



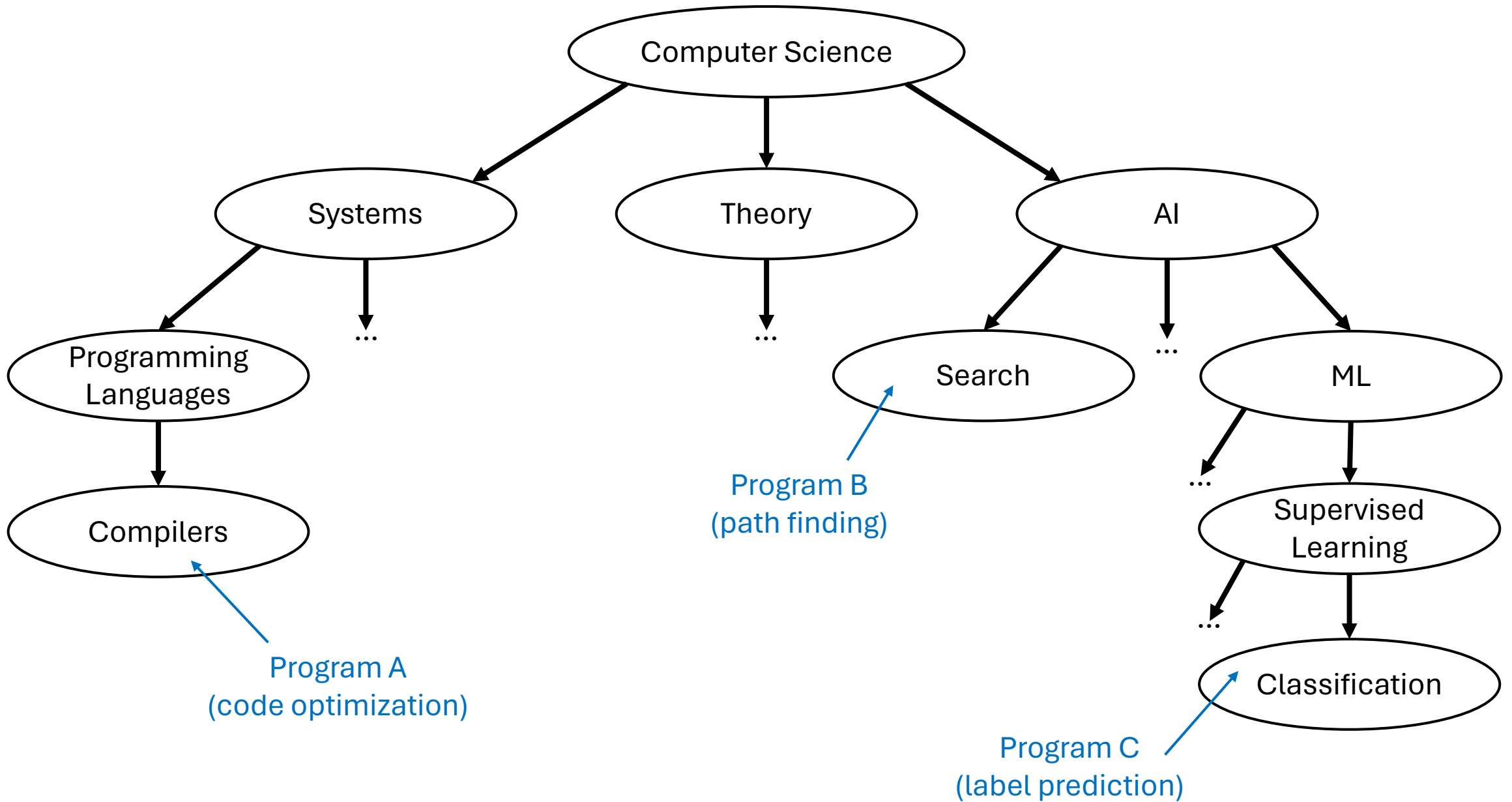
# COMPSCI 389

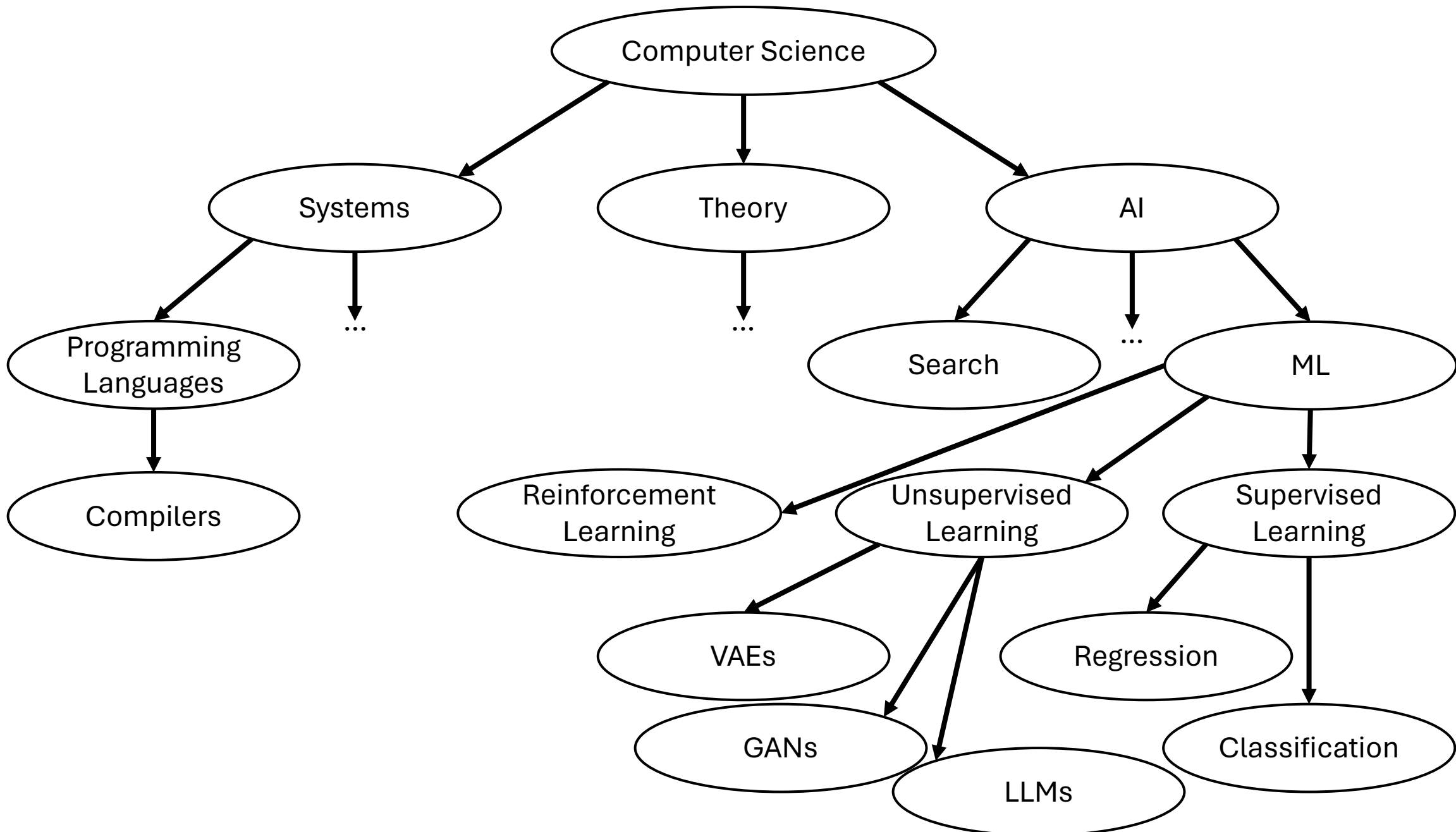
## Introduction to Machine Learning

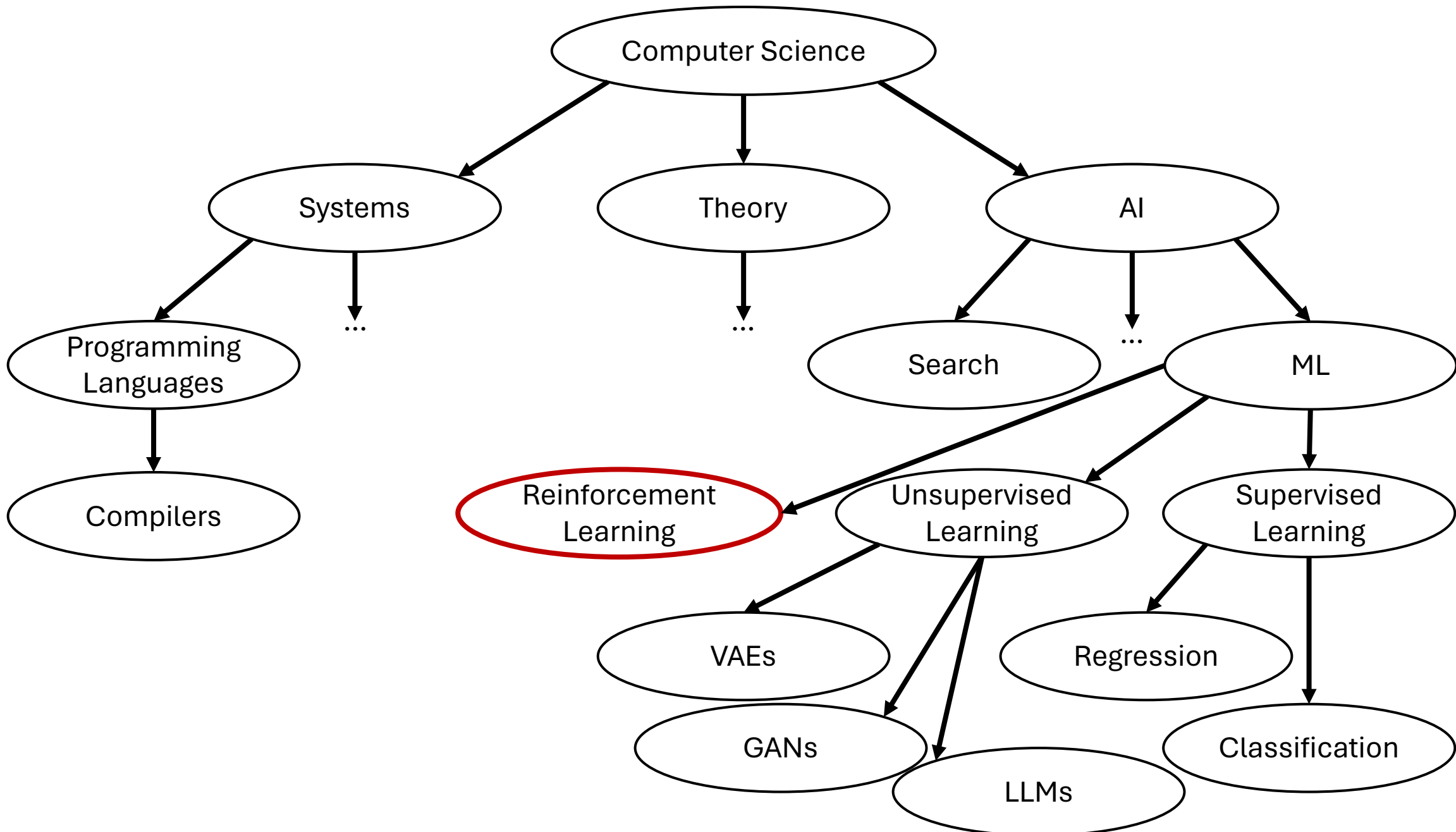
**Days:** Tu/Th. **Time:** 2:30 – 3:45 **Building:** Morrill 2 **Room:** 222

### **Topic 13.0: Reinforcement Learning**

Prof. Philip S. Thomas (pthomas@cs.umass.edu)







# Supervised and Unsupervised Learning

- Algorithm learns from a fixed data set
- Supervised: Data includes labels
- Unsupervised: Data does not include labels
- **Semi-Supervised Learning:** Some data includes labels
  - Use unlabeled data to learn a representation (e.g., features)
  - Use labeled data to train a model using the learned representation
  - Not discussed further in this class

