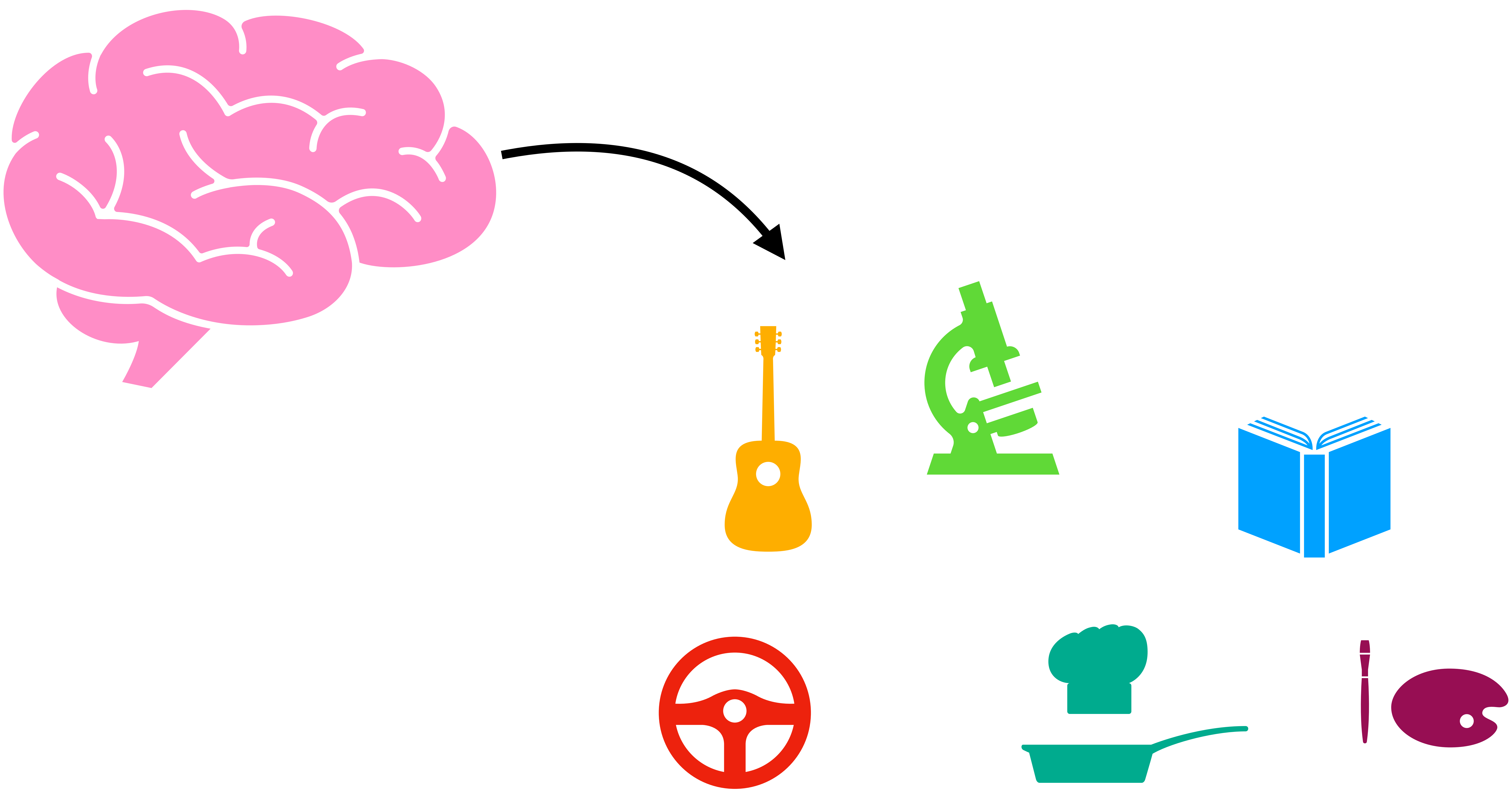
Generalization in Reinforcement Learning

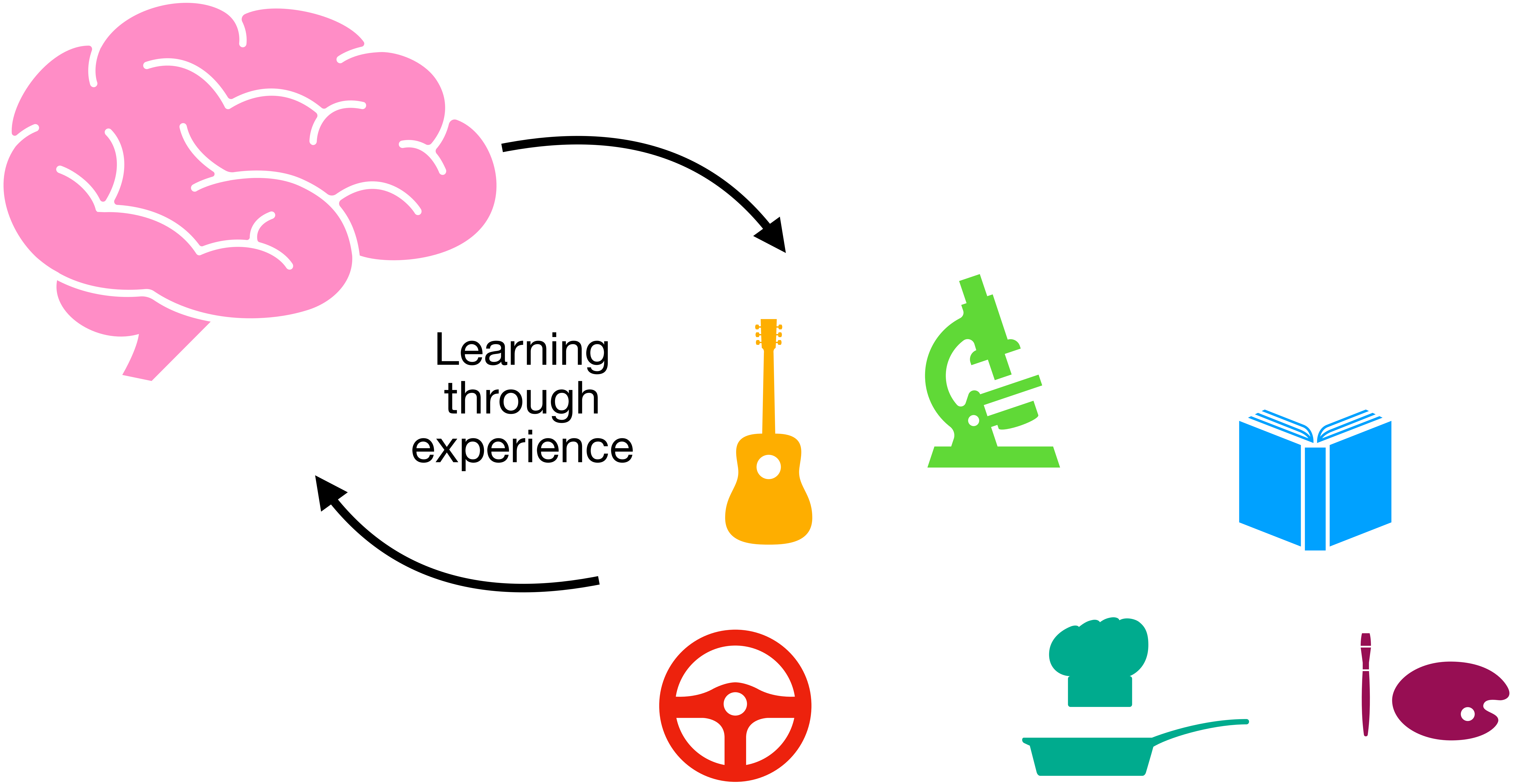
Charley Wu
Human and Machine Cognition Lab
hmc-lab.com

Brains control behavior

Brains control behavior

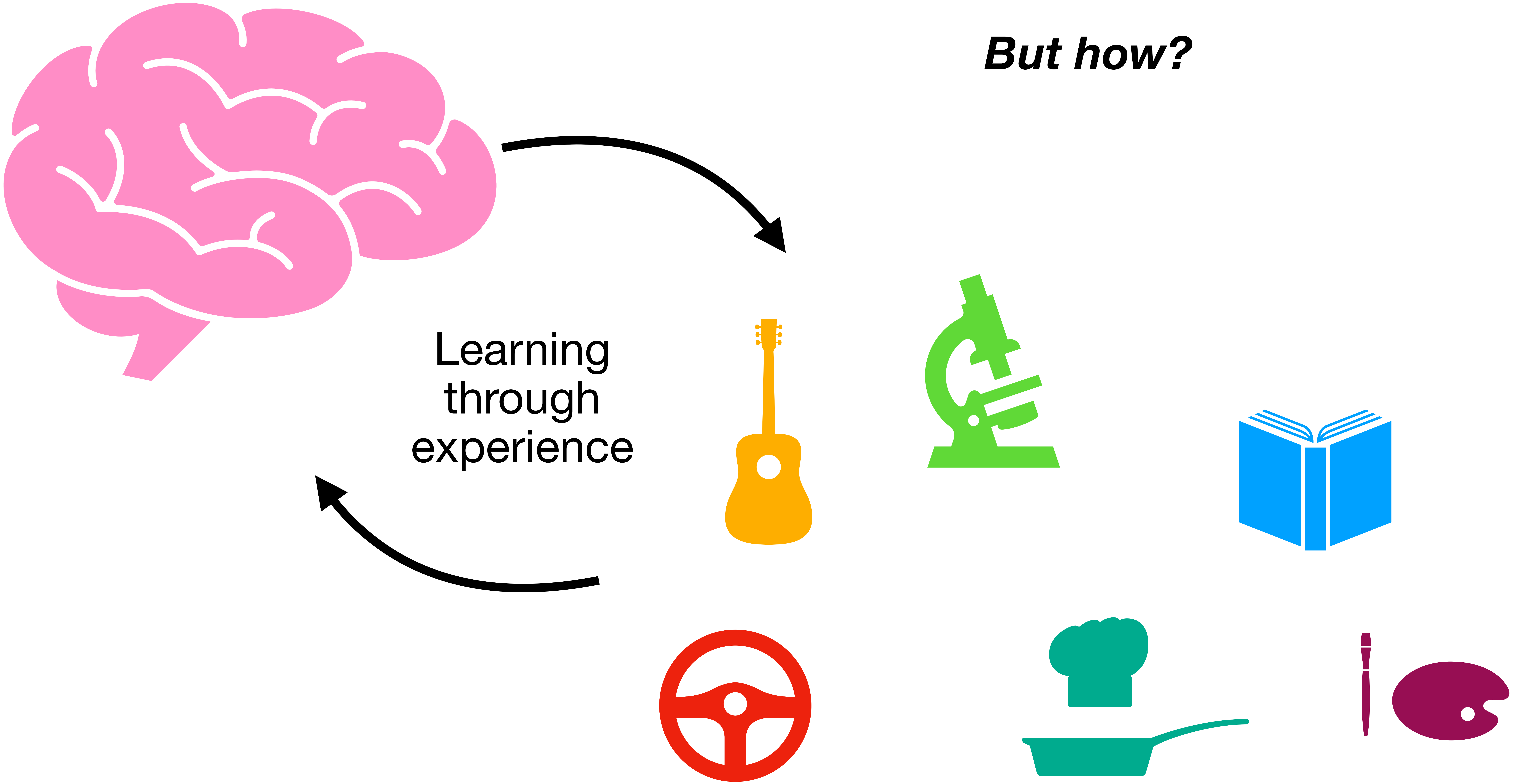


Brains control behavior

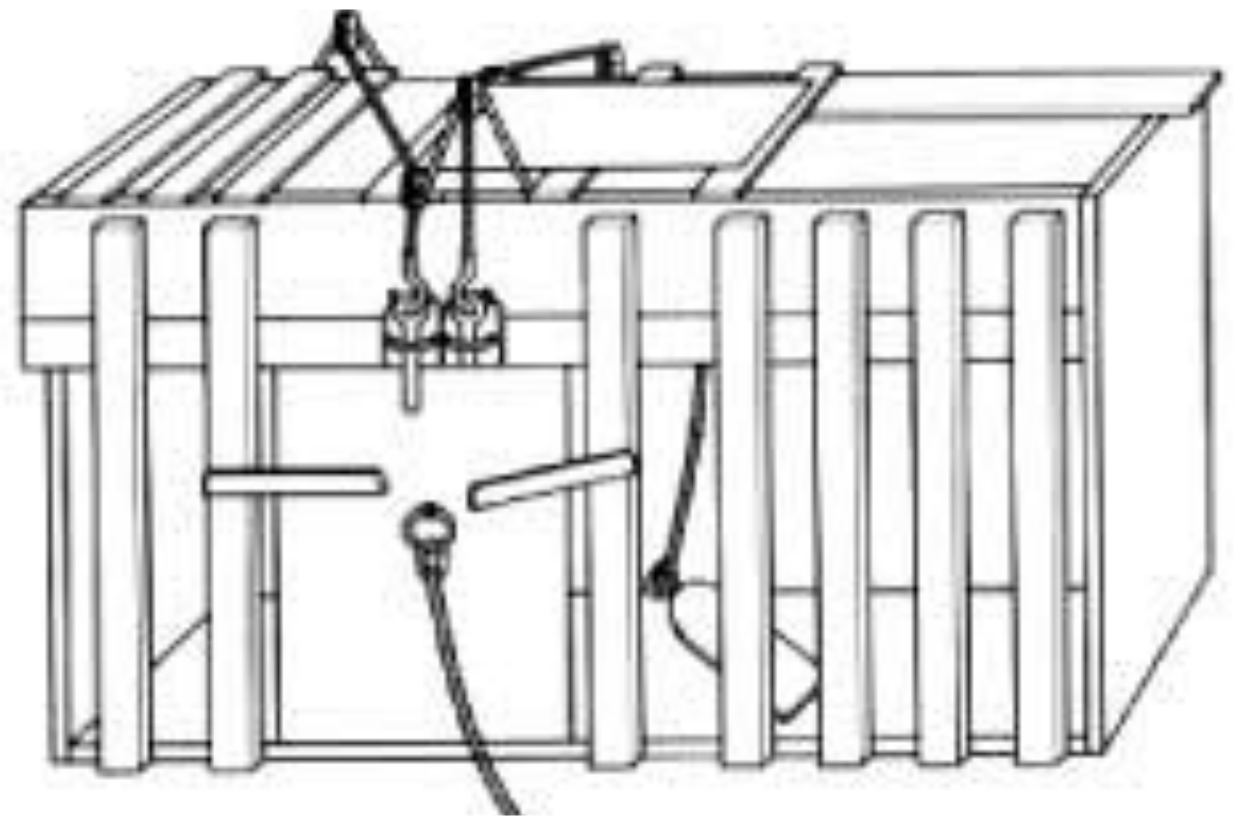


Brains control behavior

But how?



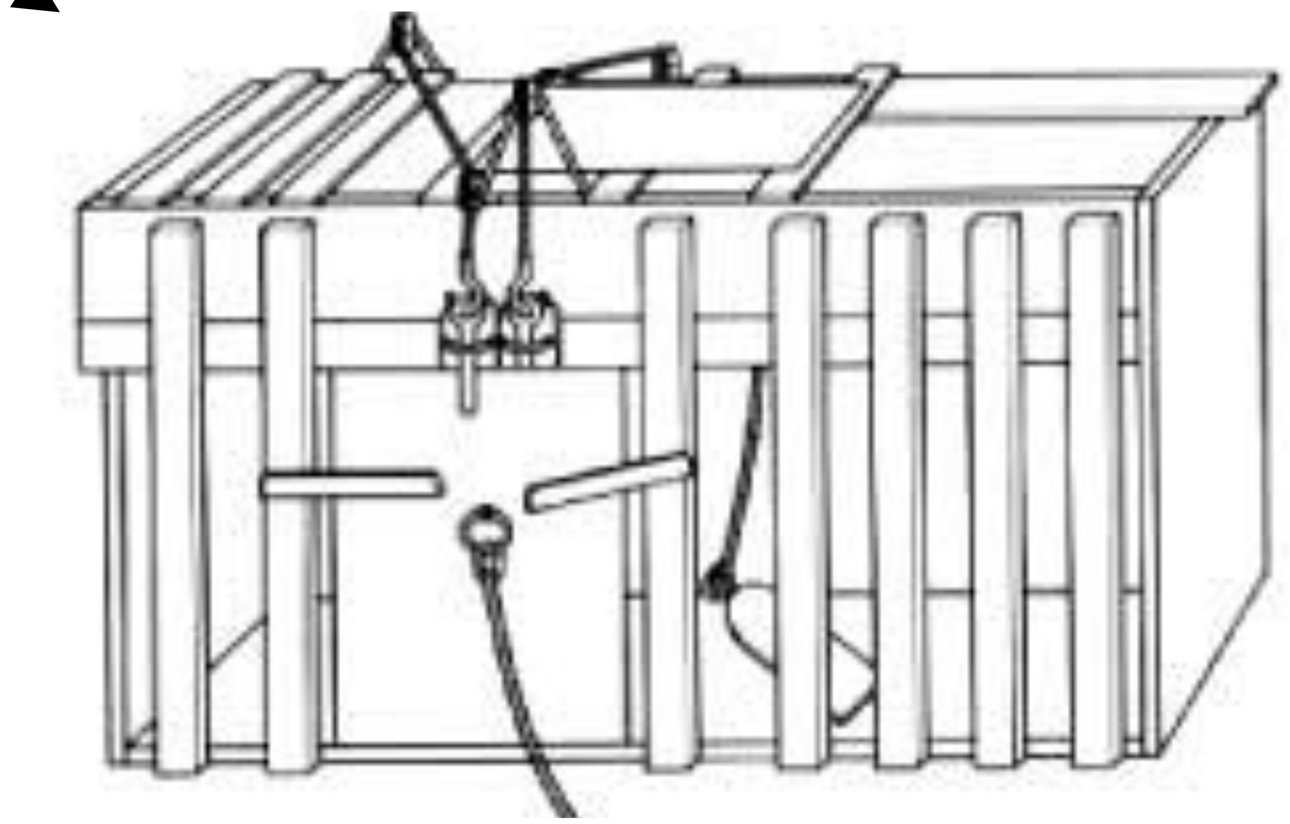
Thorndike's (1898) Law of Effect



Puzzle Box

Thorndike's (1898) Law of Effect

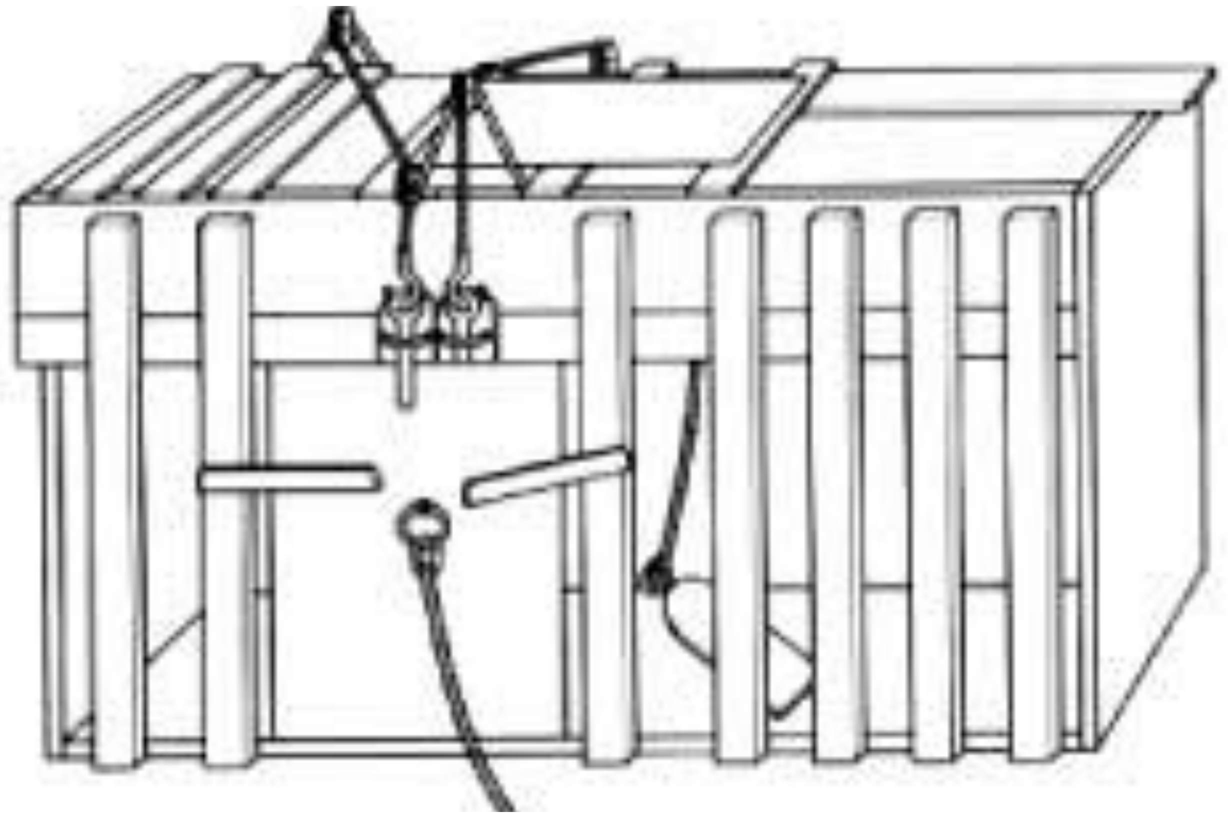
Cat



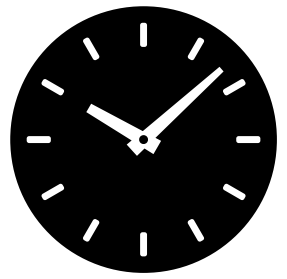
Puzzle Box

Thorndike’s (1898) Law of Effect

Cat



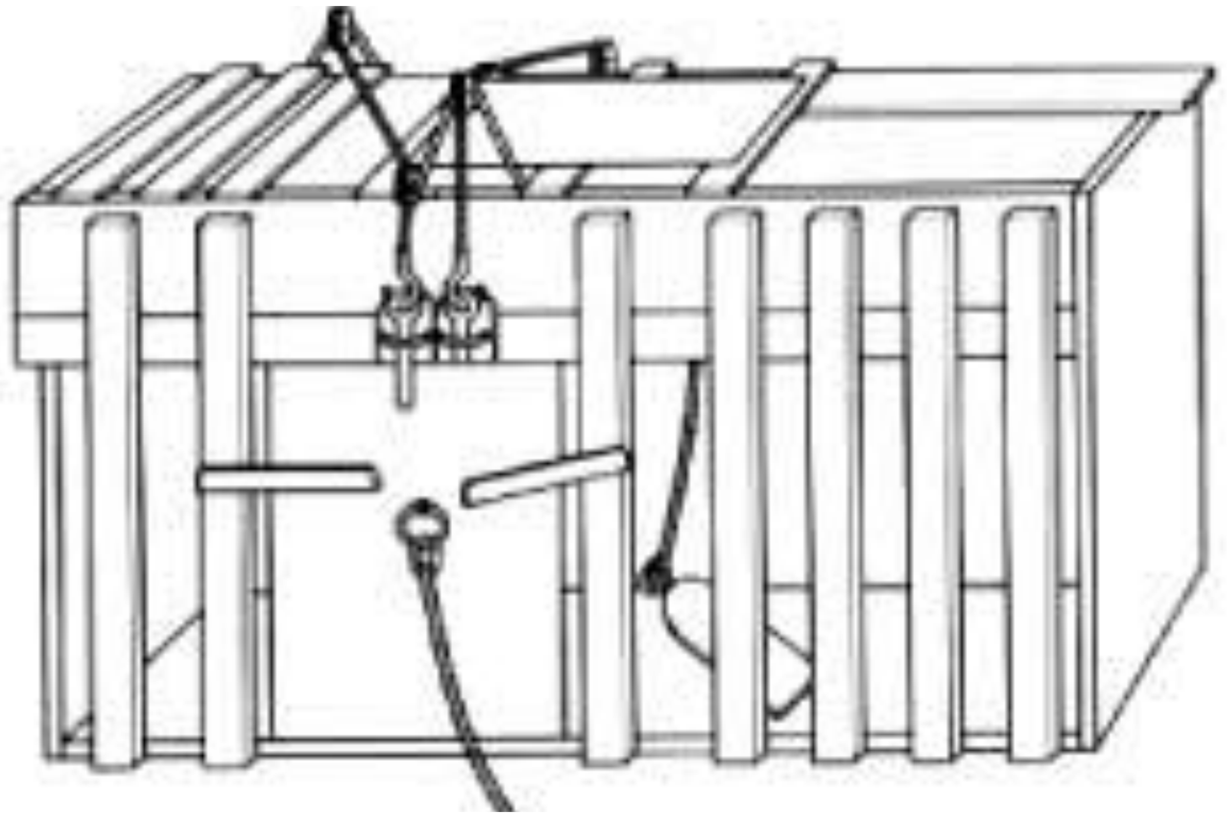
Puzzle Box



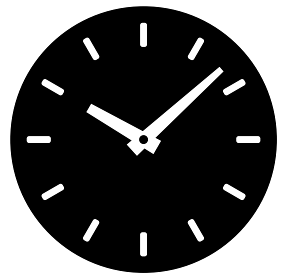
Time to escape

Thorndike’s (1898) Law of Effect

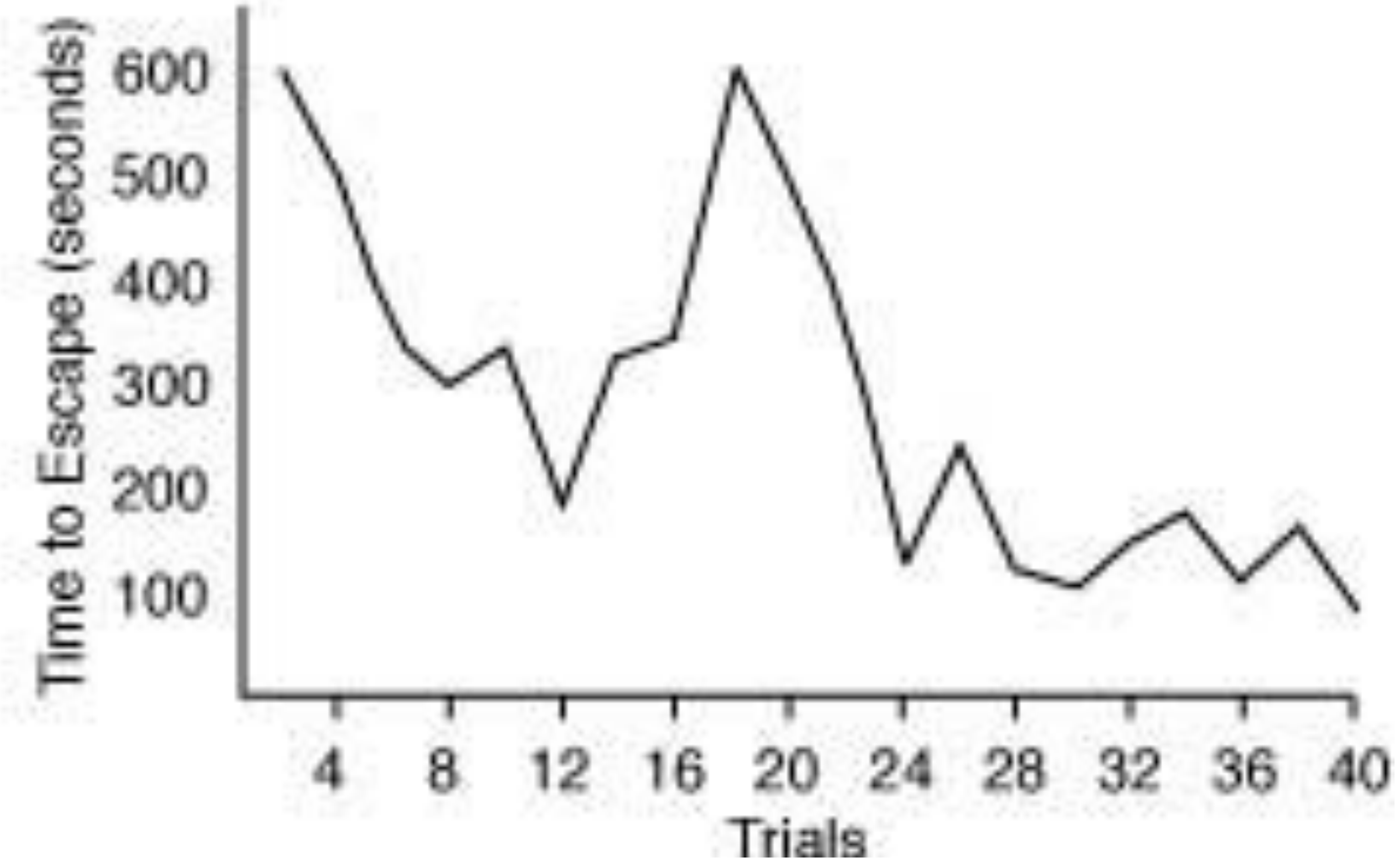
Cat



Puzzle Box



Time to escape

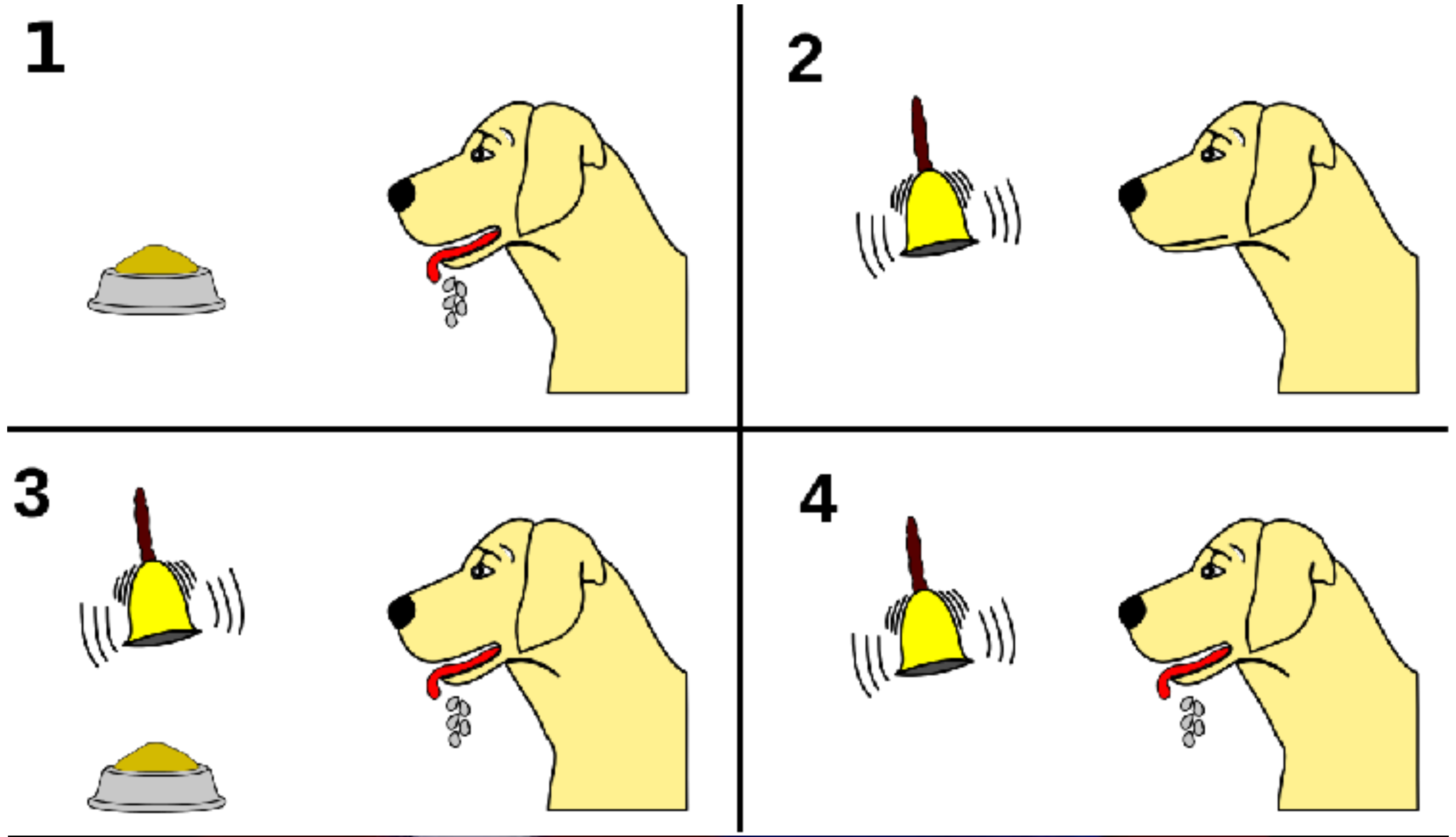


Actions associated with satisfaction are strengthened, while those associated with discomfort become weakened.

