

From Experience to Reason: Dyna-Style Planning with Linear Function Approximation and Prioritized Sweeping

Rich Sutton

with

Csaba Szepesvari

Alborz Geramifard

Michael Bowling

David Silver

and others at the

University of Alberta

take-home points re Dyna and model-based RL

- an appealing view of planning as day-dreaming
 - more relevant to AI than other fast RL methods
- linear case works almost as well as the tabular
 - great freedom in choosing dream-start states
 - thus great freedom in search control
- in the general case, we can see glimmers of the machinery for representation change

the constructivist view of AI

- AI = making machines that can predict and control their low-level input-output stream of sensation and action
- construct a model of the world; find its key states and state transitions
- continually predict, control, plan
- continually reformulate representations for rapid, adaptive decision-making

Dyna is AI in a nutshell

1. learn a model of the world
2. use the model to find a good policy (planning)
3. find a good policy without making or using a model (classical RL)

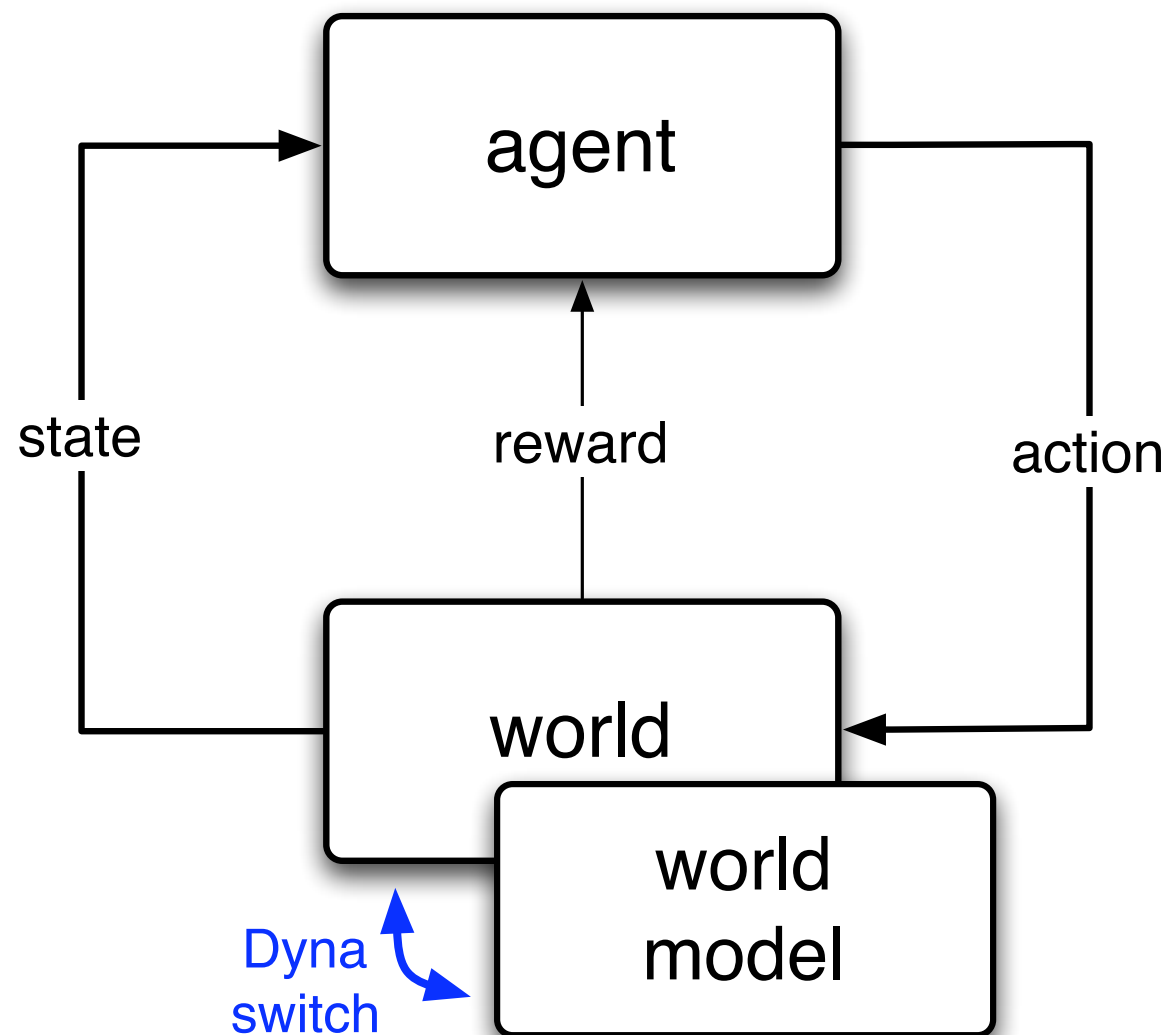
do all of these all the time without stopping

