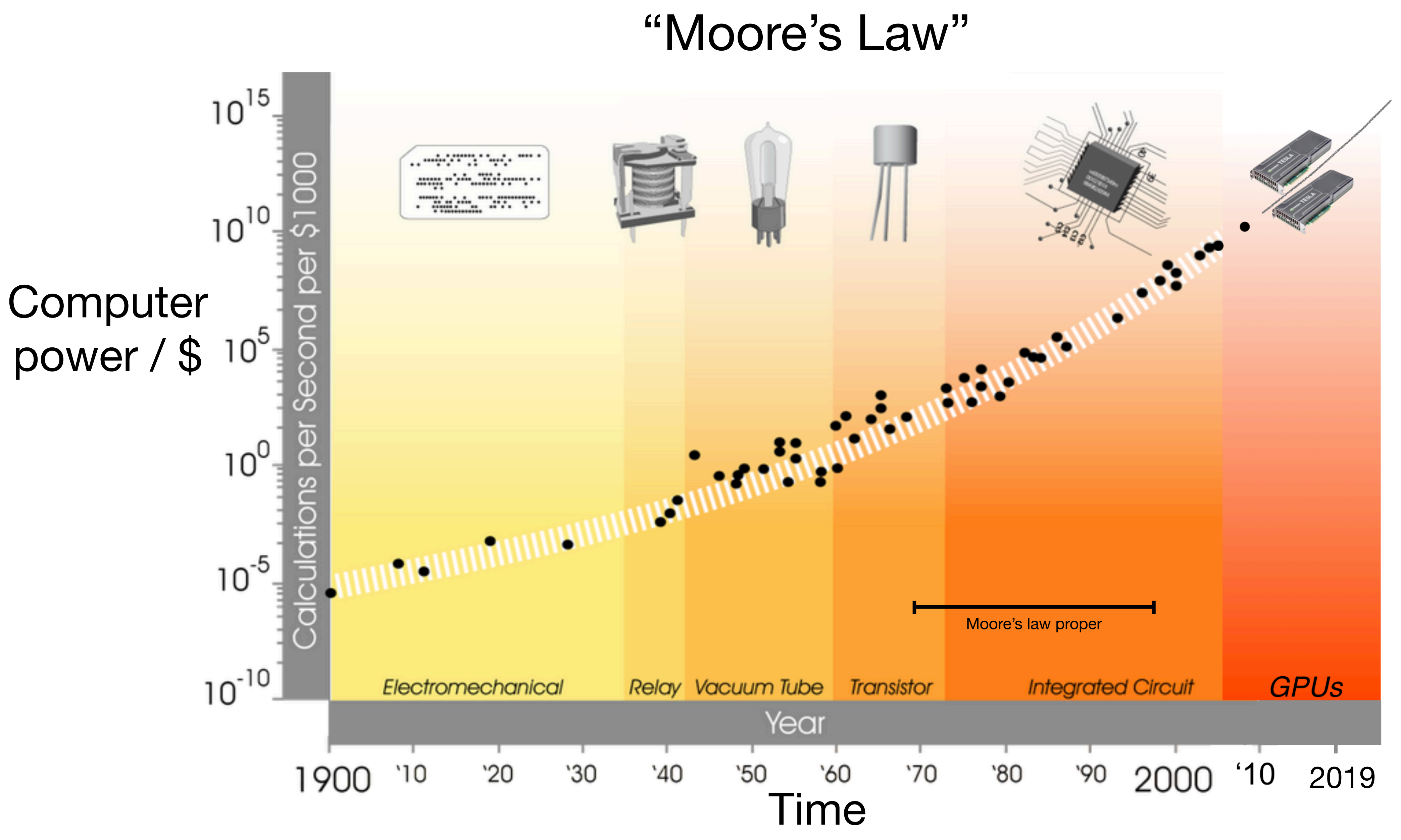
Are You Ready to Fully Embrace Approximation?