Classical and Operant Conditioning

Classical Condition (Pavlov, 1927)

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



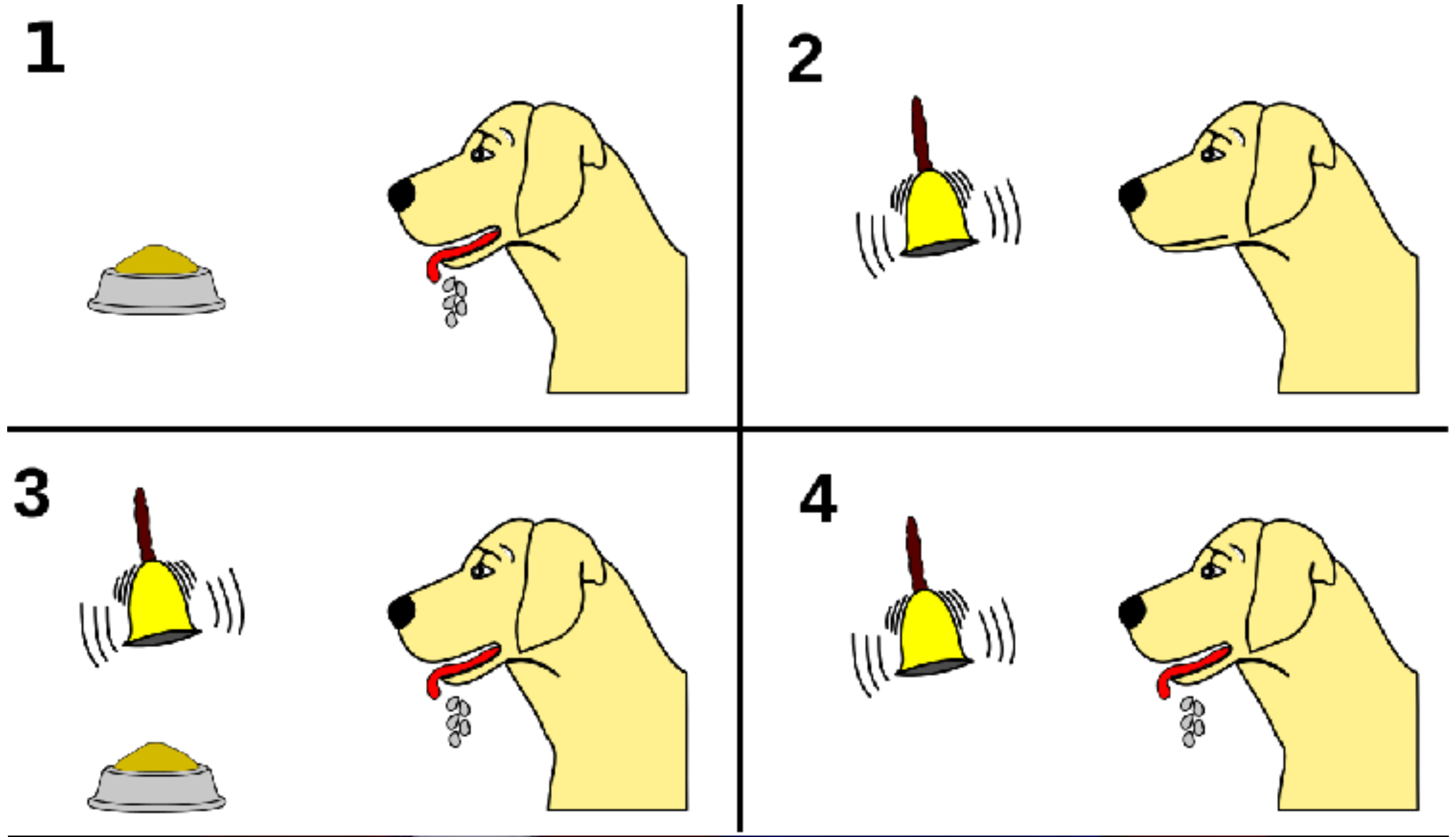
Operant Condition (Skinner, 1938)

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

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

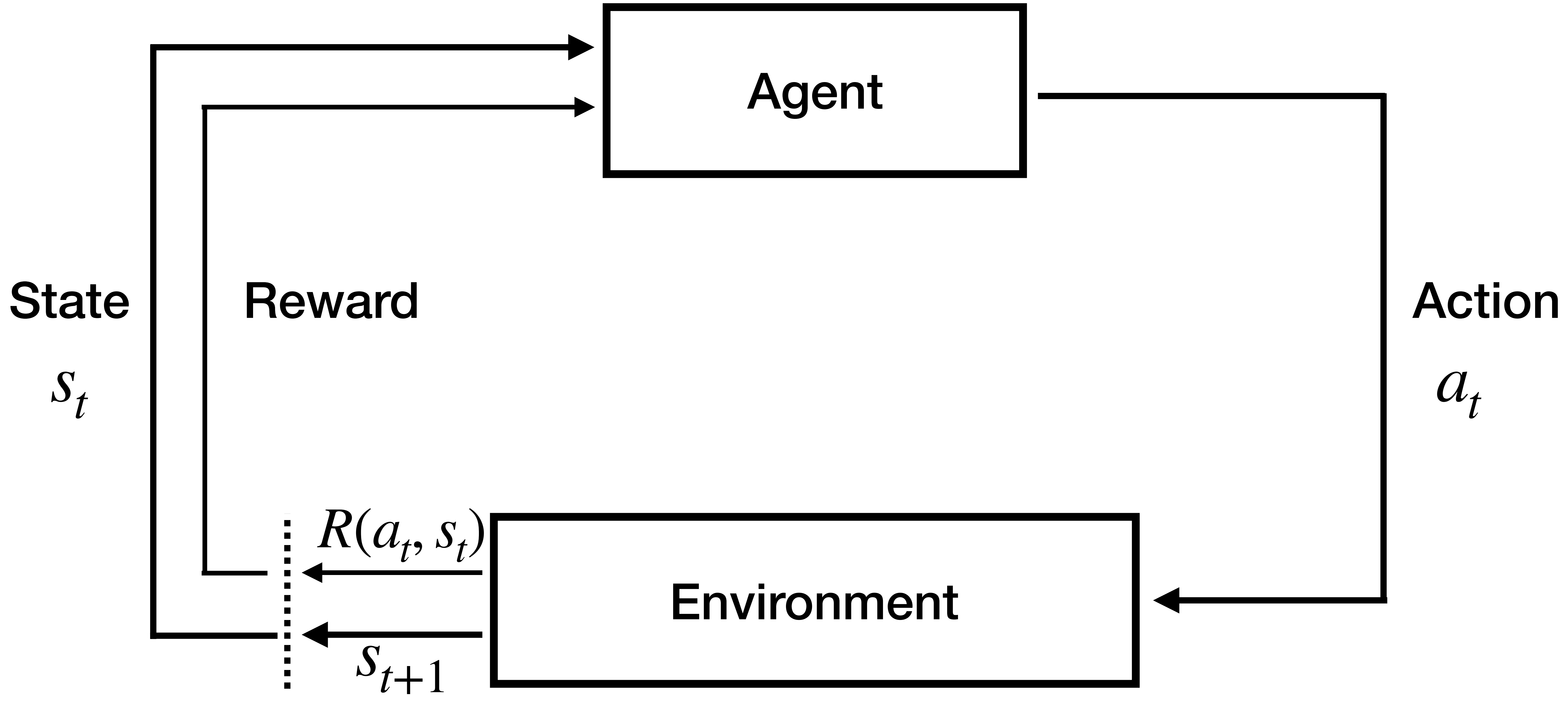
Reinforcement Learning

The Environment:

- governs the transition between states
- provides rewards

The Agent:

- Learns a *value function* mapping actions onto onto rewards
- Implements a policy, selecting actions based on their value

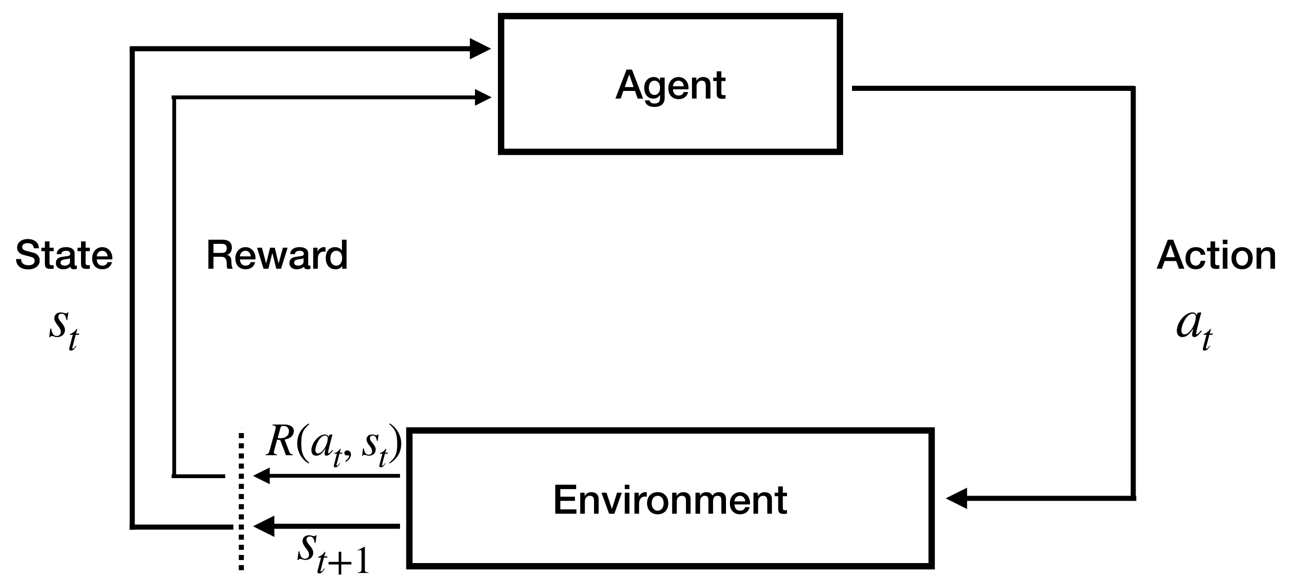


Sutton and Barto (2018 [1998])

Neuroscience

AI and Machine Learning

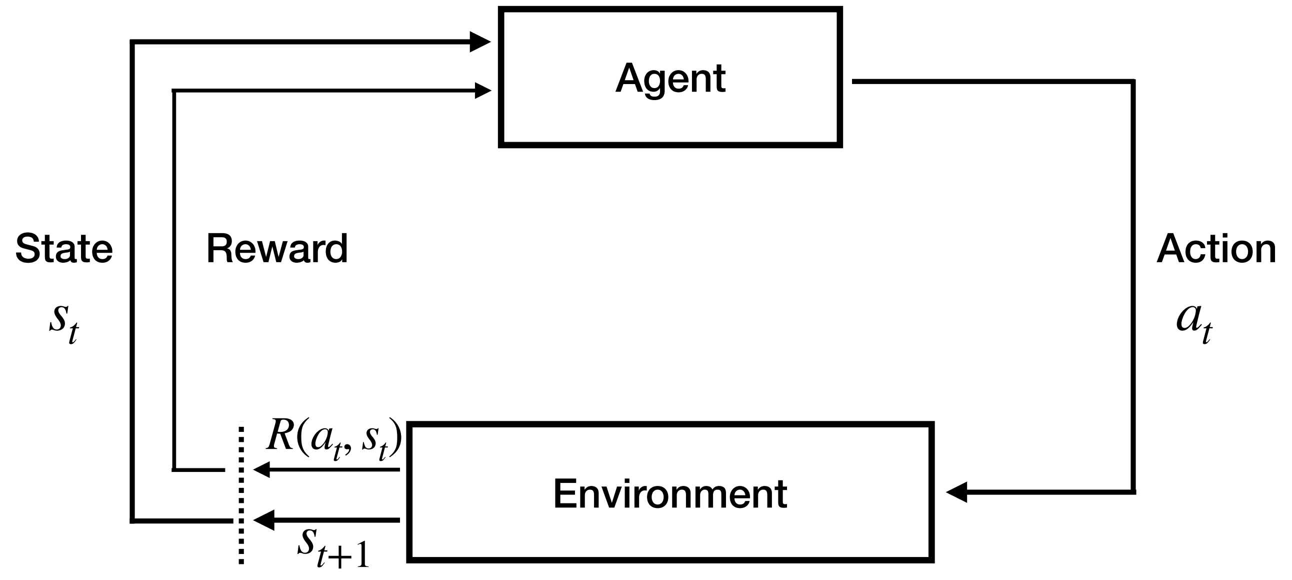
Reinforcement Learning



Neuroscience

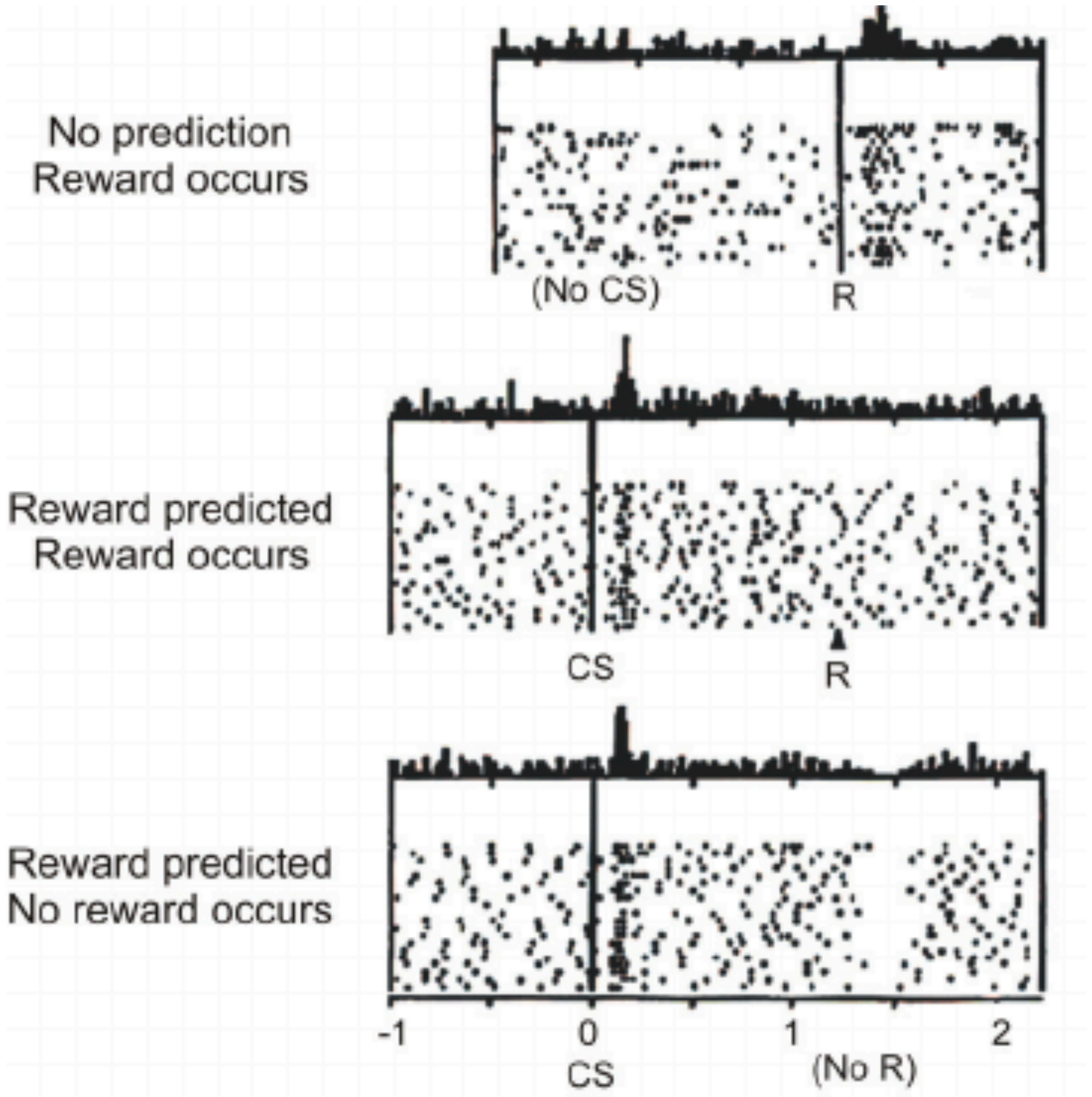
AI and Machine Learning

Reinforcement Learning



Temporal Difference Learning

Dopamine Reward Prediction Error

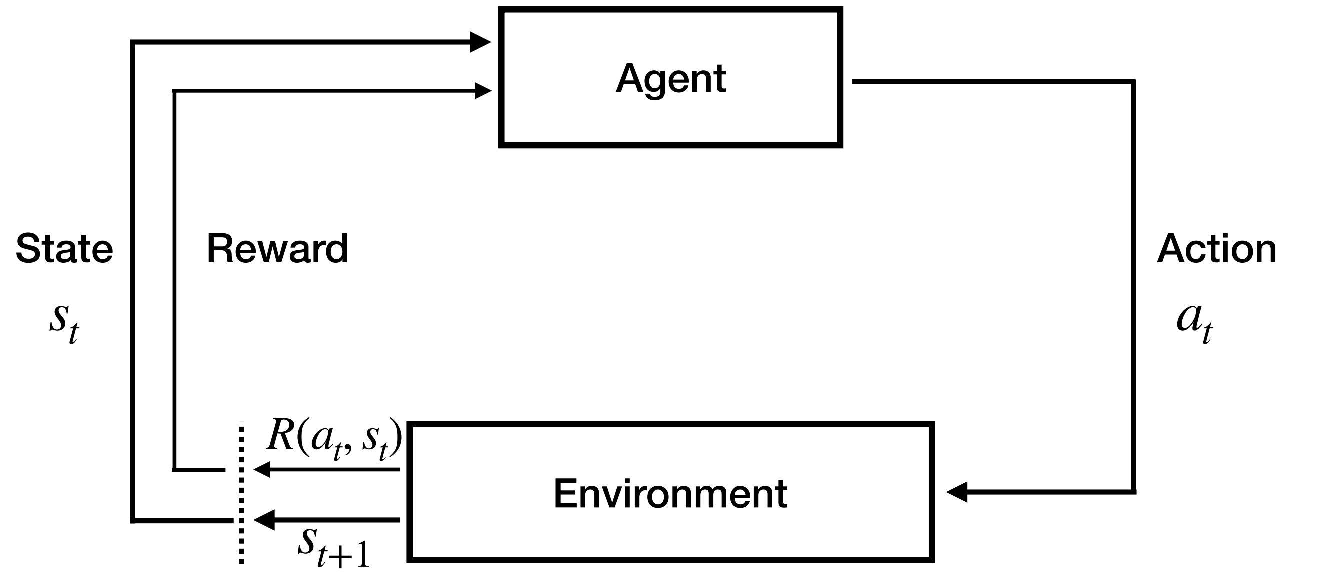


Schultz et al. (1997)

Neuroscience

AI and Machine Learning

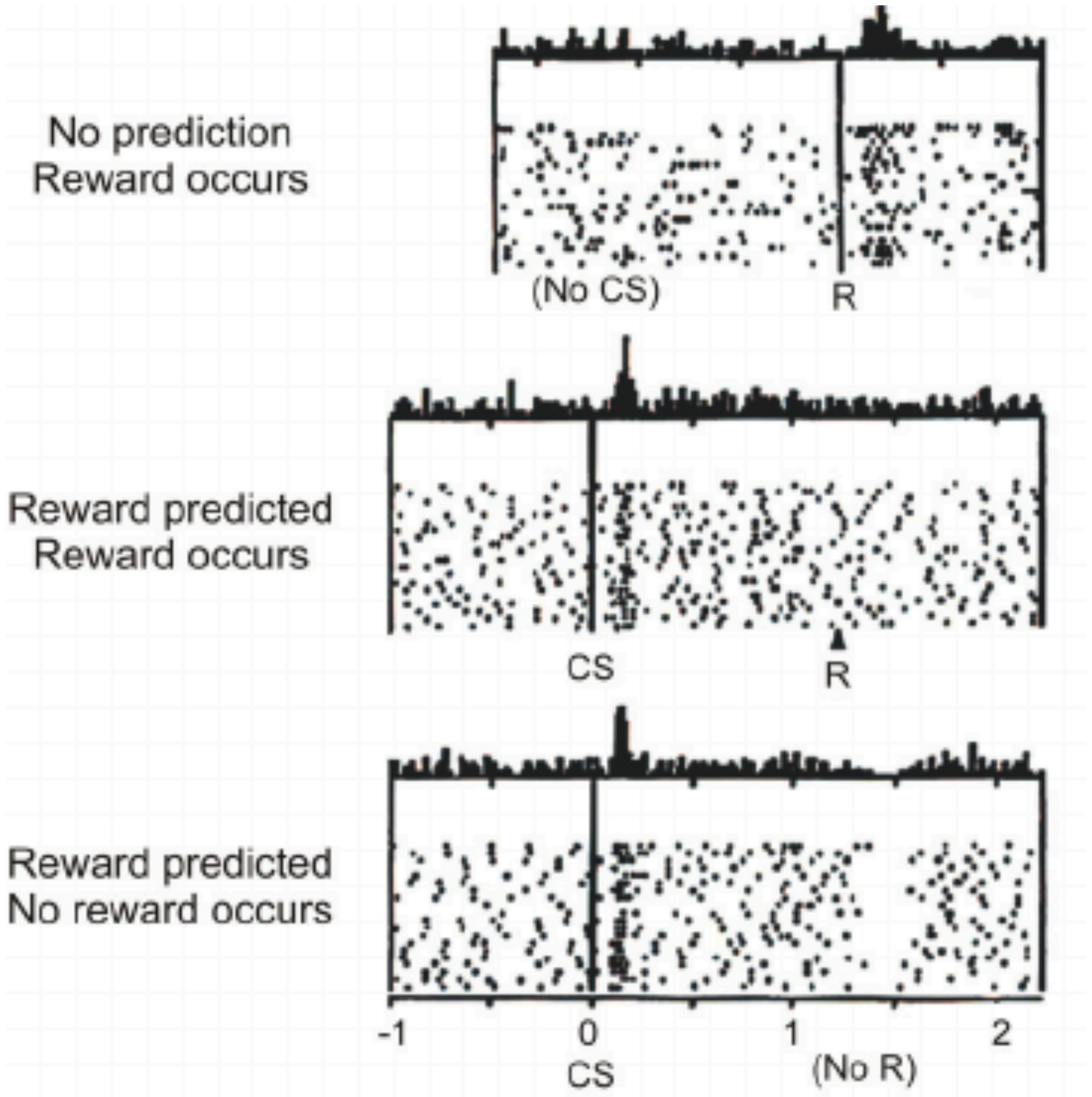
Reinforcement Learning



Temporal Difference Learning

Monte Carlo Tree Search

Dopamine Reward Prediction Error



Schultz et al. (1997)

AlphaGo

Silver et al. (2016)

Outline

Part 1. Overview of Reinforcement Learning

- Value functions and policies
- Tabular methods vs. value-function approximation
- Multi-armed Bandit problem
- Models of human learners

Outline

Part 1. Overview of Reinforcement Learning

- Value functions and policies
- Tabular methods vs. value-function approximation
- Multi-armed Bandit problem
- Models of human learners

~~~~~~~~~~~~~ Break ~~~~~~~~~~~~~
~~~~~~~~~~~~~

Outline

Part 1. Overview of Reinforcement Learning

- Value functions and policies
- Tabular methods vs. value-function approximation
- Multi-armed Bandit problem
- Models of human learners

~~~~~~~~~~~~~ Break ~~~~~~~~~~~~~

Part 2. Generalization guided learning

- Search in vast spaces (Wu et al., *NHB* 2018)
- Learning like a child (Schulz et al., *PsychSci* 2019; Meder et al., *DevSci* 2021; Giron et al., *in prep*)
- Connecting spatial and conceptual search (Wu et al., *PLoS CompBio* 2020)
- Graph-structured generalization (Wu et al., *CBB* 2020)
~~~~~~~~~~~~~

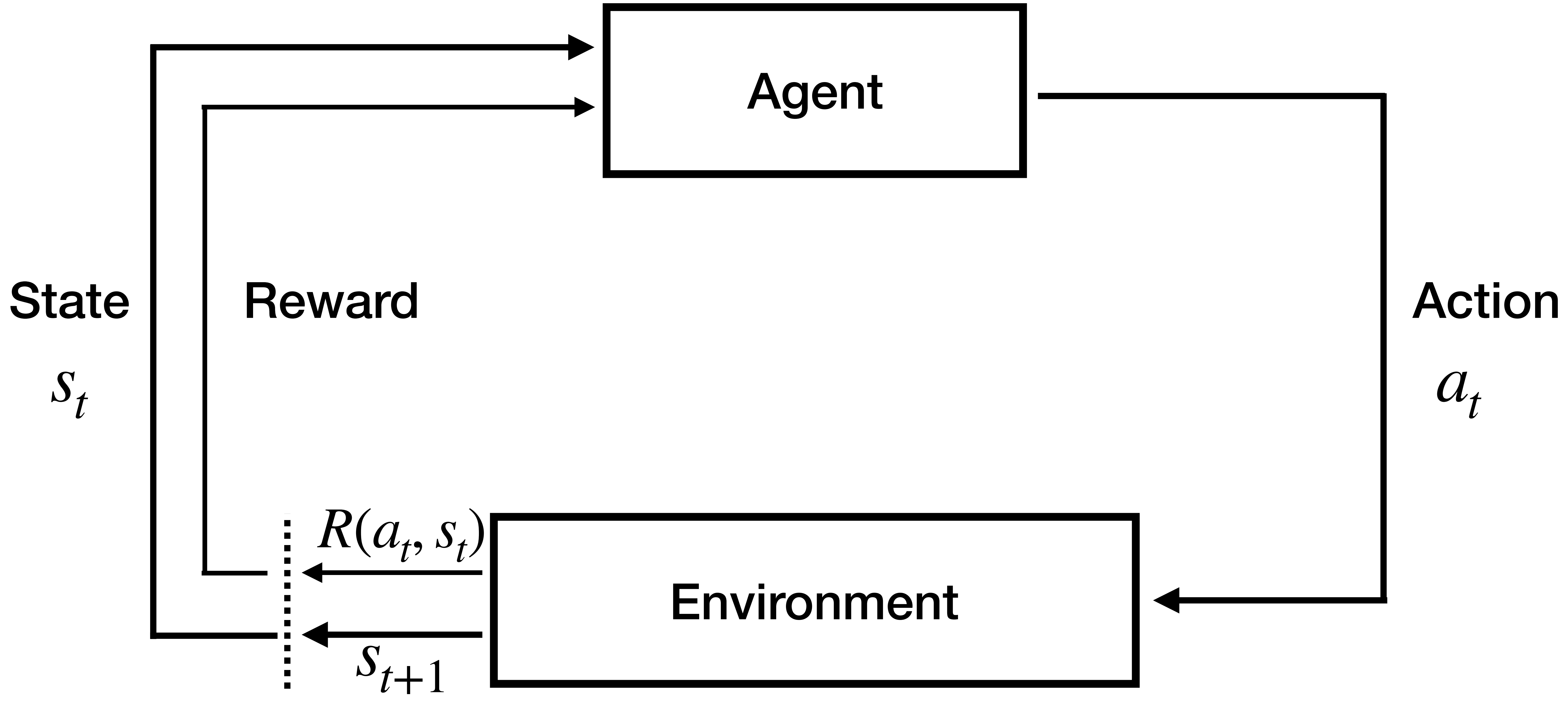
Reinforcement Learning

The Environment:

- governs the transition between states
- provides rewards

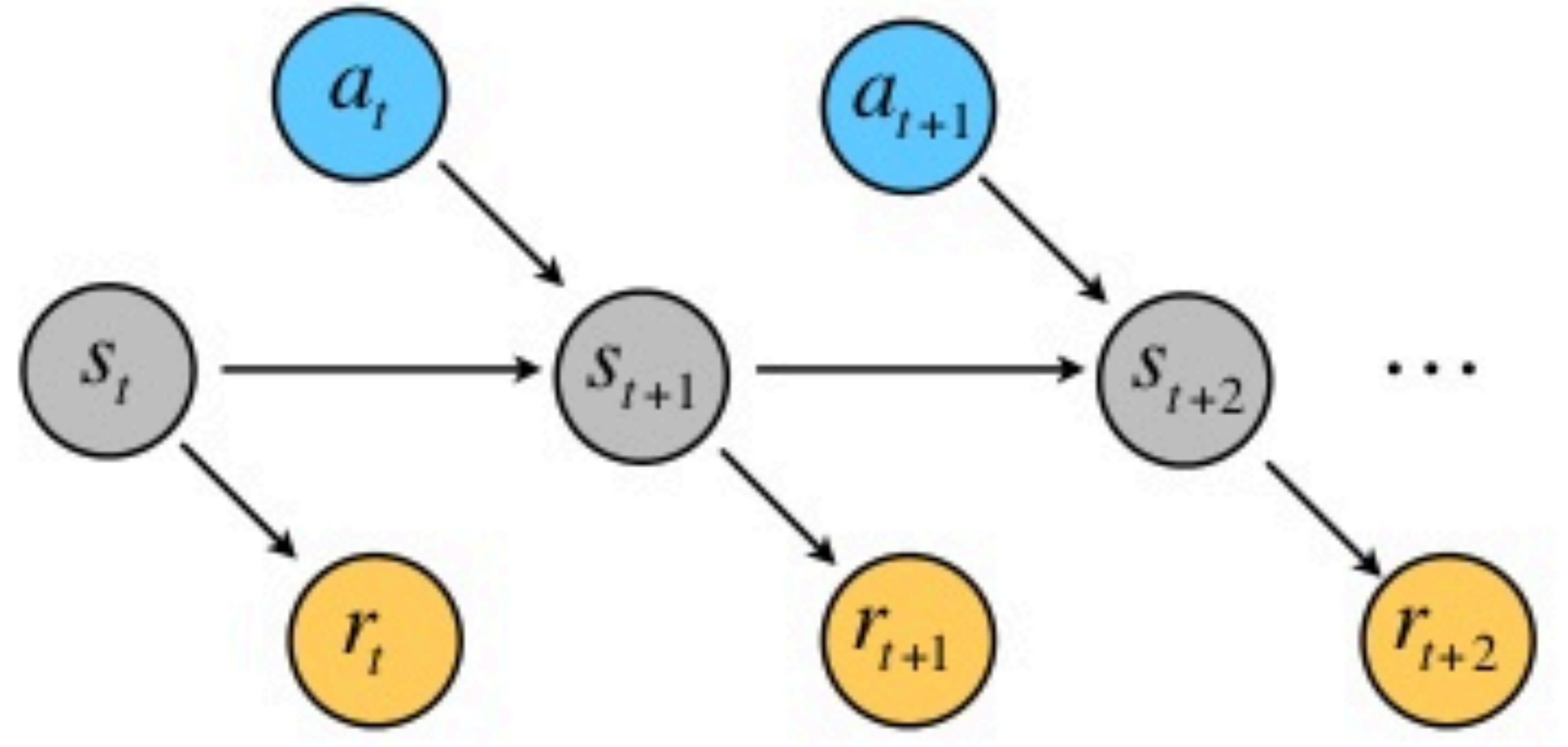
The Agent:

- Learns a *value function* mapping actions onto onto rewards
- Implements a policy, selecting actions based on their value



Sutton and Barto (2018 [1998])

