Introduction to Reinforcement Learning

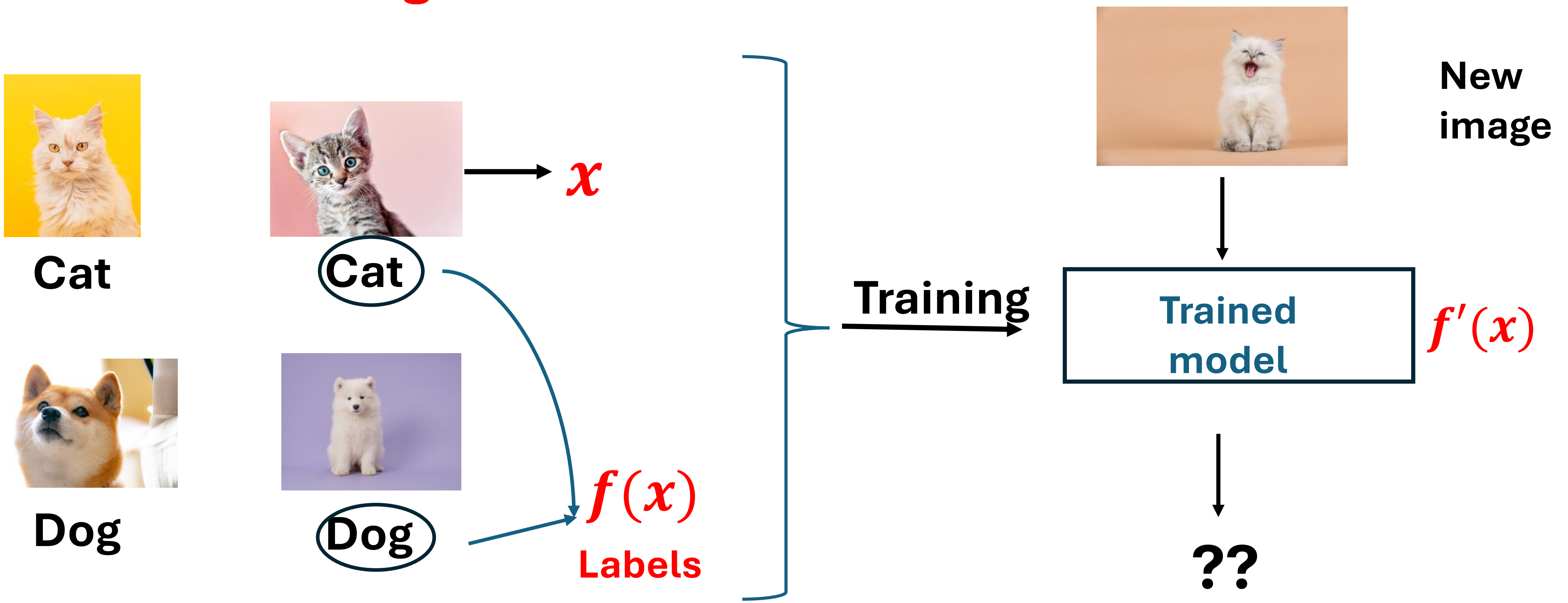
Subrahmanya Swamy Peruru

Paradigms of Machine Learning

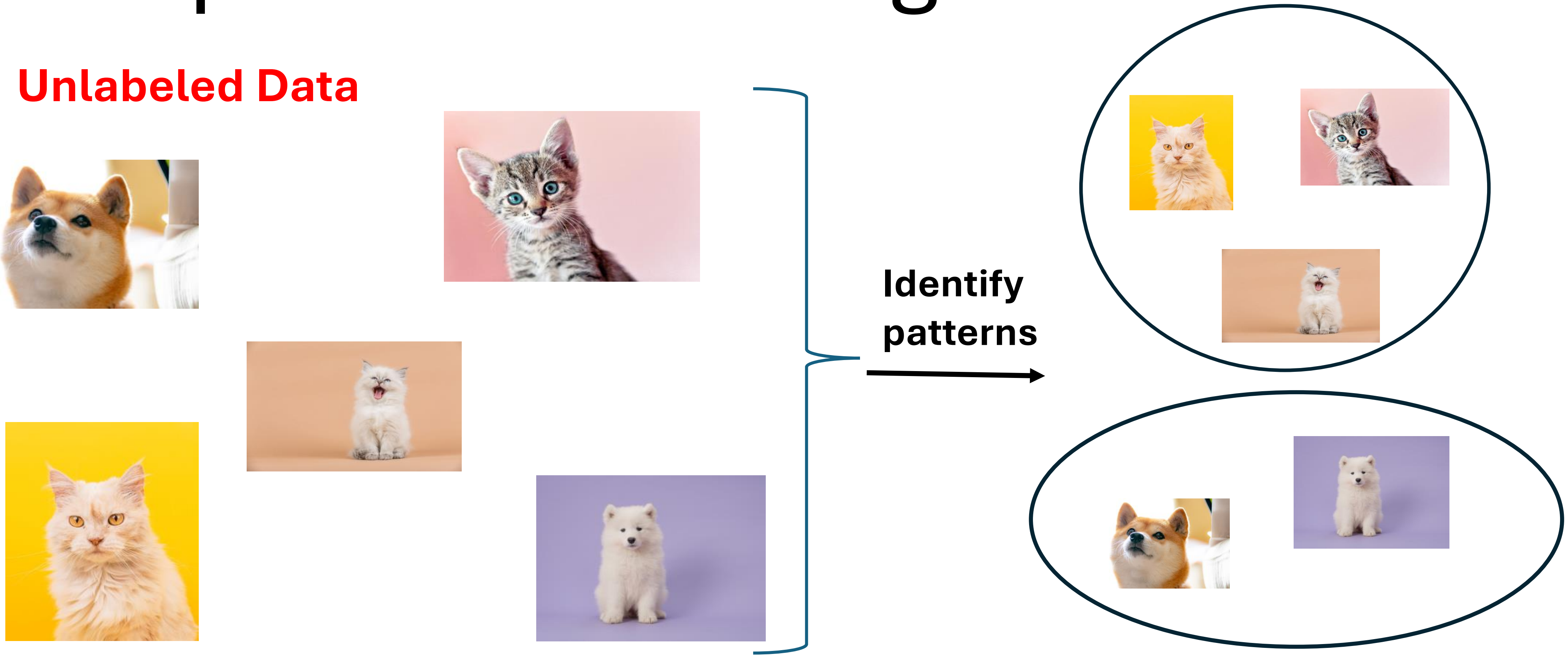
- Supervised Learning
- Unsupervised Learning
- Reinforcement Learning

Supervised Learning

Labeled Training Data



Unsupervised Learning



