



rlai.net

# Reinforcement Learning and Artificial Intelligence



The RL&AI group at  
the Univ. of Alberta  
in 2011

Principal investigators:  
Rich Sutton  
Michael Bowling  
Csaba Szepesvari  
Dale Schuurmans  
Patrick Pilarski  
et al.

# The Future of Artificial Intelligence

*Rich Sutton*

Reinforcement Learning and Artificial Intelligence Laboratory  
Department of Computing Science  
University of Alberta, Canada



# Outline:

## Understanding AI in the...

- Present
  - Success, excitement, and fear
  - Moore's law (generalized) drives it all
- Past
  - The impact of Moore's law can be seen throughout the history of AI
  - The longest trend: Scalable methods are initially disfavoured, but eventually win
- Future
  - A key remaining challenge: Knowledge (of the world's state & dynamics)
  - How can we make knowledge scalable with Moore's law?

# Advances in AI abilities are coming faster; in the last 5 years:

- IBM's Watson beats the best human players of *Jeopardy!* (2011)
- Deep neural networks greatly improve the state of the art in speech recognition and computer vision (2012–)
- Google's self-driving car becomes a plausible reality ( $\approx$ 2013)
- Deepmind's DQN learns to play Atari games at the human level, from pixels, with no game-specific knowledge ( $\approx$ 2014, *Nature*)
- Univ of Alberta's Cepheus solves Poker (2015, *Science*)
- Deepmind's AlphaGo defeats the European Go champion 5-0, vastly improving over all previous programs (2016, *Nature*)

# Corporate investment in AI is way up

- Google's prescient AI buying spree: Boston Dynamics, Nest, Deepmind Technologies, ...
- New AI research labs at Facebook (Yann LeCun), Baidu (Andrew Ng), Allen Institute (Oren Etzioni), Maluuba...
- Also enlarged corporate AI labs: Microsoft, Amazon, Adobe...
- Yahoo makes major investment in CMU machine learning department
- Many new AI startups getting venture capital

Why are these things  
happening now?

Is it because of big progress in AI  
algorithms?

Or...



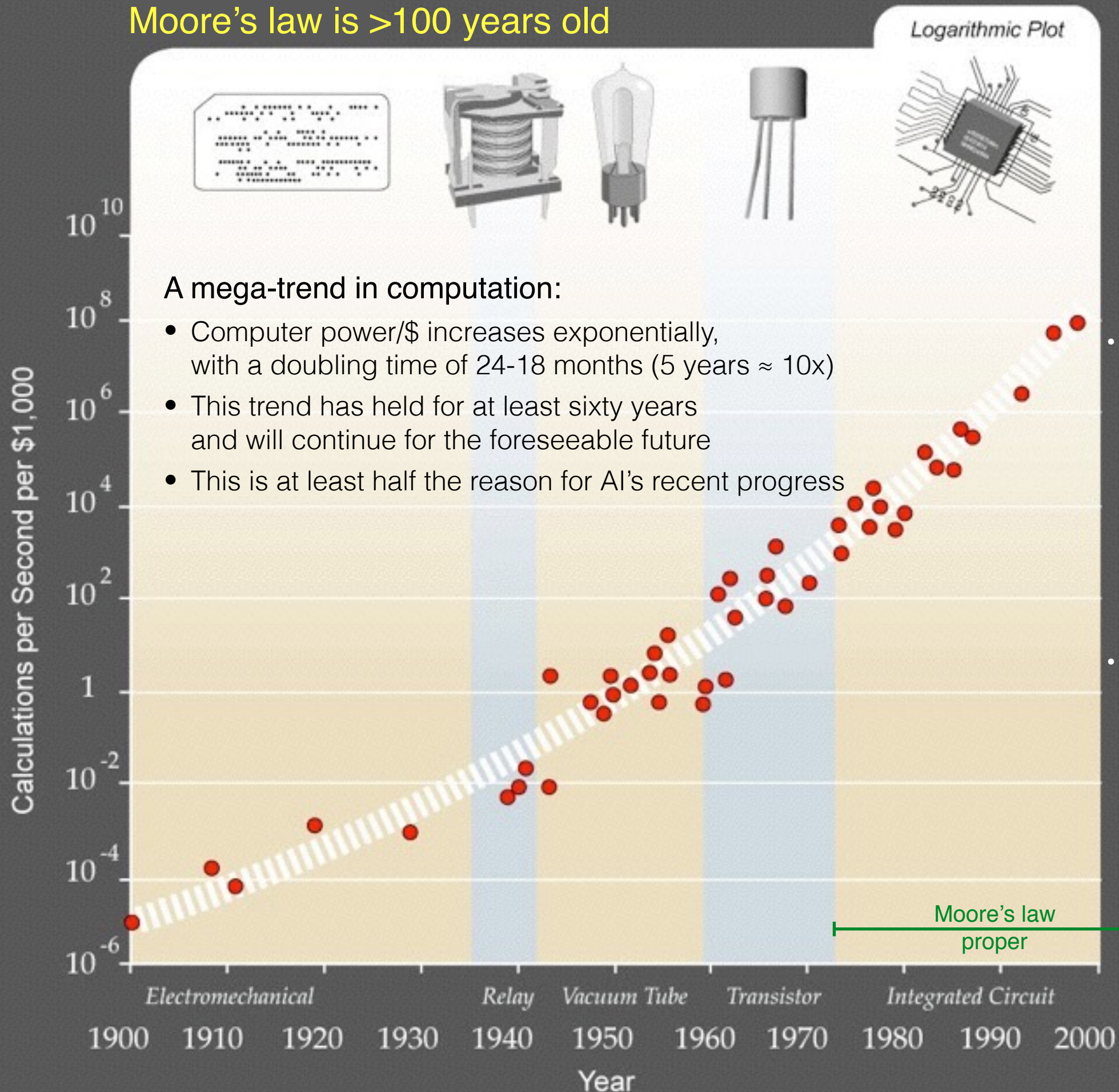
# Moore's Law

The long-term exponential improvement in computer hardware

Moore's law proper: The number of transistors that can be placed inexpensively on an integrated circuit doubles approximately every two years



## Moore's law is >100 years old



### • Why is this happening?

- because we use each generation of computers to create the next
- because it is so economically valuable
- because so many engineers are working on it

