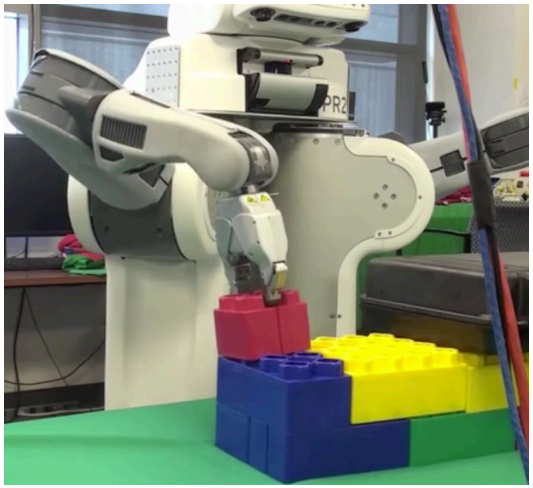
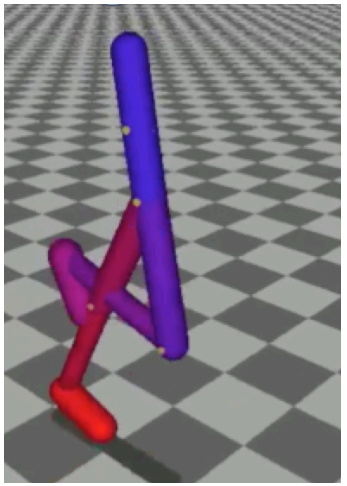
Deep Reinforcement Learning for Manipulation, Locomotion, and Then Some

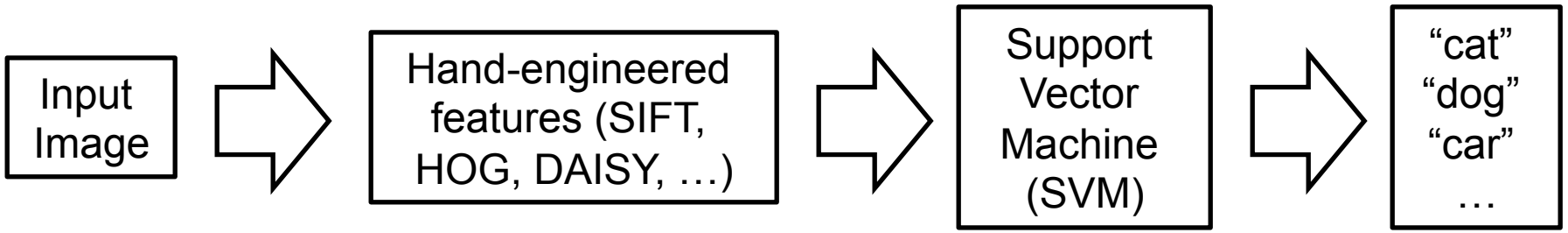


Pieter Abbeel
UC Berkeley EECS

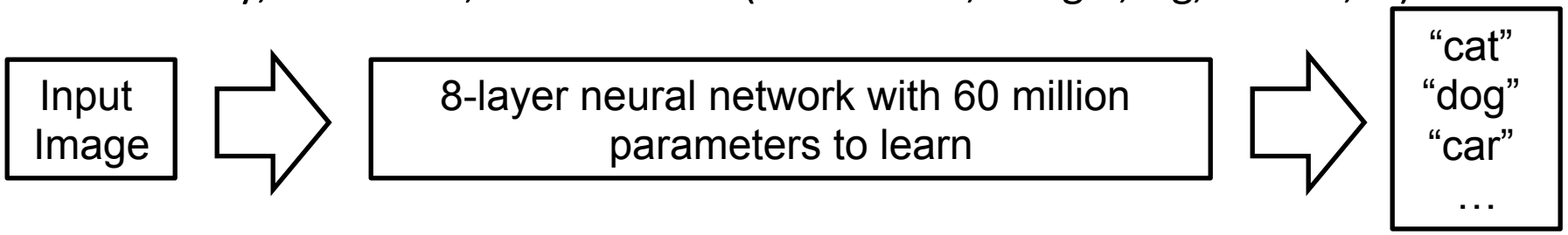


Object Detection in Computer Vision

- State-of-the-art object detection until 2012:

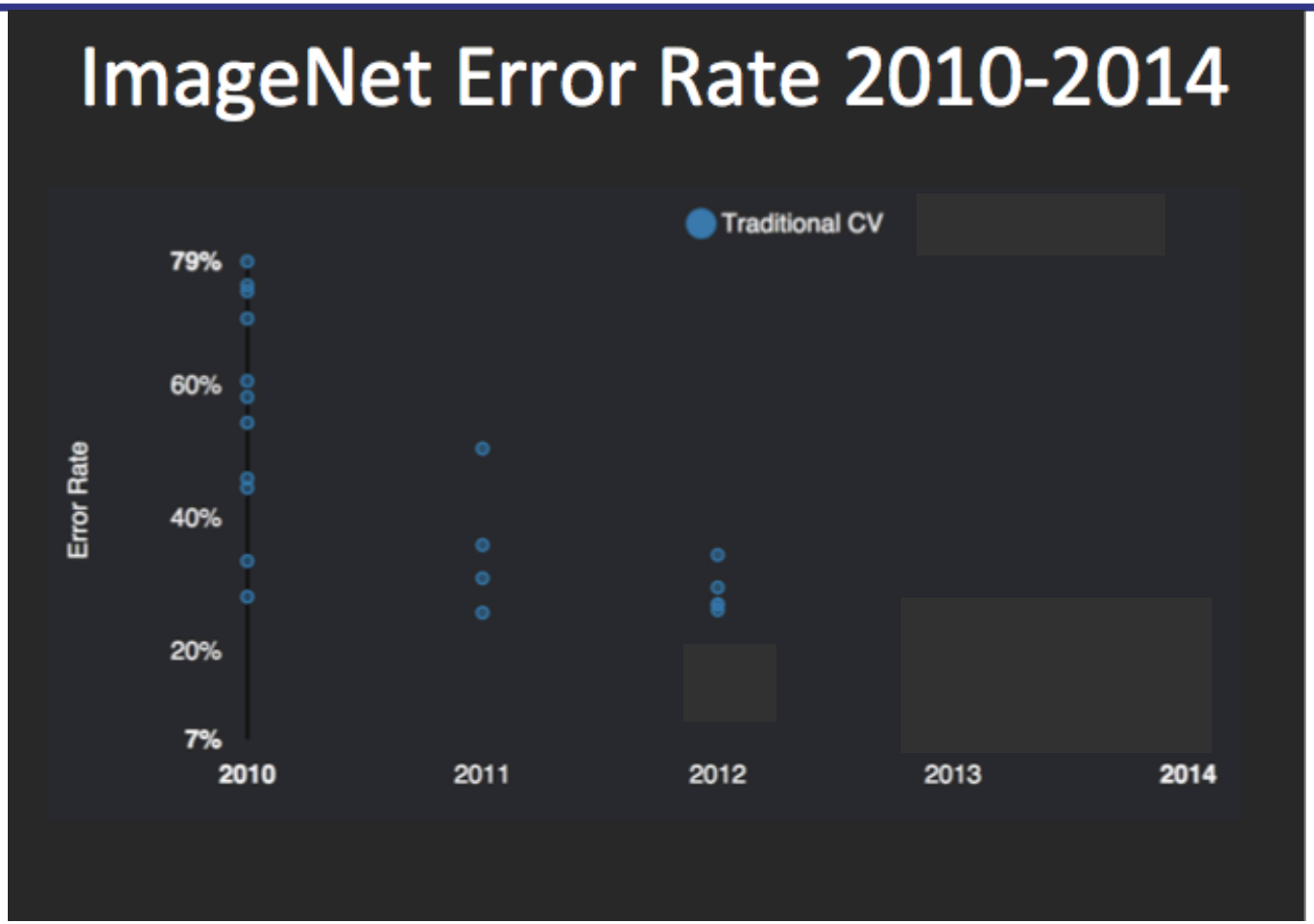


- Krizhevsky, Sutskever, Hinton 2012 (also: Lecun, Bengio, Ng, Darrell, …):



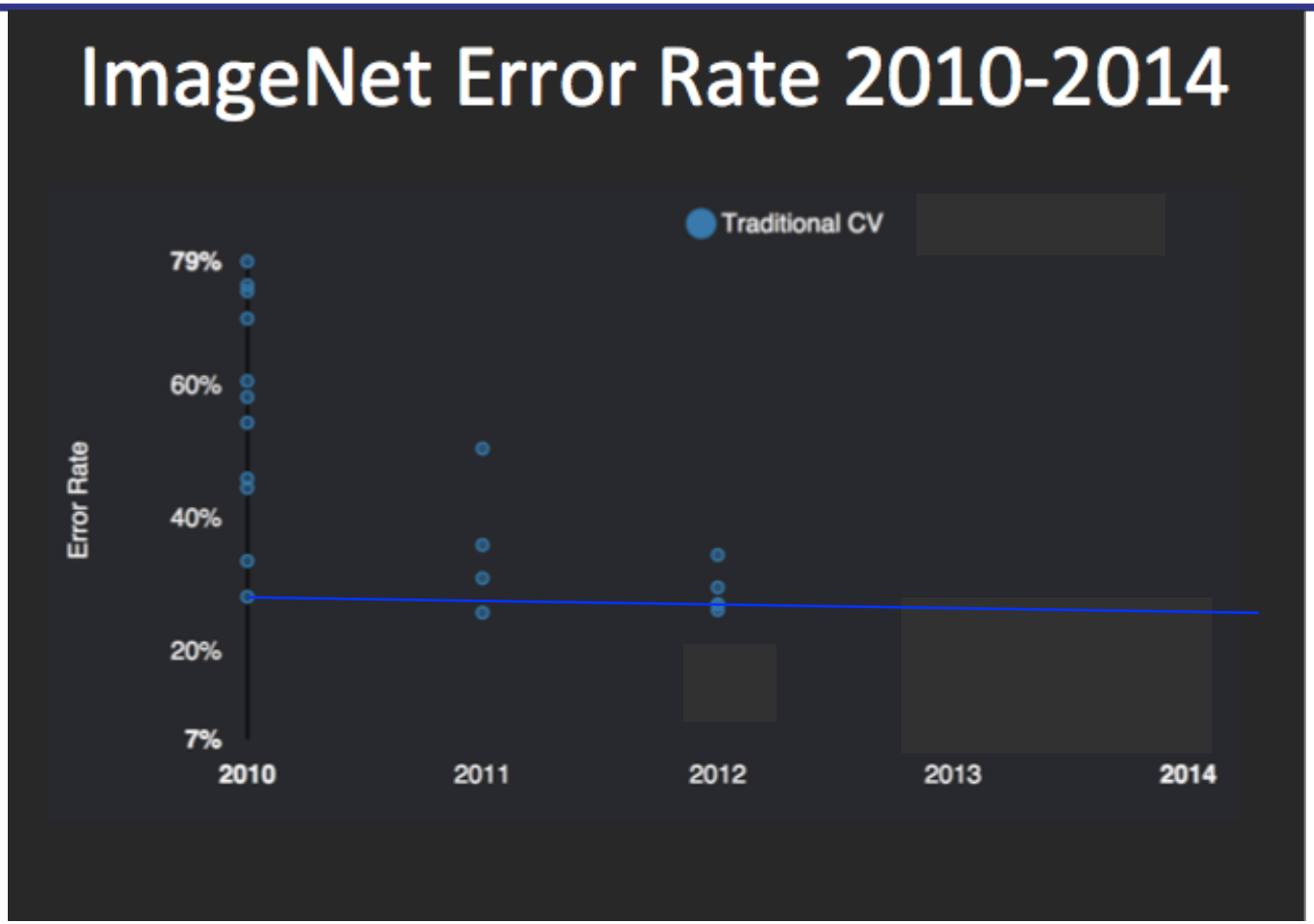
  - 60 million learned parameters (since then, billions of parameters)
  - ~1.2 million training images

