DSC291: Machine Learning with Few Labels

Generative Adversarial Learning

Zhiting Hu
Lecture 12, May 8, 2025

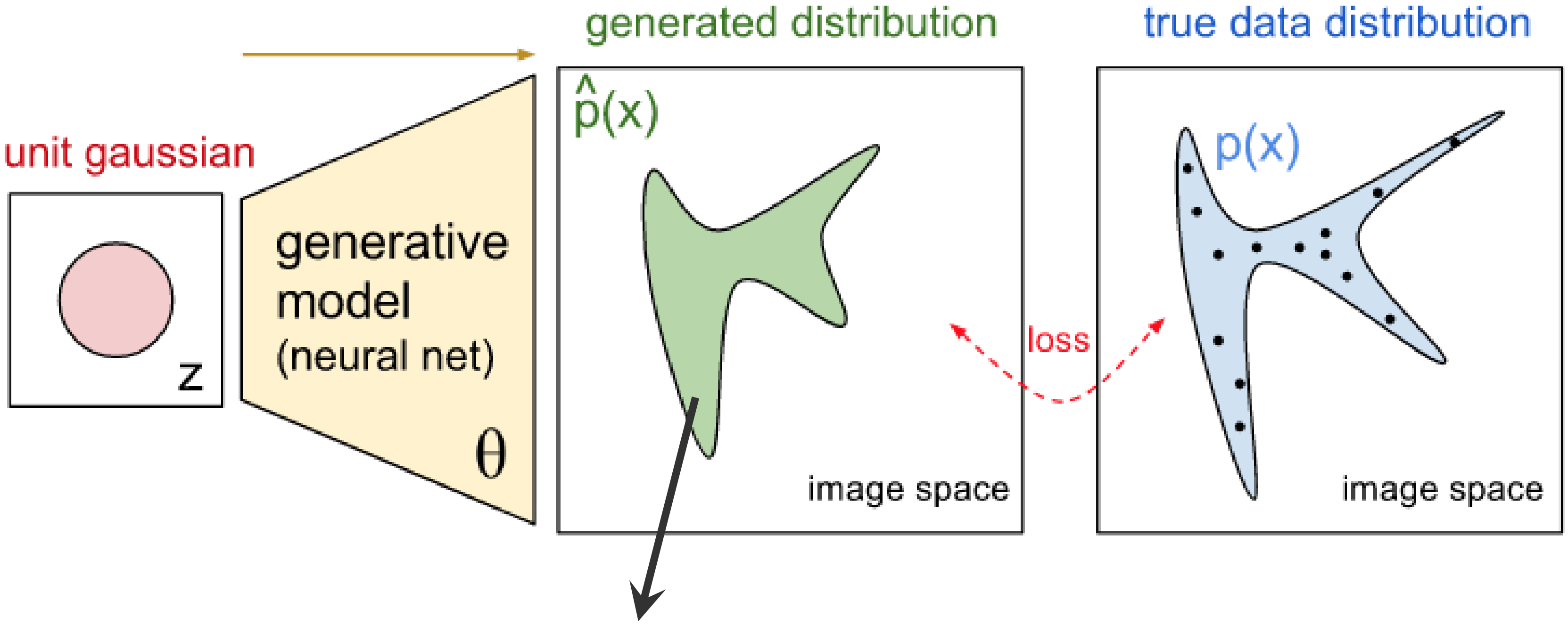
UC San Diego
HALICIOĞLU DATA SCIENCE INSTITUTE

Outline

- Deep Generative Models
  - Generative adversarial learning

- Paper presentation:
  - Letong Liang: “DeepSeek-Prover-V2”
  - Ali El Lahib, Darin Djapri: “TD-MPC2: Scalable, Robust World Models for Continuous Control”

Recap: Implicit Generative Models



https://blog.openai.com/generative-models/

Courtesy: Grosse CSC321 Lecture 19

Recap: Generative Adversarial Nets (GANs)

- Learning
  - A minimax game between the generator and the discriminator
  - Train D to maximize the probability of assigning the correct label to both training examples and generated samples
  - Train G to fool the discriminator

$$\max_D \mathcal{L}_D = \mathbb{E}_{\boldsymbol{x}\sim p_{data}(\boldsymbol{x})}\left[\log D(\boldsymbol{x})\right] + \mathbb{E}_{\boldsymbol{x}\sim G(\boldsymbol{z}),\boldsymbol{z}\sim p(\boldsymbol{z})}\left[\log(1 - D(\boldsymbol{x}))\right]$$
$$\min_G \mathcal{L}_G = \mathbb{E}_{\boldsymbol{x}\sim G(\boldsymbol{z}),\boldsymbol{z}\sim p(\boldsymbol{z})}\left[\log(1 - D(\boldsymbol{x}))\right].$$

