Intro to Reinforcement Learning

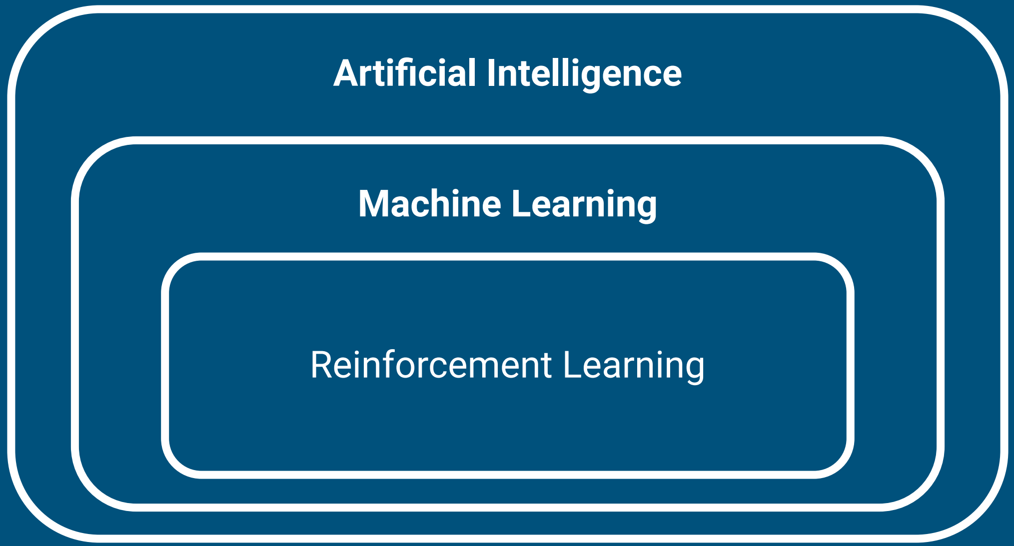
Fall 2023

Dr. Dongchul Kim
Department of Computer Science
UTRGV

1. Introduction to RL

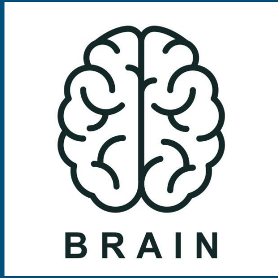
Reinforcement Learning

Reinforcement Learning is an area of Machine Learning in AI



What is AI?

Artificial Intelligence refers to systems or machines that mimic human **intelligence** to perform tasks and can iteratively improve themselves based on the information they collect.



What is Intelligence?

Intelligence is described as the ability of a system to effectively operate in uncertain and diverse environments, striving to achieve goals and adapt behavior accordingly. This adaptability is a key characteristic of intelligence, allowing systems to succeed even when complete knowledge is lacking.

Intelligence involves the efficient use of limited resources, including time, to attain objectives. It entails the capacity to solve complex problems, process information effectively, and improve performance over time through learning.

