



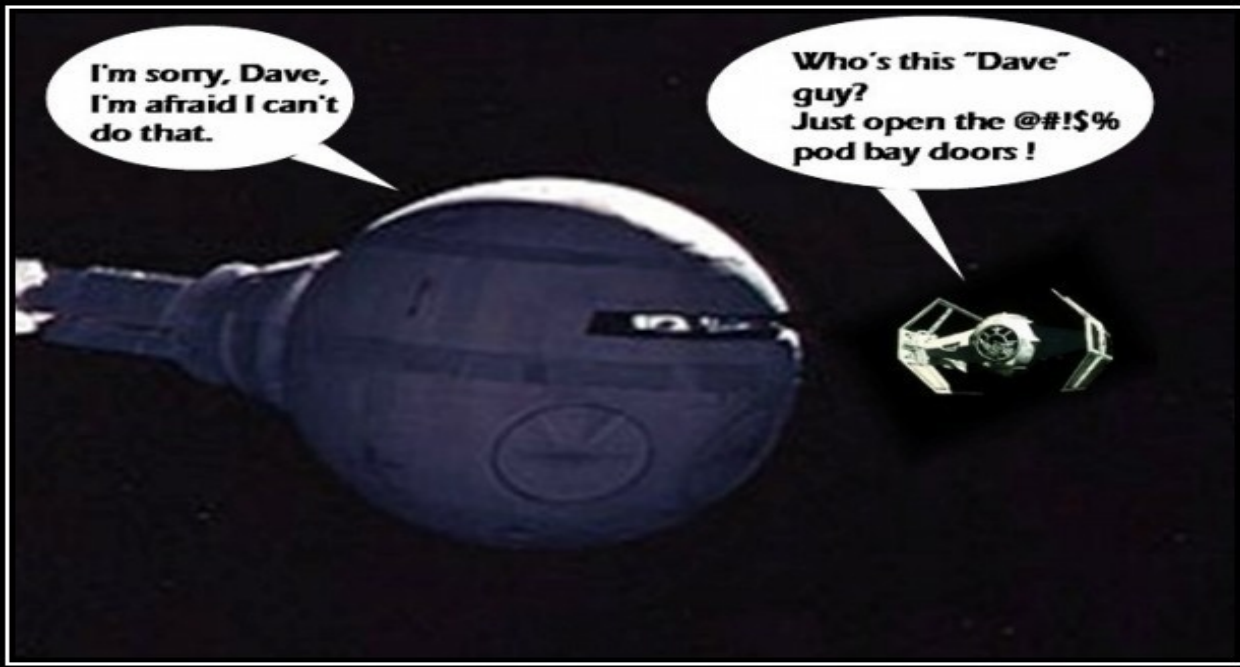
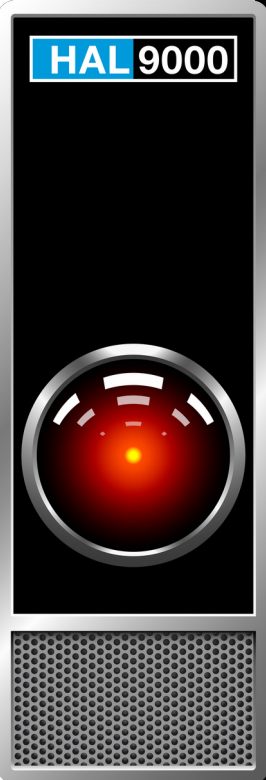
# A Path to AI

Yann LeCun

Facebook AI Research & NYU



# How can machines be intelligent and beneficial



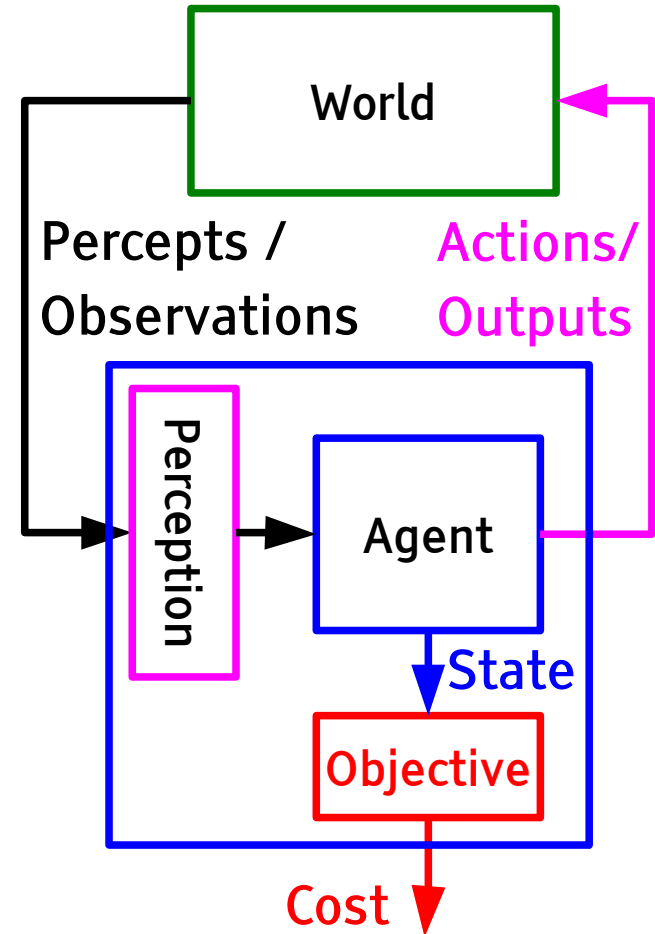
**HAL 9000**  
Worst garage door opener. Ever.





# The Architecture of an AI system

- ▶ **Perception:** estimates the state of the world
  - ▶ Vision, Speech, Audio
- ▶ **Agent**
  - ▶ Prediction, Causal Inference
  - ▶ Planning, Reasoning, Working Memory
- ▶ **Objective:** Measures the agent's "happiness"
  - ▶ Agent acts to optimize the objective
  - ▶ Drives the system to do what we want
  - ▶ Parts of it are hardwired and immutable
  - ▶ Parts of it is trainable



# Obstacles to AI

How could machines acquire common sense?