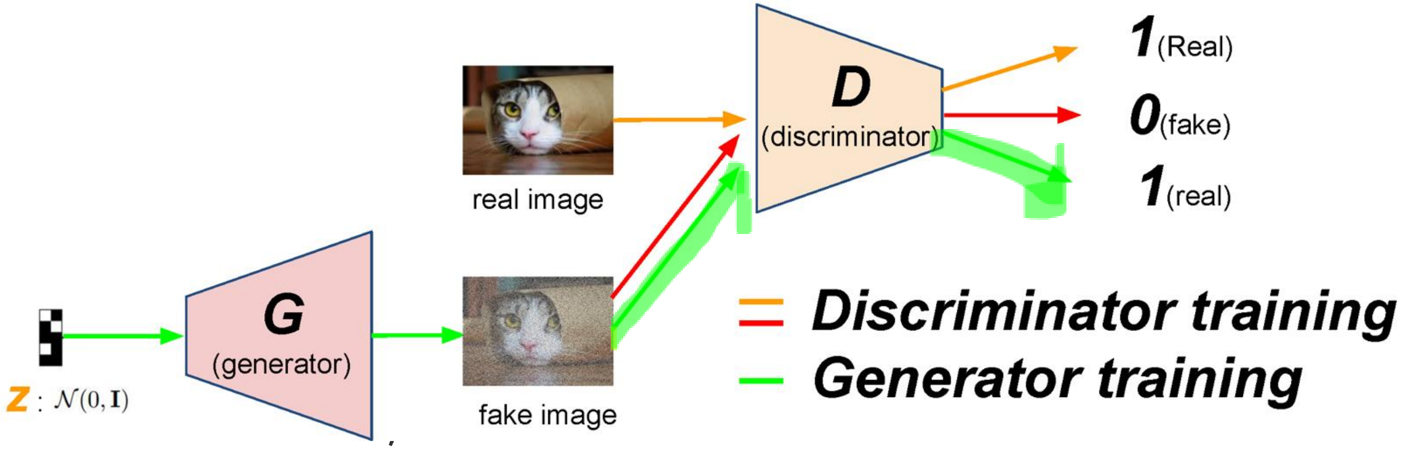


Figure courtesy: Kim

Recap: Optimality of GANs

Question: in practice, we're unlikely to get the optimal D^*. In this case, what is the minimax game truly optimizing?

- The minimax game can now be reformulated as

$$
\begin{aligned}
C(G) &= \max_D V(G, D) \\
&= \mathbb{E}_{\boldsymbol{x} \sim p_{\text{data}}}[\log D_G^*(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z} \sim p_{\boldsymbol{z}}}[\log(1 - D_G^*(G(\boldsymbol{z})))] \\
&= \mathbb{E}_{\boldsymbol{x} \sim p_{\text{data}}}[\log D_G^*(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{x} \sim p_g}[\log(1 - D_G^*(\boldsymbol{x}))] \\
&= \mathbb{E}_{\boldsymbol{x} \sim p_{\text{data}}}\left[\log \frac{p_{\text{data}}(\boldsymbol{x})}{P_{\text{data}}(\boldsymbol{x}) + p_g(\boldsymbol{x})}\right] + \mathbb{E}_{\boldsymbol{x} \sim p_g}\left[\log \frac{p_g(\boldsymbol{x})}{p_{\text{data}}(\boldsymbol{x}) + p_g(\boldsymbol{x})}\right]
\end{aligned}
$$

Theorem 1. *The global minimum of the virtual training criterion $C(G)$ is achieved if and only if $p_g = p_{data}$. At that point, $C(G)$ achieves the value $-\log 4$.*

$$
C(G) = -\log(4) + KL\left(p_{\text{data}} \left\| \frac{p_{\text{data}} + p_g}{2}\right.\right) + KL\left(p_g \left\| \frac{p_{\text{data}} + p_g}{2}\right.\right)
$$

$$
= -\log(4) + 2 \cdot JSD\left(p_{\text{data}} \| p_g\right) \quad \text{Jensen-Shannon Divergence}
$$

[Goodfellow et al., 2014]

Wasserstein GAN (WGAN)

- If our data are on a low-dimensional manifold of a high dimensional space, the model's manifold and the true data manifold can have a negligible intersection in practice

[Arjovsky et al., 2017] Slide adapted from bhiksha

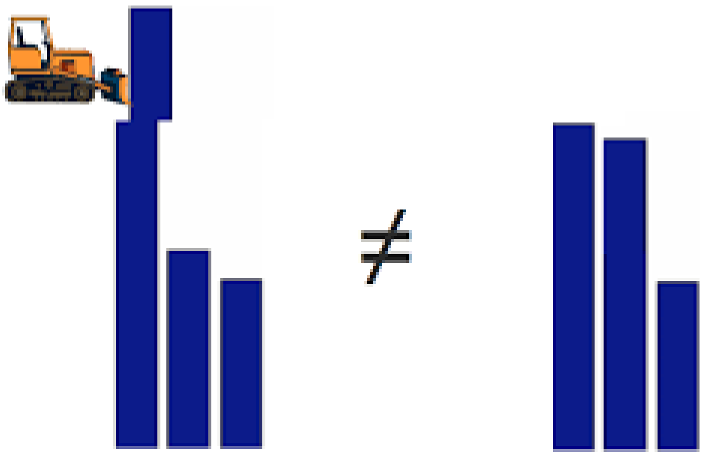
Wasserstein GAN (WGAN)

- If our data are on a low-dimensional manifold of a high dimensional space, the model's manifold and the true data manifold can have a negligible intersection in practice
- The loss function and gradients may not be continuous and well behaved

[Arjovsky et al., 2017] Slide adapted from bhiksha

Wasserstein GAN (WGAN)

- If our data are on a low-dimensional manifold of a high dimensional space, the model's manifold and the true data manifold can have a negligible intersection in practice
- The loss function and gradients may not be continuous and well behaved
- The Wasserstein Distance is well defined
  - Earth Mover's Distance
  - Minimum transportation cost for making one pile of dirt in the shape of one probability distribution to the shape of the other distribution



[Arjovsky et al., 2017] Slide adapted from bhiksha

Wasserstein GAN (WGAN)

- Objective

$$W(p_{data}, p_g) = \frac{1}{K} \sup_{||D||_L \leq K} \mathrm{E}_{x\sim p_{data}}[D(x)] - \mathrm{E}_{x\sim p_g}[D(x)]$$

- $||D||_L \leq K$: K- Lipschitz continuous
- Use gradient-clipping to ensure D has the Lipschitz continuity