⇒ models and planning are 2/3s of AI

what i care about

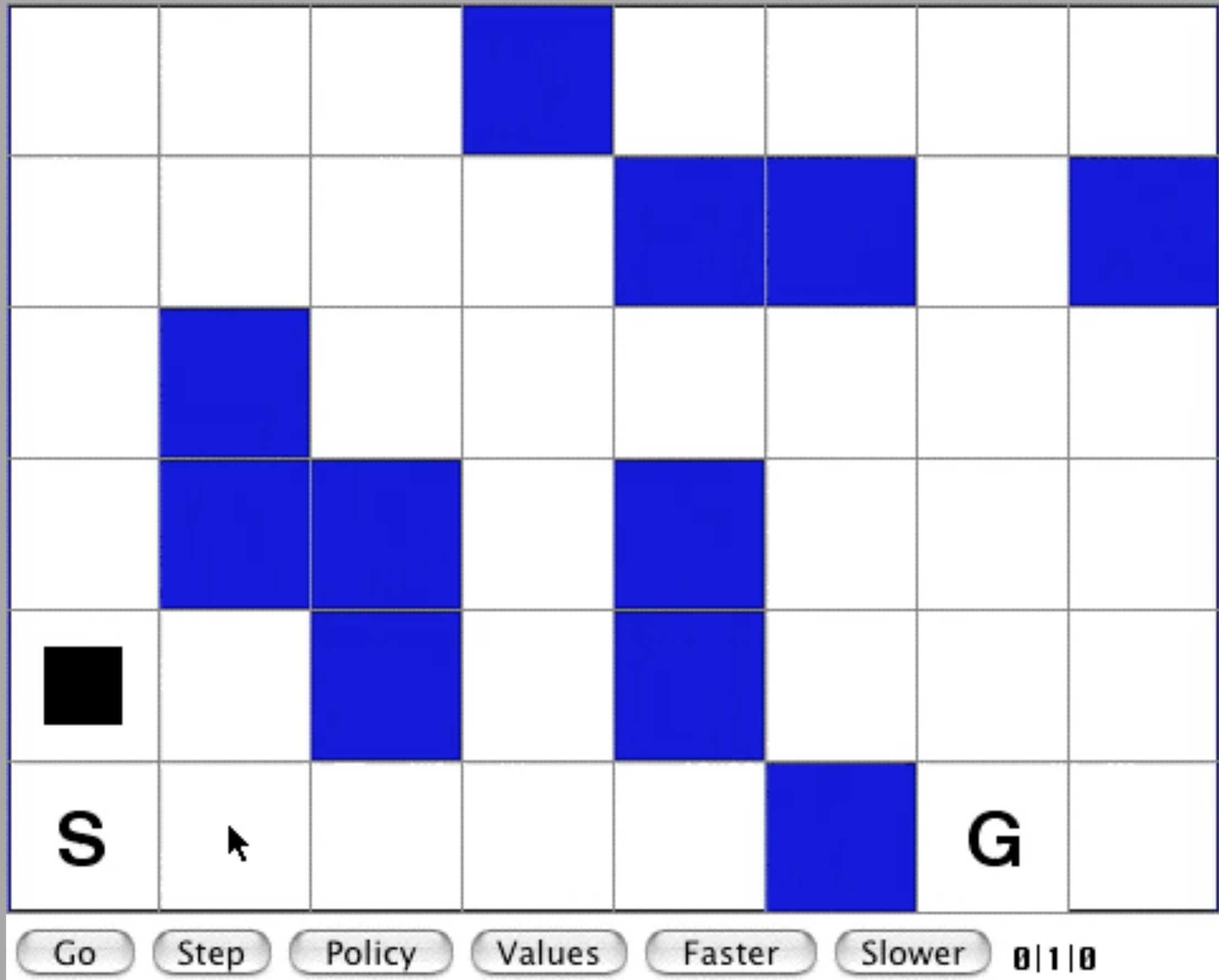
- solving AI
- understanding what *mind* is well enough to make some
- making machines that can predict and control their input-output stream
- maximizing reward is a good formal spec
- learning, rapid adaptation, responsiveness
- planning and foresight, reason
- constructivism. constructing an understanding of the world. abstractions. new state variables. new high-level operators. new features
- the special thing about people is that we are ready to think about the world at the right levels.
representation is all.