Rich Sutton
DeepMind, Amii, RLAI, and UAlberta

Machine Intelligence Today



- Increasing computational power (Moore’s Law) drives progress
- Methods that *scale with computation* are the most impactful
- Thus the current successes of machine learning and deep learning

Scaling with computation is new. Not the usual in CS

- **The usual scaling** is scaling with problem size
    - bigger problem ⇒ more computation needed *to solve it exactly*
- Now we assume the problem could *never* be solved exactly
- **The new scaling** is scaling with computation
    - more computation ⇒ a better approximate answer

We need methods that scale with increasing computation

We need methods that scale with increasing computation

Search and Learning.

We need methods that scale with increasing computation
to better approximate answers.

Search and Learning.

We need methods that scale with increasing computation
to better approximate answers.

Search and Learning. With approximation.

RL has scaled with computation pretty well

- It has embraced function approximation.
- It has embraced Deep Learning.
- It has embraced learning from unprepared experience.
- It has embraced search, particularly MCTS.
- It has embraced replay and (to some extent) planning.
- All these things scale with computational resources

But RL has held back.
It has not fully embraced approximation

- RL is grounded in finite MDPs and tabular methods
- To really abandon finite MDPs challenges us psychologically, requires strong discipline
- If we fully embraced approximation we would lose so much!
  - We lose discounted reward and all the theory built on it
  - We lose Bellman Errors
  - We lose Markov state, thus transition probabilities and expectations, including all true value functions v_π, v_*, q_π, q_*

How has RL dealt with the loss?

"The five stages of grieving"

Denial

Anger

Bargaining

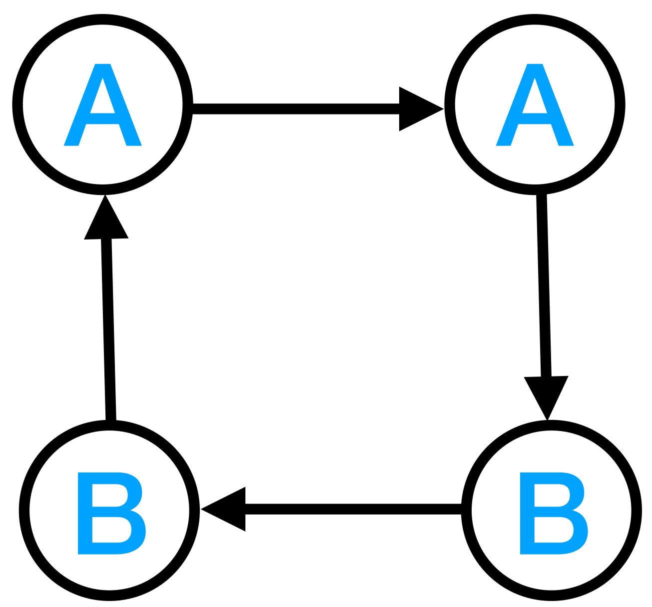
Depression

Acceptance

Approximation in Reinforcement Learning

- World (environment) states map to feature vectors $\phi_t = \phi(S_t) \in \Re^d, d \ll |\mathcal{S}|$
- Then all agent operations use only the feature vectors ϕ_t
- Thus, we may talk about a value function $\hat{v}_{\mathbf{w}}(s)$, but really it is $s \rightarrow \phi \rightarrow \hat{v}$
- Note ϕ_t is not Markov;
  what happens next will depend on past feature vectors (and actions)
  - e.g., Pr$\left[\phi_{t+1} = \phi' \mid \phi_t = \phi\right]$ is not defined

