General Principles of Human and Machine Learning

Lecture 11: General Principles

Dr. Charley Wu
Dr. Charline Tessereau

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

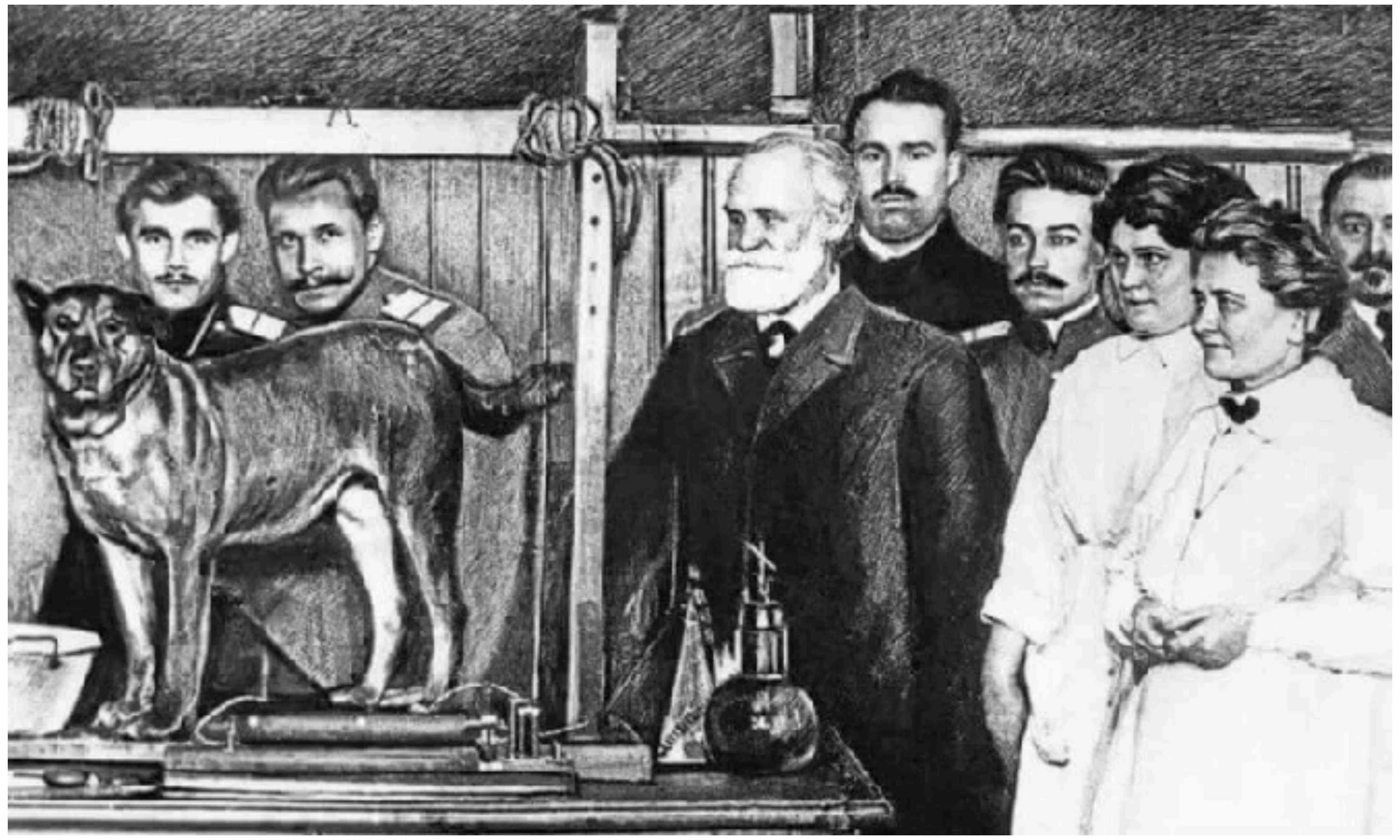
Exam

- Combination of multiple choice and short answer questions
  - No complex calculations are needed
  - No need to memorize formulas or dates
  - Focus is on understanding the main theoretical ideas and how they connect together across fields
- Thursday July 27th, 10:30-12:00
  - same room/time as the lecture
  - Bring pens/pencils
  - Register on ALMA if possible, otherwise we can enter the grades manually if your study program doesn't allow it
- Second taking is scheduled for Oct 12, 10:30-12:00
  - Please contact us if you are interested in taking it by Oct 1st

Revisiting our original questions

- What is learning?
- What aspects of learning are the same across biological and artificial systems? What is different?
- What has the study of biological intelligence informed us about artificial systems?
- What can artificial intelligence teach us about biological intelligence?

Foundations of Biological Learning



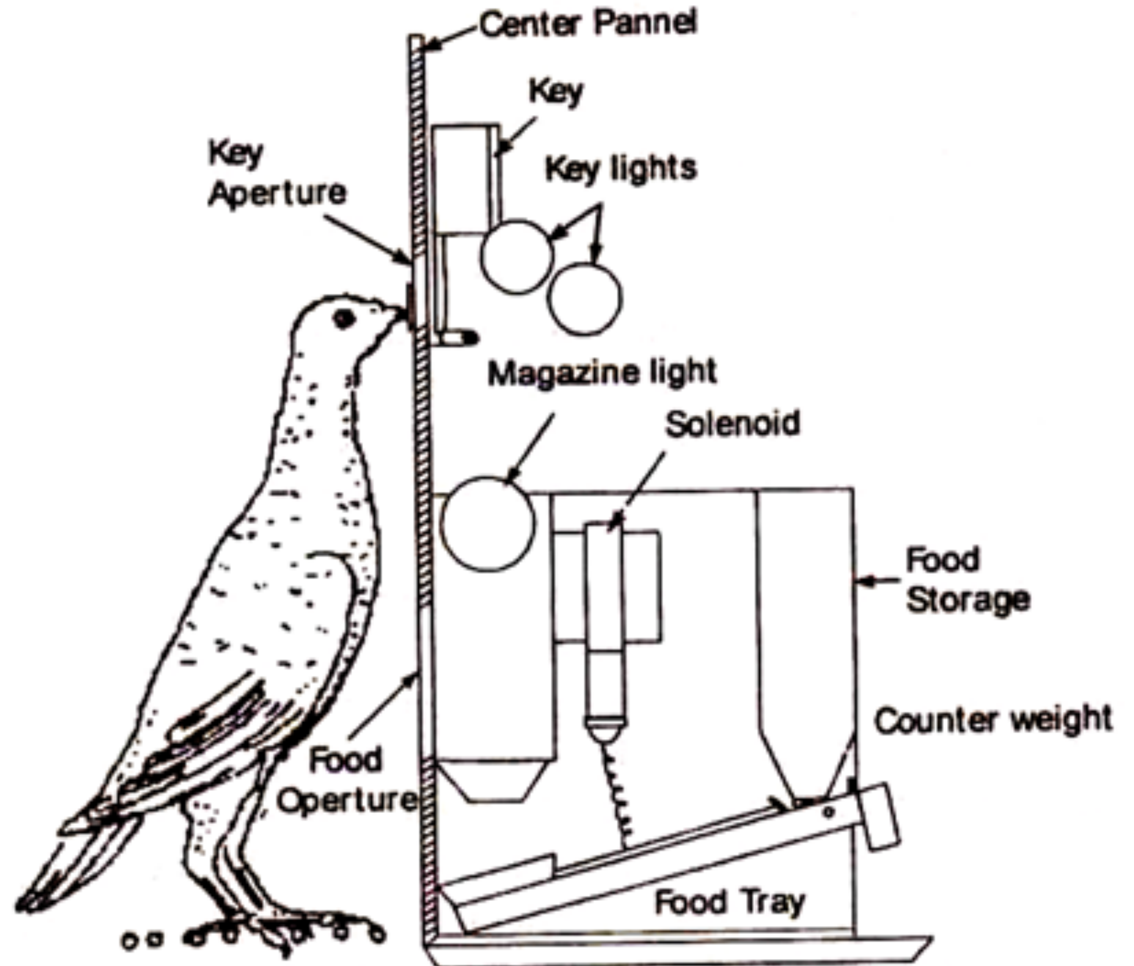
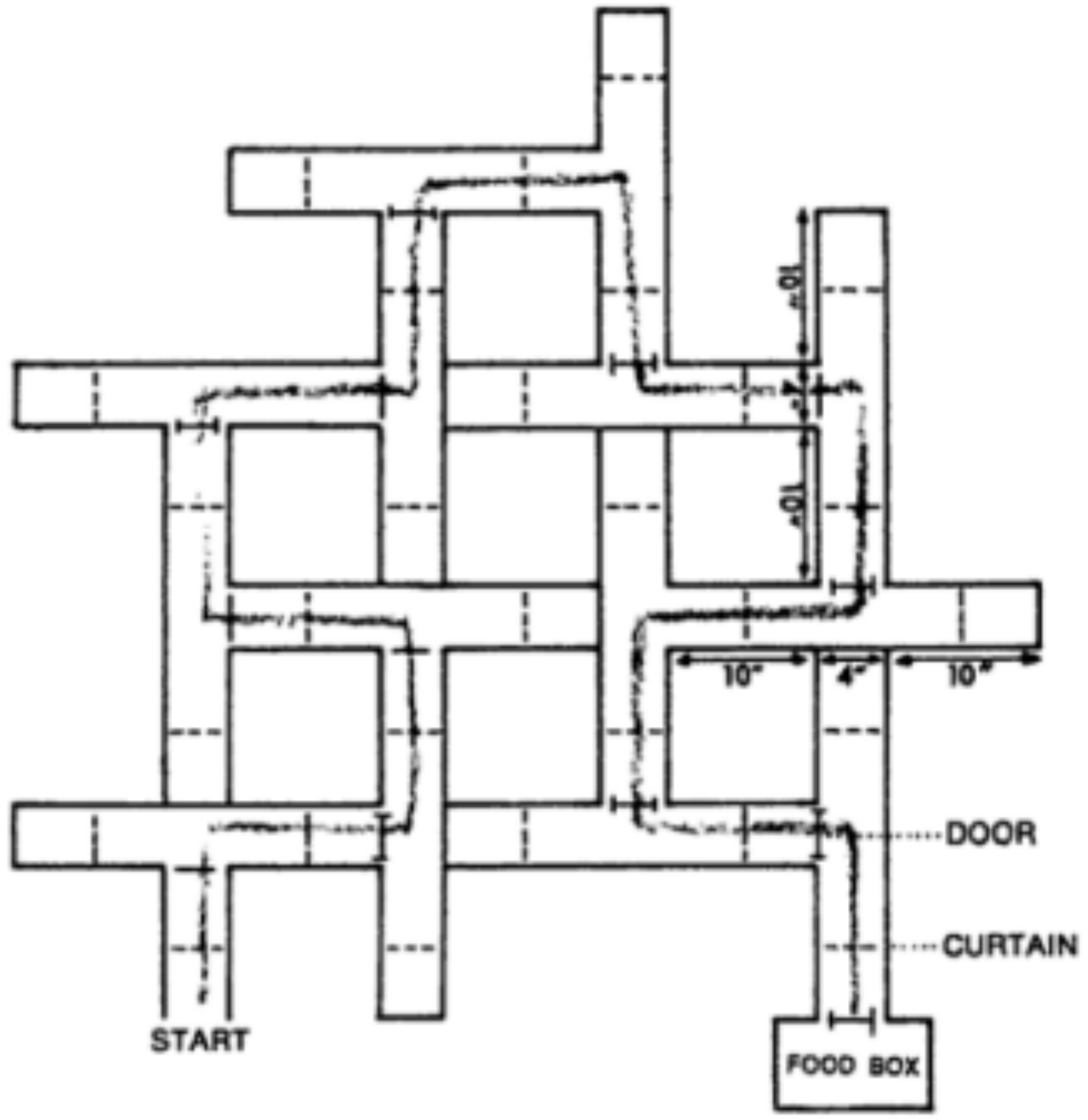
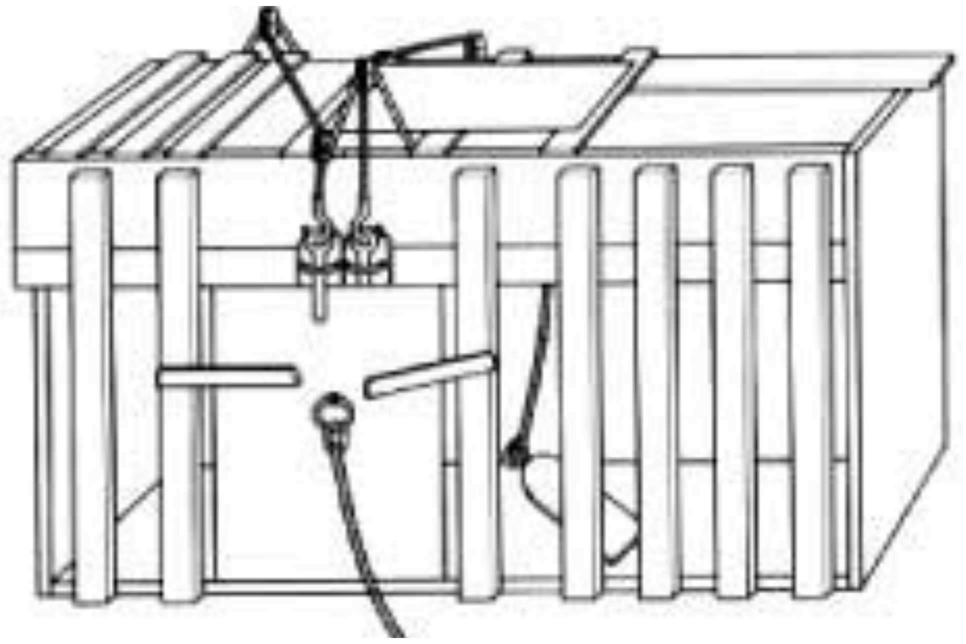


Illustration. Skinner box as adapted for the pigeon.



A brief timeline of early research on biological learning

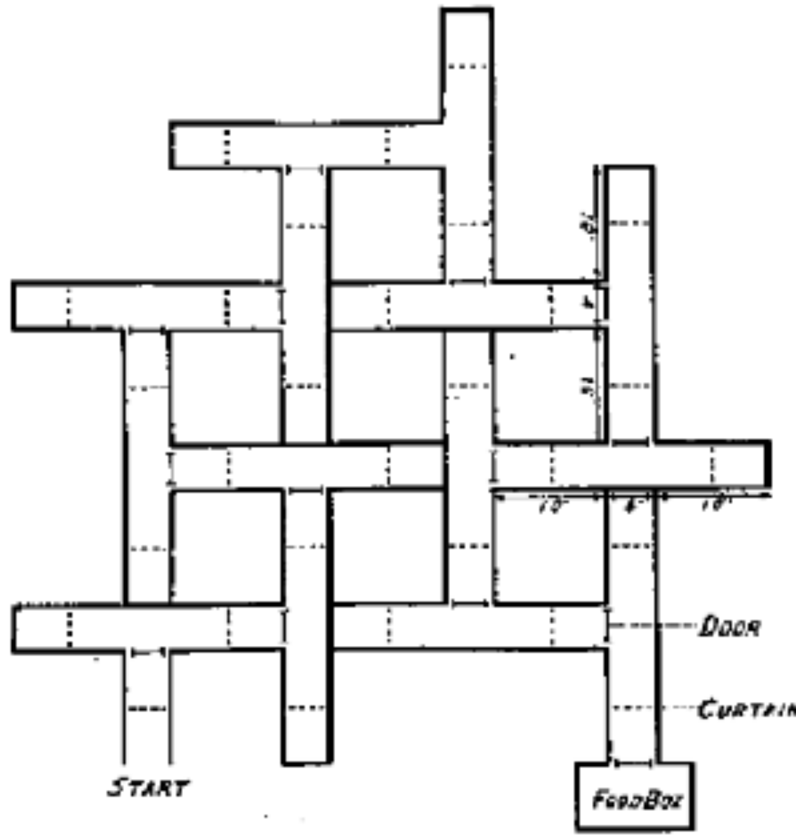


Thorndike (1911)

Pavlov (1927)

Skinner (1938)

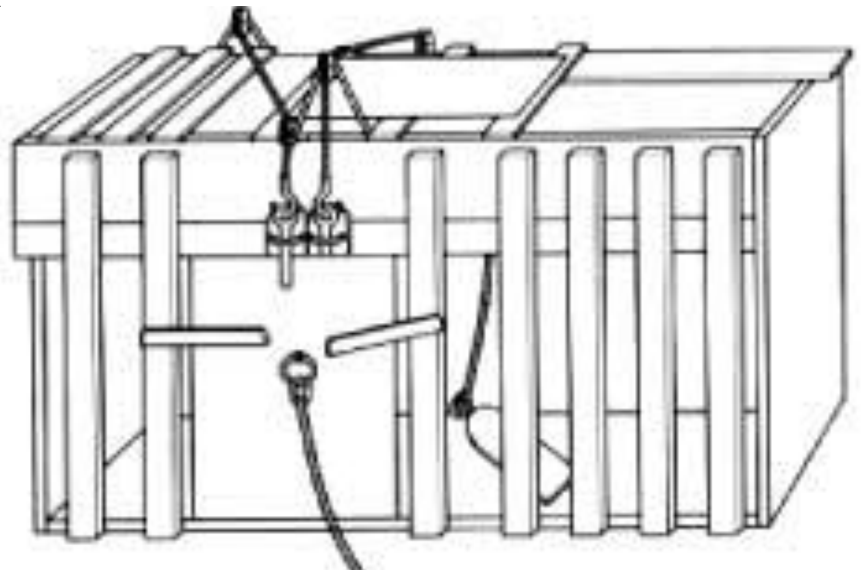
Tolman (1948)



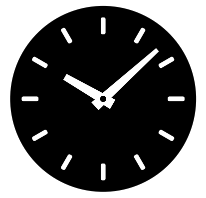
Plan of maze
14-Unit T-Alley Maze
FIG. 1
(From M. H. Elliott, The effect of change of reward on the maze performance of rats. *Univ. Calif. Publ. Psychol.*, 1928, 4, p. 20.)

Thorndike’s Laws

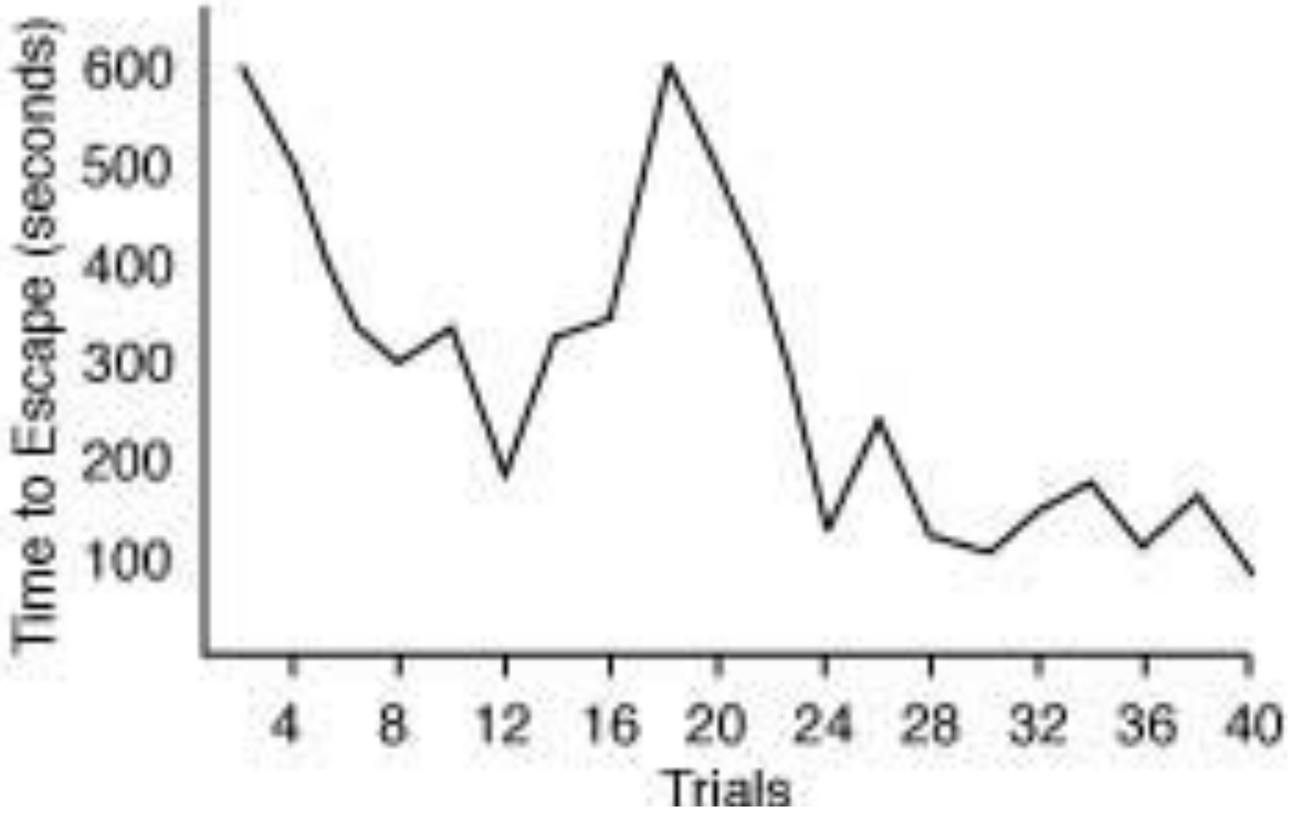
Law of Effect



Puzzle Box



Time to escape

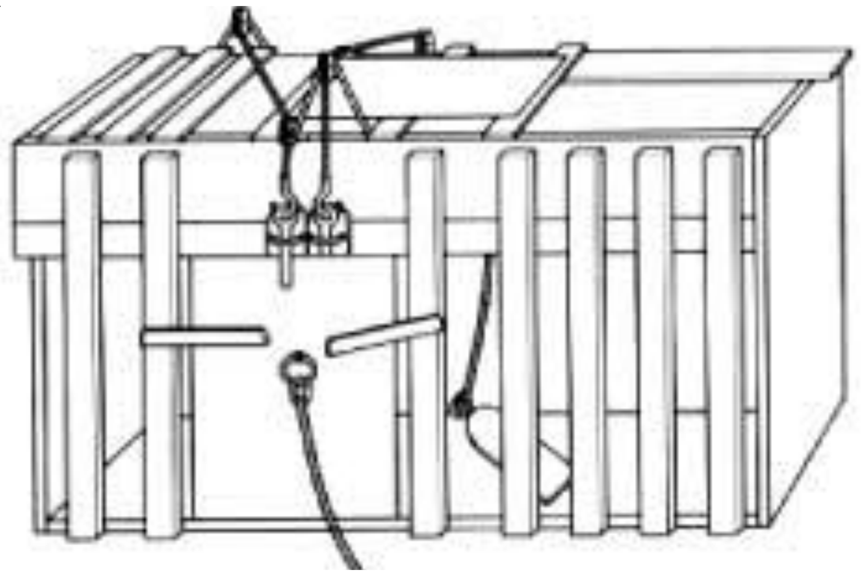


Law of Exercise

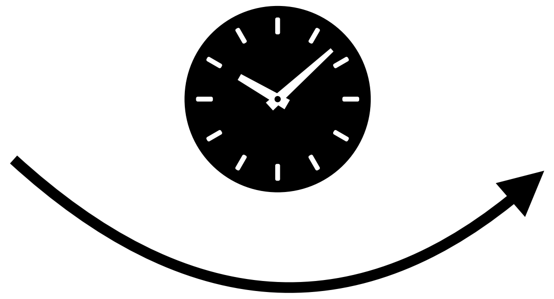
Thorndike's Laws

Law of Effect

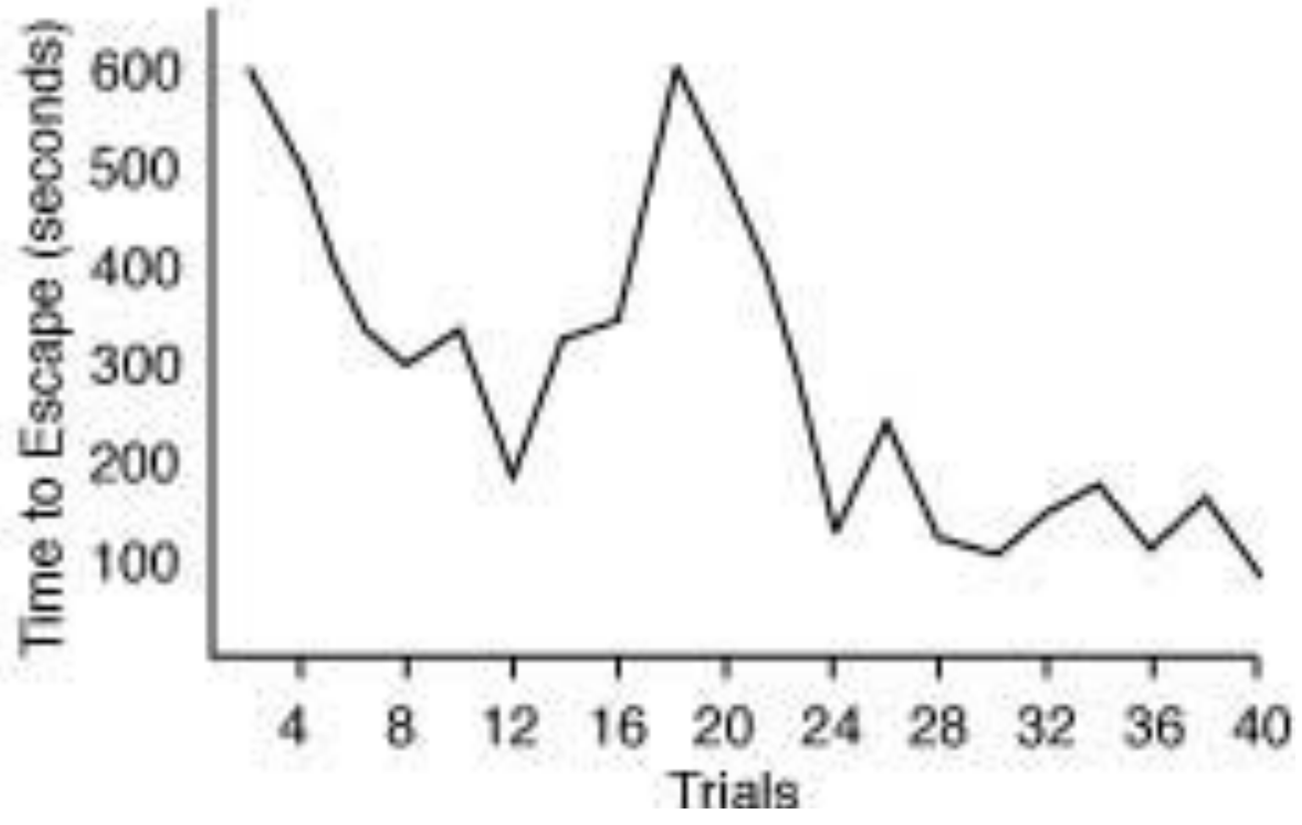
Cat



Puzzle Box



Time to escape



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

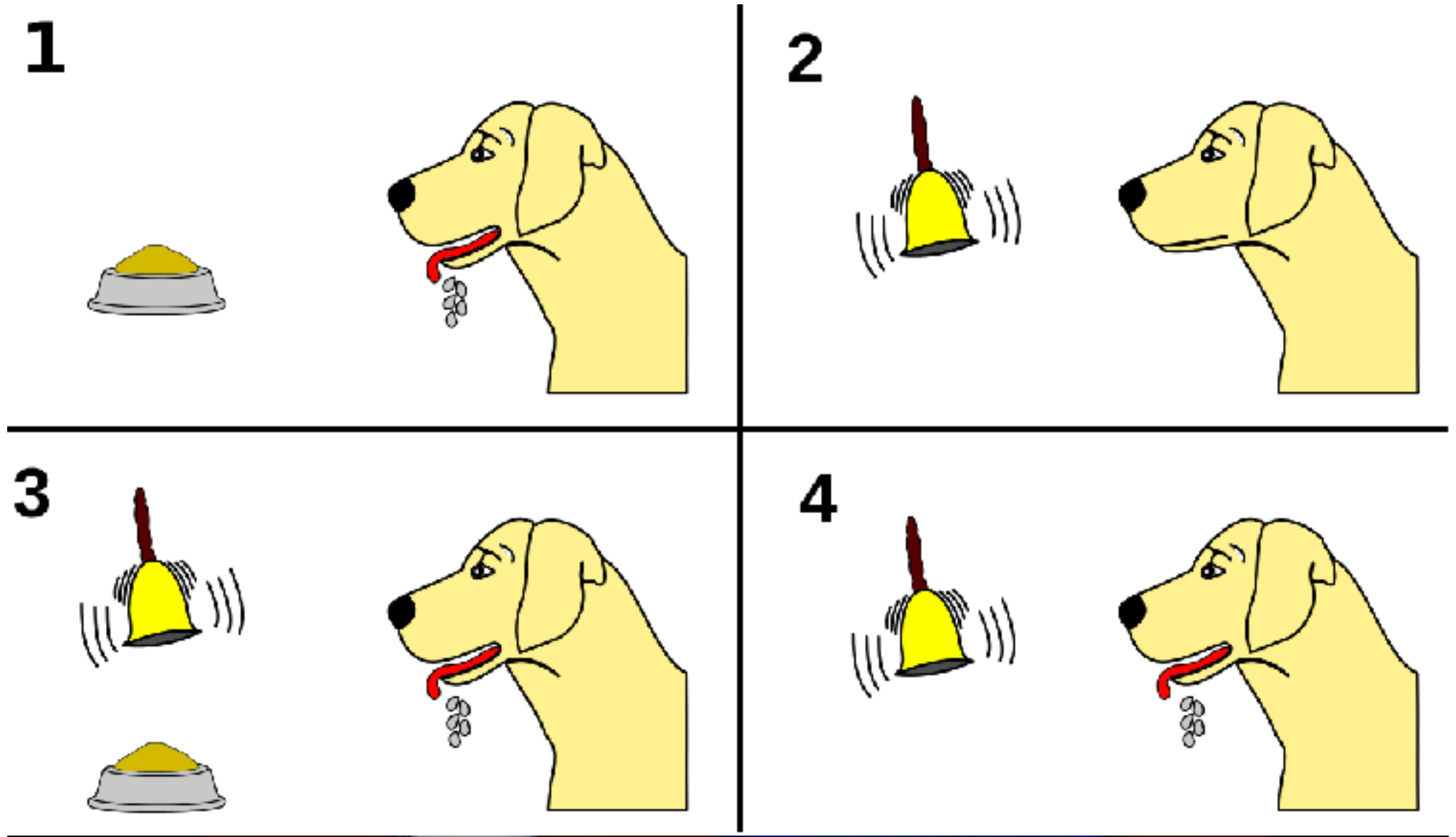
Law of Exercise

Independent of success, exercising connection between stimulus and response strengthens the association (i.e., habits)

Classical and Operant Conditioning

Classical Condition (Pavlov, 1927)

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

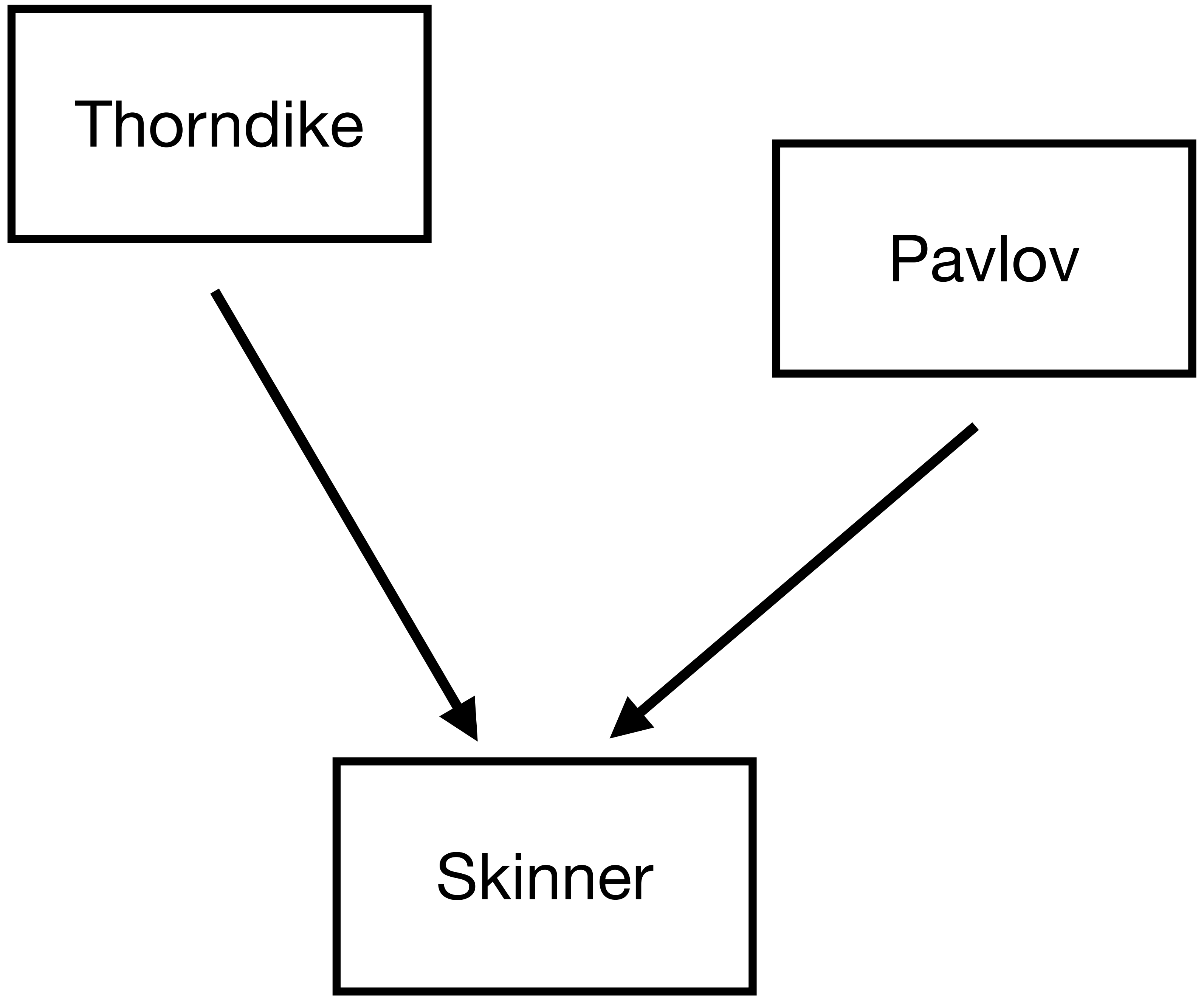


Operant Condition (Skinner, 1938)

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

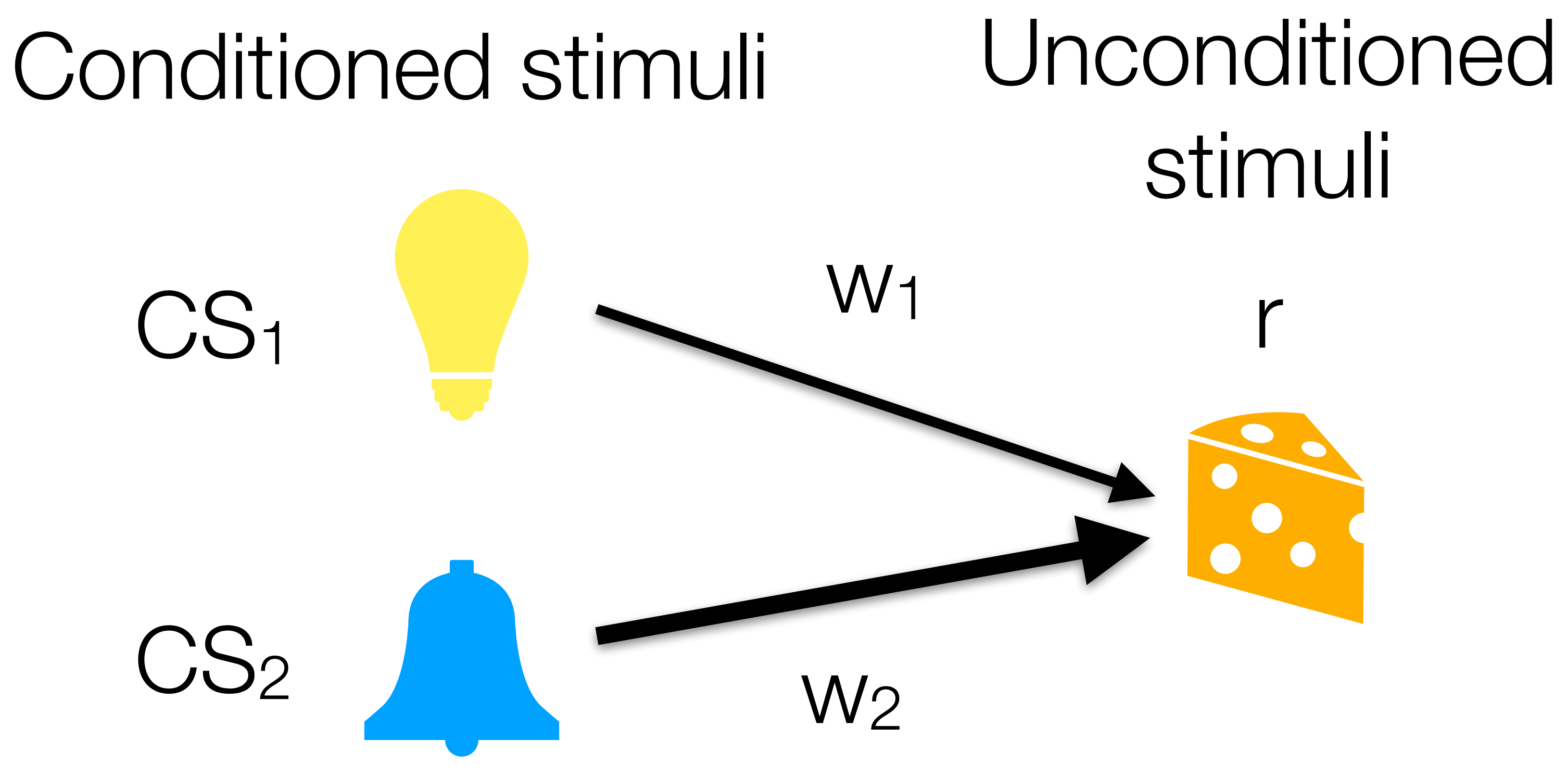
What is the relationship between Thorndike, Pavlovian condition, and Operant condition?

- Each are verbal theories, describing a pattern of behavioral phenomenon
  - Thorndike: successful actions get strengthened
  - Pavlov: resonse to US get transferred to CS
  - Skinner: conditioning not only applies to responses, but also actions/behavior



Rescorla-Wagner

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



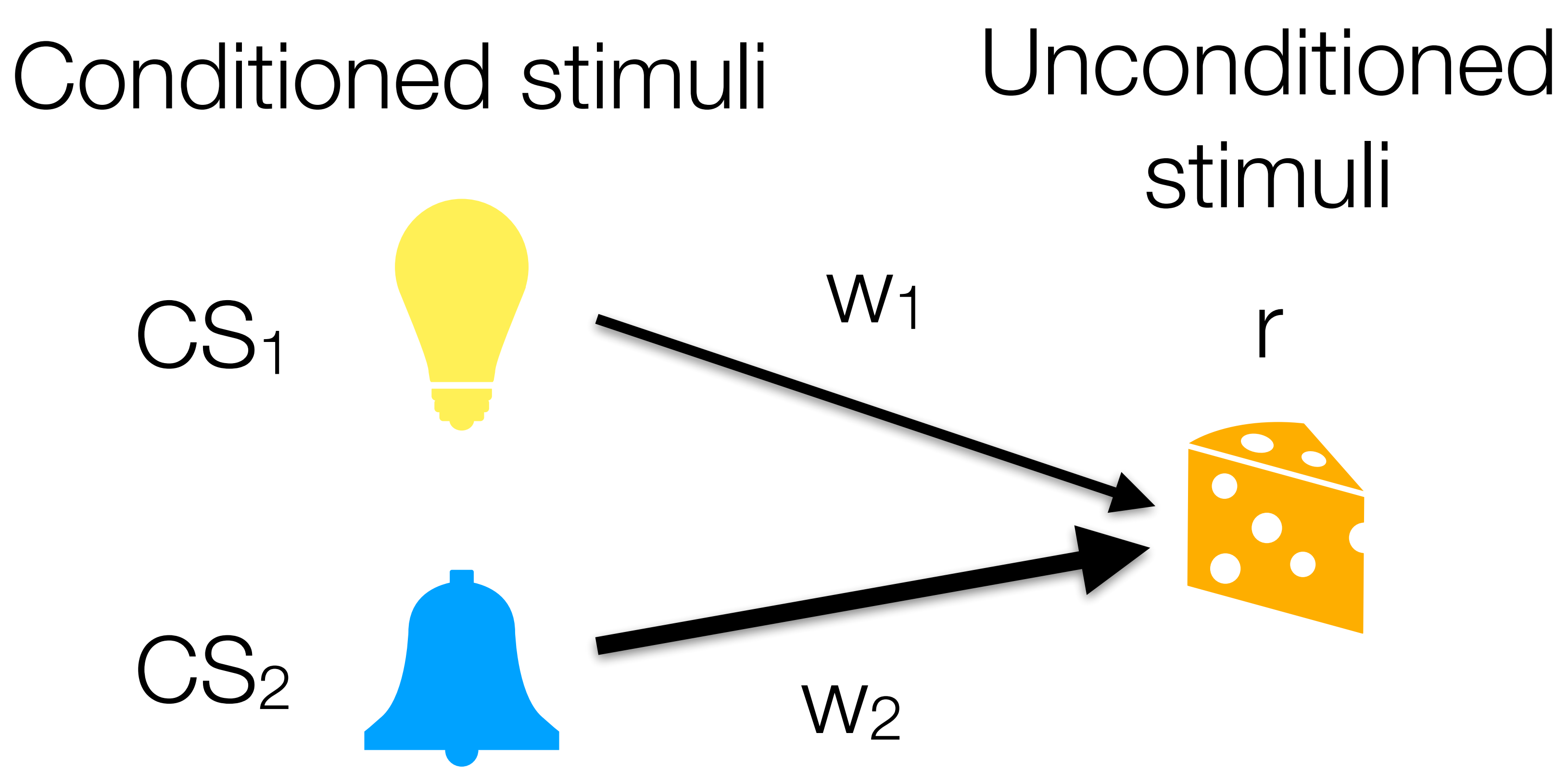
Reward prediction

$$\hat{r}_t = \sum_i \mathrm{CS}_i^t w_i$$

Weight update

$$w_i \leftarrow w_i + \eta(r_t - \hat{r}_t)$$

Rescorla-Wagner



Rescorla-Wagner model

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

Reward prediction

$$\hat{r}_t = \sum_i \mathrm{CS}_i^t w_i$$

Reward expectation

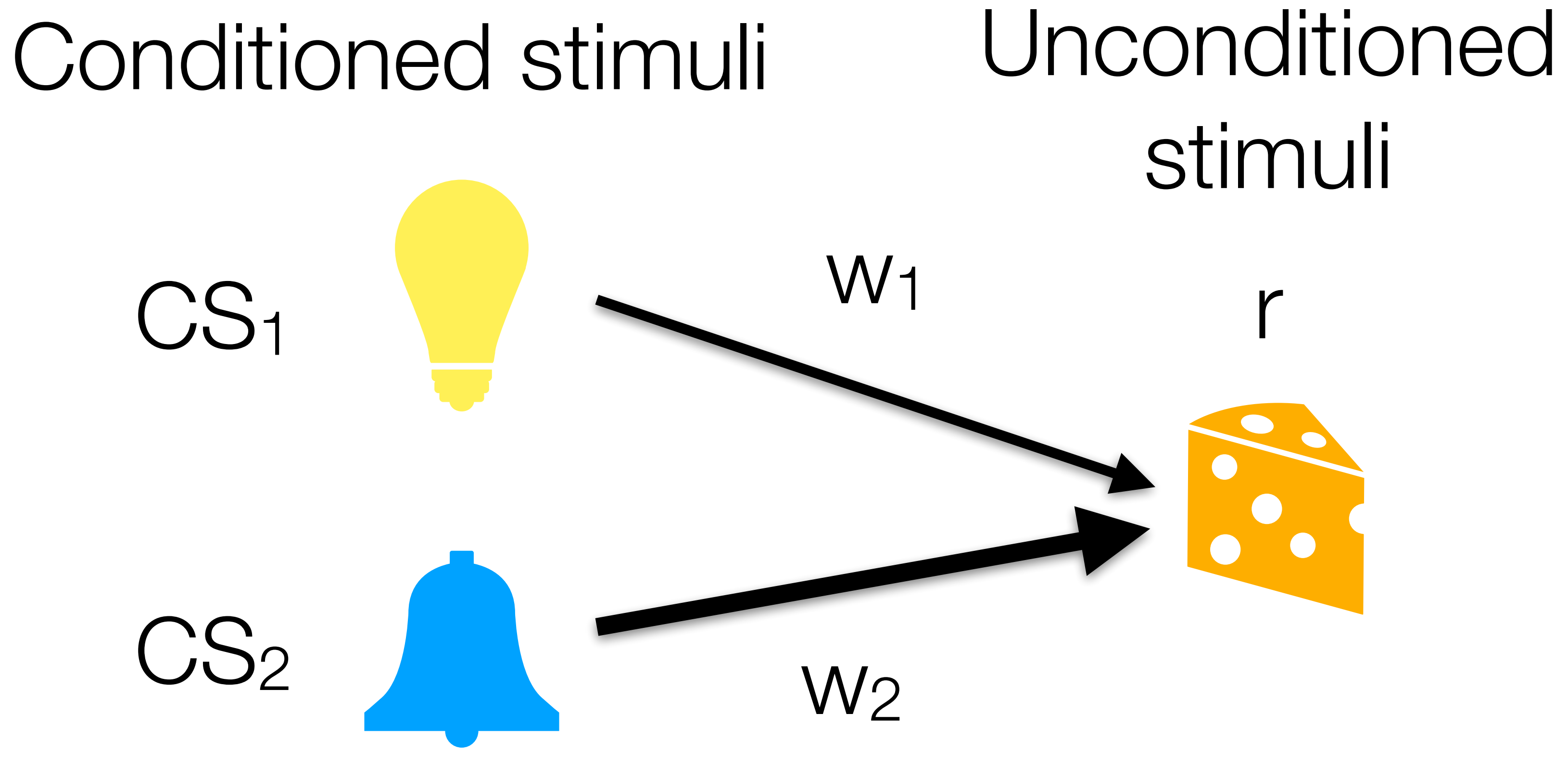
CS i on trial t

Associative strength

Weight update

$$w_i \leftarrow w_i + \eta(r_t - \hat{r}_t)$$

Rescorla-Wagner



Rescorla-Wagner model

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

Reward prediction

$$\hat{r}_t = \sum_i \mathrm{CS}_i^t w_i$$

Reward expectation

CS i on trial t

Associative strength

Weight update

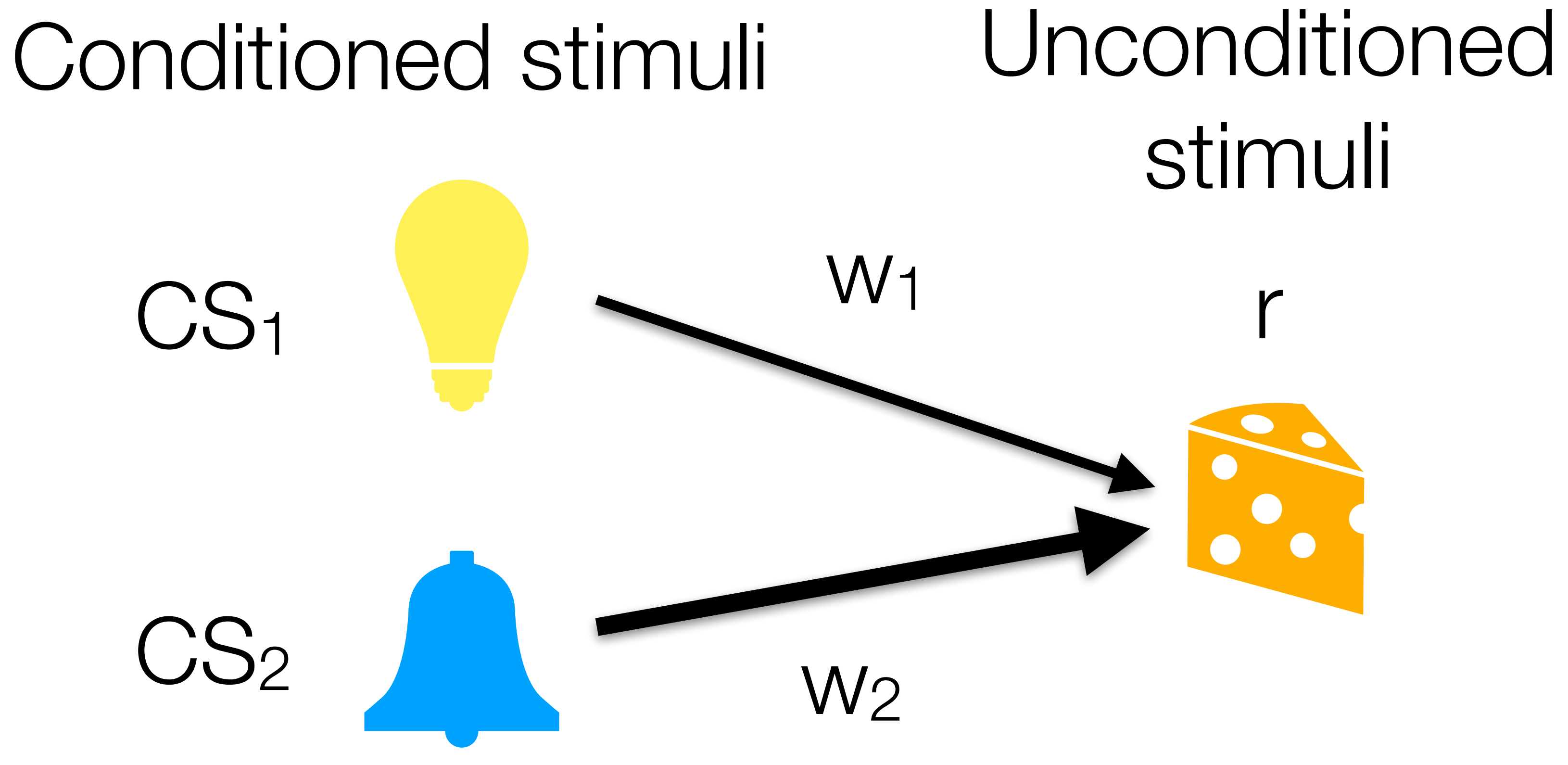
$$w_i \leftarrow w_i + \eta(r_t - \hat{r}_t)$$

Learning rate

Observed outcome

Predicted outcome

Rescorla-Wagner



Rescorla-Wagner model

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

Reward prediction

$$\hat{r}_t = \sum_i \mathrm{CS}_i^t w_i$$

Reward expectation

CS i on trial t

Associative strength

Weight update

$$w_i \leftarrow w_i + \eta(\underbrace{r_t - \hat{r}_t}_{\delta})$$

Learning rate

Observed outcome

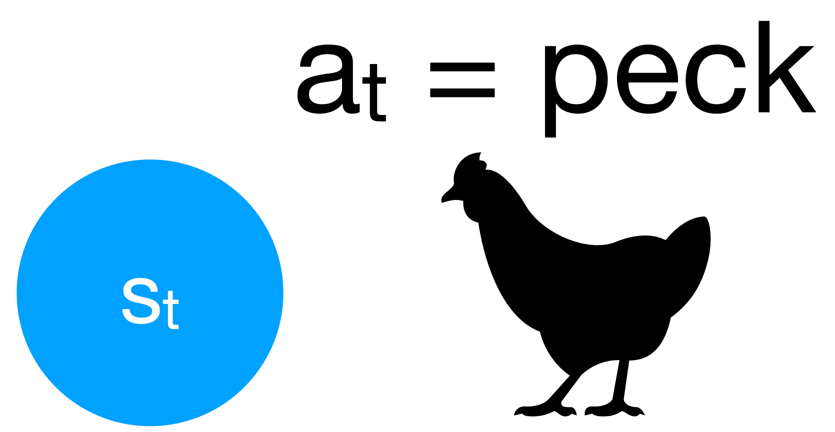
Predicted outcome

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

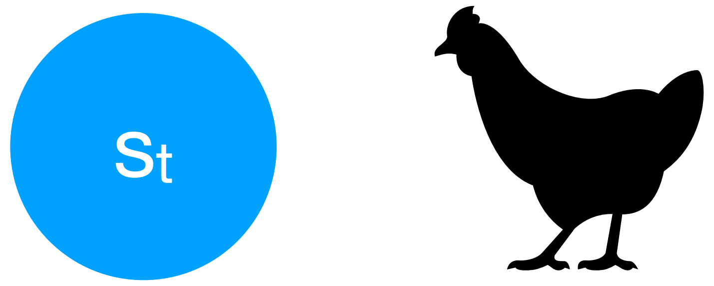


From Rescorla-Wagner to Q-learning

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

Q-learning
(Watkins, 1989)

a_t = peck



From Rescorla-Wagner to Q-learning

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

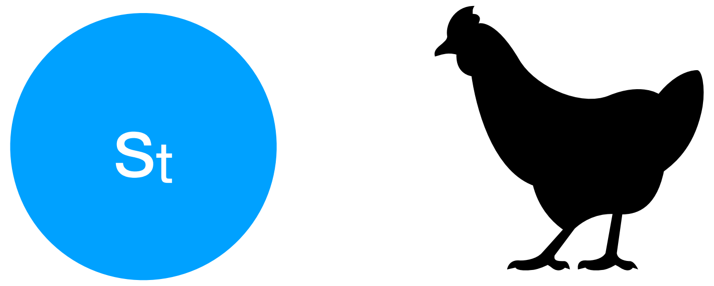
Q-learning
(Watkins, 1989)

$$\hat{r}_t = \sum_i \mathrm{CS}_i^t w_i$$

Reward estimate

$$Q(s_t, a_t)$$

a_t = peck



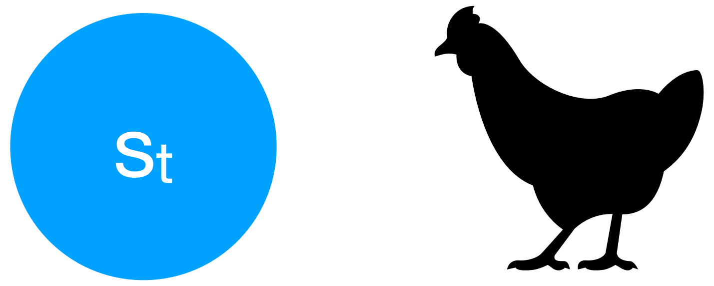
From Rescorla-Wagner to Q-learning

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

Q-learning
(Watkins, 1989)