Dyna architecture

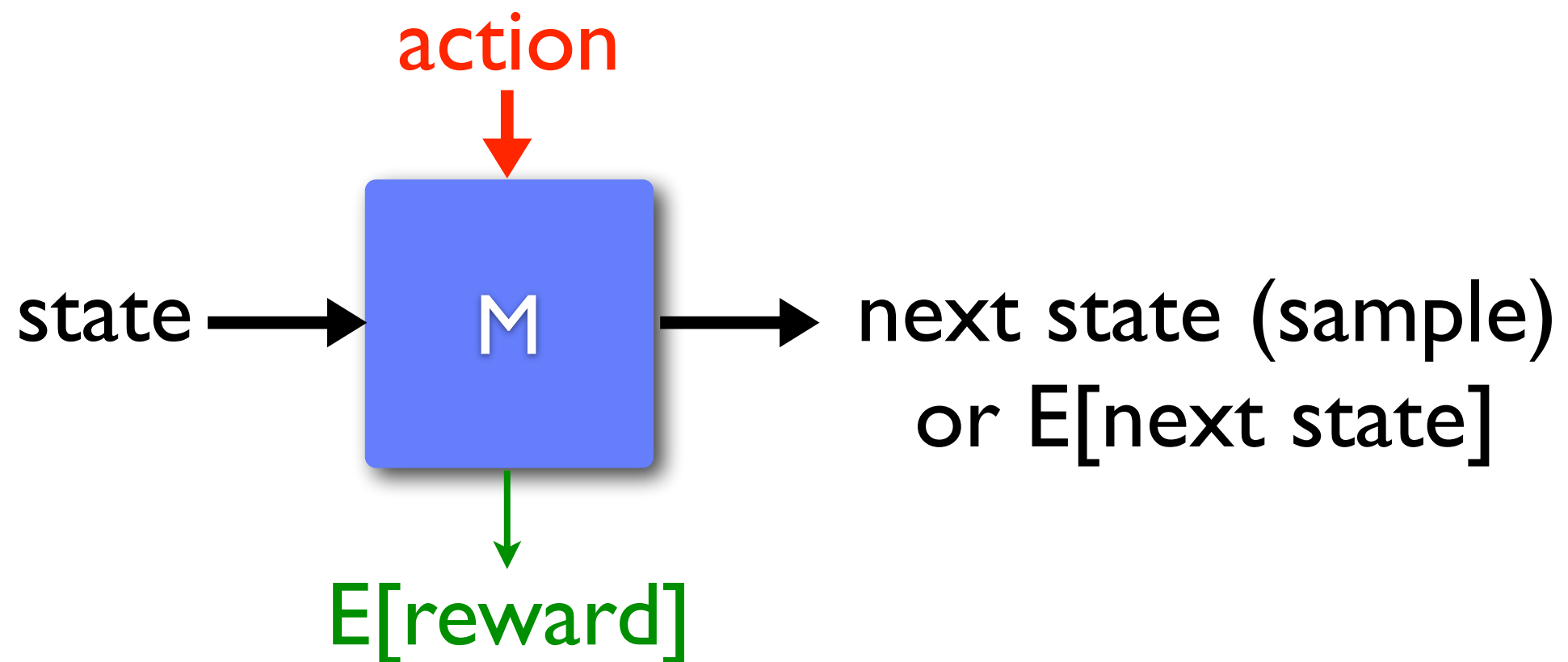


- learning and planning are achieved by the same RL algorithm (e.g., TD(0), Sarsa)
- applied to real or simulated experience
- altering the same policy and/or value function
- learning and planning are incremental, simultaneous, and asynchronous – nothing waits on anything else