- TL;DR:
  - Machines don't have common sense
  - To acquire it, they must learn how the world works by observation
  - Predictive/unsupervised learning is the missing link



# Today, AI = Supervised (Deep) Learning

NEW YORK UNIVERSITY

LAGR

netSCALE  
Technologies, Inc

100-dimensional  
Feature Vector

100@25x121



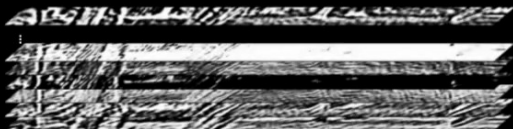
CONVOLUTIONS (6x6)

20@30x125



MAX SUBSAMPLING (1x4)

20@30x484

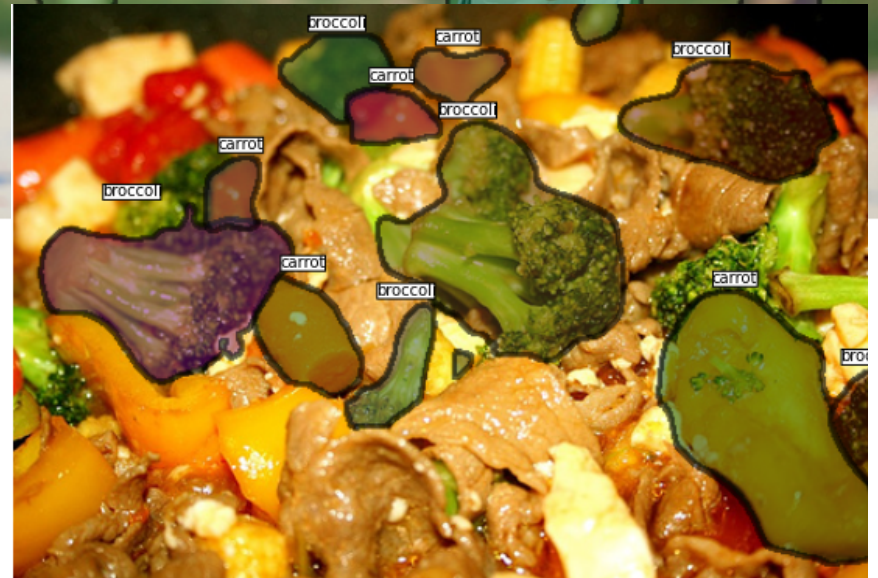


CONVOLUTIONS (7x6)FEAT

3@36x484



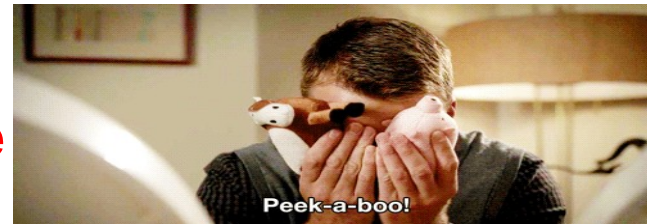
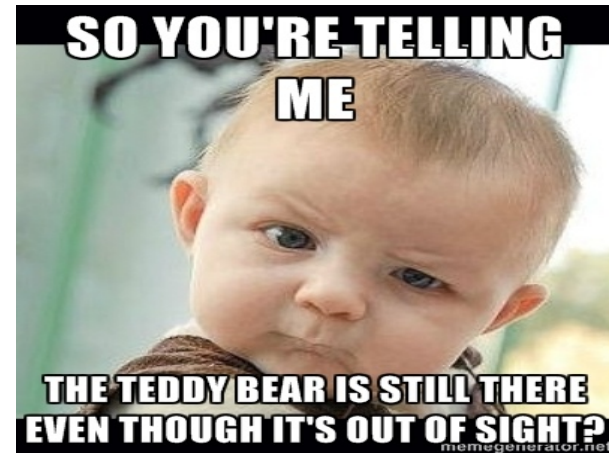
YUV Input





# But Supervised Learning is Insufficient for “Real” AI

- ▶ Most of animals and humans learning is unsupervised, through interaction with the world
- ▶ We learn how the world works by observing it
  - ▶ We learn many simple things: depth and 3-dimensionality, gravity, **object permanence**, ...
- ▶ **We build models of the world through predictive unsupervised learning**
- ▶ World models give us “common sense”





# What is Common Sense?

- ▶ **“The trophy doesn’t fit in the suitcase because it’s too large/small”**
  - ▶ (Winograd Schema)
  
- ▶ **“Mike picked up his bag and left the room”**
  
- ▶ We have common sense because we know how the world works
  
- ▶ **How do we get machines to learn that?**



# Common Sense is the ability to fill in the blanks

- ▶ Infer the state of the world from partial information
- ▶ Infer the future from the past and present
- ▶ Infer past events from the present state
- ▶ Filling in the visual field at the retinal blind spot
- ▶ Filling in occluded images
- ▶ Filling in missing segments in text, missing words in speech.
- ▶ Predicting the consequences of our actions
- ▶ Predicting the sequence of actions leading to a result
- ▶ **Predicting any part of the past, present or future percepts from available information.**
- ▶ That's what **predictive unsupervised learning** is

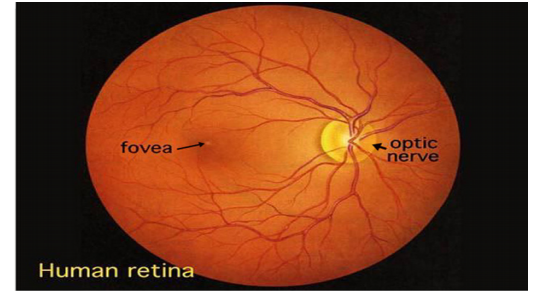


Fig. 1. Human retina as seen through an ophthalmoscope.







# The Necessity of Unsupervised / Predictive Learning

- ▶ **The number of samples required to train a large learning machine (for any task) depends on the amount of information that we ask it to predict.**
  - ▶ The more you ask of the machine, the larger it can be.
- ▶ “The brain has about  $10^{14}$  synapses and we only live for about  $10^9$  seconds. So we have a lot more parameters than data. This motivates the idea that we must do a lot of unsupervised learning since the perceptual input (including proprioception) is the only place we can get  $10^5$  dimensions of constraint per second.”
  - ▶ Geoffrey Hinton (in his 2014 AMA on Reddit)
  - ▶ (but he has been saying that since the late 1970s)
- ▶ Predicting human-provided labels is not enough
- ▶ Predicting a value function is not enough



# Three Types of Learning

## ▶ Reinforcement Learning

▶ The machine predicts a scalar reward given once in a while

▶ **A few bits trial**

## ▶ Supervised Learning

▶ The machine predicts a category or a few numbers for each input

▶ **10→10,000 bits per trial**

## ▶ Unsupervised Learning

▶ The machine predicts any part of its input for any observed part.

