General Principles of Human and Machine Learning

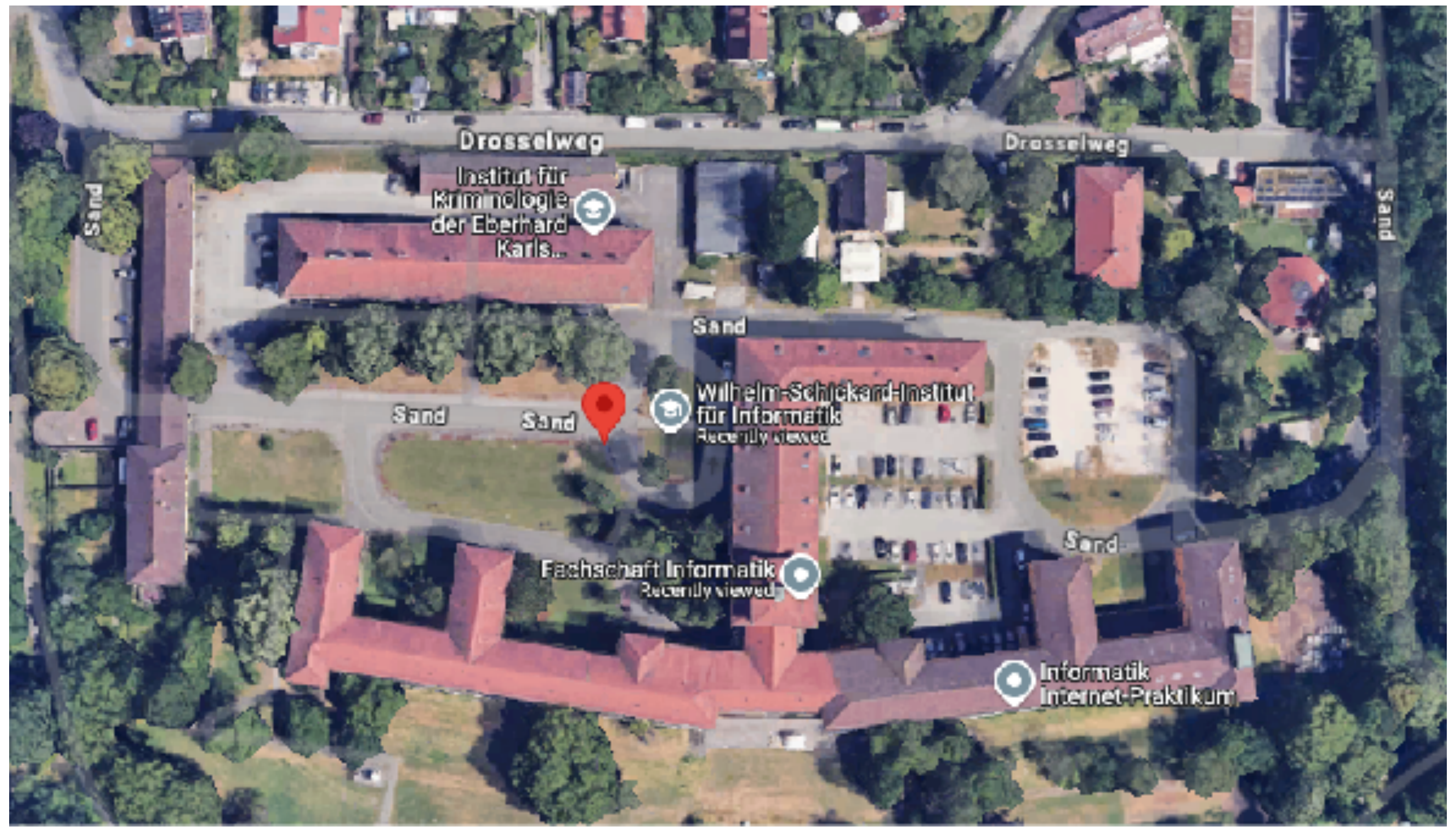
Lecture 12: General Principles

Dr. Charley Wu

https://hmc-lab.com/GPHML.html

Exam

- Combination of multiple choice and short answer questions
  - No complex calculations are needed 
  - No need to memorize formulas or dates
  - Focus on understanding the main theoretical ideas and how they connect across fields
  - Bring pens/pencils
- First taking: Friday, Feb 21st, 13:00 -15:00
  - Hörsaal 1, F119 (SAND 6/7)
- Second taking: Friday April 11th, 12:00 –14:00
  - Ground floor lecture room, AI building (Maria-von-Linden-Str. 6)



Revisiting our original questions

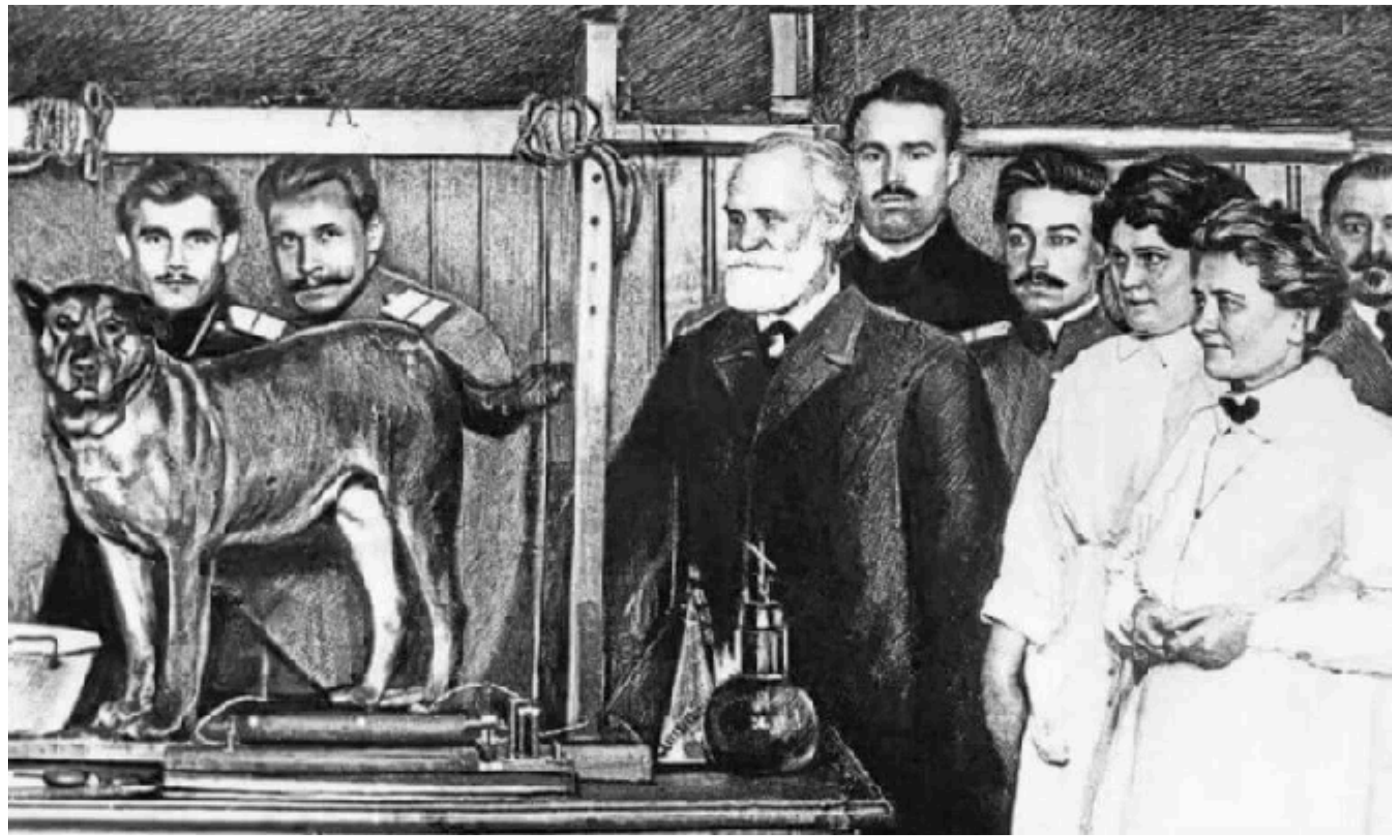
What are the guiding principles of human and machine learning?

How have these two fields informed one another?

Which mechanisms of learning are shared across fields?

Where have we seen convergence?

Foundations of Biological Learning



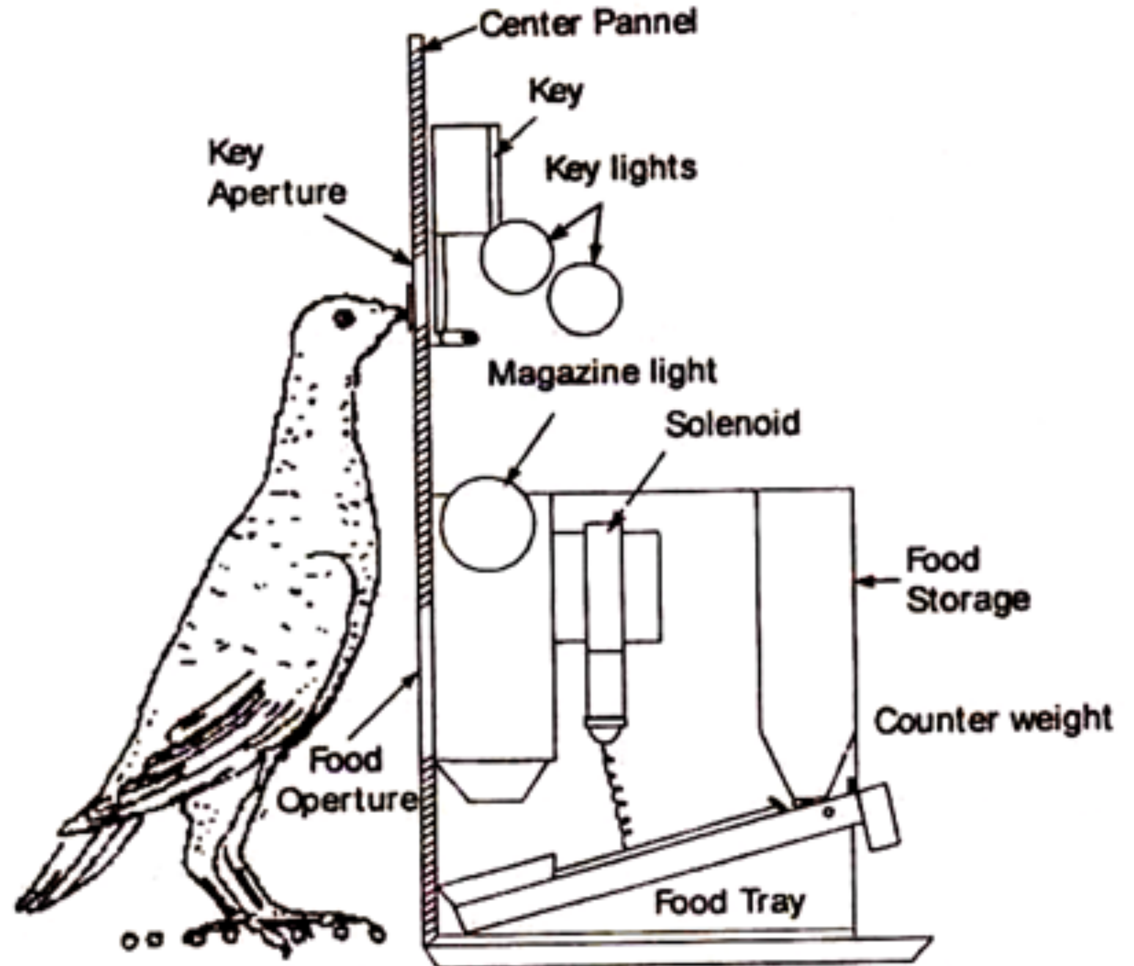
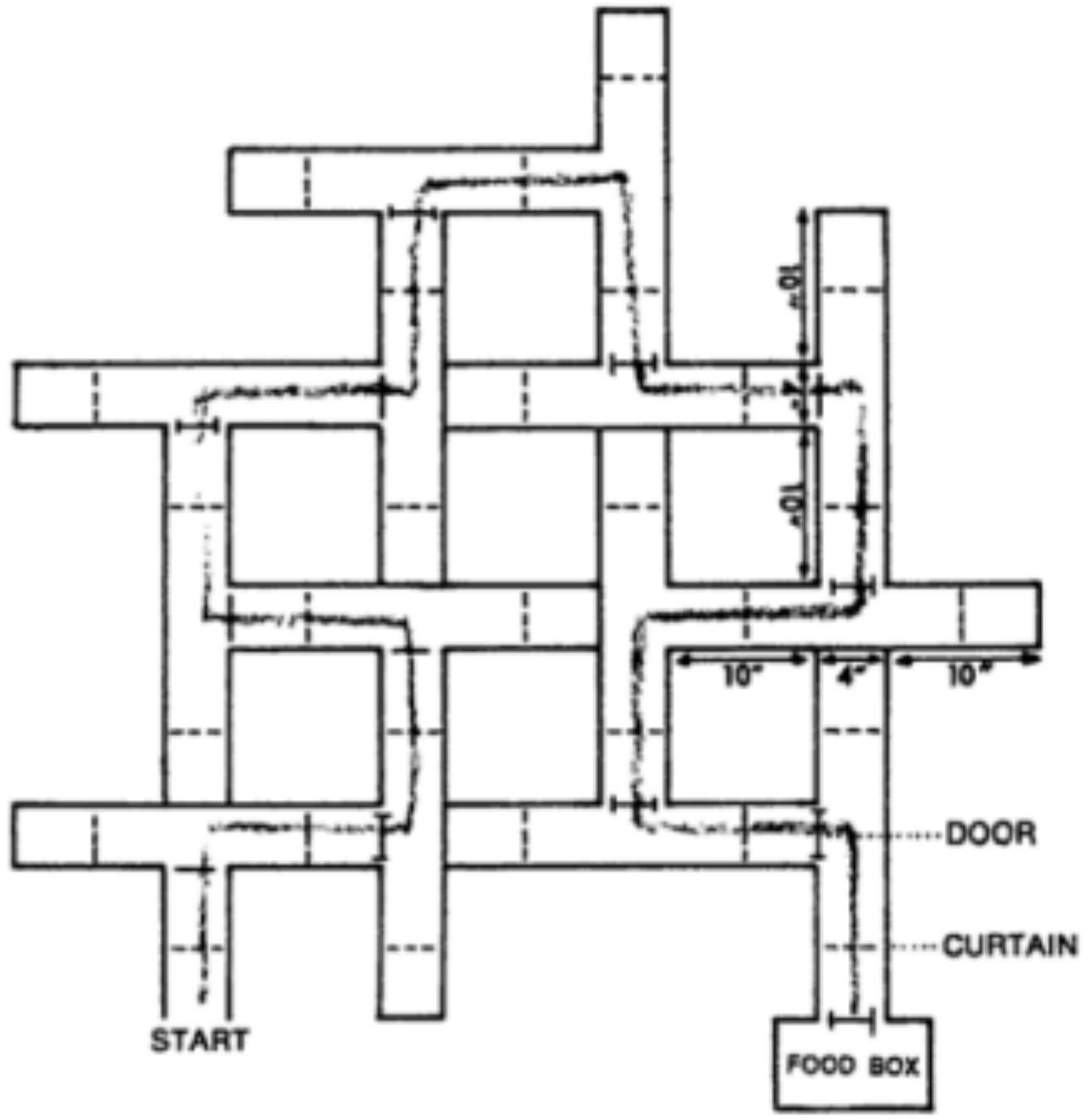
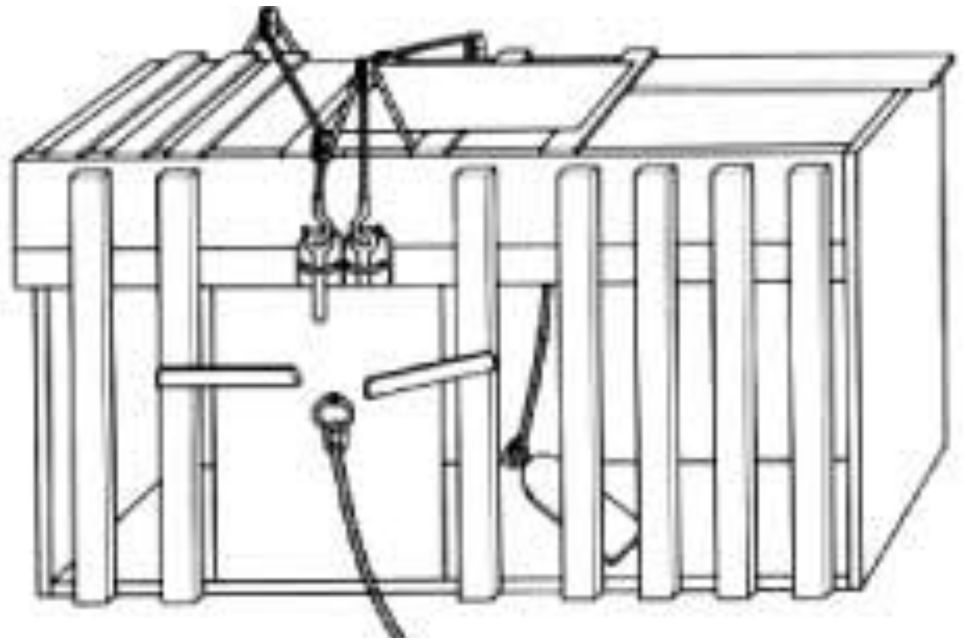


Illustration. Skinner box as adapted for the pigeon.



A brief timeline of early research on biological learning

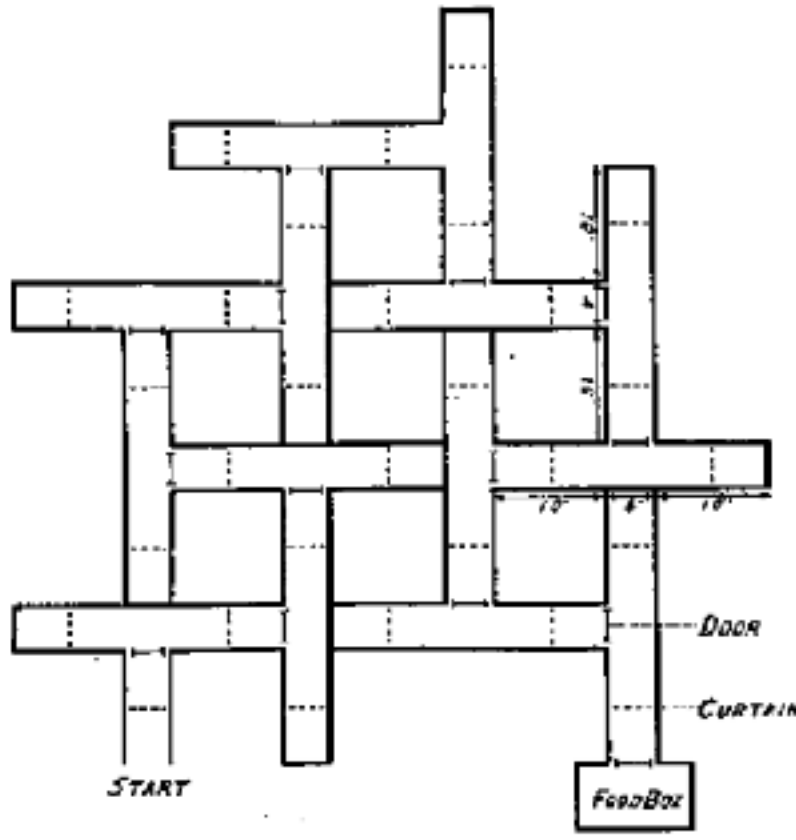


Thorndike (1911)

Pavlov (1927)

Skinner (1938)

Tolman (1948)



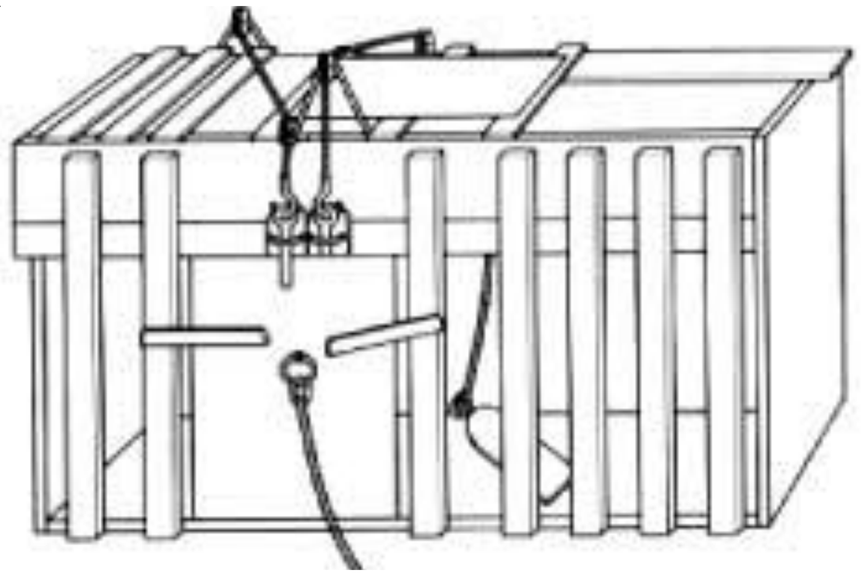
Plan of maze
14-Unit T-Alley Maze

Fig. 1

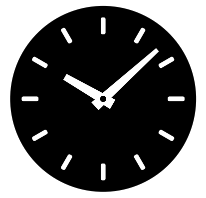
(From M. H. Elliott, 'The effect of change of reward on the maze performance of rats.' *Univ. Calif. Publ. Psychol.*, 1928, 4, p. 20.)

Thorndike's Laws

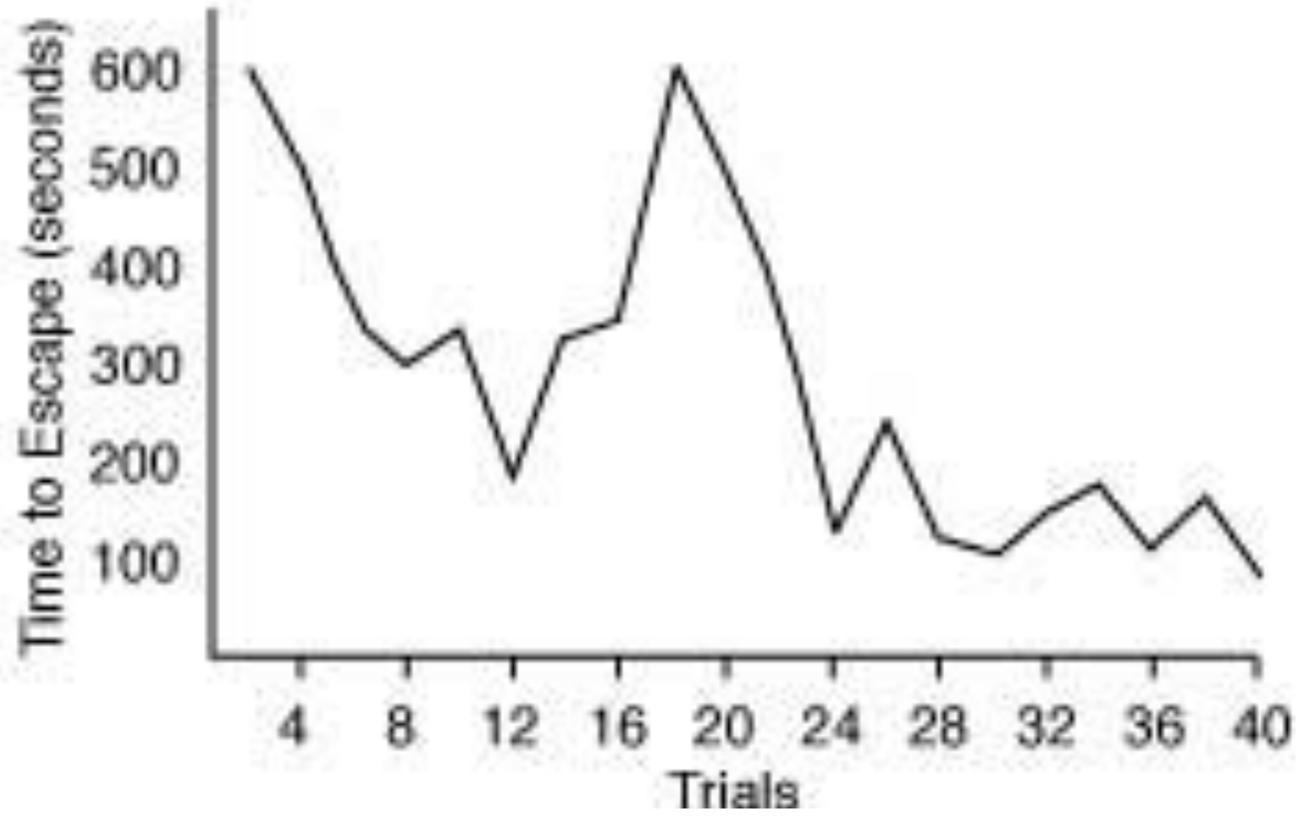
Cat



Puzzle Box



Time to escape



Law of Effect

Actions associated with satisfaction are strengthened, while those associated with discomfort become weakened

Law of Exercise

Any response to a stimulus will be strengthened proportional to how often it has been associated in the past

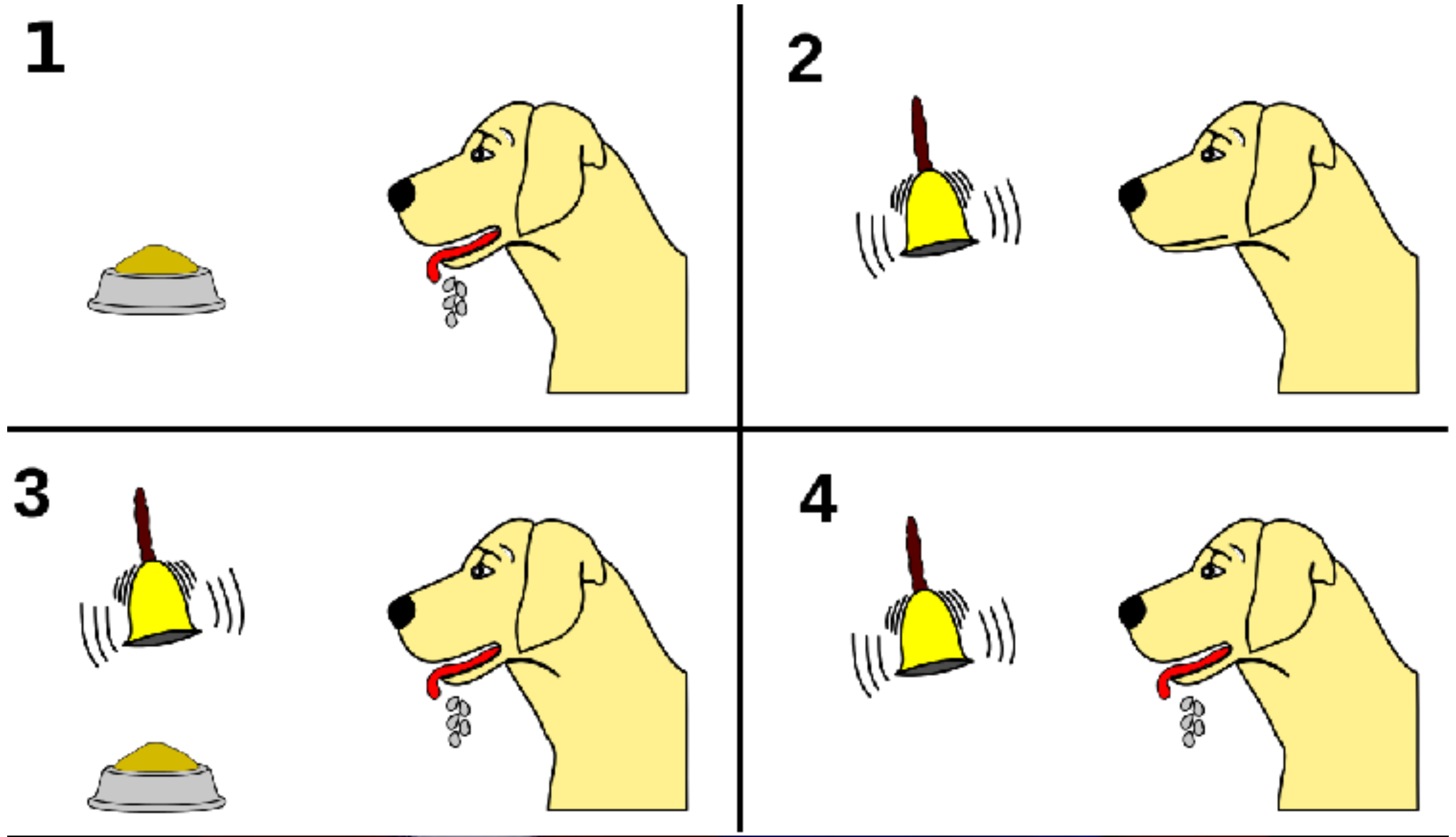
Classical and Operant Conditioning

Classical Condition (Pavlov, 1927)

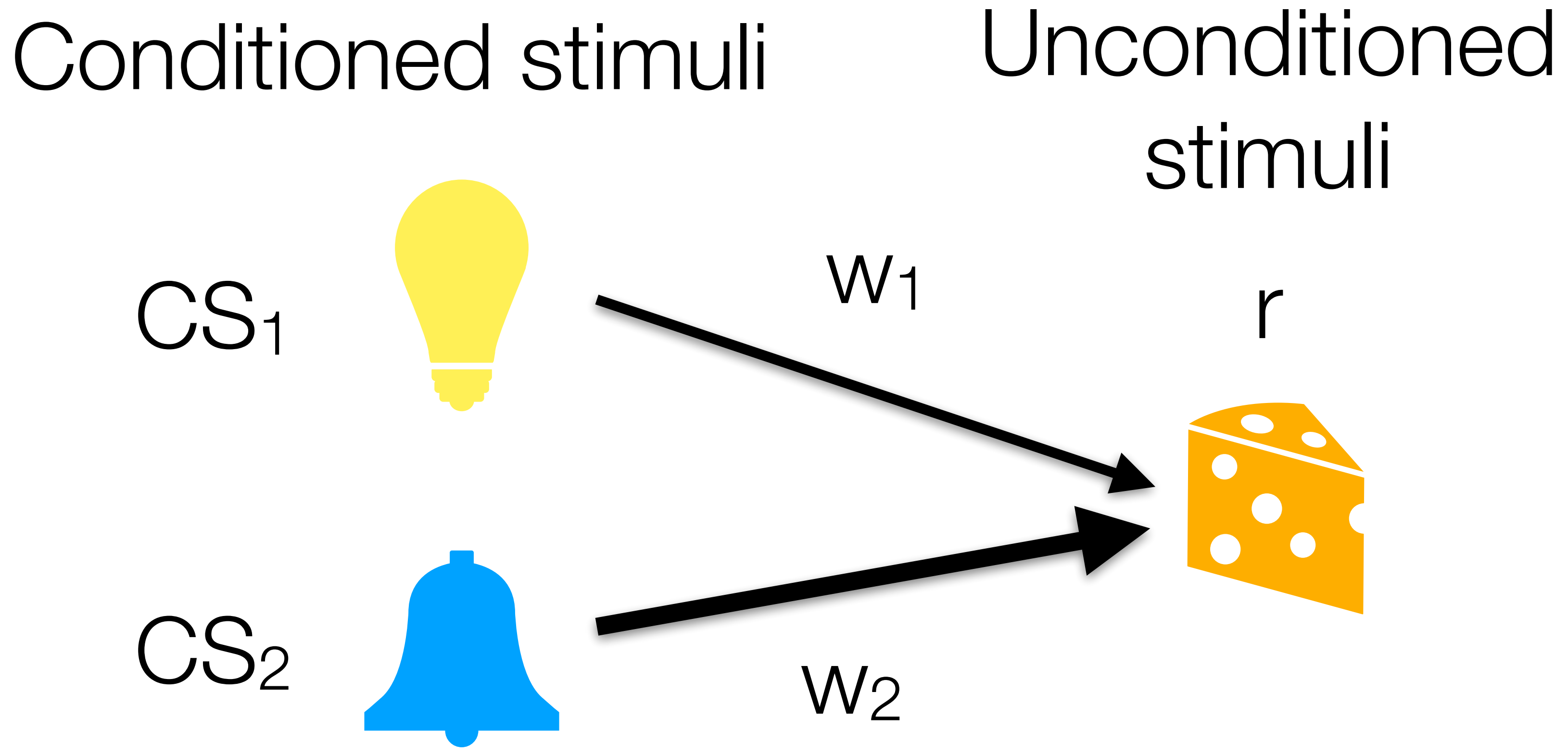
Learning as the *passive* coupling of stimulus (bell ringing) and response (salivation), anticipating future rewards

Operant Condition (Skinner, 1938)

Skinner (1938): Learning as the *active* shaping of behavior in response to rewards or punishments



Rescorla-Wagner



Rescorla-Wagner model

(Bush & Mosteller, 1951; Rescorla & Wagner, 1972)

Reward prediction

$$\underbrace{\hat{r}_t}_{\text{Reward expectation}} = \sum_i \underbrace{\mathrm{CS}_i^t}_{\text{CS i on trial t}} \underbrace{w_i}_{\text{Associative strength or weight}}$$

Weight update for i where $CS_i = 1$:

$$w_i \leftarrow w_i + \underbrace{\eta}_{\text{Learning rate}} \underbrace{(\underbrace{r_t}_{\text{Observed outcome}} - \underbrace{\hat{r}_t}_{\text{Predicted outcome}})}_{\delta}$$

Reward prediction error (RPE)

RW Model

- Reward prediction is the sum of CS stimuli x weights
- Weights are updated via the delta-rule

The delta-rule of learning:

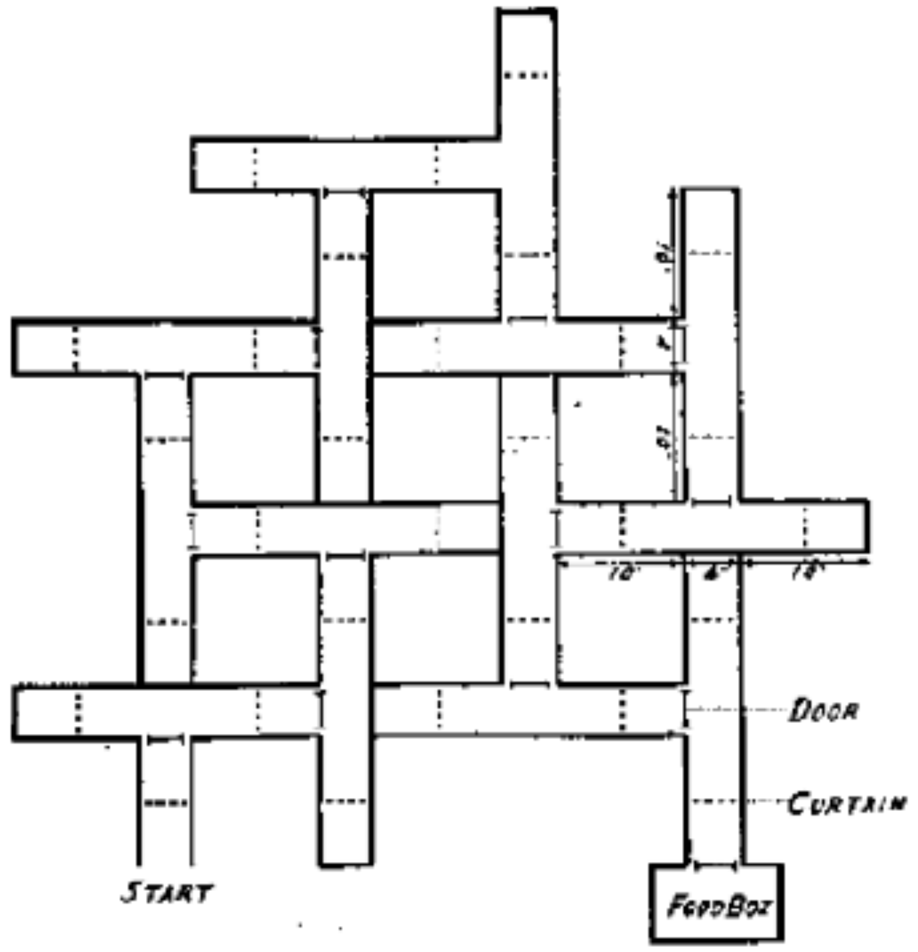
- Learning occurs only when events violate expectations ($\delta \neq 0$)
- The magnitude of the error corresponds to how much we update our beliefs

Tolman and Cognitive maps

- Learning is not just a telephone switchboard connecting incoming sensory signals to outgoing responses (S-R Learning)
- Rather, “latent learning” establishes something like a “field map of the environment” gets etablished (S-S learning)

Stimulus-Response (S-R) Learning

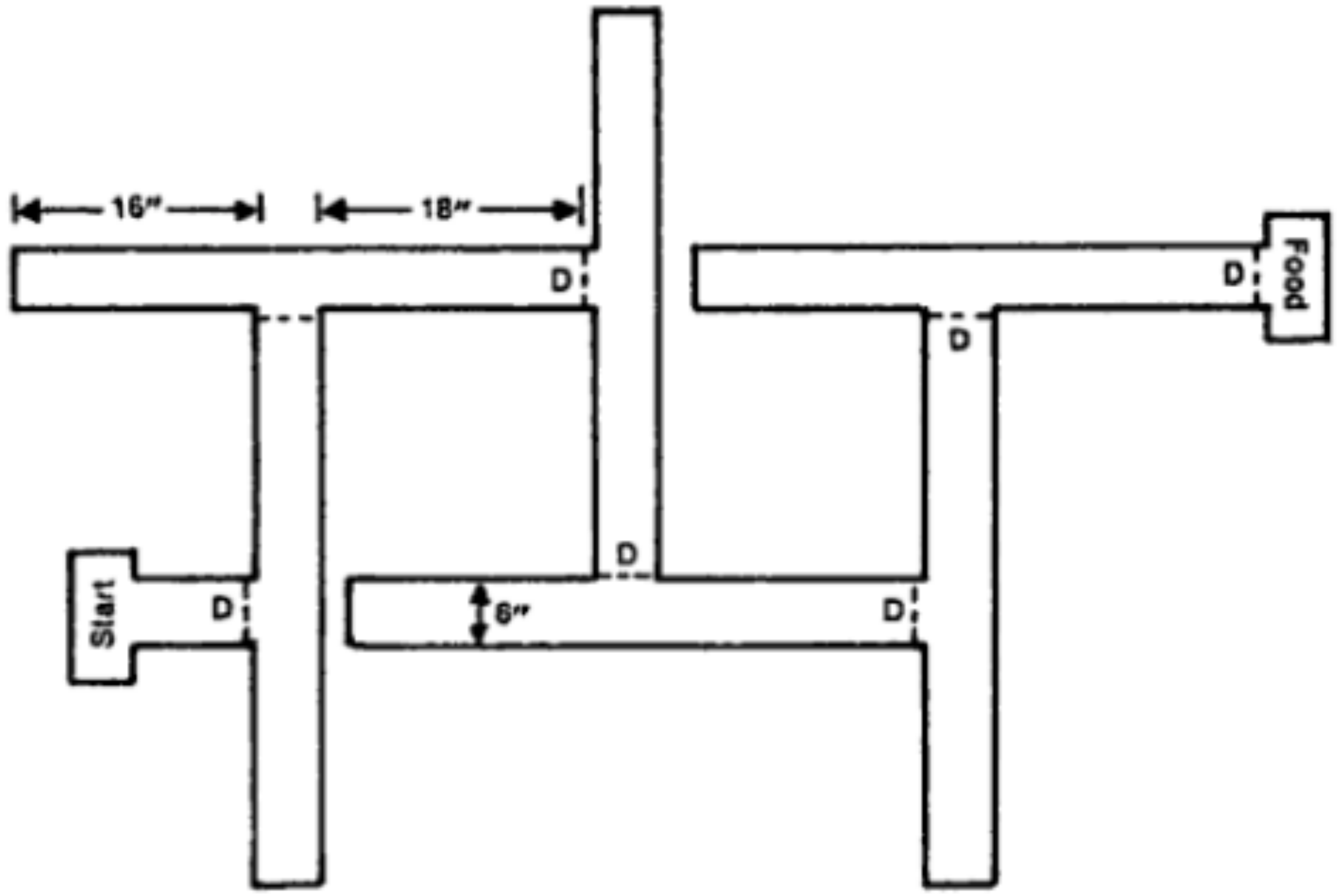
Stimulus-Stimulus (S-S) Learning



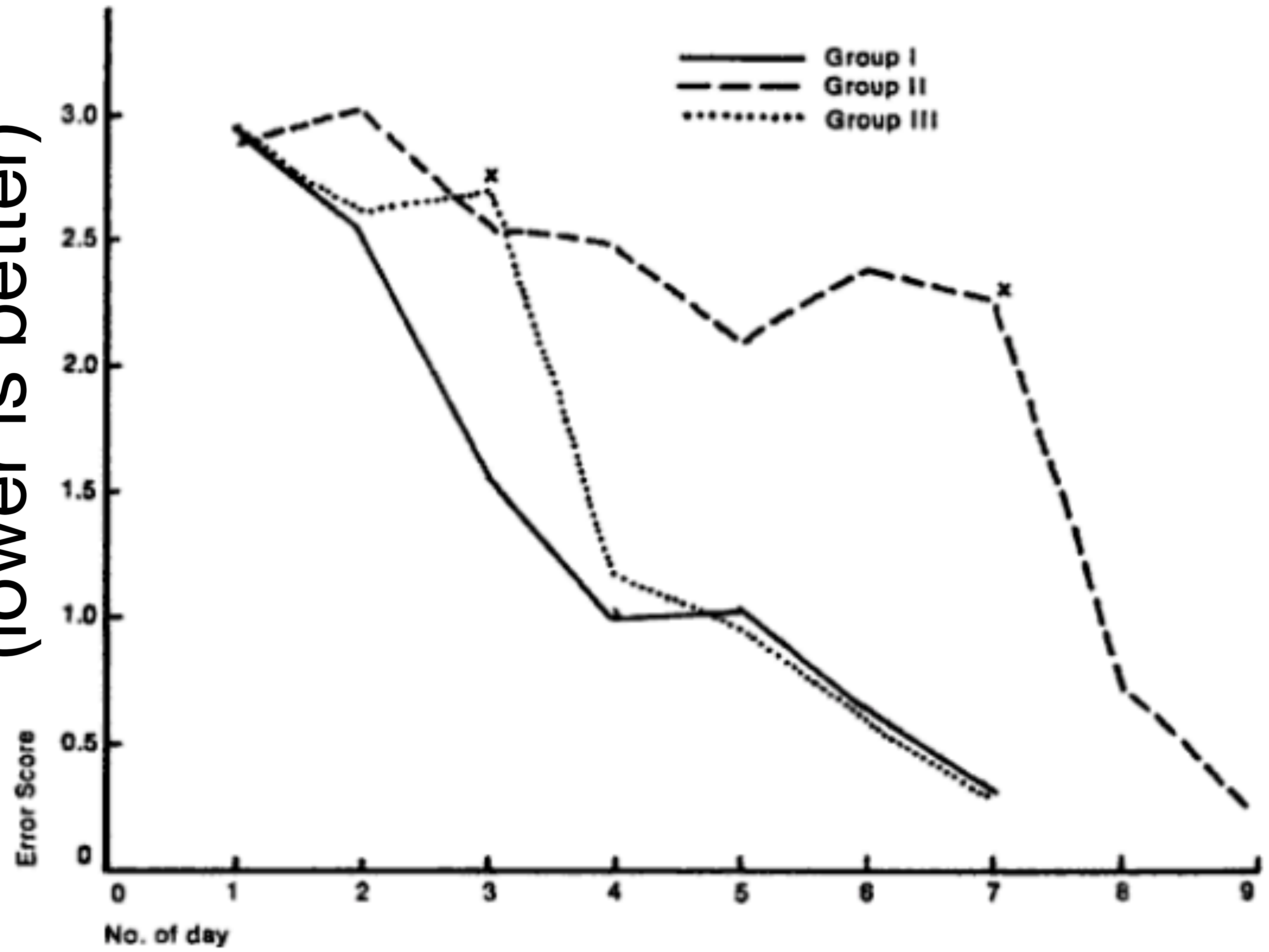
Plan of maze
14-Unit T-Alley Maze
Fig. 1
(From M. H. Elliott, The effect of change of reward on the maze performance of rats. *Univ. Calif. Publ. Psychol.*, 1928, 4, p. 20.)

Latent Learning

- Blodgett (1929) Maze navigation task
  - **Group 1** [Control]: one trial a day with food in the goal box at the end
  - **Group 2** [Late food] No food in the maze for days 1-6, then food provided at the end on day 7
  - **Group 3** [Early food] … food added on day 3
- Learning curves dropped dramatically when food was added
  - This suggests latent learning prior to reward
  - “They had been building up a ‘map’”
  - Once the reward was added, they could use the map rather than starting from scratch

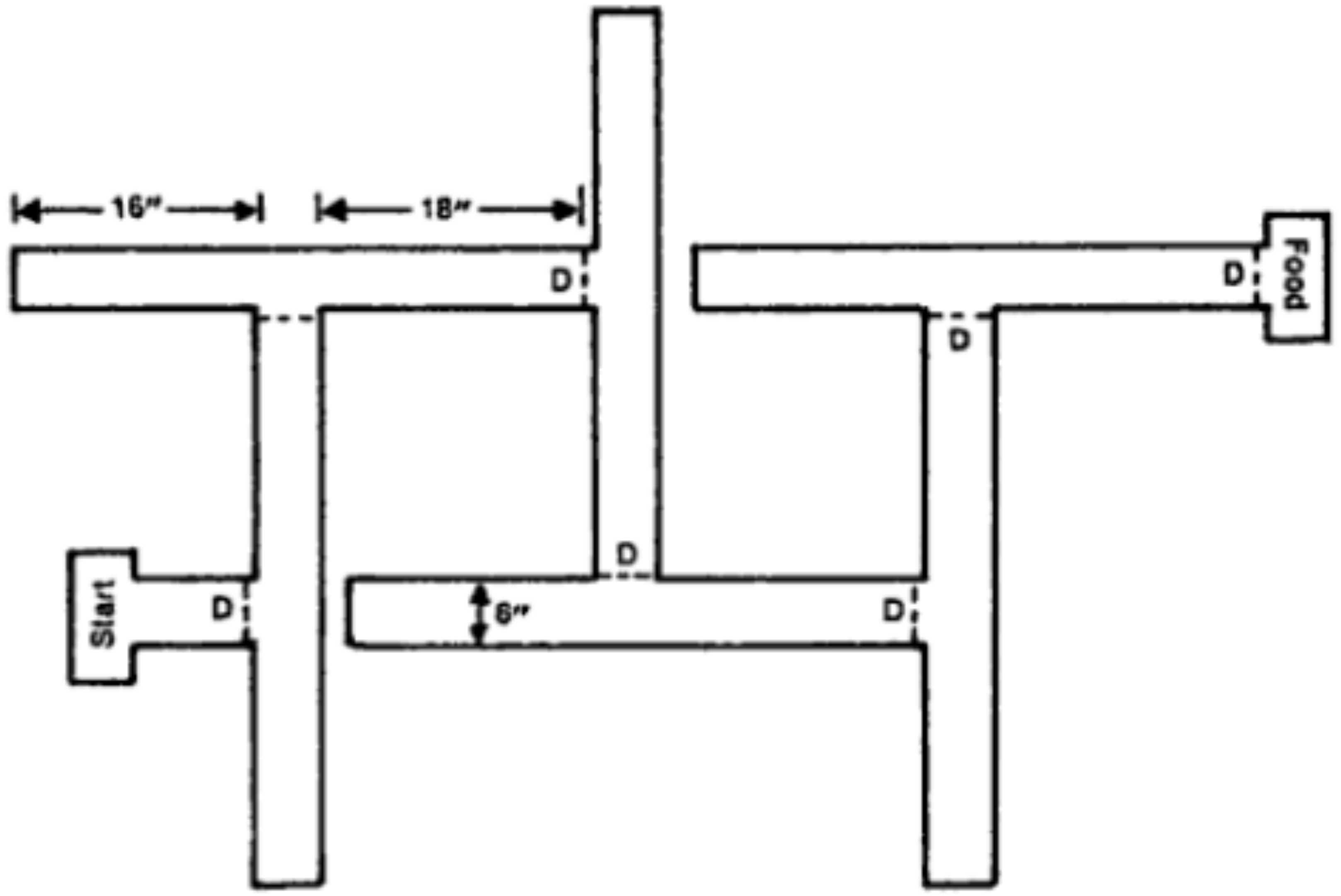


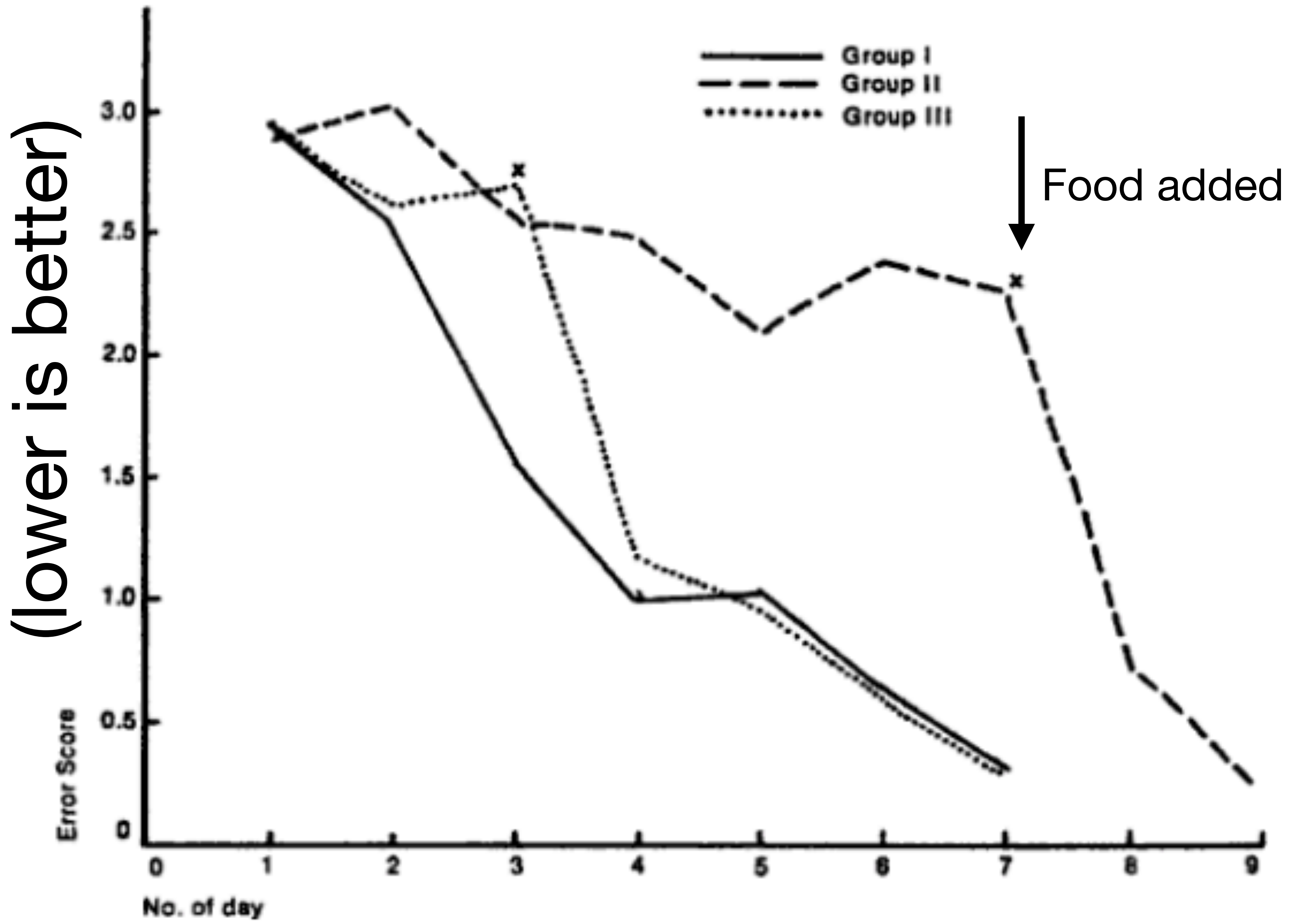
(lower is better)



Latent Learning

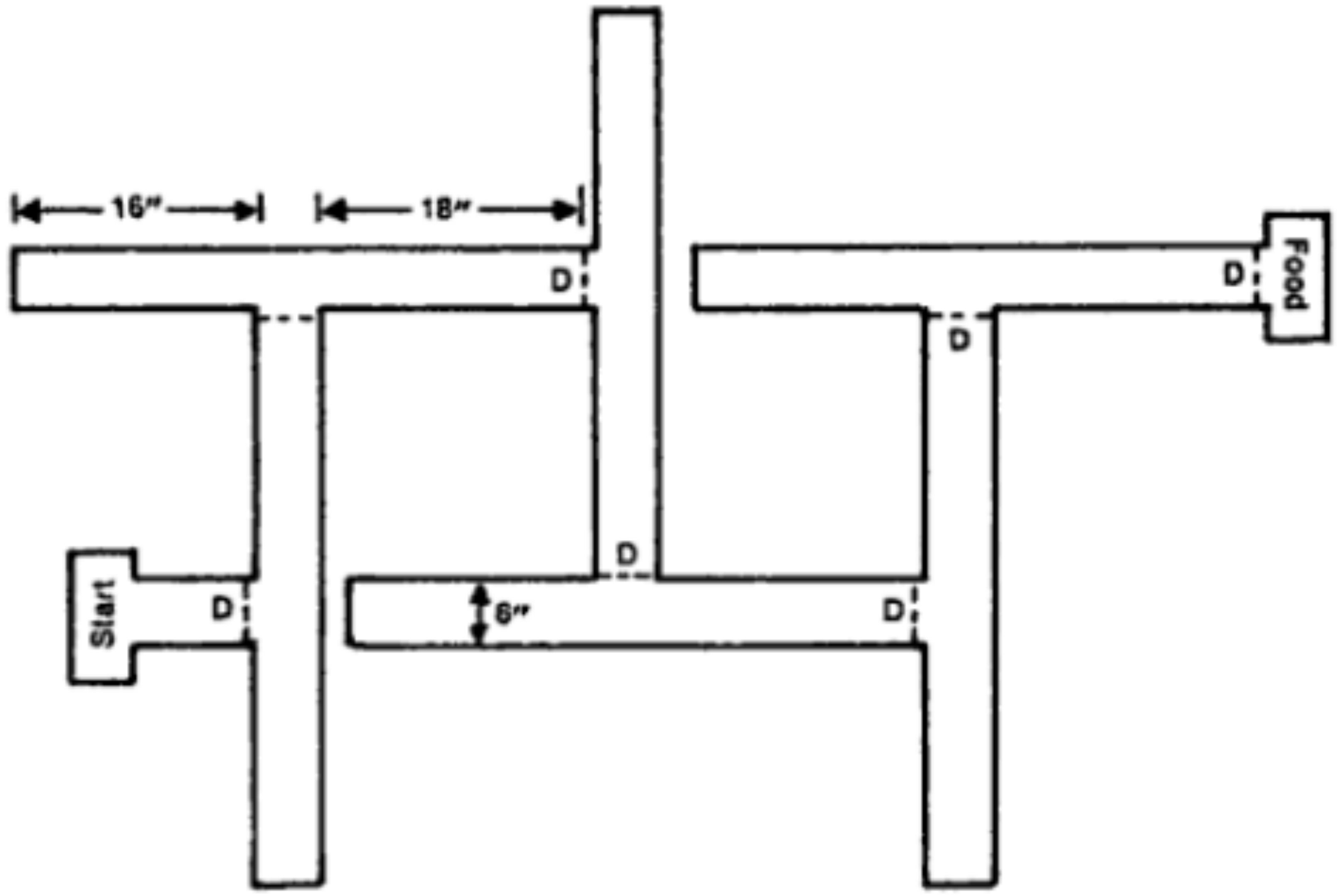
- Blodgett (1929) Maze navigation task
    - **Group 1** [Control]: one trial a day with food in the goal box at the end
    - **Group 2** [Late food] No food in the maze for days 1-6, then food provided at the end on day 7
    - **Group 3** [Early food] … food added on day 3
- Learning curves dropped dramatically when food was added
    - This suggests latent learning prior to reward
    - “They had been building up a ‘map’”
    - Once the reward was added, they could use the map rather than starting from scratch