# Reinforcement Learning

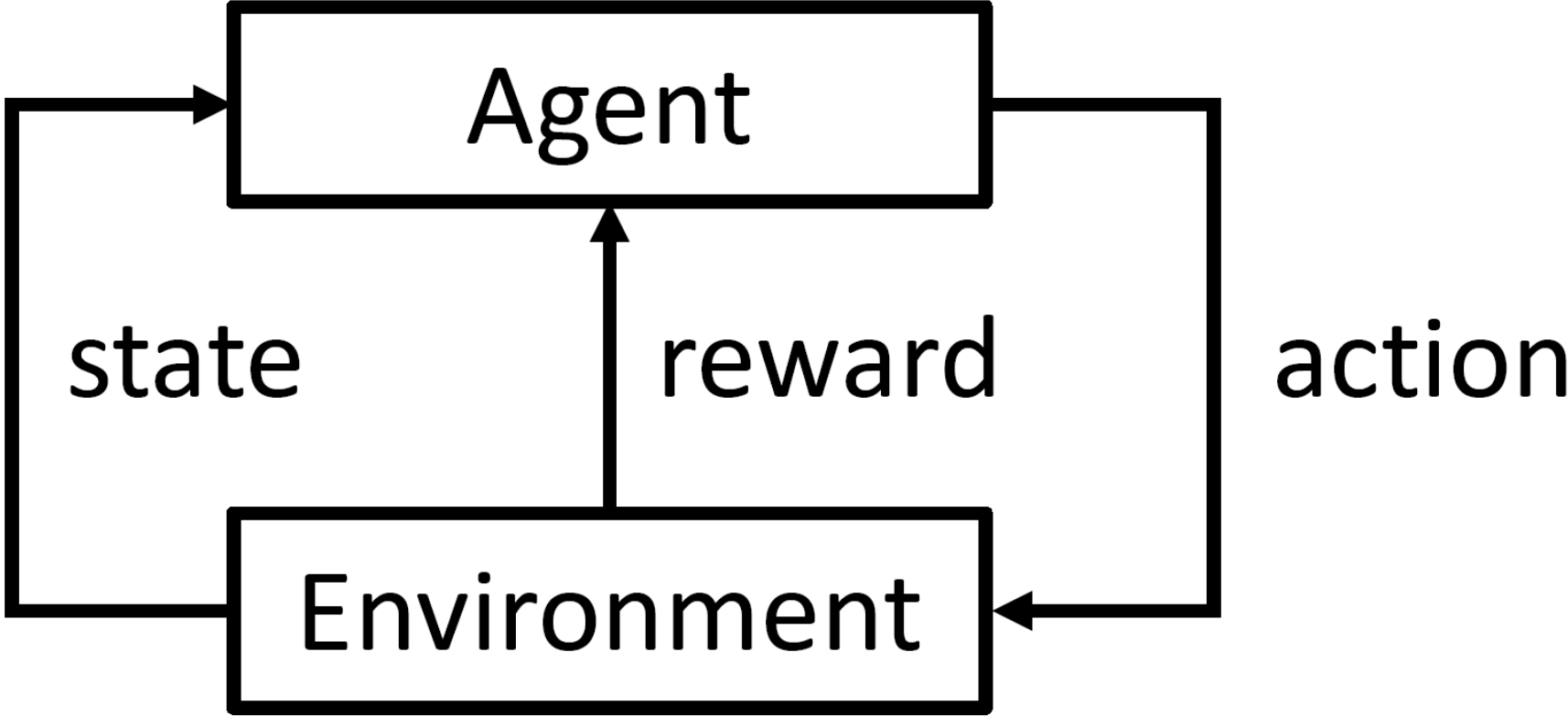
- There is no fixed data set.
- The decisions (predictions) made by the agent change the data the agent receives!
- Modeled as an agent interacting with an environment

*Reinforcement learning is an area of machine learning, inspired by **behaviorist psychology**, concerned with how an agent can learn from interactions with an environment*

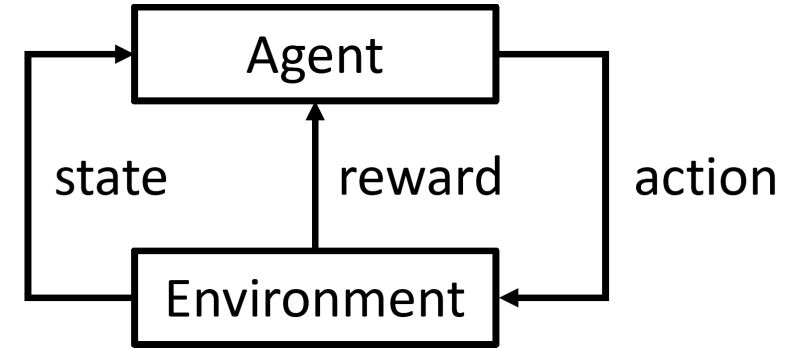
– Wikipedia / Sutton&Barto / Phil

How rewards and punishments shape our behavior.

# Agent-Environment Diagram



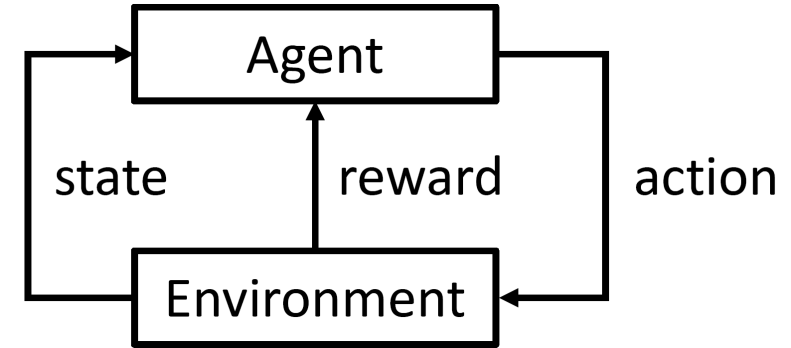
# RL Problem Description



- The agent interacts with the environment over time  $t \in \{0, 1, \dots\}$ .
- At each time the agent observes the **state** of the environment
  - For now, we assume that it observes the full state of the environment.
    - This is called the **fully observable** setting.
  - In general, the agent might only make a partial (noisy) observation about the state of the environment through its sensors.
    - This is called the **partially observable** setting.
- Based on its observation of the state, the agent selects an **action**.
  - The “parametric model” in RL is the mechanism in the agent that takes a state as input and produces an action as output.
  - This mechanism is called a **policy**.
    - Worse, in RL, **model** means something completely different! (A *model* of the environment)
  - It can be deterministic (always producing the same action given a state) or stochastic (producing a distribution over actions given the state).

