t_{\tau-3}$
$t_{\tau-2}$
$t_{\tau-1}$
(c) Reward Prediction
Q^{aux}
(b) Pixel Control

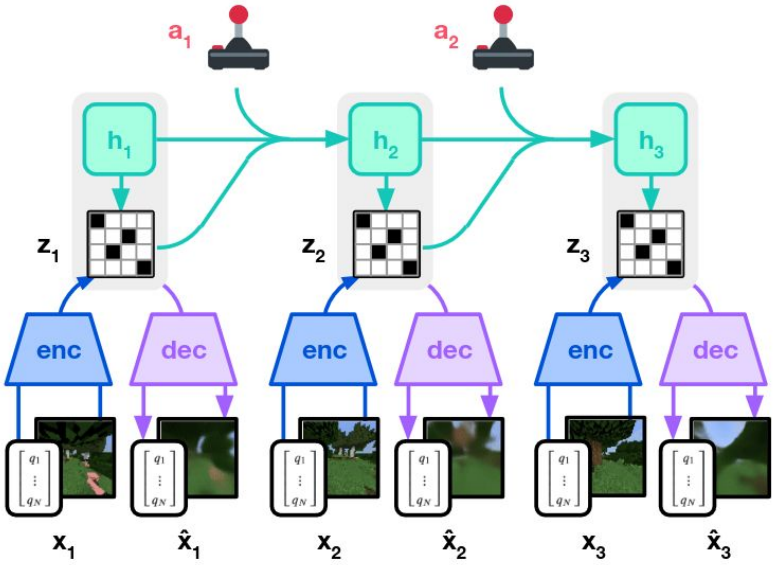
MBRL - UNREAL

Components:

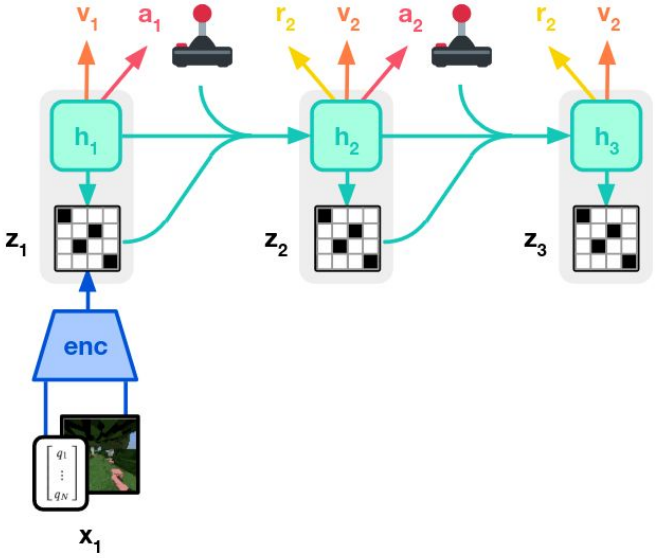
- Base A3C Agent: The component that uses A3C's on-policy training
- Auxiliary Control Tasks: Additional tasks preset by the user giving pseudo-rewards to the agent for specific behavior
- Auxiliary Reward Tasks: Additional reward prediction tasks that help extract relevant features from the environment
- Value Function Replay: Additional off-policy training for the value function

MBRL - Latent Variable Models

- Recent advances like Dreamer (Hafner et. al.) and stochastic Latent Actor Critic (Lee et. al.) leverage variational inference to learn latent variable models of environment dynamics.