Rescorla-Wagner model		Q-learning
$\hat{r}_t = \sum_i \mathrm{CS}_i^t w_i$	Reward estimate	$Q(s_t, a_t)$
$w_i \leftarrow w_i + \eta(r_t - \hat{r}_t)$	Prediction error learning	$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \eta[r - Q(s_t, a_t)]$

a_t = peck



From Rescorla-Wagner to Q-learning

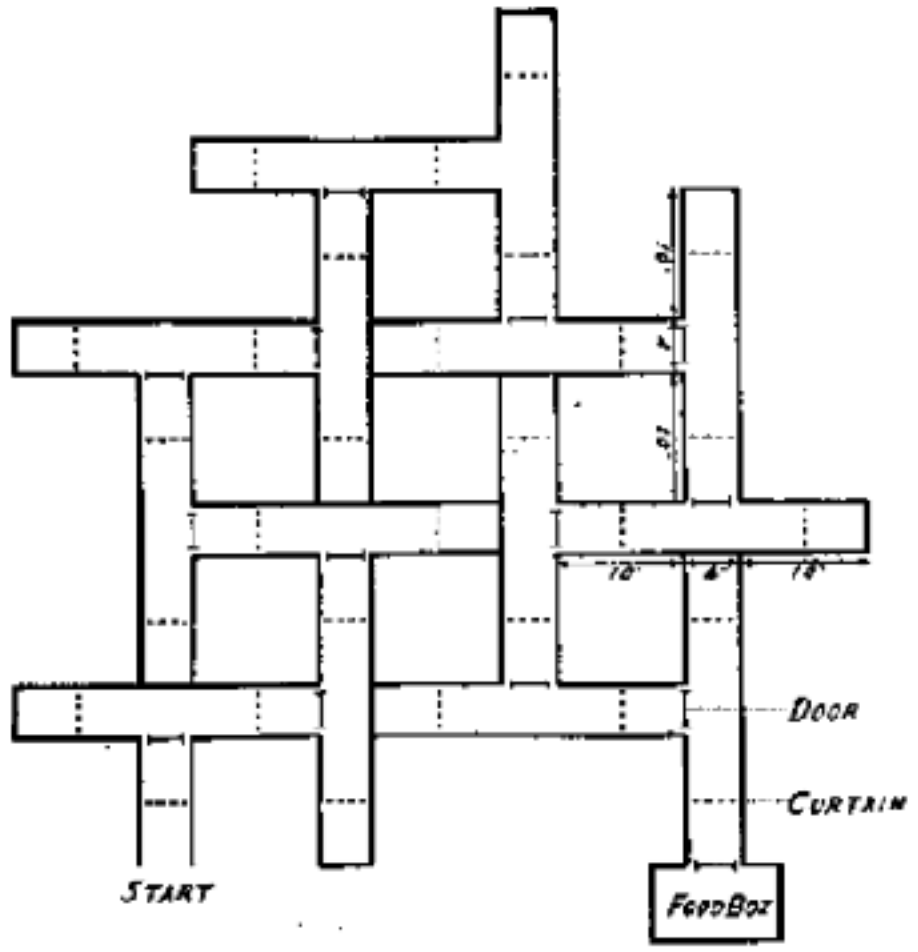
Rescorla-Wagner model (Bush & Mosteller, 1951; Rescorla & Wagner, 1972)		Q-learning (Watkins, 1989)
$\hat{r}_t = \sum_i \mathrm{CS}_i^t w_i$	Reward estimate	$Q(s_t, a_t)$
$w_i \leftarrow w_i + \eta(r_t - \hat{r}_t)$	Prediction error learning	$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \eta[r - Q(s_t, a_t)]$
?	Behavioral policy	$\pi(a_t \mid s_t) \propto \exp(Q(s_t, a_t)/\tau)$

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



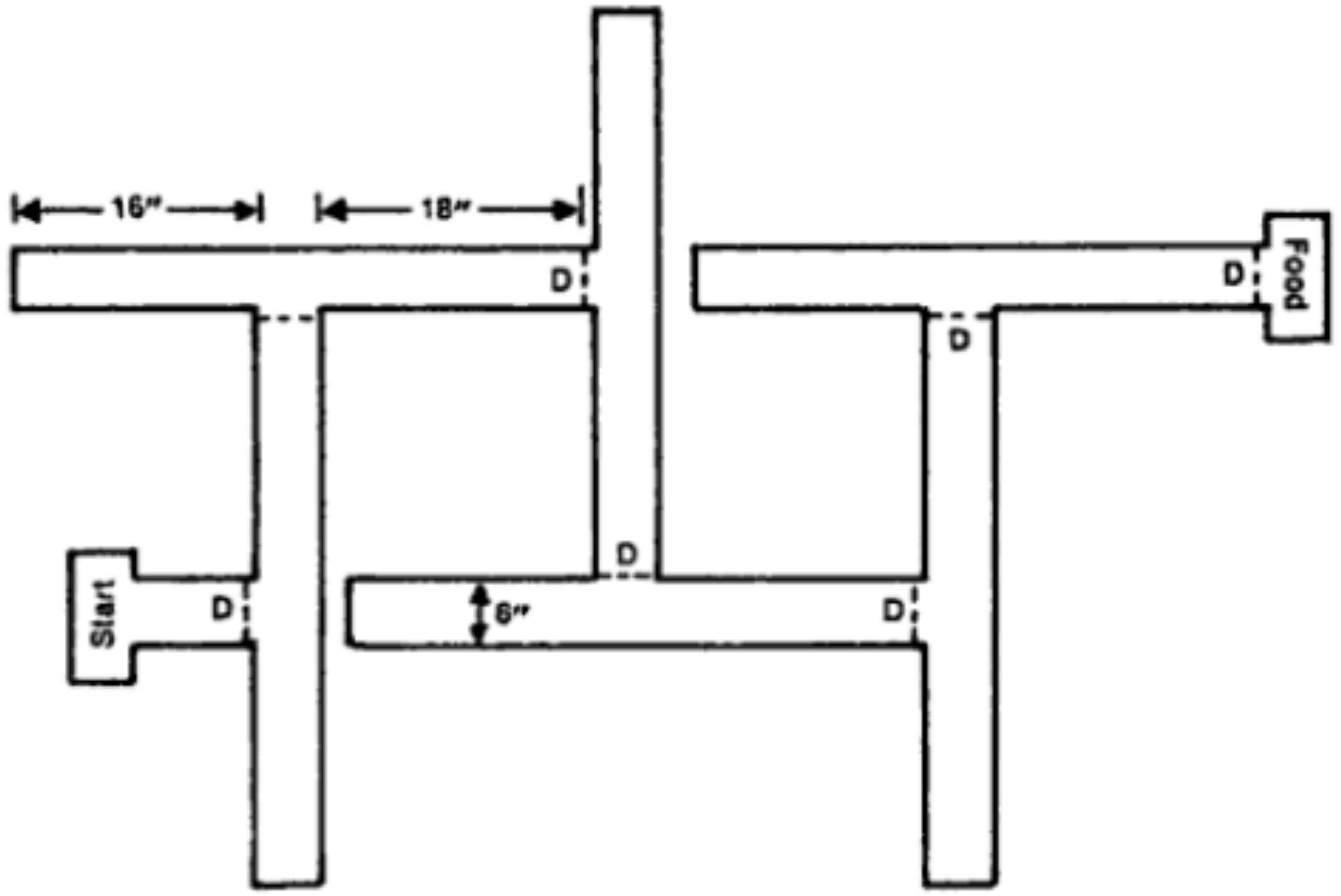
Plan of maze
14 Unit T-Alley Maze

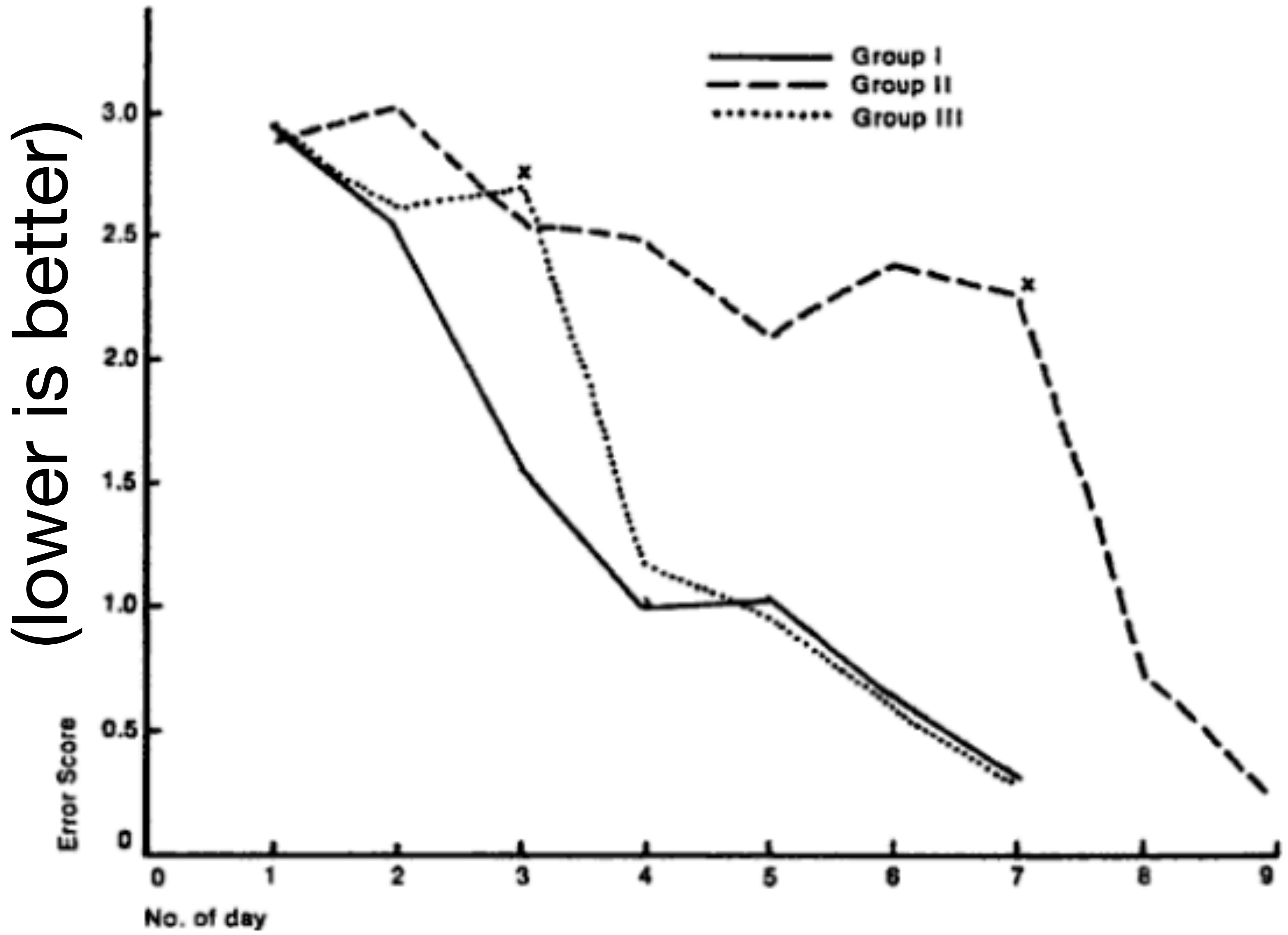
FIG. 1

(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

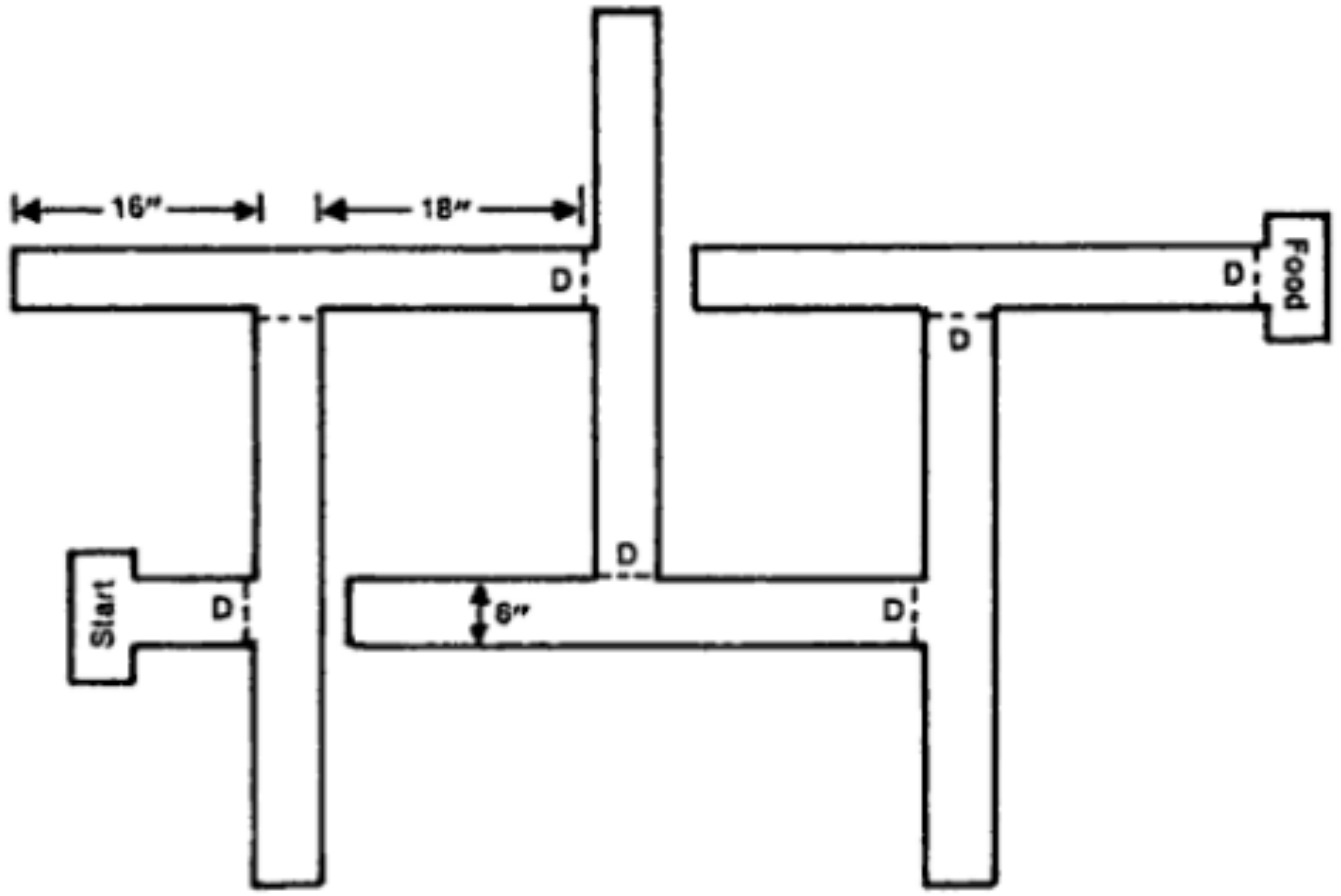
- Blodgett (1929) Maze navigation task
  - **Group 1** [Control]: one trial a day with food in the goal box at the end
  - **Group 2** [Late food] No food in the maze for days 1-6, then food provided at the end on day 7
  - **Group 3** [Early food] … food added on day 3
- Learning curves dropped dramatically 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

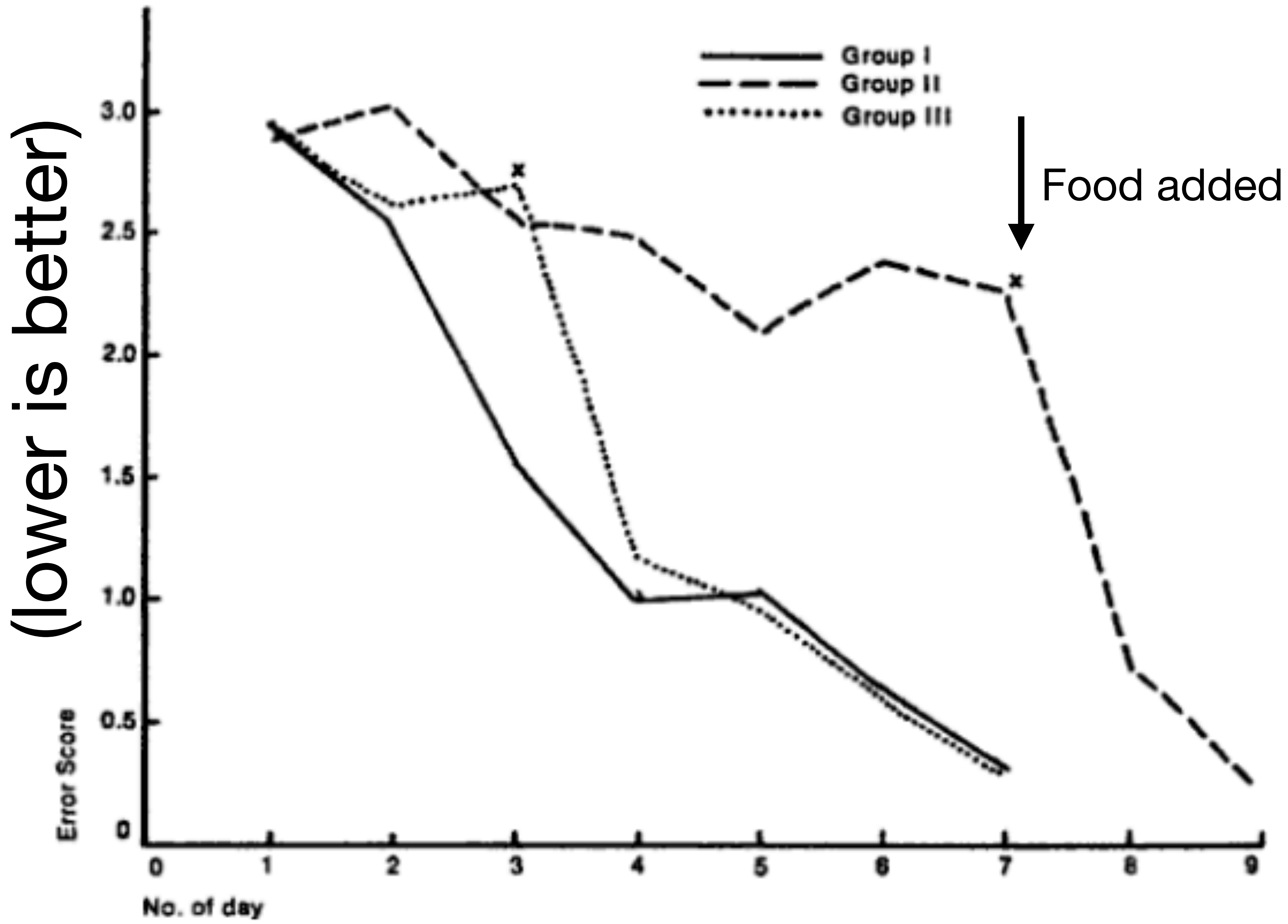




Latent Learning

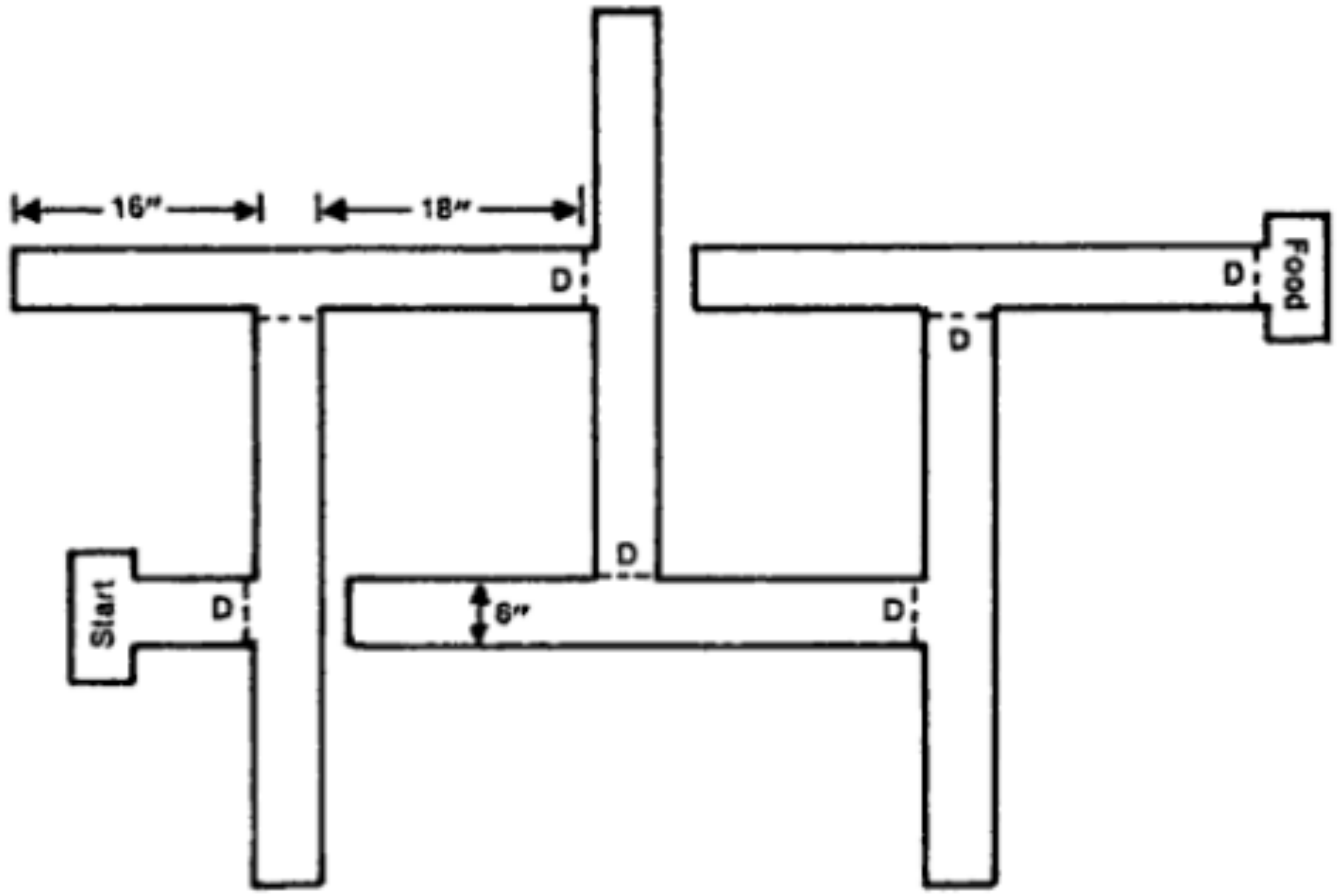
- Blodgett (1929) Maze navigation task
    - **Group 1** [Control]: one trial a day with food in the goal box at the end
    - **Group 2** [Late food] No food in the maze for days 1-6, then food provided at the end on day 7
    - **Group 3** [Early food] … food added on day 3
- Learning curves dropped 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

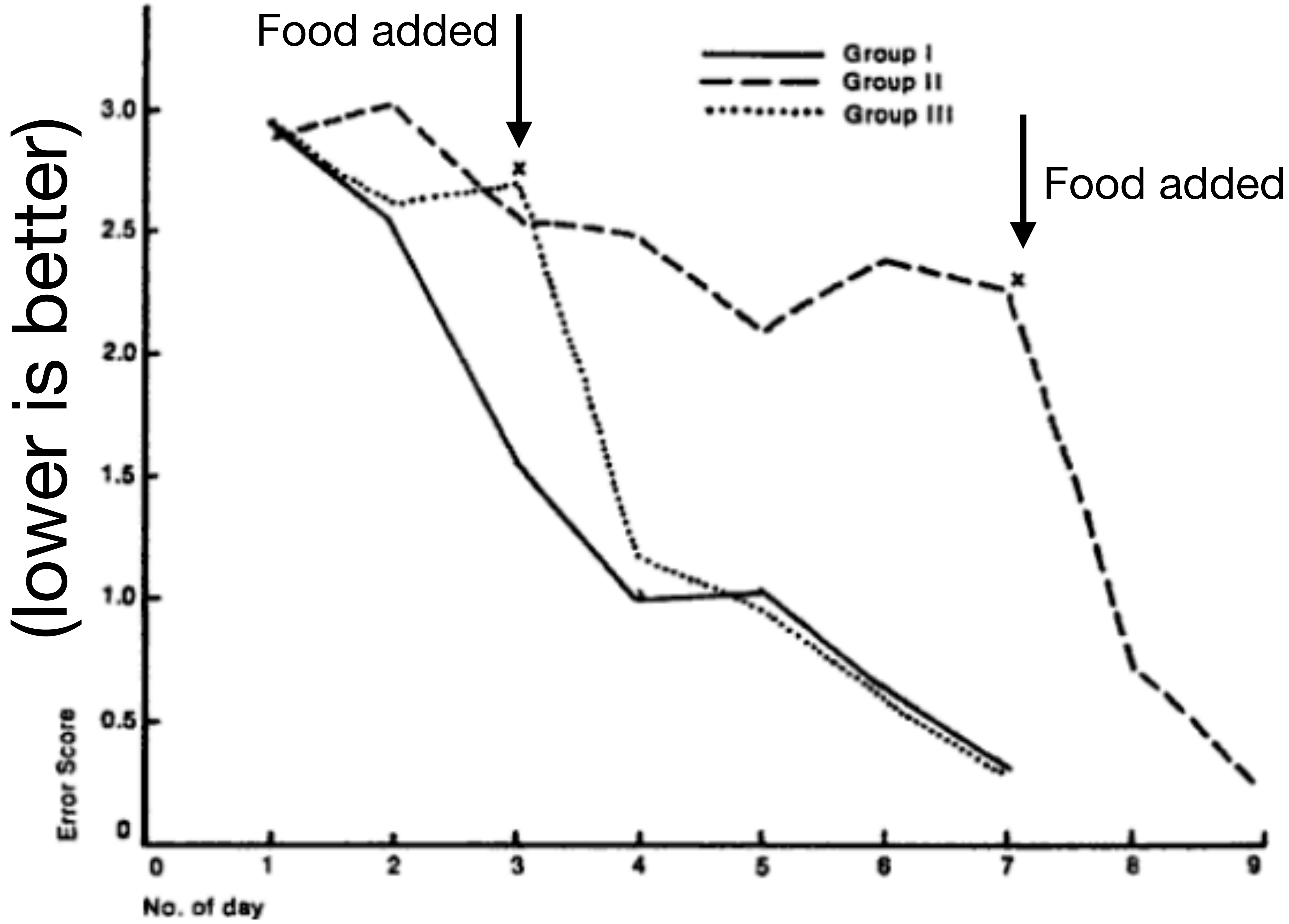




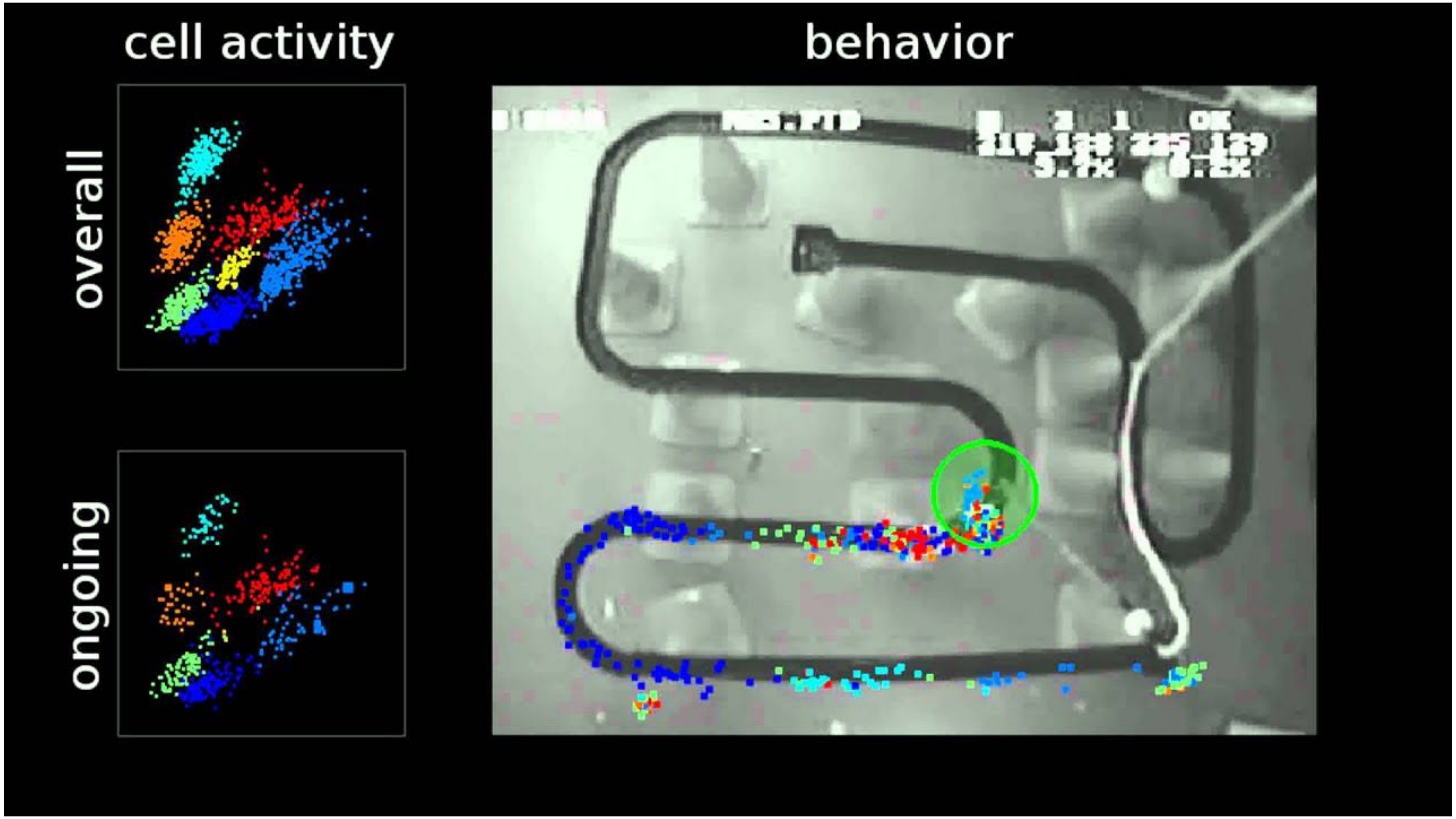
Latent Learning

- 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

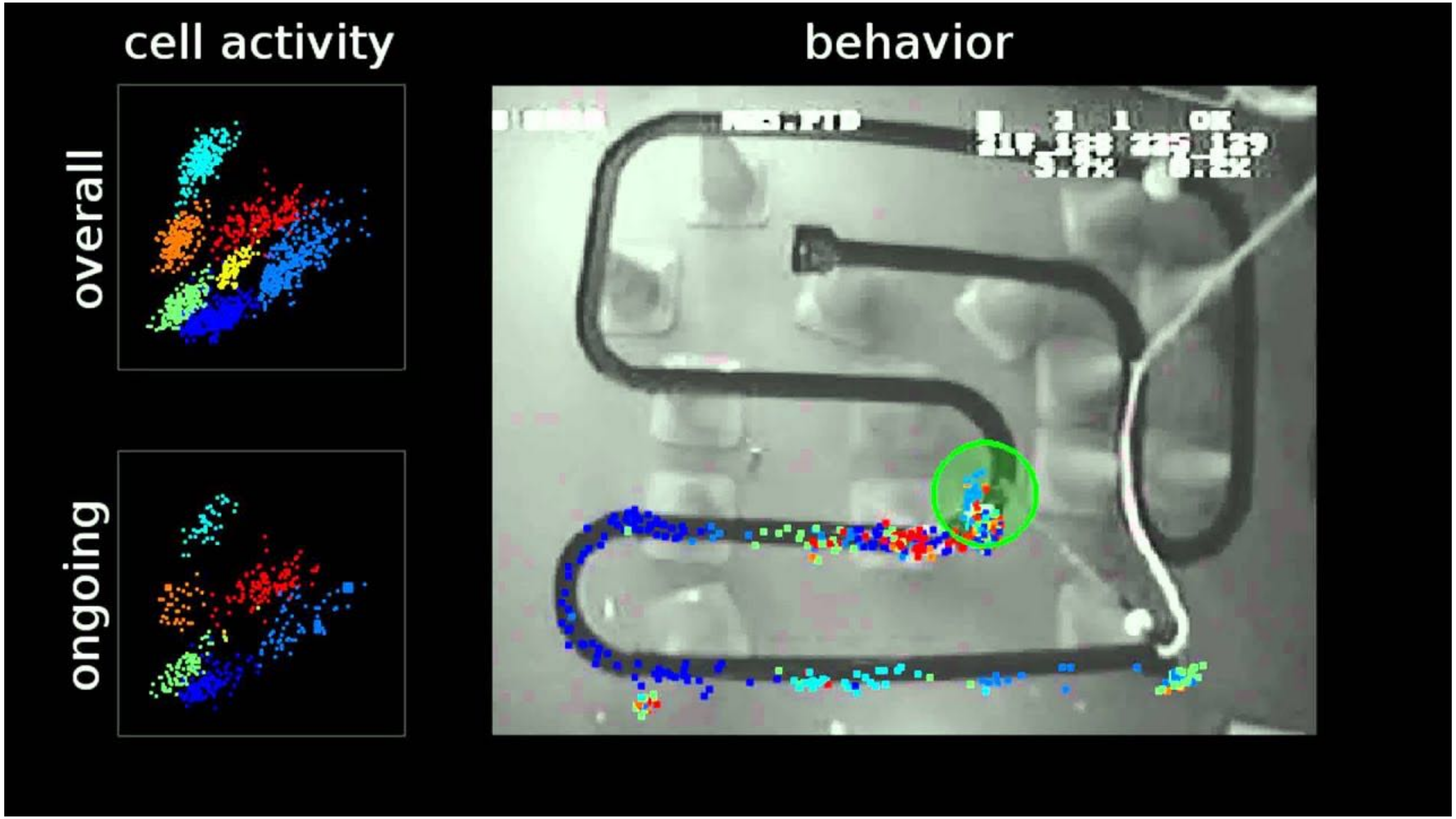


John O'Keefe
Nobel Prize in Physiology or Medicine 2014

Wilson Lab (MIT)

Place cells in the hippocampus represent location in an environment

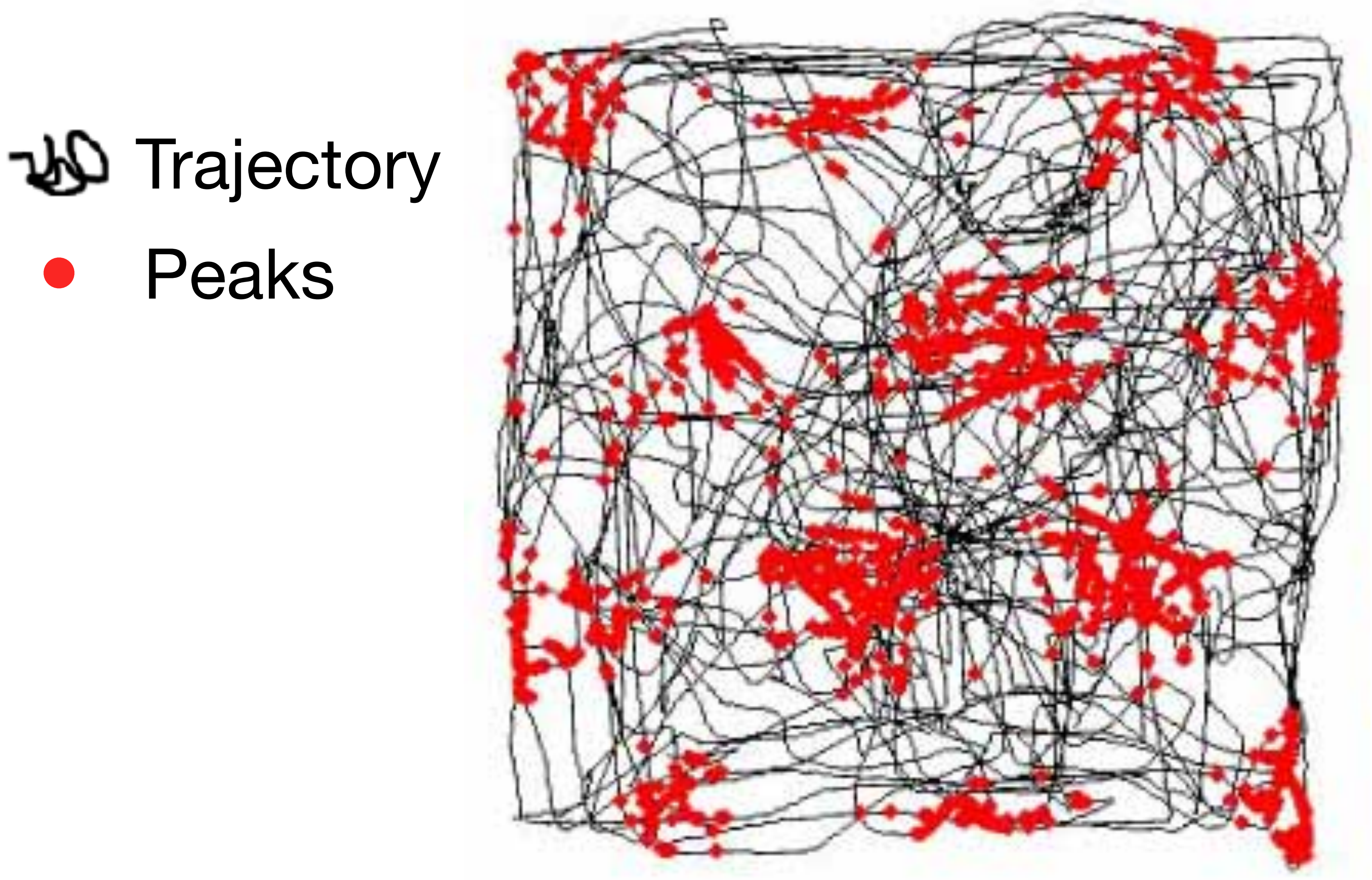
(O'keefe & Nadel 1978)

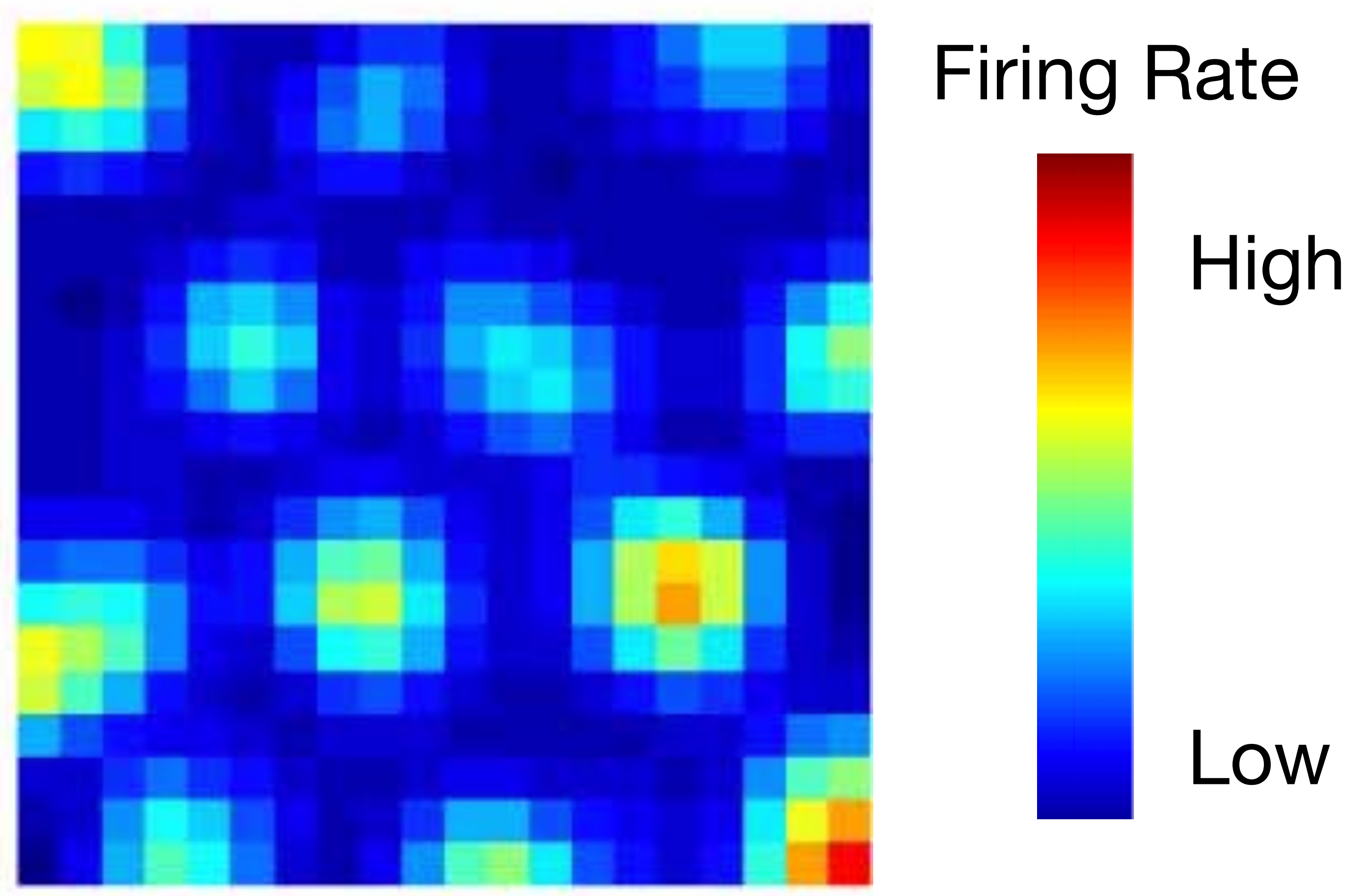


John O'Keefe
Nobel Prize in Physiology or Medicine 2014

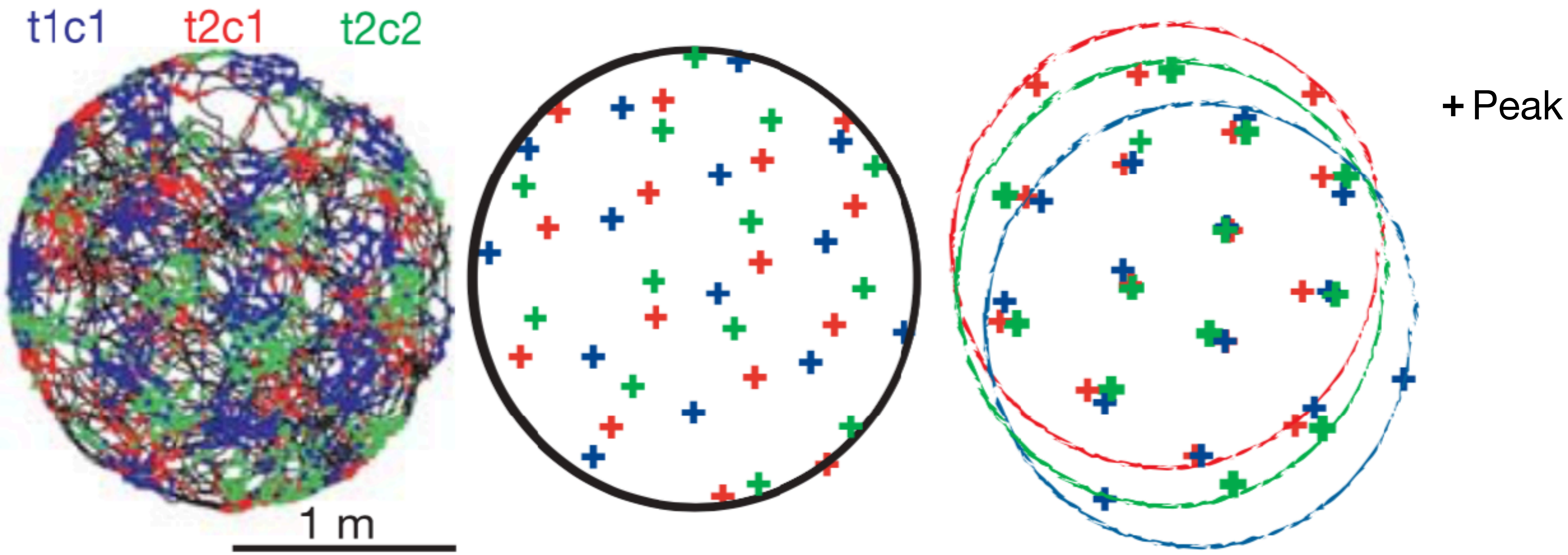
Wilson Lab (MIT)

Grid cells in the Entorhinal Cortex provide a coordinate system



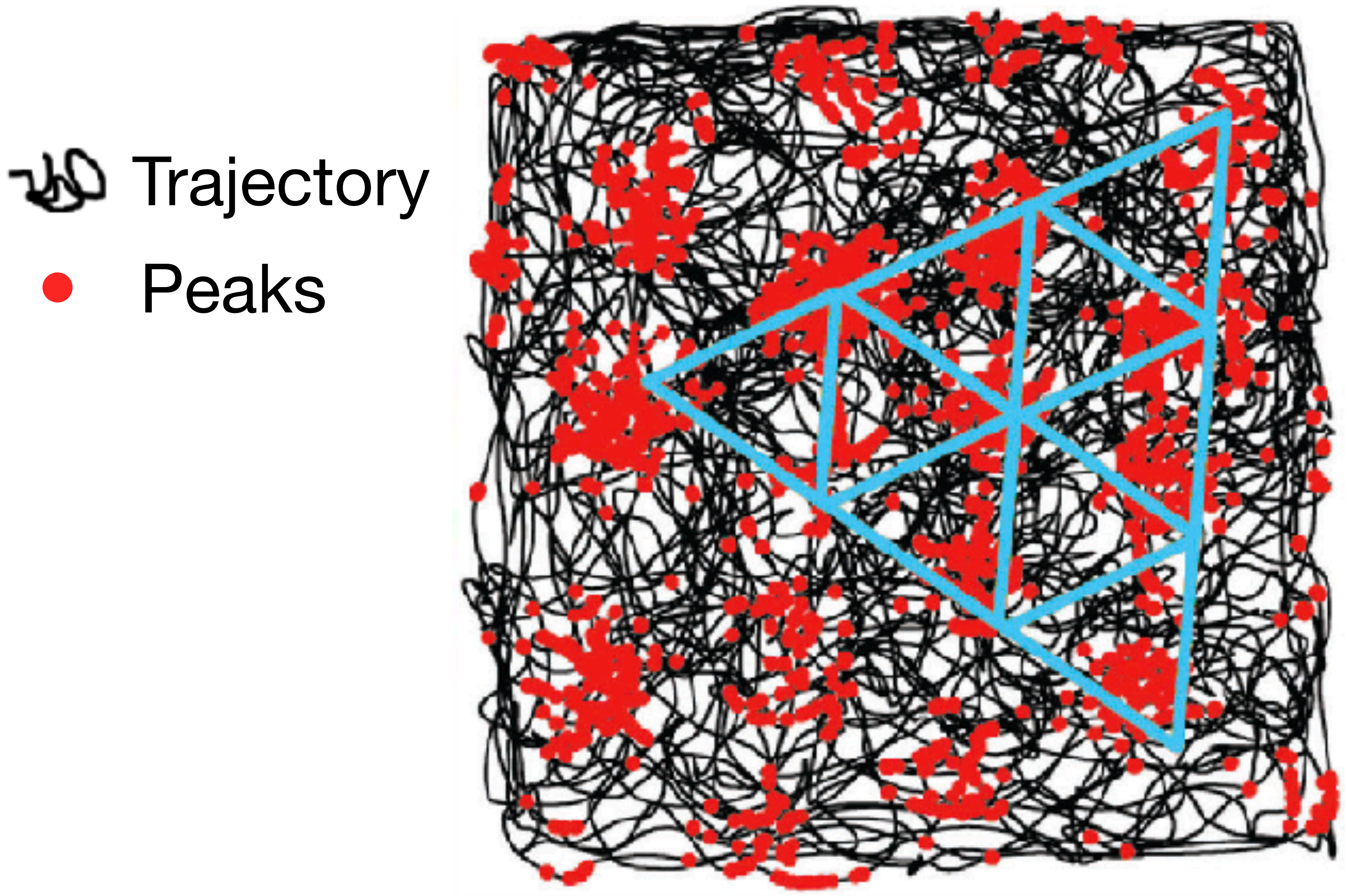


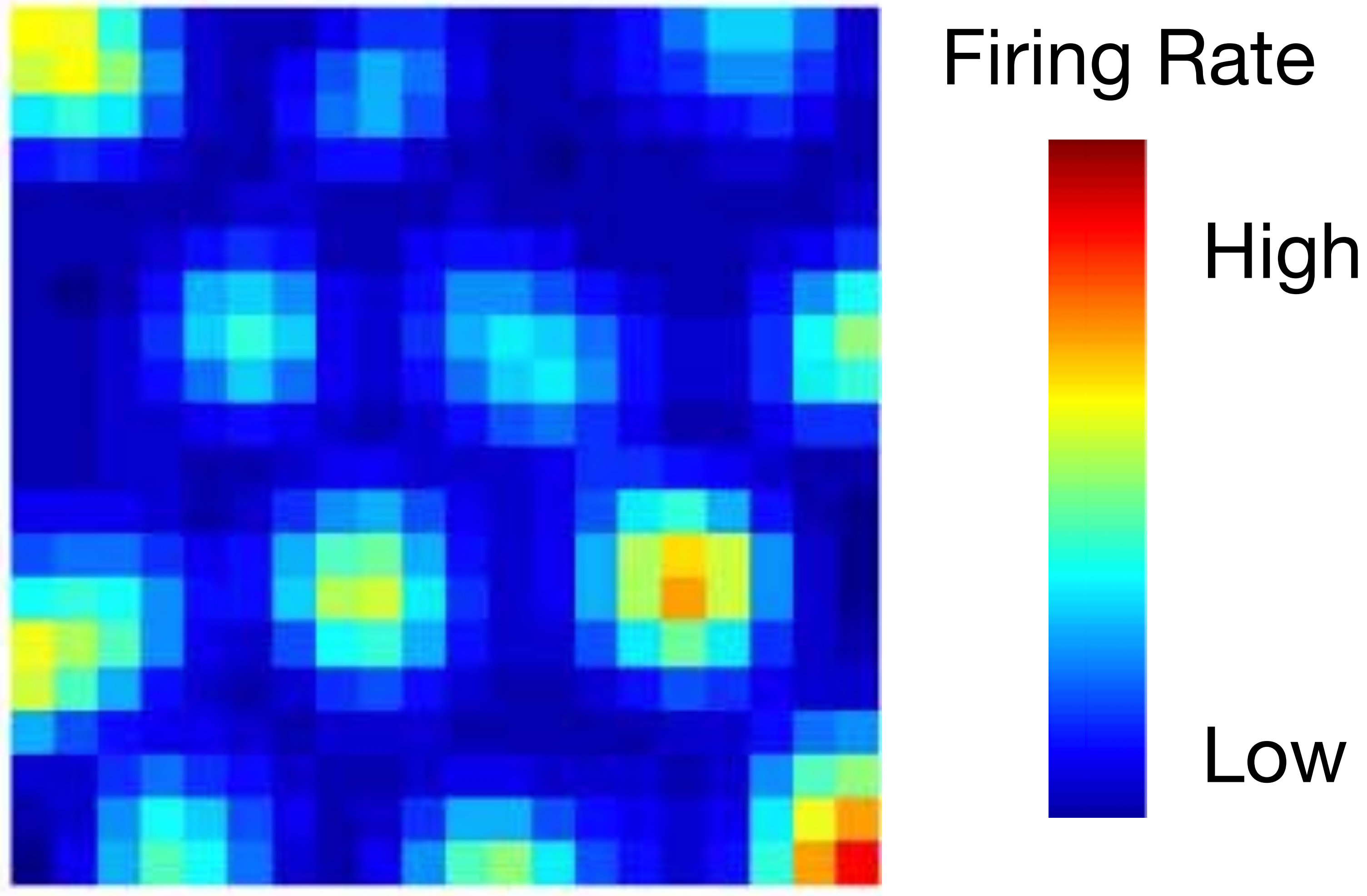
Edvard and Maj-Britt Moser
Nobel Prize in Physiology or Medicine 2014

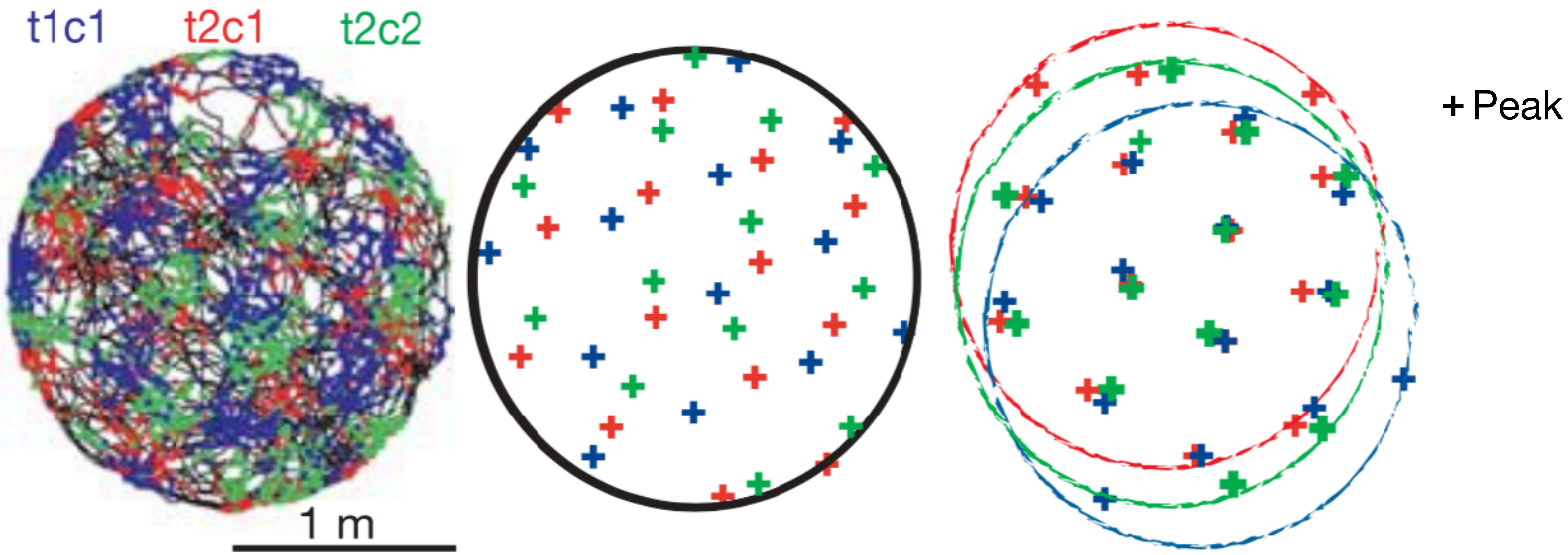


Hafting *et al* (Nature, 2005)