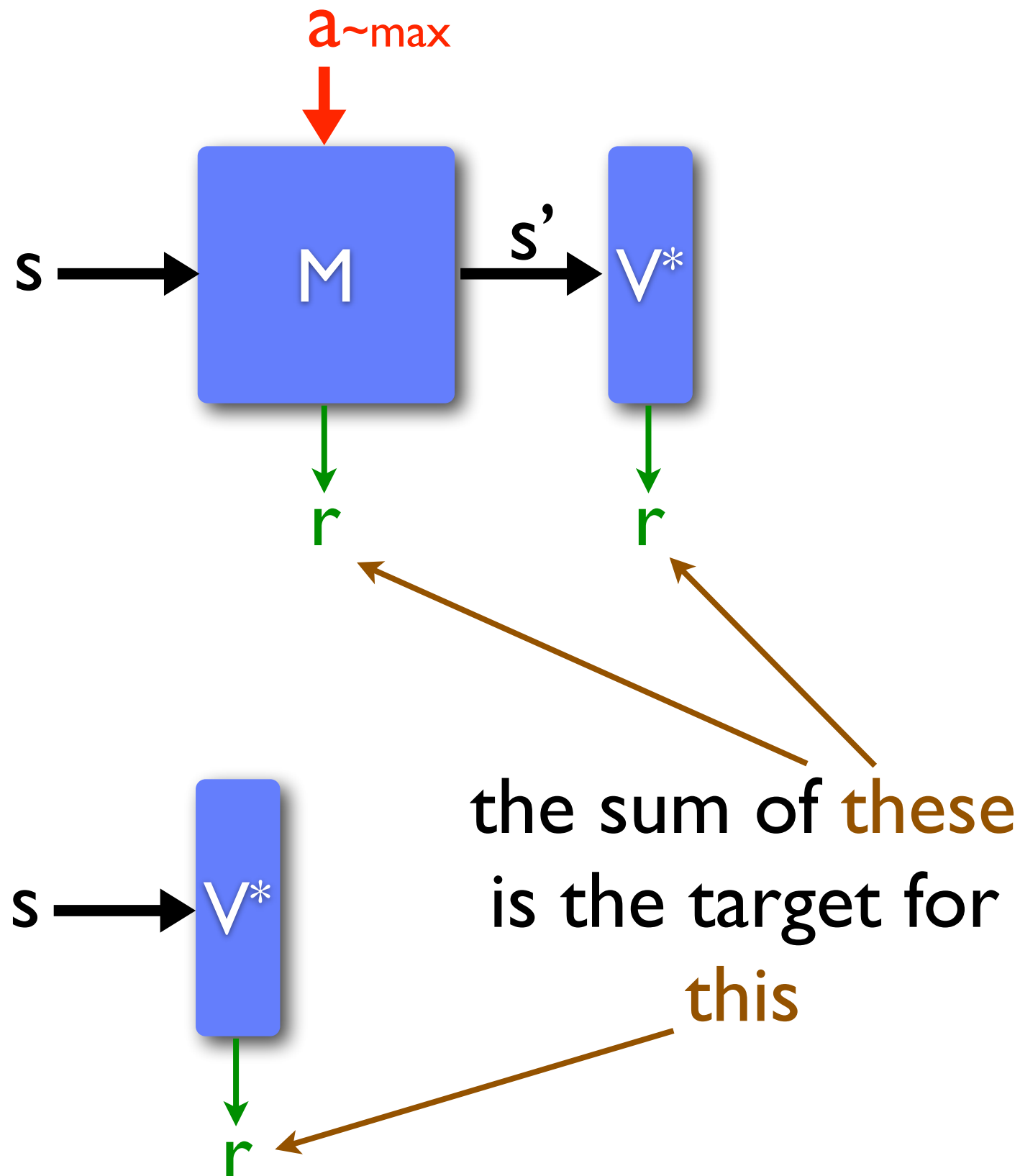
Dyna circa 1990



fundamental operation of a model - *projection*



planning
update,
in pictures



other methods for online learning and planning

- essentially there aren't any
 - ...with any claim to generality
 - ...that use general function approximation
- adaptive control methods limited to LQR systems
- LS methods are poorly suited to online use
- Paduraru's thesis is an exception

where is Dyna now?

- ☒ online anytime reacting/learning/planning
- ☒ data-efficient learning control
- ☒ flexible search control (prioritized sweeping)
- ☒ consistent with flexible, powerful temporal abstraction (options)
- ☐ works with flexible, general function approximation
- ☐ works with constructivism

in a linear model

- states are represented by feature vectors

$$s \longrightarrow \phi_s \qquad s_t \longrightarrow \phi_t \qquad \in \mathbb{R}^n$$

- the model is a set of matrix-vector pairs

$$M = \{F_a, b_a\}_{a \in Actions}$$

expected transition
matrix

$$E\{\phi_{t+1} | \phi_t = \phi, a_t = a\} = F_a \phi$$

$$E\{r_{t+1} | \phi_t = \phi, a_t = a\} = b_a^\top \phi$$

expected reward
vector

Linear model

$$F_a \phi = \phi'$$
$$b_a^\top \phi = r$$

start state vector

ϕ

transition model

F_a

=

ϕ'

predicted
next-state vector

reward model

b_a

r

predicted reward

Linear model

$$F_a \phi = \phi'$$
$$b_a^\top \phi = r$$

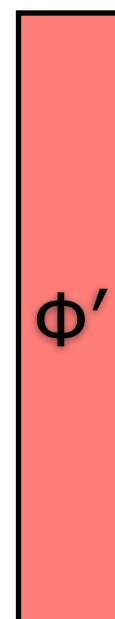
start state vector

transition model

reward model



=



predicted
next-state vector

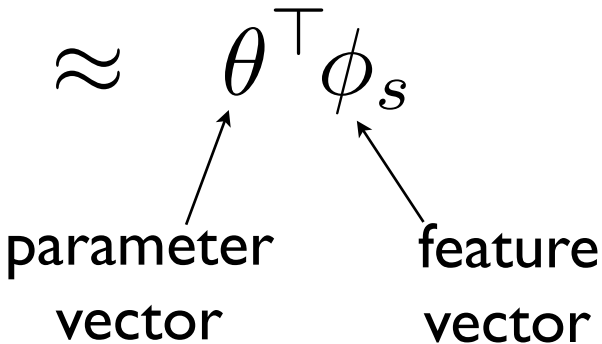


predicted reward

- linear state-value function approximation:

$$V^\pi(s) = E \left[\sum_{k=1}^{\infty} \gamma^{k-1} r_k \mid s_0 = s, \pi \right]$$
$$\approx \theta^\top \phi_s$$

parameter vector feature vector



- action-value approximation:

$$Q^\pi(s, a) = E \left[\sum_{k=1}^{\infty} \gamma^{k-1} r_k \mid s_0 = s, a_0 = a, \pi \right]$$
$$\approx \theta_a^\top \phi_s$$

Dyna-style policy evaluation

$F, b \rightarrow \theta$ via one-step projections and backups

repeat forever:

- generate starting feature vector: $\phi \sim \mu$
- simulate one step forward
 - next feature vector: $\phi' = F\phi$
 - immediate reward: $r = b^\top \phi$
- perform a TD backup:

$$\theta \leftarrow \theta + \alpha(r + \gamma\theta^\top \phi' - \theta^\top \phi)\phi \quad \text{TD}(0)$$

Dyna-style policy evaluation

$F, b \rightarrow \theta$ via one-step projections and backups

repeat forever:

- generate starting feature vector: $\phi \sim \mu$
- simulate one step forward
 - next feature vector: $\phi' = F\phi$
 - immediate reward: $r = b^\top \phi$
- perform a TD backup:

$$\theta \leftarrow \theta + \alpha(r + \gamma\theta^\top \phi' - \theta^\top \phi)(\phi - \phi')$$

residual
gradient
Baird 95

- does this converge?
- what does it converge to?
- does it depend on the starting distribution μ ?
- does it depend on TD(0) vs residual gradient?

