Model-based RL and Abstraction

Frans A. Oliehoek
TU Delft→ EEMCS → Intelligent Systems → Sequential Decision Making

CWI, March 25, 2025

Today: model-based RL

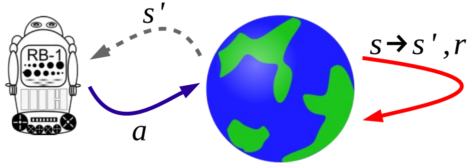
- Fully deep-learning based
- A comment on abstraction

Other MBRL, but not today:

- Learning partial local models (“Influence-based abstraction”) [e.g., Suau et al. ‘22 NeurIPS]
- Bayesian model-based RL for POMDPs [e.g., Katt et al. ‘22 AAMAS]
- Offline model-based RL with confounding [Azizi et al. ‘24 EWRL]
- Does MuZero learn good models? [He et al. 2024 ECAI]

Why RL? Real World is sequential

- Sequential decision making problems
  - actions can have long term effects
  - Markov decision process



What configuration to select for the traffic lights?

Picture by Ahmed Rabea

MDP: <S,A,T,R>

- ►*S – set of states*
- ►*A* – set of actions
- ►transitions: *T(s'|s,a)*
- ►rewards: *R(s,a)*

What is model-based RL?

(or “some terminology to sure we are on the same page”)

RL Nomenclature

Terminology in RL sometimes confusing…

- model available → ‘planning’
  - small problems: exact planning (DP, VI, PI, etc.)
  - large problems: simulation-based planning
    (aka approximate DP, neurodynamic programming, … etc.)

- model not available → ‘reinforcement learning’
  - model-based RL: learns a model
  - model-free RL: does not learn a model
    - value-based: directly learn value function
    - policy search: directly learn policy

RL Nomenclature

Terminology in RL sometimes confusing…

- model available → ‘planning’
  - small problems: exact planning (DP, VI, PI, etc.)
  - large problems: simulation-based planning
    (aka approximate DP, neurodynamic pr

Common confusion #1: mixing up these

► Of course: MBRL typically **uses** planning

- model not available → ‘reinforcement learning’
  - model-based RL: learns a model
  - model-free RL: does not learn a model
    - value-based: directly learn value function
    - policy search: directly learn policy

So why care about model-based RL?

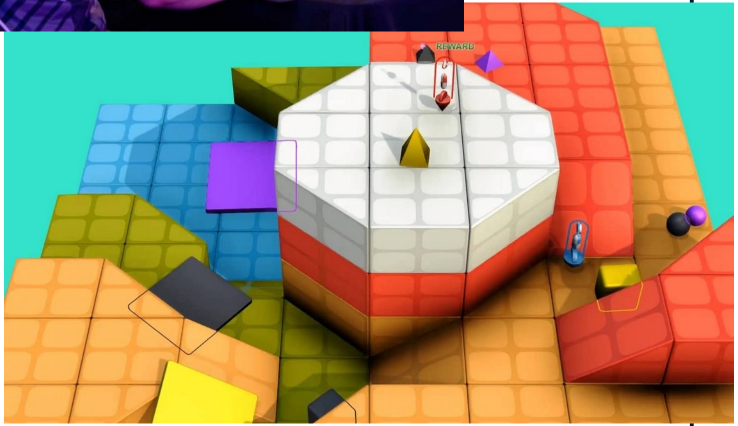
(can’t we just apply model-free RL methods directly on the world?)

Successes: Deep RL

- Atari

- Dota 2

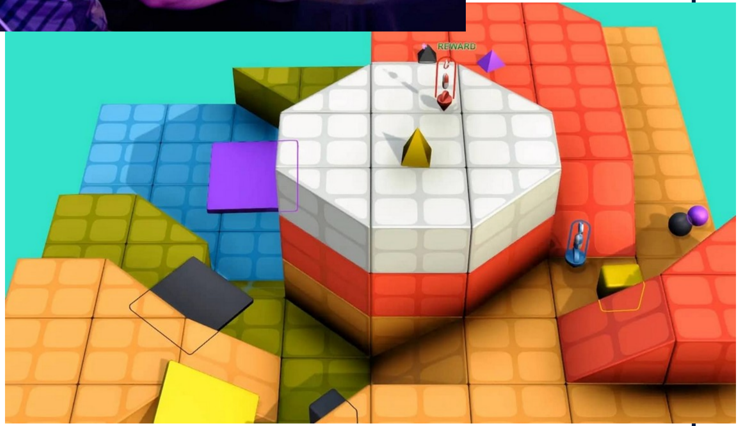
- XLand



** training used 'frame skipping' so 200M frames from environment needed
*** also 'frameskipping' so almost x4 frames from environment

Deep RL methods are data hungry





- Atari: DQN was using
  50 million frames** per game
  (38 days of play by a human)

- Dota 2 ***:
  1–3M steps per batch
  estimated **9.7 trillion steps**

- XLand
  - `fine tuning' --- **100M steps**
  - training of last (5th) generation
    > **100 billion steps**

** training used 'frame skipping' so 200M frames from environment needed
*** also 'frameskipping' so almost x4 frames from environment

Deep RL methods are data hungry

- Atari: DQN was using **50 million frames**** per game (38 days of play by a human)
- 1-7 days on 1 GPU

- Dota 2 ***: 1–3M steps per batch estimated **9.7 trillion steps**
- 10 months 80k—173k CPUs: 7.5 steps/s 1000s of GPUs

- XLand
  - `fine tuning' --- **100M steps**
  - training of last (5th) generation > **100 billion steps**
- 8 TPUv3s
  - 30mins
  - 23 days

** training used 'frame skipping' so 200M frames from environment needed
*** also 'frameskipping' so almost x4 frames from environment

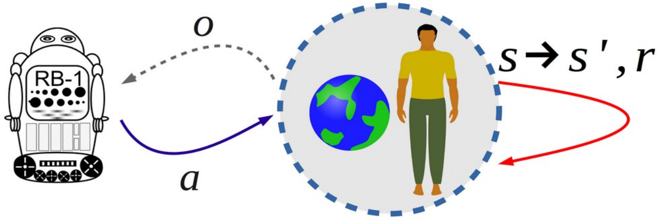
Hmmm... :(
want to learn about the *real* world...!

- ...it will not give us so much samples...

Picture by Ahmed Rabea

What if we want to learn to adapt to humans?

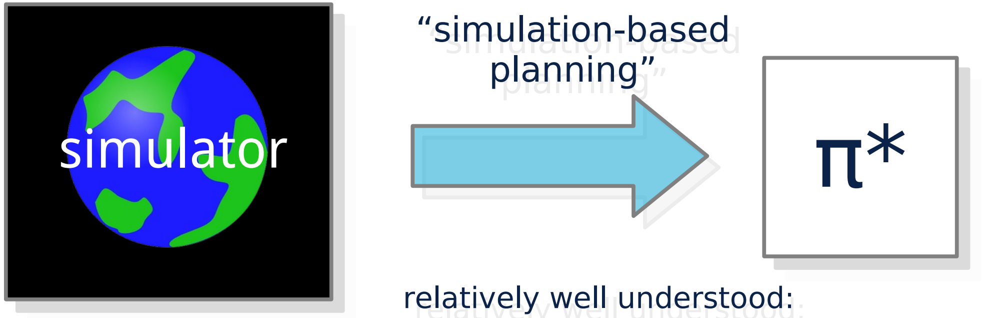
- ▶ they will not give us billions of attempts...



MBRL is potentially promising to overcome this problem

Simulators are great!