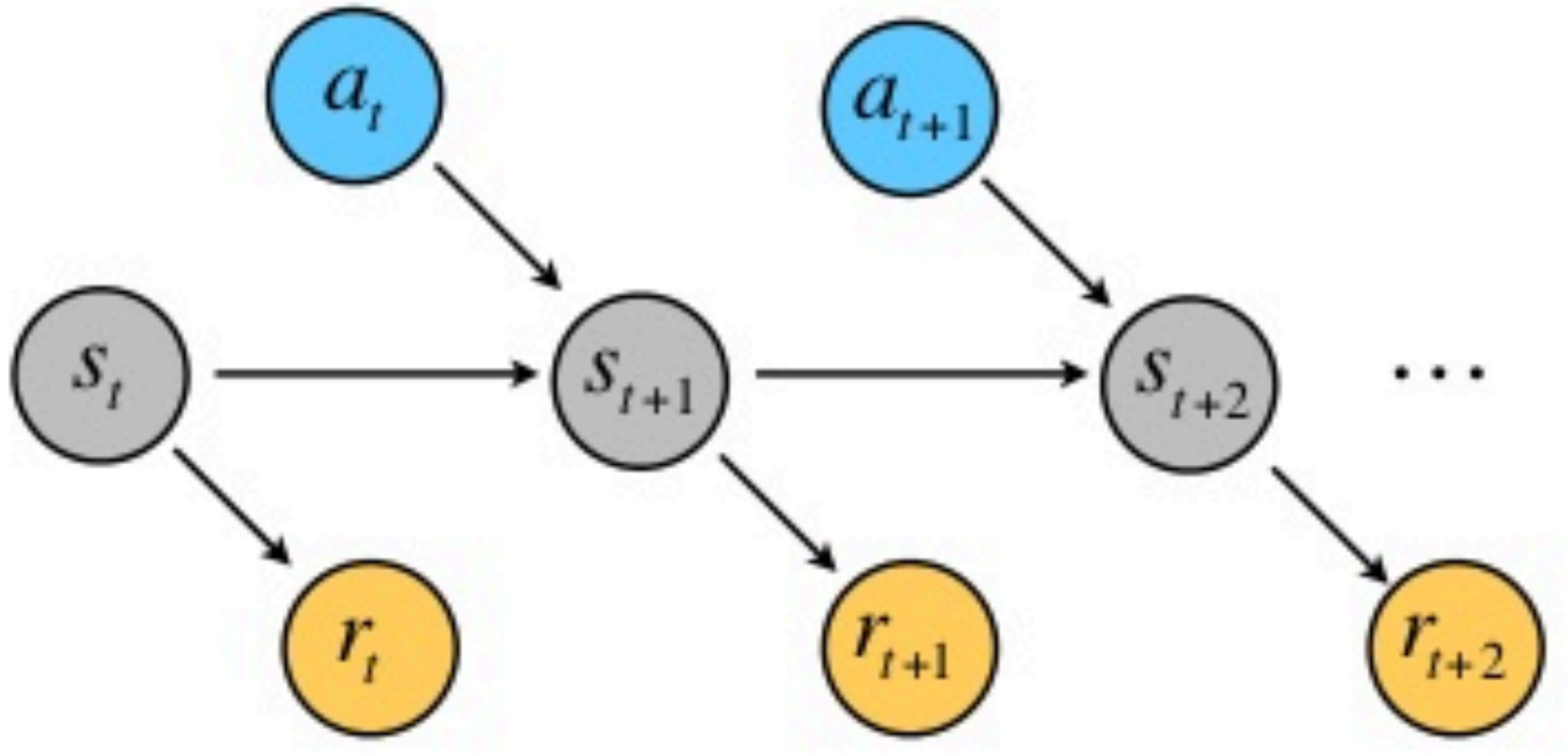
Environment



Markov Decision Process (MDP)

- Simplifying assumption that the system is fully defined by only the previous state (i.e., Markov Principle): $P(s_{t+1} \mid s_t, a_t)$

Environment

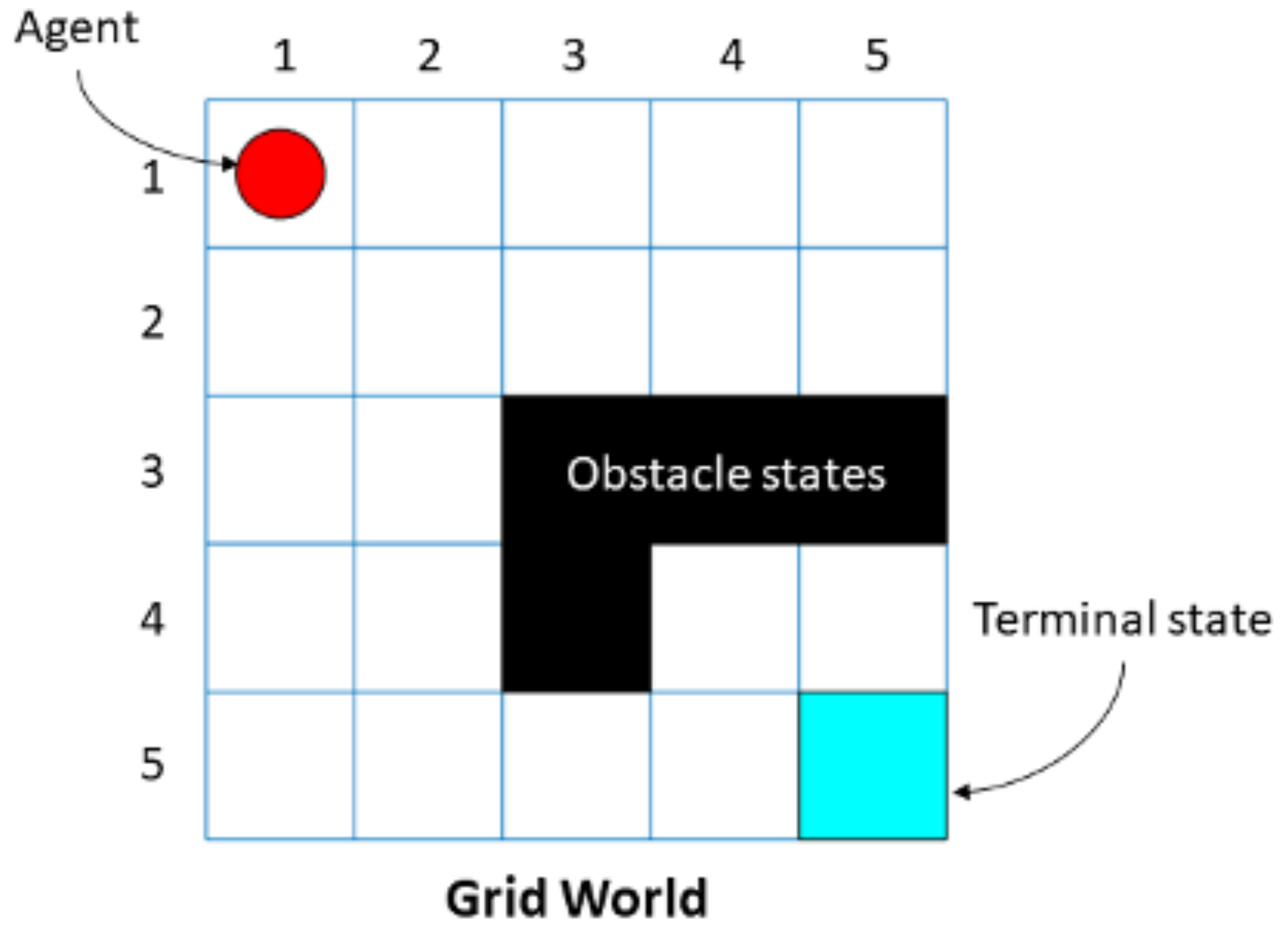


Markov Decision Process (MDP)

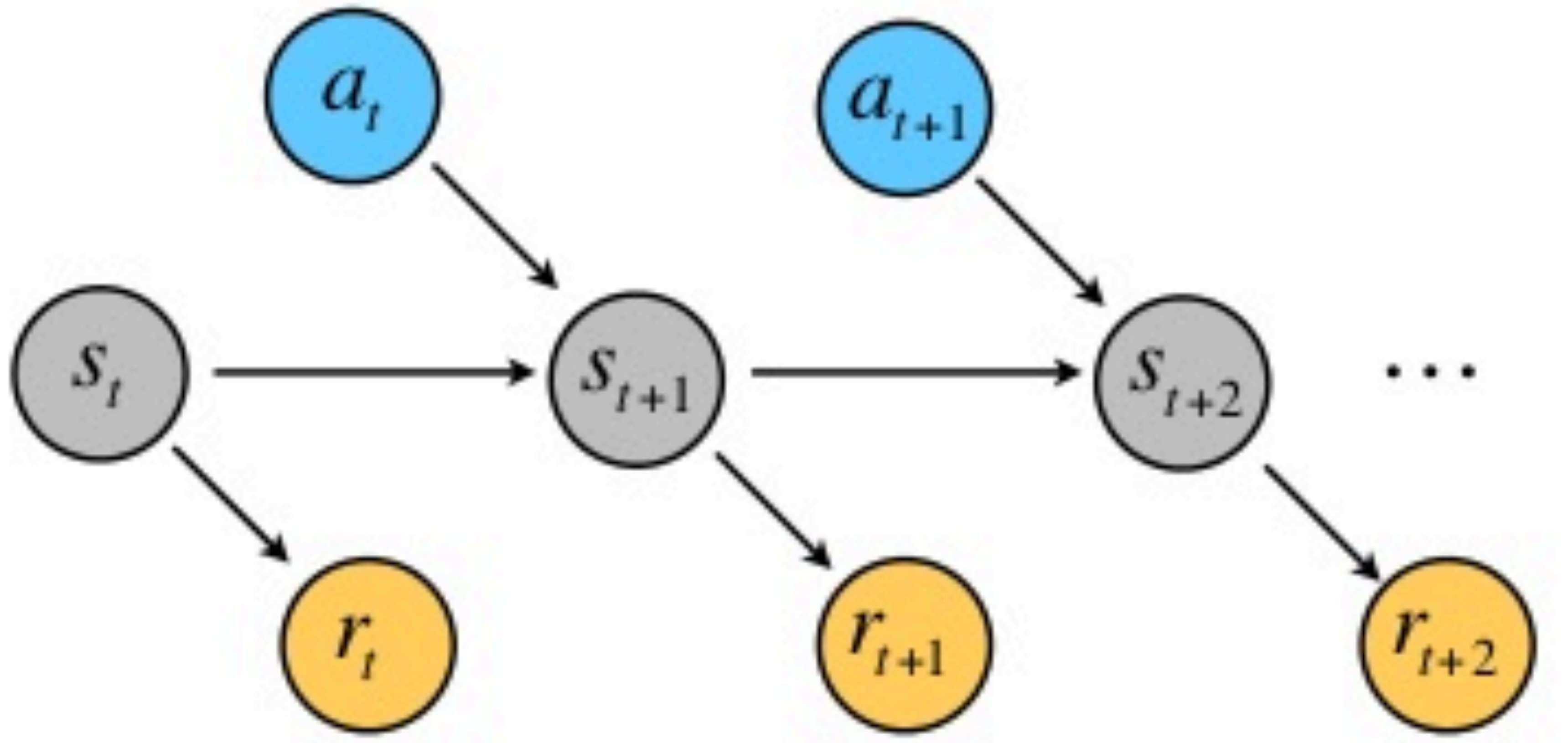
- Simplifying assumption that the system is fully defined by only the previous state (i.e., Markov Principle): $P(s_{t+1} \mid s_t, a_t)$

What are the states?

- Discrete locations, pixels on a screen, a set of feature values, etc…



Environment

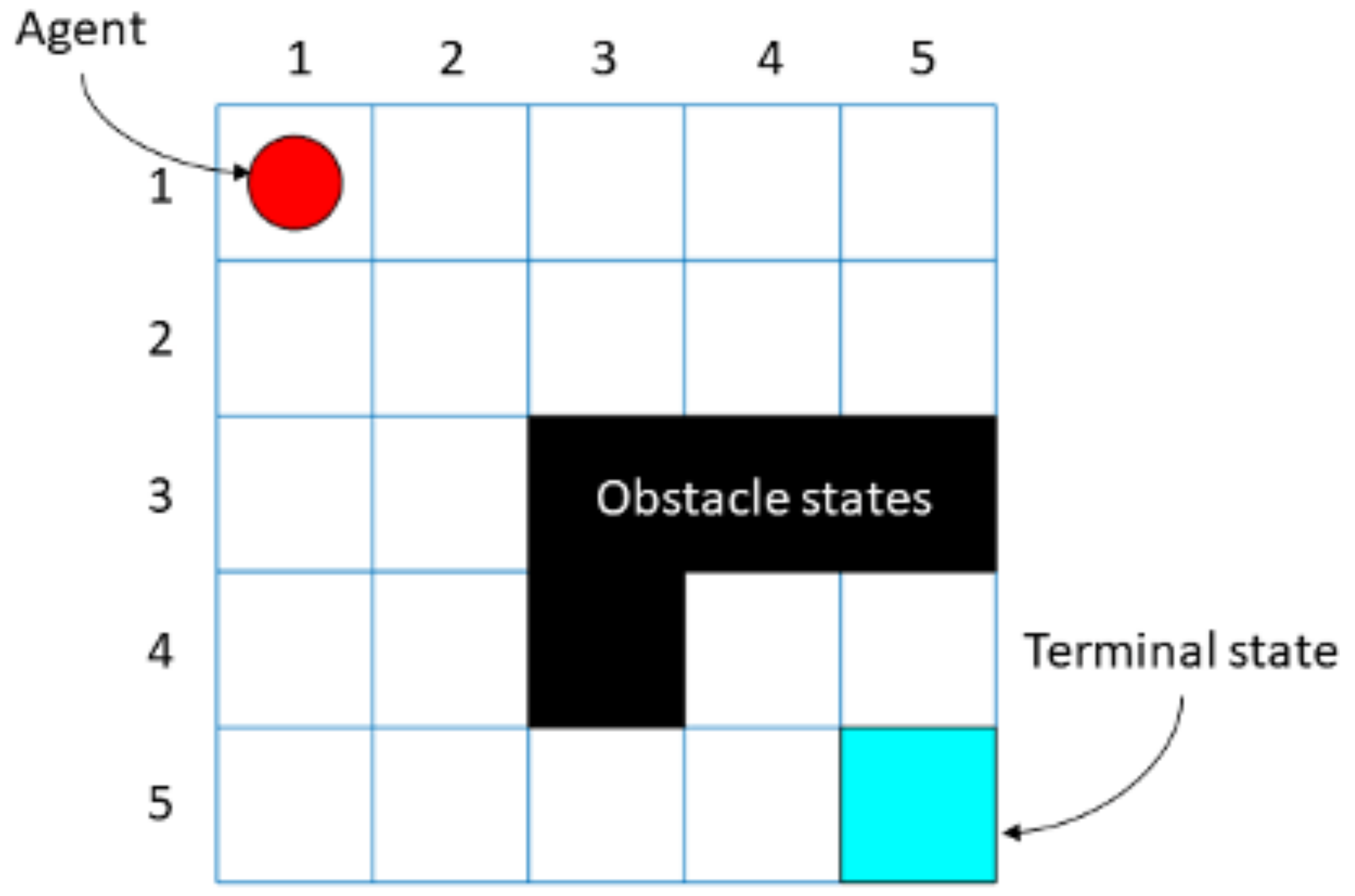


Markov Decision Process (MDP)

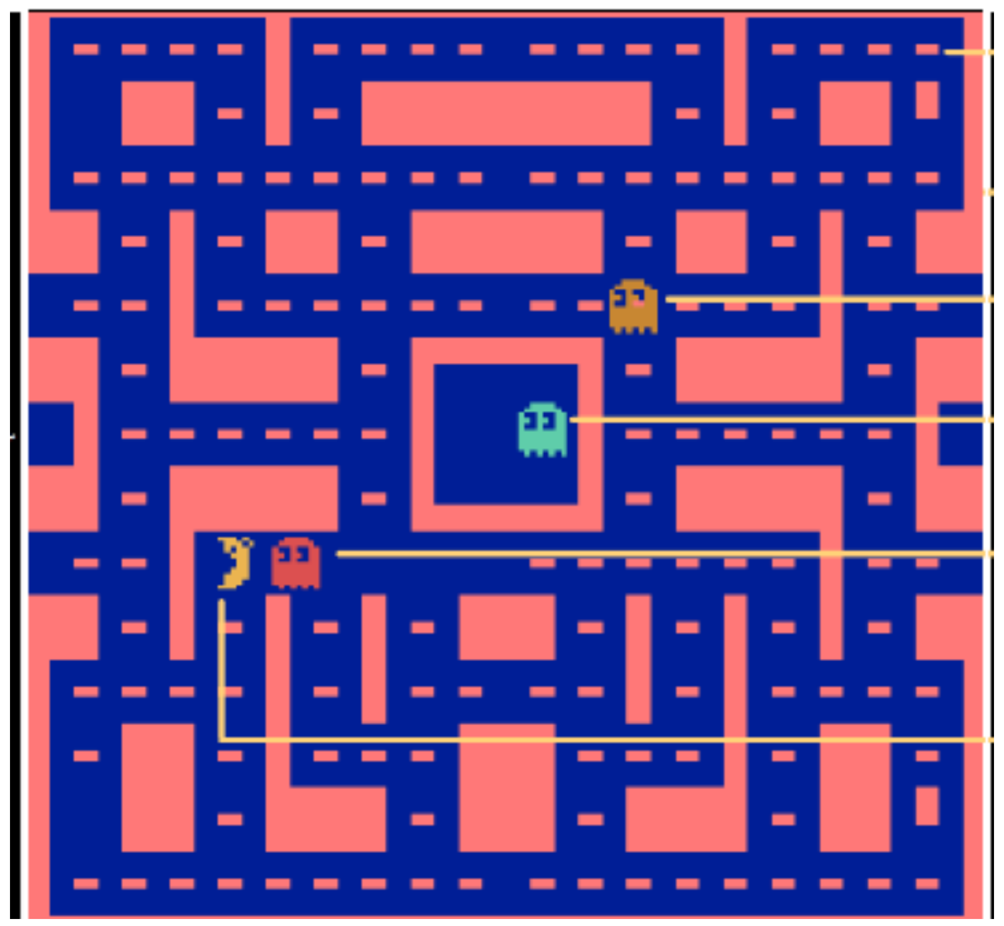
- Simplifying assumption that the system is fully defined by only the previous state (i.e., Markov Principle): $P(s_{t+1} \mid s_t, a_t)$

What are the states?

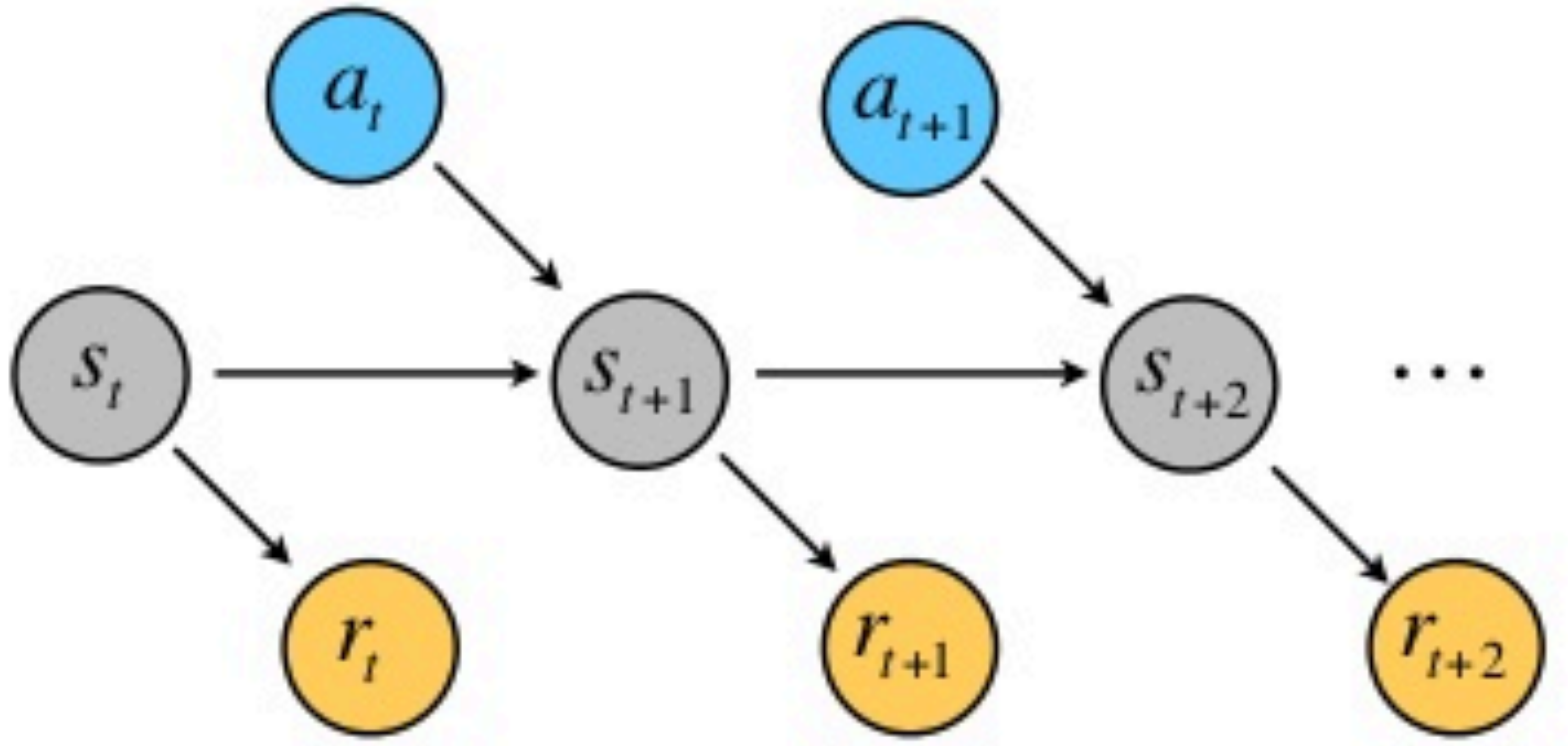
- Discrete locations, pixels on a screen, a set of feature values, etc…



Grid World



Environment

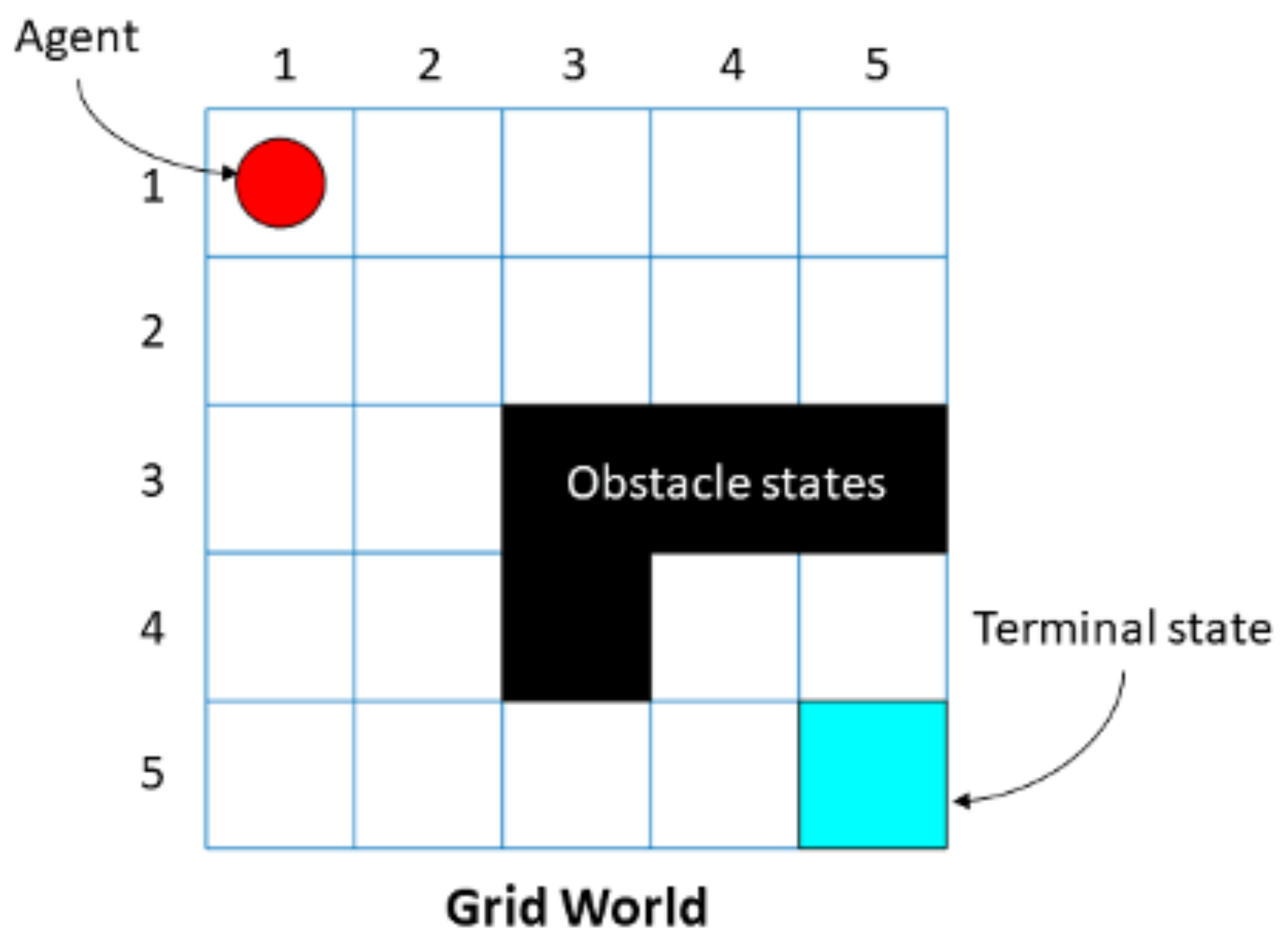


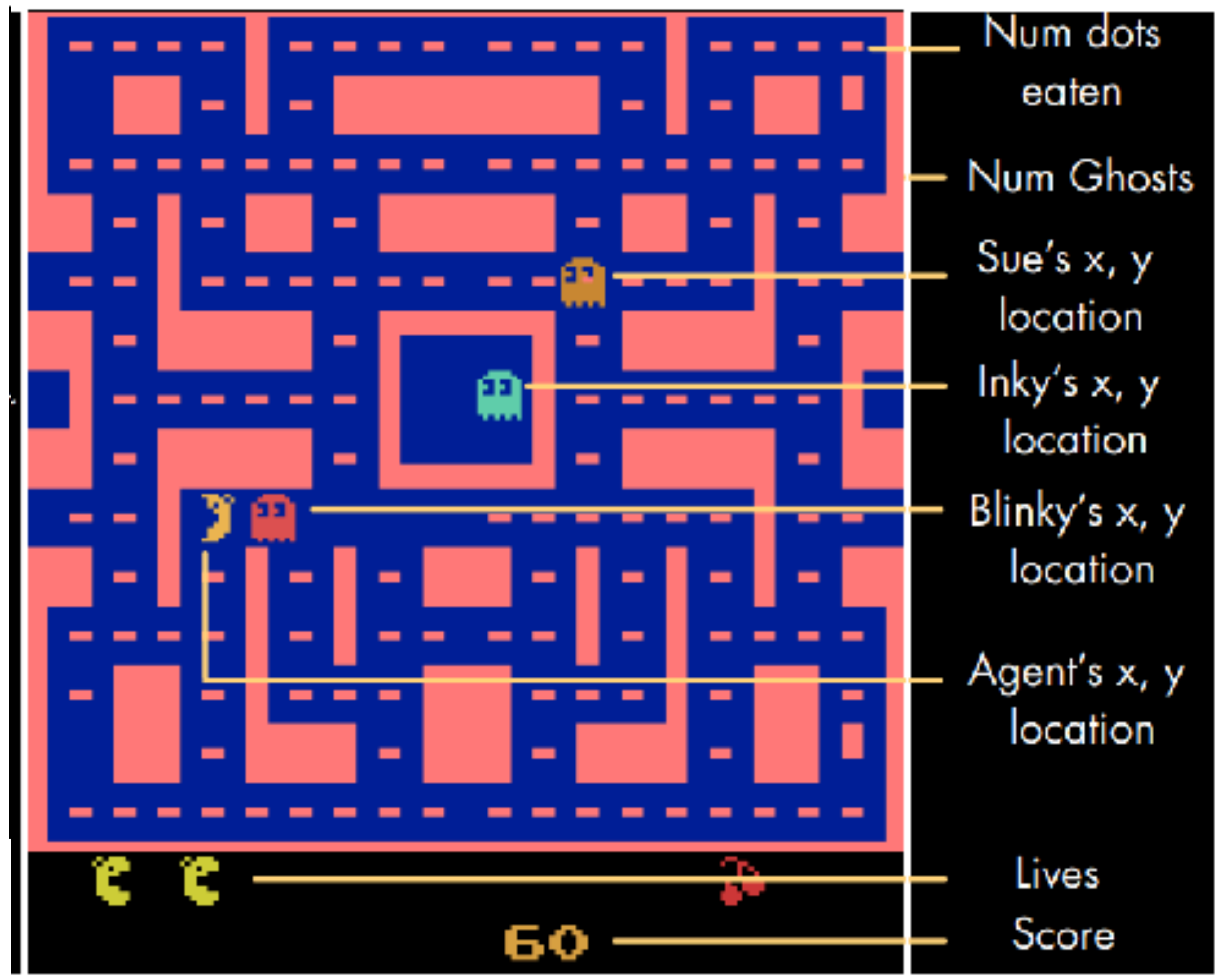
Markov Decision Process (MDP)

- Simplifying assumption that the system is fully defined by only the previous state (i.e., Markov Principle): $P(s_{t+1} \mid s_t, a_t)$

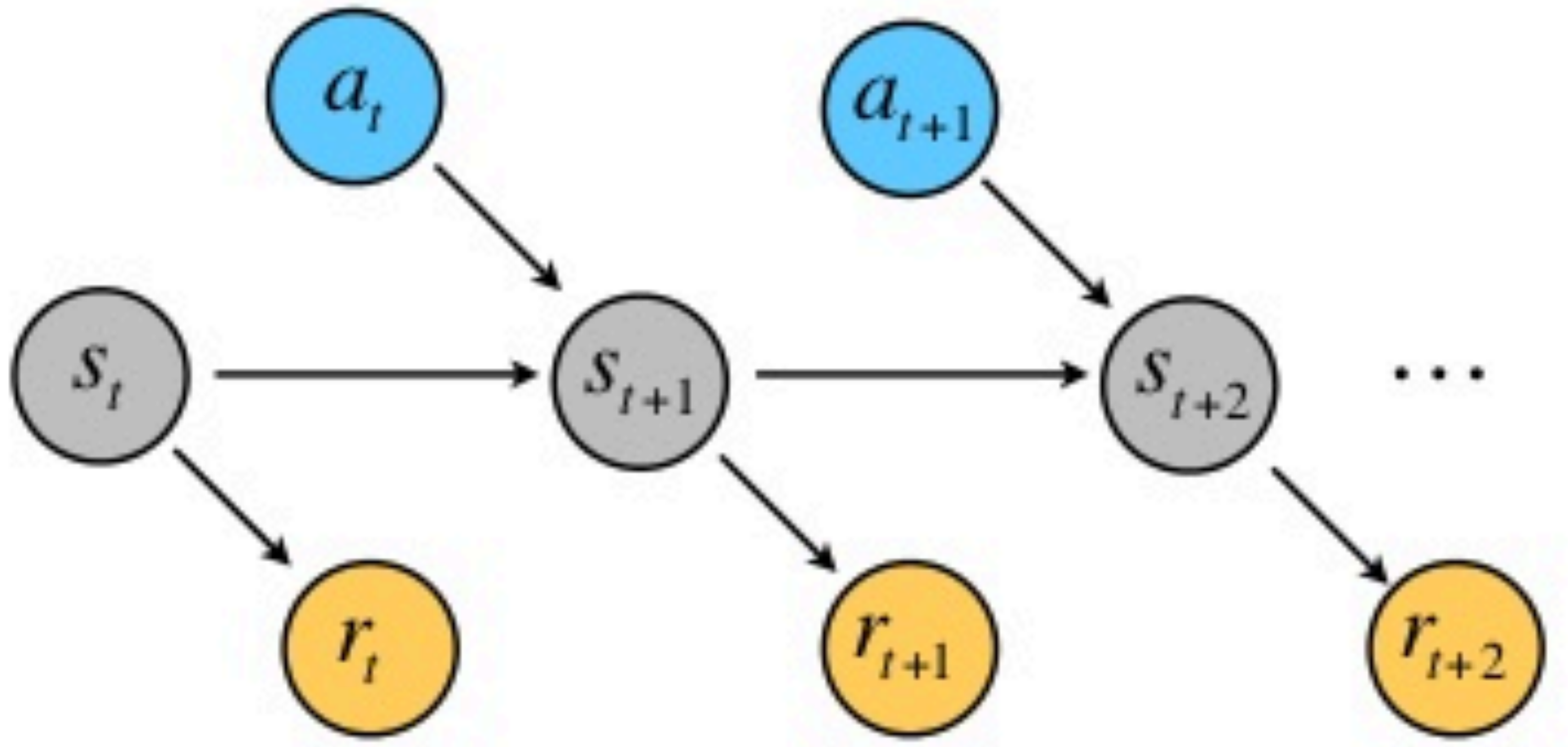
What are the states?