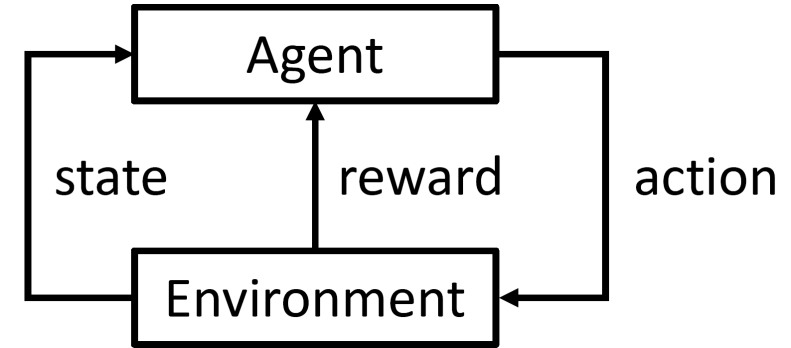


# RL Problem Description



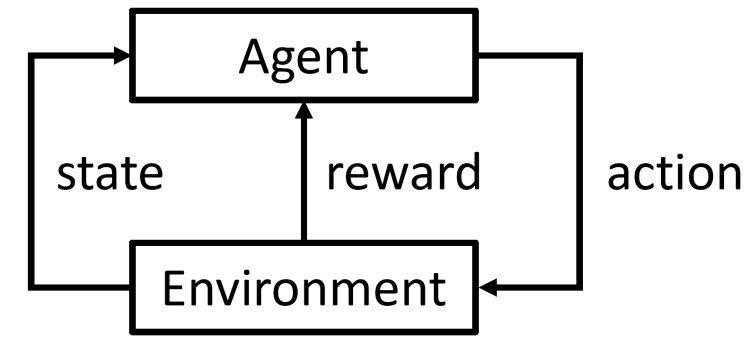
- The agent interacts with the environment over time  $t \in \{0, 1, \dots\}$ .
- At each time the agent observes the **state** of the environment
- Based on its observation of the state, the agent selects an **action**.
  - The **policy** is the mechanism that determines the action given the state.
- The action causes the state of the environment to change.
  - This is called a **state transition**.
- When the state transitions, the environment also emits a scalar **reward**.
  - Intuitively, this reward indicates how “good” the current state is in the short term.
  - Sometimes it is called the **immediate reward** to emphasize the short-term nature of its evaluation.
- The sequence of agent-environment interactions can end, and the process restarts.
  - Each sequence of agent-environment interactions starting from time 0 is called an **episode**
  - This is the **episodic setting**. If the sequence of interactions never ends, it is called the **continuing setting**.
- The agent’s goal is to find a policy that maximizes the total amount of reward that it receives.
  - The **return** is the sum of rewards that the agent receives during one episode.
  - The same policy can produce different returns during different episodes due to stochasticity in the state transitions, rewards, and policy.
  - The agent’s goal is to maximize the **expected return**.

# RL Problem Description



- The agent interacts with the environment over time  $t \in \{0, 1, \dots\}$ .
- At each time the agent observes the **state** of the environment
- Based on its observation of the state, the agent selects an **action**.
  - The **policy** is the mechanism that determines the action given the state.
- The action causes a **state transition**.
- When the state transitions, the environment also emits a scalar **reward**.
- Each sequence of agent-environment interactions starting from time 0 is called an **episode**. Episodes can end (terminate).
- The **return** is the sum of rewards that the agent receives during one episode.
- The agent's goal is to maximize the **expected return**.

# Key Properties of RL



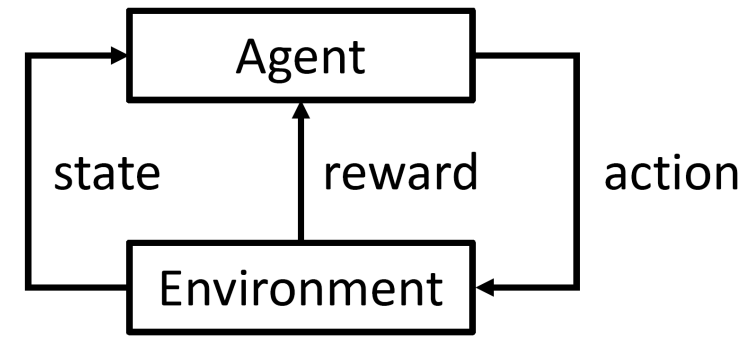
- **Evaluative feedback, not instructive feedback**

- **Instructive feedback** tells an agent what the correct decisions would have been
  - Labels in supervised learning provide instructive feedback.
- **Evaluative feedback** tells an agent how good its decisions were
  - Rewards in RL provide evaluative feedback.
  - Evaluative feedback can be noisy (random)
  - The range of possible feedback values may not be known.
    - Is a reward of +10 good or bad? The agent must interact with the environment to figure this out!

- **Sequential**

- The agent's goal is to maximize the expected return (expected sum of rewards).
- This can require it to forgo larger short-term rewards to obtain larger rewards in the future.

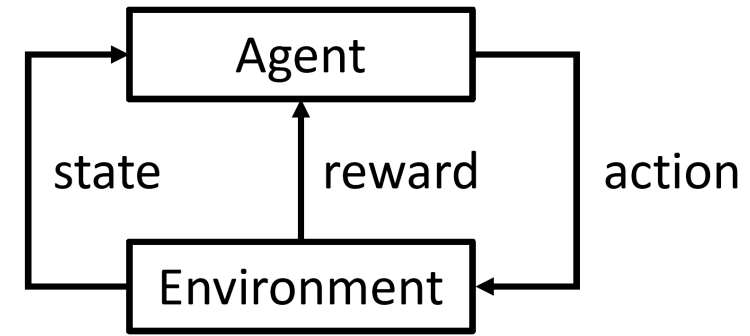
# RL Examples



- **Agent:** Child
- **Environment:** World
- **Goal:** The child may learn to grasp an object or get a parent's attention
  
- **State:** The state of the world around the child (partially observed!)
- **Action:** Decision of how much to activate each muscle
- **Reward:** Positive when an object is picked up, negative when an object is dropped, positive when a parent responds, etc.

# RL Examples

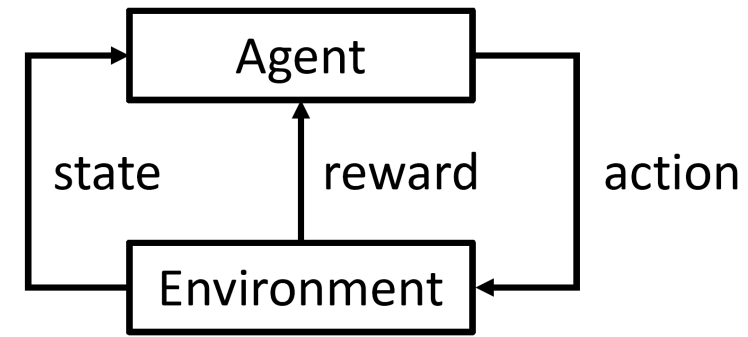
- **Agent:** Dog
- **Environment:** World
- **Goal:** Learn to fetch or catch





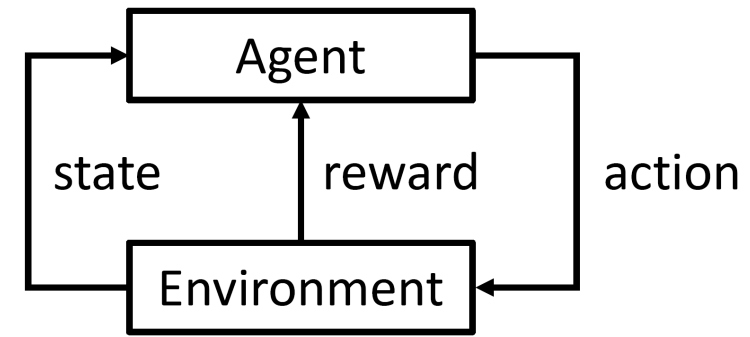
# RL Examples

- **Agent:** Dog
- **Environment:** World
- **Goal:** Learn to fetch or catch
  
- **State:** The state of the world around the dog (partially observed!)
- **Action:** Decision of how much to activate each muscle
- **Reward:** Positive when food is obtained
  
- **Note:** Each catching attempt can be viewed as an **episode**.