Dyna-style policy evaluation

$F, b \rightarrow \theta$ via one-step projections and backups

repeat forever:

- generate starting feature vector: $\phi \sim \mu$
- simulate one step forward
 - next feature vector: $\phi' = F\phi$
 - immediate reward: $r = b^\top \phi$
- perform a TD backup:

$$\theta \leftarrow \theta + \alpha (r + \gamma \theta^\top \phi' - \theta^\top \phi) \phi \quad \text{TD}(0)$$

$= 0$

analysis

at convergence:

$$r + \gamma \theta^\top \phi' - \theta^\top \phi = 0 \quad \forall \phi \sim \mu$$

$$b^\top \phi + \gamma \theta^\top F \phi - \theta^\top \phi = 0 \quad \underline{\forall \phi \sim \mu}$$

if μ spans the full vector space, then this implies

$$b^\top + \gamma \theta^\top F - \theta^\top = 0$$

$$\Rightarrow \theta = (I - \gamma F^\top)^{-1} b$$

which is independent of μ

and which is the classical TD/LSTD solution

convergence thms

- for the TD(0) iteration, F must have spectral radius less than or equal to one $\max_{||x||=1} x^\top F x \leq 1$
- for the residual gradient iteration, convergence is assured whenever the fixed-point exists, i.e., when $I - \gamma F$ is invertible
- both conditions pertain to the stability of the model
- decreasing step sizes (probably not needed)
- extends to control

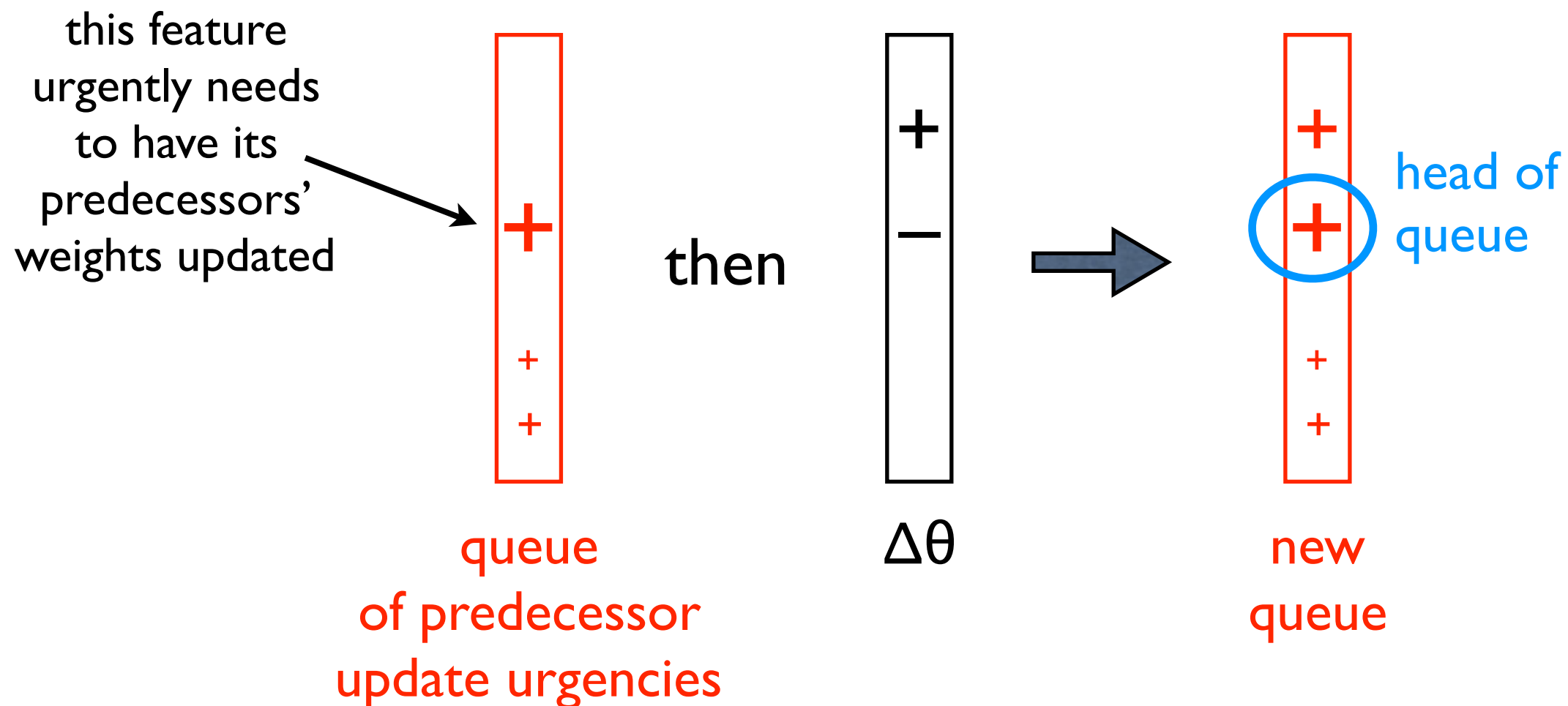
conclusions (I) re linear Dyna

- planning diverges only if the model is unstable (which would mean the planning problem itself was ill-posed)
- Dyna-style planning seems to work out well with linear FA
- we can pick starting state vectors however we want, opening the door to prioritized sweeping