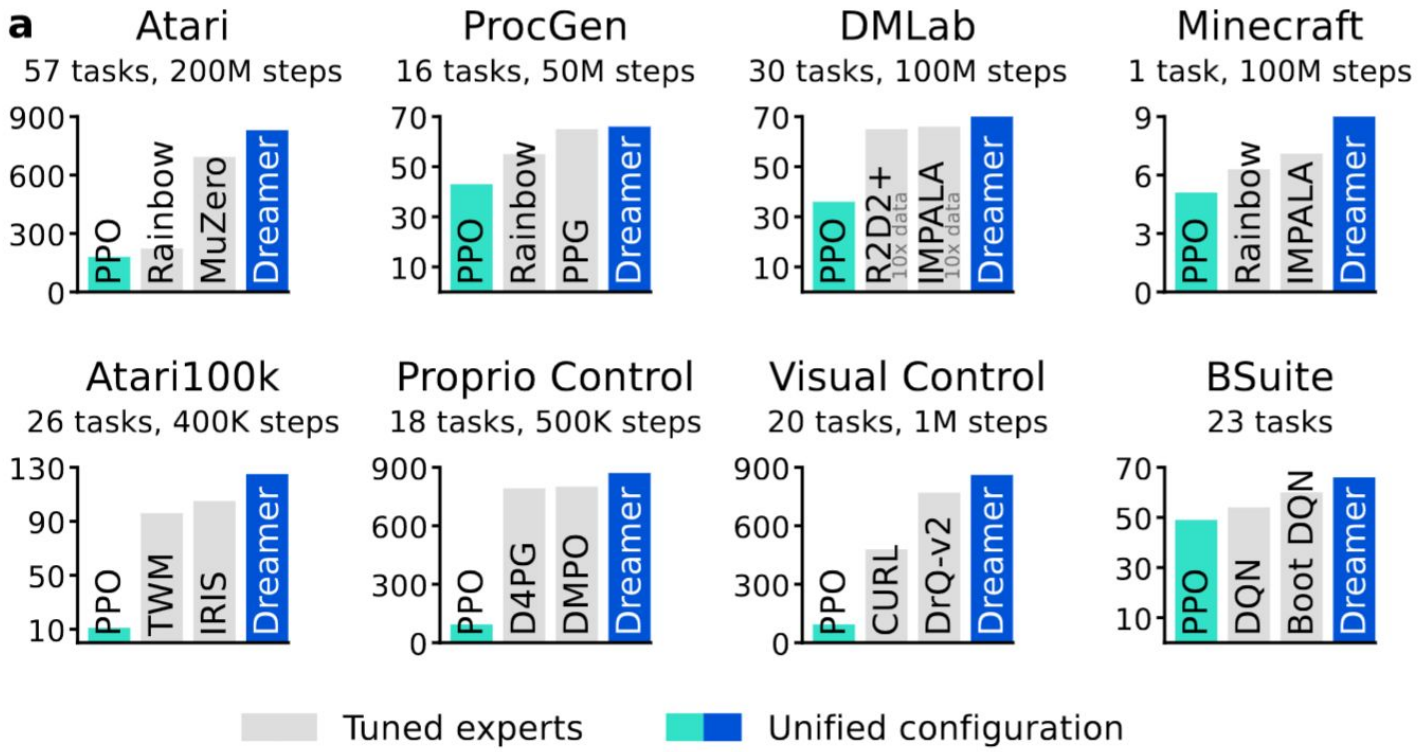
(a) World Model Learning



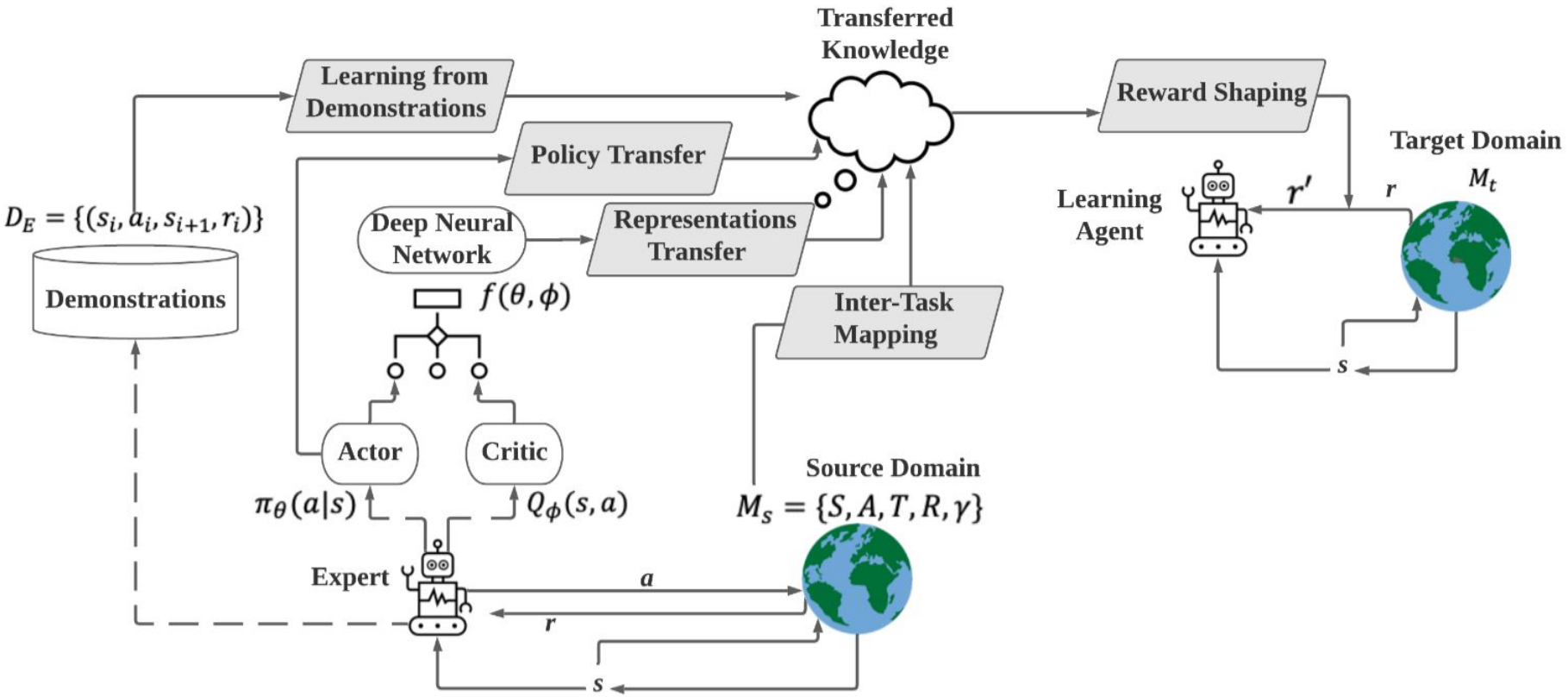
(b) Actor Critic Learning

MBRL - Dreamer v3

- Dreamer-v3 outperforms specialized methods across over 150 diverse tasks *using fixed hyperparameters*