Progressive GAN

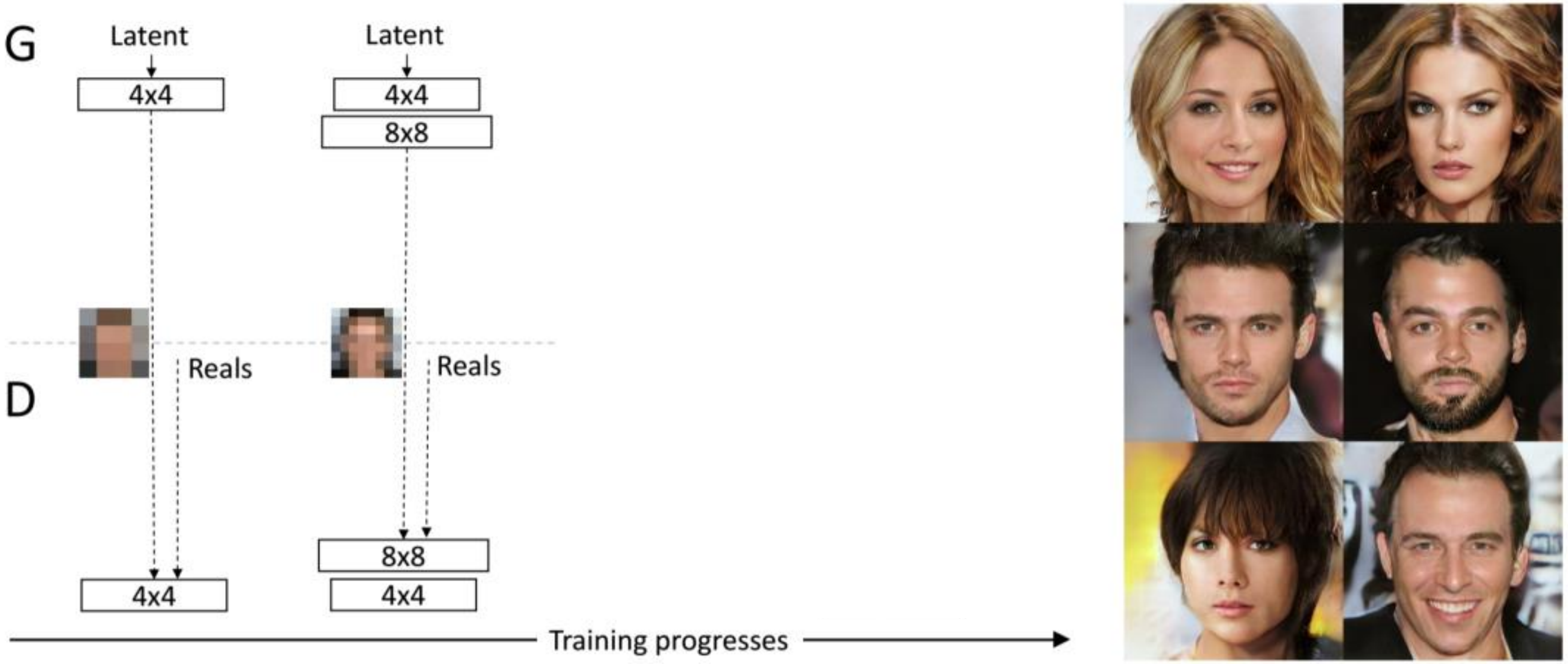
Low resolution images

[Karras et al., 2018]

Progressive GAN

Low resolution images

add in
additional
layers



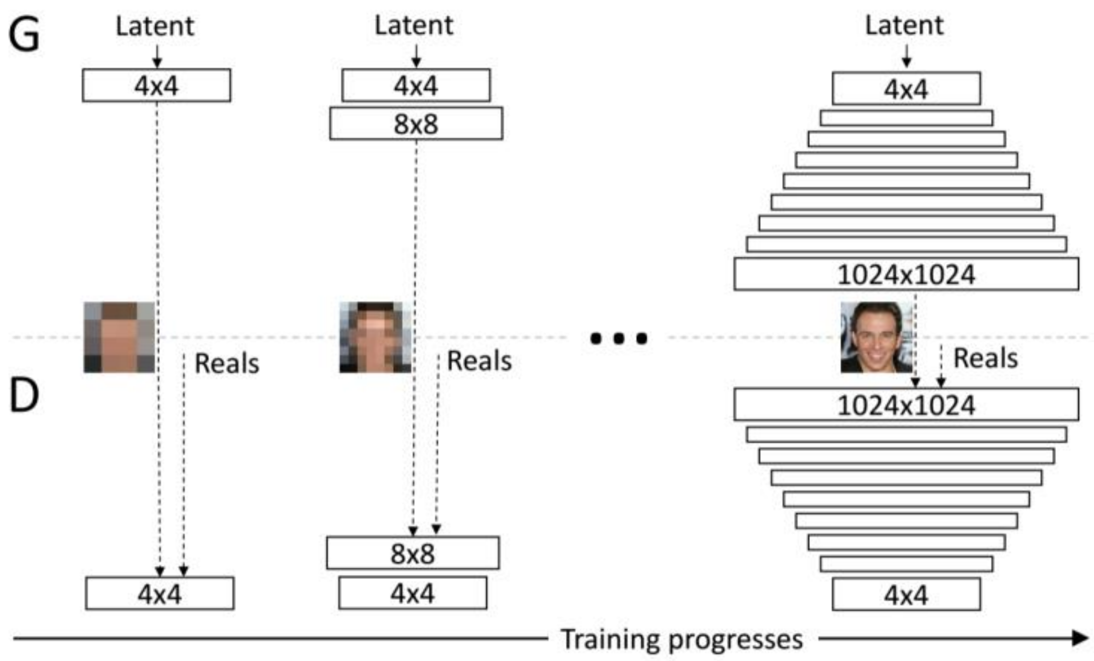
[Karras et al., 2018]

Progressive GAN

Low resolution images

add in
additional
layers

High resolution images



[Karras et al., 2018]

BigGAN

[Brock et al., 2018]

BigGAN

- GANs benefit dramatically from scaling

[Brock et al., 2018]

BigGAN

- GANs benefit dramatically from scaling
- 2x – 4x more parameters
- 8x larger batch size
- Simple architecture changes that improve scalability

[Brock et al., 2018]

BigGAN

- GANs benefit dramatically from scaling
- 2x – 4x more parameters
- 8x larger batch size
- Simple architecture changes that improve scalability

[Brock et al., 2018]

BigGAN

- GANs benefit dramatically from scaling
- 2x
- 8x
- Sim

[Brock et al., 2018]