▶ Predicts future frames in videos

▶ **Millions of bits per trial**

▶ **But these are unreliable bits!**



PLANE



CAR





# How Much Information does the Machine Need to Predict?

## ▶ **Pure Reinforcement Learning (cherry)**

- ▶ The machine predicts a scalar reward given once in a while.

## ▶ **A few bits for some samples**

## ▶ **Supervised Learning (icing)**

- ▶ The machine predicts a category or a few numbers for each input

## ▶ **10→10,000 bits per sample**

## ▶ **Unsupervised/Predictive Learning (génénoise)**

- ▶ The machine predicts any part of its input for any observed part.

- ▶ Predicts future frames in videos

## ▶ **Millions of bits per sample**



- ▶ **Unsupervised Learning is the Dark Matter (or Dark Energy) of AI**



# Not a jab at RL (we do RL at FAIR)

► Model-free RL works in games, but it doesn't really work in the real world



FAIR won the VizDoom 2016 competition.  
[Wu & Tian, submitted to ICLR 2017]



TorchCraft: interface between Torch and StarCraft (on github)  
[Usunier et al, submitted to ICLR 2017]



# Sutton's Dyna Architecture: "try things in your head before acting"

- ▶ Dyna: an Integrated Architecture for Learning, Planning and Reacting
- ▶ [Rich Sutton, ACM SIGART 1991]

The main idea of Dyna is the old, commonsense idea that planning is 'trying things in your head,' using an internal model of the world (Craik, 1943; Dennett, 1978; Sutton & Barto, 1981). This suggests the existence of a more primitive process for trying things *not* in your head, but through direct interaction with the world. *Reinforcement learning* is the name we use for this more primitive, direct kind of trying, and Dyna is the extension of reinforcement learning to include a learned world model.

REPEAT FOREVER:

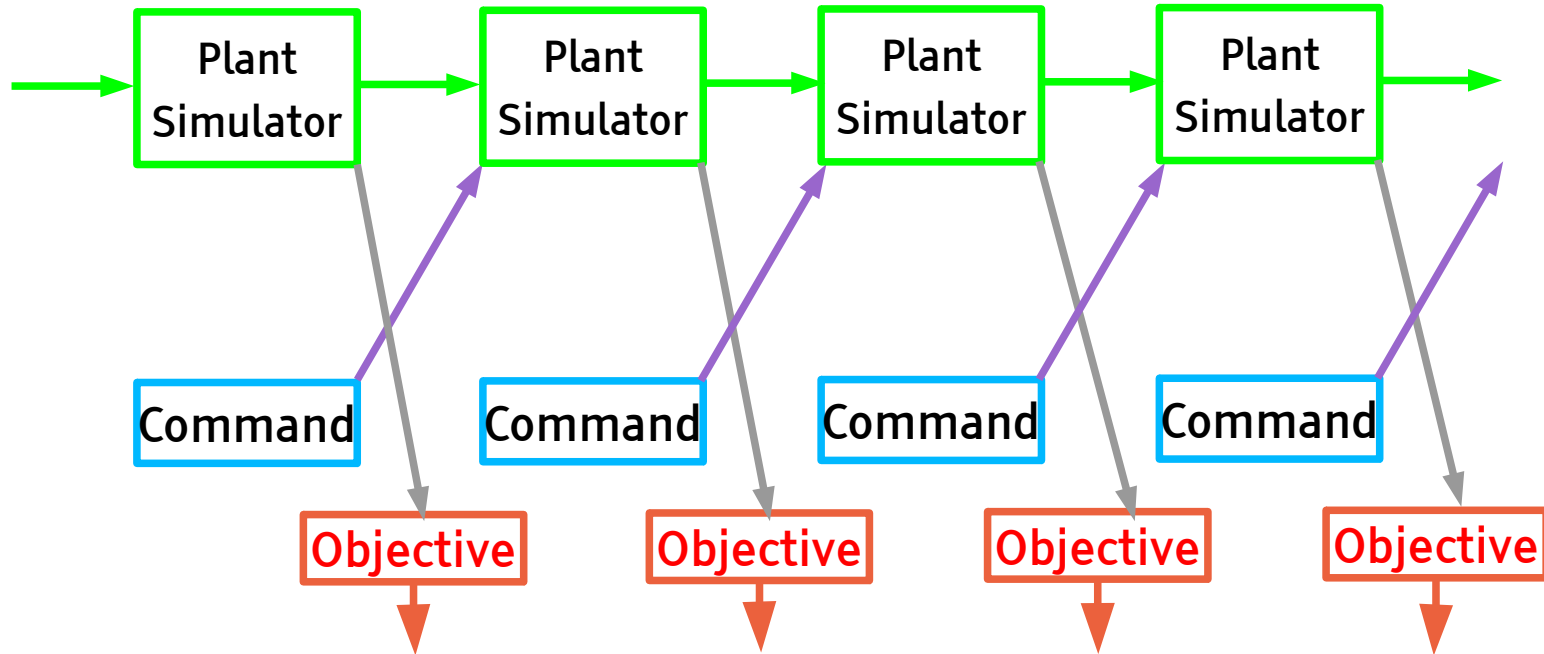
1. Observe the world's state and reactively choose an action based on it;
2. Observe resultant reward and new state;
3. Apply reinforcement learning to this experience;
4. Update action model based on this experience;
5. Repeat  $K$  times:
  - 5.1 Choose a hypothetical world state and action;
  - 5.2 Predict resultant reward and new state using action model;
  - 5.3 Apply reinforcement learning to this hypothetical experience.