Grid cells in the Entorhinal Cortex provide a coordinate system



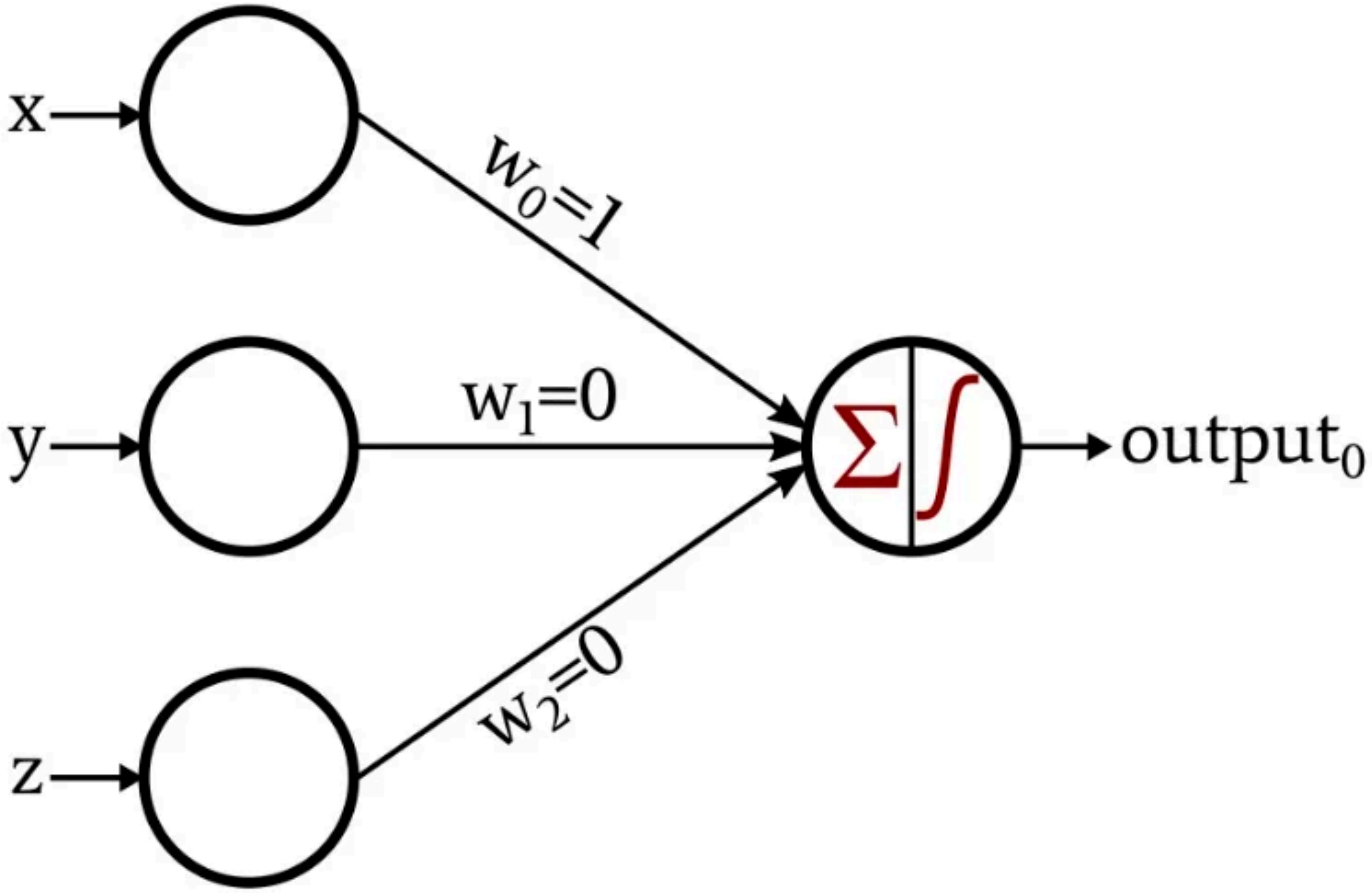




Edvard and Maj-Britt Moser
Nobel Prize in Physiology or Medicine 2014

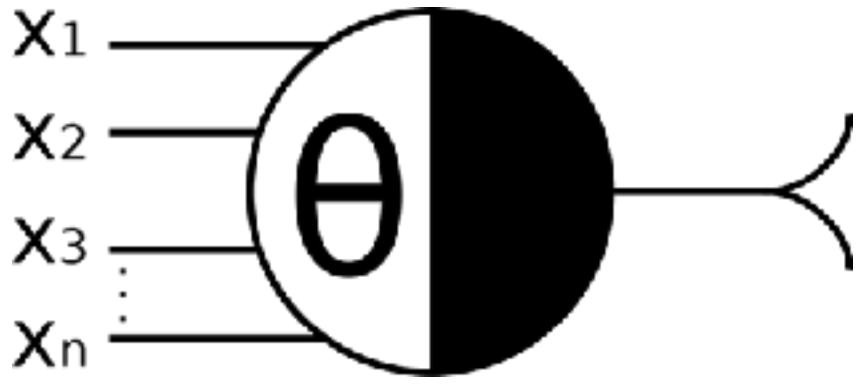
Hafting *et al* (Nature, 2005)

Origins of Artificial Learning



Timeline of early Artificial Neural Networks

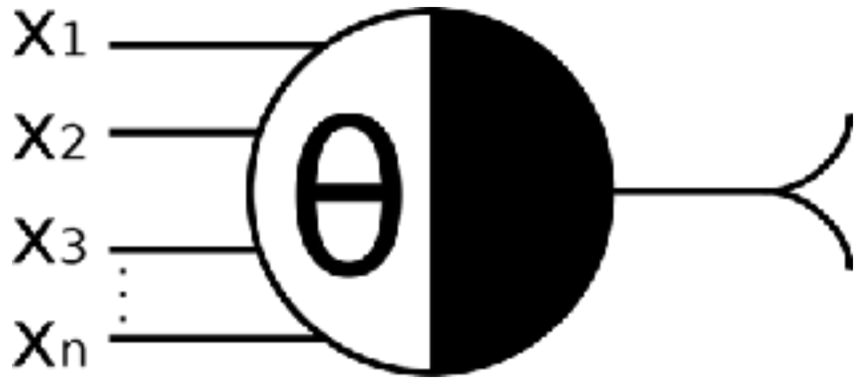
Timeline of early Artificial Neural Networks



McCulloch & Pitts
(1943) neuron

Timeline of early Artificial Neural Networks

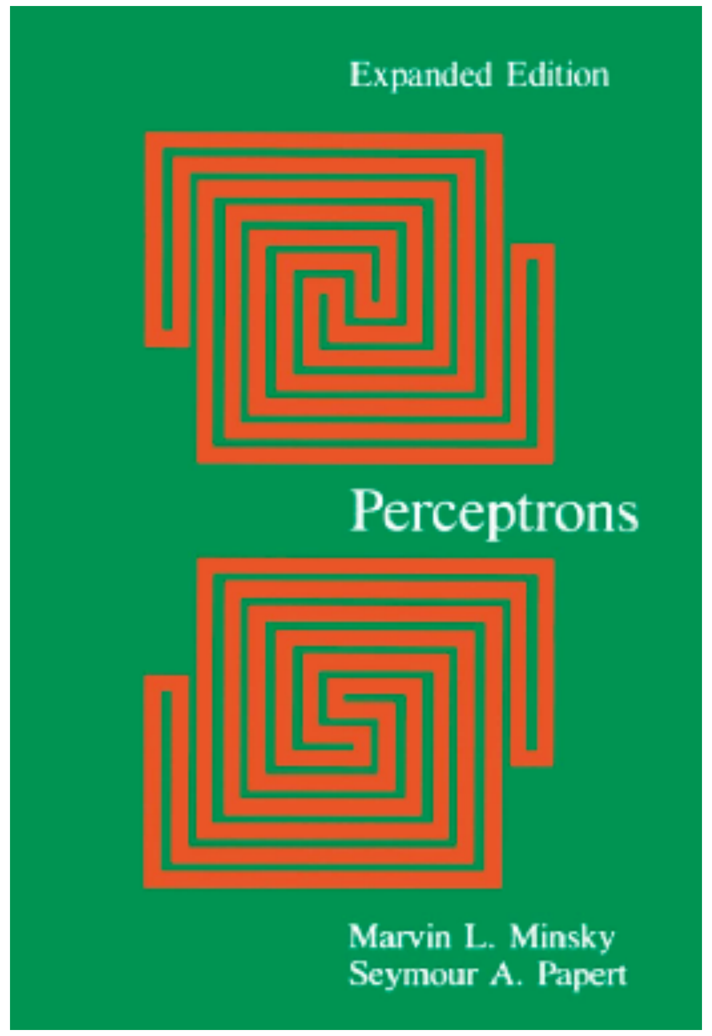
Rosenblatt (1958) Perceptron



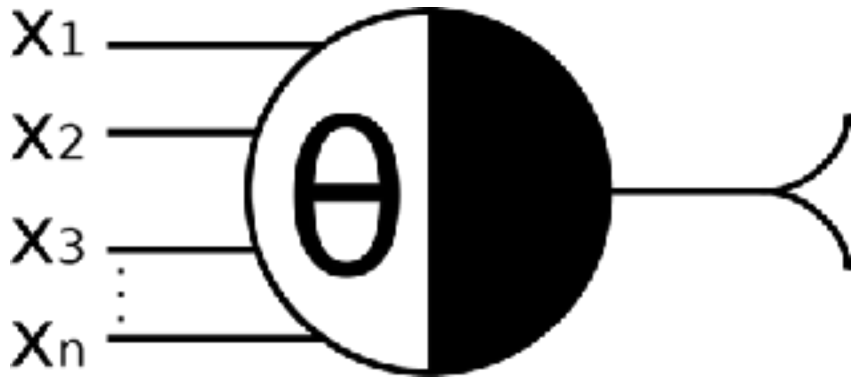
McCulloch & Pitts
(1943) neuron

Timeline of early Artificial Neural Networks

Minsky & Parpert (1969)



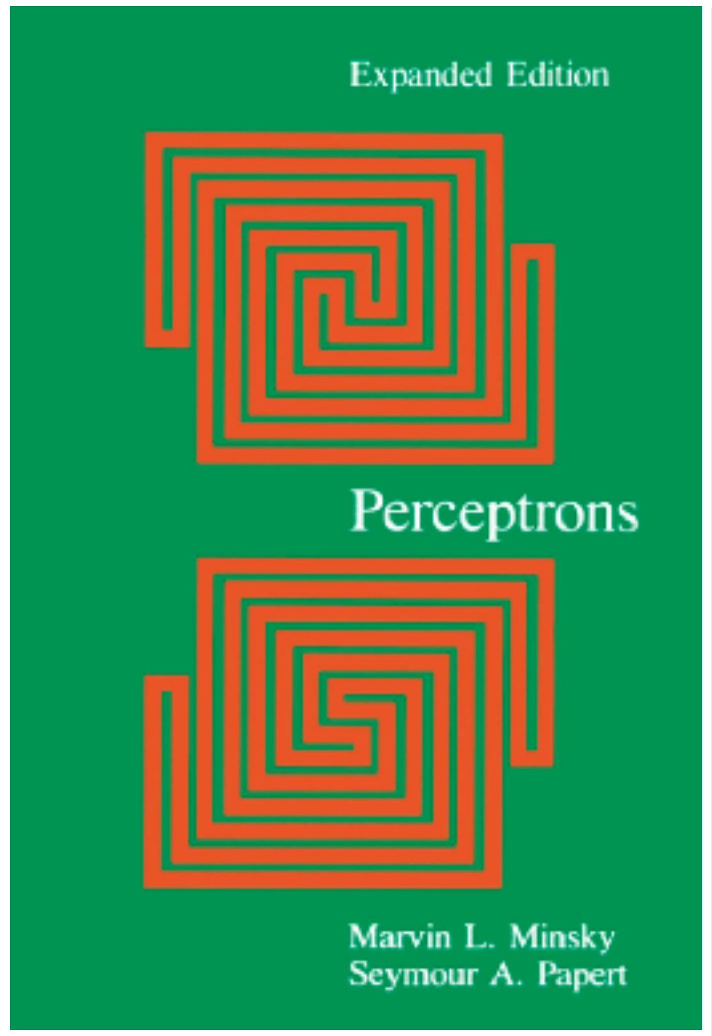
Rosenblatt (1958) Perceptron



McCulloch & Pitts
(1943) neuron

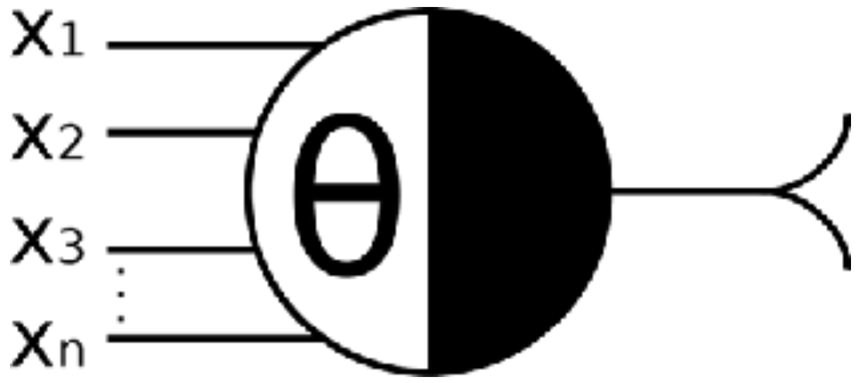
Timeline of early Artificial Neural Networks

Minsky & Parpert (1969)



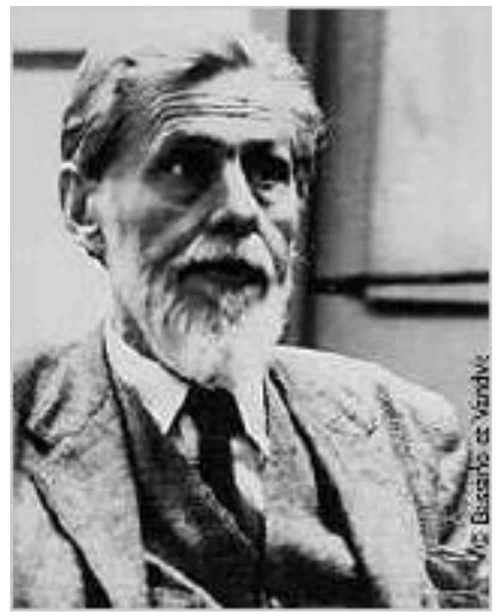
Rosenblatt (1958) Perceptron

AI Winter



McCulloch & Pitts
(1943) neuron

McCulloch & Pitts (1943)



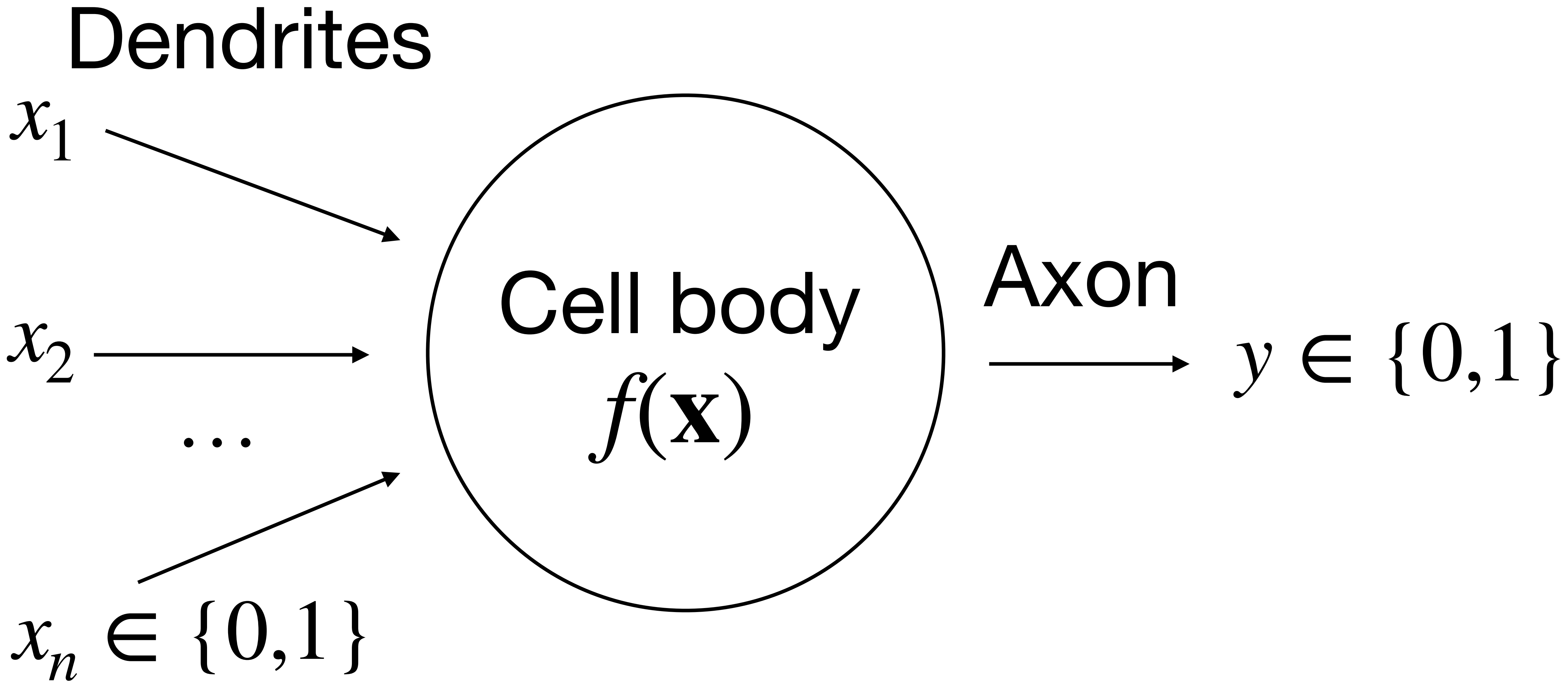
Warren McCulloch

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

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

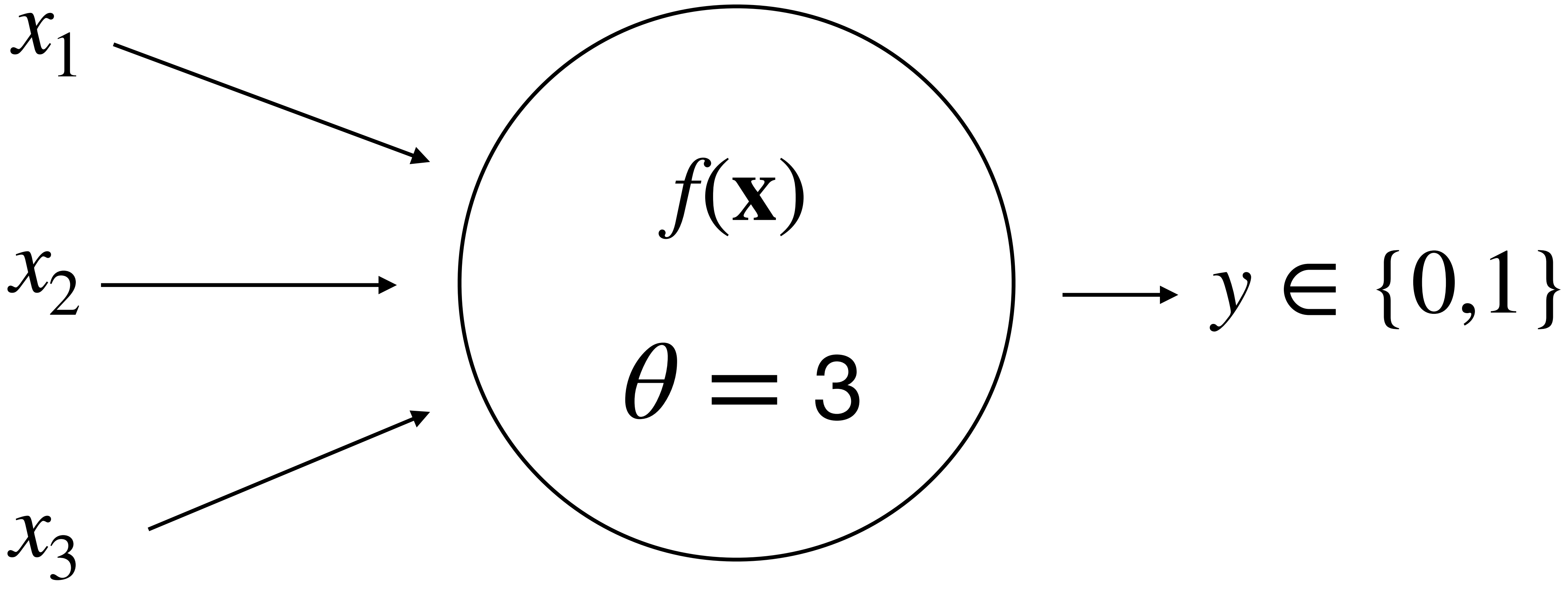
- The axon produces the output



McCulloch & Pitts (1943)

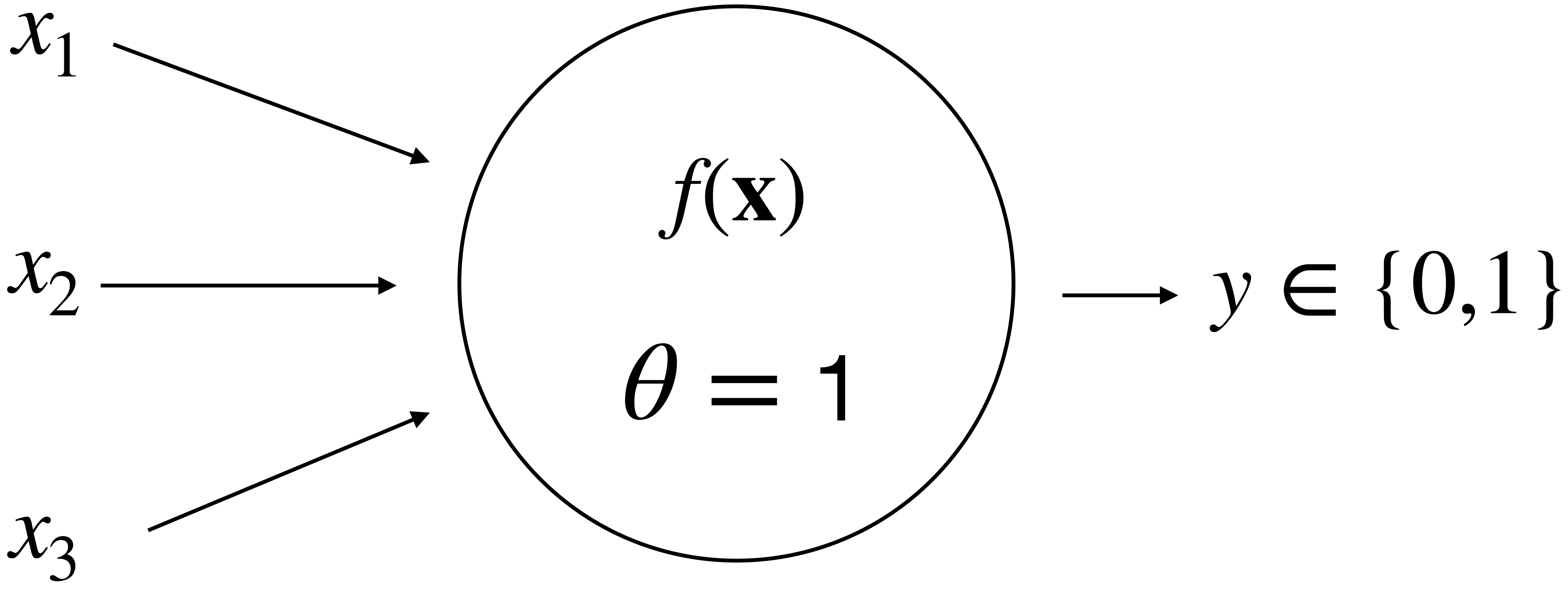
$$f(\mathbf{x}) = \begin{cases} 1 & \text{if } \sum x_i \geq \theta \\ 0 & \text{else} \end{cases}$$

AND function



All inputs need to be on for the neuron to fire

OR function

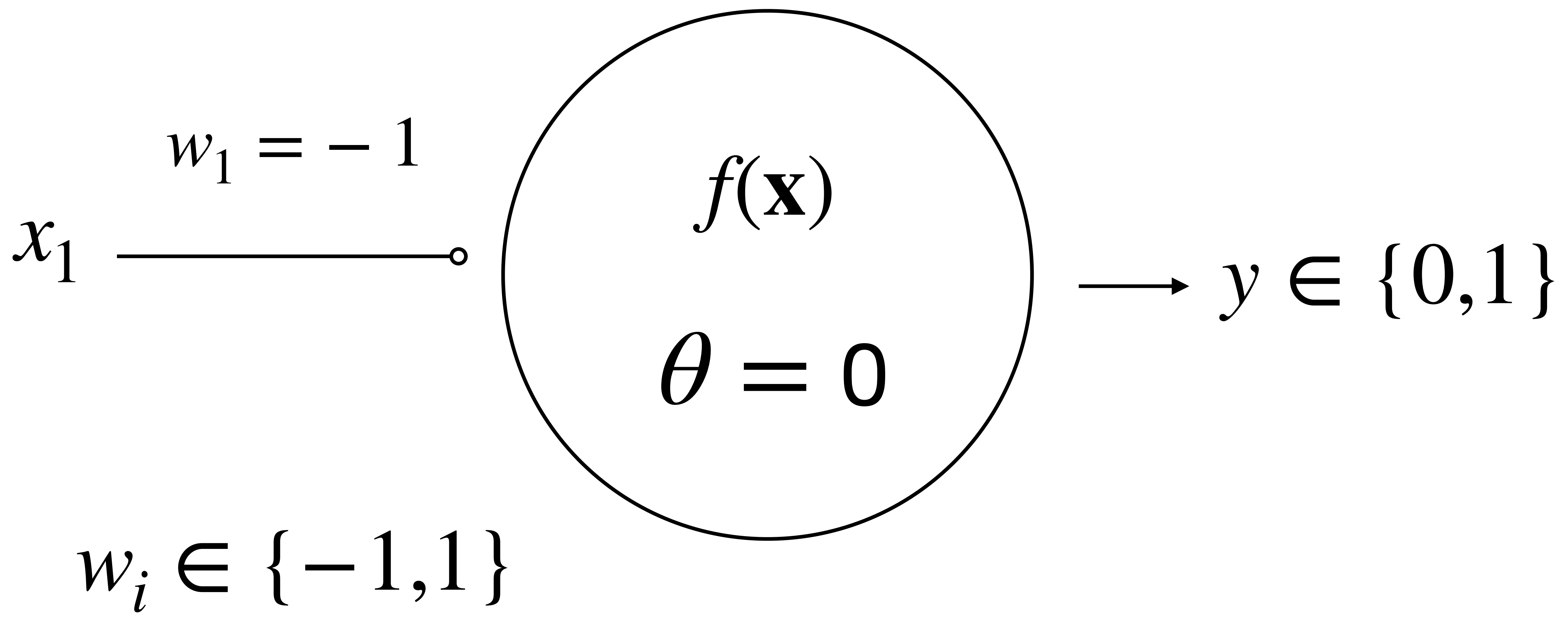


Neuron fires if any input is on

McCulloch & Pitts (1943)

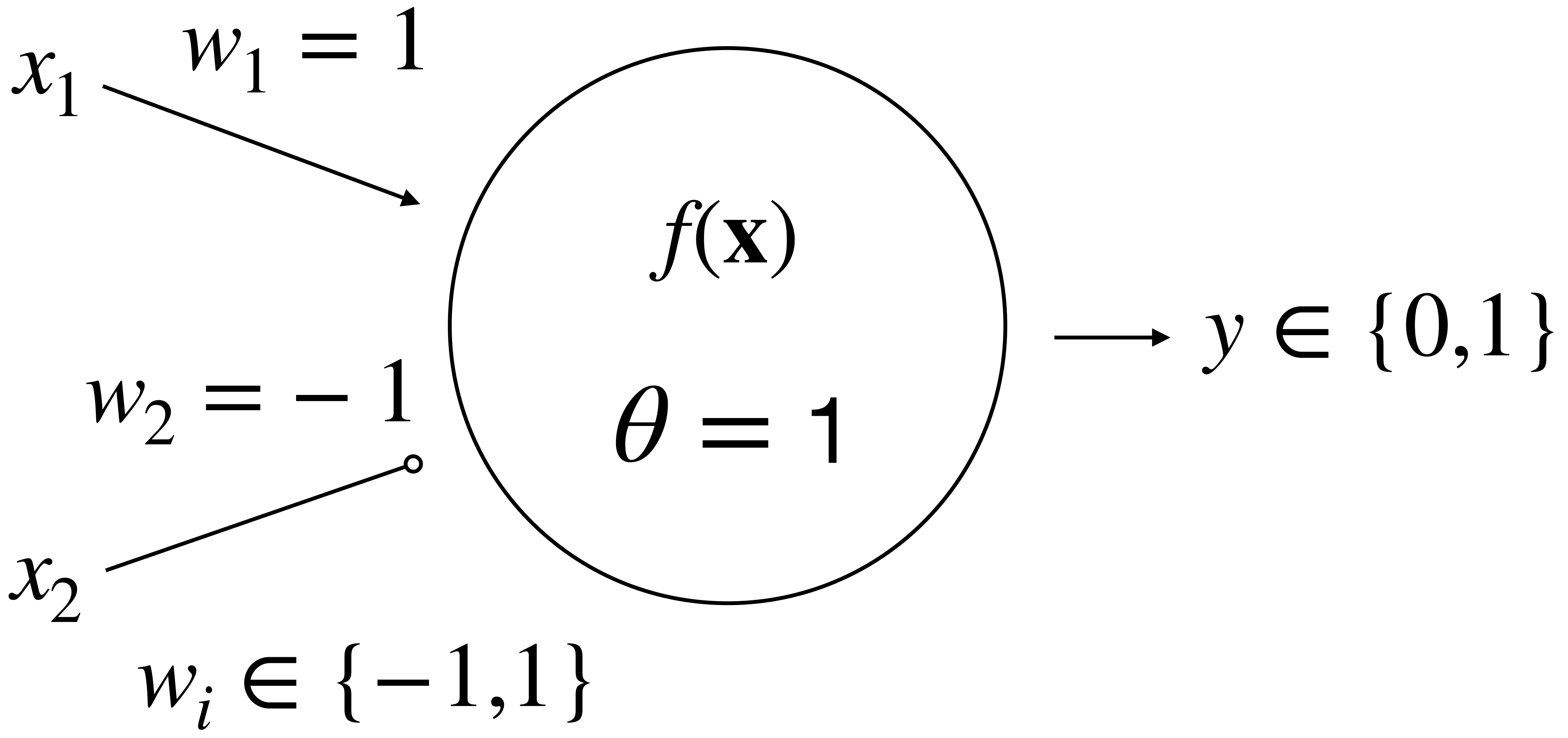
$$f(\mathbf{x}) = \begin{cases} 1 & \text{if} \sum w_i x_i \geq \theta \\ 0 & \text{else} \end{cases}$$

NOT function



Neuron fires if no inputs are on

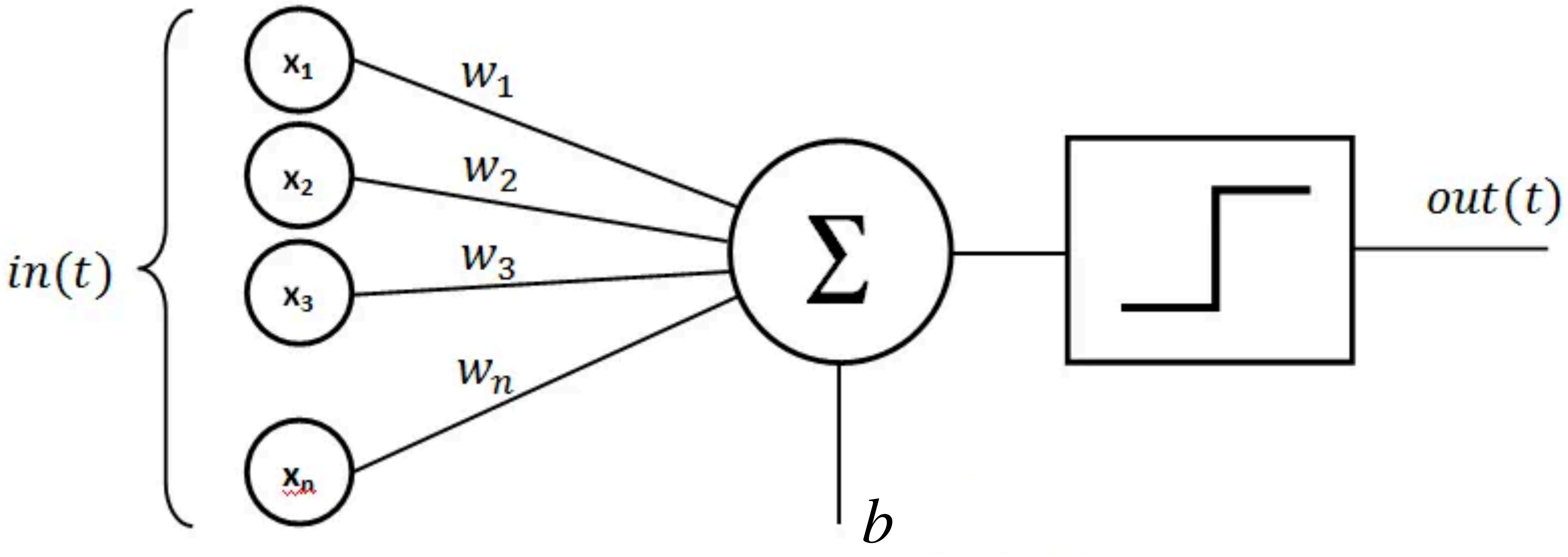
NAND



Neuron fires when x_1 is on AND x_2 not on

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.

      $b = b + error$

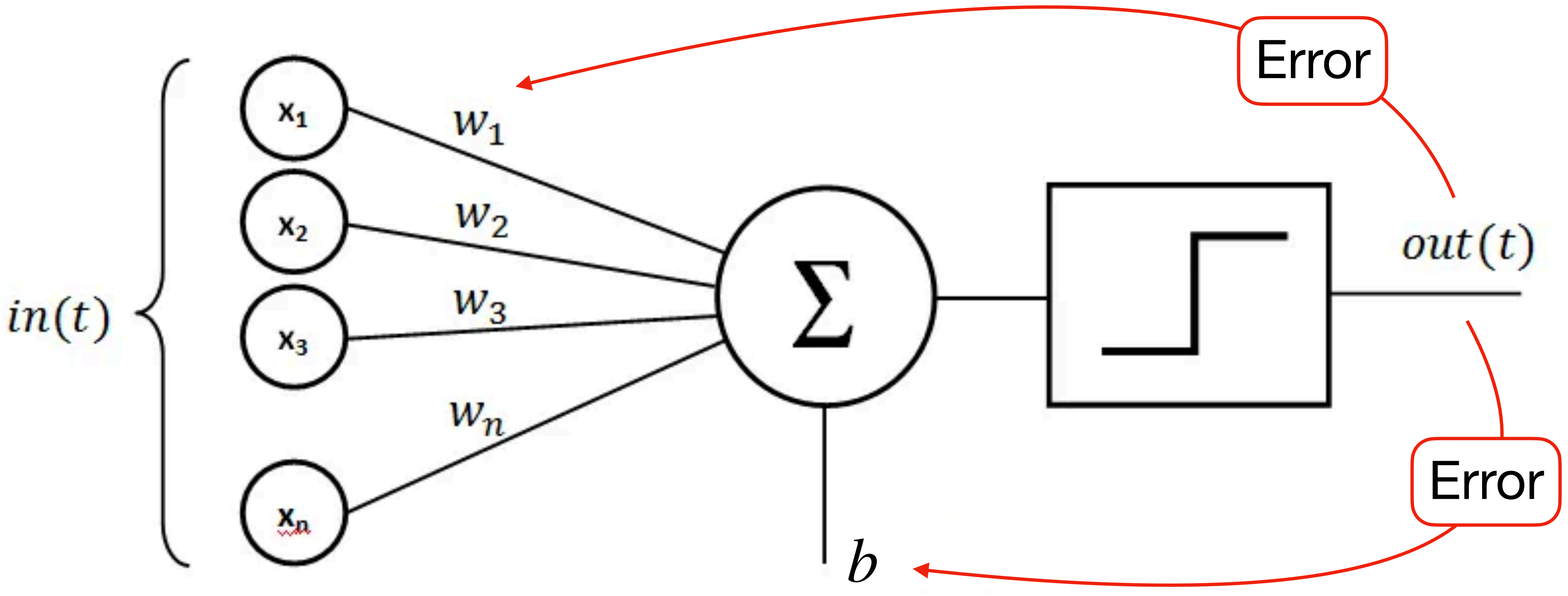
  Test for convergence

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

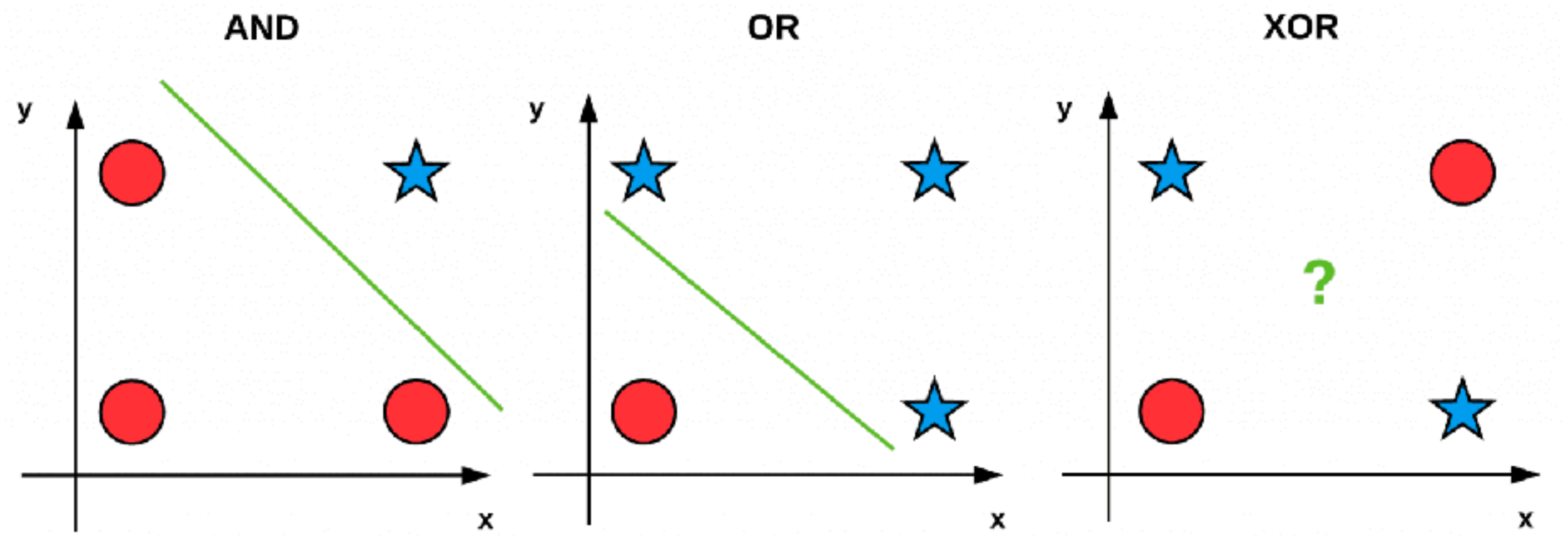
Output: Set of weights $\mathbf{w}$ and bias b for the perceptron.

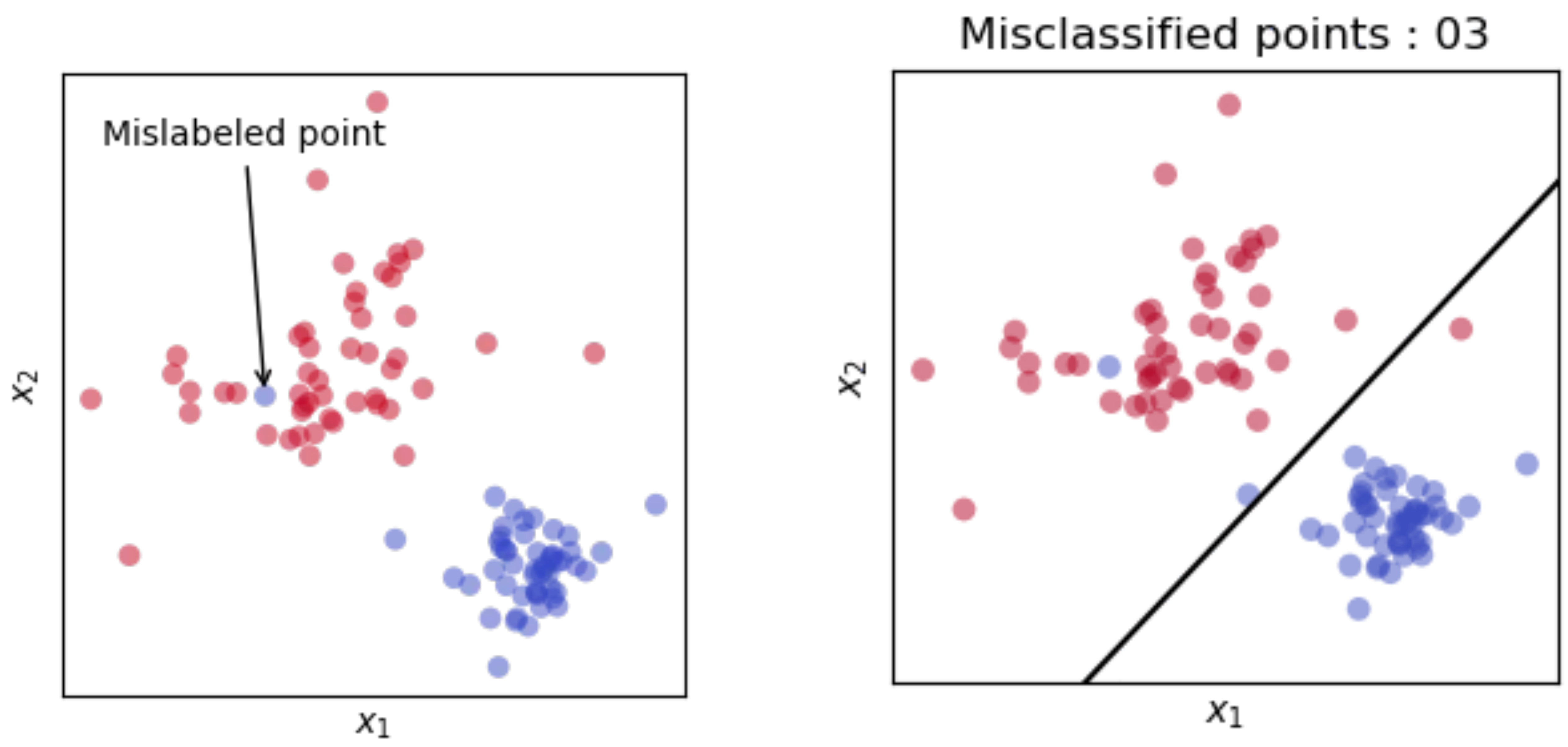
not on the exam

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

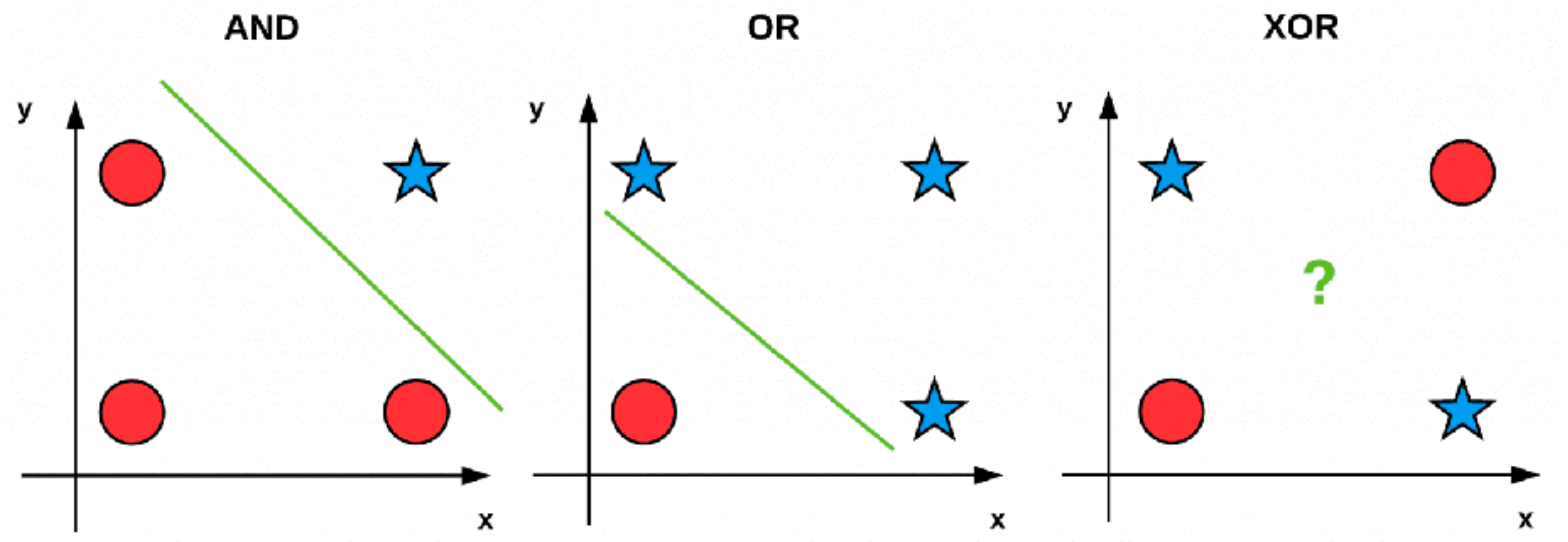


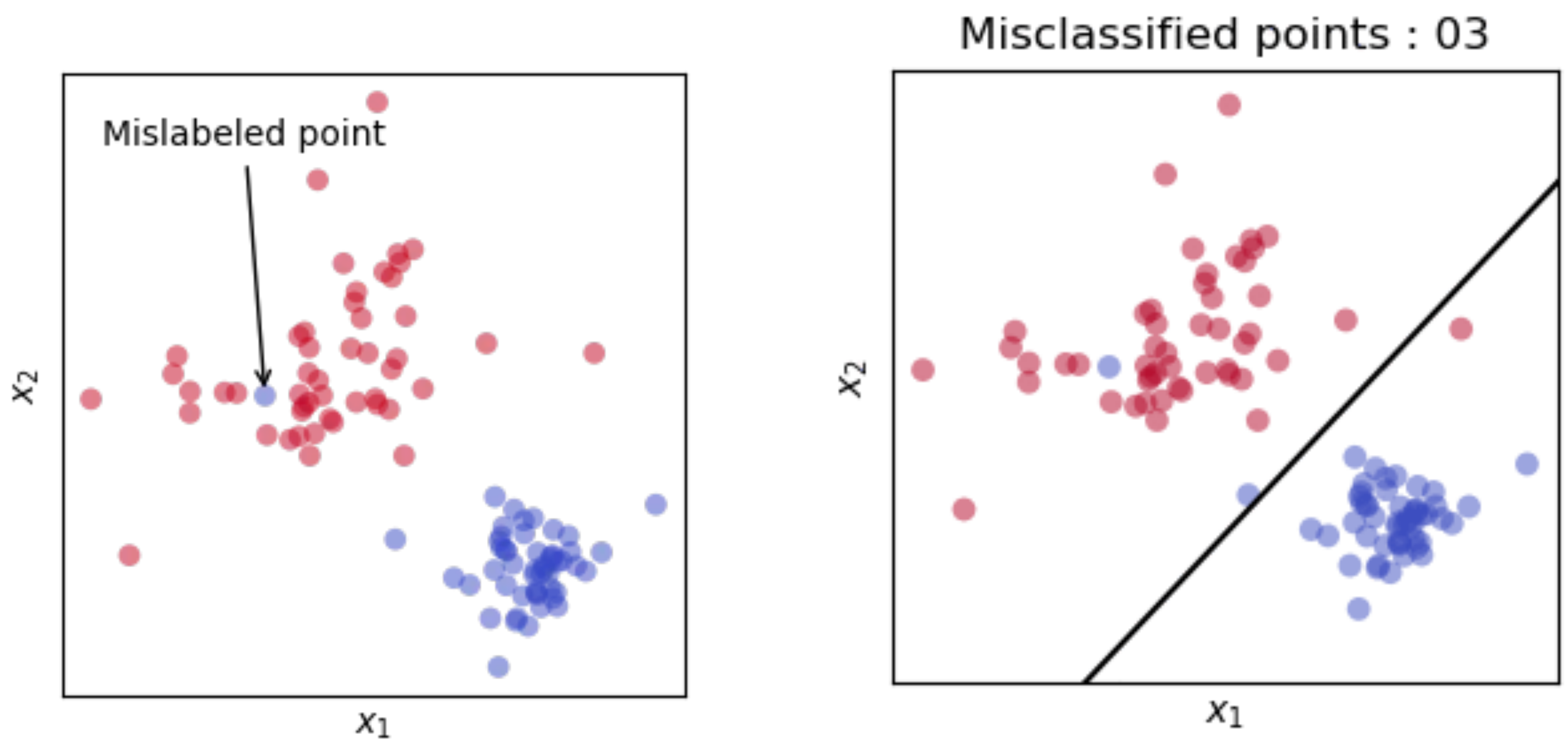


Limitations of linear separability

Adrian Rosebrock

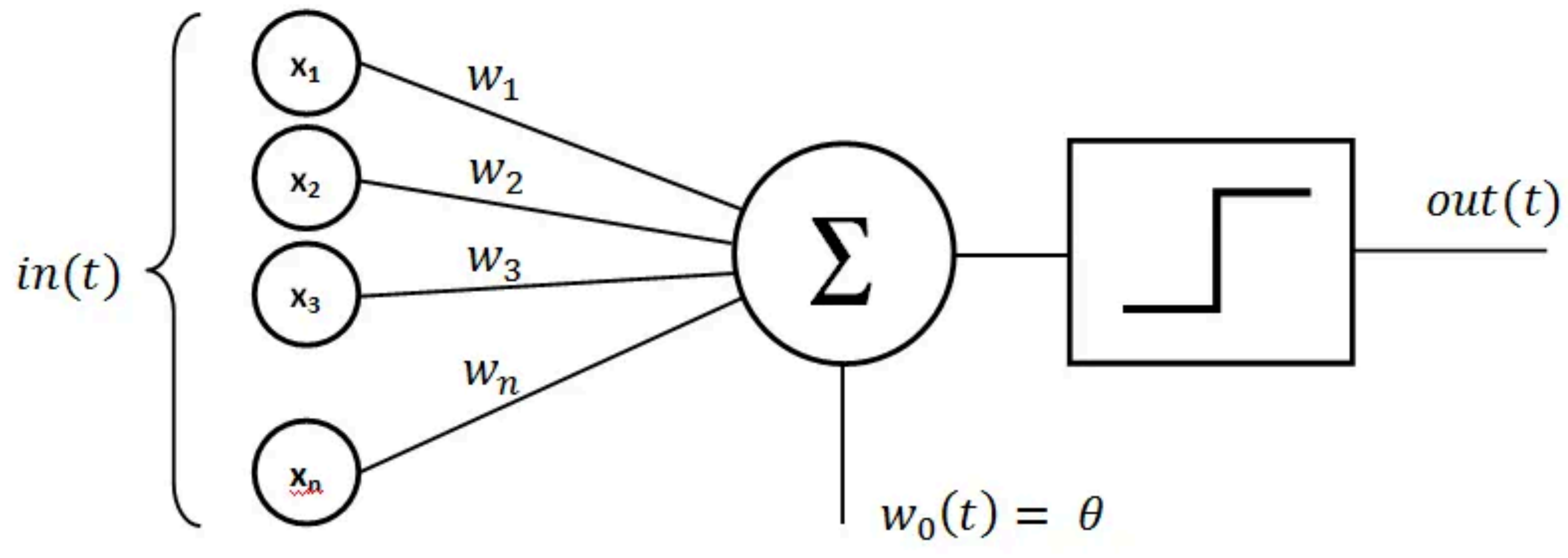
- The perceptron can learn any linearly separable problem
    - But not all problems are lineary separable
- Even a single mislabeled data point in the data will throw the algorithm into chaos
- Enter the **XOR problem** and Minsky & Parpert (1969) critique
    - Argument: because a single neuron is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

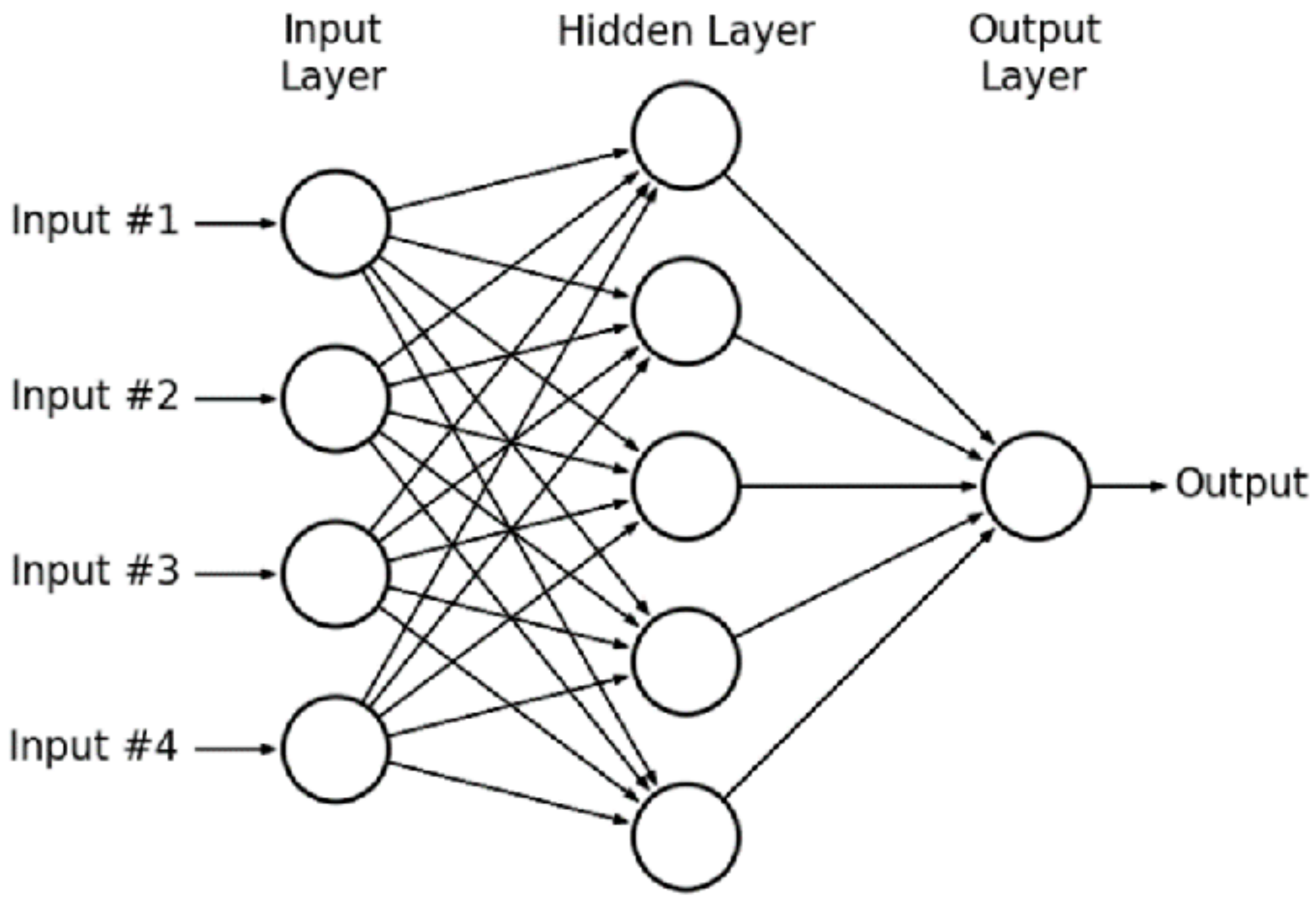




Multilayer Perceptrons

- 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

- After the disappointment of early neural networks, there was a brief boom period of “expert systems” using **symbolic AI**
- Limitations of expert systems caused a 2nd AI winter, which ended with modern advances in pattern recognition and deep neural networks (i.e., machine learning)