Performance



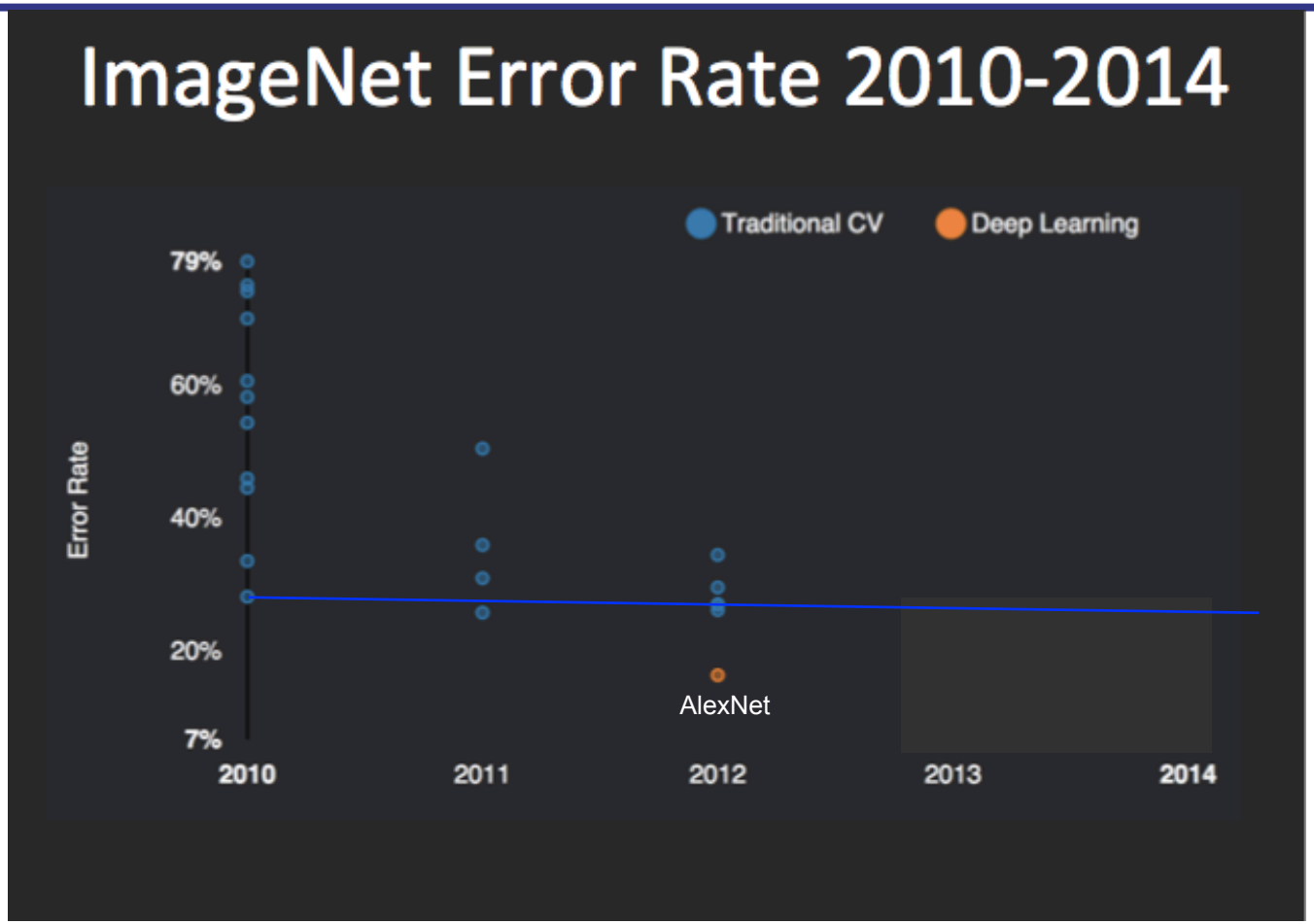
graph credit Matt Zeiler, Clarifai

Performance



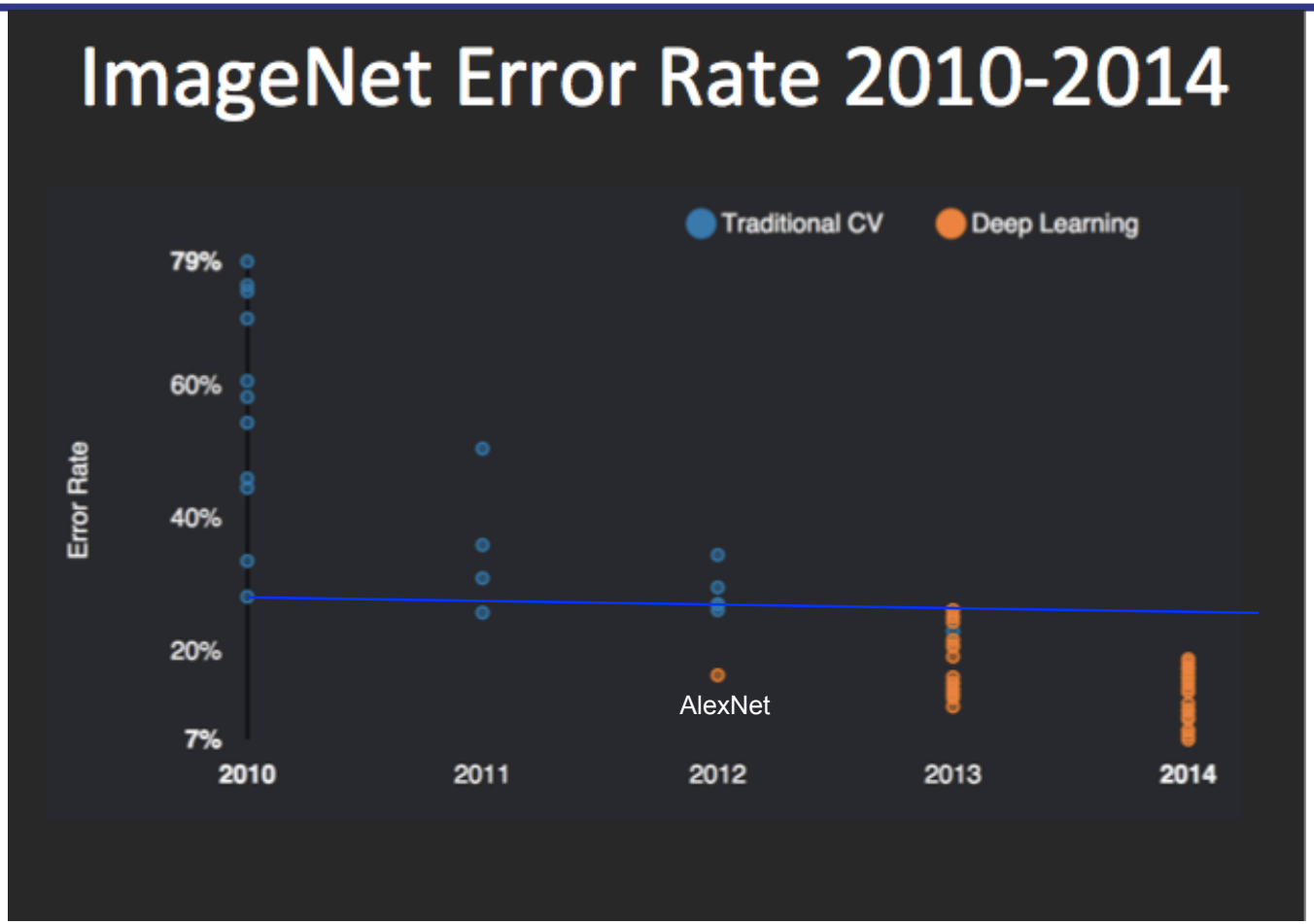
graph credit Matt Zeiler, Clarifai

Performance



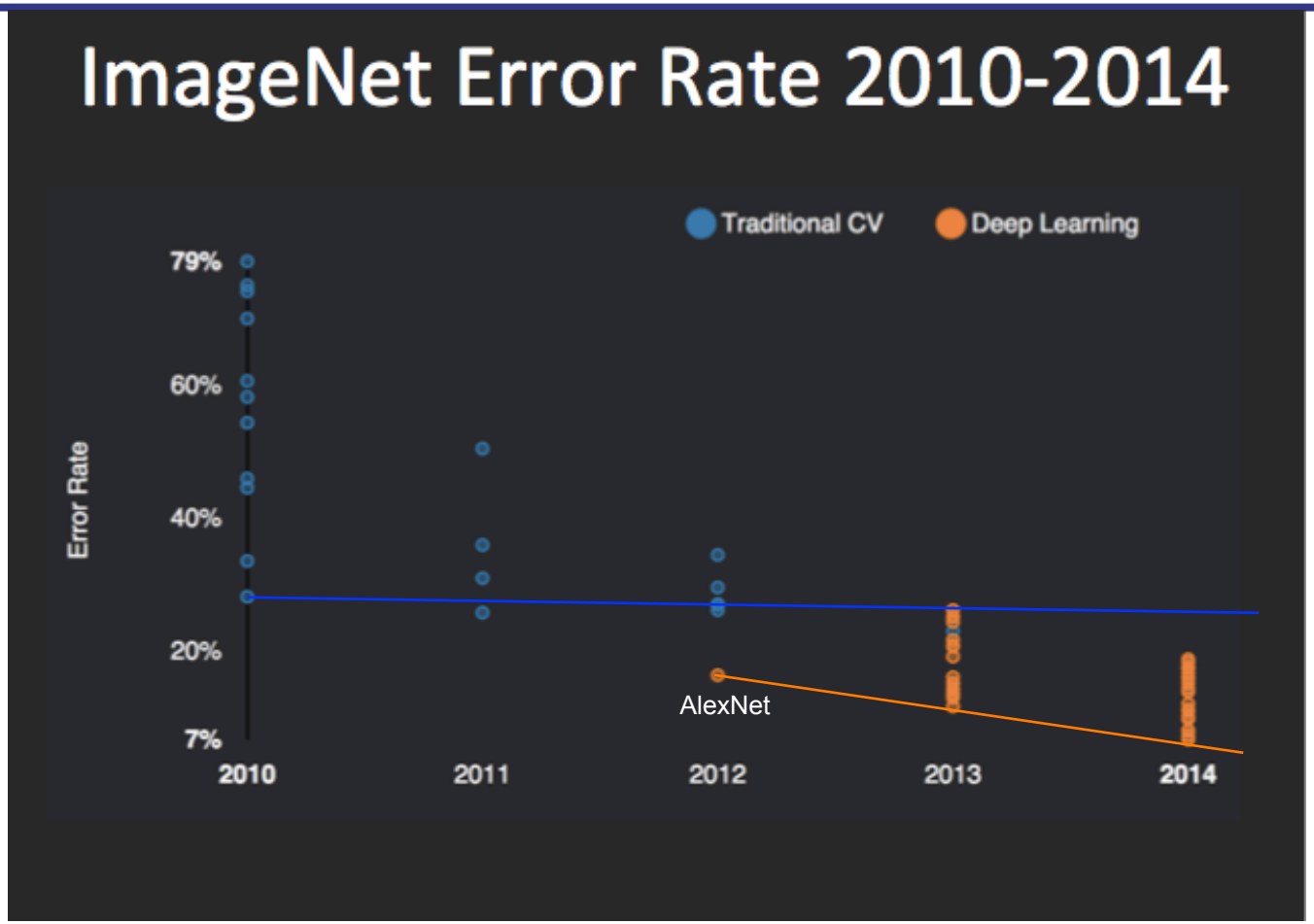
graph credit Matt Zeiler, Clarifai

Performance



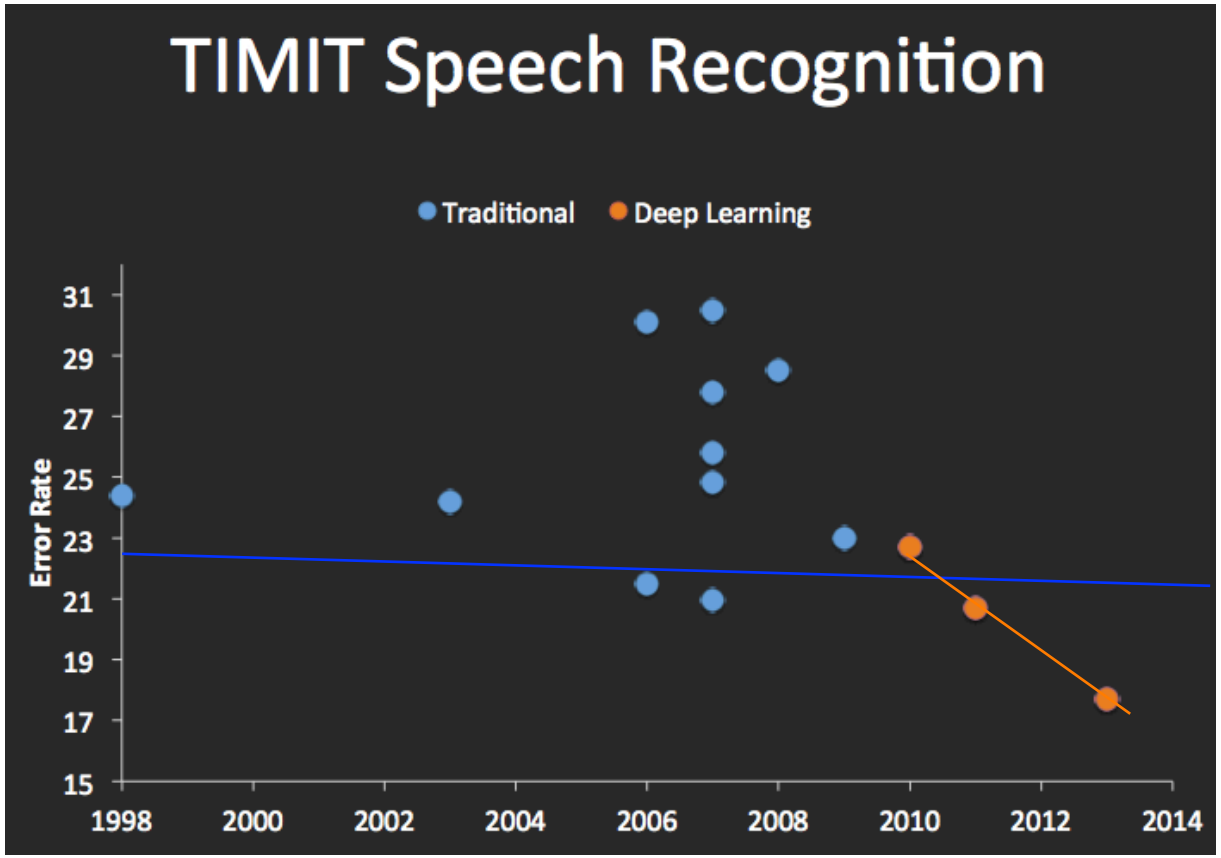
graph credit Matt Zeiler, Clarifai

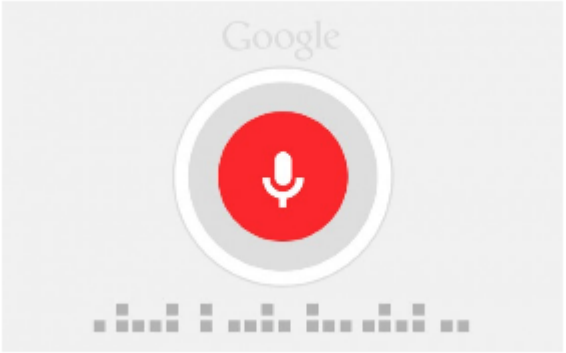
Performance



graph credit Matt Zeiler, Clarifai

Speech Recognition

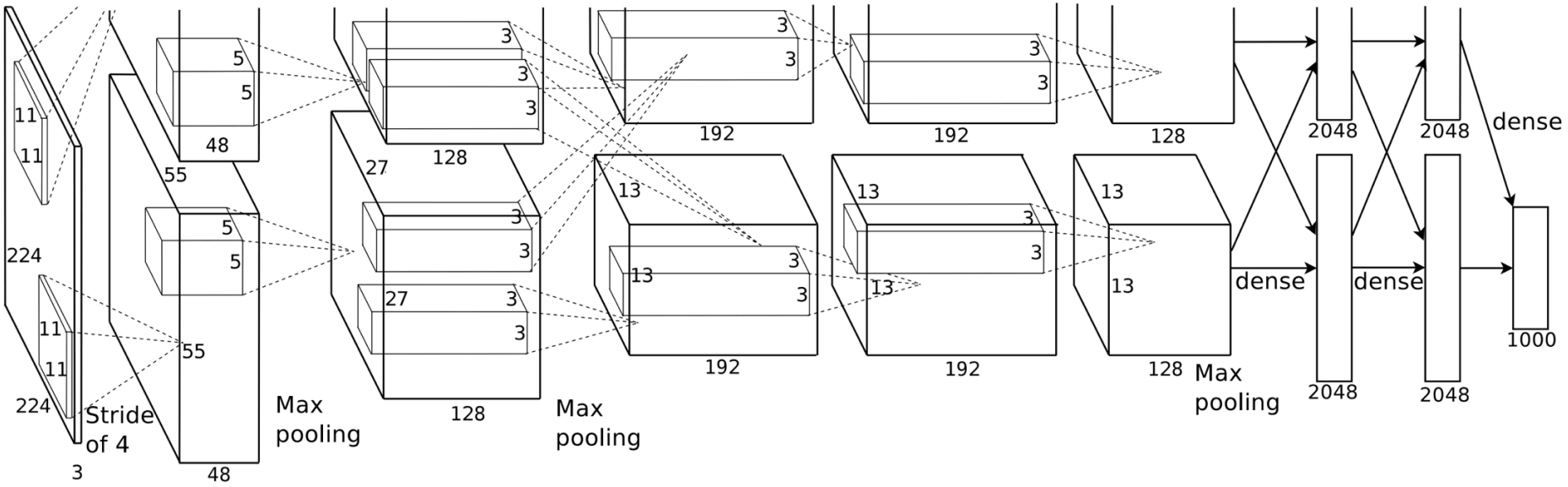




graph credit Matt Zeiler, Clarifai

What's Under the Hood

