This lecture will be recorded!!!

Welcome to

DS595
Reinforcement Learning

Prof. Yanhua Li

Time: 6:00pm –8:50pm W
Zoom Lecture
Spring 2022

Project 4 Due 4/25 M mid-night

- https://github.com/yingxue-zhang/DS595-RL-Projects/tree/master/Project4
- Important Dates:
- ~~Project Proposal: 3/30/2022~~
- ~~Project Progress: 4/13/2022~~
  - ~~Brief intro of your project progress today if time allows.~~
- Final Project: Monday 4/25/2022
  - **Project report due on 4/25/2022 midnight on Canvas discussion board.**
  - **Self-and-cross evaluation form due 4/27/2022 midnight on canvas.**
  - **Poster session on 4/27/2022 on Zoom**

Project 4 Poster Session Next Wed

Each project has an ID.

- 1. Training Agent w/ RL to race in TrackMania 2020
- 2. Playing Doom with a Deep Recurrent Q Network
- 3. Robotic Grasping with Reinforcement Learning
- 4. Comparisons of Deep Q-learning (DQN) with Proximal Policy Optimization (PPO) on Lunar Lander
- 5. AlphaGomoku: Mastering The Game of Gomoku using The AlphaZero Learning Method
- 6. Comparison of Reinforcement Learning models applied to MineRL challenge
- 7. Reinforcement Learning for Autonomous Driving using CARLA Simulator
- 8. Super Mario Bros (w/ A2C, PPO, DQN, etc.)
- 9. Continuous & Mapless Navigation for robots using Policy Gradient Algorithm
- 10. Training Agent w/ RL to the puzzle video game: Minesweeper
- 11. Comparison of Discrete and Continuous Action Spaces in Car Racing Environment

https://github.com/yingxue-zhang/DS595-RL-Projects/tree/master/Project4

Team project #4

11 Teams (Last offering in Fall 2020: 9 teams)

Final report due 4/25/2022 M, submit it to Canvas discussion board in teams.

- https://github.com/yingxue-zhang/DS595-RL-Projects/tree/master/Project4

- We will host a poster session next Wed on 4/27/2022 with breakout rooms for group "posters". ☺

- We will let participates to choose a breakout room

- Intro of breakout room feature.

- https://www.youtube.com/watch?v=GDKJM6JhyUY

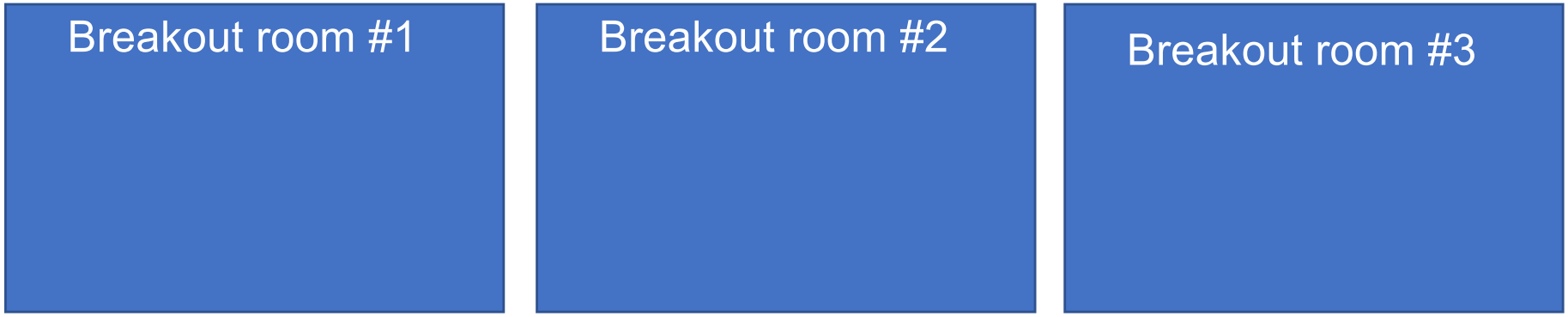
Team project #4 (on 4/27)

Final Poster Session in Zoom breakout rooms.

	Breakout room #1	Breakout room #2	Breakout room #3
6pm-6:25pm	Team 1 TA: Yingxue Zhang	Team 4 PhD student: Xin Zhang	Team 7 Prof Yanhua Li
6:30pm-6:55pm	Team 2	Team 5	Team 8
7:00pm-7:25:pm	Team 3	Team 6	Team 9
7:30pm-7:55:pm	Team 10	Team 11	Closed

8pm-8:50pm Breakout rooms all closed. Free discussion in the main Zoom meeting room.

Let's try breakout rooms now



Team project #4 Survey

1. How many people do you have in your group? 4 or 5?

Example answer: 5

Due 4/27 W on Canvas

Answer:

2. List all people names of your group, and indicate how much each groupmate (including yourself) contributes to the team project, in percentage (from 0% to 100%)?

Note that if you have a group of 5 people, and everyone contributes equally, each groupmate should contribute 20%. Please make sure that the sum of all contributions to be 100%.

(Optionally, you can also provide more details (if you want), about the details of the contributions made by each individual, such as ``programming", ``data collection", ``report writing", ``analyzing and plotting the figures", ``come up with the idea of using XYZ feature for analyzing the popularity", or anything, you name it.)

Example answer: A(myself): 20%, B: 20%, C: 20%, D: 20%, E: 20%

Final Grading

The grading system for this course is A,B,C,D,F (without +/-).

Oral Work: 10%,

Project #4 poster presentation.

Quizzes/Exams: 30%

5 counted quizzes (Done)

Class projects: 60%

Project 1 for 5%, (Done)

Project 2 for 10%, (Done)

Project 3 for 15%, (Done)

Project 4 for 30% (On-going)

	Reinforcement Learning	Inverse Reinforcement Learning
Single Agent	**Tabular representation of reward** *Model-based control* *Model-free control* *(MC, SARSA, Q-Learning)*	**Linear reward function learning** Imitation learning Apprenticeship learning Inverse reinforcement learning
	Function representation of reward *1. Linear value function approx (MC, SARSA, Q-Learning)* *2. Value function approximation (Deep Q-Learning, Double DQN, prioritized DQN, Dueling DQN)* *3. Policy function approximation (Policy gradient, PPO, TRPO)* *4. Actor-Critic methods (A2C, A3C, Pathwise Derivative PG)* *Advanced topics in RL (Sparse Rewards)*	**Non-linear reward function learning** Generative adversarial imitation learning (GAIL) Adversarial inverse reinforcement learning (AIRL) https://arxiv.org/abs/1710.11248
	Review of Deep Learning *As bases for non-linear function approximation (used in 2-4).*	Review of Generative Adversarial nets as bases for non-linear IRL
Multiple Agents	**Multi-Agent Reinforcement Learning** Multi-agent Actor-Critic etc. **Meta-RL**	**Multi-Agent Inverse Reinforcement Learning** MA-GAIL

This Lecture

- Review Generative Adversarial Networks (GANs)
- IRL as GAN
  - Generative Adversarial Imitation Learning (GAIL)
- Multi-Agent IRL
  - MA-GAIL
- Meta-Learning
  - Meta-Reinforcement Learning
- Class Review

This Lecture

- Review Generative Adversarial Networks (GANs)
- IRL as GAN
  - Generative Adversarial Imitation Learning (GAIL)
- Multi-Agent IRL
  - MA-GAIL
- Meta-Learning
  - Meta-Reinforcement Learning
- Class Review

Introduction of Generative Adversarial Network (GAN)

Generation

Image Generation

$$\begin{bmatrix} 0.3 \\ -0.1 \\ \vdots \\ -0.7 \end{bmatrix} \begin{bmatrix} 0.1 \\ -0.1 \\ \vdots \\ 0.7 \end{bmatrix} \begin{bmatrix} -0.3 \\ 0.1 \\ \vdots \\ 0.9 \end{bmatrix}$$

In a specific range

→ NN Generator →

Sentence Generation

$$\begin{bmatrix} 0.3 \\ -0.1 \\ \vdots \\ -0.7 \end{bmatrix} \begin{bmatrix} 0.1 \\ -0.1 \\ \vdots \\ 0.2 \end{bmatrix} \begin{bmatrix} -0.3 \\ 0.1 \\ \vdots \\ 0.5 \end{bmatrix}$$

→ NN Generator →

How are you?

Good morning.

Good afternoon.

Powered by: http://mattya.github.io/chainer-DCGAN/

Basic Idea of GAN

It is a neural network (NN), or a function.

vector → Generator → image

high dimensional vector

$\begin{bmatrix} 0.1 \\ -3 \\ \vdots \\ 2.4 \\ 0.9 \end{bmatrix}$ → Generator → 

Each dimension of input vector represents some characteristics.

$\begin{bmatrix} 3 \\ -3 \\ \vdots \\ 2.4 \\ 0.9 \end{bmatrix}$ → Generator → 

Longer hair

$\begin{bmatrix} 0.1 \\ 2.1 \\ \vdots \\ 5.4 \\ 0.9 \end{bmatrix}$ → Generator → 

blue hair

$\begin{bmatrix} 0.1 \\ -3 \\ \vdots \\ 2.4 \\ 3.5 \end{bmatrix}$ → Generator → 

Open mouth

Basic Idea of GAN

It is a neural network (NN), or a function.

image → Discri-minator → scalar

Larger value means real, smaller value means fake.

 → Discri-minator → 1.0

 → Discri-minator → 1.0

 → Discri-minator → 0.1

 → Discri-minator → 0.1

Basic Idea of GAN

Brown

veins

Generator

Butterflies are not brown

Butterflies do not have veins

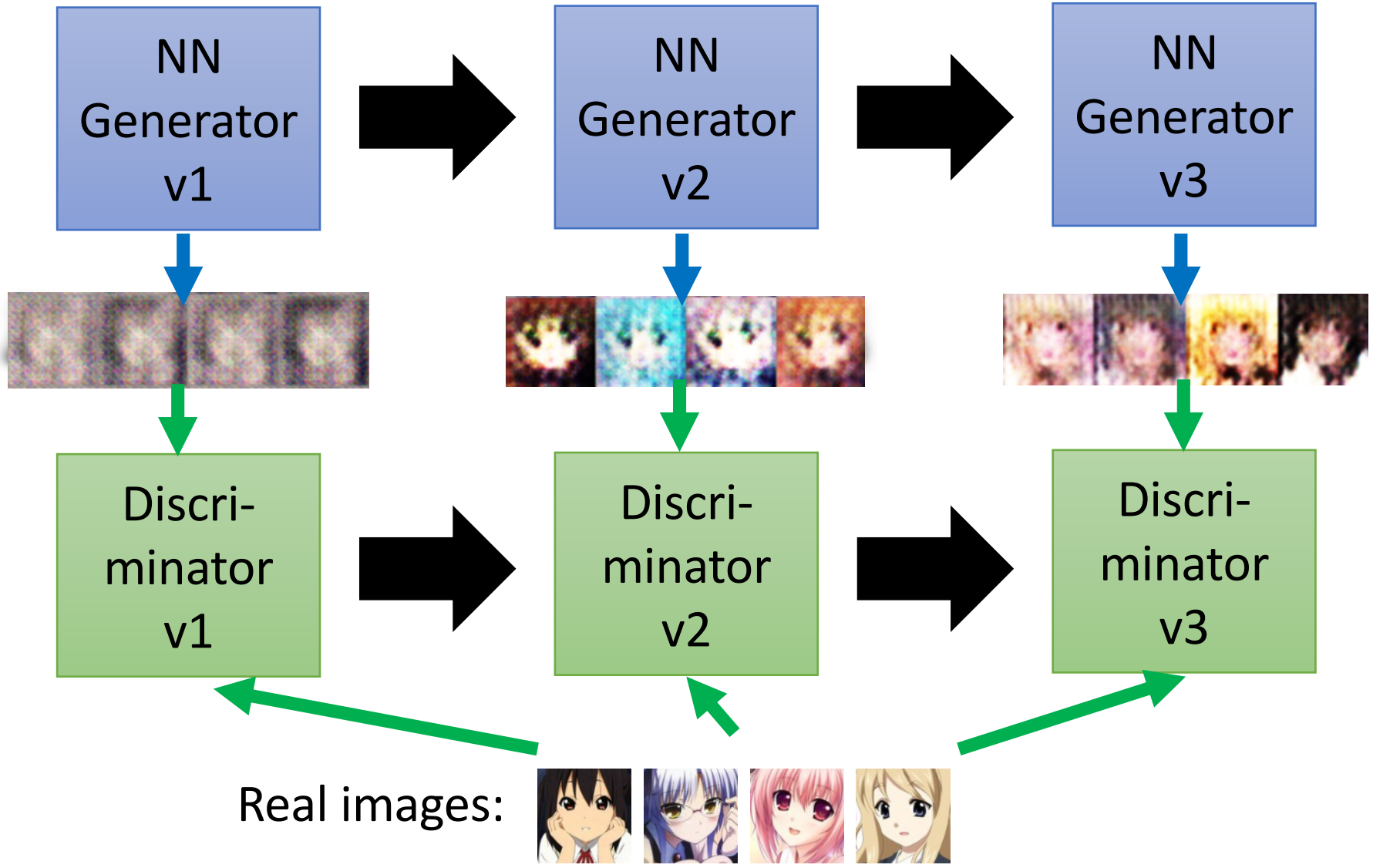
.....