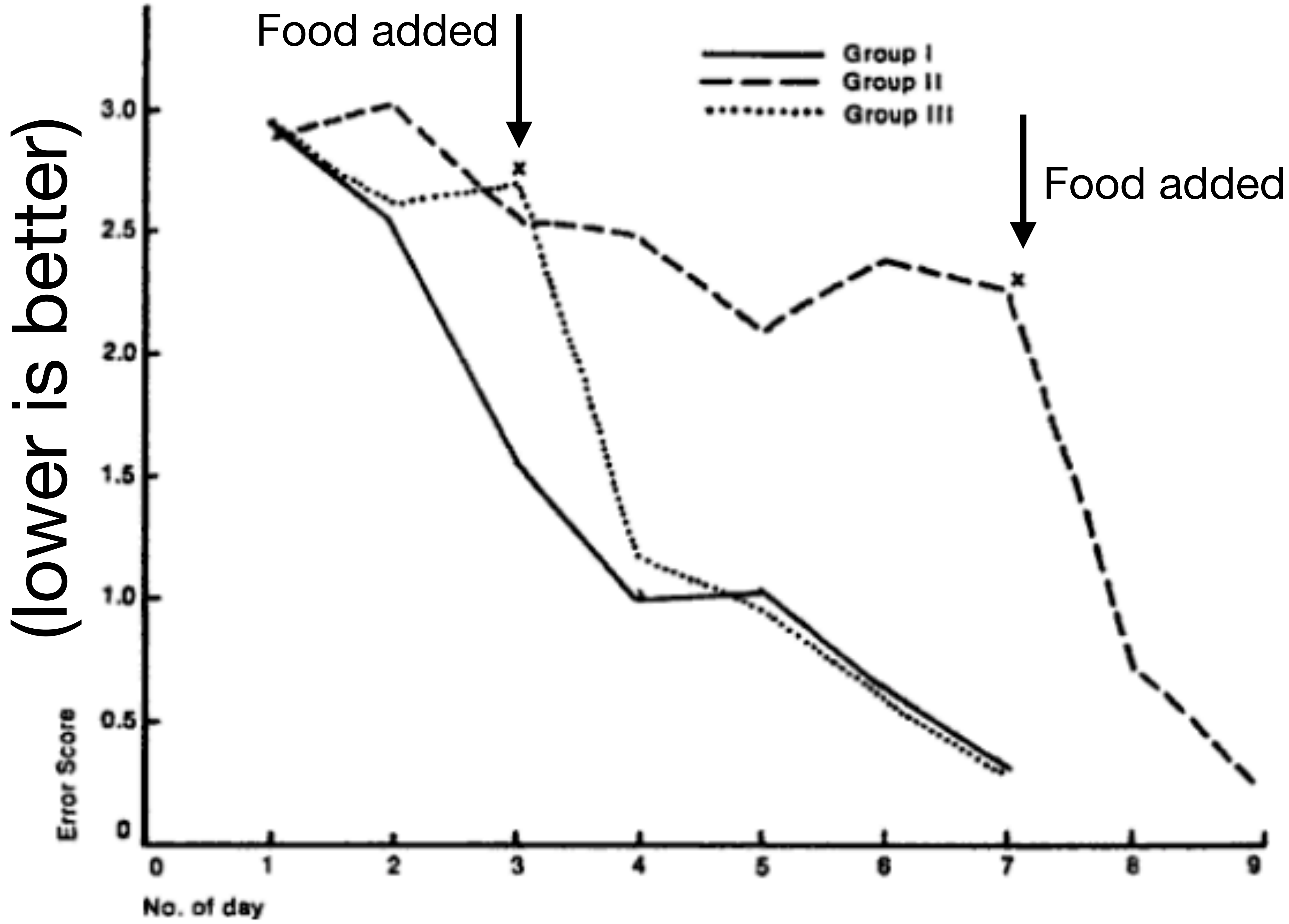




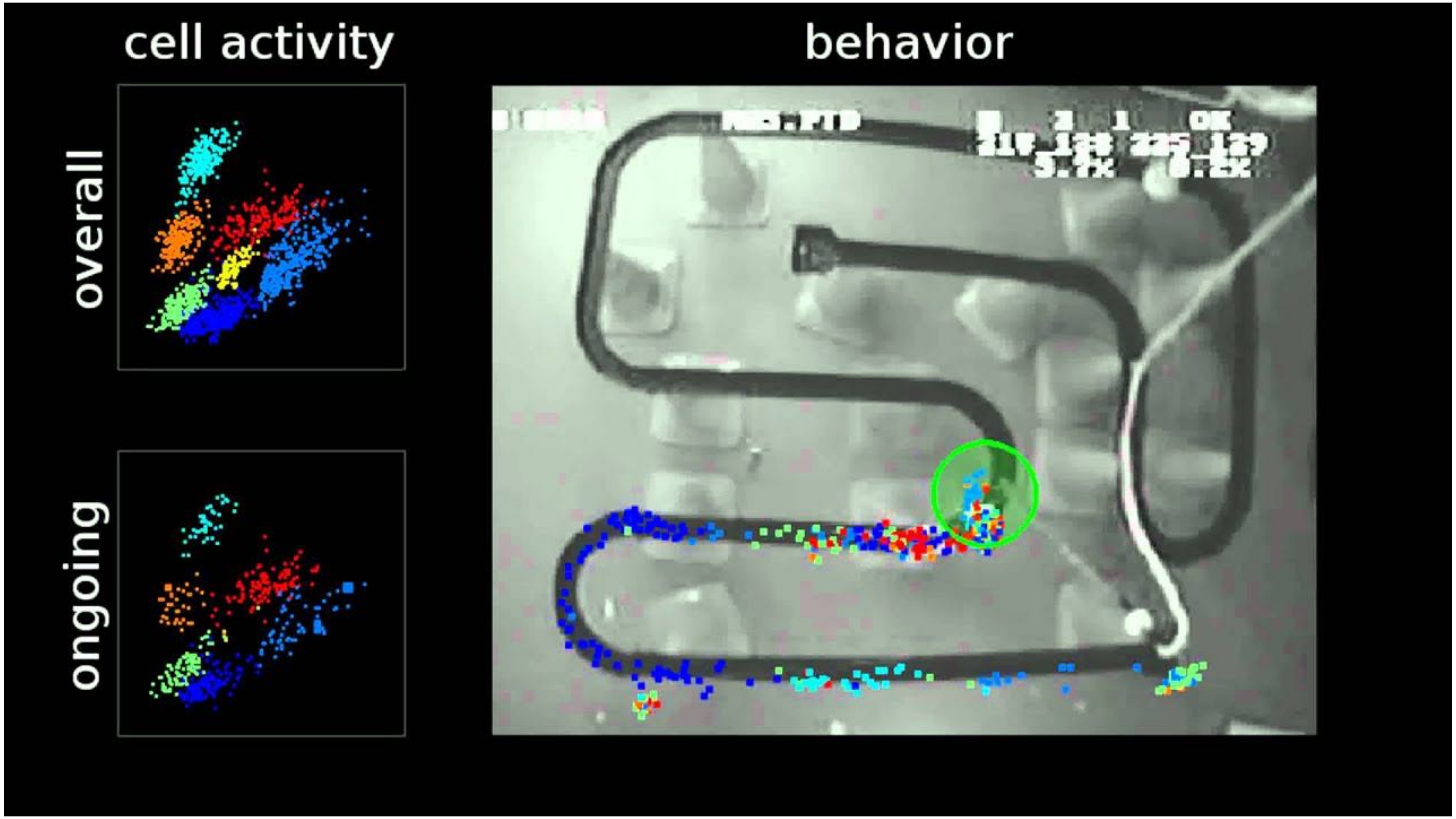
Latent Learning

- Blodgett (1929) Maze navigation task
    - **Group 1** [Control]: one trial a day with food in the goal box at the end
    - **Group 2** [Late food] No food in the maze for days 1-6, then food provided at the end on day 7
    - **Group 3** [Early food] … food added on day 3
- Learning curves dropped dramatically when food was added
    - This suggests latent learning prior to reward
    - “They had been building up a ‘map’”
    - Once the reward was added, they could use the map rather than starting from scratch





Place cells in the hippocampus represent location in an environment



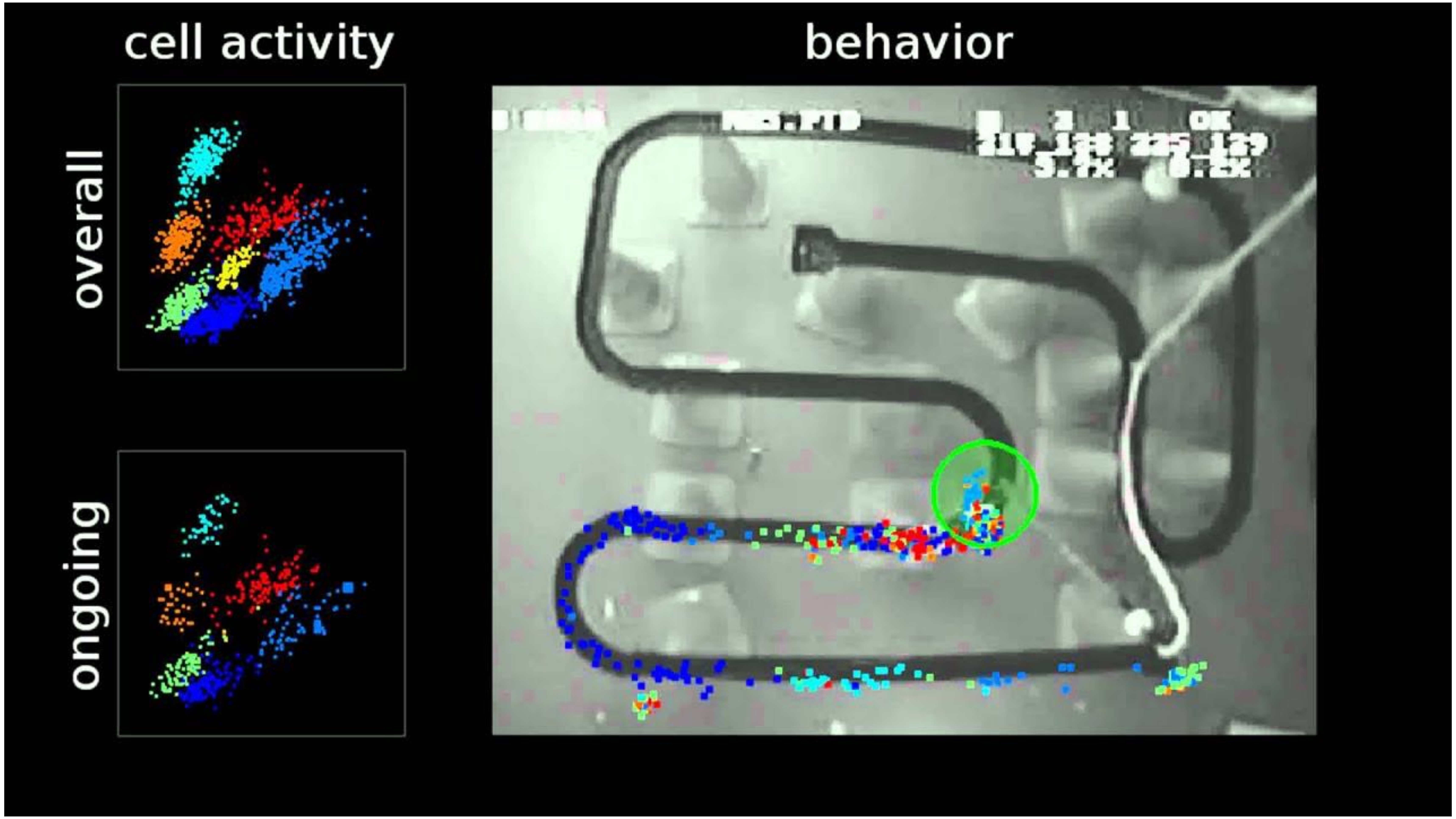
John O'Keefe
Nobel Prize in Physiology or Medicine 2014

Wilson Lab (MIT)

Place cells in the hippocampus represent location in an environment

Place Cell

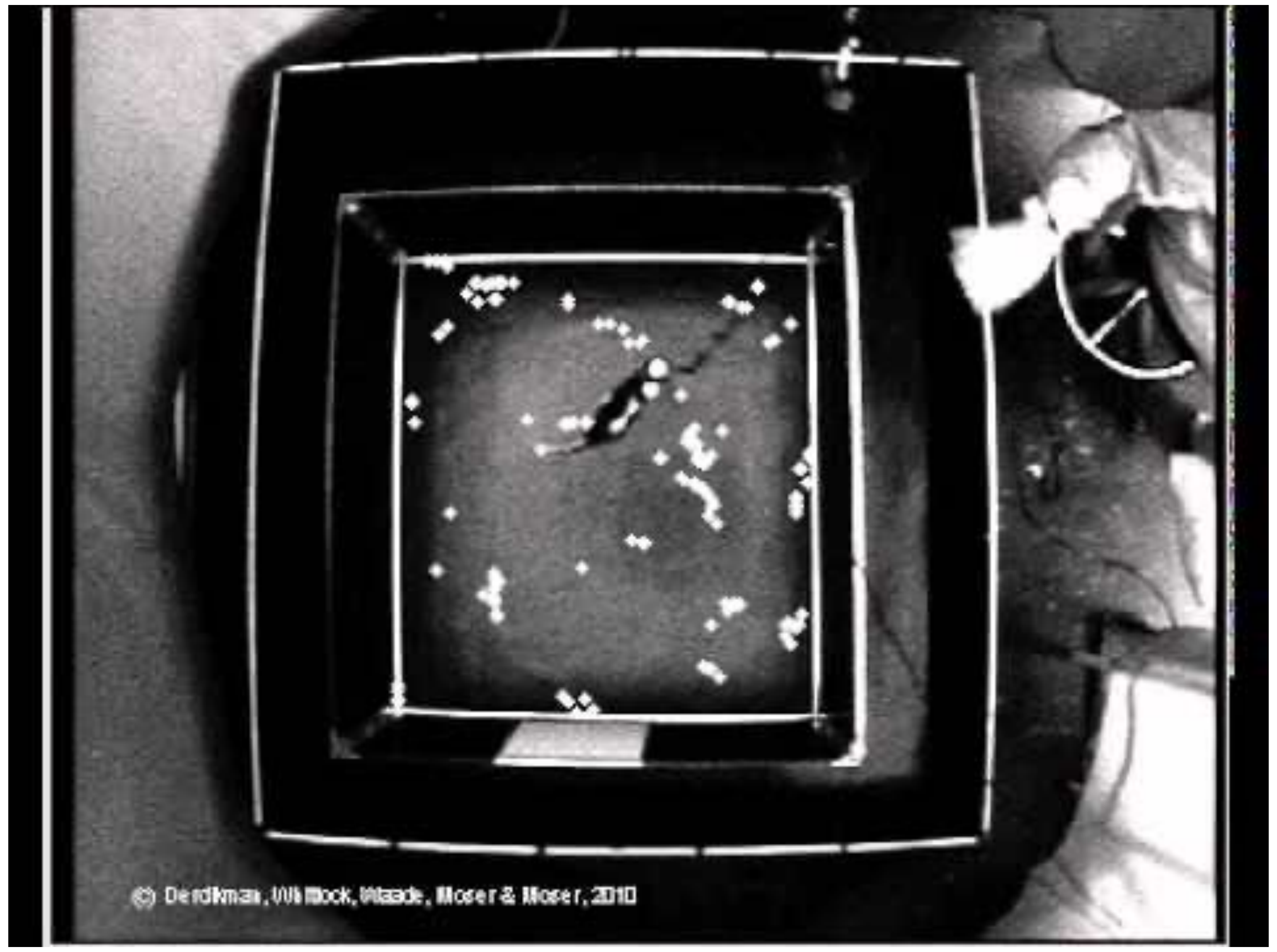
(O'keefe & Nadel 1978)

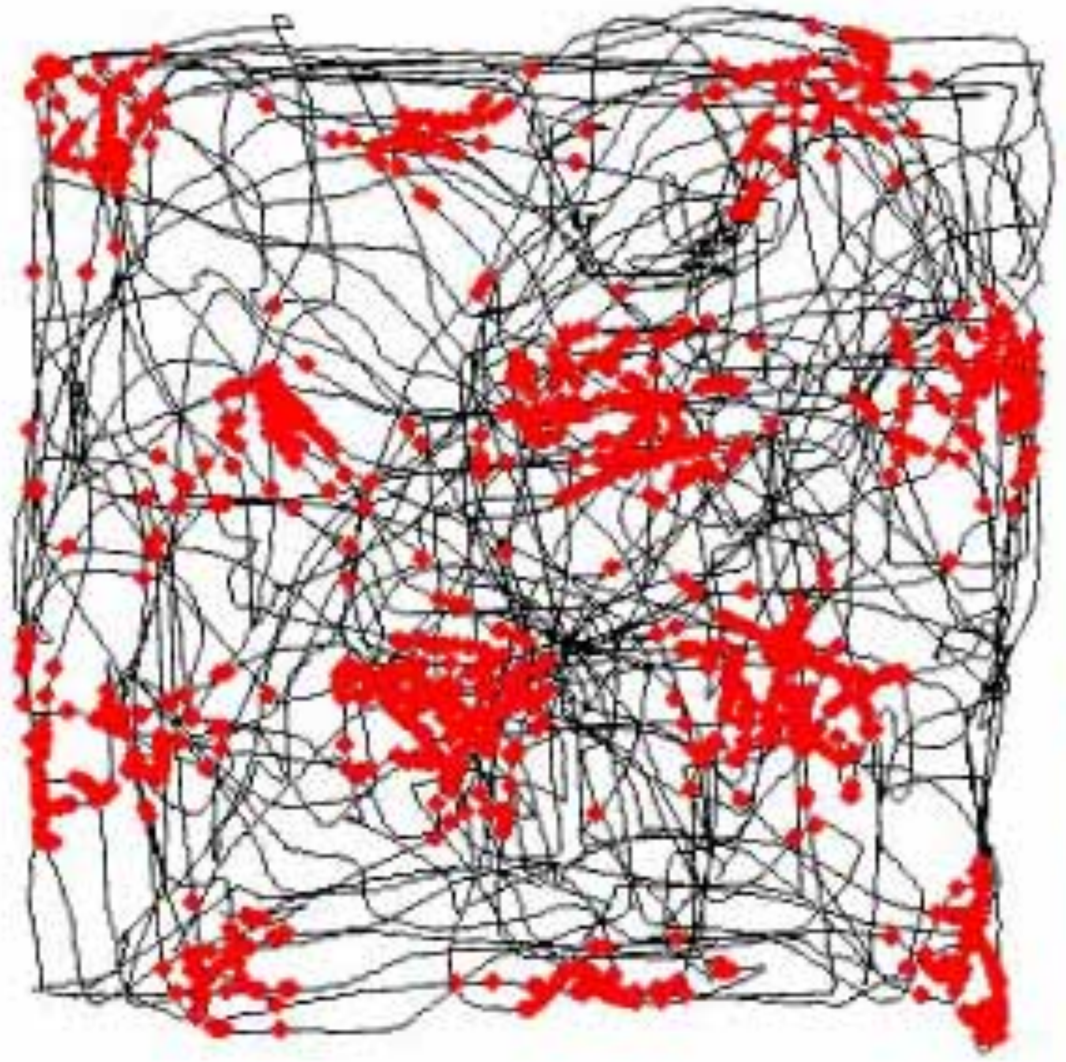


John O'Keefe
Nobel Prize in Physiology or Medicine 2014

Wilson Lab (MIT)

Grid cells in the Entorhinal Cortex provide a coordinate system

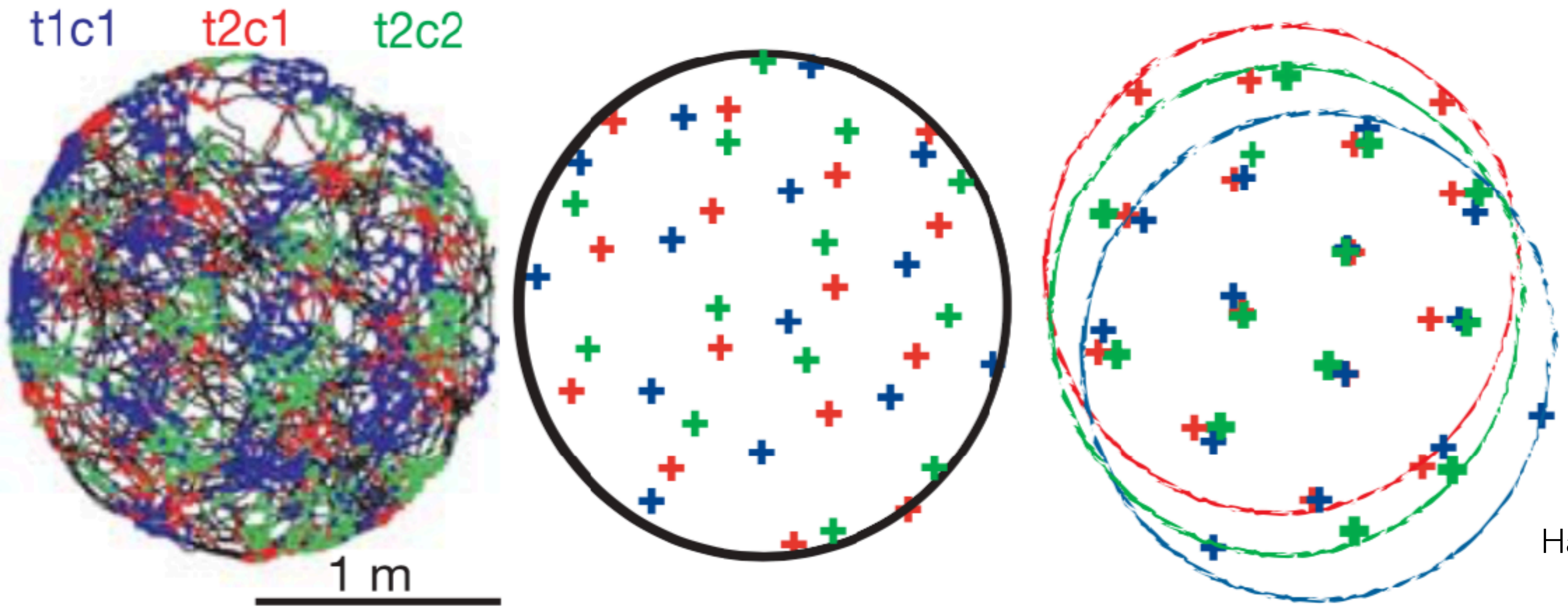




Trajectory

Peaks

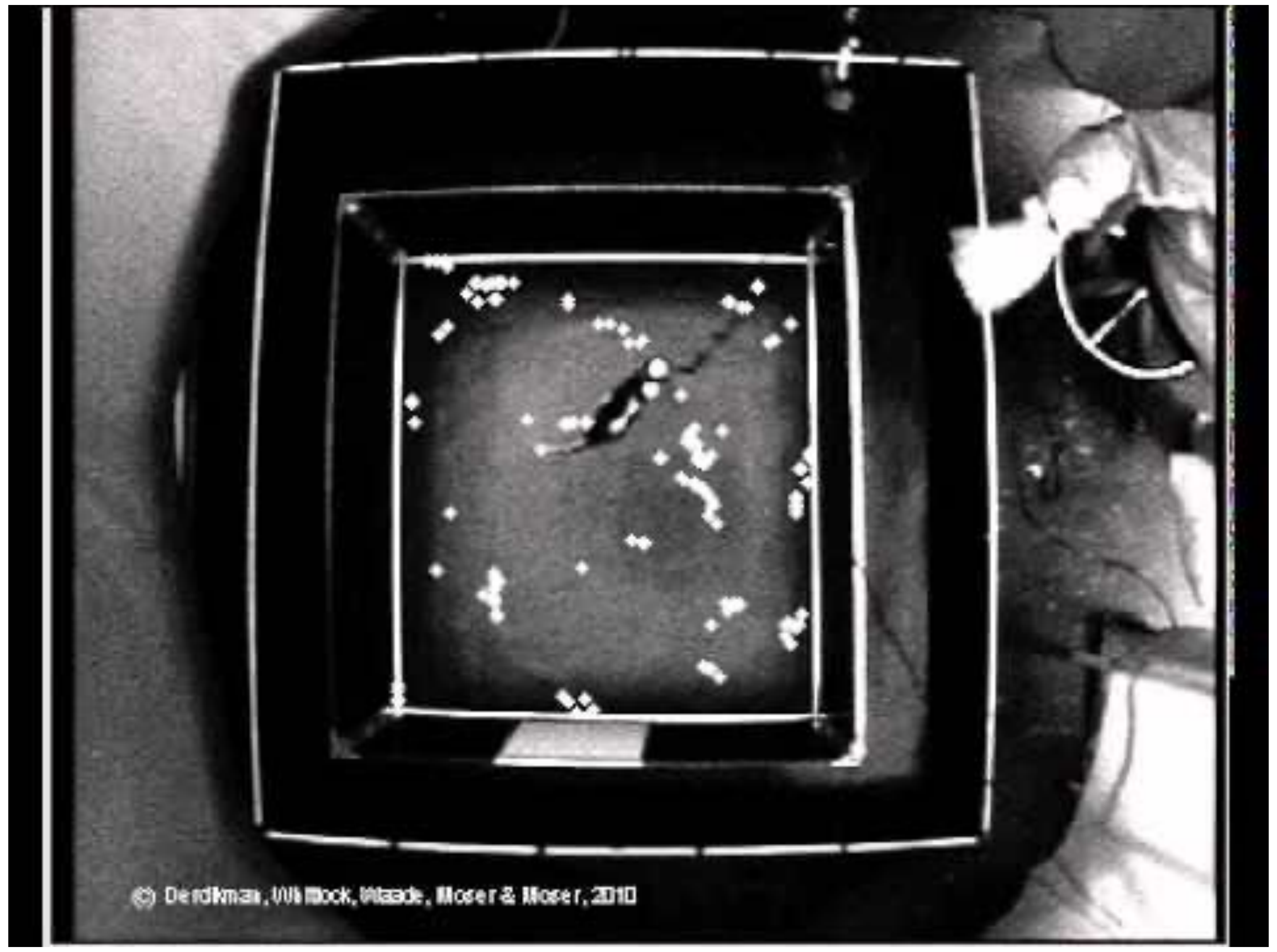
Edvard and Maj-Britt Moser
Nobel Prize in Physiology or Medicine 2014

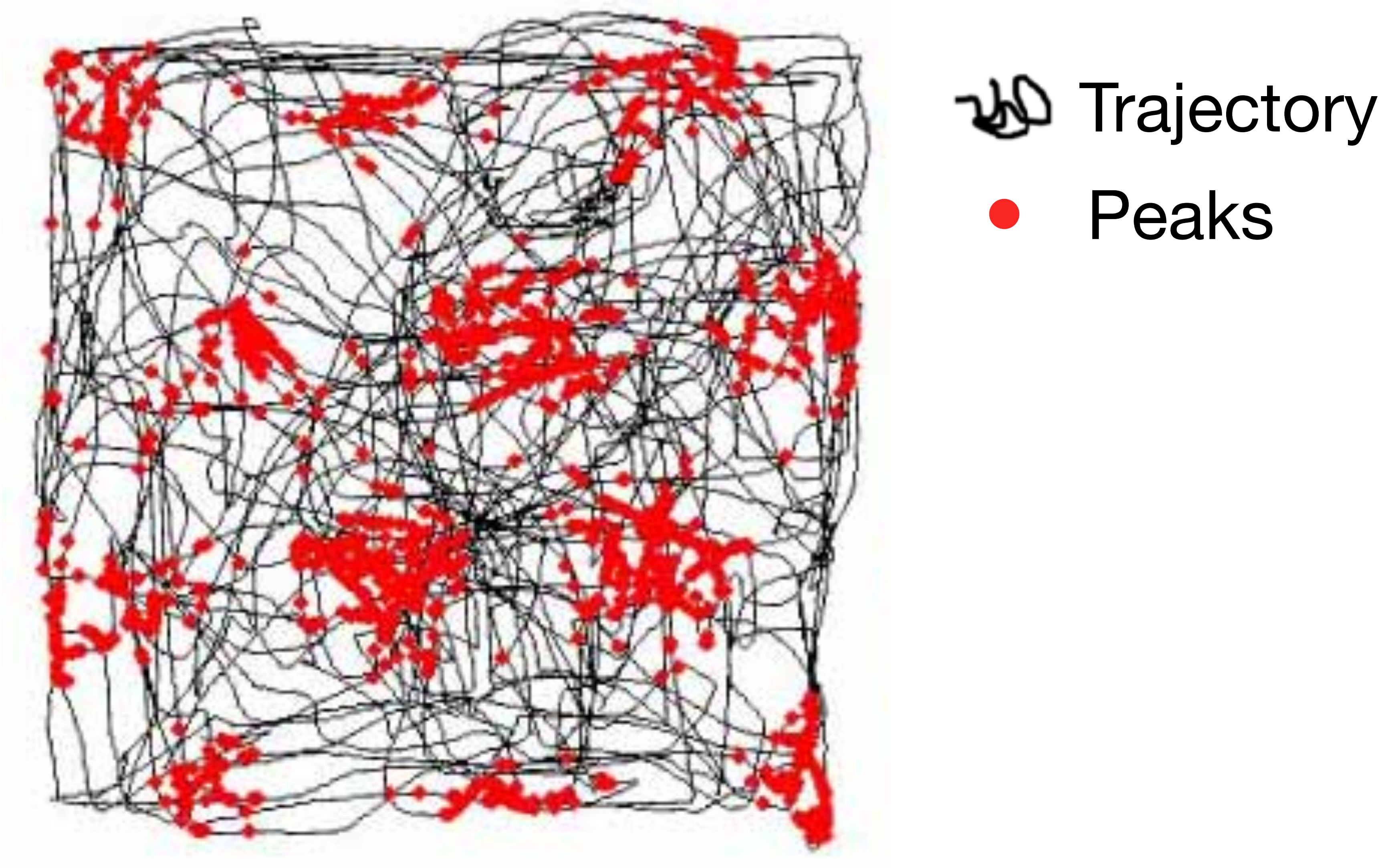


+ Peak

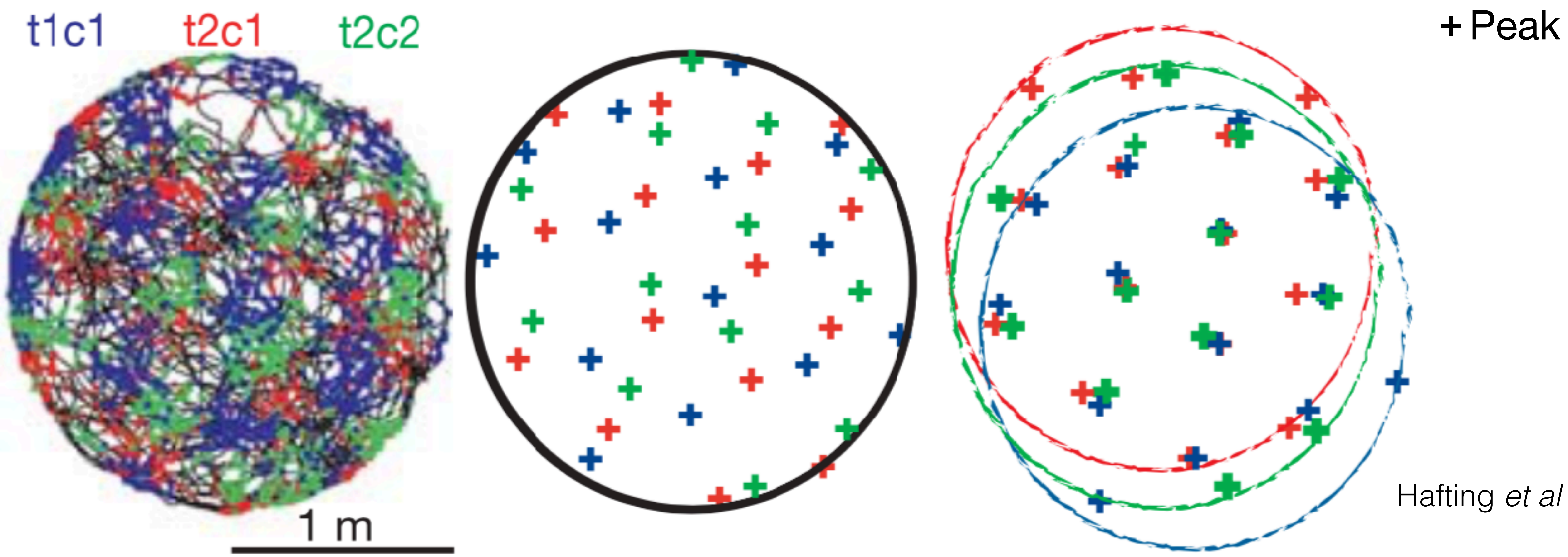
Hafting *et al* (Nature, 2005)

Grid cells in the Entorhinal Cortex provide a coordinate system



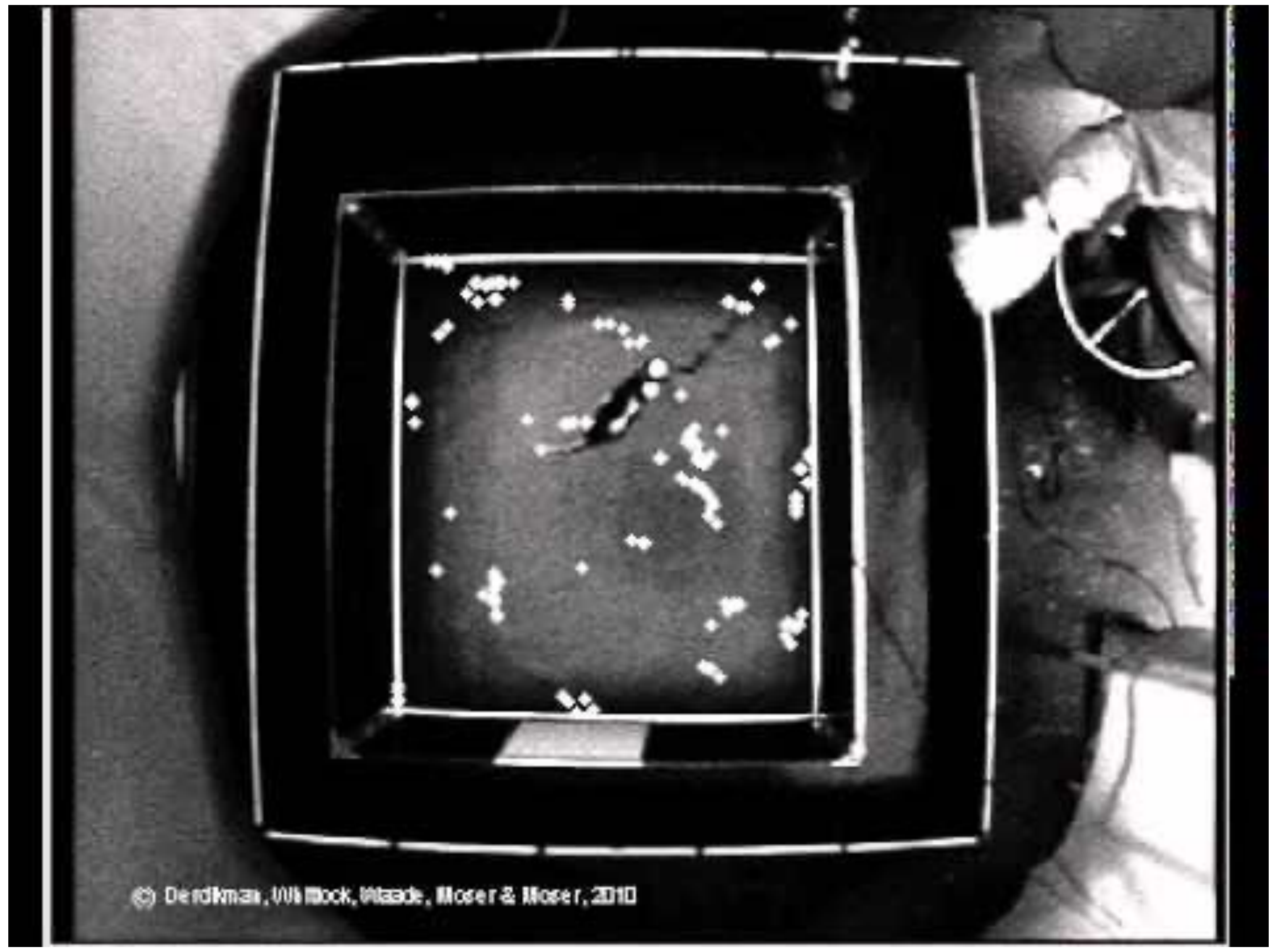


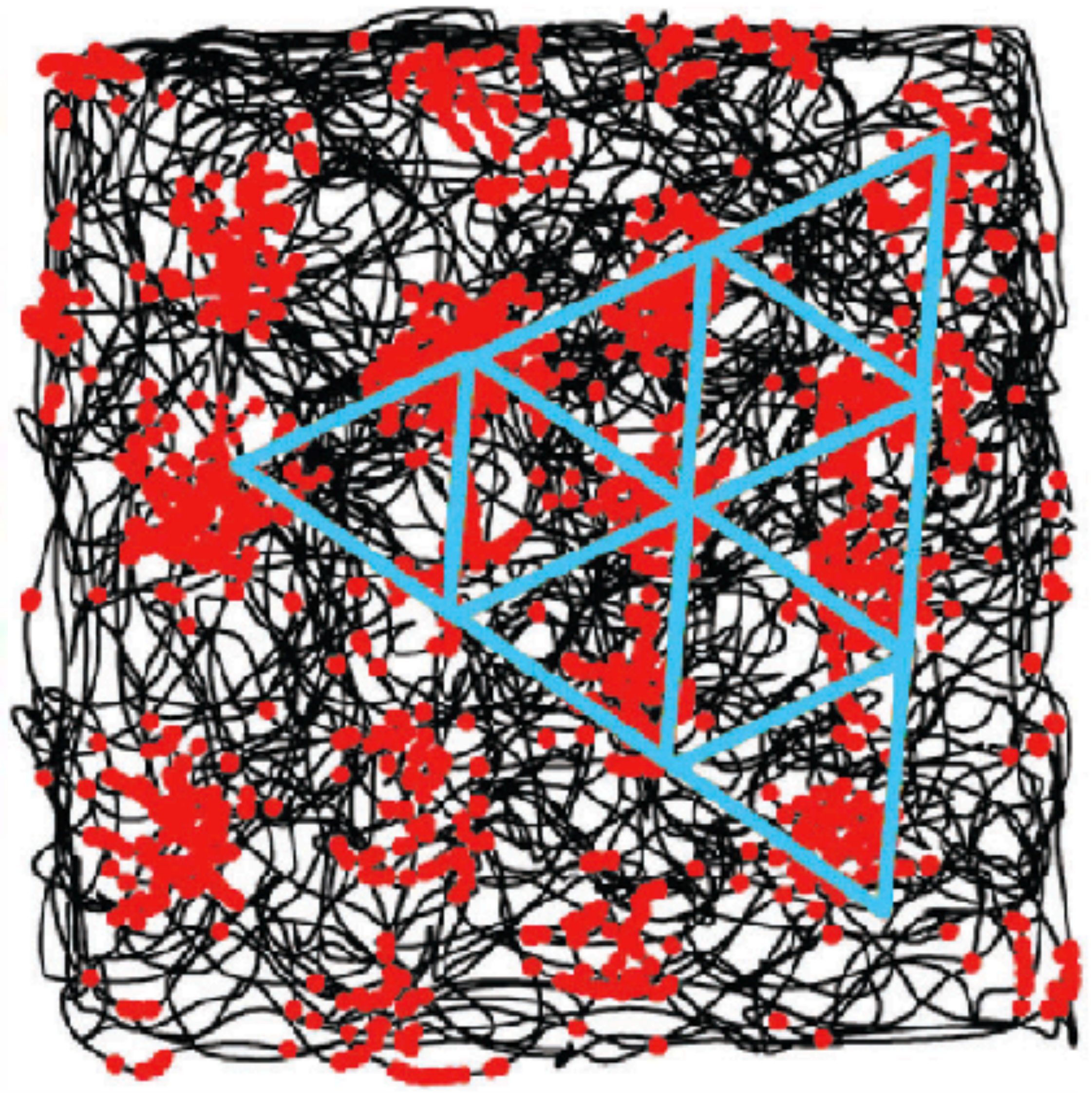
Edvard and Maj-Britt Moser
Nobel Prize in Physiology or Medicine 2014



Hafting *et al* (Nature, 2005)

Grid cells in the Entorhinal Cortex provide a coordinate system

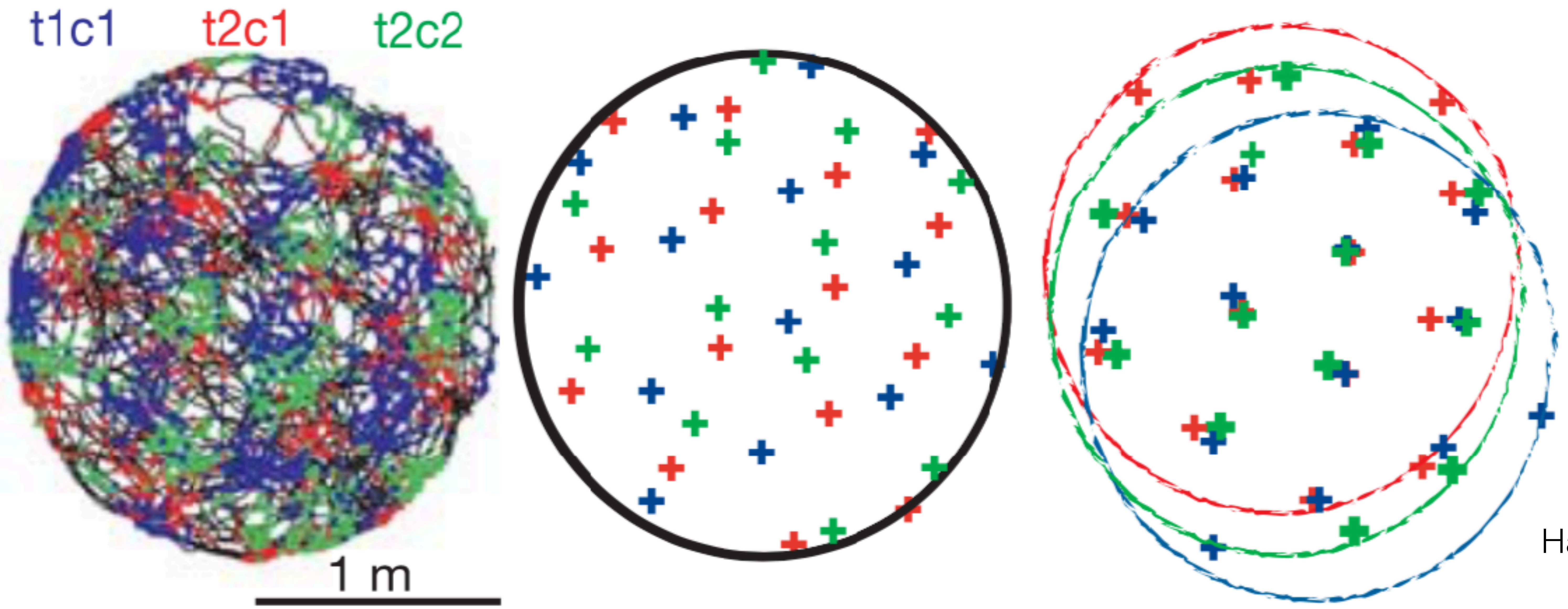




Trajectory

Peaks

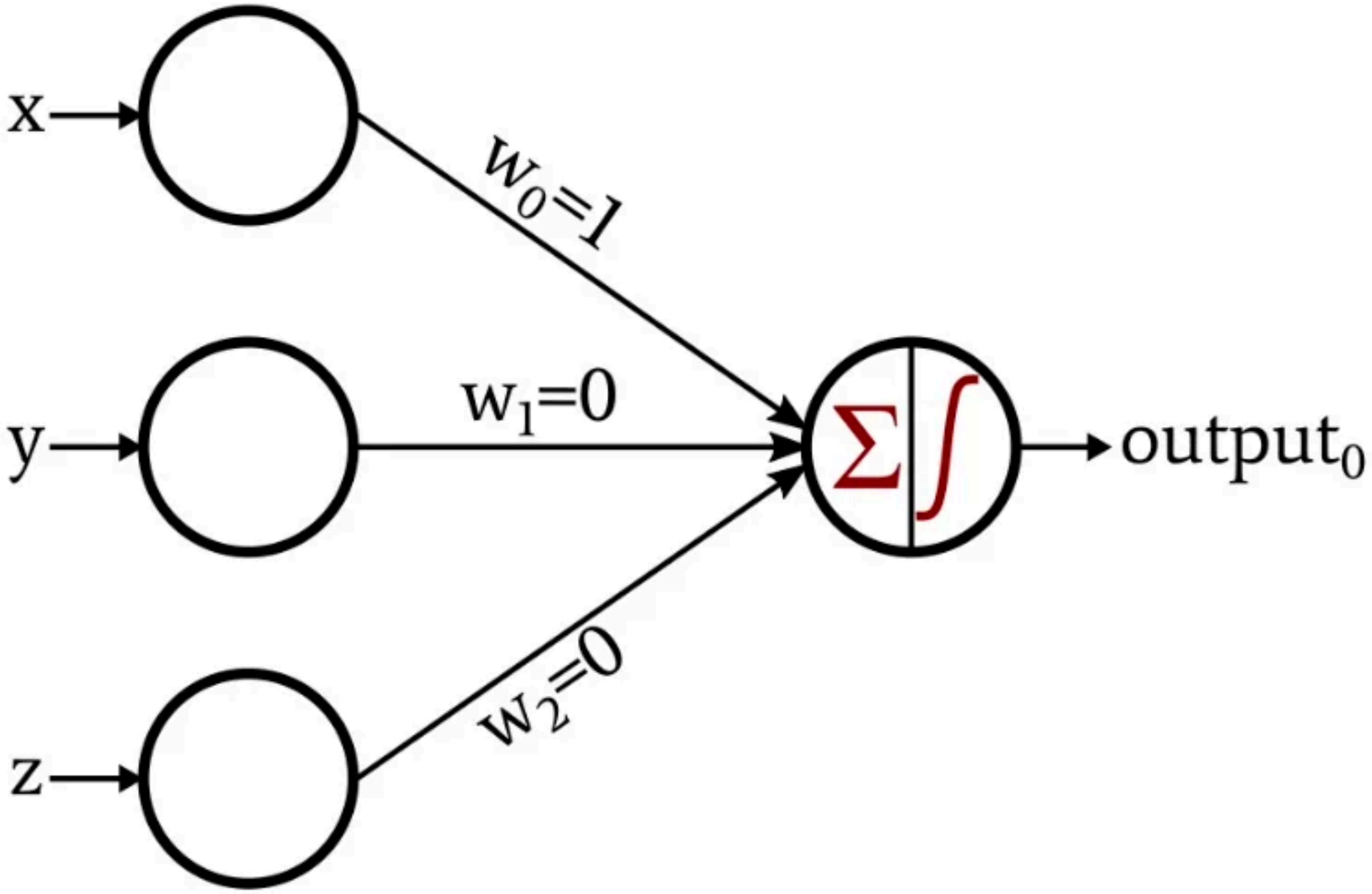
Edvard and Maj-Britt Moser
Nobel Prize in Physiology or Medicine 2014



+ Peak

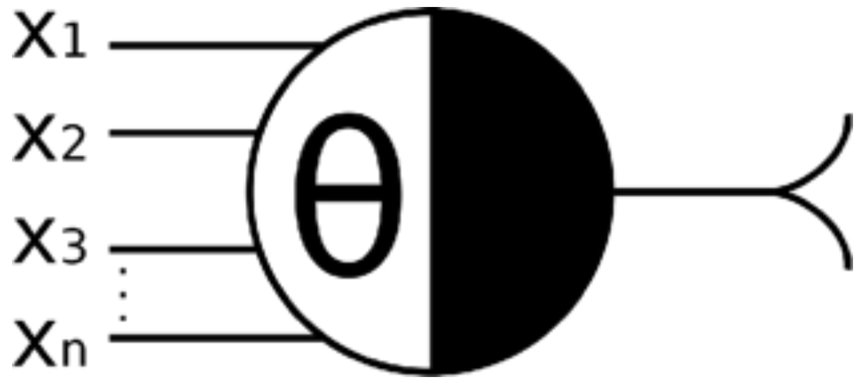
Hafting *et al* (Nature, 2005)

Origins of Artificial Learning



Timeline of Artificial Neural Networks

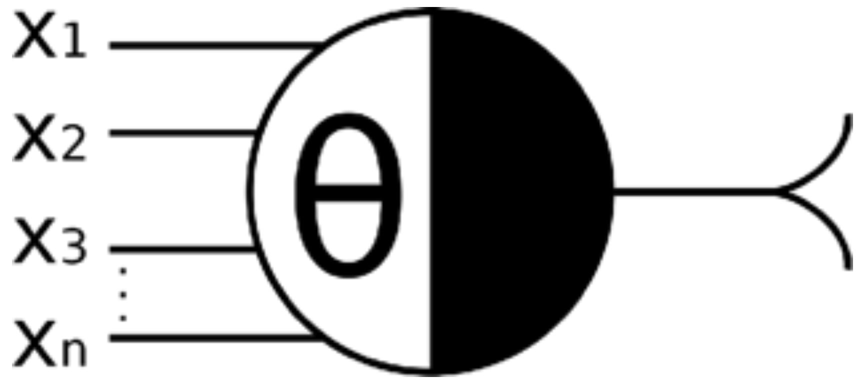
Timeline of Artificial Neural Networks



McCulloch & Pitts
(1943) Perceptron

Timeline of Artificial Neural Networks

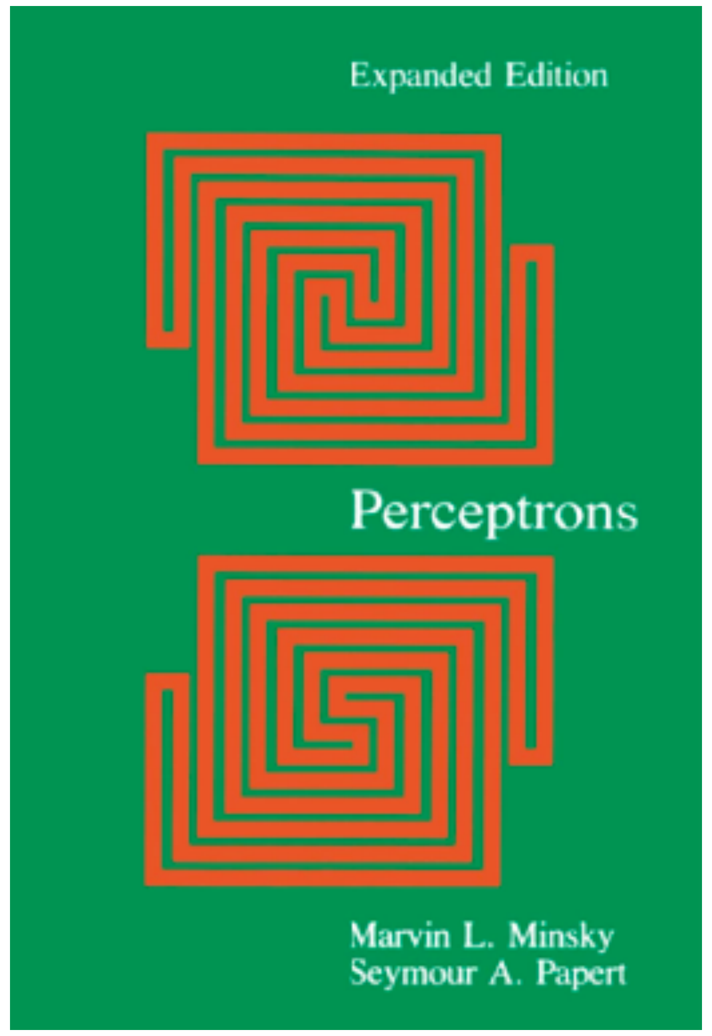
Rosenblatt (1958) Perceptron



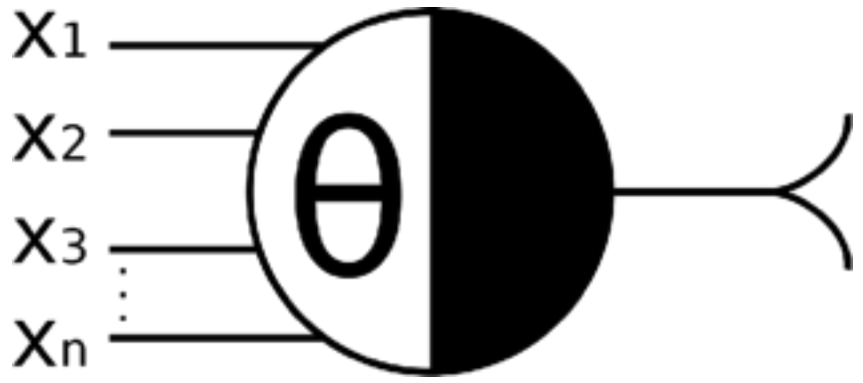
McCulloch & Pitts
(1943) Perceptron

Timeline of Artificial Neural Networks

Minsky & Parpert (1969)



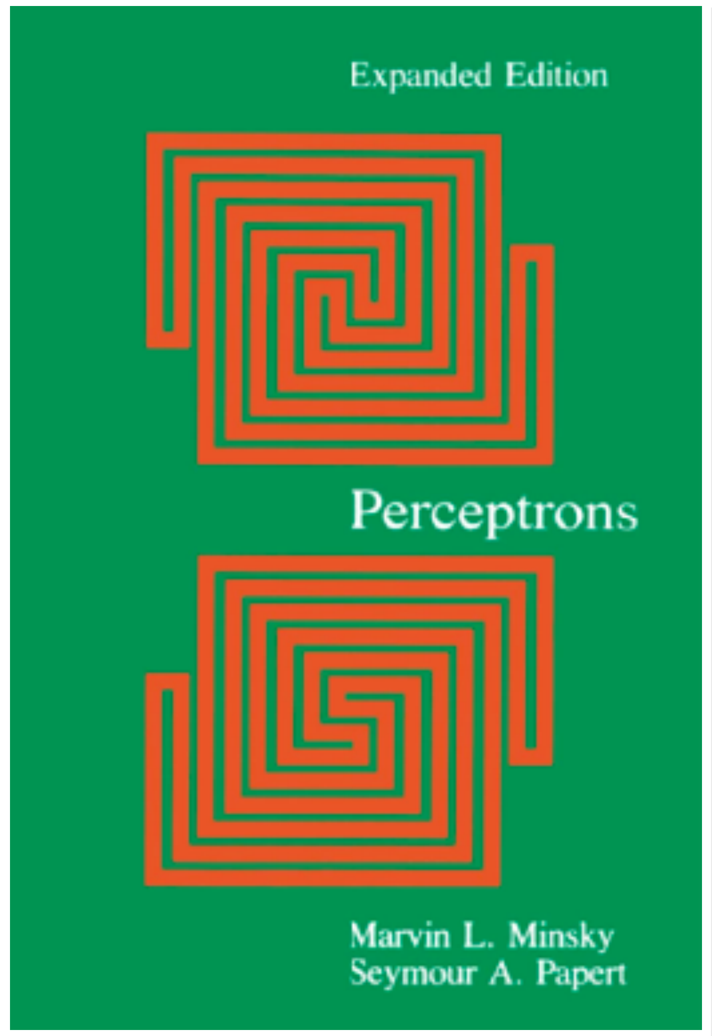
Rosenblatt (1958) Perceptron



McCulloch & Pitts
(1943) Perceptron

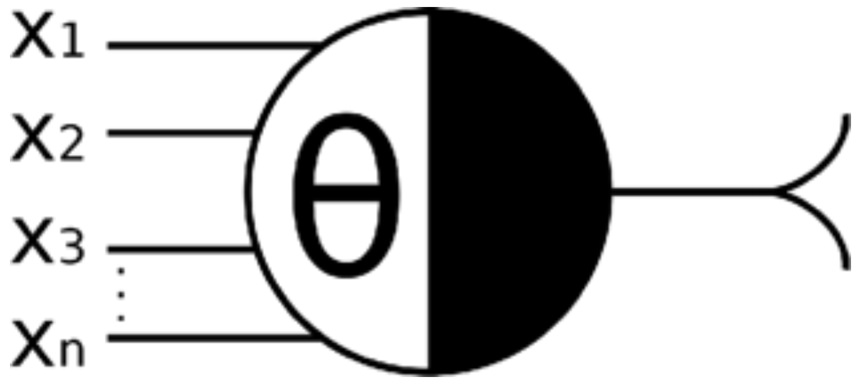
Timeline of Artificial Neural Networks

Minsky & Parpert (1969)