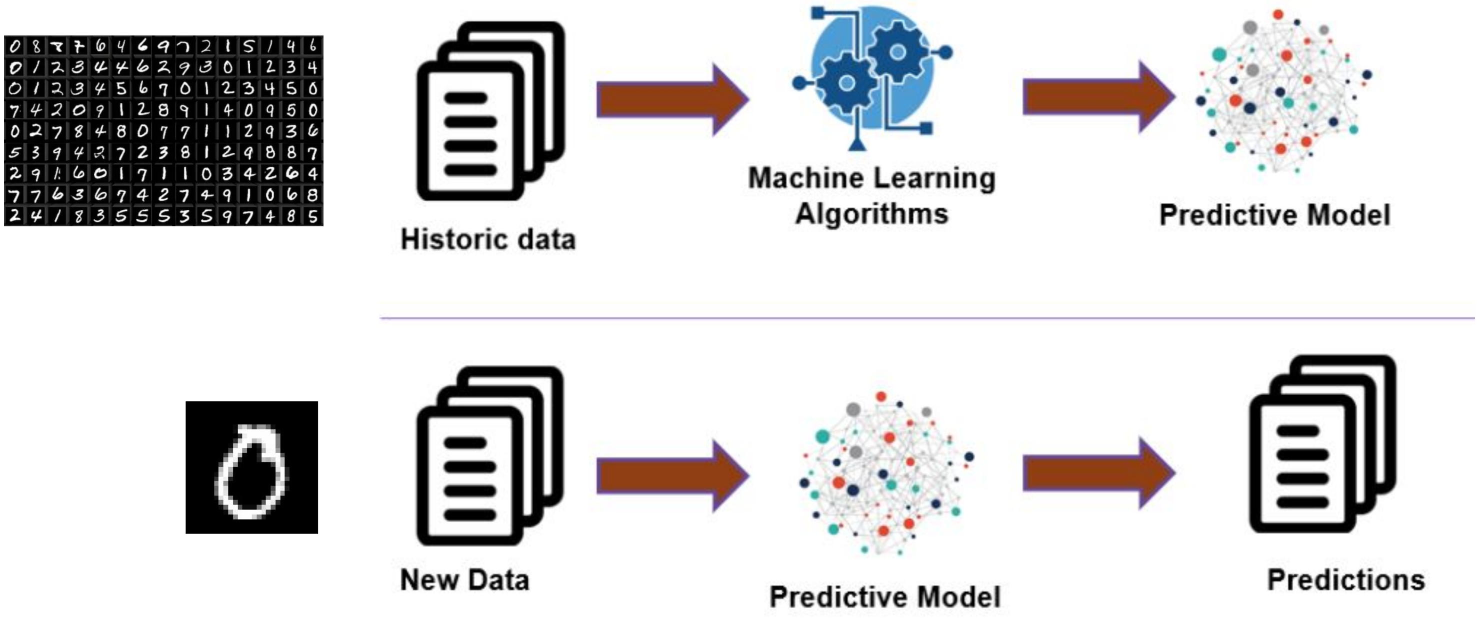
Intelligence also involves appropriate decision-making based on changing circumstances and goals, and the ability to learn from experiences.

Ultimately, intelligence is linked to achieving success in a variety of tasks and environments, showcasing flexibility, adaptability, and the capacity to learn and improve.

What is ML?