Discriminator

Basic Idea of GAN

This is where the term "***adversarial***" comes from.

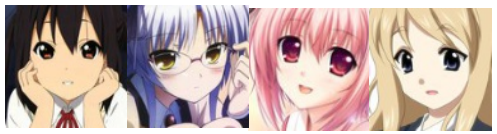
You can explain the process in different ways…….



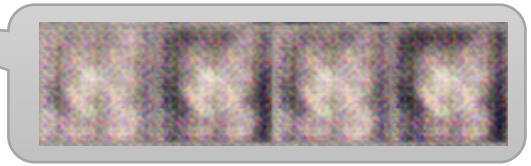
Basic Idea of GAN

Generator (student)

Discriminator (teacher)



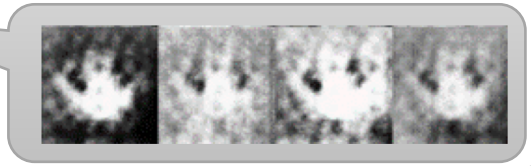
Generator v1



Discriminator v1

Two eyes

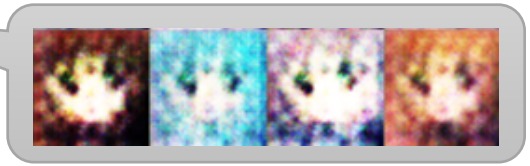
Generator v2



Discriminator v2

Back-&-White

Generator v3



Only use G ?

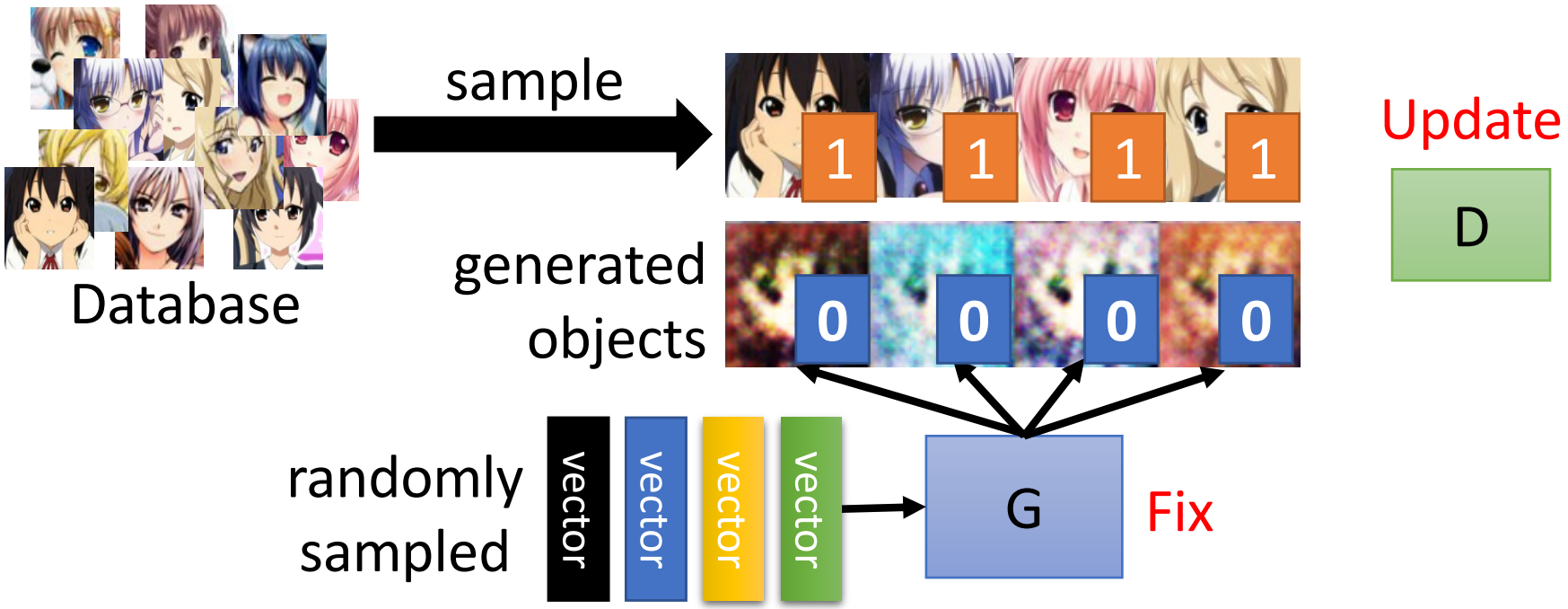
Only Use D ?

Algorithm

- Initialize generator and discriminator

- In each training iteration:

Step 1: Fix generator G, and update discriminator D



Discriminator learns to assign high scores to real objects and low scores to generated objects.

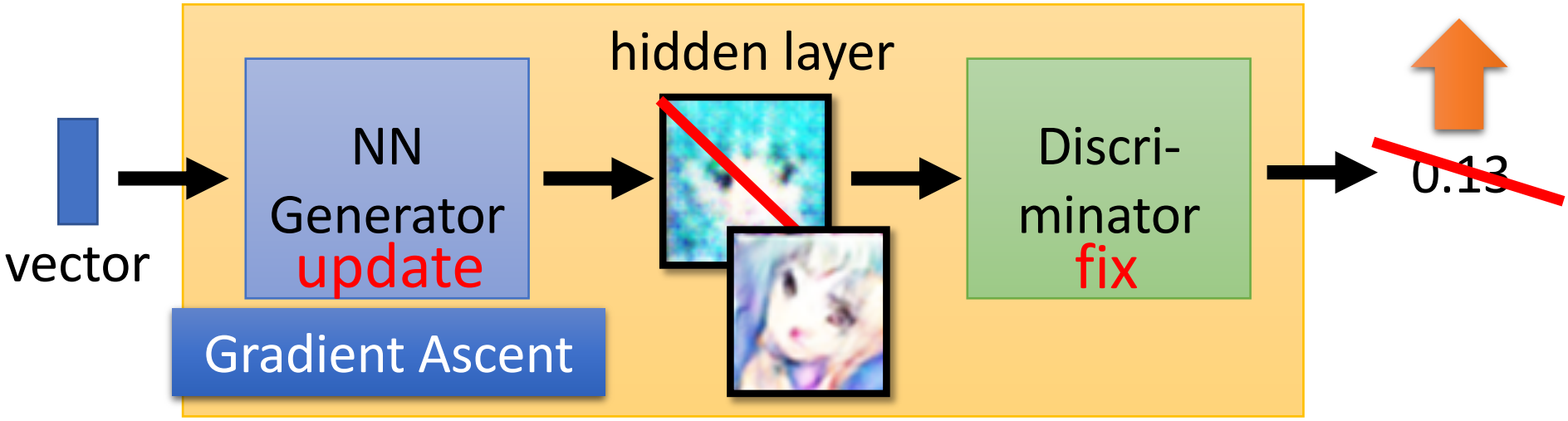
Algorithm

- Initialize generator and discriminator

- In each training iteration:

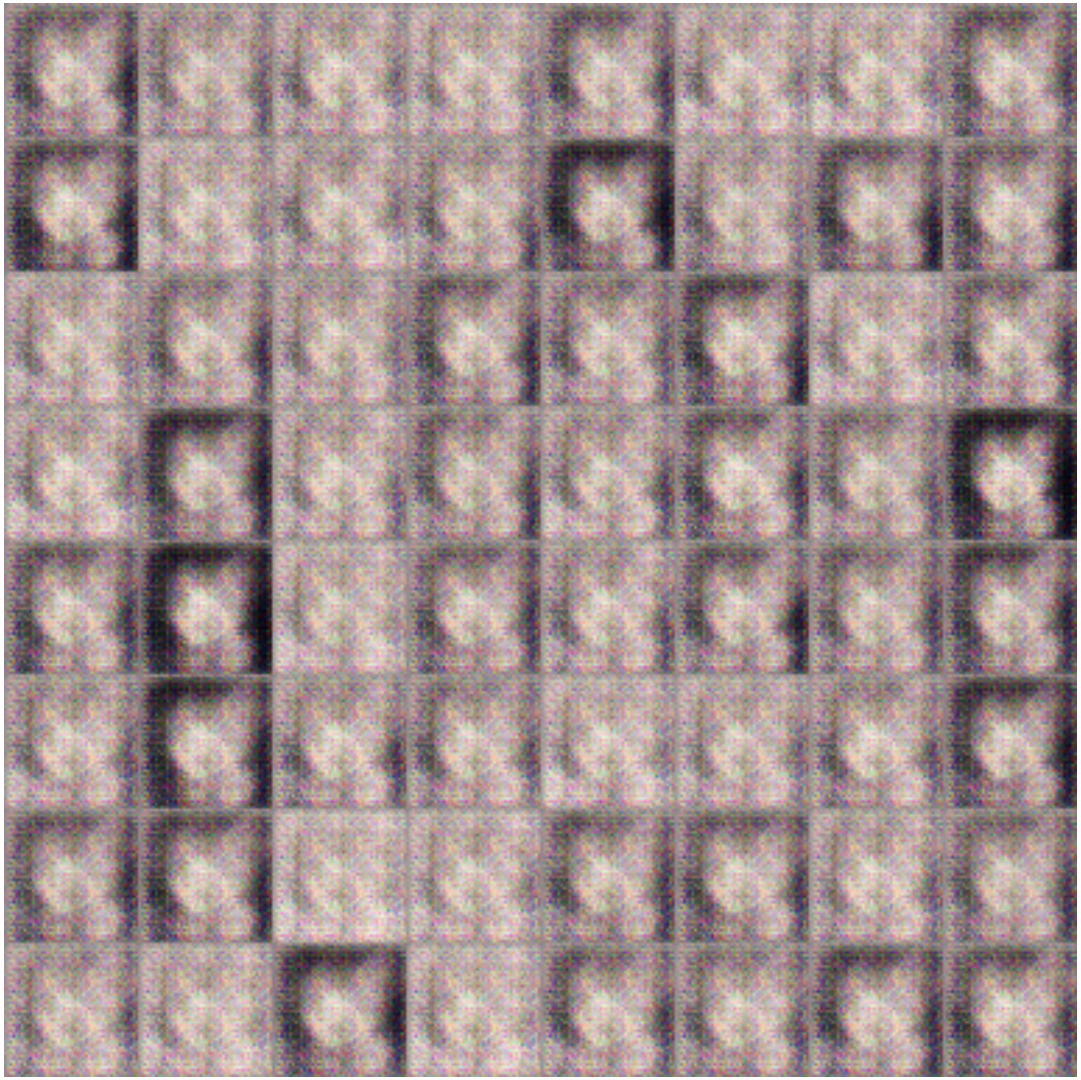
Step 2: Fix discriminator D, and update generator G