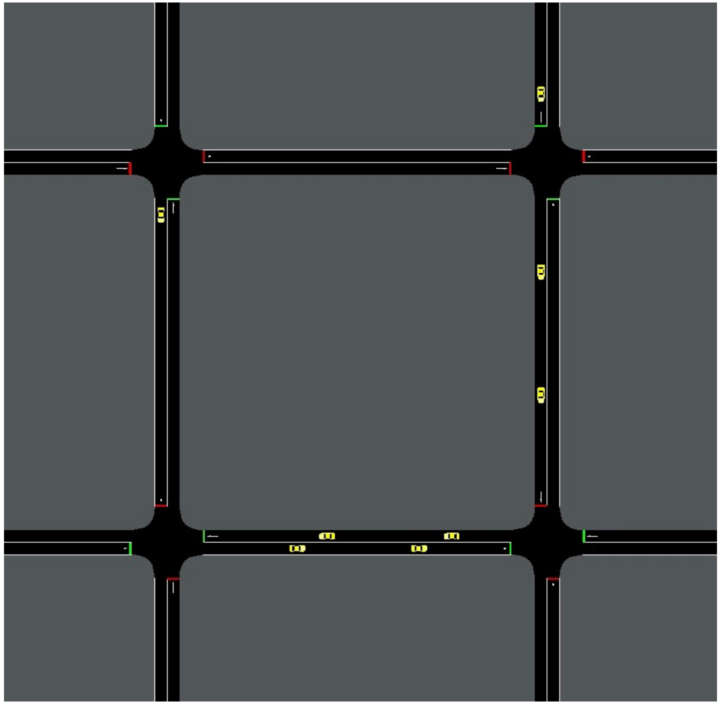
Can do **simulation-based planning**!



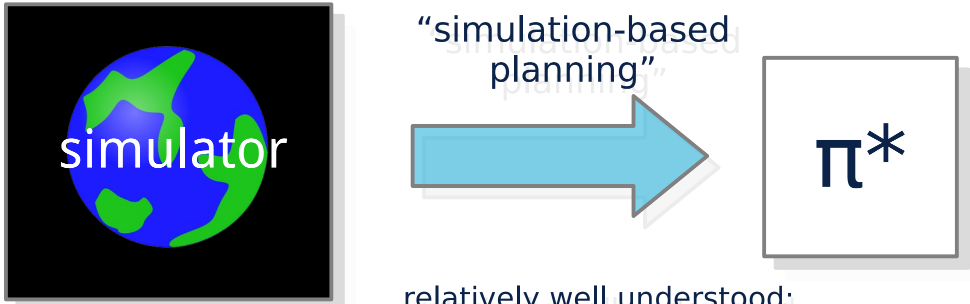
relatively well understood:
- online planning
- model-predictive control
- deep RL

Simulators are great!

Can do **simulation-based planning**!



abstraction can scale deep RL
to 100 intersections



relatively well understood:

- online planning
- model-predictive control
- deep RL

Suau, et al. Distributed Influence-Augmented Local Simulators for Parallel MARL in Large Networked Systems. NeurIPS 2022.

Simulators are great!

Transfer to the real world…

sim2real gap…

but if these are similar enough, we can expect π* to do well in the real world

"simulation-based planning"

π*

relatively well understood:
- online planning
- model-predictive control
- deep RL

Simulators are great! If you have them…

Otherwise learn them: **Model-based RL**

what techniques?

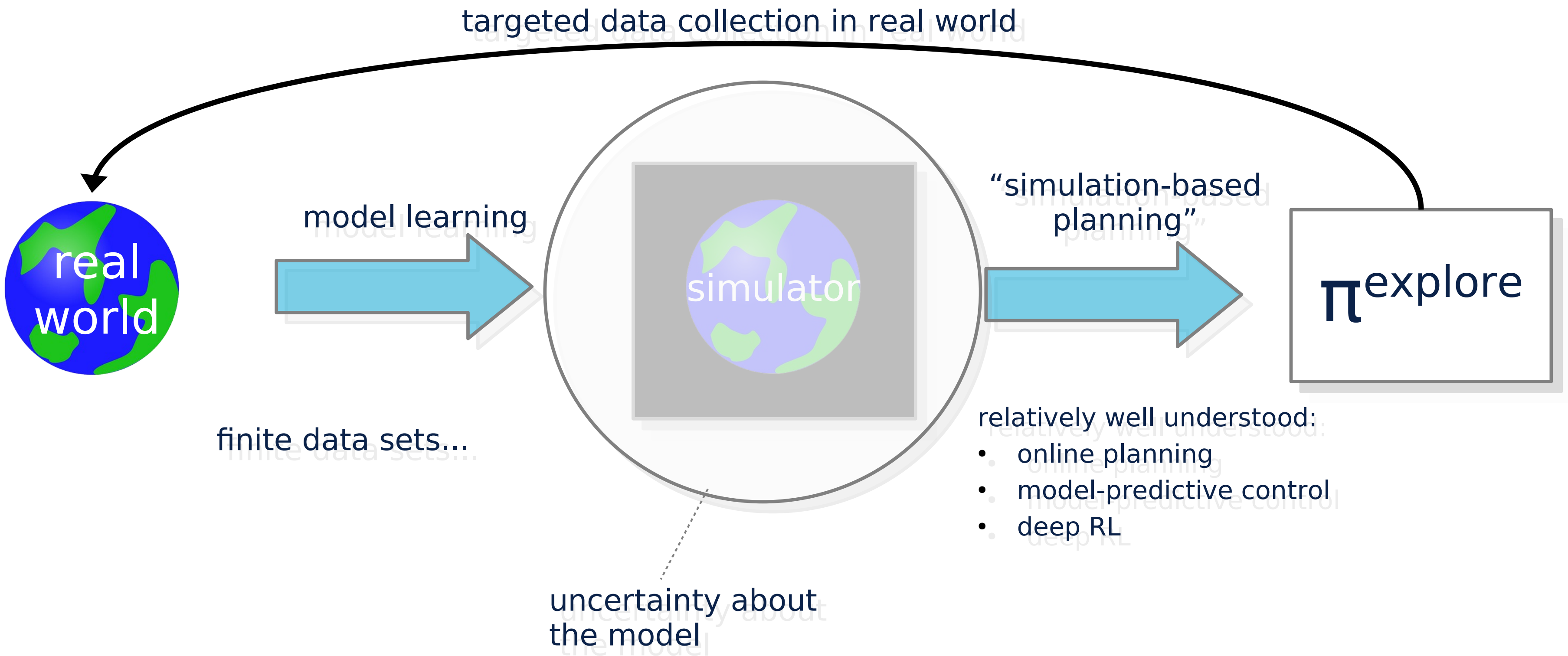
simulator needs to predict outcome of **all actions** well! (“causal model”)

relatively well understood:

- online planning
- model-predictive control
- deep RL

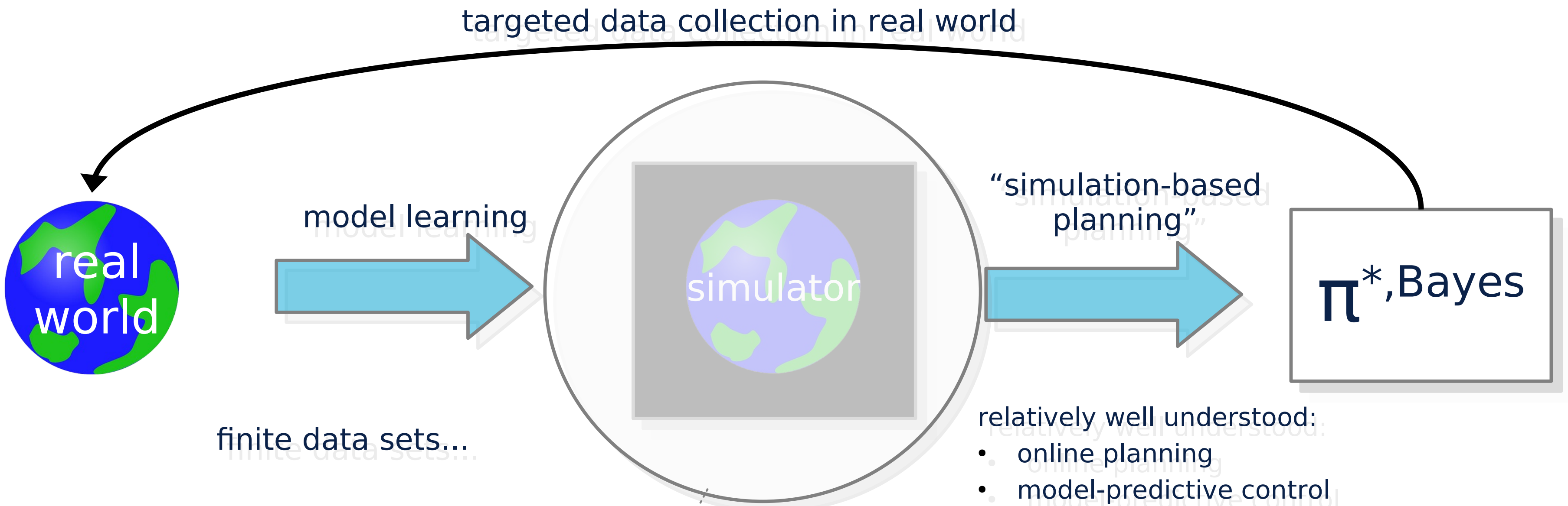
Bayesian RL: Active learning / active perception

In ideal case we close the loop: targeted data collection!