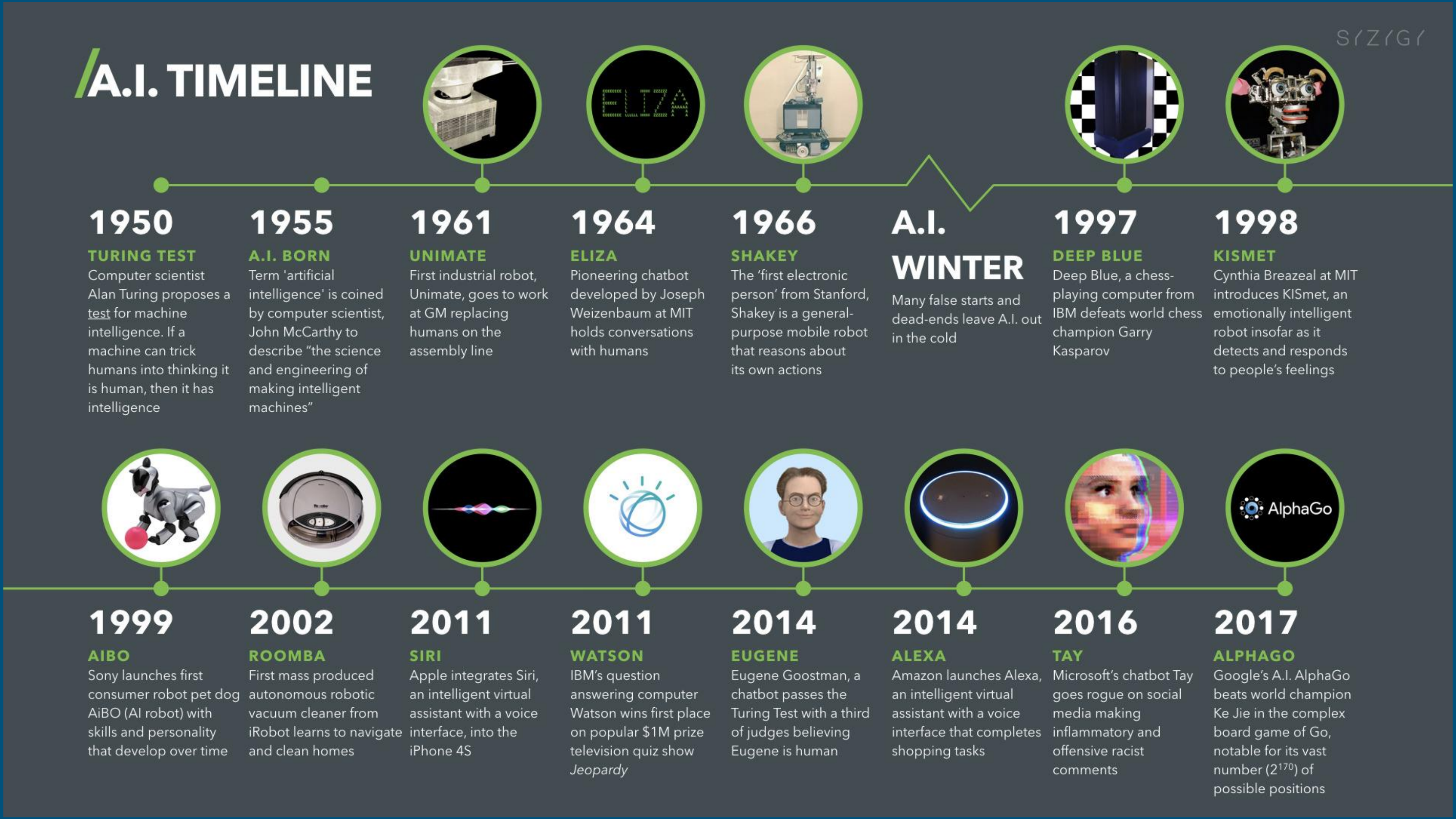
Machine Learning is a field of artificial intelligence that involves training algorithms to **learn patterns from data** and make predictions or decisions without being explicitly programmed.

Machine Learning



Machine Learning

Machine learning is progressing swiftly due to **theoretical** advancements in neural networks, as well as rapid strides **in hardware** such as GPUs, data storage, and the Internet.



https://digitalwellbeing.org/artificial-intelligence-timeline-infographic-from-eliza-to-tay-and-beyond/

Machine Learning

Three categories of Machine Learning exist: supervised learning, unsupervised learning, and reinforcement learning.