- Discrete locations, pixels on a screen, a set of feature values, etc…





Environment

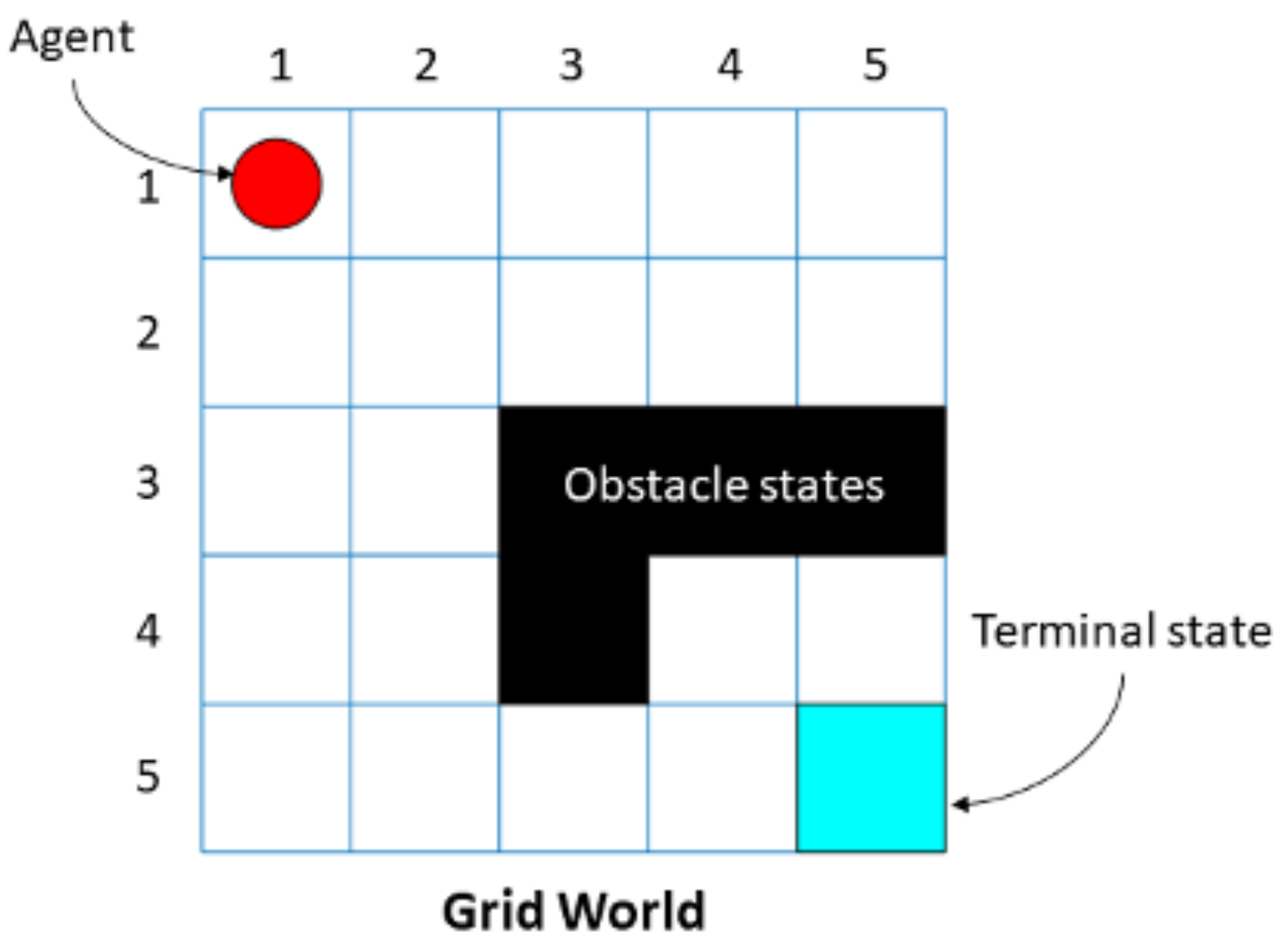


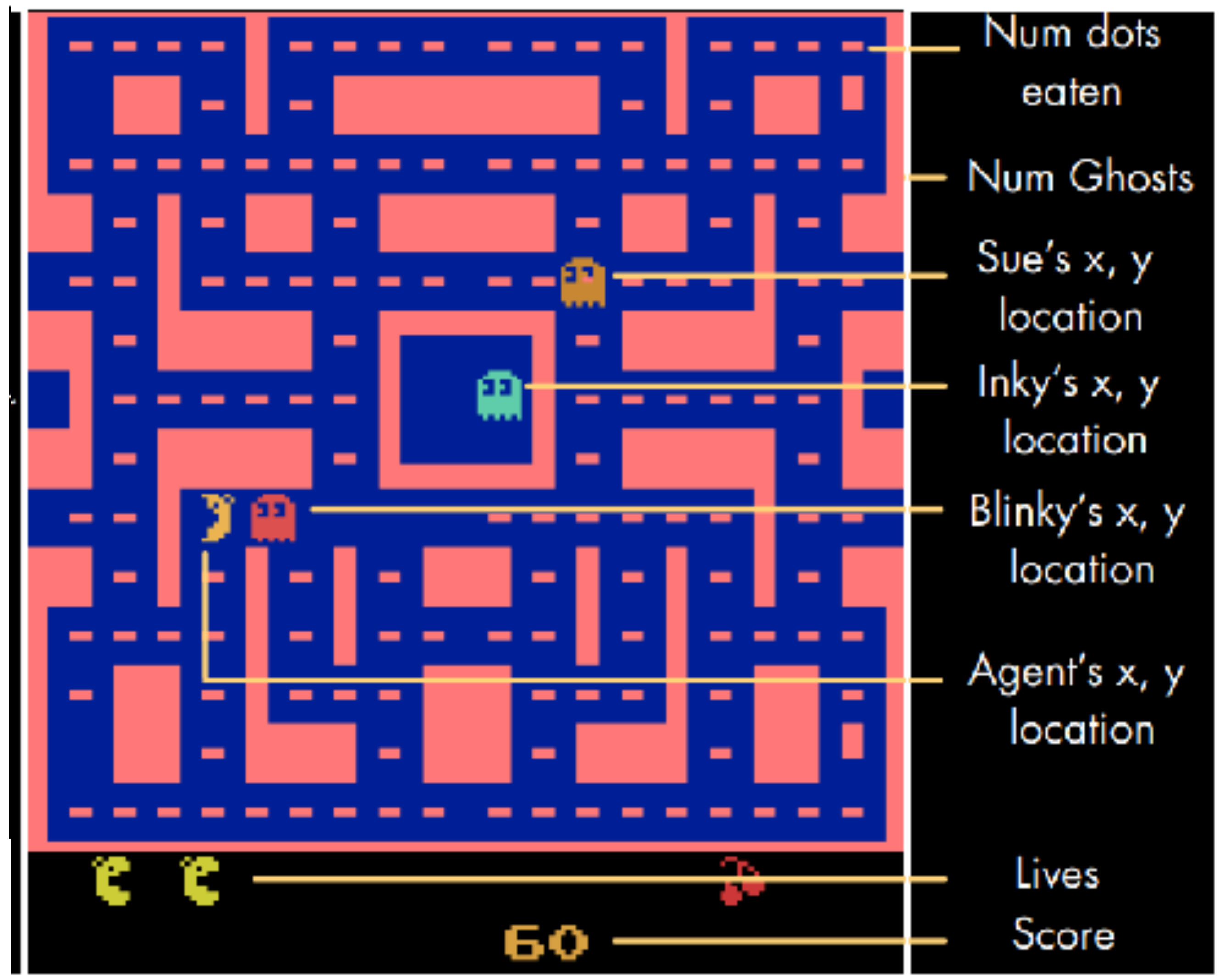
Markov Decision Process (MDP)

- Simplifying assumption that the system is fully defined by only the previous state (i.e., Markov Principle): $P(s_{t+1} \mid s_t, a_t)$

What are the states?

- Discrete locations, pixels on a screen, a set of feature values, etc…





Partially Observable MDP (POMDP)

Agent

Agent

- **Experiences Rewards**

- **Learns a Policy**

Agent

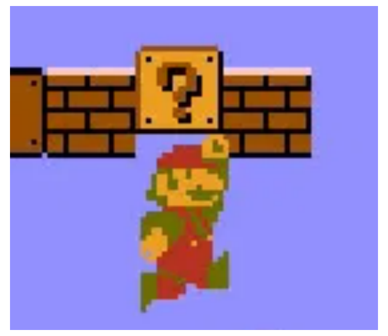
- **Experiences Rewards**
  - How good is a given state? $r_t = R(s_t)$

- **Learns a Policy**

Agent

- **Experiences Rewards**

  - How good is a given state? $r_t = R(s_t)$
  - How good is a state-action pair? $r_t = R(s_t, a_t)$

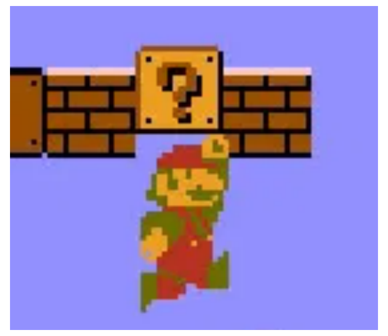


- **Learns a Policy**

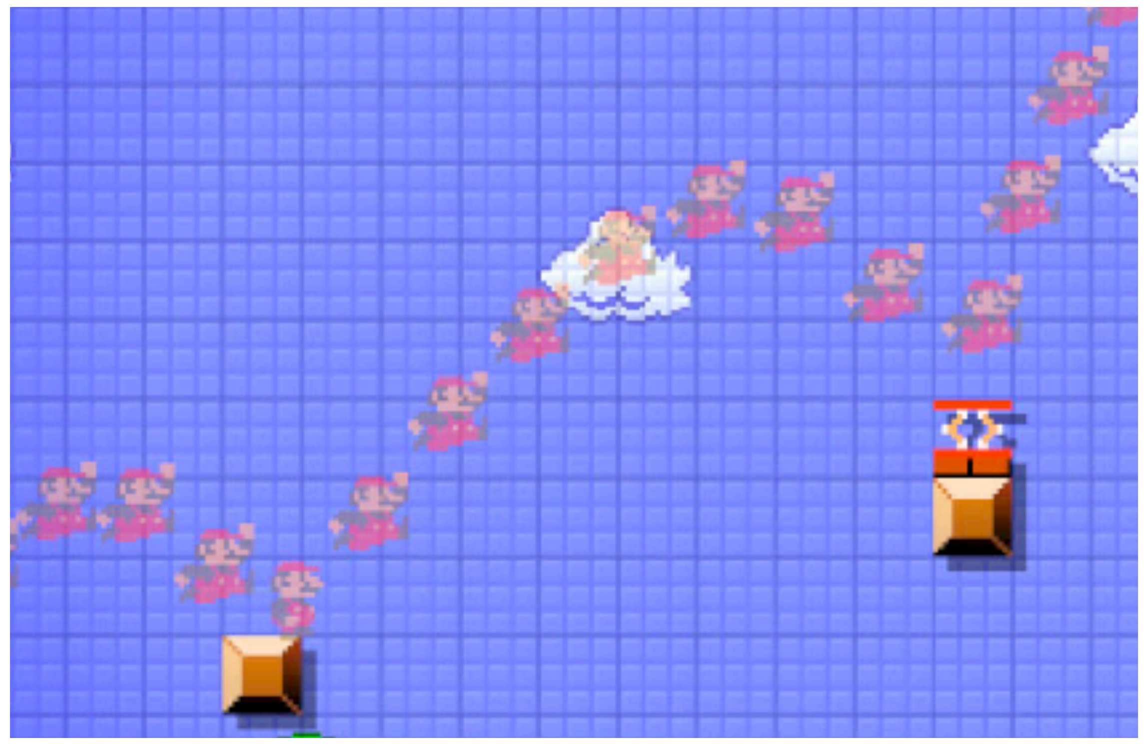
Agent

- **Experiences Rewards**

  - How good is a given state? $r_t = R(s_t)$



  - How good is a state-action pair? $r_t = R(s_t, a_t)$



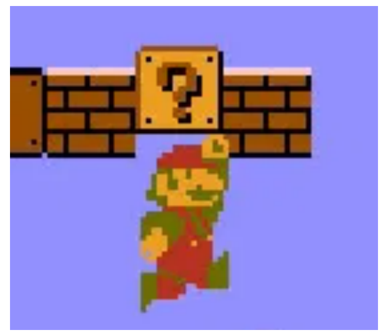
  - How good is a *trajectory* $\tau = (s_0, a_0, s_1, a_1, \ldots)$:

    $R(\tau) = \sum_{t=0}^{\infty} \gamma^k r_t$ where $\gamma \in [0,1]$ is the temporal discount

- **Learns a Policy**

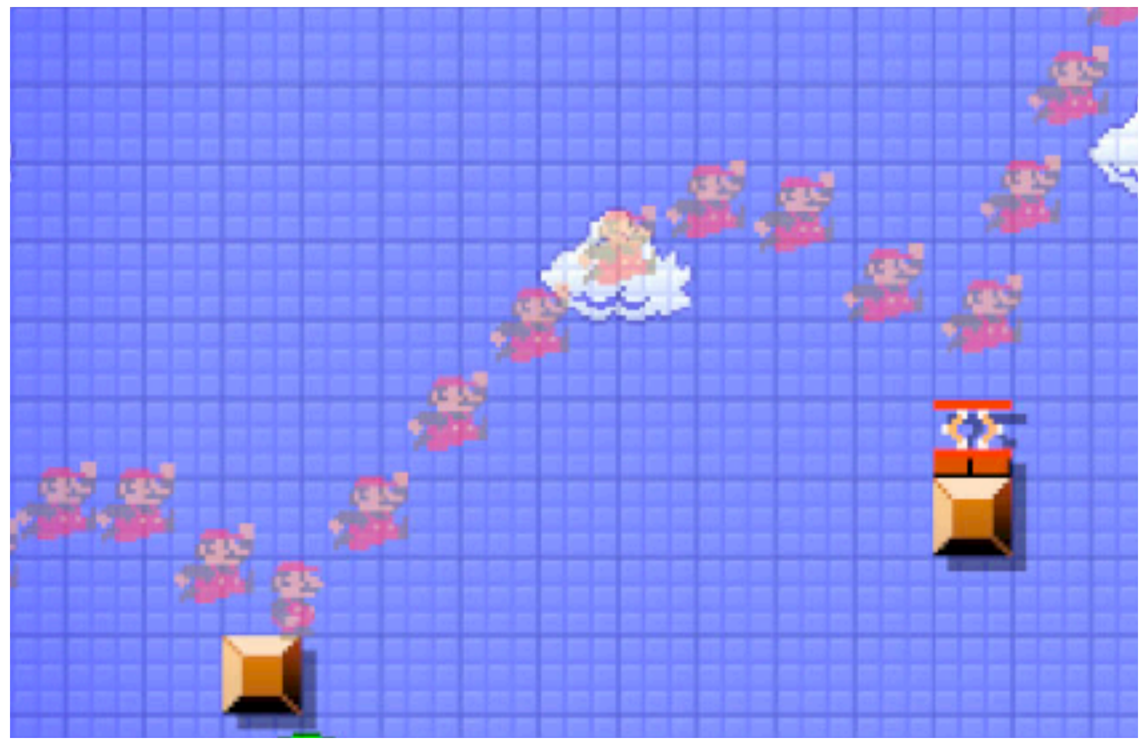
Agent

- **Experiences Rewards**

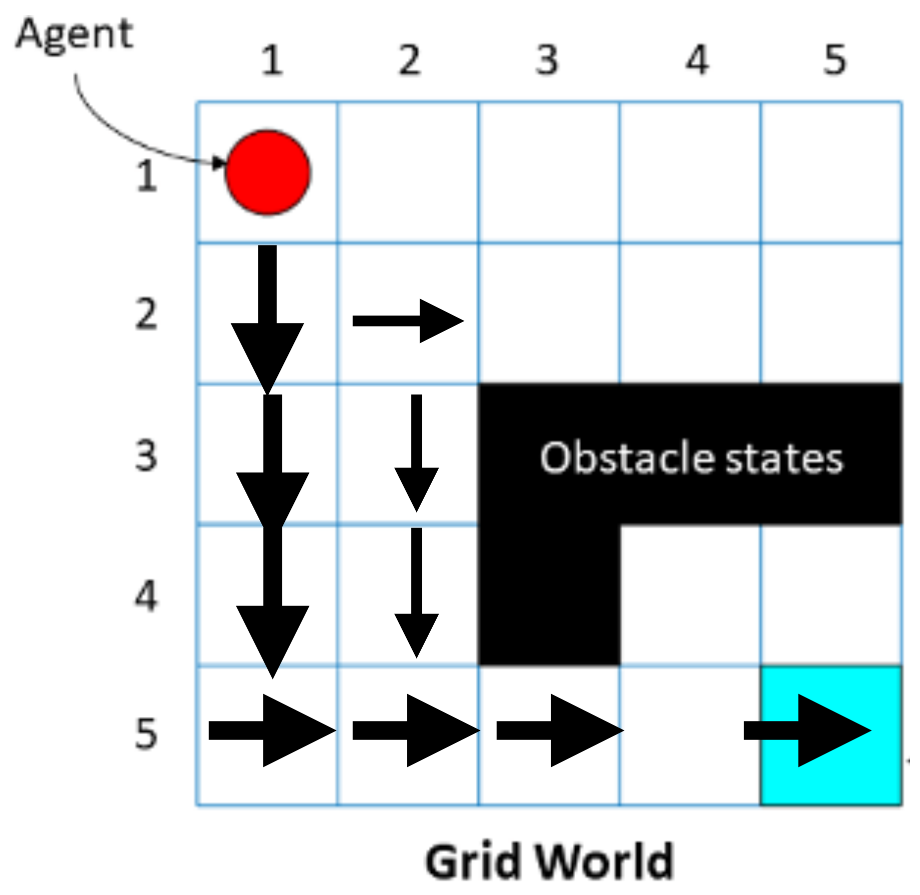


  - How good is a given state? $r_t = R(s_t)$
  - How good is a state-action pair? $r_t = R(s_t, a_t)$
  - How good is a *trajectory* $\tau = (s_0, a_0, s_1, a_1, \ldots)$:

$$R(\tau) = \sum_{t=0}^{\infty} \gamma^k r_t$$ where $\gamma \in [0,1]$ is the temporal discount



- **Learns a Policy**
  - π defines how to act, where $\pi(a \mid s)$ is the probability of selecting action a in state s
  - sample trajectories from the policy $\tau \sim \pi$



The RL Problem

The RL Problem

Select a policy π^* that maximizes expected rewards

The RL Problem

Select a policy π^* that maximizes expected rewards

For this, we need to define a **value function:**

- The expected discounted returns under a policy:

$$V_\pi(s) = \mathbb{E}_{\tau \sim \pi}[R(\tau) \mid s_0 = s]$$

The RL Problem

Select a policy π^* that maximizes expected rewards

For this, we need to define a **value function:**

- The expected discounted returns under a policy:

$$V_\pi(s) = \mathbb{E}_{\tau \sim \pi}[R(\tau) \mid s_0 = s]$$

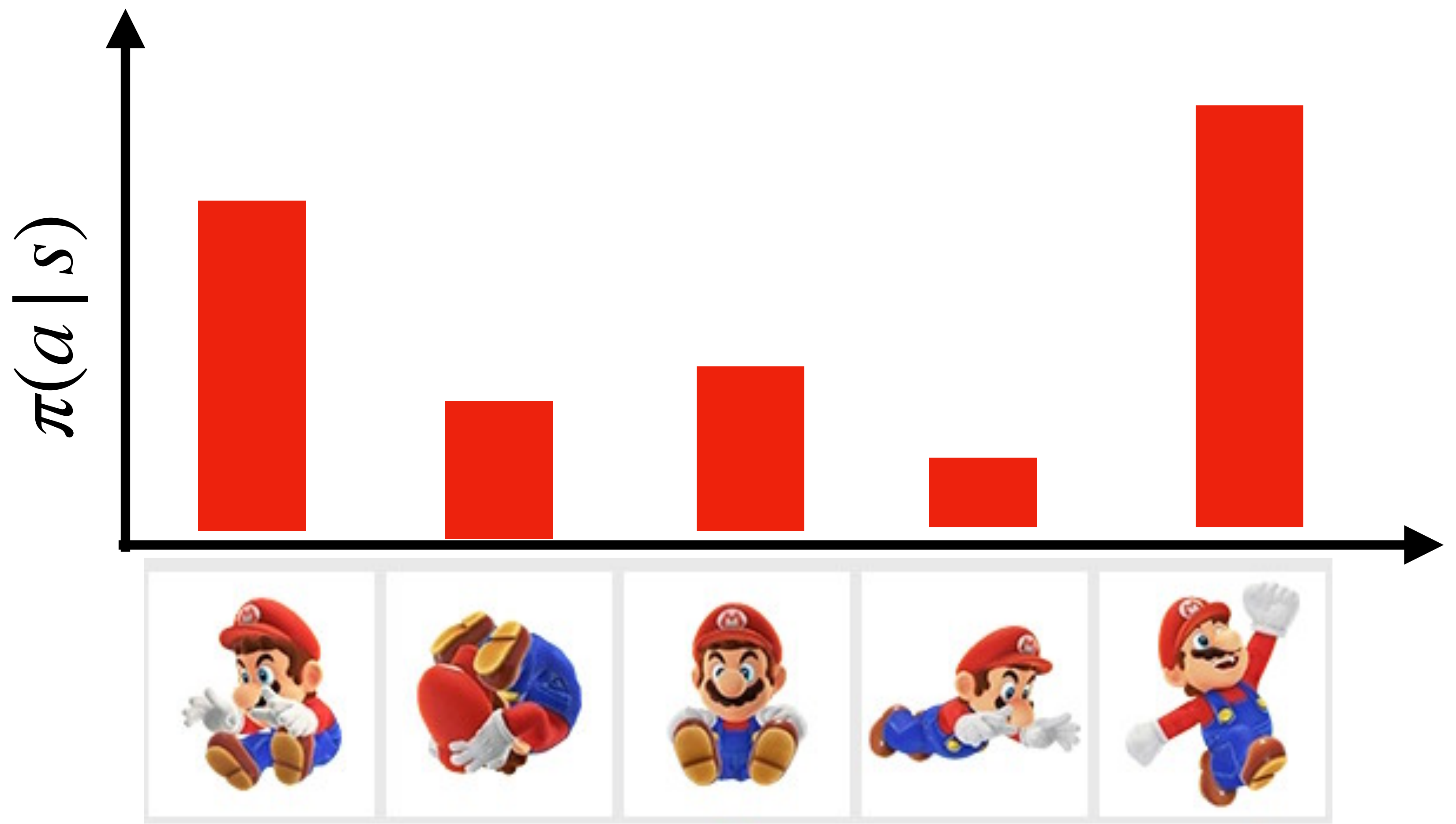
- Let's unpack that a bit:

$$V_\pi(s) = \sum_a \pi(a \mid s) \sum_{s'} P(s' \mid s, a) \left[R(s', a) + \gamma V_\pi(s') \right]$$

  - The expectation over a policy is equivalent to summing up all actions and their new state transitions, weighted by their probability

  - Then we sum up the (discounted) future expected rewards

The RL Problem



Select a policy π^* that maximizes expected rewards

For this, we need to define a **value function:**

- The expected discounted returns under a policy:

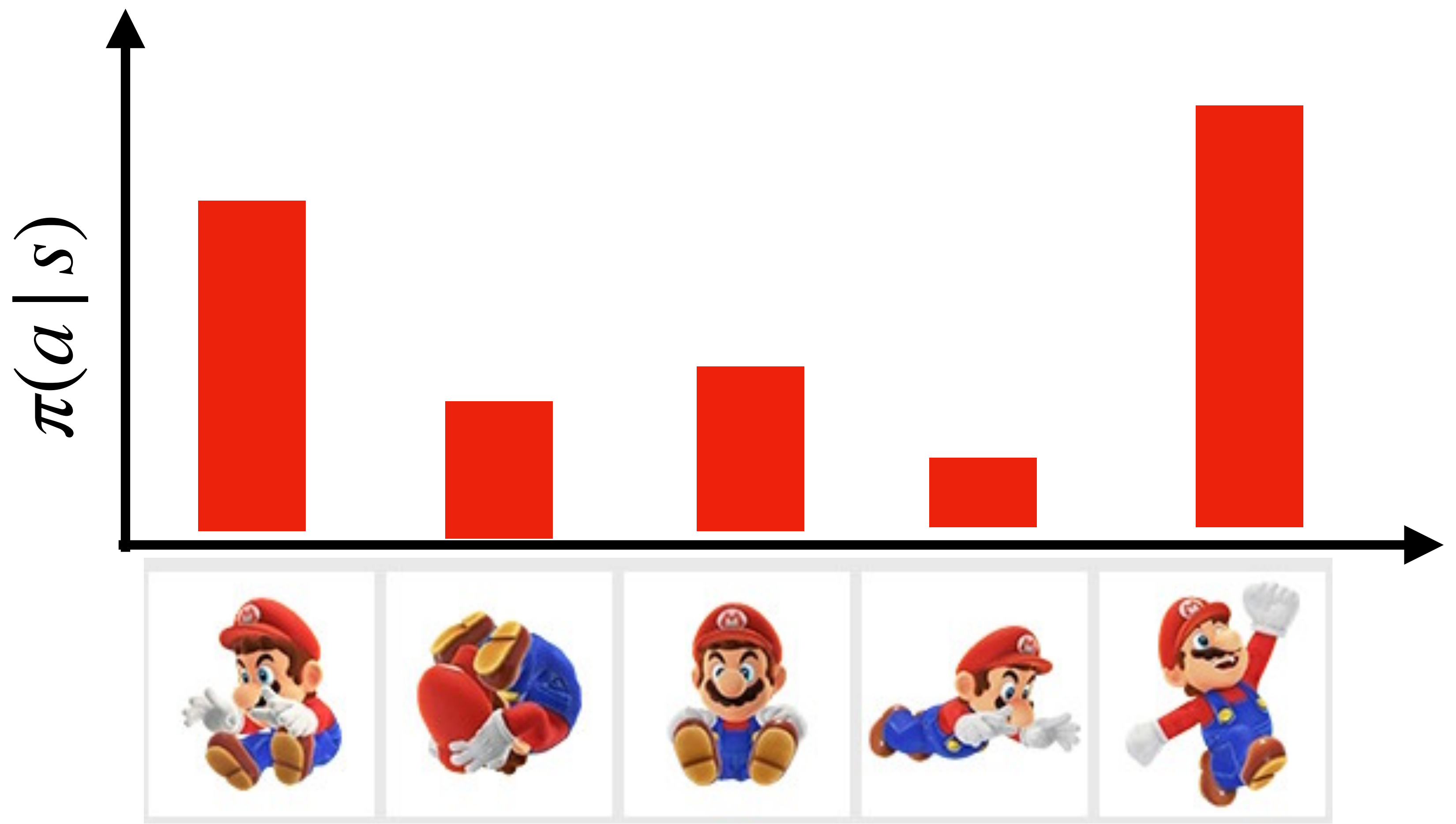
$$V_\pi(s) = \mathbb{E}_{\tau \sim \pi}[R(\tau) \mid s_0 = s]$$

- Let's unpack that a bit:

$$V_\pi(s) = \sum_a \pi(a \mid s) \sum_{s'} P(s' \mid s, a)\left[R(s', a) + \gamma V_\pi(s')\right]$$

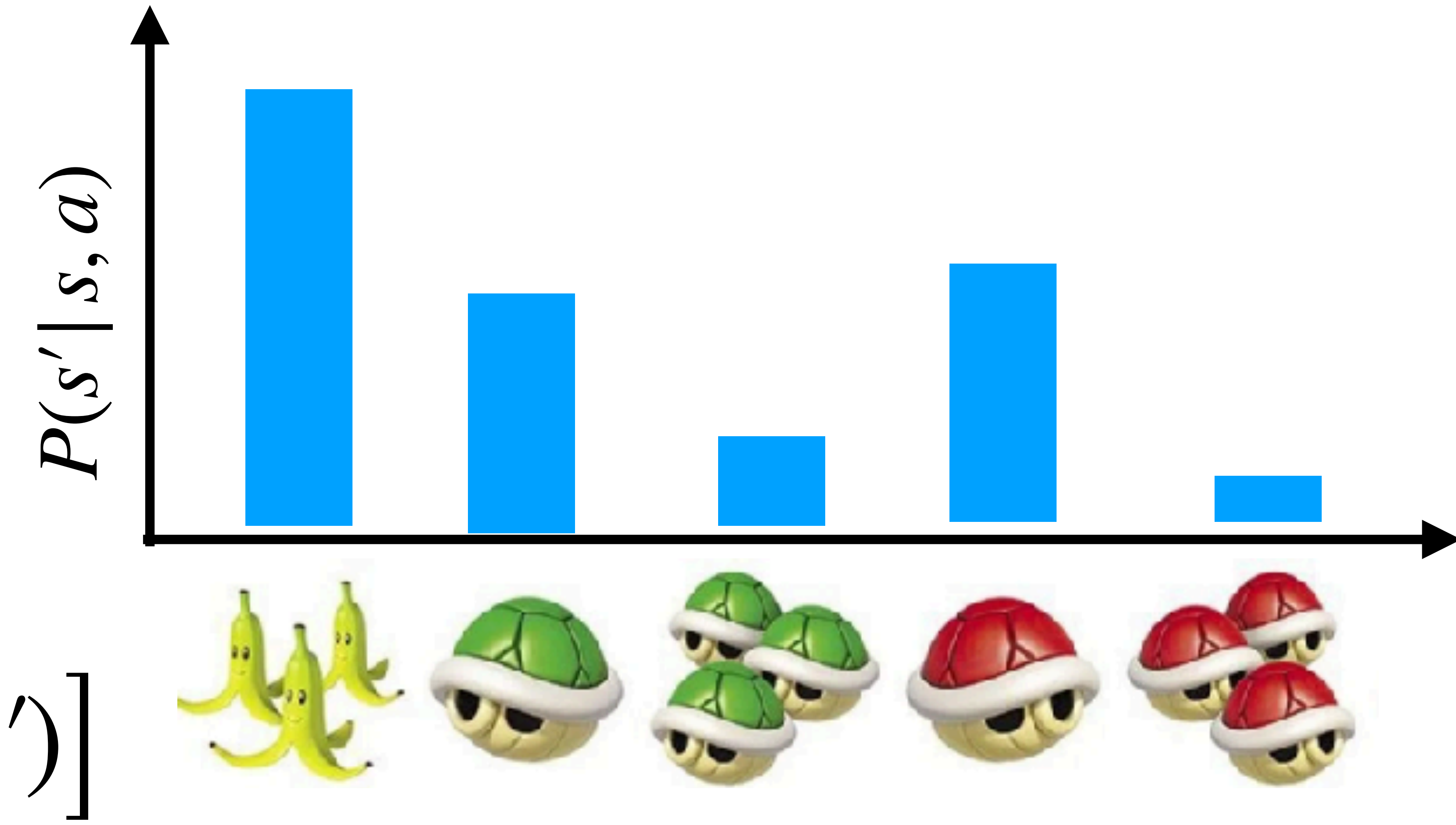
  - The expectation over a policy is equivalent to summing up all actions and their new state transitions, weighted by their probability
  - Then we sum up the (discounted) future expected rewards

The RL Problem



Select a policy π^* that maximizes expected rewards

For this, we need to define a **value function:**



- The expected discounted returns under a policy:

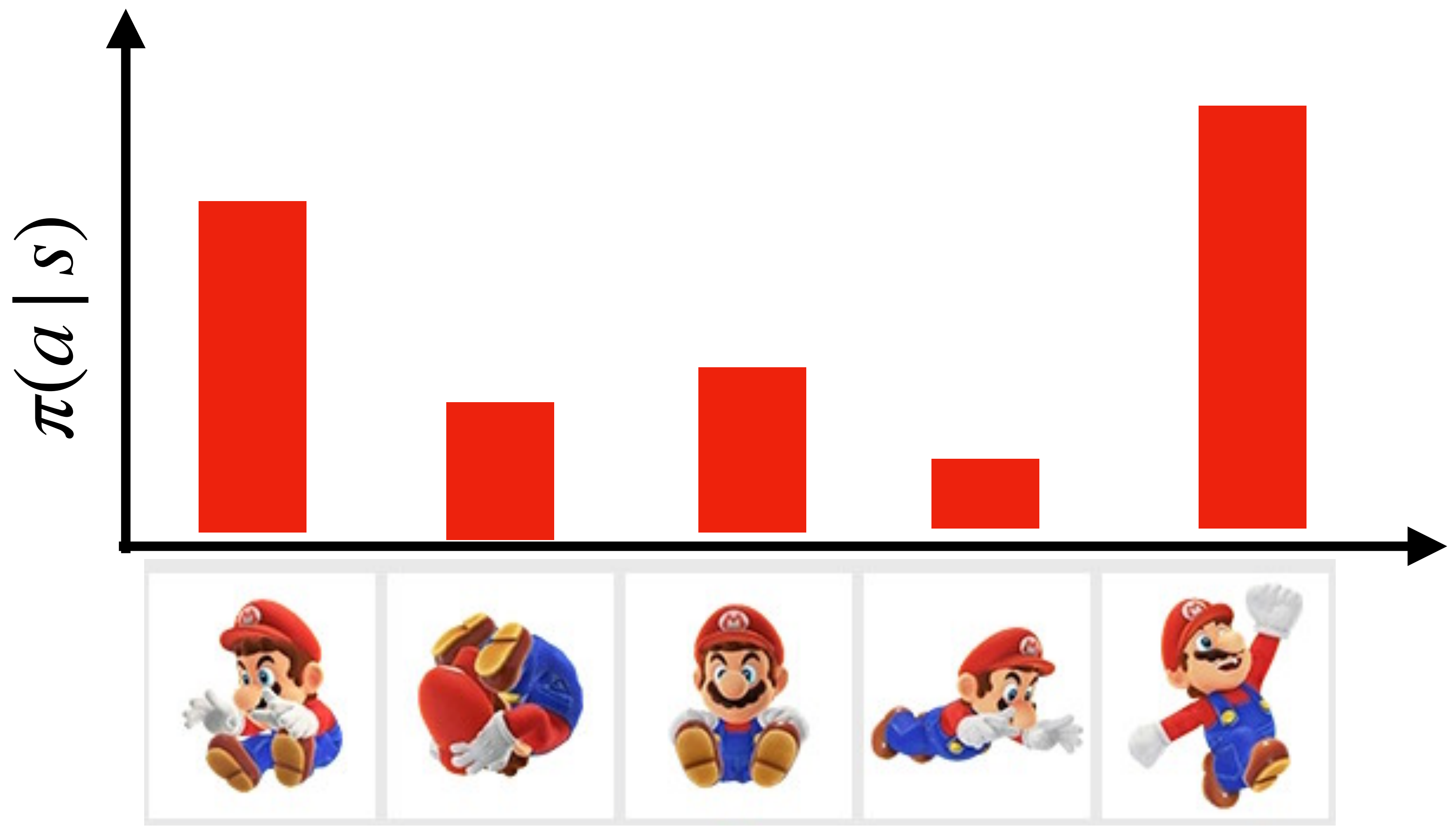
$$V_\pi(s) = \mathbb{E}_{\tau \sim \pi}[R(\tau) \mid s_0 = s]$$