Bayesian RL: Active learning / active perception

In ideal case we close the loop: targeted data collection!



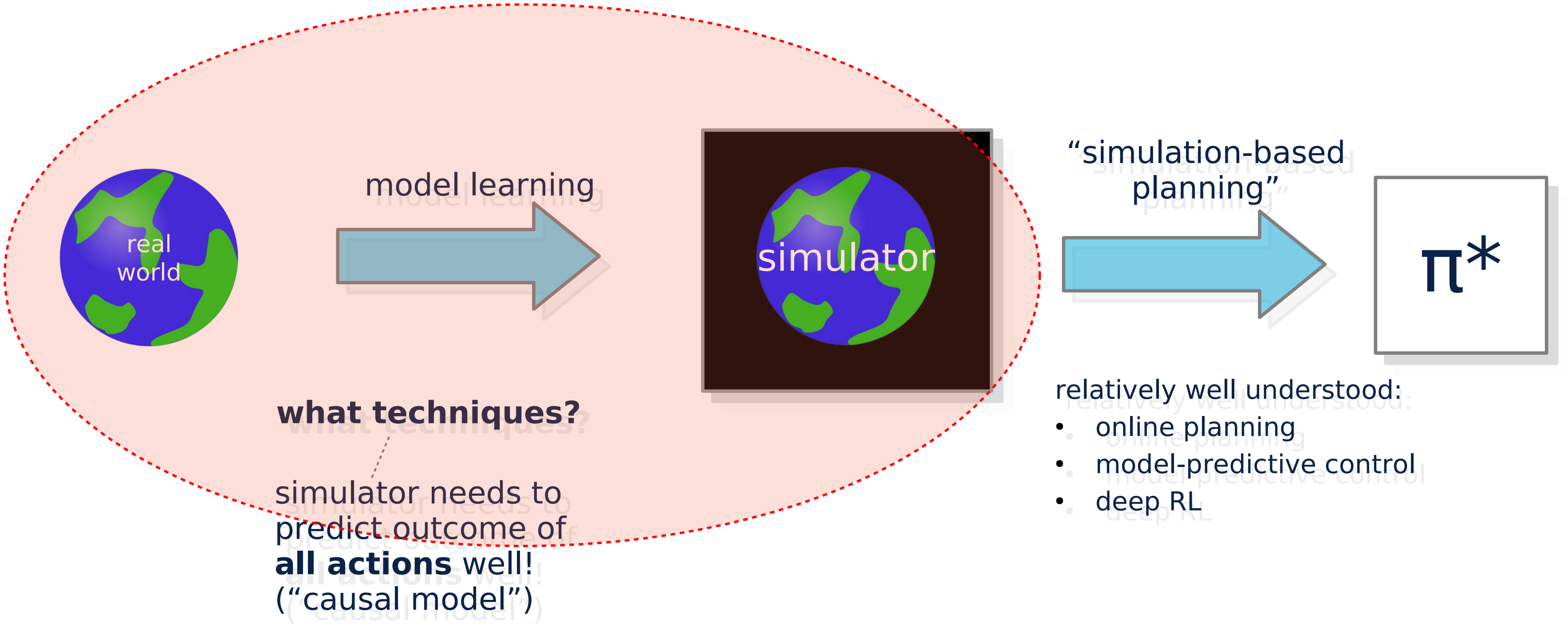
relatively well understood:

- online planning
- model-predictive control
- deep RL

If we have priors
→ true Bayesian RL:
optimally (!) trade off exploration/exploitation

One step back

Let’s zoom in on the model learning:



One step back

Let's zoom in on the model learning:

what techniques?

simulator needs to predict outcome of **all actions** well! ("causal model")

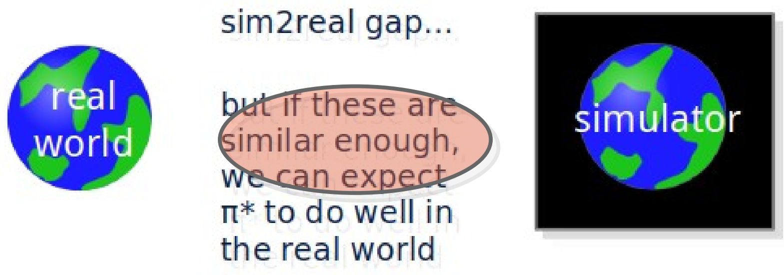
additionally:
this needs to work with abstraction!

relatively well understood:
- online planning
- model-predictive control
- deep RL

Does MBRL work?

Remember: Models *are* abstractions!

- Models **are an abstraction of reality**
- Rare to encounter a true MDP…
- ‘tabular’ MBRL uses human-defined state spaces
  → typically abstractions
- ‘deep’ MBRL learns its own state representation
  → they are abstractions
- **→ So we would hope that things like MBRL also work on abstractions, right? ←**



MBRL + abstractions…?

- Turns out that it is not that simple…!
- E.g., consider R-max
  - it's theory is based on being in an MDP!
    (critically depends on Markov property)

Theorem 2 *Let M be an SG with N states and k actions. Let $0 < \delta < 1$, and $\epsilon > 0$ be constants. Denote the policies for M whose ϵ-return mixing time is T by $\Pi_M(\epsilon, T)$, and denote the optimal expected return achievable by such policies by $Opt(\Pi_M(\epsilon, T))$. Then, with probability of no less than $1 - \delta$ the R-MAX algorithm will attain an expected return of $Opt_M(\Pi(\epsilon, T)) - 2\epsilon$ within a number of steps polynomial in $N, k, T, \frac{1}{\epsilon}$, and $\frac{1}{\delta}$.*

and clearly… in deep MBRL this can also give issues…

R-Max for MDPs:

After (s,a,r',s'):

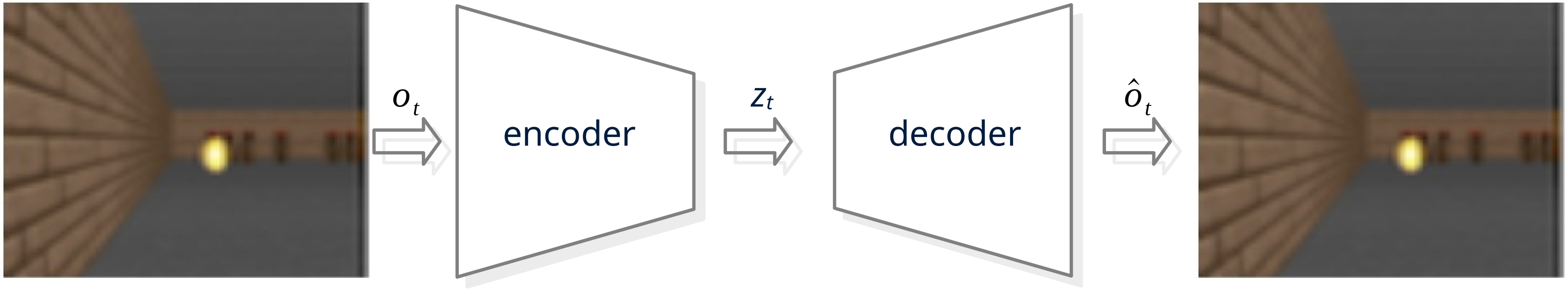
- Store reward r'
- Store transition:
  N[s',s,a] += 1, N[s,a] += 1
- if N[s,a] == m:
  - R(s,a) := mean (Rset(s,a))
  - P(s'|s,a) := N[s',s,a] / N[s,a]
- Plan next step with updated model

Deep MBRL

(and an introduction to some types of abstraction)

Learning models via reconstruction

- E.g., "World models" [Ha&Schmidhuber'18 NeurIPS]
- Main idea: reconstruction to learn useful features



- Then learn a model $P(z_{t+1} \mid z_t, a_t)$, $R(z_t, a_t)$
  - details: RNNs, etc.