Rosenblatt (1958) Perceptron

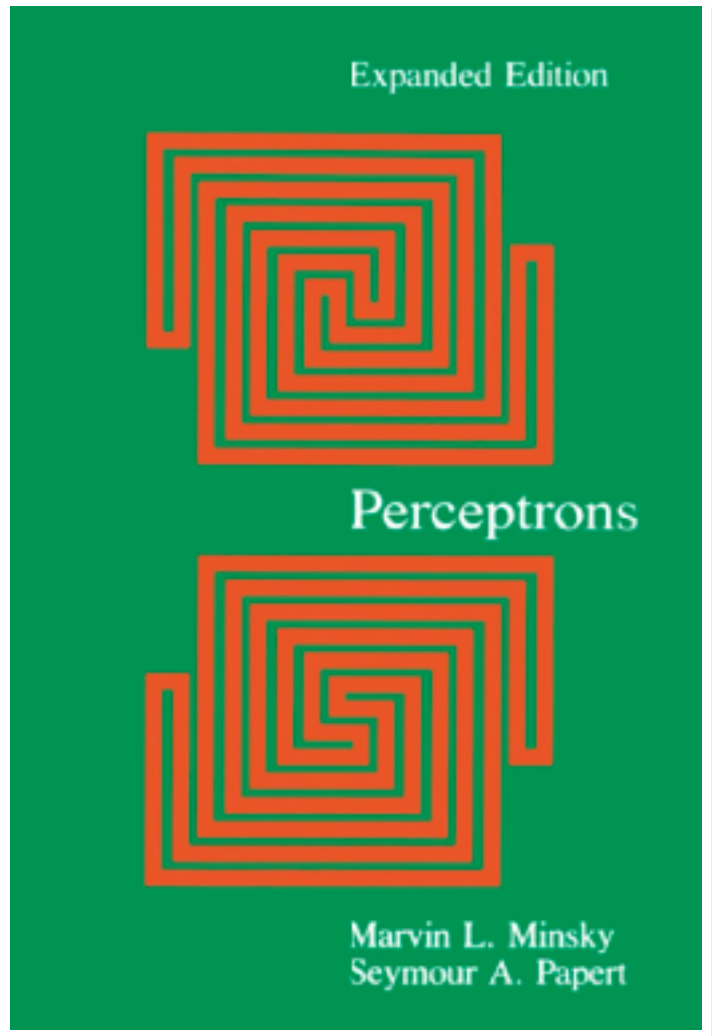
AI Winter



McCulloch & Pitts
(1943) Perceptron

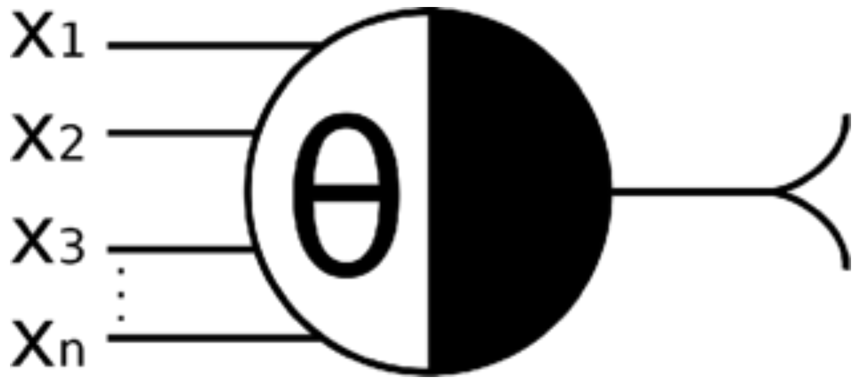
Timeline of Artificial Neural Networks

Minsky & Parpert (1969)

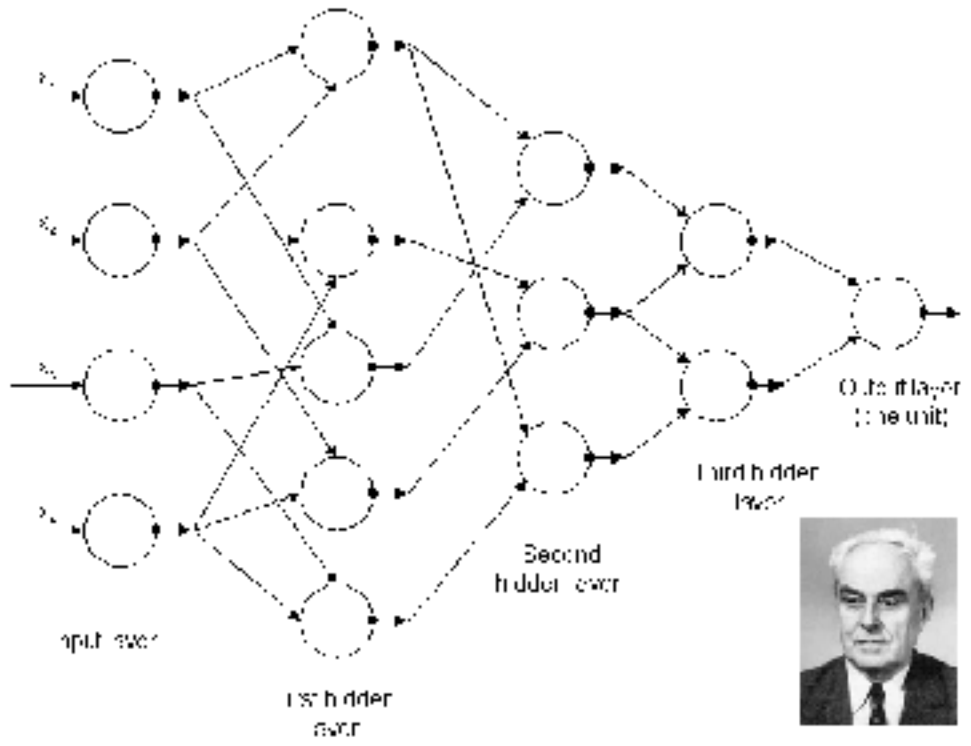


Rosenblatt (1958) Perceptron

AI Winter



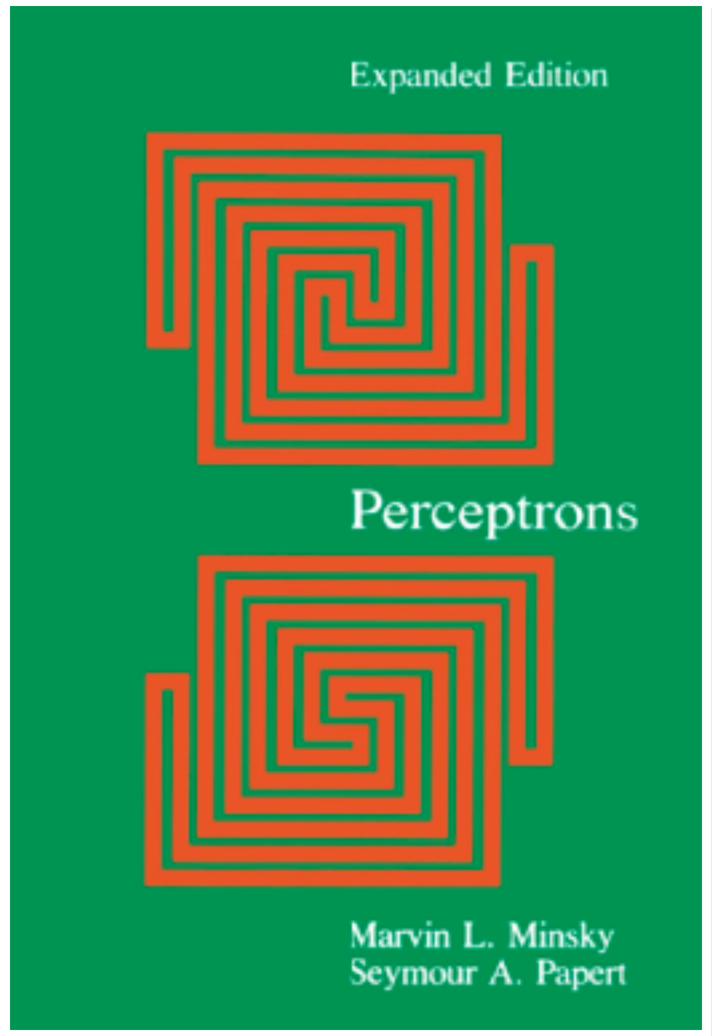
McCulloch & Pitts
(1943) Perceptron



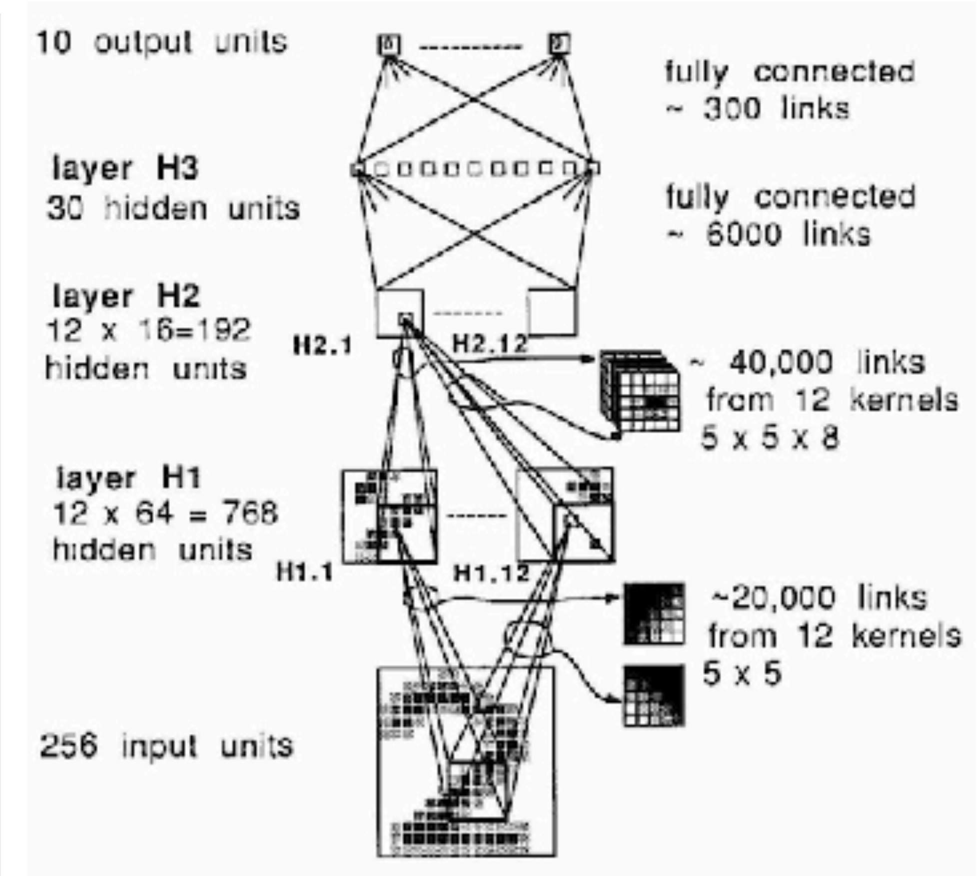
First deep network (Ivakhnenko & Lapa 1965)

Timeline of Artificial Neural Networks

Minsky & Parpert (1969)

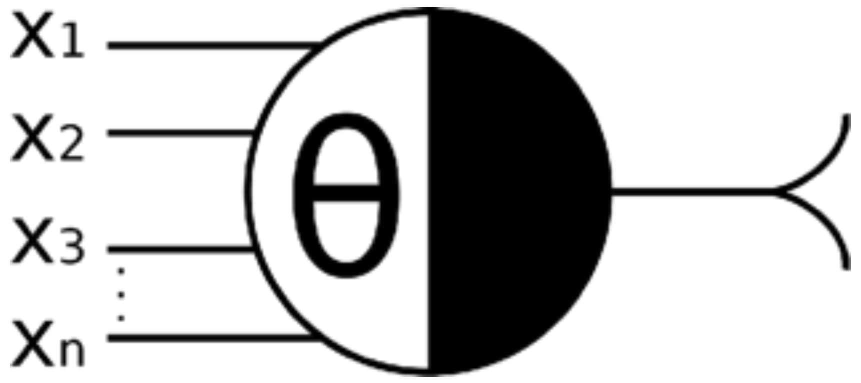


Convnets for MNIST (LeCun et al., 1989)

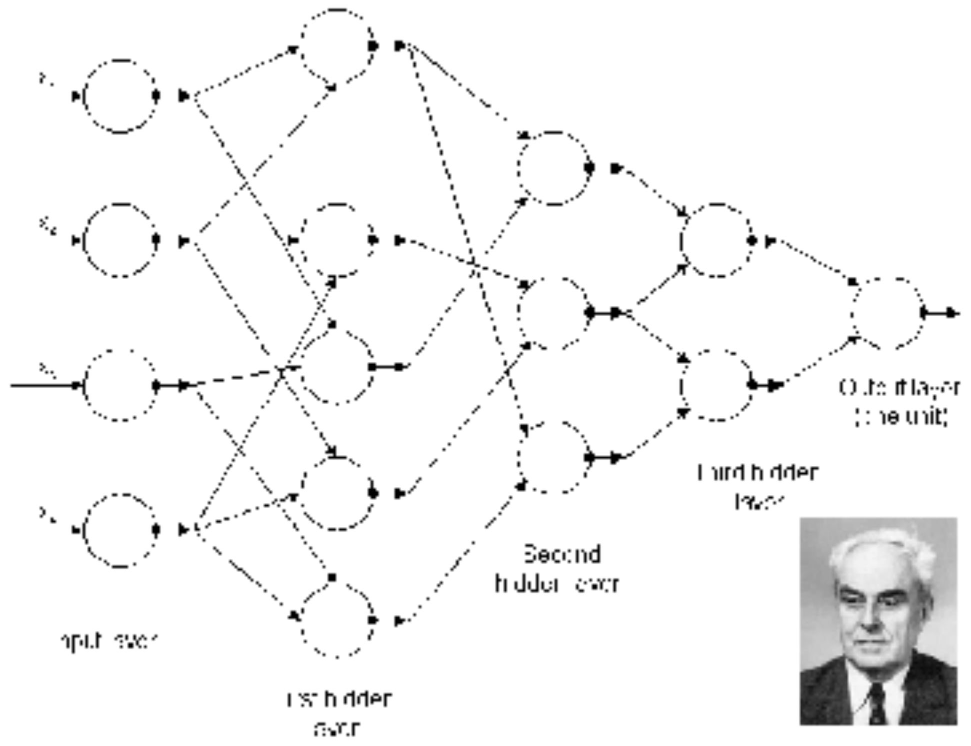


Rosenblatt (1958) Perceptron

AI Winter



McCulloch & Pitts
(1943) Perceptron



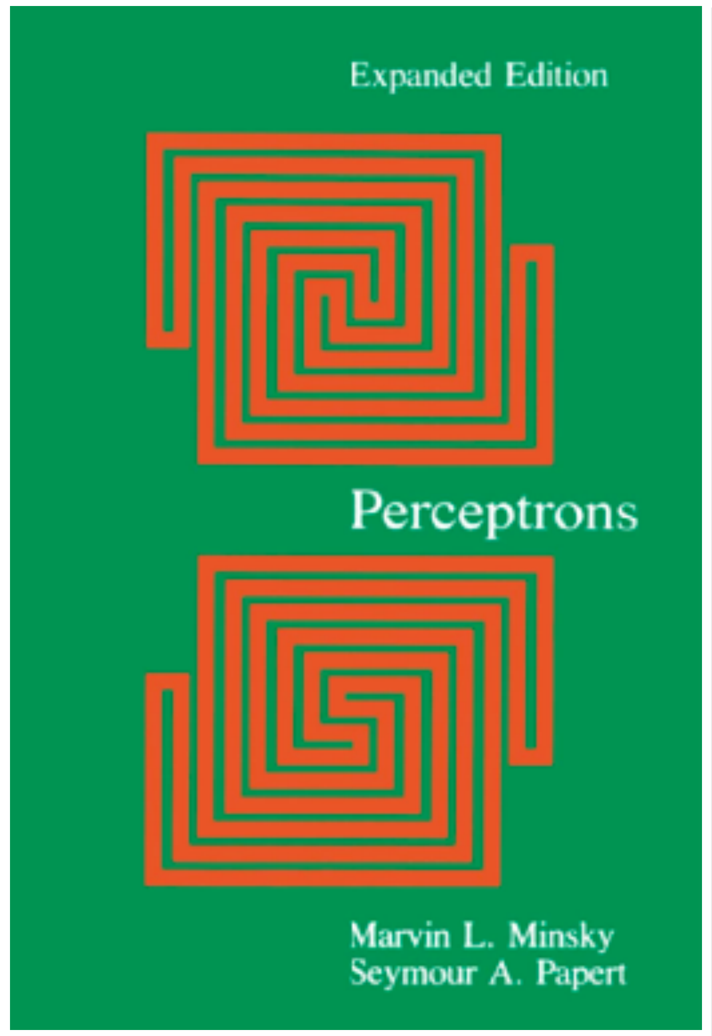
First deep network (Ivakhnenko & Lapa 1965)

Timeline of Artificial Neural Networks

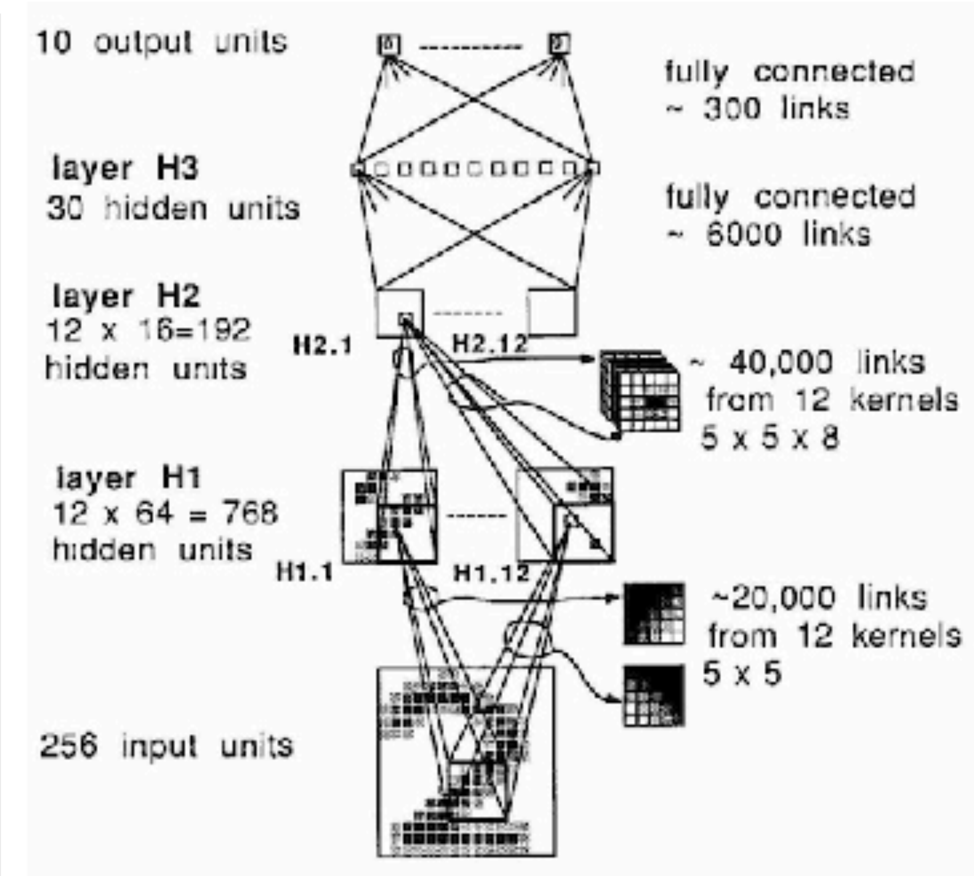
Deep Learning
revolution

Rosenblatt (1958) Perceptron

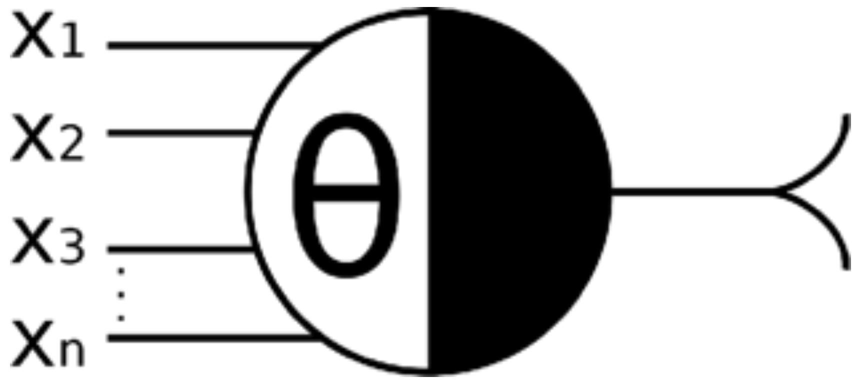
Minsky & Parpert (1969)



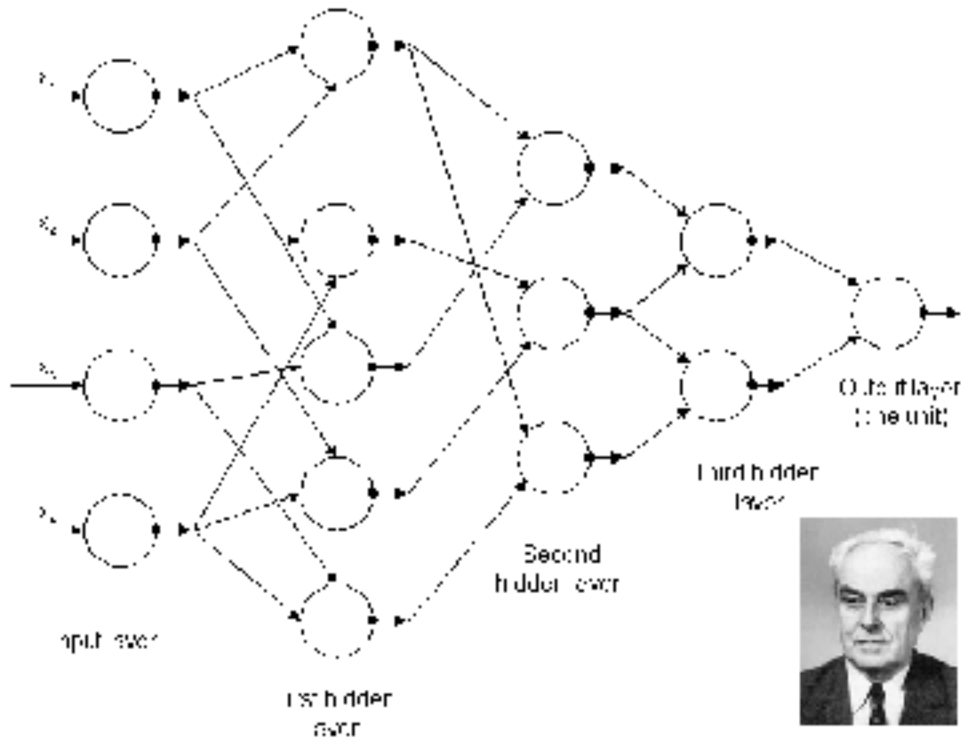
Convnets for MNIST (LeCun et al., 1989)



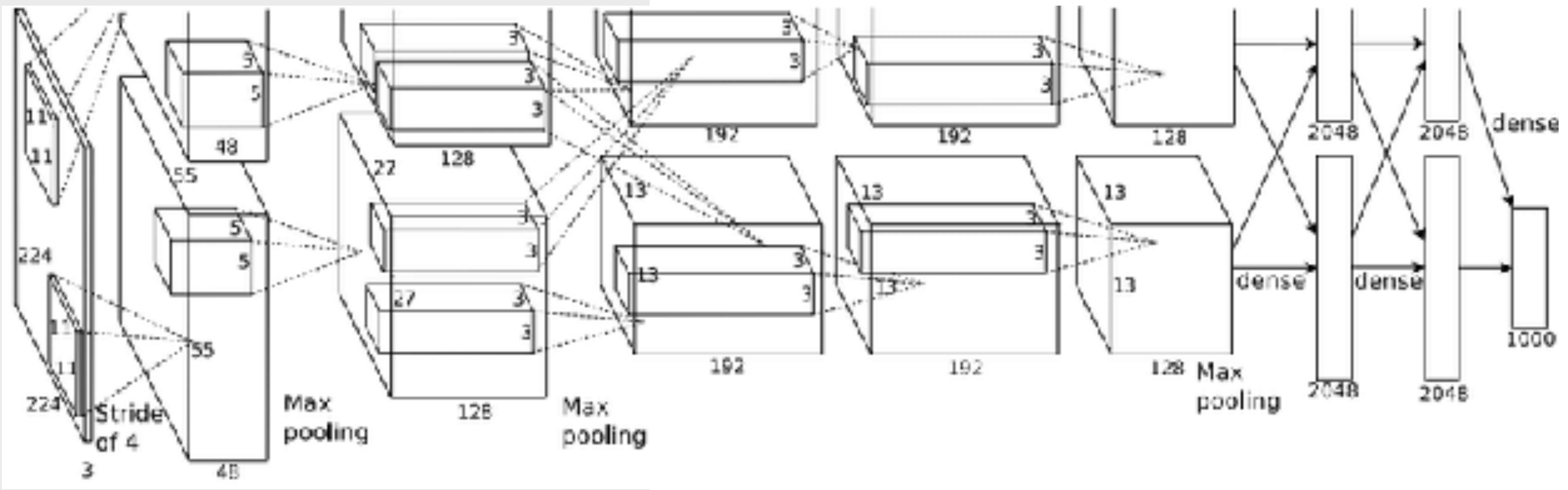
AI Winter



McCulloch & Pitts
(1943) Perceptron

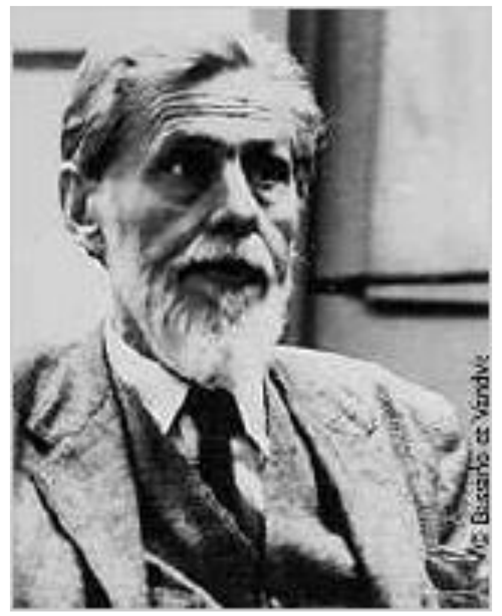


First deep network (Ivakhnenko & Lapa 1965)



ReLU & Dropout (Krizhevsky,
Sutskever, & Hinton, 2012)

McCulloch & Pitts (1943)

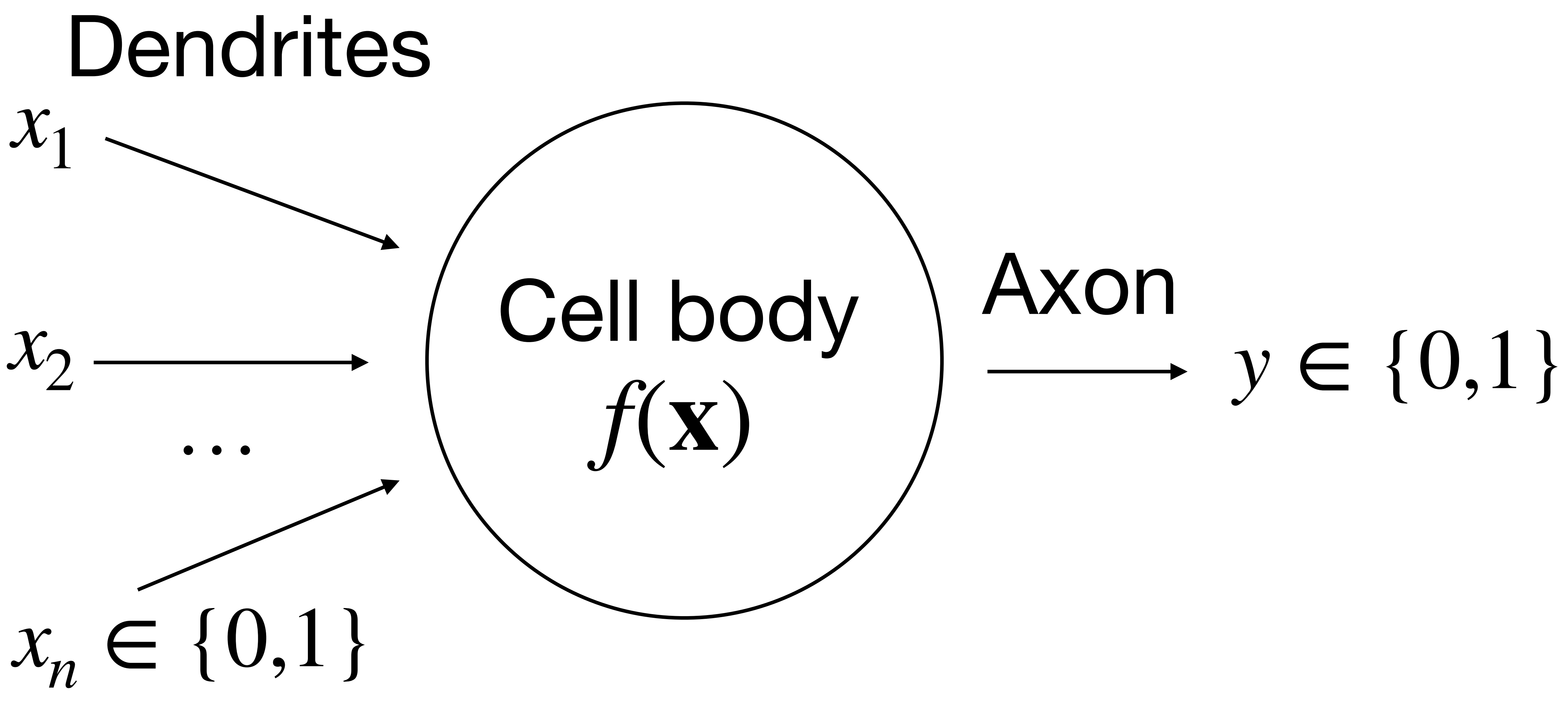

Warren McCulloch


Walter Pitts

- First computational model of a neuron
- The dendritic inputs $\{x_1, \ldots, x_n\}$ provide the input signal
- The cell body processes the signal

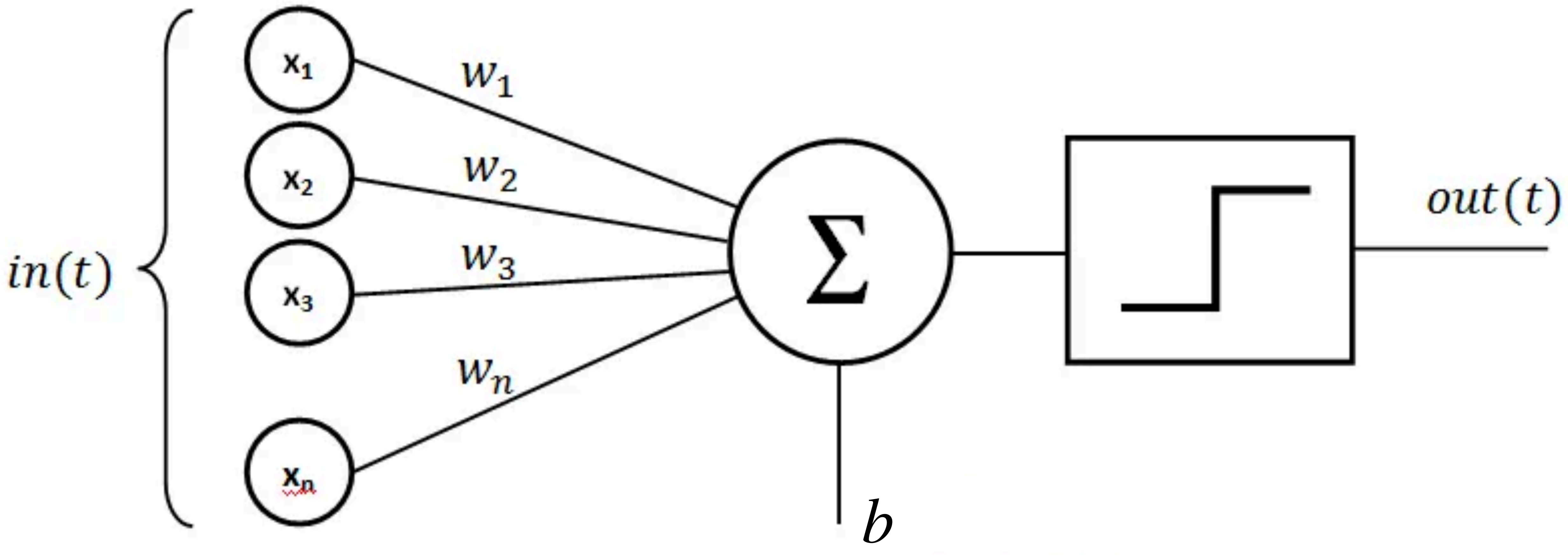
$$f(\mathbf{x}) = \begin{cases} 1 & \text{if } \sum x_i \geq \theta \\ 0 & \text{else} \end{cases}$$

- If the sum of the inputs is greater or equal to some *threshold* θ, then the axon produces the output



Rosenblatt's Perceptron

- **Added a learning rule**, allowing it to learn any binary classification problem *with linear seperability*
- Very similar to McCulloch & Pitts', but with some key differences:
  - A bias term is added b
  - Weights w_i aren't only $\in \{-1,1\}$ but can be any real number
- Weights (and bias) are updated based on error



Algorithm 1: Perceptron Learning Algorithm

Input: Training examples $\{\mathbf{x}_i, y_i\}_{i=1}^{m}$.

Initialize $\mathbf{w}$ and b randomly.

while *not converged* **do**

  ### Loop through the examples.

  for $j = 1, m$ **do**

    ### Compare the true label and the prediction.

    $error = y_j - \sigma(\mathbf{w}^T \mathbf{x}_j + b)$

    ### If the model wrongly predicts the class, we update the weights and bias.

    if *error != 0* **then**

      ### Update the weights.

      $\mathbf{w} = \mathbf{w} + error \times x_j$

      ### Update the bias.

      $b = b + error$

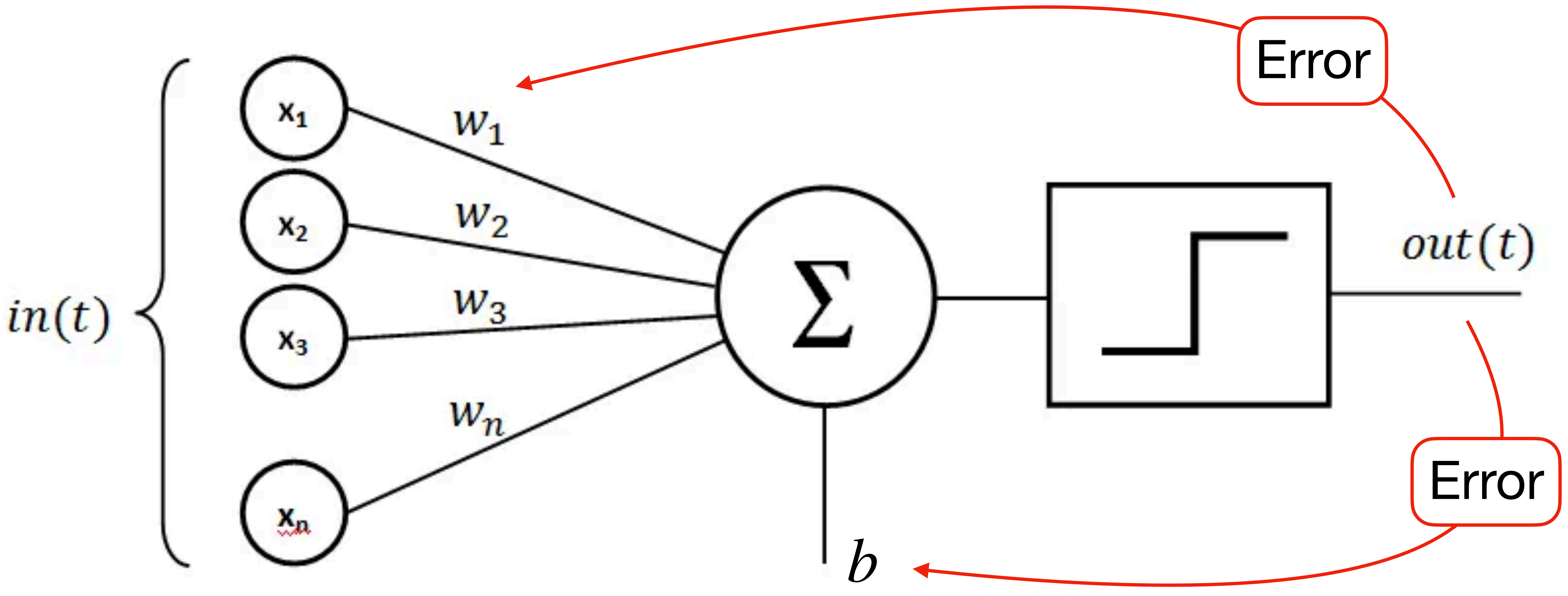
  Test for convergence

Output: Set of weights $\mathbf{w}$ and bias b for the perceptron.

not on the exam

Rosenblatt's Perceptron

- **Added a learning rule**, allowing it to learn any binary classification problem *with linear seperability*
- Very similar to McCulloch & Pitts', but with some key differences:
  - A bias term is added b
  - Weights w_i aren't only $\in \{-1,1\}$ but can be any real number
- Weights (and bias) are updated based on error



Algorithm 1: Perceptron Learning Algorithm

Input: Training examples $\{\mathbf{x}_i, y_i\}_{i=1}^m$.

Initialize $\mathbf{w}$ and b randomly.

while *not converged* **do**

  # # # Loop through the examples.

  for $j = 1, m$ **do**

    # # # Compare the true label and the prediction.

    $error = y_j - \sigma(\mathbf{w}^T\mathbf{x}_j + b)$

    ### If the model wrongly predicts the class, we update the weights and bias.

    if *error != 0* **then**

      ### Update the weights.

      $\mathbf{w} = \mathbf{w} + error \times x_j$

      ### Update the bias.

      $b = b + error$

  Test for convergence

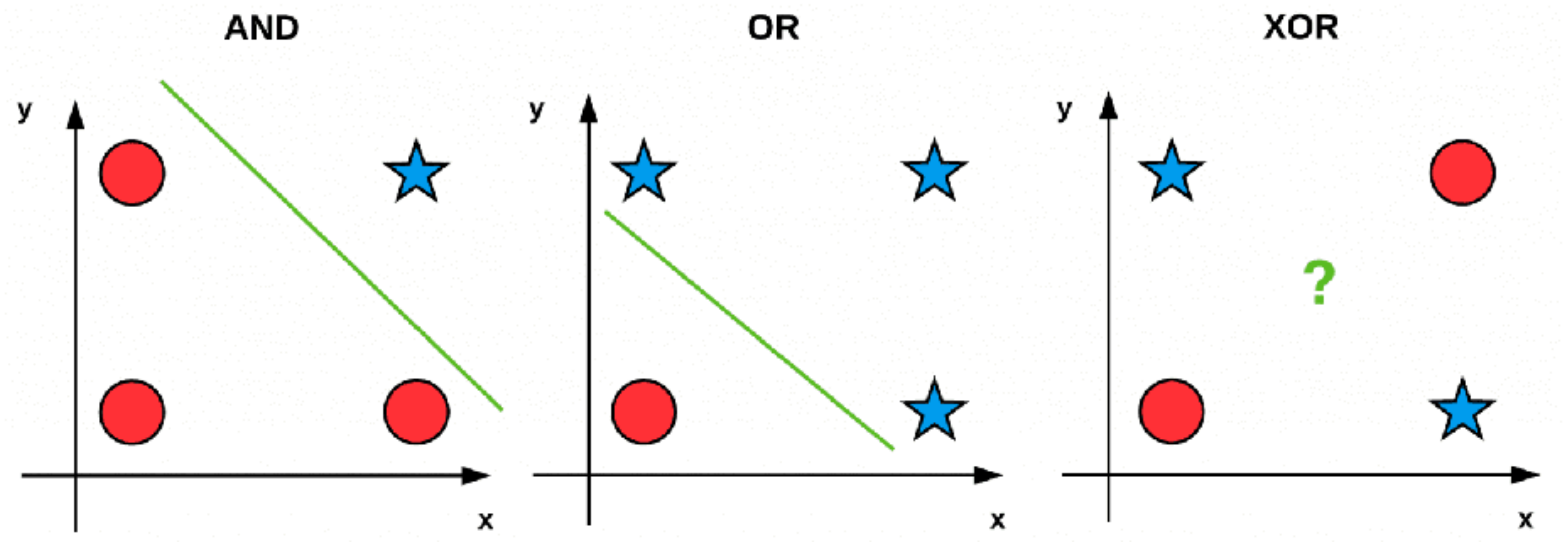
Output: Set of weights $\mathbf{w}$ and bias b for the perceptron.

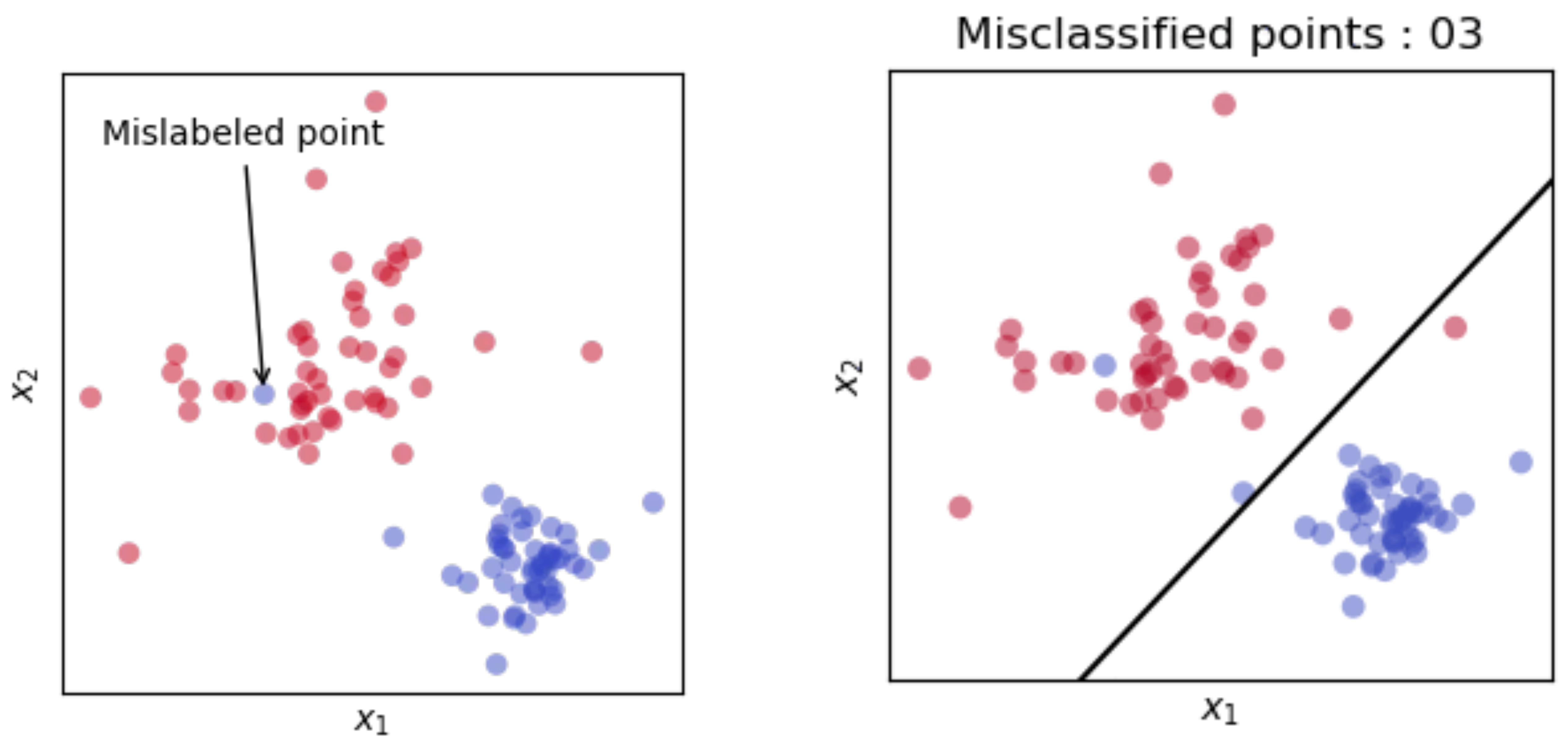
not on the exam

Limitations of linear separability

Adrian Rosebrock

- The perceptron can learn any linearly separable problem
  - But not all problems are lineary separable
- Even a single mislabeled data point in the data will throw the algorithm into chaos
- Enter the **XOR problem** and Minsky & Parpert (1969) critique
  - Argument: because a single neuron is unable to solve XOR, larger networks will also have similar problems
  - Therefore, the research program should be dropped

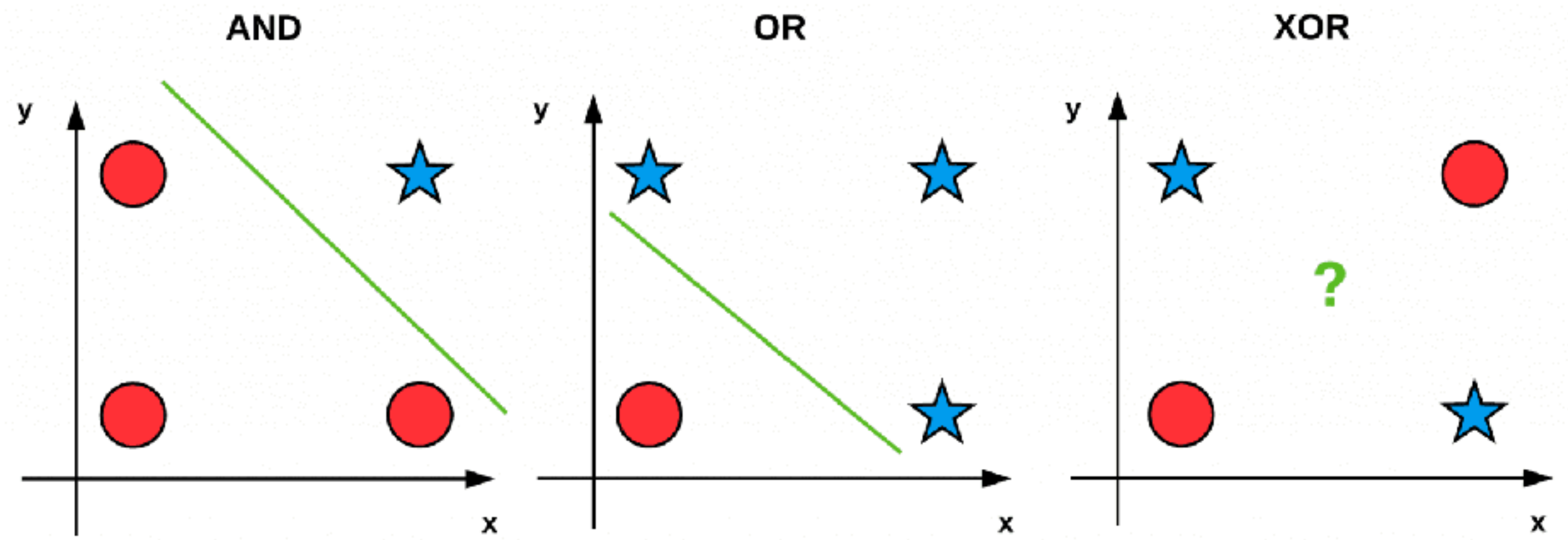


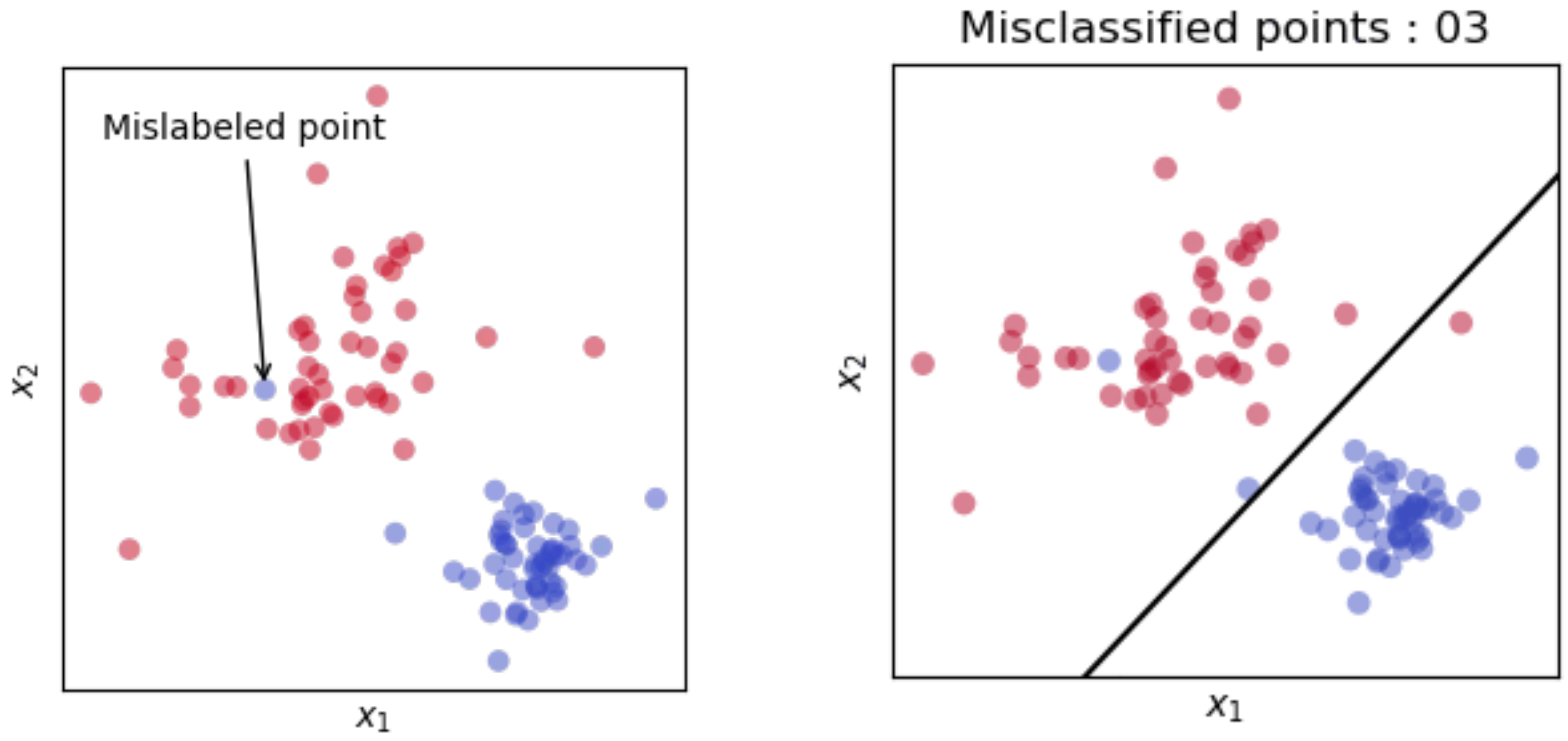


Limitations of linear separability

Adrian Rosebrock

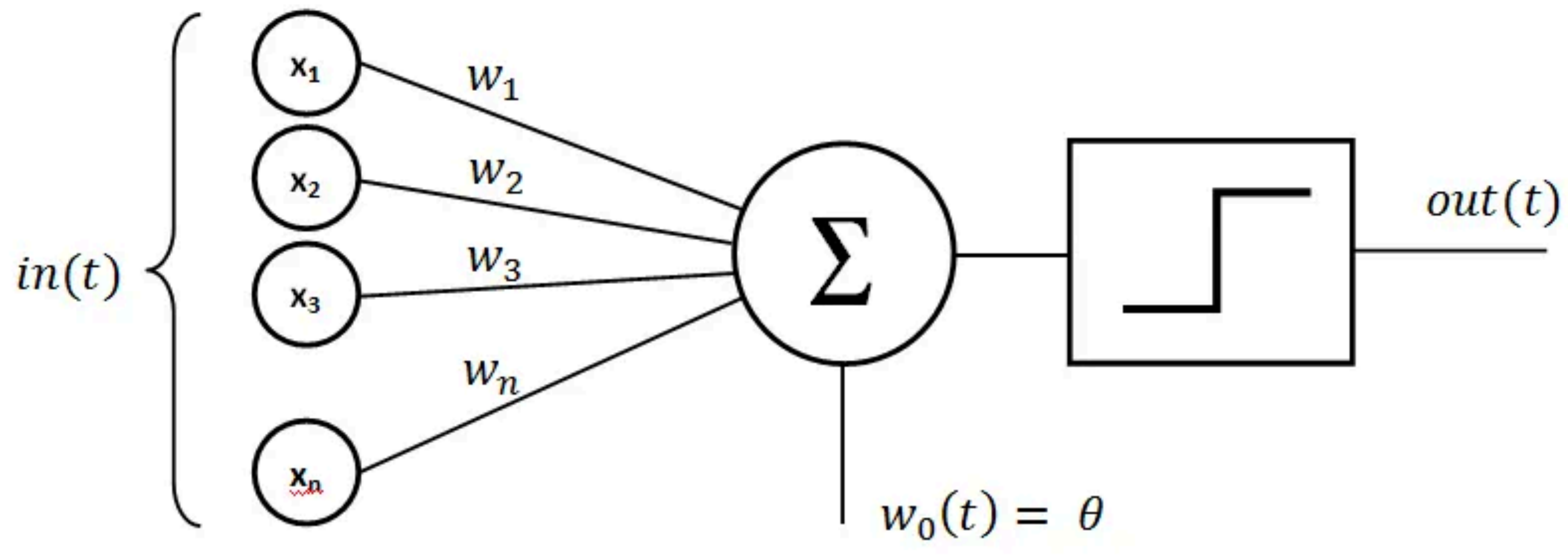
- The perceptron can learn any linearly separable problem
  - But not all problems are lineary separable
- Even a single mislabeled data point in the data will throw the algorithm into chaos
- Enter the **XOR problem** and Minsky & Parpert (1969) critique
  - Argument: because a single neuron is unable to solve XOR, larger networks will also have similar problems
  - Therefore, the research program should be dropped