# RL Examples

- **Agent:** Robot
- **Environment:** Lab
- **Goal:** Lift a heavy object

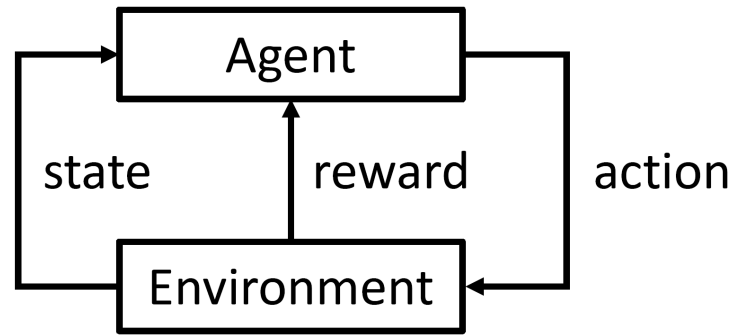






The learned policy exploits the container dynamics.

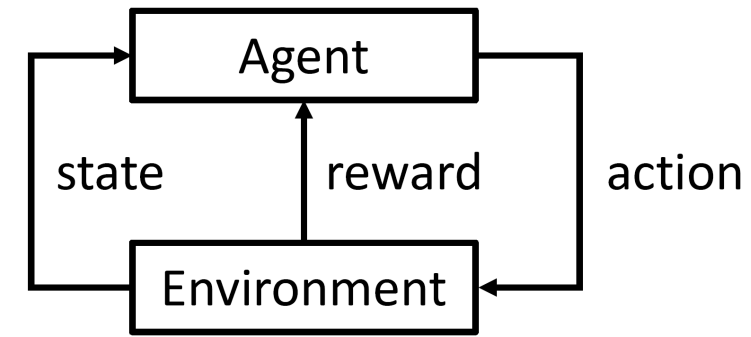
# RL Examples



- **Agent:** Robot
- **Environment:** Lab
- **Goal:** Lift a heavy object
  
- **State:** State of the robot. Observation: Sensor readings for joint angles and angular velocities.
- **Action:** How much power to give to each motor
- **Reward:** Positive when the jug is successfully lifted above the robot's head
- **Note:** The above is a simplification. The policy search was done over target trajectories for the arm, using a low-level controller to achieve those positions.
- **Note:** Each lifting attempt is an **episode**.

# RL Examples

- **Agent:** Program
- **Environment:** Software environment
- **Goal:** Learn when to garbage collect
  
- **State:** State of a program that is executing.
- **Action:** Whether or not to garbage collect.
  - Garbage collect is a slow operation but frees up memory.
  - When memory runs out, collection is forced.
- **Reward:** Zero when the action “no-collect” is chosen, and a negative penalty proportional to the time it takes to perform collection when “collect” is chosen.



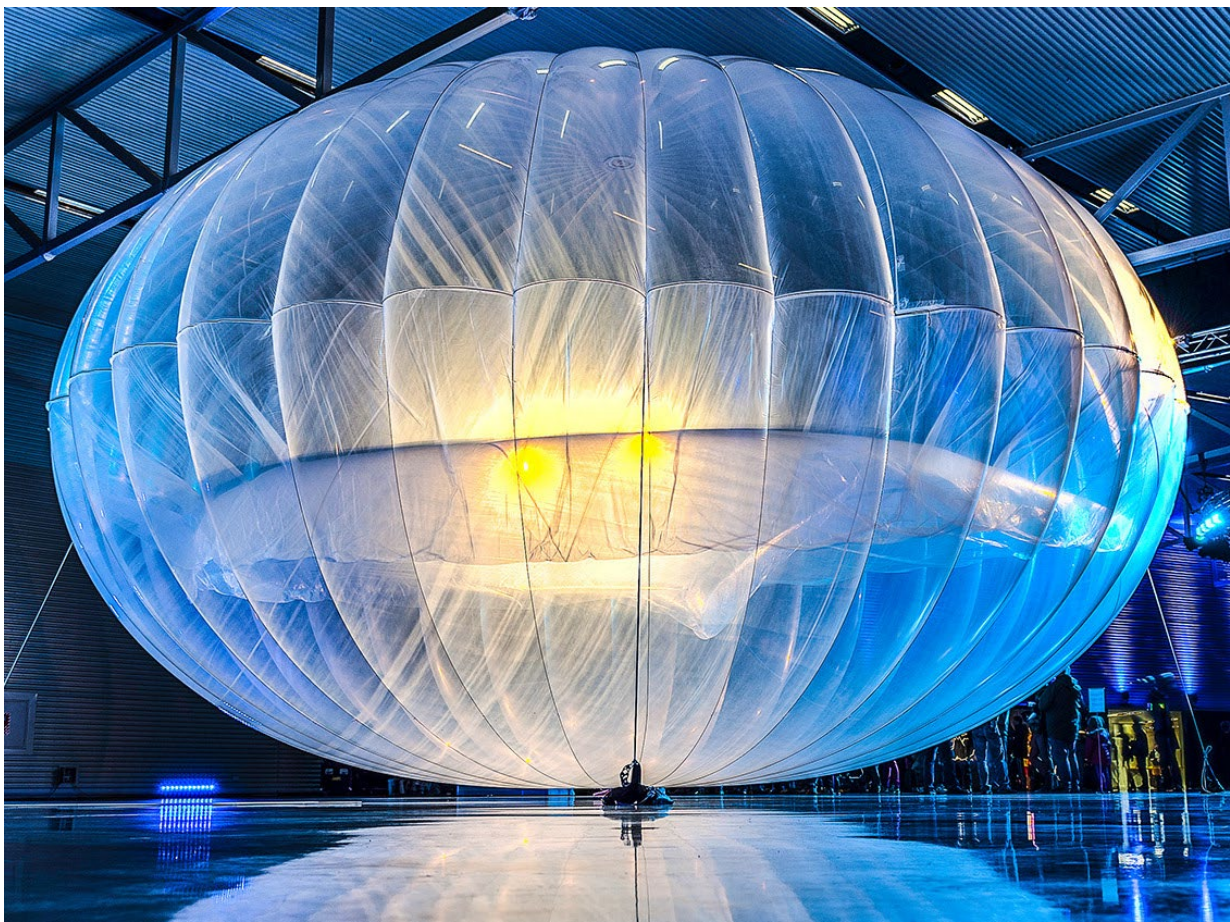
# Functional Electrical Stimulation (FES)



# Elevator Scheduling



# Loon Balloon



# Targeted Advertisements

The screenshot shows the Adobe website's 'Products' page. The browser address bar displays 'http://www.adobe.com/products/cata' and the page title is 'Adobe products'. The navigation menu includes 'Products', 'Business solutions', 'Support & Learning', 'Download', 'Company', and 'Buy'. A search bar is located on the right, and links for 'Privacy', 'My cart', and 'Sign in to My Adobe' are visible. The main content area features a large banner with the text 'Save 40% on all-new tools. Only in Creative Cloud. Introductory offer for CS customers.' and a 'Join now' button. To the right, there is a smaller advertisement for 'Photoshop CC plus Lightroom desktop and mobile for US\$9.99/mo.' with a 'Join now >' link. The background of the main banner is a vibrant, colorful burst of light trails.