# Classical model-based optimal control

- ▶ Simulate the world (the plant) with an initial control sequence
- ▶ Adjust the control sequence to optimize the objective through gradient descent
- ▶ Backprop through time was invented by control theorists in the late 1950s
  - ▶ it's sometimes called the adjoint state method
- ▶ [Athans & Falb 1966, Bryson & Ho 1969]



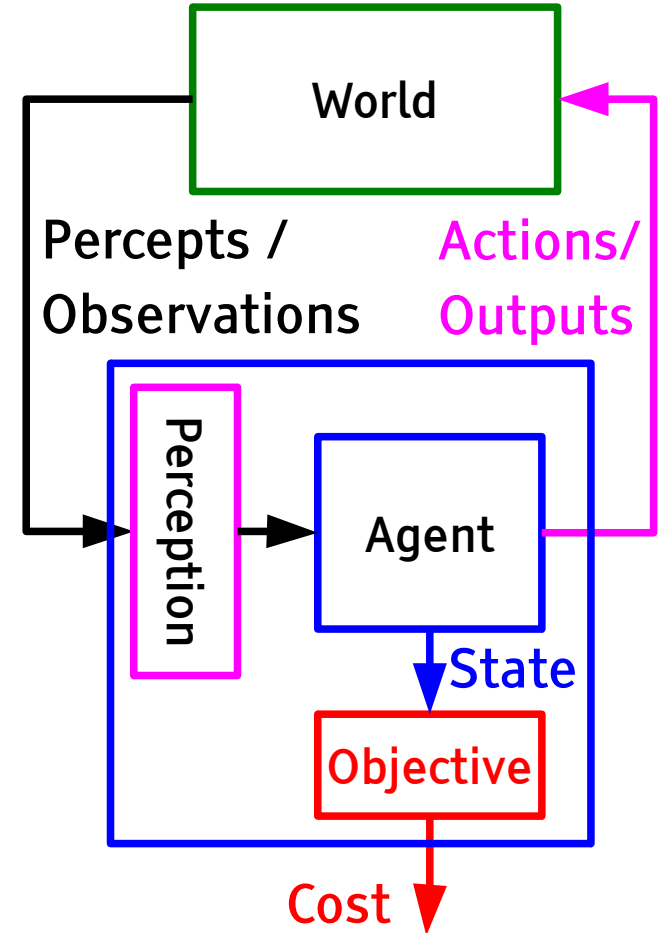
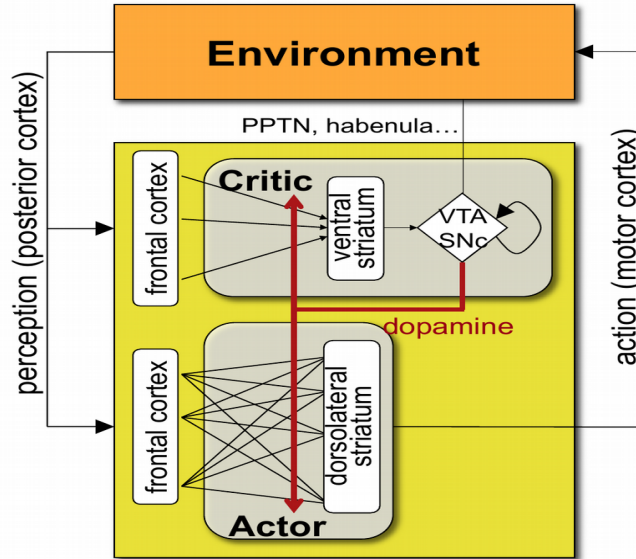
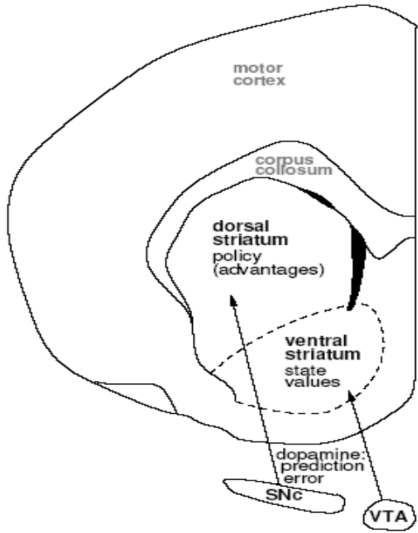
# Architecture of an AI Agent

- TL;DR:
  - AI Agent = perception + world model + actor + critic + objective



# AI System: Learning Agent + Objective

- ▶ The agent gets percepts from the world
- ▶ The agent acts on the world
- ▶ The agents tries to minimize the long-term expected cost.
- ▶ **Objective has immutable and trainable parts**