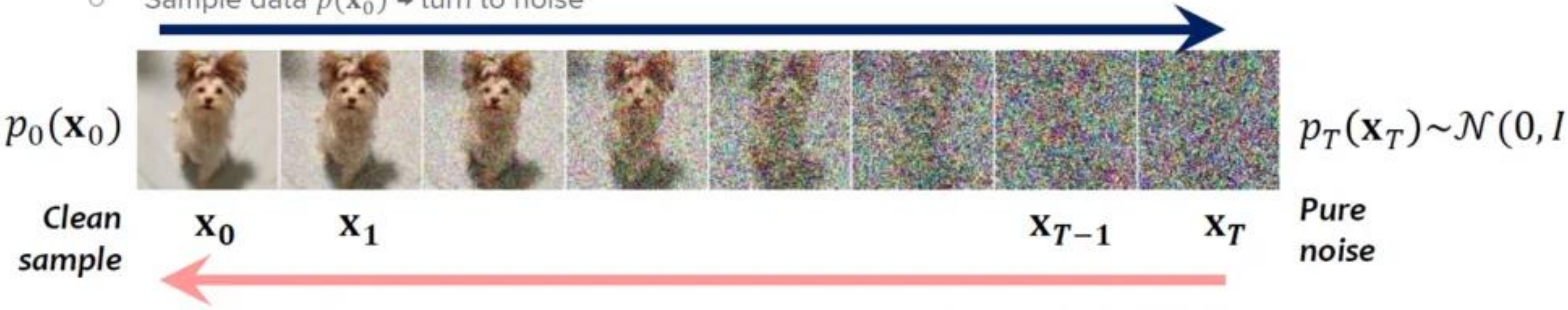
Key Takeaways

- Deep Generative Models: brief history

- GANs:
  - Implicit generative model
  - Minimax formulation
  - Wasserstein GAN

Diffusion model

- **Forward / noising process**
  - Sample data $p(\mathbf{x}_0)$ → turn to noise



- **Reverse / denoising process**
  - Sample noise $p_T(\mathbf{x}_T)$ → turn into data

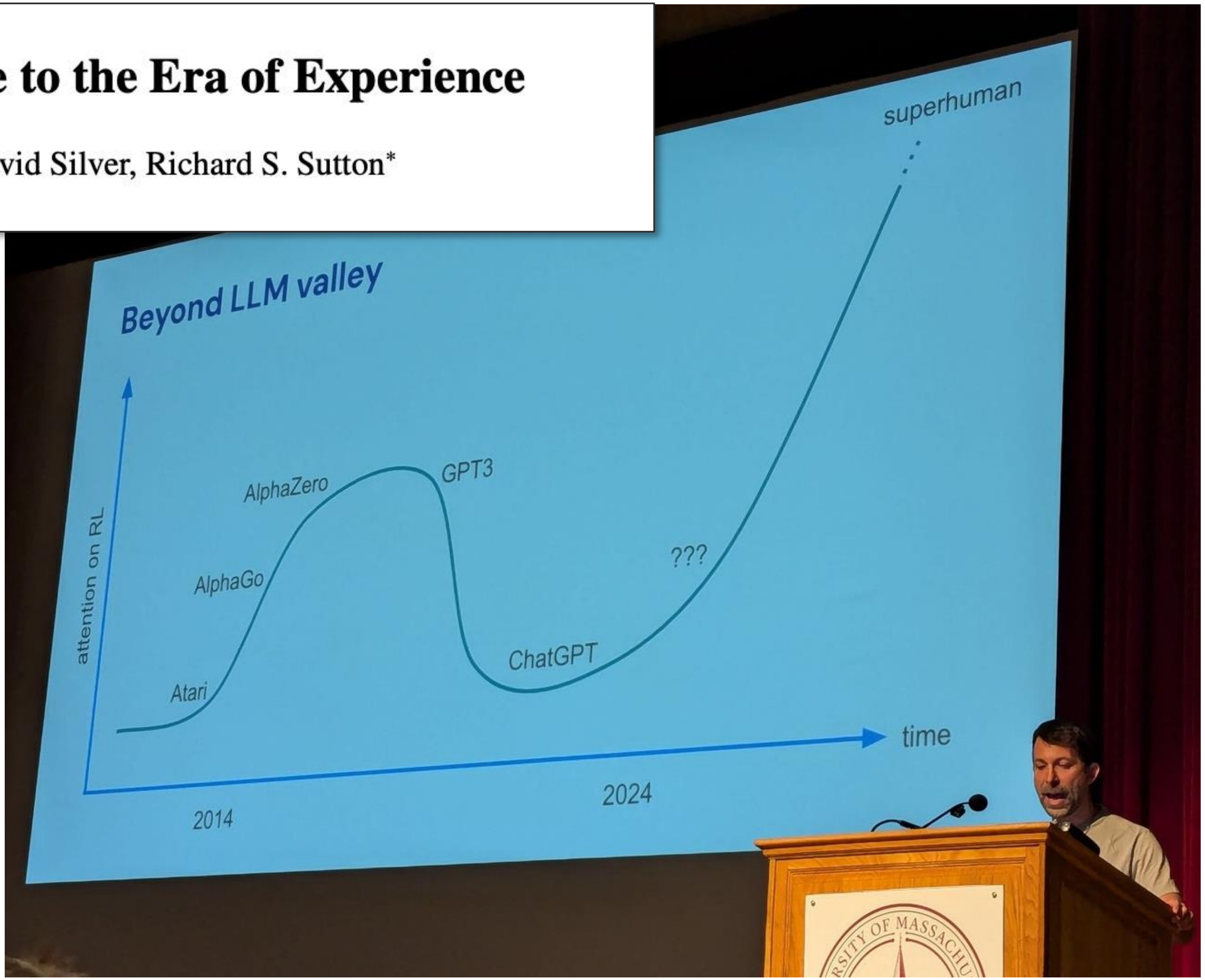
Reinforcement Learning

RL Conference 2024

RL Conference 2024

Welcome to the Era of Experience

David Silver, Richard S. Sutton*



So far… Supervised Learning

Data: (x, y)
x is data, y is label

Goal: Learn a *function* to map x -> y

Examples: Classification, regression, object detection, semantic segmentation, image captioning, etc.