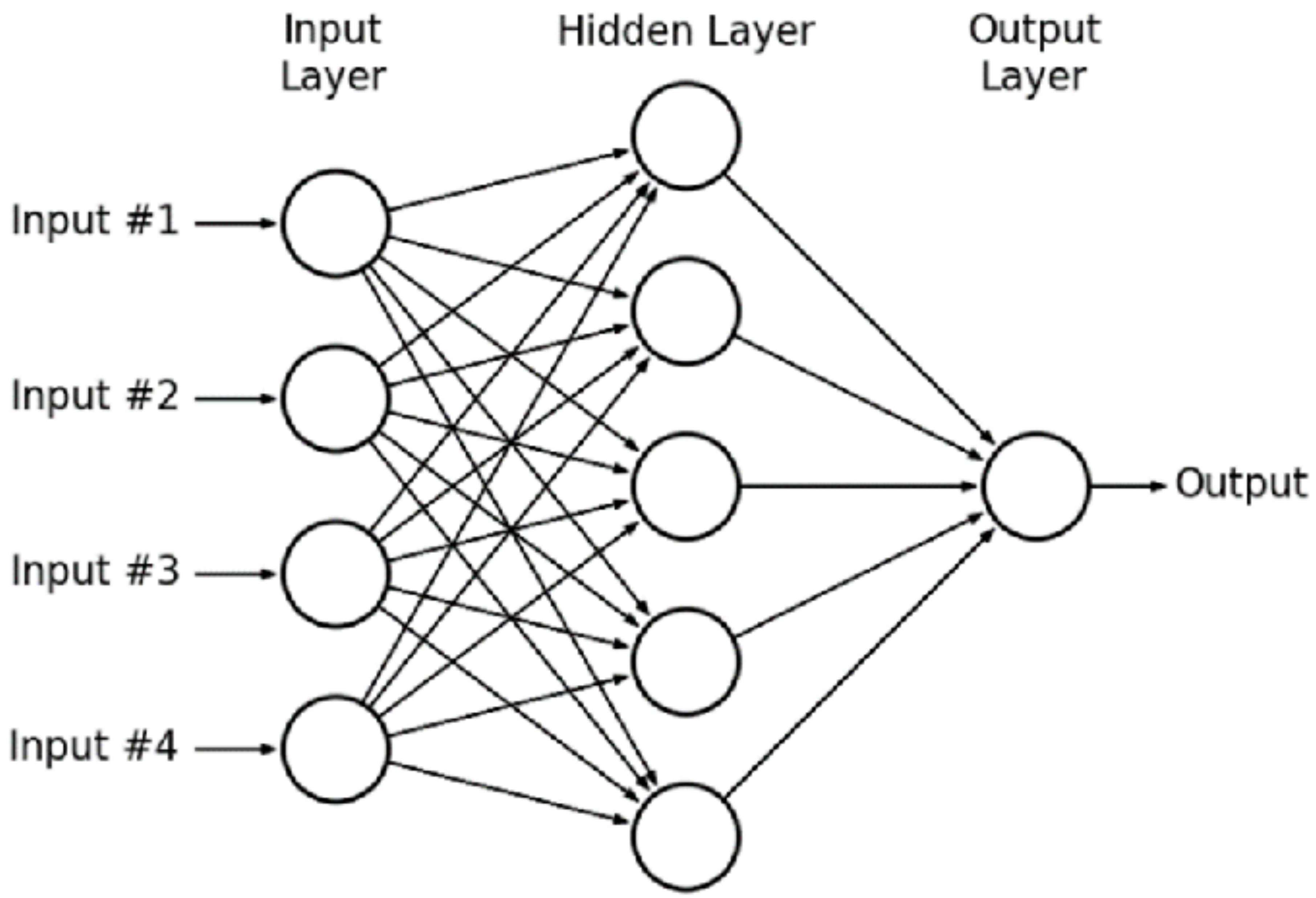




Multilayer Perceptrons

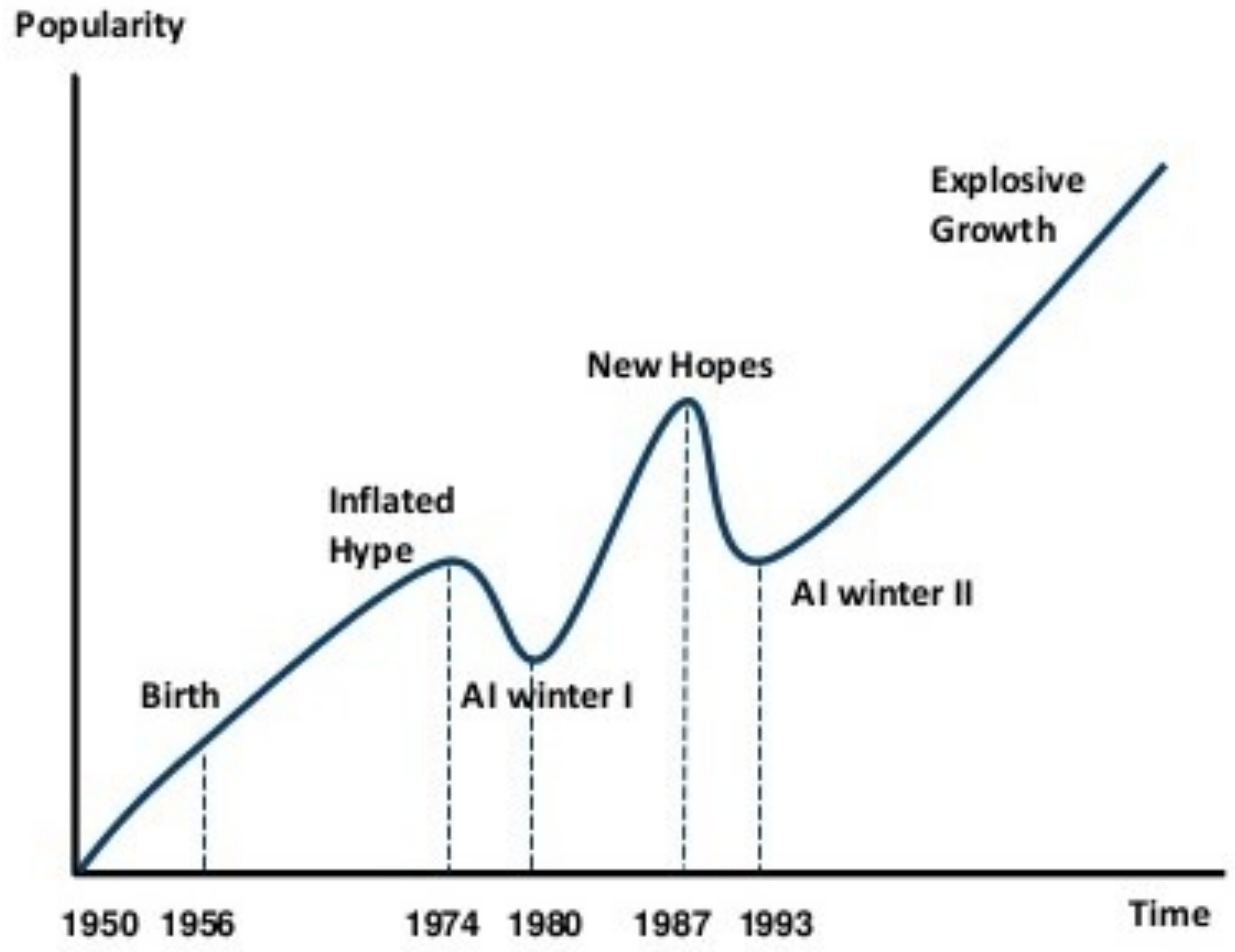
- MLPs are feedforward networks with (multiple) **hidden layers**, where we apply the same activation function at each layer
  - A single hidden layer allows us to solve XOR
- More generally, MLPs can learn any abitrary decision boundary by adding more hidden layers
- Training via gradient descent and backpropogation





The 1st AI winter and the rise of symbolic AI

- Skepticism about Perceptrons not being able to solve XOR problems led to **AI winter I**
- Afterwards, was a hopeful revival of interest based on “expert systems” using **symbolic AI**
- Limitations of expert systems caused **AI winter II**, which ended with modern advances in pattern recognition and deep neural networks (i.e., machine learning)



Symbolic AI

Herbert Simon
& Allen Newell

- **Physical Symbol System hypothesis**:
  "*A physical symbol system has the necessary and sufficient means for general intelligent action* - Allen Newell and Herbert Simon (1976)"

Symbolic AI

- **Physical Symbol System hypothesis**:
  "A physical symbol system has the necessary and sufficient means for general intelligent action - Allen Newell and Herbert Simon (1976)"

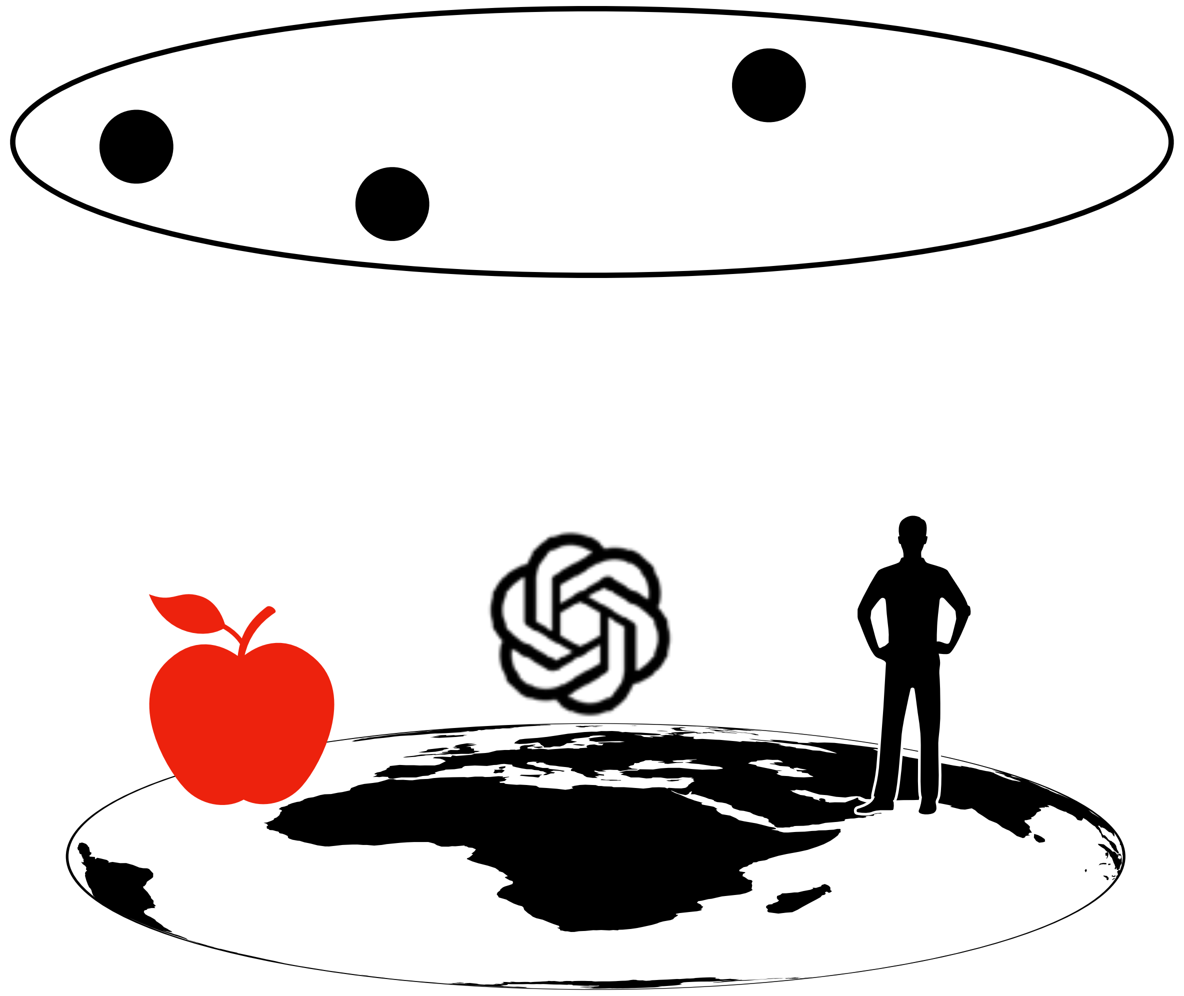
Herbert Simon
& Allen Newell

- **Symbols** can represent things in the world
  - e.g., (Apple), (ChatGPT), (Charley), etc…

Symbolic AI

Herbert Simon
& Allen Newell

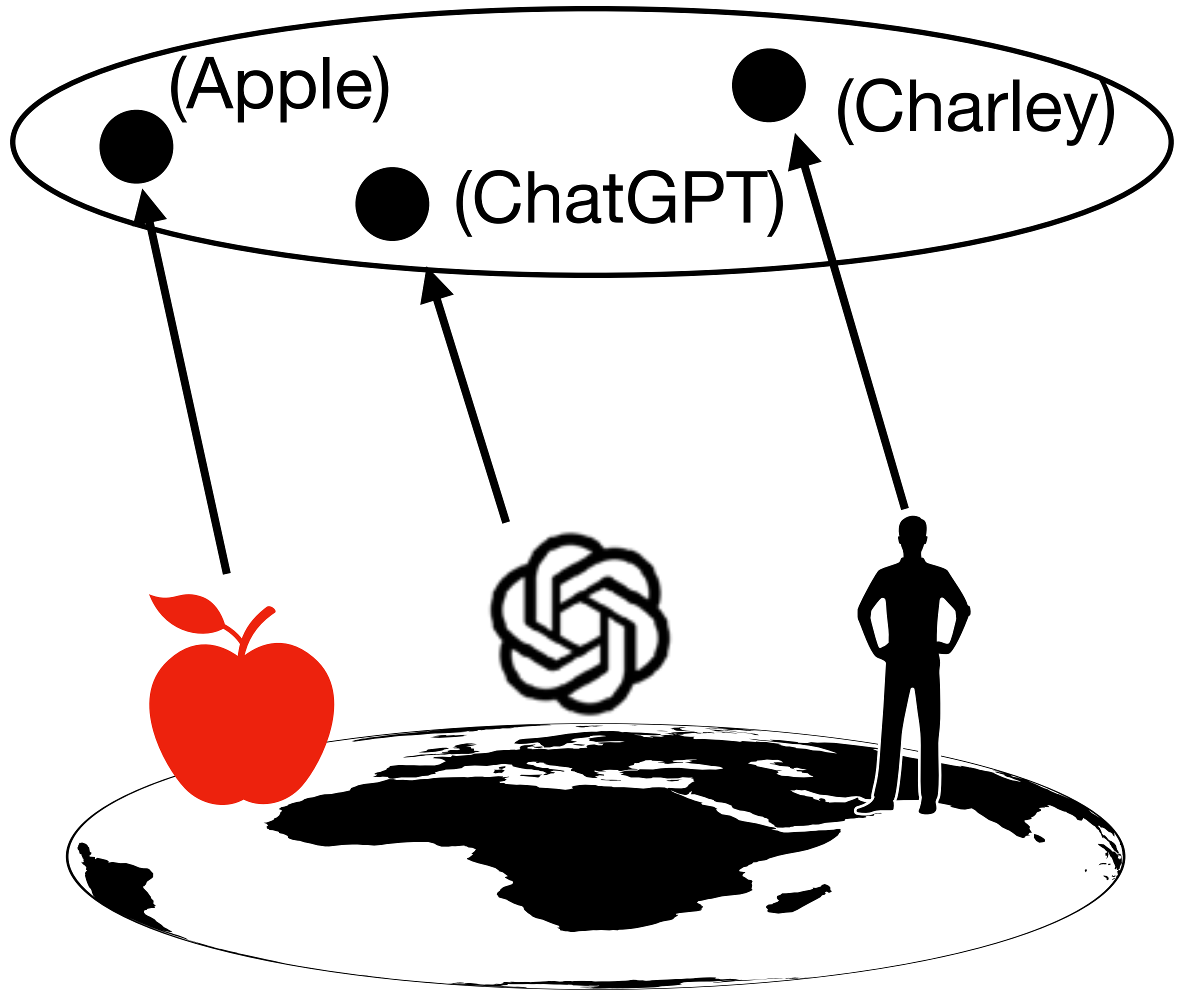
- **Physical Symbol System hypothesis**:
  "A physical symbol system has the necessary and sufficient means for general intelligent action - Allen Newell and Herbert Simon (1976)"
  - **Symbols** can represent things in the world
    - e.g., (Apple), (ChatGPT), (Charley), etc…



Symbolic AI

- **Physical Symbol System hypothesis**:
  "A physical symbol system has the necessary and sufficient means for general intelligent action - Allen Newell and Herbert Simon (1976)"
  - **Symbols** can represent things in the world
    - e.g., (Apple), (ChatGPT), (Charley), etc…

Herbert Simon
& Allen Newell

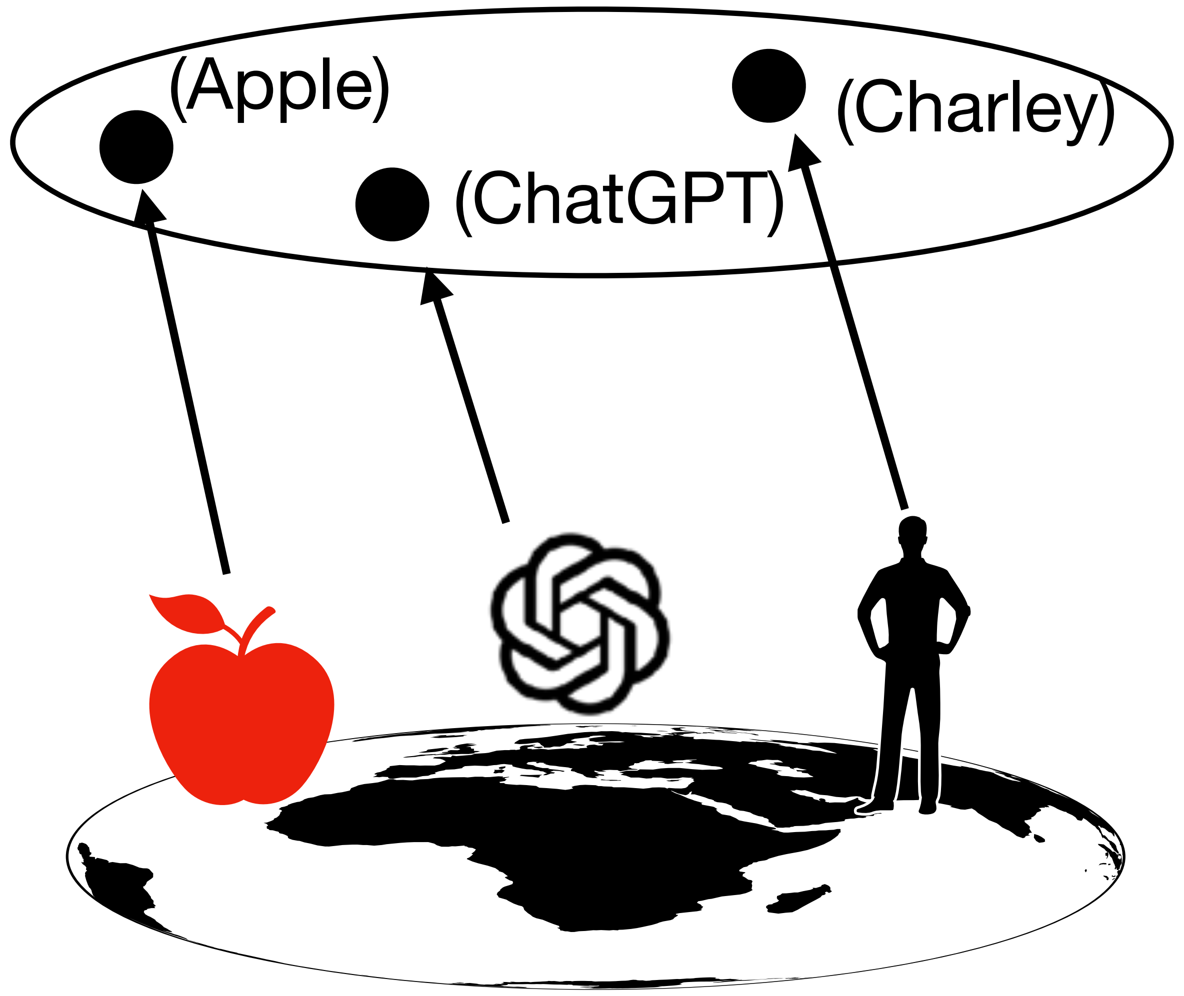


Symbolic AI

Herbert Simon
& Allen Newell

- **Physical Symbol System hypothesis**:
  "A physical symbol system has the necessary and sufficient means for general intelligent action - Allen Newell and Herbert Simon (1976)"

- **Symbols** can represent things in the world
  - e.g., (Apple), (ChatGPT), (Charley), etc…
- **Relations** can be i) predicates that describes a symbol or ii) verbs describing how symbols interact with other symbols
  - i) red(Apple), unreliable(ChatGPT), instructor(Charley)
  - ii) eat(Charley, Apple), generatePicture(ChatGPT, Apple)

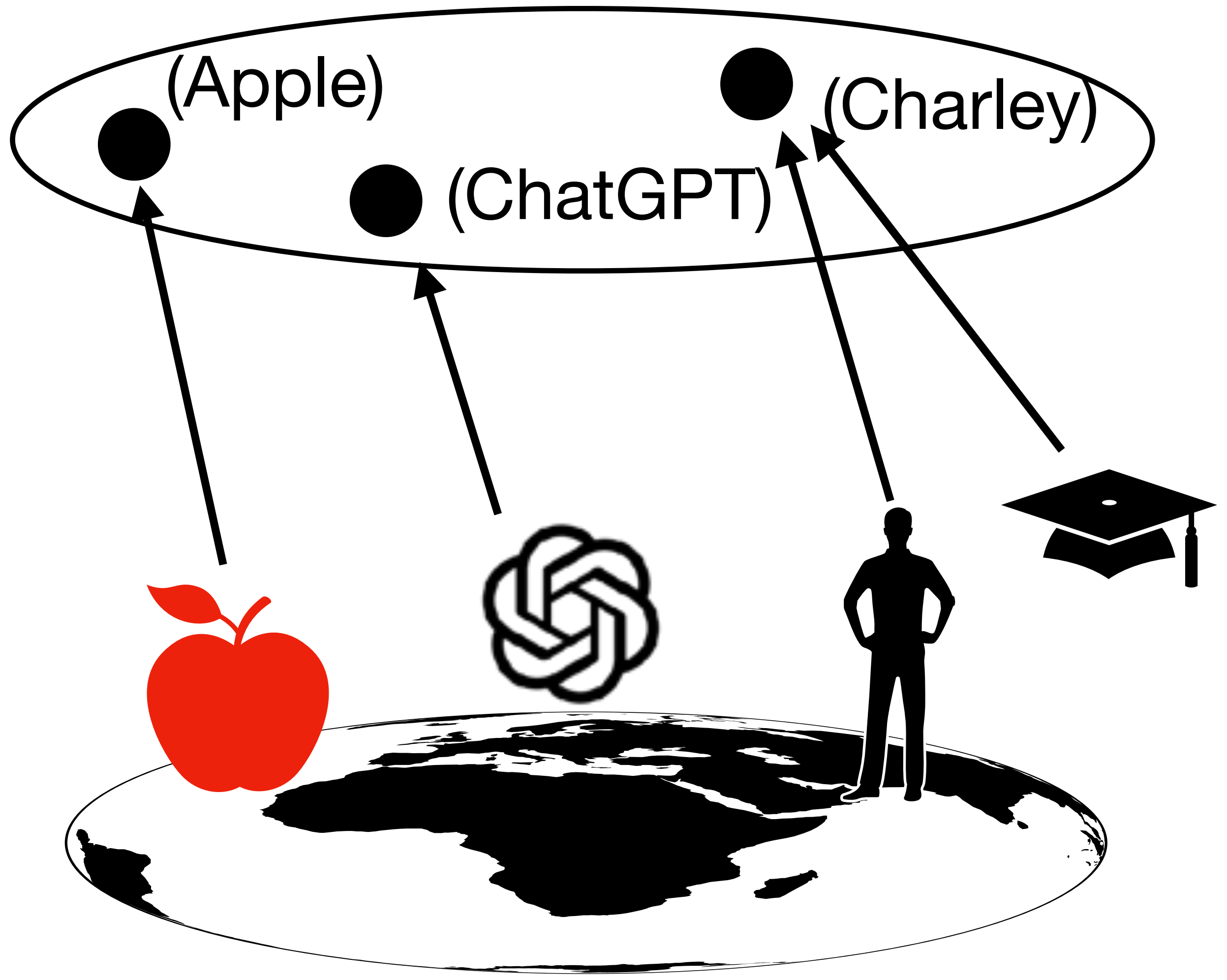


Symbolic AI

Herbert Simon
& Allen Newell

- **Physical Symbol System hypothesis**:
  "A physical symbol system has the necessary and sufficient means for general intelligent action - Allen Newell and Herbert Simon (1976)"

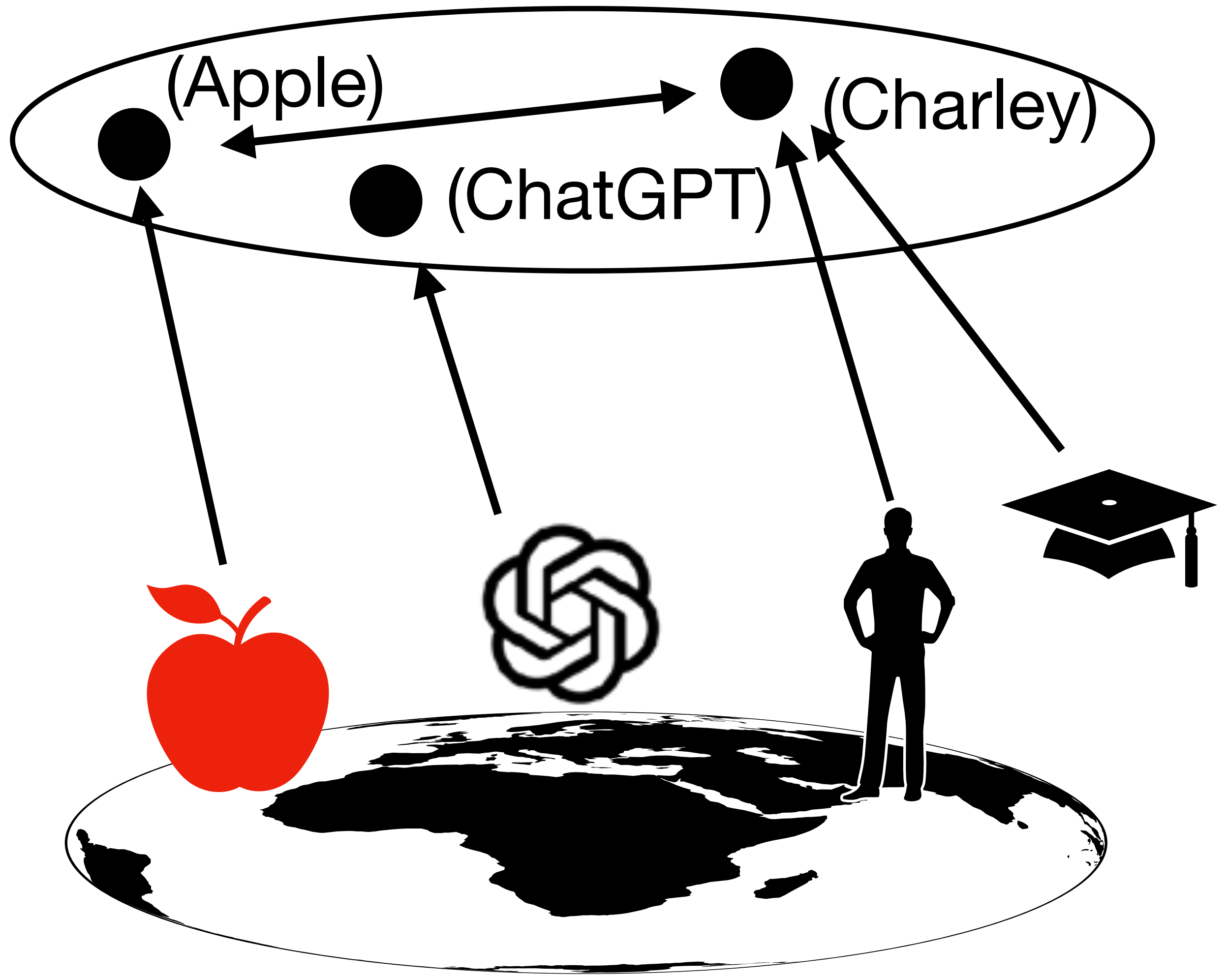
- **Symbols** can represent things in the world
  - e.g., (Apple), (ChatGPT), (Charley), etc…
- **Relations** can be i) predicates that describes a symbol or ii) verbs describing how symbols interact with other symbols
  - i) red(Apple), unreliable(ChatGPT), instructor(Charley)
  - ii) eat(Charley, Apple), generatePicture(ChatGPT, Apple)

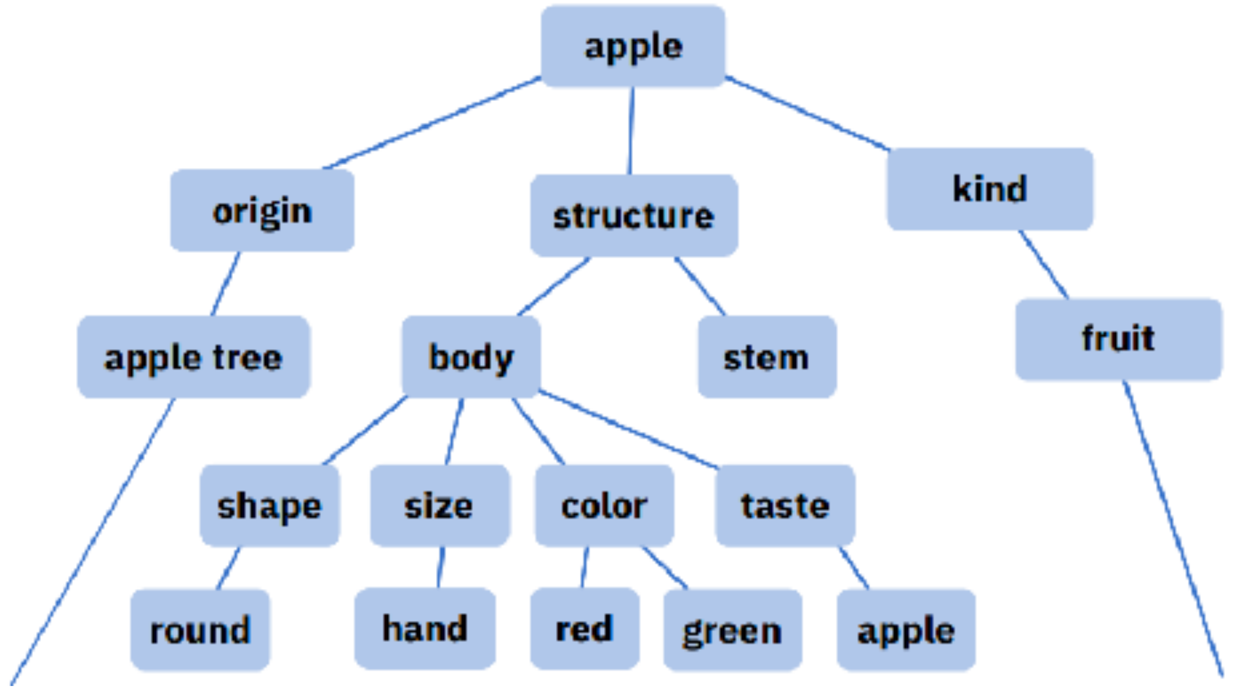


Symbolic AI

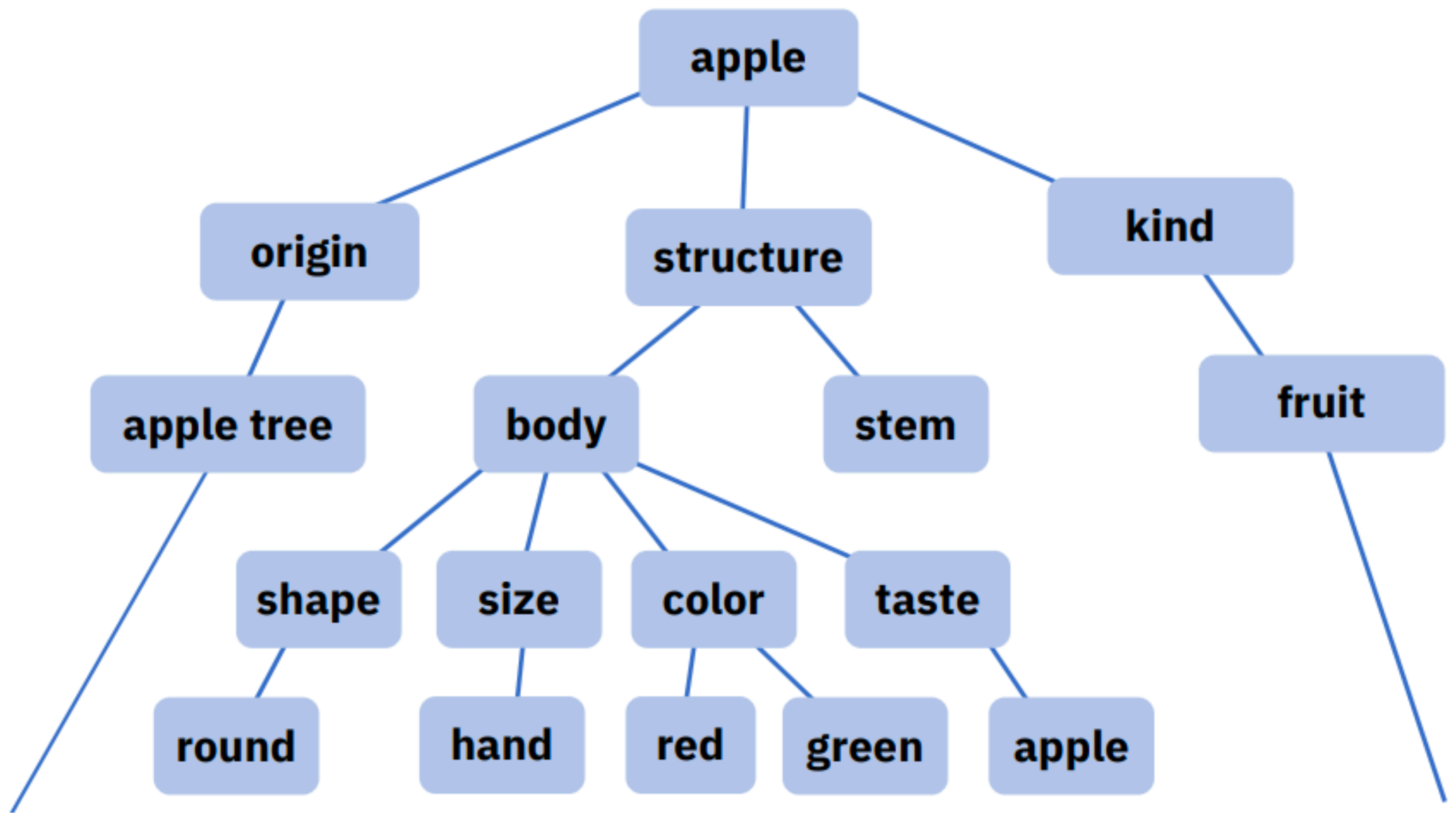
Herbert Simon
& Allen Newell

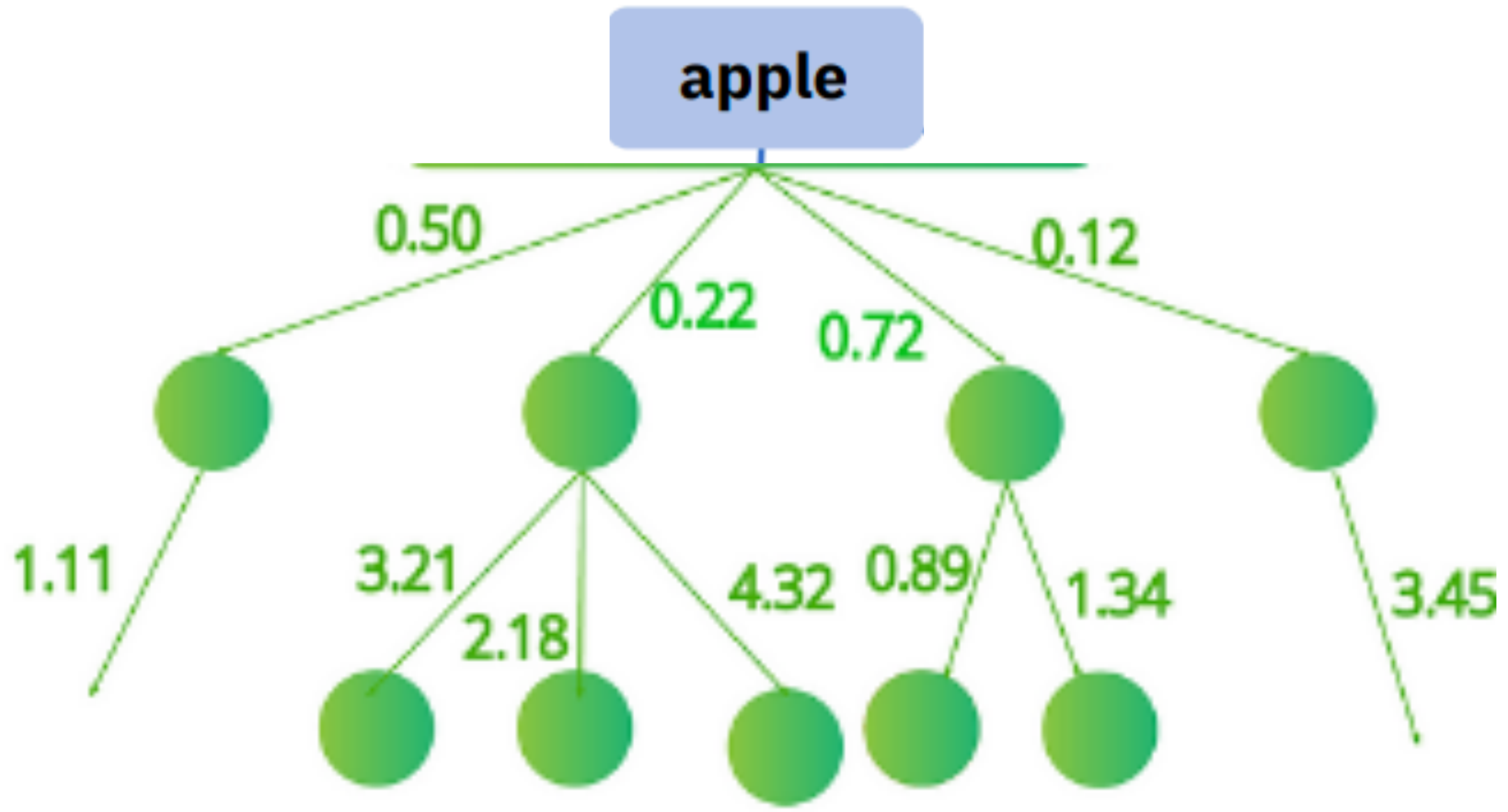
- **Physical Symbol System hypothesis**:
  "A physical symbol system has the necessary and sufficient means for general intelligent action - Allen Newell and Herbert Simon (1976)"
  - **Symbols** can represent things in the world
    - e.g., (Apple), (ChatGPT), (Charley), etc…
  - **Relations** can be i) predicates that describes a symbol or ii) verbs describing how symbols interact with other symbols
    - i) red(Apple), unreliable(ChatGPT), instructor(Charley)
    - ii) eat(Charley, Apple), generatePicture(ChatGPT, Apple)





Symbolic vs. sub-symbolic AI



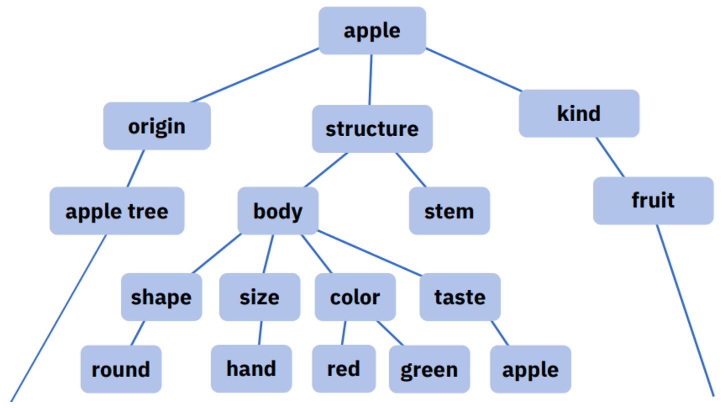


Symbolic AI

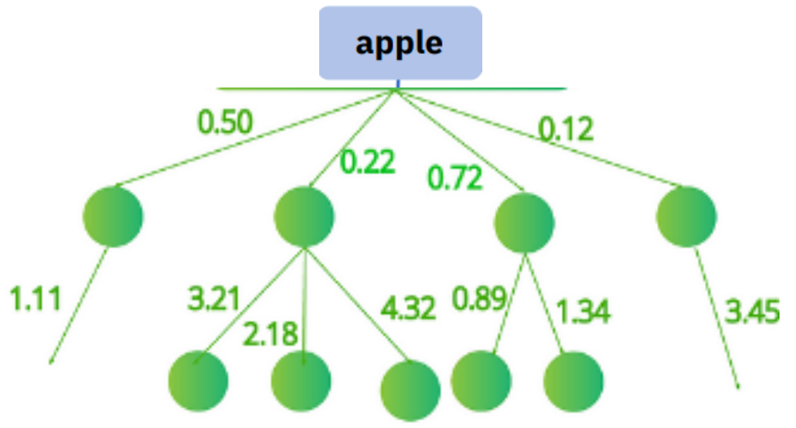
- **Symbols and relations** represent things in the world, and reasoning is just the manipulation of these entities
- Compositionality: symbols and rules can be combined to produce new representations
  - "Language of thought" (LoT) hypothesis (Fodor, 1975): concepts/knowledge represented by a language-like system
- Extracting symbolic representations and search over compositional hypothesis spaces is difficult

Sub-symbolic AI

- Representations **distributed** across connection weights, but the weights themselves don't explicitly represent anything
- Efficiency: knowledge can be implicitly learned by capturing statistical patterns
- Interpretation of representations and behavior is difficult

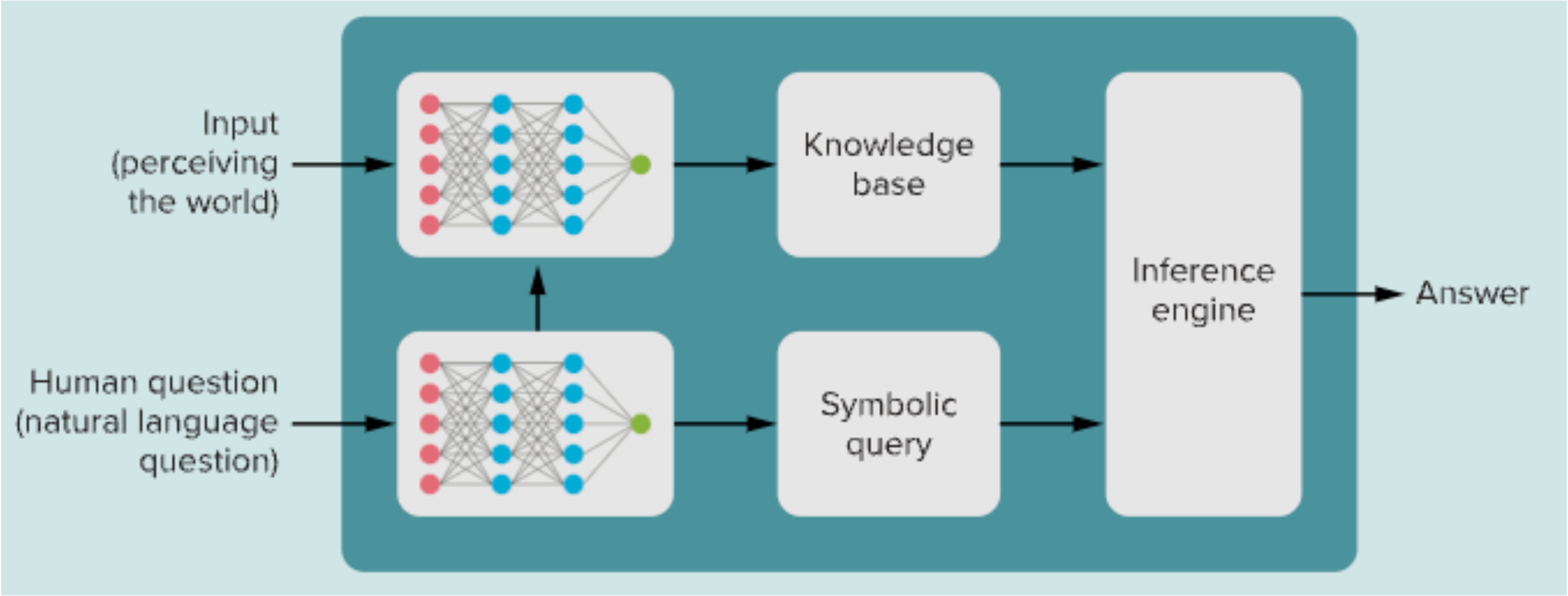


+



Neurosymbolic AI

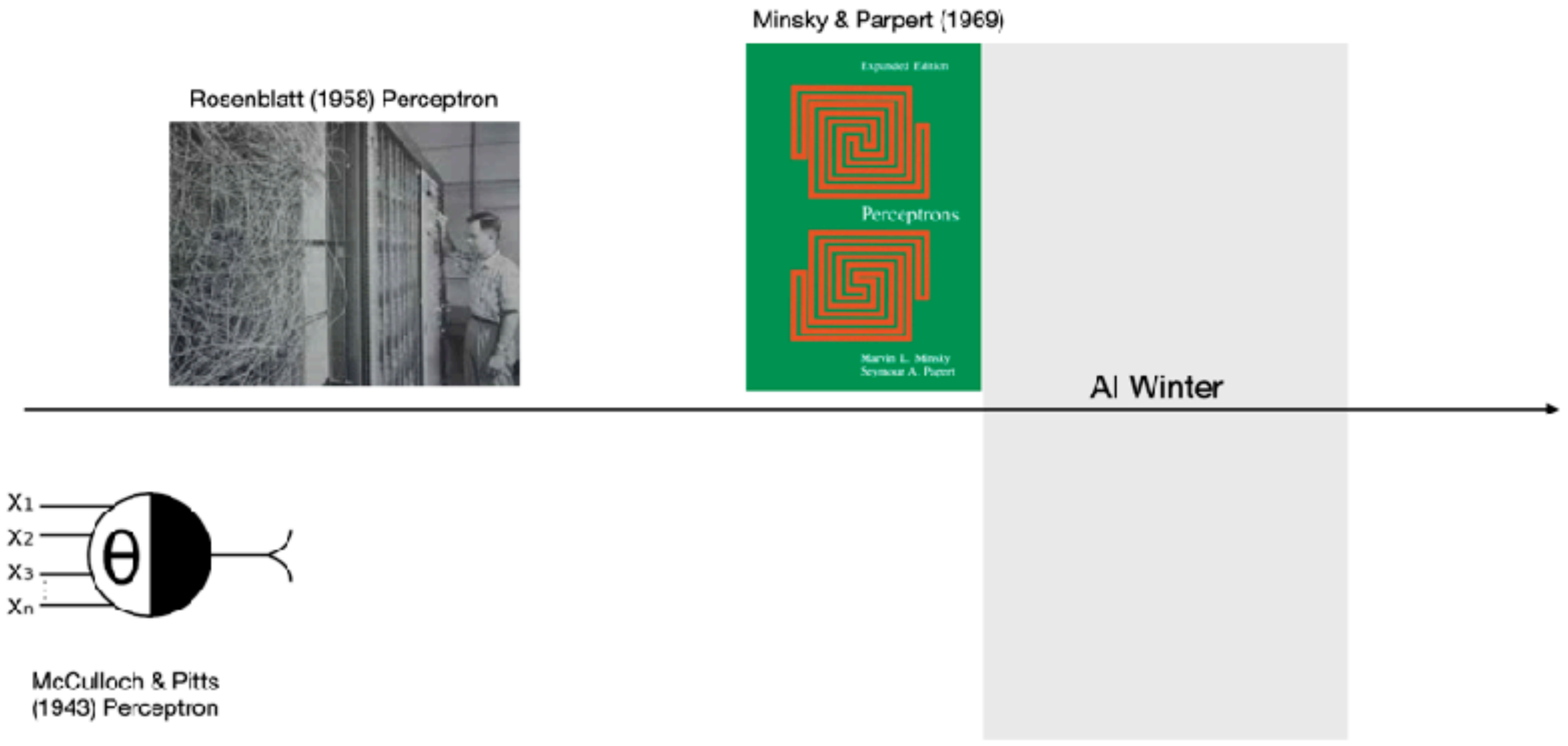
- Neurosymbolic AI aims to combine symbolic and subsymbolic approaches to get the best of both worlds
- Modern AI assistants (e.g., Siri, Google, Alexa) are essentially expert systems with ANN voice recognition and text-to-speech

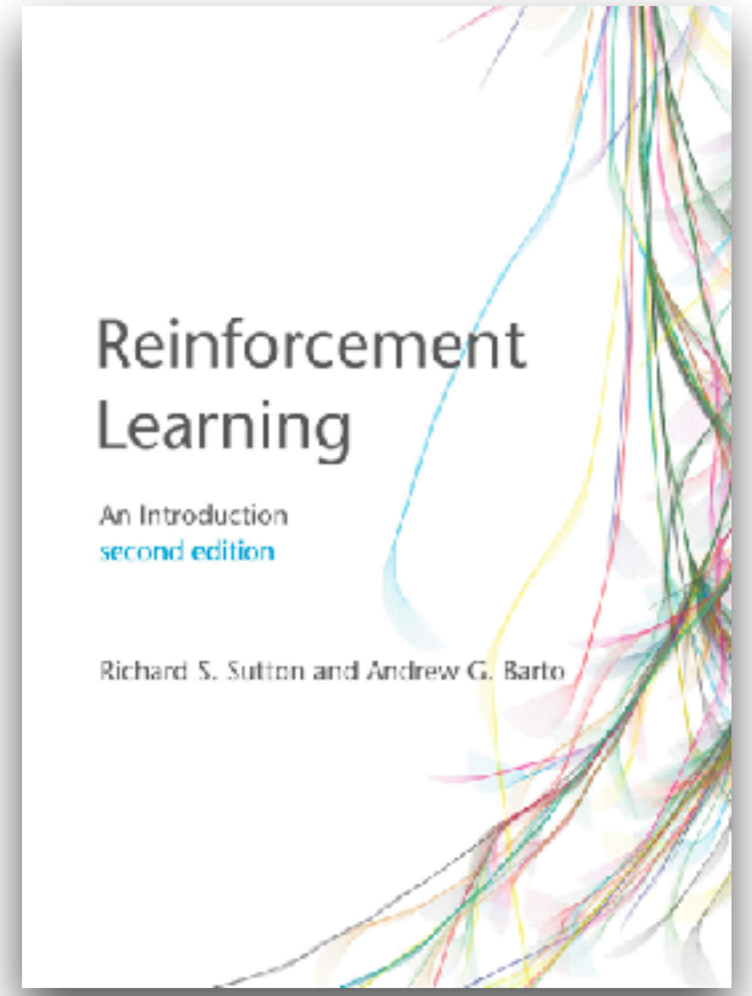


A common framework of learning?

Early biological research

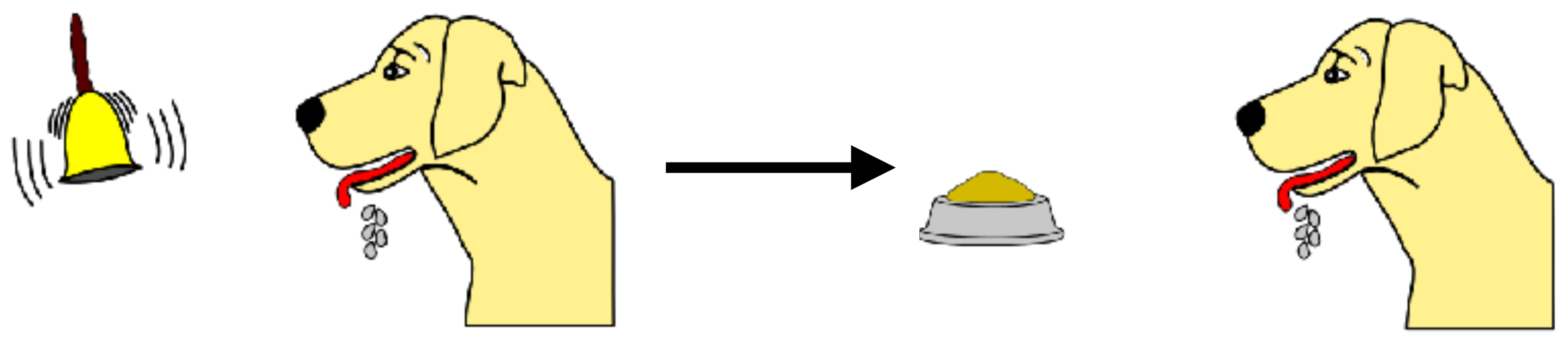
Early AI research



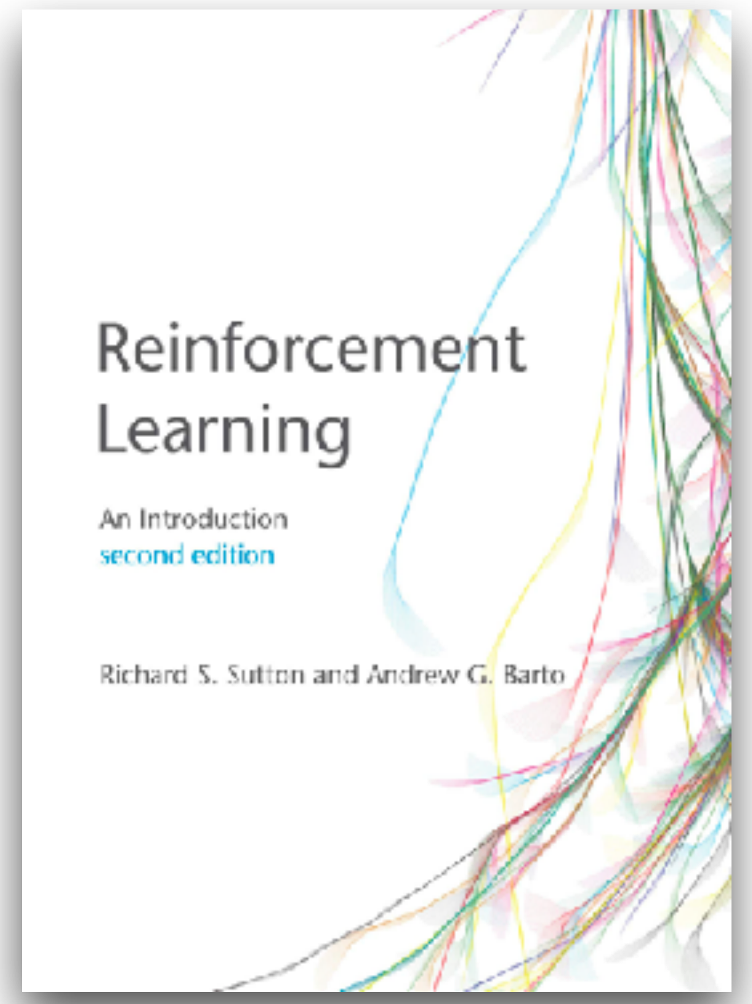


Reinforcement Learning

Pavlovian (classical) conditioning

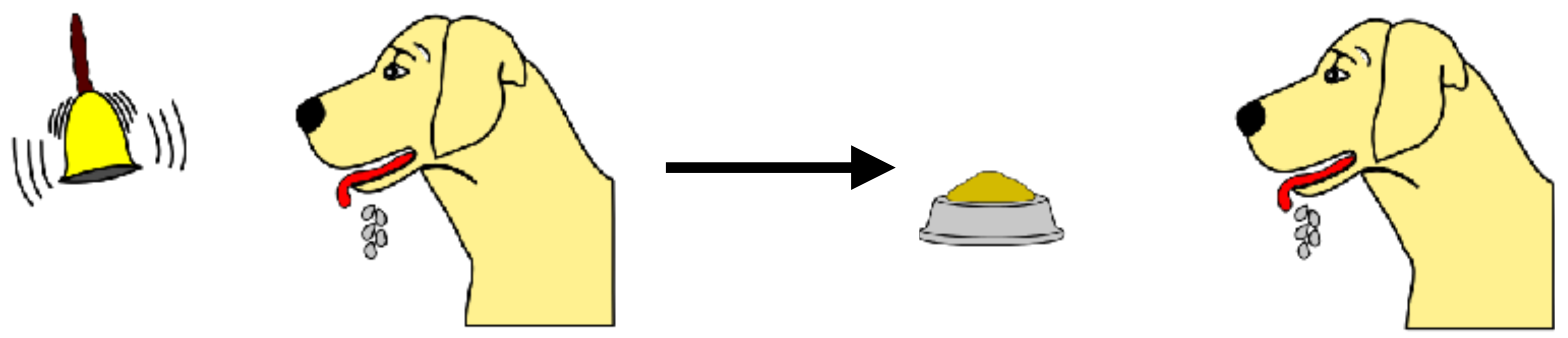


Learn which environmental cues *predict* reward

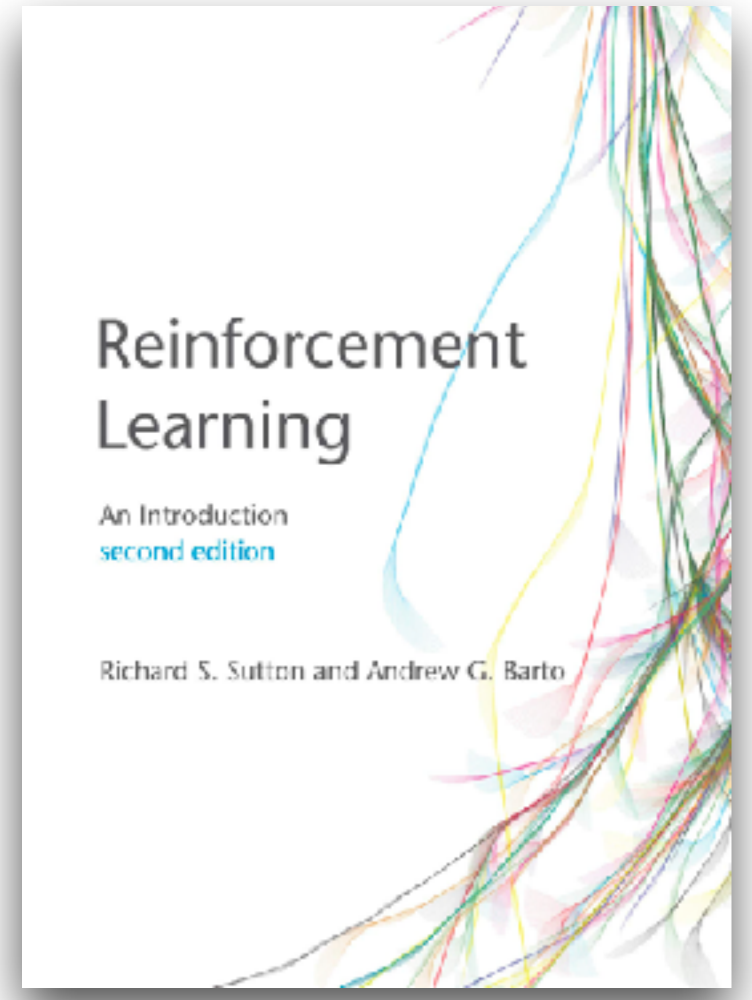


Reinforcement Learning

Pavlovian (classical) conditioning

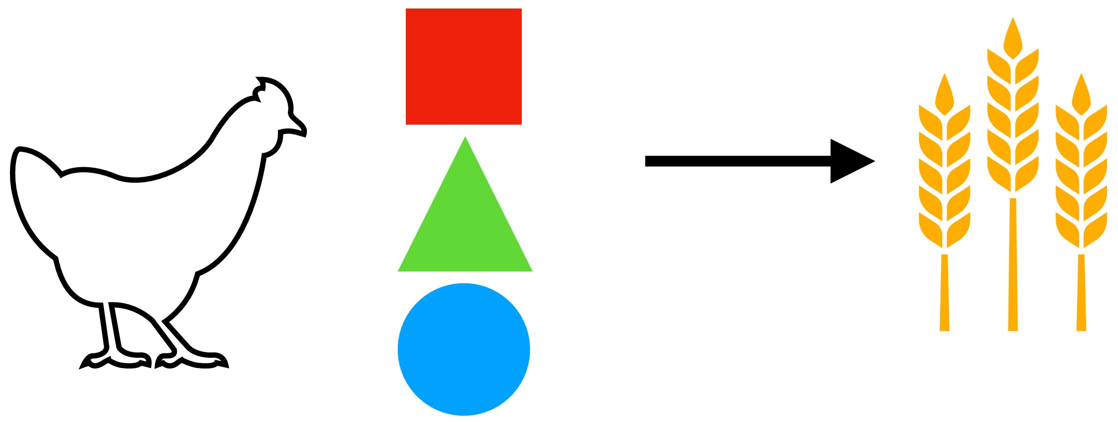


Learn which environmental cues *predict* reward

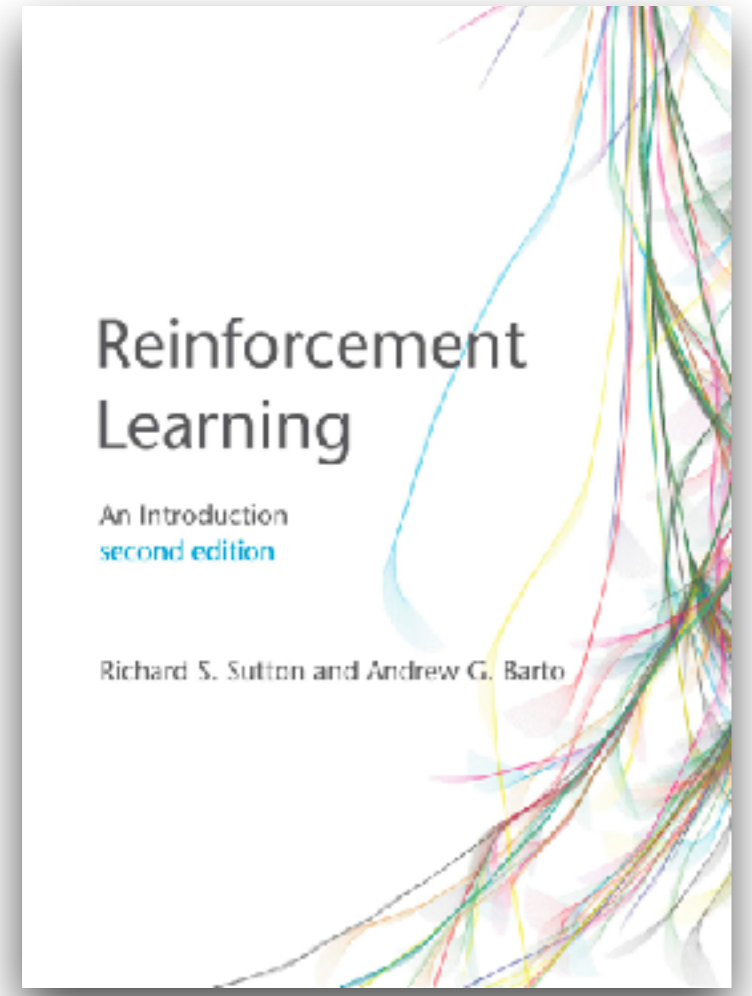


Reinforcement Learning

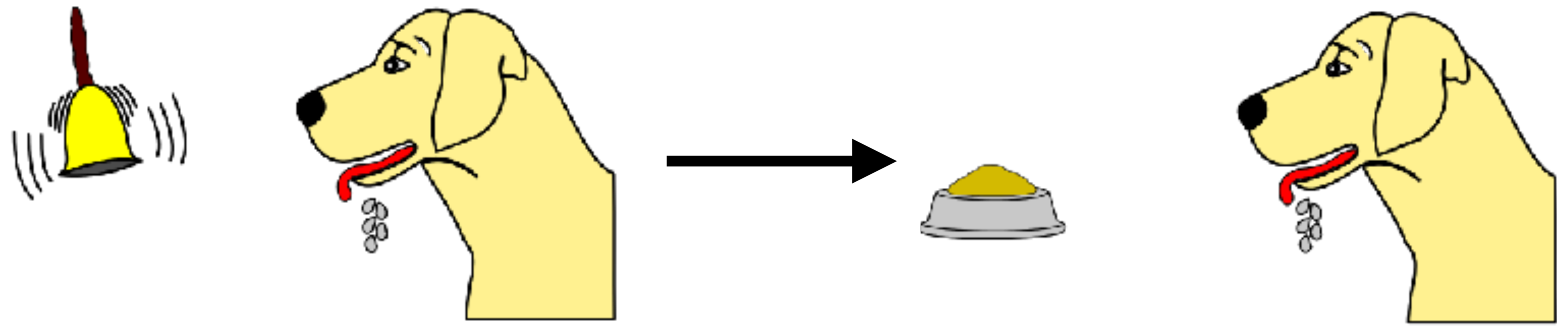
Operant (instrumental) conditioning



Learn which actions *predict* reward



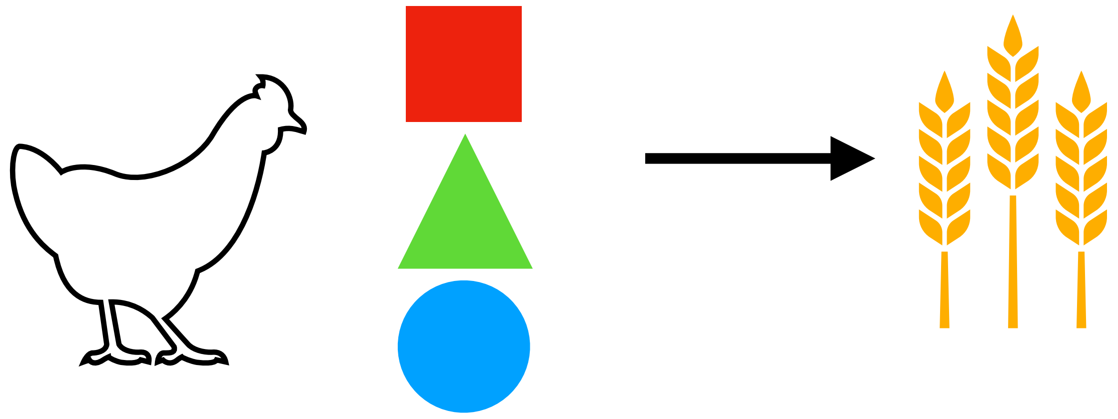
Pavlovian (classical) conditioning



Learn which environmental cues *predict* reward

Reinforcement Learning

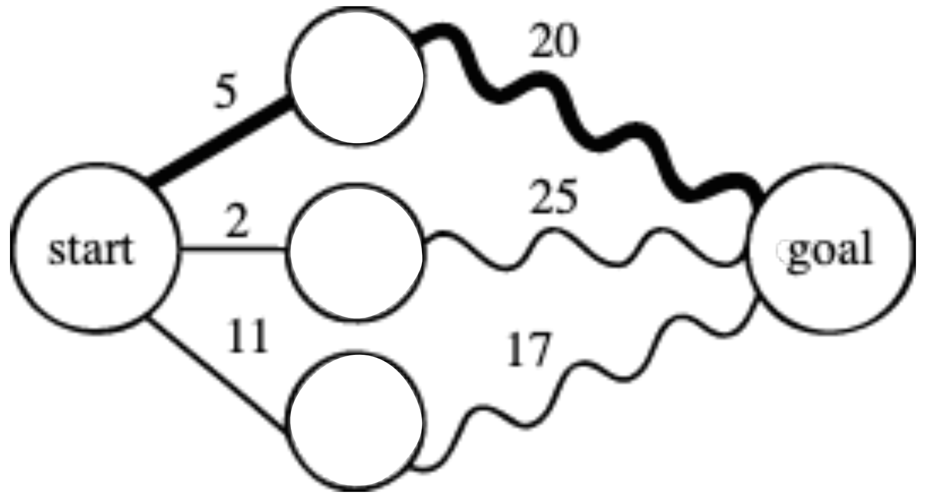
Operant (instrumental) conditioning



Learn which actions *predict* reward

Neuro-dynamic programing
Bertsekas & Tsitsiklis (1996)