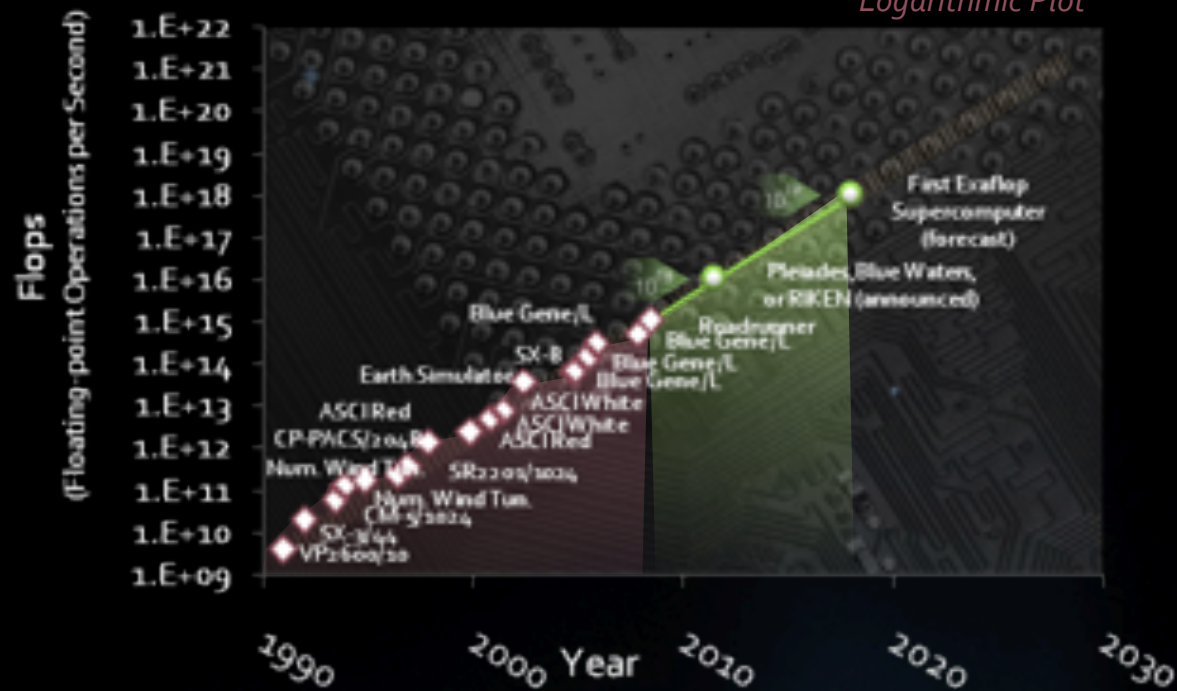
### • Can it really keep going?

- yes, as long as new technologies come along
- as they always have in the past
- the theoretical limits to computation rate are still far away



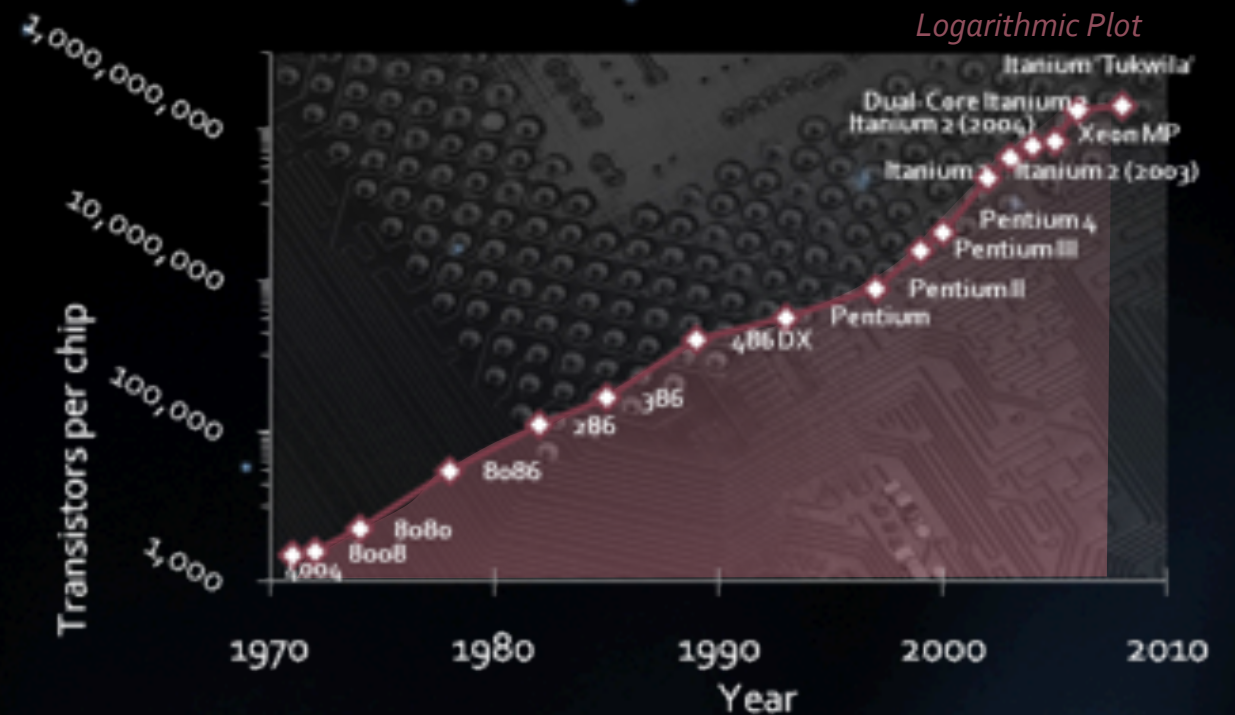
## Growth in Supercomputer Power

Logarithmic Plot



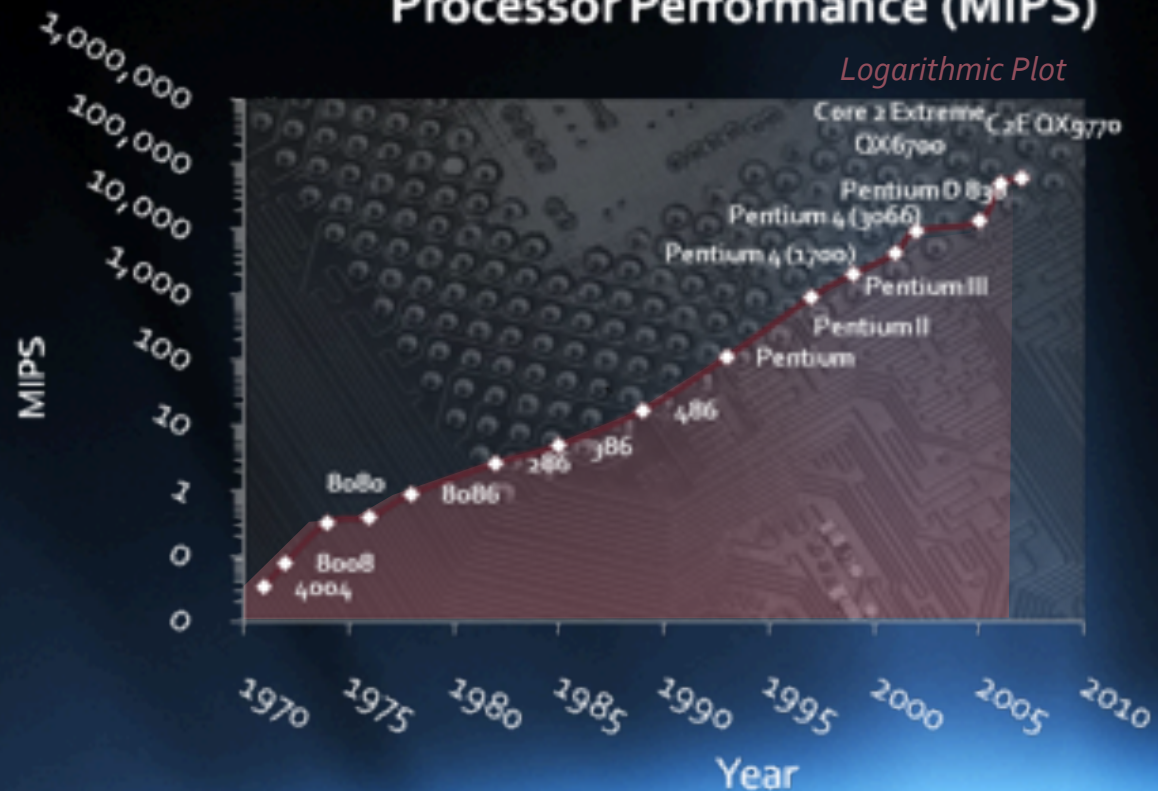
## Transistors Per Chip (Intel processors)