Photos provided by Pexels

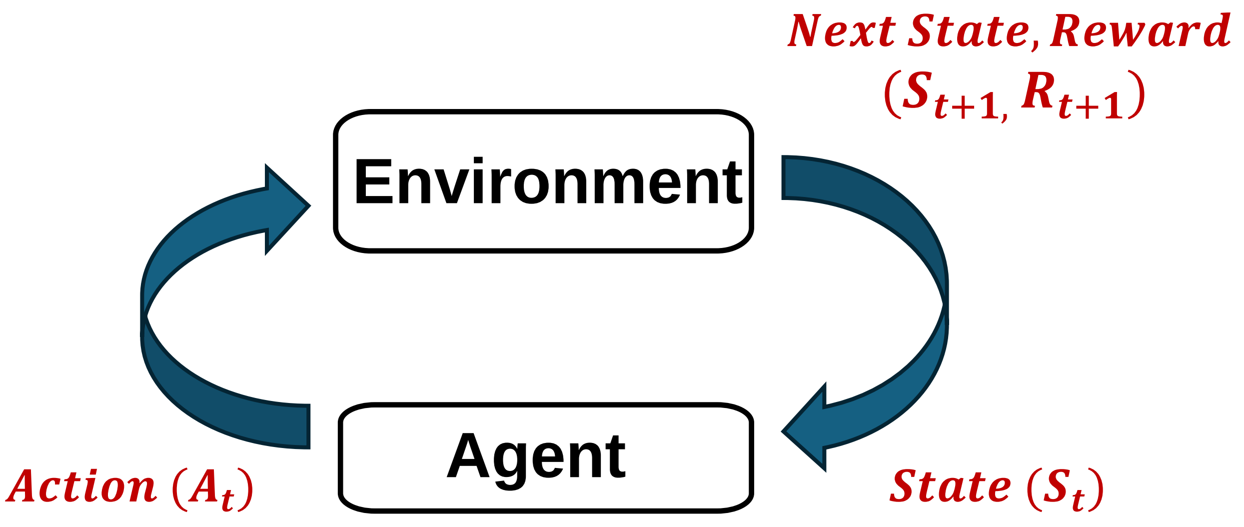
Reinforcement Learning

Feedback:
Score,
new display

State:
Display

Actions:
UP / Left /
Right / Down

Learn by Trial and Error



1. Agent observes the state and takes action
2. Environment puts the agent in a new state &
3. Gives a reward based on the action taken