- Let's unpack that a bit:

$$V_\pi(s) = \sum_a \pi(a \mid s) \sum_{s'} P(s' \mid s, a) \left[R(s', a) + \gamma V_\pi(s') \right]$$

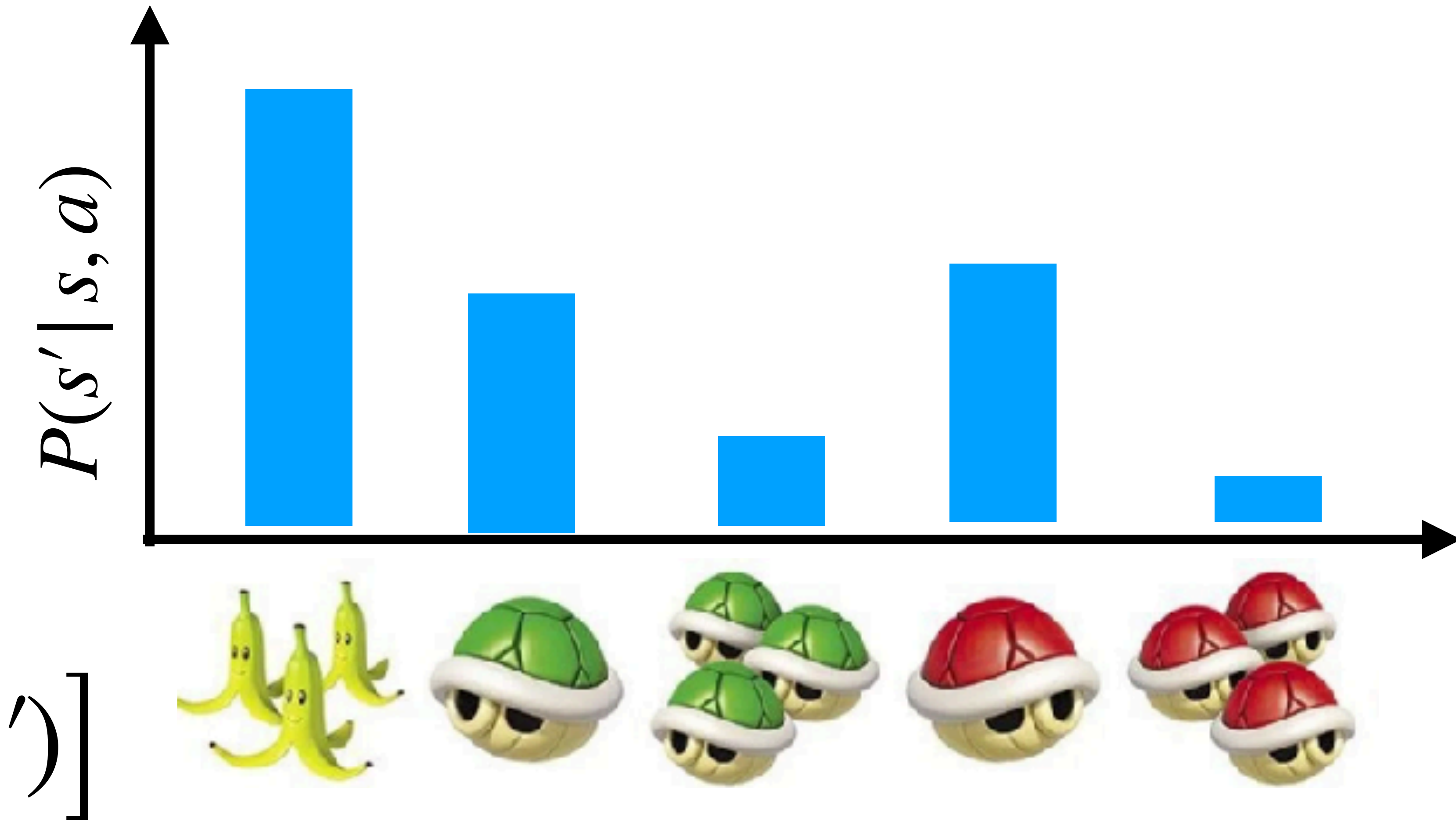
  - The expectation over a policy is equivalent to summing up all actions and their new state transitions, weighted by their probability
  - Then we sum up the (discounted) future expected rewards

The RL Problem



Select a policy π^* that maximizes expected rewards

For this, we need to define a **value function:**



- The expected discounted returns under a policy:

$$V_\pi(s) = \mathbb{E}_{\tau \sim \pi}[R(\tau) \mid s_0 = s]$$

- Let's unpack that a bit:

$$V_\pi(s) = \sum_a \pi(a \mid s) \sum_{s'} P(s' \mid s, a) \left[R(s', a) + \gamma V_\pi(s') \right]$$

  - The expectation over a policy is equivalent to summing up all actions and their new state transitions, weighted by their probability
  - Then we sum up the (discounted) future expected rewards

Finding an optimal policy via Bellman Equations

- Bellman equations are a concept from dynamic programming that provide a recursive method for optimization:

$$V_\pi(s) = \sum_a \pi(a \mid s) \sum_{s'} P(s' \mid s, a)\left[R(s', a) + \gamma V_\pi(s')\right]$$

- Theoretically, we can define an optimal value function:

$$V_*(s) = \max_a \sum_{s'} P(s' \mid s, a)\left[R(s, a) + \gamma V_*(s')\right]$$

- Optimal policy:

$$\pi_* = \arg\max_a V_*(s)$$

Finding an optimal policy via Bellman Equations

- Bellman equations are a concept from dynamic programming that provide a recursive method for optimization:

$$V_\pi(s) = \sum_a \pi(a \mid s) \sum_{s'} P(s' \mid s, a) \left[\underbrace{R(s', a) + \gamma V_\pi(s')}_{\text{recursive}} \right]$$

- Theoretically, we can define an optimal value function:

$$V_*(s) = \max_a \sum_{s'} P(s' \mid s, a) \left[R(s, a) + \gamma V_*(s') \right]$$

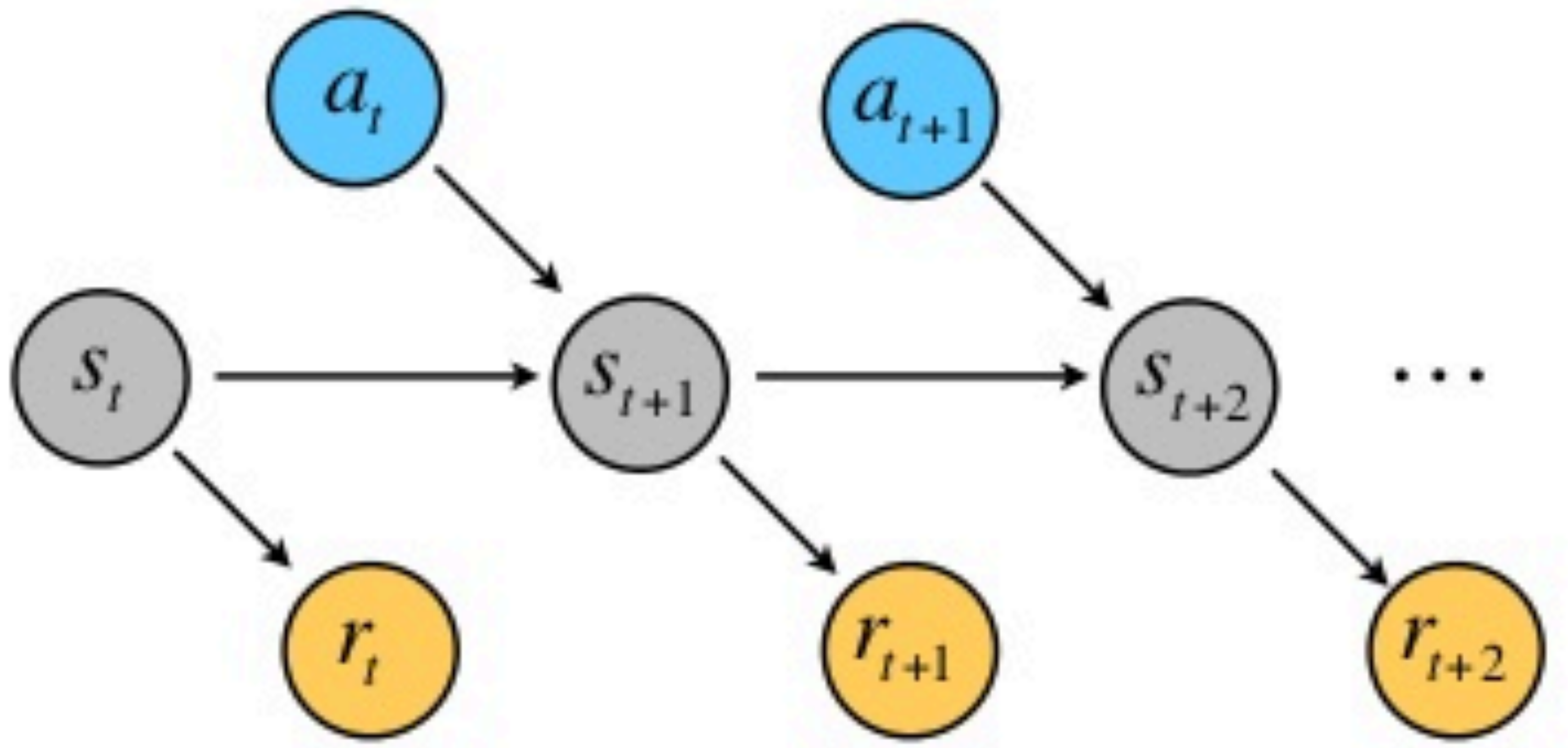
- Optimal policy:

$$\pi_* = \arg\max_a V_*(s)$$

Finding an optimal policy via Bellman Equations

- Bellman equations are a concept from dynamic programming that provide a recursive method for optimization:

$$V_\pi(s) = \sum_a \pi(a \mid s) \sum_{s'} P(s' \mid s, a)\Big[\underbrace{R(s', a) + \gamma V_\pi(s')}_{\text{recursive}}\Big]$$



- Theoretically, we can define an optimal value function:

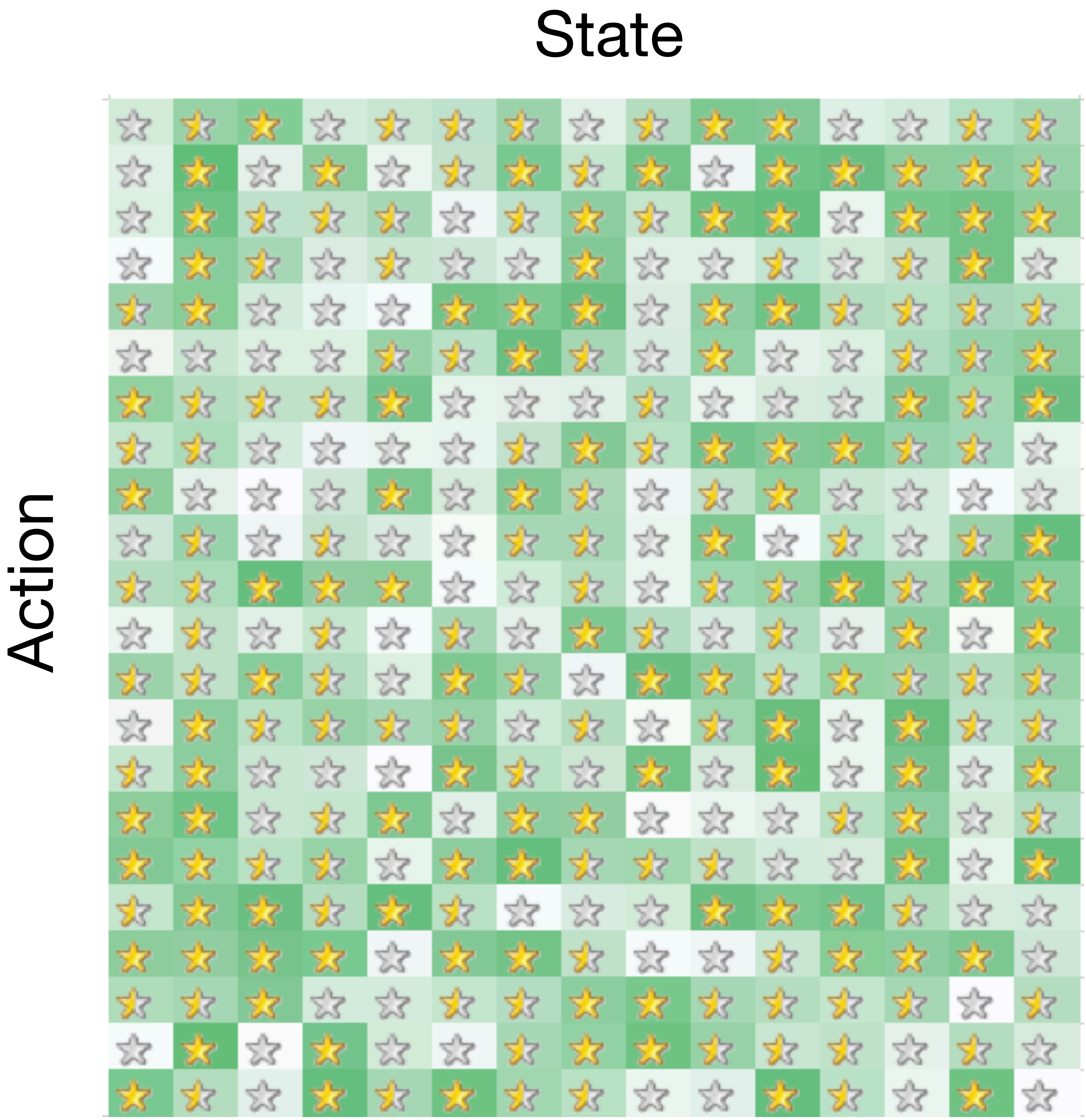
$$V_*(s) = \max_a \sum_{s'} P(s' \mid s, a)\Big[R(s, a) + \gamma V_*(s')\Big]$$

- Optimal policy:

$$\pi_* = \arg\max_a V_*(s)$$

Tabular methods

- Based on methods from Dynamic programming (Bellman, 1957), Tabular methods were first proposed as solutions for RL problems by Minsky (1961)
- Think of a giant lookup table, where we store a value representation
- Value iteration and policy iteration are examples
- Caveat: solutions require repeat visits to each state, which is infeasible in most real-world problems



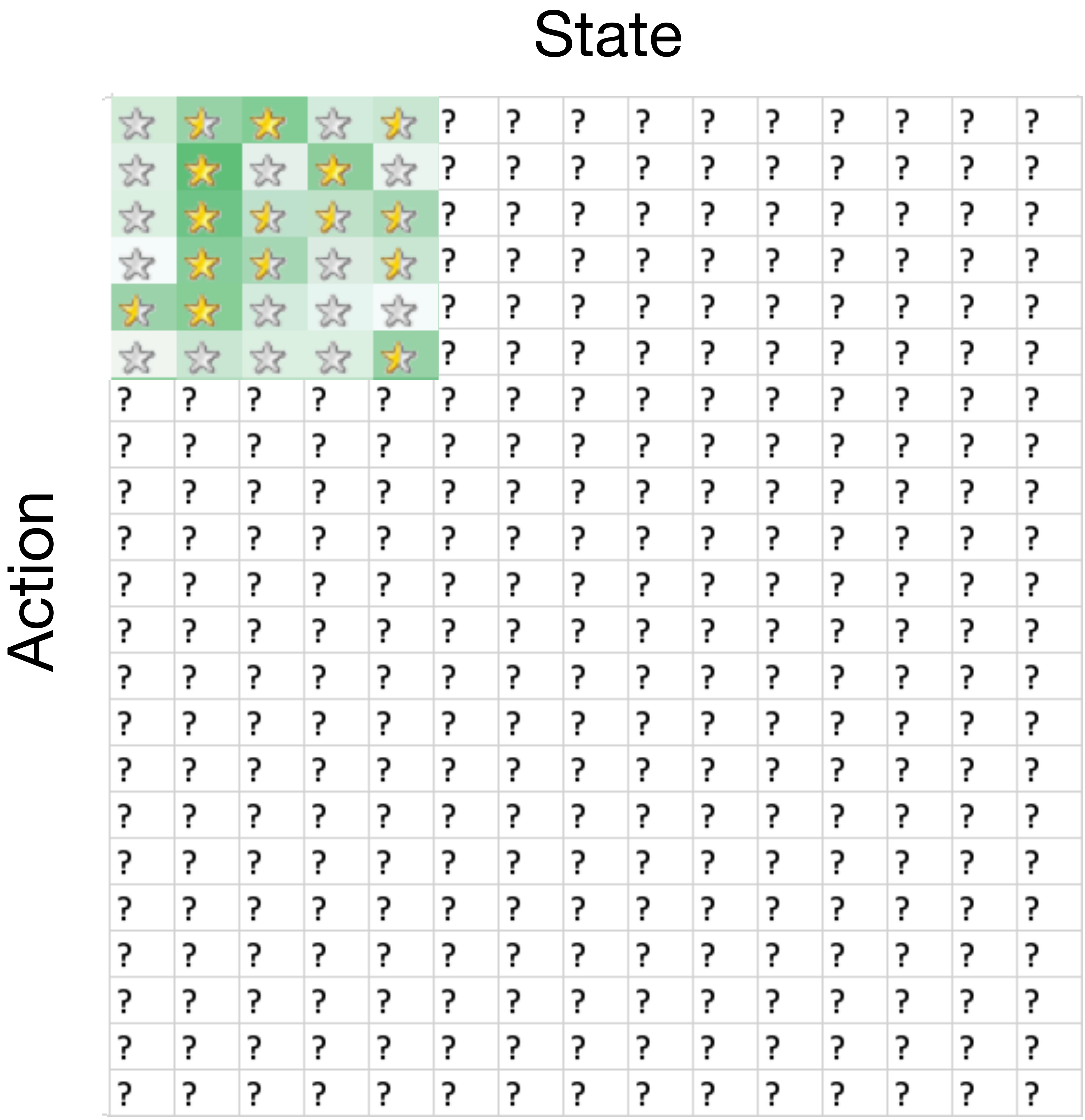
Tabular methods

- Based on methods from Dynamic programming (Bellman, 1957), Tabular methods were first proposed as solutions for RL problems by Minsky (1961)

- Think of a giant lookup table, where we store a value representation

- Value iteration and policy iteration are examples

- Caveat: solutions require repeat visits to each state, which is infeasible in most real-world problems



Value iteration

Iteratively visit all states and update the value function until a “good enough” solution has been reached.

1. Initialize the value function as $V_0(s) = 0$ for all states

2. For all s in $\mathcal{S}$:

$$V_{k+1}(s) = \max_{a \in A} \sum_{s'} P(s' \mid s, a)[R(s,a) + \gamma V_k(s')]$$

until $\max_{s \in \mathcal{S}} \mid V_k(s) - V_{k-1}(s) \mid < \theta$ Bellman residual

Value iteration

Iteratively visit all states and update the value function until a "good enough" solution has been reached.

1. Initialize the value function as $V_0(s) = 0$ for all states

2. For all s in $\mathcal{S}$:

$$V_{k+1}(s) = \max_{a \in A} \sum_{s'} P(s' | s, a)[R(s, a) + \gamma V_k(s')]$$

until $\max_{s \in \mathcal{S}} | V_k(s) - V_{k-1}(s) | < \theta$ Bellman residual

V_k converges on V_* as $k \to \infty$, and perhaps sooner, but with many costly sweeps through the state space

Policy Iteration

Alternate between evaluating a policy and then improving the policy.

Start with π_0 (typically a random policy), and then iterate for all $s \in \mathcal{S}$ in each step

- **Policy Evaluation**

$$V_{\pi_k}(s) = \mathbb{E}_{\pi_k}\left[R(s', a) + \gamma V_{\pi_k}(s')\right]$$

- **Policy Improvement**

$$\pi_{k+1} = \arg\max_a \sum_{s'} P(s' \mid s, a)\left[R(s, a) + \gamma V_{\pi_k}\right]$$

Policy Iteration

Alternate between evaluating a policy and then improving the policy.

Start with π_0 (typically a random policy), and then iterate for all $s \in \mathcal{S}$ in each step

- **Policy Evaluation**

$$V_{\pi_k}(s) = \mathbb{E}_{\pi_k}\left[R(s', a) + \gamma V_{\pi_k}(s')\right]$$

- **Policy Improvement**

$$\pi_{k+1} = \arg\max_a \sum_{s'} P(s' \mid s, a)\left[R(s, a) + \gamma V_{\pi_k}\right]$$

Policy can converge faster than value function, but still requires visiting all states $2n$ times and lacks convergence guarantees


Bill Murray
He's having the day of his life...
over and over again.
Groundhog Day

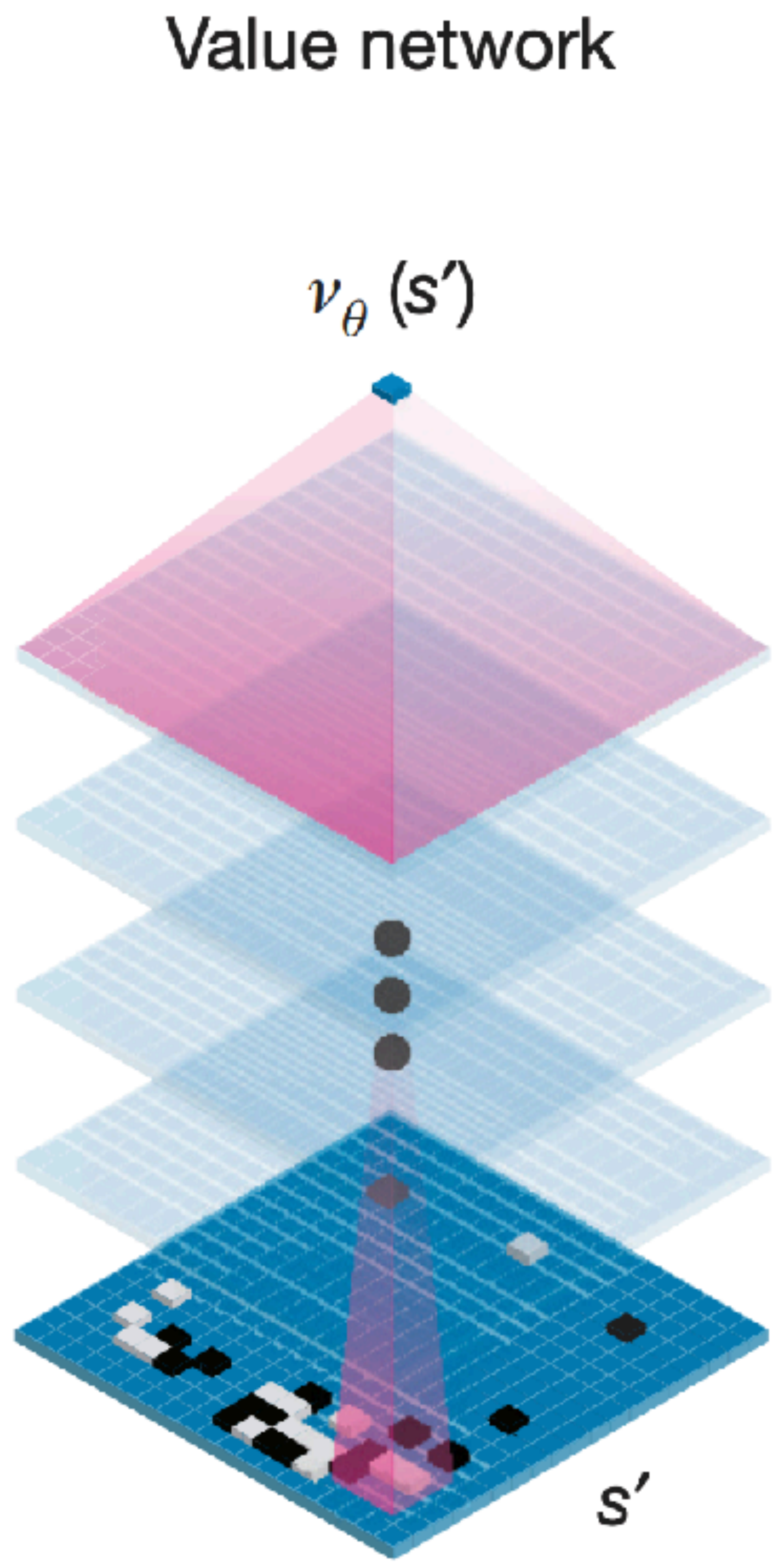
Value function approximation

Instead of learning the value for each state independently, learn a value function mapping each state to some value:

$$f : s \in \mathcal{S} \rightarrow V_\theta(s)$$

… a variety of methods are available including:

- linear function approximation (e.g,. regression)
- Neural networks
- Gaussian process regression (non-parametric)



Silver et al. (2016)

Value function approximation

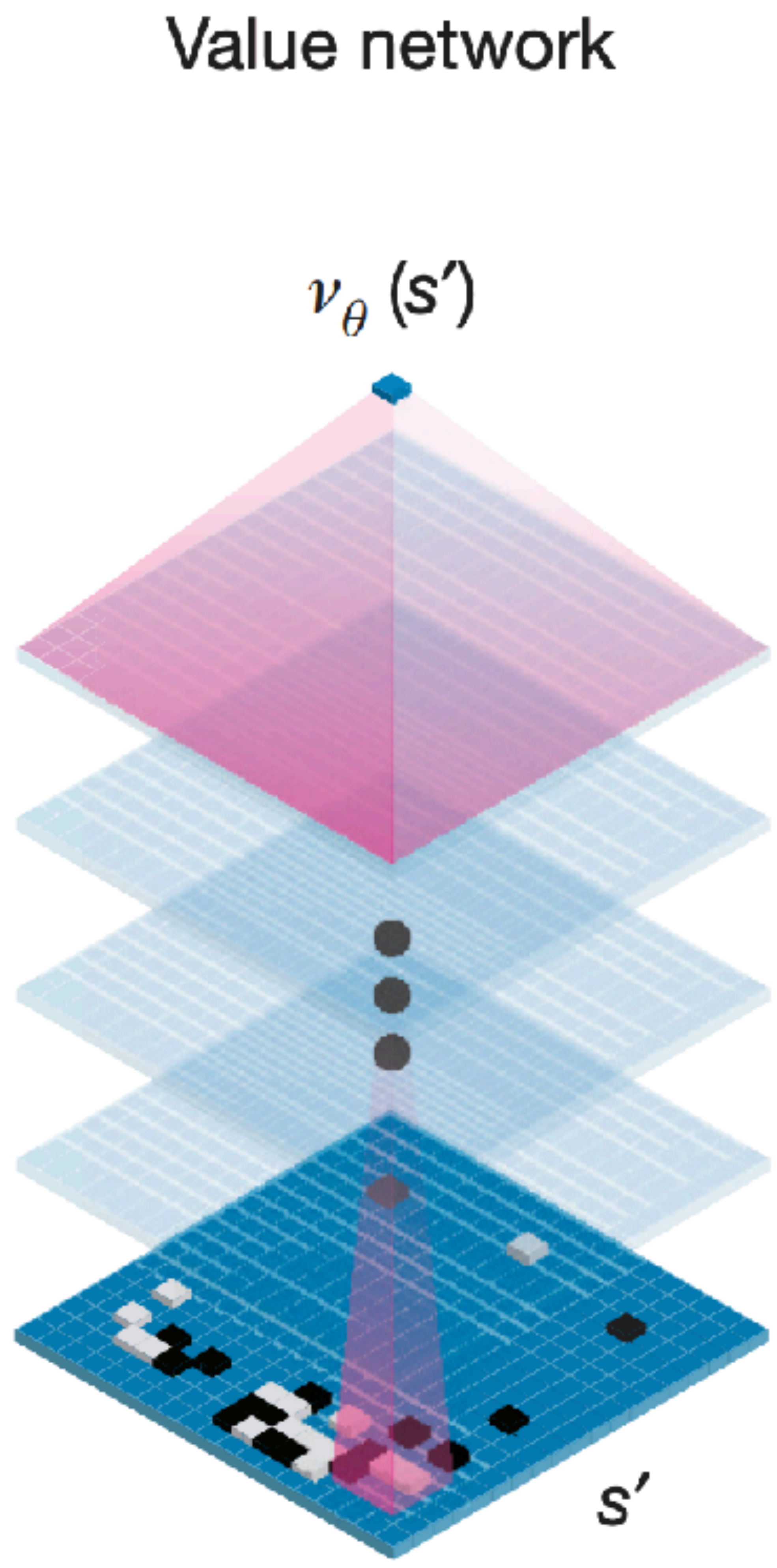
Instead of learning the value for each state independently, learn a value function mapping each state to some value:

$$f : s \in \mathcal{S} \to V_\theta(s)$$

… a variety of methods are available including:

- linear function approximation (e.g,. regression)
- Neural networks
- Gaussian process regression (non-parametric)

$$s' \longrightarrow \boxed{\text{Function Approximator}} \longrightarrow V_\theta(s')$$



Silver et al. (2016)

Value function approximation

Instead of learning the value for each state independently, learn a value function mapping each state to some value:

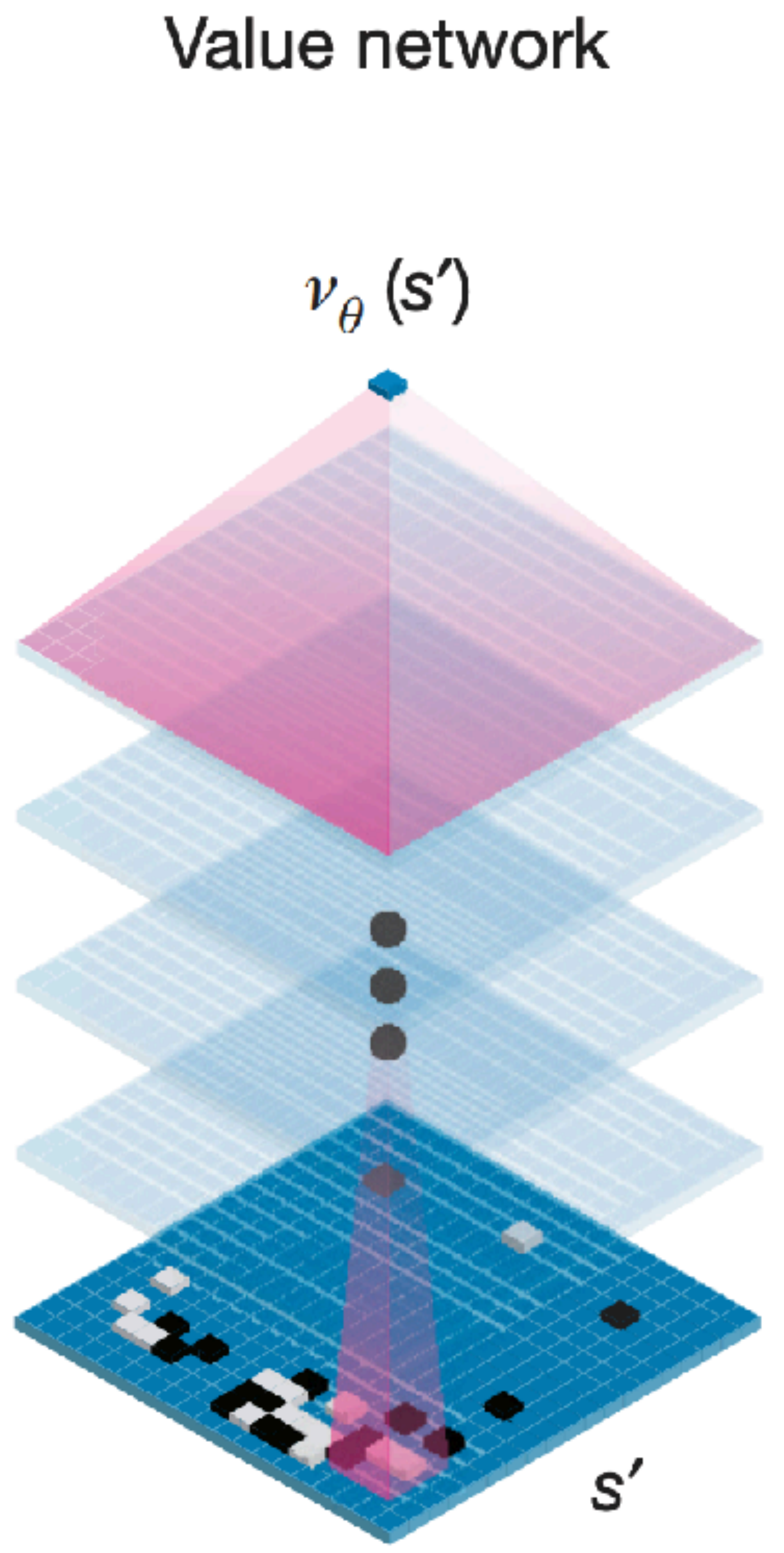
$$f : s \in \mathcal{S} \rightarrow V_\theta(s)$$