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 anything in the world
    - e.g., (Bagels), (ChatGPT), (Charley), etc…
  - **Relations** can be a predicate that describes a symbol or verbs describing how symbols interact with other symbols
    - toasted(Bagel)
    - eat(Charley, Bagel)
- By populating a **knowledge base** with symbols and relations, we can use a program to find new propositions (*inference*)

Symbolic vs. sub-symbolic AI

Symbolic models

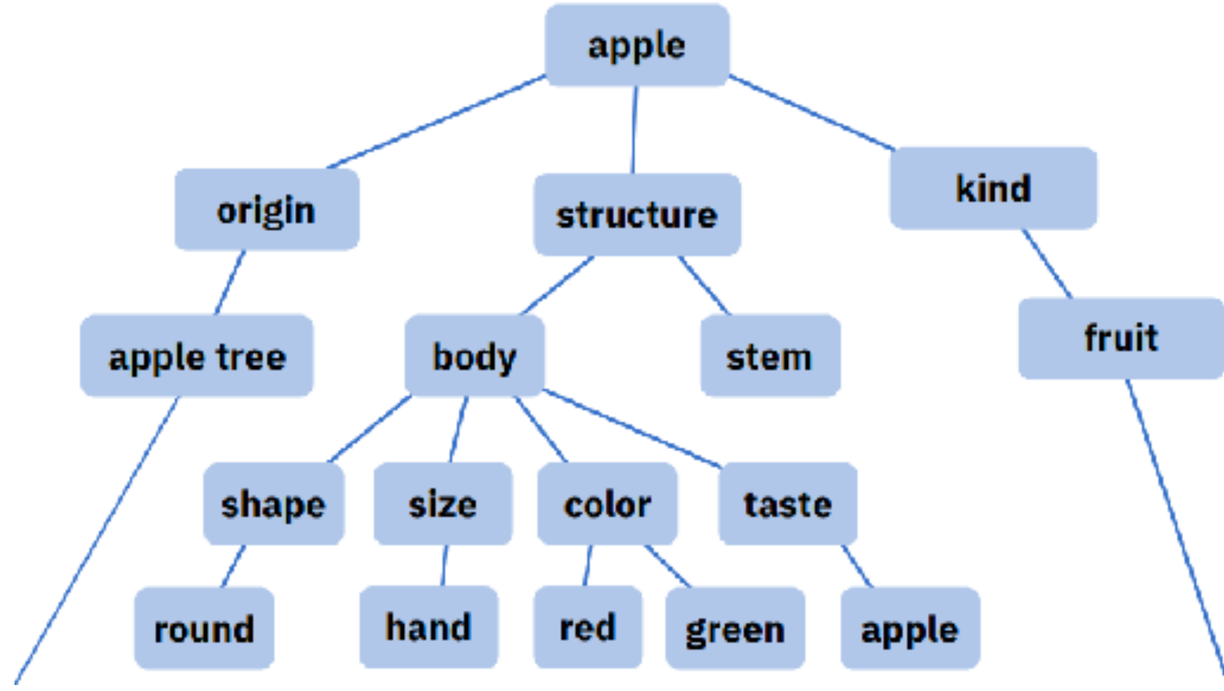
- Symbols, rules, and structured representations
    - express logical operations
- Compositionality: symbols and rules can be combined to produce new representations

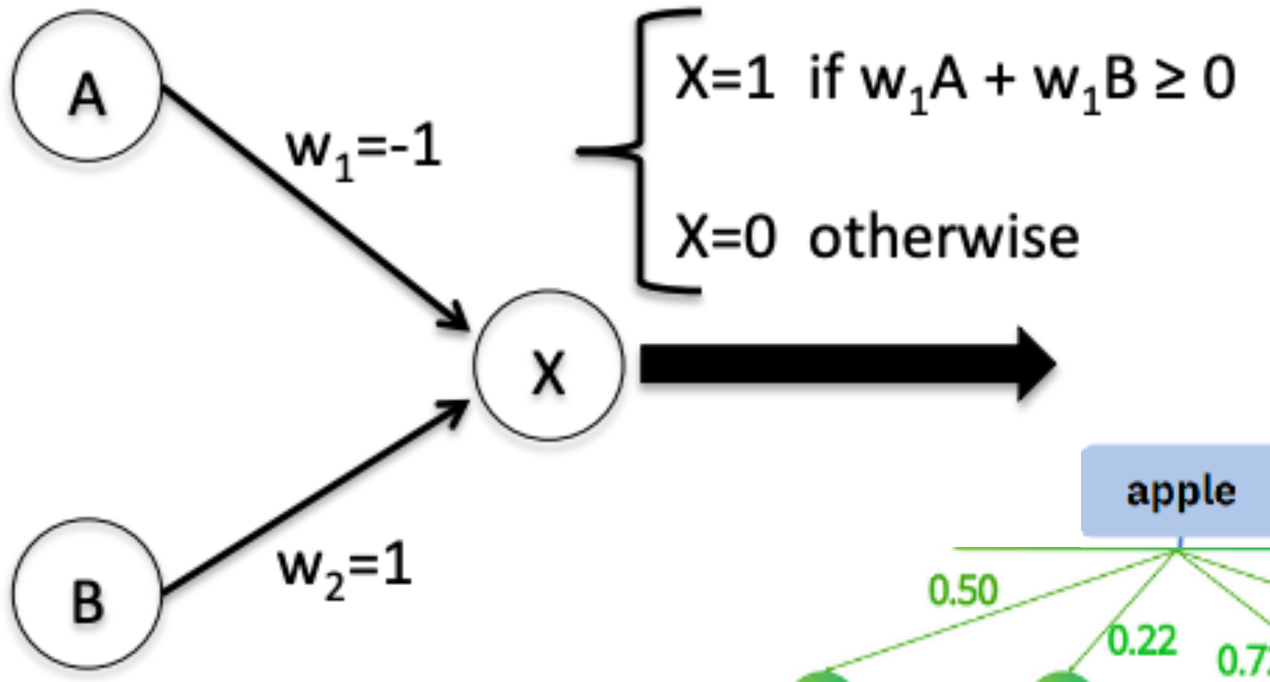
Sub-symbolic models

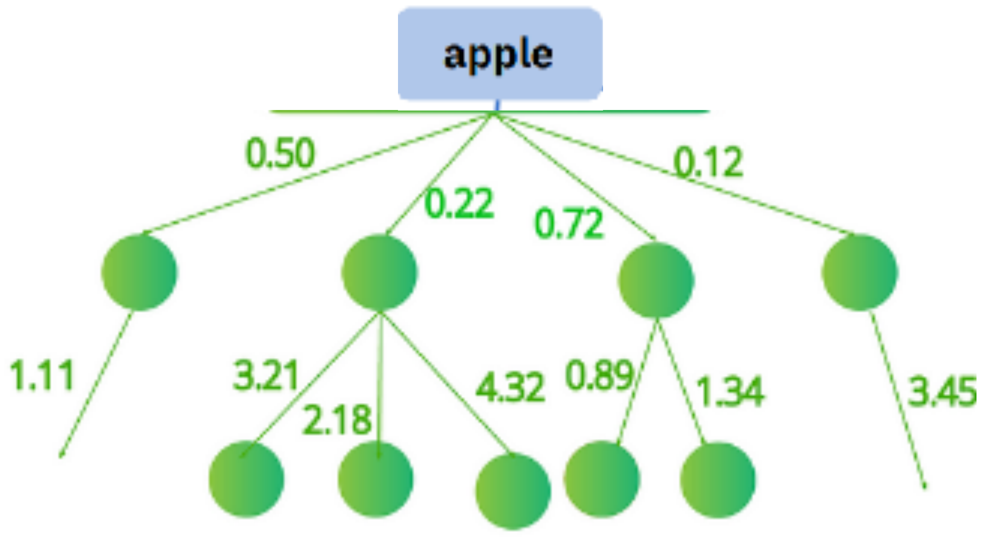
- Neural networks encoding information through connection weights
- No explicit representation of concepts or knowledge, but distributed throughout the network
- Knowledge can be implicitly learned by capturing statistical patterns
- The rise of deep learning takes advantage of the scalability of subsymbolic learning mechanisms

A => B (*"A implies B"*)

A	B	A => B
0	0	1
0	1	1
1	0	0
1	1	1

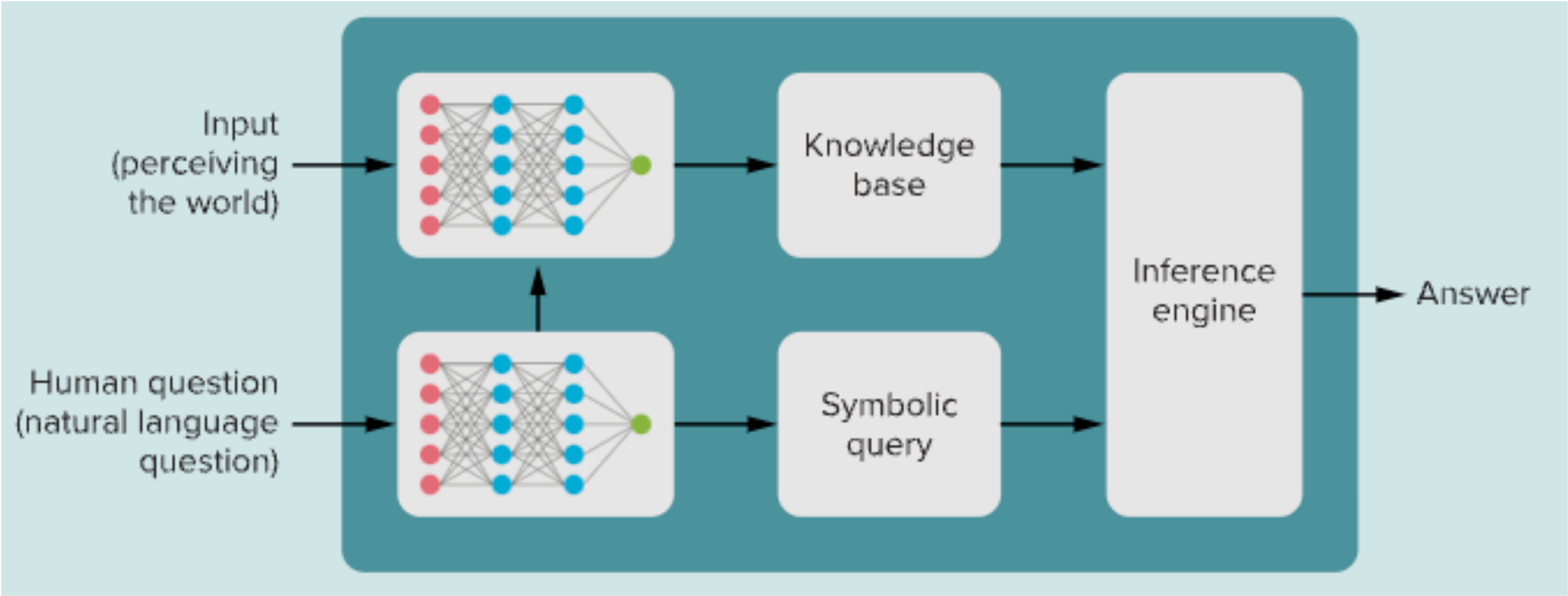




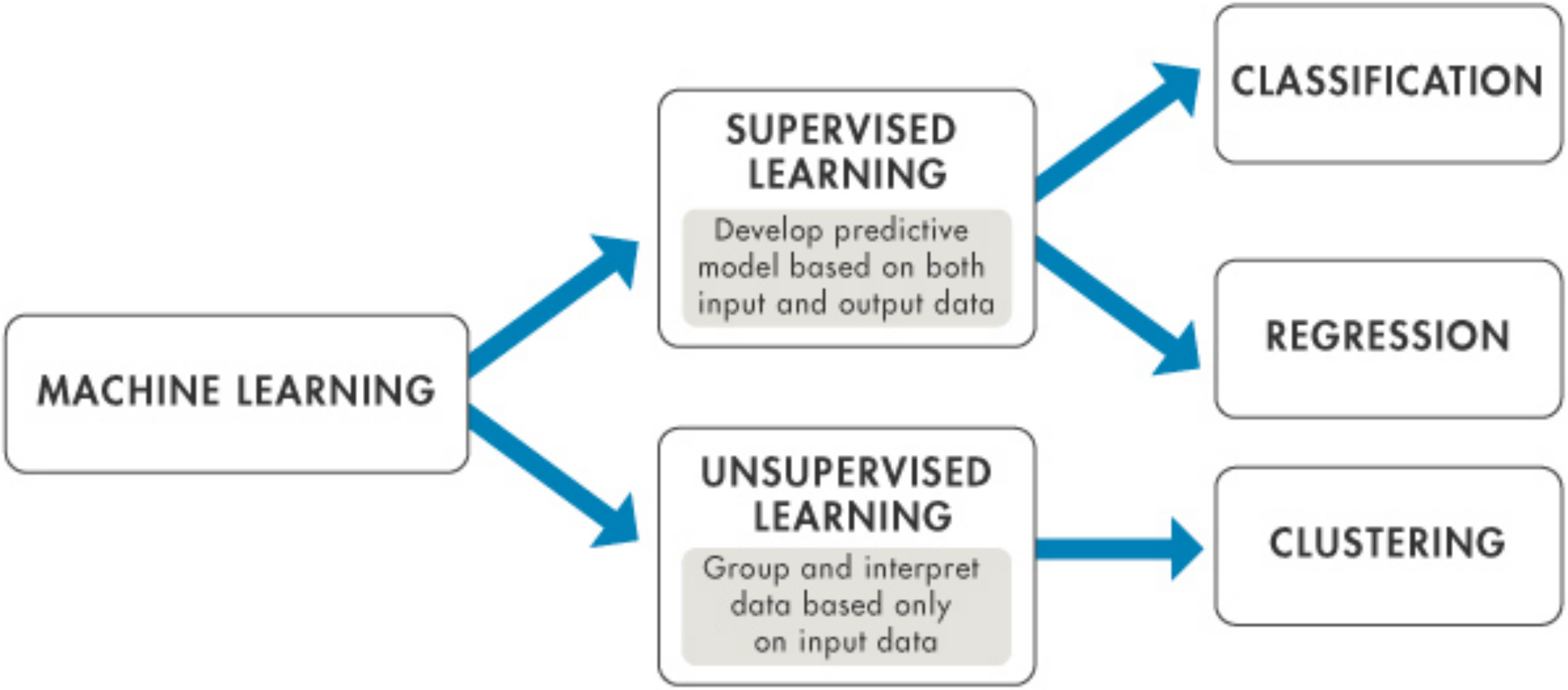


Hybrid systems: Neurosymbolic AI

- Symbolic and subsymbolic approaches can operate together to get the best of both worlds
- Subsymbolic neural networks can be used to extract symbolic representations
- Modern AI assistants (e.g., Siri, Google, Alexa) are essentially expert systems with voice recognition and text-to-speech added on