AABBAABBAABB



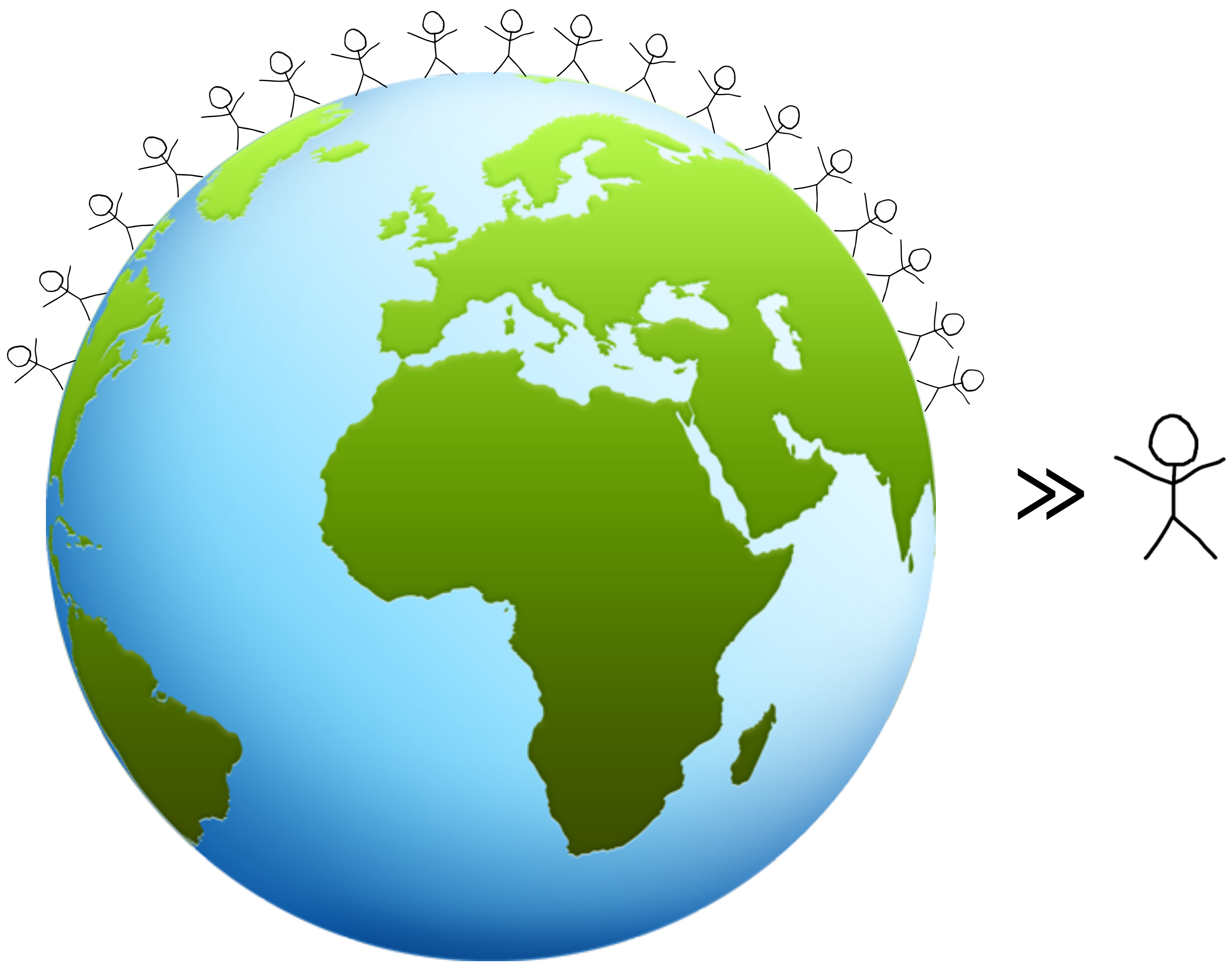
Approximation in Reinforcement Learning (2)

- World states map to feature vectors $\phi_t = \phi(S_t) \in \Re^d, d \ll |\mathcal{S}|$
- Note that there may be as many as $|\mathcal{S}|$ different feature vectors
- Thus the feature vectors cannot be treated as individuals in any way (they must be processed parametrically)
  - e.g., we couldn't approximate $\Pr\left[\phi_{t+1} = \phi' \mid \phi_t = \phi\right]$ (even if it made sense) because you would have to store things for each ϕ
  - and it would depend on the behavior policy