… a variety of methods are available including:

- linear function approximation (e.g,. regression)
- Neural networks
- Gaussian process regression (non-parametric)

$$\Delta\theta \propto \frac{\partial V_\theta(s')}{\partial \theta}\big(R(s') - V_\theta(s')\big)$$

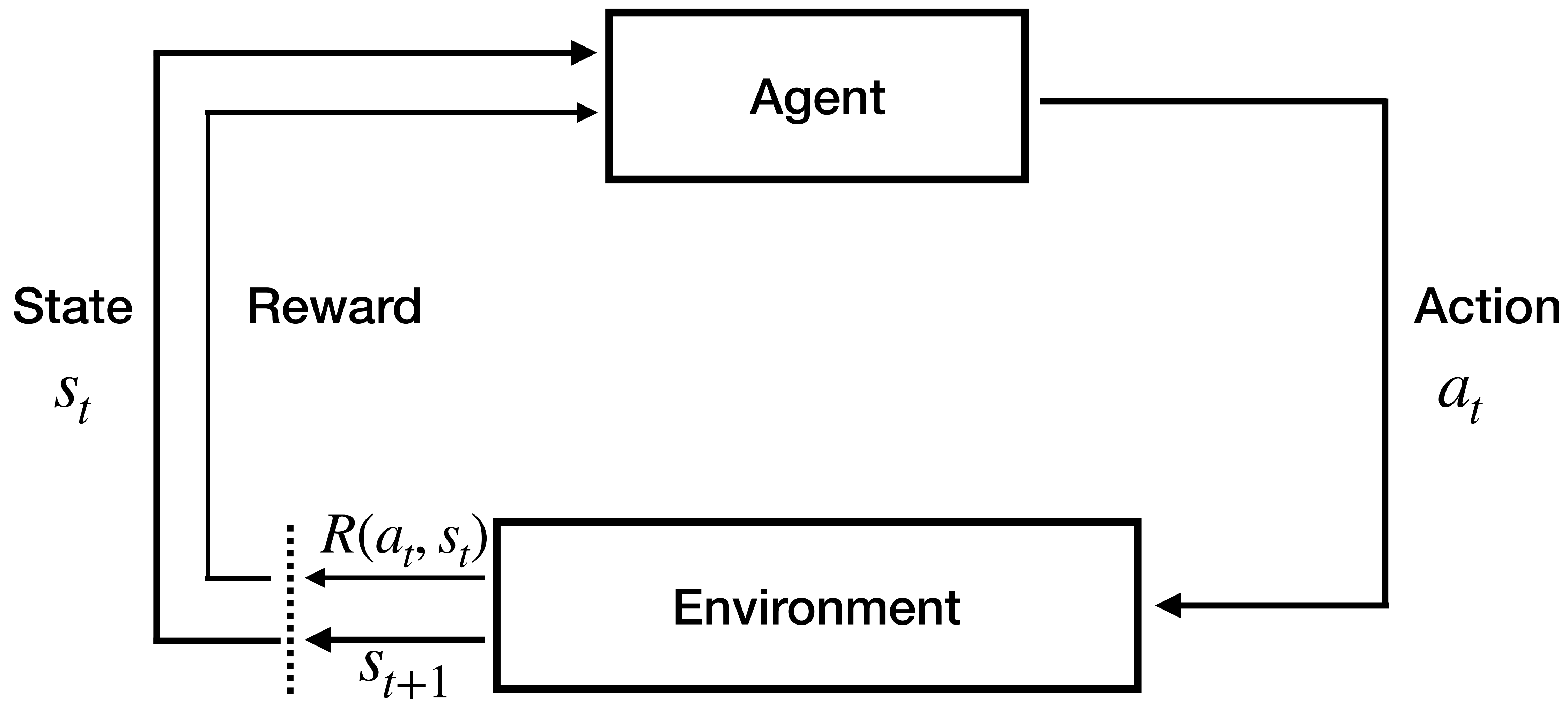
$s' \longrightarrow$ Function Approximator $\longrightarrow V_\theta(s')$

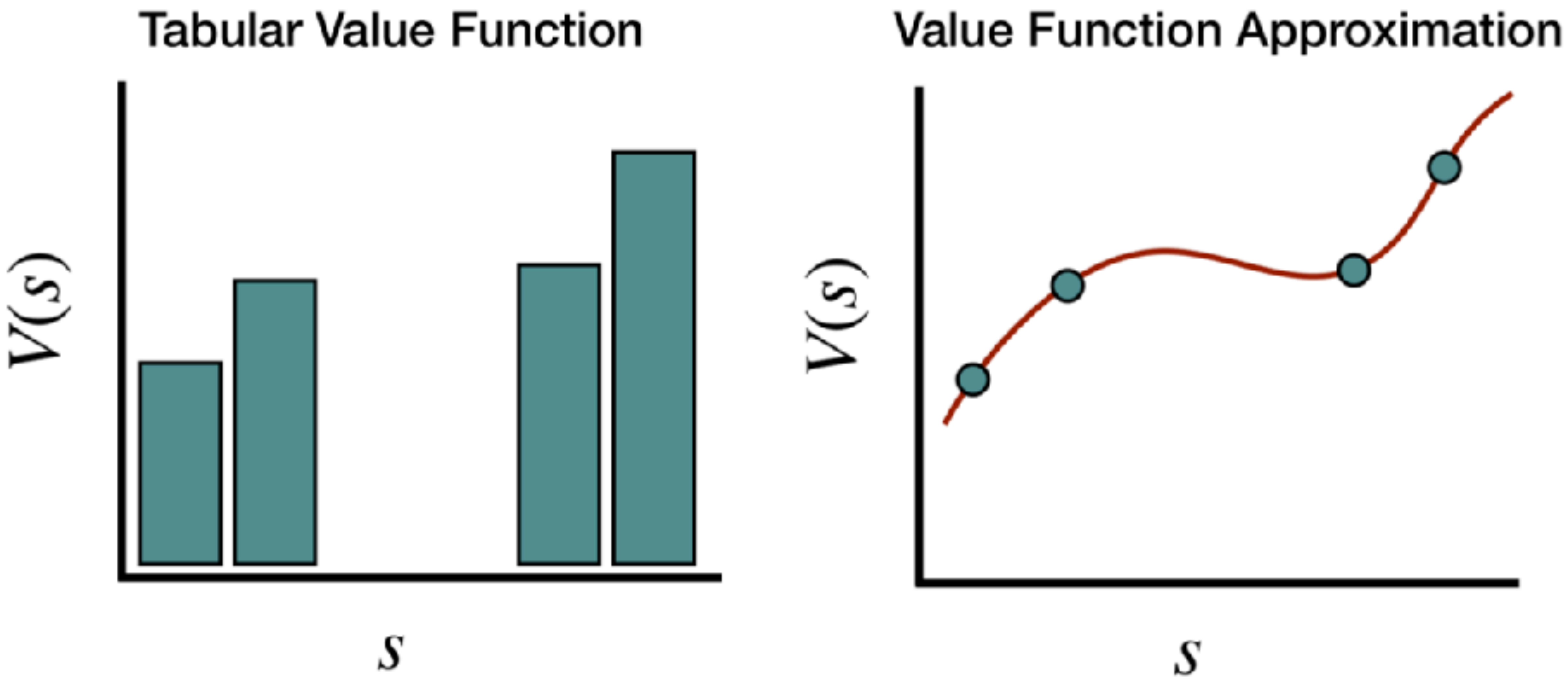


Silver et al. (2016)

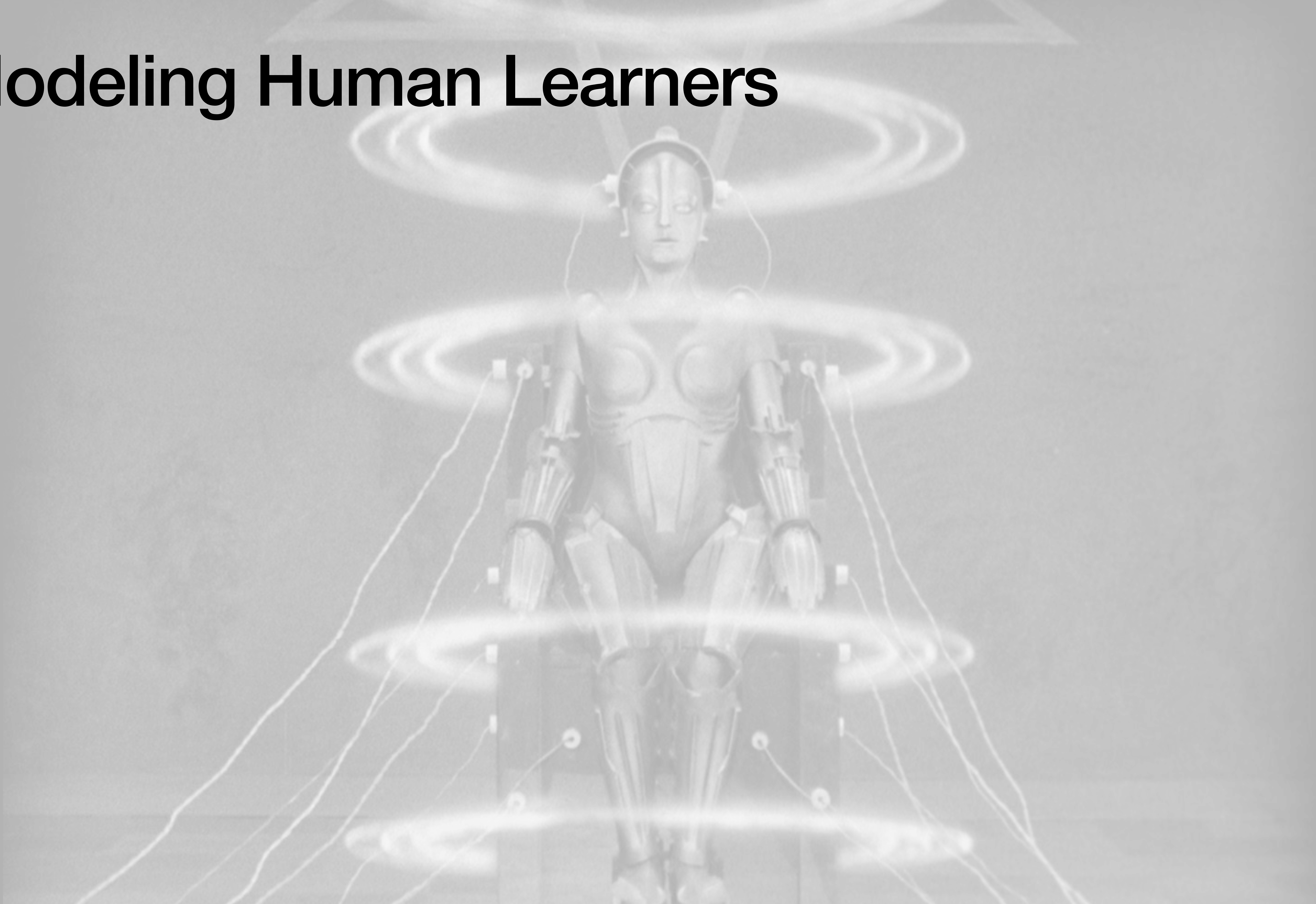
Quick recap

- RL framework defines interactions between an **agent** and the **environment**
    - The environment defines the transitions between states and provides rewards
    - The agent learns a value function and then turns this into a policy
- Traditional solutions to RL problems can be broadly classified into either **tabular methods** or **value function approximation**





Modeling Human Learners



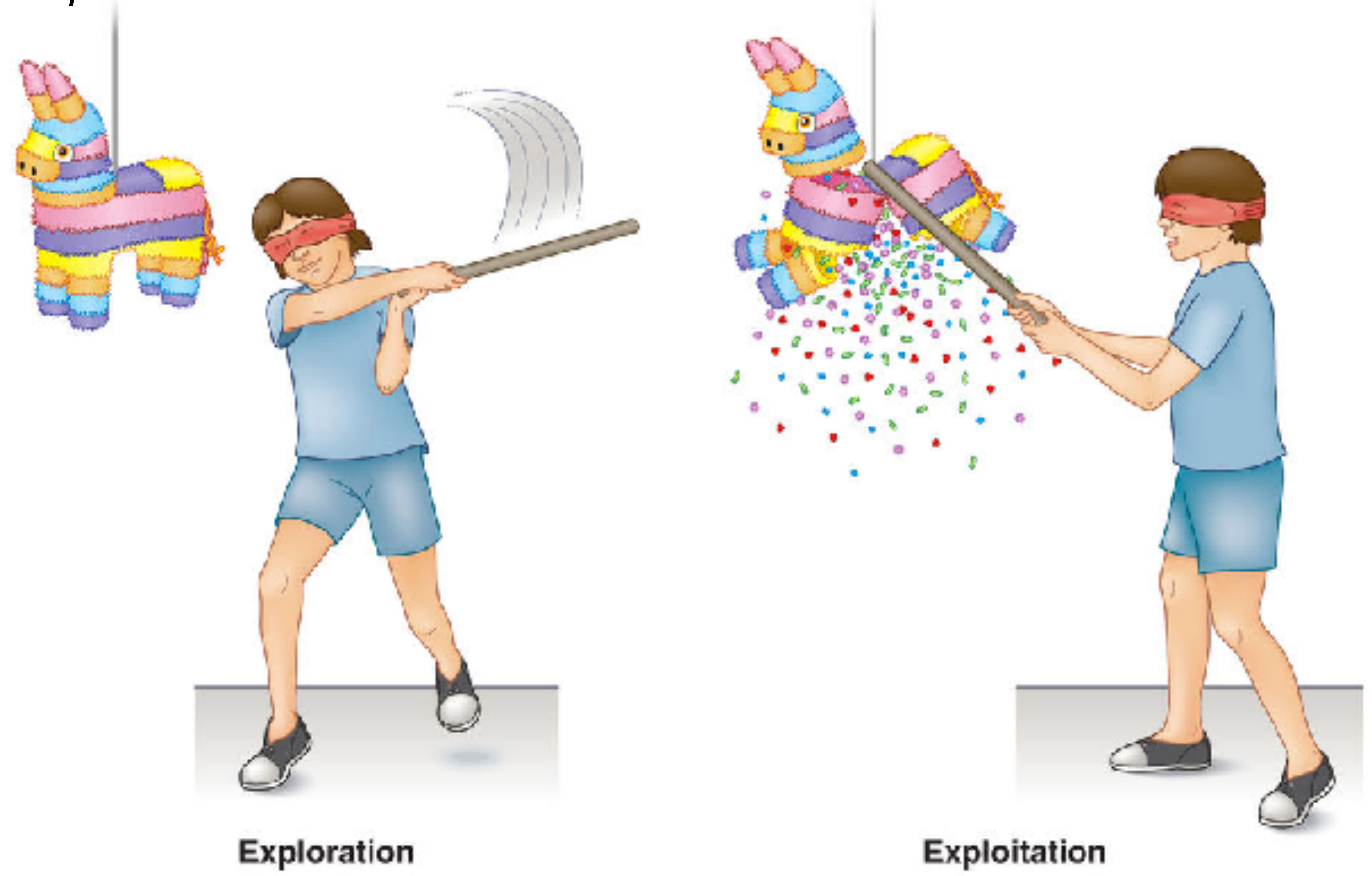
Multi-armed bandit

Multi-armed bandit

- Colorful metaphor for a row of slot machines

Multi-armed bandit

- Colorful metaphor for a row of slot machines
- One of the simplest RL problems for studying the *exploration-exploitation dilemma:*
  - **exploring** untried options to acquire information
  - **exploiting** known options to acquire immediate rewards



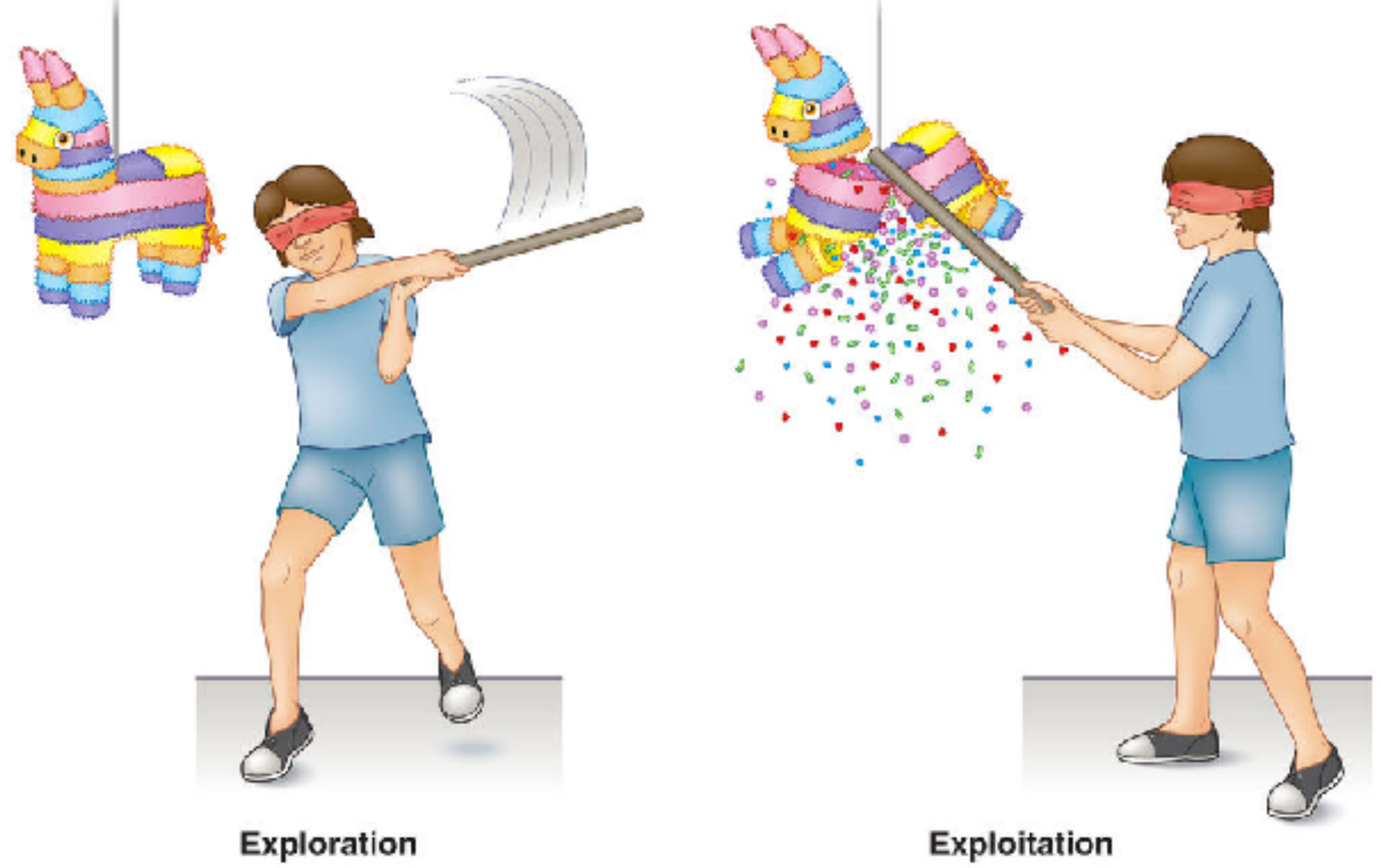
Multi-armed bandit

- Colorful metaphor for a row of slot machines
- One of the simplest RL problems for studying the *exploration-exploitation dilemma:*
  - **exploring** untried options to acquire information
  - **exploiting** known options to acquire immediate rewards

- We can think of it as an MDP with a single state:
  - No need to discount future states
  - Instead of state-value function $V(s)$ we can represent the value of an action $V(a)$ or more properly, using a action-value function $Q(s, a)$

Multi-armed bandit

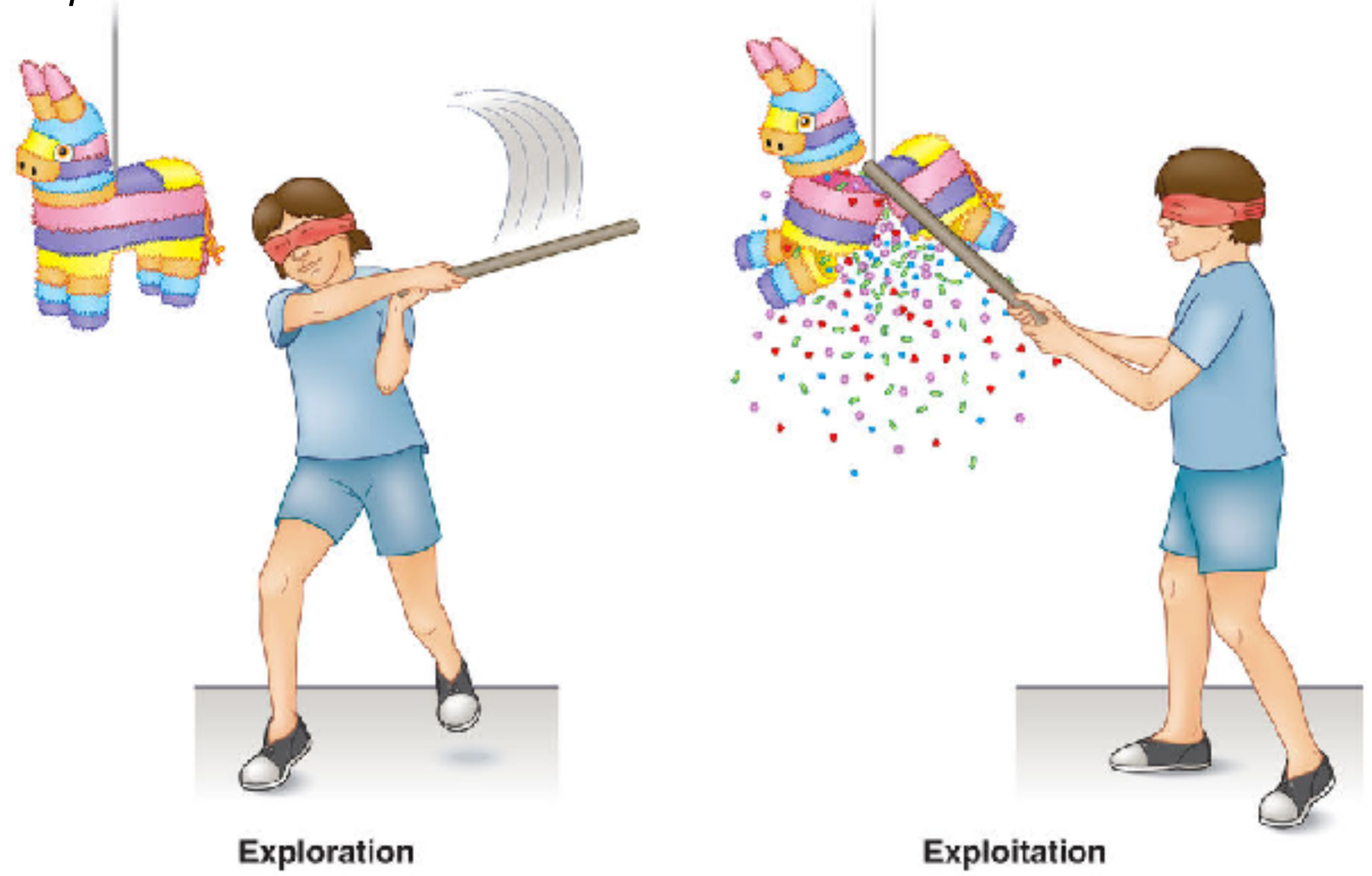
- Colorful metaphor for a row of slot machines
- One of the simplest RL problems for studying the *exploration-exploitation dilemma:*
  - **exploring** untried options to acquire information
  - **exploiting** known options to acquire immediate rewards



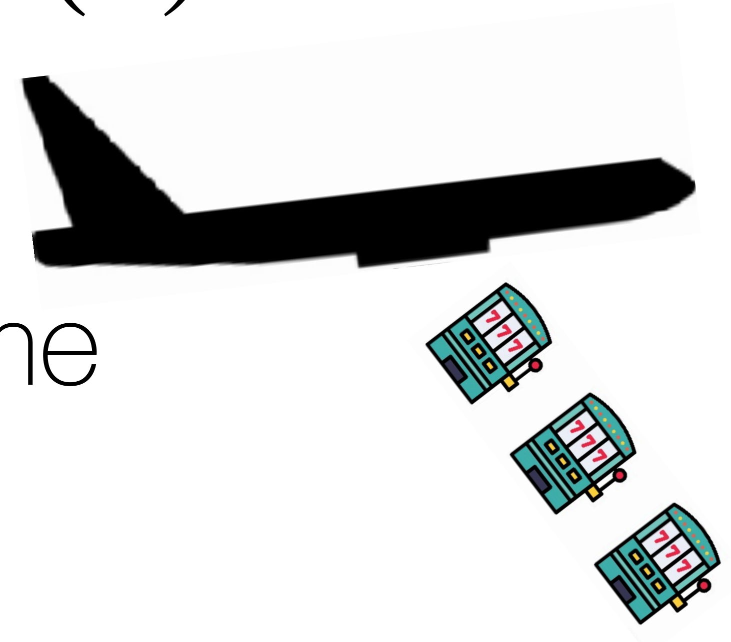
- We can think of it as an MDP with a single state:
  - No need to discount future states
  - Instead of state-value function $V(s)$ we can represent the value of an action $V(a)$ or more properly, using a action-value function $Q(s, a)$
- Despite it's simplicity, optimal solutions are generally only available with infinite time

Multi-armed bandit

- Colorful metaphor for a row of slot machines
- One of the simplest RL problems for studying the *exploration-exploitation dilemma:*
  - **exploring** untried options to acquire information
  - **exploiting** known options to acquire immediate rewards



- We can think of it as an MDP with a single state:
  - No need to discount future states
  - Instead of state-value function $V(s)$ we can represent the value of an action $V(a)$ or more properly, using a action-value function $Q(s, a)$
- Despite it's simplicity, optimal solutions are generally only available with infinite time



Brief aside: State-Value function vs. Action-value function

So far, I've focused on describing state-value functions:

$$V_\pi(s) = \mathbb{E}_{\tau \sim \pi}[R(\tau) \,|\, s_0 = s]$$

However, you can also describe a RL model using the action-value function:

$$Q_\pi(s, a) = \mathbb{E}_{\tau \sim \pi}[R(\tau) \,|\, s_0 = s, a_0 = a]$$

Both are equivalent under:

$$V_\pi(s) = \sum_{a \in A} \pi(a \,|\, s) * Q_\pi(s, a)$$

Rescorla-Wagner and Q-learning: the Delta Rule

Rescorla-Wagner and Q-learning: the Delta Rule

Rescorla-Wagner (1972) model

- Learning as an active process of making predictions about the world
- Predict the value of stimulus $\mathbf{x}_t$ with a linear combination of weights $\mathbf{w}_t$:

$$V(\mathbf{x}_t) = \mathbf{w}_t^\top \mathbf{x}_t$$

- Weights are updated based on prediction error (aka Delta rule)

$$\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t + \eta \left[r_t - V(\mathbf{x}_t) \right]$$

Rescorla-Wagner and Q-learning: the Delta Rule

Rescorla-Wagner (1972) model

- Learning as an active process of making predictions about the world
- Predict the value of stimulus $\mathbf{x}_t$ with a linear combination of weights $\mathbf{w}_t$:

$$V(\mathbf{x}_t) = \mathbf{w}_t^\top \mathbf{x}_t$$

- Weights are updated based on prediction error (aka Delta rule)

$$\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t + \eta \underbrace{\left[r_t - V(\mathbf{x}_t)\right]}_{\delta_t}$$

Rescorla-Wagner and Q-learning: the Delta Rule

Rescorla-Wagner (1972) model

- Learning as an active process of making predictions about the world
- Predict the value of stimulus $\mathbf{x}_t$ with a linear combination of weights $\mathbf{w}_t$:

$$V(\mathbf{x}_t) = \mathbf{w}_t^\top \mathbf{x}_t$$

- Weights are updated based on prediction error (aka Delta rule)

$$\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t + \eta \underbrace{\left[r_t - V(\mathbf{x}_t)\right]}_{\delta_t}$$

Q-Learning (Watkins, 1989) uses the same update rule

$$Q(s_t, a_t)_{new} \leftarrow Q(s_t, a_t)_{old} + \eta \left[R(s, a) - Q(s, a)_{old}\right]$$

Rescorla-Wagner and Q-learning: the Delta Rule

Rescorla-Wagner (1972) model

- Learning as an active process of making predictions about the world
- Predict the value of stimulus $\mathbf{x}_t$ with a linear combination of weights $\mathbf{w}_t$:

$$V(\mathbf{x}_t) = \mathbf{w}_t^\top \mathbf{x}_t$$

- Weights are updated based on prediction error (aka Delta rule)

$$\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t + \eta \underbrace{\left[r_t - V(\mathbf{x}_t)\right]}_{\delta_t}$$

Q-Learning (Watkins, 1989) uses the same update rule

			
Q	0	0	0

$$Q(s_t, a_t)_{new} \leftarrow Q(s_t, a_t)_{old} + \eta \left[R(s, a) - Q(s, a)_{old}\right]$$

Rescorla-Wagner and Q-learning: the Delta Rule

Rescorla-Wagner (1972) model

- Learning as an active process of making predictions about the world
- Predict the value of stimulus $\mathbf{x}_t$ with a linear combination of weights $\mathbf{w}_t$:

$$V(\mathbf{x}_t) = \mathbf{w}_t^\top \mathbf{x}_t$$

- Weights are updated based on prediction error (aka Delta rule)

$$\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t + \eta \underbrace{\left[r_t - V(\mathbf{x}_t)\right]}_{\delta_t}$$

Q-Learning (Watkins, 1989) uses the same update rule

			
Q	0	0	0

$$Q(s_t, a_t)_{new} \leftarrow Q(s_t, a_t)_{old} + \eta \left[R(s, a) - Q(s, a)_{old}\right]$$

0 0

Rescorla-Wagner and Q-learning: the Delta Rule

Rescorla-Wagner (1972) model

- Learning as an active process of making predictions about the world
- Predict the value of stimulus $\mathbf{x}_t$ with a linear combination of weights $\mathbf{w}_t$:

$$V(\mathbf{x}_t) = \mathbf{w}_t^\top \mathbf{x}_t$$

- Weights are updated based on prediction error (aka Delta rule)

$$\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t + \eta \underbrace{\left[r_t - V(\mathbf{x}_t) \right]}_{\delta_t}$$

Q-Learning (Watkins, 1989) uses the same update rule

			
Q	0	0	0

→ 10

$$Q(s_t, a_t)_{new} \leftarrow \underset{0}{Q(s_t, a_t)_{old}} + \eta \left[R(s, a) - \underset{0}{Q(s, a)_{old}} \right]$$

Rescorla-Wagner and Q-learning: the Delta Rule

Rescorla-Wagner (1972) model

- Learning as an active process of making predictions about the world
- Predict the value of stimulus $\mathbf{x}_t$ with a linear combination of weights $\mathbf{w}_t$:

$$V(\mathbf{x}_t) = \mathbf{w}_t^\top \mathbf{x}_t$$

- Weights are updated based on prediction error (aka Delta rule)

$$\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t + \eta \underbrace{\left[r_t - V(\mathbf{x}_t)\right]}_{\delta_t}$$

Q-Learning (Watkins, 1989) uses the same update rule

			
Q	0	0	0

→ 10

$$Q(s_t, a_t)_{new} \leftarrow Q(s_t, a_t)_{old} + \eta \left[R(s,a) - Q(s,a)_{old}\right]$$

0 10 0

Rescorla-Wagner and Q-learning: the Delta Rule

Rescorla-Wagner (1972) model

- Learning as an active process of making predictions about the world
- Predict the value of stimulus $\mathbf{x}_t$ with a linear combination of weights $\mathbf{w}_t$:

$$V(\mathbf{x}_t) = \mathbf{w}_t^\top \mathbf{x}_t$$

- Weights are updated based on prediction error (aka Delta rule)

$$\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t + \eta \underbrace{\left[r_t - V(\mathbf{x}_t)\right]}_{\delta_t}$$