Logarithmic Plot



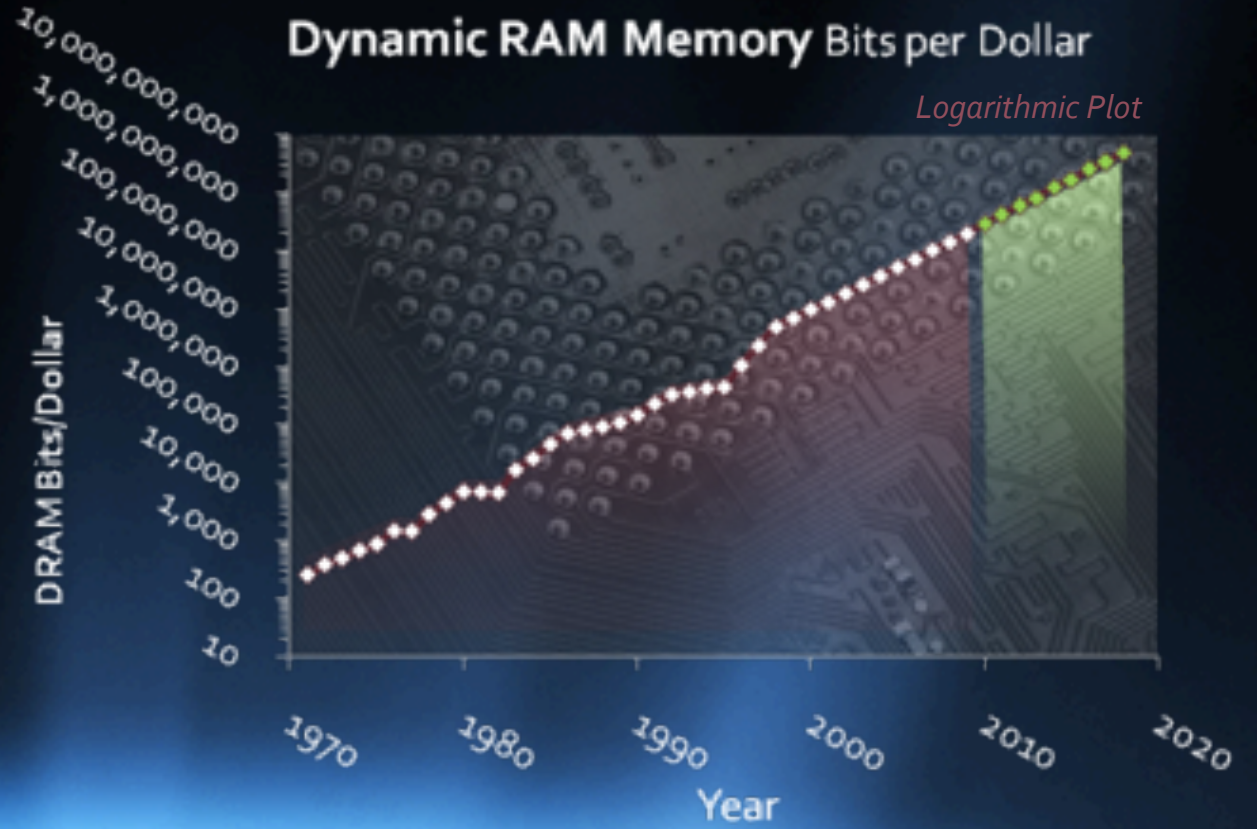
## Processor Performance (MIPS)

Logarithmic Plot



## Dynamic RAM Memory Bits per Dollar

Logarithmic Plot



(from Kurzweil AI)



# Doubling (or Halving times) in 2010

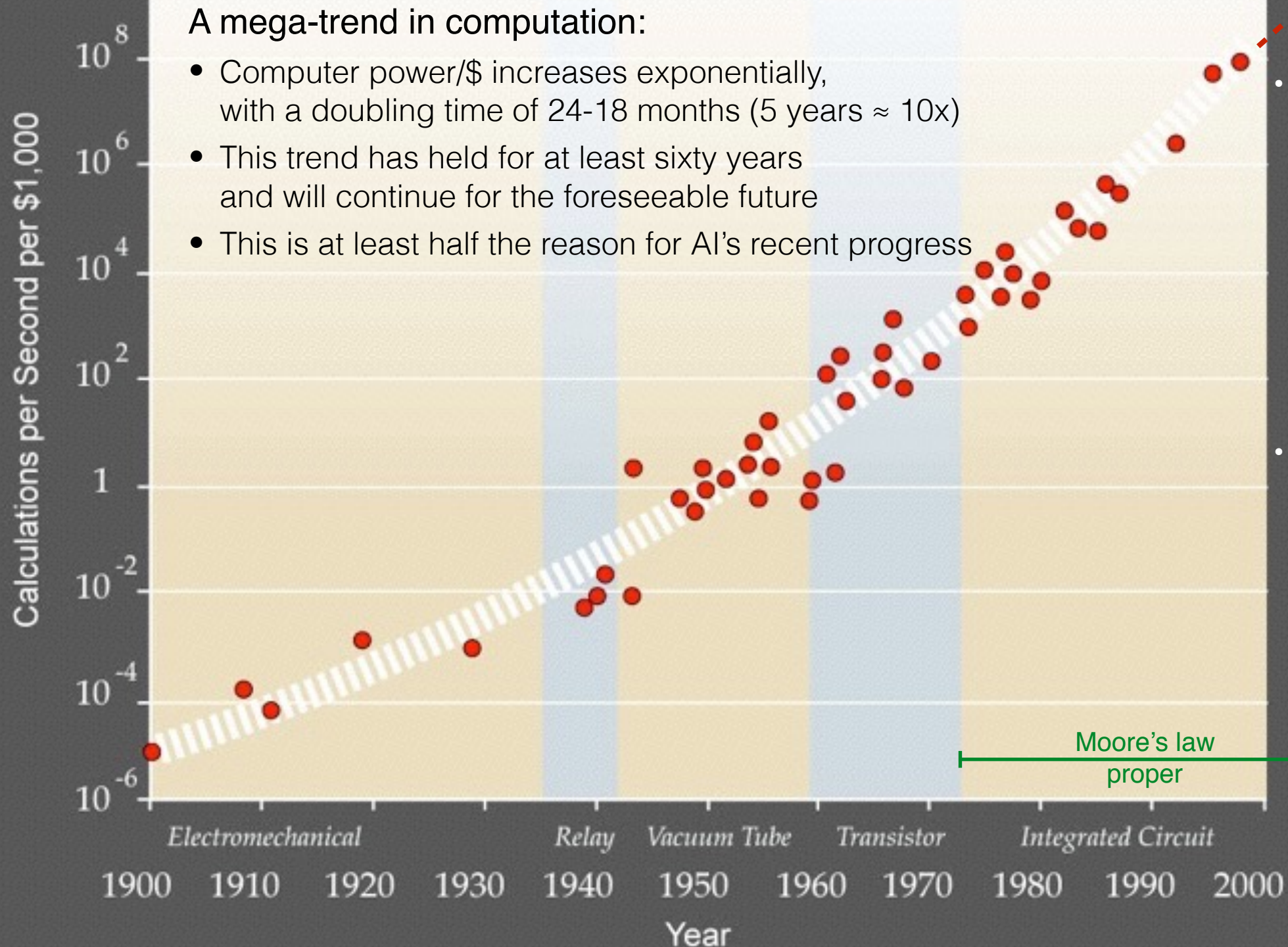
- Dynamic RAM Memory “Half Pitch” Feature Size 5.4 years
- Dynamic RAM Memory (bits per dollar) 1.5 years
- Average Transistor Price 1.6 years
- Microprocessor Cost per Transistor Cycle 1.1 years
- Total Bits Shipped 1.1 years
- Processor Performance in MIPS 1.8 years
- Transistors in Intel Microprocessors 2.0 years
- Microprocessor Clock Speed 2.7 years