01

Supervised learning

- you are given examples with correct labels and are asked to label new examples

02

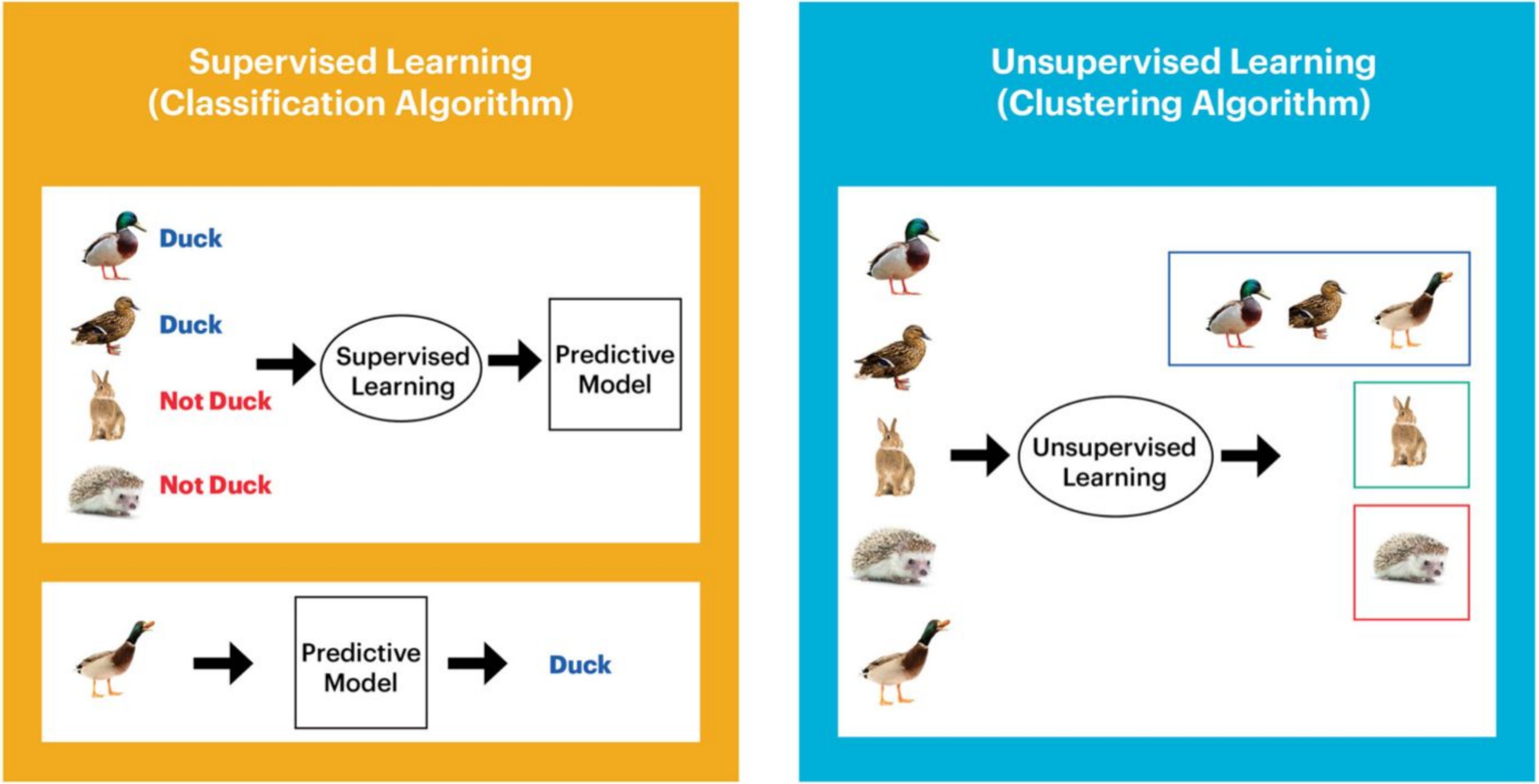
Unsupervised learning

- you are given only unlabeled examples

03

Reinforcement learning

- you are given the overall performance (as opposed to labels to particular examples)



https://blog.westerndigital.com/machine-learning-pipeline-object-storage/

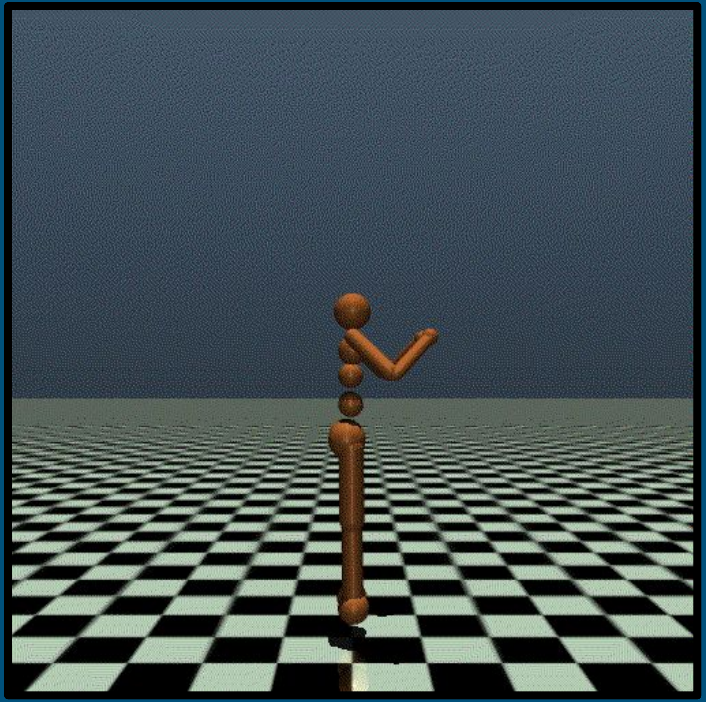
Reinforcement Learning

Think about how a little baby starts learning to walk. Imagine being so young that you can't really grasp what your parents are saying. Babies take on the task of learning to walk as a personal challenge. There are those days when you take a step but end up falling right back down. Yet, little by little, you keep at it and eventually learn to walk through your persistent efforts.