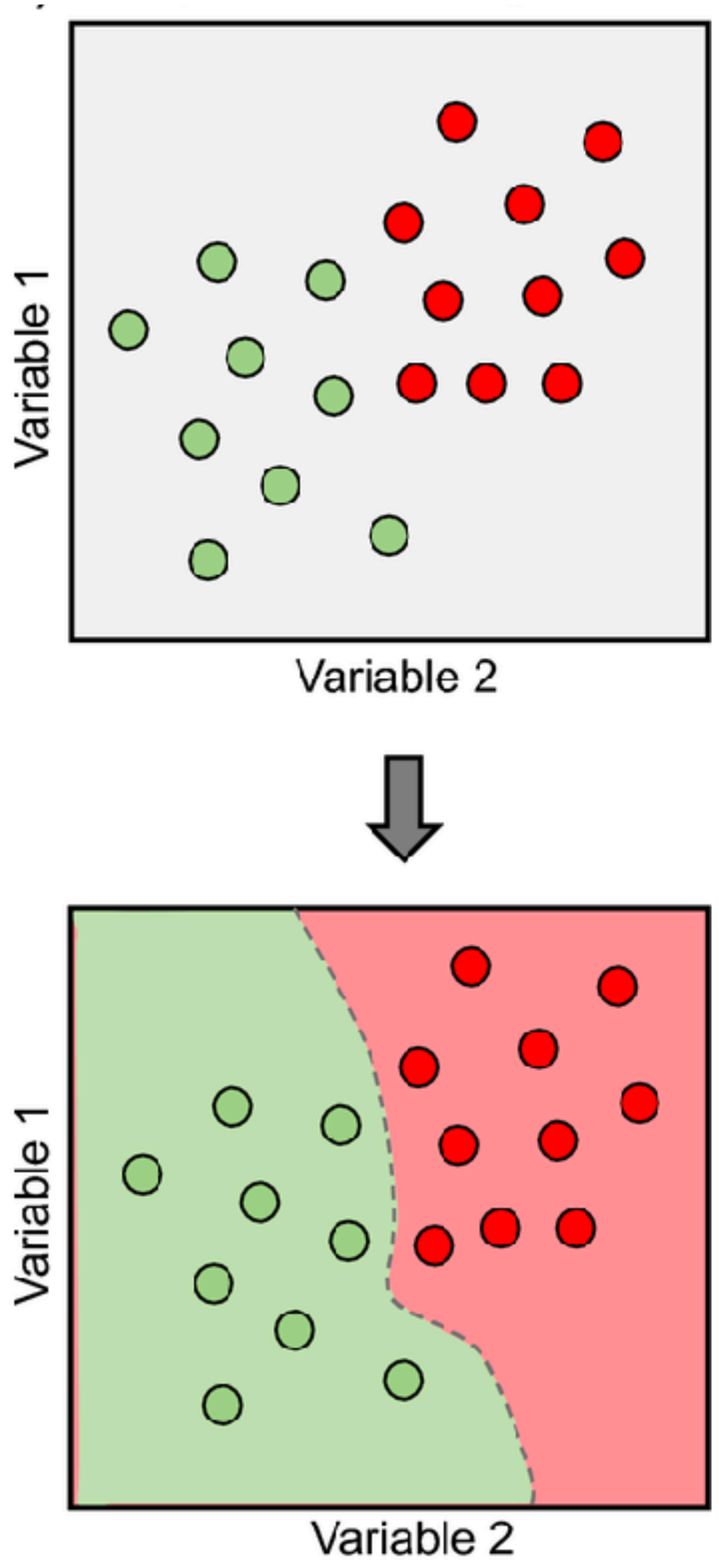


Modern Machine Learning

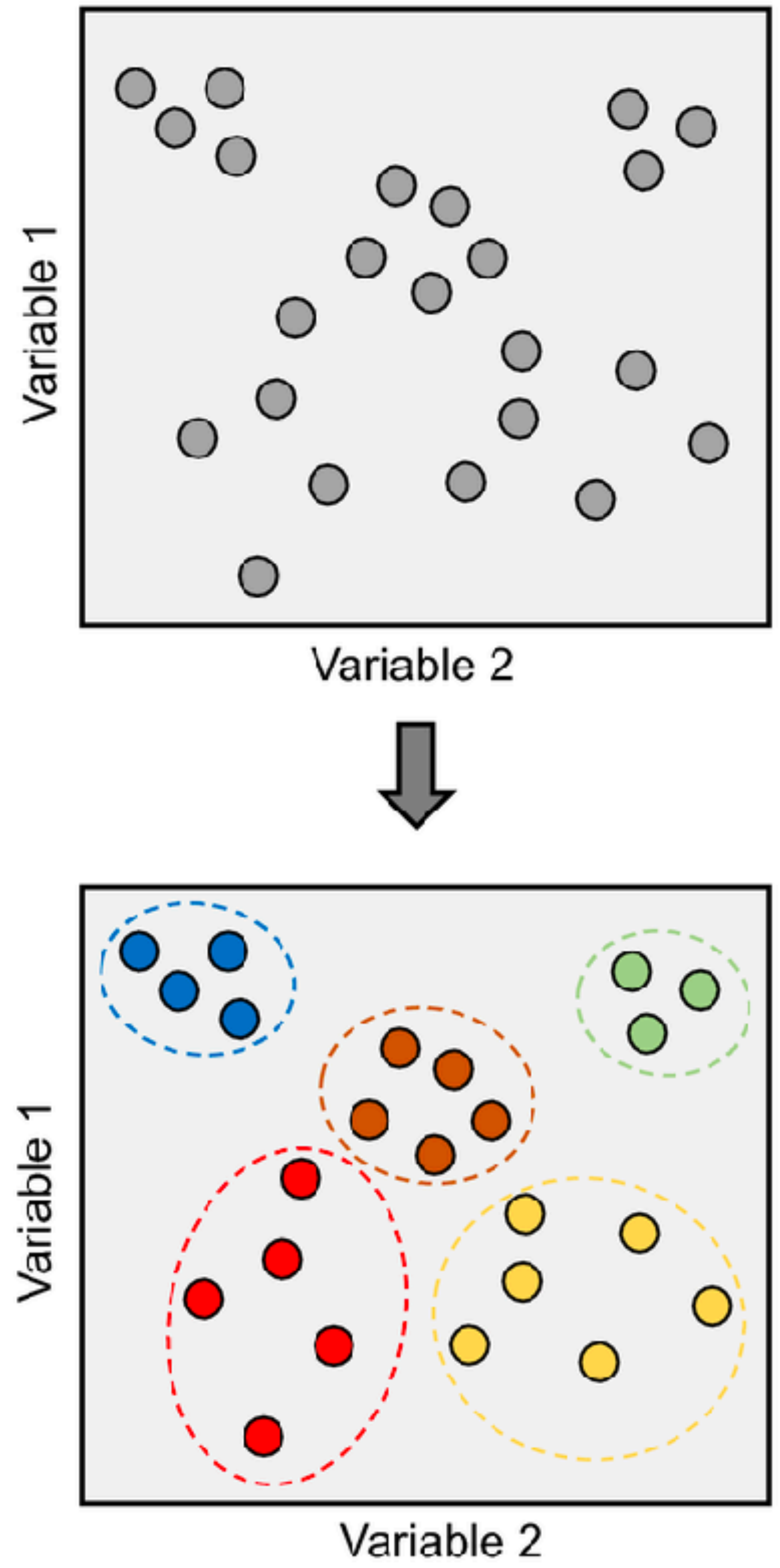


Supervised vs. unsupervised learning

Supervised

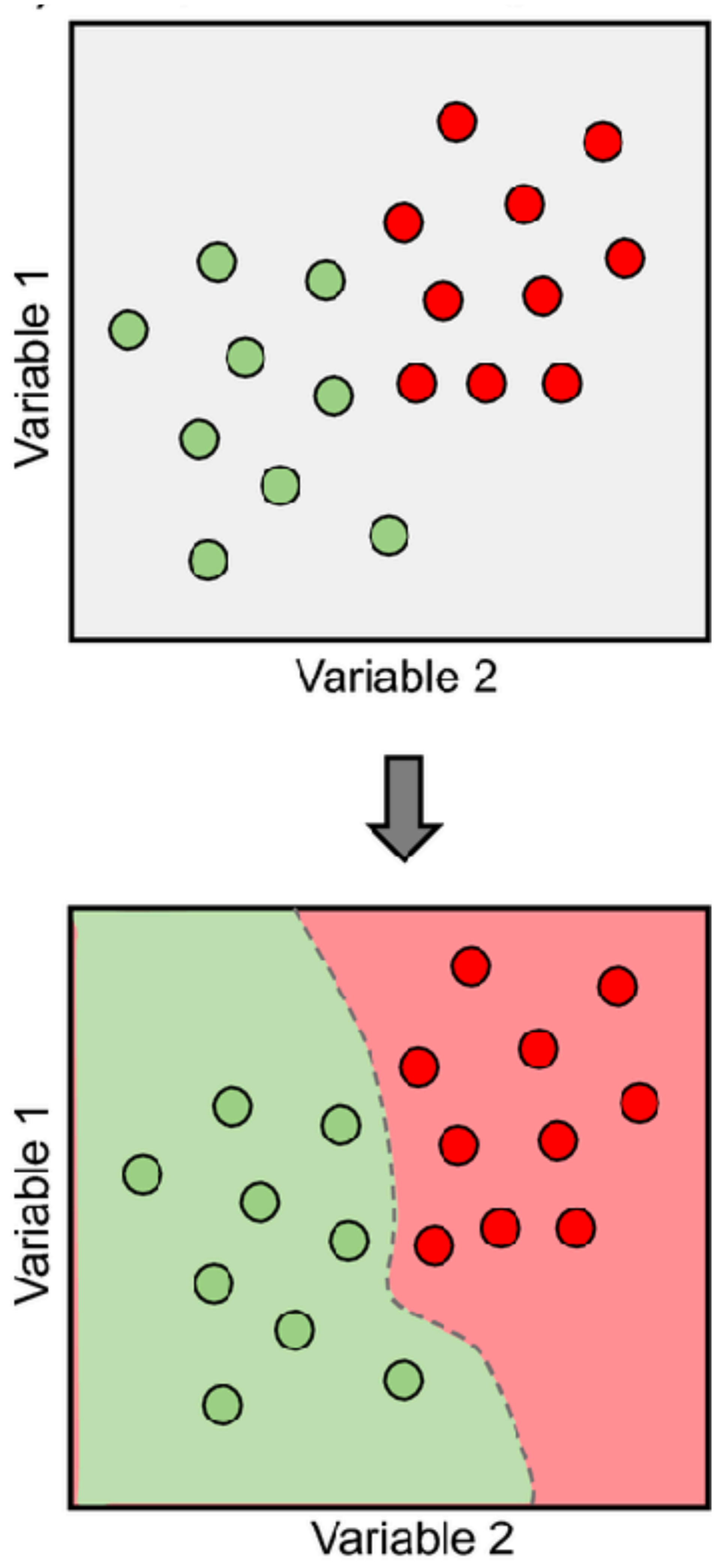


Unsupervised



Supervised vs. unsupervised learning

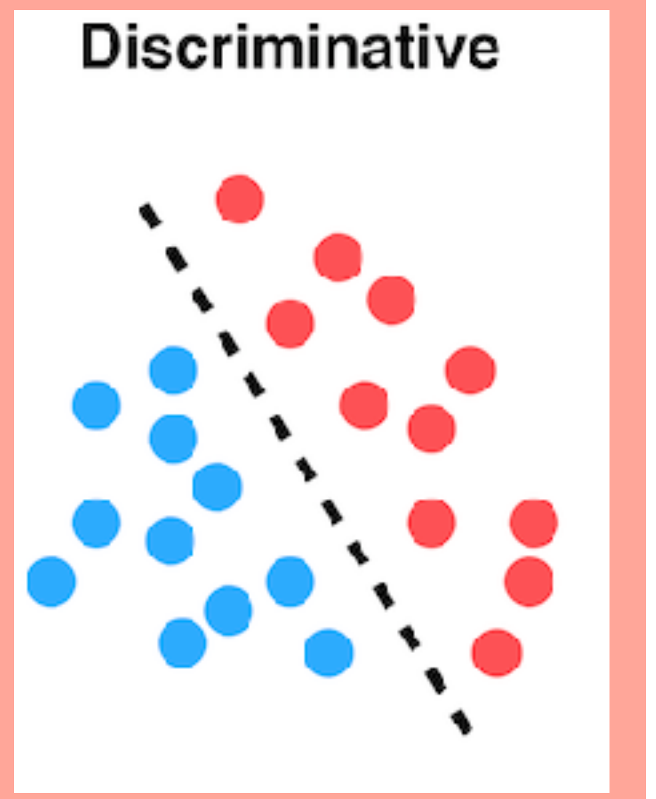
Supervised



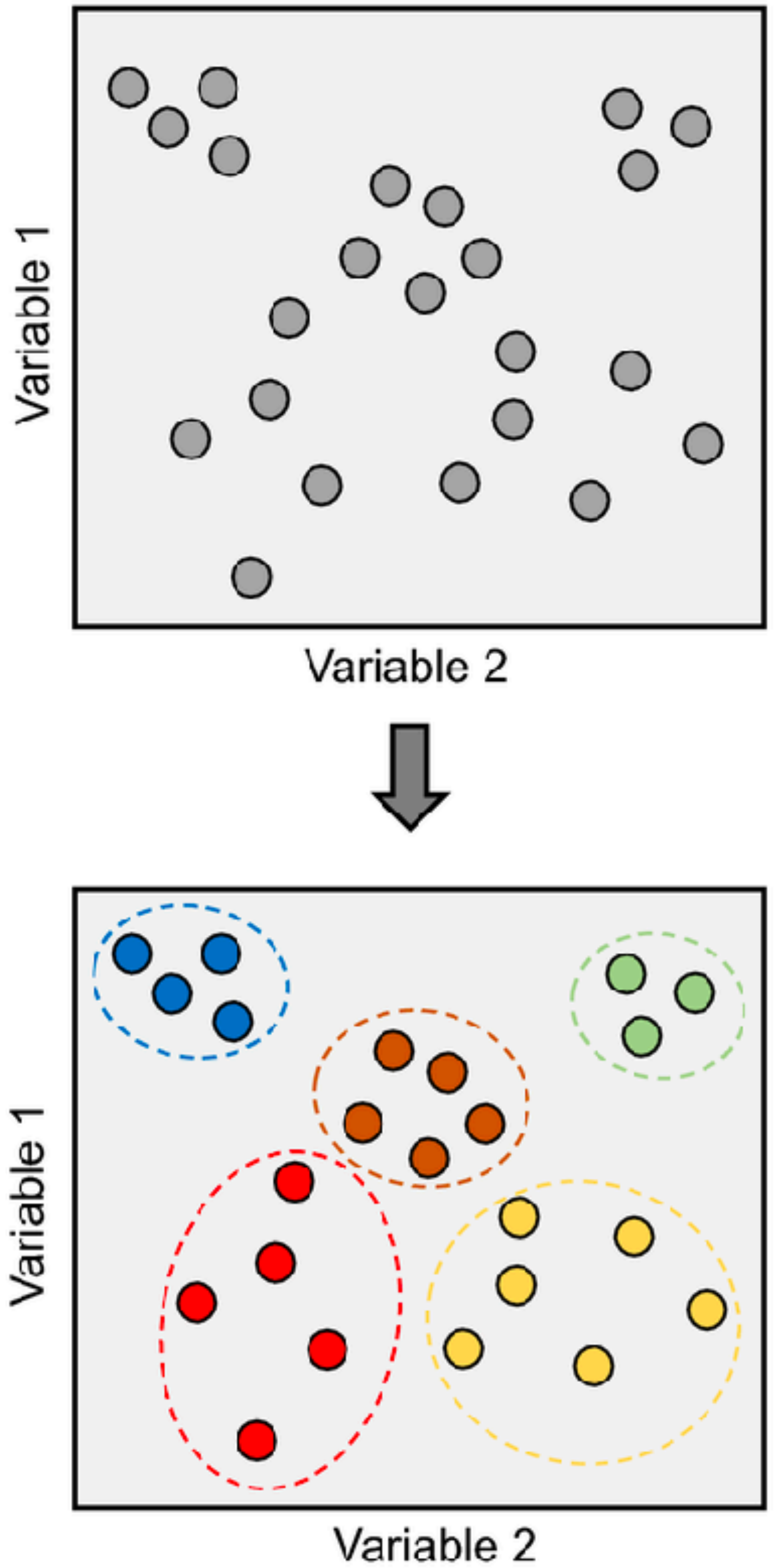
MLPs

Decision trees and random forests

SVMs

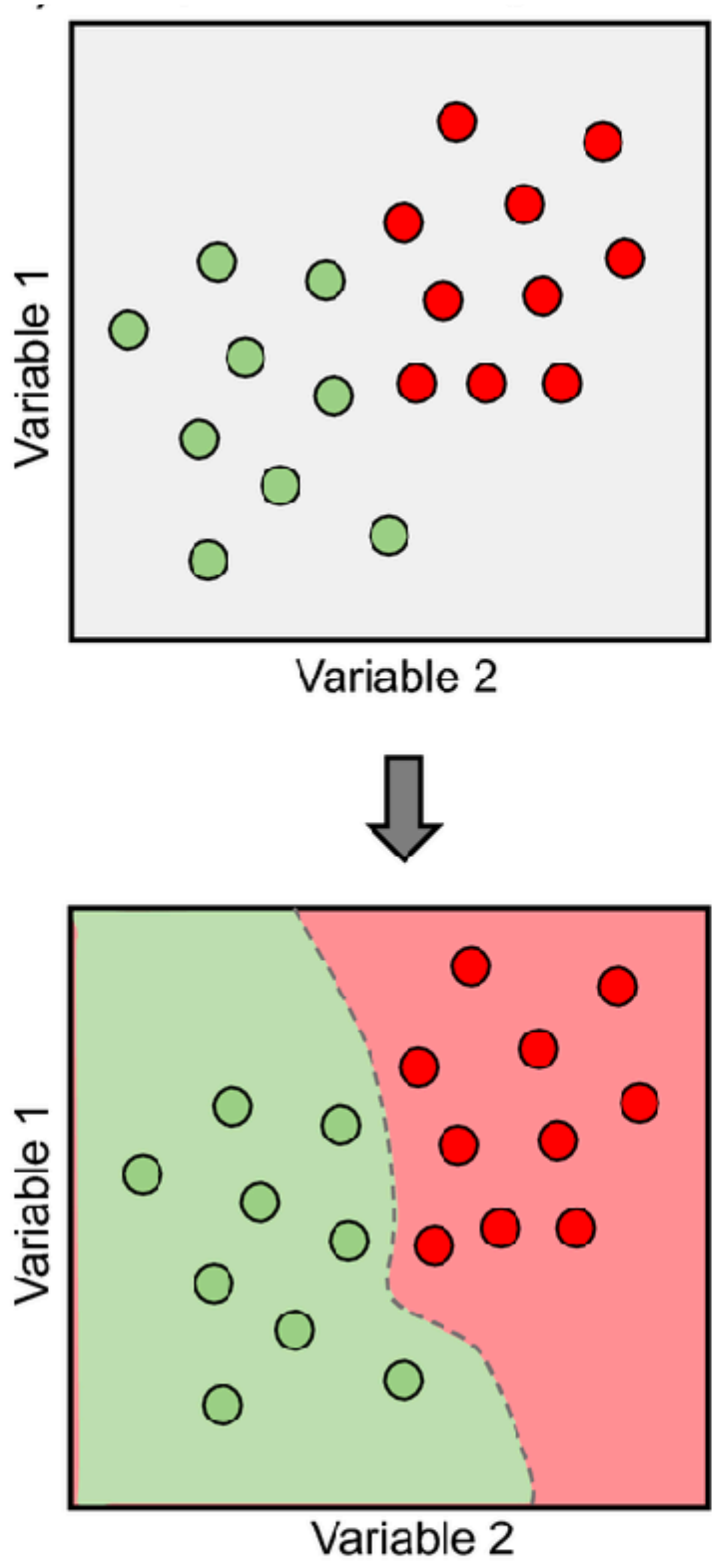


Unsupervised

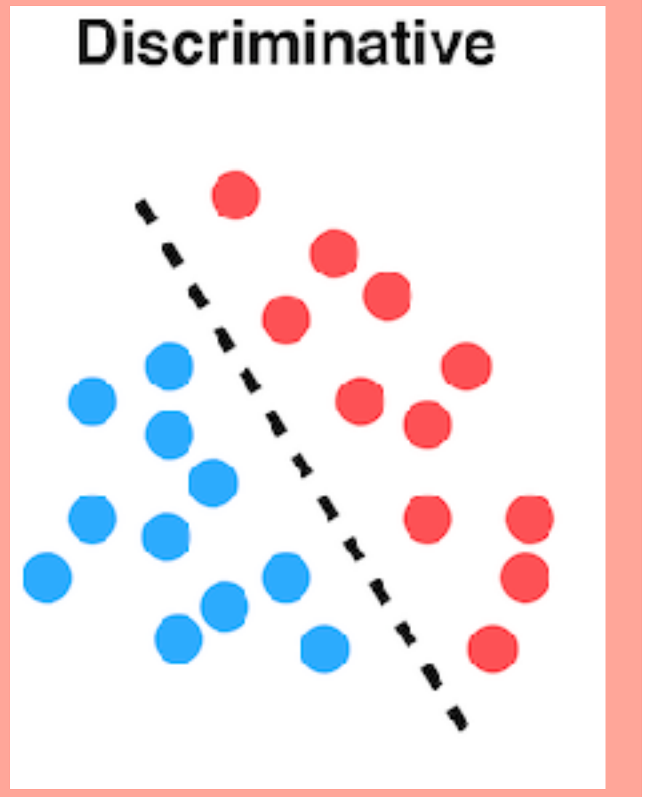


Supervised vs. unsupervised learning

Supervised



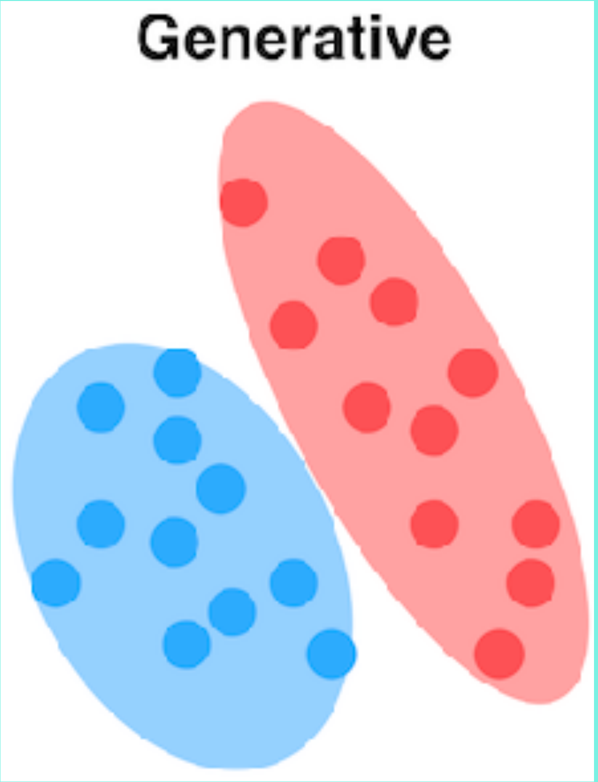
MLPs



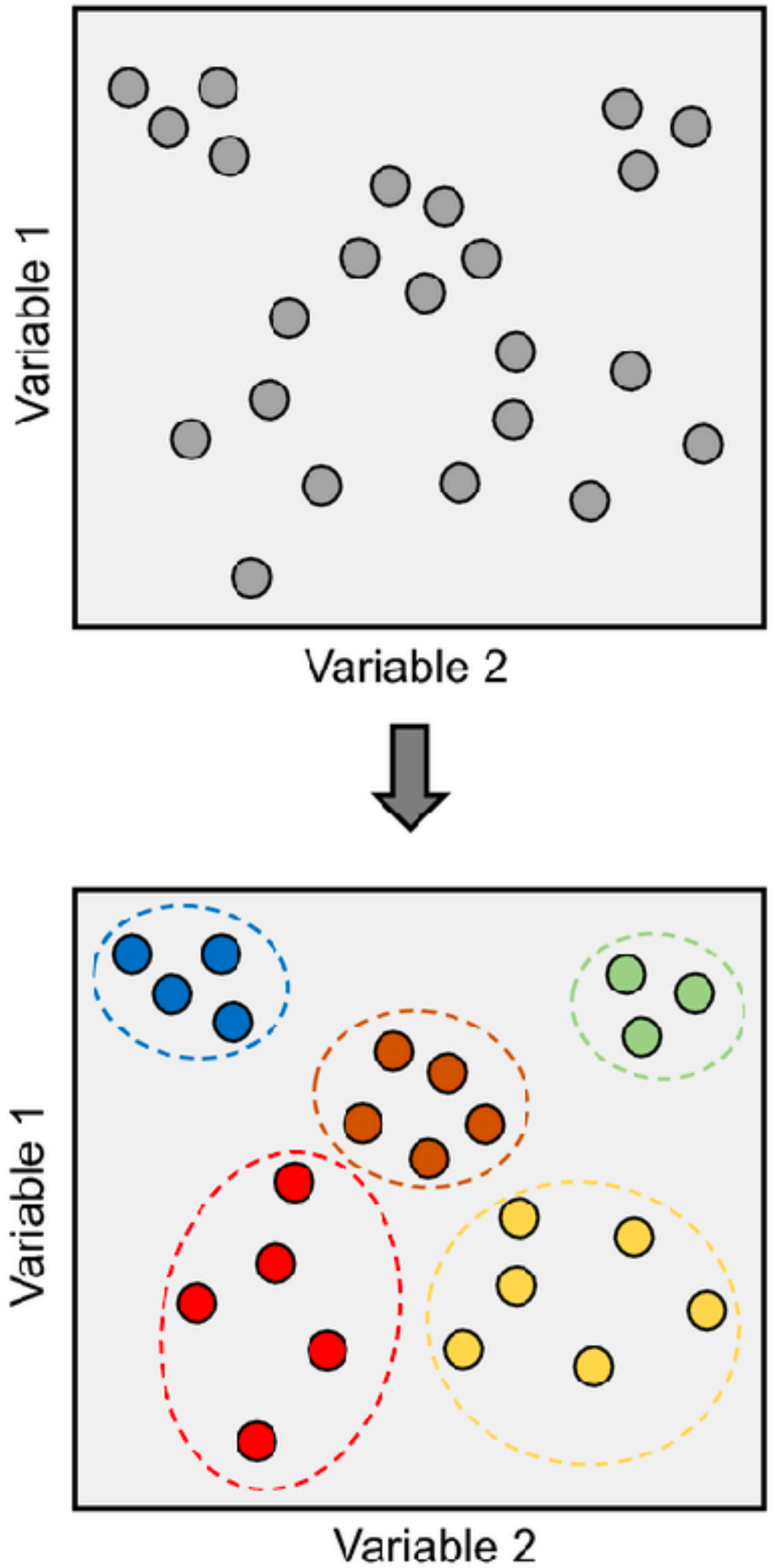
Decision trees and random forests

SVMs

Naïve Bayes

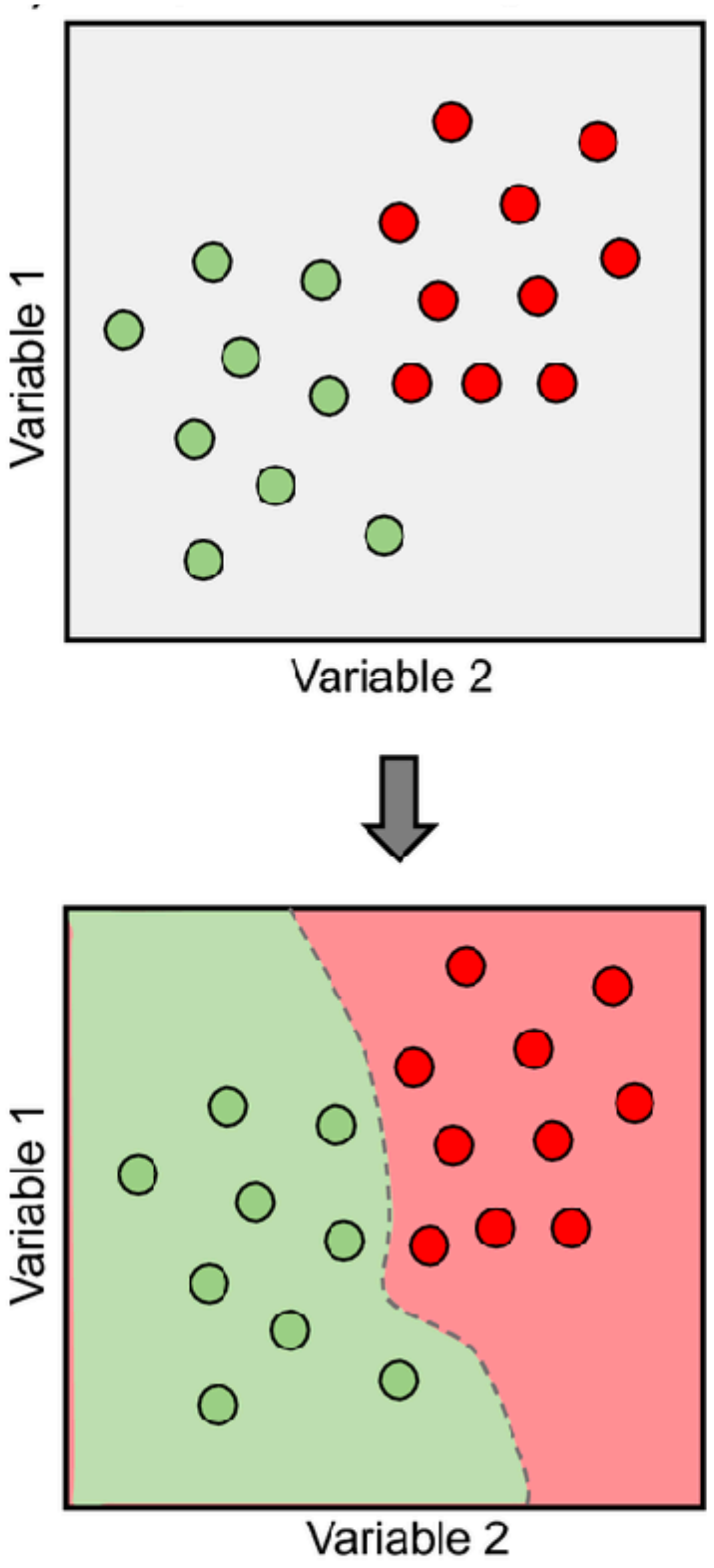


Unsupervised

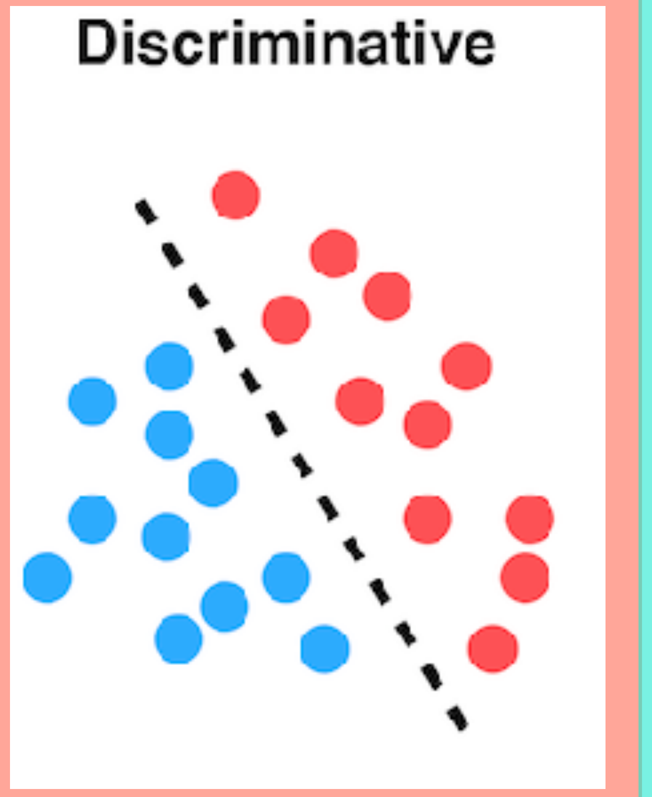


Supervised vs. unsupervised learning

Supervised



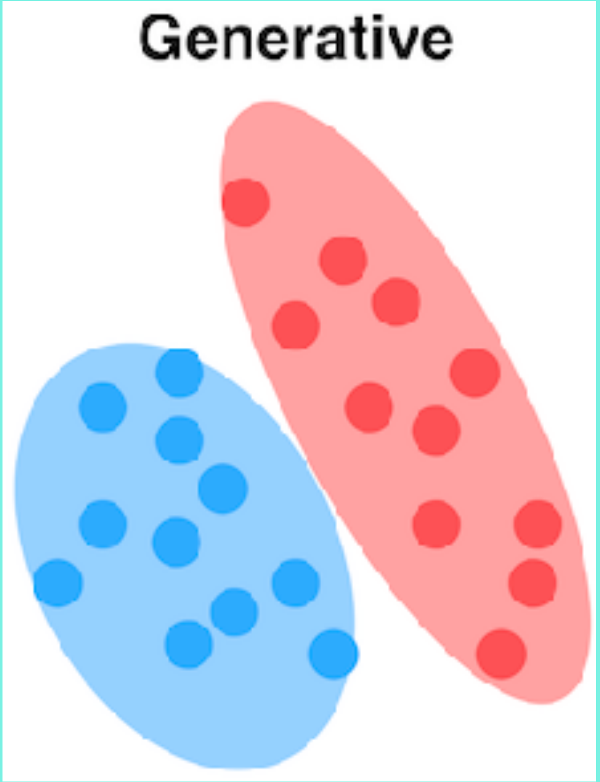
MLPs



Decision trees and random forests

SVMs

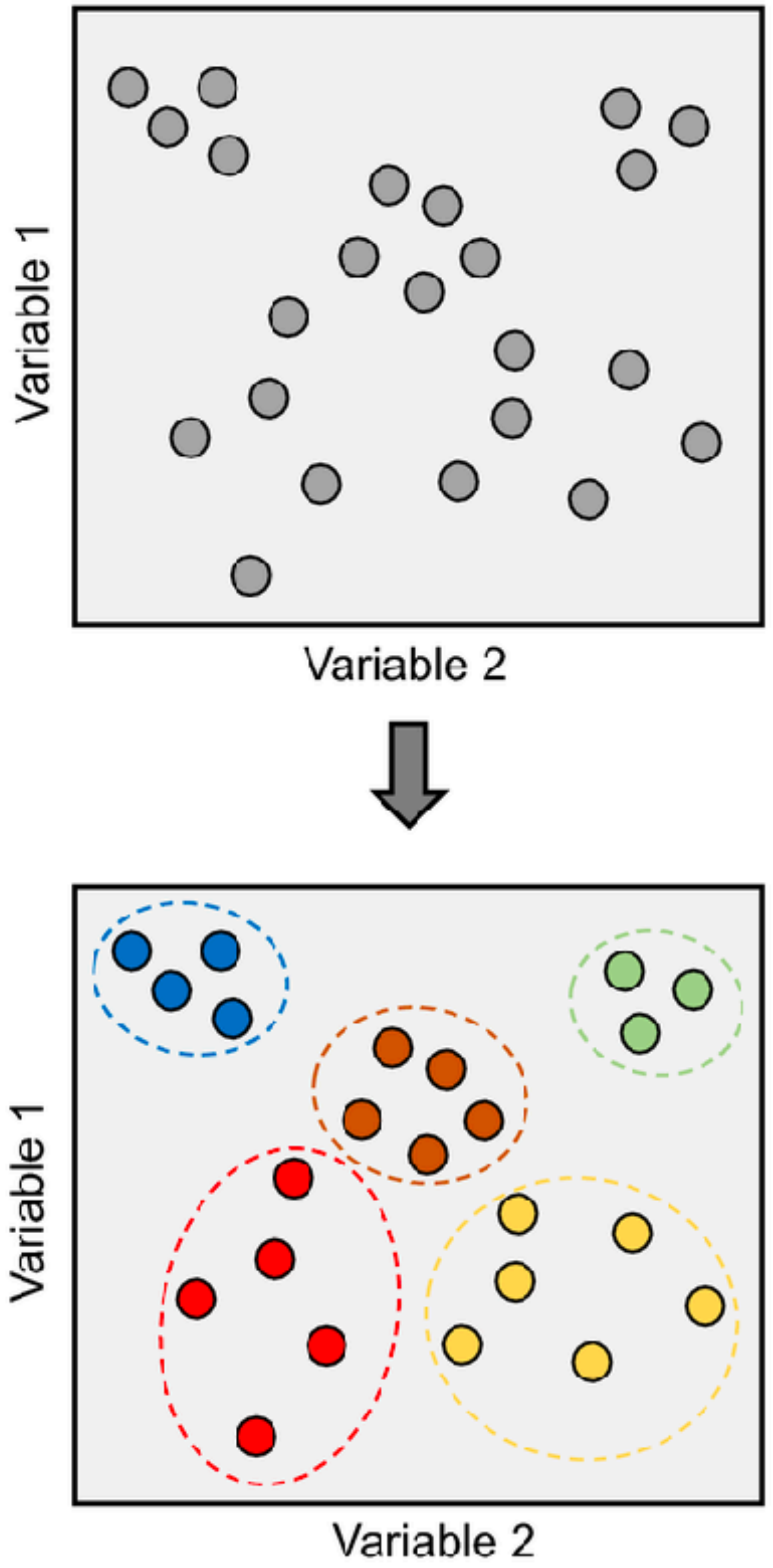
Naïve Bayes



k-Means

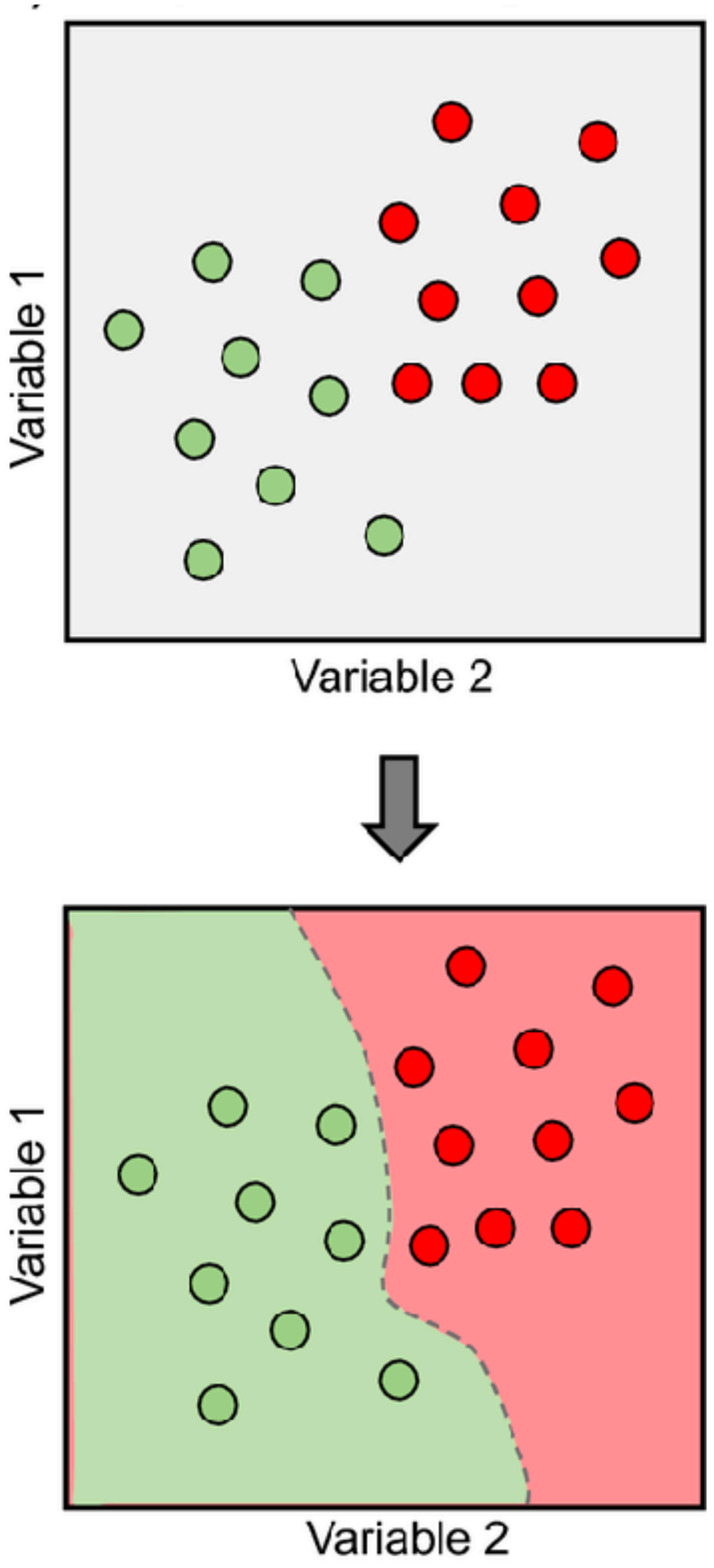
GMMs

Unsupervised

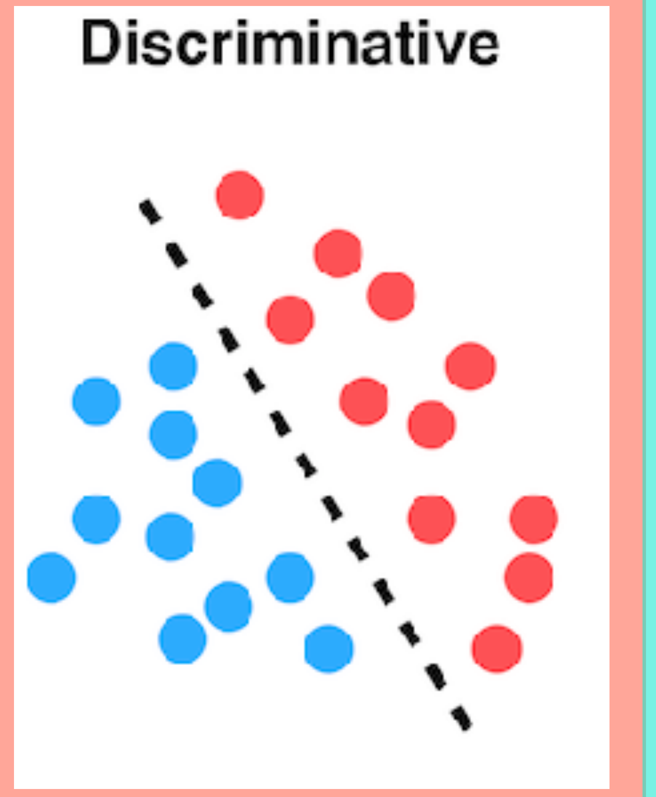


Supervised vs. unsupervised learning

Supervised



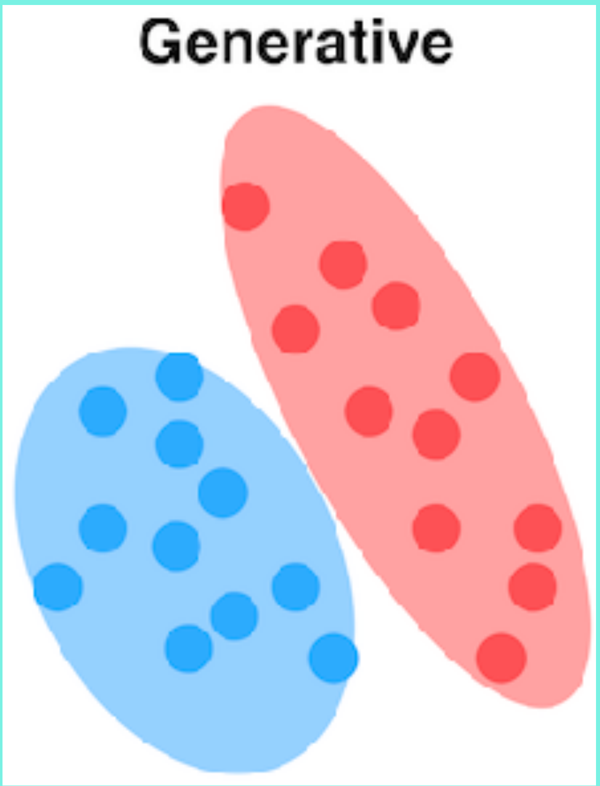
MLPs



Decision trees and random forests

SVMs

Naïve Bayes

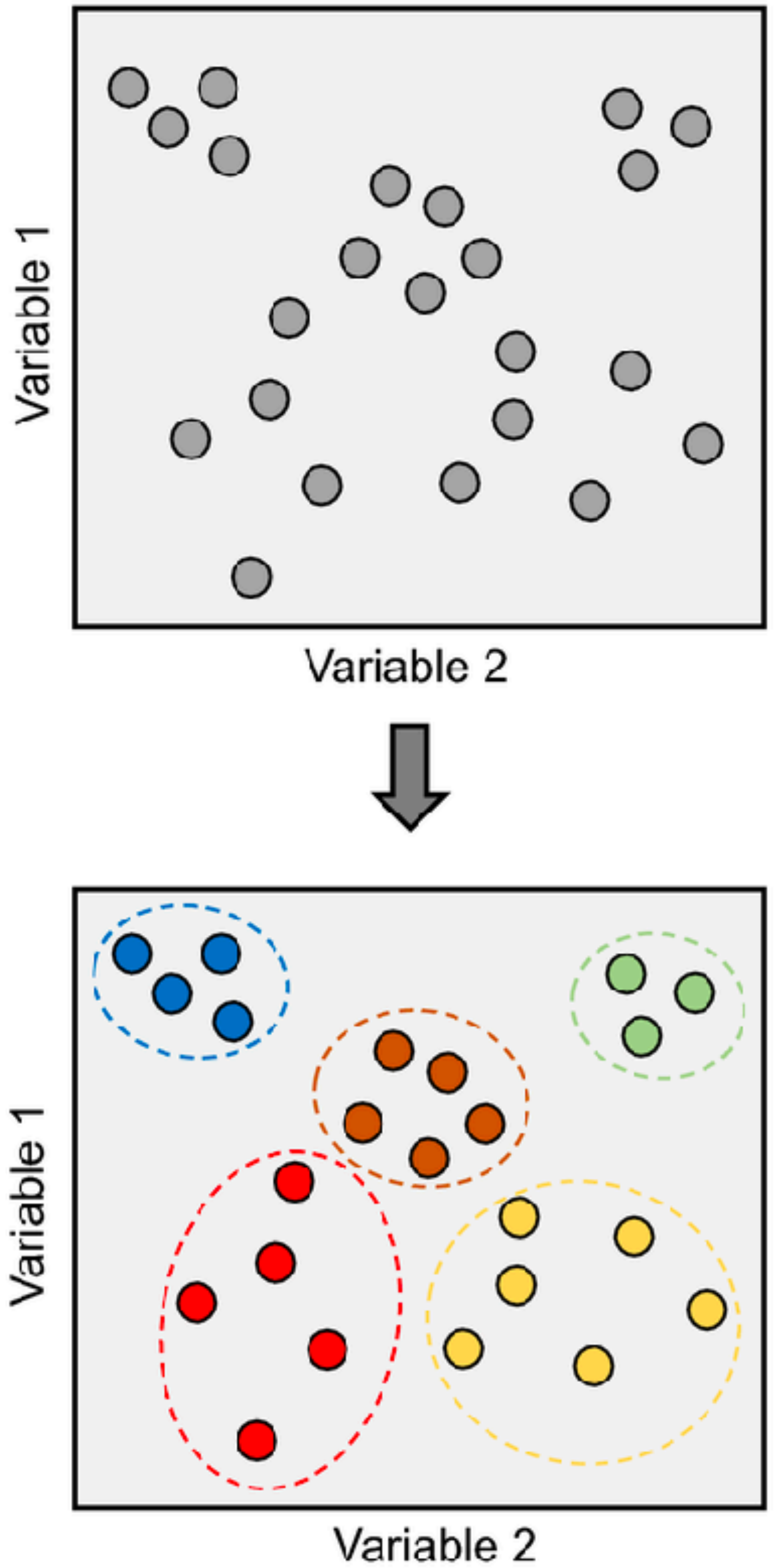


Learn data distribution

k-Means

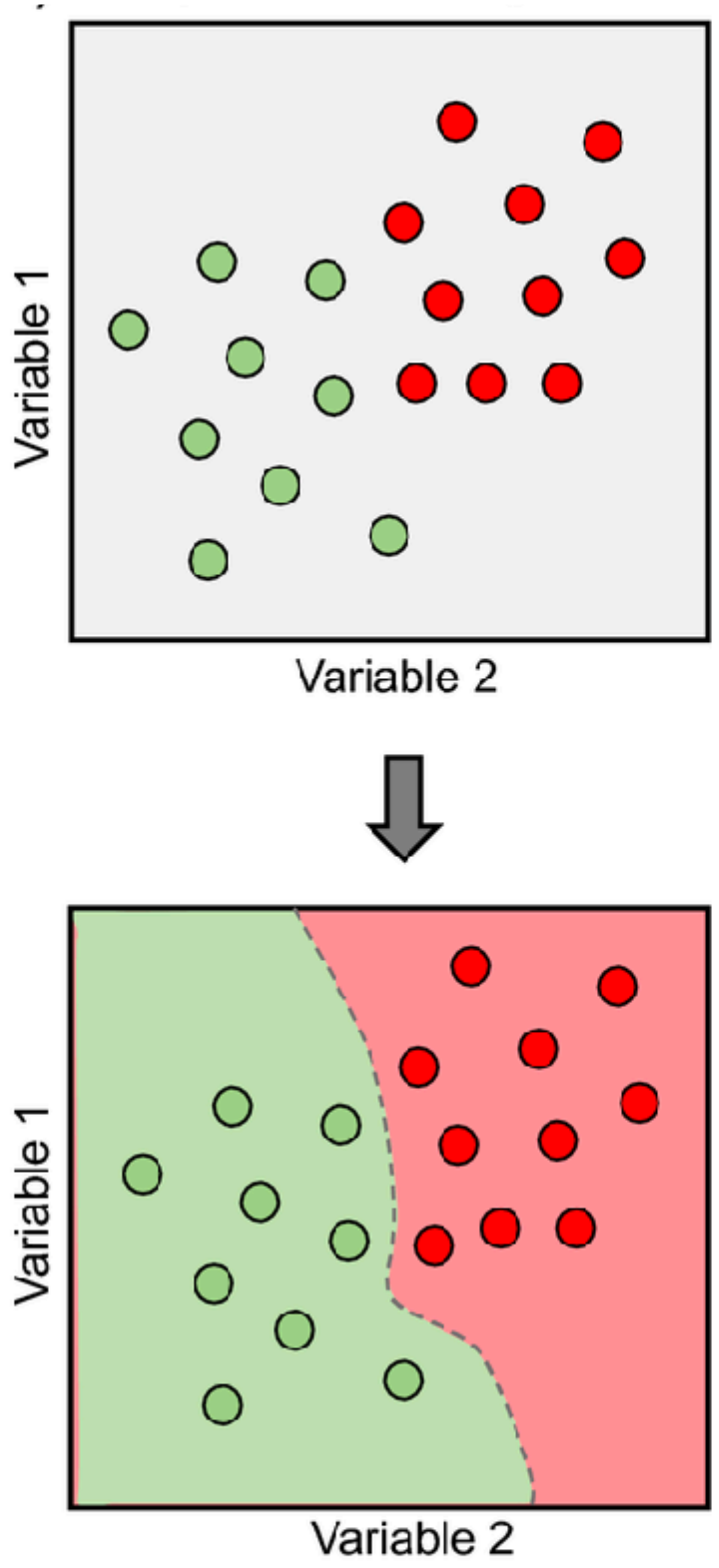
GMMs

Unsupervised



Supervised vs. unsupervised learning

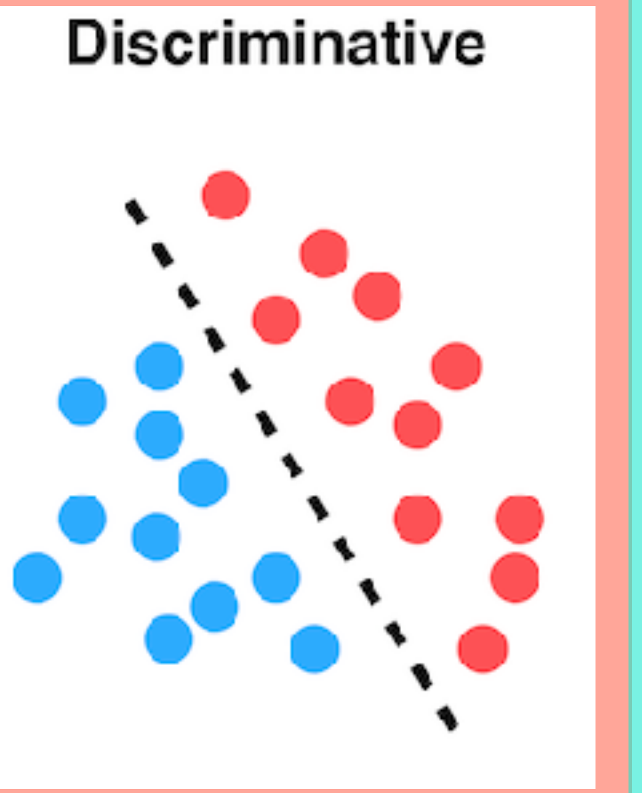
Supervised



Learn decision boundary

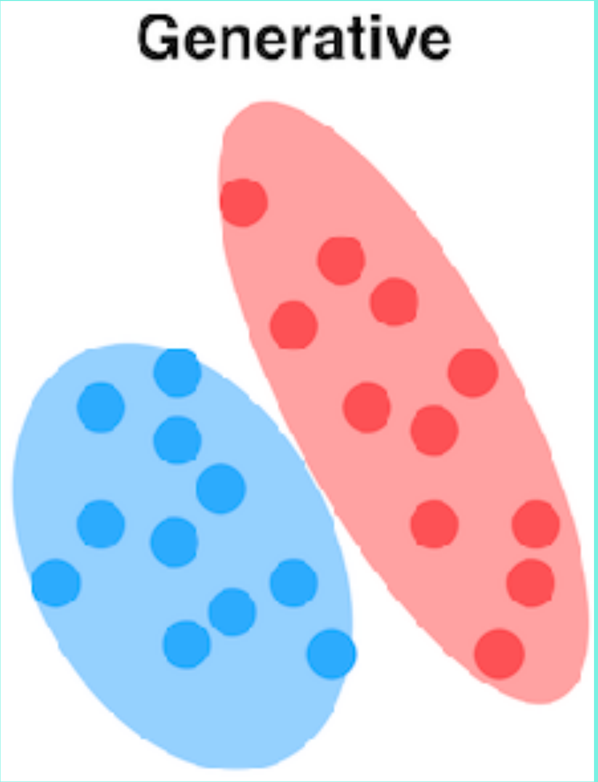
MLPs

Decision trees and random forests



SVMs

Naïve Bayes

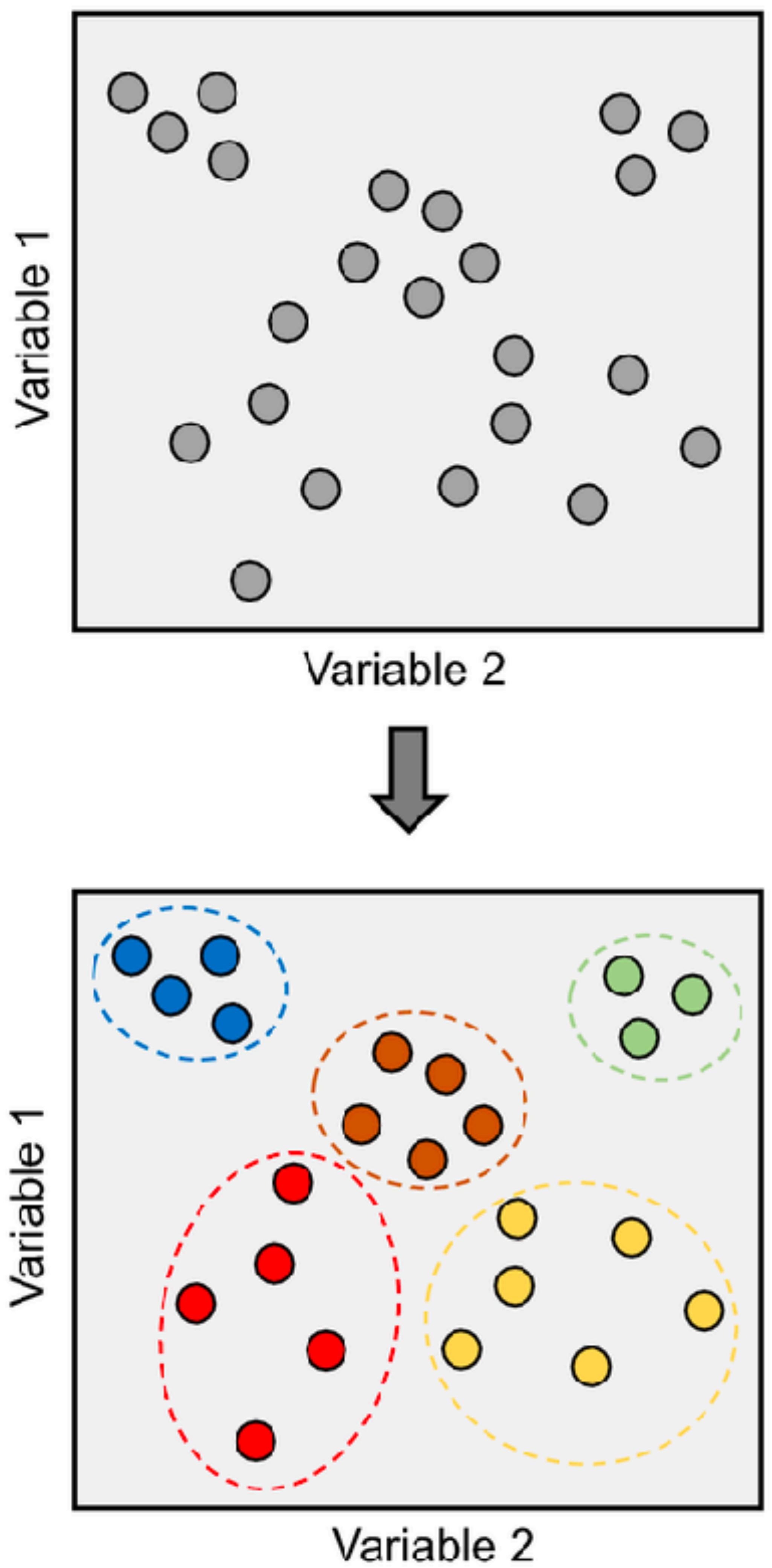


Learn data distribution

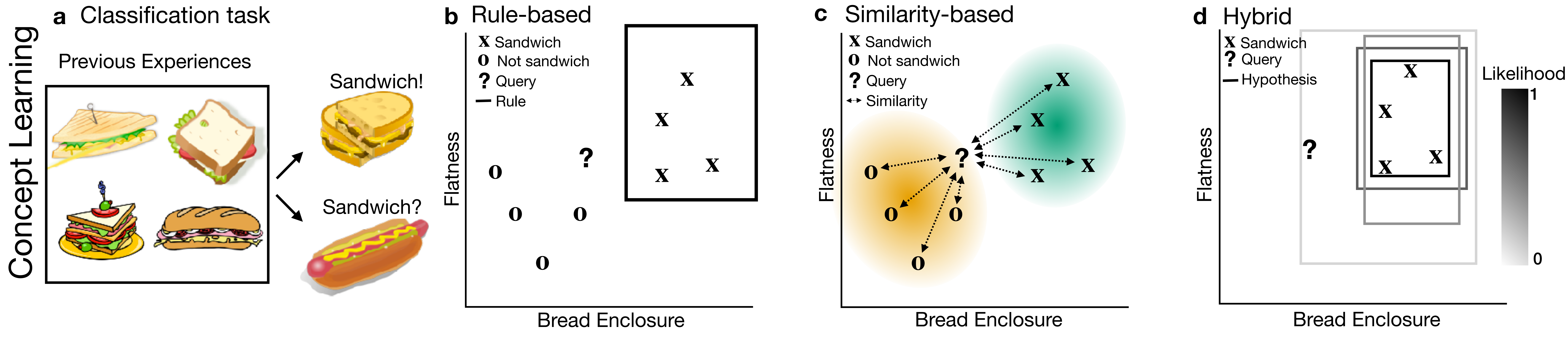
k-Means

GMMs

Unsupervised



Learning concepts



- Concepts are mental representations of categories in the world
- 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

Bayesian Concept Learning

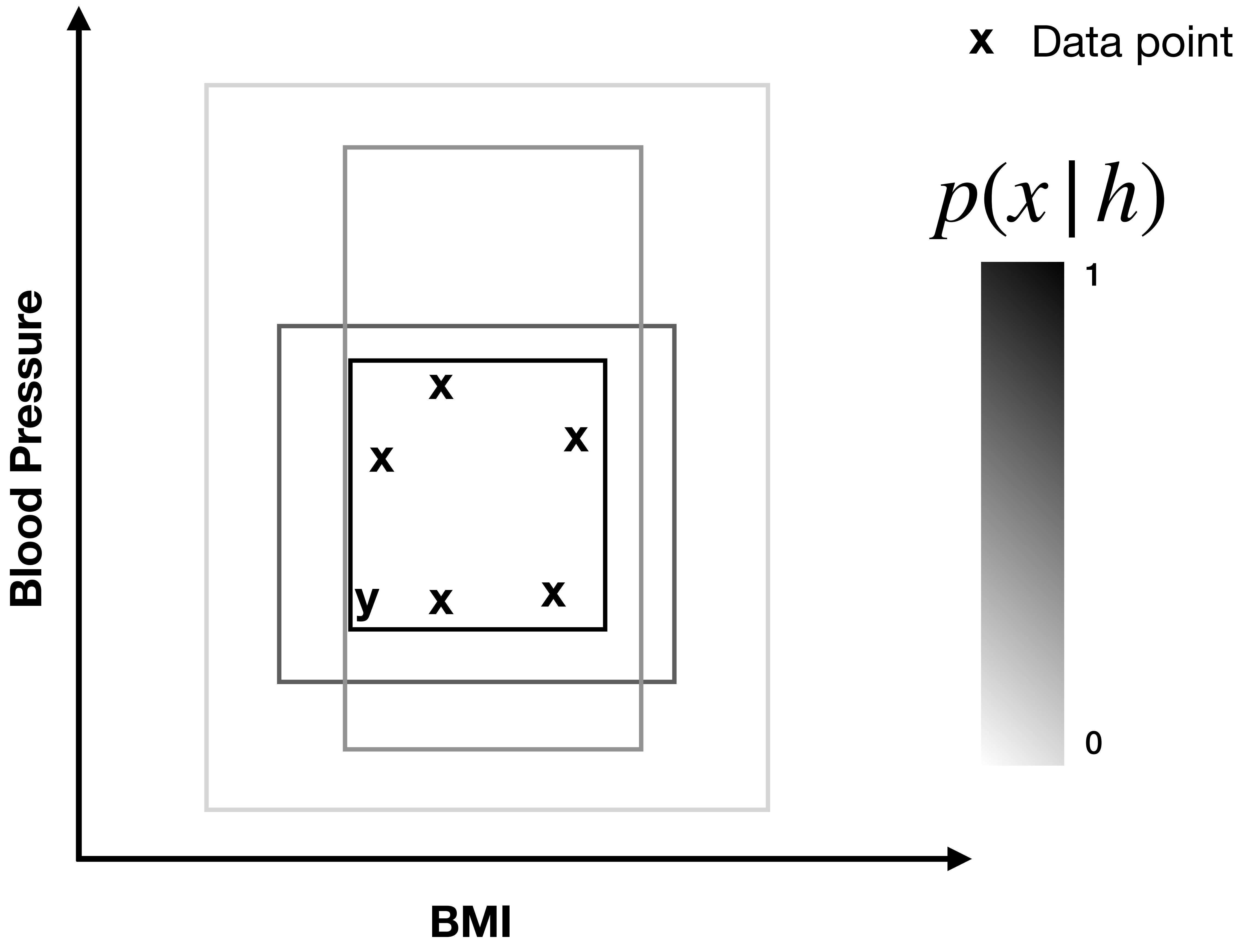
- Uses a distribution over rule-like hypotheses, and produces similarity-like generalization gradients

- The probability of y being in the same category of x is thus based on summing over all hypotheses consistent with the data

$$p(y \in C|x) = \sum_{h:y \in h} p(h|x). \quad \text{where} \quad p(h|x) = \frac{p(x|h)p(h)}{p(x)}$$

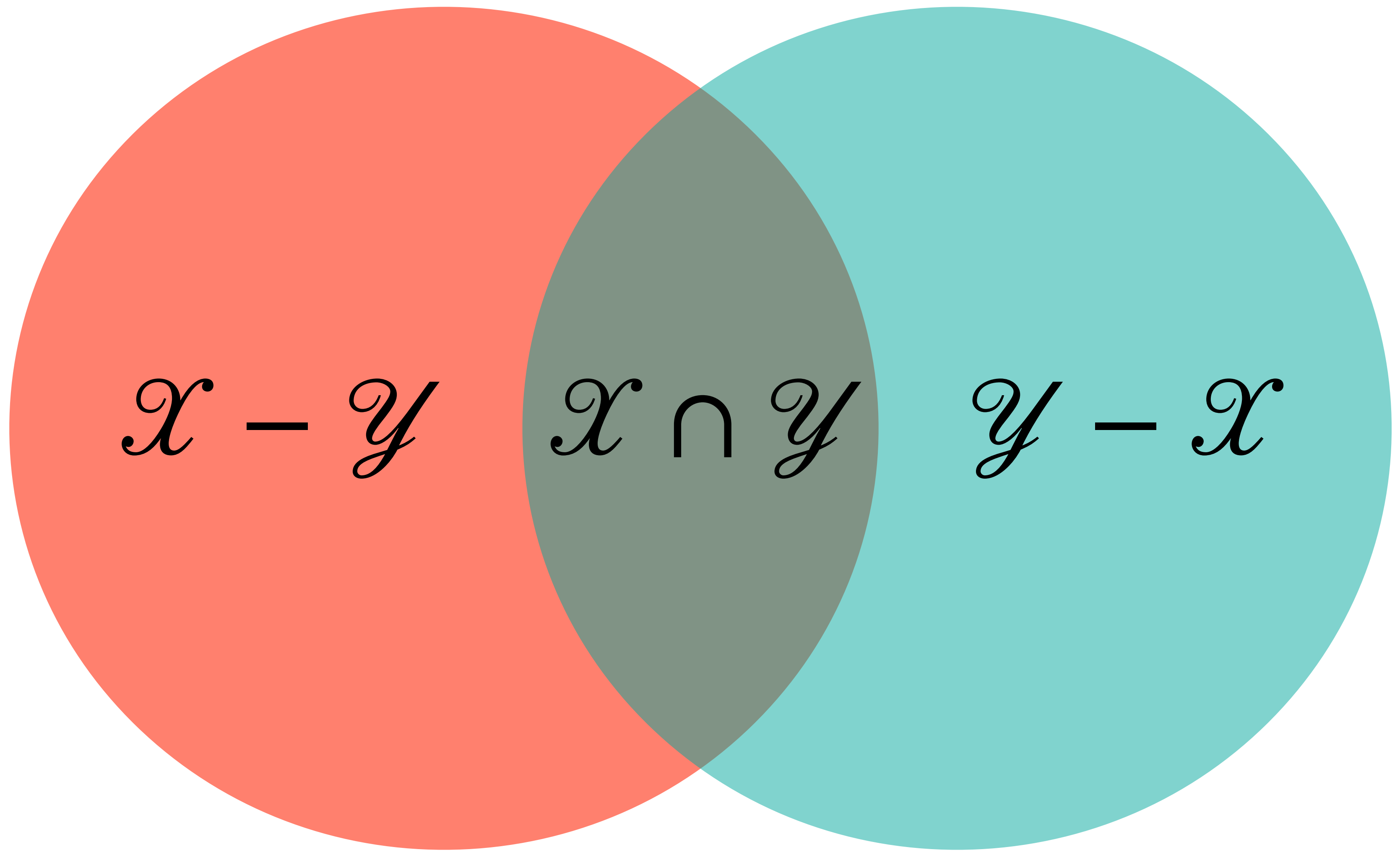
- Where narrower hypotheses are favored under strong sampling (**Bayesian Size Principle**)

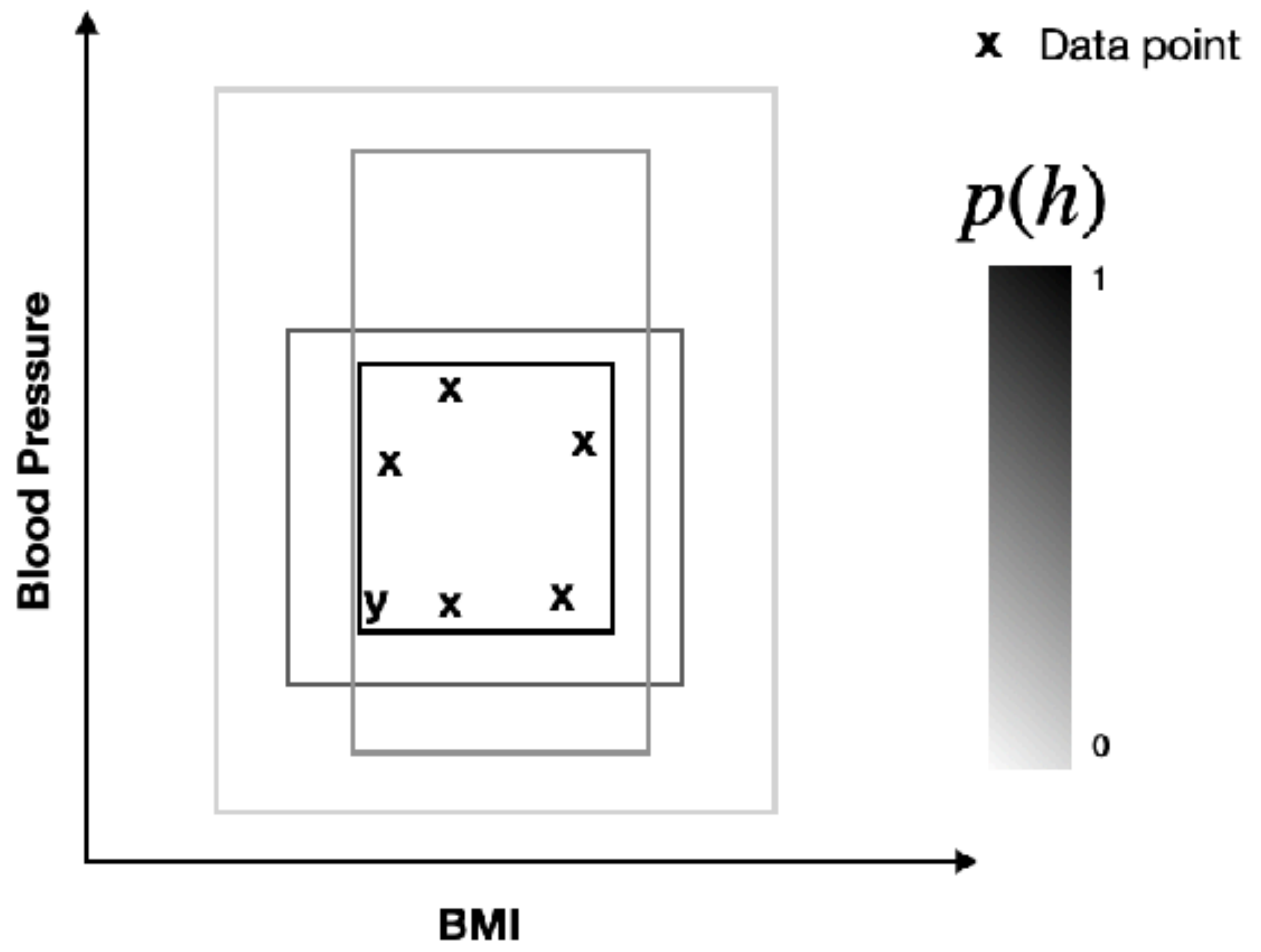
$$p(x|h) = \begin{cases} \frac{1}{|h|} & \text{if } x \in h \\ 0 & \text{otherwise} \end{cases} \qquad \text{[strong sampling]},$$



Tenenbaum (*NIPS* 1999)
Tenenbaum & Griffiths (*BBS* 2001)

Bayesian Concept Learning Subsumes Tversky's Contrast Model





Contrast model

$$S(y,x) = \theta f(\mathcal{Y} \cap \mathcal{X}) - \alpha f(\mathcal{Y} - \mathcal{X}) - \beta f(\mathcal{X} - \mathcal{Y}),$$

Ratio model (alternative form)

$$S(y,x) = 1 \Big/ \left[1 + \frac{\alpha f(\mathcal{Y} - \mathcal{X}) + \beta f(\mathcal{X} - \mathcal{Y})}{f(\mathcal{Y} \cap \mathcal{X})} \right].$$

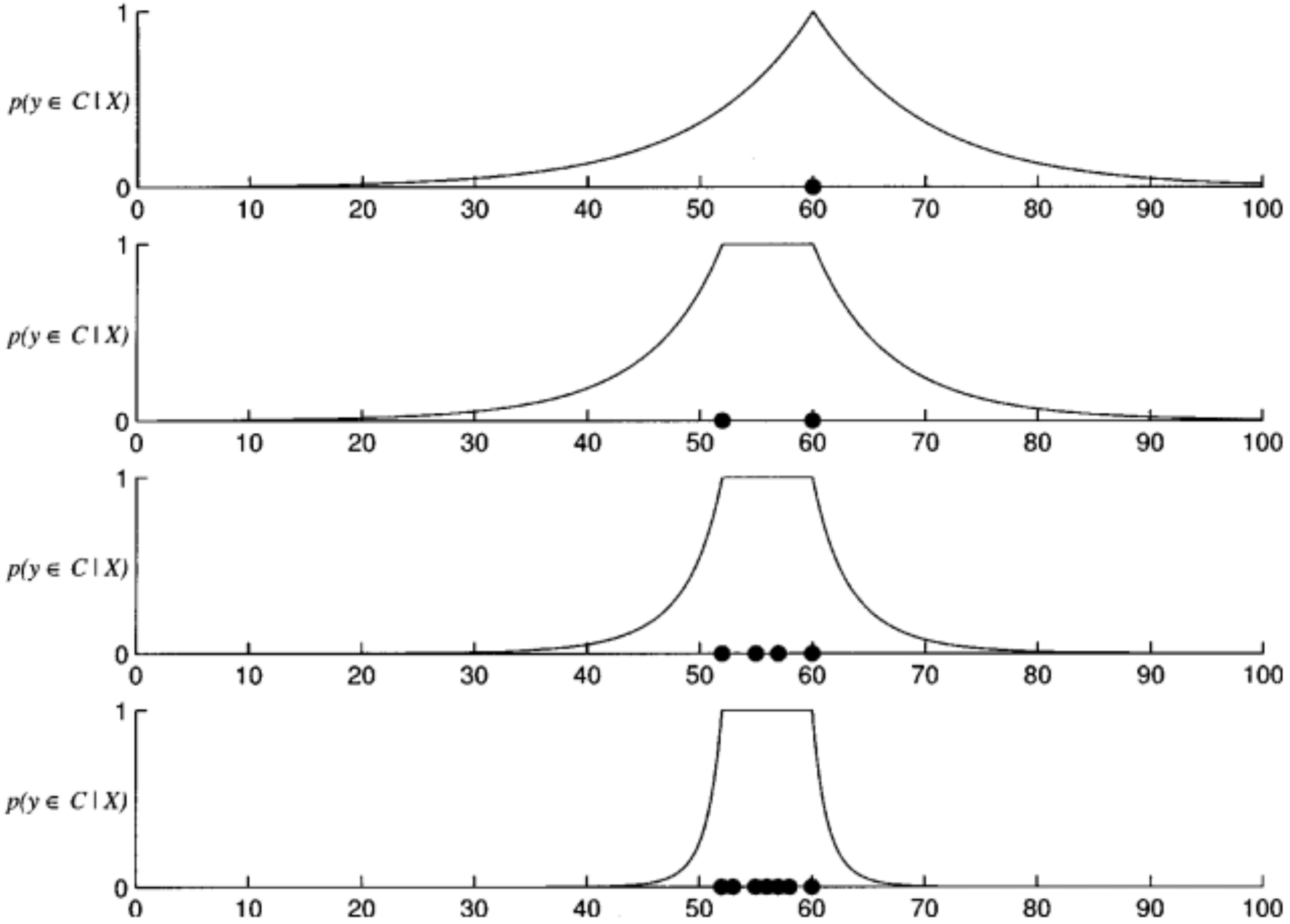
(equivalent when α=0 and β=1)

Bayesian concept learning

$$p(y \in C | x) = \sum_{h:y \in h} p(h|x).$$

$$= 1 \Big/ \left[1 + \frac{\sum_{h:x \in h, y \notin h} p(h,x)}{\sum_{h:x,y \in h} p(h,x)} \right].$$

Bayesian Concept Learning Extends Shepard's Law of Generalization to Multiple Examples



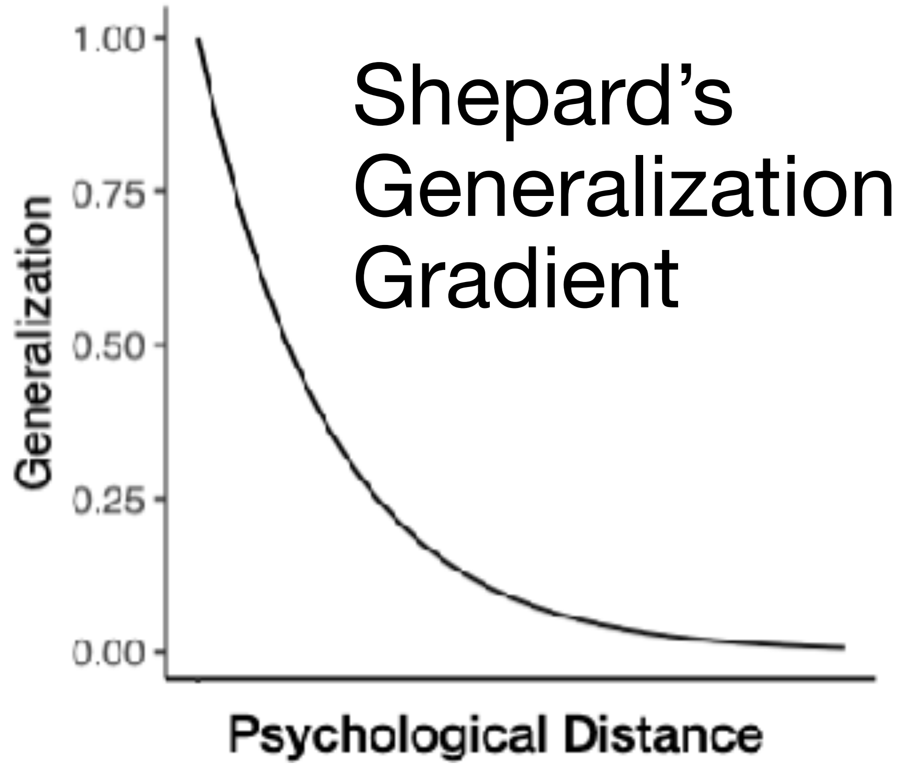
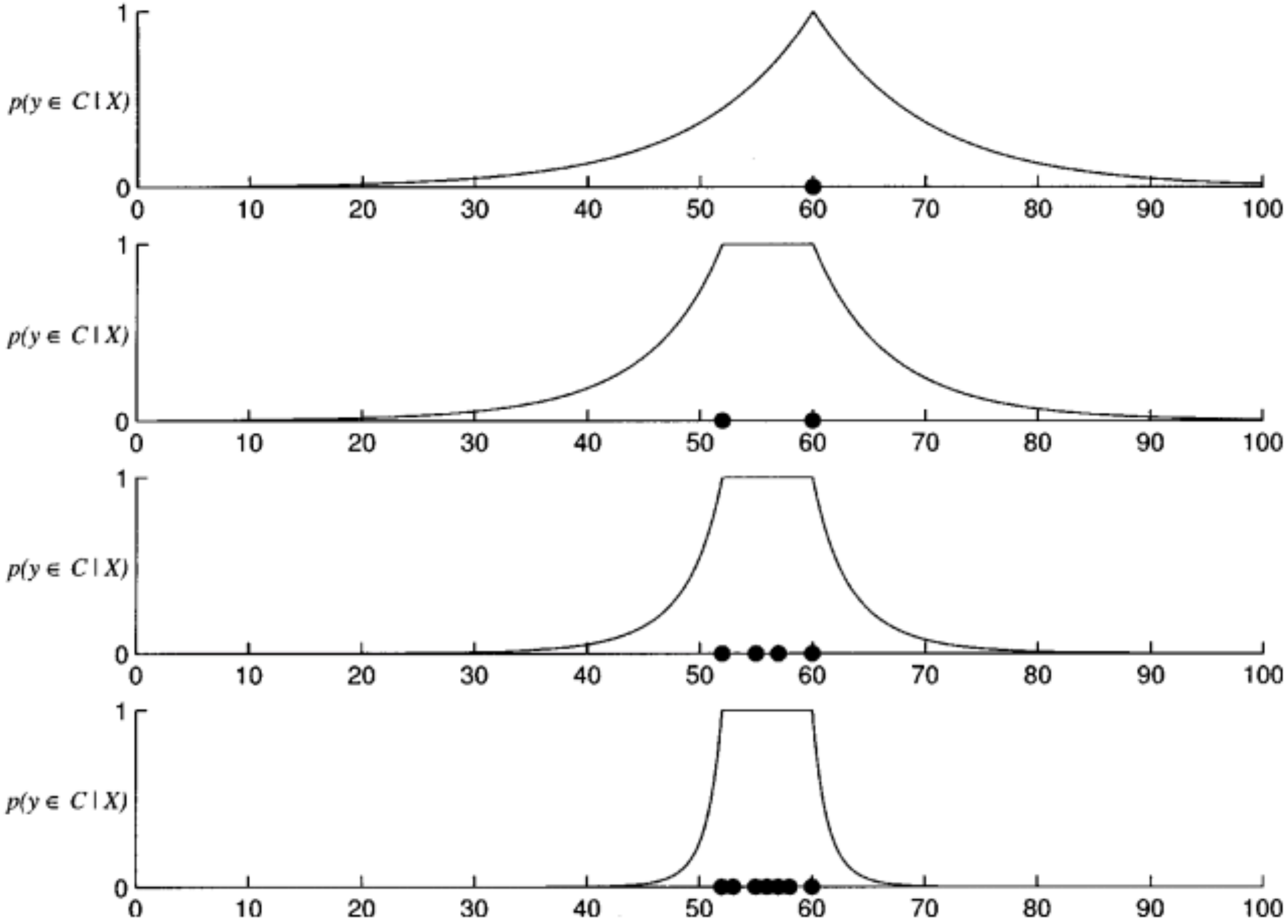
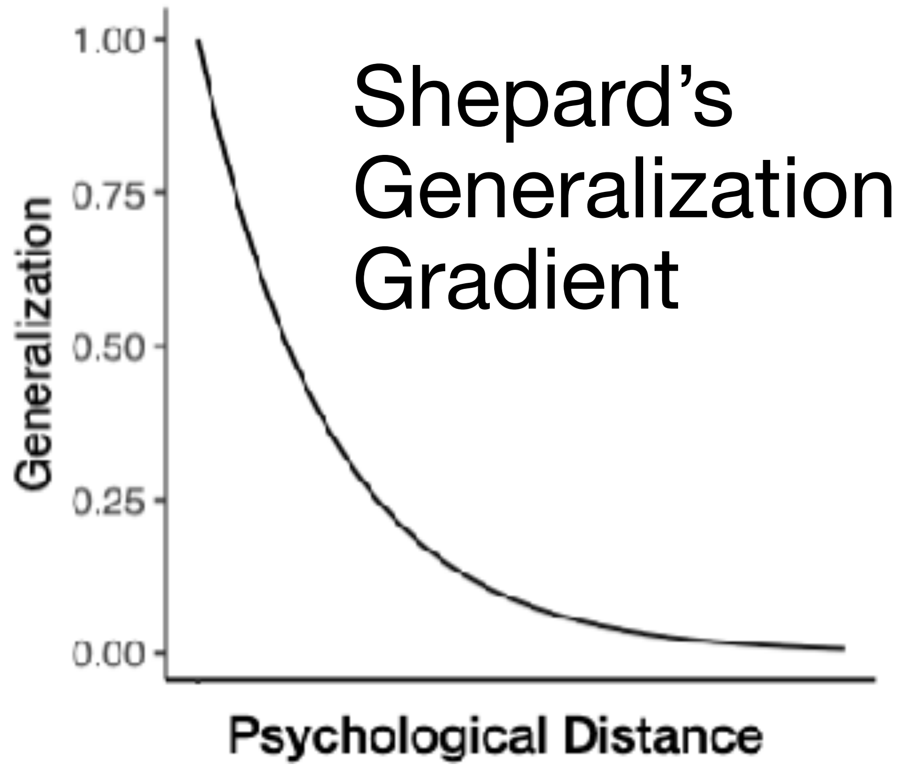


Figure 3. The effect of the number of examples on Bayesian generalization (under the assumptions of strong sampling and an Erlang prior, $\mu = 10$). Filled circles indicate examples. The first curve is the gradient of generalization with a single example, for the purpose of comparison. The remaining graphs show that the range of generalization decreases as a function of the number of examples.

Bayesian Concept Learning Extends Shepard's Law of Generalization to Multiple Examples



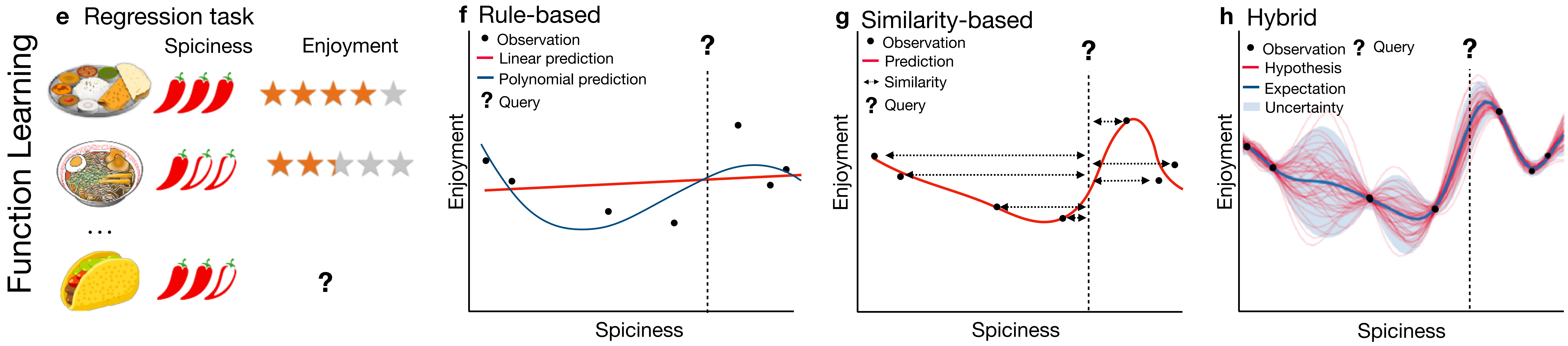


Range of generalization decreases with more examples

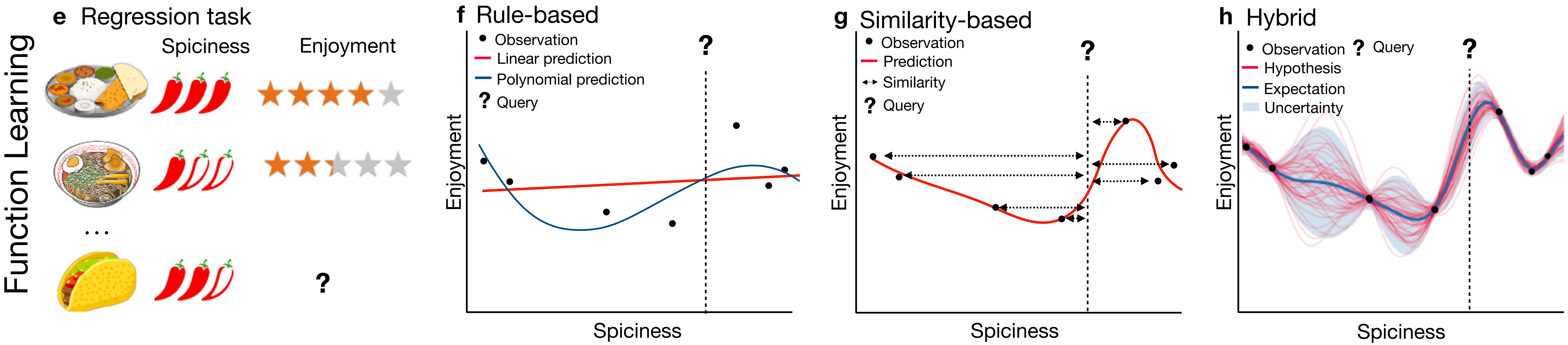
more examples = less uncertainty about the extent of consequential region

Figure 3. The effect of the number of examples on Bayesian generalization (under the assumptions of strong sampling and an Erlang prior, $\mu = 10$). Filled circles indicate examples. The first curve is the gradient of generalization with a single example, for the purpose of comparison. The remaining graphs show that the range of generalization decreases as a function of the number of examples.