Q-Learning (Watkins, 1989) uses the same update rule

			
Q	0	0	0

→ 10

$$Q(s_t, a_t)_{new} \leftarrow Q(s_t, a_t)_{old} + \eta \left[R(s, a) - Q(s, a)_{old}\right]$$

0 1 10 0

Rescorla-Wagner and Q-learning: the Delta Rule

Rescorla-Wagner (1972) model

- Learning as an active process of making predictions about the world
- Predict the value of stimulus $\mathbf{x}_t$ with a linear combination of weights $\mathbf{w}_t$:

$$V(\mathbf{x}_t) = \mathbf{w}_t^\top \mathbf{x}_t$$

- Weights are updated based on prediction error (aka Delta rule)

$$\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t + \eta \left[\underbrace{r_t - V(\mathbf{x}_t)}_{\delta_t} \right]$$

Q-Learning (Watkins, 1989) uses the same update rule

			
Q	0	0	10

→ 10

$$Q(s_t, a_t)_{new} \leftarrow Q(s_t, a_t)_{old} + \eta \left[R(s,a) - Q(s,a)_{old} \right]$$

10 0 1 10 0

Rescorla-Wagner and Q-learning: the Delta Rule

Rescorla-Wagner (1972) model

- Learning as an active process of making predictions about the world
- Predict the value of stimulus $\mathbf{x}_t$ with a linear combination of weights $\mathbf{w}_t$:

$$V(\mathbf{x}_t) = \mathbf{w}_t^\top \mathbf{x}_t$$

- Weights are updated based on prediction error (aka Delta rule)

$$\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t + \eta \underbrace{\left[r_t - V(\mathbf{x}_t)\right]}_{\delta_t}$$

Q-Learning (Watkins, 1989) uses the same update rule

			
Q	0	0	10

→ 10

$$Q(s_t, a_t)_{new} \leftarrow Q(s_t, a_t)_{old} + \eta \left[R(s,a) - Q(s,a)_{old}\right]$$

10 1 10

Rescorla-Wagner and Q-learning: the Delta Rule

Rescorla-Wagner (1972) model

- Learning as an active process of making predictions about the world
- Predict the value of stimulus $\mathbf{x}_t$ with a linear combination of weights $\mathbf{w}_t$:

$$V(\mathbf{x}_t) = \mathbf{w}_t^\top \mathbf{x}_t$$

- Weights are updated based on prediction error (aka Delta rule)

$$\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t + \eta \underbrace{\left[r_t - V(\mathbf{x}_t)\right]}_{\delta_t}$$

Q-Learning (Watkins, 1989) uses the same update rule

			
Q	0	0	10

→ 7

$$Q(s_t, a_t)_{new} \leftarrow Q(s_t, a_t)_{old} + \eta \left[R(s, a) - Q(s, a)_{old}\right]$$

10 1 7 10

Rescorla-Wagner and Q-learning: the Delta Rule

Rescorla-Wagner (1972) model

- Learning as an active process of making predictions about the world
- Predict the value of stimulus $\mathbf{x}_t$ with a linear combination of weights $\mathbf{w}_t$:

$$V(\mathbf{x}_t) = \mathbf{w}_t^\top \mathbf{x}_t$$

- Weights are updated based on prediction error (aka Delta rule)

$$\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t + \eta \underbrace{\left[r_t - V(\mathbf{x}_t)\right]}_{\delta_t}$$

Q-Learning (Watkins, 1989) uses the same update rule

			
Q	0	0	7

→ 7

$$Q(s_t, a_t)_{new} \leftarrow Q(s_t, a_t)_{old} + \eta \left[R(s, a) - Q(s, a)_{old}\right]$$

7 10 1 7 10

Temporal Difference Learning

For simplicity, I'm omitting the temporal difference (TD) error, which looks like:

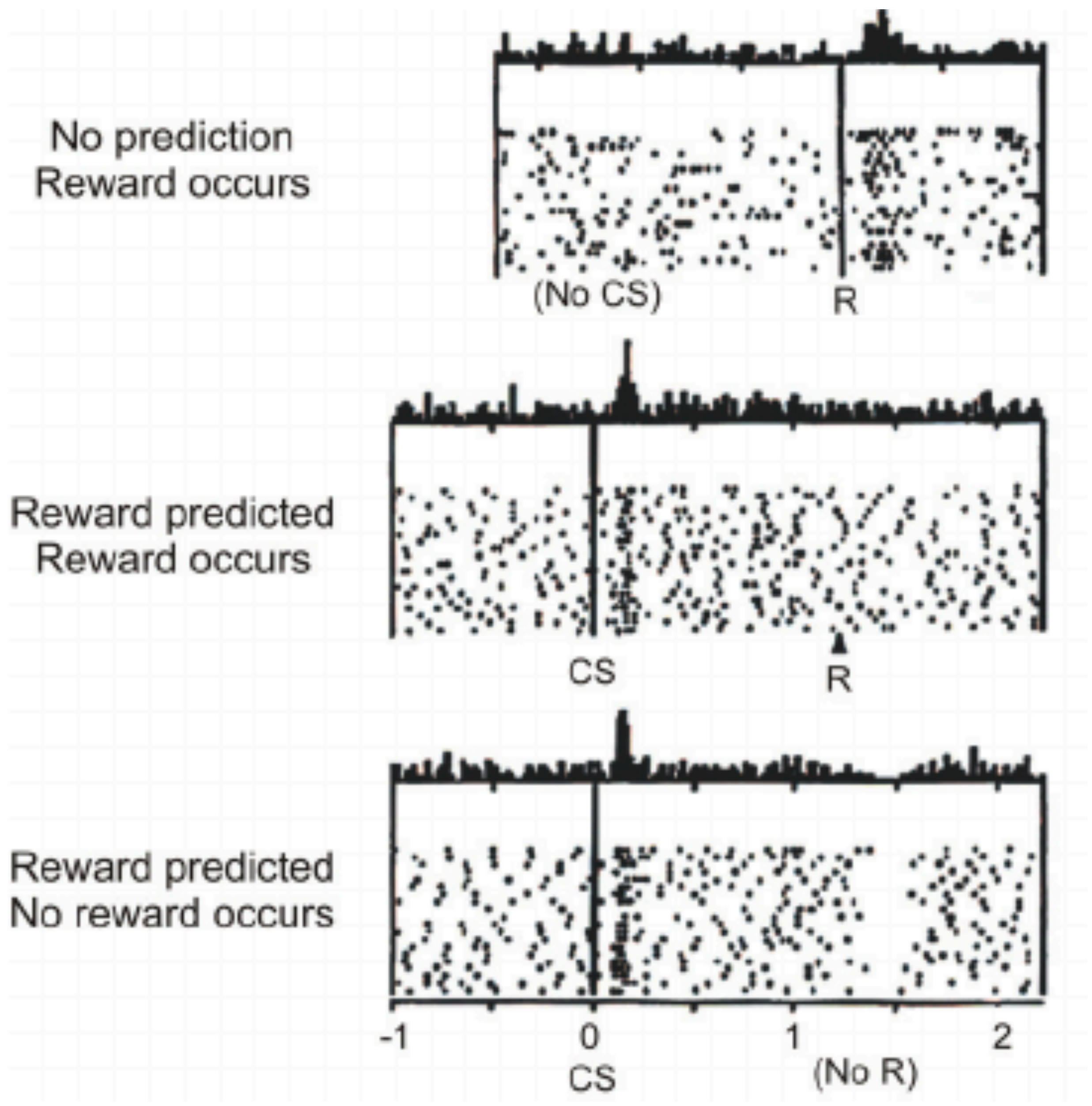
$$V(s) \leftarrow V(s) + \eta\left(r + \underline{\gamma V(s')} - V(s)\right)$$

Temporal Difference Learning

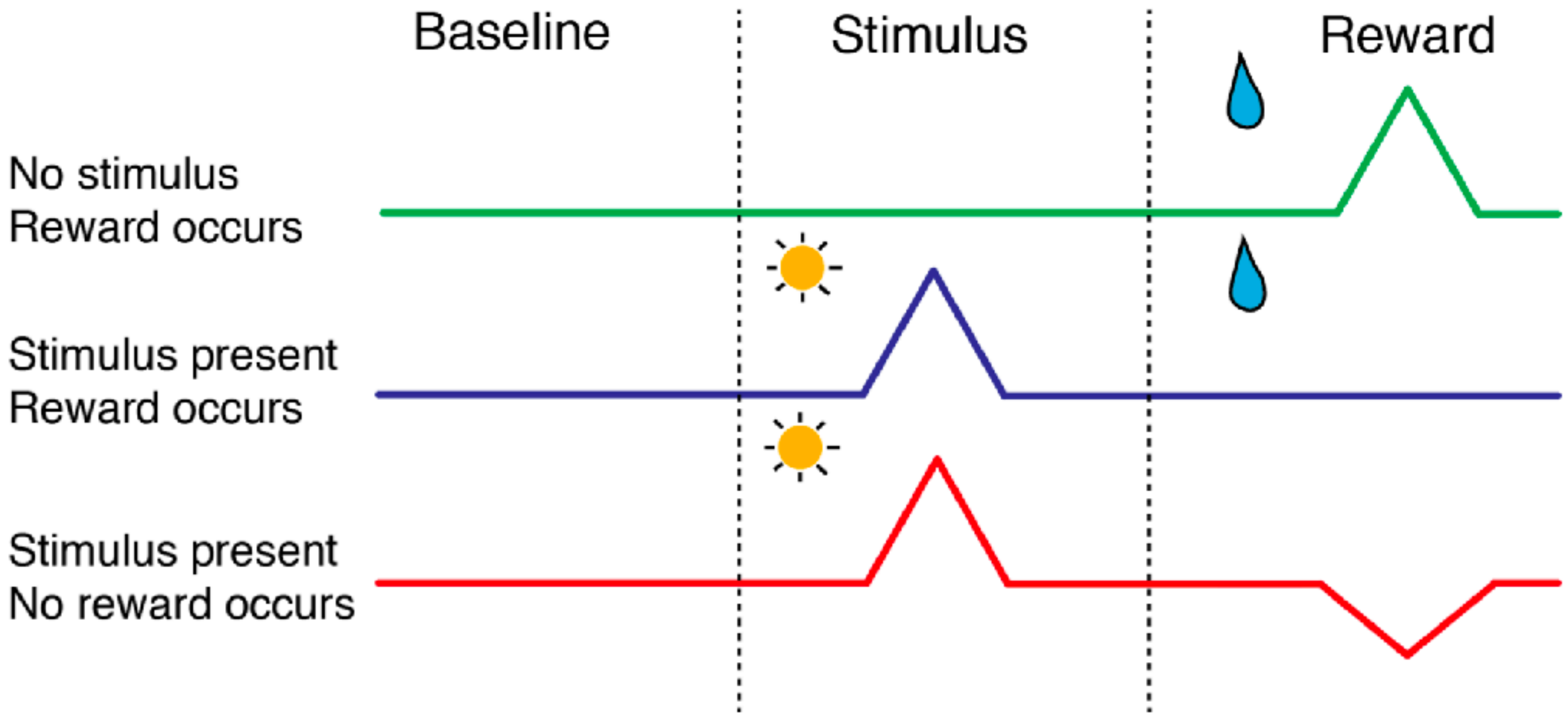
For simplicity, I'm omitting the temporal difference (TD) error, which looks like:

$$V(s) \leftarrow V(s) + \eta \left(r + \gamma V(s') - V(s) \right)$$

Dopamine Reward Prediction Error



Schultz et al. (1997)



Policy aka choice function

Goal of Human RL?

Policy aka choice function

Goal of Human RL?

Predict behavior using a model. This tells us we have understood some aspect of human learning.

Policy aka choice function

Goal of Human RL?

Predict behavior using a model. This tells us we have understood some aspect of human learning.

We want to make probabilistic predictions, since people don't behave deterministically.

Policy aka choice function

Goal of Human RL?

Predict behavior using a model. This tells us we have understood some aspect of human learning.

We want to make probabilistic predictions, since people don't behave deterministically.

Epsilon greedy:

$$\text{action} = \begin{cases} \max_a Q(a) & \text{with } p(1-\epsilon) \\ \text{random action} & \text{with } p(\epsilon) \end{cases}$$

Policy aka choice function

Goal of Human RL?

Predict behavior using a model. This tells us we have understood some aspect of human learning.

We want to make probabilistic predictions, since people don't behave deterministically.

Epsilon greedy:

$$\text{action} = \begin{cases} \max_a Q(a) & \text{with } p(1-\epsilon) \\ \text{random action} & \text{with } p(\epsilon) \end{cases}$$

Softmax:

$$P(a_i) = \frac{\exp(Q(a_i)/\tau)}{\sum_j^n \exp(Q(a_j)/\tau)}$$

Policy aka choice function

Goal of Human RL?

Predict behavior using a model. This tells us we have understood some aspect of human learning.

We want to make probabilistic predictions, since people don't behave deterministically.

Epsilon greedy:

$$\text{action} = \begin{cases} \max_a Q(a) & \text{with } p(1-\epsilon) \\ \text{random action} & \text{with } p(\epsilon) \end{cases}$$

Softmax:

Q-values → Probabilities

$$\begin{bmatrix} 1.3 \\ 5.1 \\ 0.7 \\ 1.1 \end{bmatrix} \rightarrow \frac{\exp(Q(a_i)/\tau)}{\sum_j^n \exp(Q(a_j)/\tau)} \rightarrow \begin{bmatrix} 0.002 \\ 0.90 \\ 0.05 \\ 0.02 \end{bmatrix}$$

Policy aka choice function

Goal of Human RL?

Predict behavior using a model. This tells us we have understood some aspect of human learning.

We want to make probabilistic predictions, since people don't behave deterministically.

Epsilon greedy:

$$\text{action} = \begin{cases} \max_a Q(a) & \text{with } p(1-\epsilon) \\ \text{random action} & \text{with } p(\epsilon) \end{cases}$$

Softmax:

Q-values → Probabilities

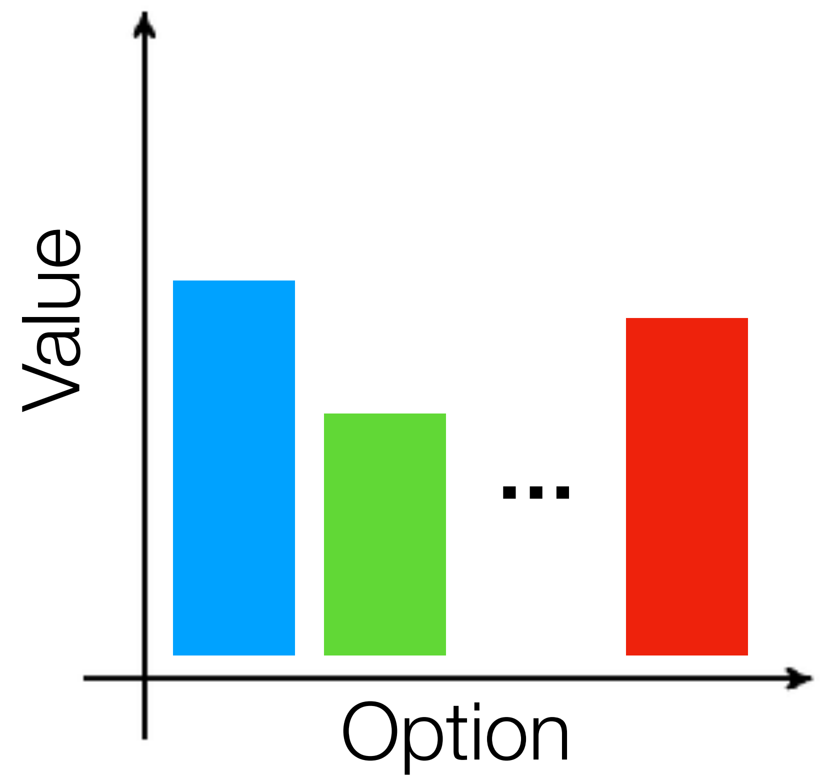
$$\begin{bmatrix} 1.3 \\ 5.1 \\ 0.7 \\ 1.1 \end{bmatrix} \rightarrow \frac{\exp(Q(a_i)/\tau)}{\sum_j^n \exp(Q(a_j)/\tau)} \rightarrow \begin{bmatrix} 0.002 \\ 0.90 \\ 0.05 \\ 0.02 \end{bmatrix}$$

ϵ and τ model forms of exploration, but don't distinguish between experienced vs. inexperienced options

Bayesian reinforcement learning

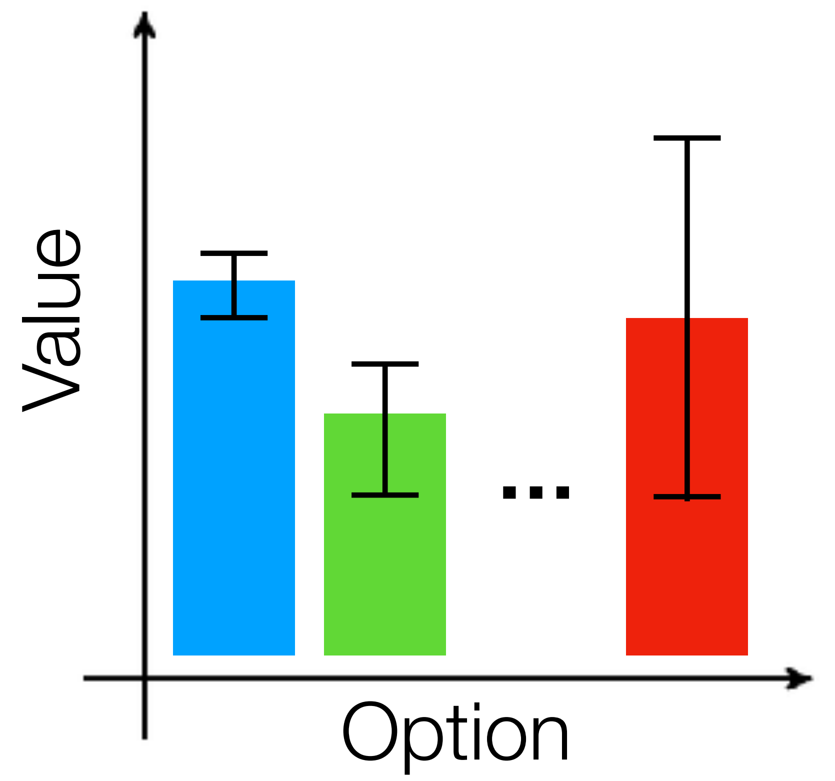
We can describe a Bayesian variant of the RW model using a Kalman filter:

Bayesian reinforcement learning



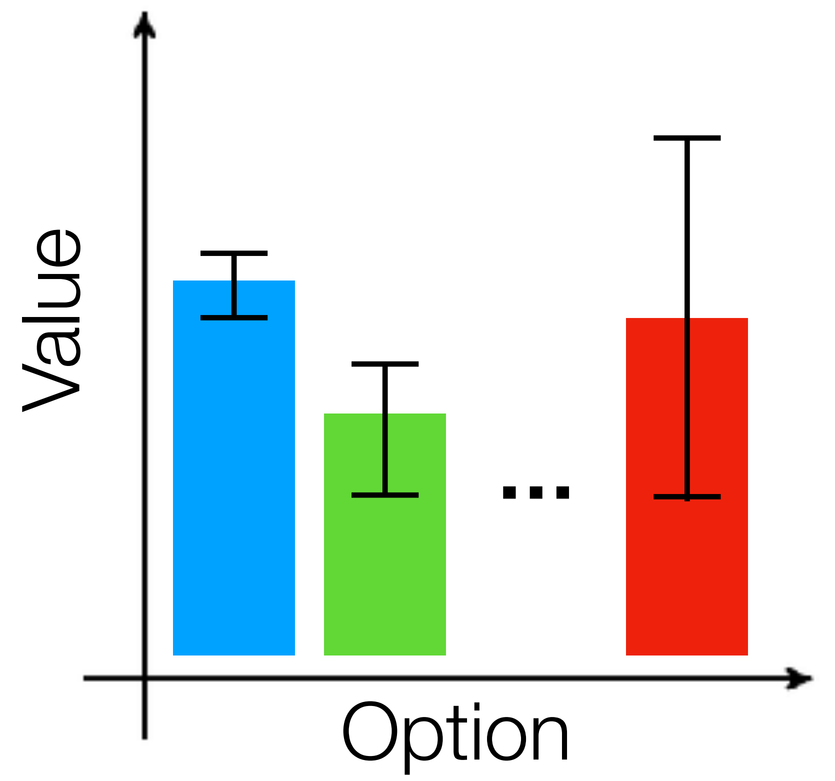
We can describe a Bayesian variant of the RW model using a Kalman filter:

Bayesian reinforcement learning



We can describe a Bayesian variant of the RW model using a Kalman filter:

Bayesian reinforcement learning



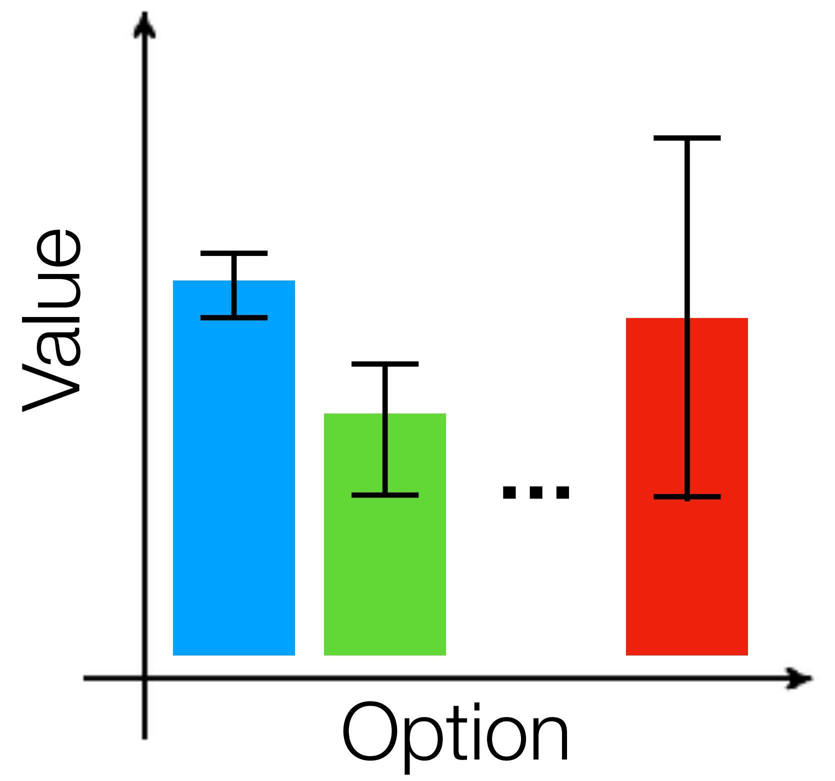
We can describe a Bayesian variant of the RW model using a Kalman filter:

Normally distributed posterior over rewards, where the mean is $Q_{i,t}$:

$$p(r_{i,t} \,|\, \mathcal{D}_{t-1}) = \mathcal{N}(Q_{i,t}, \sigma^2_{i,t})$$

where $\mathcal{D}_t = \left[a_0, r_0, a_1, r_1, \ldots\right]$ collect previous choices and rewards

Bayesian reinforcement learning



We can describe a Bayesian variant of the RW model using a Kalman filter:

Normally distributed posterior over rewards, where the mean is $Q_{i,t}$:

$$p(r_{i,t} \mid \mathcal{D}_{t-1}) = \mathcal{N}(Q_{i,t}, \sigma^2_{i,t})$$

where $\mathcal{D}_t = \left[a_0, r_0, a_1, r_1, \ldots\right]$ collect previous choices and rewards

Bayesian updates:

Mean: $$Q_{i,t+1} = Q_{i,t} + k_{i,t}\left[r_{i,t} - Q_{i,t}\right]$$

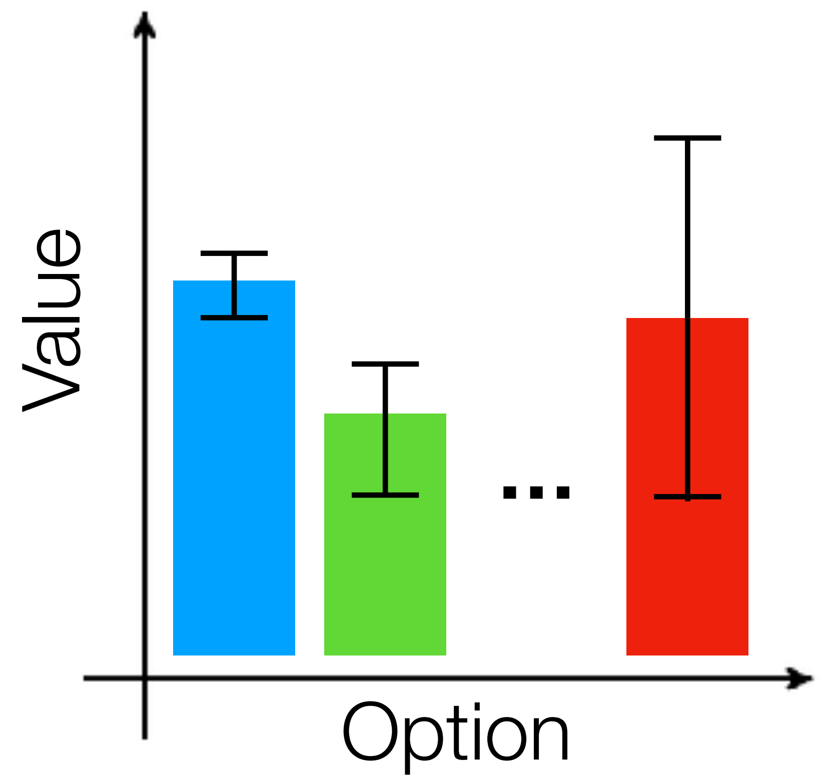
Variance: $$\sigma^2_{i,t+1} = \left[1 - k_{i,t}\right]\sigma^2_{i,t}$$

Kalman Gain (learning rate):

$$k_{i,t} = \begin{cases} \dfrac{\sigma^2_{i,t}}{\sigma^2_{i,t} + \sigma^2_\epsilon} & \text{if } a_t = i \\ 0 & \text{otherwise} \end{cases}$$

Error variance σ^2_ϵ is a free parameter

Bayesian reinforcement learning



We can describe a Bayesian variant of the RW model using a Kalman filter:

Normally distributed posterior over rewards, where the mean is $Q_{i,t}$:

$$p(r_{i,t} \mid \mathcal{D}_{t-1}) = \mathcal{N}(Q_{i,t}, \sigma^2_{i,t})$$

where $\mathcal{D}_t = [a_0, r_0, a_1, r_1, \ldots]$ collect previous choices and rewards

Bayesian updates:

Mean: $$Q_{i,t+1} = Q_{i,t} + k_{i,t}\left[r_{i,t} - Q_{i,t}\right]$$

Variance: $$\sigma^2_{i,t+1} = \left[1 - k_{i,t}\right]\sigma^2_{i,t}$$

Kalman Gain (learning rate):

$$k_{i,t} = \begin{cases} \dfrac{\sigma^2_{i,t}}{\sigma^2_{i,t} + \sigma^2_\epsilon} & \text{if } a_t = i \\ 0 & \text{otherwise} \end{cases}$$

Error variance σ^2_ϵ is a free parameter

Strictly, this is a KF variant known as a Bayesian mean tracker (BMT), assuming stationary rewards

Policies for Bayesian models

We can now use uncertainty estimates to inform our policy and explore more efficiently.

Policies for Bayesian models

We can now use uncertainty estimates to inform our policy and explore more efficiently.

Thompson Sampling:

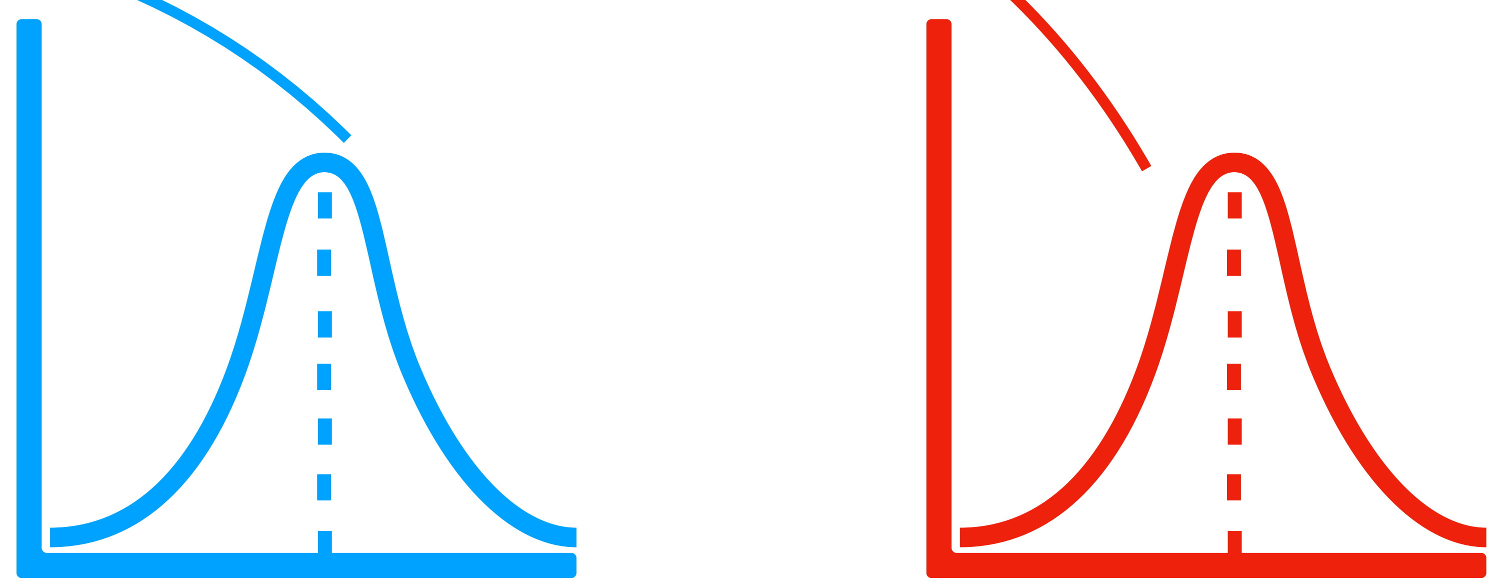
$$P(a_i) = P(r_i > r_{j \neq i})$$

Policies for Bayesian models

We can now use uncertainty estimates to inform our policy and explore more efficiently.

Thompson Sampling:

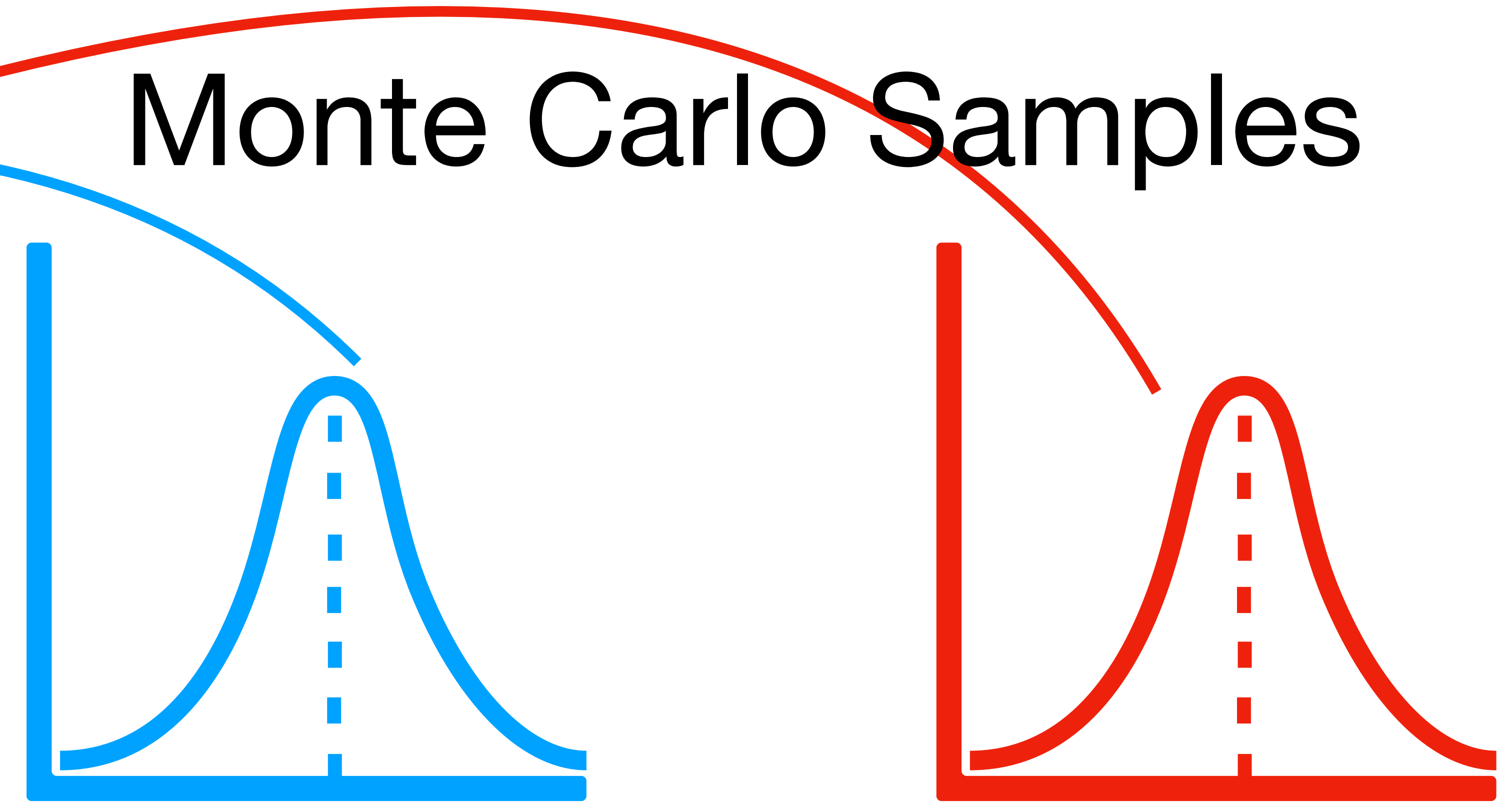
$$P(a_i) = P(r_i > r_{j \neq i})$$



Policies for Bayesian models

We can now use uncertainty estimates to inform our policy and explore more efficiently.