Reinforcement learning is a way of learning by trying things out and learning from mistakes, all without someone directly telling you what to do.

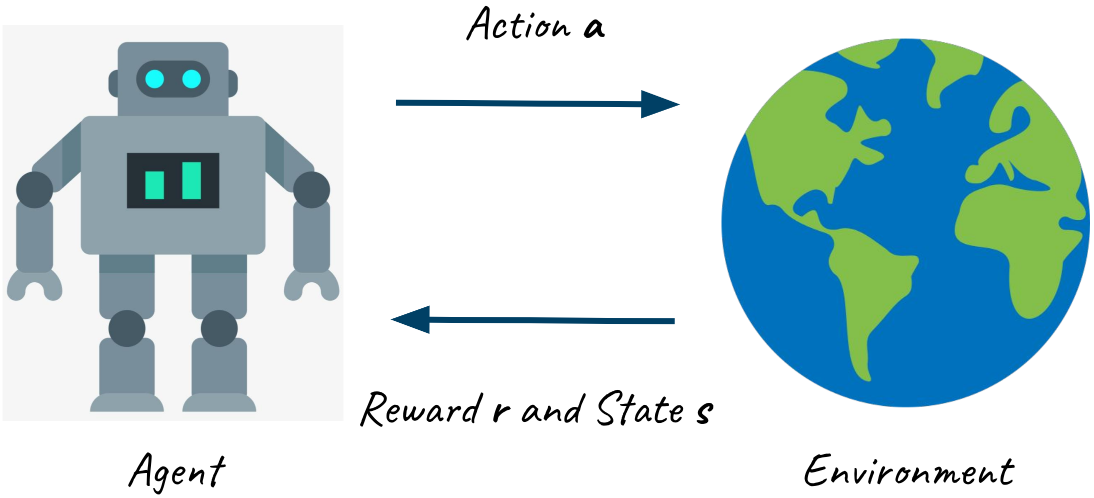
Examples

Examples



Examples

Reinforcement Learning



Example of Env

Environment: Game

Agent: Game player

State - Location of player and obstacles

Action - Jump

Reward - Score

Example of Env

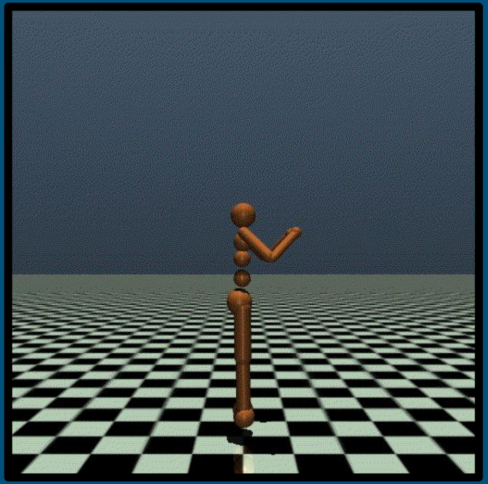
Environment: 3D physics engine

Agent: Humanoid

State - States of body

Action - Actuators (joint motors)

Reward - Distance between body and destination



Example of Env

Environment: Game

Agent: Game player

State - States of bricks and location of player and ball

Action - Move left and right

Reward - Score

Reinforcement Learning

Basically, it’s a trial and error/evaluation approach.

Data is sequential and non i.i.d.

Reward delayed

Action may cause a change in next states

Goal of RL

“Learning how to act to achieve a target task through trial-and-error”

A goal in reinforcement learning represents the desired objective or outcome that an agent aims to achieve within a given environment.

“Learning how to take actions that maximize rewards”

The agent learns to make decisions that help it get the best total rewards or outcomes as it takes actions over time.