[Figure from Krizhevsky, Sutskever, Hinton 2012]



$$\min_{\theta} \sum_{i=1}^{m} error(x^{(i)}; \theta)$$

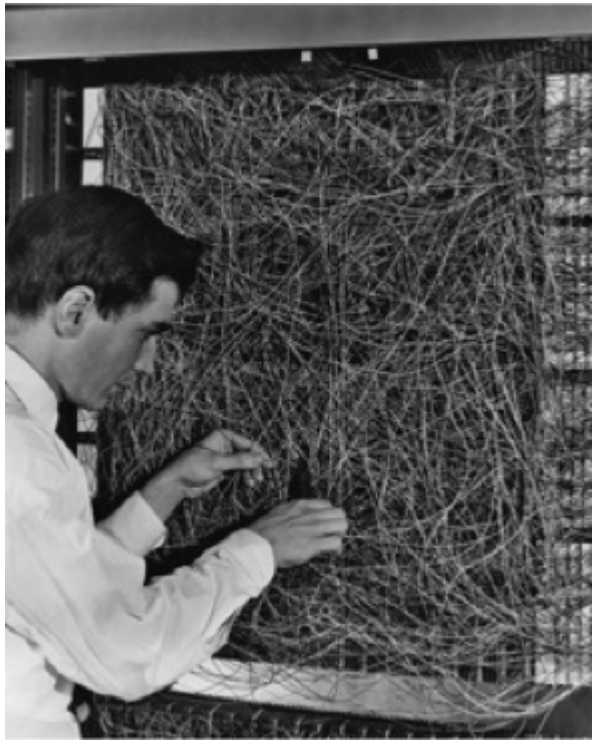
History

1958 Rosenblatt proposed perceptrons
1980 Neocognitron (Fukushima, 1980)
1982 Hopfield network, SOM (Kohonen, 1982), Neural PCA (Oja, 1982)
1985 Boltzmann machines (Ackley et al., 1985)
1986 Multilayer perceptrons and backpropagation (Rumelhart et al., 1986)
1988 RBF networks (Broomhead&Lowe, 1988)
1989 Autoencoders (Baldi&Hornik, 1989), Convolutional network (LeCun, 1989)
1992 Sigmoid belief network (Neal, 1992)
1993 Sparse coding (Field, 1993) (Olshausen, 1996)