Learning functions



Learning functions



- *Rules* describe an explicit parametric family of candidate functions (e.g., linear or polynomial)
  (Carroll, 1963; Brehmer, 1976)

- *Similarity* uses the generic principle that similar inputs produce similar outputs (often learned using ANNs) as the basis of generalization
  (McClelland et al., 1986; Busemeyer et al., 1997)

- *Hybrids* combine elements of both: Gaussian process (GP) regression uses kernel similarity to learn a distribution over functions, and can compositionally combine kernels like we can combine multiple rules
  (Rasmussen & Williams, 2005; Mercer, PhilTransRoySoc 1909; Lucas et al., PBR 2015)

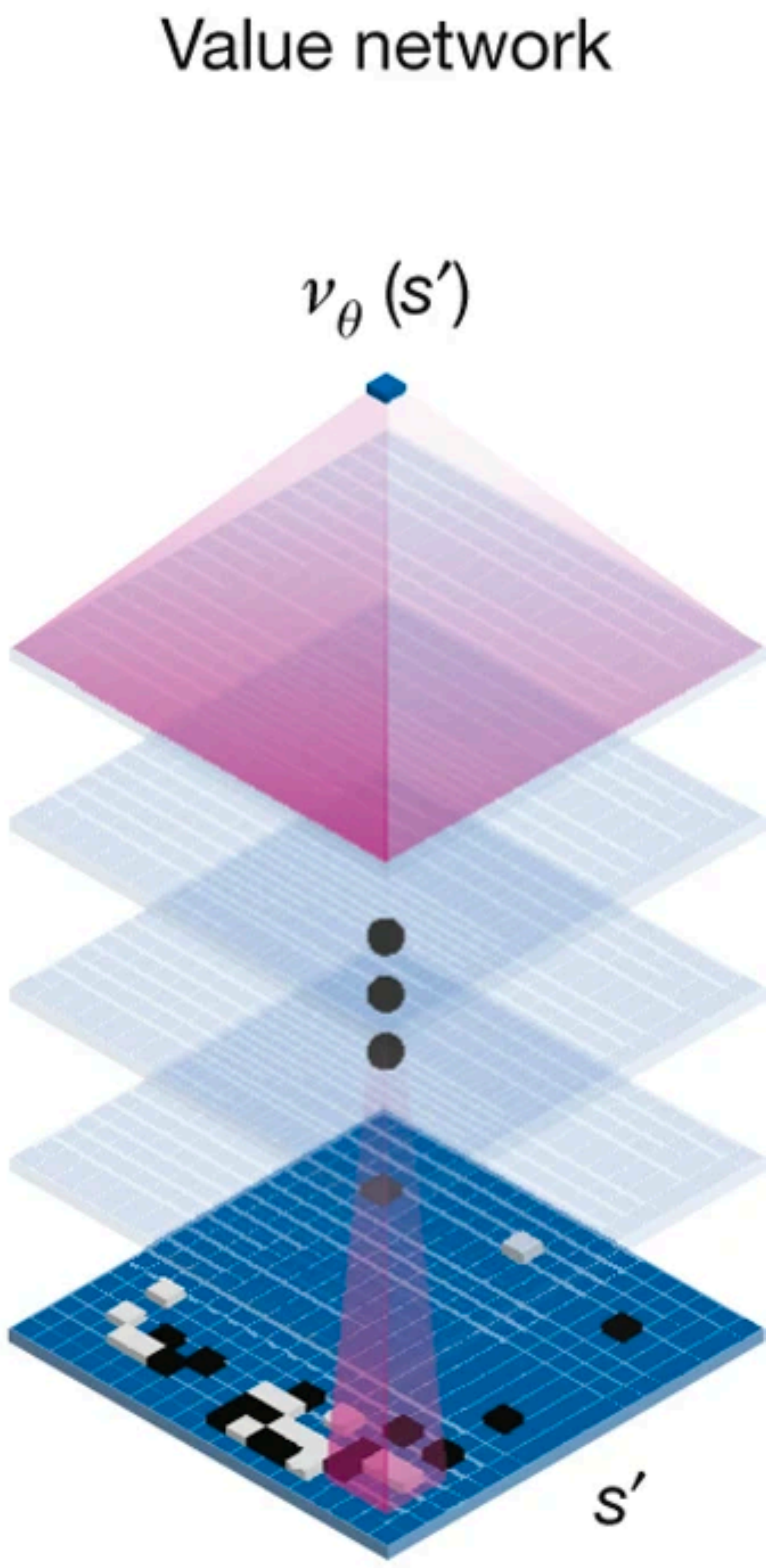
Value function approximation in RL

- Value function approximation is a key method for generalization in RL
    - Use function learning mechanisms for inferring *implicit* value of novel states: $V(s') = f(s')$
    - Implement a policy on the basis of value: $\pi(s') \propto \exp(V(\mathbf{s}'))$
- AlphaGo uses a deep neural network for value function approximation

Silver et al., (*Nature* 2016)

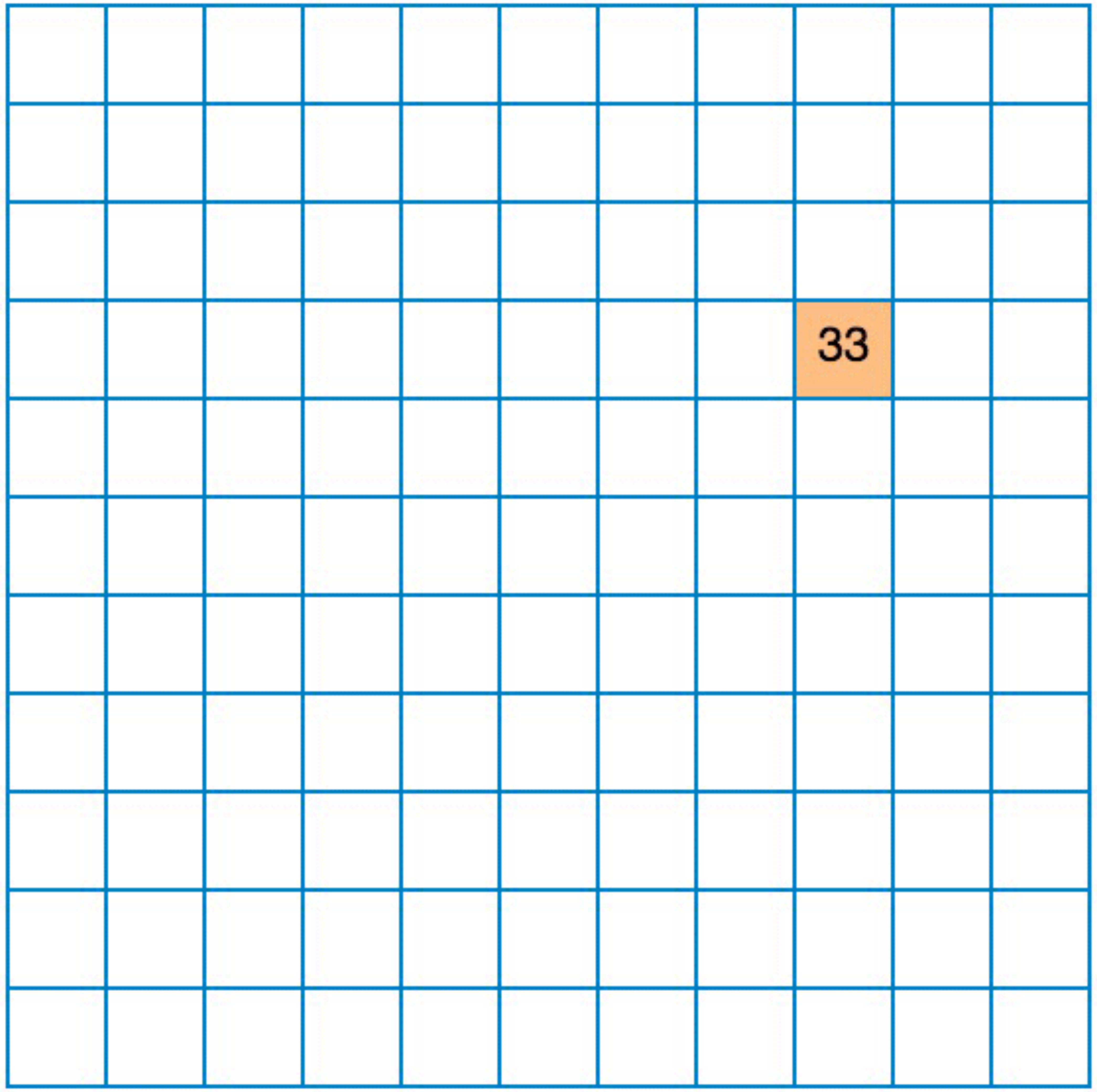
Value function approximation in RL

- Value function approximation is a key method for generalization in RL
    - Use function learning mechanisms for inferring *implicit* value of novel states: $V(s') = f(s')$
    - Implement a policy on the basis of value: $\pi(s') \propto \exp(V(\mathbf{s}'))$
- AlphaGo uses a deep neural network for value function approximation



Silver et al., (*Nature* 2016)

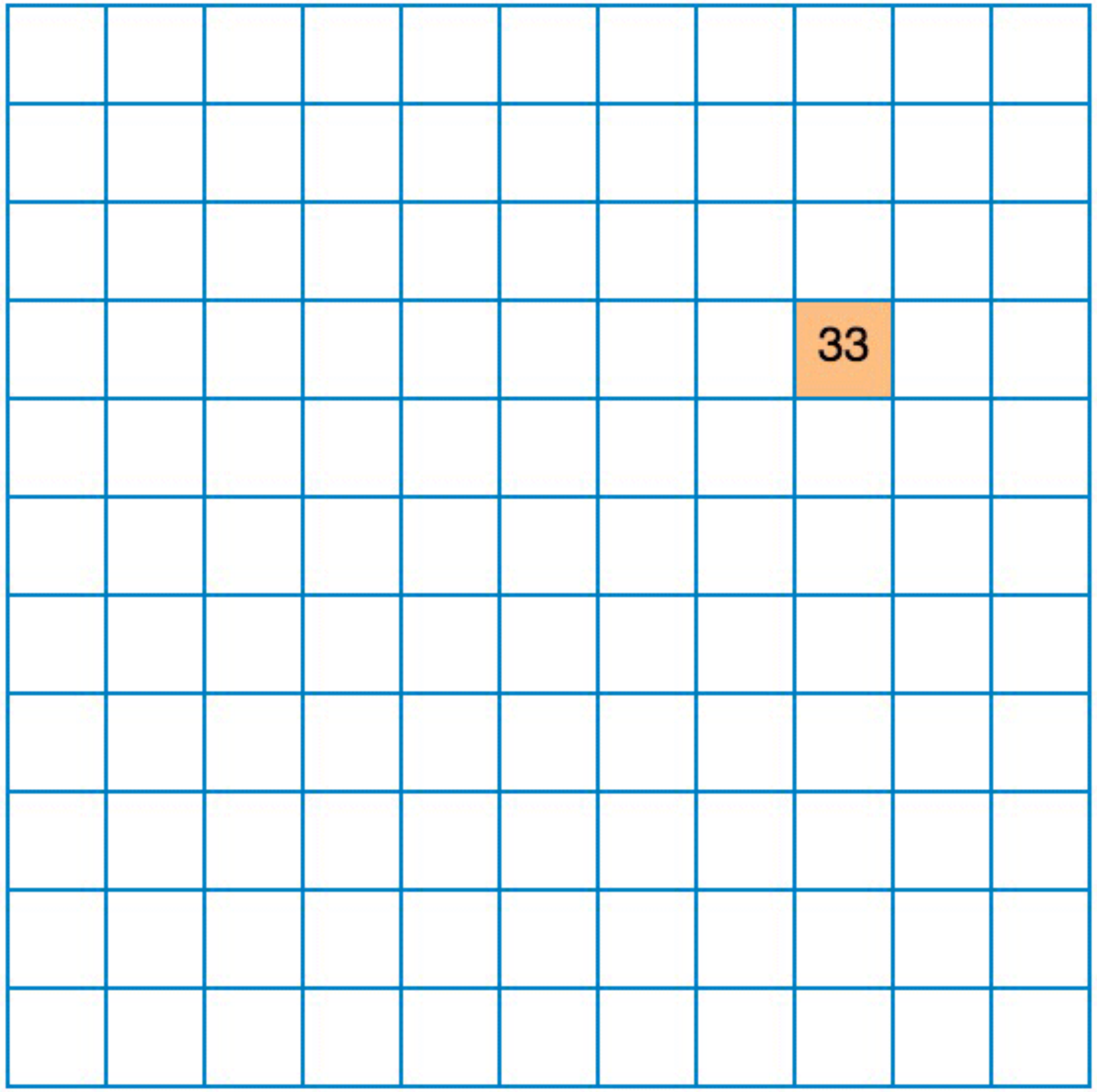
Spatially Correlated Bandit



Wu et al., (*Nature Human Behaviour* 2018)

- click tiles on the grid
- maximize reward
- each tile has normally distributed rewards
- limited search horizon
- nearby tiles have similar rewards

Spatially Correlated Bandit



Wu et al., (*Nature Human Behaviour* 2018)

- click tiles on the grid
- maximize reward
- each tile has normally distributed rewards
- limited search horizon
- nearby tiles have similar rewards

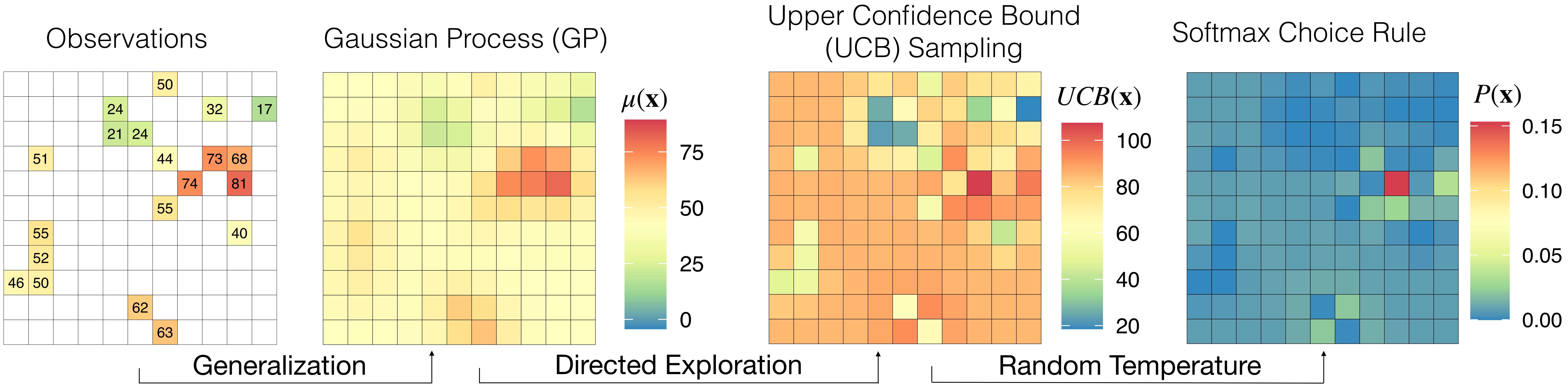
Spatially Correlated Bandit

7	5	10	22	32	32	28	24	22	26	33
6	11	19	29	38	41	42	40	37	36	40
22	27	30	35	43	50	53	53	51	49	46
45	44	38	36	40	46	47	49	54	55	48
61	55	46	40	37	32	27	31	44	52	44
62	59	57	54	44	27	14	17	33	46	45
53	59	68	71	59	36	17	15	28	45	51
46	57	71	77	67	47	26	18	27	45	56
45	56	65	67	60	46	29	20	27	42	55
51	57	58	53	47	40	30	23	28	40	49
60	62	58	47	39	38	35	31	35	41	46

Wu et al., (*Nature Human Behaviour* 2018)

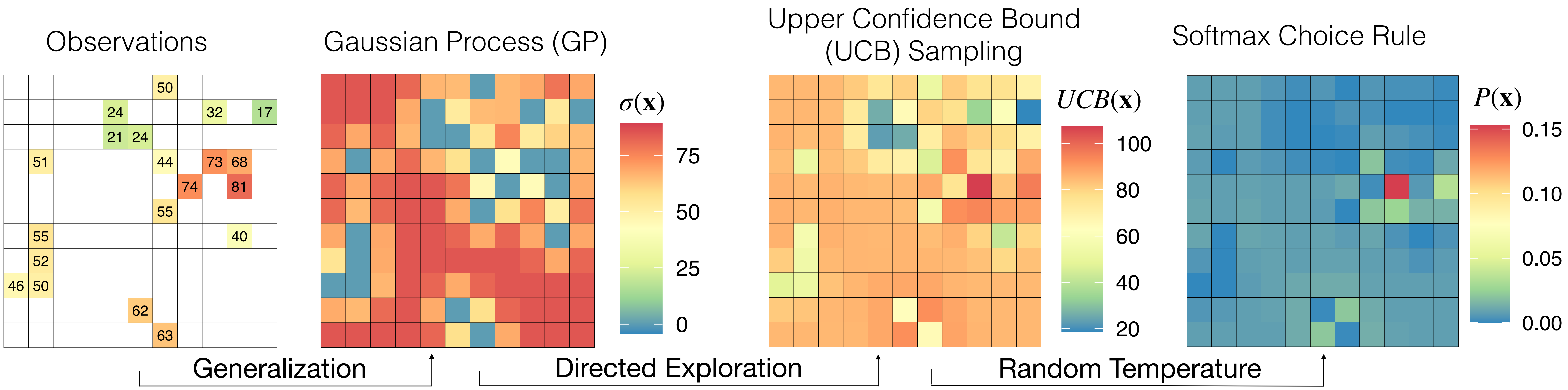
- click tiles on the grid
- maximize reward
- each tile has normally distributed rewards
- limited search horizon
- nearby tiles have similar rewards

GP-UCB Model



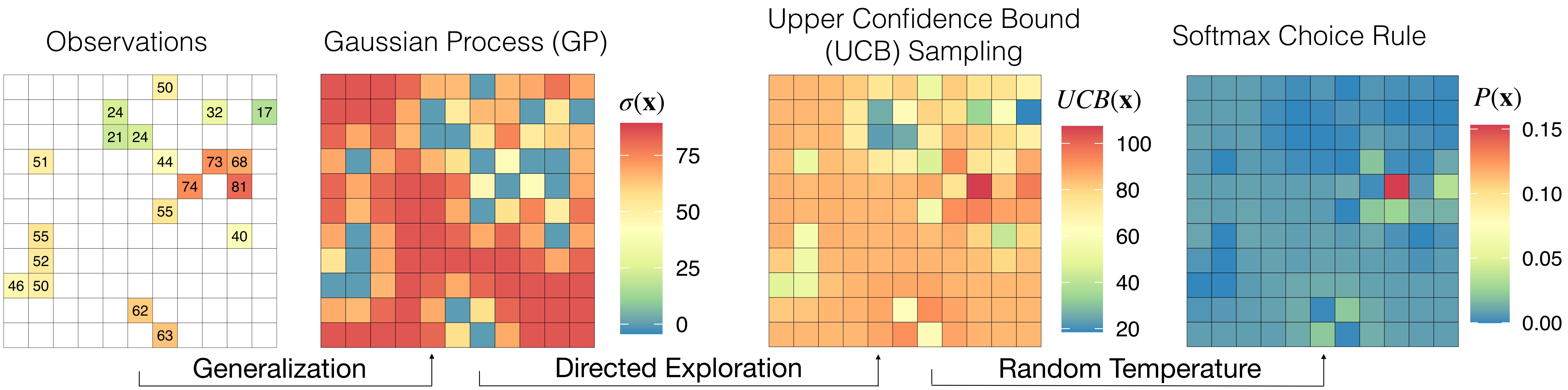
Observations
Gaussian Process (GP)
Upper Confidence Bound
(UCB) Sampling
Softmax Choice Rule
50
24
32
17
21
24
51
44
73
68
74
81
55
55
40
52
46
50
62
63
$\mu(\mathbf{x})$
75
50
25
0
$UCB(\mathbf{x})$
100
80
60
40
20
$P(\mathbf{x})$
0.15
0.10
0.05
0.00
Generalization
Directed Exploration
Random Temperature

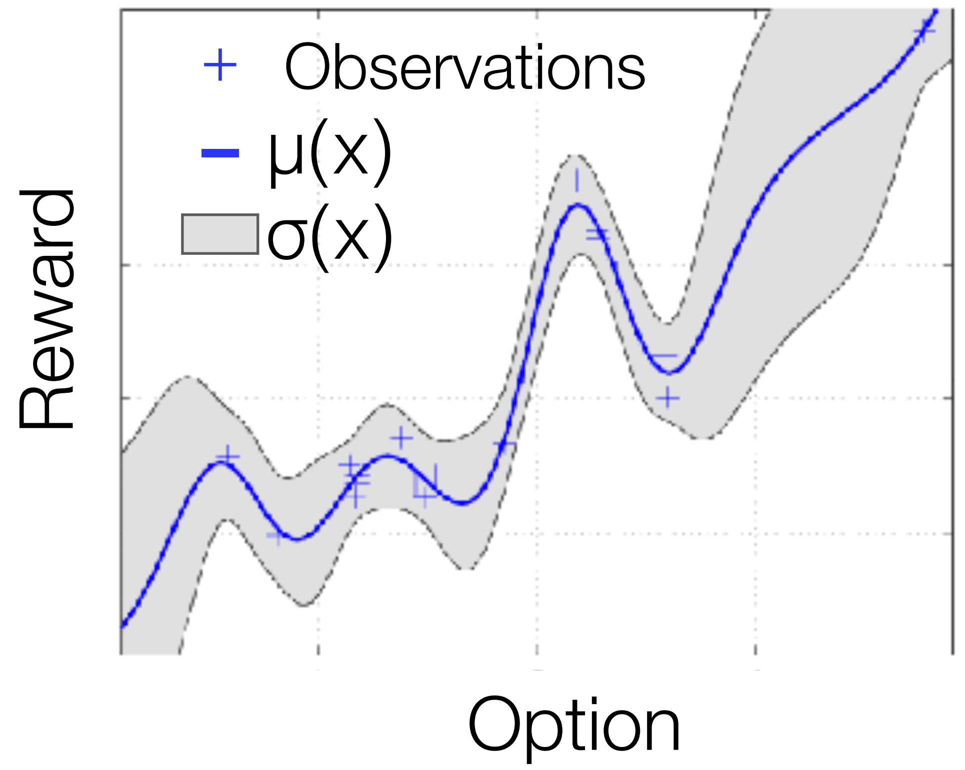
GP-UCB Model



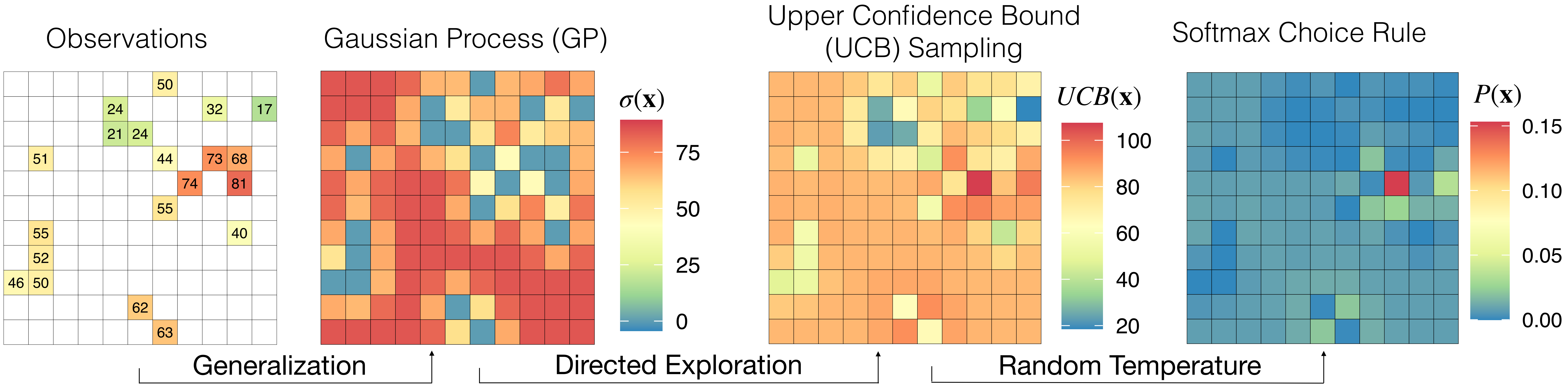
Observations
Gaussian Process (GP)
Upper Confidence Bound (UCB) Sampling
Softmax Choice Rule
50
24
32
17
21
24
51
44
73
68
74
81
55
55
40
52
46
50
62
63
$\sigma(\mathbf{x})$
75
50
25
0
$UCB(\mathbf{x})$
100
80
60
40
20
$P(\mathbf{x})$
0.15
0.10
0.05
0.00
Generalization
Directed Exploration
Random Temperature

GP-UCB Model

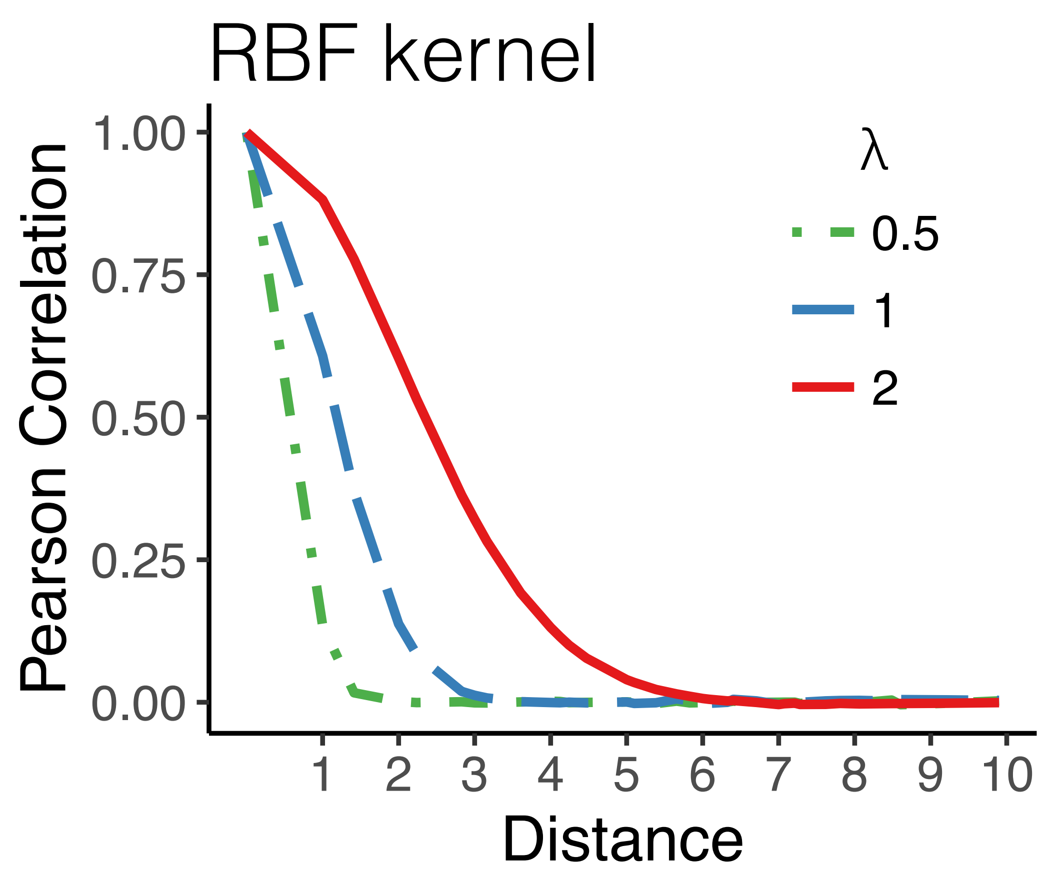

Observations
Gaussian Process (GP)
Upper Confidence Bound (UCB) Sampling
Softmax Choice Rule
σ(x)
75
50
25
0
UCB(x)
100
80
60
40
20
P(x)
0.15
0.10
0.05
0.00
Generalization
Directed Exploration
Random Temperature


Observations
μ(x)
σ(x)
Reward
Option

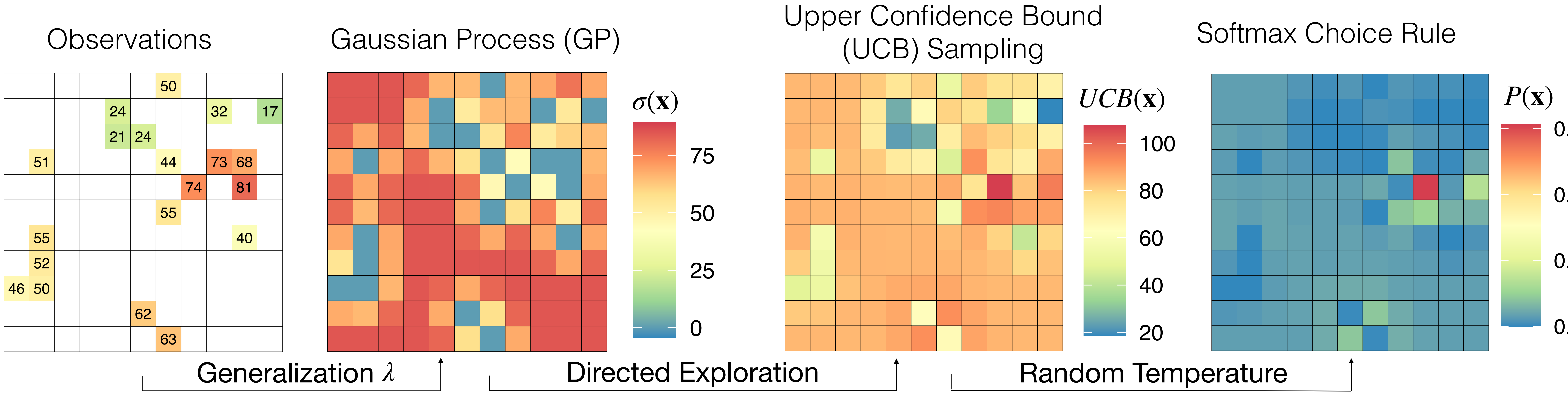
GP-UCB Model



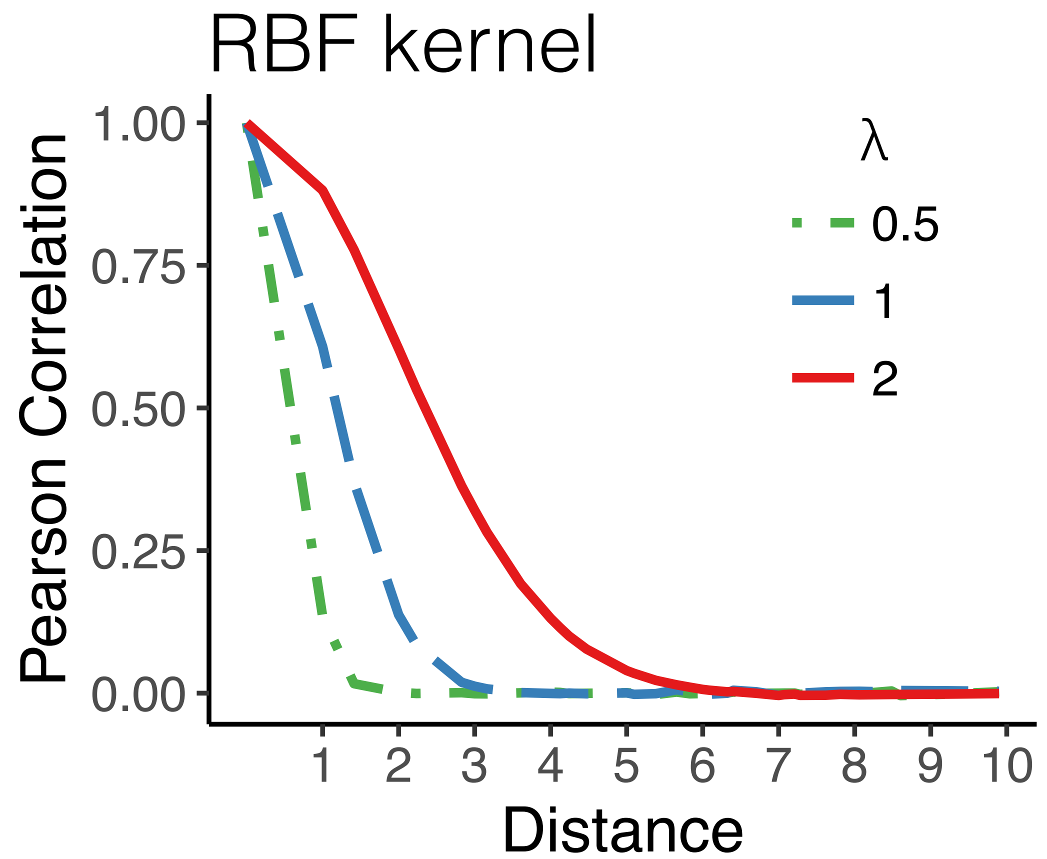
$$k_{RBF}(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{||\mathbf{x} - \mathbf{x}'||^2}{2\lambda^2}\right)$$



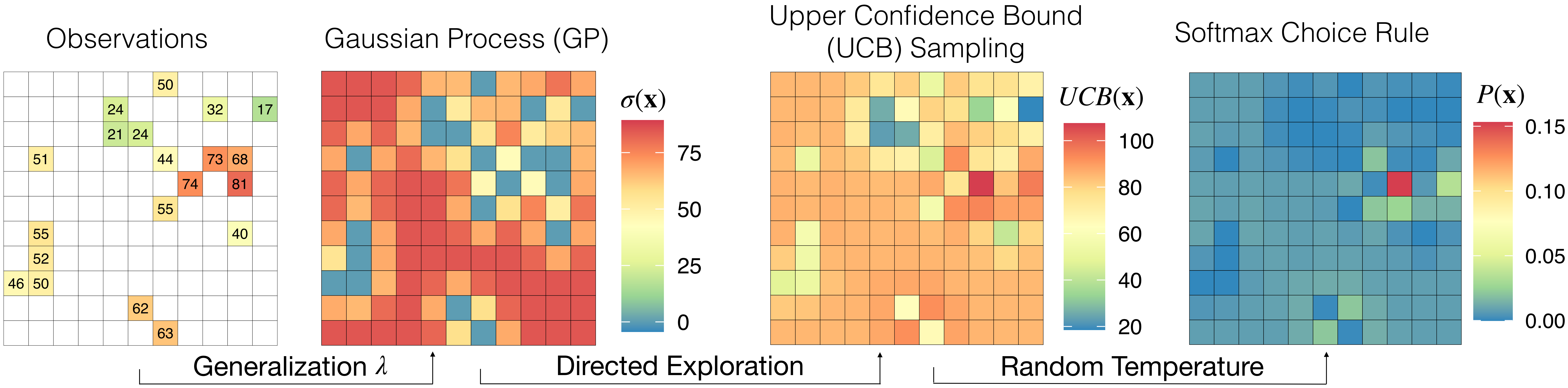
GP-UCB Model



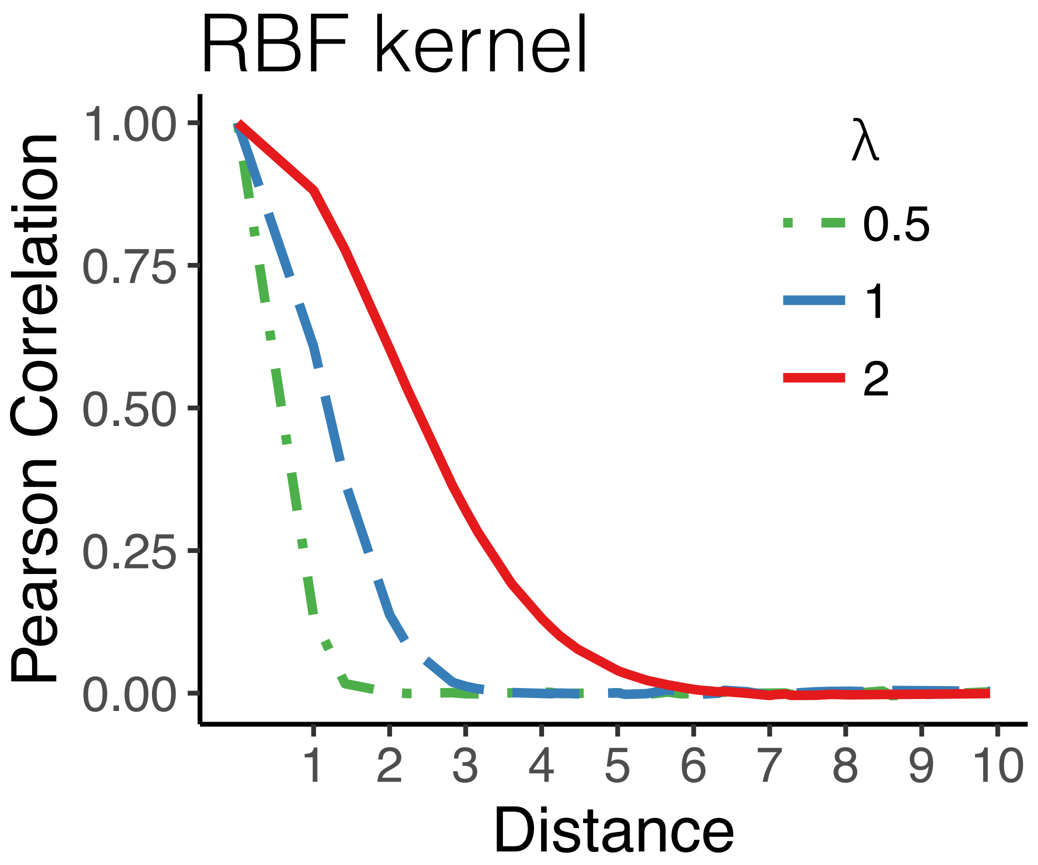
$$k_{RBF}(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{||\mathbf{x} - \mathbf{x}'||^2}{2\lambda^2}\right)$$

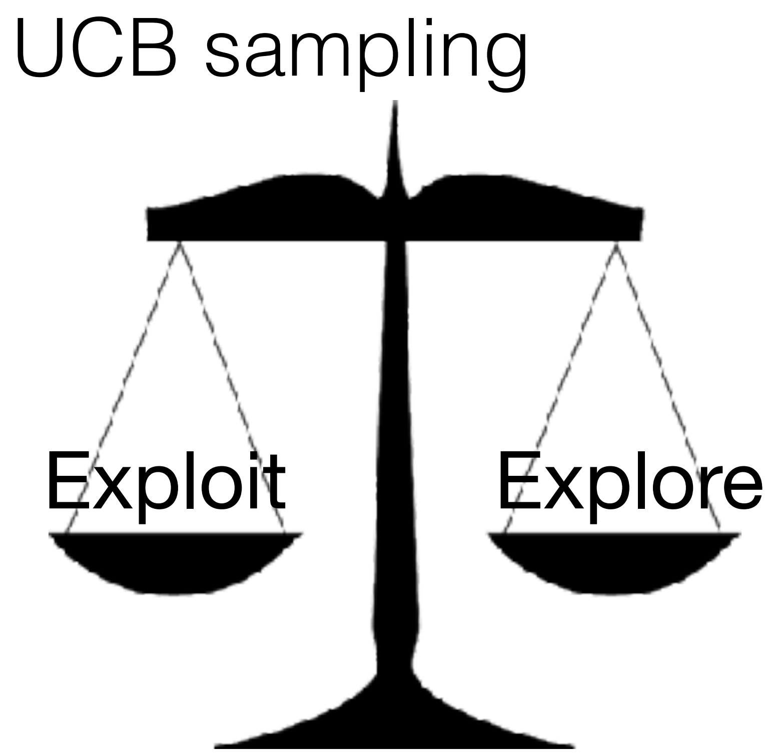


GP-UCB Model

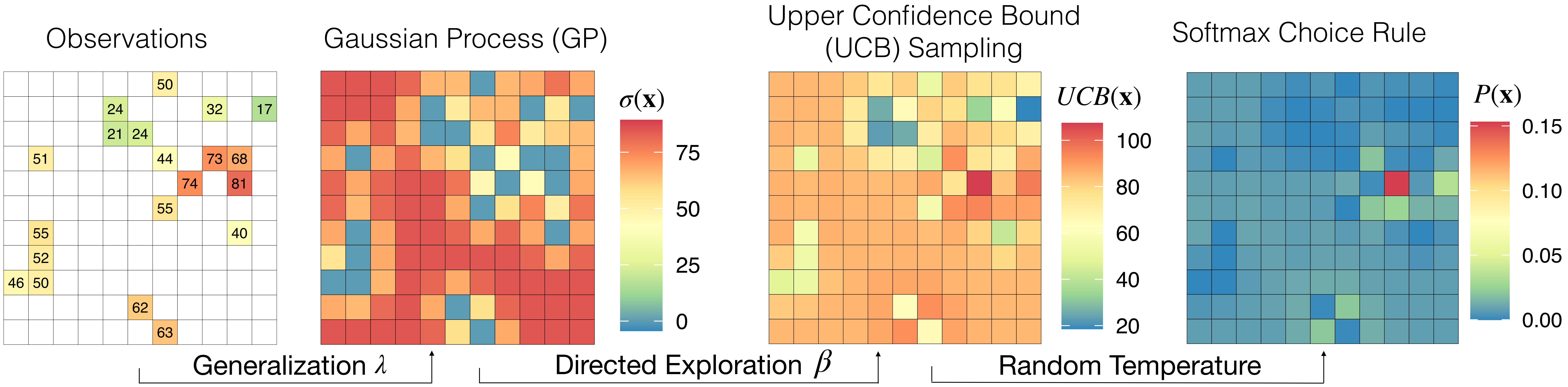


$$k_{RBF}(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{||\mathbf{x} - \mathbf{x}'||^2}{2\lambda^2}\right) \qquad UCB(\mathbf{x}) = \mu(\mathbf{x}) + \beta\sigma(\mathbf{x})$$

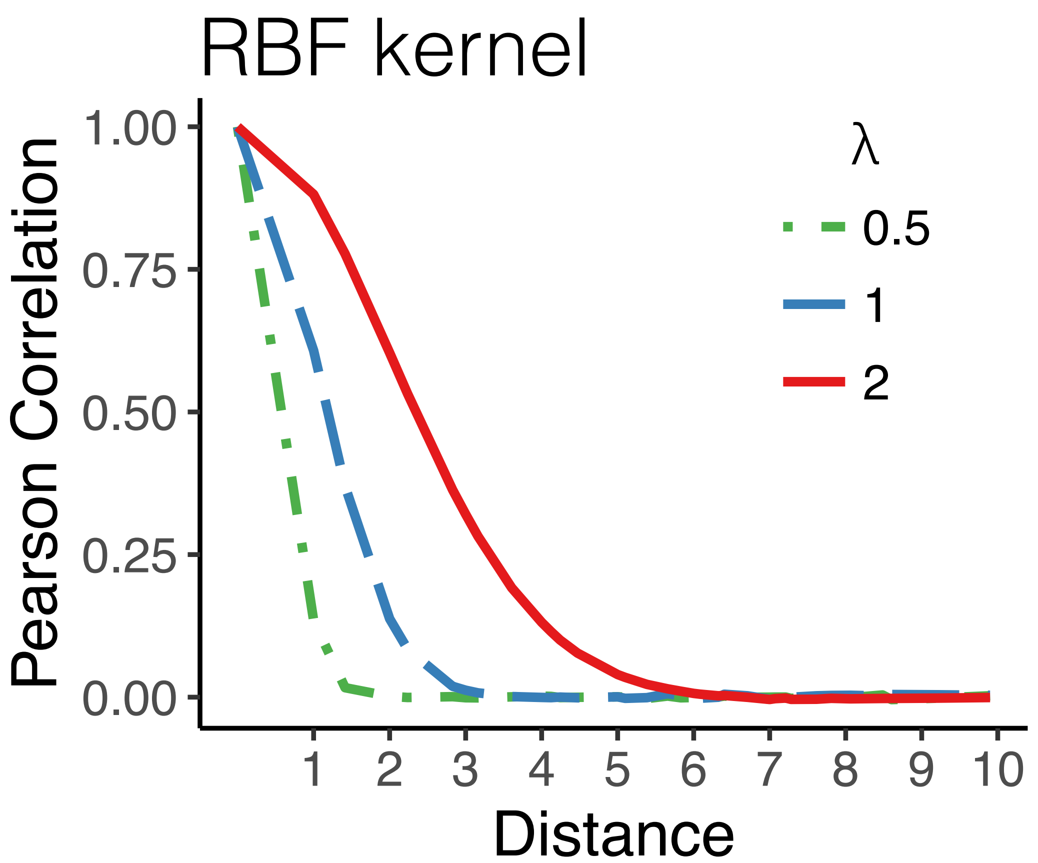


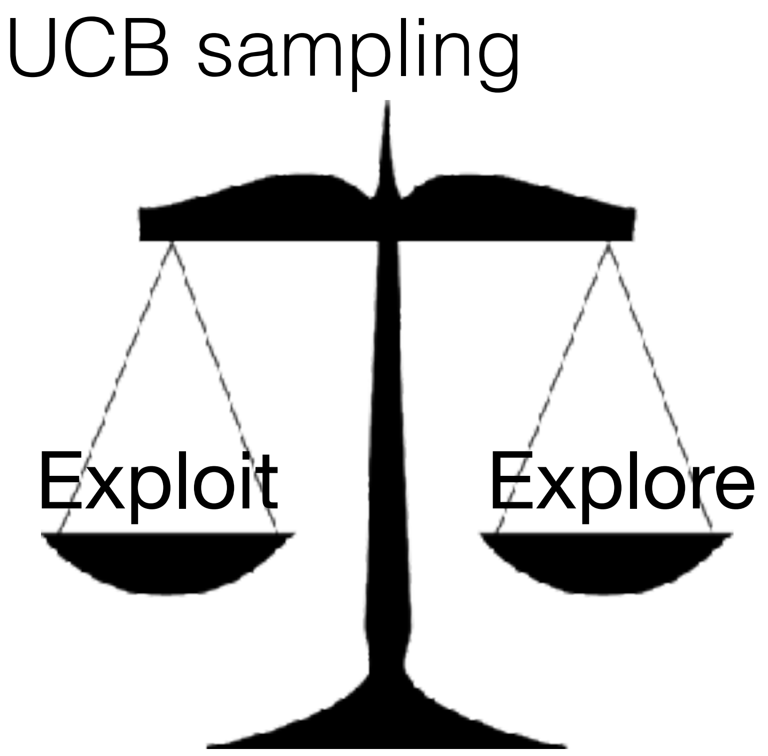


GP-UCB Model

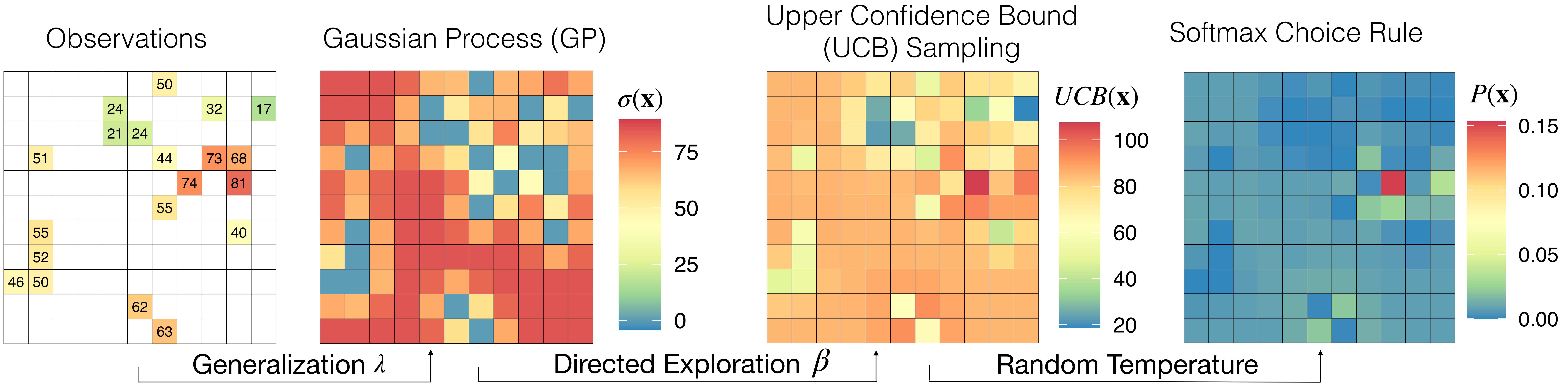


$$k_{RBF}(\mathbf{x},\mathbf{x}') = \exp\left(-\frac{||\mathbf{x}-\mathbf{x}'||^2}{2\lambda^2}\right) \qquad UCB(\mathbf{x}) = \mu(\mathbf{x}) + \beta\sigma(\mathbf{x})$$

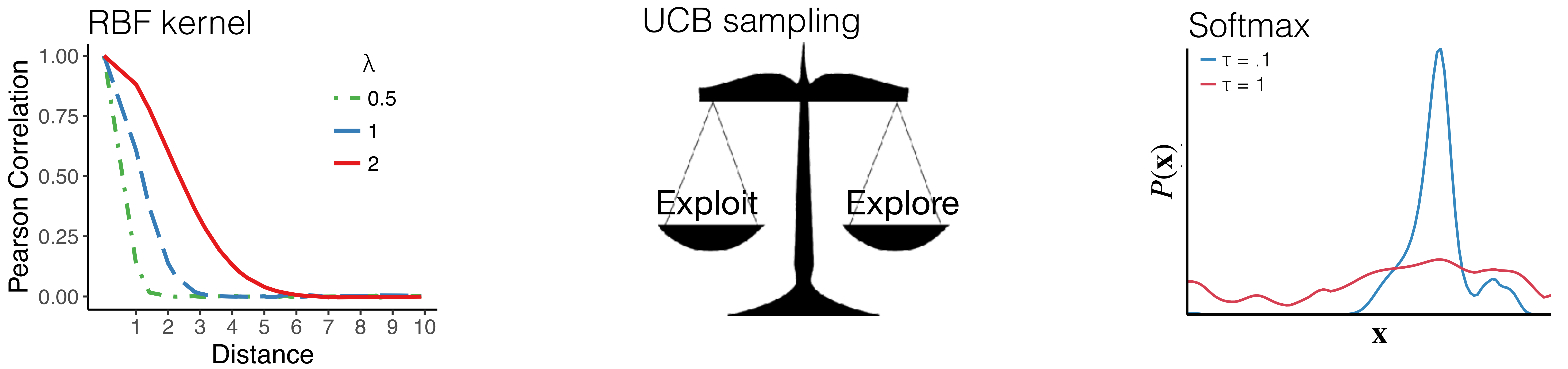




GP-UCB Model

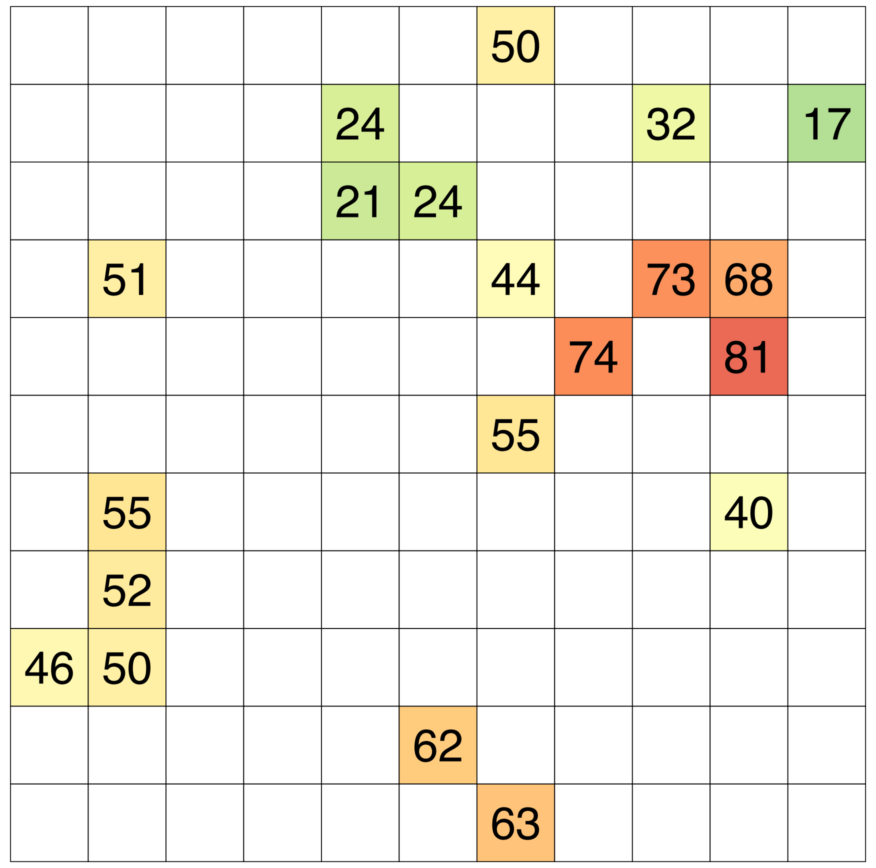


$$k_{RBF}(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{||\mathbf{x} - \mathbf{x}'||^2}{2\lambda^2}\right) \qquad UCB(\mathbf{x}) = \mu(\mathbf{x}) + \beta\sigma(\mathbf{x}) \qquad P(\mathbf{x}) \propto \exp(UCB(\mathbf{x})/\tau)$$

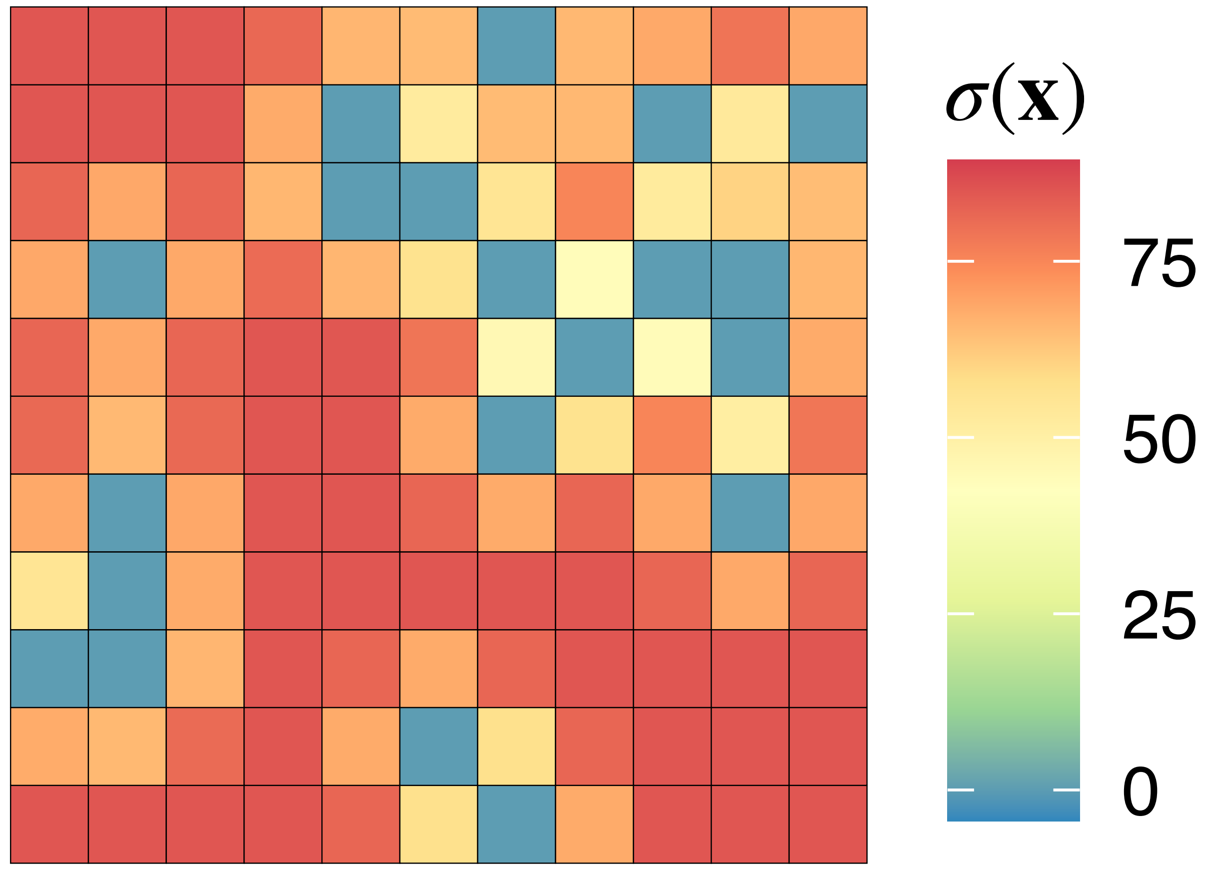


GP-UCB Model

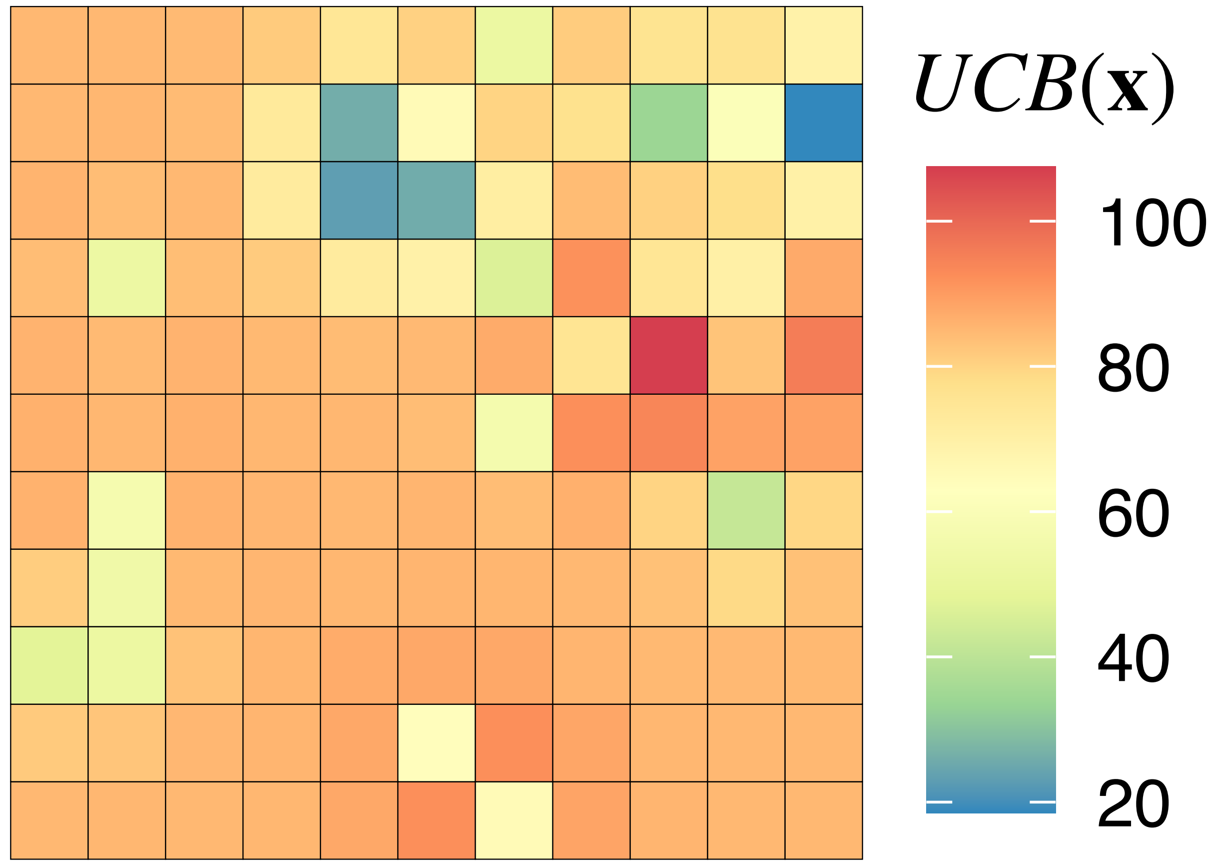
Observations



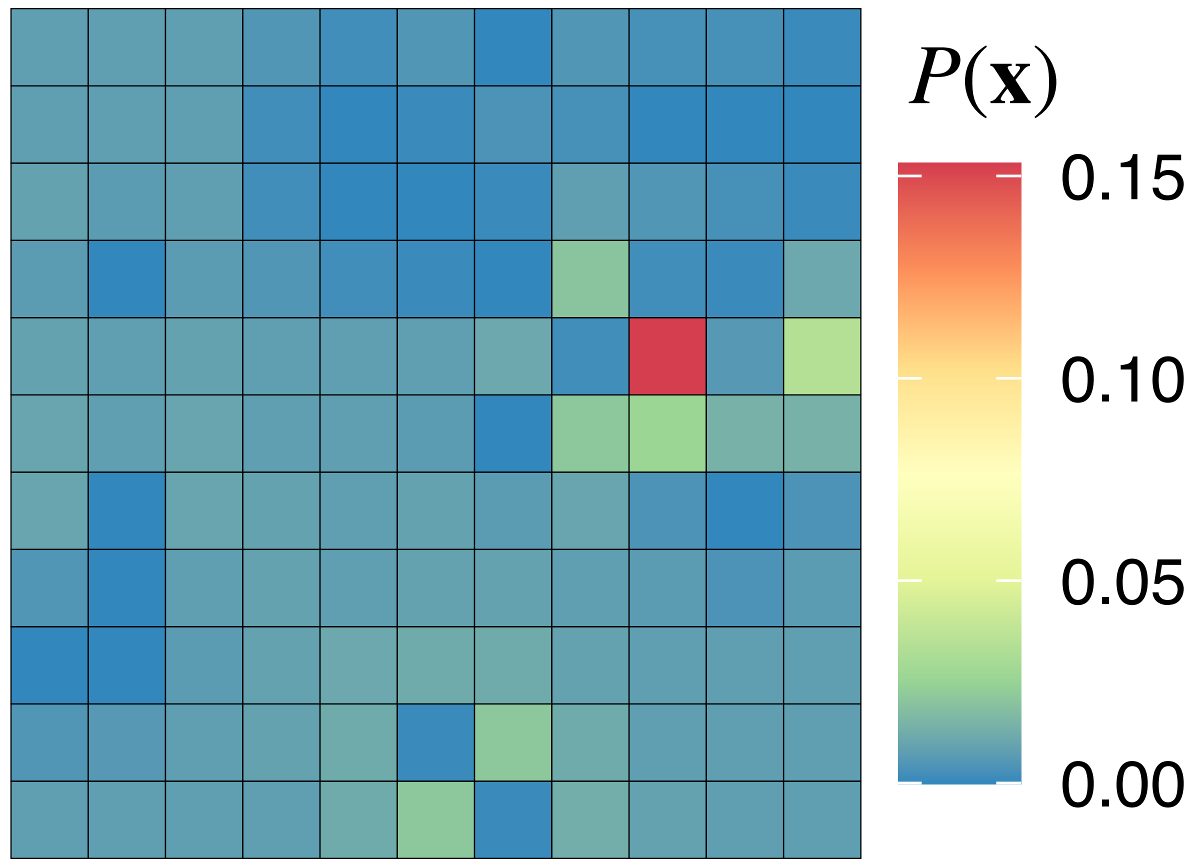
Gaussian Process (GP)