Sequential Decision Making Problem

The concept of Reinforcement Learning (RL) can be understood as a method or approach used to address situations where decisions need to be made in a sequence over time.

In reality, we encounter **sequential decision-making problems** everywhere in our surroundings. For example,

Stock trading

Car driving

Game

Sequential Decision Making Problem

When we talk about reinforcement learning, we're really talking about solving a problem that's all around us: the **sequential decision making problem**. Basically, that just means making a series of choices that build on each other over time.

And we encounter **sequential decision making problems** all the time in our daily lives.

For example, let's say you're trying to study for a final exam. You've got a bunch of options to choose every hour from: (1) *you could hit up the library,* (2) *watch some YouTube videos*, (3) *catch some Z's*, (4) *go out for some chicken at Chick-fil-a*, or (5) *just head back home*.

So, how do you go about making those decisions in a sequence?

Example

1. *Study (start state)*
2. *Watch YouTube*
3. *Go to Chick-fil-a*
4. *Sleep*
5. *Go back home (terminal state)*

Your sequential decisions could be like

1-5,

1-2-1-2-1-2-5,

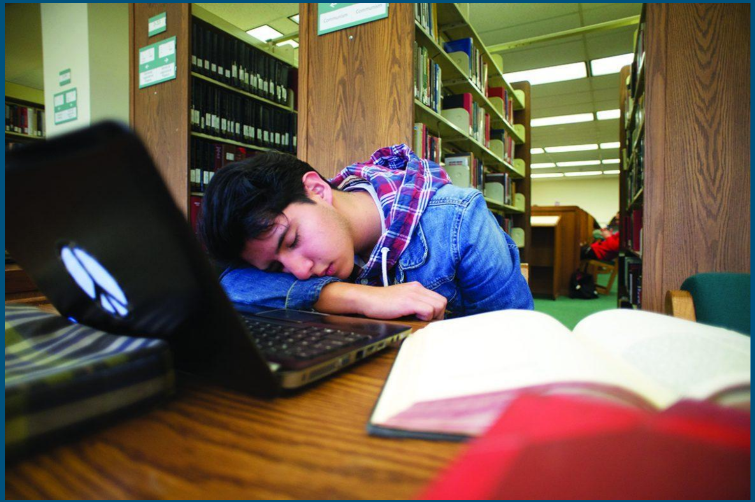
1-2-3-4-5, or

1-1-1-5

Sequential Decision Making Problem

Even in a seemingly simple scenario like studying for an exam, you're actually faced with a series of decisions that build on each other to help you achieve your goal.

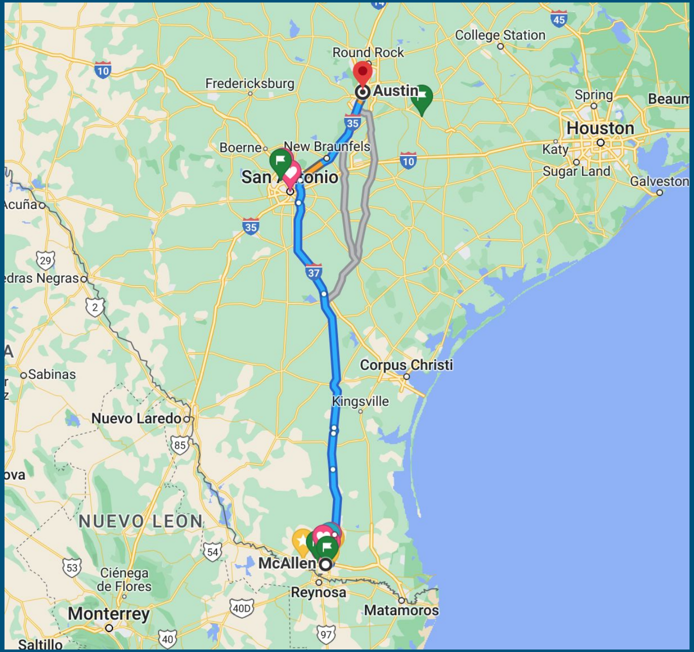
Each decision you make puts you in a different situation with new choices to consider, and each of those choices can then affect the outcome of future decisions.