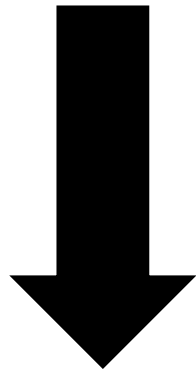
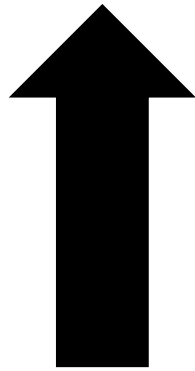
# Atari 2600



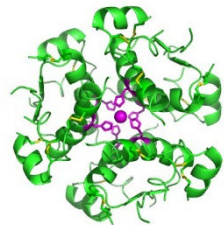


# Example Applications: Diabetes Treatment

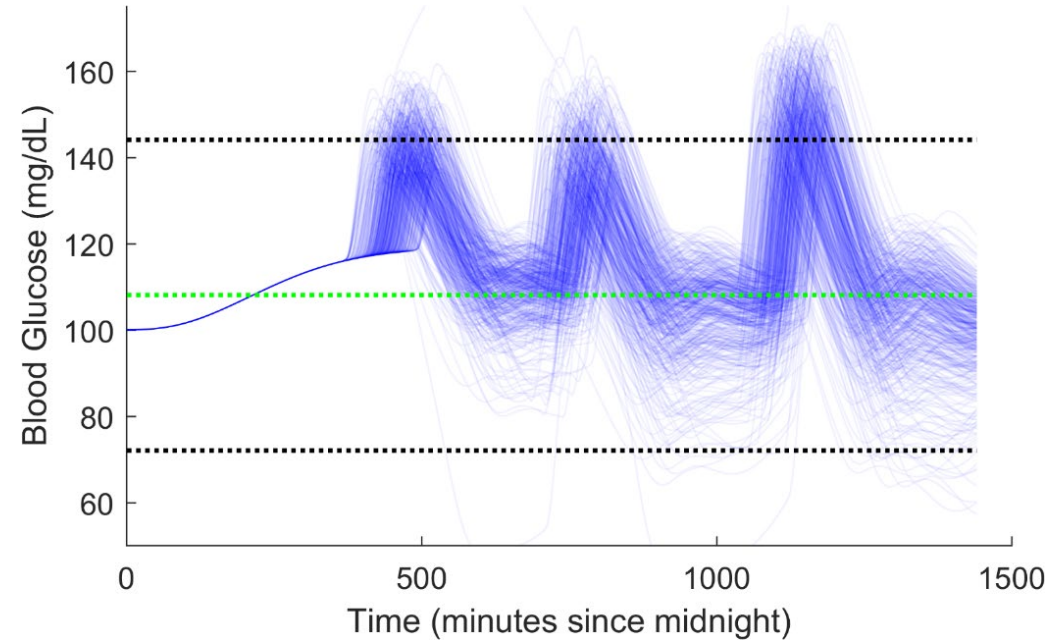
Blood Glucose  
(sugar)



Eat Carbohydrates

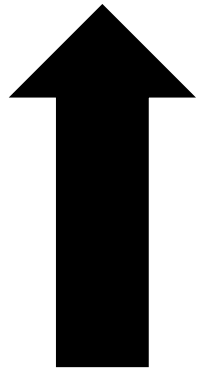


Release Insulin

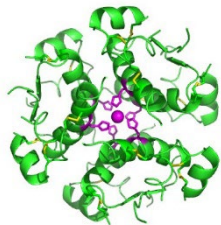


# Type 1 Diabetes Treatment

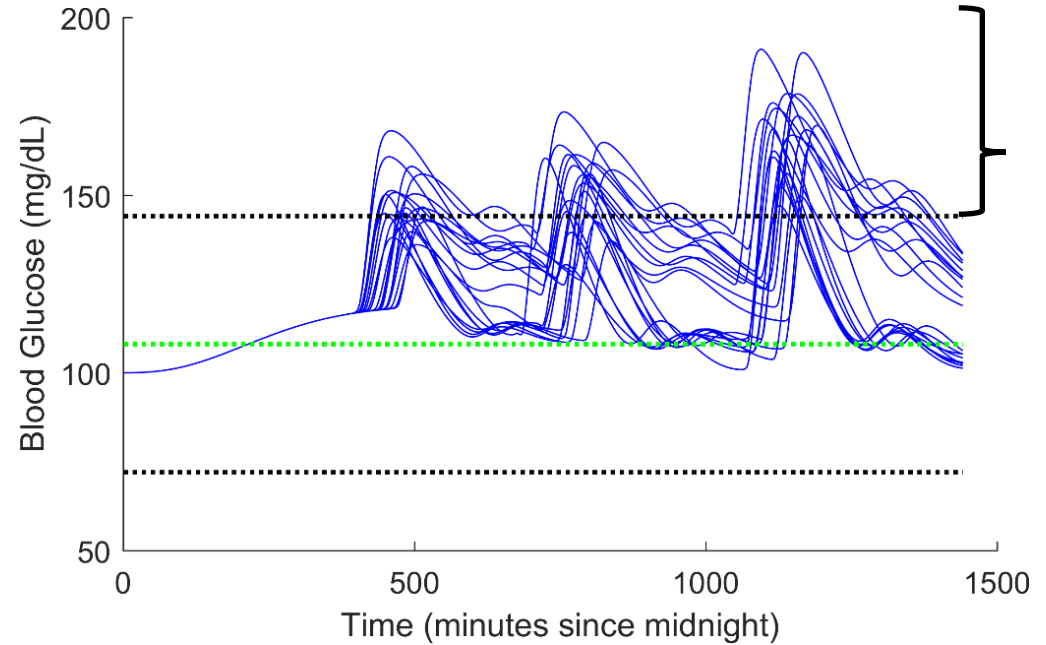
Blood Glucose  
(sugar)