Generator learns to “fool” the discriminator



large network

Anime Face Generation



100 updates

Anime Face Generation

1000 updates

Anime Face Generation

2000 updates

Anime Face Generation

5000 updates

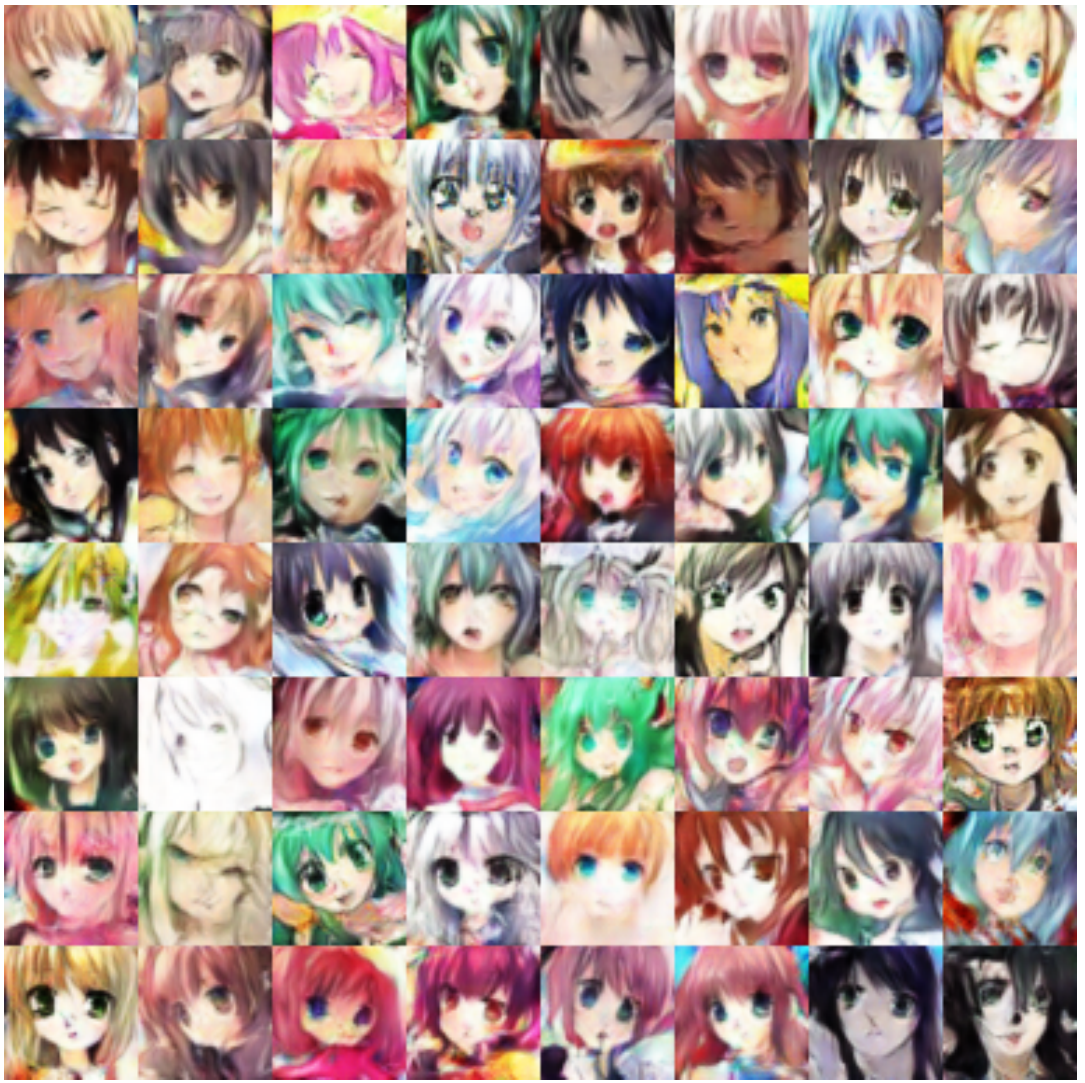
Anime Face Generation

10,000 updates

Anime Face Generation

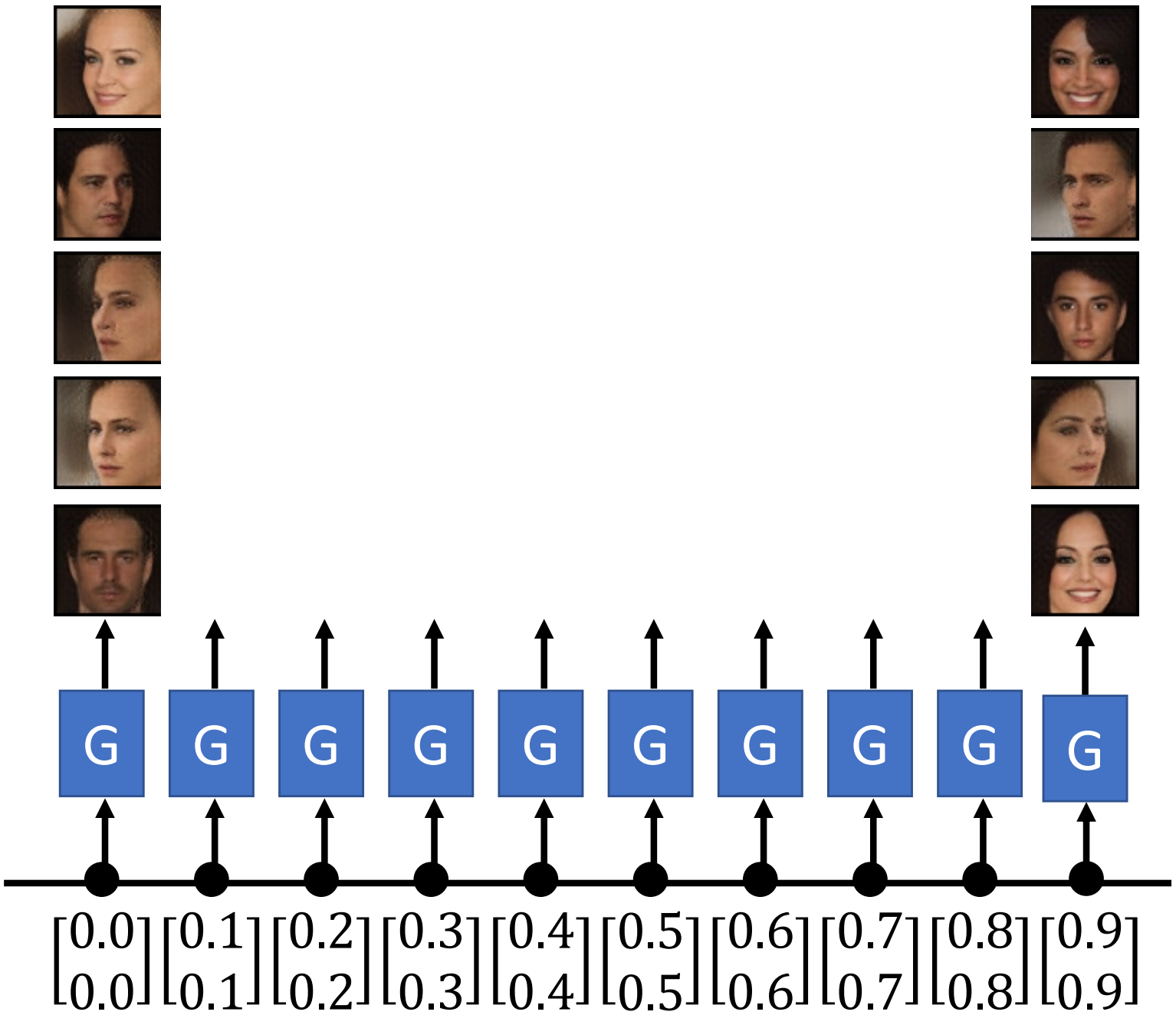
20,000 updates

Anime Face Generation



50,000 updates

The faces generated by machine.

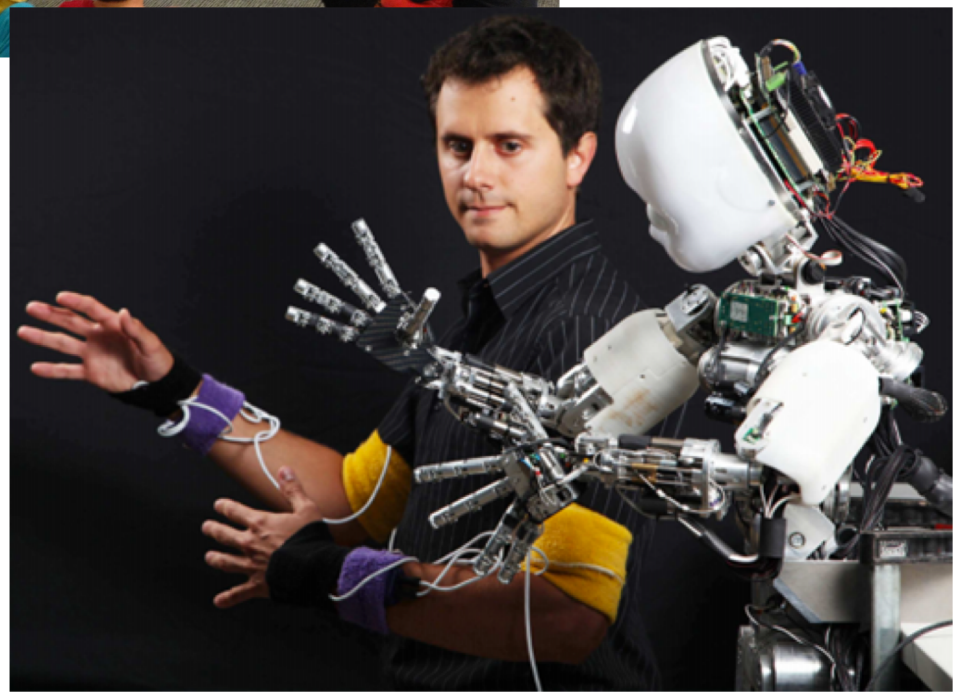

G G G G G G G G G G
[0.0 0.0] [0.1 0.1] [0.2 0.2] [0.3 0.3] [0.4 0.4] [0.5 0.5] [0.6 0.6] [0.7 0.7] [0.8 0.8] [0.9 0.9]

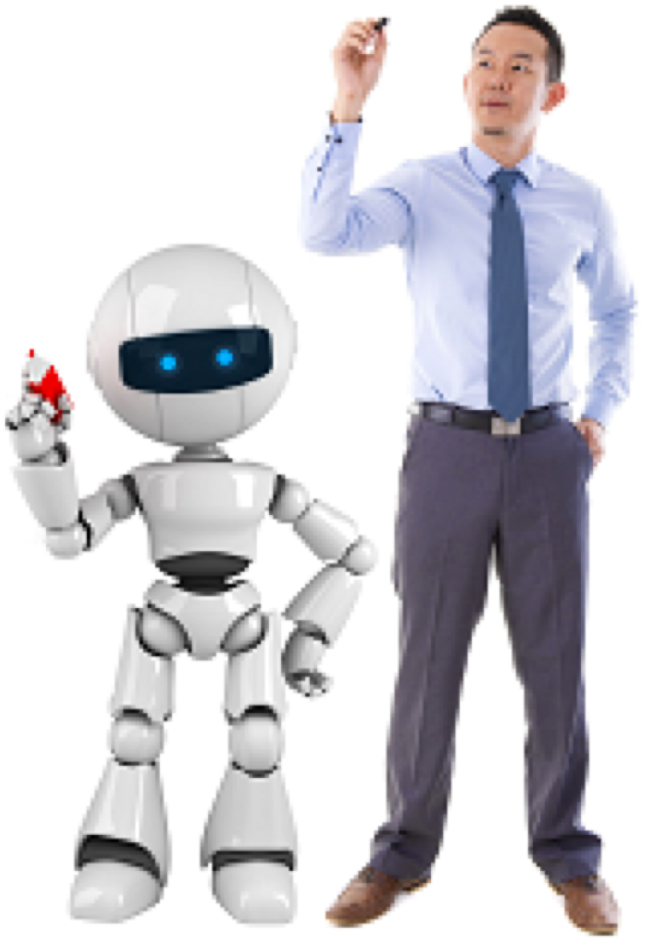
This Lecture

- Review Generative Adversarial Networks (GANs)
- IRL as GAN
  - Generative Adversarial Imitation Learning (GAIL)
- Multi-Agent IRL
  - MA-GAIL
- Meta-Learning
  - Meta-Reinforcement Learning
- Class Review

Inverse reinforcement learning

- Also called Learning from Demonstration (LfD)

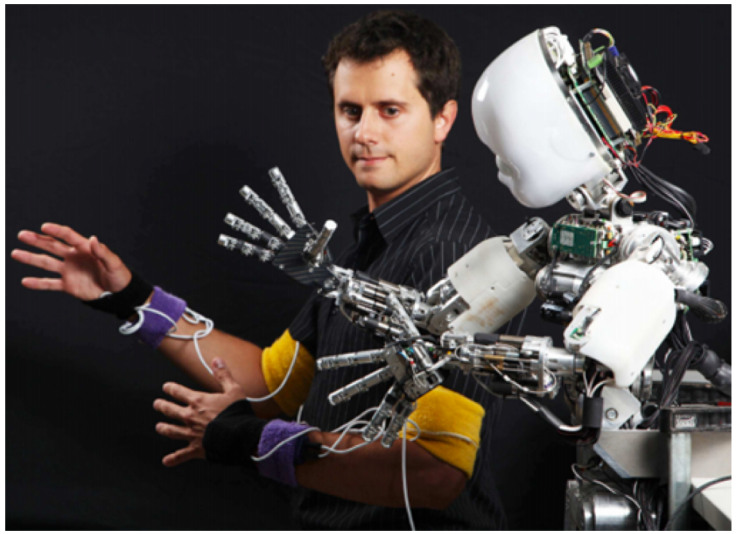




Inverse reinforcement learning

- Also called Learning from Demonstration (LfD)

- Input:
  - Trajectory set from experts $\tau=(s_1,a_1,\ldots,s_n,a_n)$
  - *State space: S*
  - *Action space: A*