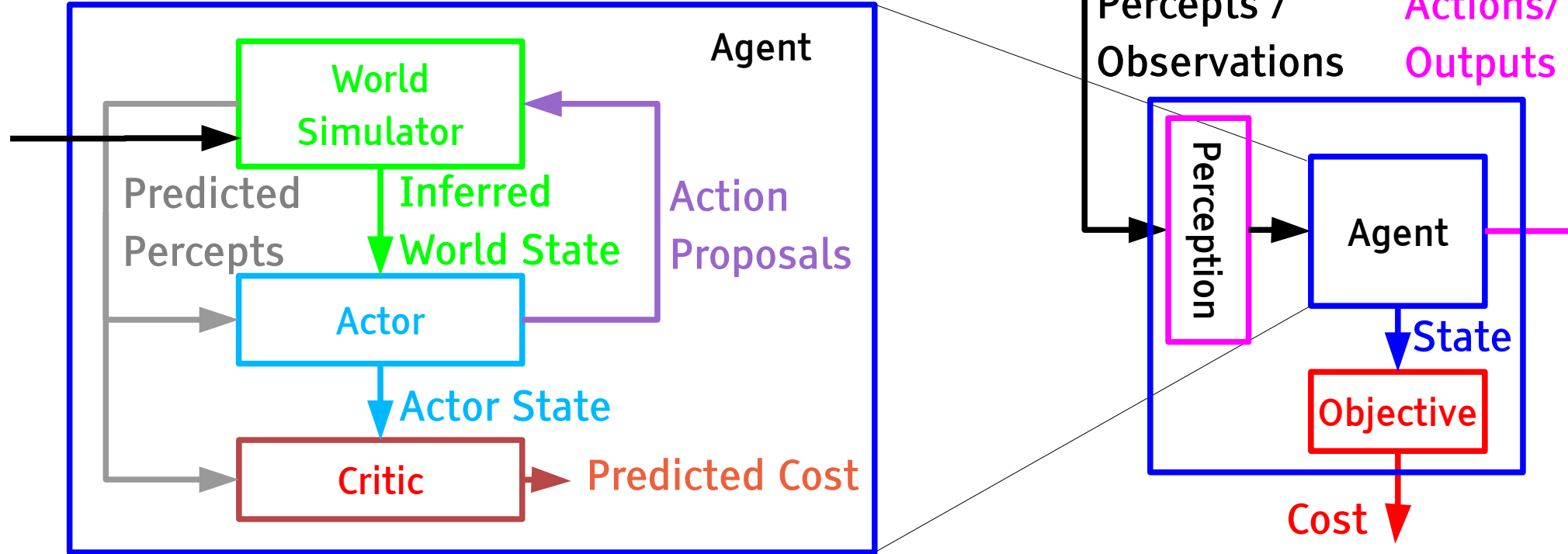






# AI Agent: Reasoning = Prediction + Planning

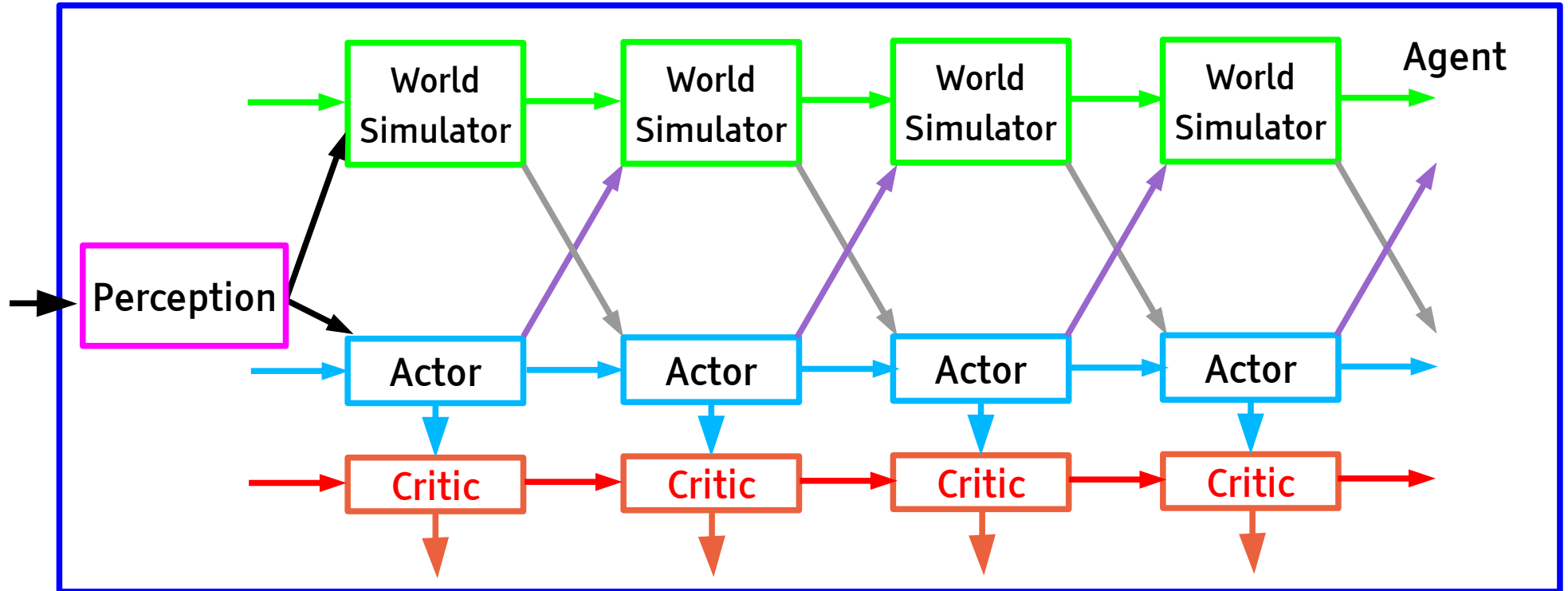
- ▶ The essence of intelligence is the ability to predict
- ▶ To plan ahead, we simulate the world
- ▶ The action taken minimizes the predicted cost





# AI Agent: Reasoning = Simulating

- ▶ The essence of intelligence is the ability to predict
- ▶ To plan ahead, we must simulate the world
- ▶ The action taken minimizes the predicted cost