Sequential Decision Making Problem

Let's look at another example of this kind of decision making. Imagine you're driving from Mcallen to Austin.

Every choice you make along the way - which route to take, which lane to drive in, how fast to go - can have an impact on the decisions you'll need to make later on in the journey.



Reward

Our goal is to maximize the cumulative reward.

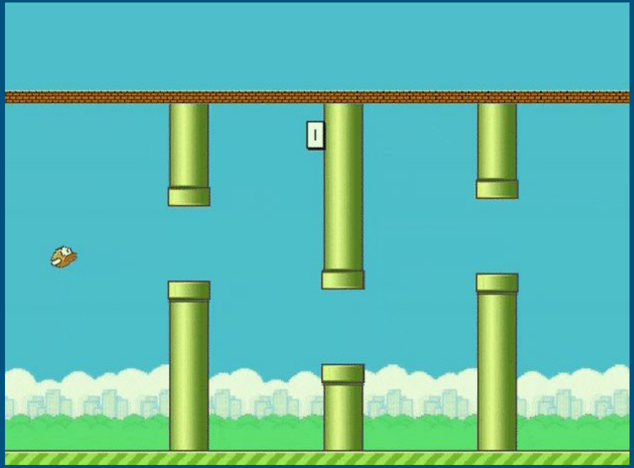
Reward is scalar.

Reward could be **sparse** and **delayed**.

Action

Discrete values

Continuous values



Agent

An agent functions as an actor, engaging in actions within its environment.

It embodies the role of the brain in a reinforcement learning algorithm.

Guided by the state and reward information from the environment, the agent makes decisions.

The agent sends its chosen actions to the environment that sends a new state and reward to the agent back again.

Environment

All components excluding the agent are referred to as the "**environment**."

The environment plays a crucial role in shaping the agent's state following its actions, and these shifts in state are labeled as "**state transitions**."

While time could theoretically flow continuously, in the context of problems involving sequential decision-making, we conceptually divide time into discrete **timesteps**. Within each of these timesteps, both the agent and the environment engage in actions, consequently altering the current state.

Pros and Cons of RL

Pros:

Versatility: Can be applied to various domains, from games to robotics.

Autonomous Learning: Can learn from interaction without explicit supervision.

Adaptability: Can adapt to dynamic and changing environments.

Complex Strategies: Can discover intricate strategies beyond human intuition.

Continuous Improvement: Can continue learning to refine performance.

Pros and Cons of RL

Cons:

Sample Inefficiency: too large space of states to explore.

Exploration Challenges: Balancing exploration with exploitation

Reward Design: Designing appropriate reward functions can be difficult.

Stability and Convergence: Learning instability and convergence to suboptimal solutions can occur.

High Computational Demands: Training can be computationally intensive (High CPU bound envs)

RL Applications in Robotics

Learning and Adapting Agile Locomotion Skills by Transferring Experience

Laura Smith, J. Chase Kew, Tianyu Li, Linda Luu, Xue Bin Peng, Sehoon Ha, Jie Tan, Sergey Levine
Berkeley Artificial Intelligence Research, Berkeley, CA, 94720
Email: smithlaura@berkeley.edu