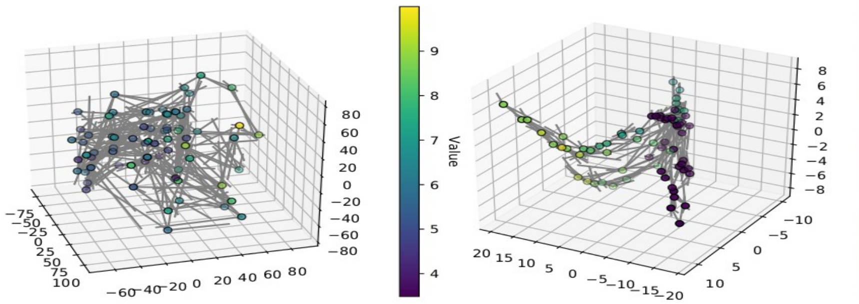
Results

- Can work on complex image-based domains

Source: https://worldmodels.github.io/

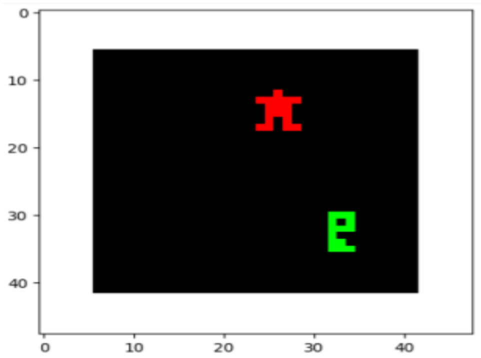
Resulting Latent Space…

- How do these learned state spaces look like?



(a) WM-AE Baseline (b) LD-AE Baseline

Abstract MDP. Nodes: abstract states, edges: abstract transitions, color: predicted value.



not clear if these are the best abstractions…

→ what are good abstractions?

[van der Pol, Kipf, Oliehoek & Welling, AAMAS, 2020]

“Model irrelevance abstractions”
MDP bisimulations & homomorphisms

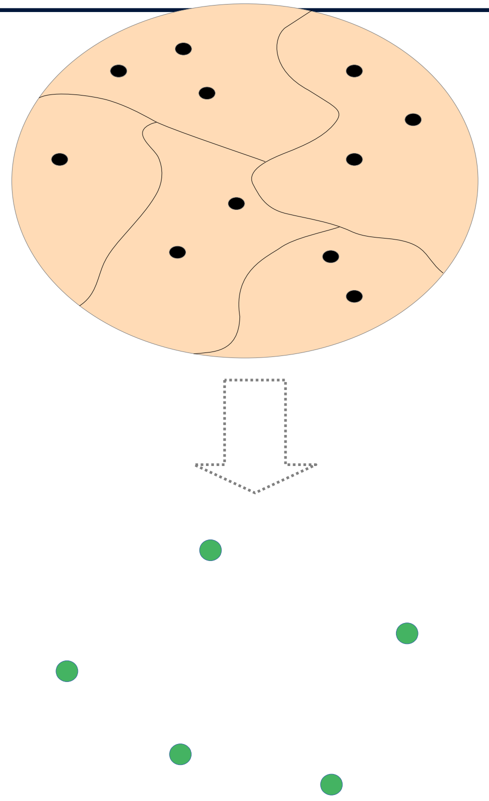
Abstract MDPs

- Abstractions partition the state space

- **Given an MDP** and **some φ**....
  ....can create an **abstract MDP:**

- Weighting function $\omega_{\varphi}(s)$
  - specifies the assumed state probabilities
  - for each abstract state φ

- Transitions:
  $T(\varphi' | \varphi, a) = \Sigma_{s' \in \varphi'} \Sigma_{s \in \varphi} T(s' | s, a)\, \omega_{\varphi}(s)$
- Rewards:
  $R(\varphi, a) = \Sigma_{s \in \varphi} R(s, a)\, \omega_{\varphi}(s)$



Exact MDP bisimulations

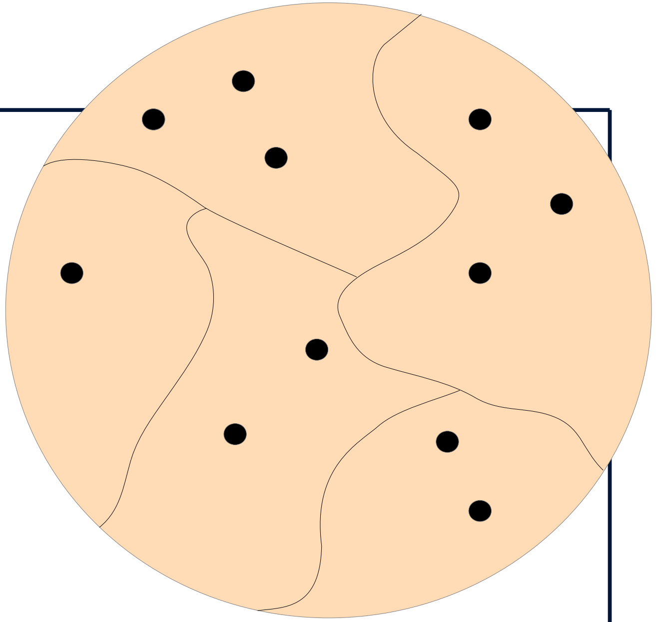
- An abstraction $\varphi(s)$ is a **stochastic bisimulation** [Givan et al. 2003] if
  - whenever s_1, s_2 in same an abstract state φ …

    …they have same rewards

    $R(s_1,a) = R(s_2,a) = R(\varphi,a)$

    ….they have same abstract transitions

    $P(\varphi' \mid s_1, a) = P(\varphi' \mid s_2, a) = P(\varphi' \mid \varphi, a)$



$P(\varphi' \mid s_1, a) =$
$\Sigma_{s' \in \varphi'} \; P(\varphi' \mid s_1, a)$

- Also "model irrelevance abstraction"
  - implies equal Q-values ("ISA Q^{Π} abstraction")
  - i.e. **no value loss**

Approximate: ε-model similarity

- Whenever s_1, s_2 in same an abstract state φ, then, for all a,φ':
  - $|R(s_1,a) - R(s_2,a)| < \varepsilon_R$
  - $|P(\varphi' \mid s_1, a) - P(\varphi' \mid s_2, a)| = |\Sigma_{s' \in \varphi'} P(s' \mid s_1, a) - P(s' \mid s_1, a)| < \varepsilon_T$
  - → approx. same probability of next clusters φ'

- Value-loss when **planning** [Starre'23, and others before]:

$$V^*(s) - V^{\bar{\pi}^*}(s) \leq \begin{cases} \dfrac{2\varepsilon_R}{(1-\gamma)} + \dfrac{2\gamma\varepsilon_T\,|\bar{S}|\max|R|}{(1-\gamma)^2} & \gamma\text{-discounted infinite horizon} \\ h\,\varepsilon_R + (h+1)h\,\varepsilon_T\,|\bar{S}|\max|R| & h \text{ finite horizon} \end{cases}$$

  - $|\bar{S}|$ is the number of abstract states

Starre et al. 2023 TMLR "An Analysis of Abstracted Model-Based Reinforcement Learning"

Approximate: ε-model similarity

- Whenever s_1, s_2 in same an abstract state φ, then, for all a,φ':
  - $|R(s_1,a) - R(s_2,a)| < \varepsilon_R$
  - $|P(\varphi' \mid s_1, a) - P(\varphi' \mid s_2, a)| = |\Sigma_{s' \in \varphi'} P(s' \mid s_1, a) - P(s' \mid s_1, a)| < \varepsilon_T$
  - → approx. same probability of next clusters φ'

- Value-loss when **planning** [Starre'23, and others before]:

$$V^*(s) - V^{\bar{\pi}^*}(s) \leq$$

- $|\bar{S}|$ is the num

Starre et al. 2023 TMLR "An A
Model-Based Reinforcement L

Motivates "Bisimulation metrics" [Ferns, Panangaden & Precup 2004, UAI]