Stochastic approximations to dynamic programing problems



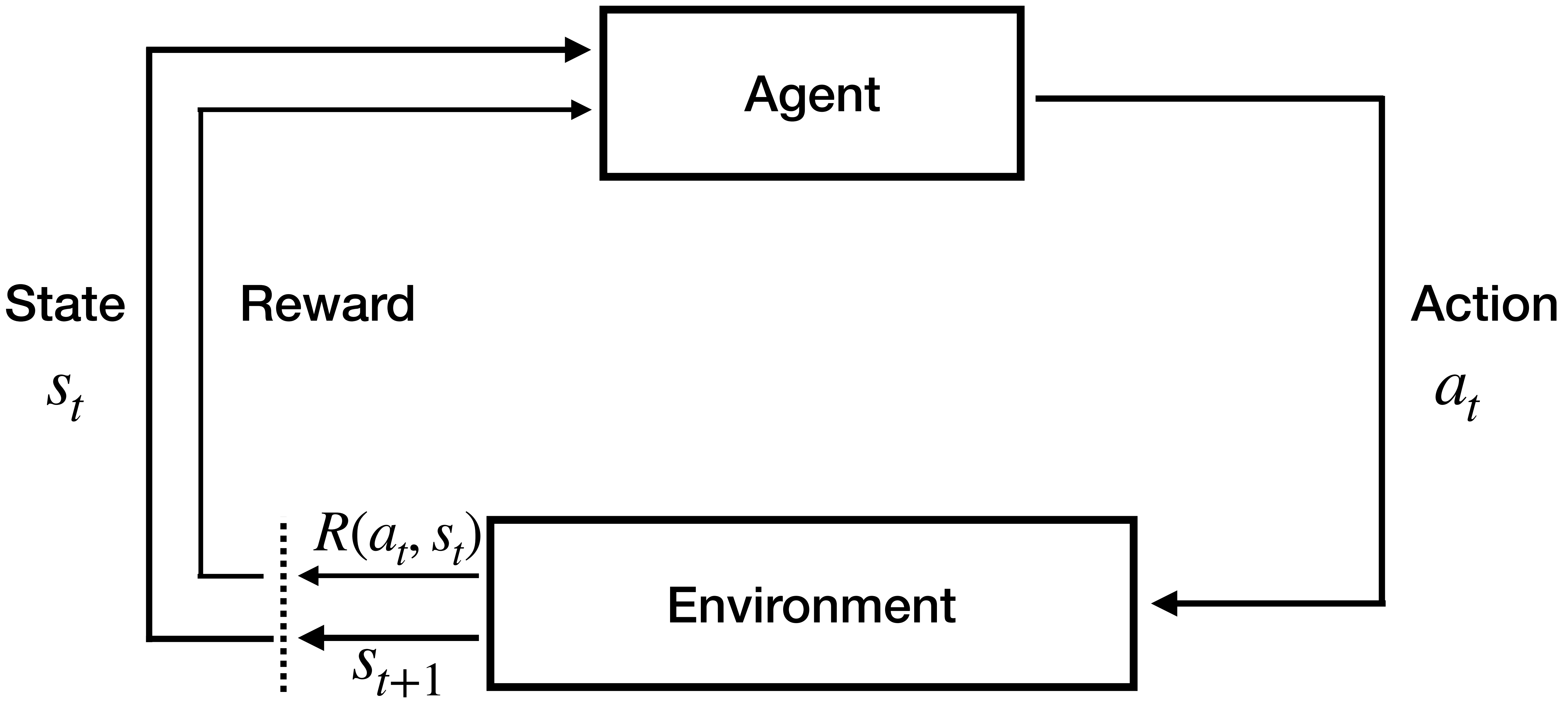
Reinforcement Learning

The Agent:

- Iteratively selects actions a_t based on a policy π
- Receives feedback from the environment in terms of new states s_{t+1} and rewards $R(a_t, s_t)$
- Updates internal representations
    - value $Q(s, a)$ or $V(s)$
    - model of the environment
        - reward function R
        - transitions $T(s' \mid s)$

The Environment:

- governs the transition between states $s_t \rightarrow s_{t+1}$
- provides rewards $R(a_t, s_t)$



Sutton and Barto (2018 [1998])

Reinforcement Learning

The Agent:

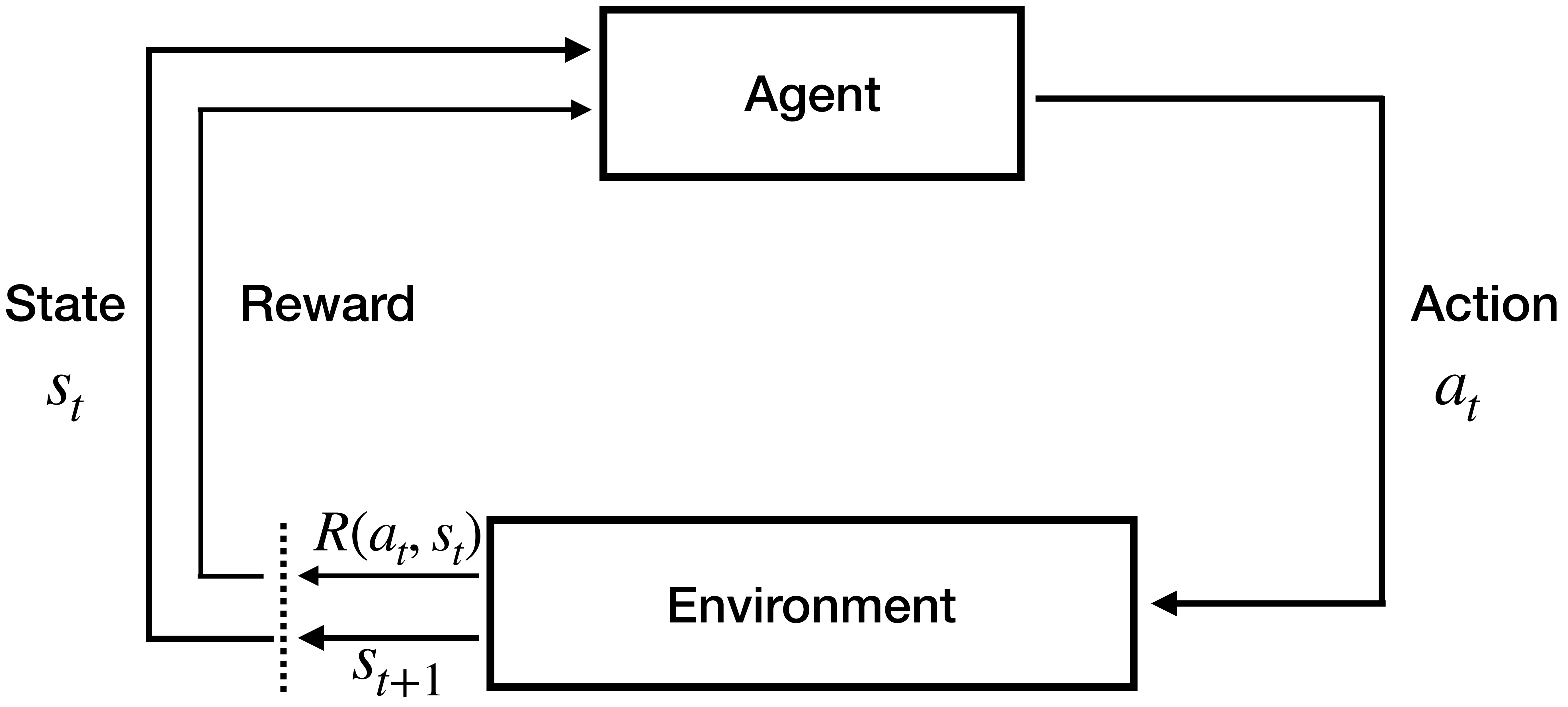
- Iteratively selects actions a_t based on a policy π
- Receives feedback from the environment in terms of new states s_{t+1} and rewards $R(a_t, s_t)$
- Updates internal representations
  - value $Q(s, a)$ or $V(s)$
  - model of the environment
    - reward function R
    - transitions $T(s' \mid s)$

The Environment:

- governs the transition between states $s_t \rightarrow s_{t+1}$
- provides rewards $R(a_t, s_t)$

Delta-rule of learning

Belief-updates are proportional to the magnitude of the reward predition error (RPE)

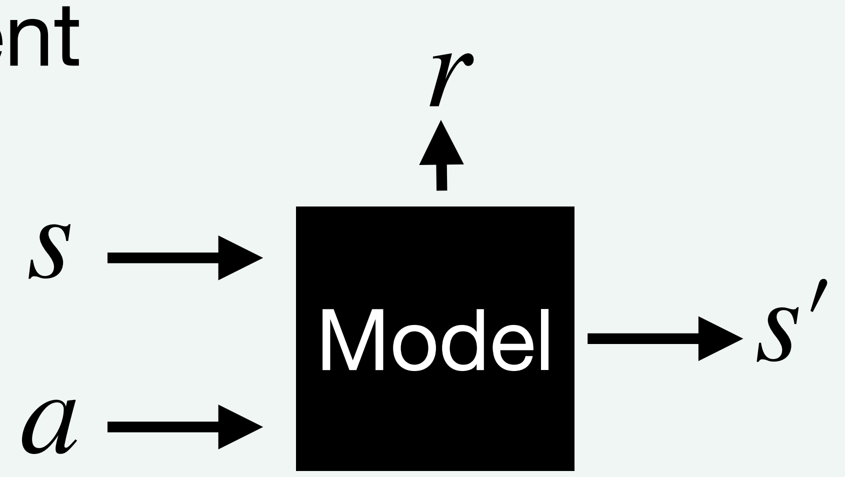


Sutton and Barto (2018 [1998])

Reinforcement Learning

The Agent:

- Iteratively selects actions a_t based on a policy π
- Receives feedback from the environment in terms of new states s_{t+1} and rewards $R(a_t, s_t)$
- Updates internal representations
  - value $Q(s, a)$ or $V(s)$
  - model of the environment
    - reward function R
    - transitions $T(s' | s)$

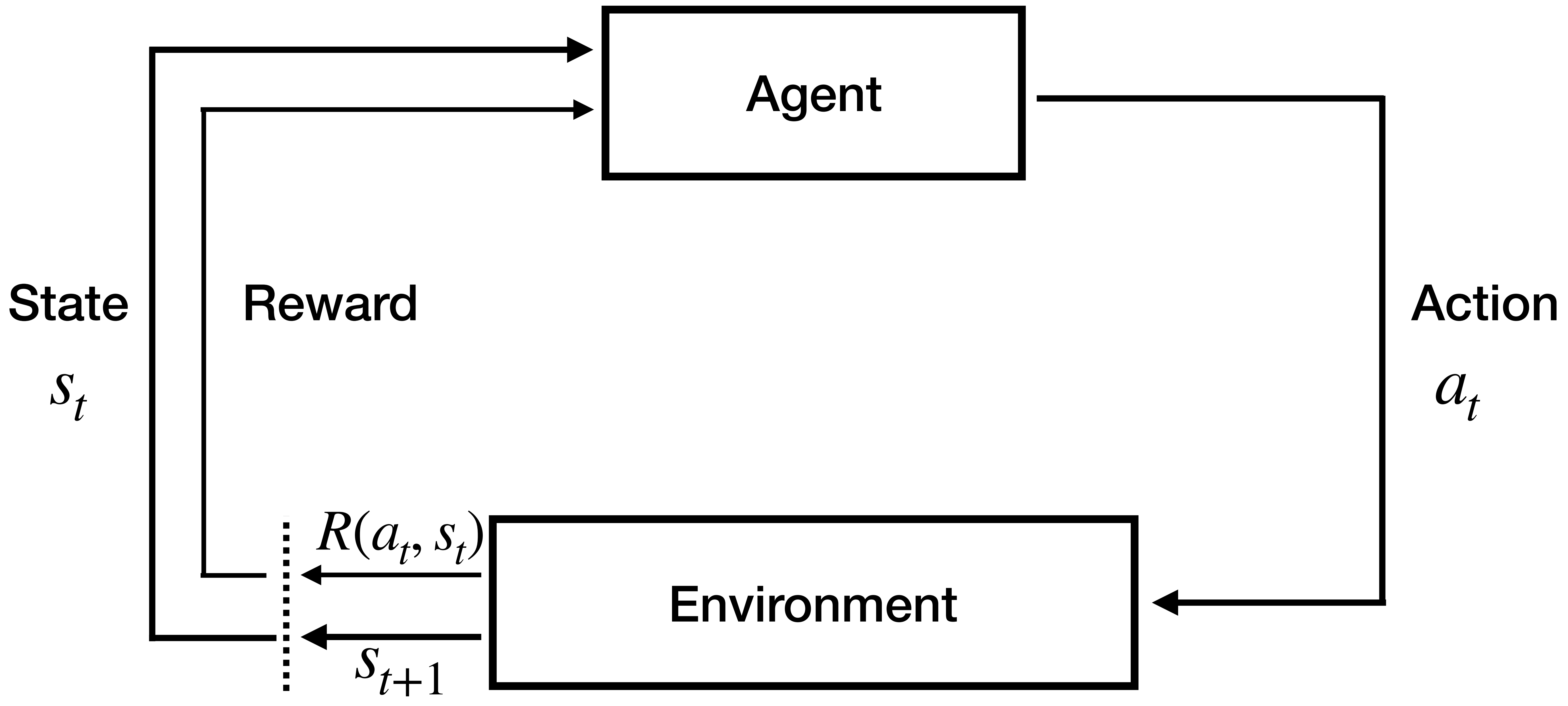


The Environment:

- governs the transition between states $s_t \rightarrow s_{t+1}$
- provides rewards $R(a_t, s_t)$

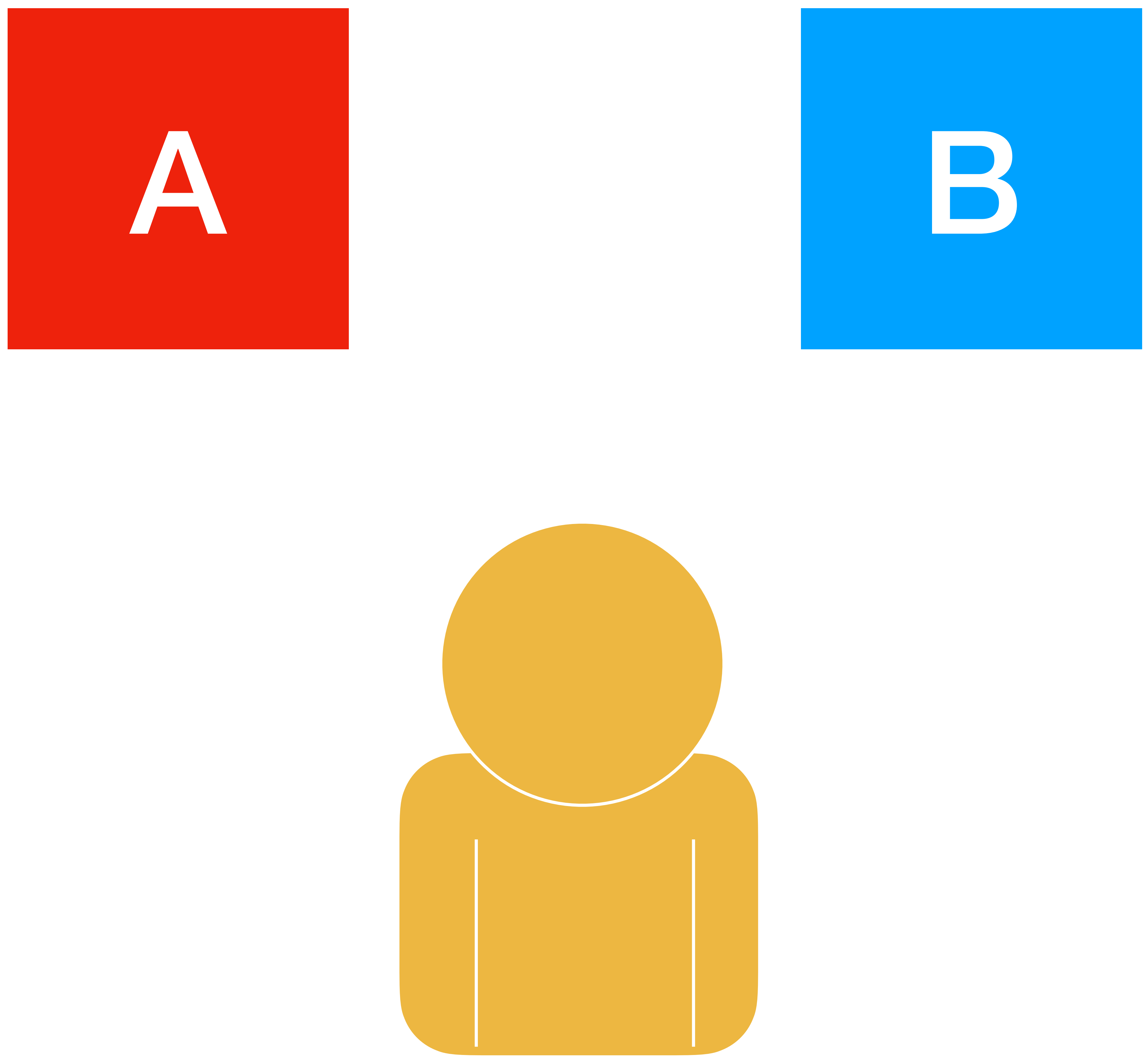
Delta-rule of learning

Belief-updates are proportional to the magnitude of the reward predition error (RPE)

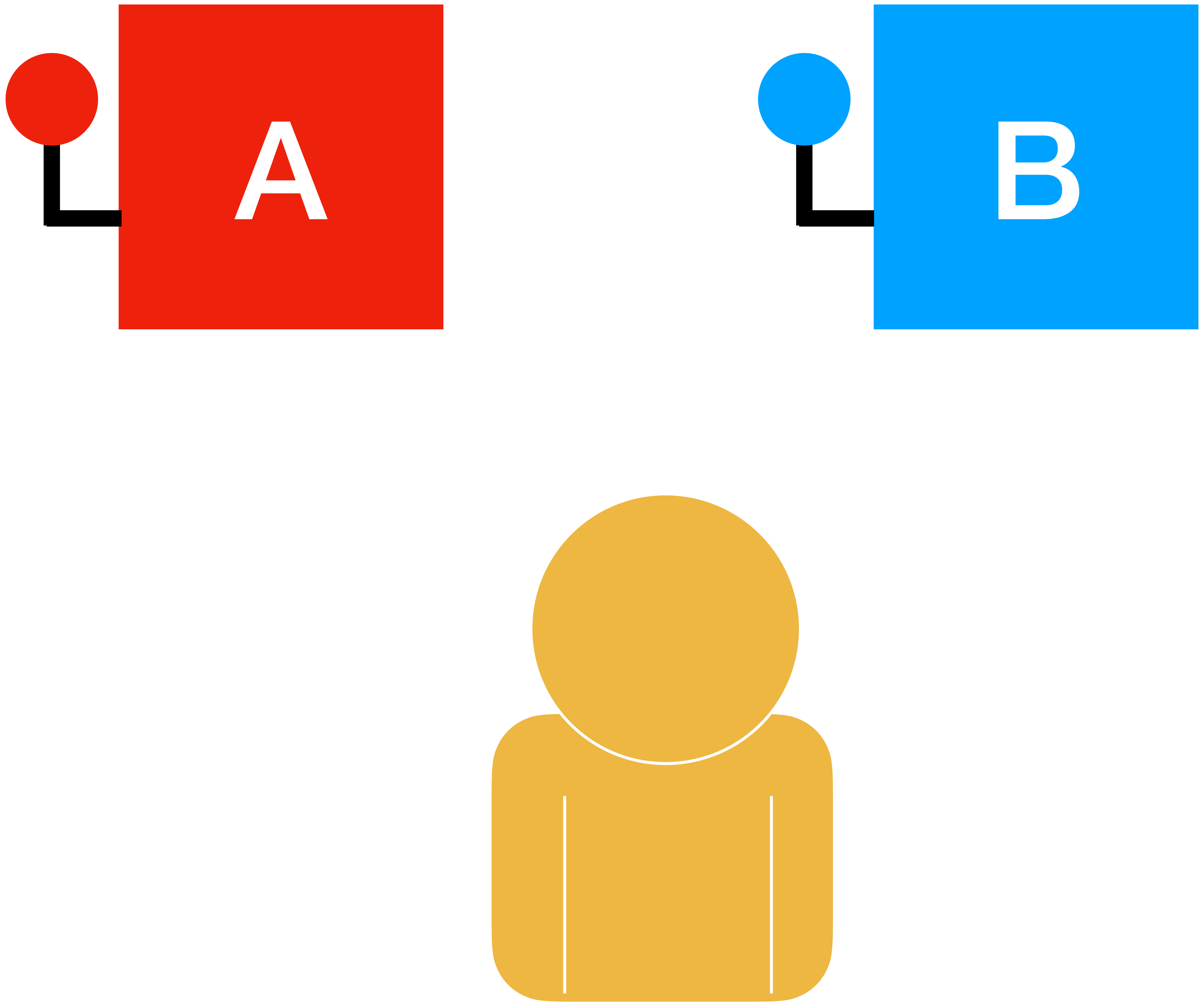


Sutton and Barto (2018 [1998])

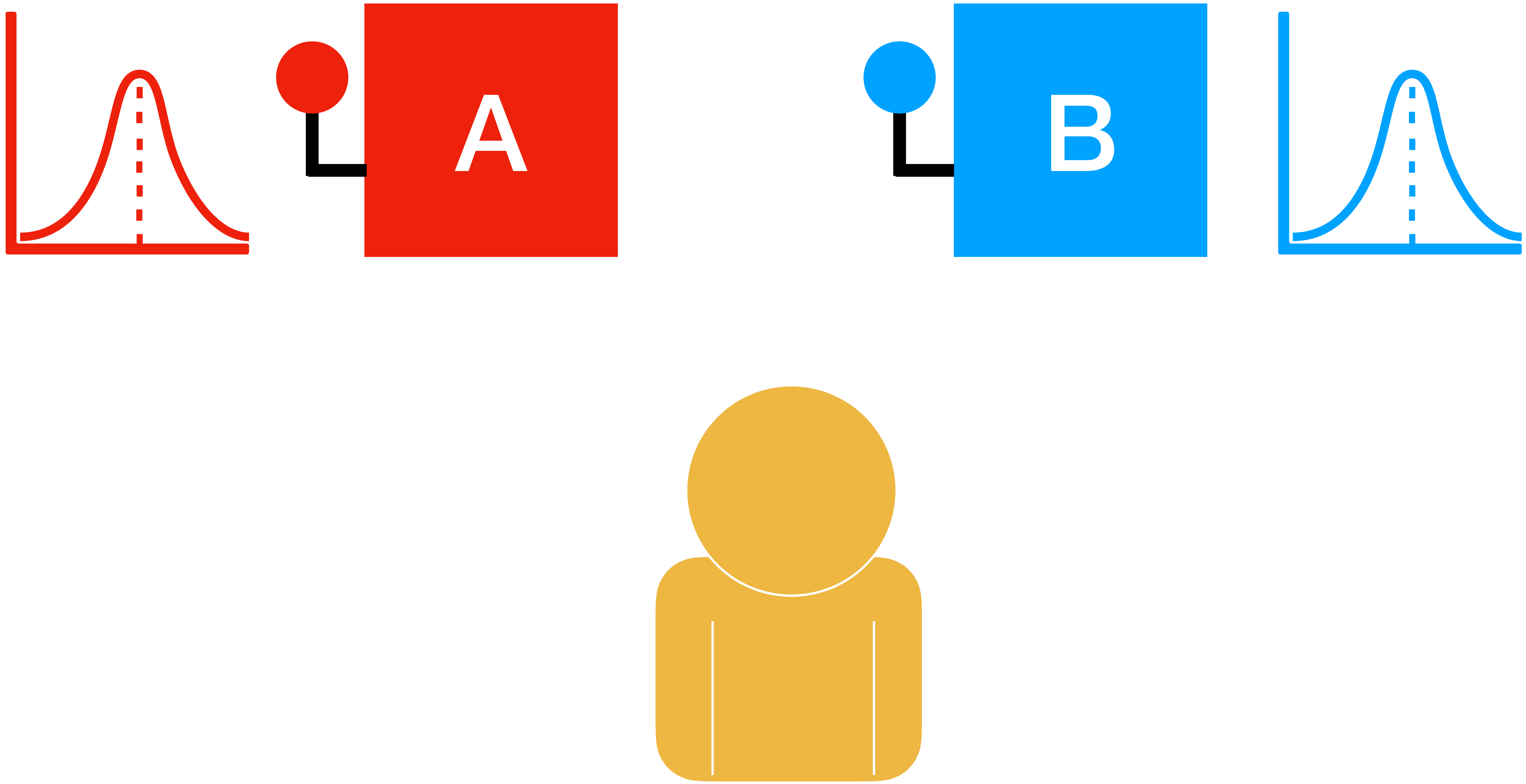
2-Armed Bandit Problem



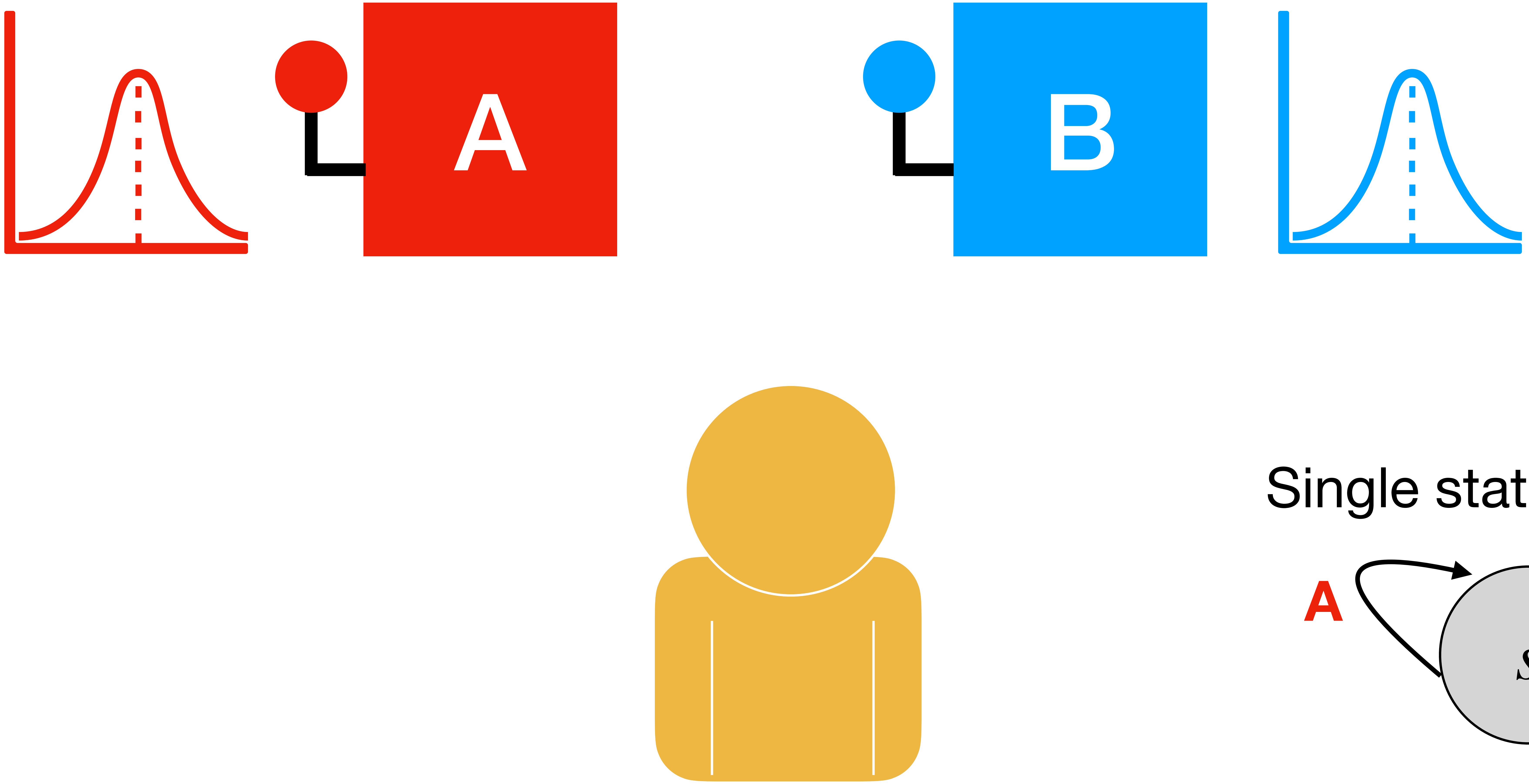
2-Armed Bandit Problem



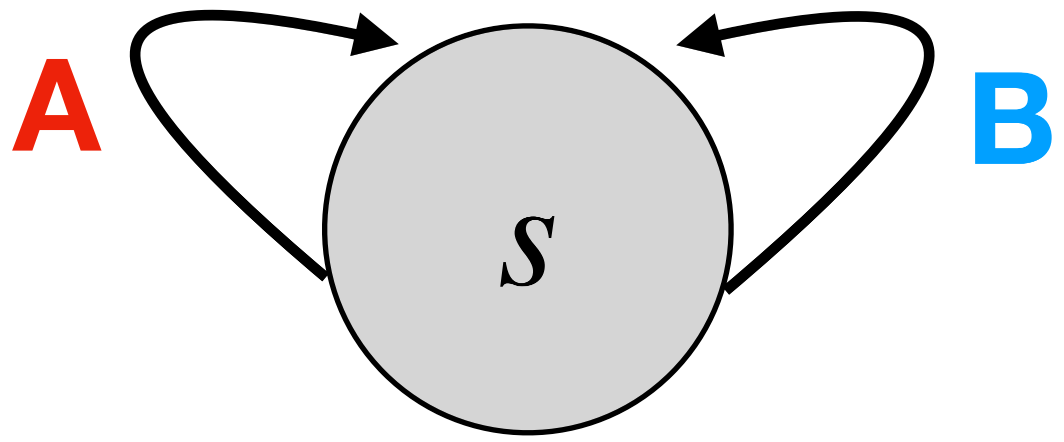
2-Armed Bandit Problem



2-Armed Bandit Problem



Single state problem



Q-Learning (Watkins, 1989)

Value learning

Q-Learning (Watkins, 1989)

Value learning

$$Q_{t+1}(a) \leftarrow Q_t(a) + \eta \left[r - Q_t(a) \right]$$

Q-Learning (Watkins, 1989)

Value learning

$$Q_{t+1}(a) \leftarrow Q_t(a) + \eta \, \underbrace{\left[\underset{\substack{\uparrow \\ \text{Observed reward}}}{r} - \underset{\substack{\uparrow \\ \text{Predicted reward}}}{Q_t(a)} \right]}_{\delta}$$

Reward prediction error (RPE)

Q-Learning (Watkins, 1989)

Value learning

$$Q_{t+1}(a) \leftarrow Q_t(a) + \eta \, [\underbrace{r - Q_t(a)}_{\delta}]$$

Observed reward

Predicted reward

δ

Reward prediction error (RPE)

The delta-rule of learning:

- Learning occurs only when events violate expectations ($\delta \neq 0$)
- The magnitude of the error corresponds to how much we update our beliefs

Q-Learning (Watkins, 1989)

Value learning

$$Q_{t+1}(a) \leftarrow Q_t(a) + \eta\left[r - Q_t(a)\right]$$

learning rate

Observed reward

Predicted reward

δ

Reward prediction error (RPE)

The delta-rule of learning:

- Learning occurs only when events violate expectations ($\delta \neq 0$)
- The magnitude of the error corresponds to how much we update our beliefs

Q-Learning (Watkins, 1989)

Value learning

$$Q_{t+1}(a) \leftarrow Q_t(a) + \eta \left[r - Q_t(a) \right]$$

learning rate

Observed reward

Predicted reward

δ

Reward prediction error (RPE)

The delta-rule of learning:

- Learning occurs only when events violate expectations ($\delta \neq 0$)
- The magnitude of the error corresponds to how much we update our beliefs

Exercise 1: Compute Q-values

A B

assume $\eta = .9$

	$Q(A)$	$Q(B)$	a	r	δ
t=1	0	0	A	5	
t=2			B	12	
t=3			B	4	
t=4			A	8	

Q-Learning (Watkins, 1989)

Value learning

$$Q_{t+1}(a) \leftarrow Q_t(a) + \underset{\text{learning rate}}{\eta} \underbrace{[\underset{\text{Observed reward}}{r} - \underset{\text{Predicted reward}}{Q_t(a)}]}_{\delta \text{ Reward prediction error (RPE)}}$$

Exercise 1: Compute Q-values

A B assume $\eta = .9$

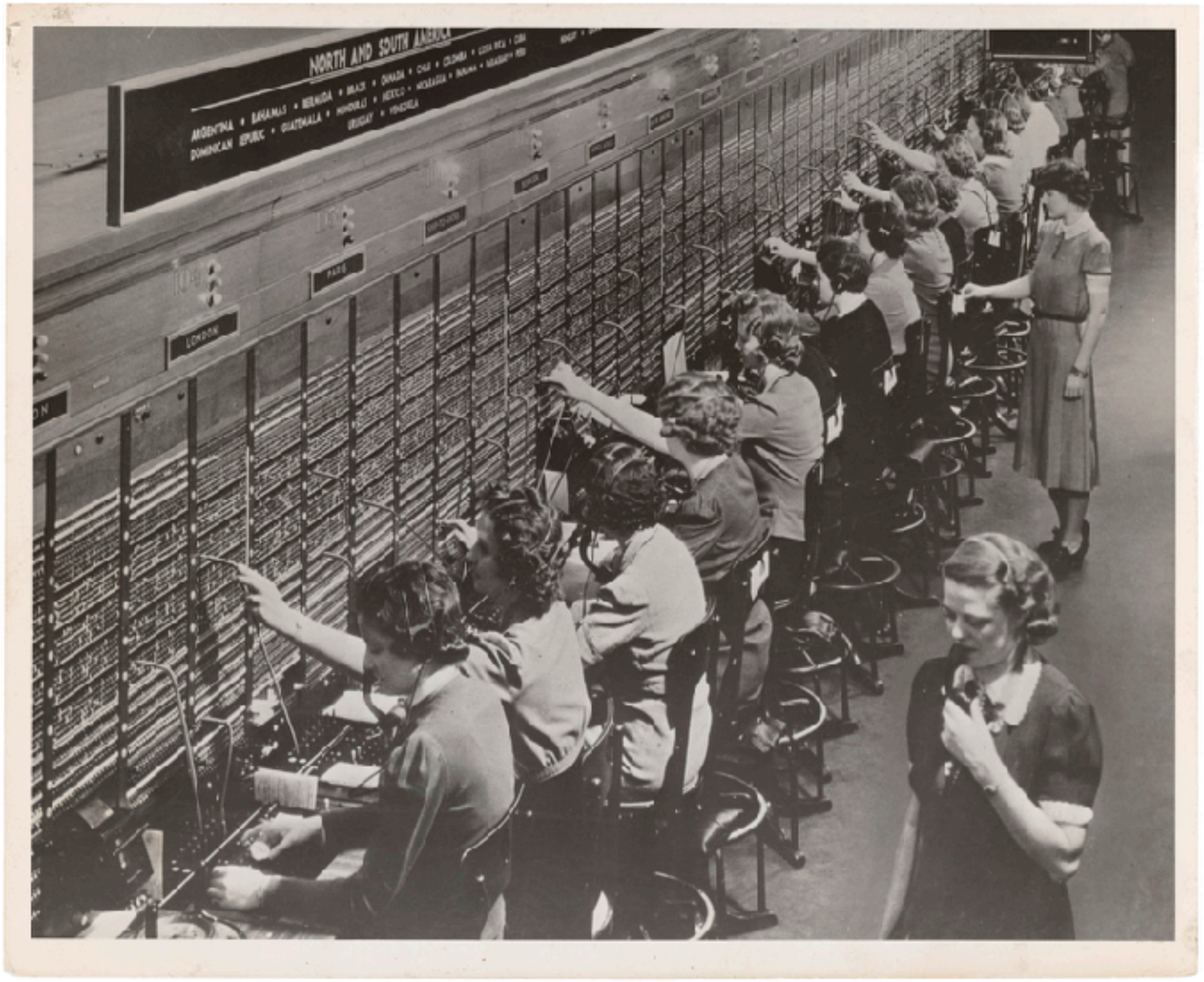
	$Q(A)$	$Q(B)$	a	r	δ
t=1	0	0	A	5	5
t=2	4.5	0	B	12	12
t=3	4.5	10.8	B	4	-6.8
t=4	4.5	4.68	A	8	3.5

The delta-rule of learning:

- Learning occurs only when events violate expectations ($\delta \neq 0$)
- The magnitude of the error corresponds to how much we update our beliefs

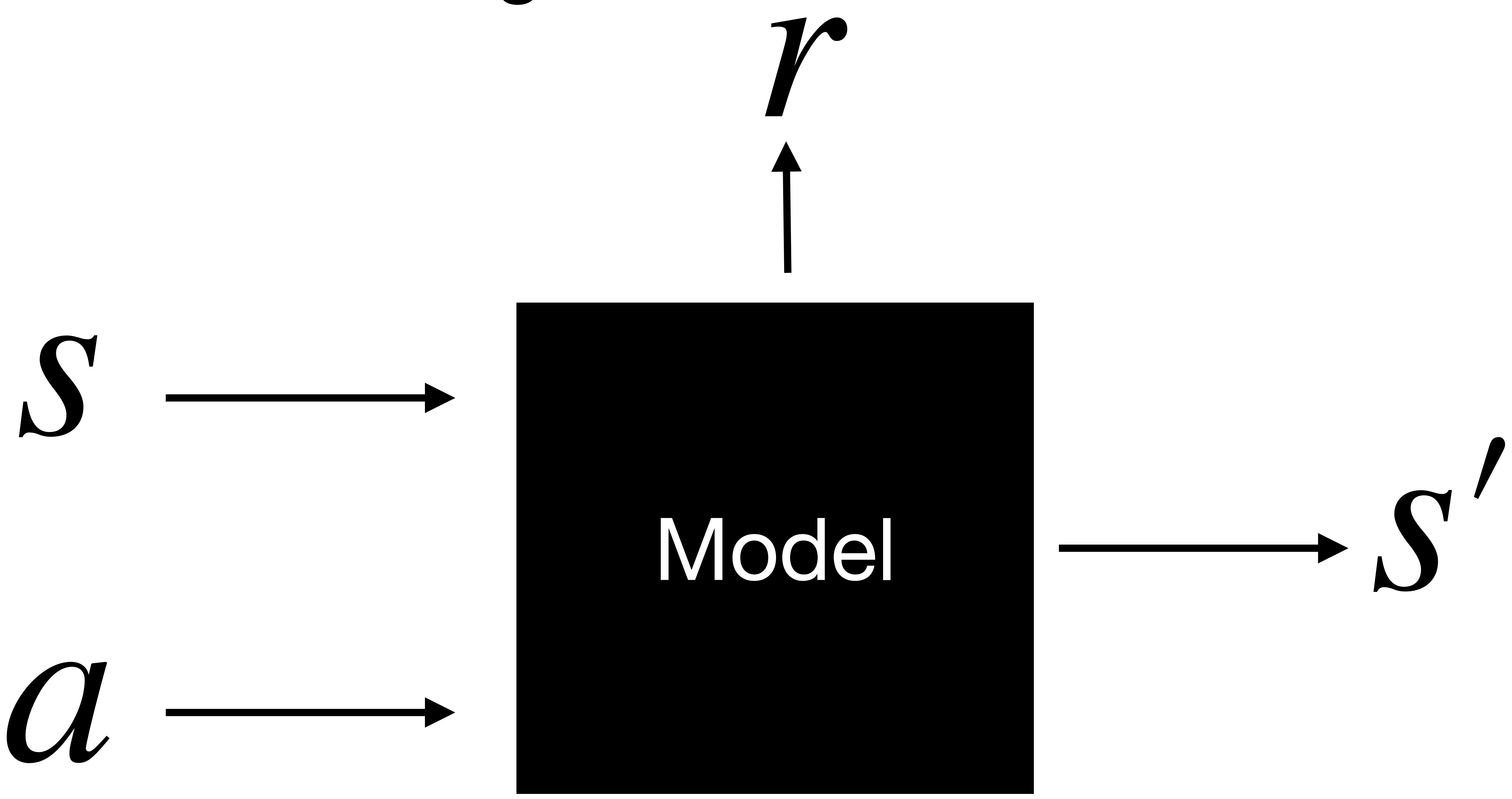
Model-free

S-R learning



Model-based

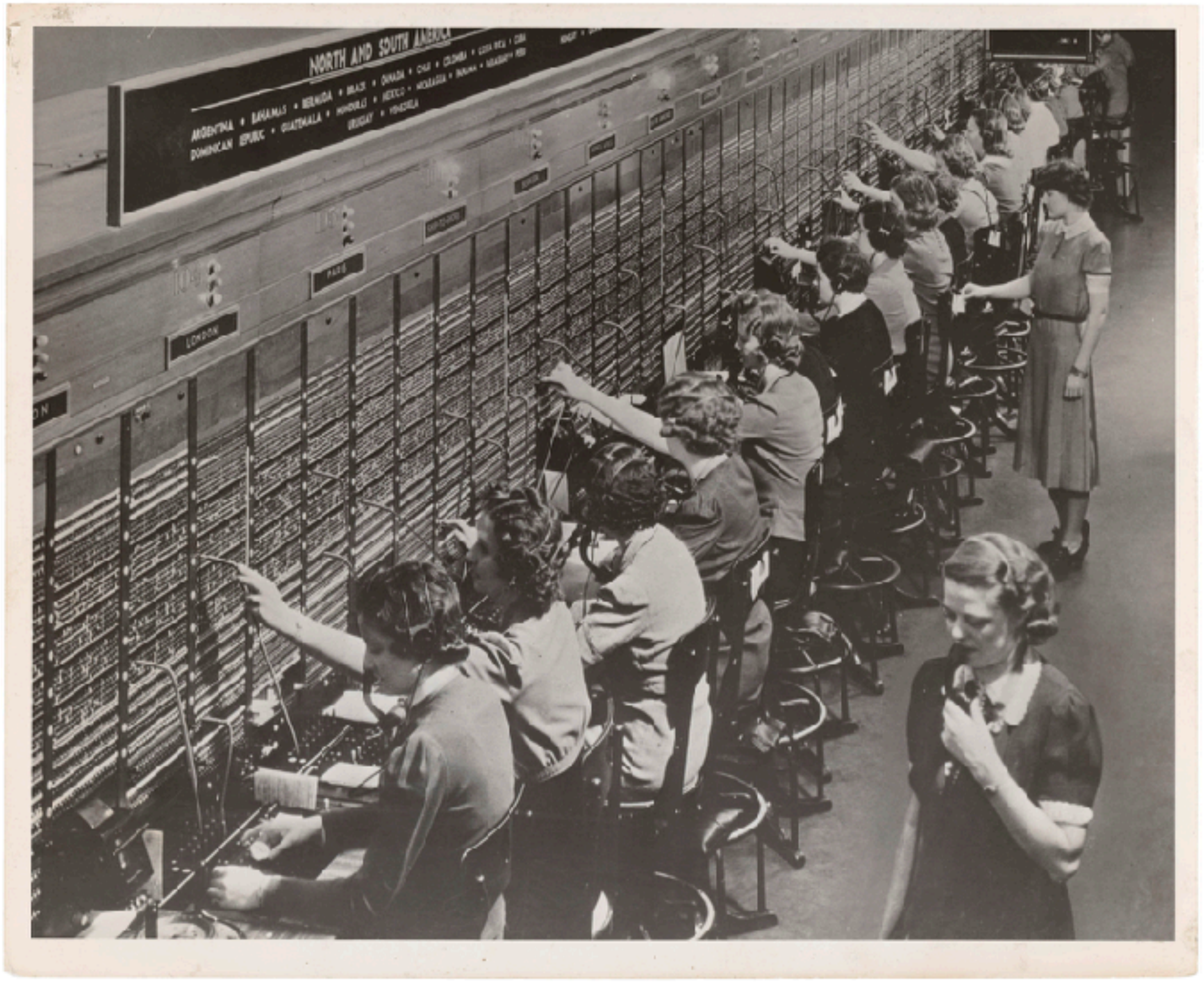
S-S learning



Tolman (1948)

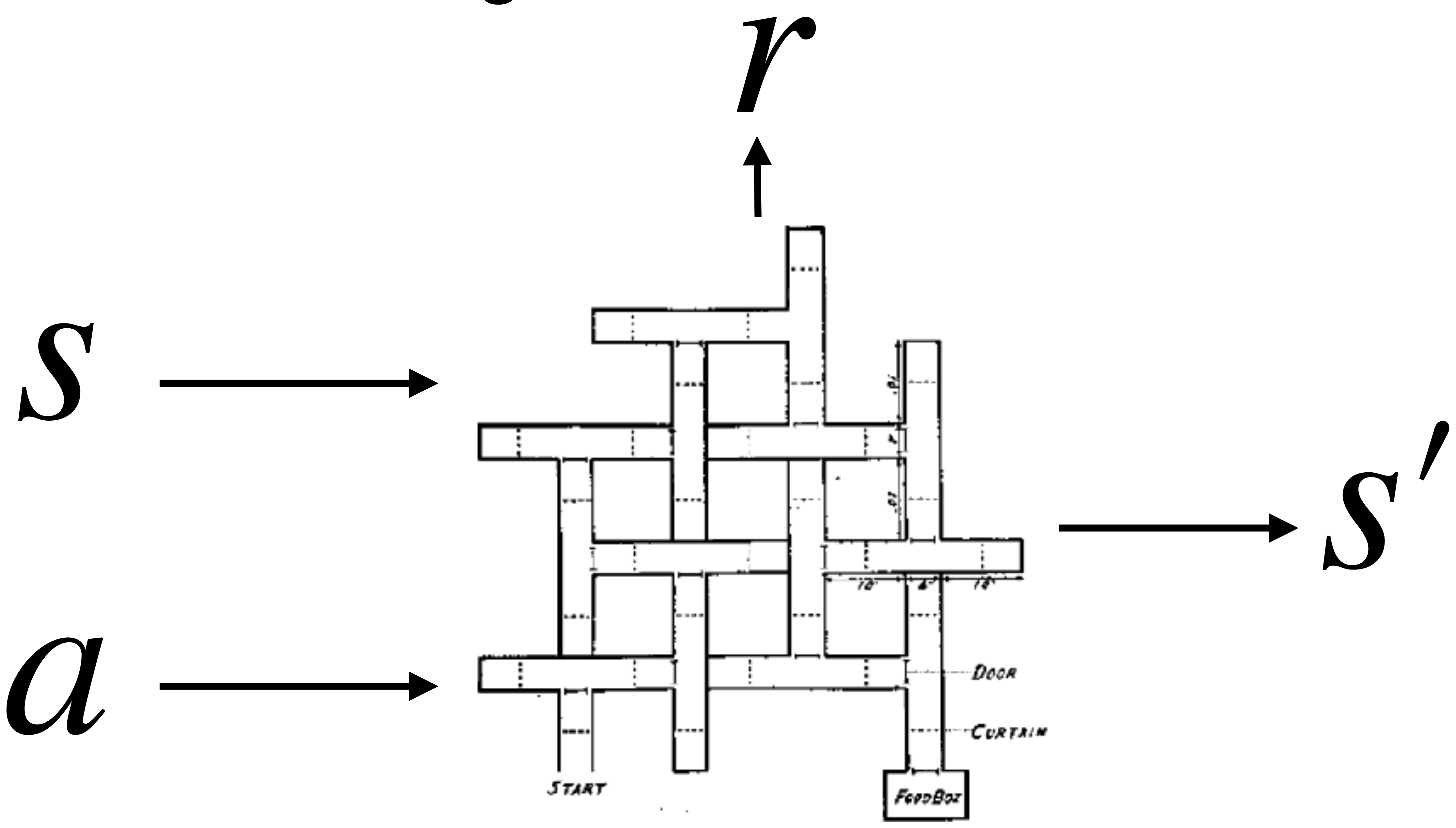
Model-free

S-R learning



Model-based

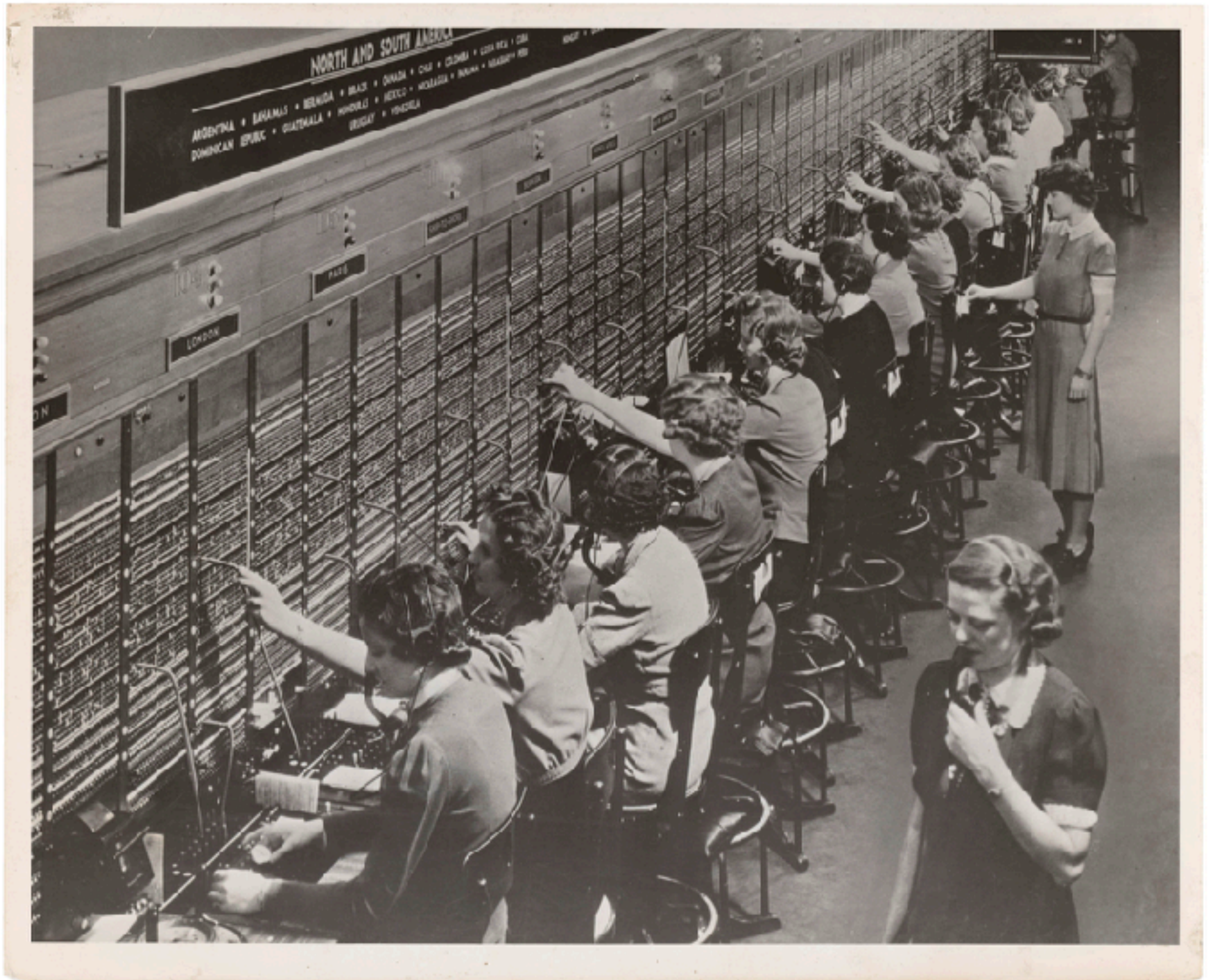
S-S learning



Tolman (1948)