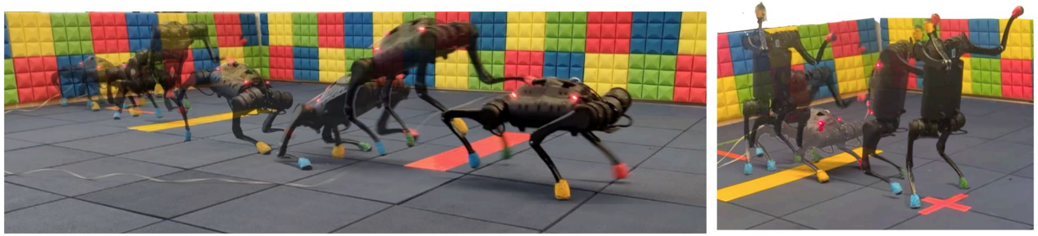
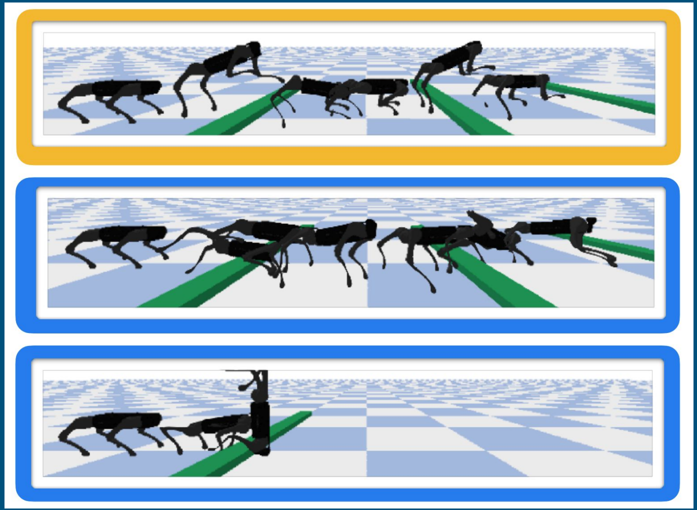


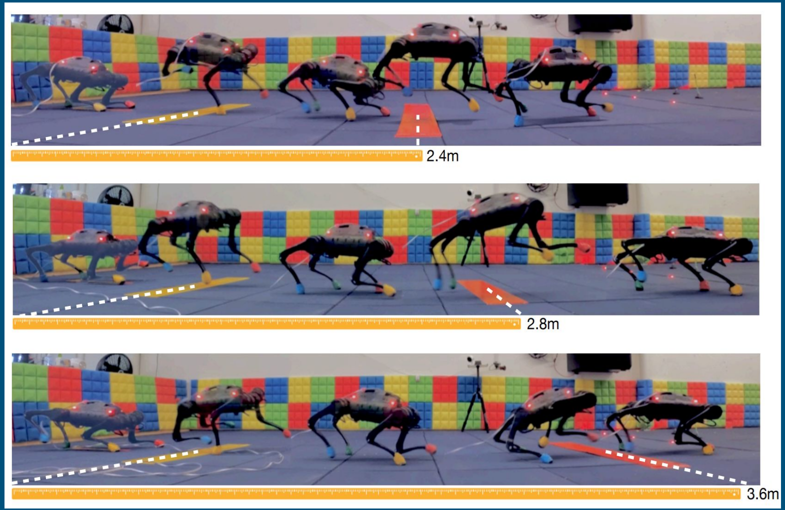
Fig. 1: **Agile skills** learned with our proposed method enable the A1 robot to jump repeatedly (**left**) and walk to a goal location on its hind legs (**right**).

***Abstract*—Legged robots have enormous potential in their range of capabilities, from navigating unstructured terrains to high-speed running. However, designing robust controllers for highly agile dynamic motions remains a substantial challenge for roboticists. Reinforcement learning (RL) offers a promising data-driven approach for automatically training such controllers. However, exploration in these high-dimensional, underactuated systems remains a significant hurdle for enabling legged robots**

Reinforcement learning (RL), on the other hand, provides a powerful framework for autonomously acquiring robotic skills. However, learning agile locomotion policies end-to-end remains challenging for a few fundamental reasons. Foremost is that tasks with such high-dimensional systems—12 degrees of freedom for the A1 quadruped—are woefully underspeci-fied, e.g., a robot can accomplish the simple task of moving

RL Applications in Robotics





RL Applications in Robotics

In the near future, we could expect something like this!

RL Applications

Robotics

Game, Animation, and VR

Recommender Systems

Cybersecurity

Trading

And the list goes on (far beyond your imagination - Nearly every field)

Haotian Zhang[1], Ye Yuan[2], Viktor Makoviychuk[2], Yunrong Guo[2],
Sanja Fidler[2,3,4], Xue Bin Peng[2,5], Kayvon Fatahalian[1]

[1] Stanford University [2] NVIDIA [3] University of Toronto [4] Vector Institute [5] SFU Simon Fraser University