## Kurzweil’s Law of Accelerating Returns:

An analysis of the history of technology shows that technological change is exponential  
Information technology, in particular, exhibits fast, long-lasting exponential improvements

(from Kurzweil AI)

# Moore's law is >100 years old



## • Why is this happening?

- because we use each generation of computers to create the next
- because it is so economically valuable
- because so many engineers are working on it

## • Can it really keep going?

- yes, as long as new technologies come along
- as they always have in the past
- the theoretical limits to computation rate are still far away



# Fear of AI is also up

- Many people fear the success of AI, that it may be unsafe and threaten humanity
- One fear is that AIs will be much smarter than us
  - Nick Bostrom, author of “Superintelligence: Paths, Dangers, Strategies,” worries that the first strong AI might take over and cause an “existential catastrophe”
  - Elon Musk — “[Strong AI would be] releasing the demon” “our greatest existential threat” “there should be some regulatory oversight” “I think there is potentially a dangerous outcome there”
  - Stephen Hawking — “The development of full artificial intelligence could spell the end of the human race” “It would take off on its own, and re-design itself at an ever increasing rate” “world militaries are considering autonomous-weapon systems that can choose and eliminate targets” “humans, limited by slow biological evolution, couldn’t compete and would be superseded by AI”
- AI researchers are sometimes too dismissive of these fears
  - Andrew Ng compares worrying about strong AI to worrying about over-population on Mars
  - Geoff Hinton says that if strong AI does ever happen it won’t be for a long while

# An imaginary conversation

Smart Fearful Person: Eventually the machines will be smarter than us!

Me: Yes, at that time we may have to either (a) subjugate them, or (b) risk that they subjugate or destroy us

SFP: We must ensure that we can subjugate them!

...even though it seems morally questionable

...and may be very hard to maintain indefinitely

...and even though if they do escape, then they will be pissed at us

*This is our fear!*

Me: Or, we could choose option (b) and not have to worry about all that  
What might happen then?

We may still be of some value and live on

Or we may be useless and in the way, and go extinct

Both of these may be preferable to option (a) *succeeding*

# My view

- Understanding human-level AI will be a profound scientific achievement and economic boon which may well happen by 2030 (25% chance) or 2040 (50% chance) — or never (10% chance)
- It will bring great changes! We should certainly prepare ourselves
- But the fear is overblown, unhelpful, misplaced, and poorly expressed
  - AI will arrive much slower than feared, at the rate of Moore's law
  - The greatest risks come not from AI as much as from the people who would misuse it; this is a pre-existing, ongoing problem with our societies
    - The problems that need solving are not primarily technical or mathematical, but societal
  - One big fear is that strong AIs will escape our control; this is likely, but not to be feared



# Four metaphors for the impact of AI on humanity

1. Ho hum. It's just another round of technology
2. Yikes! It's the end of humanity!
3. Wow. It could be the next step for humanity
4. Hmm. It could be quite complex and diverse

Several of these may happen, one after the other, or even at the same time

in conclusion, about the present:

# AI is not like other sciences

- AI has Moore's law, an enabling technology racing alongside it, making the present special
- Moore's law is a slow fuse, leading to the greatest scientific prize of all time
- So slow, so inevitable, yet so uncertain in timing
- The present is a special time for humanity, as we prepare for, wait for, and strive to create strong AI

# Outline:

## Understanding AI in the...

- Present

- Success, excitement, and fear
- Moore's law (generalized) drives it all

- Past

- The impact of Moore's law can be seen throughout the history of AI
- The longest trend: Scalable methods are initially disfavoured, but eventually win

- Future

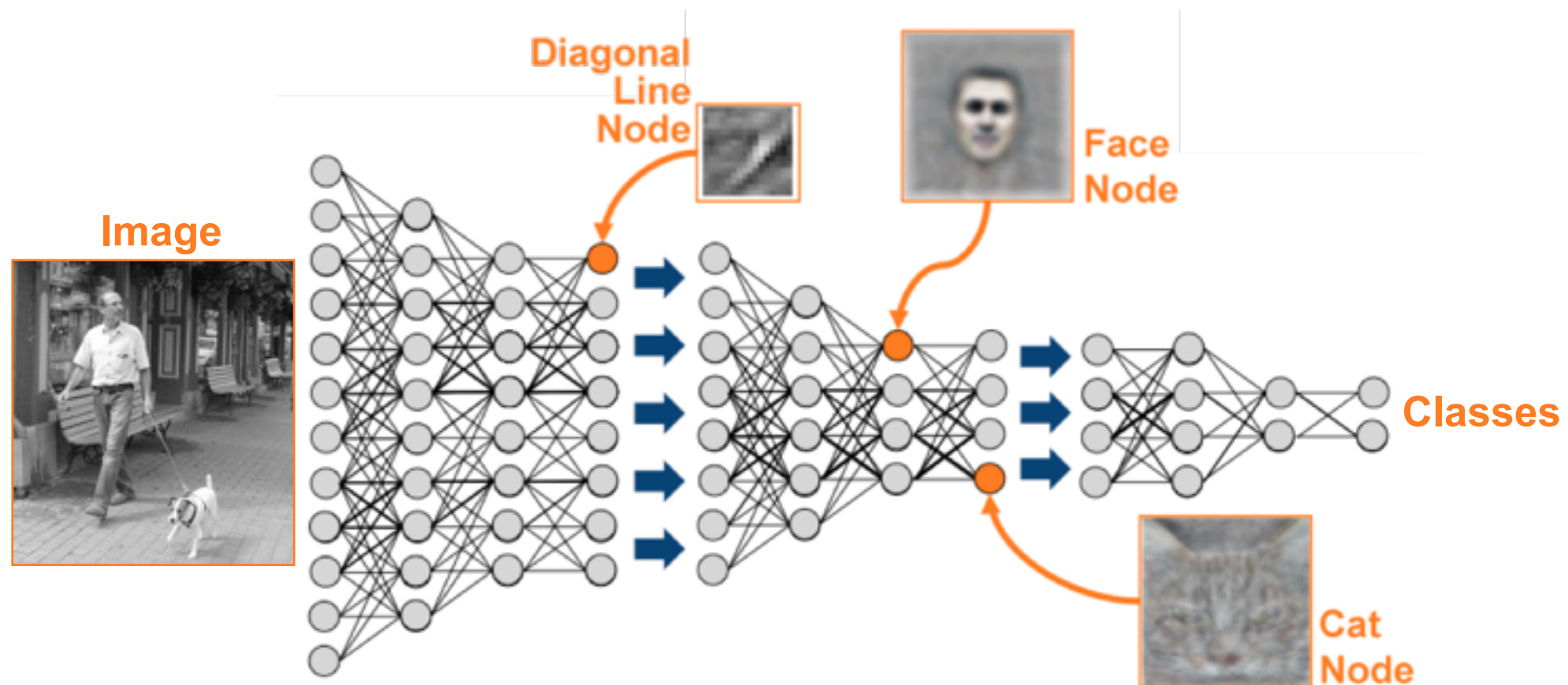
- A key remaining challenge: Knowledge (of the world's state & dynamics)
- How can we make knowledge scalable with Moore's law?