Eat Carbohydrates

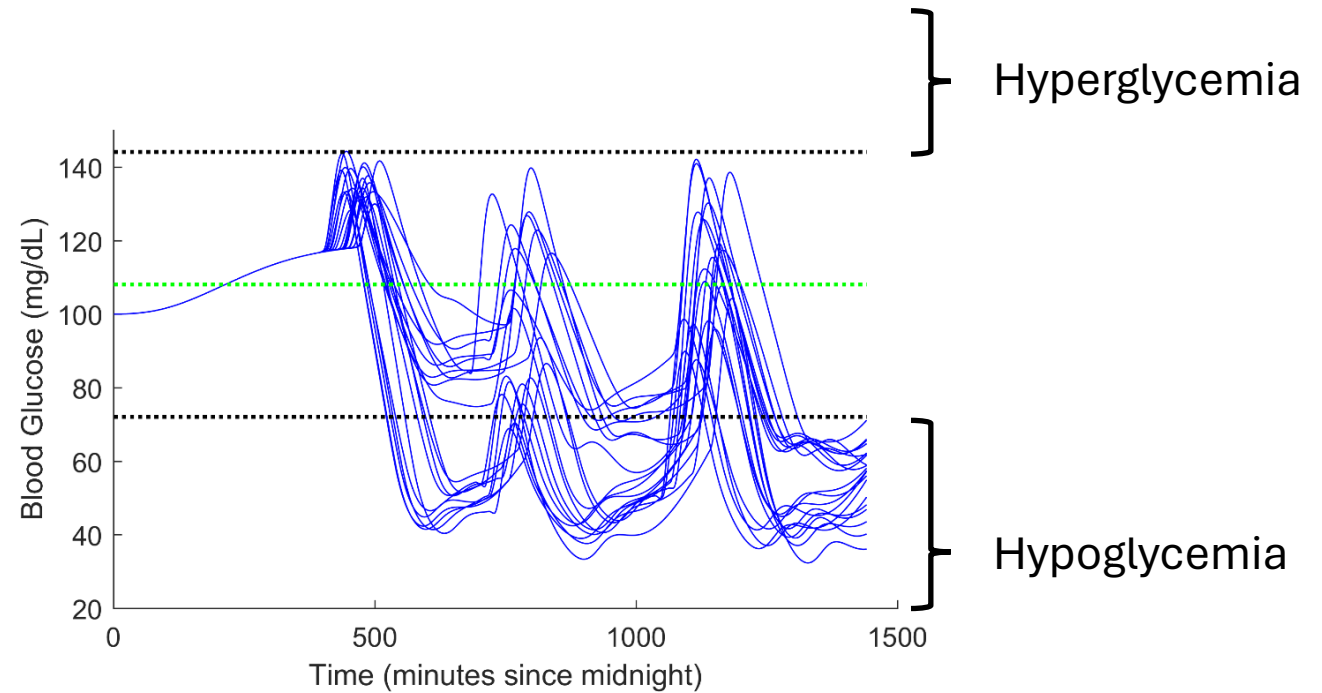
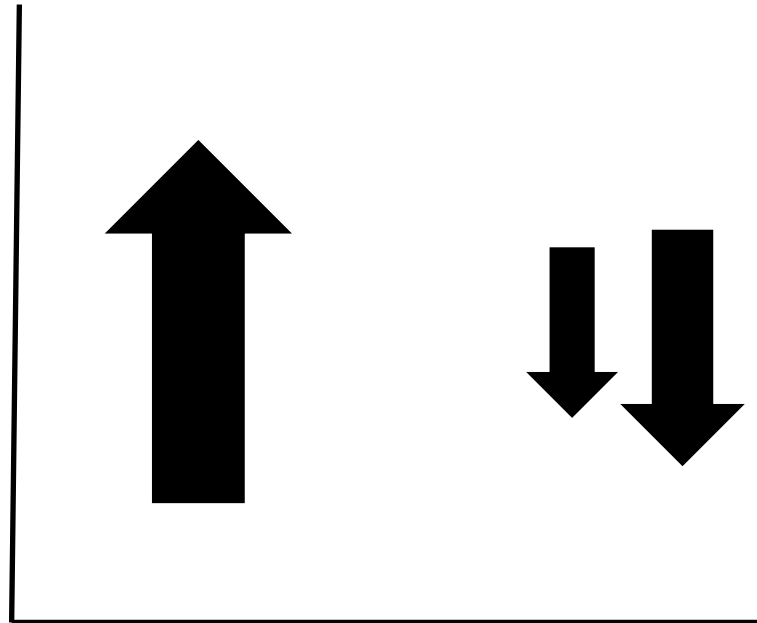
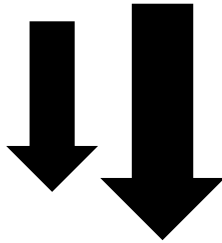
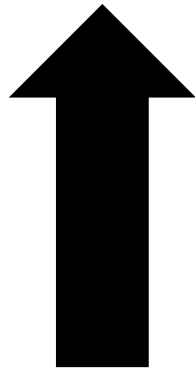


Release Insulin

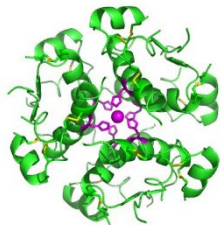


# Example Applications: Diabetes Treatment

Blood Glucose  
(sugar)



Eat Carbohydrates

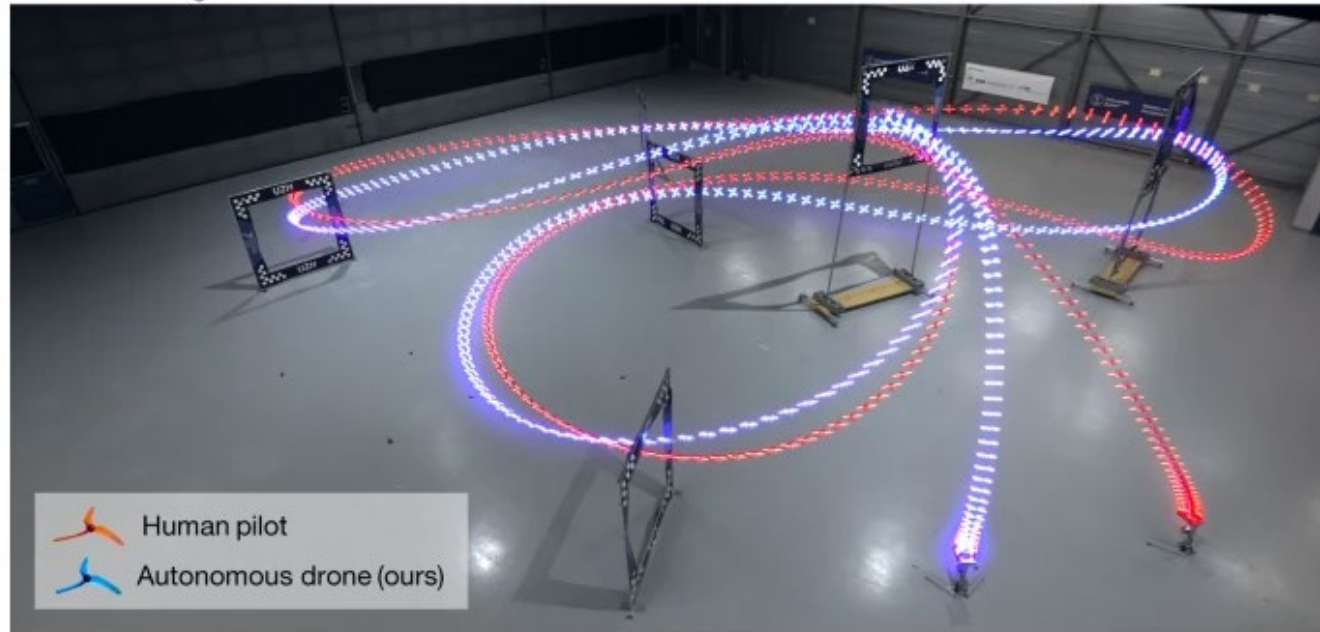


Release Insulin

$$\text{injection} = \frac{\text{blood glucose} - \text{target blood glucose}}{CF} + \frac{\text{meal size}}{CR}$$

## Fig. 1: Drone racing.

**a** Drone racing: human versus autonomous



**b** Head-to-head competition



**c** Human champions



**a**, Swift (blue) races head-to-head against Alex Vanover, the 2019 Drone Racing League world champion (red). The track comprises seven square gates that must be passed in order in each lap. To

# Sepsis Treatment

## Sepsis FACTS

Sepsis arises when the body's response to an infection injures its own tissues and organs. It may lead to shock, multi-organ failure, and death - especially if not recognized early and treated promptly. Sepsis is the final common pathway to death from most infectious diseases worldwide, including viruses such as SARS-CoV-2.