2000s Sparse, Probabilistic, and Energy models (Hinton, Bengio, LeCun, Ng)

Is deep learning 3, 30, or 60 years old?

based on history by K. Cho



Rosenblatt's Perceptron

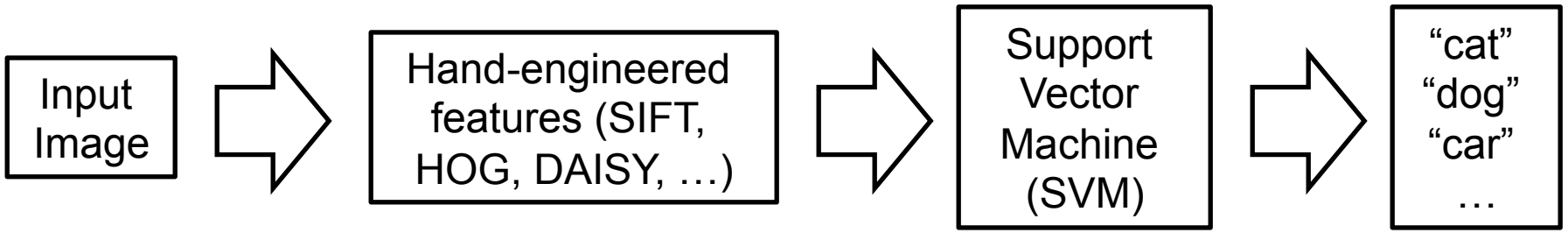
What's Changed

- Data
  - 1.2M training examples
  - * 2048 (shifts)
  - * 90 (PCA re-coloring)
  - 1.2M * 2k *90 ~ 0.216 trillion
  - Human eye: 1k frames/s

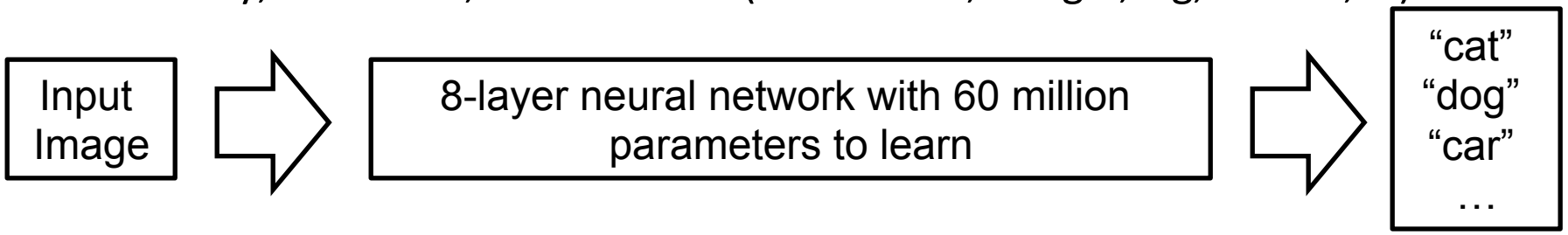
    → ~6.84yrs of experience
- Compute power
  - Two NVIDIA GTX 580 GPUs
  - 5-6 days of training time
- Nonlinearity
  - Sigmoid → ReLU
- Regularization
  - (Training data augmentation)
  - Drop-out
- Exploration of model structure
- Optimization know-how

Object Detection in Computer Vision

- State-of-the-art object detection until 2012:



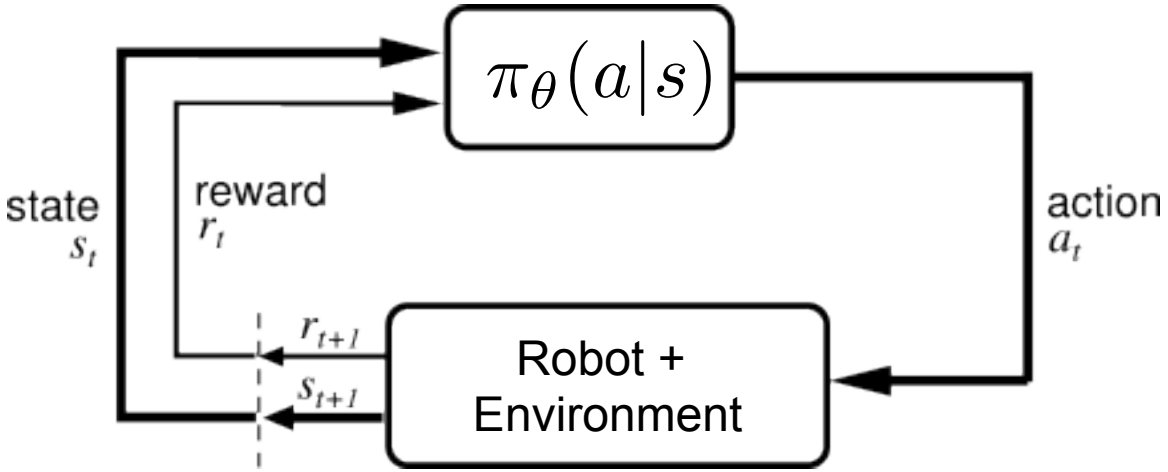
- Krizhevsky, Sutskever, Hinton 2012 (also: Lecun, Bengio, Ng, Darrell, …):



  - 60 million learned parameters (since then, billions of parameters)
  - ~1.2 million training images

Reinforcement Learning

probability of taking action a in state s



$$\max_\theta \quad \mathrm{E}[\sum_{t=0}^{H} R(s_t)|\pi_\theta]$$

- Robotics
- Marketing / Advertising
- Dialogue
- Optimizing operations / logistics
- Queue management
- …

Examples of RL in Robotics

[Kohl and Stone, ICRA 2004]

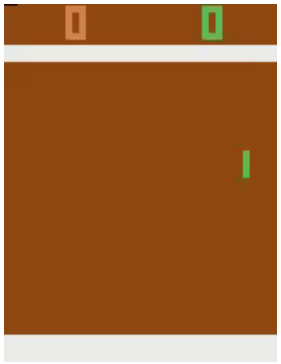
[Tedrake, Zhang, Seung 2005]

[Ng + al, ISER 2004]

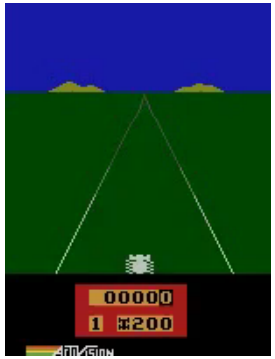
…

Number of parameters learned: tens

How About Deep RL?