# Deep learning

≡ deep neural networks

≡ multi-layer neural networks with many layers



- Each line has a learned connection weight
- Each node combines its weighted inputs, then applies a nonlinear transformation
- For each image, the network produces class labels as output, and true class labels are provided by people (supervised learning)
- Then each weight is incremented so as to reduce the squared error (stochastic gradient descent, backpropagation)

# 3 waves of neural networks

- First explored in the 1950-60s: Perceptron, Adaline...
  - only one learnable layer
- Revived in the 1980-90s as Connectionism, Neural Networks
  - exciting multi-layer learning using backpropagation (SGD); many successful applications; remained popular in engineering
- Revived again in ~2010 as Deep Learning
  - dramatically improved over state-of-the-art in speech recognition and visual object recognition, *transforming these fields*
  - the best algorithms were essentially the same as in the 1980s, except with faster computers and larger training sets

i.e., NNs won (eventually) because their performance scaled with Moore's law, whereas competing methods did not

# Visual object recognition (crudely)

- Objective: Given an image, label and locate the objects within it
- Input: Many images with the objects labeled
- Early methods (dating back to the 1960s) used CAD-like models of the objects, or generalized-cylinder models, geometric models
- Later methods use more generic features, like edges, gradients, Hessian and difference-of-Gaussian detectors, then SIFT and SURF features, and finally matched to models or dictionaries; these methods scaled better and eventually worked better
- All this is thrown out in deep learning, which performs better and is easier to design. Features are learned instead of being built in. The only things built in are invariance to translation and scale.



# Scalable methods

- A method is *scalable with the computational mega-trend (Moore's law)* to the extent that its performance improves roughly in proportion to the quantity of computation it is given
- Scalable means you can take advantage of (use effectively) an arbitrarily large amount of computation (e.g., learning, search)
- A method is not scalable if the improvement it gives is not much affected by the computation available (e.g., the opening book in chess)
- Search and learning are scalable; prior knowledge, human assistance, and taking advantage of special-case structure are not
- By definition then, scalable methods improve automatically with time; they tend to be disfavoured initially but perform better in the end
- This is a pattern that can be seen over and over in the history of AI

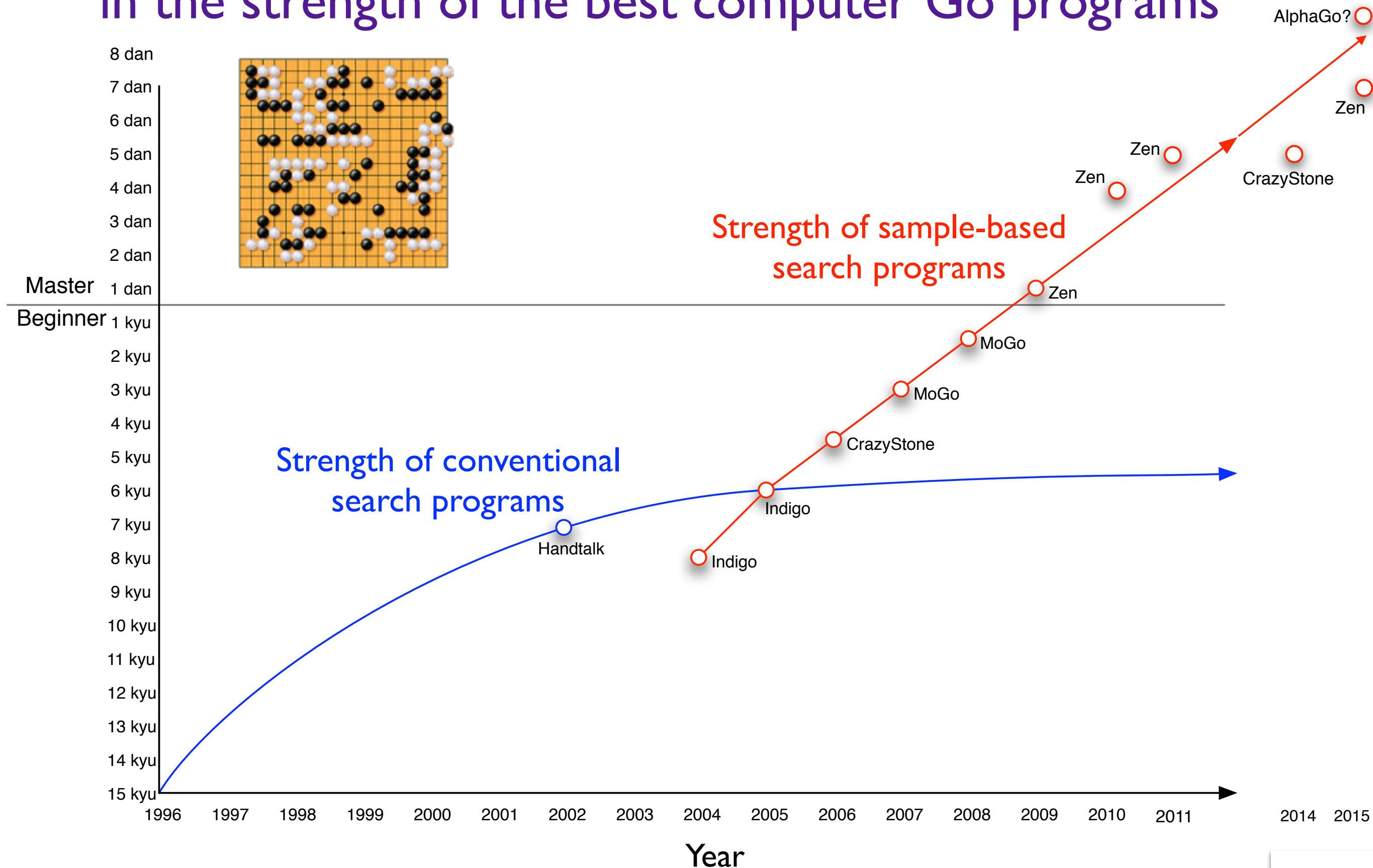
# Computer chess

- Early computer pioneers had hoped to program computers to play chess much like humans do, by relying primarily on chess heuristics (rules of thumb) to choose the best moves –The Computer History Museum
- By the 1970s, a new generation of chess machines arose that gave up on playing like people and focused on optimizing search
  - this was controversial, and the results were initially mixed
- In 1997, IBM's Deep Blue machine, using specialized hardware to do a full width search (all moves considered), defeated Gary Kasparov, the reigning world chess champion
  - At the time, many found Deep Blue's victory unsatisfying (calling it a "brute force" solution and "not the way people play chess")
- The real lesson: Scalable methods eventually win

# Computer Go

- Chess machines were based on  $\alpha$ - $\beta$  **search** and **state-evaluation functions**, but neither of these worked well for Go
- Again, heuristic methods were tried and gave modest improvements, but not strong play
- In 2006, a new kind of search based on running sample trajectories to the end of the game, called Monte Carlo Tree Search (MCTS), was introduced, greatly improved performance, and transformed the field
  - Almost all the heuristics of previous programs were left out in MCTS
- In 2016, deep learning and reinforcement learning were used to find an effective state-evaluation function, dramatically improving performance