# Learning predictive models

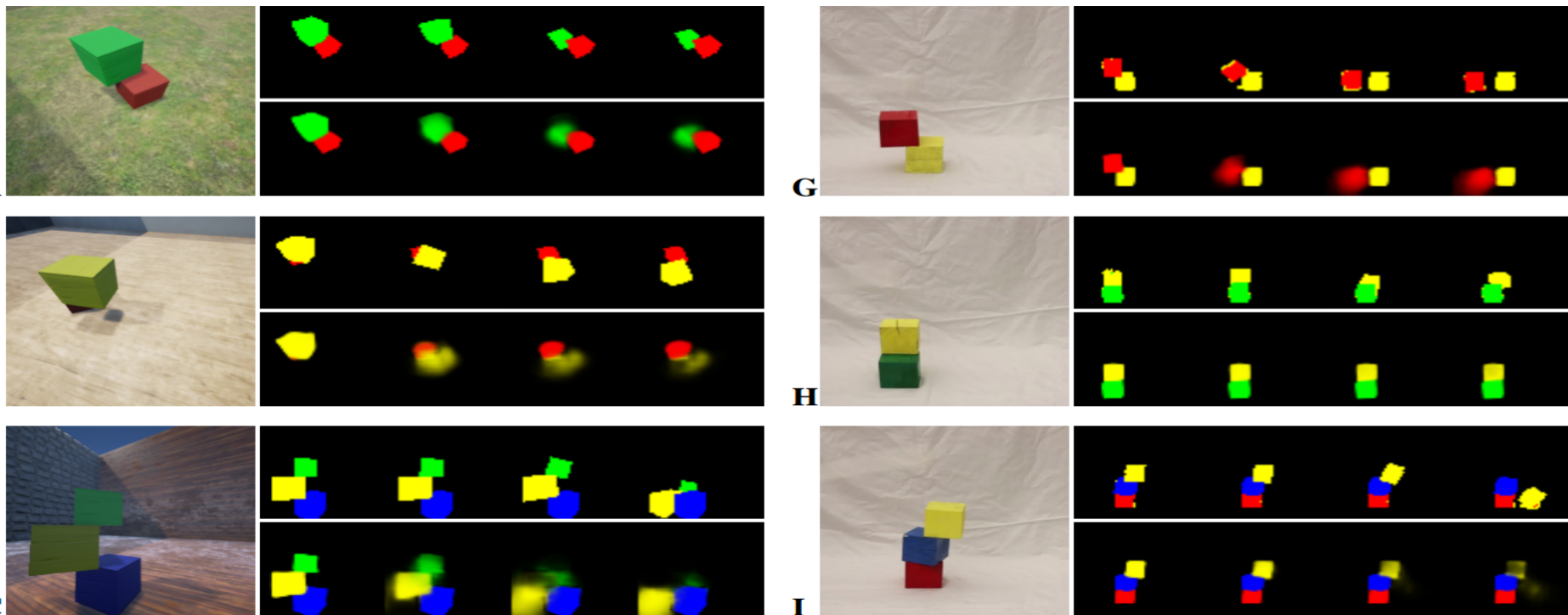
The missing link to AI

- TL;DR:
  - Generative Adversarial Networks are extremely promising



# Learning Physics (PhysNet) [Lerer, Gross, Fergus, ICML'16]

- ▶ ConvNet predicts the trajectories of falling blocks
- ▶ Uses the Unreal game engine hooked up to Torch.





# But in the real world, the future is uncertain...

## ▶ Naïve predictive learning

- ▶ Minimize the prediction error
- ▶ Predict the average of all plausible futures
- ▶ Blurry results



## ▶ Better predictive learning

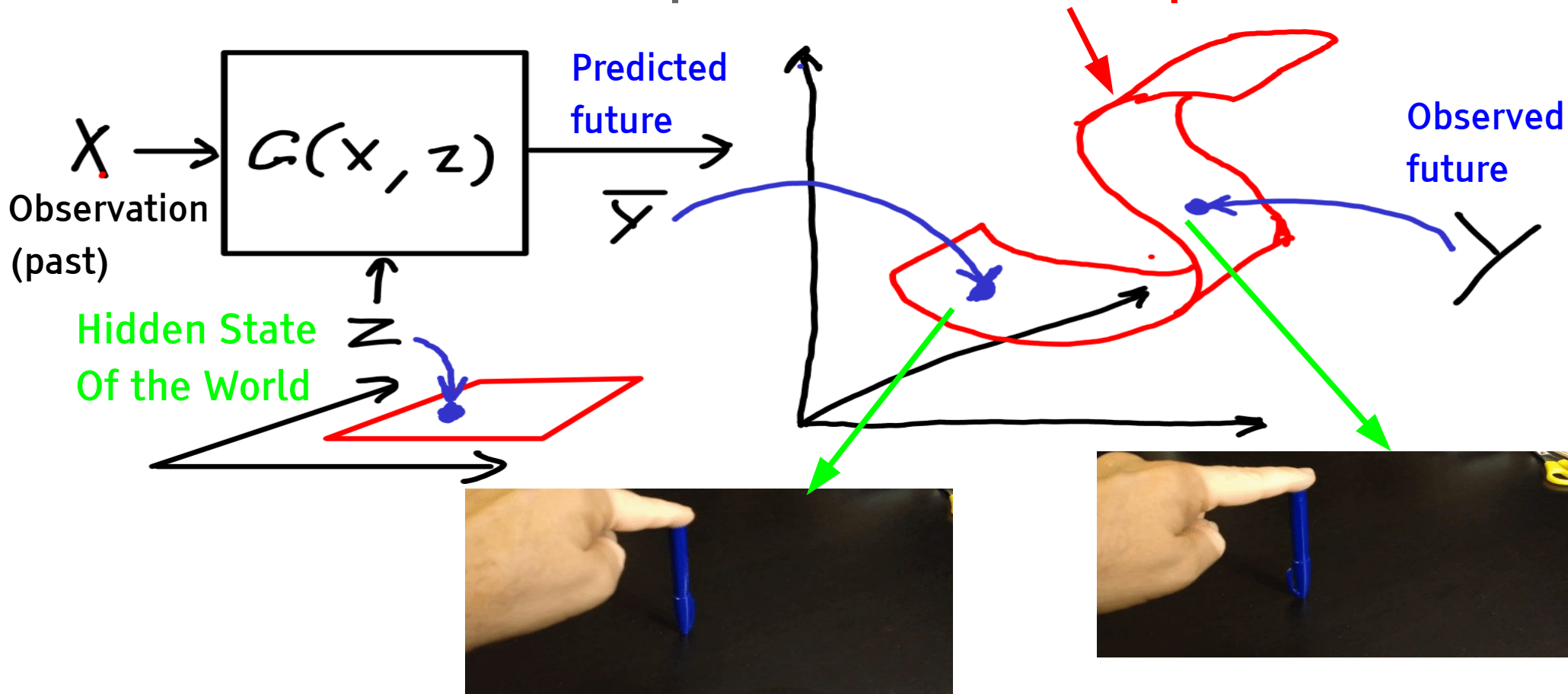
- ▶ Learning the loss function
- ▶ Predict one plausible future among many
- ▶ Sharper results