Fully embracing approximation means

- the agent can’t store things for individual states
- the agent can’t do anything that treats individual feature vectors distinctly
- the state the agent works with will not be Markov
- never converging to the exactly correct anything, even in the limit
- the world is much bigger (more complex) than the agent
  - even as the agent’s computational complexity grows exponentially!
- experience is too big to be fully processed by the agent, particularly in real time
- the best approximations will change over time, thus learning must be online

The world is much more complicated than you



- Thus, approximation must be embraced.
- Anything you try to learn can only be learned approximately:
  - value functions,
  - policies,
  - models,
  - states.
- Violating this principle is the most important problem with the use of simulated worlds.

Big world ⇒ apparent non-stationarity

⇒ changing *approximate* value function

Acceptance and opportunity (1):

Function approximation when there is no ideal

- Approximation is okay, we can still do things.
  It's just different. Probably better, certainly real-er.

- Transition probabilities and expectations
  are replaced by a function approximator with a loss

- There are not usable "true" value functions
  - but we can have approximations with a loss
  - and we still do have mean squared return error (for a fixed policy):

$$\mathrm{MSRE}(\underbrace{\mathbf{w}}_{\text{weight vector}}) \doteq \lim_{n\to\infty} \frac{1}{n} \sum_{k=1}^{n} \Big[\underbrace{\hat{v}(S_{t+k}, \mathbf{w})}_{\text{approx value of state}} - \underbrace{G_{t+k}}_{\text{return}}\Big]^2, \text{ if } \underbrace{A_i}_{\text{action}} \text{ were selected} \sim \underbrace{\pi}_{\text{policy}}, \forall i \geq t$$

Acceptance and opportunity (2):

Discounting ⇒ Maximize average reward rate

- All policies π are ranked according to their reward rate:

$$r(\pi) \doteq \lim_{n\to\infty} \frac{1}{n} \sum_{k=1}^{n} \underset{\text{reward}}{R_{t+k}}, \quad \text{if } \underset{\text{action}}{A_i} \sim \pi, \forall i \geq t$$

- Returns are defined relative to $r(\pi)$:

$$G_t \doteq R_{t+1} - r(\pi) + R_{t+2} - r(\pi) + R_{t+3} - r(\pi) + \cdots$$

- Learning and planning algorithms are less developed,

  but Yi Wan and Abhishek Naik have just made good progress (NeurIPS)

Acceptance and opportunity (3):
Feature function ⇒ state-update function

- Instead of an unknowable function ϕ accessing an unknowable world state
- We have a *known* state-update function, operating on *known* experience, with a *known*, improvable objective (summarizing the past to predict the future):

$$\underset{\text{state}}{S_t} = \underset{\text{state-update function}}{u}(S_{t-1}, \underset{\text{action}}{A_{t-1}}, \underset{\text{observation}}{O_t})$$

- This is just a better way to get a non-Markov state
- Our Agent-State Research Group is working on this

Acceptance and opportunity (4):

Converging ⇒ tracking

- Approximation means accepting that the world is big, you can't get anything exactly right
- You could converge to the best approximate static solution, balancing all the errors, or you could track the current best approximation
- Surprisingly, you can *do better by tracking*, maybe *much better* (see ICML2007 paper by Dave Silver, Anna Koop, and me)
- Tracking means learning and relearning, continually, online, like an endless sequence of related learning problems, but all from one base problem
- Thus approximation provides a new basis, a new rationale, for on-line learning, meta learning, generalization, and representation learning!

Conclusion

- Approximation is key to future advances in machine intelligence
- As the premiere RL research institution, we should be leading the advances in approximation within RL
- Approximation seems a difficult challenge, but it is necessary,
  - and will yield great dividends if we fully embrace it
- Fully embracing approximation is on the critical path to the future of machine intelligence