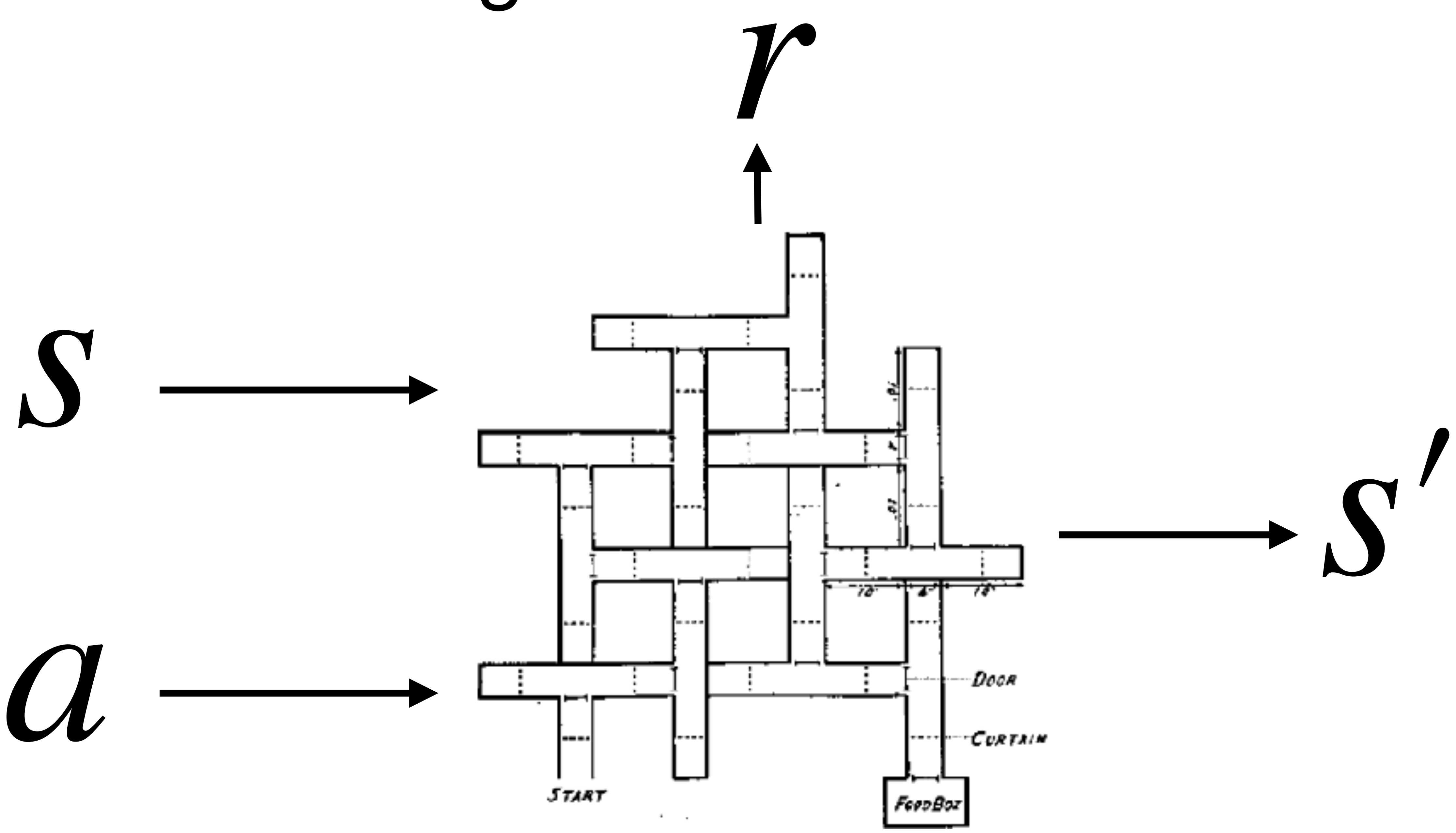
Model-free

S-R learning



Model-based

S-S learning



Tolman (1948)

image credit Alyssa Dayan
(from Dolan & Dayan, 2013)

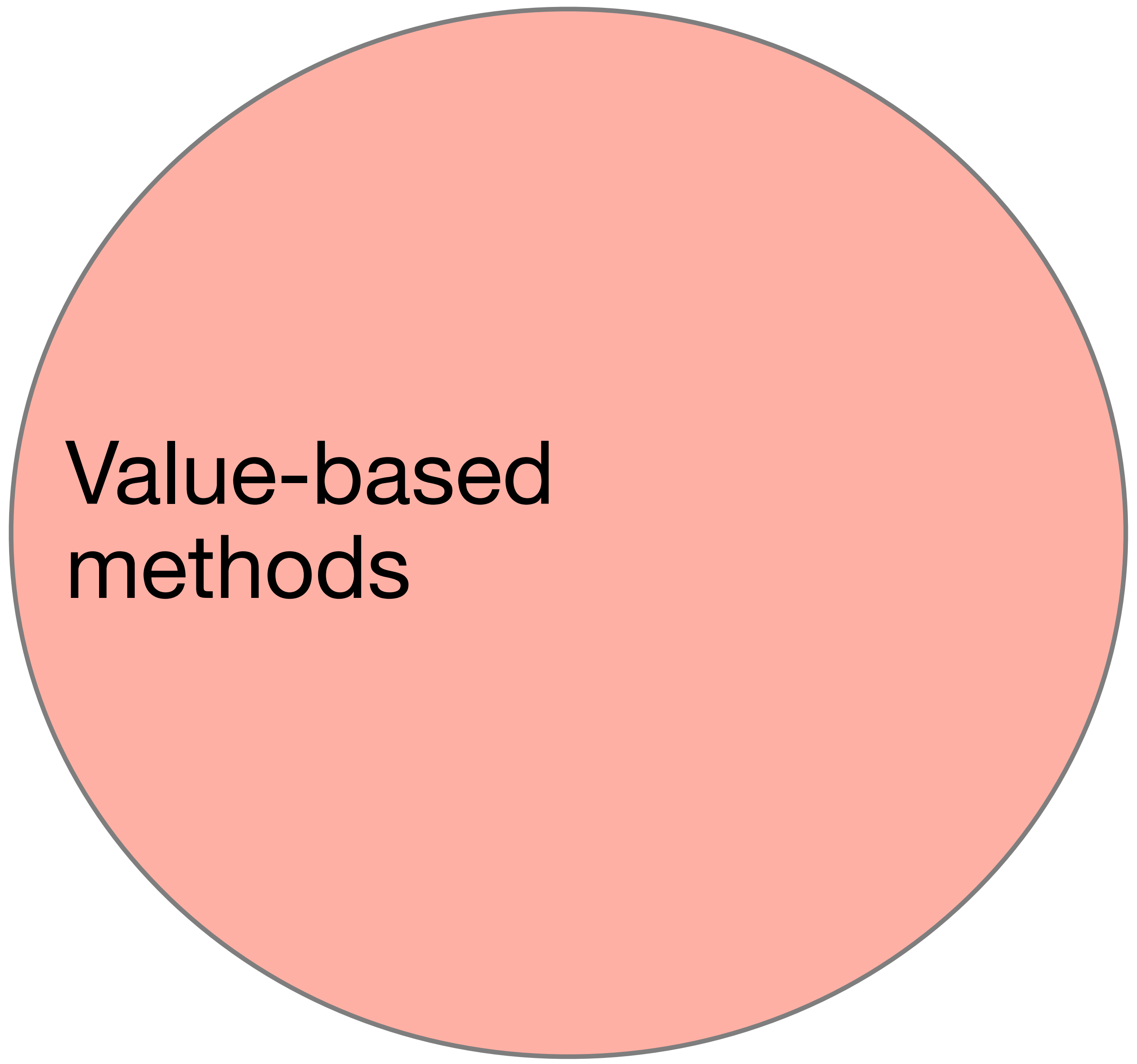
Advances in RL

Advances in RL

- Modern model-free methods can be categorized as **Value-based**, **Policy-based**, or **Actor**-**Critic**

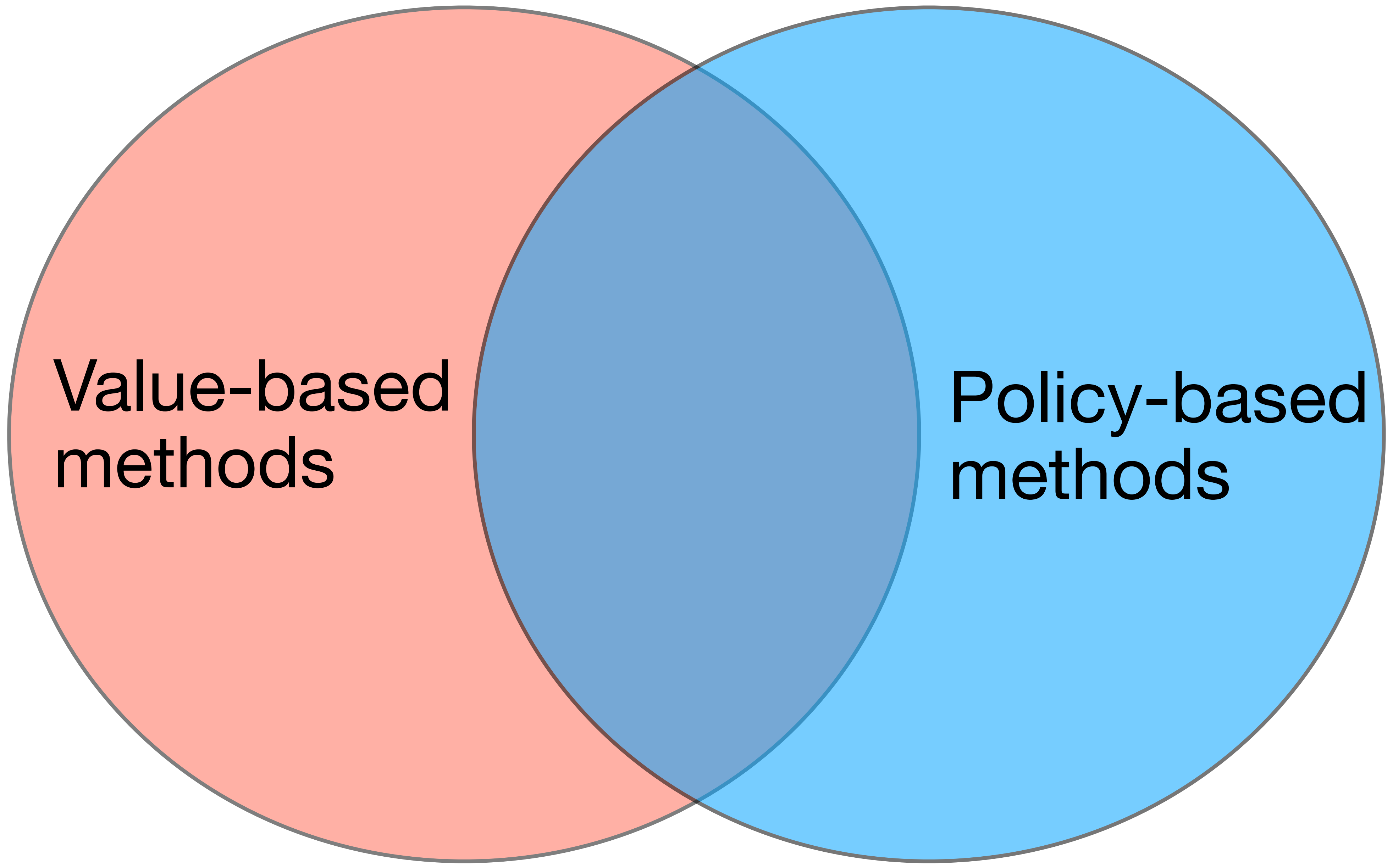
Advances in RL

- Modern model-free methods can be categorized as **Value-based**, **Policy-based**, or **Actor**-**Critic**



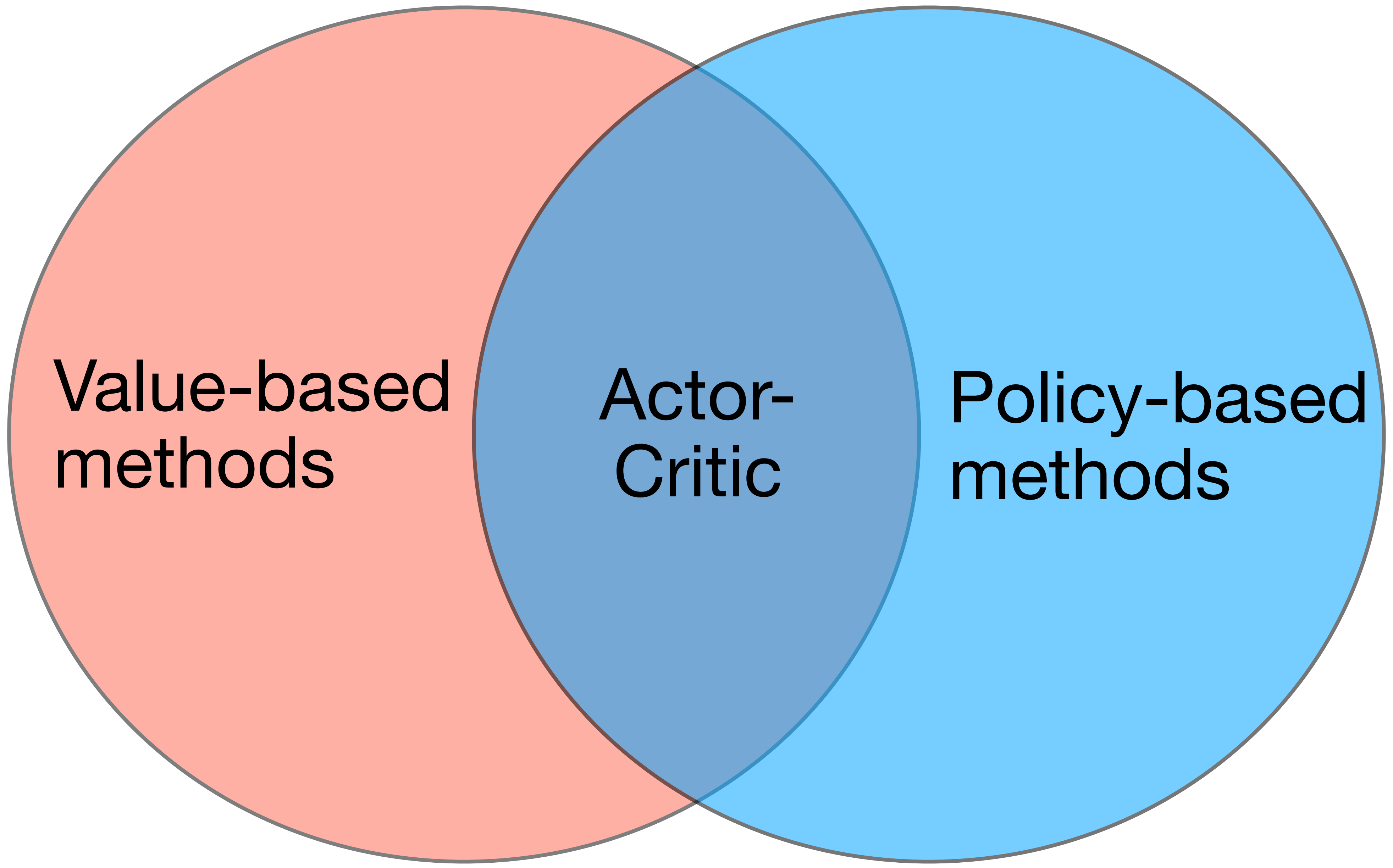
Advances in RL

- Modern model-free methods can be categorized as **Value-based**, **Policy-based**, or **Actor**-**Critic**



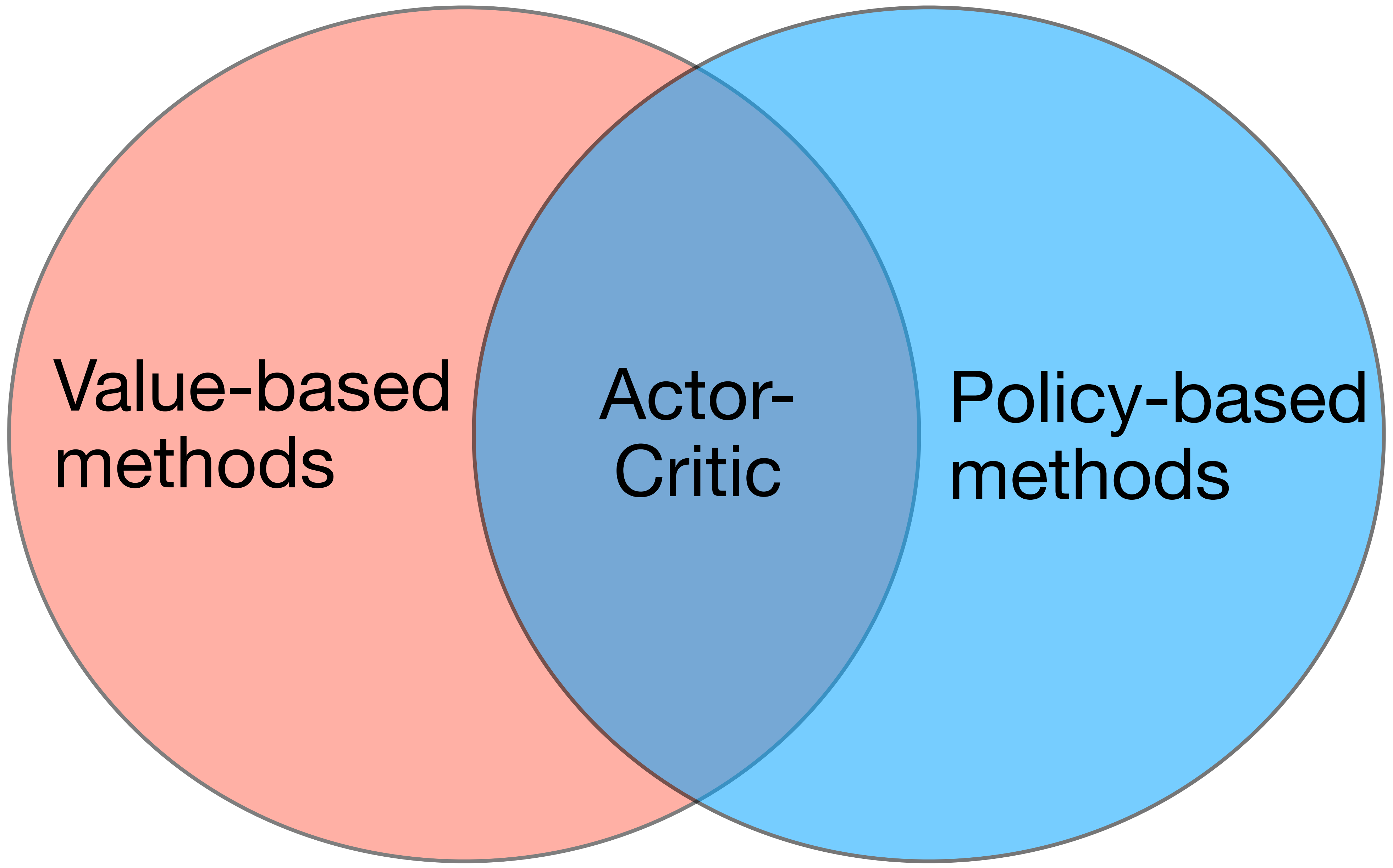
Advances in RL

- Modern model-free methods can be categorized as **Value-based**, **Policy-based**, or **Actor-Critic**



Advances in RL

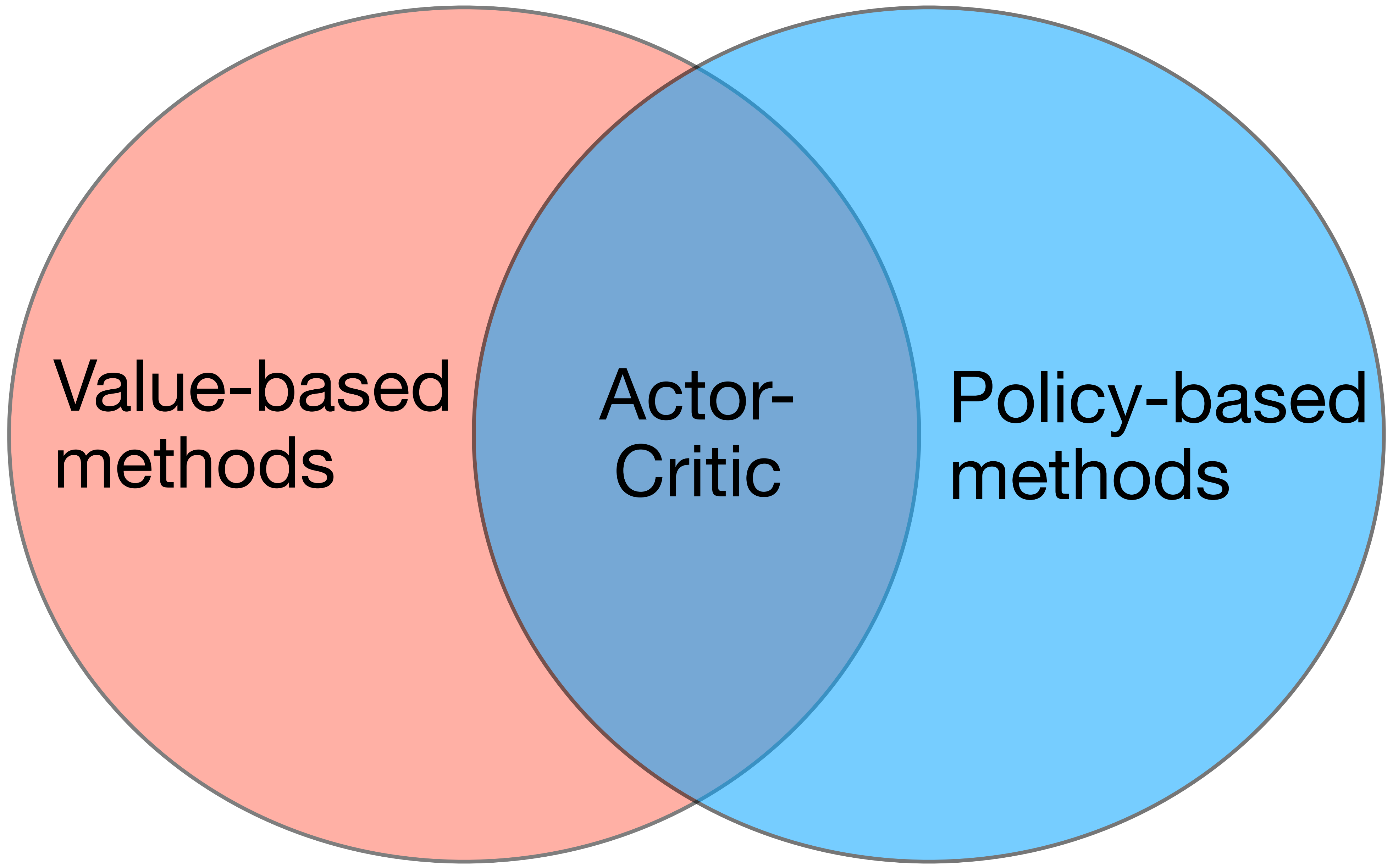
- Modern model-free methods can be categorized as **Value-based**, **Policy-based**, or **Actor**-**Critic**
  Deep Q-learning



Advances in RL

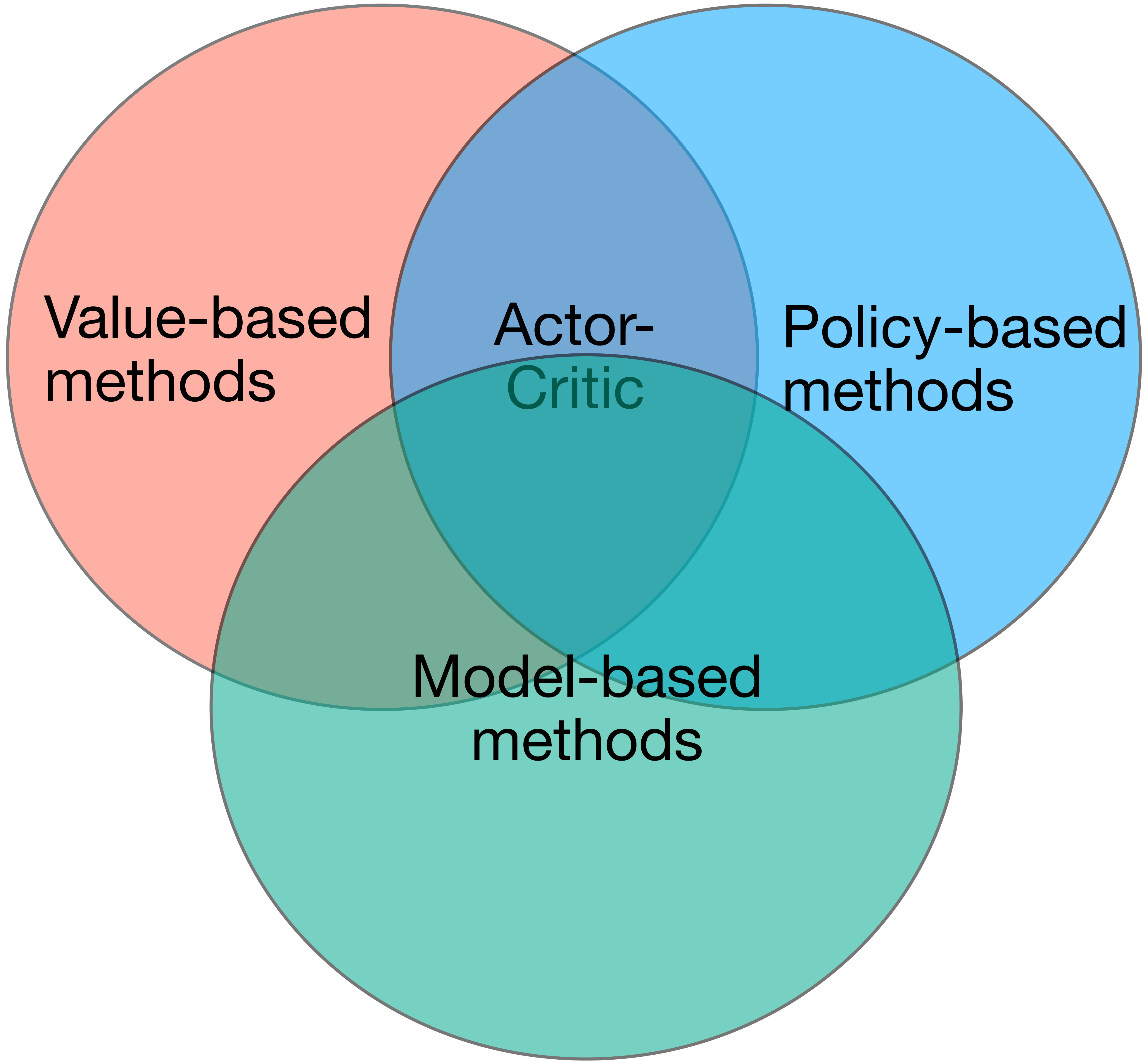
- Modern model-free methods can be categorized as **Value-based**, **Policy-based**, or **Actor**-**Critic**

  Deep Q-learning Policy gradient



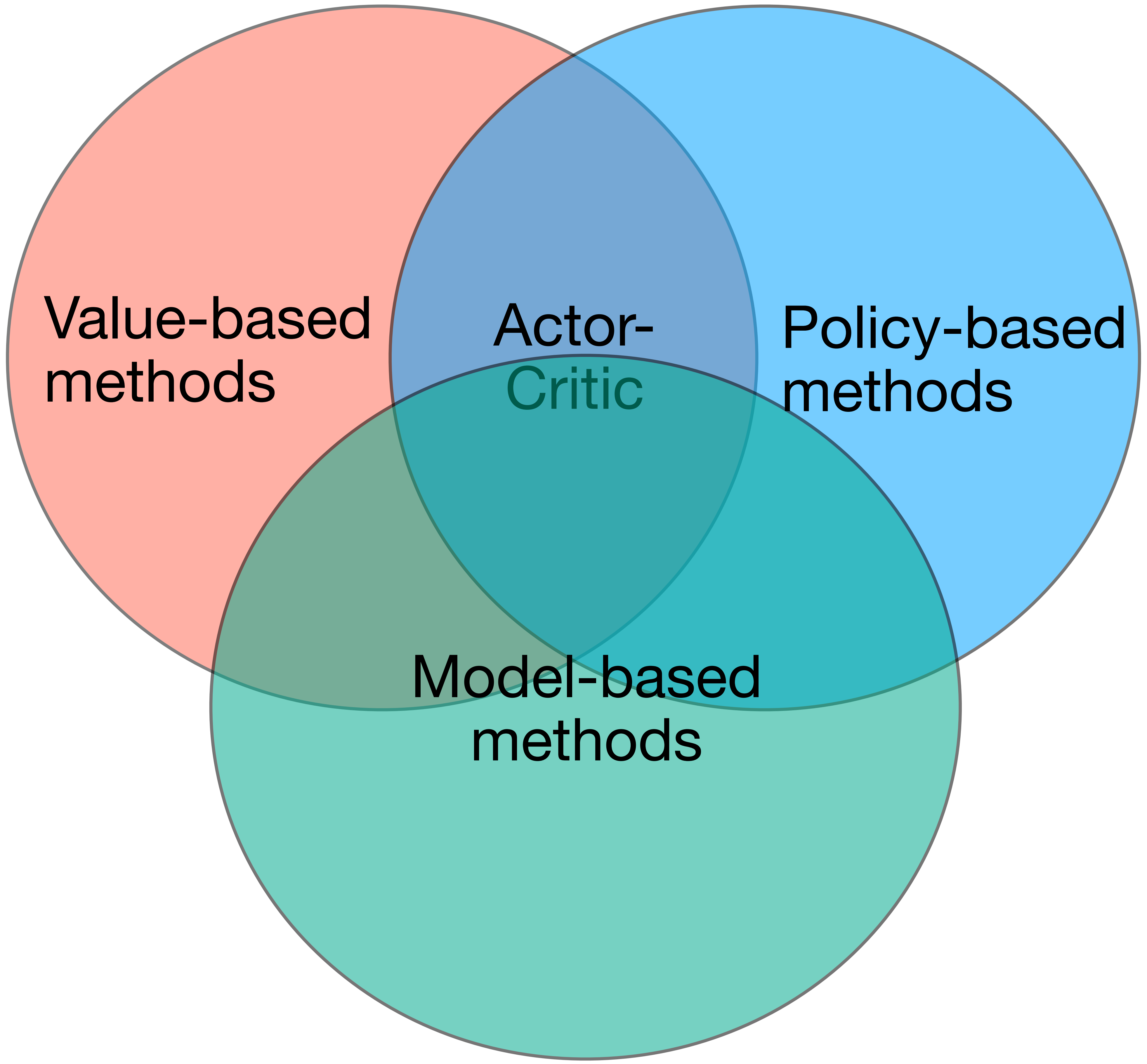
Advances in RL

- Modern model-free methods can be categorized as **Value-based**, **Policy-based**, or **Actor-Critic**
  Deep Q-learning Policy gradient
- Model-based methods can as well…



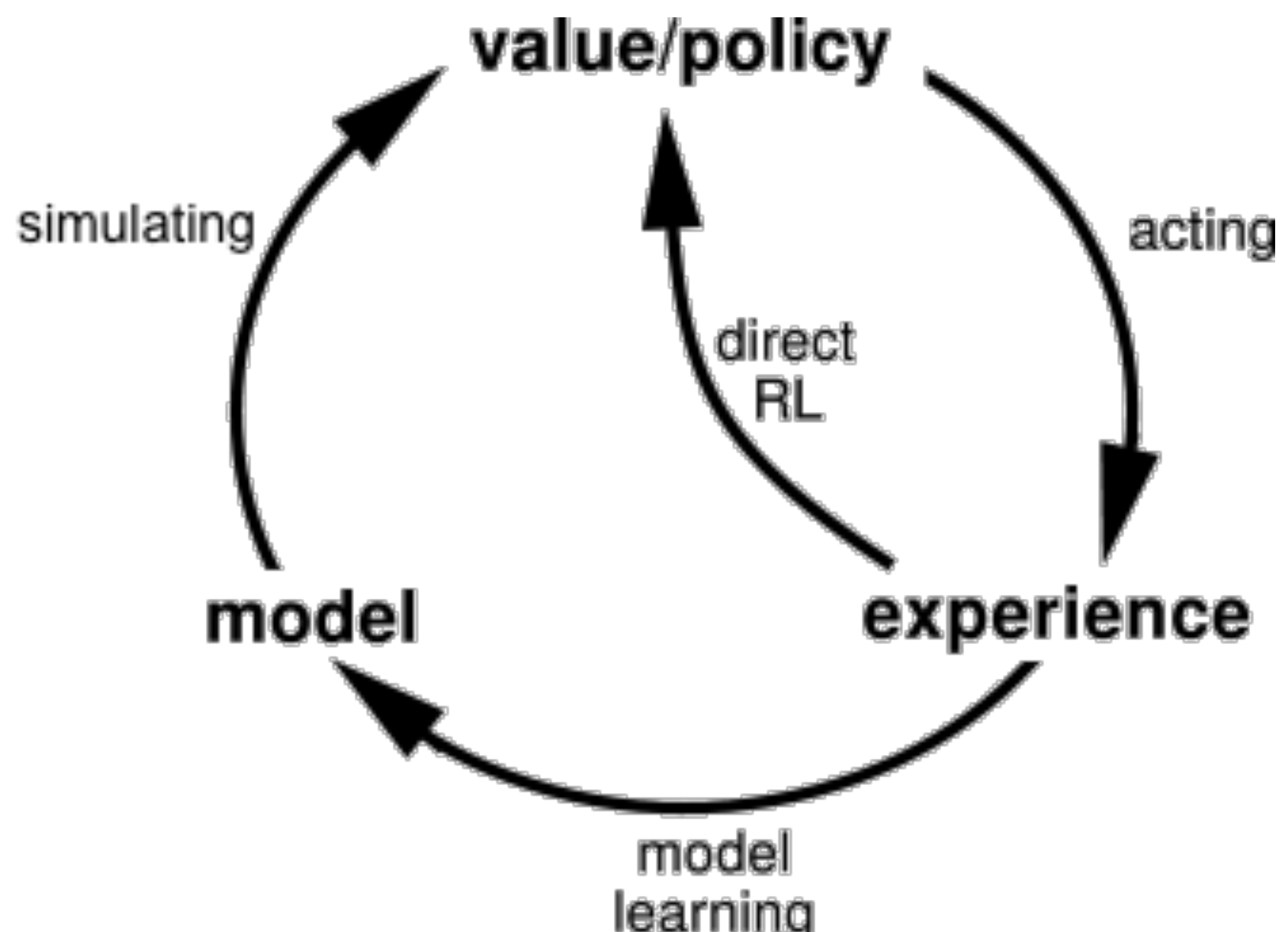
Advances in RL

- Modern model-free methods can be categorized as **Value-based**, **Policy-based**, or **Actor-Critic**
  Deep Q-learning Policy gradient
- Model-based methods can as well…



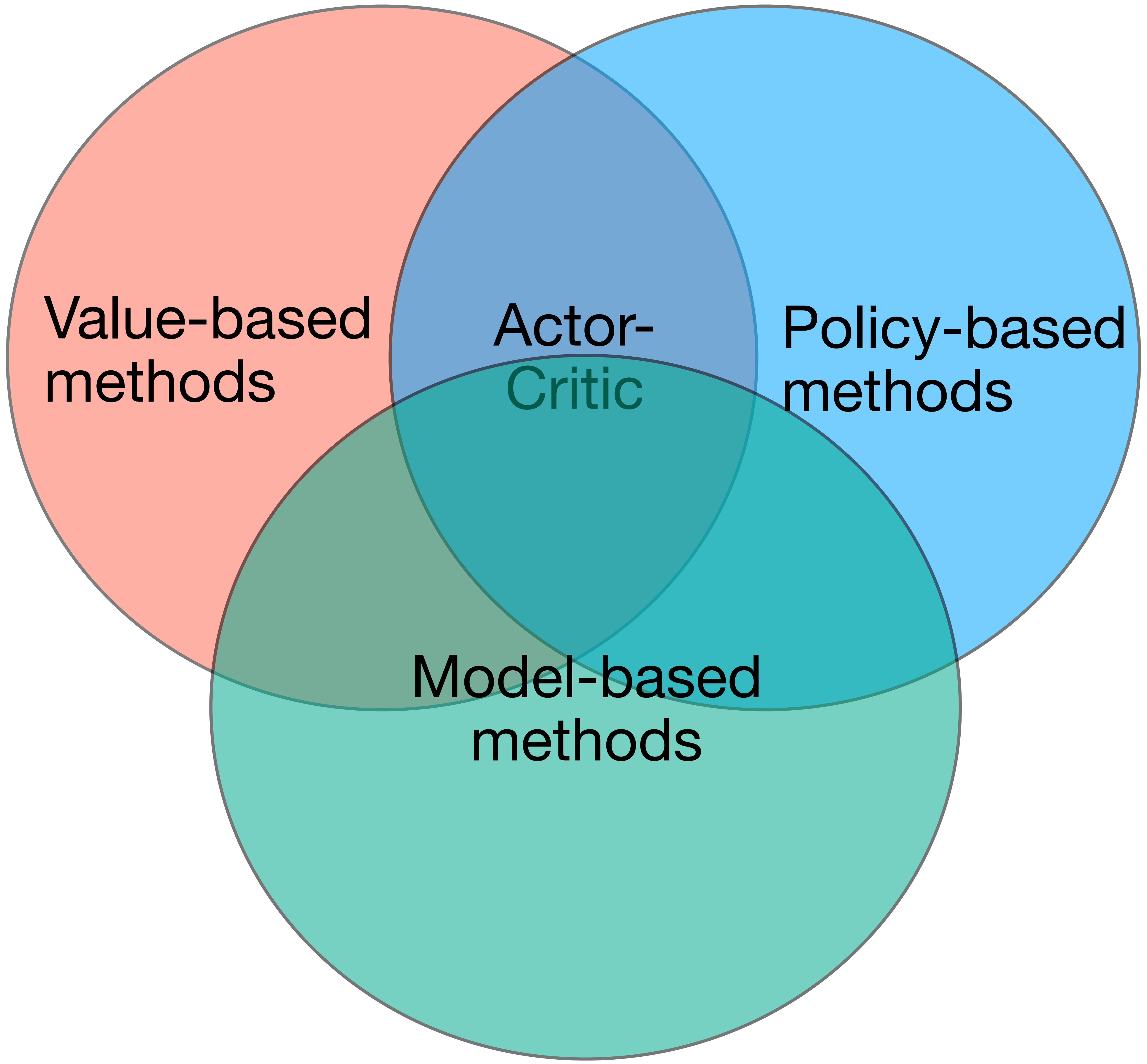
The model can be used to **simulate experiences** for updating the value/policy

These simulations are **computationally costly**, but supplement direct RL, leading to **faster learning** and **greater flexibility**



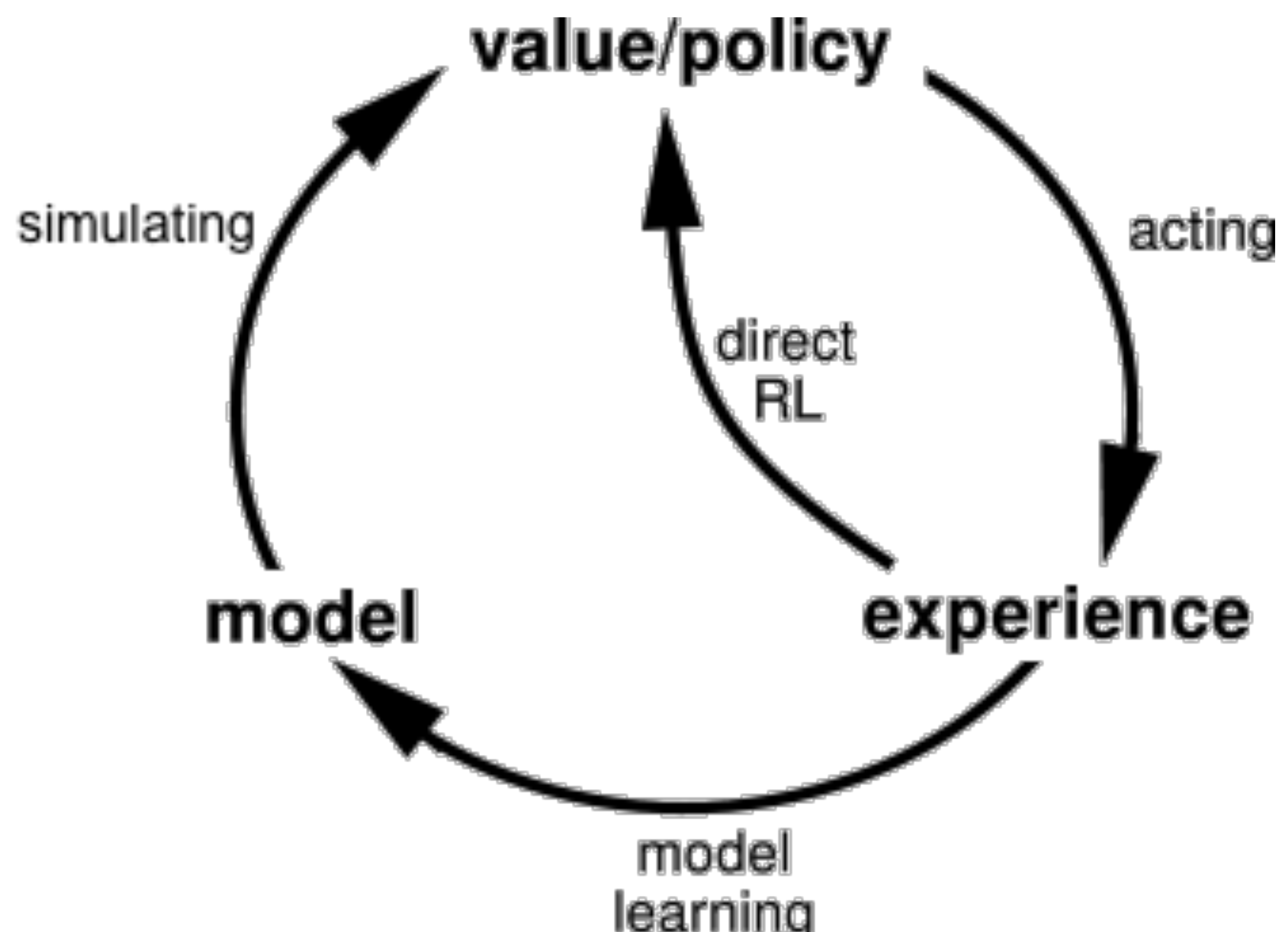
Advances in RL

- Modern model-free methods can be categorized as **Value-based**, **Policy-based**, or **Actor-Critic**
  Deep Q-learning Policy gradient
- Model-based methods can as well…
  - DYNA (**Model** & **Value-based**)



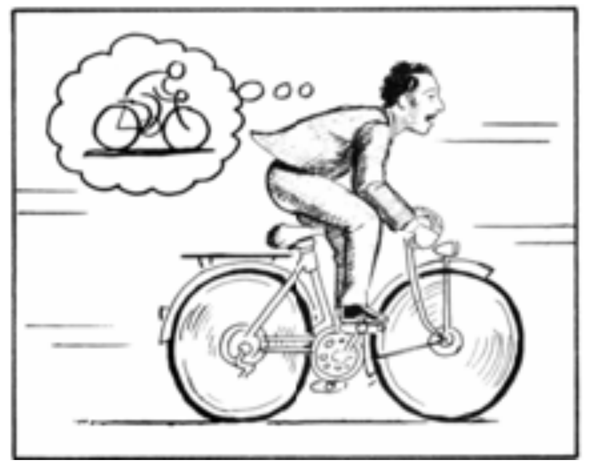
The model can be used to **simulate experiences** for updating the value/policy

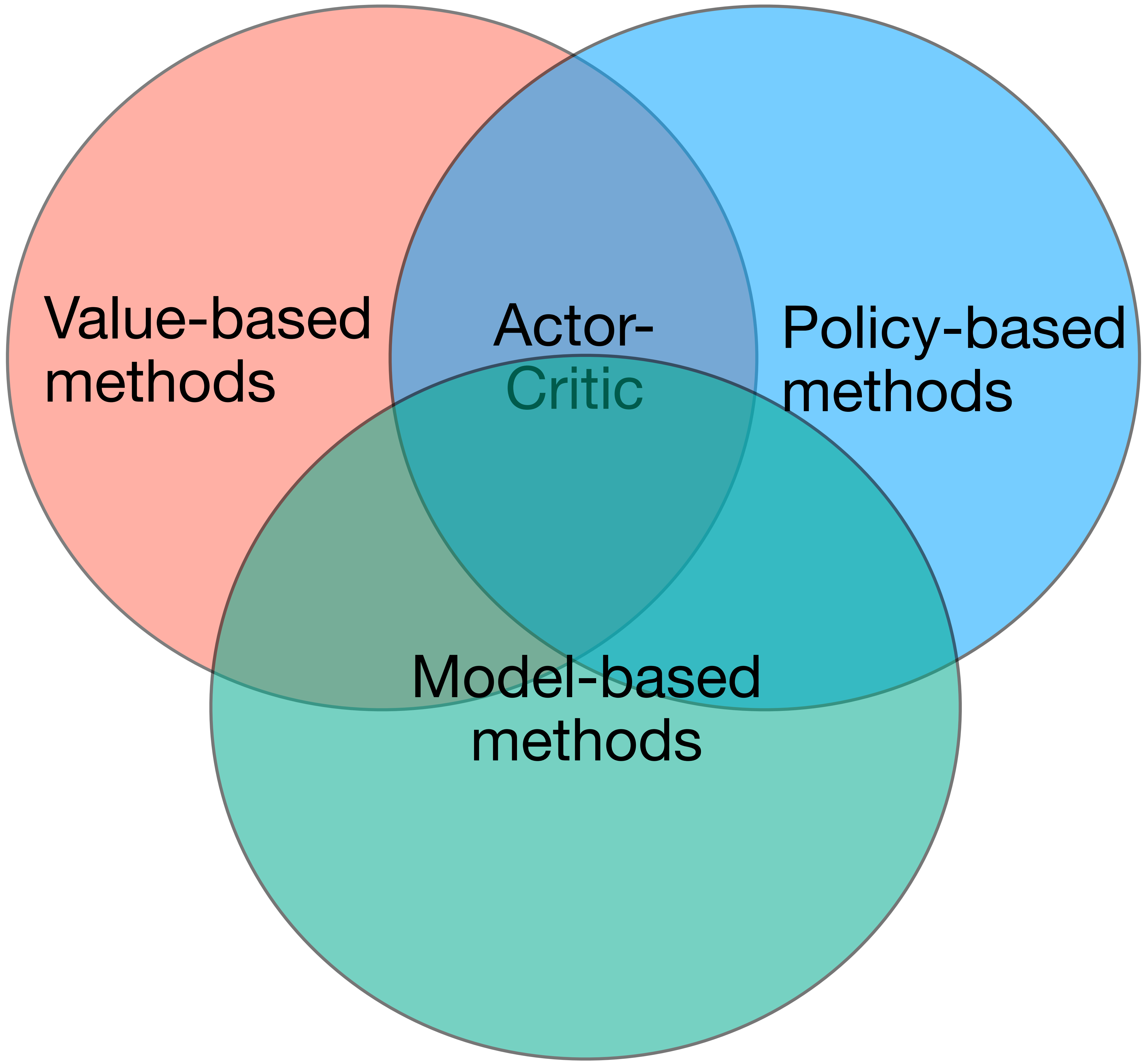
These simulations are **computationally costly**, but supplement direct RL, leading to **faster learning** and **greater flexibility**



Advances in RL

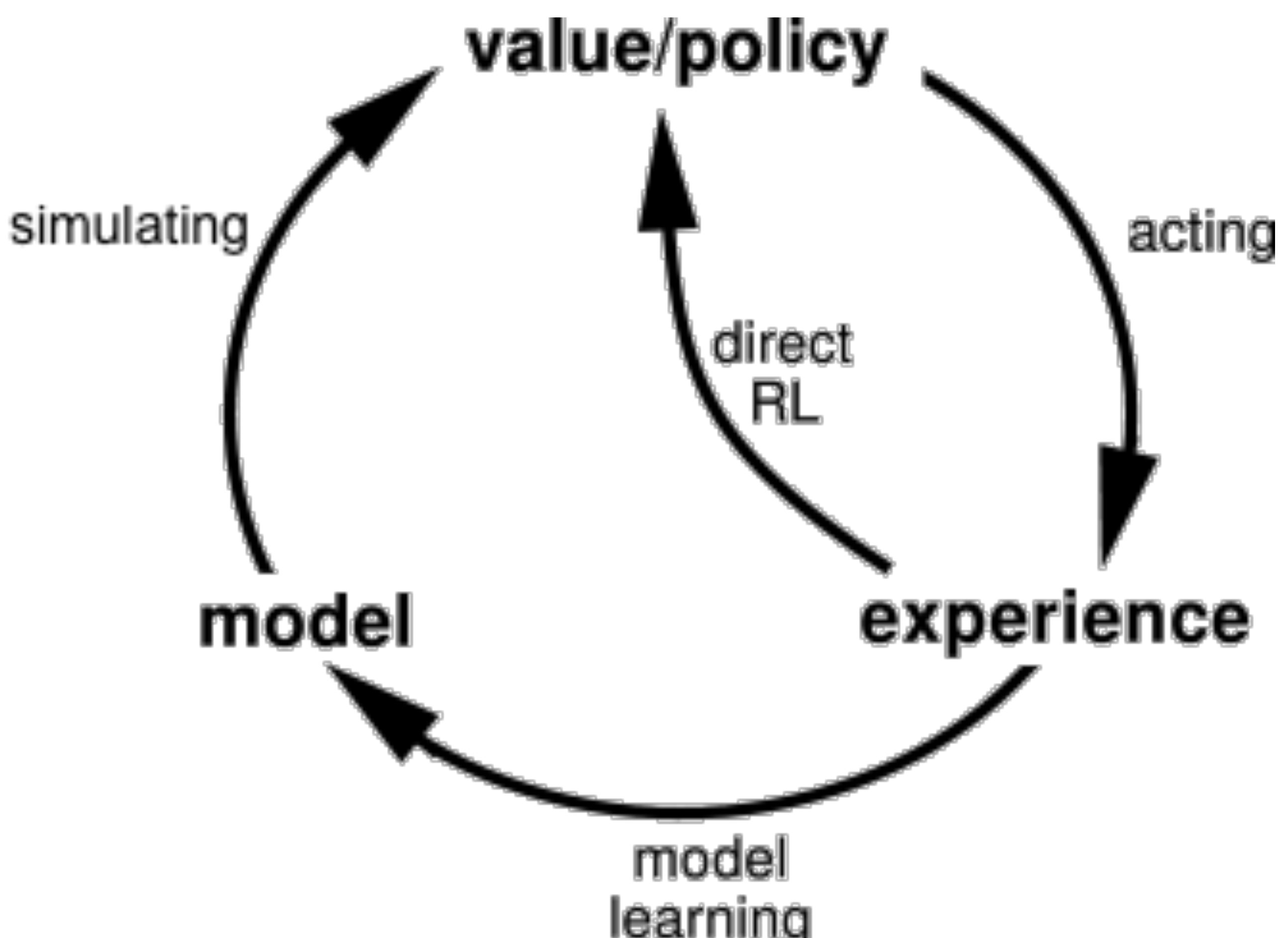
- Modern model-free methods can be categorized as **Value-based**, **Policy-based**, or **Actor-Critic**
  Deep Q-learning Policy gradient
- Model-based methods can as well…
  - DYNA (**Model** & **Value-based**)
  - World Models (**Model** & **Policy-based**)





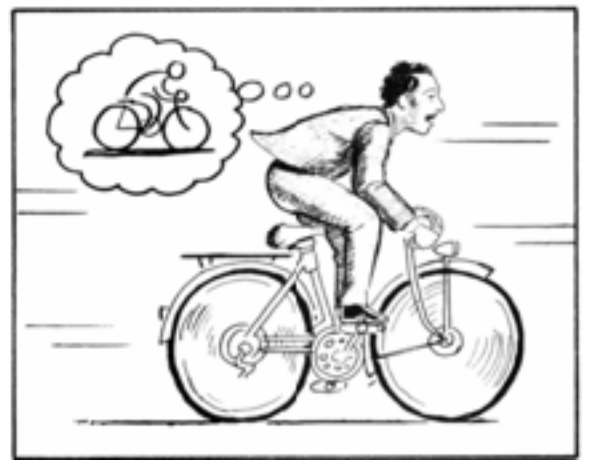
The model can be used to **simulate experiences** for updating the value/policy

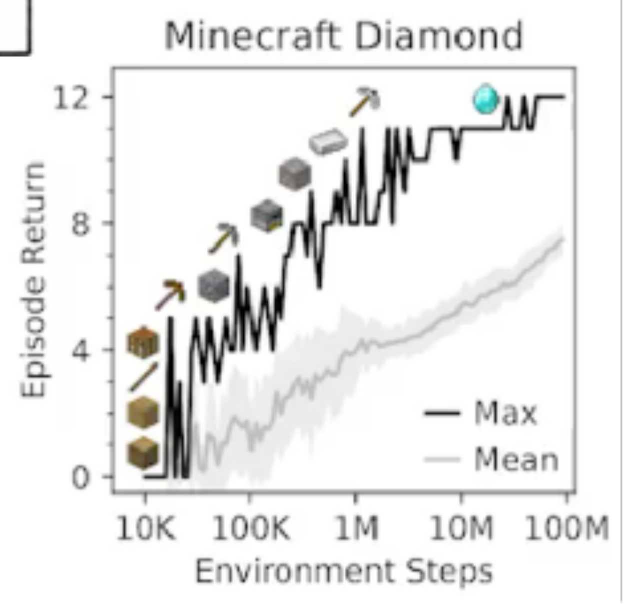
These simulations are **computationally costly**, but supplement direct RL, leading to **faster learning** and **greater flexibility**