► Turn this into a sort of metric to be used for optimization

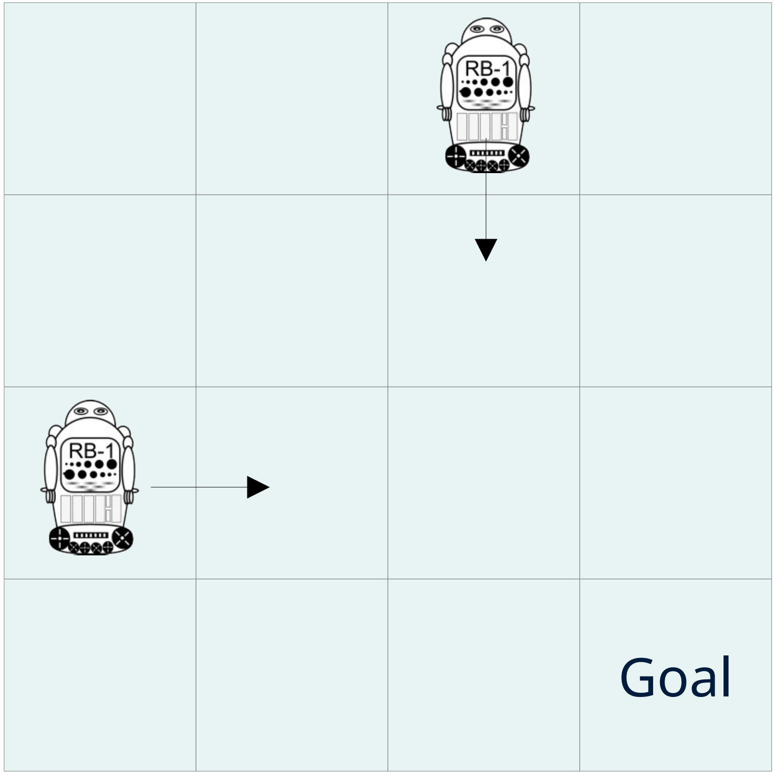
► General form

$$d(s,s') = max_a(c_R|r_s^a - r_{s'}^a| + c_T d_P(T_s^a, T_{s'}^a))$$

MDP homomorphisms

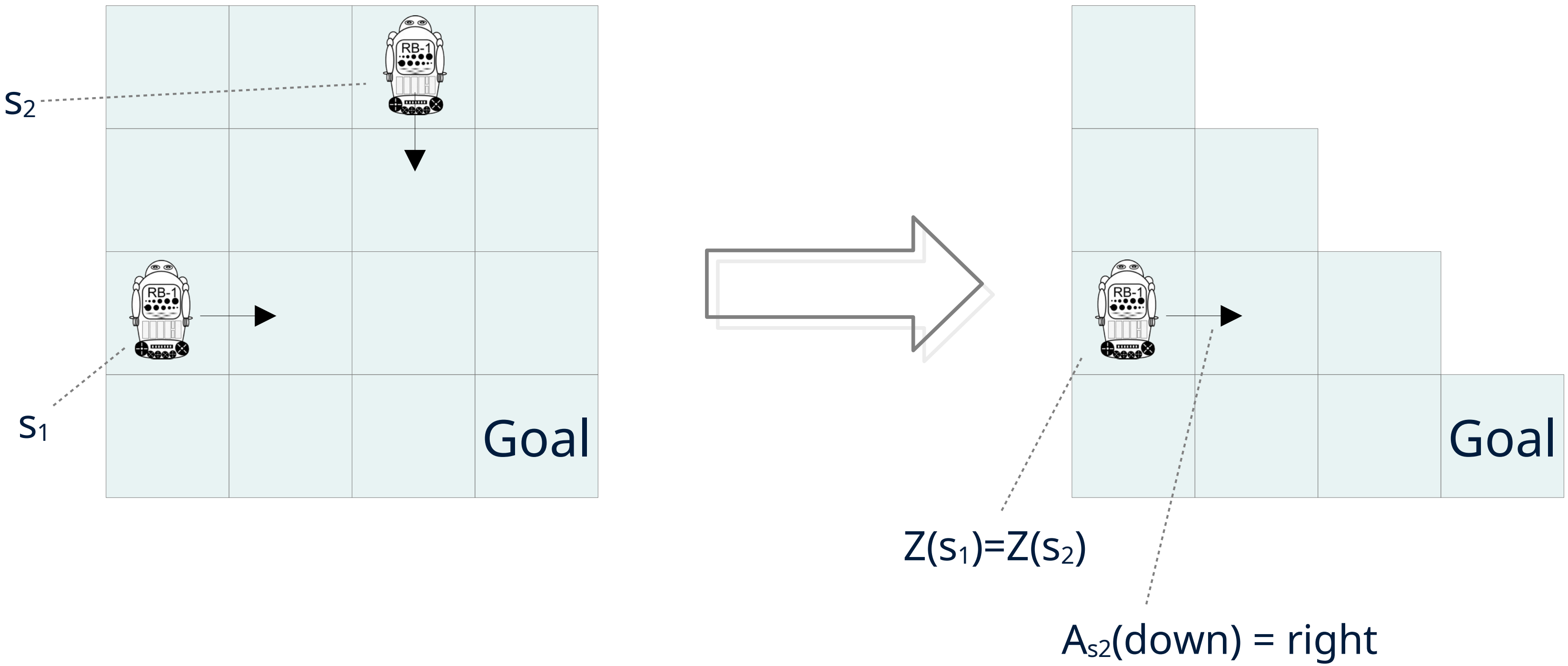
- MDP bisimulation:
  criteria need to hold for all actions

- But sometimes different
  actions have similar effects…



MDP homomorphisms

- Transform states $Z(s)$ and actions $A_s(a)$



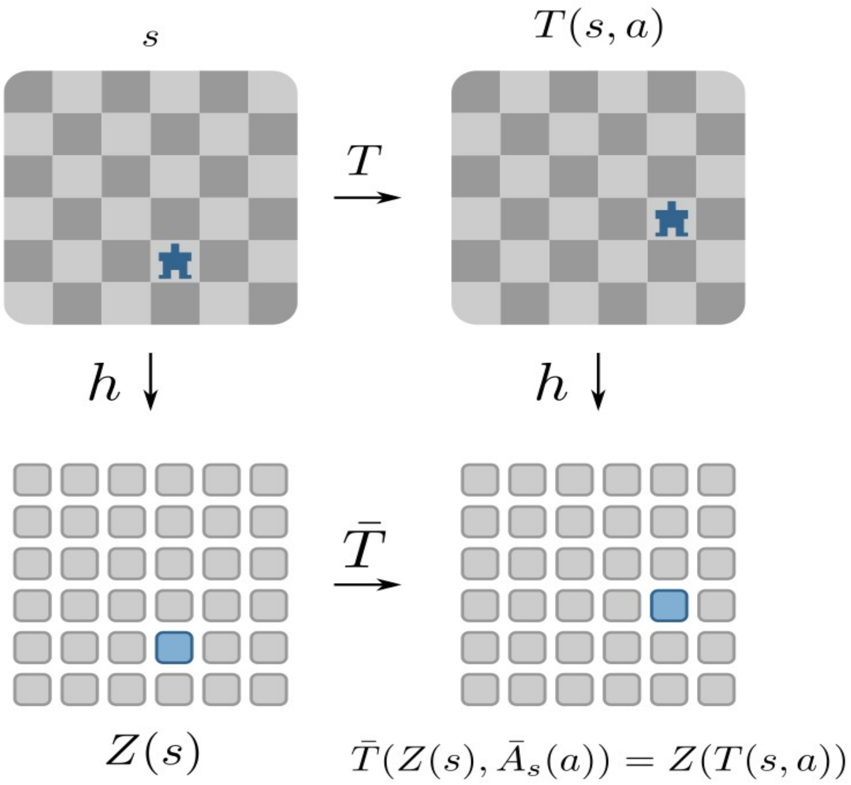
Put constraints on the Latent Space

- Enforce consistency
- Deterministic transitions
- "MDP homomorphism loss"

$$L(\theta,\phi,\zeta)=\frac{1}{N}\sum_{n=1}^{N} d\left(Z_\theta(s_n{}'),\bar{T}_\phi\left(Z_\theta(s_n),\bar{A}_\phi(z,a)\right)\right)$$
$$+d\left(R(s_n),\bar{R}_\zeta\left(Z_\theta(s_n)\right)\right)$$

- Contrastive loss:

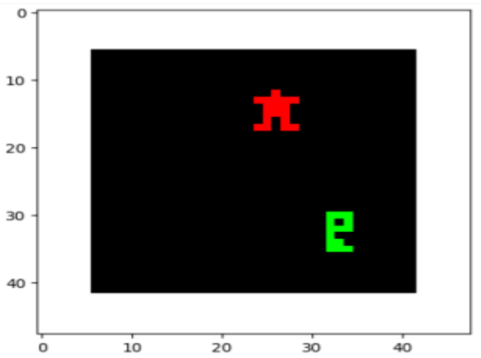
$$\frac{1}{N}\sum_{n=1}^{N}\sum_{s_\neg\in S_\neg} max\left(0,\epsilon-d\left(Z_\theta(s_\neg),\bar{T}_\phi\left(Z_\theta(s_n),\bar{A}_\phi(z,a)\right)\right)\right)$$

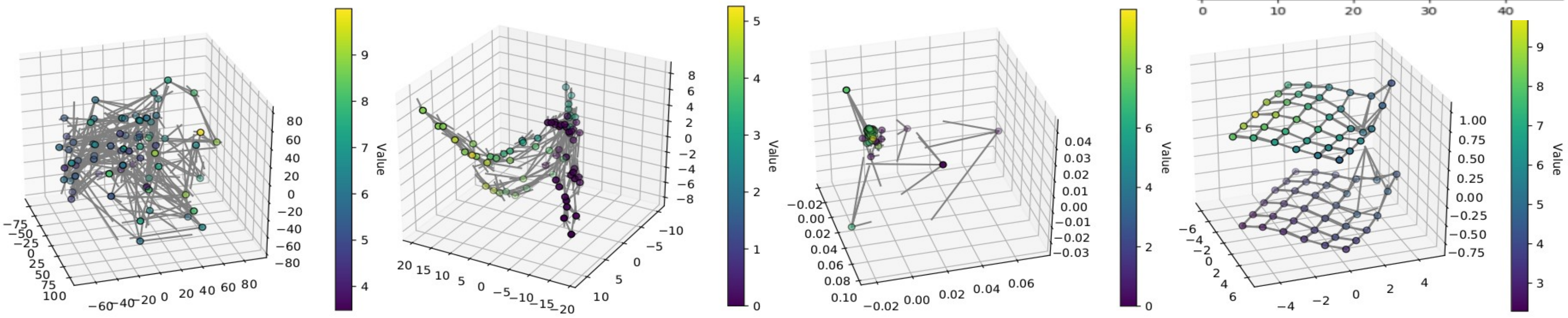


[van der Pol, Kipf, Oliehoek & Welling, AAMAS, 2020]
Also "consistency loss" [Ye et al. 2021 NeurIPS].
Closely related: Wasserstein believer [Avalos et. al 2024 ICLR]

Put constraints on the Latent Space

- Much recent work: learn latent representation
  → But need appropriate constraints!





(a) WM-AE Baseline (b) LD-AE Baseline (c) DMDP-HOM Baseline (d) This paper

Abstract MDP. Nodes: abstract states, edges: abstract transitions, color: predicted value.

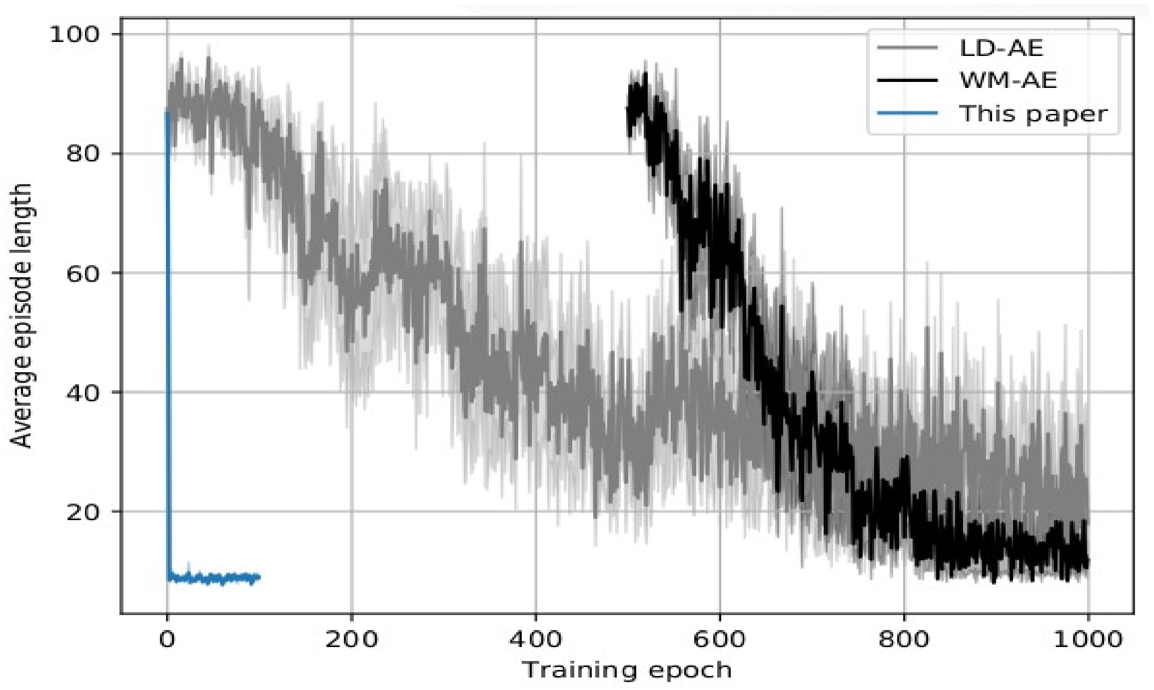
[van der Pol, Kipf, Oliehoek & Welling, AAMAS, 2020]

Leads to “Plannable models”

The models support planning

- ►Discretize
- ►Estimate state transitions
- ►Apply standard planning (**plain value iteration**)

- ►much better sample complexity



Abstract MDP. Nodes: abstract states, edges: abstract transitions, color: predicted value.

[van der Pol, Kipf, Oliehoek & Welling, AAMAS, 2020]

So a success story…?

- Yes… empirically, but let's reflect what we did… we:

- …collected data
- …used it to learn a state representation (with "MDP homomorphism loss")
- …estimated a model on top of these abstract states
- …and hoped for the best

Would we not like any theory that would say:
"assuming your abstraction is quite good, your value loss will be limited"
?

Subtleties in model learning for (actual) tabular MDPs

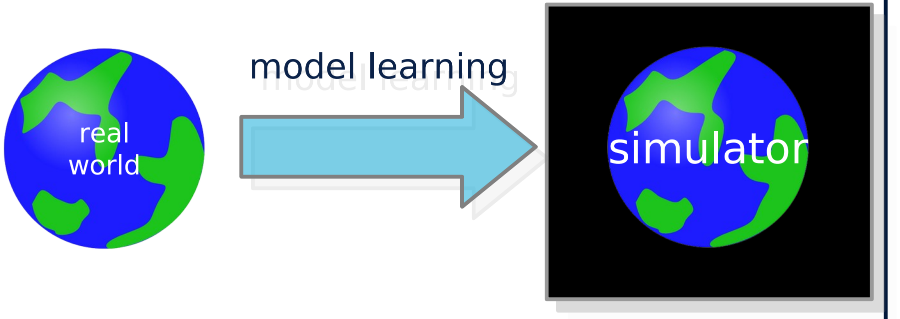
MBRL: estimating conditionals P(s'|s,a)

Tabular model-based RL requires: ***

1) estimate P(s'|s,a), R(s,a)

2) have some confidence on accuracy

   – guarantees
   – exploration



- *** we really need P(s'|do(s,a)), but in MDPs we do not need to worry: P(s'|do(s,a))= P(s'|s,a)

E.g.: Weissman et al. (2003)

- L1 error of estimated transitions T_Y w.r.t. the true T:

$$||T_Y(\cdot|s,a) - T(\cdot|s,a)||_1 \triangleq \sum_{s' \in S} |T_Y(s'|s,a) - T(s'|s,a)|.$$

- Can be bounded:

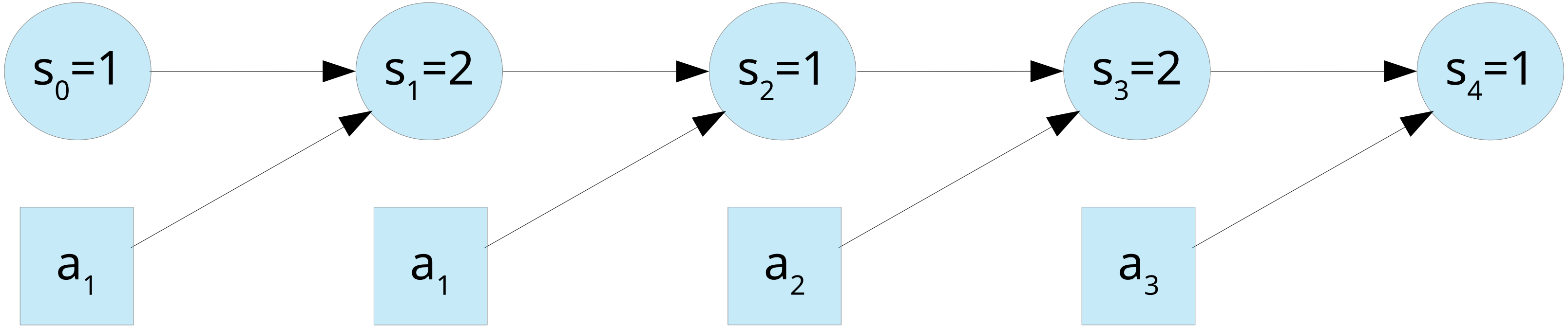
Lemma 1 (L_1 inequality (Weissman et al., 2003)). *Let* $Y_{s,a} = Y_{s,a}^{(1)}, Y_{s,a}^{(2)}, \cdots, Y_{s,a}^{(N(s,a))}$ *be i.i.d. random variables distributed according to* $T(\cdot|s,a)$*. Then, for all* $\epsilon > 0$,

$$\Pr(||T_Y(\cdot|s,a) - T(\cdot|s,a)||_1 \geq \epsilon) \leq (2^{|S|} - 2)e^{-\frac{1}{2}N(s,a)\epsilon^2}. \quad (3)$$

Tsachy Weissman, Erik Ordentlich, Gadiel Seroussi, Sergio Verdu, and Marcelo J Weinberger. Inequalities for the l1 deviation of the empirical distribution. Hewlett-Packard Labs, Tech. Rep, 2003.

But Markov != independent variables

- It **does not make different visits to a state independent**



- $P(s_2=1 | s_1=2,a_1) \neq P(s_2=1 | s_1=2,a_1,s_4=1)$
  - Since reaching s_4=42 in 2 time steps gives information about what s_2 was!
- So *estimating the conditionals* is possibly problematic...!

E.g, when conditioning on number of visits...

- To estimate this accuracy, we typically use large deviation bounds (e.g. Hoeffding bound).
  - Roughly "**Given m samples** of s' for a particular (s,a) with prob. 1-δ the estimated transition probability P(.|s,a) is ε-accurate"

E.g. R-Max

After (s,a,r',s'):

- Store reward r'
- Store transition:
  N[s',s,a] += 1, N[s,a] += 1
- **if N[s,a] == m:**
  - R(s,a) := mean (Rset(s,a))
  - P(s'|s,a) := N[s',s,a] / N[s,a]
- Plan next step with updated model

Example [Strehl & Littman 2008]

- What probability will I estimate for P(. | s=2) given that I require m=5 samples?

- The fact that I revisit state 2 fully determines the outcome of the previous visit!

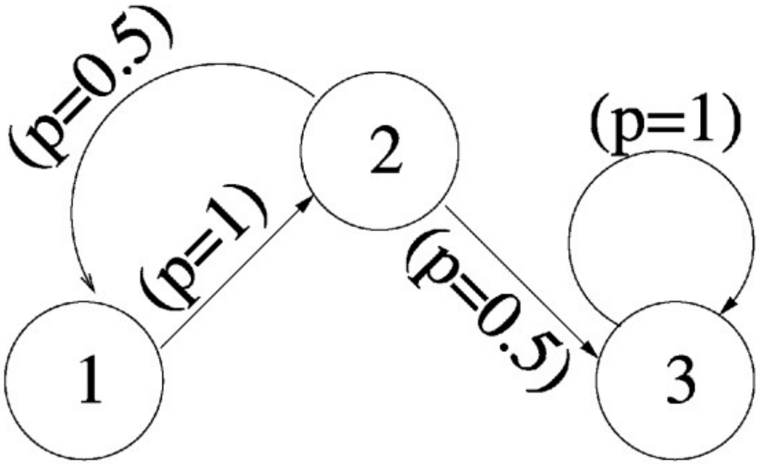


image reproduced from Strehl & Littman (2008)

Strehl, Alexander L., and Michael L. Littman. "An analysis of model-based interval estimation for Markov decision processes." Journal of Computer and System Sciences 74.8 (2008): 1309-1331.

OK, so now what?

- So we cannot use Hoeffding bounds, etc?
- No, fortunately Strehl & Littman (2008) also show that
  - the probability of a sequence of outcomes from a Markov chain
  - is upper bounded by a process that makes independent draws.

- Strehl & Littman (2008):

"we **may assume the samples are independent** if we only use this assumption **when upper bounding the probability of certain sequences of next-states or rewards**. This is valid because, although the samples may not be independent, any upper bound that holds for independent samples also holds for samples obtained in an online manner by the agent."

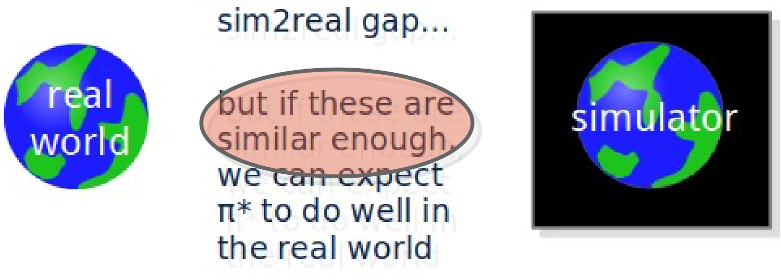
- i.e., can still use Hoeffding bound, etc.
  - **as long as samples from an MDP: they need to be identically distributed**

MBRL with abstraction

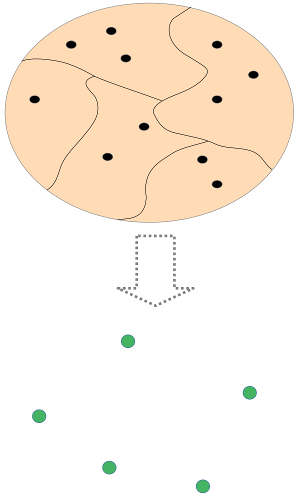
Recap: Models *are* abstractions!

- Models **are an abstraction of reality**
- Rare to encounter a true MDP…

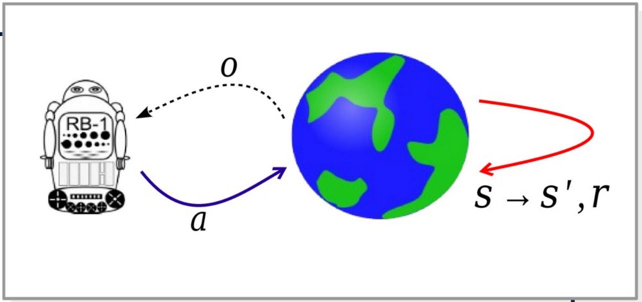
  → So we would hope that things like MBRL also work on abstractions ←



- Abstract MDPs can be **constructed**
  - Given an MDP, φ, and weighting function $\omega_\varphi(s)$
  - $T(\varphi' | \varphi, a) = \Sigma_{s' \in \varphi'} \Sigma_{s \in \varphi} T(s' | s, a)\, \omega_\varphi(s)$
  - $R(\varphi, a) = \Sigma_{s \in \varphi} R(s, a)\, \omega_\varphi(s)$



Abstraction as a POMDP



- Abstraction can be thought of as a POMDP!
  - abstract states are observations: $\varphi \leftrightarrow o$
- When entering φ, there is a distribution over states
  - there is a true belief, that depends on history $h_t = (\varphi_0, a_0, \ldots, a_{t-1}, \varphi_t)$
  - in an Abstract MDP $\omega_\varphi(s)$ approximates that belief
- An Abstract MDP is an MDP → can plan with it → value loss bounded
- An Abstract MDP can be constructed, **it can not be 'experienced' !**

Combining RL and Abstraction

- When the MDP is not known…
  → learn about abstract states directly?