→ Cat

Classification

So far… Unsupervised Learning

Data: x
no labels!

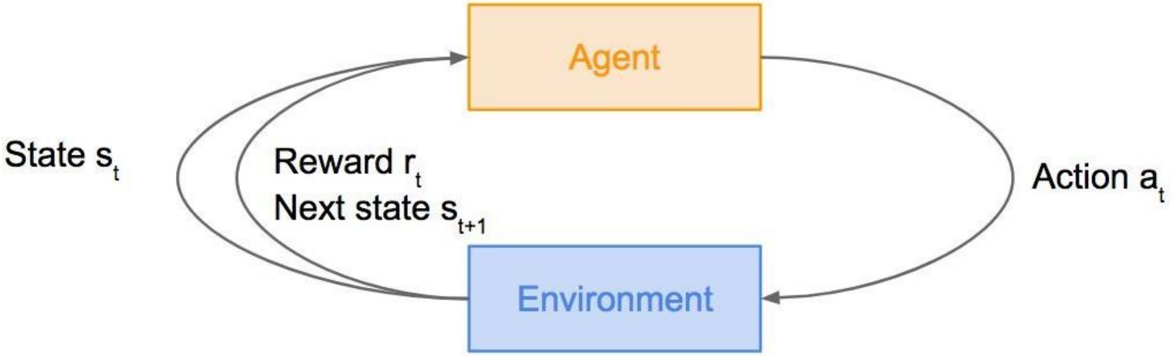
Goal: Learn some underlying hidden *structure* of the data

Examples: Clustering, dimensionality reduction, feature learning, density estimation, etc.

Today: Reinforcement Learning

Problems involving an **agent** interacting with an **environment**, which provides numeric **reward** signals

Goal: Learn how to take actions in order to maximize reward



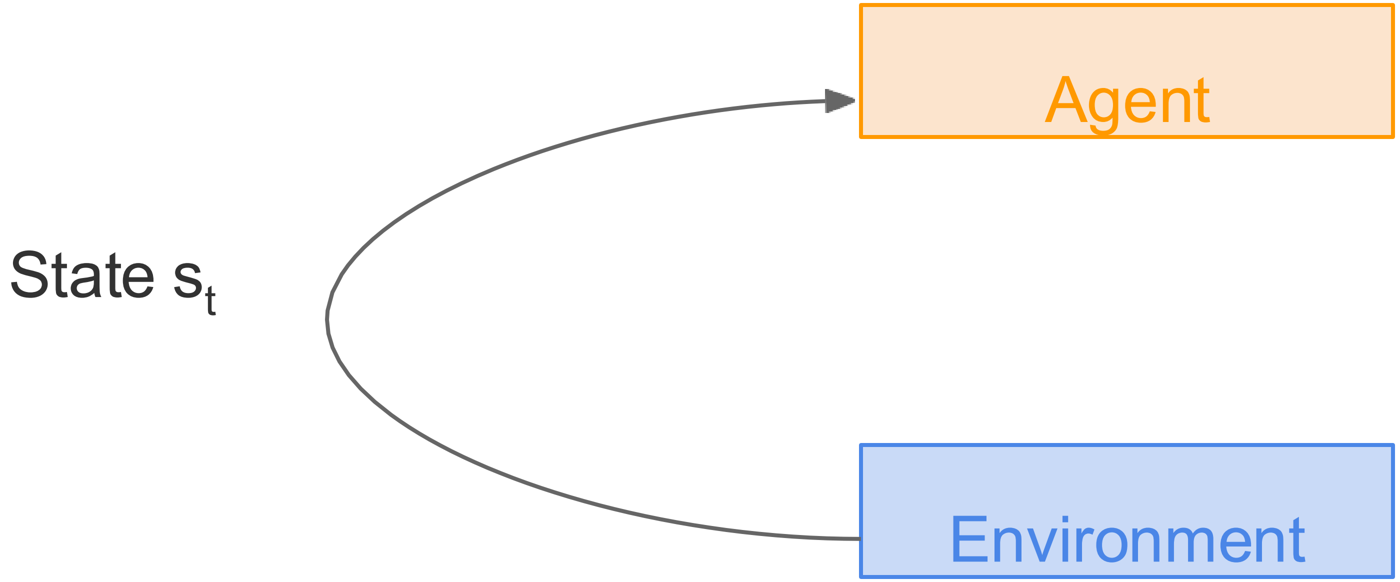
Atari games figure copyright Volodymyr Mnih et al., 2013. Reproduced with permission.

Overview

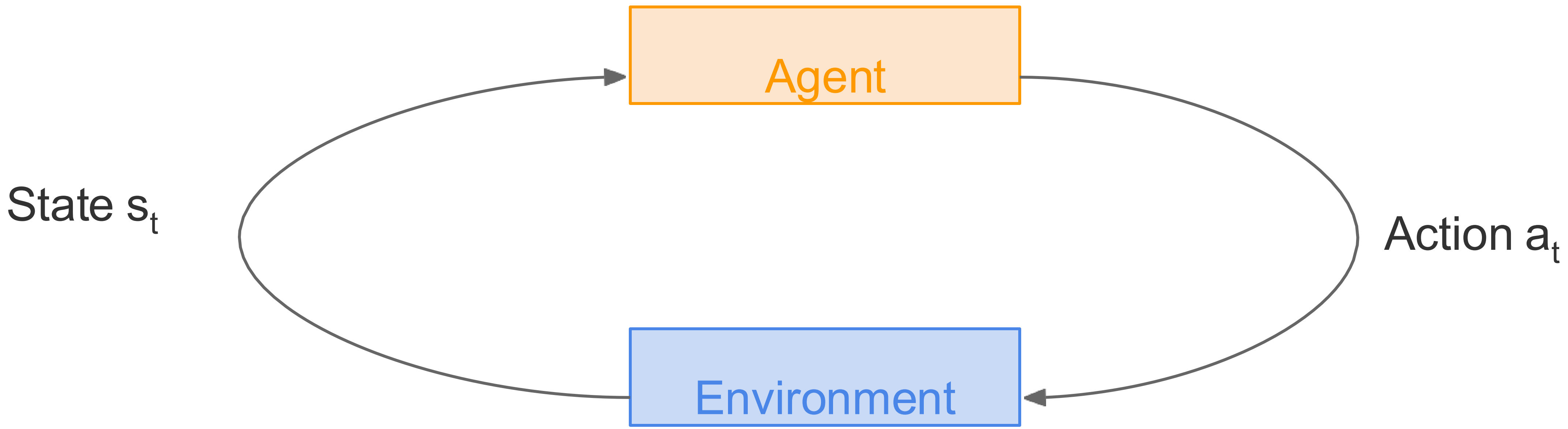
- What is Reinforcement Learning?
- Markov Decision Processes
- Q-Learning
- Policy Gradients