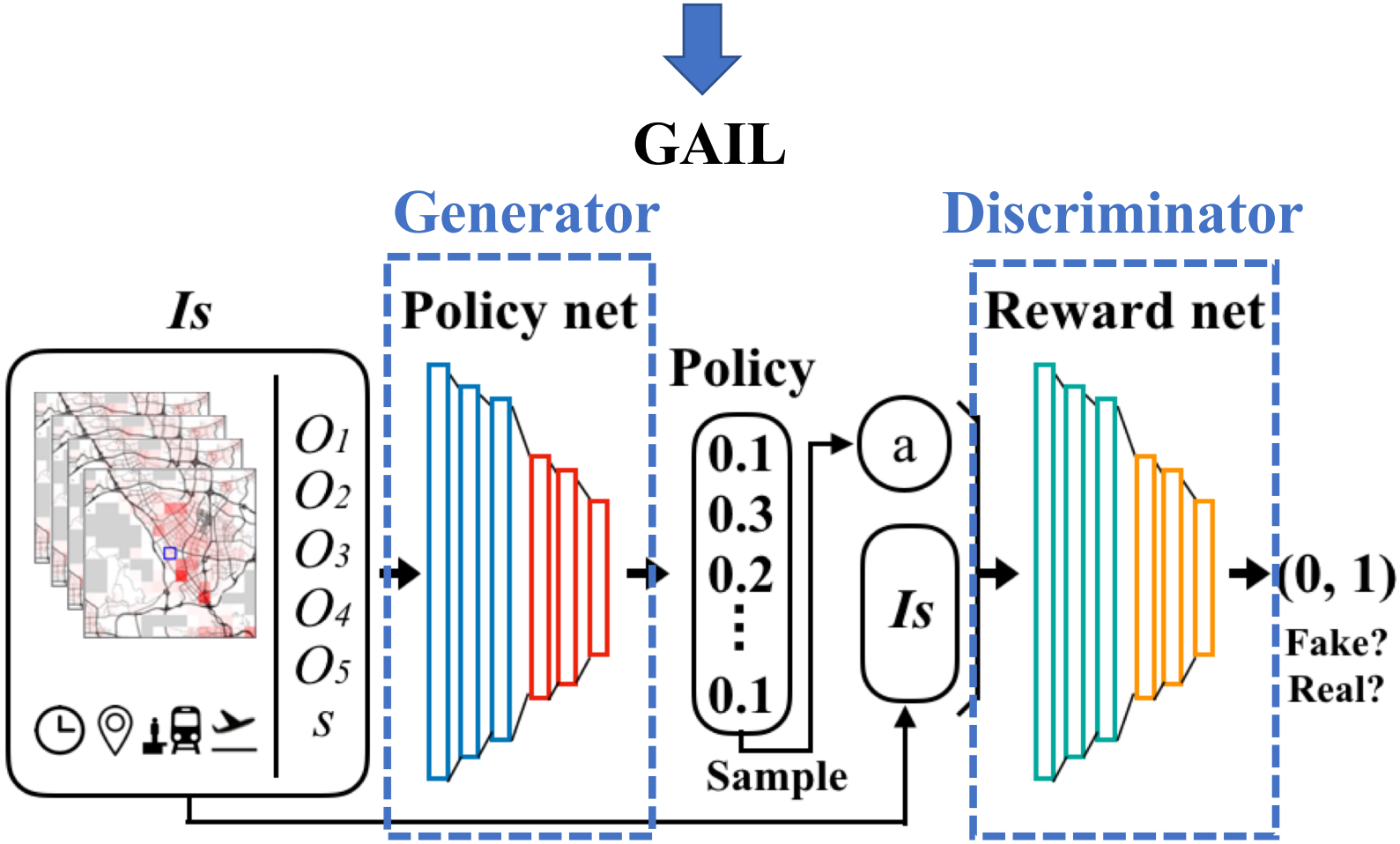


- *Output:*
  - *Policy: R(s,a)*
  - *Reward: π(s,a)*

- *Model-free problem: Unknown P* and γ.

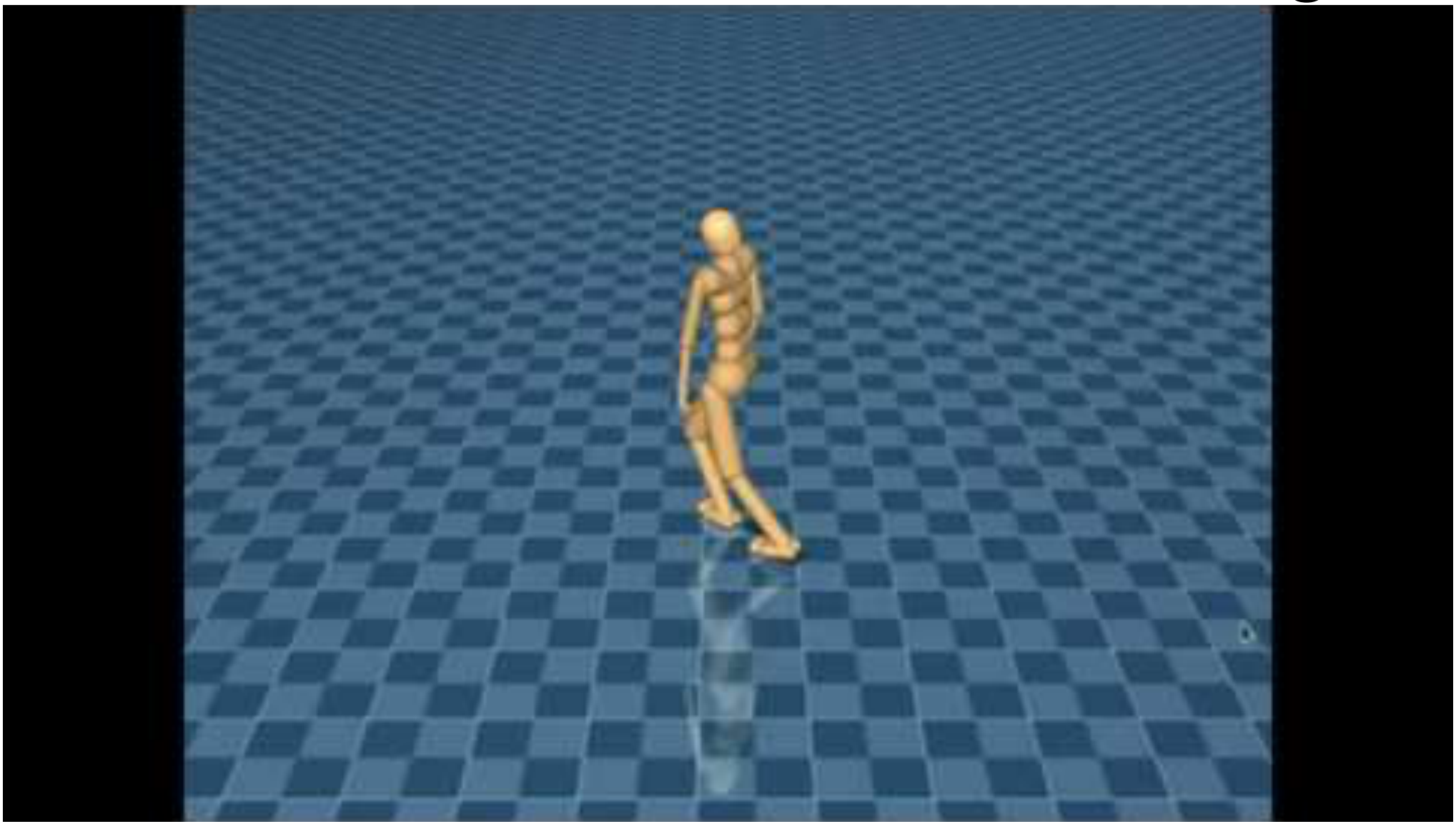
GAIL: Generative Adversarial Imitation Learning

Generative Adversarial Nets (**GAN**) + Imitation Learning (**IL**)



https://arxiv.org/abs/1606.03476

Inverse reinforcement learning



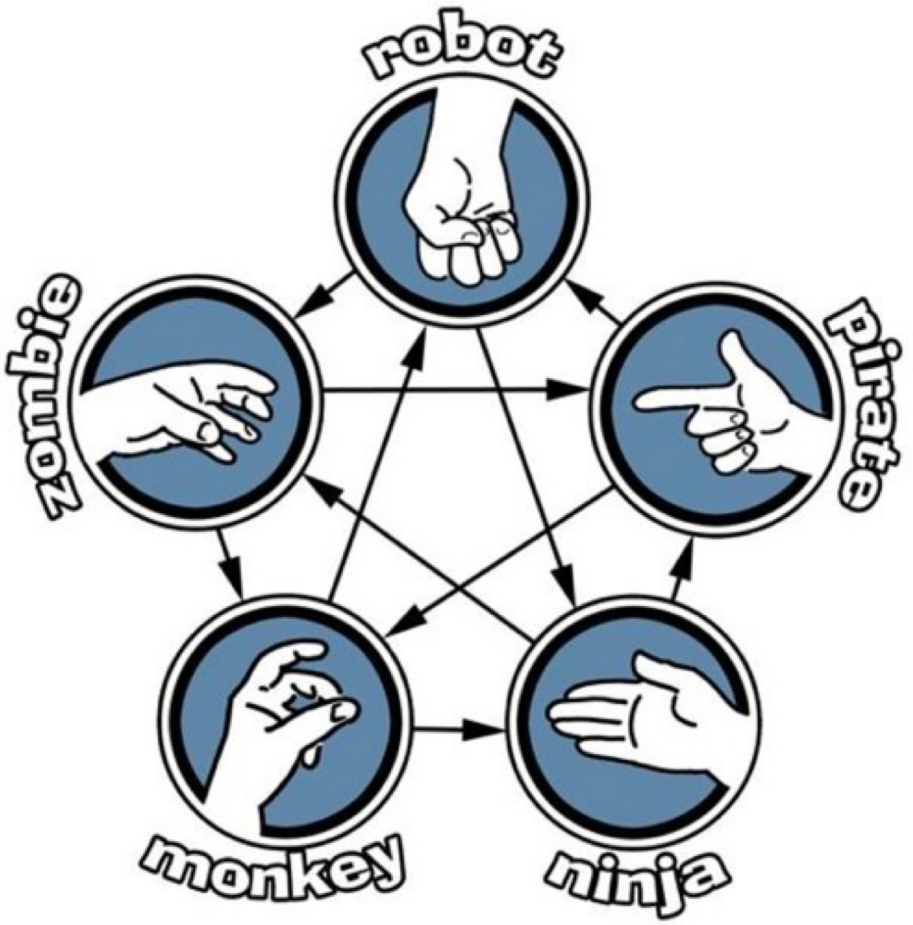
https://youtu.be/YsxN3uRBupc

This Lecture

- Review Generative Adversarial Networks (GANs)
- IRL as GAN
  - Generative Adversarial Imitation Learning (GAIL)
- Multi-Agent IRL
  - MA-GAIL
- Meta-Learning
  - Meta-Reinforcement Learning
- Class Review