GOAL: Learn policy to maximize the cumulative reward $\sum_t R\_t$

Photos provided by Pexels

Paradigms of Machine Learning

- Supervised Learning

- Unsupervised Learning

- Reinforcement Learning

Photos provided by Pexels

Paradigms of Machine Learning

- Supervised Learning
  - Fitting a function for the given labeled data (x, y)
  - $y \approx f(x)$

- Unsupervised Learning

- Reinforcement Learning

Paradigms of Machine Learning

- Supervised Learning
  - Fitting a function for the given labeled data (x, y)
  - $y \approx f(x)$
- Unsupervised Learning
  - Identifying patterns in unlabeled data
  - E.g. Clustering
- Reinforcement Learning

Photos provided by Pexels

Paradigms of Machine Learning

- Supervised Learning
  - Fitting a function for the given labeled data (x, y)
  - $y \approx f(x)$
- Unsupervised Learning
  - Identifying patterns in unlabeled data
  - E.g. Clustering
- Reinforcement Learning
  - Learning sequential tasks through trial and error
  - Feedback through reward/penalty

RL Demonstrations

Autonomous Helicopter

Pong game

AlphaGo by DeepMind

Photos provided by Pexels

RL Demonstrations

Autonomous Helicopter

Pong game

AlphaGo by DeepMind

Photos provided by Pexels

RL Demonstrations

Autonomous Helicopter

Pong game

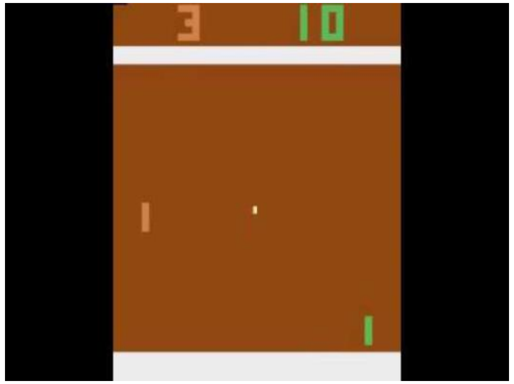
AlphaGo by DeepMind

Photos provided by Pexels

RL Demonstrations

Autonomous Helicopter

Pong game



AlphaGo by DeepMind

Photos provided by Pexels

One State RL: Multi-arm Bandits

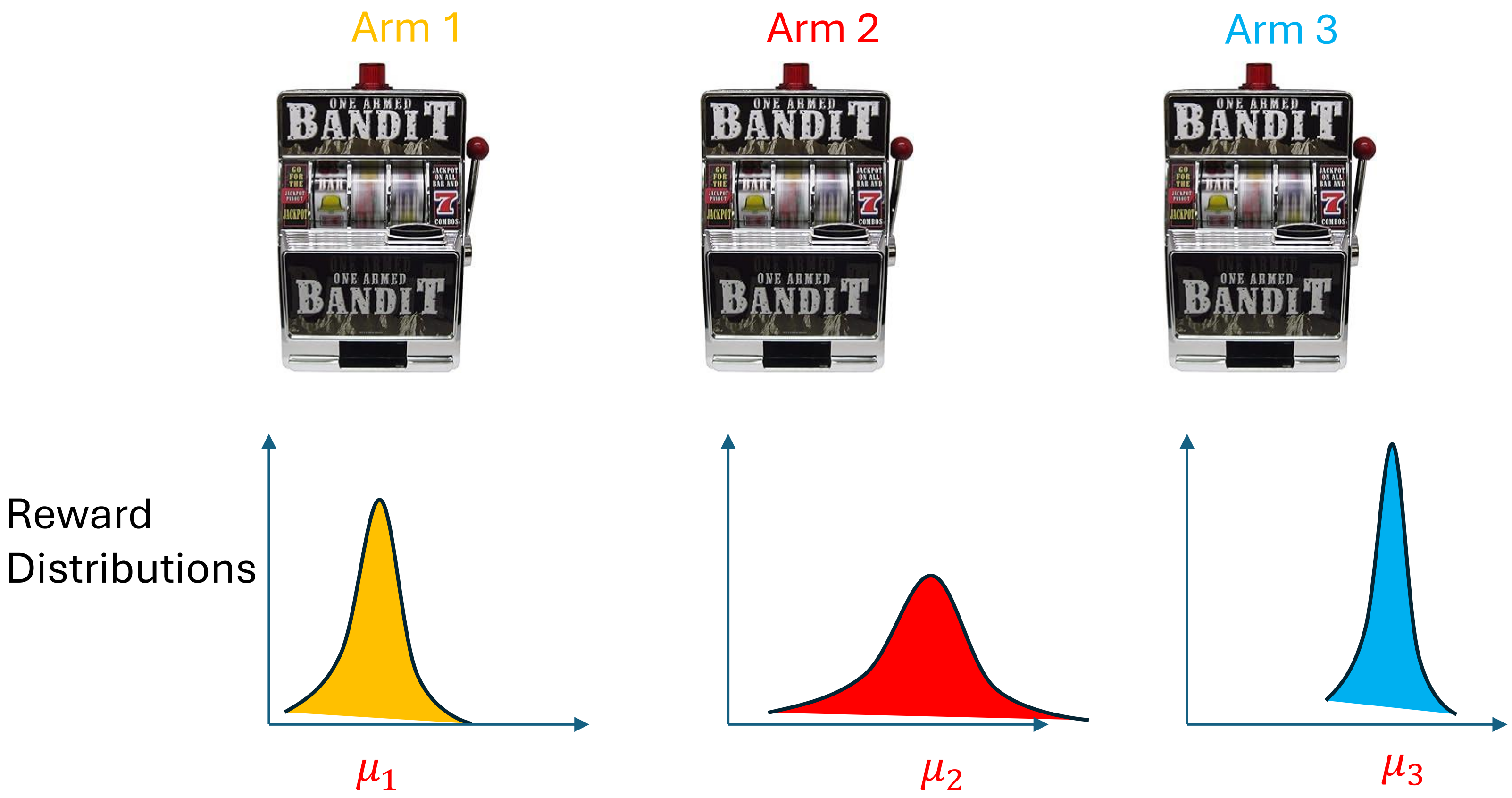
- Simplified version of RL problem: "Multi-arm Bandit" problem.
  - Only one state
  - Multiple actions (a.k.a. arms)
    - $\mathcal{A}$ – Action set