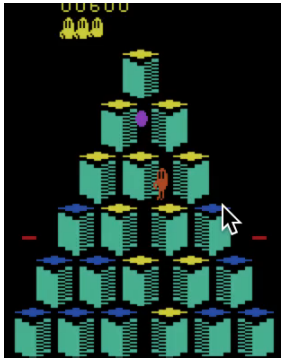


Pong

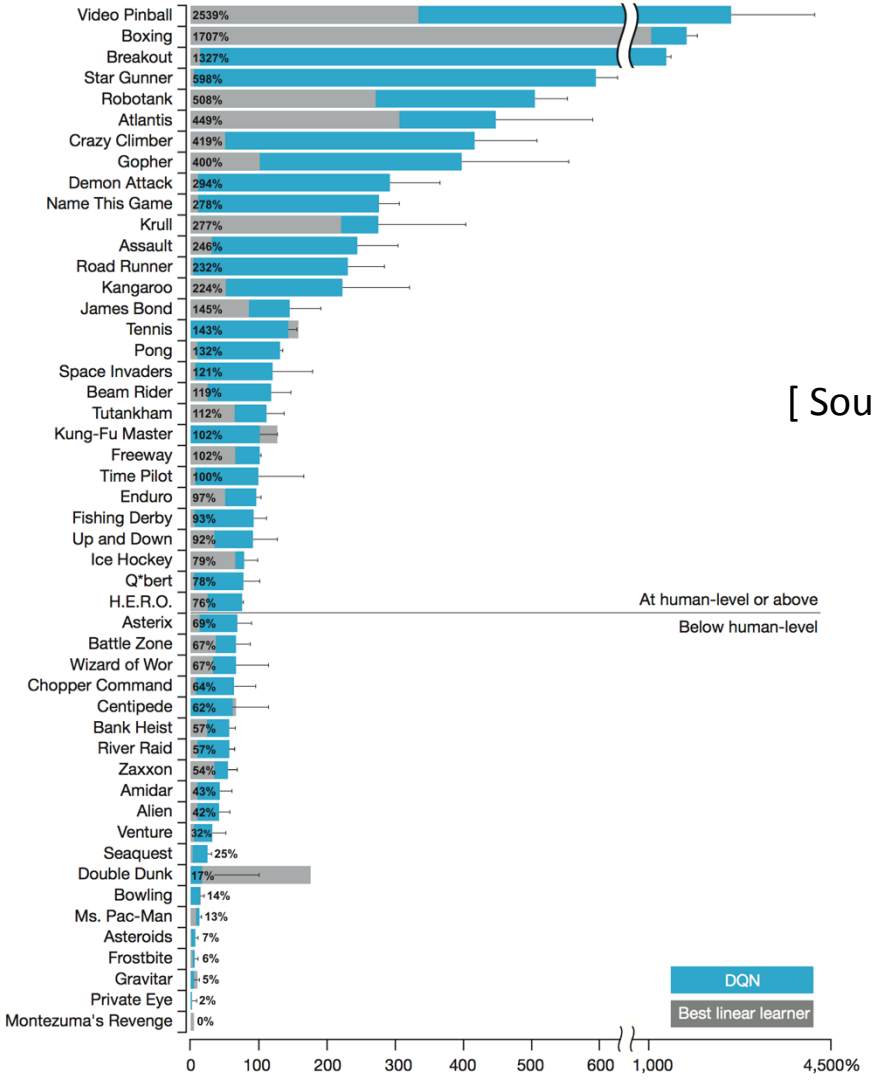


Enduro

Beamrider

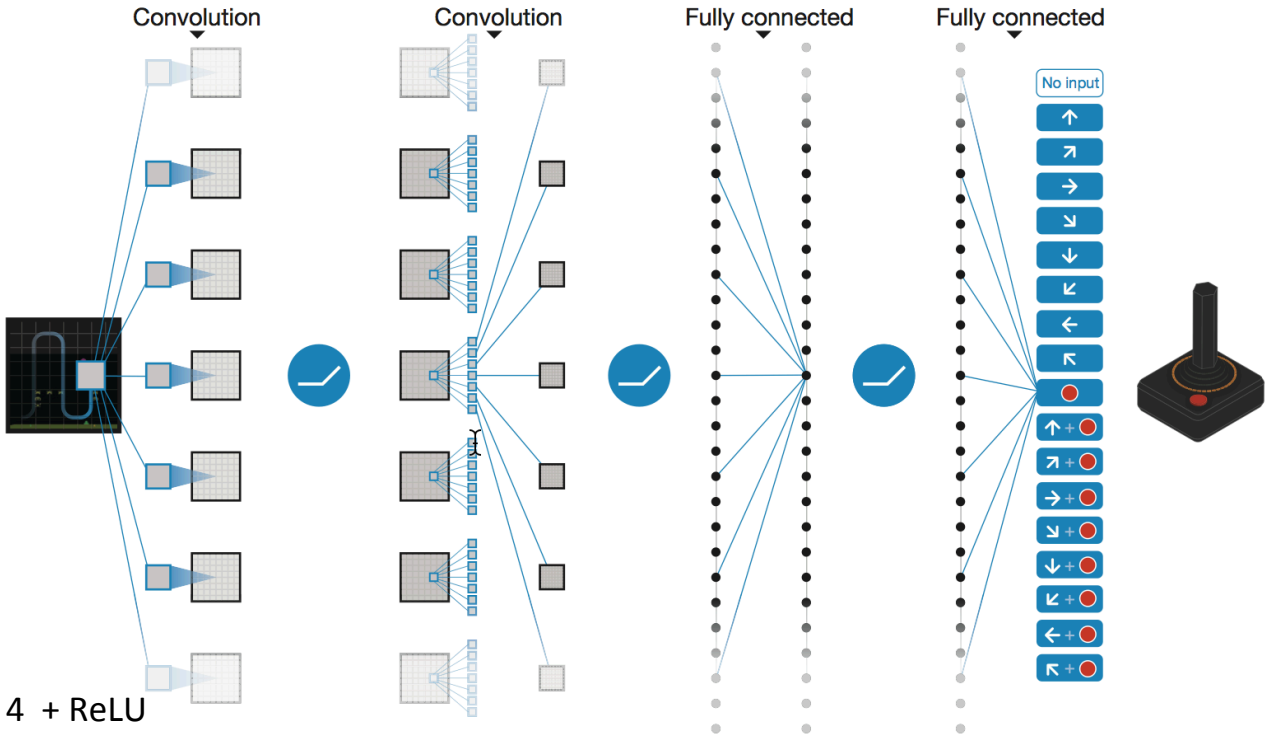


Q*bert



[Source: Mnih et al., Nature 2015 (DeepMind)]

From Pixels to Joystick Commands



32 8x8 filters with stride 4 + ReLU
64 4x4 filters with stride 2 + ReLU
64 3x3 filters with stride 1 + ReLU
fully connected 512 units + ReLU
fully connected output units, one per action

[Source: Mnih et al., Nature 2015 (DeepMind)]

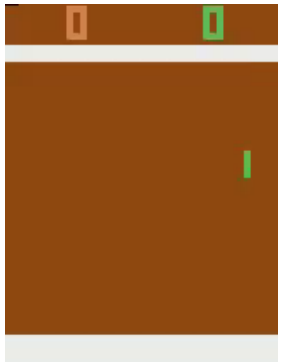
Approach

- Q-learning with deep network as function approximator
- mini-batches of size 32
- e-greedy annealed linearly from 1 to 0.1 over the first million frames, and fixed at 0.1 thereafter
- trained for a total of 50 million frames (=38 days of game experience) and use a replay memory of one million most recent frames

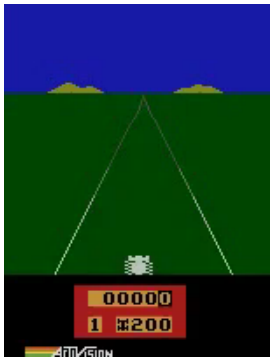
What's new?

Deep Reinforcement Learning for Atari Games

- Deep Q-learning [Mnih et al, 2013]
- Monte Carlo Tree Search [Xiao-Xiao et al, 2014]
- Trust region policy iteration [Schulman, Levine, Moritz, Jordan, A., 2014]

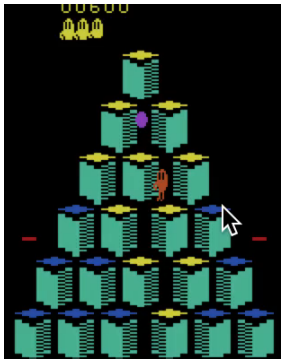


Pong



Enduro

Beamrider

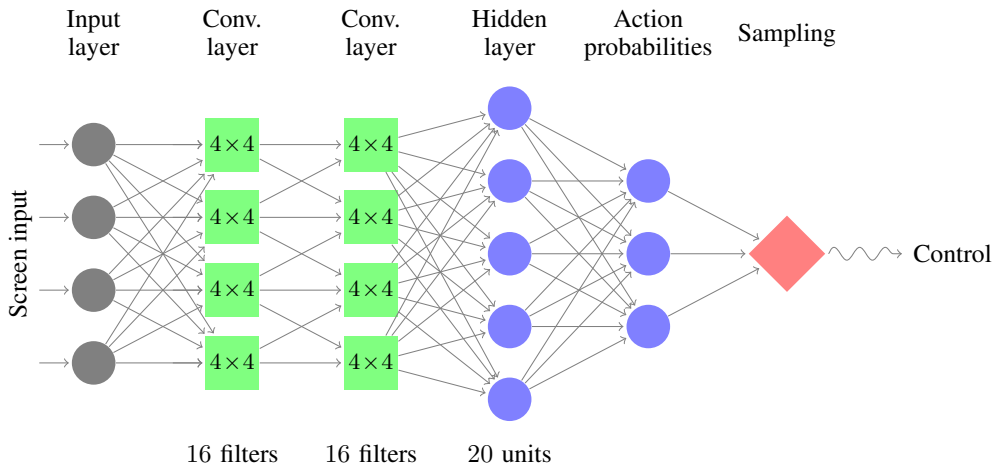


Q*bert

Policy Structure for Atari Games

Input: 110x84x3 images
Output: game controls

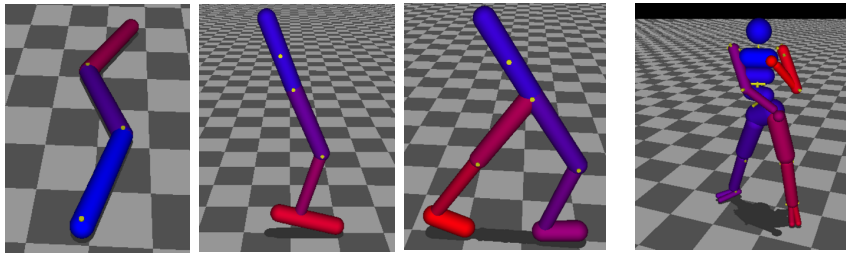
Neural network architecture:



35,000 learned parameters for each game

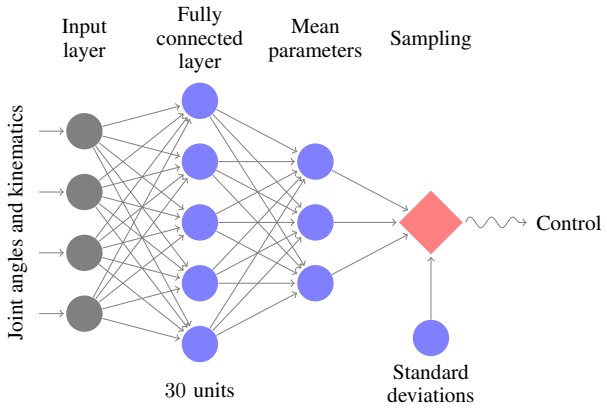
Locomotion

Robot models in physics simulator (MuJoCo, from Emo Todorov)