MA-GAIL: Multi-Agent Generative Adversarial Imitation Learning

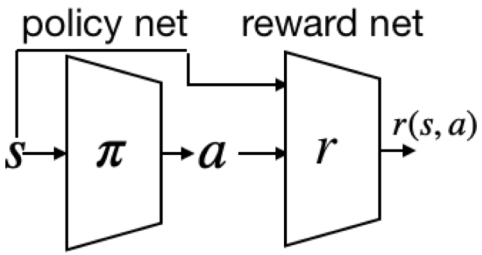
Rock-paper-scissors



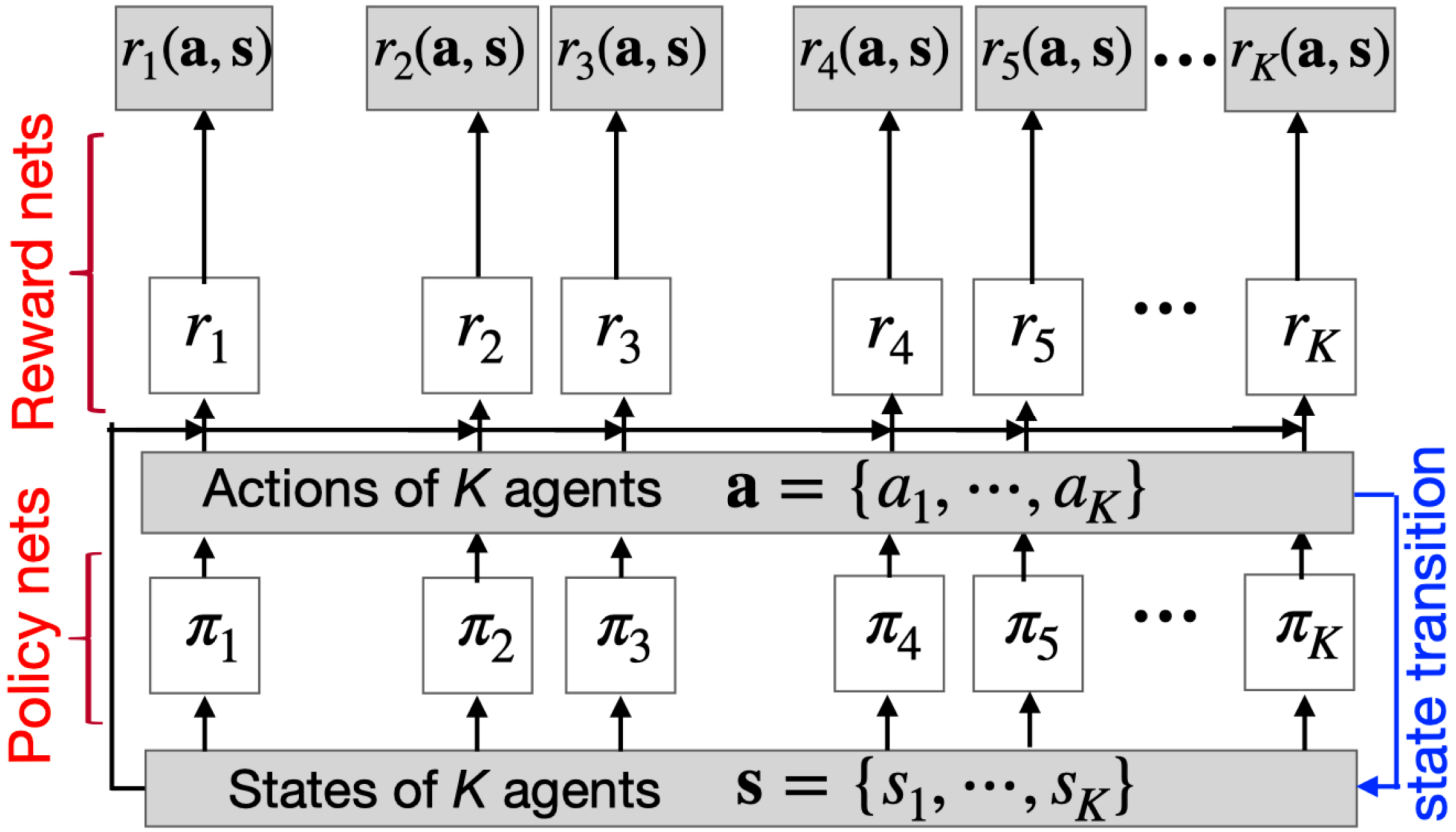
Monkey-Pirate-Robot-Ninja-Zombie

MA-GAIL: Multi-Agent Generative Adversarial Imitation Learning

GAIL



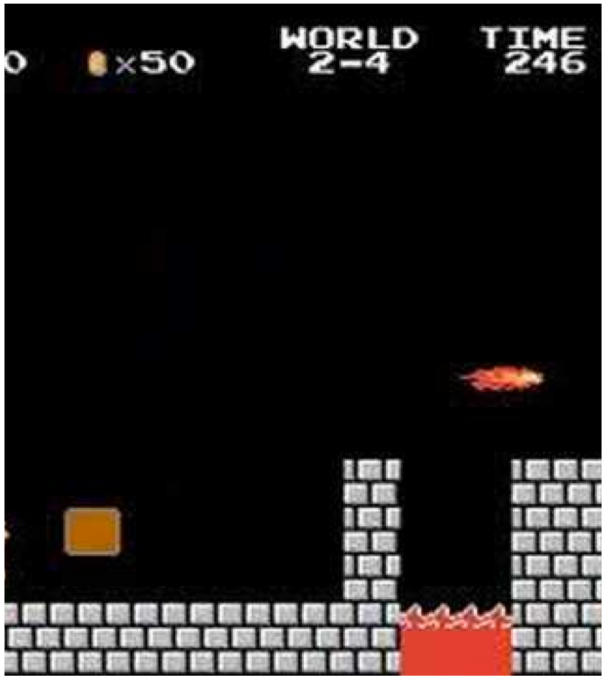
MA-GAIL



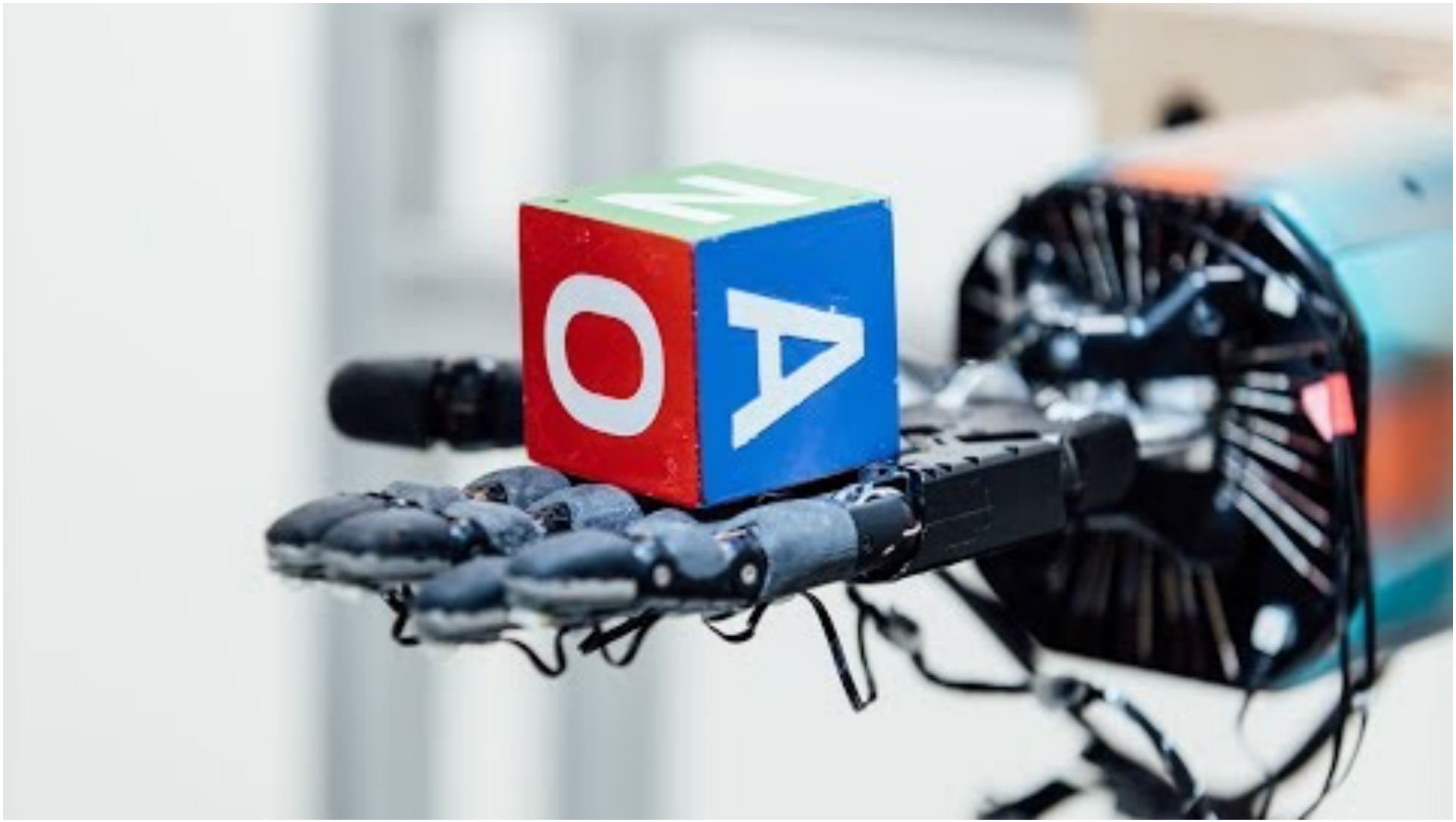
https://arxiv.org/abs/1807.09936

This Lecture

- Review Generative Adversarial Networks (GANs)
- IRL as GAN
  - Generative Adversarial Imitation Learning (GAIL)
- Multi-Agent IRL
  - MA-GAIL
- Meta-Learning
  - Meta-Reinforcement Learning
- Class Review