Reinforcement Learning

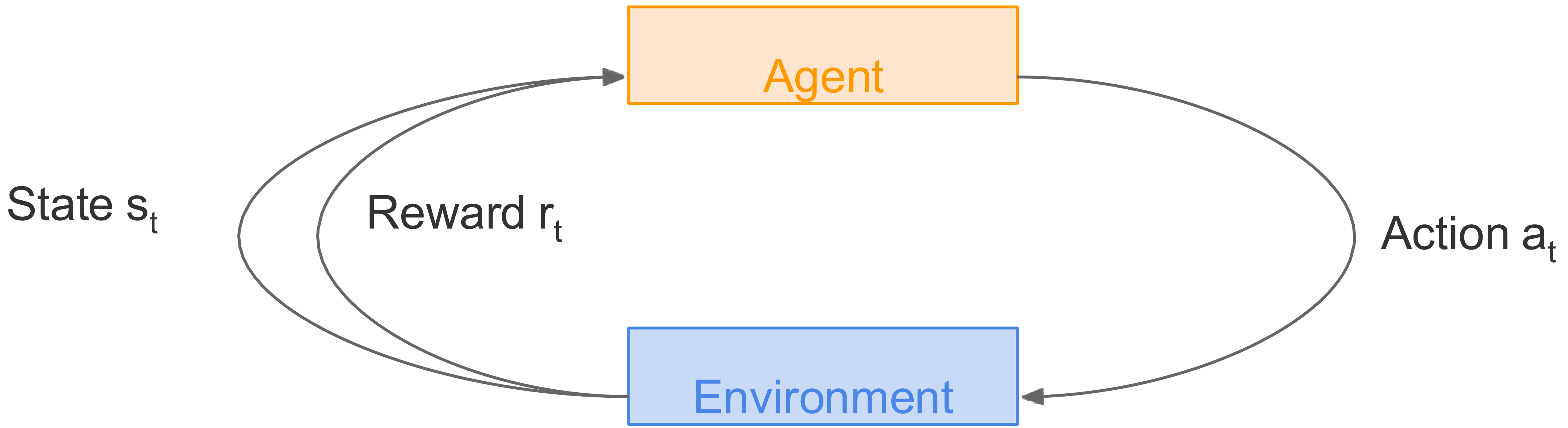
Reinforcement Learning



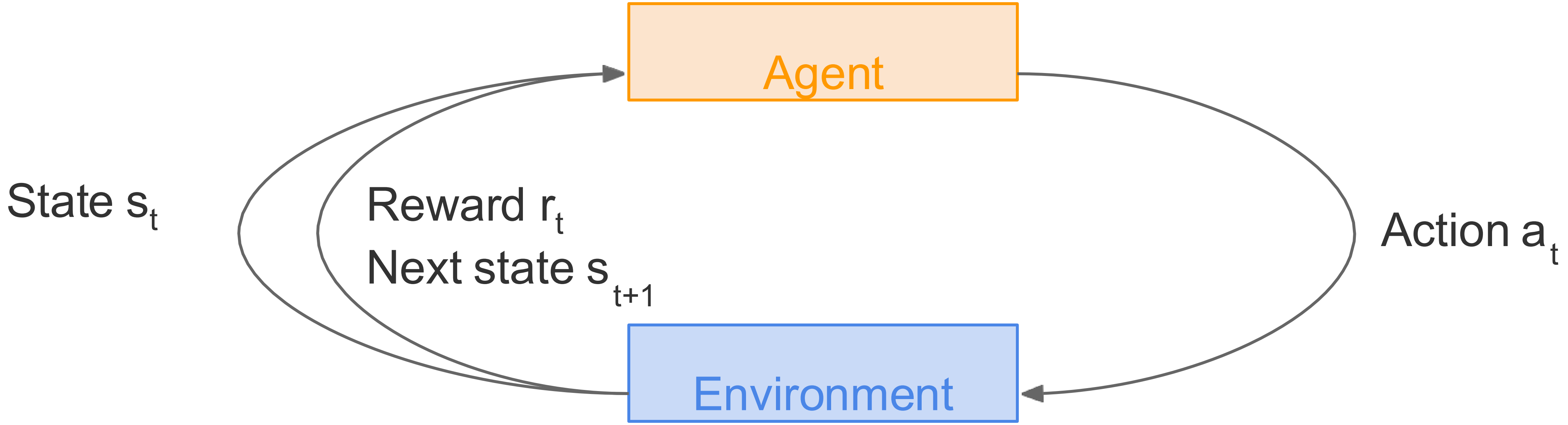
Reinforcement Learning



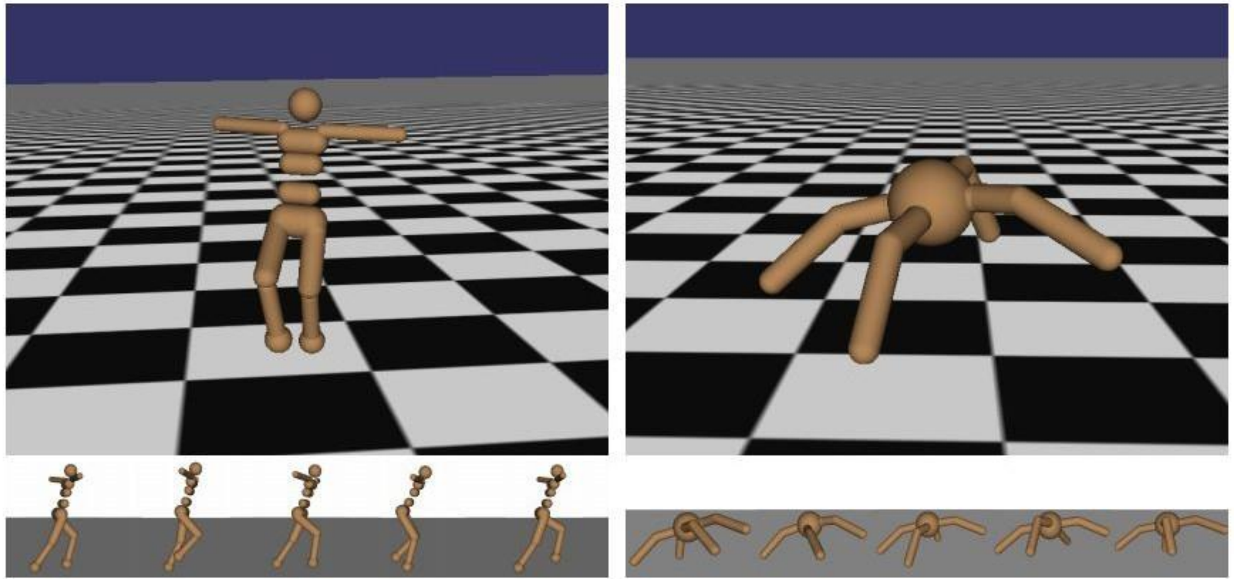
Reinforcement Learning



Reinforcement Learning



Robot Locomotion



Objective: Make the robot move forward

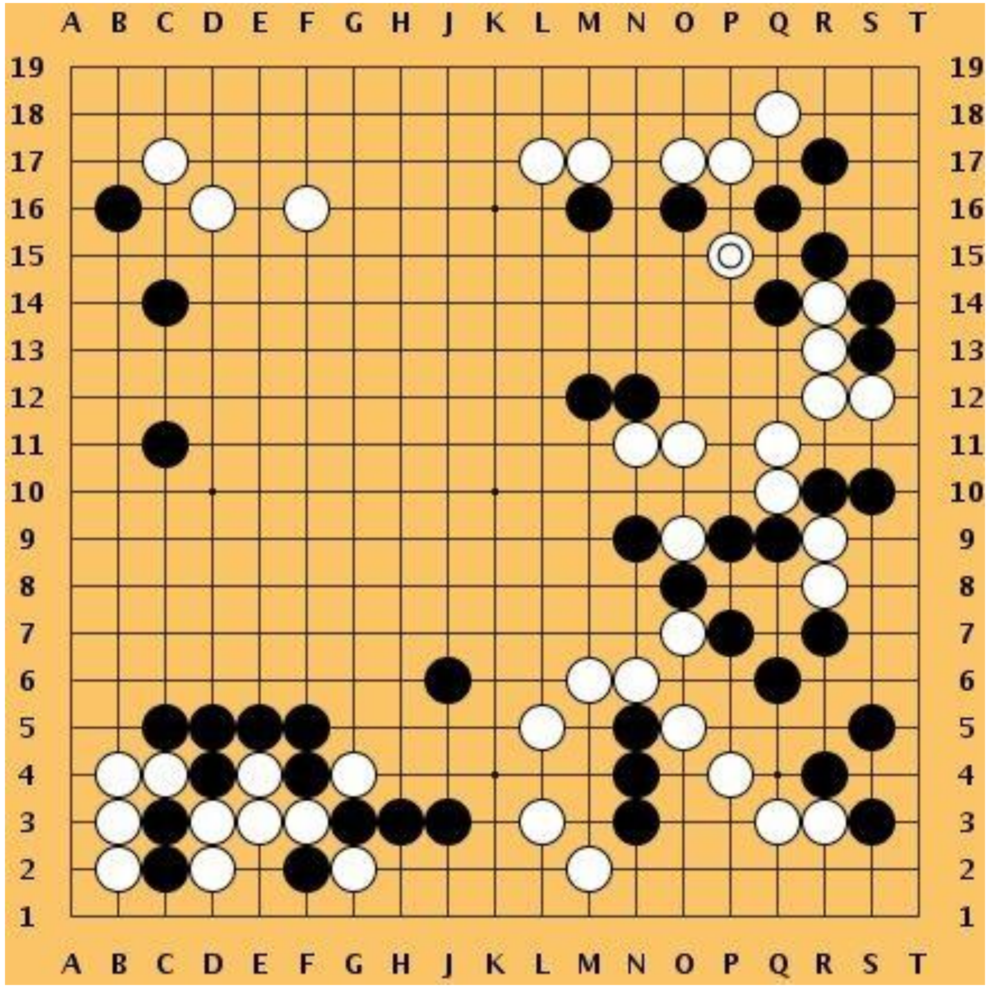
State: Angle and position of the joints
Action: Torque applied on joints
Reward: 1 at each time step upright + forward movement

Atari Games

Objective: Complete the game with the highest score

State: Raw pixel inputs of the game state
Action: Game controls e.g. Left, Right, Up, Down
Reward: Score increase/decrease at each time step