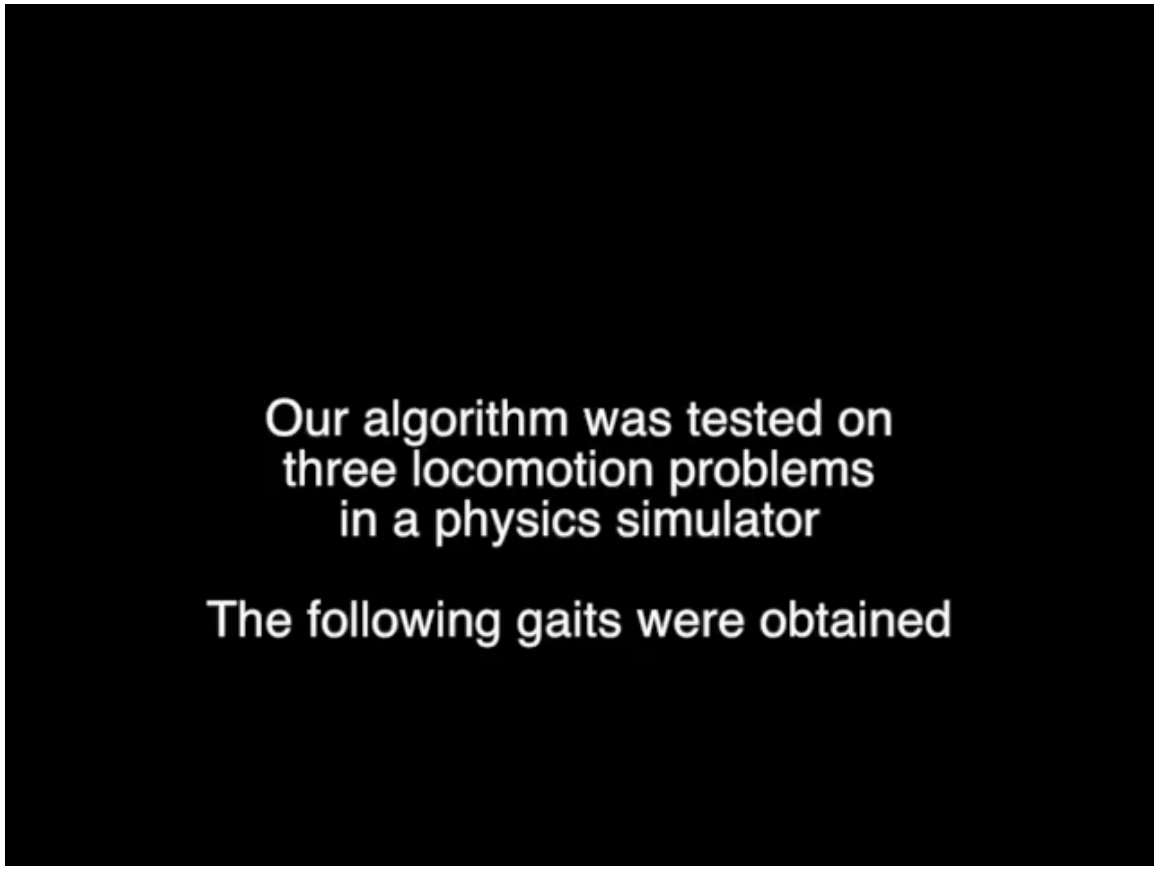


Input: joint angles and velocities
Output: joint torques

Neural network architecture:

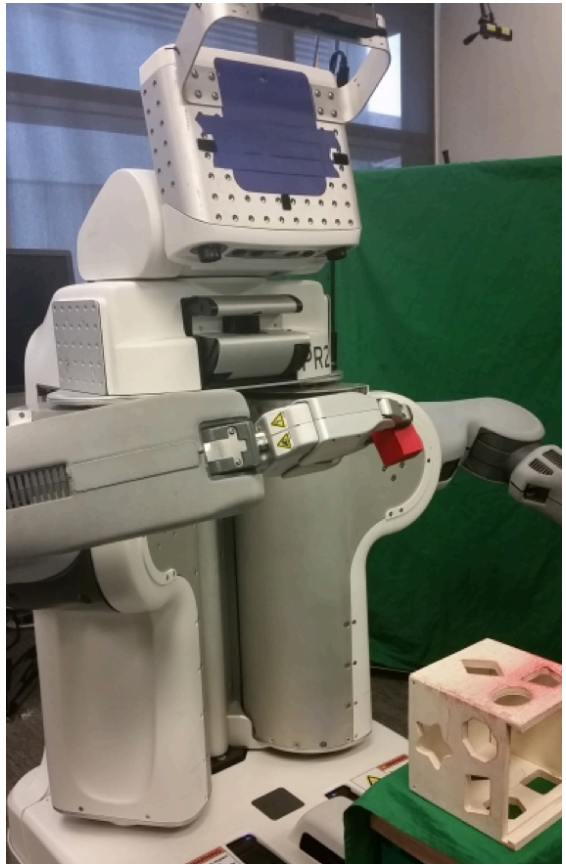


Experiments in Locomotion

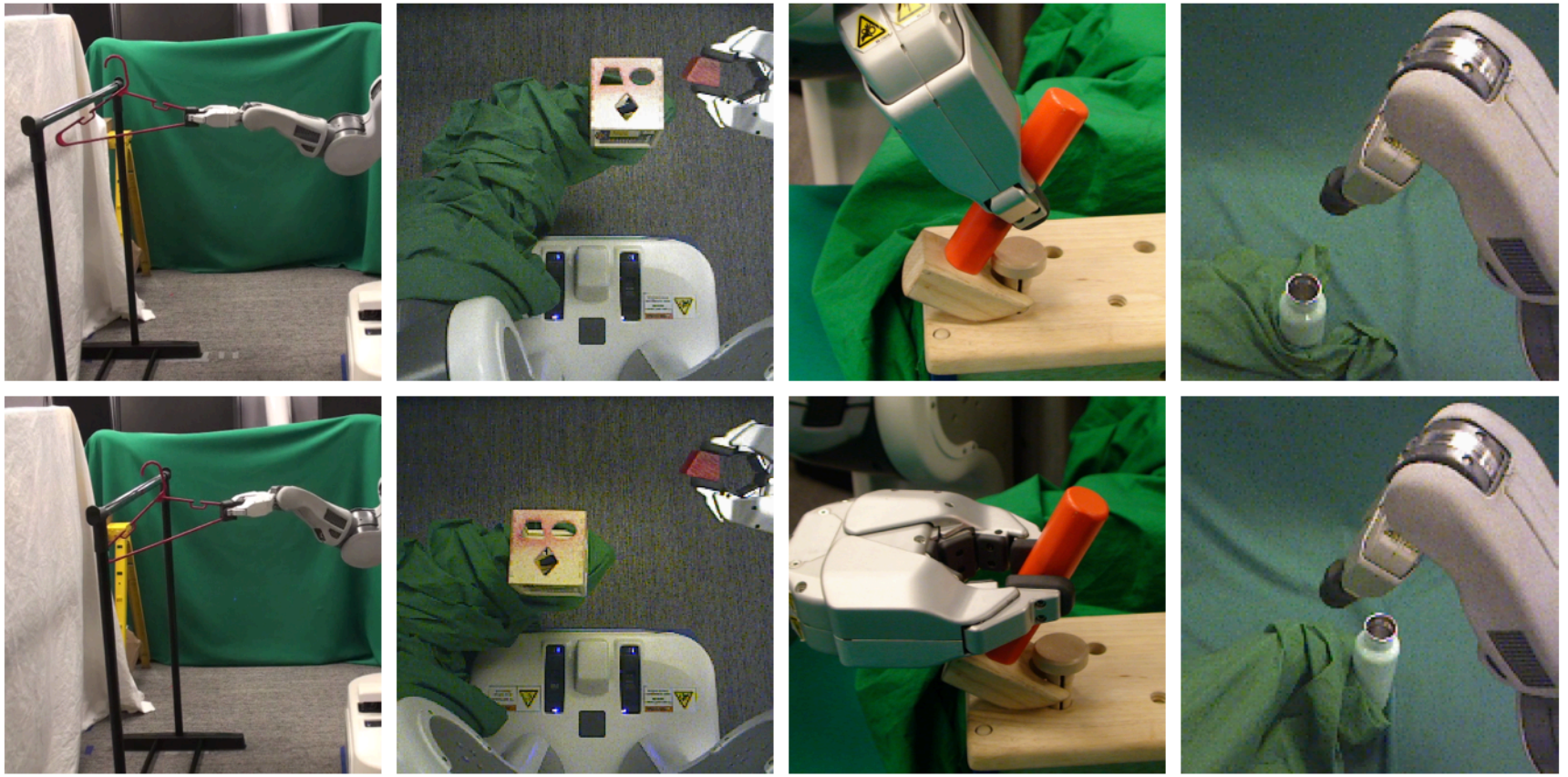


[Schulman, Levine, Moritz, Jordan, A., 2014]

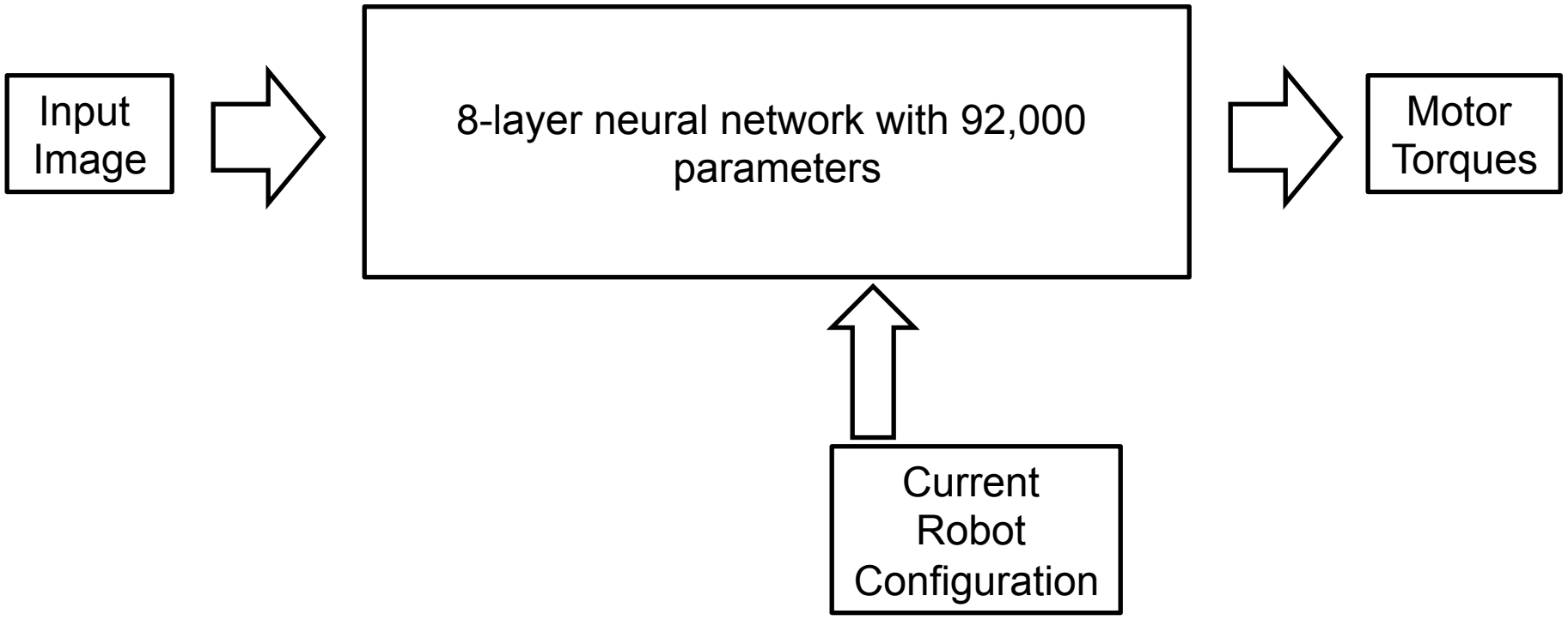
How About Real Robotic Visuo-Motor Skills?