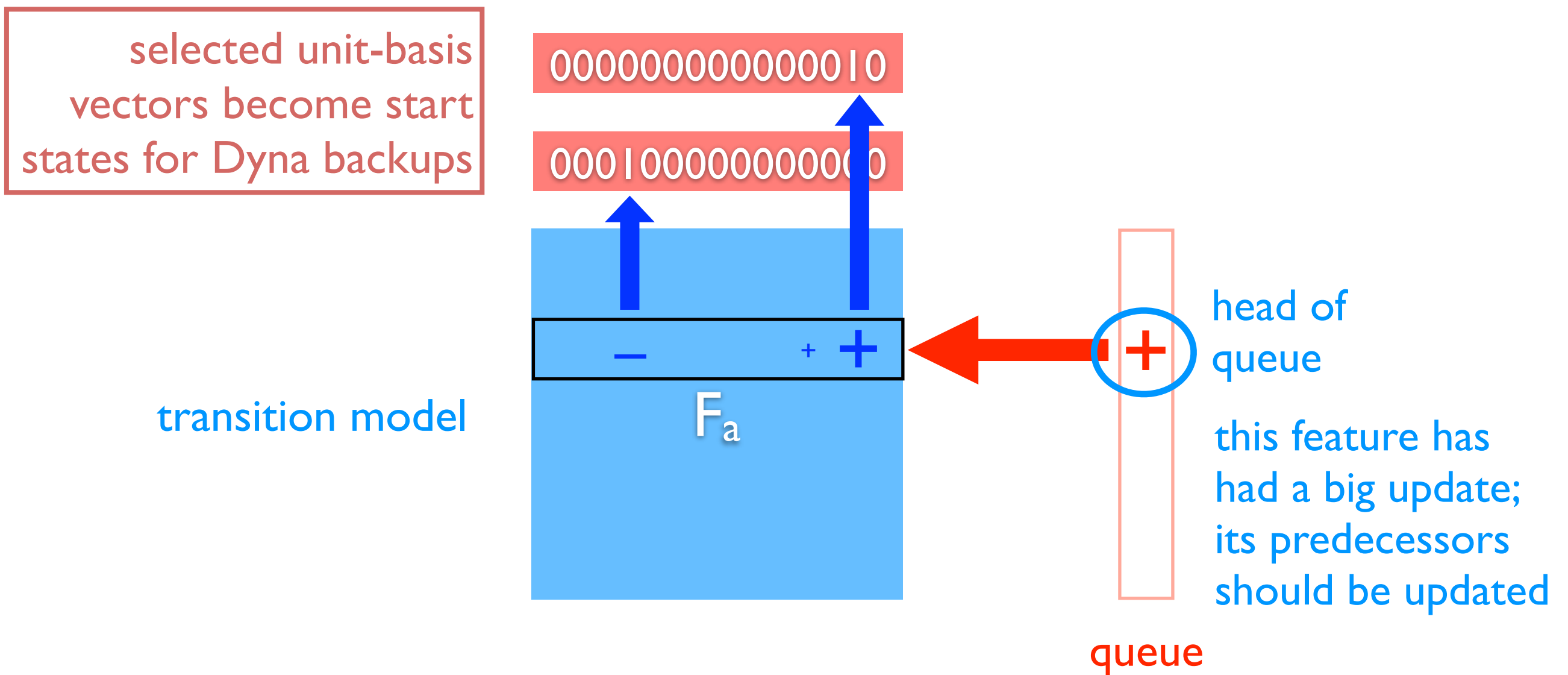
prioritized sweeping

- a clever way of selecting start-state vectors
 - to speed planning (a form of search control)
 - by working backwards from goal states (and states of recently changed value)
- previously it has worked state-by-state
(Moore & Atkeson 1993; Peng & Williams 1993; McMahan & Gordon 2005)
- now must be feature-by-feature