MBRL - Transfer

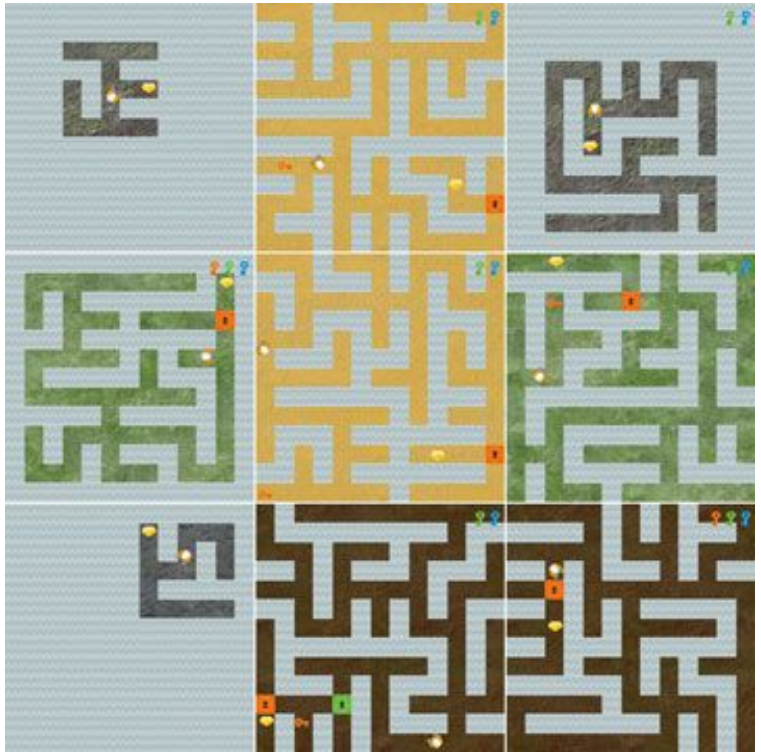


Re-use information from a source task to speed-up learning on a new task

MBRL - Transfer

Transfer of a dynamics model

- Similar dynamics function but different reward function, eg. new level in video game
- Slightly changed transition dynamics, eg. transfer from simulation to real-world tasks

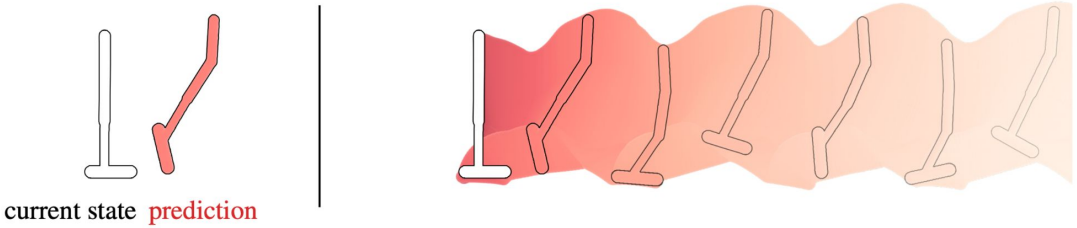


Procgen Benchmark: Different levels of the same game

MBRL - Transfer

Similar dynamics with different reward

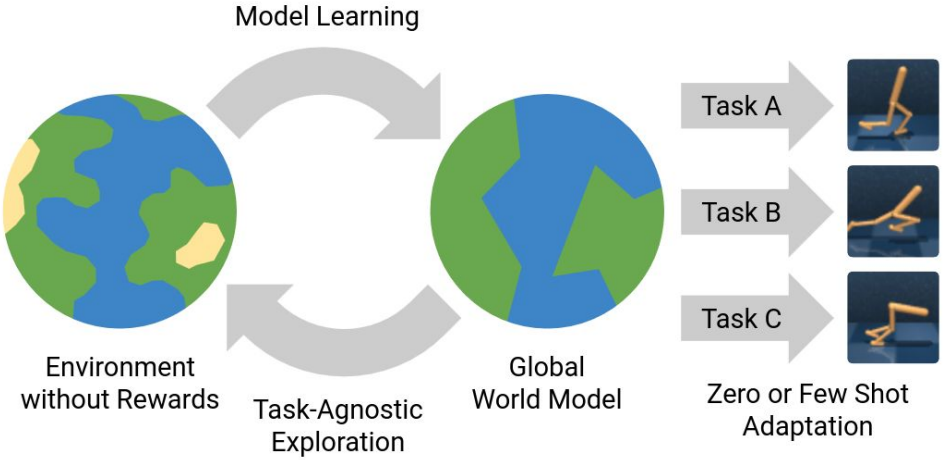
- Can be formulated as a multi-objective RL problem
- A multi-objective MDP has a single dynamics function but multiple reward functions
- These rewards can be combined in different ways, each of which lead to a new task specification
- Successor representations: Another method for changing reward functions
- Summarize the model in the form of future state occupancy statistics