- A reward distribution corresponding to each arm
  - $\mathcal{R}_a$ – Reward distrution for action a
  - $\mu_a = \mathbb{E}[\mathcal{R}_a]$ – Expected reward for action a

- Applications: Recommendation systems, Ad placement, ...

Photos provided by Pexels

Multi-arm Bandits



Problem:

- Reward distributions are **unknown**
- Given **T chances** to pull the arms
- Which arms should be pulled to **maximize the total reward** in those T rounds

Exploration
Vs
Exploitation dilemma

ETC (Explore-Then-Commit)

1. **Explore:** Play each arm N times

2. Compute the sample average rewards $\bar{\mu}(a) = \frac{1}{N}\sum_{t=1}^{KN} R_t \, 1\{a_t = a\}$ for each arm $a \in \mathcal{A}$

3. **Commit:** Play the arm with the highest sample average for the remaining $T - KN$ rounds

μ^* - Optimal arm's expected reward

a_t - Arm played in round t

K - Number of arms

R_t - Sample reward obtained in round t

T - Total number of rounds

Performance (ETC Vs Best possible reward) : $T\mu^* - \sum_{t=1}^{T} \mathbb{E}[R_t]$

How much to Explore? $N \approx (\frac{T}{K})^{\frac{2}{3}}$

Photos provided by Pexels

ϵ-Greedy (Explore uniformly)

1. Play each arm once

2. In each round t:
    - Toss a coin with bias ϵ.
    - If it lands in head: Explore - Play any arm randomly
    - Else: Exploit - Play the arm with the highest sample average so far

What ϵ to choose? $\epsilon \approx (\frac{K}{T})^{\frac{1}{3}}$

Photos provided by Pexels

UCB (Upper Confidence Bound)

Optimism under UnCertainty

1. Play each arm once in the first K rounds

2. For $t > K$:
   - Play the arm with the highest $UCB_t(a) = \underbrace{\overline{\mu_{t-1}}(a)}_{\text{Exploit}} + \underbrace{\sqrt{\frac{2\log T}{n_{t-1}(a)}}}_{\text{Explore}}$
   - Based on the observed sample reward R_t, update $n_t(a_t)$ and $\overline{\mu_t}(a_t)$
     - $n_t(a_t) = n_{t-1}(a_t) + 1$
     - $\overline{\mu_t}(a_t) = \frac{1}{n_t(a_t)}[(n_t(a_t) - 1)\, \overline{\mu_{t-1}}(a_t) + R_t]$

Exploit: High sample reward arms are favoured
Explore: Least played arms are favoured

Contextual Bandits – Multiple states

- News article Recommendation systems

- Articles – arms
- Like / Dislike – Reward
- User – State

Different users have different preferences to articles

Photos provided by Pexels