- Setting: **“RL from abstracted observations” (RLAO)**
  - E.g., directly learn T(φ’|φ,a), R(φ,a)
    using model-based RL?

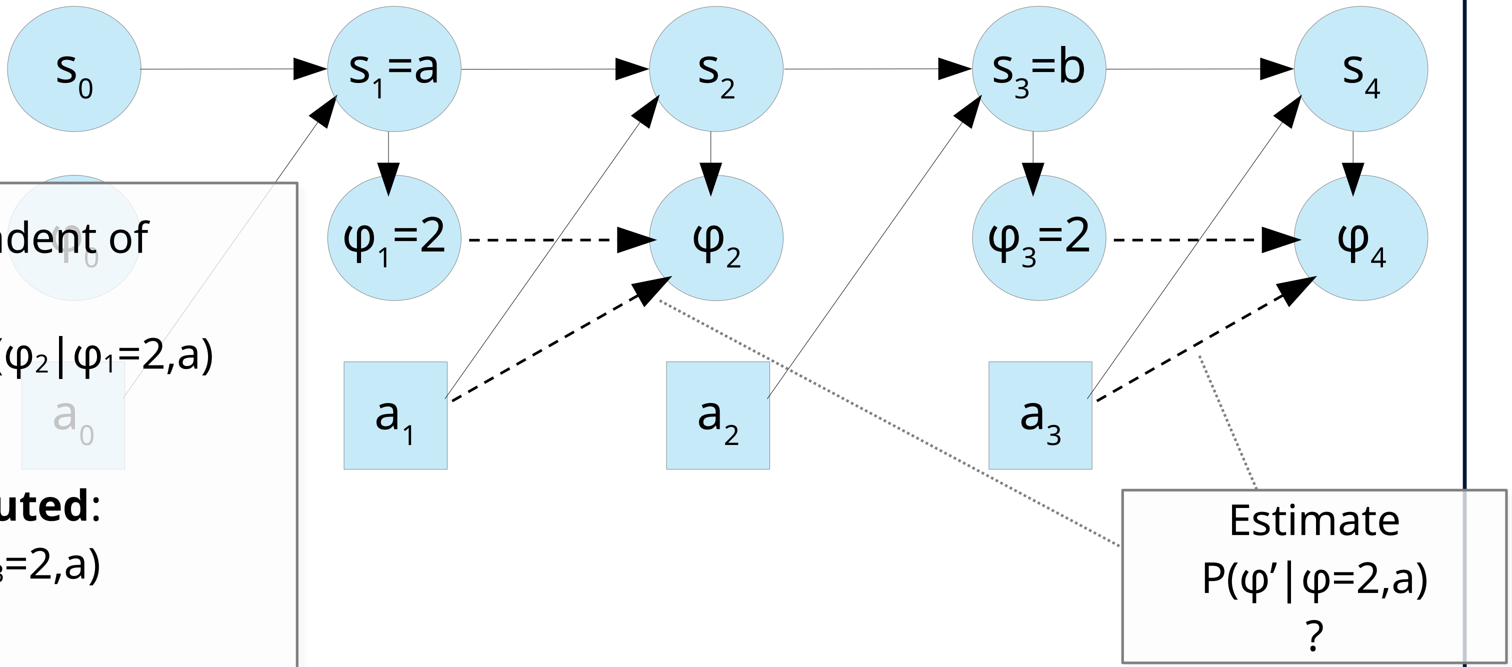
- **Guarantees for MBRL method may not hold!**
  - These proofs are typically based on independence of samples
    (Hoeffding, Weissman)

Wait…! Strehl & Littman (2008) demonstrated that we can still use these results, right?

→ yes, in MDPs, but we are in a POMDP!

Combining RL and Abstraction - details

- Yes… *in an MDP* we can still use our bounds [Strehl & Littman (2008)]
- But the result is specific for MDPs: uses the Markov property!



- (As before) not independent of future (so rest of data):
  $P(\varphi_2|\varphi_1=2,a,\varphi_3,\varphi_4) \neq P(\varphi_2|\varphi_1=2,a)$

But also:

- **not identically distributed**:
  $P(\varphi_2|\varphi_1=2,a) \neq P(\varphi_4|\varphi_3=2,a)$
- **nor Markov**:
  $P(\varphi_4|a_3,\varphi_3) \neq$
  $P(\varphi_4|a_3,\varphi_3,a_2,\varphi_2,a_1,\varphi_1,a_0,\varphi_0)$

Combining RL and Abstraction - results

- **Fix by resorting to Martingale bounds** [Starre et al. 2023]

Theorem 2 (Abstract L1 inequality). *If an agent has access to a state abstraction function ϕ and uses this to collect data for any abstract state-action pair $(\bar{s}, a)$ by acting in an MDP M according to a policy $\bar{\pi}$, we have that the following holds with a probability of at least $1 - \delta$ for a fixed value of $N(\bar{s}, a)$:*

$$||\bar{T}_Y(\cdot|\bar{s}, a) - \bar{T}_{\omega_X}(\cdot|\bar{s}, a)||_1 \leq \epsilon, \tag{37}$$

where $\delta = 2^{|\bar{S}|} e^{-\frac{1}{8} N(\bar{s},a)\epsilon^2}$.

empirical model estimated
from abstract state data
{<φ,a,φ’>}

An **abstract MDP**
that we would form
when we could observe full
state data {<s,a,s’>}

Starre et al. 2023 TMLR “An Analysis of Abstracted Model-Based Reinforcement Learning”

Combining RL and Abstraction – results

- So...
  - ...can bound difference with an abstract MDP...
  - ...that abstract MDP has bounded performance loss for ε-model similarity abstraction...

- → bound on total loss of RLAO with an additional ε term

- In our paper: apply this to R-max
  - Other algorithms left for future work.

assuming your abstraction is quite good, your value loss will be limited!

Starre et al. 2023 TMLR “An Analysis of Abstracted Model-Based Reinforcement Learning”

Conclusions

Conclusions

- RL holds a lot of promise, but… sample efficiency is an issue.
- Possible solution: learning models!

- Fully deep-learning based:
  - reconstruction loss $\rightarrow$ perhaps too complex?
  - MDP homomorphism loss + contrastive loss $\rightarrow$ seems promising

- A reassuring theory:
  we *can* do model-based RL on abstracted data
  - provided the abstraction is good enough

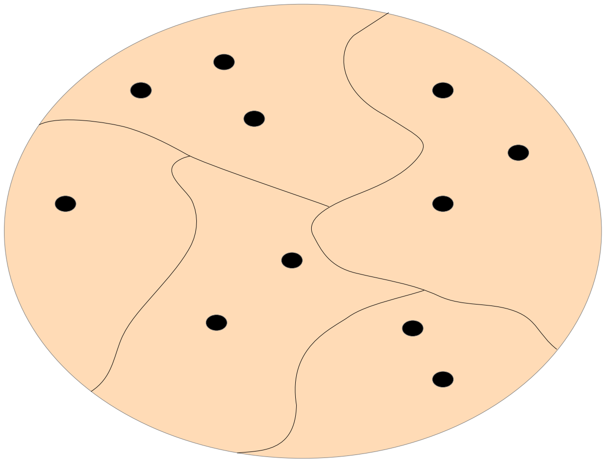
Abstractions partition the state space

- Abstract state φ = cluster of states
  - What are good abstractions?
  - how to cluster…?
- Different types of abstractions:
  - φ_0 — identity (i.e., no abstraction)
  - φ_m — model irrelevance, preserve R,T
  - φ_Π — Q^Π irrelevance (for all $\pi \in \Pi$), preserves Q-values
  - φ_{Q^*} — Q* irrelevance, preserves all optimal Q-values
  - φ_{a^*} — a* irrelevance, preserve Q(., a*)
  - φ_{π^*} — π* irrelevance, preserves optimal action

coarser

- Hierarchy:

φ_0 ISA φ_m ISA φ_Π ISA φ_{Q^*} ISA φ_{Q^*} ISA φ_{a^*} ISA φ_{π^*}



Li, Lihong, Thomas J. Walsh, and Michael L. Littman. "Towards a unified theory of state abstraction for MDPs." AI&M 1.2 (2006): 3.