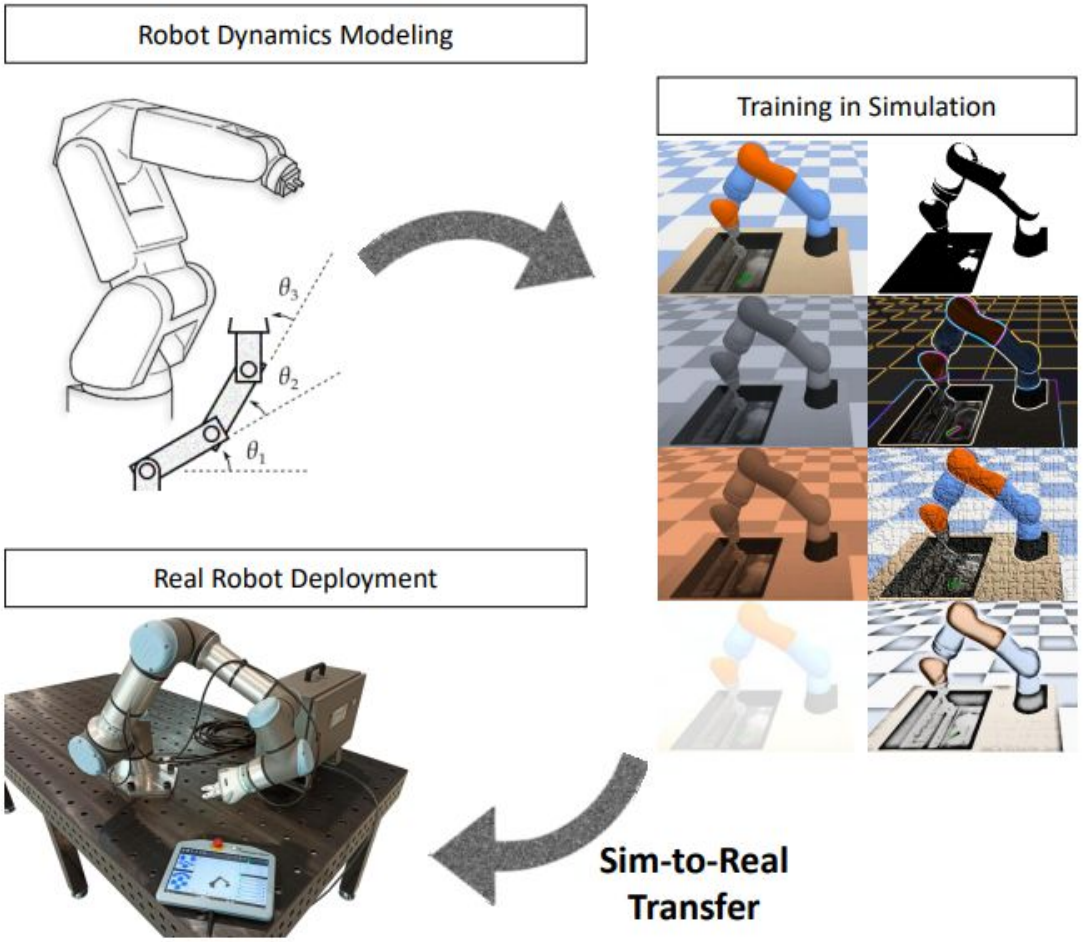
MBRL - Transfer

Slightly changed transition dynamics

- Simulation-to-real transfer is popular in robotics
- Learn a global neural network initialization that can quickly adapt to new tasks eg. Plan2Explore
- Multi-task learning: Learn a distribution over the task space. When a new task comes in, we may quickly identify in which cluster of known tasks (dynamics models) it belongs



MBRL - Sim2Real



MBRL - Safety

- Safety is an important issue, especially when learning on real-world systems
- Ex: With ε-greedy exploration it is easy to break a robot before any learning takes place
- Model-based learning has been used for safety:
  - Given a 'safe region' of the current policy, explores while ensuring return to safe region if necessary (Berkenkamp et al.)
  - Maintain two policies using two models:
    - Use the first model for exploration
    - The second model has uncertainty bounds and is used for verification of the safety of the policy of first model

MBRL - Explainability

- Relatively recent sub-field in RL and builds on model-based methods:
  - Model reconciliation for explicable and legible robot behavior

- Explainability is now widely regarded as a crucial prerequisite for AI to enter society