# Steady, exponential improvement (since MCTS, 2006) in the strength of the best computer Go programs





# Scalability is the key, but it tends to be correlated with other issues

- **Symbolic** vs **statistical**,  
**hand-crafted** vs **learned**,  
**domain-specific** vs **general-purpose**
  - **Symbolic, hand-crafted, and domain-specific** methods all rely more on human understanding and participation in their design; they begin non-scalable and tend to stay that way
  - Over the history of AI, **statistical, learned, and general-purpose** methods have steadily increased in relative importance
- In the early days of AI (pre-1980), a similar distinction was made between **“strong”** methods (powered by human input) and **“weak”** methods (relying on general principles)
  - The terminology is telling; the founding fathers favorite methods failed to scale and have fallen from favor; the **weak** have inherited AI

in conclusion, about the past

# The choice is always before the AI researcher: To work on what scales, or what does not?

- We almost always reach for what does not scale
- It is usually easier, less abstract, and quicker to payoff
  - Improving a scalable method may bring little payoff for years
- But Moore's law is progressing; with each further doubling in computation the relative advantage of scalable methods increases and becomes more quickly visible
- If you want to have a long-term impact, you should work on methods that scale with computation; timing is important

# Outline:

## Understanding AI in the...

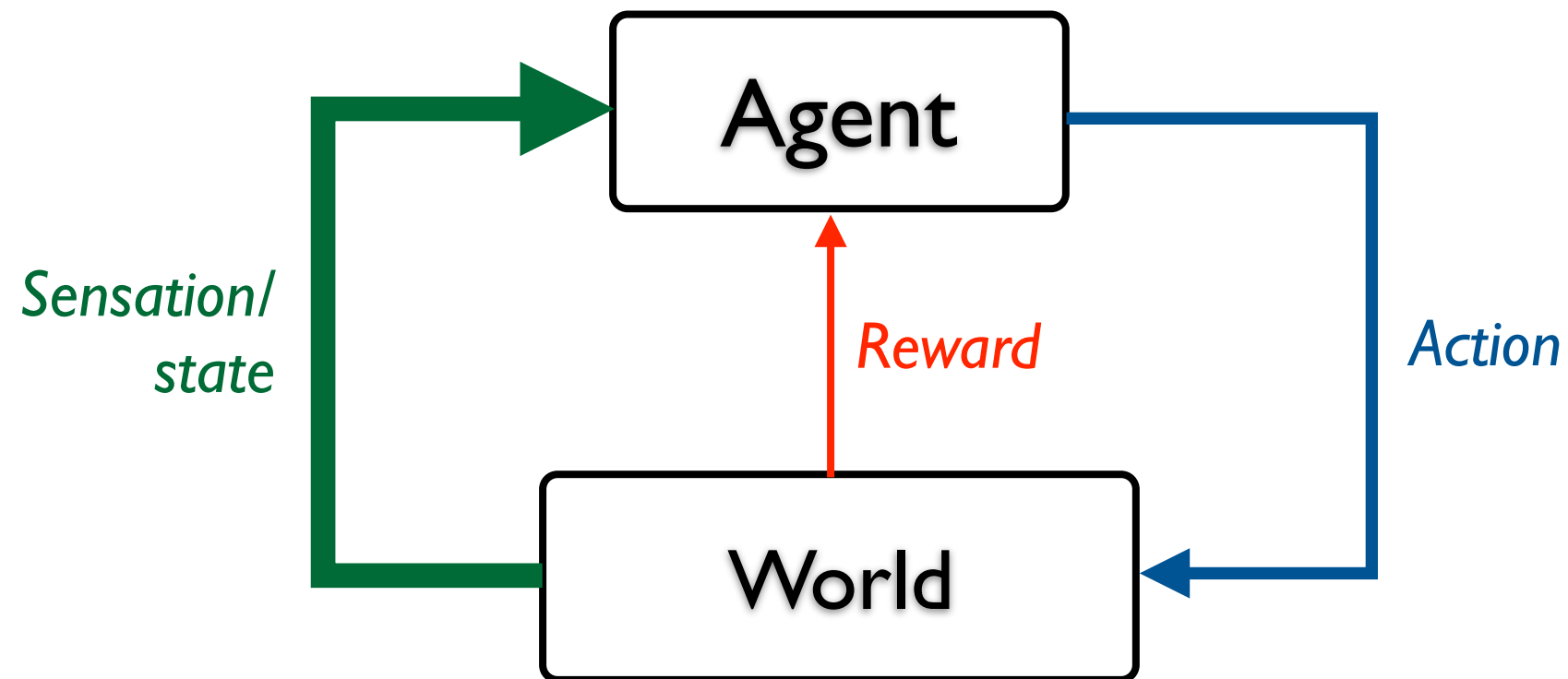
- Present
  - Success, excitement, and fear
  - Moore's law (generalized) drives it all
- Past
  - The impact of Moore's law can be seen throughout the history of AI
  - The longest trend: Scalable methods are initially disfavoured, but eventually win
- Future
  - A key remaining challenge: Knowledge (of the world's state & dynamics)
  - How can we make knowledge scalable with Moore's law?

# How scalable is reinforcement learning?

- In classic, model-free RL, we learn a policy (a mapping from states to actions) and a value function (a mapping from states to future reward)
  - these can be learned by trial and error, by trying actions and seeing what rewards follow
  - no labels are required (good for scaling)  
and it is computationally cheap (good? or bad?)
- If experience is plentiful (e.g., self-play) then RL scales beautifully
- But in the classic, model-free case, you do just a small computation per time step, and then there is nothing else to do; there is little scaling (the policy and value mappings can be made more complex)