# The Hard Part: Prediction Under Uncertainty

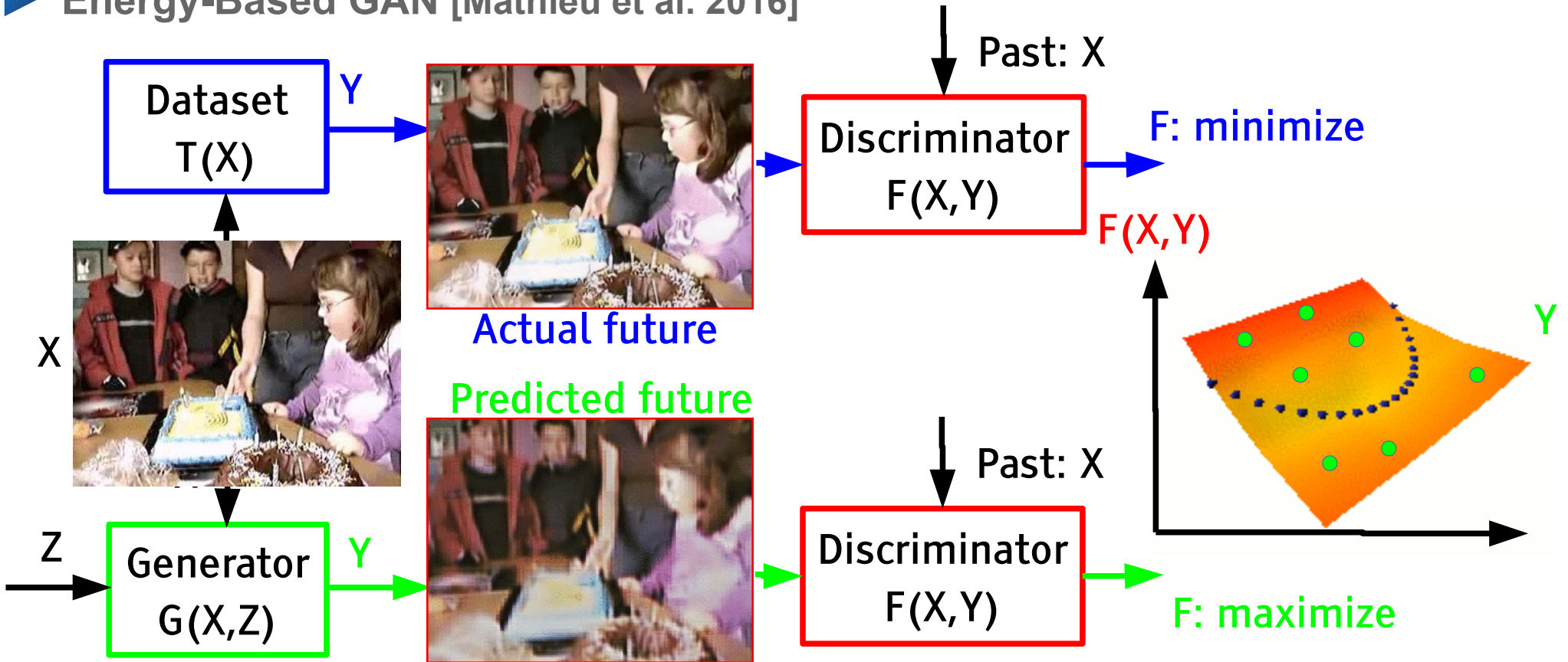
- ▶ The observed future is a representative of a **set of plausible futures**





# Adversarial Training: the key to predicting under uncertainty

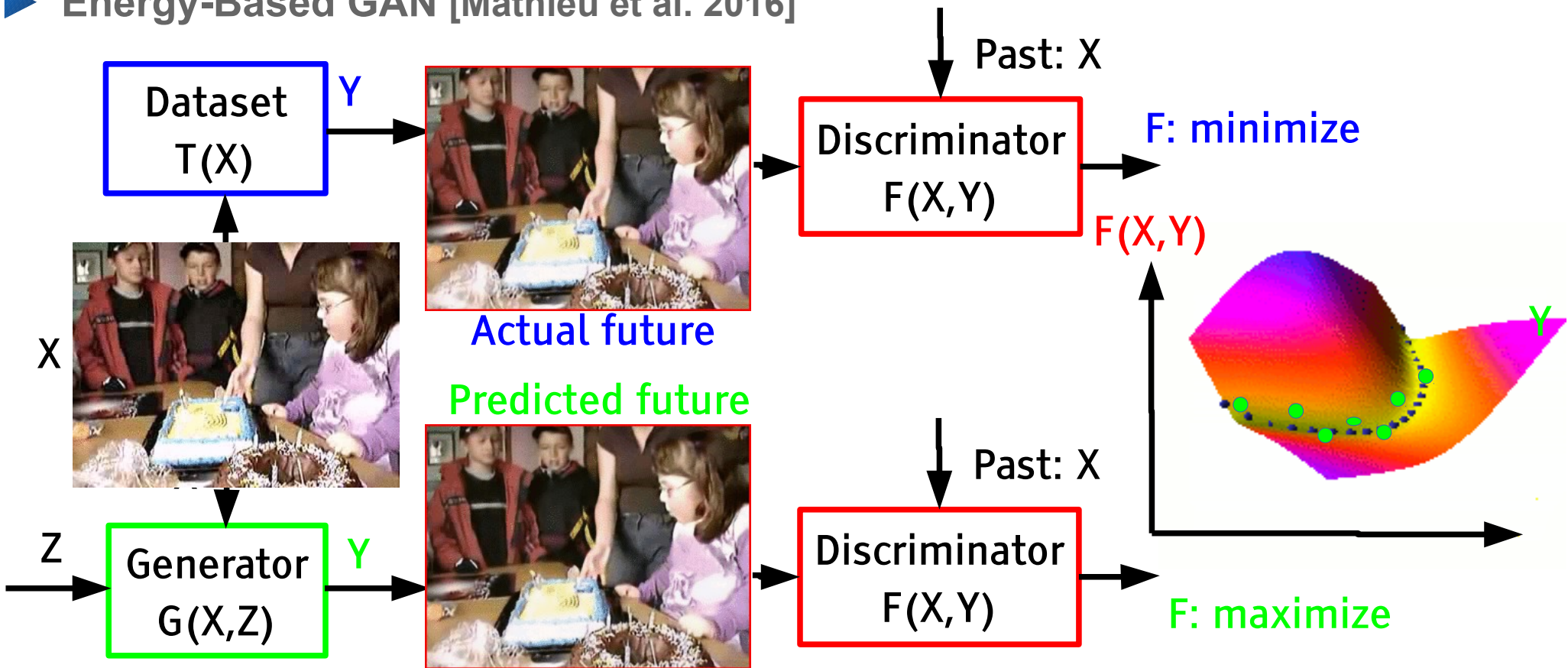
- ▶ Generative Adversarial Networks (GAN) [Goodfellow et al. NIPS 2014],
- ▶ Energy-Based GAN [Mathieu et al. 2016]





# Adversarial Training: the key to predicting under uncertainty

- ▶ Generative Adversarial Networks (GAN) [Goodfellow et al. NIPS 2014],
- ▶ Energy-Based GAN [Mathieu et al. 2016]







# Energy-Based GAN trained on ImageNet at 256x256 pixels

► Trained on dogs





# Video Prediction with GAN

- ▶ [Mathieu, Couprie, LeCun arXiv:1511:05440]
- ▶ But we are far from a complete solution.