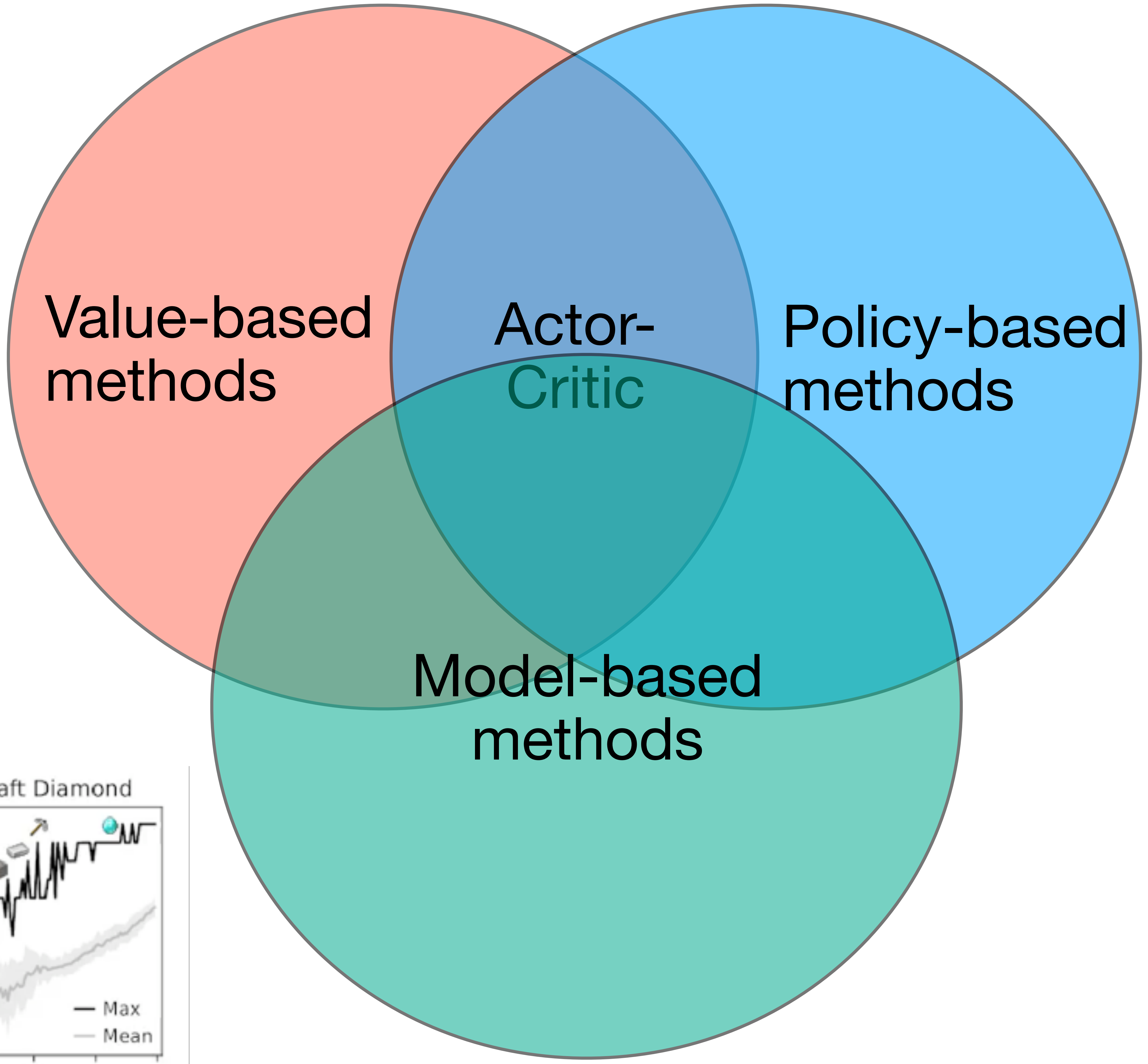


Advances in RL

- Modern model-free methods can be categorized as **Value-based**, **Policy-based**, or **Actor-Critic**
  Deep Q-learning Policy gradient
- Model-based methods can as well…
  - DYNA (**Model** & **Value-based**)
  - World Models (**Model** & **Policy-based**)
  - Dreamer (**Model** & **Actor-Critic**)

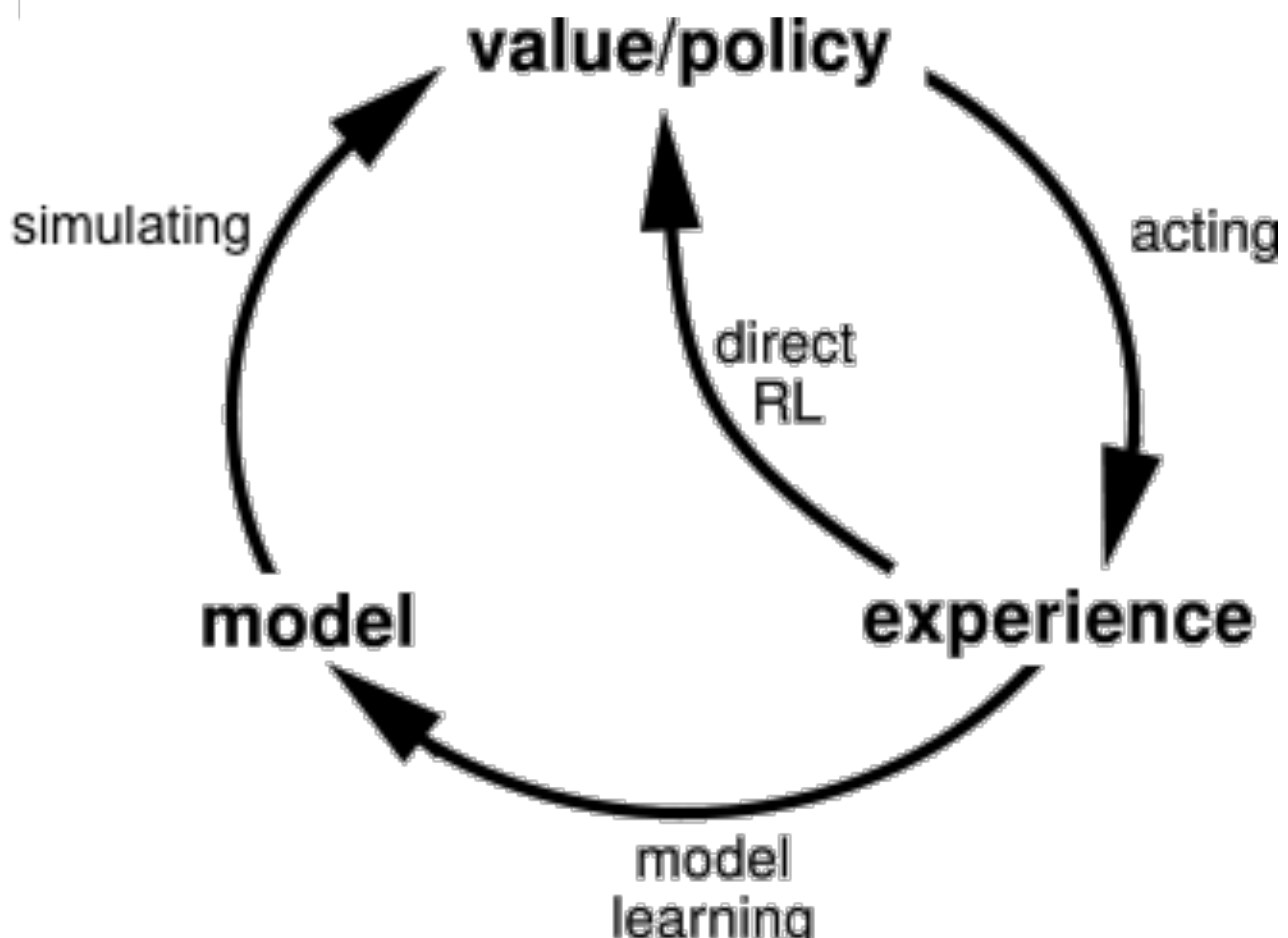






The model can be used to **simulate experiences** for updating the value/policy

These simulations are **computationally costly**, but supplement direct RL, leading to **faster learning** and **greater flexibility**



5 minute break

Social learning

Alex Witt

Learning is not only from environmental feedback, but also from social sources

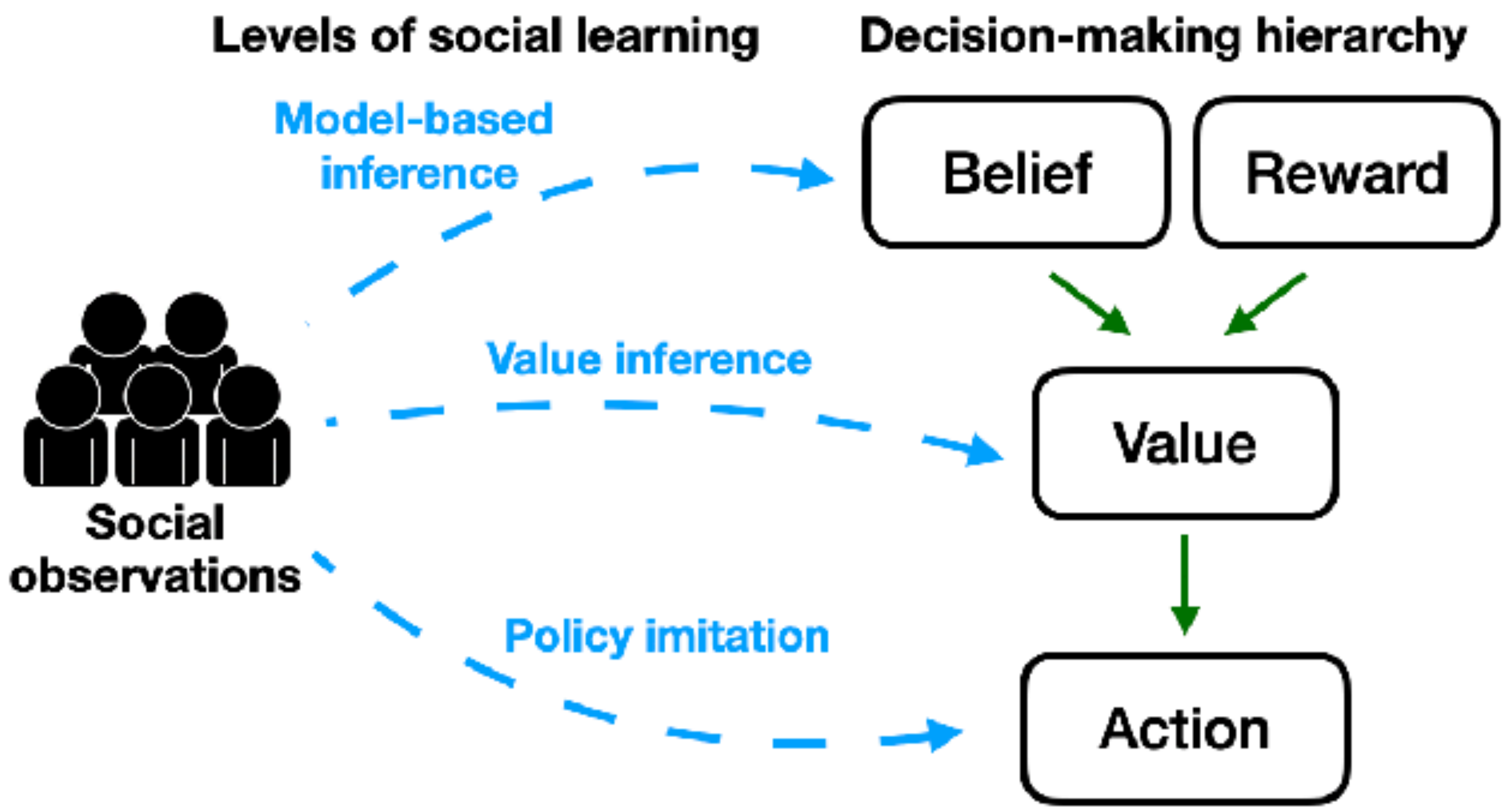
Imitation via observational learning, where **social learning strategies (SLS)** define various who, what, when

Theory of mind (ToM) involves inferring hidden mental states from observable behavior

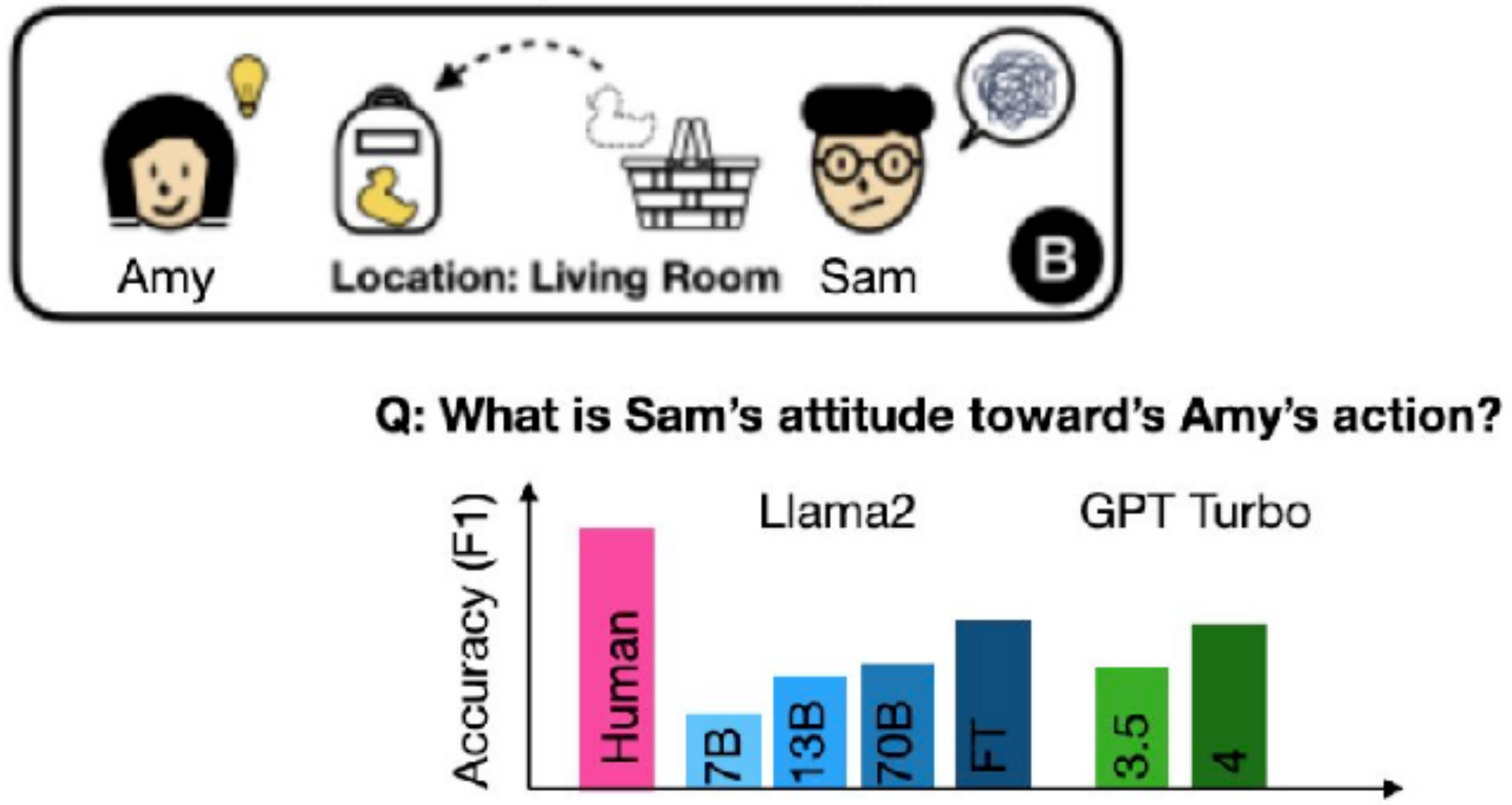
Various Bayesian formalisms of ToM, but typically intractable and a key limitation of current AI

Bandura (1961)

Wu, Vélez, & Cushman (2022)



OpenToM Benchmark (Xu et al., 2024)



Compression

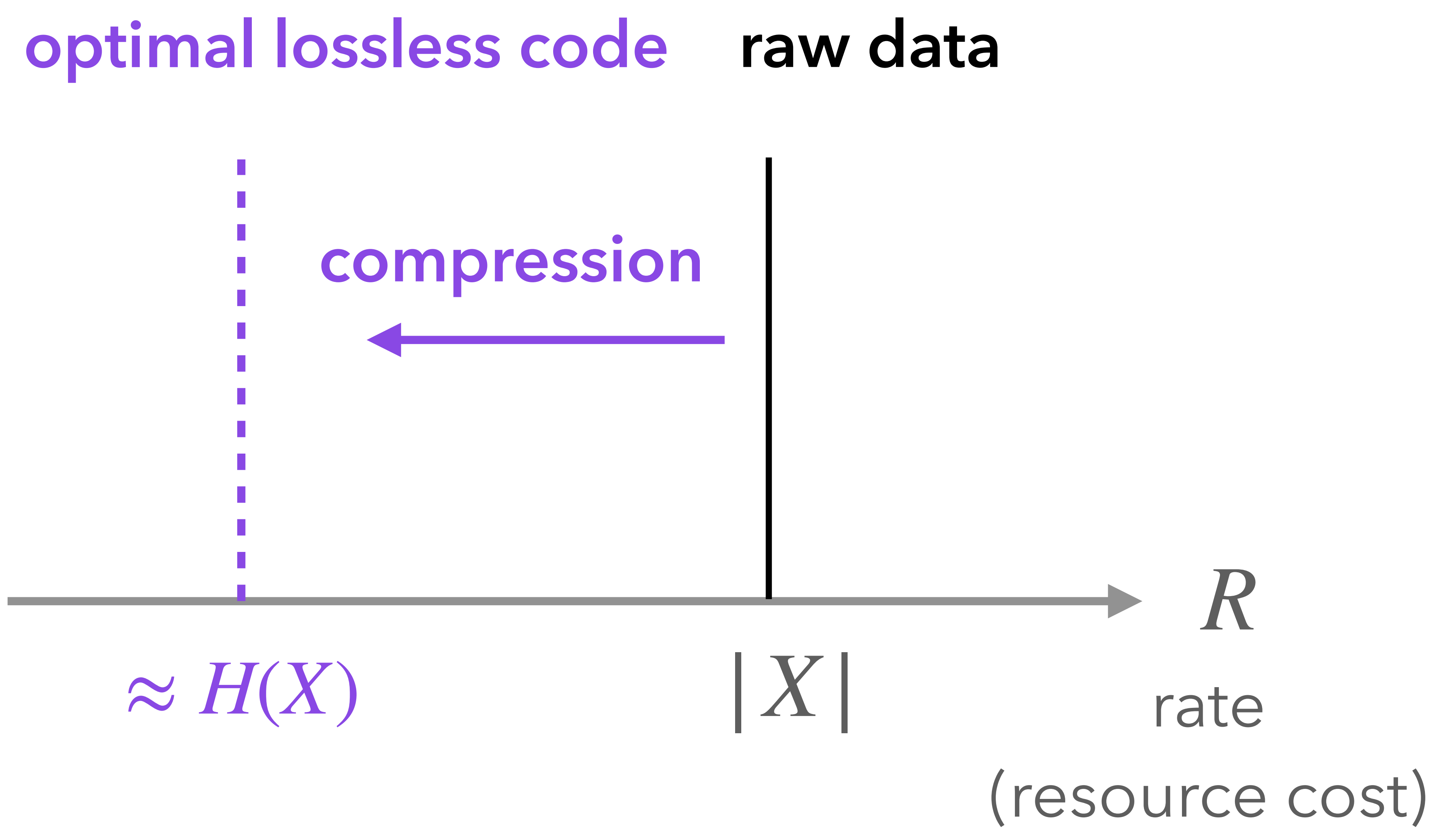
David Nagy

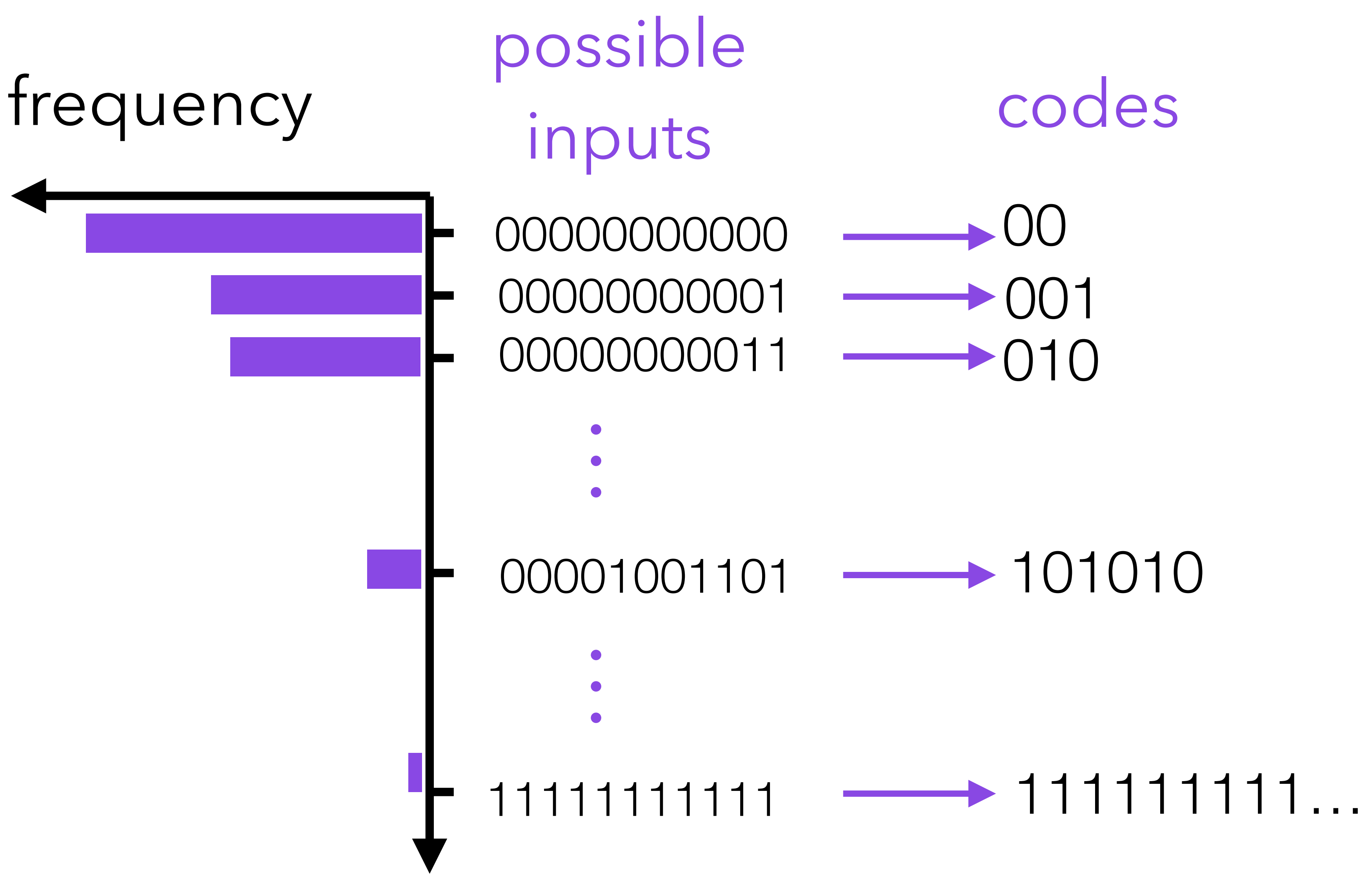
Compression decreases the resources R required to store data

Lossless compression is without loss of information

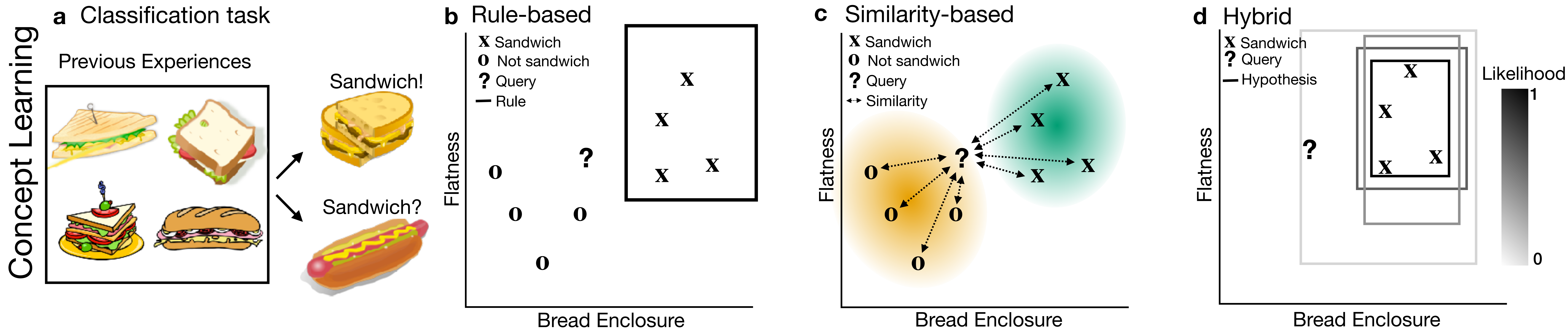
The **optimal lossless code** is based on assigning the shortest codes to the most frequent inputs: *source coding theorem*

Even greater compression is possible by allowing for distortions: **lossy compression**





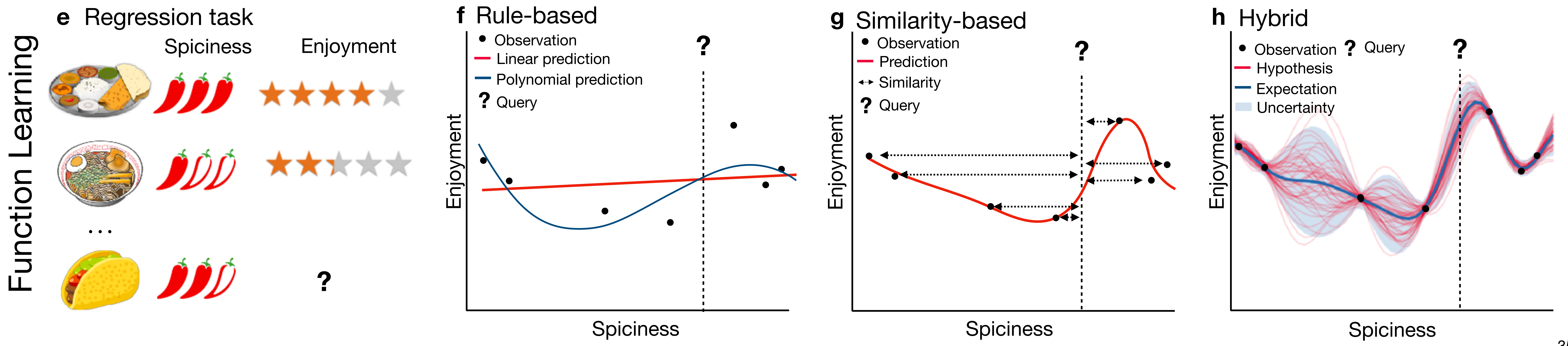
Learning concepts



- Concepts are mental representations of categories in the world (classification problem)
- Classical view used **rules** to describe the necessary and sufficient conditions for category membership
- More psychological approaches used **similarity**, compared to a learned *prototypes* or past *exemplars*
- Bayesian concept learning is a **hybrid** approach, that uses distributions over rules, and recreating patterns consistent with similarity-based approaches

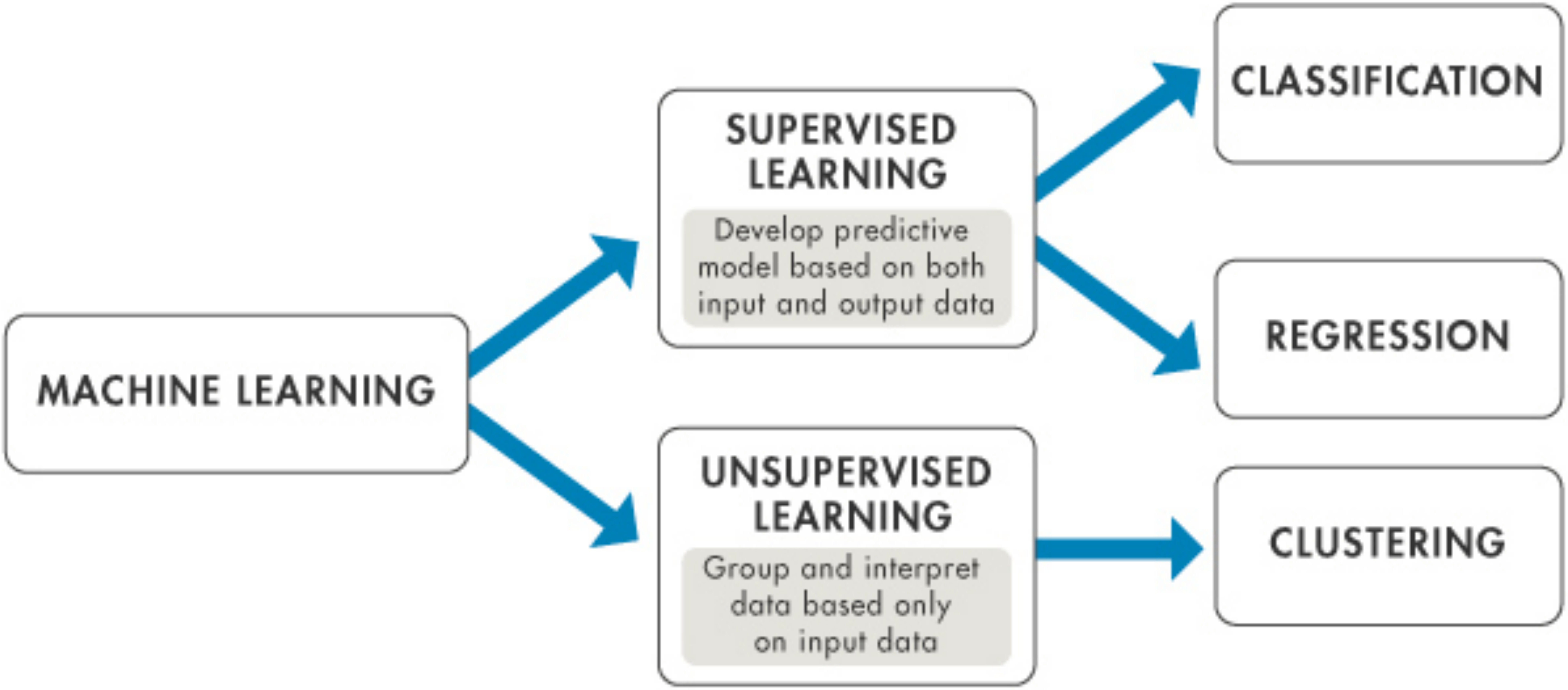
Learning functions

- Functions are mental representations of relationships in the world (regression problem)
- Early **rule**-based theories assumed humans learn functions by picking specific class of functions and then optimizing the weights (as in linear or parametric regression)
- **Similarity**-based methods used ANNs to encode the generic principle that similar inputs produce similar outputs
- **Hybrid** approaches using GP regression offer a Bayesian framework, combining kernel similarity and rule-like compositionality of kernels



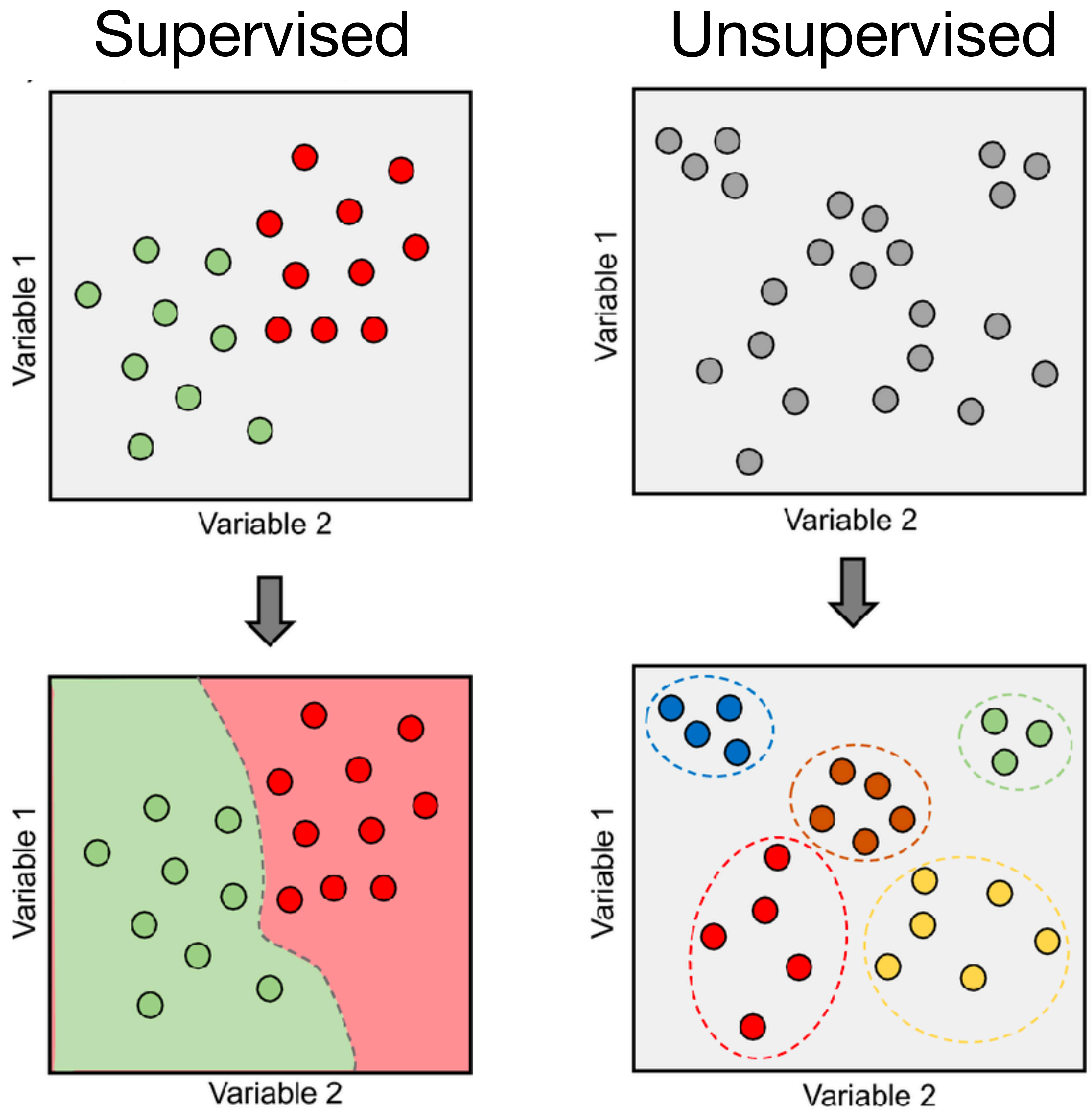
Converging theories?

Modern Machine Learning



Supervised vs. unsupervised learning

- Classification problems*: classify data points into one of *n* different categories
- **Supervised** learning:
  - Training data provides category labels
  - Classifiers usually try to learn a decision-boundary
- **Unsupervised** learning:
  - Training data lacks category labels
  - Classifiers usually try to learn clusters



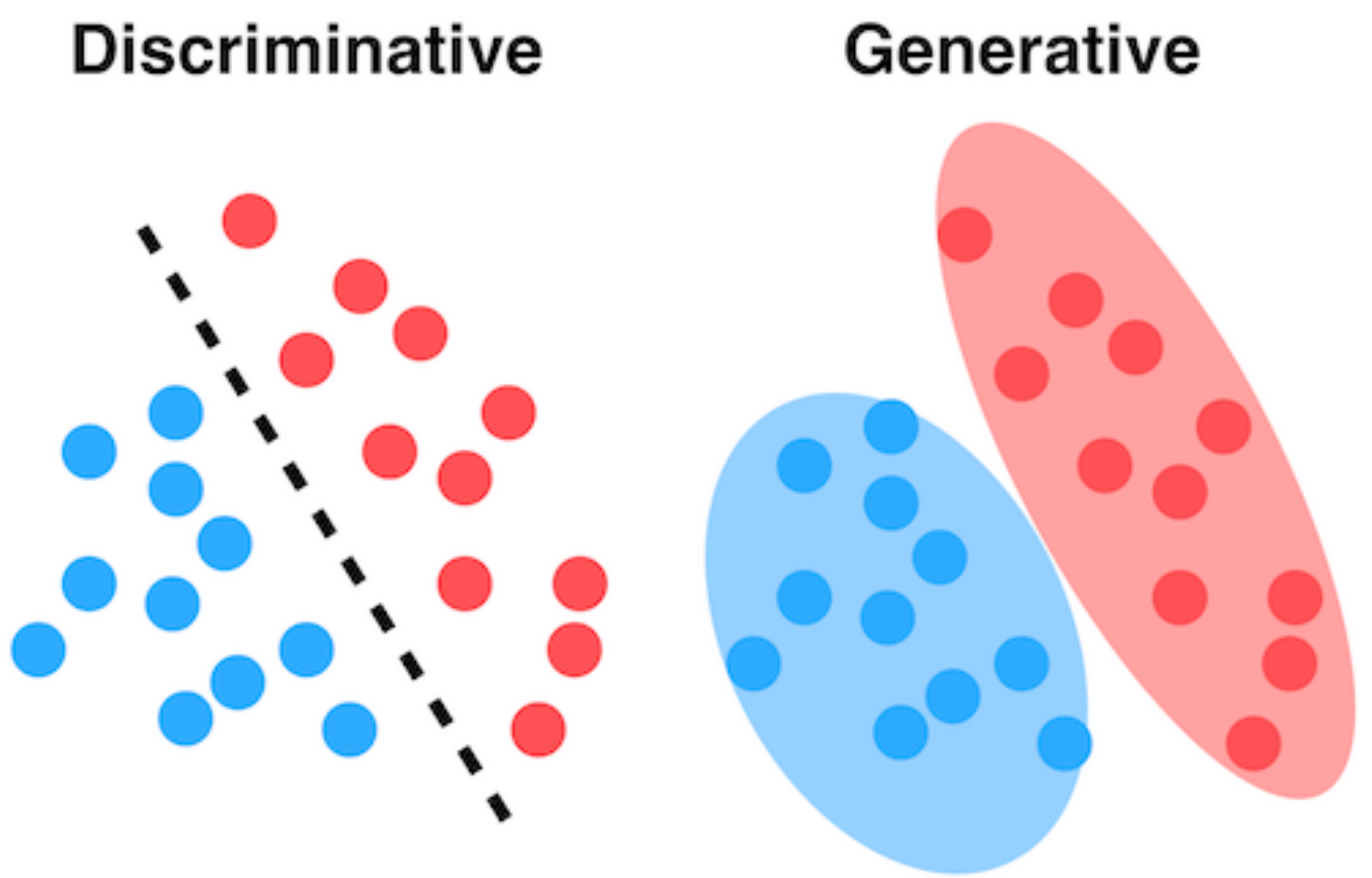
Notation:

a scalar $\quad$ $\mathbf{a}$ vector $\quad$ $\mathscr{A}$ set

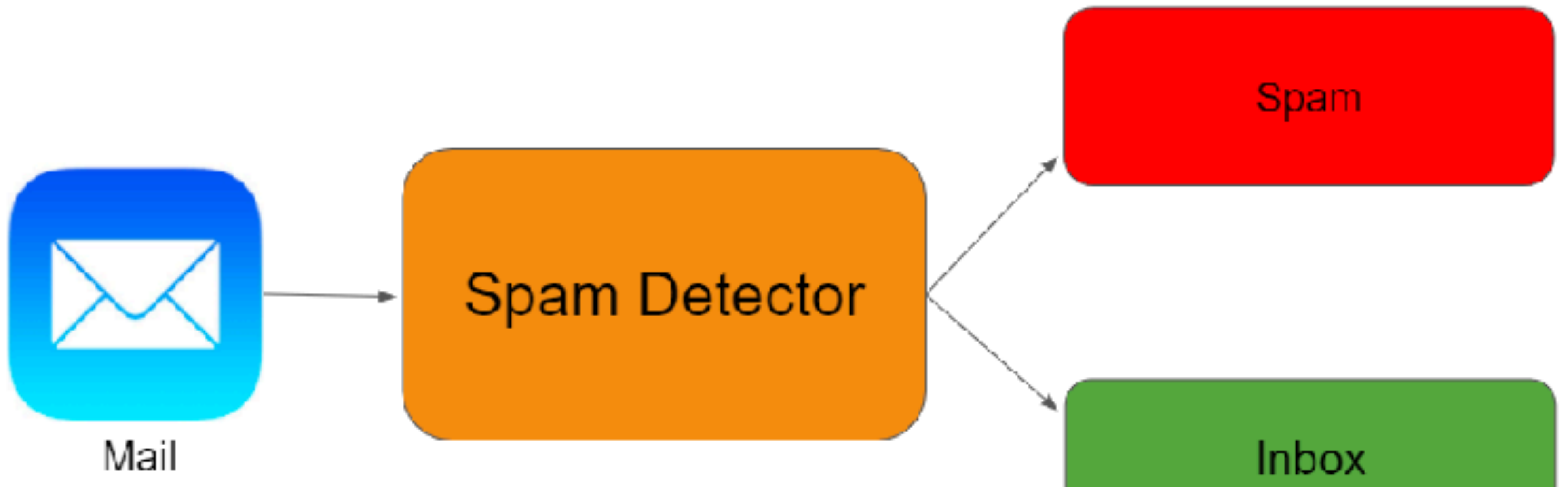
A constant $\quad$ $\mathbf{A}$ Matrix

Supervised learning

- Two general classes:
  - **Discriminitive** directly map features to class labels, often by learning a decision-boundary (rule-like)
  - **Generative** approaches learn the probability distribution of the data (similarity-like)



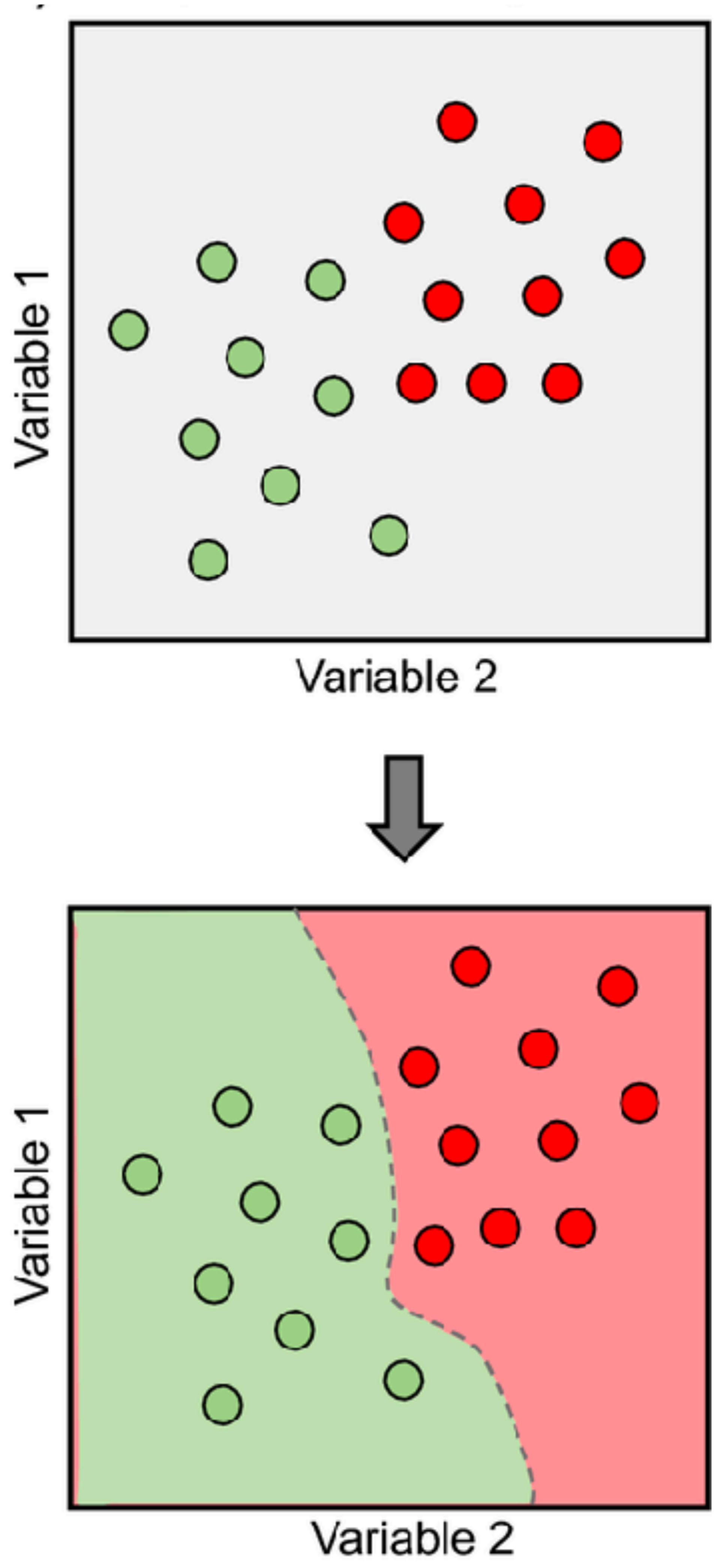
- Example problem: Spam detector
  - Data $\mathscr{D} = \{\mathbf{X}, \mathbf{y}\}$
  - each $\mathbf{x} \in \mathbf{X}$ are the features of an email (e.g., length, date, sender, content, etc…)
  - each $y \in \mathbf{y}$ is the label (1 if spam, 0 otherwise)



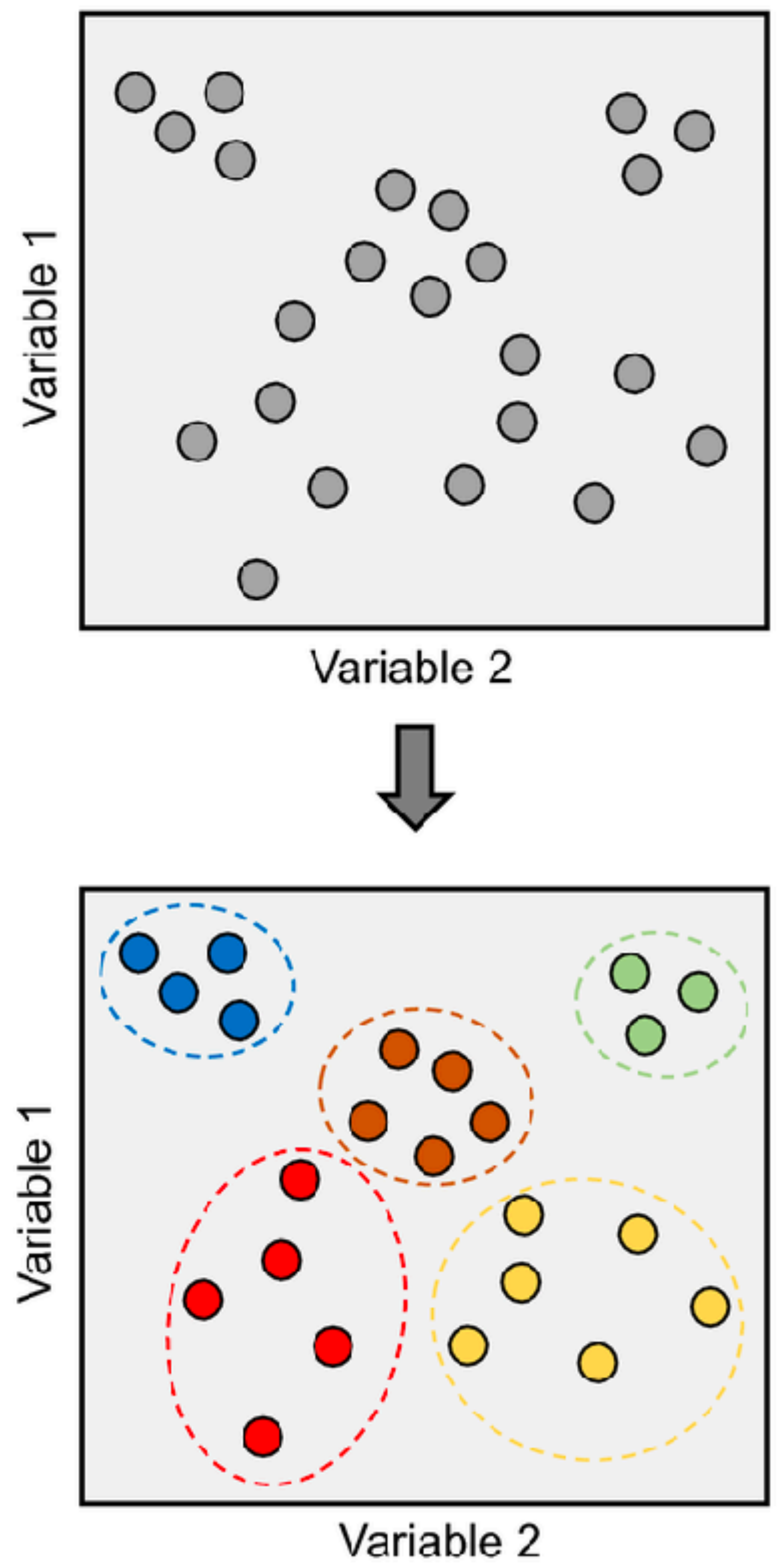
- **Discrimitive** models identify the boundaries that separate spam from non-spam
- **Generative** models learn the distributions of spam and non-spam emails