Upper Confidence Bound (UCB) Sampling



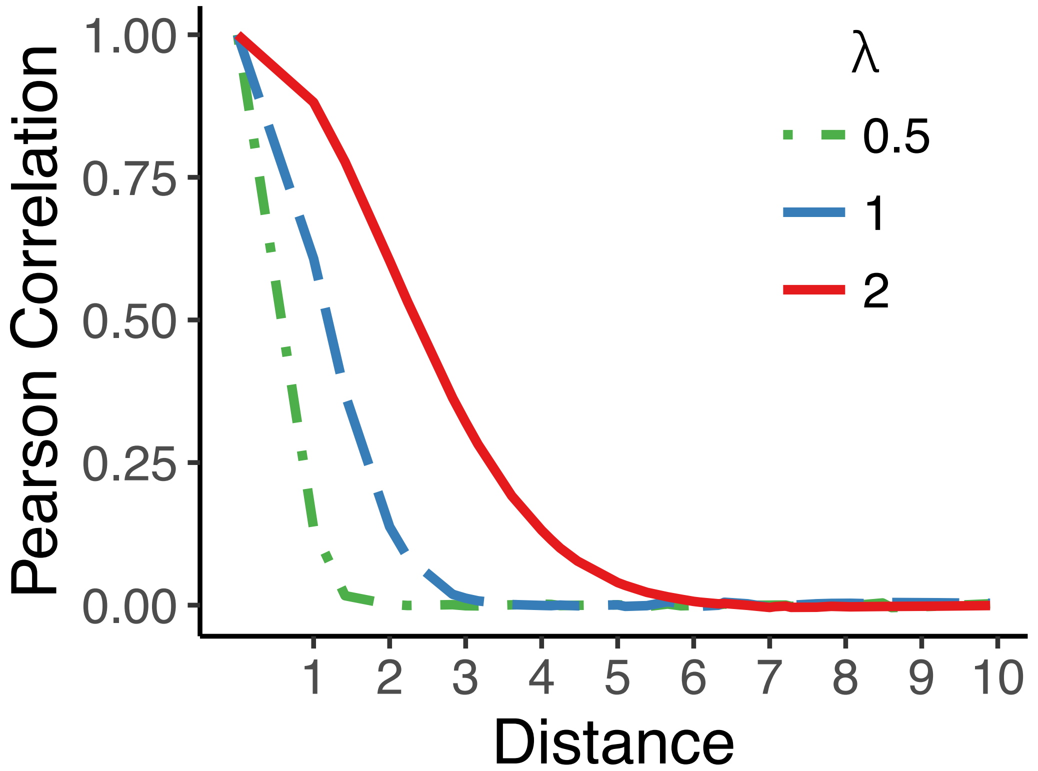
Softmax Choice Rule



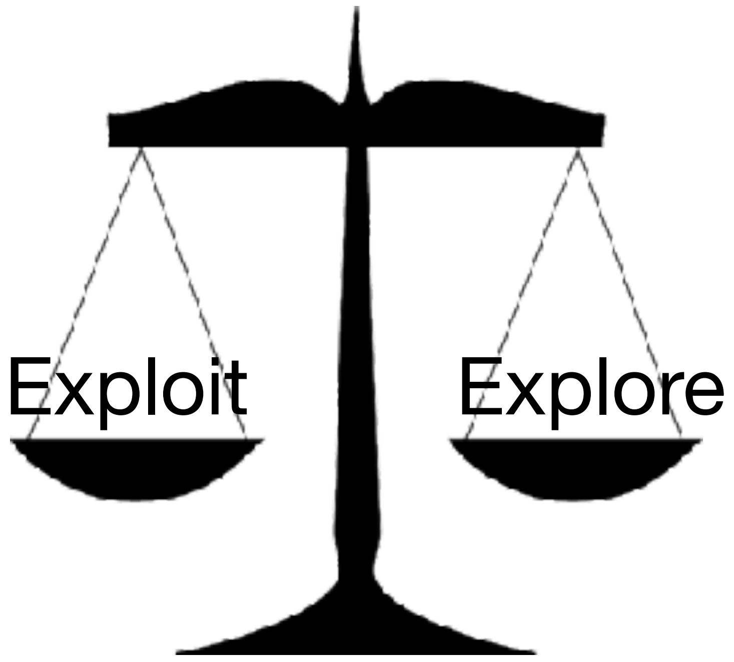
Generalization λ Directed Exploration β Random Temperature τ

$$k_{RBF}(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{||\mathbf{x} - \mathbf{x}'||^2}{2\lambda^2}\right) \qquad UCB(\mathbf{x}) = \mu(\mathbf{x}) + \beta\sigma(\mathbf{x}) \qquad P(\mathbf{x}) \propto \exp(UCB(\mathbf{x})/\tau)$$

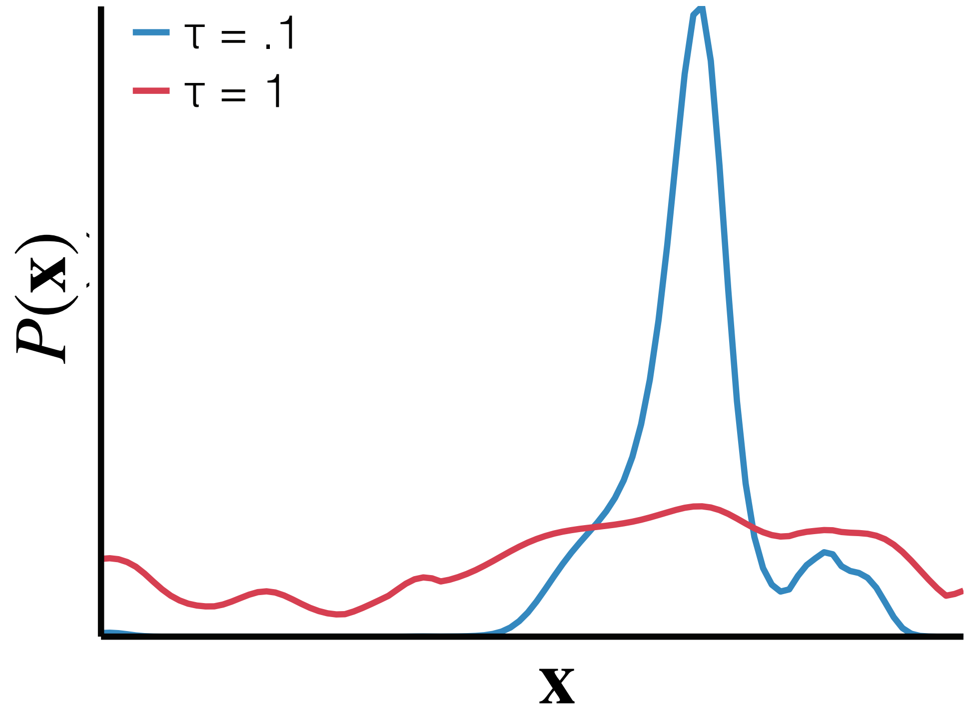
RBF kernel



UCB sampling

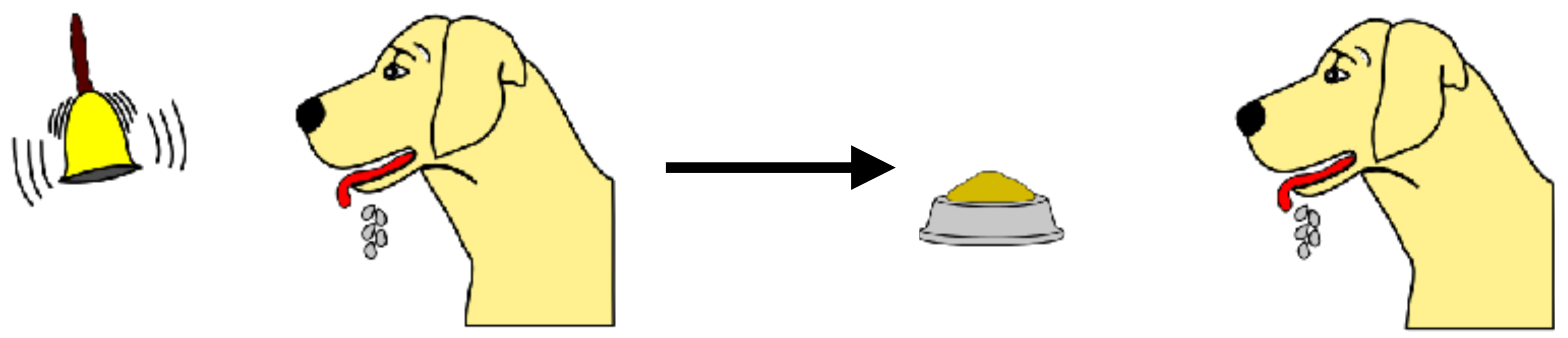


Softmax



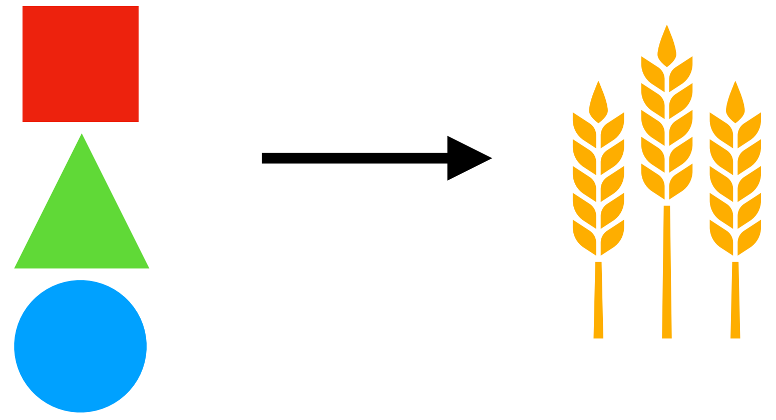
5 minute break

Pavlovian (classical)
conditioning



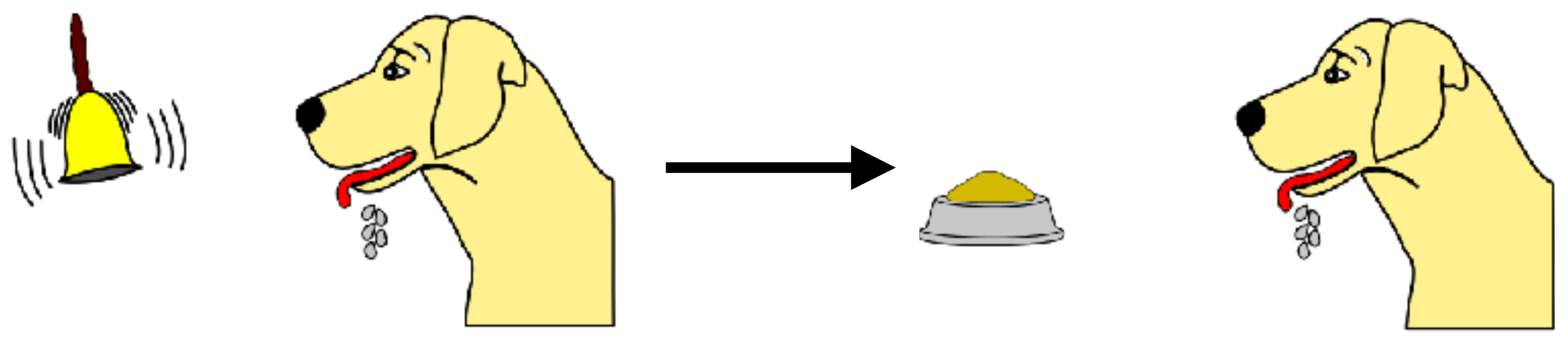
Learn which environmental cues *predict* reward

Operant (instrumental)
conditioning

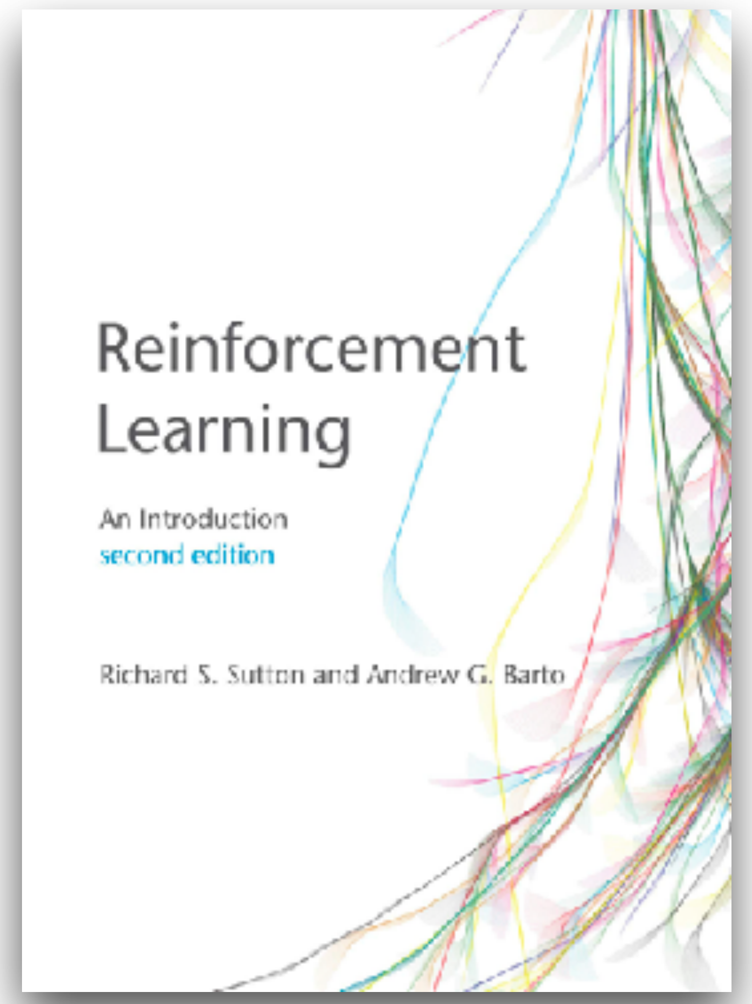


Learn which actions *predict* reward

Pavlovian (classical) conditioning

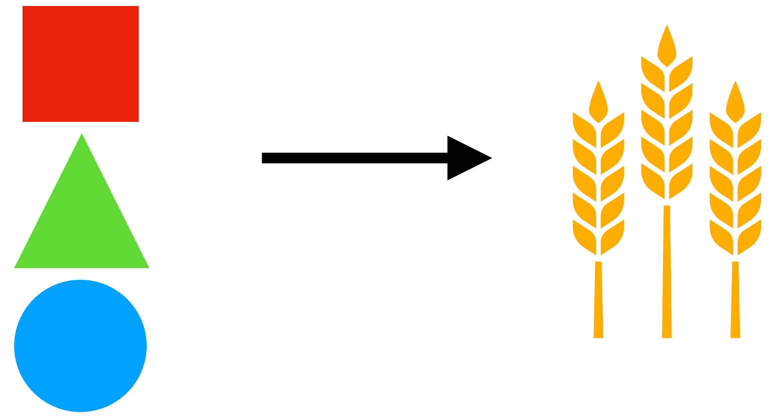


Learn which environmental cues *predict* reward



Reinforcement Learning

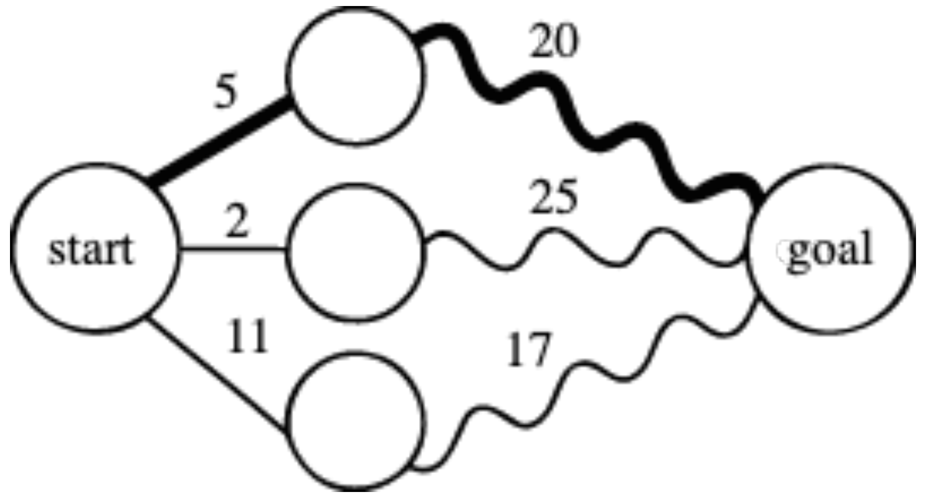
Operant (instrumental) conditioning



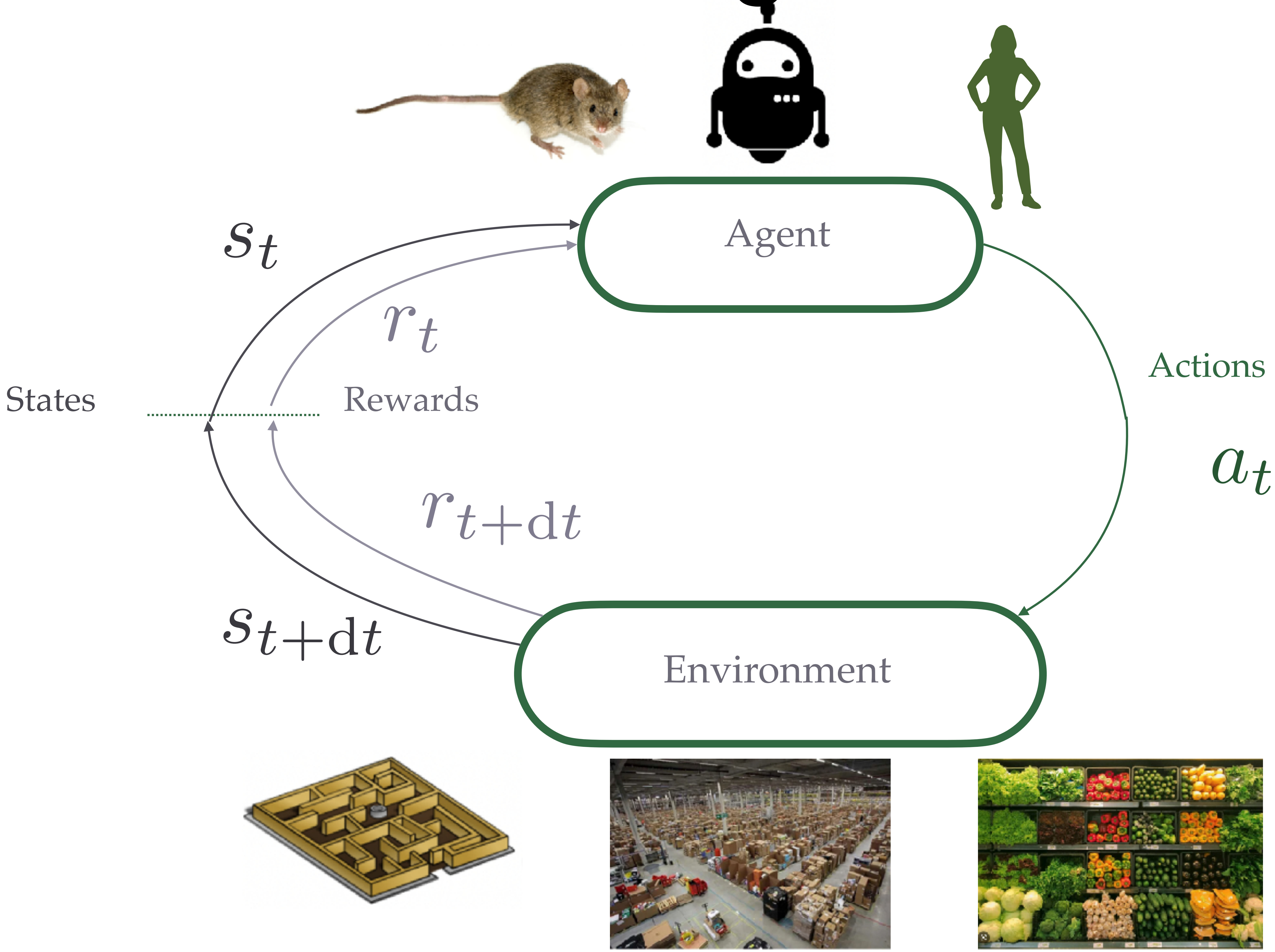
Learn which actions *predict* reward

Neuro-dynamic programing
Bertsekas & Tsitsiklis (1996)

Stochastic approximations to dynamic programing problems



Reinforcement Learning

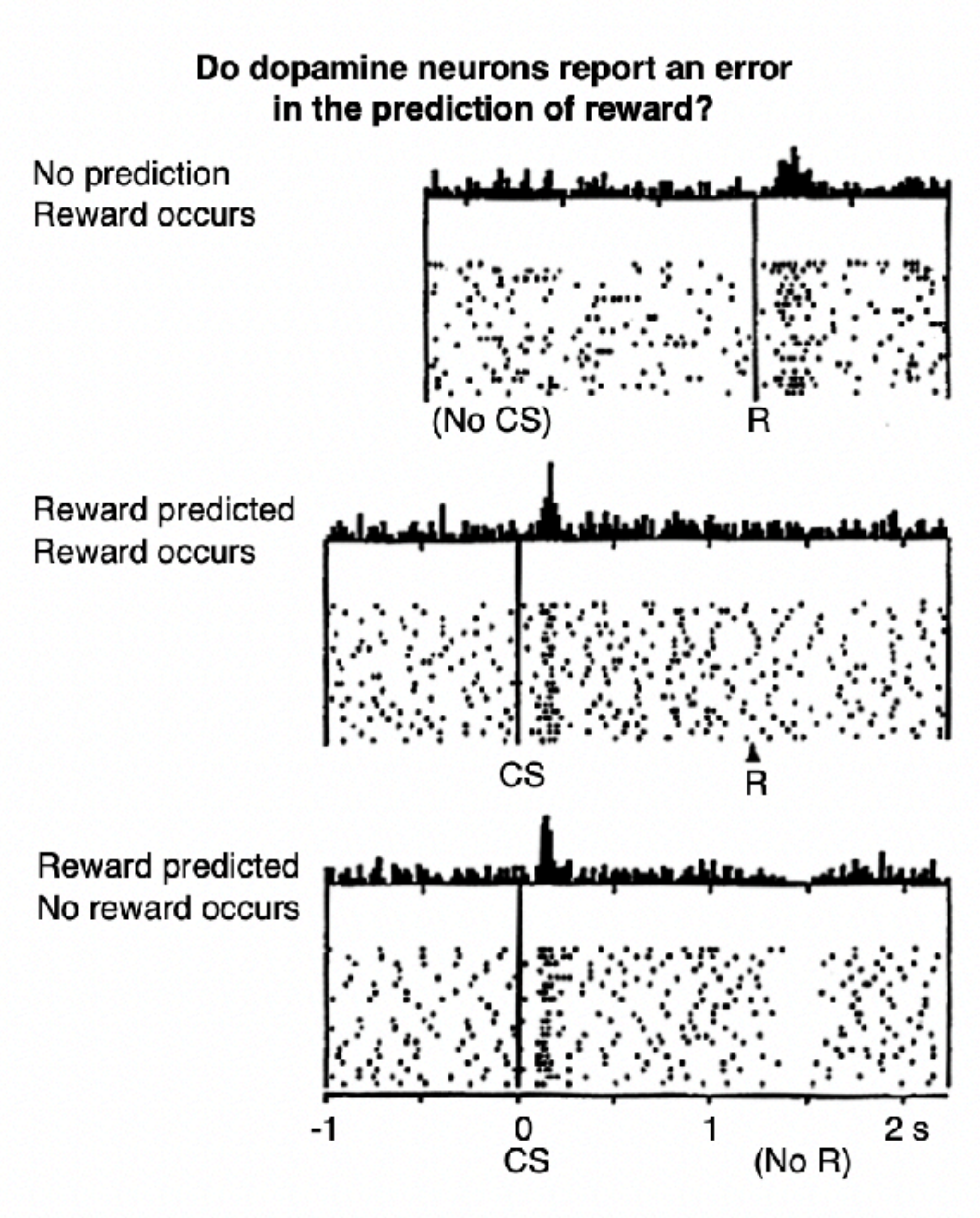


Sutton and Barto. *Reinforcement learning: An introduction*. MIT press. (2018)

Temporal Difference Error and Dopamine neurons:

RL enables to quantify prediction mechanisms:

- In behavioural science
- In neuroscience
- In psychiatry
- And of course in AI!

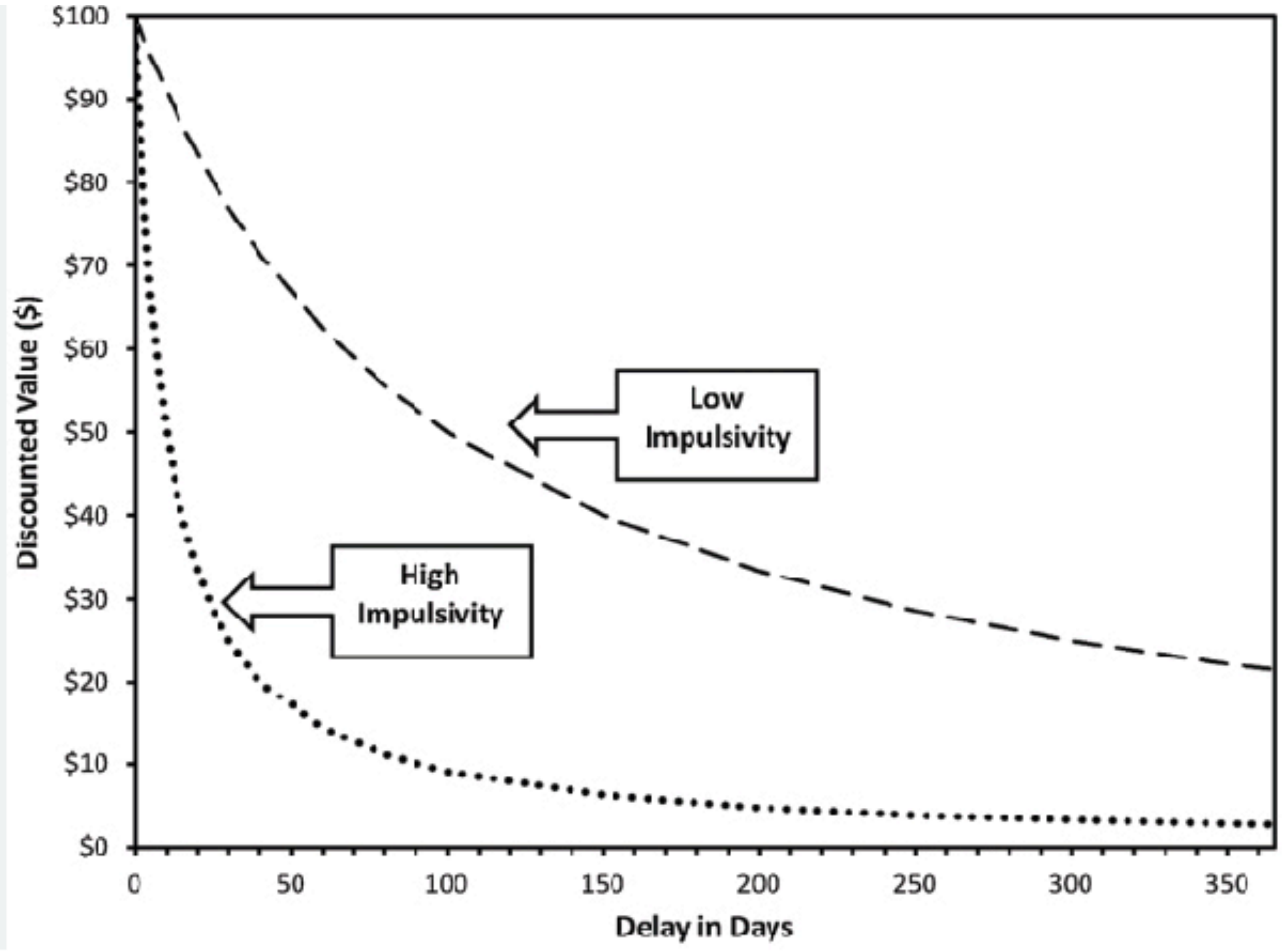


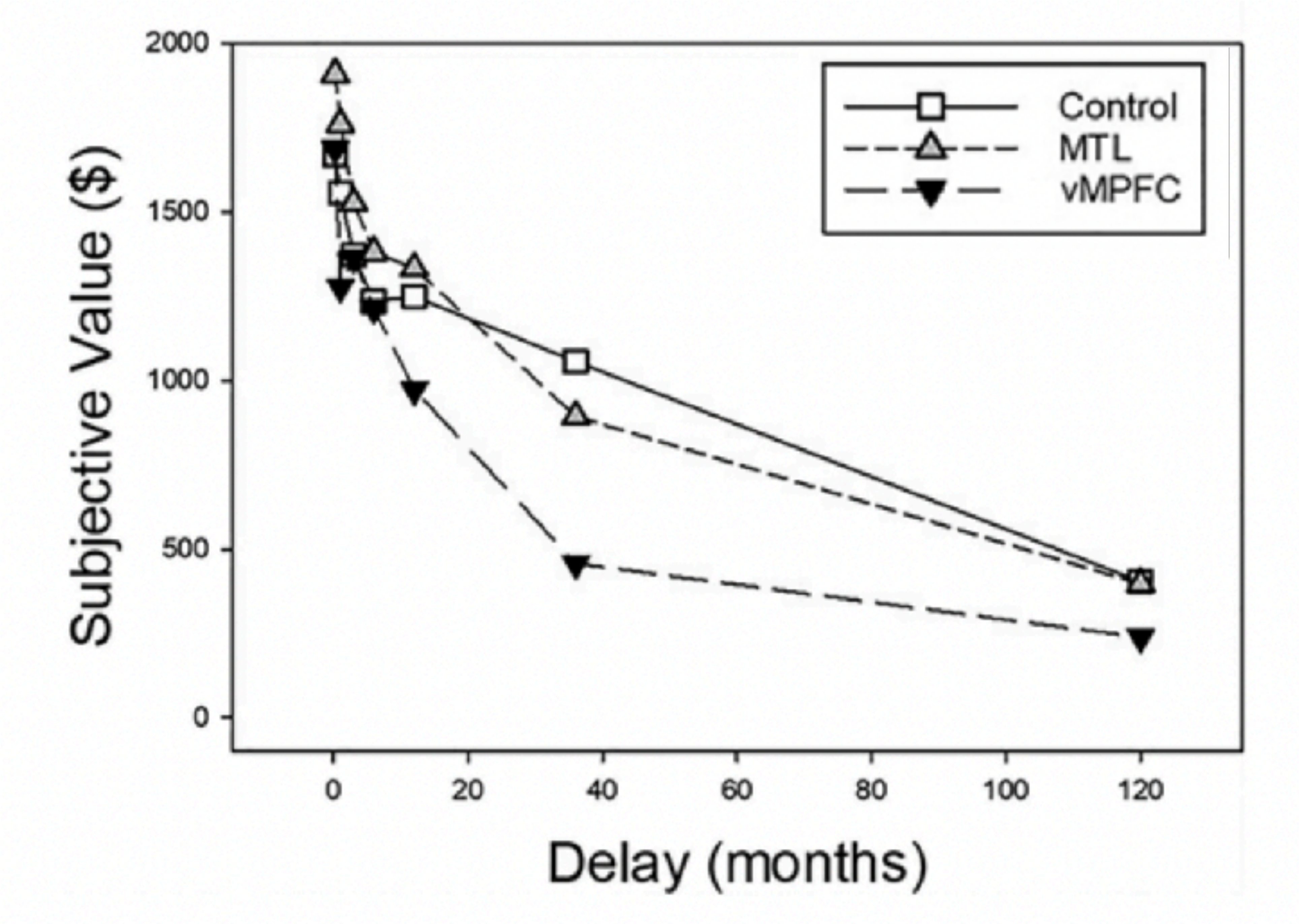
Schultz, Dayan, Montague, 1997

Discount factor and impulsivity:

RL parameters enables the characterization of populations or environments

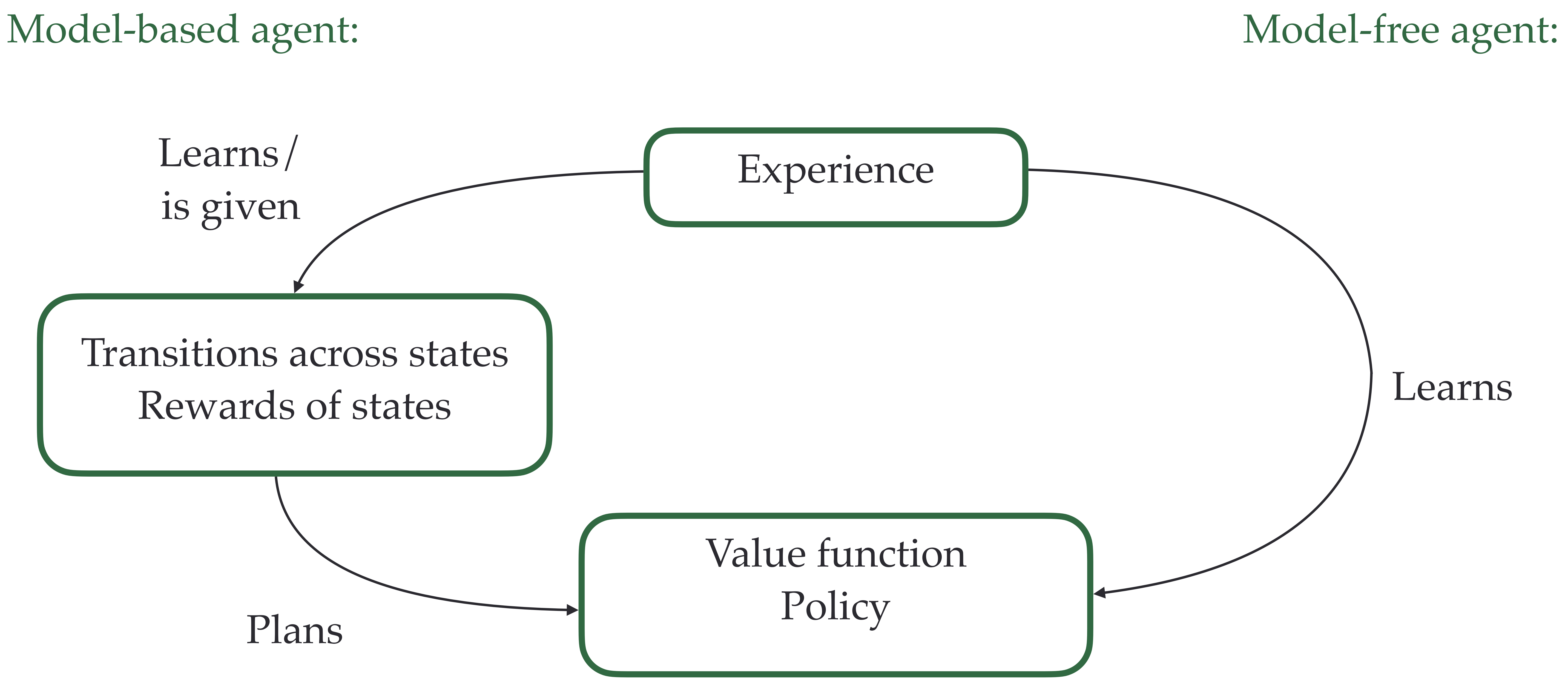
- Discount factor characterizes impulsivity/myopia
- Learning rate characterizes volatility
- …





Mok, Jenkin NY, et al. *Journal of Cognitive Neuroscience*. (2021)

Model-based vs model-free mechanisms

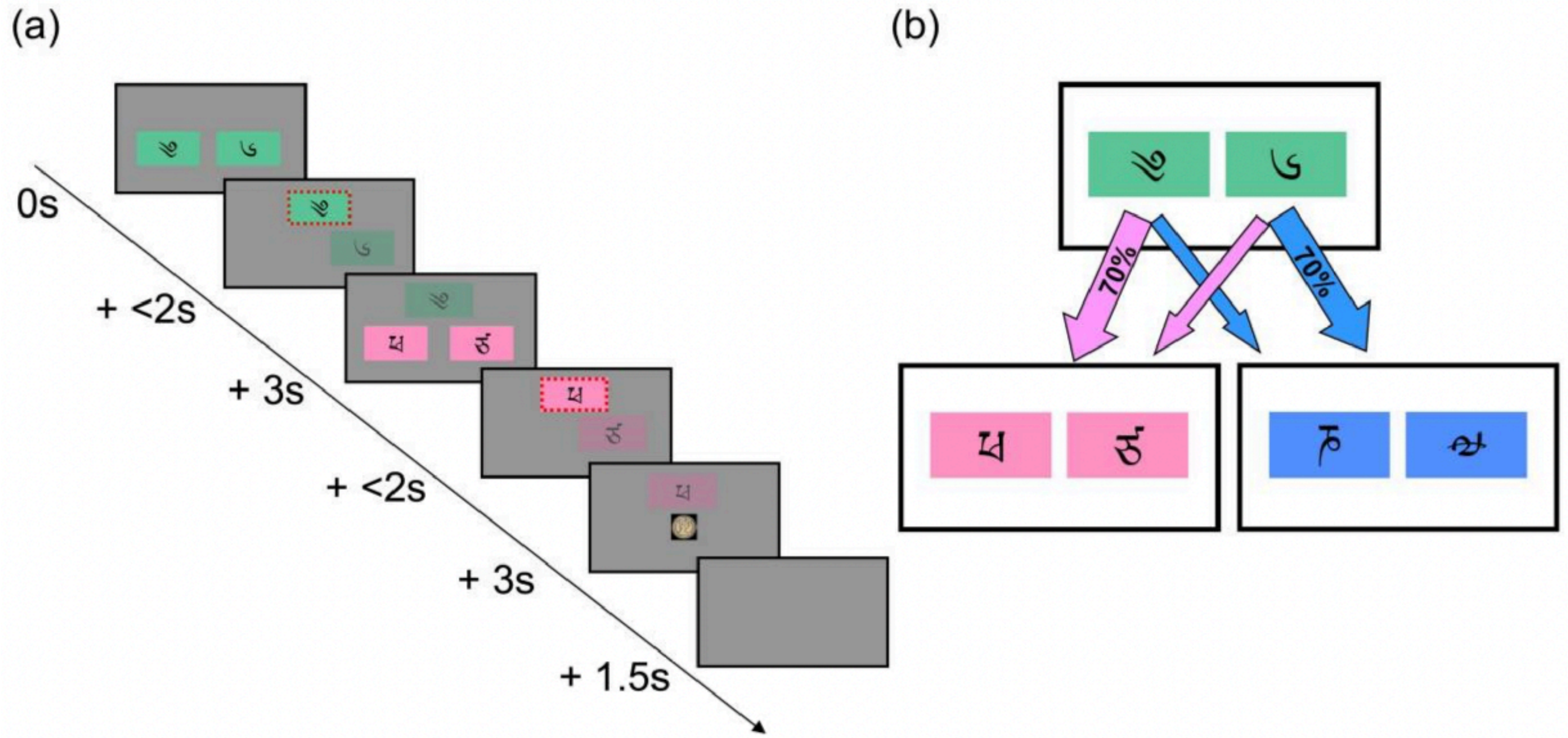


Model-based vs model-free mechanisms

Comparing behaviors/ brain region activity to model-based/model-free agents enables us to:

- assess how omniscient humans and animals are about a situation
- quantify the breadth of planning

Key example: the 2-steps task (Daw, *et al*. Neuron, 2011):



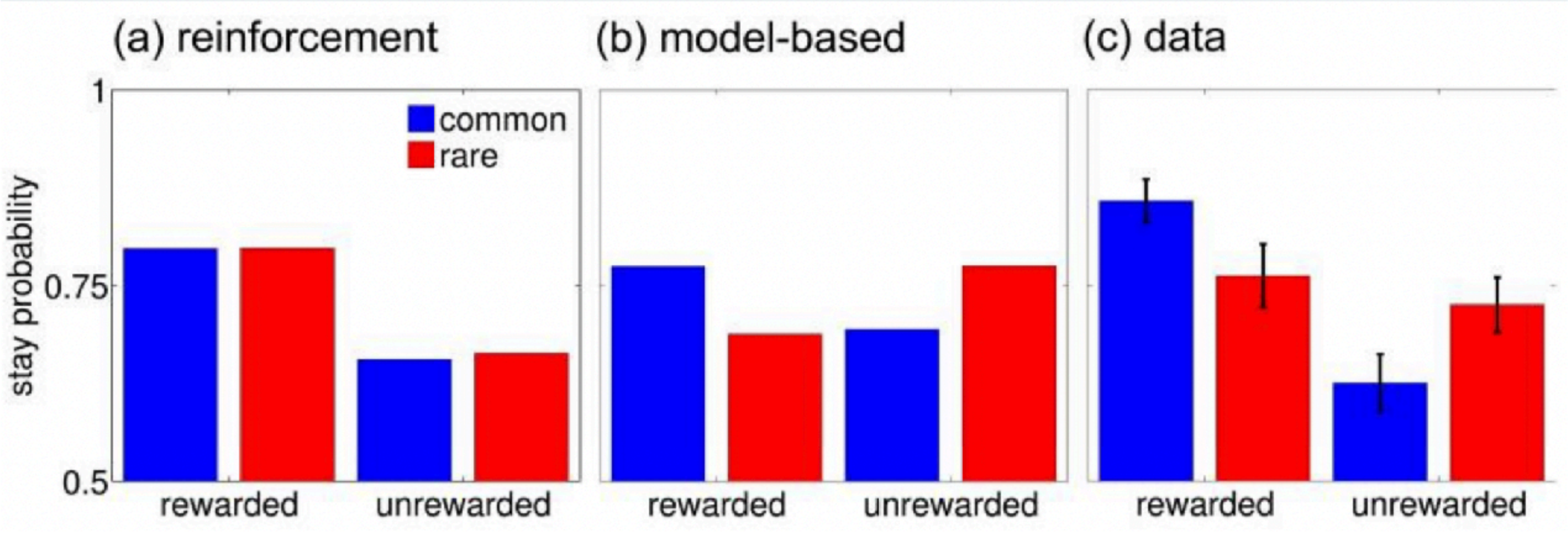
Facing biological constrains

2-steps task (Daw, *et al*. Neuron, 2011):

Comparing behaviors to model-based/model-free agents enables us to:

- quantify biological computational constrains
- quantify mechanisms of arbitration between the two

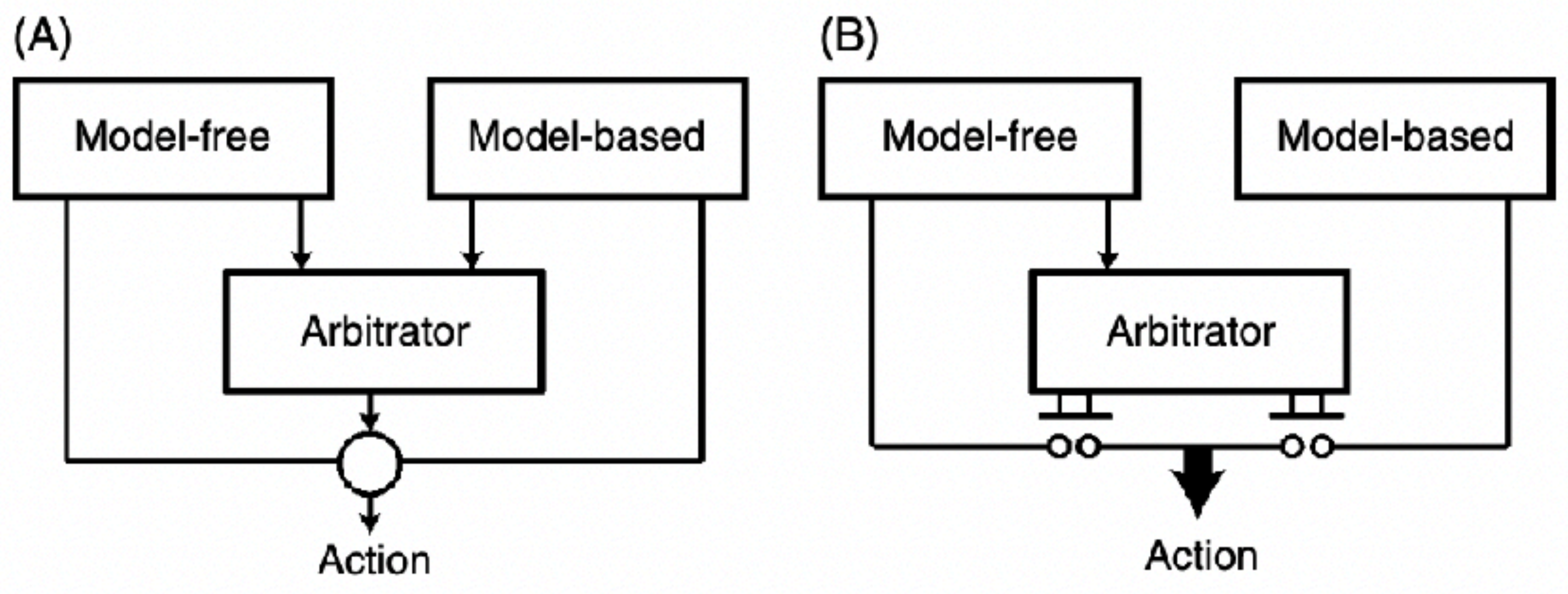
Humans are in between…

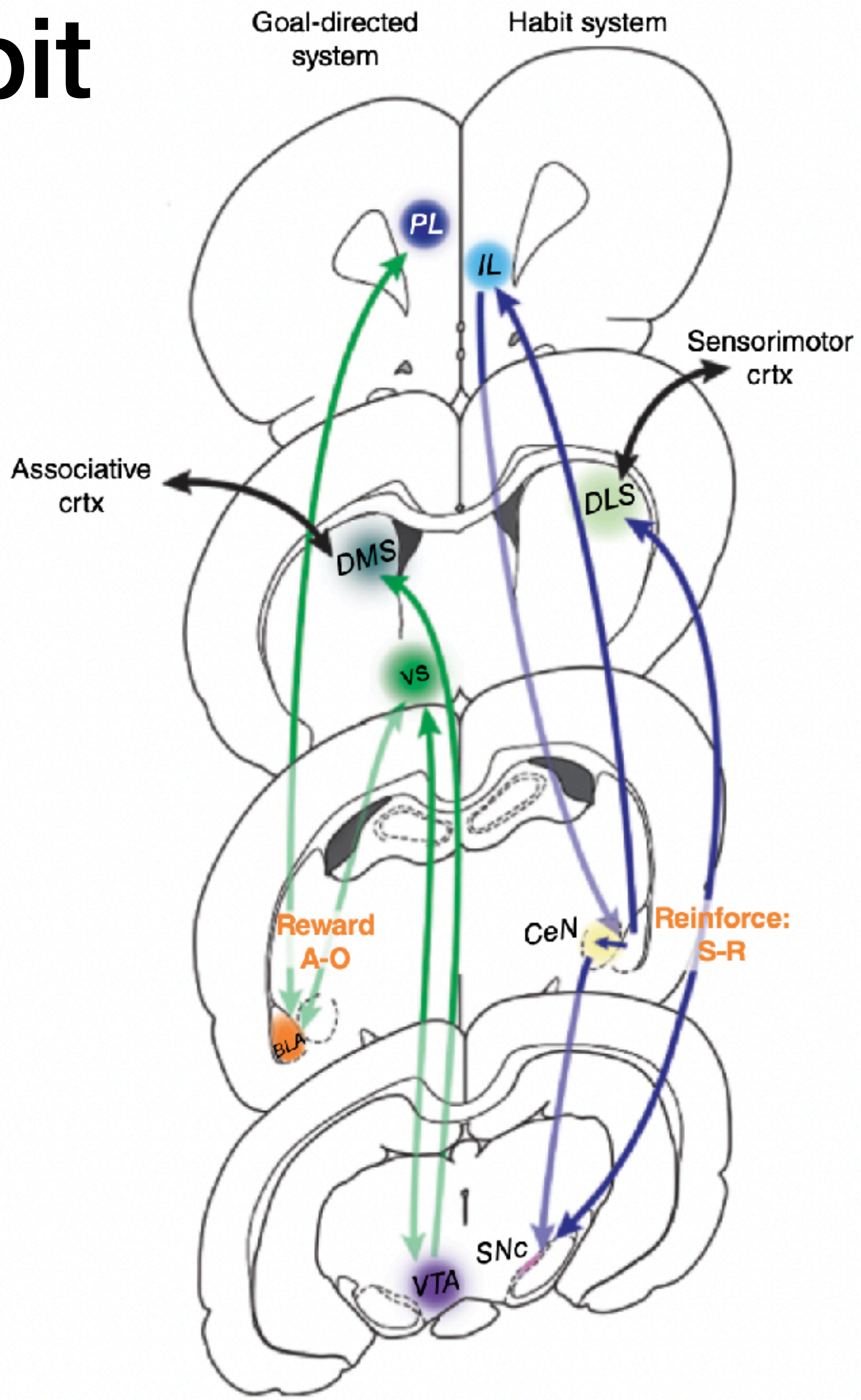


Uncovering mechanisms of habit formation

Comparing neural activity and behaviors to model-based / model-free agents enables us to:

- track emergence of habits
- quantify mechanisms of arbitration between the two and their neurological gating





Lingawi, Dezfouli, and Balleine. *The Wiley handbook on the cognitive neuroscience of l*

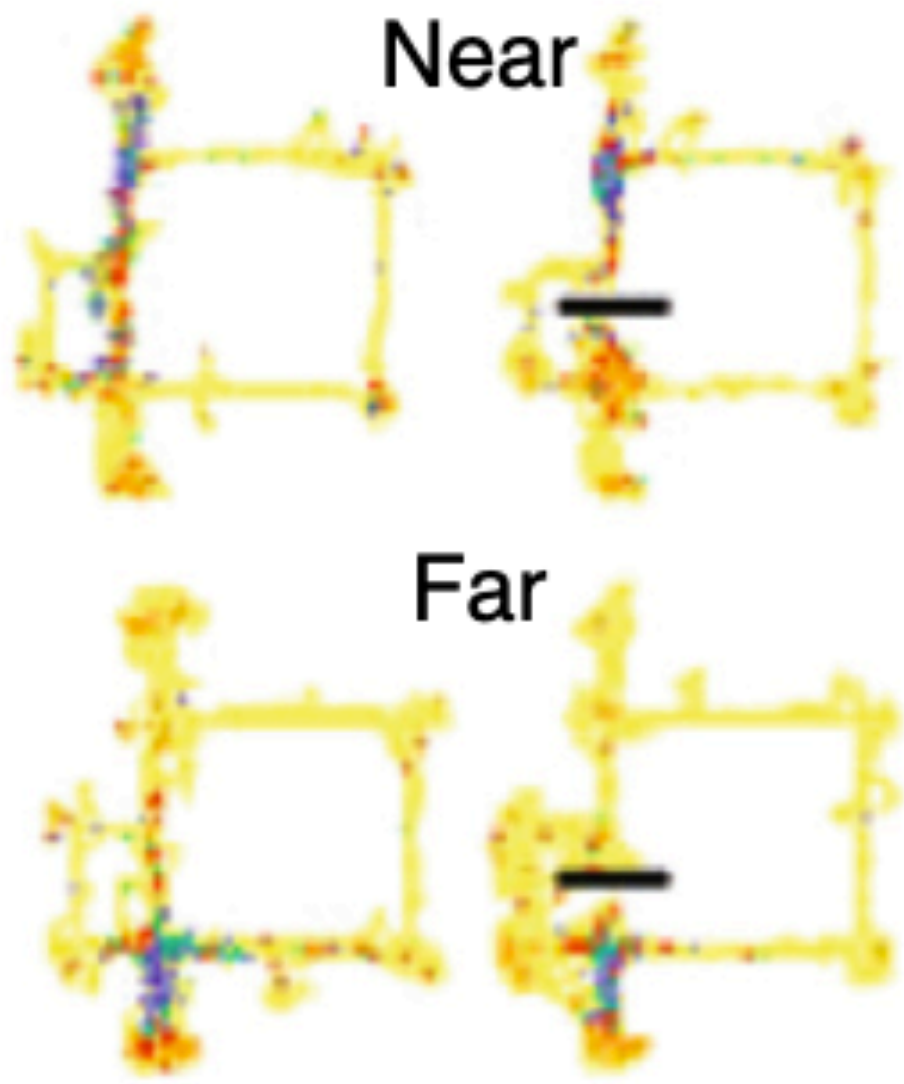
Uncovering mechanisms of prediction

Comparing neural activity and behaviors to an SR-based agent:

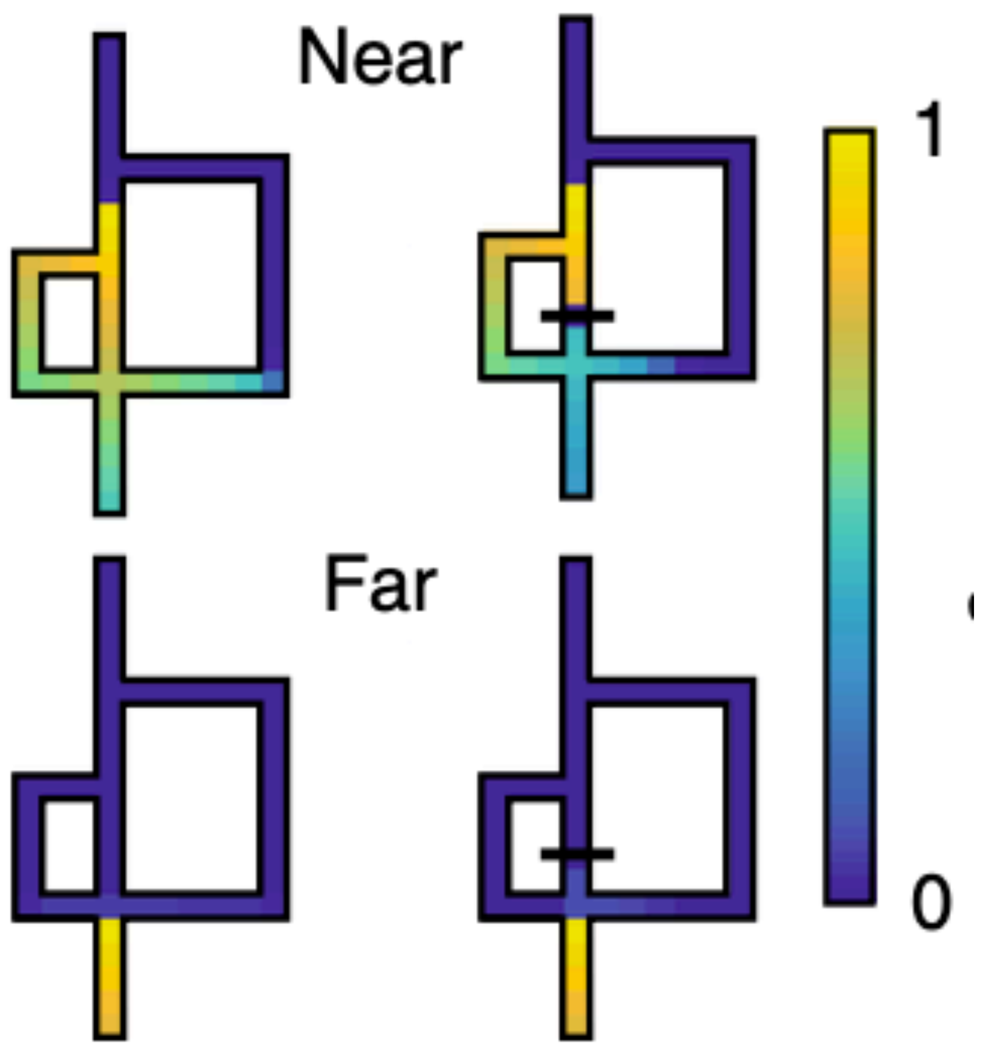
- maps timescales of predictions in the brain