Multiple (Similar) Tasks



https://youtu.be/jwSbzNHGflM

Introduction

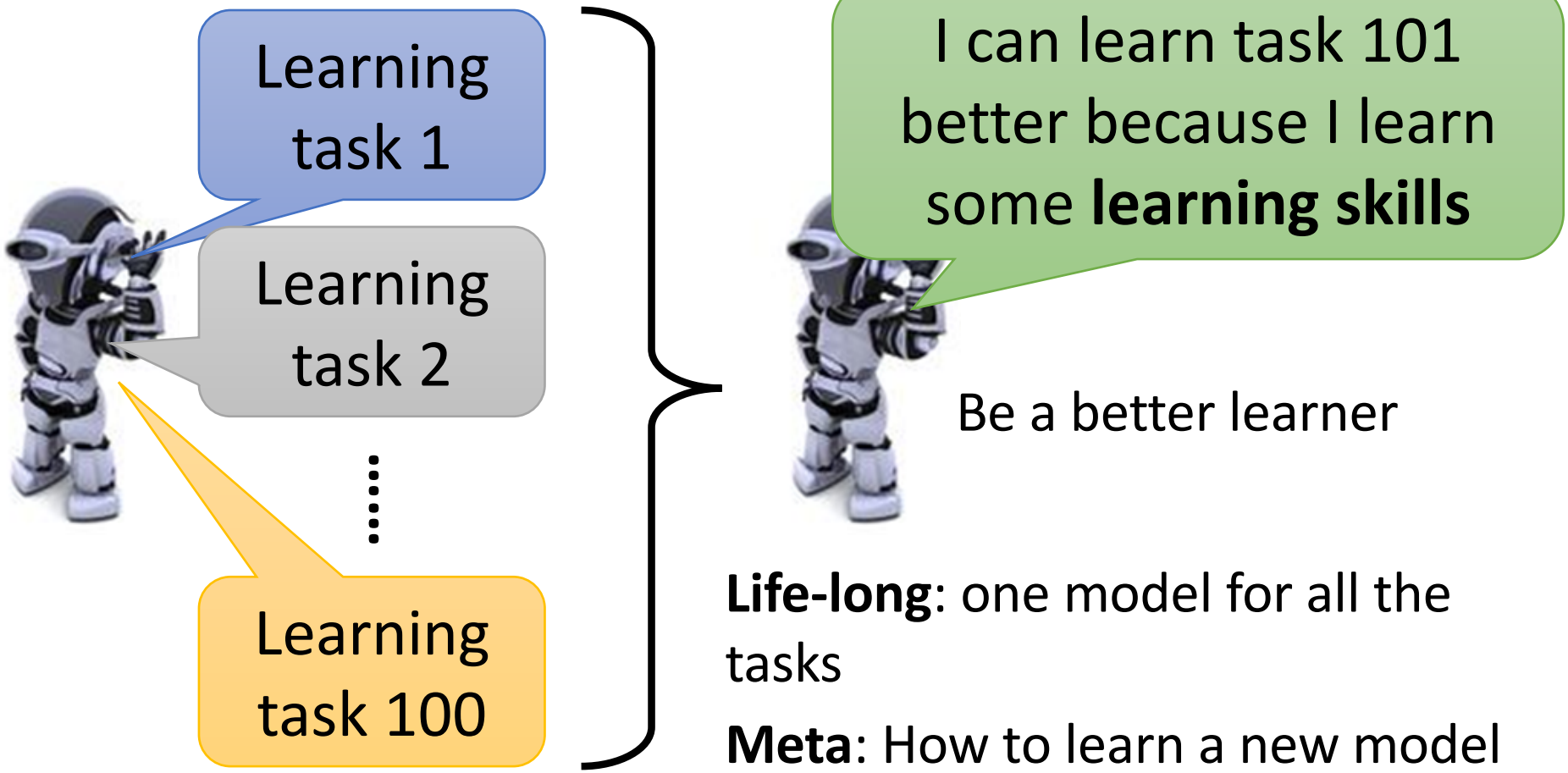
Task 1: speech recognition

Task 2: image recognition

⋮

Task 100: text classification

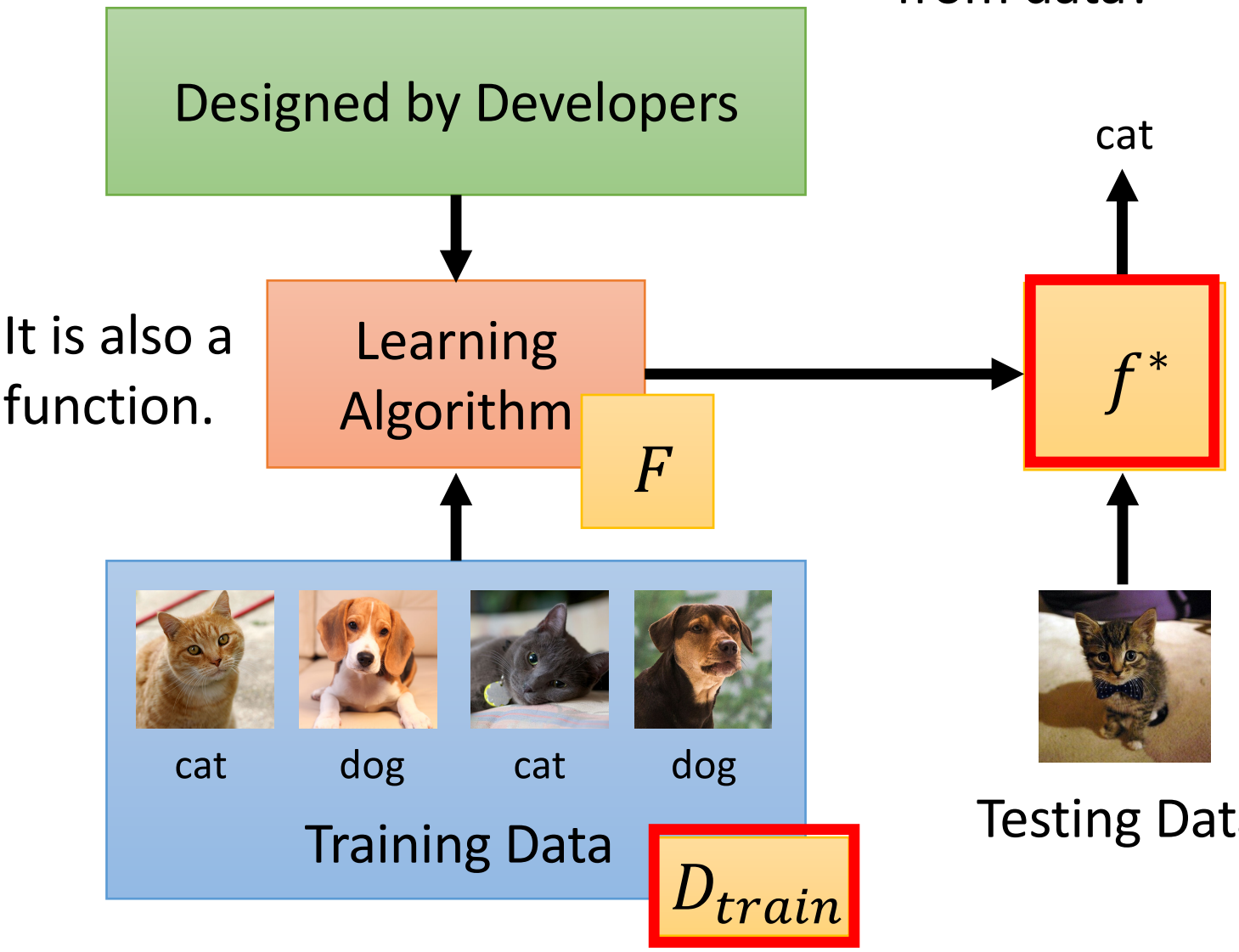
- Meta learning = Learn to learn



Meta Learning

$$f^* = F(D_{train})$$

Can machine find F from data?



Meta Learning

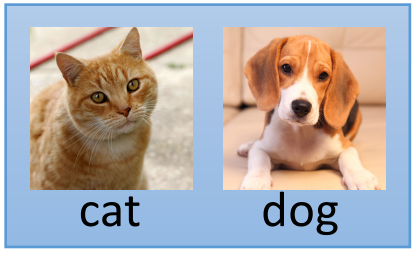
Machine Learning ≈ from data to model $f^*(.)$

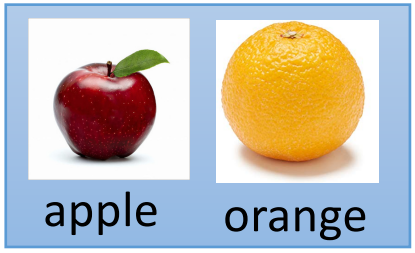
$F($ 

$) = f^*$ $f^*($  $) =$ "Cat"

Meta Learning

≈ from data of various tasks to F(.) that outcomes *f*(.)

$F_{meta}($ 



$) = F^*$ $F^*($ 

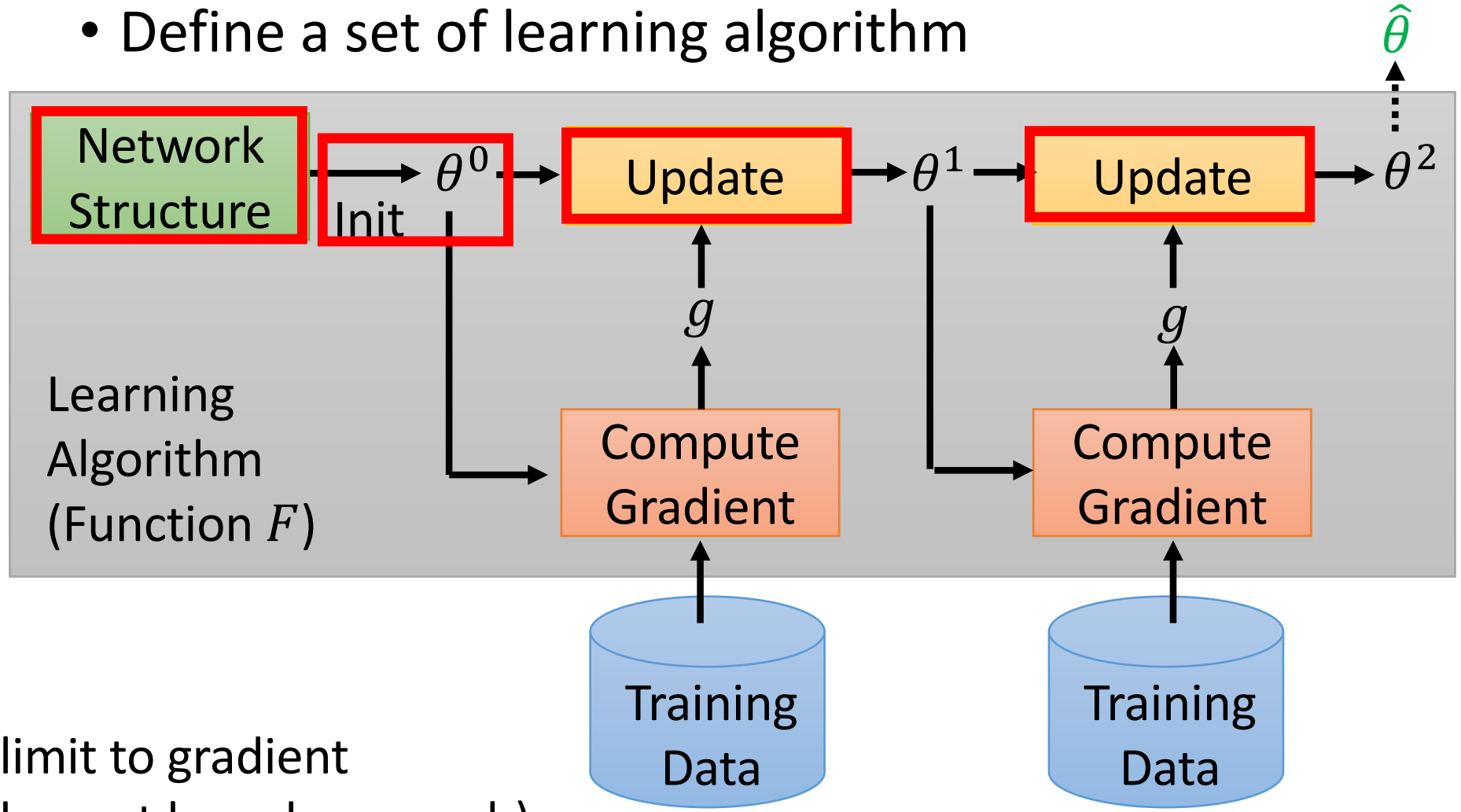
$) = f^*$

...

Meta Learning

Different decisions in the red boxes lead to different algorithms. What happens in the red boxes is decided by humans until now.

- Define a set of learning algorithm



(limit to gradient descent based approach)

More on meta-learning? Take DS504/CS586☺.