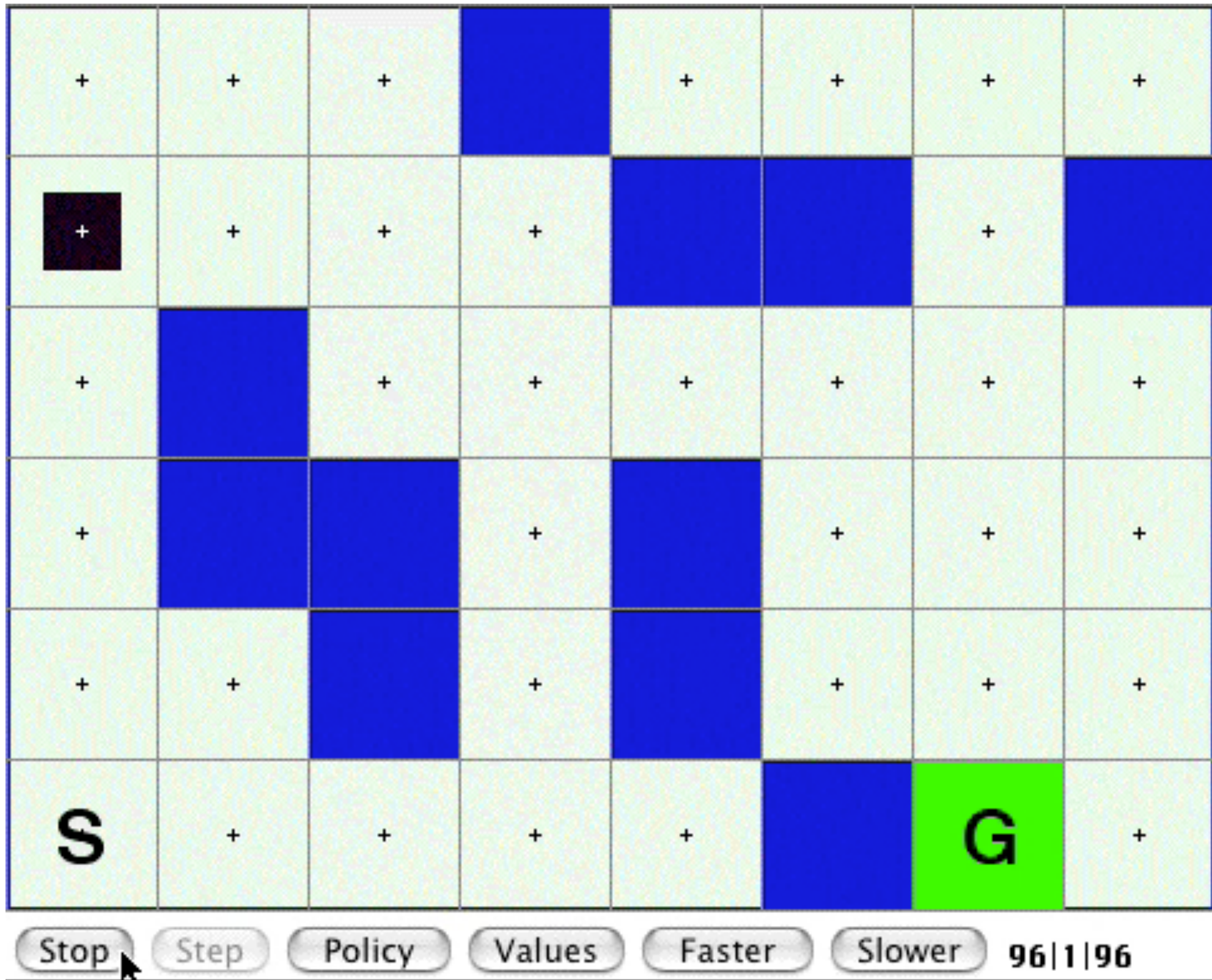
Not so much

# Reinforcement learning





# Model-based RL: GridWorld Example



# The grand challenge of knowledge

- By knowledge I mean empirical knowledge of the world
  - Analogous to the laws of physics,  
to knowing how the pieces move in chess,  
to knowing what causes what,  
to being able to predict what will happen next for various actions
- The knowledge must be
  - Expressive: able to represent all the important things, including abstractions like objects, space, people, and extended actions
  - Learnable: from data without labels or supervision (for scalability)
  - Suitable for supporting planning/reasoning
- There is a substantial body of technical machinery for this in RL (options, PSRs, TD nets); but I will just sketch the challenge and its scalability



# Skilled perception and action...learned without labels



TWO  
40x Slower



a view of (empirical) knowledge:

# Knowledge is about the world's state and dynamics

- **State** is a summary of the agent's past that it uses to predict its future
- To have **state knowledge** is to have a *good summary*, one that enables the predictions to be accurate
- The predictions themselves are the **dynamics knowledge**
- The most important things to predict are *states* and *rewards*, which of course depend on what the agent does
  - if these are predicted in the right way, then the predictions can be used as a **model of the world** to support planning (the analog of self-play and reasoning)
- How can such knowledge be learned, represented, and used in a scalable way?

# The one-step trap:

Thinking that one-step predictions are sufficient

- That is, at each step predict the state and observation *one step later*
- Any long-term prediction can then be made by simulation
- In theory this works, but not in practice
  - Making long-term predictions by simulation is exponentially complex
  - and amplifies even small errors in the one-step predictions
- Falling into this trap is very common: POMDPs, Bayesians, control theory, compression enthusiasts



# Predicting right and left bumps conditional on going forward



pred data



left bump

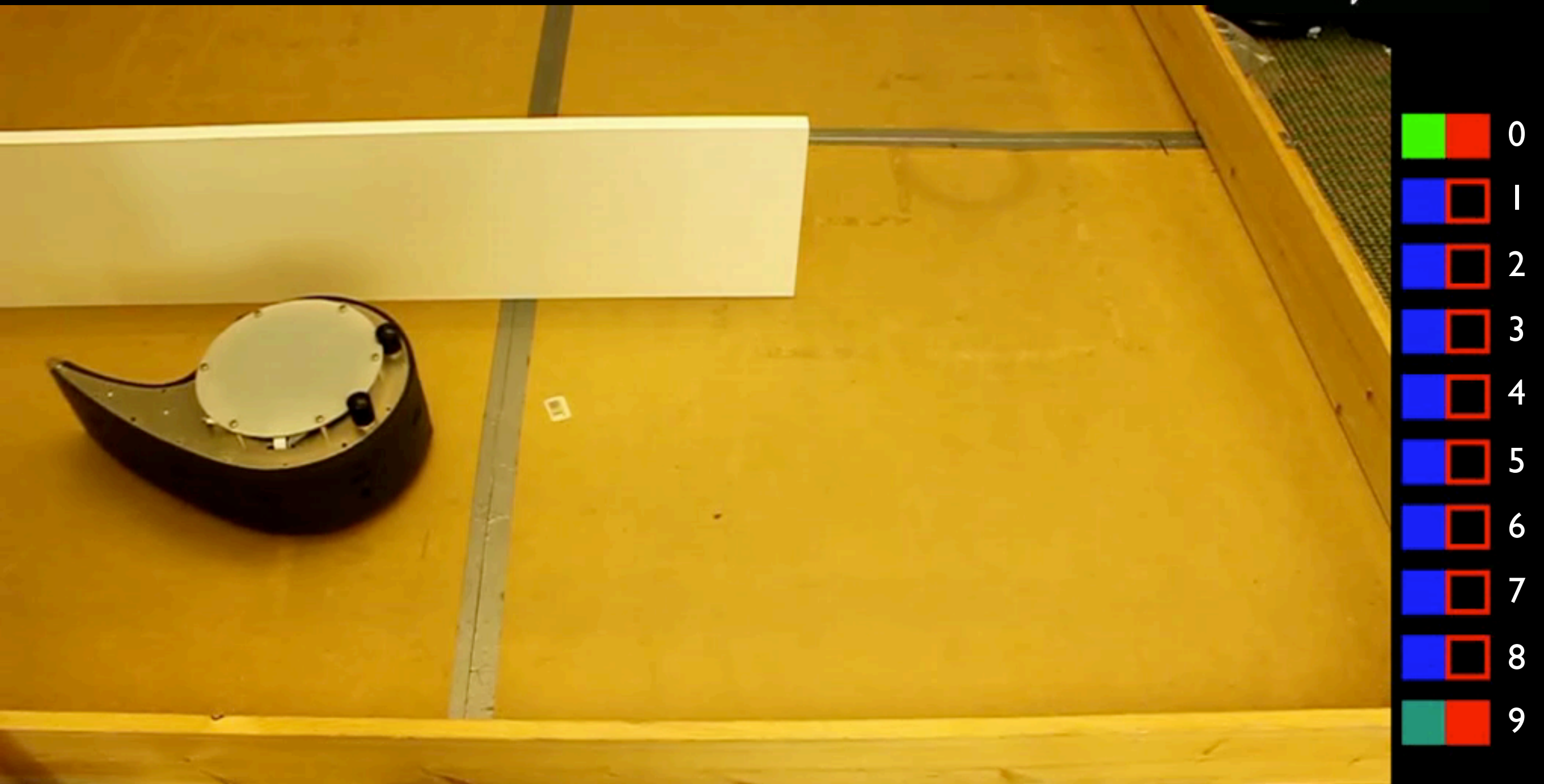
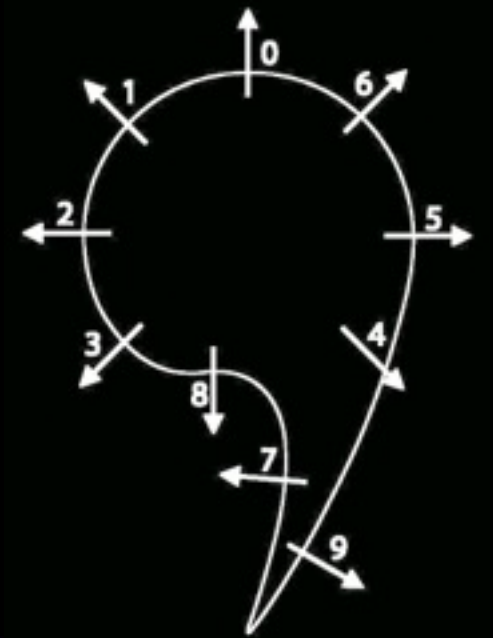






