Example Tasks

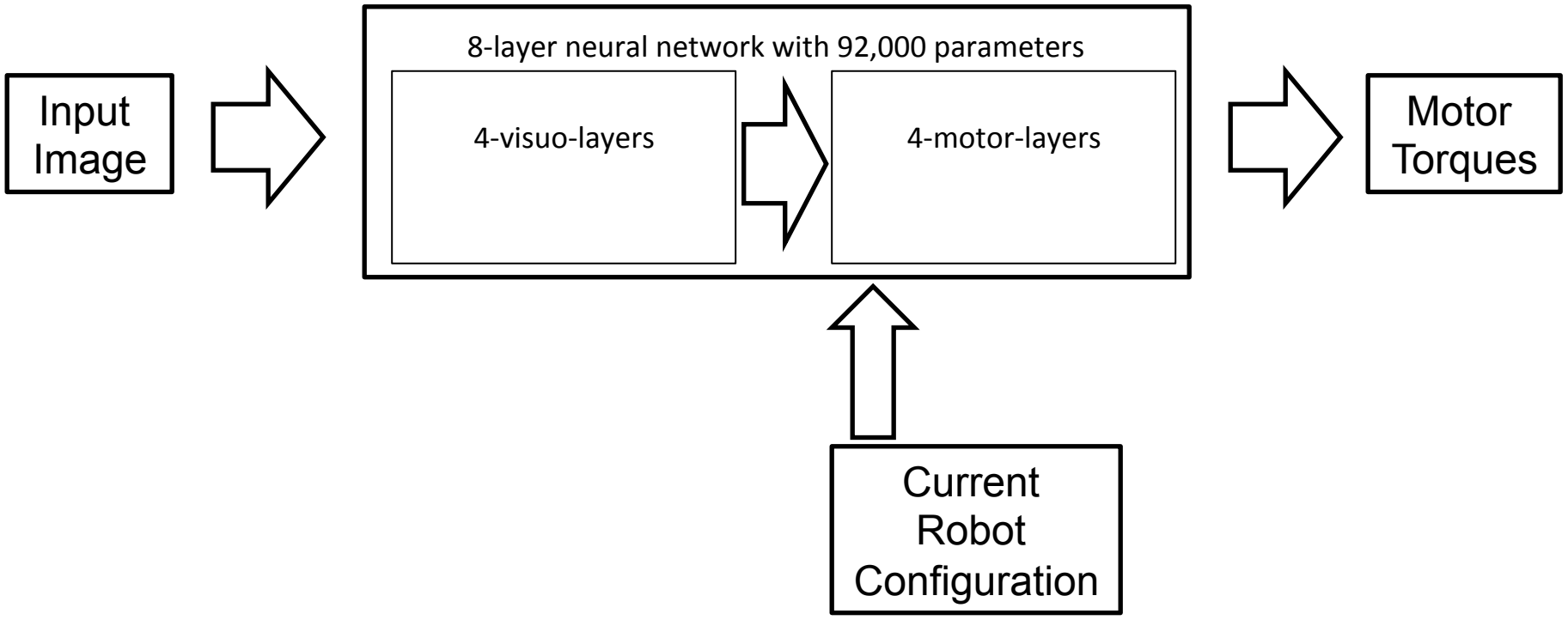


Architecture



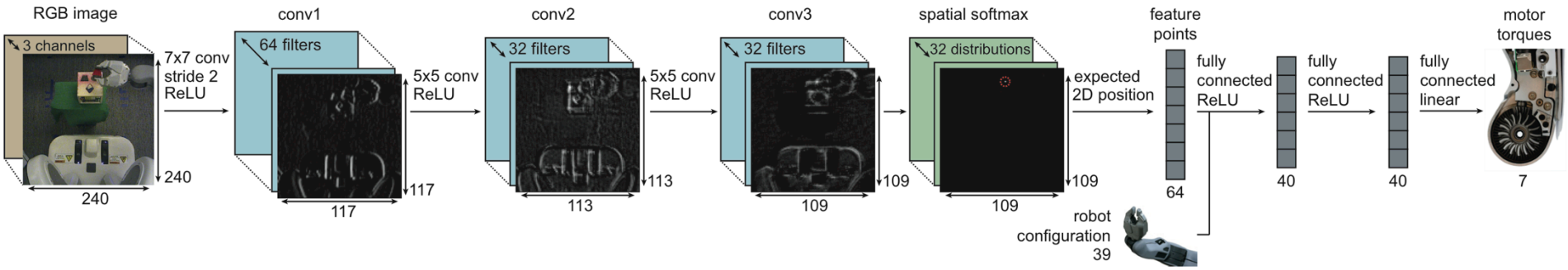
[Levine*, Finn*, Darrell, A., 2015
TR at: tinyurl.com/visuomotor]

Architecture



[Levine*, Finn*, Darrell, A., 2015
TR at: tinyurl.com/visuomotor]

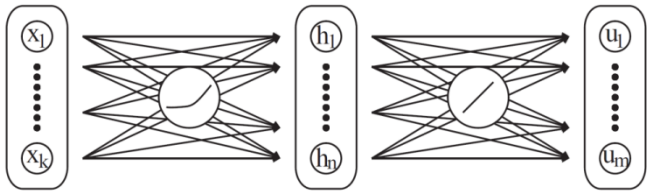
Architecture (92,000 parameters)

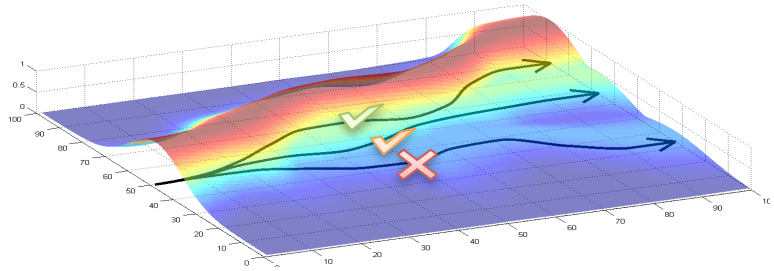


[Levine*, Finn*, Darrell, A., 2015
TR at: tinyurl.com/visuomotor]

Training: Guided Policy Search

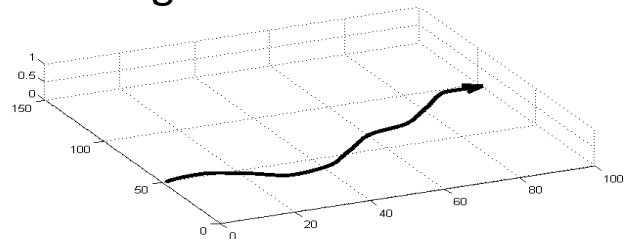
general-purpose neural network controller



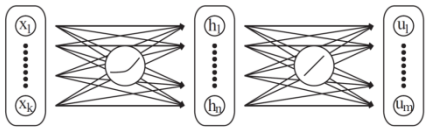


policy search (RL)	complex dynamics	complex policy	HARD
supervised learning	~~complex dynamics~~	complex policy	EASY
single instance solutions	complex dynamics	~~complex policy~~	EASY