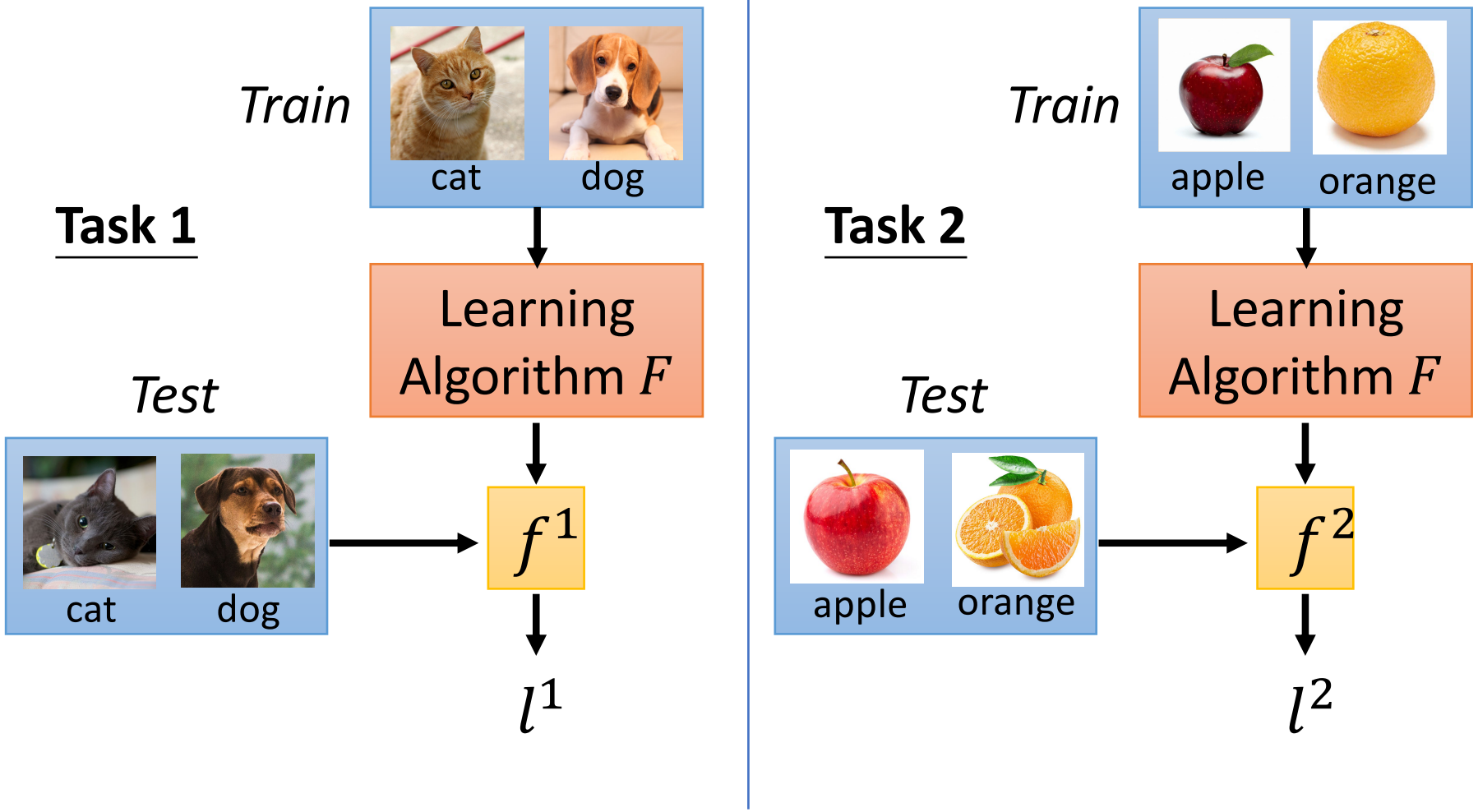
Meta Learning

$$L(F) = \sum_{n=1}^{N} l^n$$

N tasks

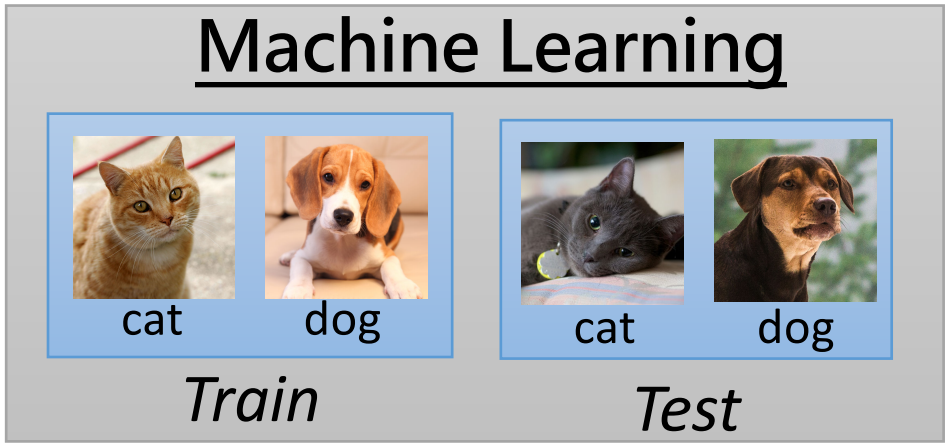
Testing loss for task n after training

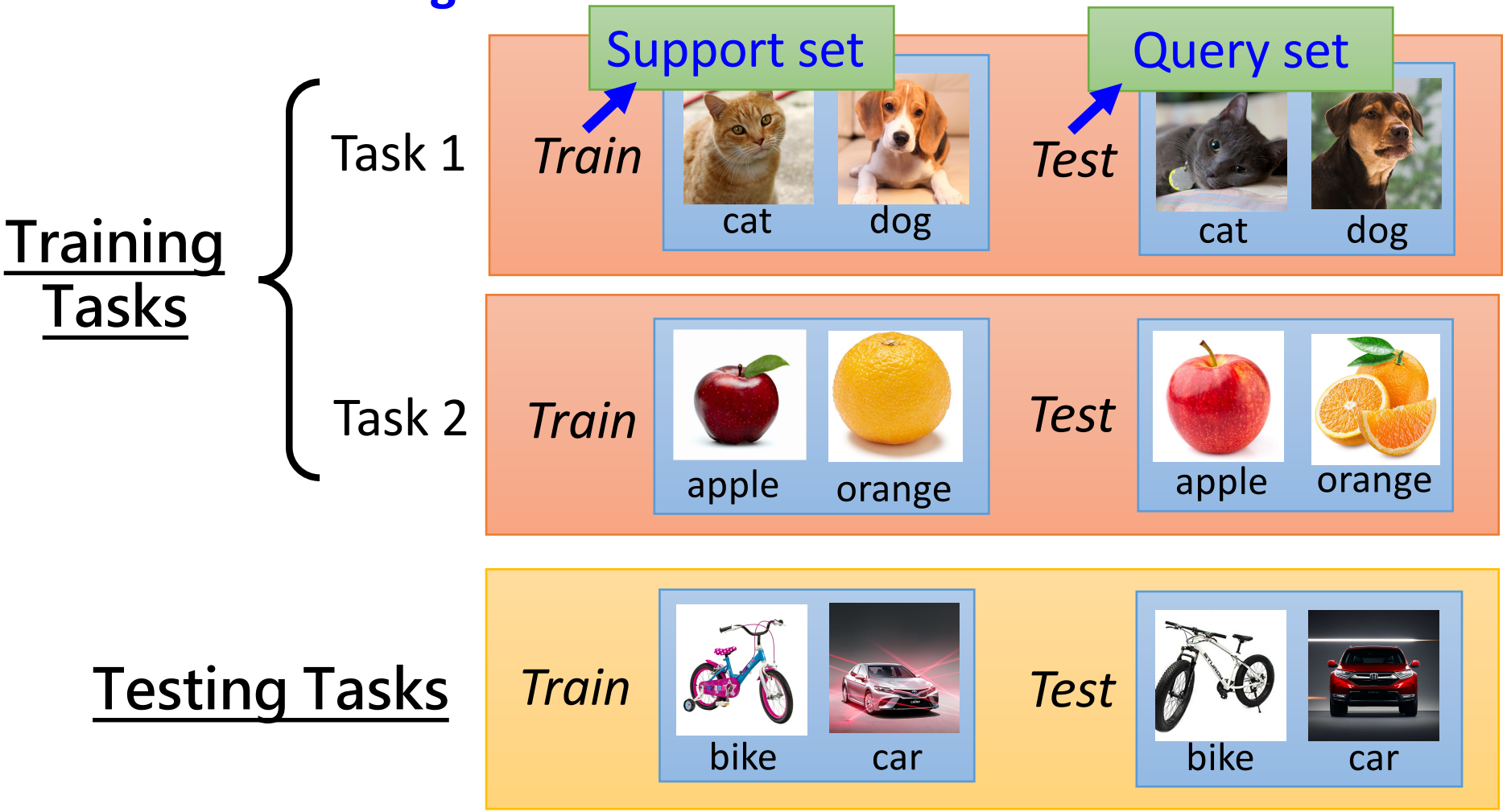
- Defining the goodness of a function F



Meta Learning

Widely considered in
few-shot learning





Meta Learning

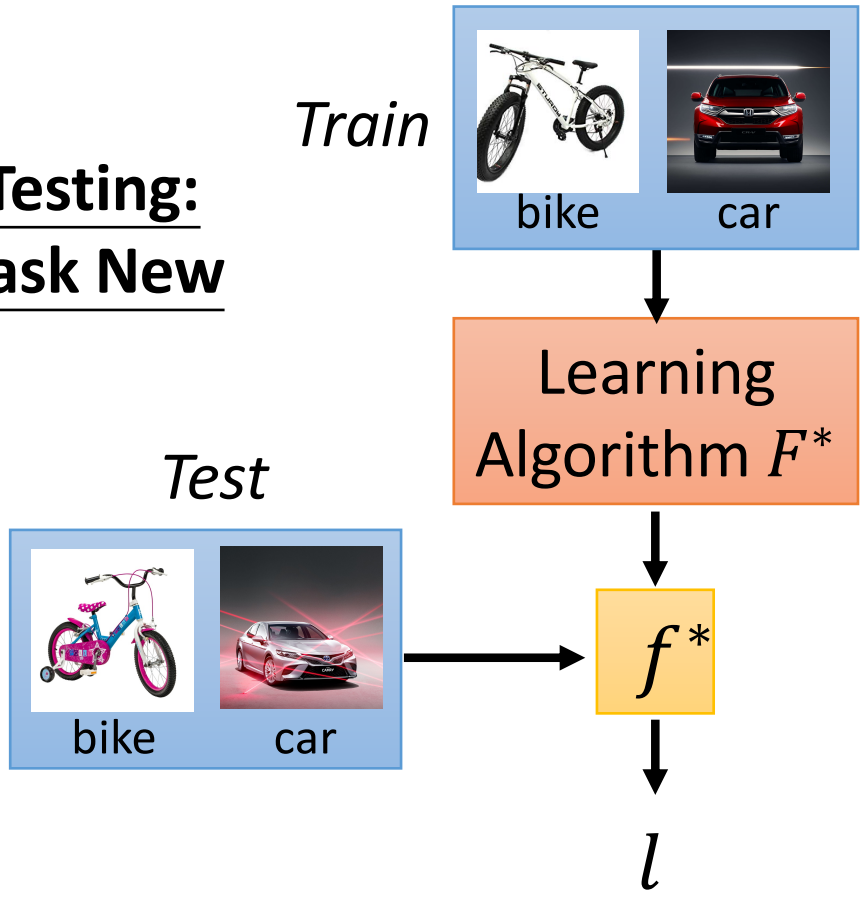
- Defining the goodness of a function F

$$L(F) = \sum_{n=1}^{N} l^n$$

- Find the best function F^*

$$F^* = arg \min_F L(F)$$

Testing:
Task New



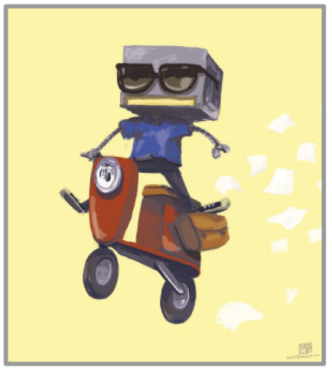
Meta-learning problem statement

supervised learning

"Dalmation" "German shepherd" "Pug"

corgi ???

reinforcement learning



Robot art by Matt Spangler, *mattspangler.com*

Source: http://rail.eecs.berkeley.edu/deeprlcourse/static/slides/lec-20.pdf

Meta-RL problem statement

Regular RL: learn policy for single task

$$\theta^\star = \arg\max_\theta E_{\pi_\theta(\tau)}[R(\tau)]$$

$$= f_{\text{RL}}(\mathcal{M})$$

MDP

Meta-RL: learn adaptation rule

$$\theta^\star = \arg\max_\theta \sum_{i=1}^{n} E_{\pi_{\phi_i}(\tau)}[R(\tau)]$$

Meta-training / Outer loop

where $\phi_i = f_\theta(\mathcal{M}_i)$

Adaptation / Inner loop

MDP for task i

$\mathcal{M}_1$ $\mathcal{M}_2$ $\mathcal{M}_3$ $\mathcal{M}_{test}$

Source: http://rail.eecs.berkeley.edu/deeprlcourse/static/slides/lec-20.pdf

Meta-RL (with Policy Gradient) (One of many Meta-RL paradigms)

while training:
  for i in tasks:
    1. sample k episodes $\mathcal{D}_i = \{(s, a, s', r)\}_{1:k}$ from π_θ
    2. compute adapted parameters $\phi_i = \theta - \alpha \nabla_\theta \mathcal{L}_i(\pi_\theta, \mathcal{D}_i)$
    3. sample k episodes $\mathcal{D}'_i = \{(s, a, s', r)_{1:k}\}$ from π_ϕ
  update policy parameters $\theta \leftarrow \theta - \nabla_\theta \sum_i \mathcal{L}_i(\mathcal{D}'_i, \pi_{\phi_i})$

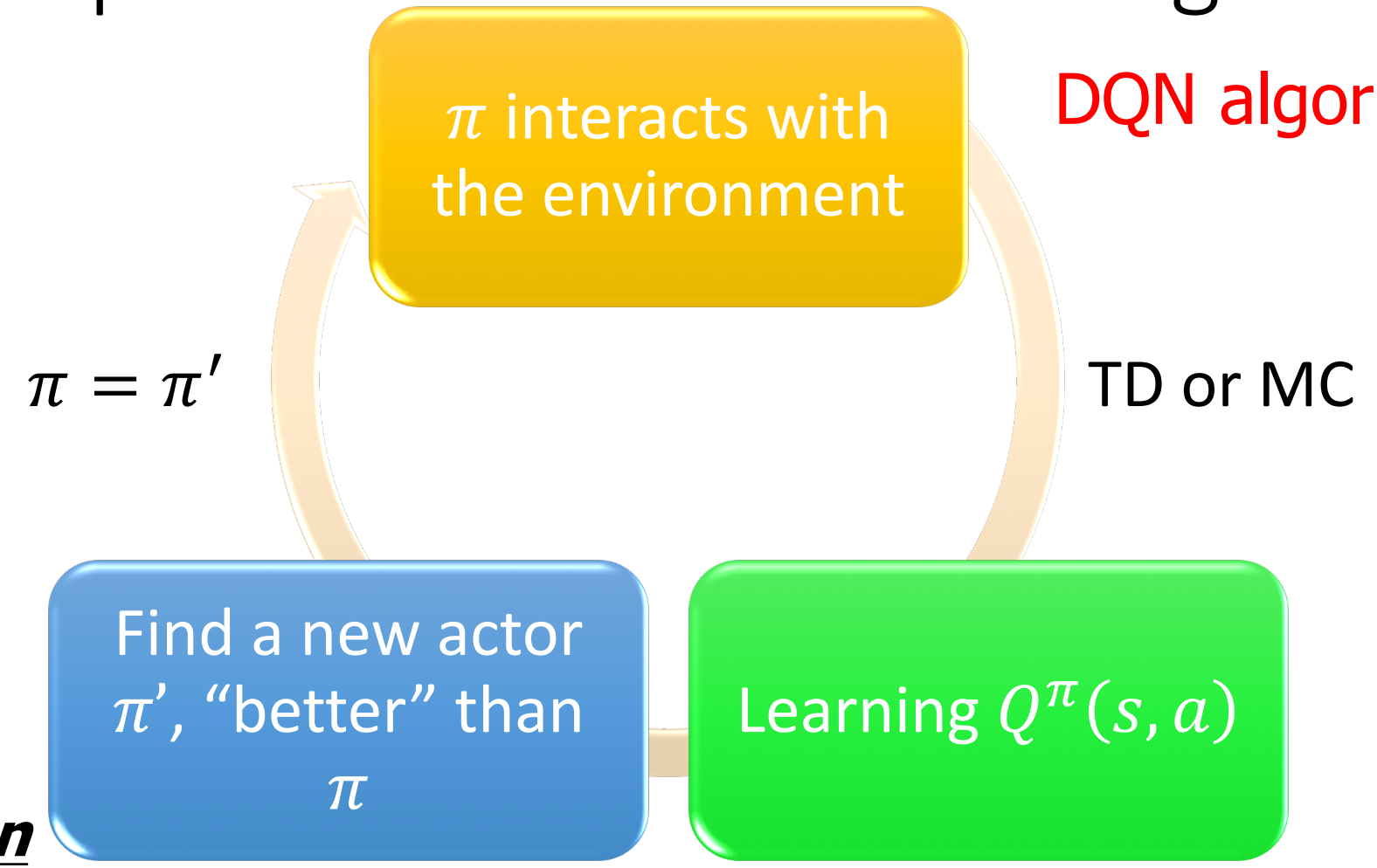
Finn et al. 2017. Fig adapted from Finn et al. 2017

This Lecture

- Review Generative Adversarial Networks (GANs)
- IRL as GAN
  - Generative Adversarial Imitation Learning (GAIL)
- Multi-Agent IRL
  - MA-GAIL
- Meta-Learning
  - Meta-Reinforcement Learning
- Class Review