Data- place field

(Alvernhe et al., *Eur. J. Neurosci.*,2011)

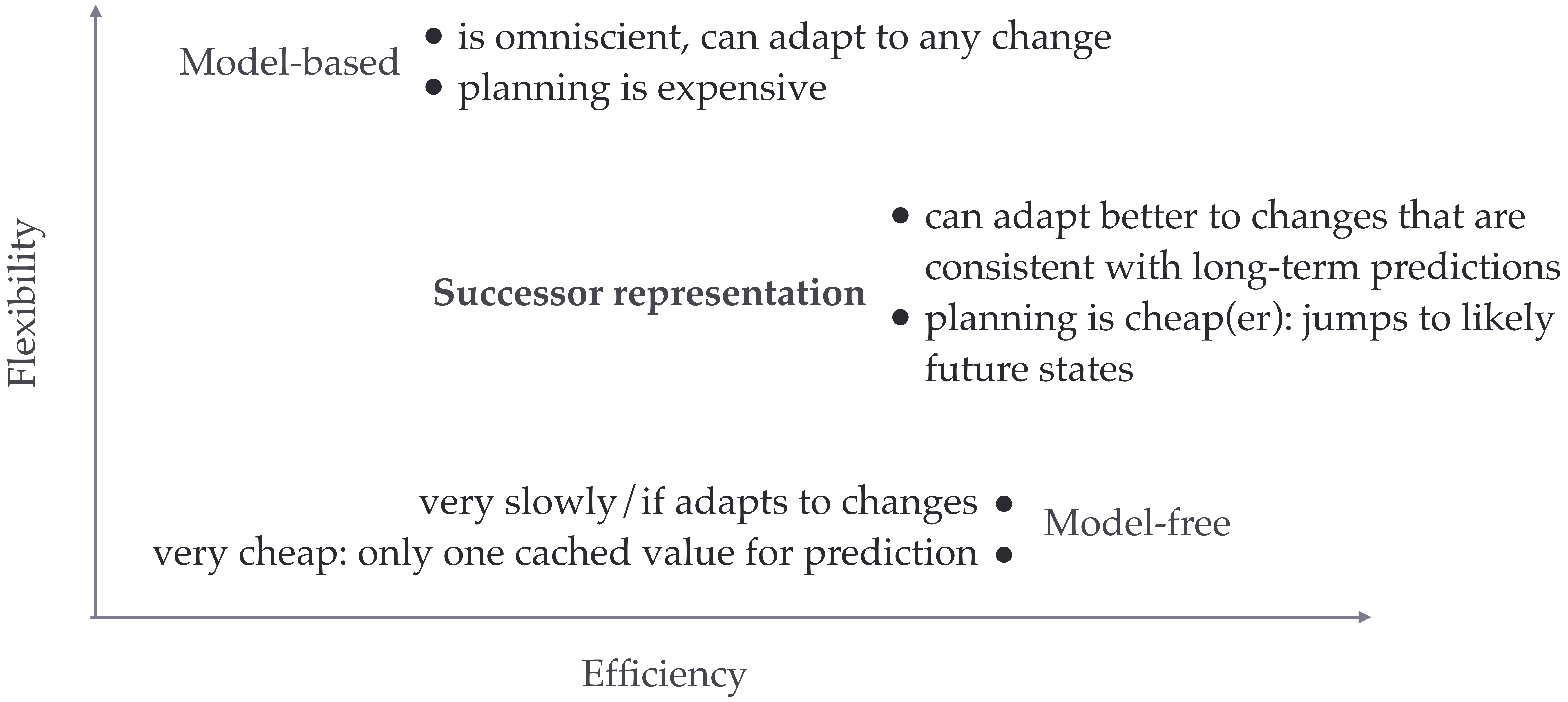


SR- place field



Stachenfeld, Botvinick and Gershman, *Nat. Neurosci.*, 2017

Quantifying flexibility from change



Lingawi, Dezfouli, and Balleine. *The Wiley handbook on the cognitive neuroscience of learning* (2016)

Quantifying randomness, moderating exploration/exploitation

Fitting exploration parameters/ terms enables us to:

- quantify degree of randomness
- quantify directed exploration

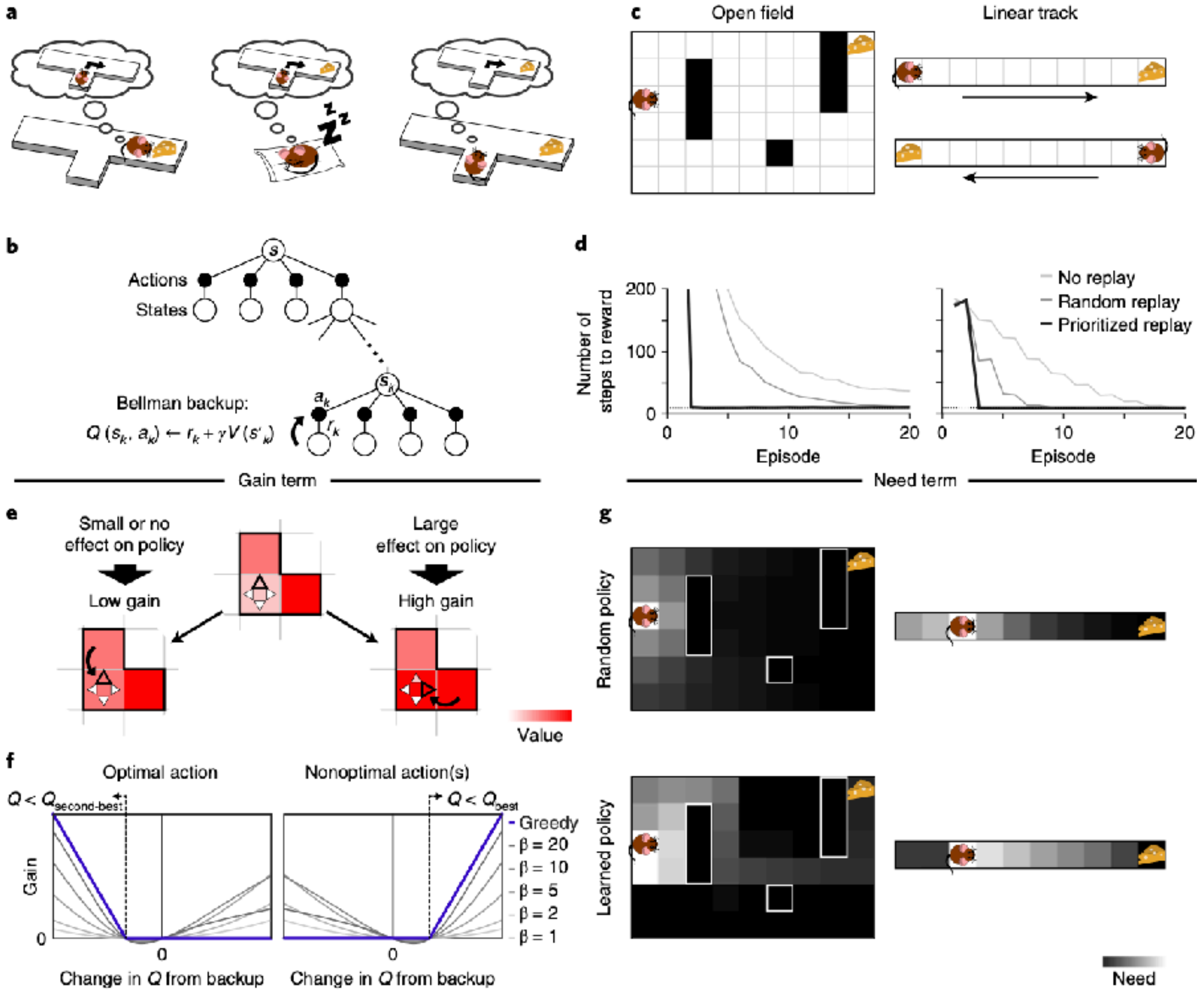
ε - greedy policy:

- $\pi(s, a_j) = \max_{a_j} Q(s, a_j)$ with probability 1- ε
- $\pi(s, a_j) = \text{Random}$ with probability ε

$$\pi(s, a_j) = \frac{\exp(\beta Q(s, a_j))}{\sum_{k=1}^{N_A} \exp\left(\beta Q(s, a_k)\right)}.$$

Quantifying replay/ offline learning

Comparing offline neural activity / replay behaviors (text, audio reports, dreams) to Dyna agent helps unravel and quantify the needs and gains for / from replay



Mattar and Daw. *Nature*. (2018)

Summary RL: quantifying mechanisms of learning and decision making

- Value/Q-learning: formalizes operant and pavlovian conditioning
- Policy gradient: formalizes ‘repeat bias’/‘win-stay’ behaviors
- Actor-critic: investigates boundary/interplay between values and actions
- Actor-critic VS policy gradient: to study value representation and action selection
- Deep Q-learning: transfer learning across states - large states and action spaces - generalization - mechanisms of layer processing in the brain

- Hierarchical RL: investigates how we break-out tasks
- Model-based: how we plan ahead
- Model-free vs model-based and their interplay: how sensitive one is to a specific experience/ how structural is the knowledge/ limitations of both

- SR: prediction: how far in the future do we plan ahead? How far in the past do we integrate information from? What is the relationship between timescale of prediction and precision of prediction?
- Dyna: how replay influences performance/learning & how experience influences replay

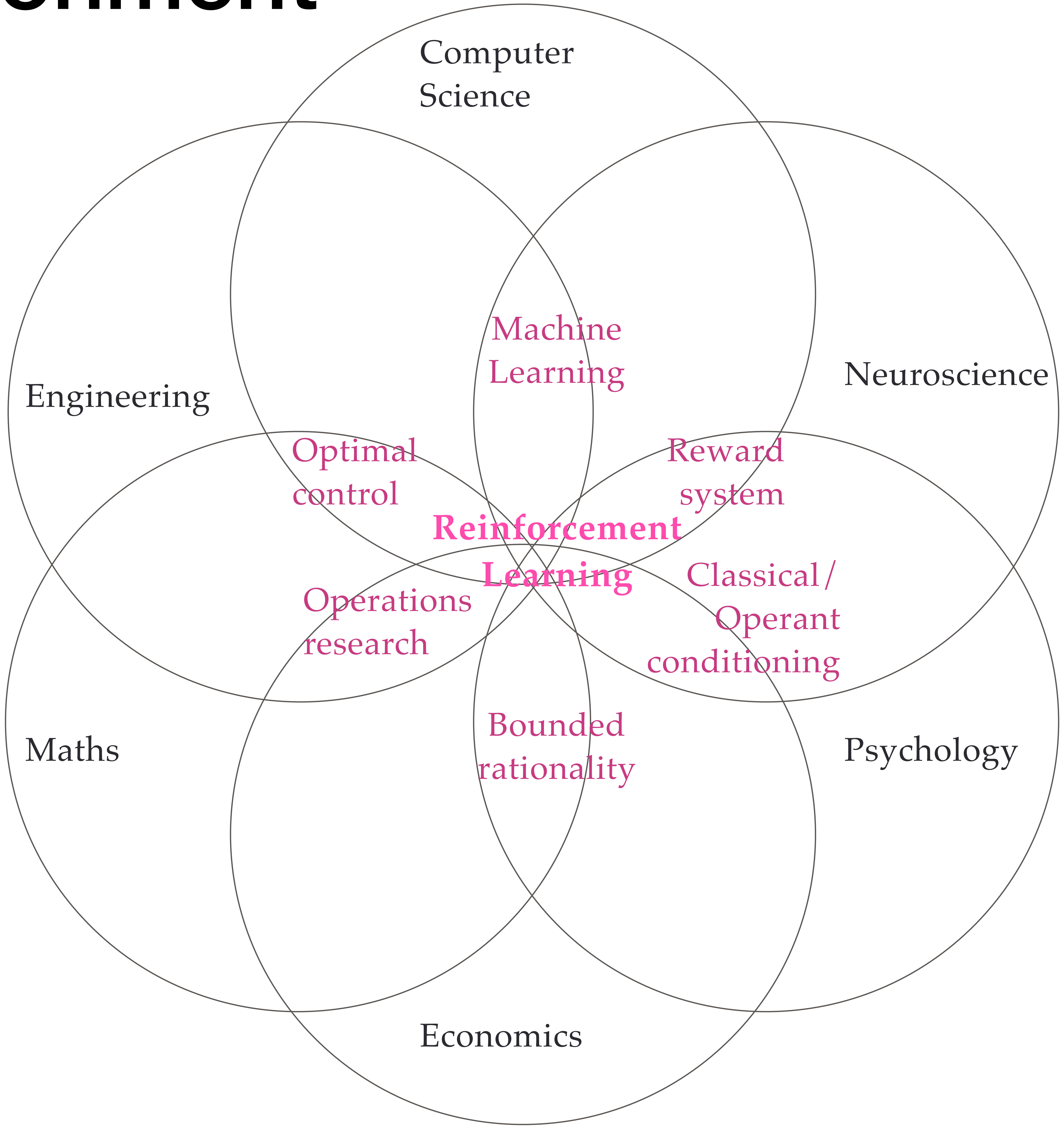
Summary RL: Science of learning to make decisions from interaction with the environment

A broad investigation of behaviour, learning and prediction

- Reward-related prediction
- Transition-related prediction
- State representation and generalization
- Action representation and generalization
- Experience-modulated learning

Interplay between neuroscience, behavior, cognitive science, machine learning, robotics:

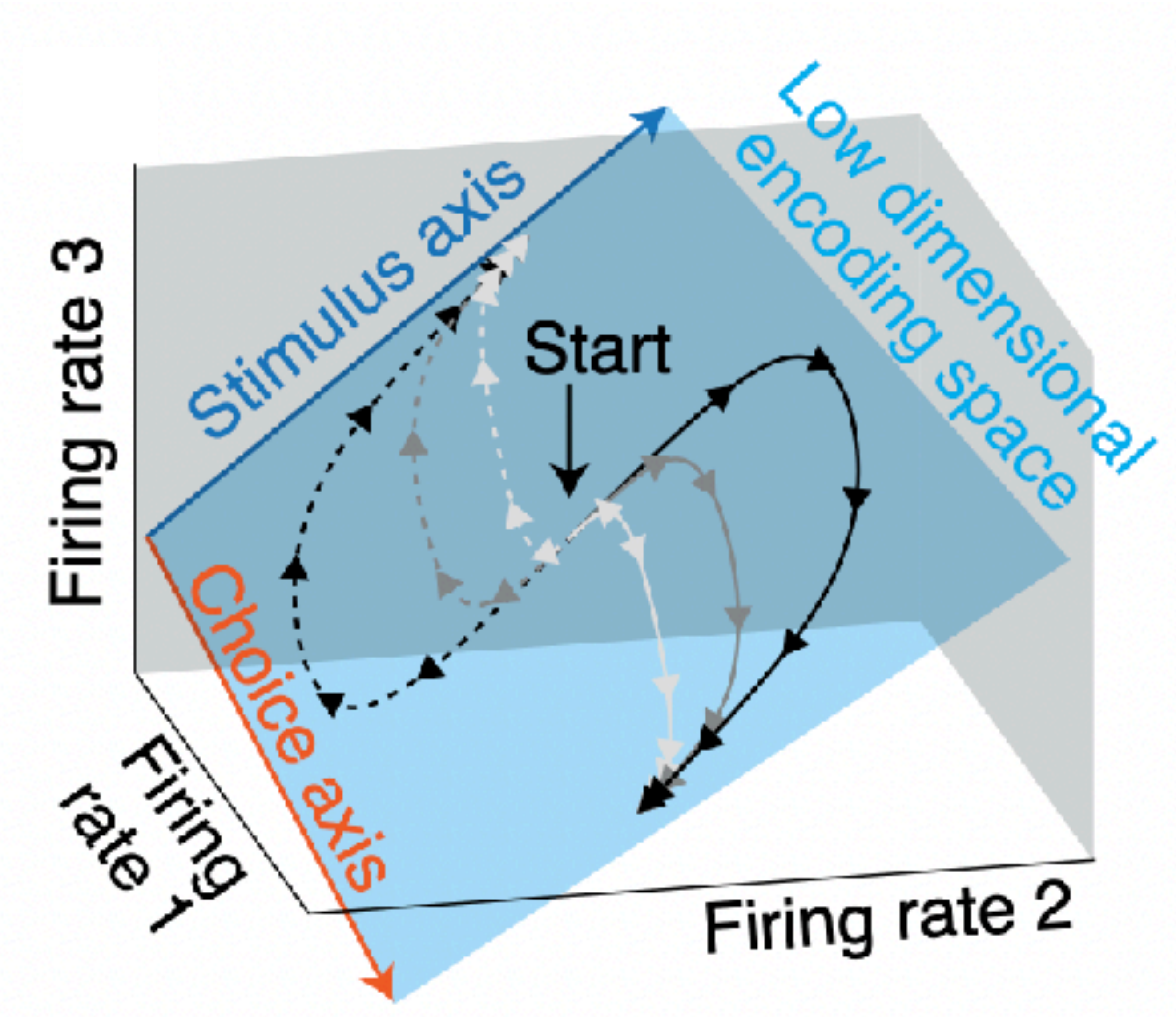
- all of those approaches enable to study behaviors
- they also improve from neurosciences advances
  - in particular, we need more flexible agents!

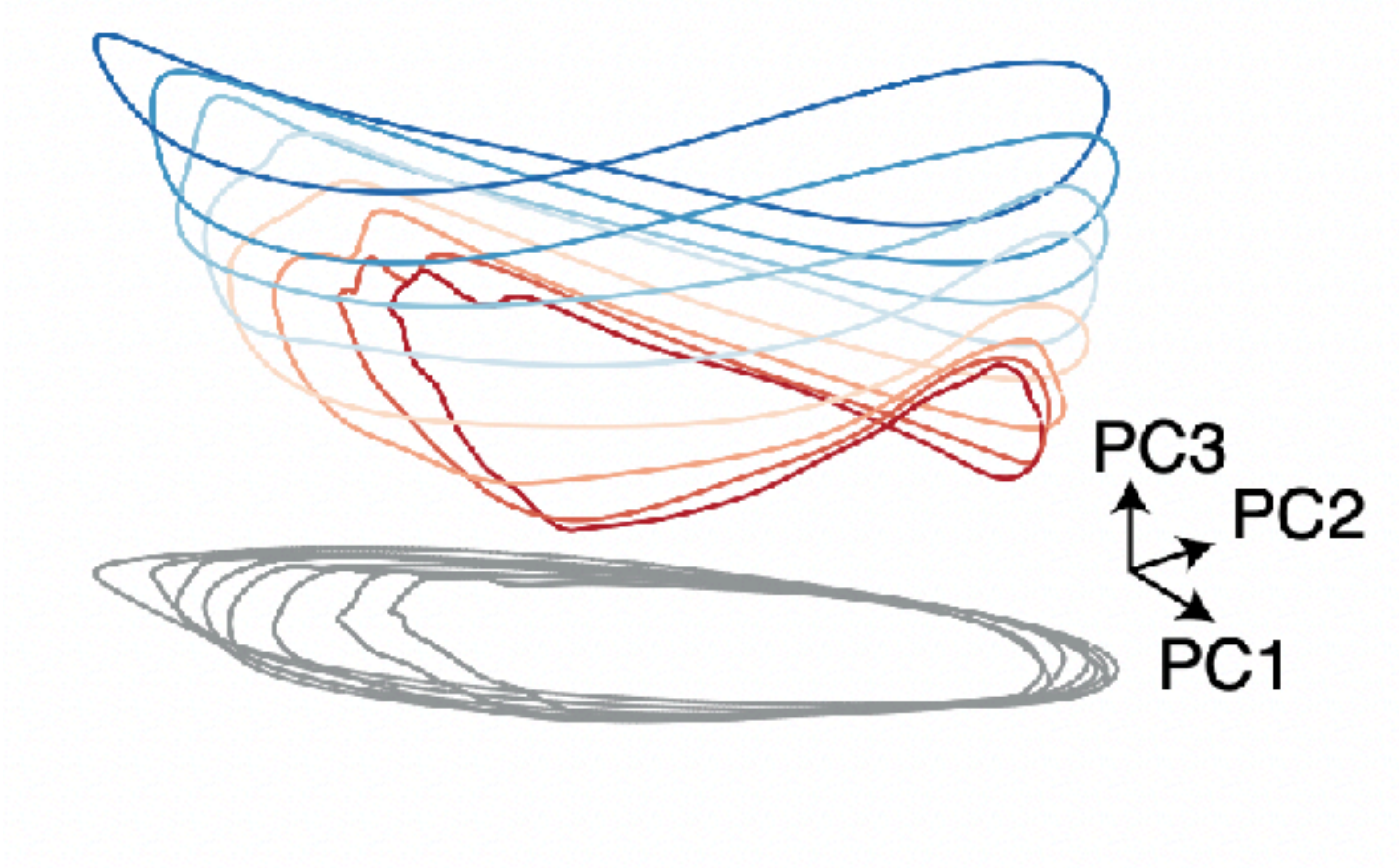


Dimensionality reduction

Manifold:

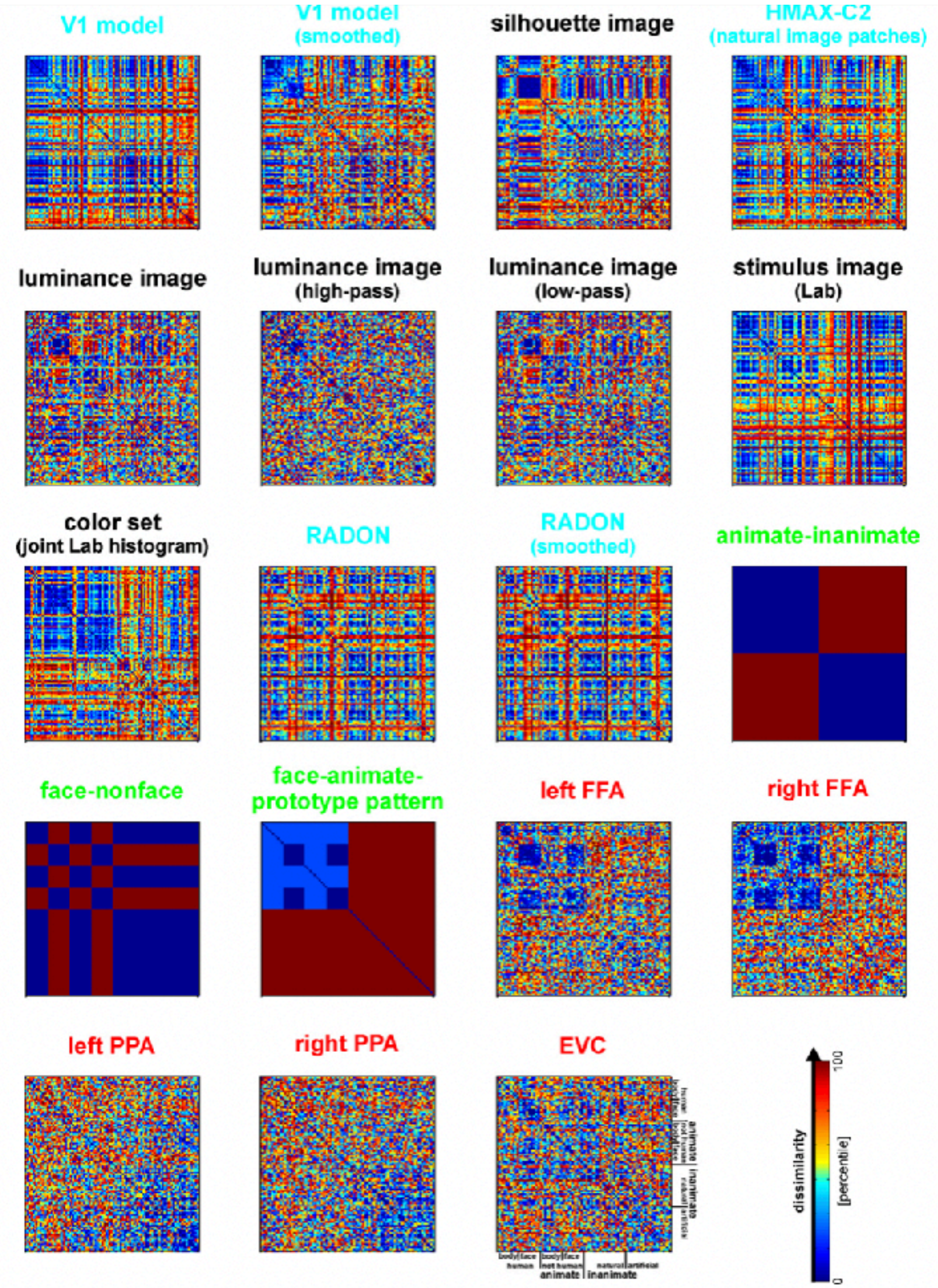
- Can capture task-relevant dimensions of neural activities - practical for dimension reduction
- Comparing those dimension and the stability of the dynamics using modeling enables to shed light on neural computations
- Embedding useful task-related dynamics within neural network can help perform tasks





Representation similarity analysis

- Can capture representational and functional marker of a brain region/model by looking at its pattern of activity correlations
- RDMs capture how different do their react to stimuli/experimental conditions
- Can be used to compare to models and or other brain regions:
  - With models, it gives information on the encoding of the region
  - With brain regions, it can be used to infer connectivity between brain regions
  - It can be used to infer/design clever connecting weights in RNNs



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 langauge)
  - Thus, there is a missing factor and that factor is 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 = UG + input

Landauer & Dumais (1997)

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

- Simple idea: *Represent the meaning of words based on the company they keep*
- **Input**: a matrix (A) containing counts of which words occur in which contexts (i.e., texts)
- **Process**: matrix factorization using singular value decomposition (SVD)
- **Outputs**:
  - Word vectors (B) and Context vectors (C)
  - Both are mapped to the same high-dimensional latent space (300 dims)
  - The distance between word vectors captures similarity, which can be used to generalize

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)

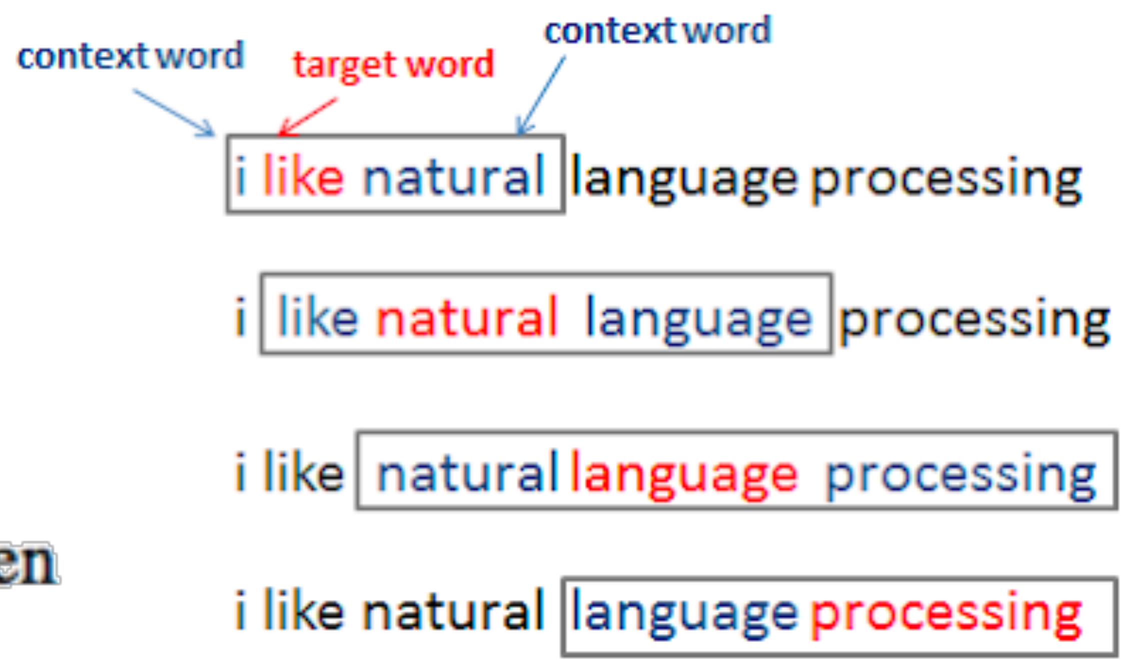
Word/	1	.	.	.	300
1	y	.	.	.	y
.	y	.	.	.	y
.	.	.	.	.	.
.	.	.	.	.	.
.	.	.	.	.	.
.	y	.	.	.	y
60,000	y	.	.	.	y

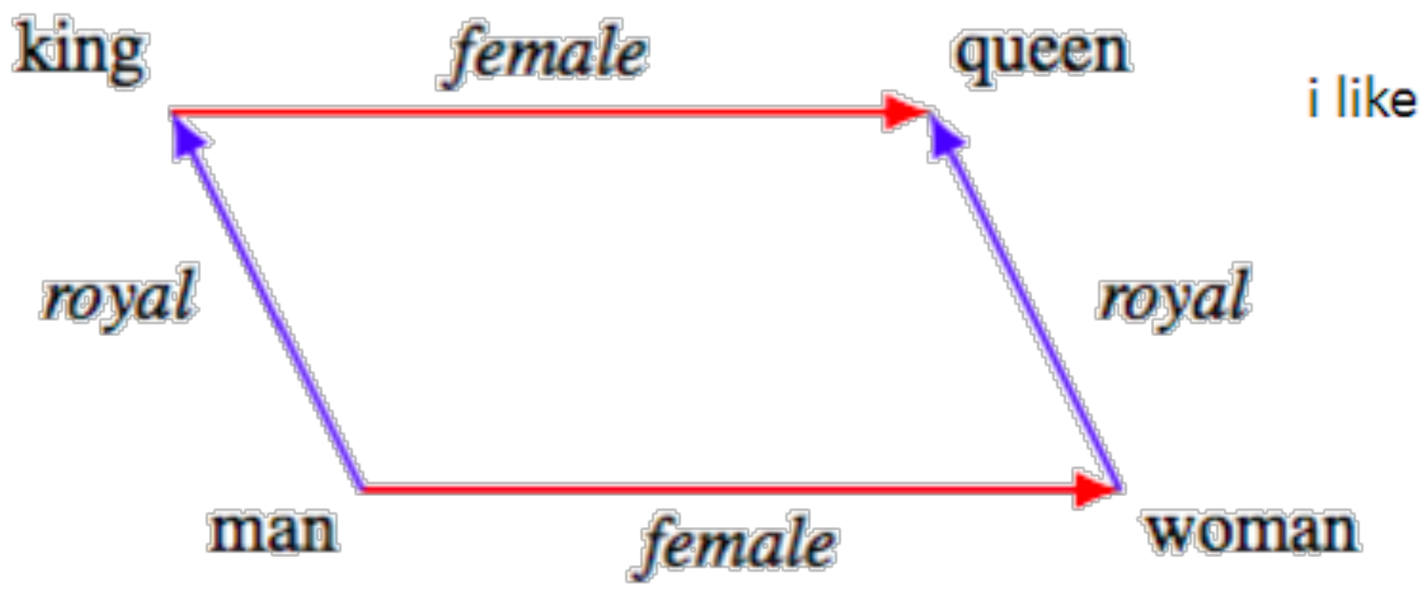
C Factor (dimension)

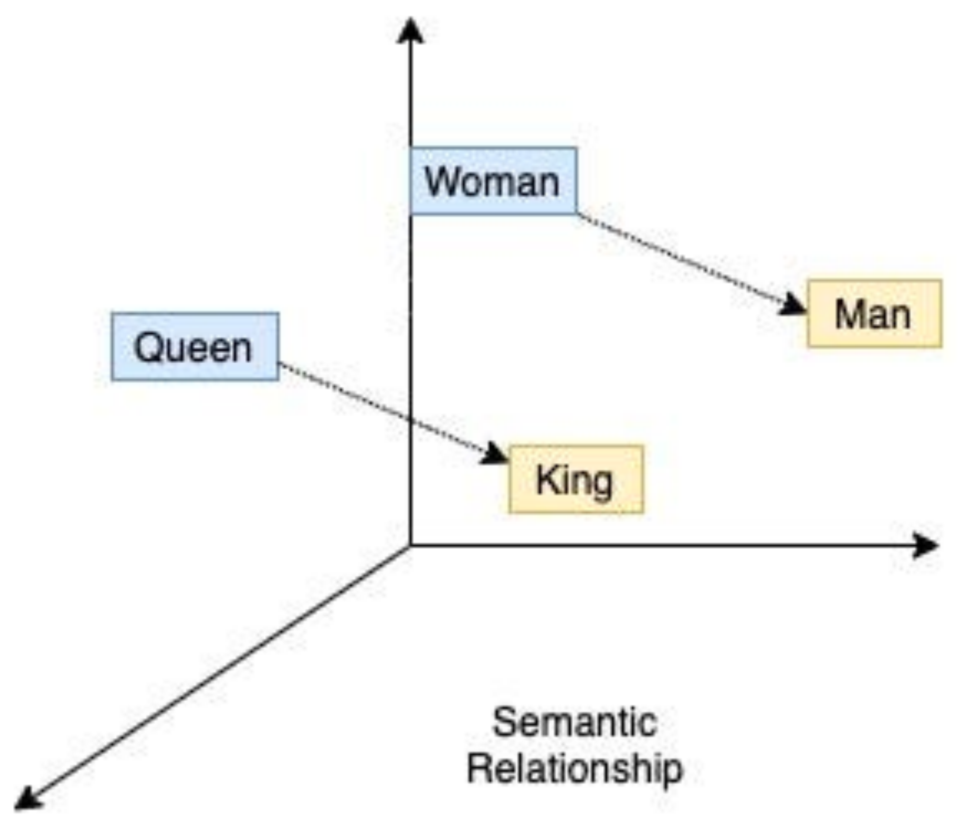
Sample/	1	.	.	.	300
1	z	.	.	.	z
.	.	.	.	.	z
.	z	.	.	.	z
.	z	.	.	.	z
30,000	z	.	.	.	z

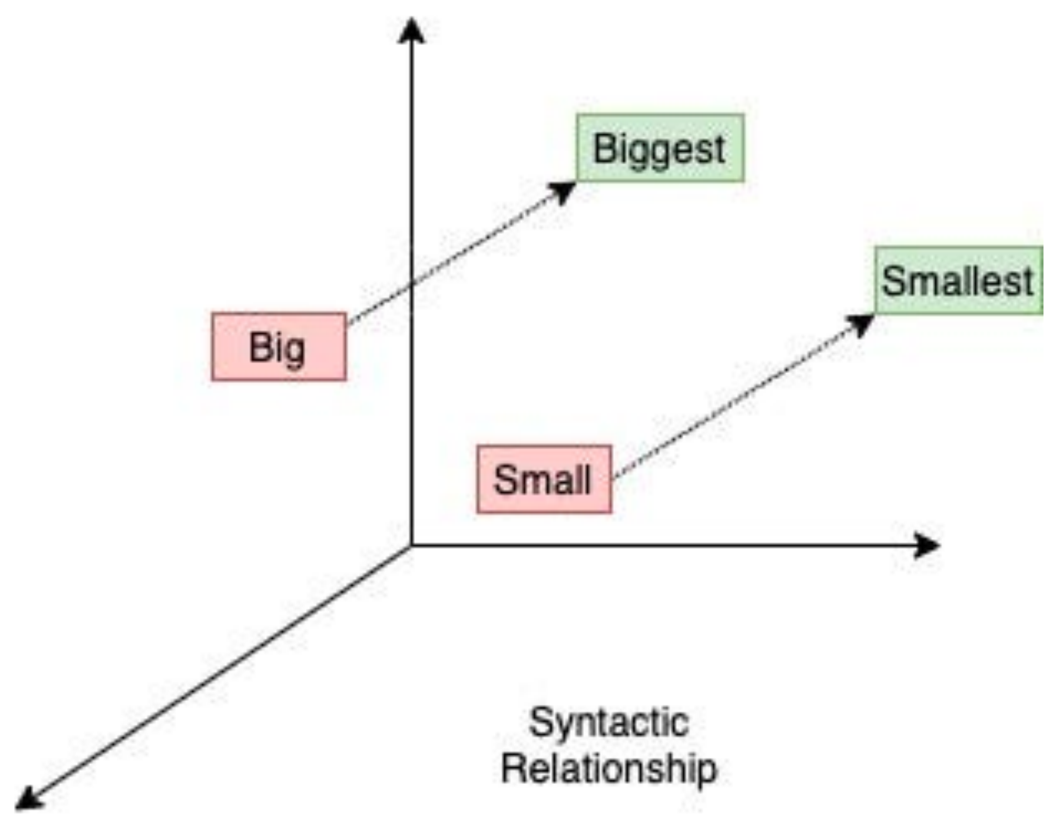
Word2vec, RNNs, and LSTMs

context window

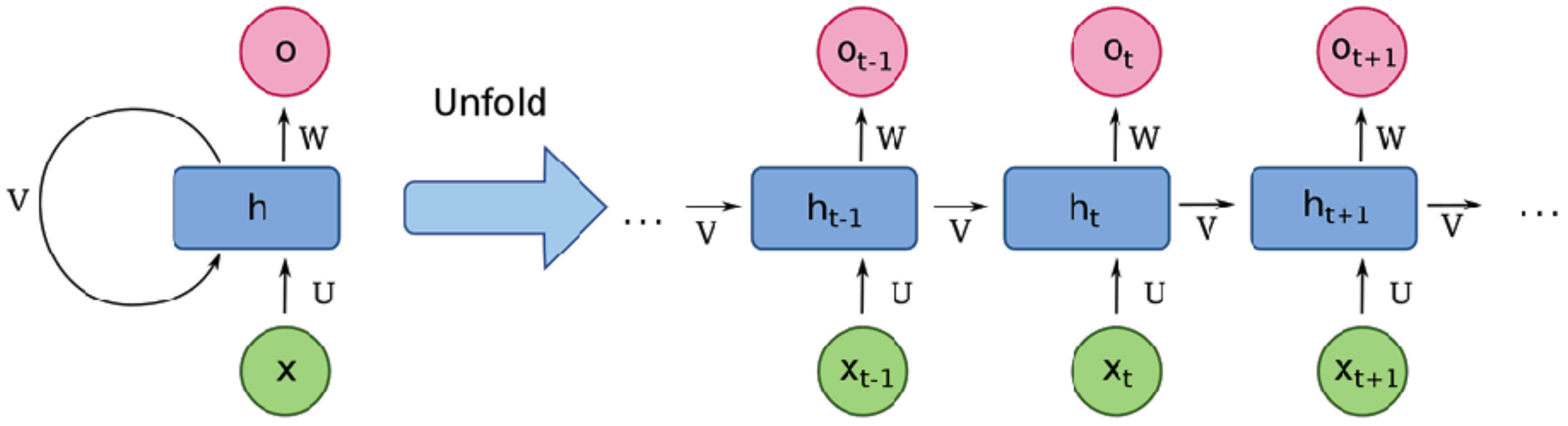




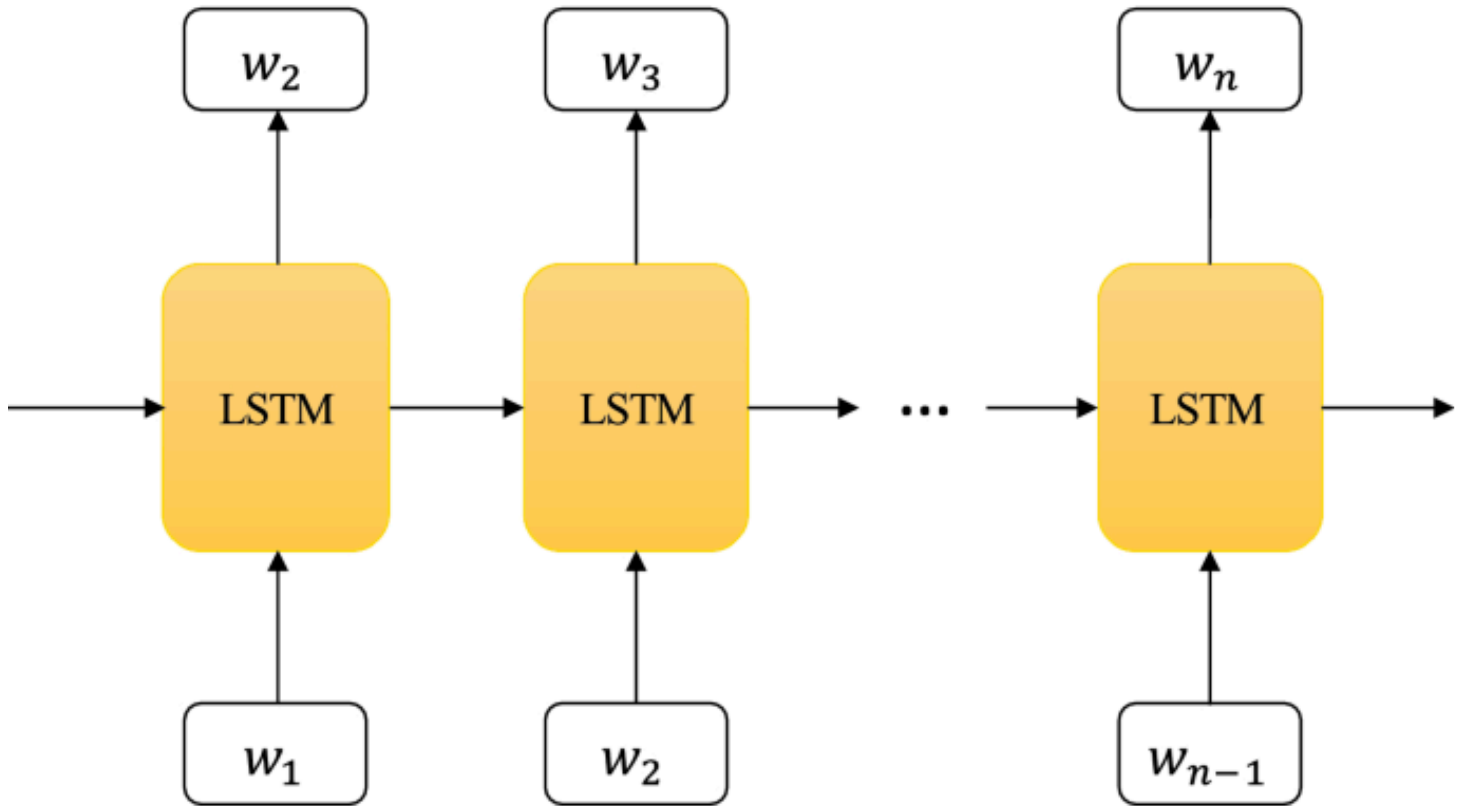




RNNs



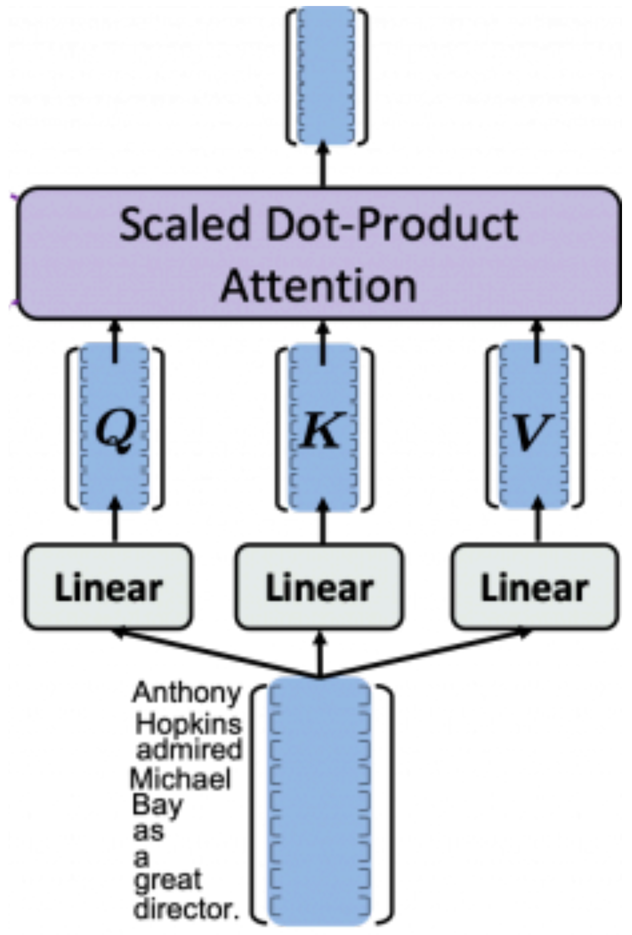
LSTMs

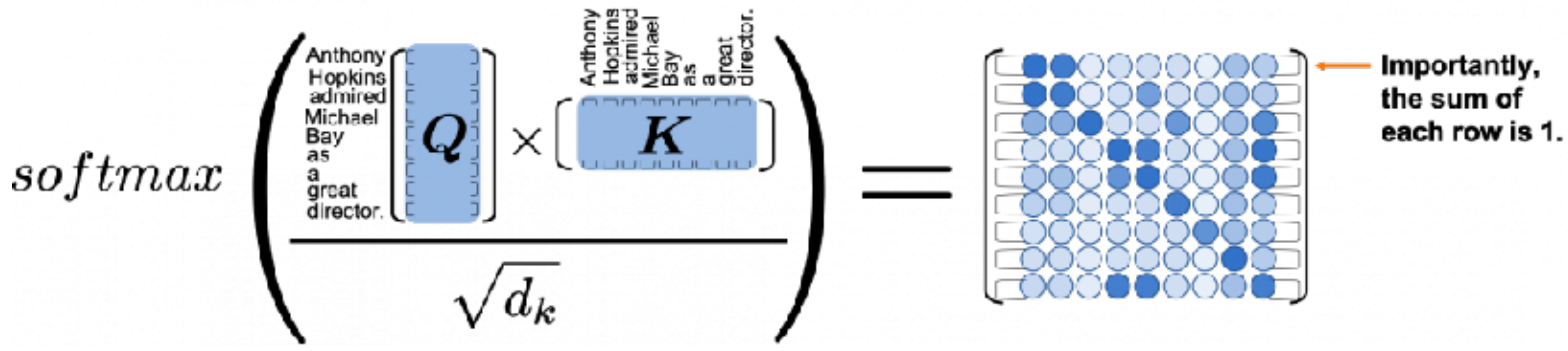


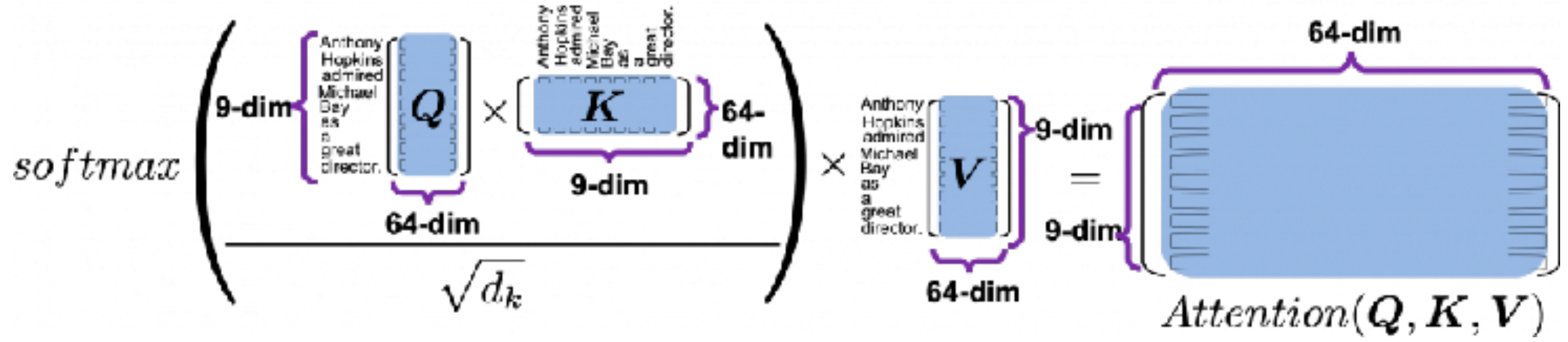
Self-attention in Transformer Networks

- Self-attention captures relationships between different words/tokens in a sequence, capturing contextual information and complex dependencies
- Each input is mapped to Query, Key, and Value representations through linear operations (fully connected layers)
    - Analogous to information retrieval (e.g., searching for videos on youtube): the search engine maps **query** (text in search bar) to **keys** (video title/description) associated with each candidate, and then presents us with a set of matches (**values**)
- $QK^\top$ produces a *score*, which is then put through a softmax to weight the relative importance of each word for each other word
- This is then multipled against Value representations to generate a contextualized representation of the text

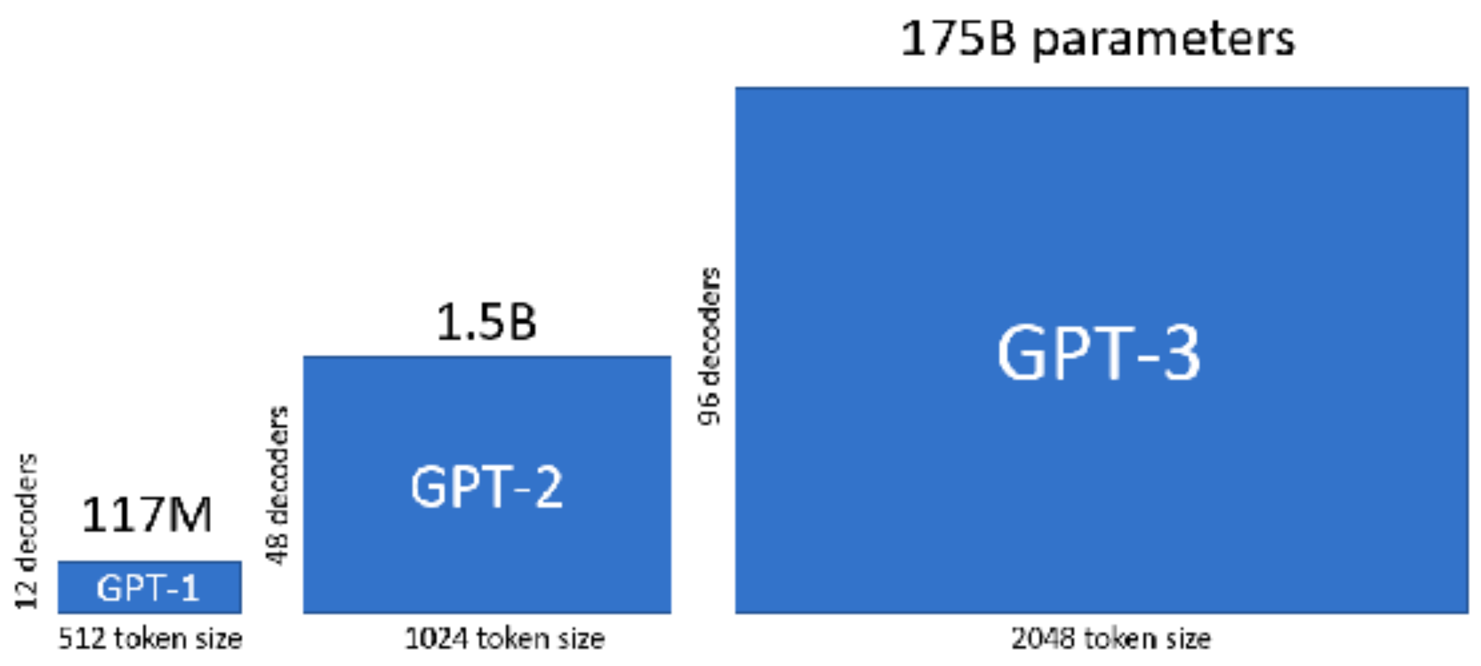
$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V^\top$$







What are LLMs?



- Self-attention mechanism used in massively hierarchical architecture of transformers networks
- Context window prediction (similar to word2vec)
- Various forms of training
  - Unsupervised text prediction
  - Supervised training on labeled data
  - Reinforcement learning from human feedback (RLHF)
- In-context learning and prompt engineering

How well have we answered these original questions?

- What is learning?
- What aspects of learning are the same across biological and artificial systems? What is different?
- What has the study of biological intelligence informed us about artificial systems?
- What can artificial intelligence teach us about biological intelligence?

What is learning?

- Forming expectations about the environment and updating through prediction error (delta-rule)
  - Reward: Rescorla-Wagner, RL, and ANNs via gradient descent
  - State transitions: Model-based RL and Successor Representation
- Generalization from local to global patterns
  - Concepts and rule learning, value function approximation, latent semantic analysis
- Combination of different systems
  - habit and planning
  - symbolic and subsymbolic
  - rules and similarity

What aspects of learning are the same across biological and artificial systems?

- Generalization mechanisms
- Structured hypotheses
  - hierarchical organization
  - concept boundaries/functional relationships
- Functional separation
  - Different mechanisms focusing on different subproblems (e.g., value function and policy)

What is different?

- Amount of training data and computational power
- General Intelligence (still missing AGI)
  - Flexibility in a variety of real-world environments
- Social, pedagogical, and cultural learning
- Biological development

What has the study of biological intelligence informed us about artificial systems?

- Prediction-error learning
- Language of Thought (LoT) and symbolic representations
- Representations of the environment (Tolman)
- Combining different learning systems
  - Rules and similarity
  - Habits and planning
  - Symbolic and subsymbolic
  - Across different timescales and hierarchies (e.g., subgoals)
- Poverty of the stimulus
  - How humans learn so much from so little: *infinite use of finite means* (A. Humboldt)

What can artificial intelligence teach us about biological intelligence?

- Precise computational specification of verbal/conceptual theories
  - with testable outcomes via simulations
- Ability to address normative questions
  - Which learning systems work well in which situations?
  - And why do they fail?
- Resolutions to debates through model comparison
- Access to learned representations that are difficult to study in living brains
  - e.g., analyzing weights of ANNs
- Failures of AI to capture human behavior point towards promising research directions

Tutorial tomorrow

- Exam preparation
- Bring in 2-3 candidate exam questions
  - Short answer questions
  - You are incentivized to bring plausible questions that would be sufficiently challenging, thought provoking, and feasible
  - Good questions will be included on the exam