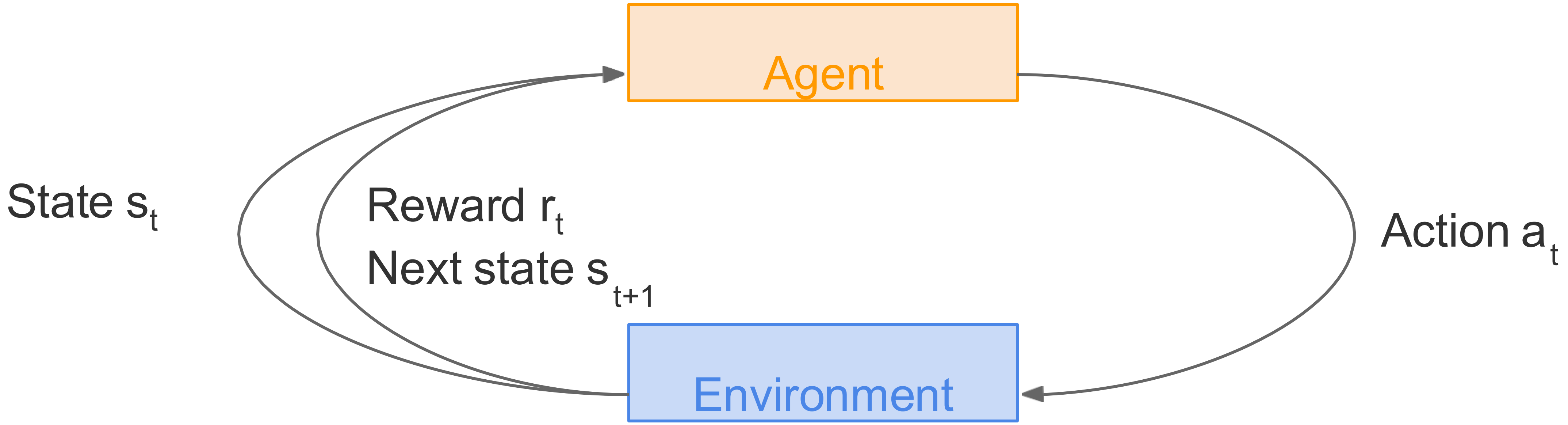
Go



Objective: Win the game!

State: Position of all pieces
Action: Where to put the next piece down
Reward: 1 if win at the end of the game, 0 otherwise

How can we mathematically formalize the RL problem?



Markov Decision Process

- Mathematical formulation of the RL problem
- **Markov property**: Current state completely characterises the state of the world

Defined by: $(\mathcal{S}, \mathcal{A}, \mathcal{R}, \mathbb{P}, \gamma)$

$\mathcal{S}$: set of possible states
$\mathcal{A}$: set of possible actions
$\mathcal{R}$: distribution of reward given (state, action) pair
$\mathbb{P}$: transition probability i.e. distribution over next state given (state, action) pair
γ : discount factor

Markov Decision Process

- At time step t=0, environment samples initial state $s_0 \sim p(s_0)$
- Then, for t=0 until done:
  - Agent selects action a_t
  - Environment samples reward $r_t \sim R(. | s_t, a_t)$
  - Environment samples next state $s_{t+1} \sim P(. | s_t, a_t)$
  - Agent receives reward r_t and next state s_{t+1}

- A policy π is a function from S to A that specifies what action to take in each state
- **Objective**: find policy π* that maximizes cumulative discounted reward: $\sum_{t \geq 0} \gamma^t r_t$

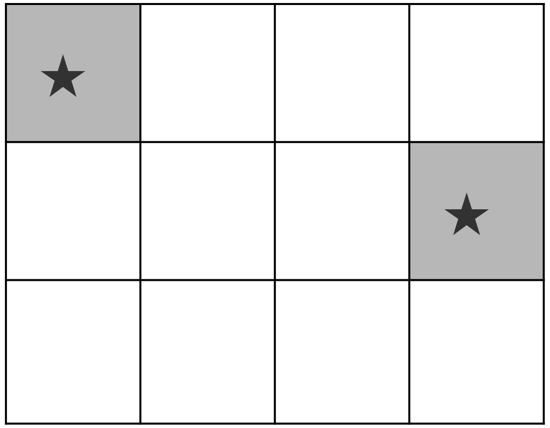
A simple MDP: Grid World

actions = {

1. right
2. left
3. up
4. down

}

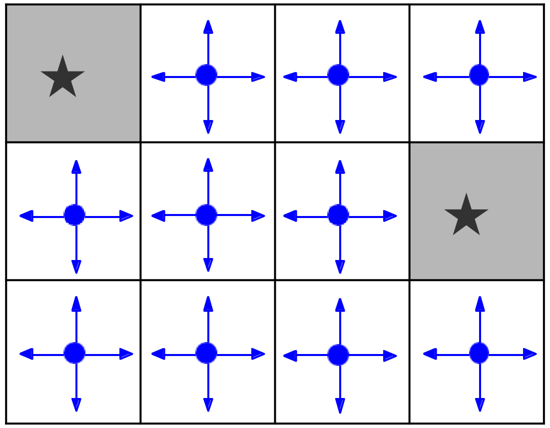
states



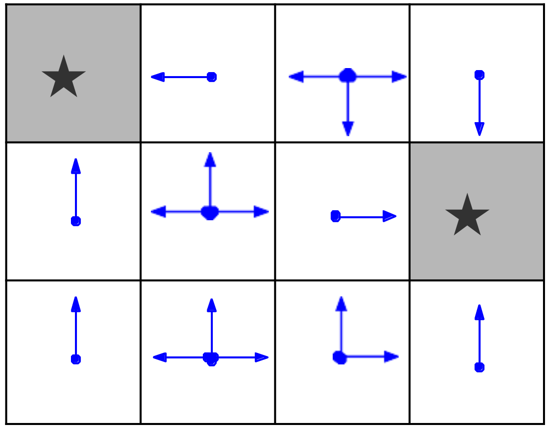
Set a negative “reward” for each transition (e.g. $r = -1$)

Objective: reach one of terminal states (greyed out) in least number of actions

A simple MDP: Grid World



Random Policy



Optimal Policy

Questions?