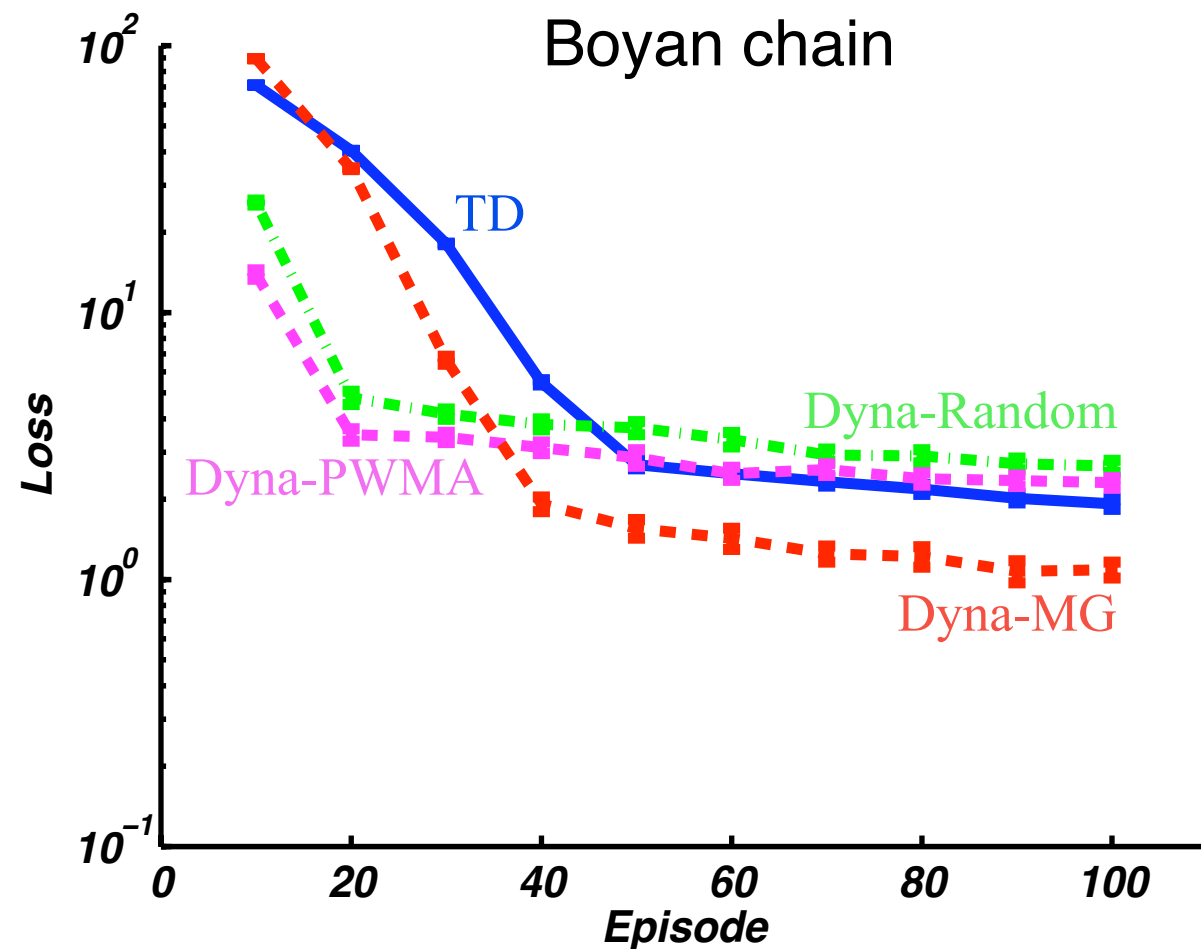
feature-wise priority queue



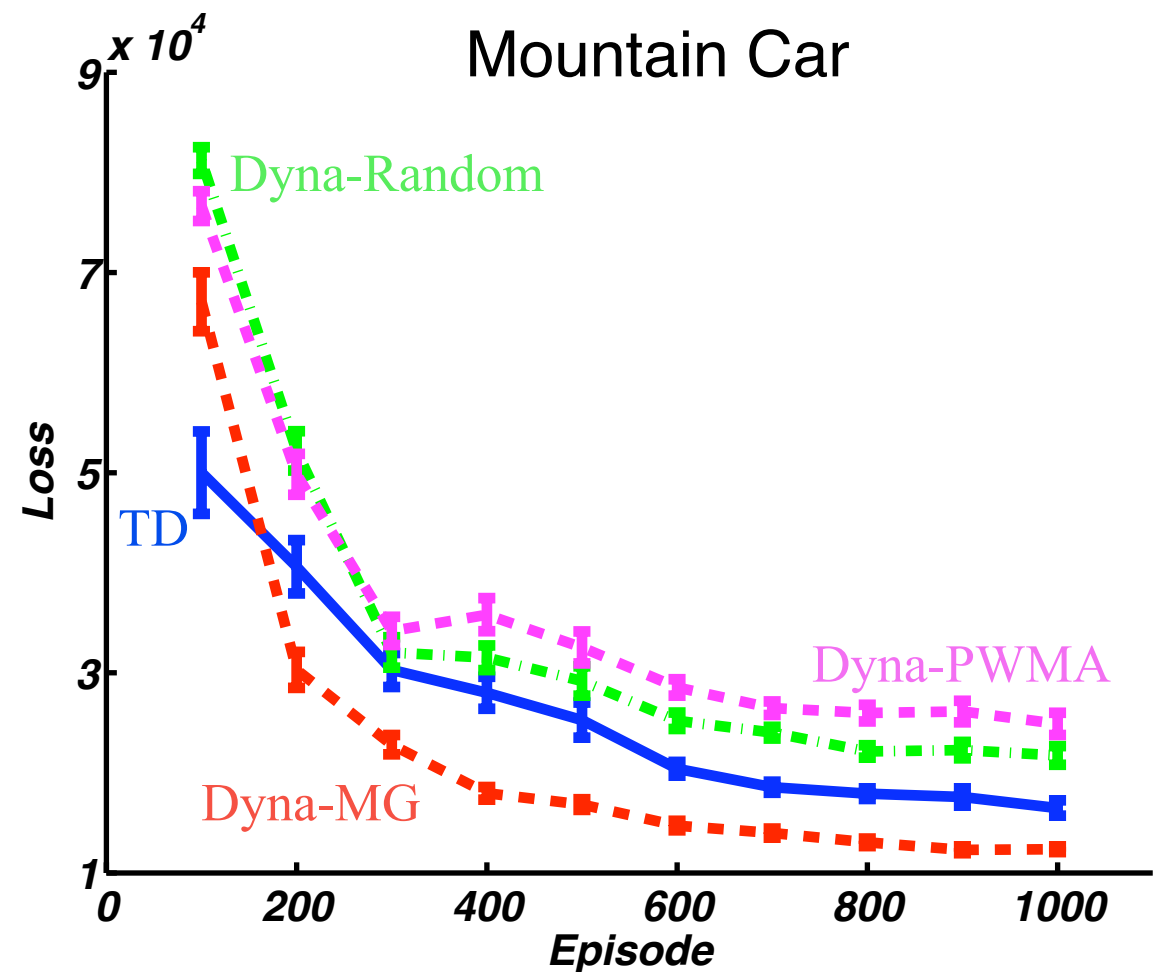
sweeping back to predecessor features



policy evaluation results

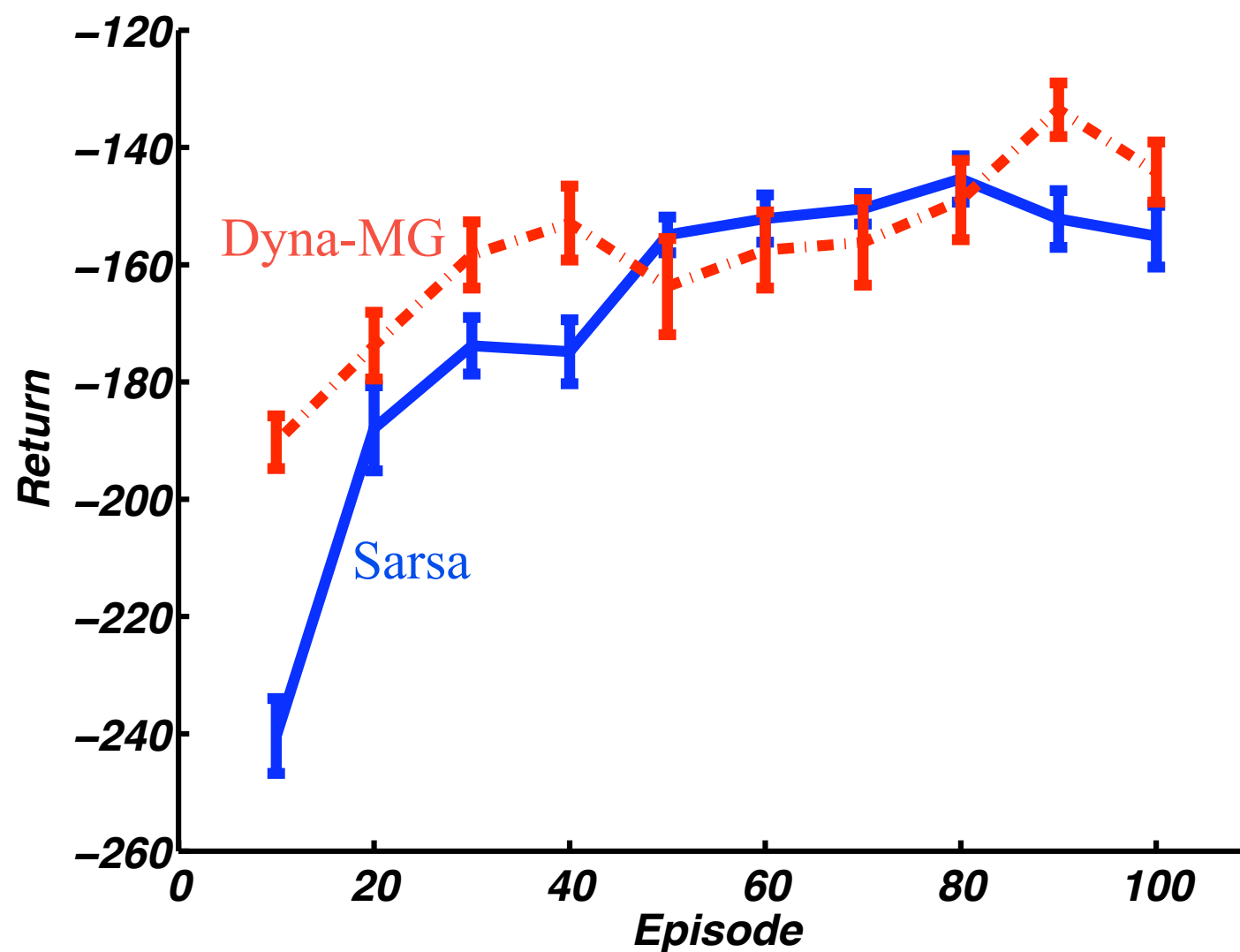


98 states
25 features



2D state space
10 tilings as in Sutton, 1996
hashed to 10,000 features

control results (mountain car)



remarks

- it is our convergence result that makes feature-wise prioritized sweeping possible
- our system (linear Dyna with PS) is probably the best online RL controller
 - better than LSPI, LSS, SPPI, DBNs...
 - the advantage of incremental computation is immense in online control

more remarks

- actual experience is important
 - planning only systems performed poorly
 - we see this also in experience replay

linearity

- is it really a good idea?
 - divergence
 - explosive multi-step projection
- should we add a sigmoid?
 - would this radically change the theory?

future work

- more experimental work
 - demonstrate advantage over LS methods
- semi-linear models
- sampling models?
- forward focusing, multi-step projection
- combining with 2nd-order methods
 - de-correlating the feature vectors

(in conclusion)

take-home points re Dyna and model-based RL

- an appealing view of planning as day-dreaming
 - more relevant to AI than other fast RL methods
- linear case works almost as well as the tabular
 - great freedom in choosing dream-start states
 - thus great freedom in search control
- in the general case, we can see glimmers of the machinery for representation change

thank you for your attention