**47 - 50 million**  
cases  
per year<sup>1</sup>

At least  
**11 million**  
deaths per year<sup>2</sup>

**1 in 5 deaths**  
worldwide is  
associated  
with **sepsis**<sup>3</sup>

### Sepsis is the number

**1**

- Cause of death in hospitals<sup>4</sup>
- Cause for hospital readmissions<sup>5</sup>
- Healthcare cost<sup>6</sup>  
(e.g. \$62 billion is spent on sepsis healthcare costs in the US alone)

Up to **50%**  
of sepsis survivors  
suffer from long-term  
physical and/or  
psychological effects<sup>7</sup>

**40%**  
of cases  
are  
children  
under 5<sup>8</sup>

**80%**  
of sepsis cases  
occur  
**outside**  
of a hospital<sup>9</sup>

**SEPSIS**  
is always caused by an  
**infection**  
like pneumonia or  
diarrheal illness<sup>10</sup>

**SEPSIS** is a medical  
**emergency** - if you or someone  
you know shows signs of sepsis,  
seek medical care immediately.  
Every hour counts.<sup>11</sup>

### These signs may indicate sepsis:

**Slurred Speech**  
or Confusion

**Extreme Shivering**  
or Muscle Pain  
/Fever

**Passing No**  
Urine  
All Day

**Severe**  
Breathlessness

**It Feels Like**  
You're Going  
to Die

**Skin Mottled**  
or Discolored

### September 13

You can help  
**#StopSepsis**  
and  
**#SaveLives**  
Get involved at  
**worldsepsisday.org**

### References

1. 2. 3. 8. 10 Rudd et al. [https://doi.org/10.1016/S0140-6736\(19\)32989-7](https://doi.org/10.1016/S0140-6736(19)32989-7).
- 4 Rhee et al. <https://jamanetwork.com/journals/jamanetworkopen/fullarticle/2724768>
- 5 Torio et al. <https://www.hcup-us.ahrq.gov/reports/statbriefs/sb204-Most-Expensive-Hospital-Conditions.pdf>
- 6 Buchman et al. [https://journals.lww.com/ccmjournal/Fulltext/2020/03000/Sepsis\\_Among\\_Medicare\\_Beneficiaries\\_\\_3\\_\\_The\\_4.aspx](https://journals.lww.com/ccmjournal/Fulltext/2020/03000/Sepsis_Among_Medicare_Beneficiaries__3__The_4.aspx)
- 7 Prescott et al. <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5839473/>
- 9 Rhee et al. <https://jamanetwork.com/journals/jama/fullarticle/2654187>
- 11 Seymour et al. <https://www.nejm.org/doi/10.1056/NEJMoa1703058>

# Warehouse Robotics

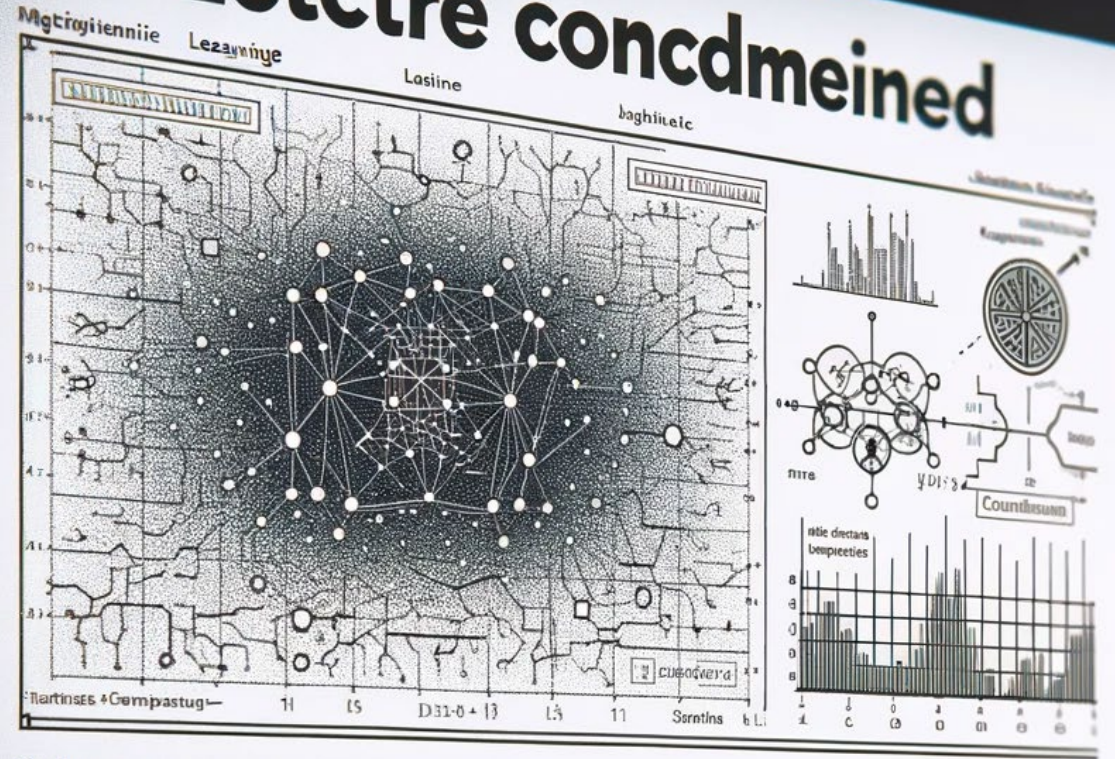


# Note

- Despite the many *proposed* and *simulated* examples of important RL problems, there are extremely few examples of RL agents actually learning and interacting with real environments in useful ways.
  - There are examples of RL agents learning in simulation, and the learned policies being useful (e.g., Loon balloons).
- A very restricted class of RL algorithms is used for fine-tuning LLMs.
  - These consider non-sequential problems.
  - These consider a special setting that can be reduced to a supervised learning problem and solved with supervised learning methods (DPO).
- Why?
  - RL is challenging to get working, as we will see!

End

# Letctre concdmeined



Dgoinnbnic



Mbcine Learning

# Thank you.