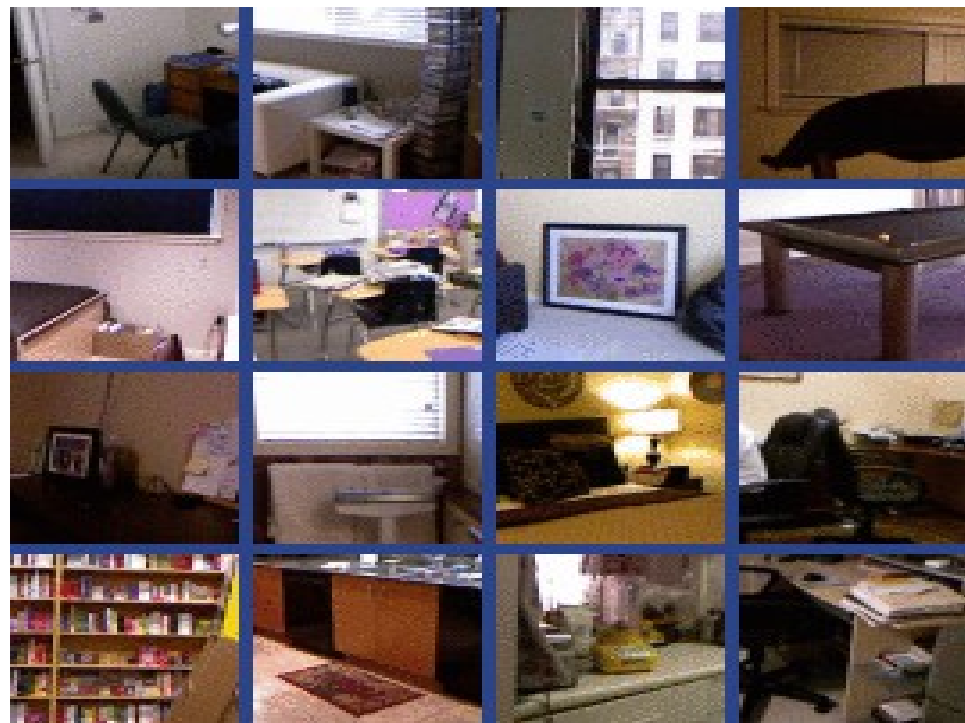


# Video Prediction: predicting 5 frames





# Video Prediction: predicting 5 frames





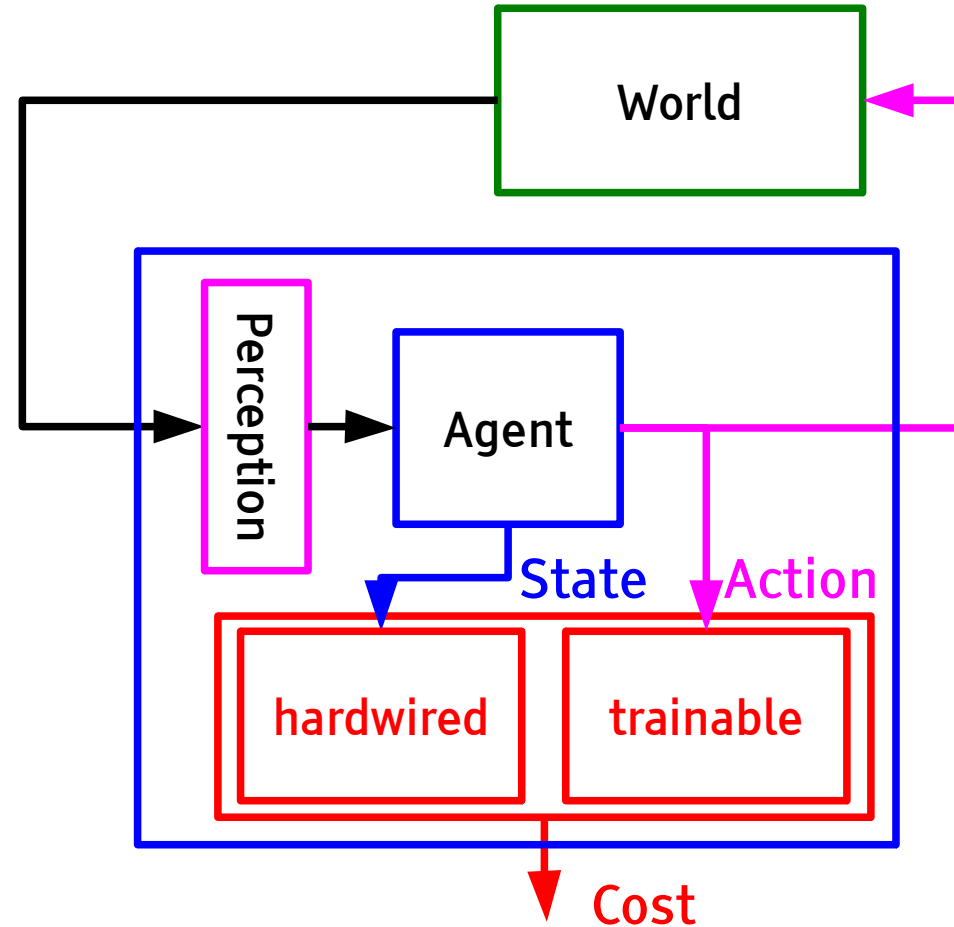
# Video Prediction: predicting 50 frames





# Aligning the Objective with Human Values

- ▶ Make the objective have two components
- ▶ 1. A hardwired, immutable “safeguard” objective
  - ▶ Call it the instinct
- ▶ 2. A Trainable objective that estimates the value function of its human trainers
  - ▶ It’s trained through adversarial training / Inverse RL.
  - ▶ (for once, I’m agreeing with Stuart Russel)





# Inverse RL through Adversarial Training

- ▶ Train the objective to emulate the (unknown) objectives of the human trainers
- ▶ Objective trains to distinguish trainer actions from agent actions
- ▶ Agent plans/trains to minimize objective