	Reinforcement Learning	Inverse Reinforcement Learning
Single Agent	**Tabular representation of reward** *Model-based control* *Model-free control* *(MC, SARSA, Q-Learning)*	**Linear reward function learning** Imitation learning Apprenticeship learning Inverse reinforcement learning
	Function representation of reward *1. Linear value function approx (MC, SARSA, Q-Learning)* *2. Value function approximation (Deep Q-Learning, Double DQN, prioritized DQN, Dueling DQN)* *3. Policy function approximation (Policy gradient, PPO, TRPO)* *4. Actor-Critic methods (A2C, A3C, Pathwise Derivative PG)* *Advanced topics in RL (Sparse Rewards)*	**Non-linear reward function learning** Generative adversarial imitation learning (GAIL) Adversarial inverse reinforcement learning (AIRL) https://arxiv.org/abs/1710.11248
	Review of Deep Learning *As bases for non-linear function approximation (used in 2-4).*	Review of Generative Adversarial nets As bases for non-linear IRL
Multiple Agents	**Multi-Agent Reinforcement Learning** Multi-agent Actor-Critic etc. **Meta-RL**	**Multi-Agent Inverse Reinforcement Learning** MA-GAIL

Deep Reinforcement Learning



Epsilon Greedy

$$a = \begin{cases} arg\max_{a} Q(s,a), & with\ probability\ 1-\varepsilon \\ random, & otherwise \end{cases}$$

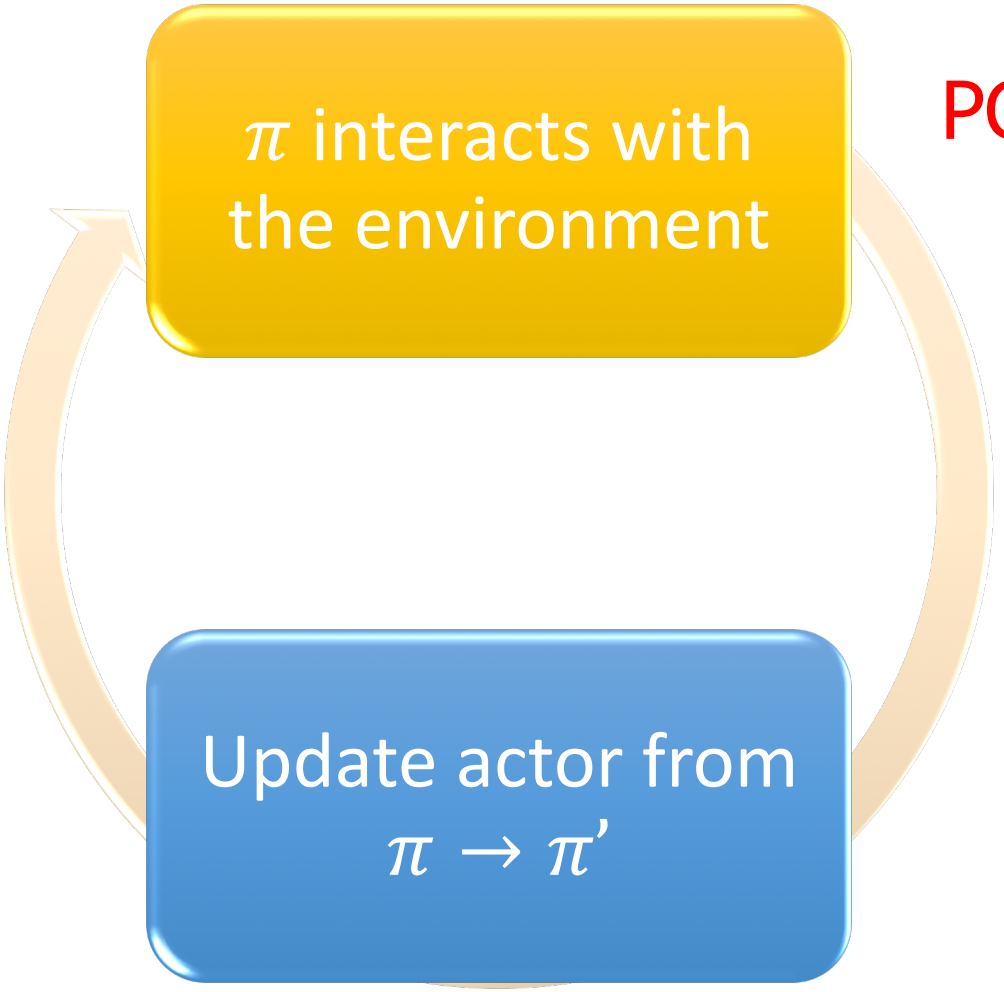
DQN

DQN

- DQN
- Double DQN
- Dueling DQN
- Prioritized DQN
- Multi-Step DQN
- Noisy Net DQN
- Distributional DQN
- DQN with continuous action space
- Rainbow

Deep Reinforcement Learning

PG algorithms



$\pi = \pi'$

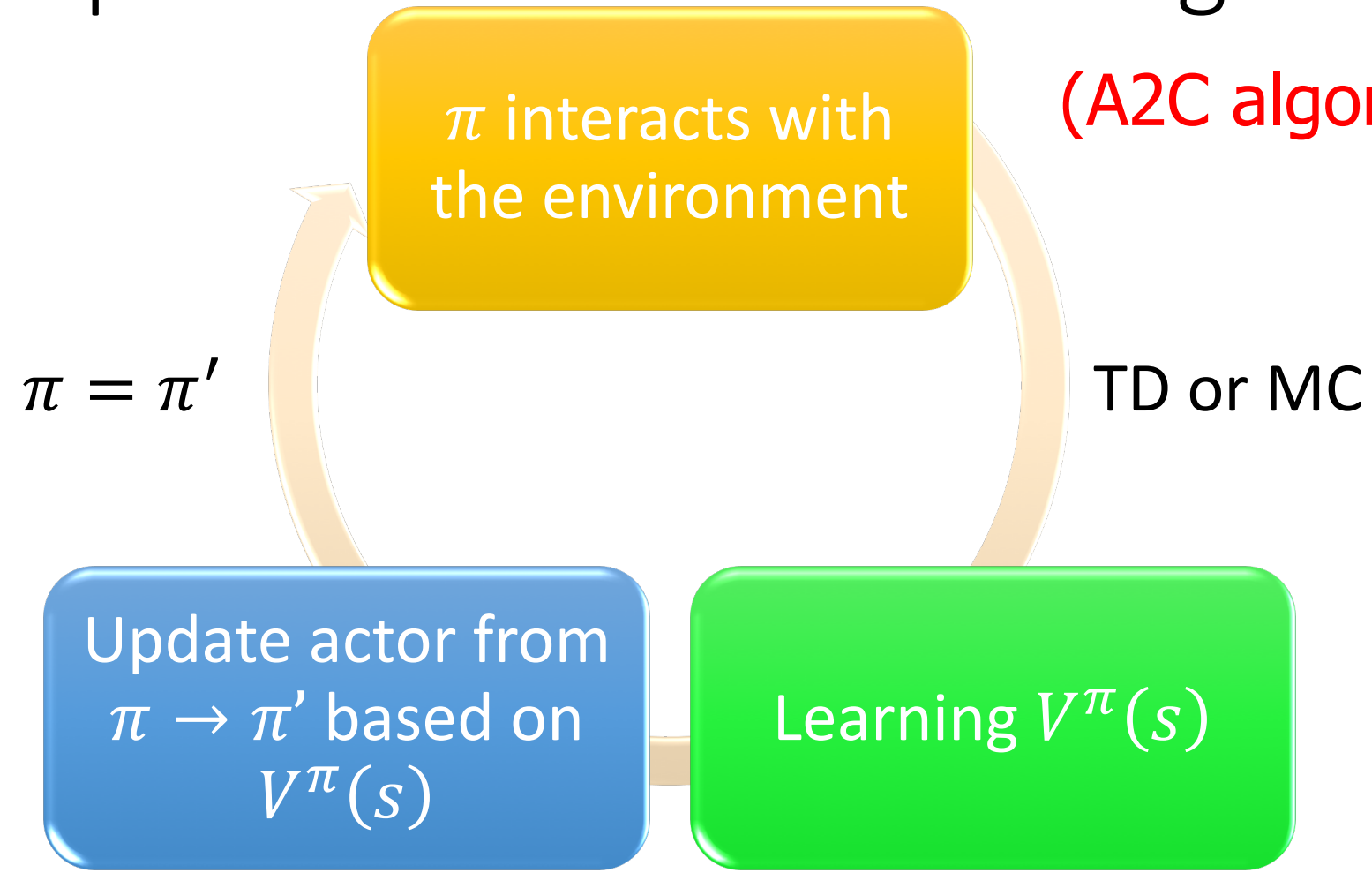
$$\nabla \bar{R}_\theta \approx \frac{1}{N} \sum_{n=1}^{N} \sum_{t=1}^{T_n} \left(\sum_{t'=t}^{T_n} \gamma^{t'-t} r_{t'}^n - b \right) \nabla log p_\theta(a_t^n | s_t^n)$$

Policy Gradient

Policy Gradient

- Basic PG
- Vanilla
- REINFORCE
- PPO
- TRPO
- PPO2

Deep Reinforcement Learning



$$\nabla \bar{R}_{\theta} \approx \frac{1}{N} \sum_{n=1}^{N} \sum_{t=1}^{T_n} \left(r_t^n + V^{\pi}(s_{t+1}^n) - V^{\pi}(s_t^n) \right) \nabla log p_{\theta}(a_t^n | s_t^n)$$

Actor Critic

Actor Critic

- A2C
- A3C
- Pathwise Derivative Policy Gradient
- Deep Deterministic Policy Gradient

More topics

- Sparse reward
    - Reward shaping, Curiosity module
    - Curriculum learning, Hierarchical RL
- Meta-RL
- Multi-agent Reinforcement Learning
- Inverse Reinforcement Learning
    - Multi-agent IRL

	Reinforcement Learning	Inverse Reinforcement Learning
Single Agent	**Tabular representation of reward** *Model-based control* *Model-free control* *(MC, SARSA, Q-Learning)*	**Linear reward function learning** Imitation learning Apprenticeship learning Inverse reinforcement learning
	Function representation of reward *1. Linear value function approx (MC, SARSA, Q-Learning)* *2. Value function approximation (Deep Q-Learning, Double DQN, prioritized DQN, Dueling DQN)* *3. Policy function approximation (Policy gradient, PPO, TRPO)* *4. Actor-Critic methods (A2C, A3C, Pathwise Derivative PG)* *Advanced topics in RL (Sparse Rewards)*	**Non-linear reward function learning** Generative adversarial imitation learning (GAIL) Adversarial inverse reinforcement learning (AIRL) https://arxiv.org/abs/1710.11248
	Review of Deep Learning *As bases for non-linear function approximation (used in 2-4).*	Review of Generative Adversarial nets As bases for non-linear IRL
Multiple Agents	**Multi-Agent Reinforcement Learning** Multi-agent Actor-Critic etc. **Meta-RL**	**Multi-Agent Inverse Reinforcement Learning** MA-GAIL

Class review

- Please spend time to fill out the online class survey.

- Our goal is to offer this course as a permanent course from next fall.

More New and Advanced Learning techniques in DS504/CS586 in Spring 2023

- Explainable AI (XAI)
- Meta-learning
- Few-shot learning
- Adversarial attack to Deep neural networks
- Multi-Task Learning, Transfer Learning, Life-long Learning

I am lecturing
Big Data Analytics
in 2023 Spring
with these AI topics

Plan: CS586/DS504-2023Spring

5. Applications

Image Processing, High dimension Data (audio, text, trajectory) analysis

4. Big Data Mining

Deep Neural Networks, Generative Models, XAI, Attacks/Defense, Compression

3. Data Management

Indexing, Query Processing

2. Data Preprocessing/Cleaning

Error Correction, Map-Matching

1. Data Acquisition & Measurement

Representative data collection: Sampling and estimation **Project 1**

Techniques

Deep Neural Networks

1. DNN Recipe **Project 2**
2. Meta Learning **Project 4**

Generative Models

1. GANs **Project 3**
2. VAEs
3. Flow based models

More techniques

1. XAI
2. Adversarial attacks/defense
3. Network Compression

Questions?

Algorithm Initialize θ_d for D and θ_g for G

- In each training iteration:

Learning D

- Sample m examples $\{x^1, x^2, \dots, x^m\}$ from database
- Sample m noise samples $\{z^1, z^2, \dots, z^m\}$ from a distribution
- Obtaining generated data $\{\tilde{x}^1, \tilde{x}^2, \dots, \tilde{x}^m\}$, $\tilde{x}^i = G(z^i)$
- Update discriminator parameters θ_d to maximize
  - $\tilde{V} = \frac{1}{m}\sum_{i=1}^{m} log D(x^i) + \frac{1}{m}\sum_{i=1}^{m} log\left(1 - D(\tilde{x}^i)\right)$
  - $\theta_d \leftarrow \theta_d + \eta \nabla \tilde{V}(\theta_d)$

Learning G

- Sample m noise samples$\{z^1, z^2, \dots, z^m\}$ from a distribution
- Update generator parameters θ_g to maximize
  - $\tilde{V} = \frac{1}{m}\sum_{i=1}^{m} log\left(D\left(G(z^i)\right)\right)$
  - $\theta_g \leftarrow \theta_g - \eta \nabla \tilde{V}(\theta_g)$