Overview of methods

Supervised

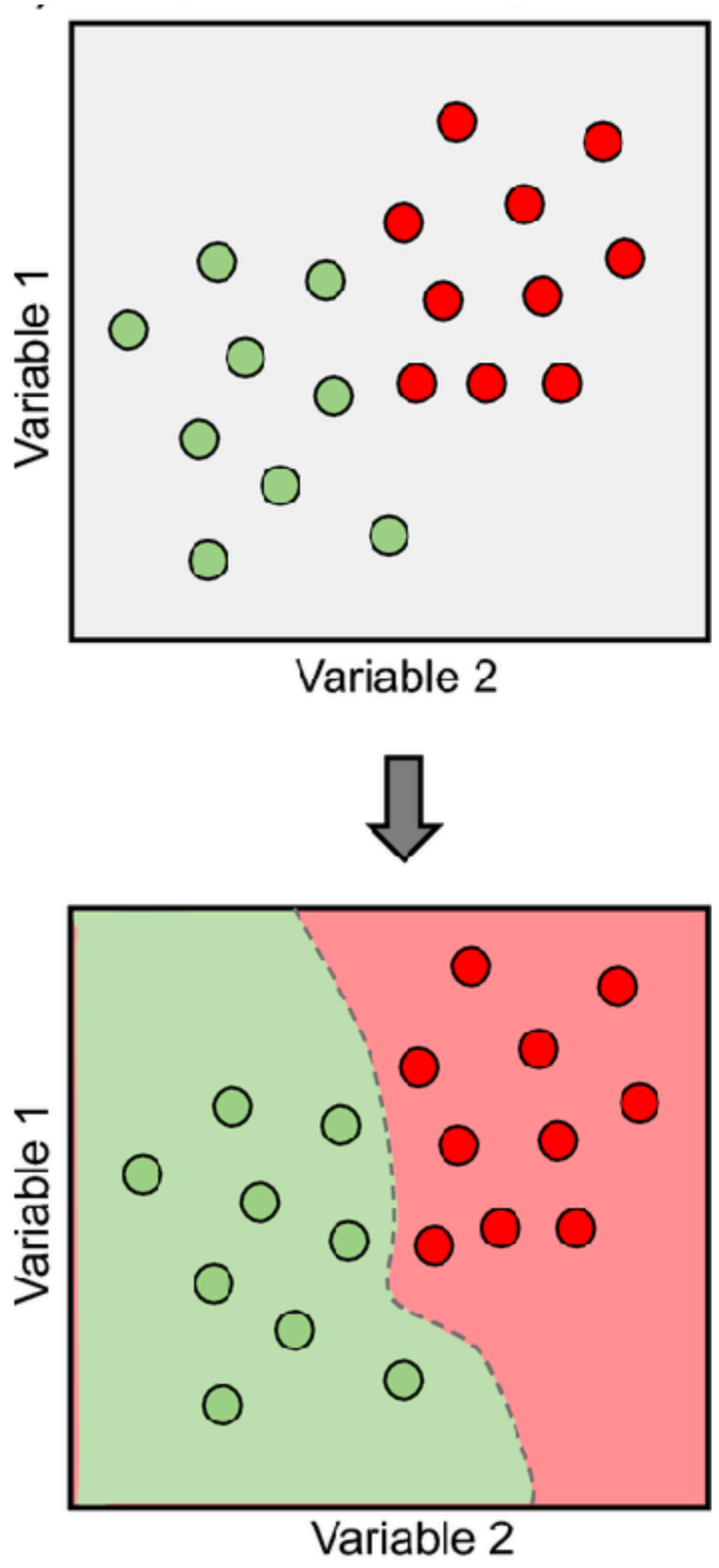


Unsupervised

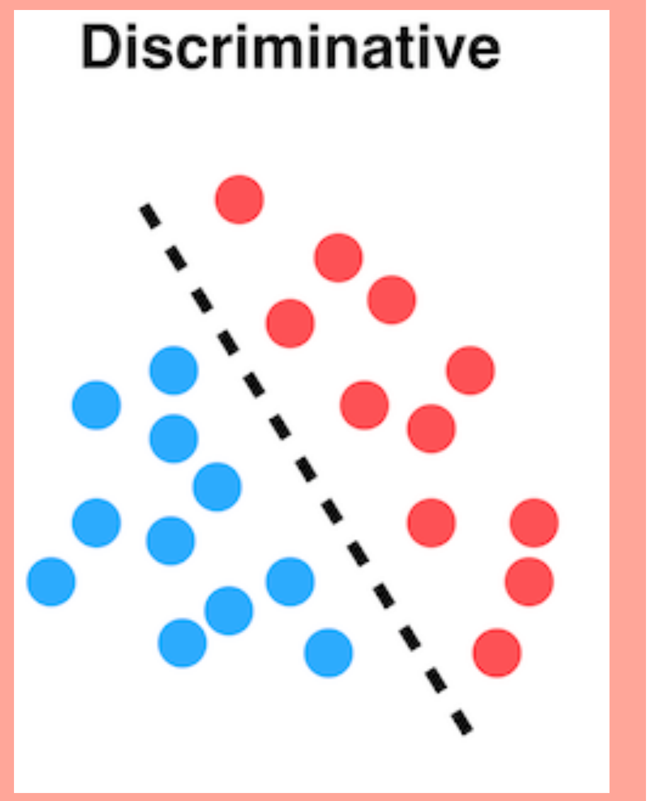


Overview of methods

Supervised



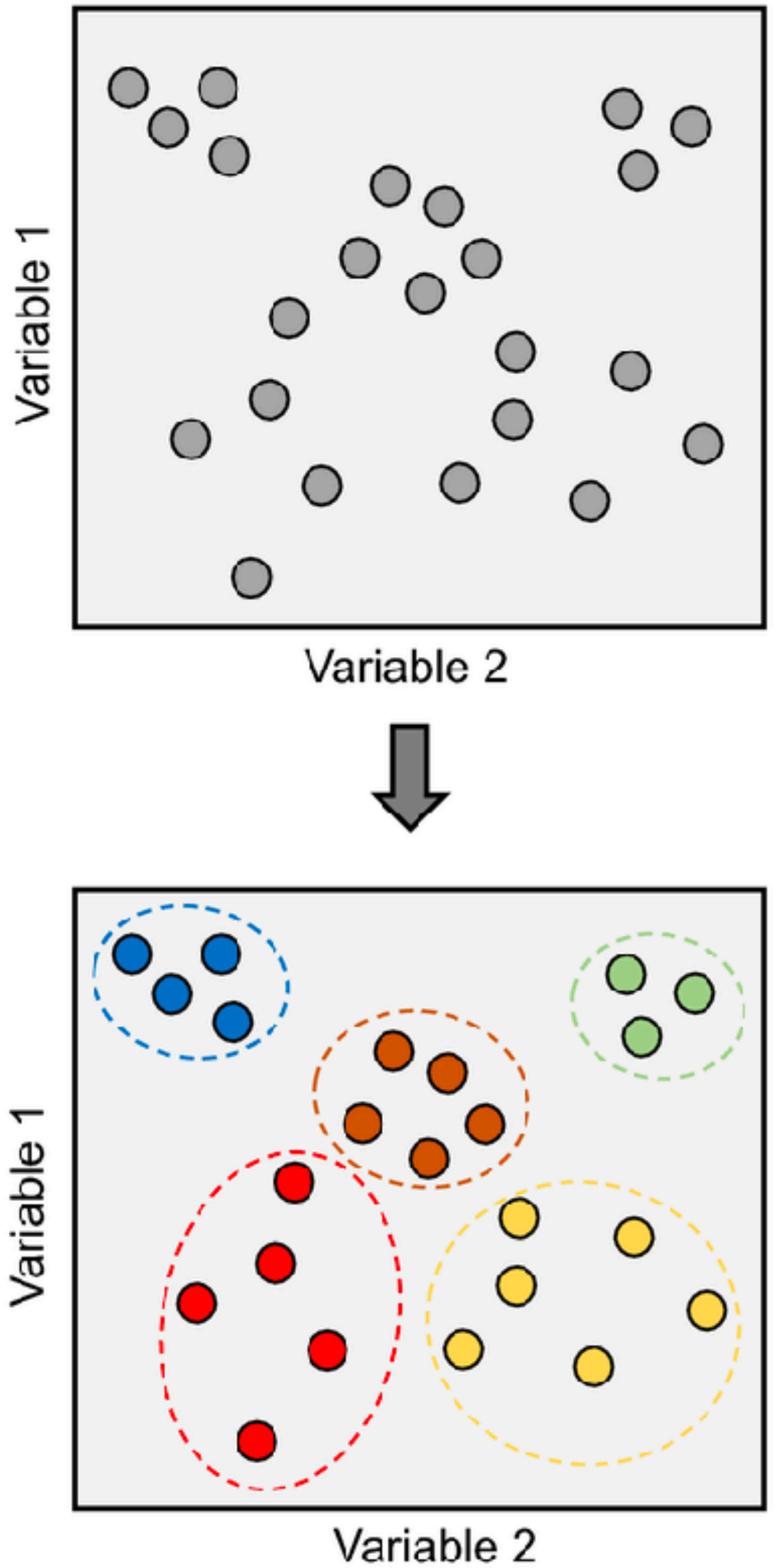
MLPs



Decision trees and random forests

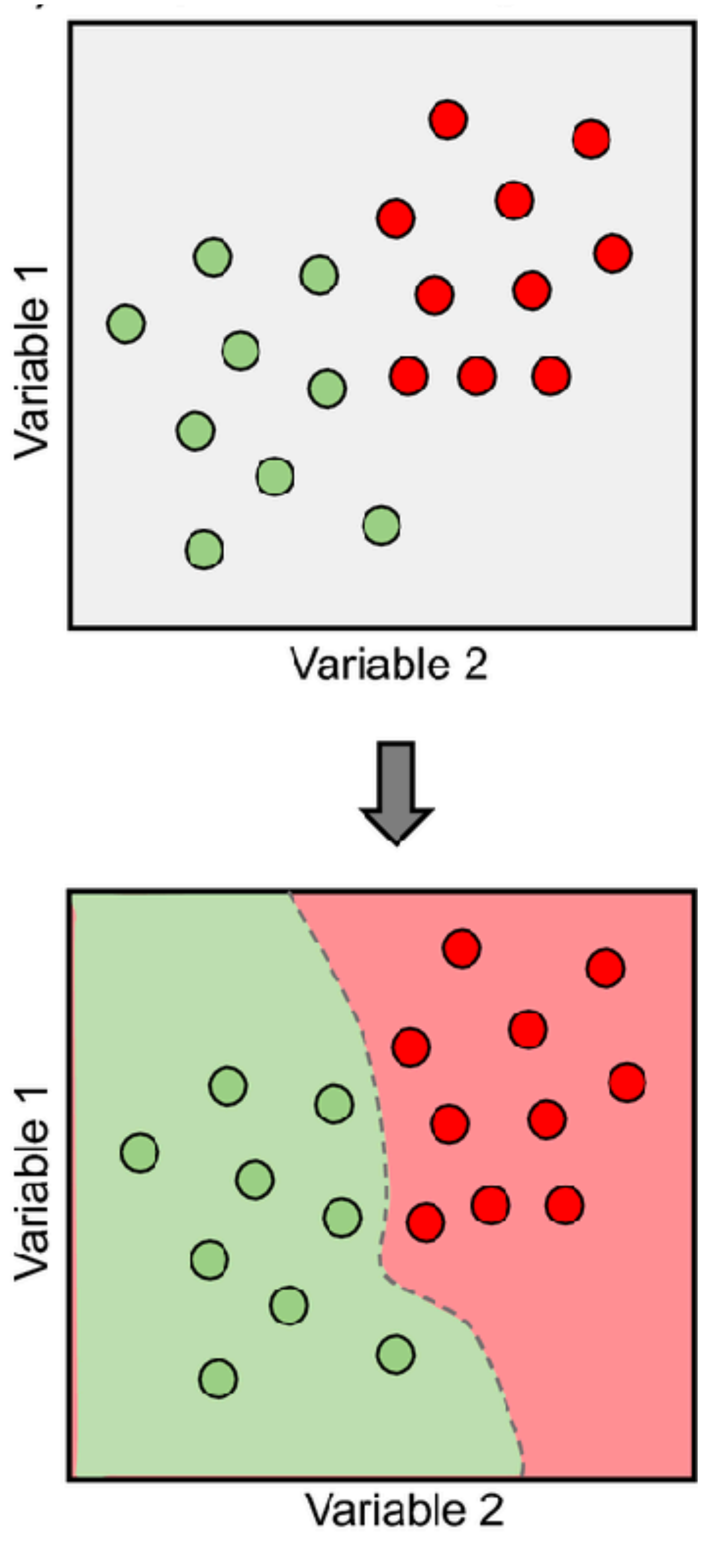
SVMs

Unsupervised



Overview of methods

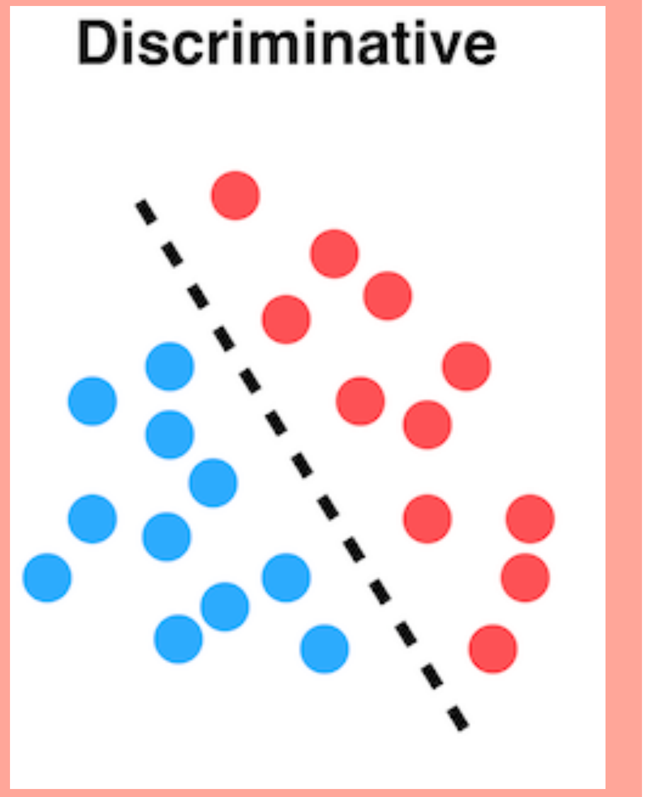
Supervised



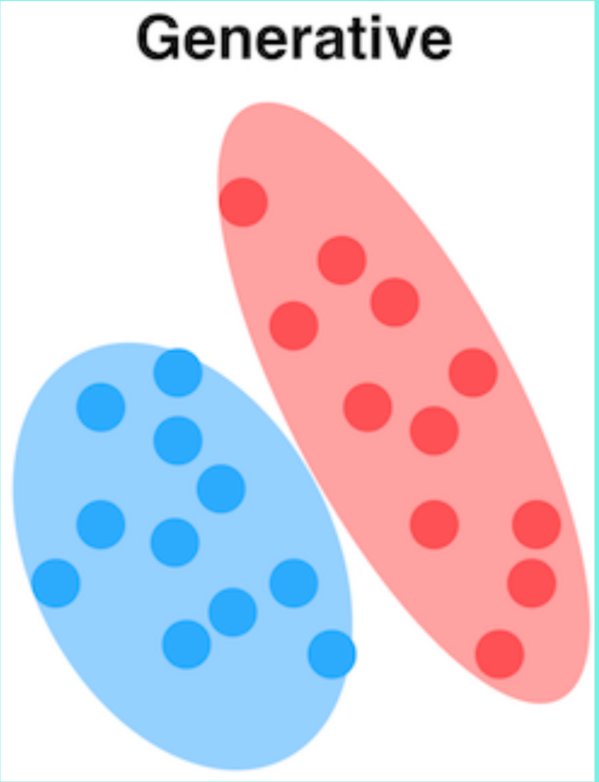
MLPs

Decision trees and random forests

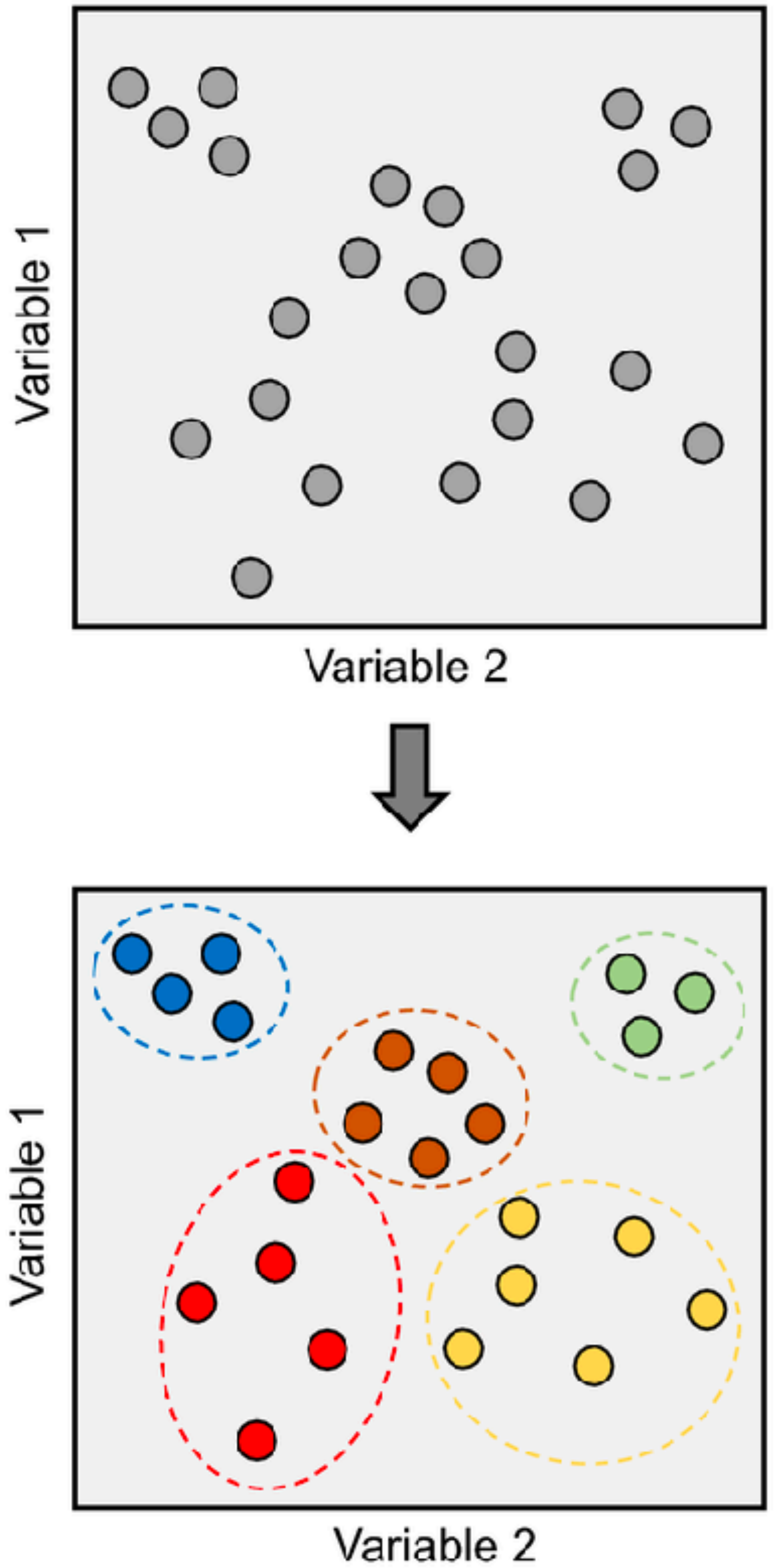
SVMs



Naïve Bayes

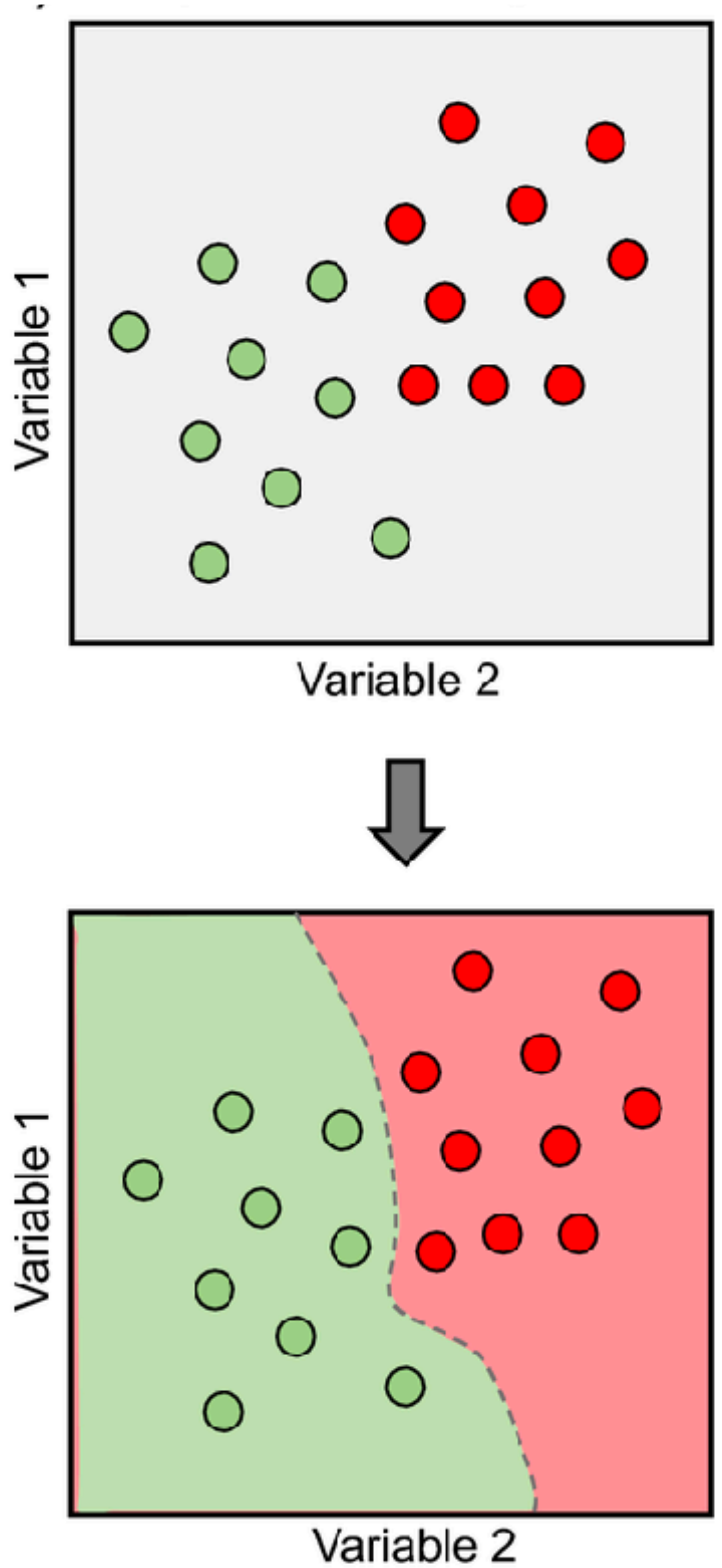


Unsupervised



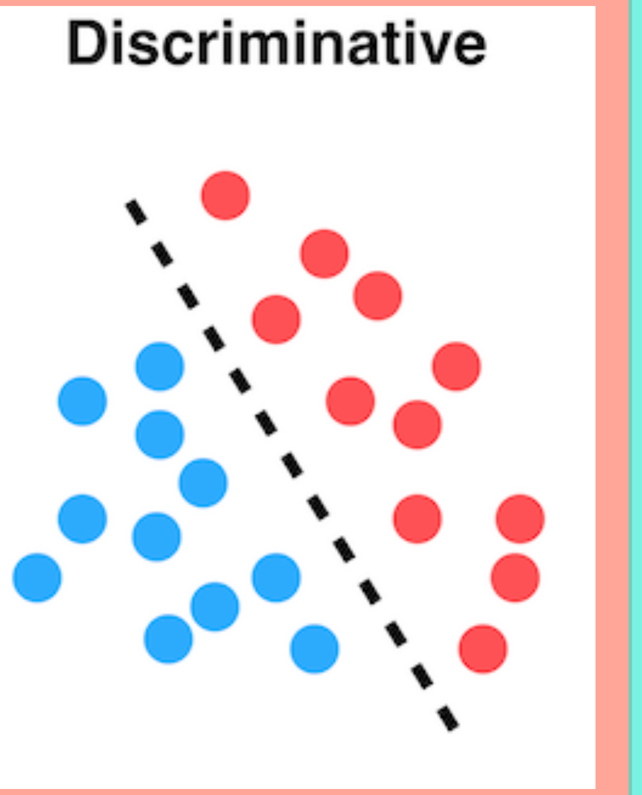
Overview of methods

Supervised



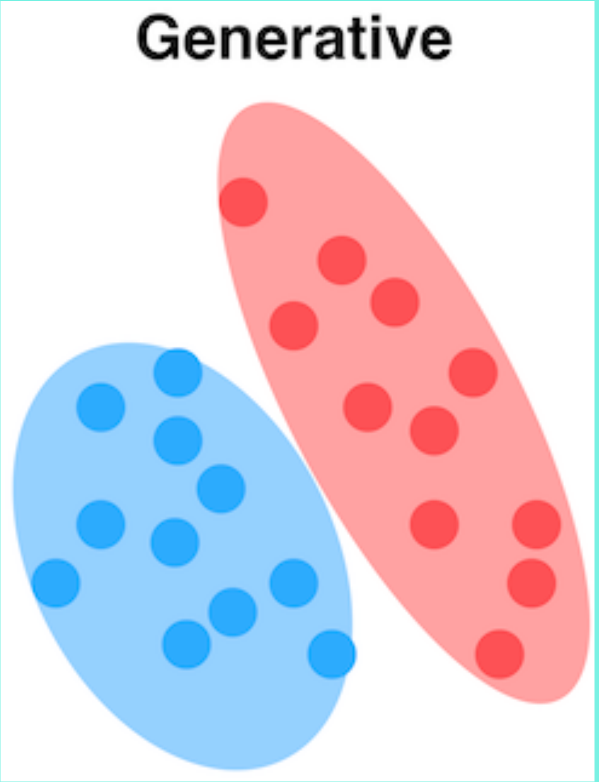
MLPs

Decision trees and random forests



SVMs

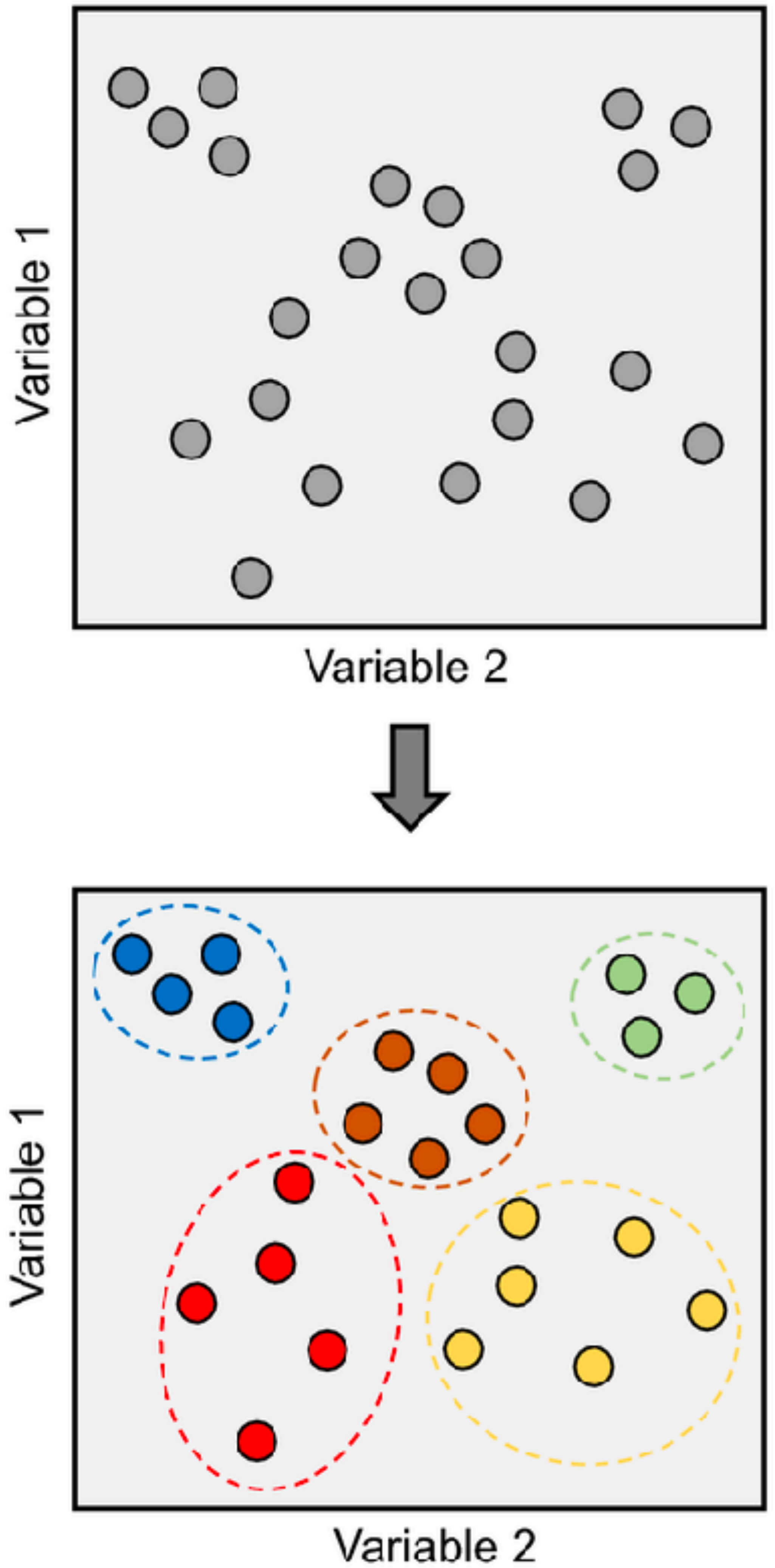
Naïve Bayes



k-Means

GMMs

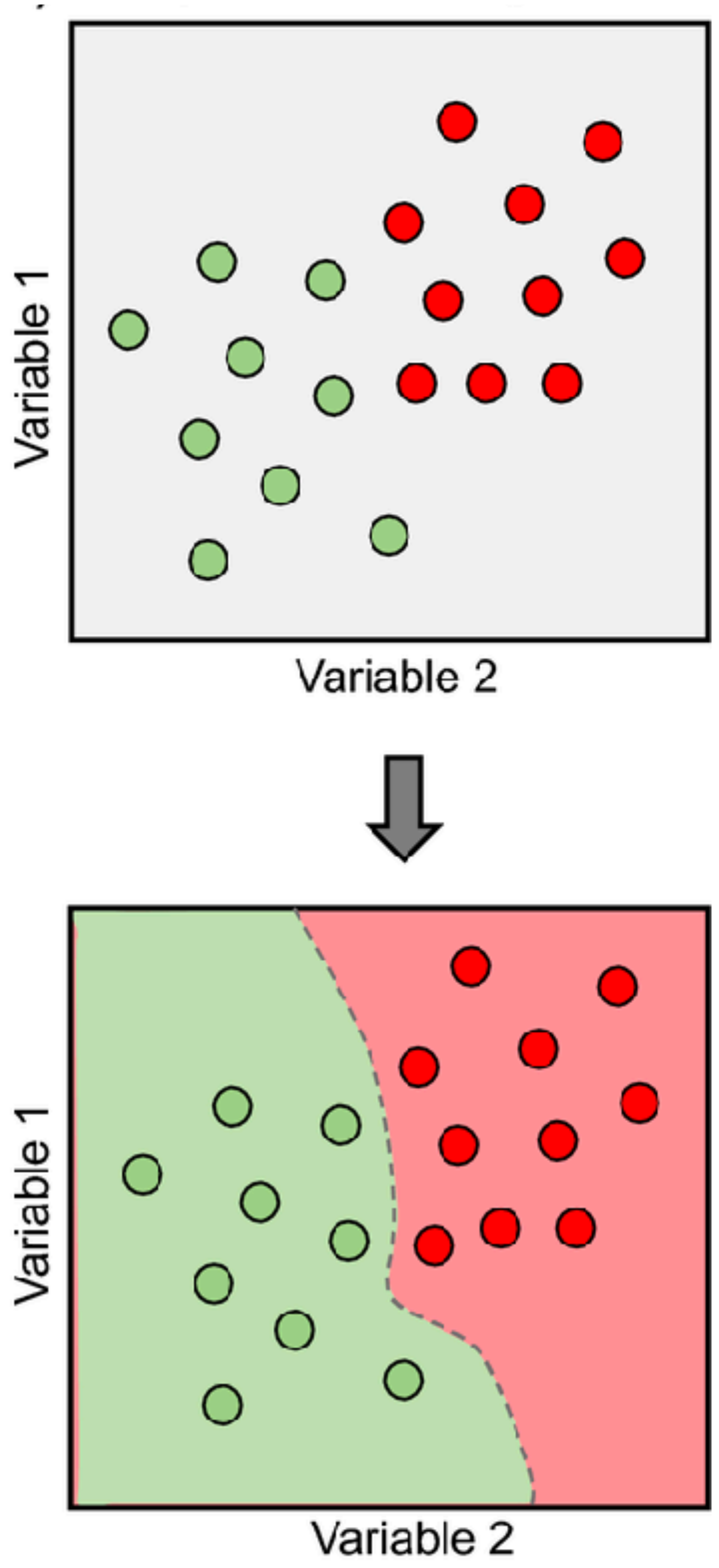
Unsupervised



Overview of methods

Which cognitive theories have similar mechanisms?

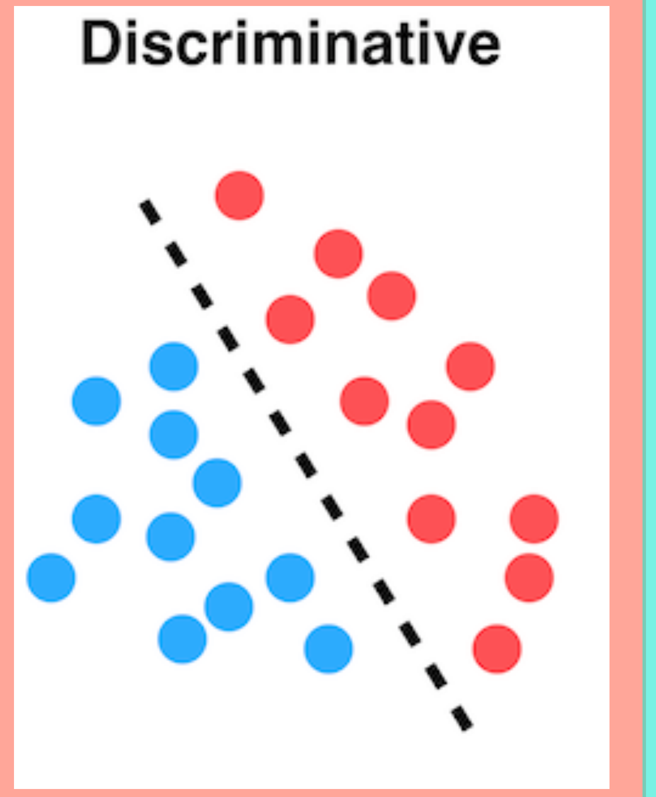
Supervised



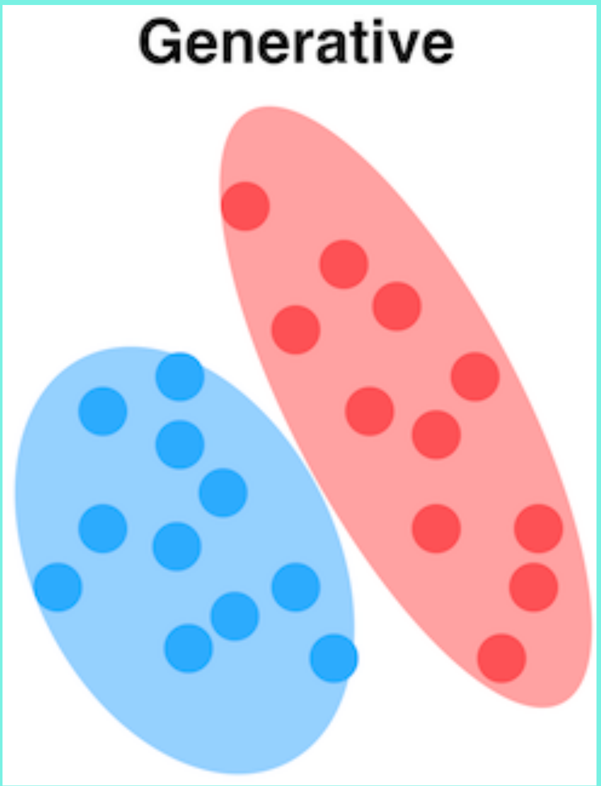
MLPs

Decision trees and random forests

SVMs



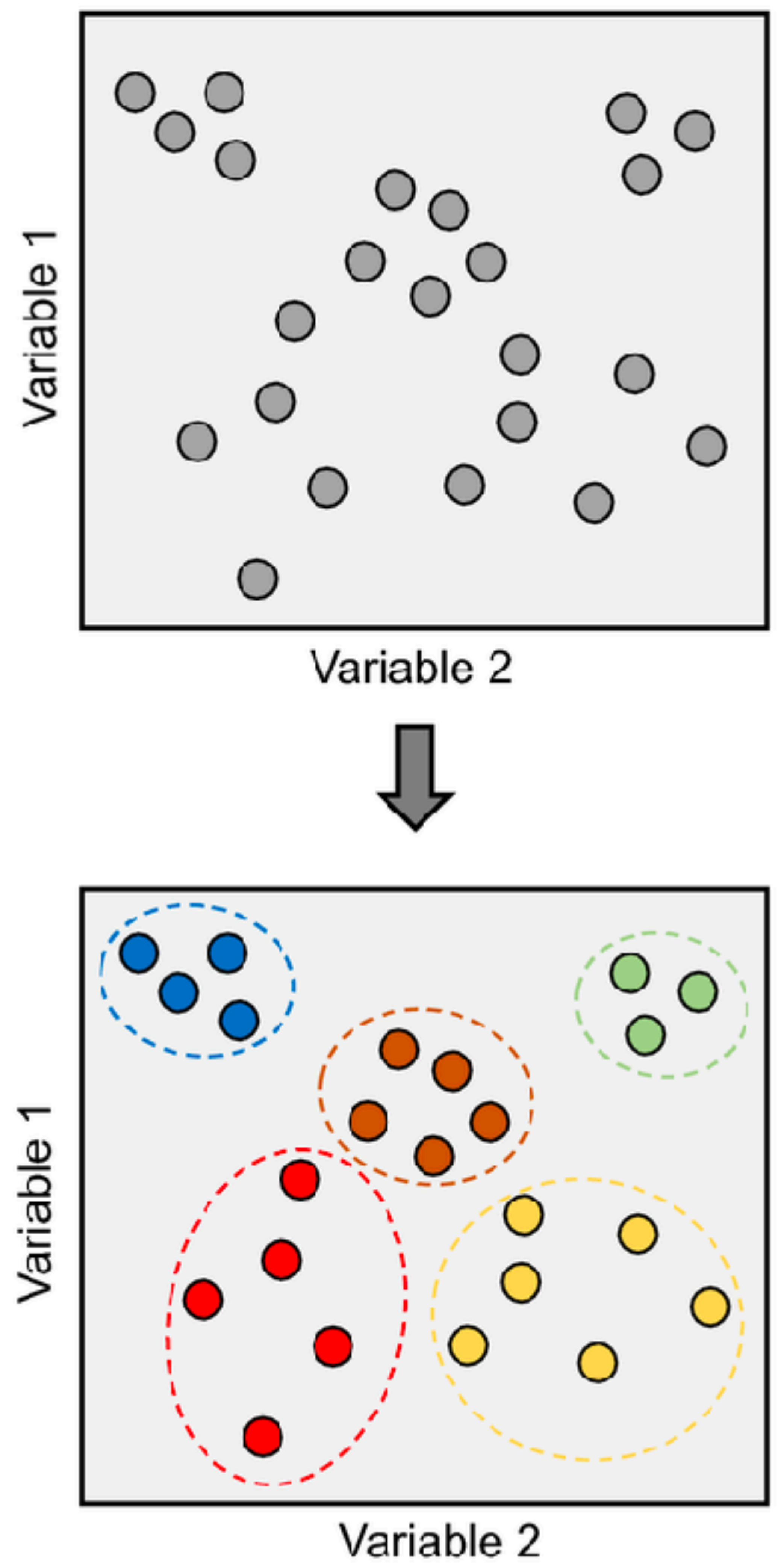
Naïve Bayes



k-Means

GMMs

Unsupervised



Chomsky: Universal Grammar (UG)

- **Plato's problem** (Chomsky, 1986): "How comes it that human beings, whose contacts with the world are brief and personal and limited, are nevertheless able to know as much as they do know?"
    - Language acquisition in children suggests they "attain infinitely more than they experience"
- **Poverty of the stimulus**: it seems like there is a disparity between the amount of input (experience) and the output (acquired language)
    - Thus, there is a missing factor and that factor is Universal Grammar (UG): "*the system of categories, mechanisms, and constraints that shared by all human languages and considered to be innate*"
- Output (language ability) > input (experience)
- Therefore: language = input + UG

Landauer & Dumais (1997)

Solving Plato's Problem with Latent Semantic Analysis (LSA)

- **Latent semantic analysis (LSA)**
  - Describe the *similarity* between words based on the similarity of contexts in which they occur
- One of the first computational approaches to solving Plato's problem
  - Focusing on semantic learning (i.e., the meaning of words) rather than grammar learning (the relational structure or syntax between words)
  - Specifically modeling "induction" (reasoning beyond the available evidence) in semantics

A

Text sample (context)

Word/	1	.	.	.	.	.	.	.	.	.	.	.	.	.	30,000
1	x	x	x	x	x	x	.	.	.	x	x	x	x	x	x
.	x	x	x	x	x	x	.	.	.	x	x	x	x	x	x
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
.	x	x	x	x	x	x	.	.	.	x	x	x	x	x	x
60,000	x	x	x	x	x	x	.	.	.	x	x	x	x	x	x

B

Factor (dimension)

Word/	1	.	.	.	300
1	y	.	.	.	y
.	y	.	.	.	y
.	.	.	.	.	.
.	.	.	.	.	.
.	.	.	.	.	.
.	y	.	.	.	y
60,000	y	.	.	.	y

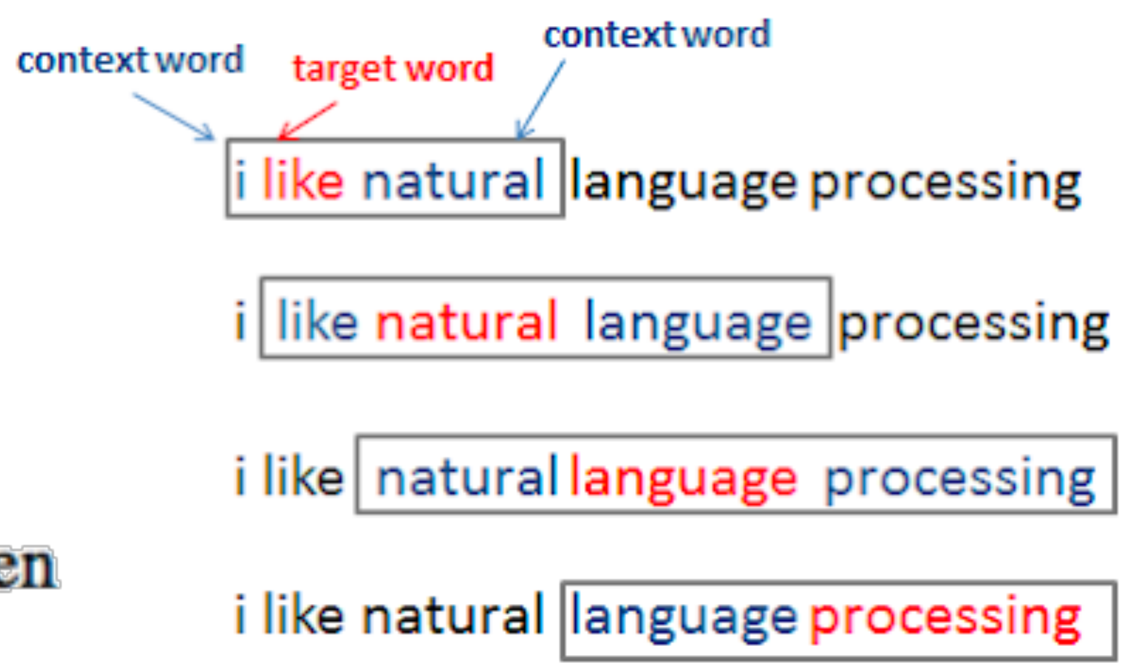
C

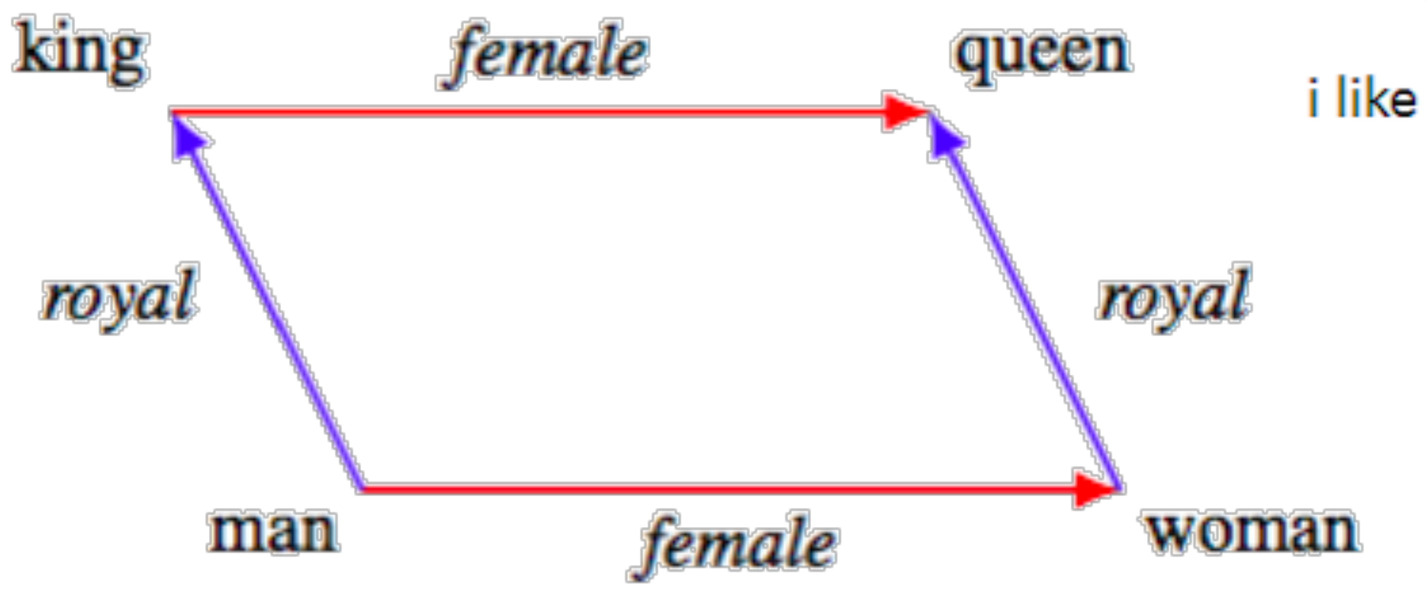
Factor (dimension)

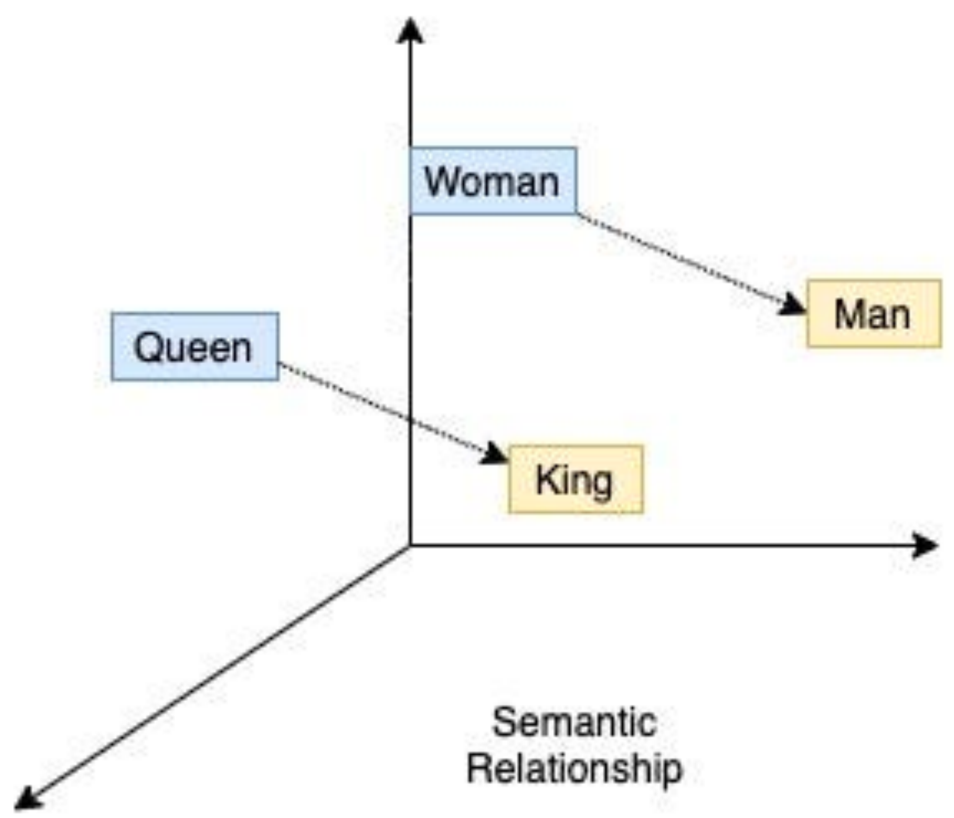
Sample/	1	.	.	.	300
1	z	.	.	.	z
.	.	.	.	.	z
.	z	.	.	.	z
.	z	.	.	.	z
30,000	z	.	.	.	z

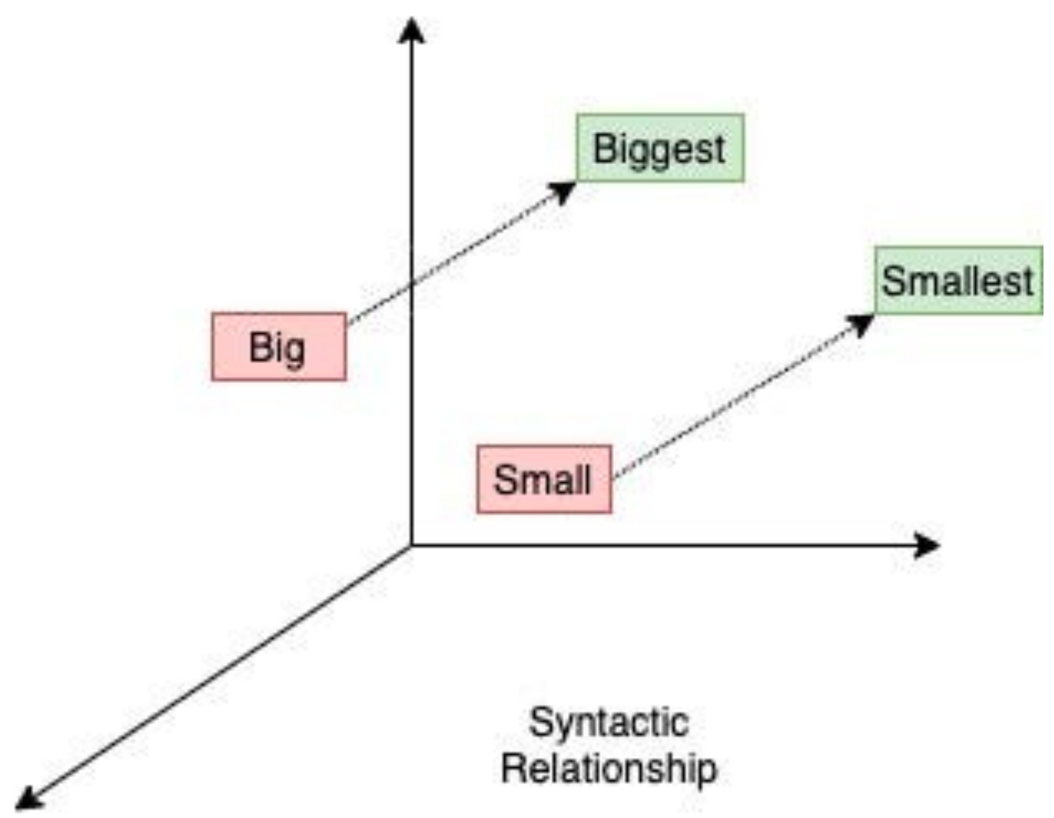
Word2vec, RNNs, and LSTMs

context window

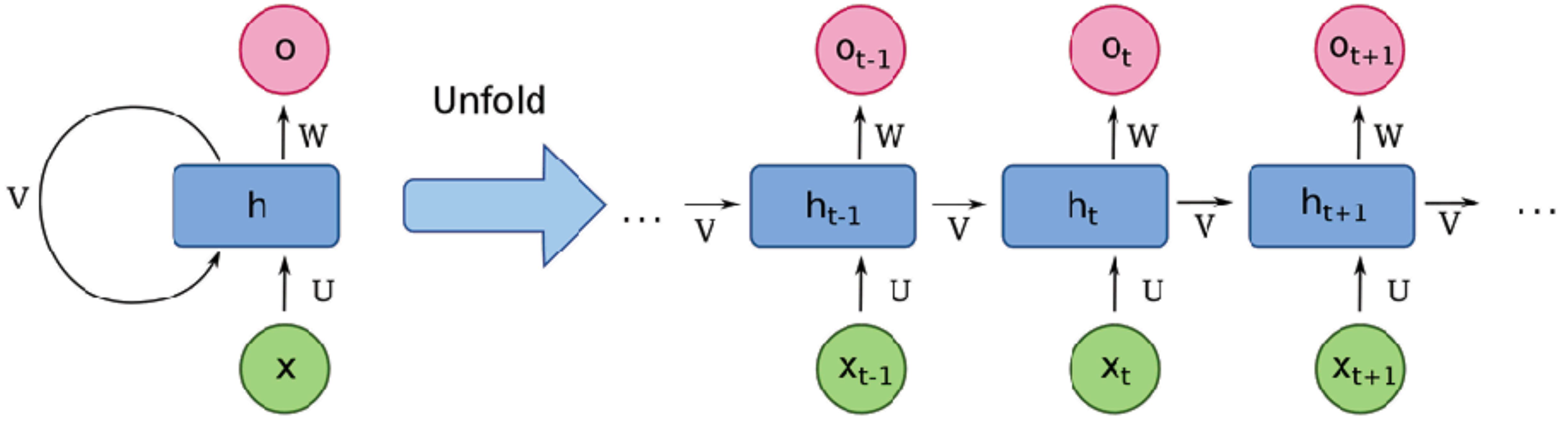




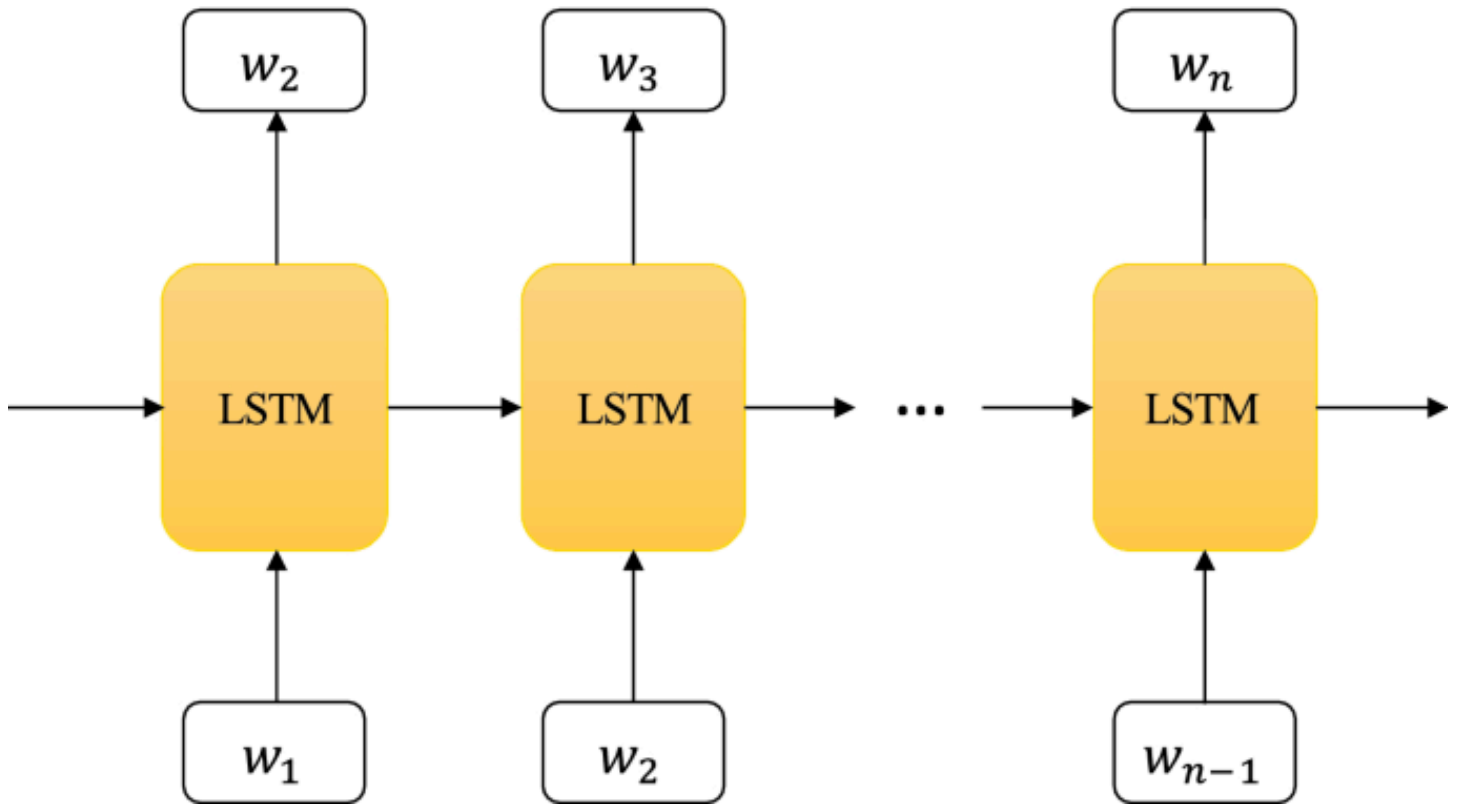




RNNs

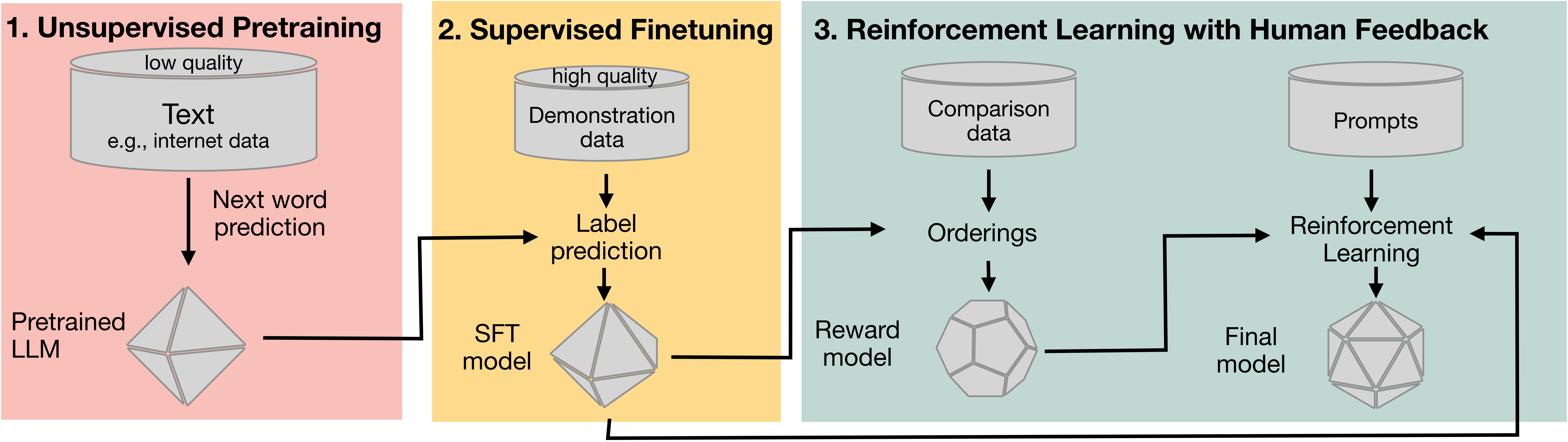


LSTMs

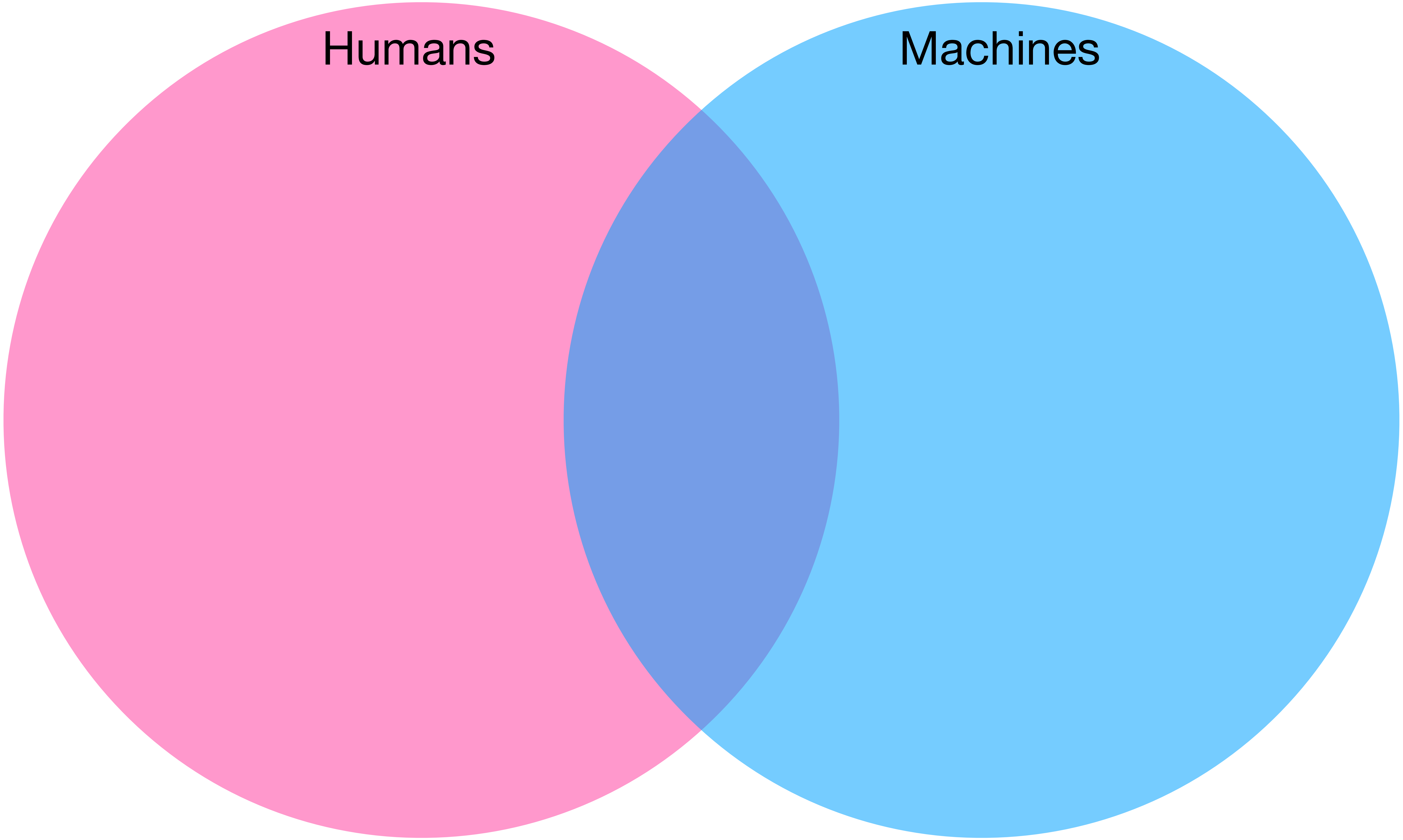


How do LLMs learn

- Combination of multiple Machine Learning techniques
  1. ***Unsupervised*** pre-training: predict the next word in a sentence
  2. ***Supervised*** fine-tuning: predict hand-curated labels
  3. ***Reinforcement learning*** with human feedback: adapt policy based on human raters



General Principles



Google doc

Final tutorial

- For **tomorrow's tutorial,** please prepare 2-3 candidate exam questions:
  - Short answer question format
  - You are incentivized to bring plausible questions that would be sufficiently challenging, thought provoking, and feasible
  - Good questions will be included on the exam
- We will go over these questions and you can ask me anything else about questions you still have about the exam or about anything else you like