Single instance solutions

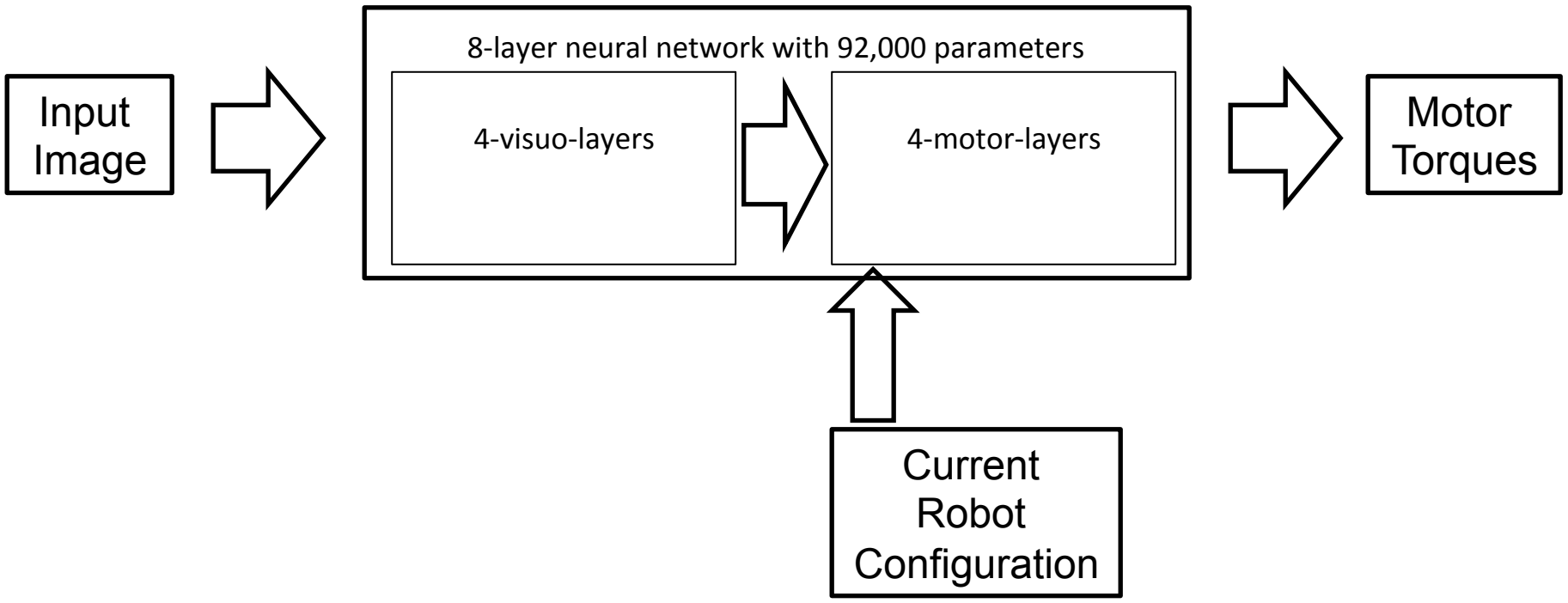


Supervised learning

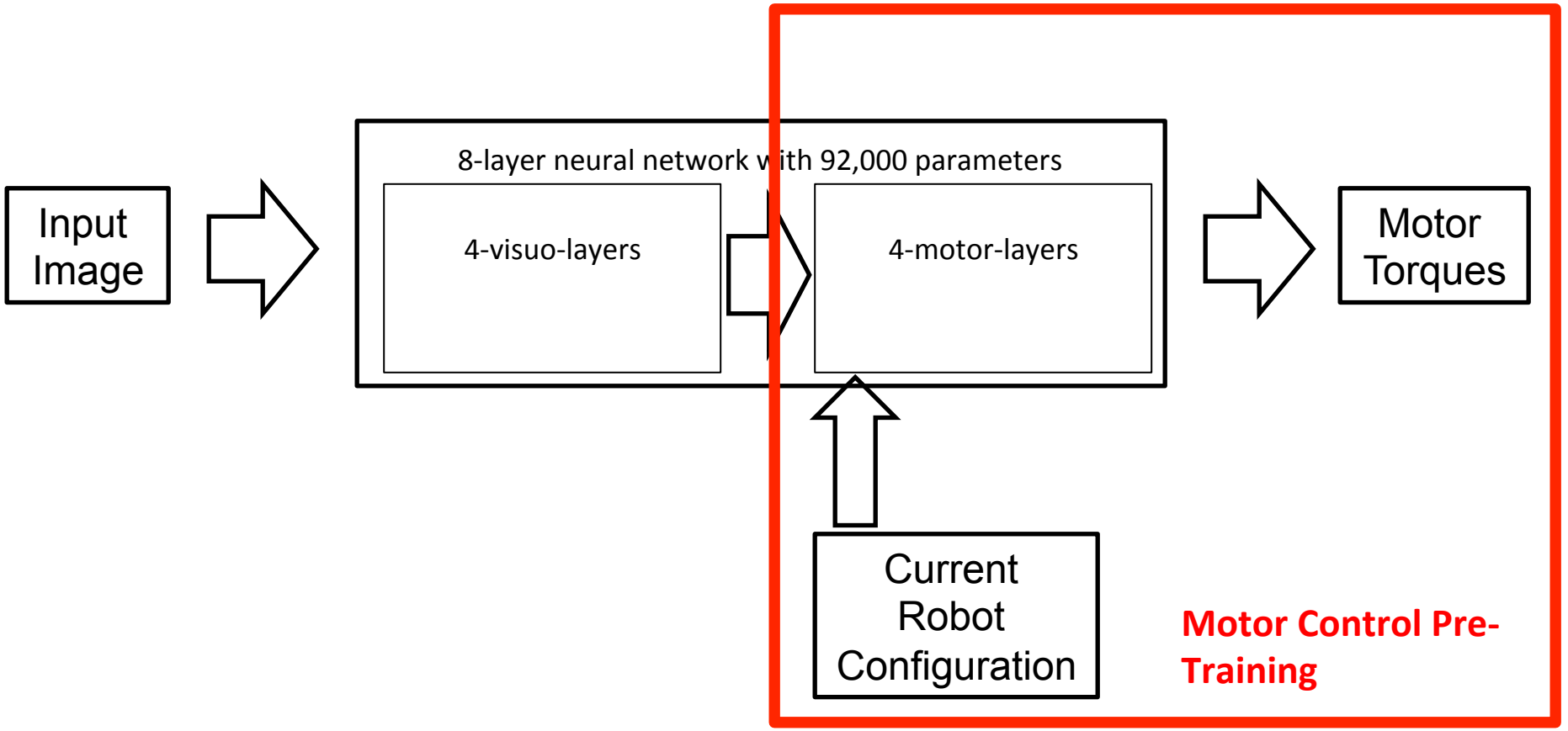


[Levine & Koltun ICML '14; Levine & Abbeel NIPS '14]

Training with Guided Policy Search



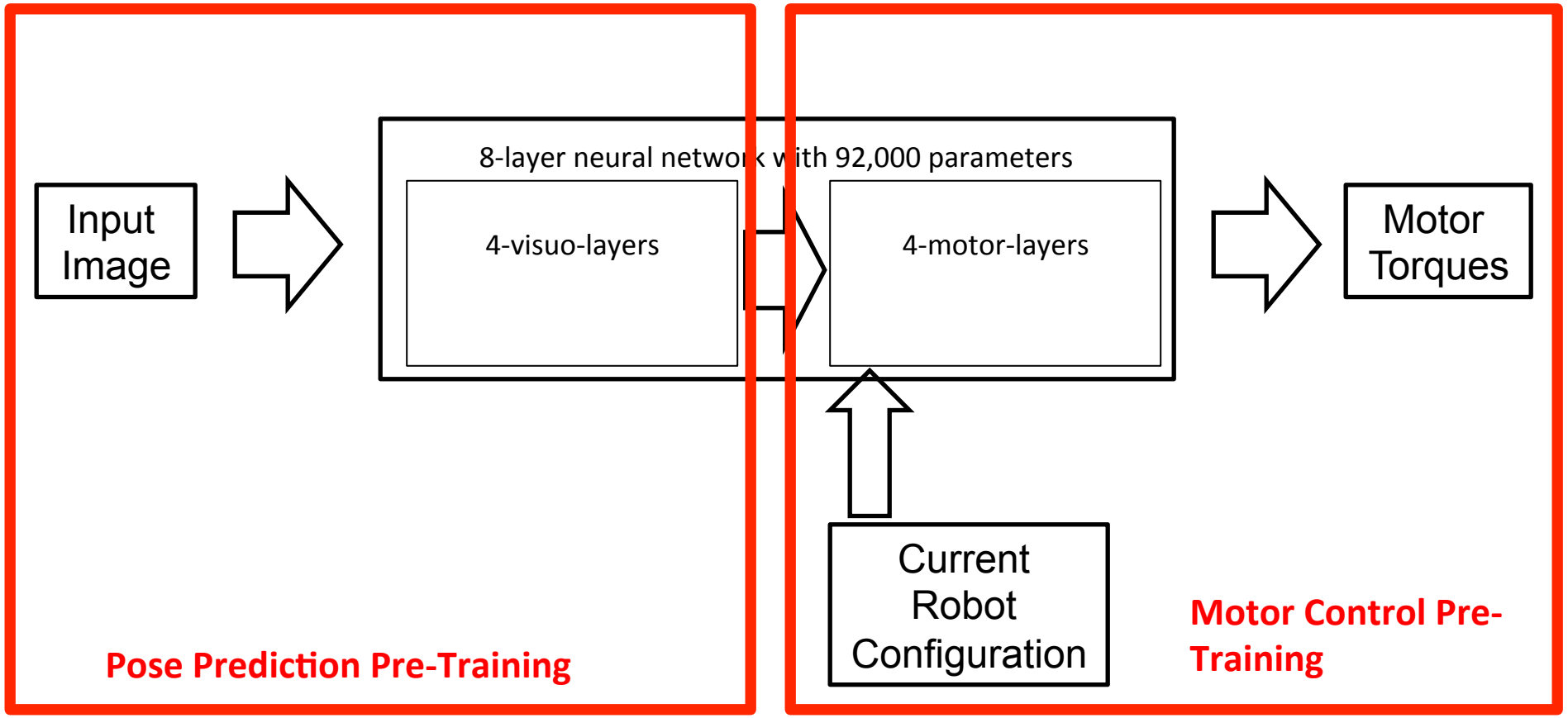
Initialization from Pre-Training



Block Stacking – Learning the Controller for a Single Instance

Block Stacking – Learned Neural Net Policy

Initialization from Pre-Training



Learned Skills

Locomotion through Guided Policy Search

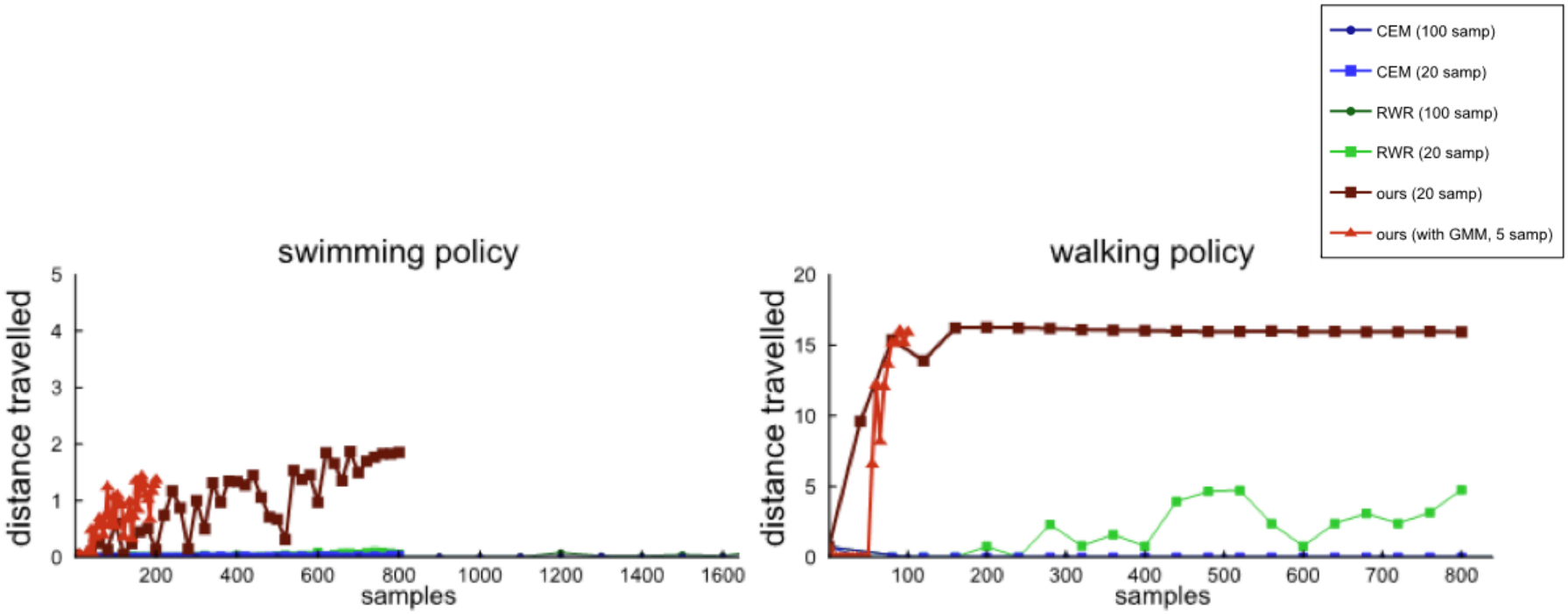
Swimming

learned policy

[neural network]

[Levine and A., NIPS 2014]

Comparison



[Levine and A., NIPS 2014]

Frontiers / Limitations

- Architectures for shared learning / transfer learning
  - Multiple robots and sensors (including simulation)
  - Multiple tasks
  - Simulation – Real world
- Leverage simultaneously learning: dynamics, Q, policy
- Exploration beyond e-greedy
- Controllers that require memory / estimation
- Hierarchical policies

Applications: Manipulation, Locomotion, Vision-based Flight, …