both bump



right bump

# Predicting 10 infrared proximity signals



		0
		1
		2
		3
		4
		5
		6
		7
		8
		9

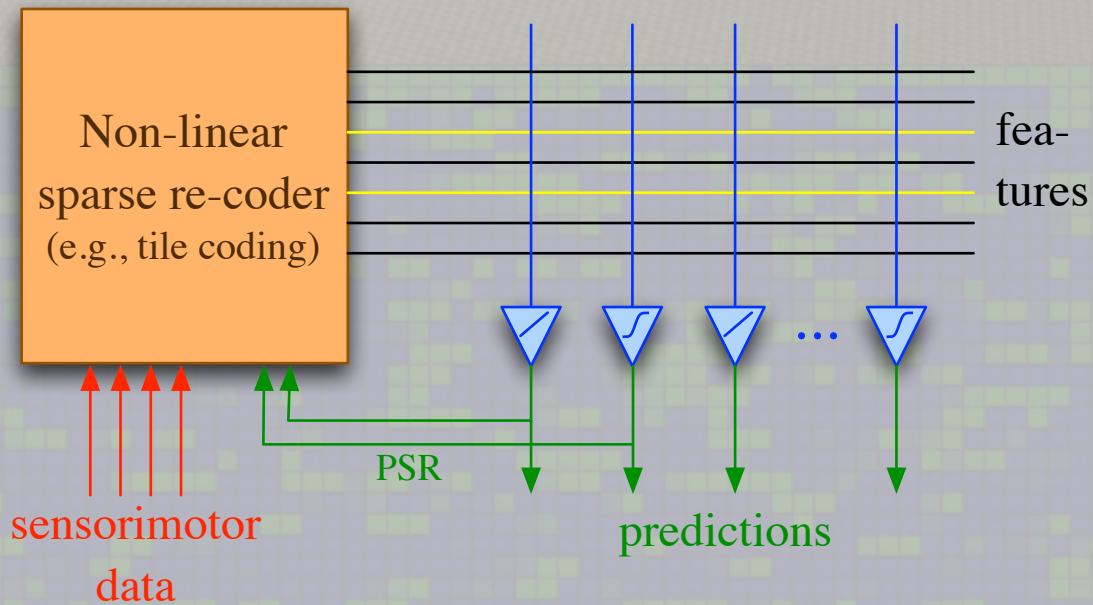


# Massive real-time prediction learning

## Up to one billion weight updates/second

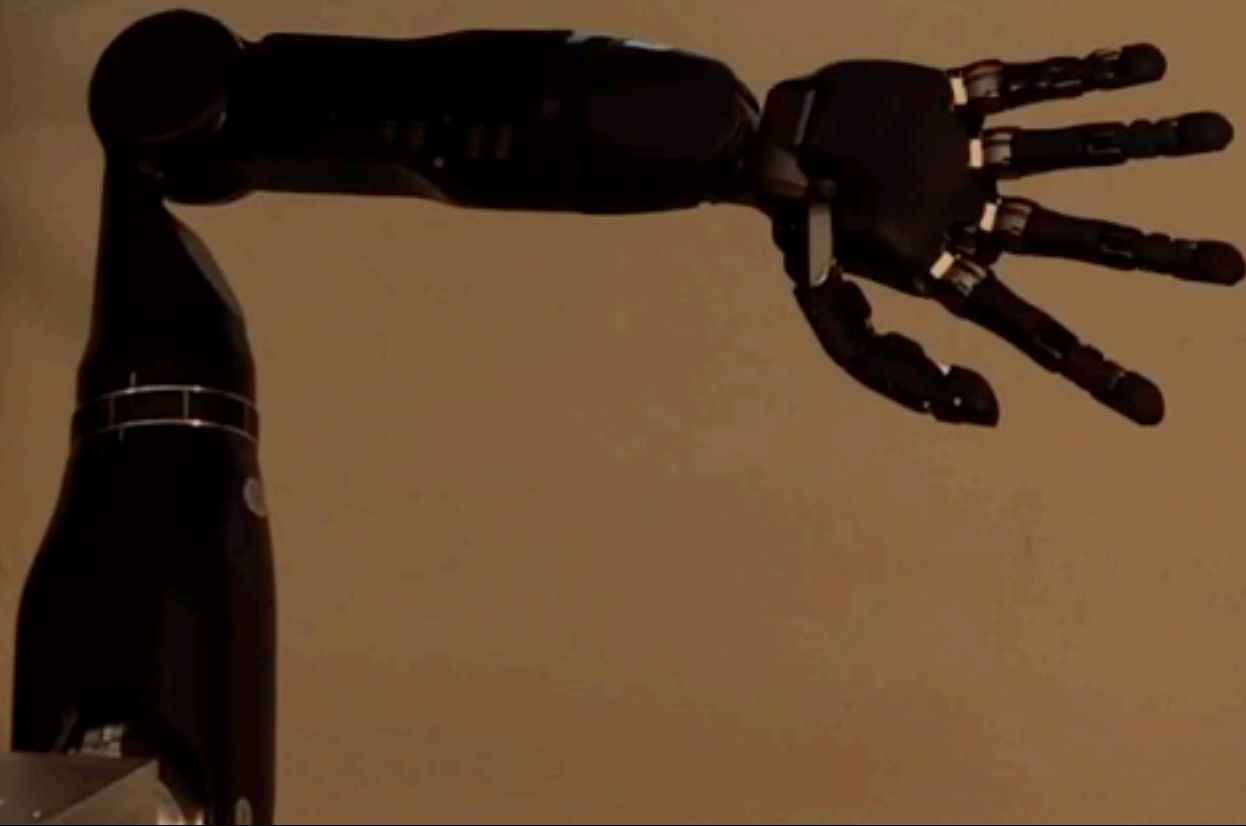
continuous observation data x 69

sparse binary  
features x 3200  
(tile coding)



predictions  
x 6000





Real-time  
prediction learning  
on a prosthetic arm



# An old, ambitious goal:

To understand the world in terms of sensorimotor data

- Making predictions at multiple levels of abstraction
- Finding the abstractions that carve the world at its joints
- Expressing cause and effect compactly, supporting planning and decision making
- This goal is well suited to scaling
  - it can utilize arbitrarily large quantities of computation in learning the predictions and in searching for the best abstractions, yet has no minimum requirement

# New tools

- General value functions (GVFs) provide a uniform language for efficiently learnable predictive knowledge
- Options and option models (temporal abstraction)
- Predictive state representations
- New off-policy learning algorithms (gradient-TD, emphatic-TD)
- Temporal-difference networks
- Deep learning, representation search
- Moore's law!

# Conclusion:

## Moore's law strongly impacts AI

- It makes the present special, as hardware races alongside the algorithmic developments
- In the past, it has caused scalable methods to have the greatest long-term impact
- These lessons should guide our future research
- Our plans should be ambitious, scalable, and patient/stubborn
  - Like my plan for a sensorimotor understanding of the world

# Thank you for your attention

and thanks to



Rupam Mahmood, Adam White, Joseph Modayil,  
Harm van Seijen, Doina Precup, Hado van Hasselt,  
Thomas Degris