Thompson Sampling:



$$P(a_i) = P(r_i > r_{j \neq i})$$

Upper Confidence Bound Sampling:

$$UCB(a_i) = Q_i + \beta \sqrt{\sigma_i^2}$$

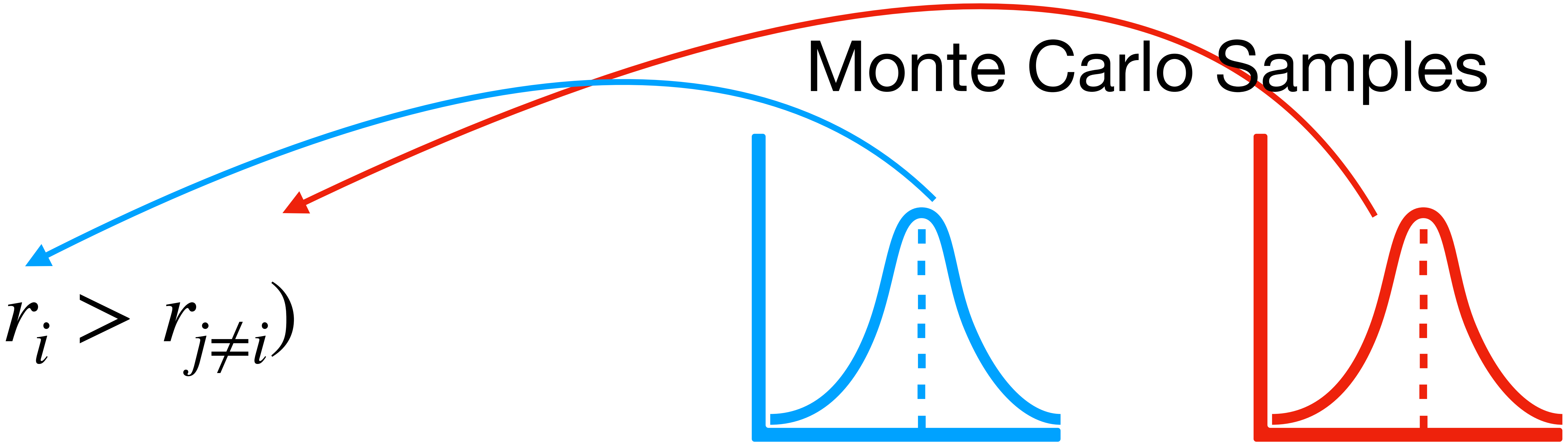
Policies for Bayesian models

We can now use uncertainty estimates to inform our policy and explore more efficiently.

Thompson Sampling:

$$P(a_i) = P(r_i > r_{j \neq i})$$

Monte Carlo Samples

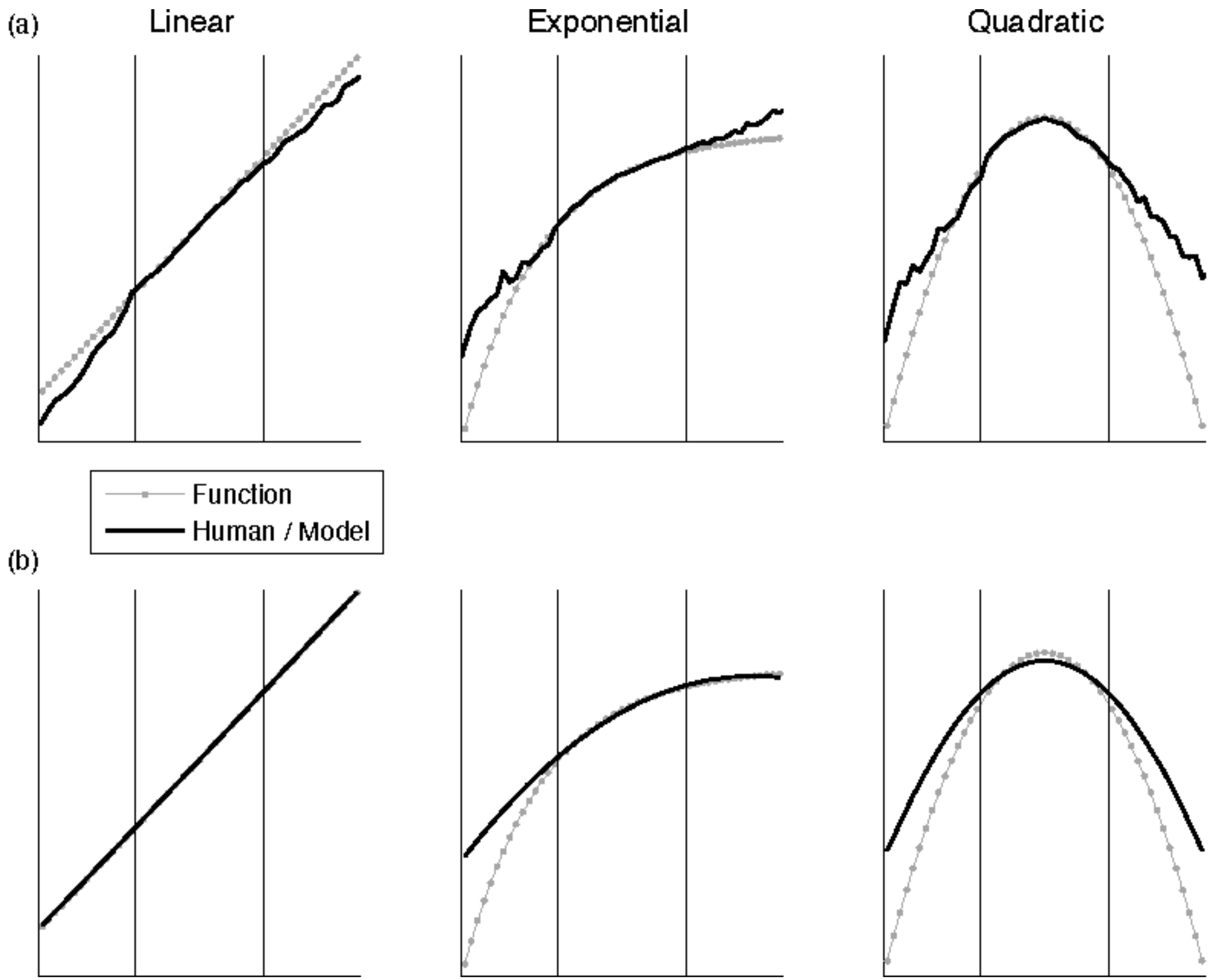


Upper Confidence Bound Sampling:

$$UCB(a_i) = Q_i + \beta \sqrt{\sigma_i^2}$$

β **models exploration directed towards uncertain choices**

Function approximation?



(c)

Model	Linear	Exponential	Quadratic
EXAM	.999	.997	.961
Linear	.999	.989	.470
Quad	.997	.997	.901
RBF	.999	.997	.882
LQ	.999	.997	.886
LR	.999	.997	.892
RQ	.998	.994	.878
LRQ	.999	.995	.877

Griffiths et al., (2008)

Summary

Summary

RL framework for learning value functions and policies

Summary

RL framework for learning value functions and policies

- Tabular methods vs. value function approximation

Summary

RL framework for learning value functions and policies

- Tabular methods vs. value function approximation
- Exhaustive iteration over states vs. predictions about novel states

Summary

RL framework for learning value functions and policies

- Tabular methods vs. value function approximation
- Exhaustive iteration over states vs. predictions about novel states

Modeling human learners with RL models

Summary

RL framework for learning value functions and policies

- Tabular methods vs. value function approximation
- Exhaustive iteration over states vs. predictions about novel states

Modeling human learners with RL models

- Can we understand something about the efficiency of human learning?

Summary

RL framework for learning value functions and policies

- Tabular methods vs. value function approximation
- Exhaustive iteration over states vs. predictions about novel states

Modeling human learners with RL models

- Can we understand something about the efficiency of human learning?
  - How humans navigate the exploration-exploitation dilemma

Summary

RL framework for learning value functions and policies

- Tabular methods vs. value function approximation
- Exhaustive iteration over states vs. predictions about novel states

Modeling human learners with RL models

- Can we understand something about the efficiency of human learning?
  - How humans navigate the exploration-exploitation dilemma
- RW and Q-learning resemble tabular methods that have access to random or noisey exploration

Summary

RL framework for learning value functions and policies

- Tabular methods vs. value function approximation
- Exhaustive iteration over states vs. predictions about novel states

Modeling human learners with RL models

- Can we understand something about the efficiency of human learning?
    - How humans navigate the exploration-exploitation dilemma
- RW and Q-learning resemble tabular methods that have access to random or noisey exploration
- Kalman filter is a Bayesian model, which can use uncertainty-informed exploration

Summary

RL framework for learning value functions and policies

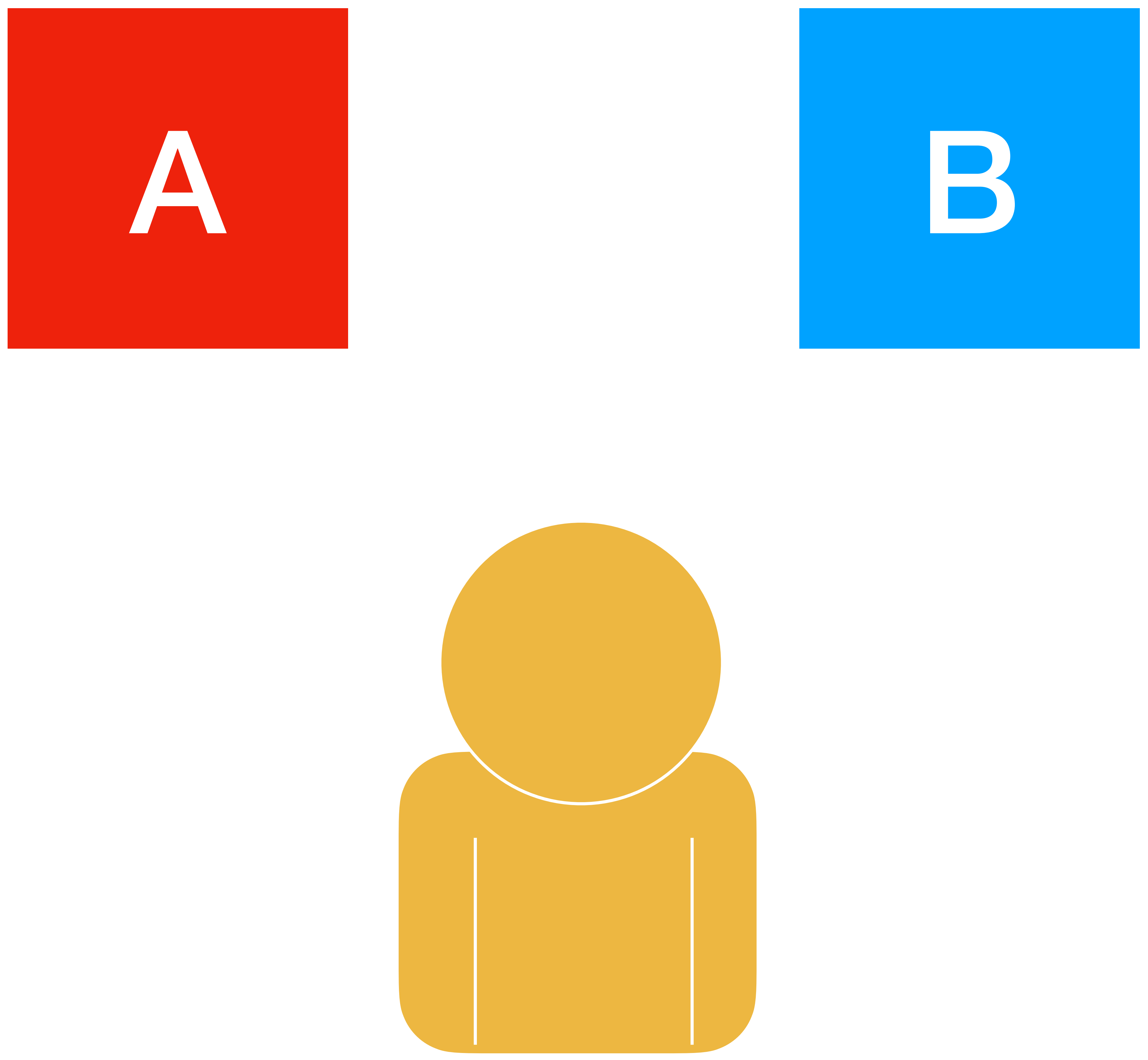
- Tabular methods vs. value function approximation
- Exhaustive iteration over states vs. predictions about novel states

Modeling human learners with RL models

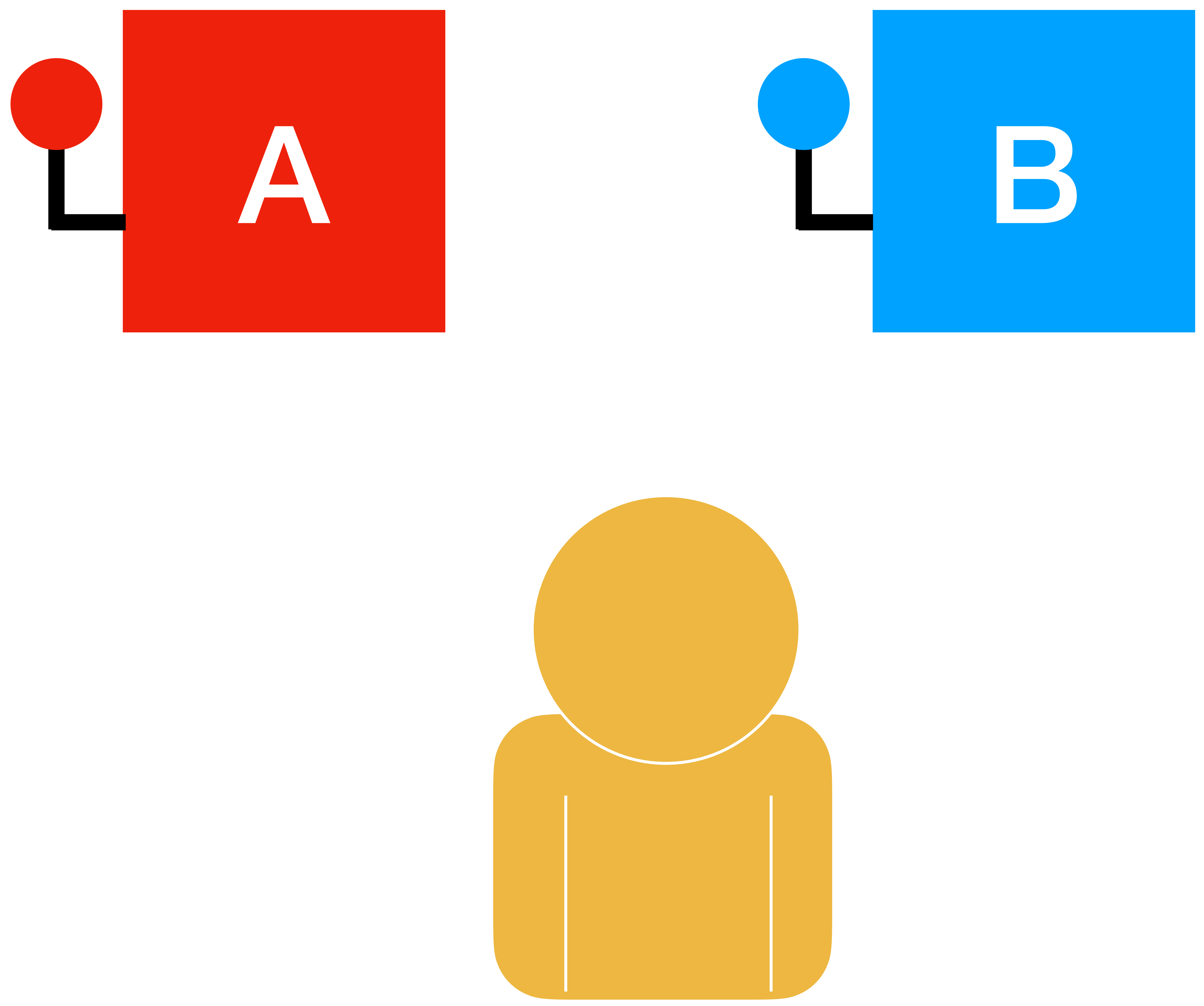
- Can we understand something about the efficiency of human learning?
  - How humans navigate the exploration-exploitation dilemma
- RW and Q-learning resemble tabular methods that have access to random or noisey exploration
- Kalman filter is a Bayesian model, which can use uncertainty-informed exploration
- But what about function approximation?

Break

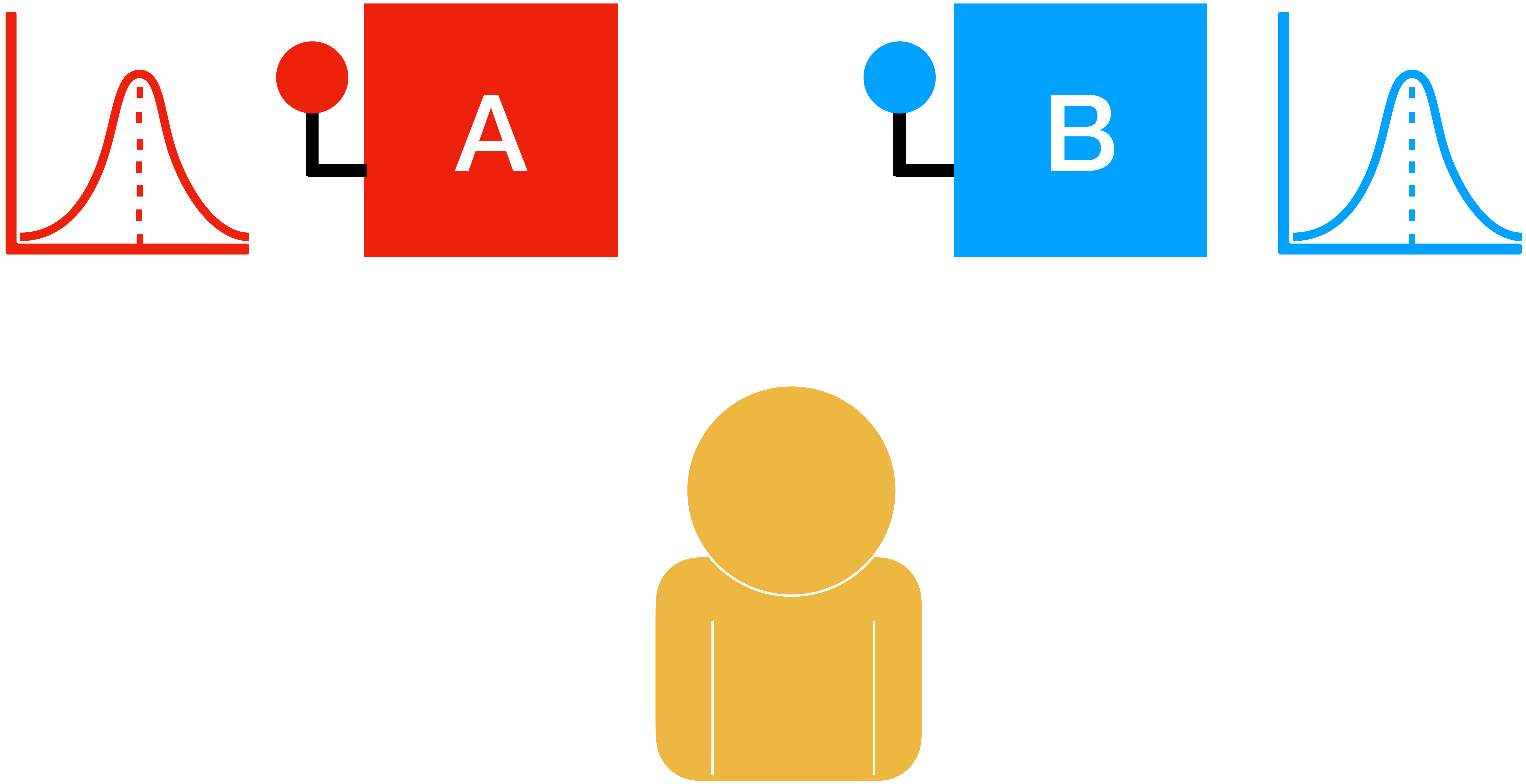
Human learning in the lab



Human learning in the lab

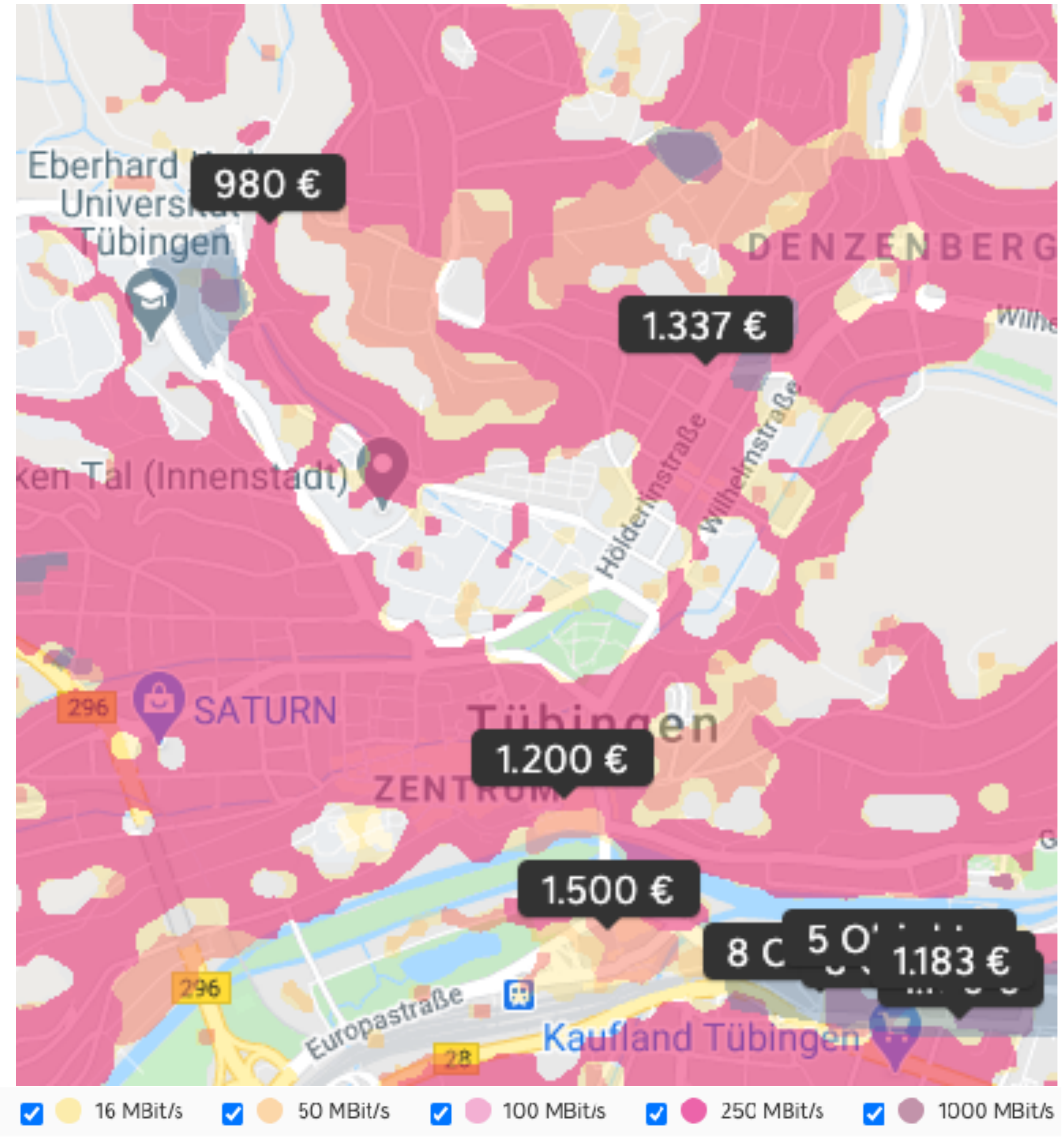


Human learning in the lab



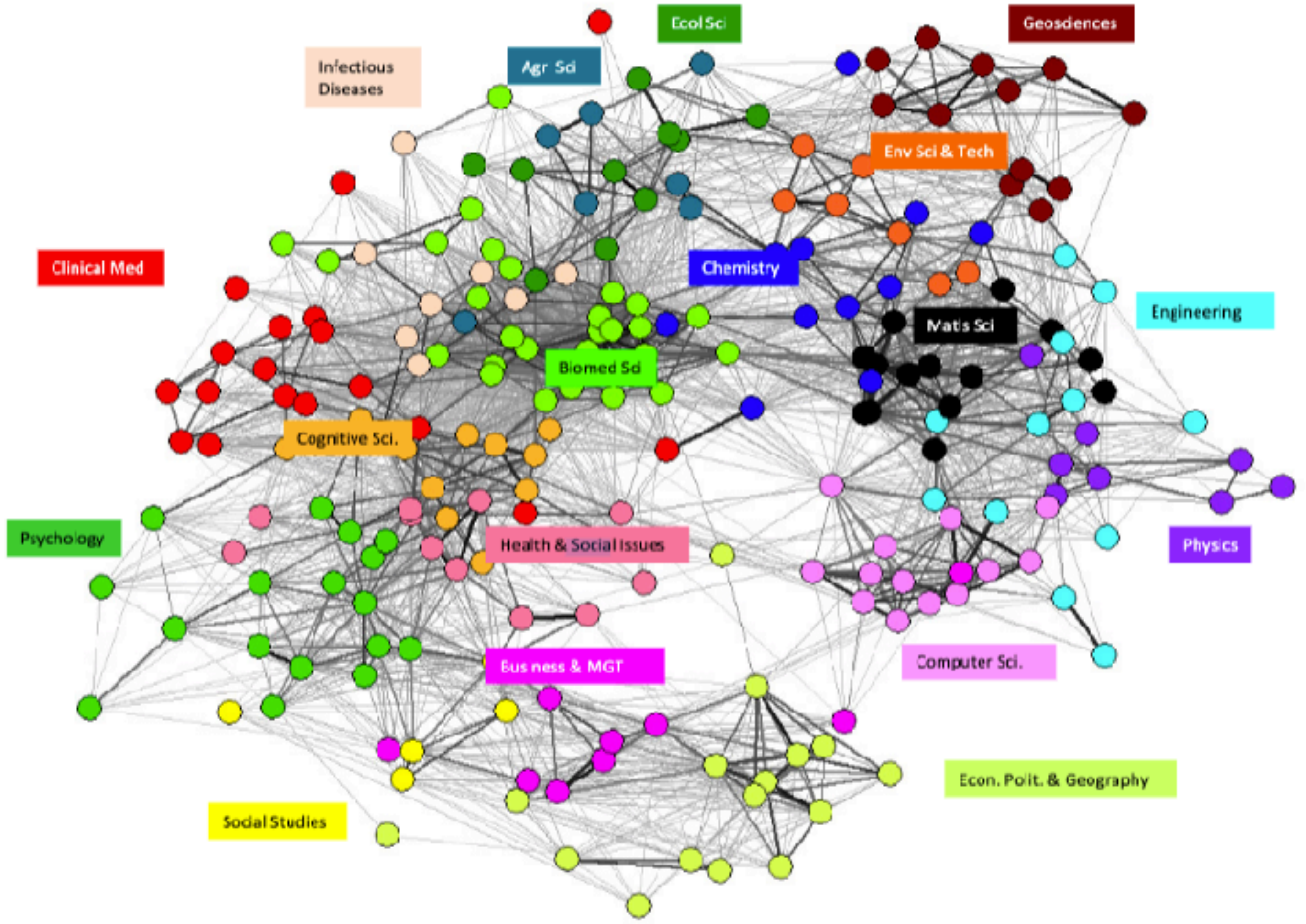
Real life problems

Finding a place to live

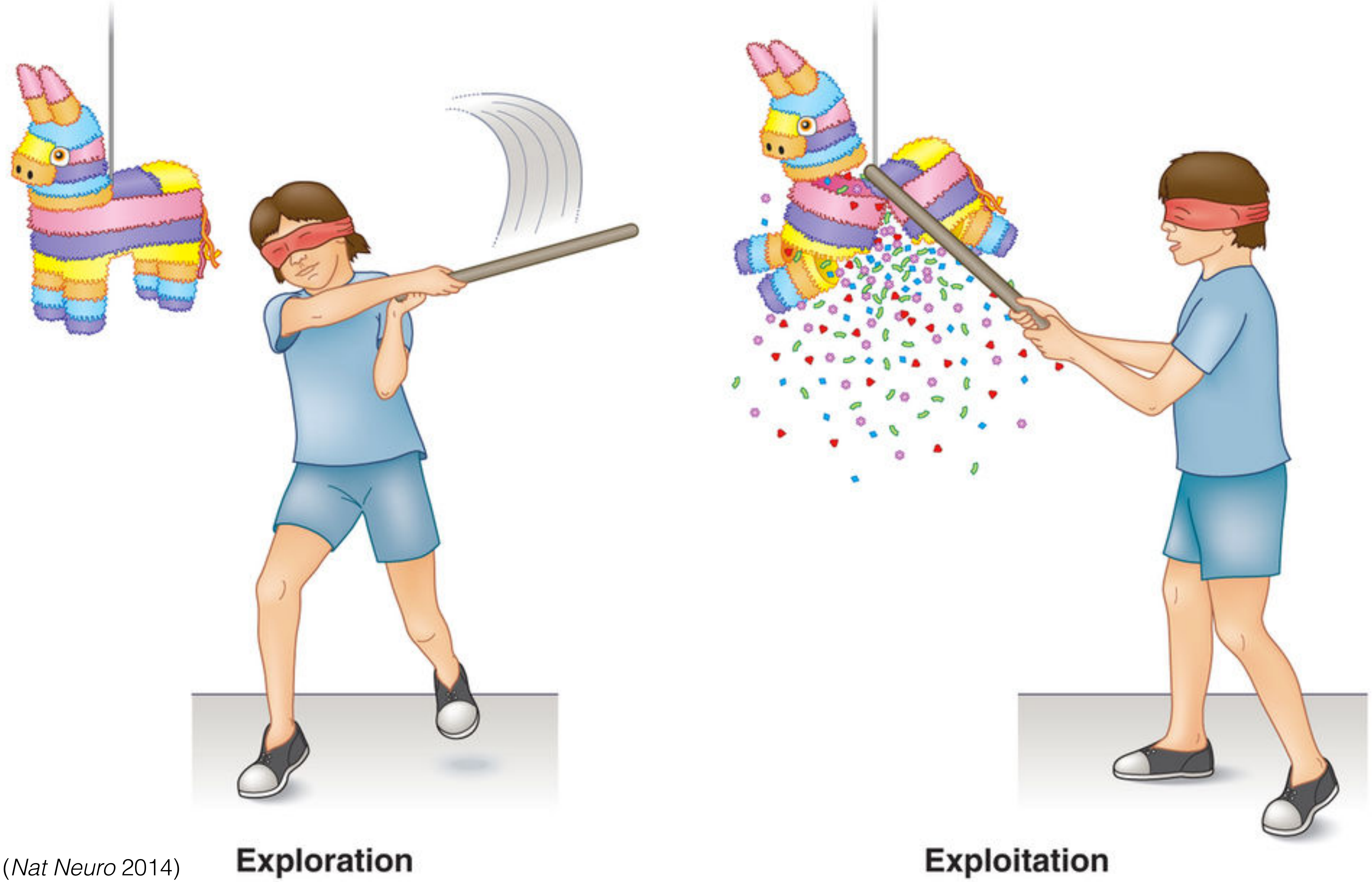


Picking what to eat

Choosing a research topic



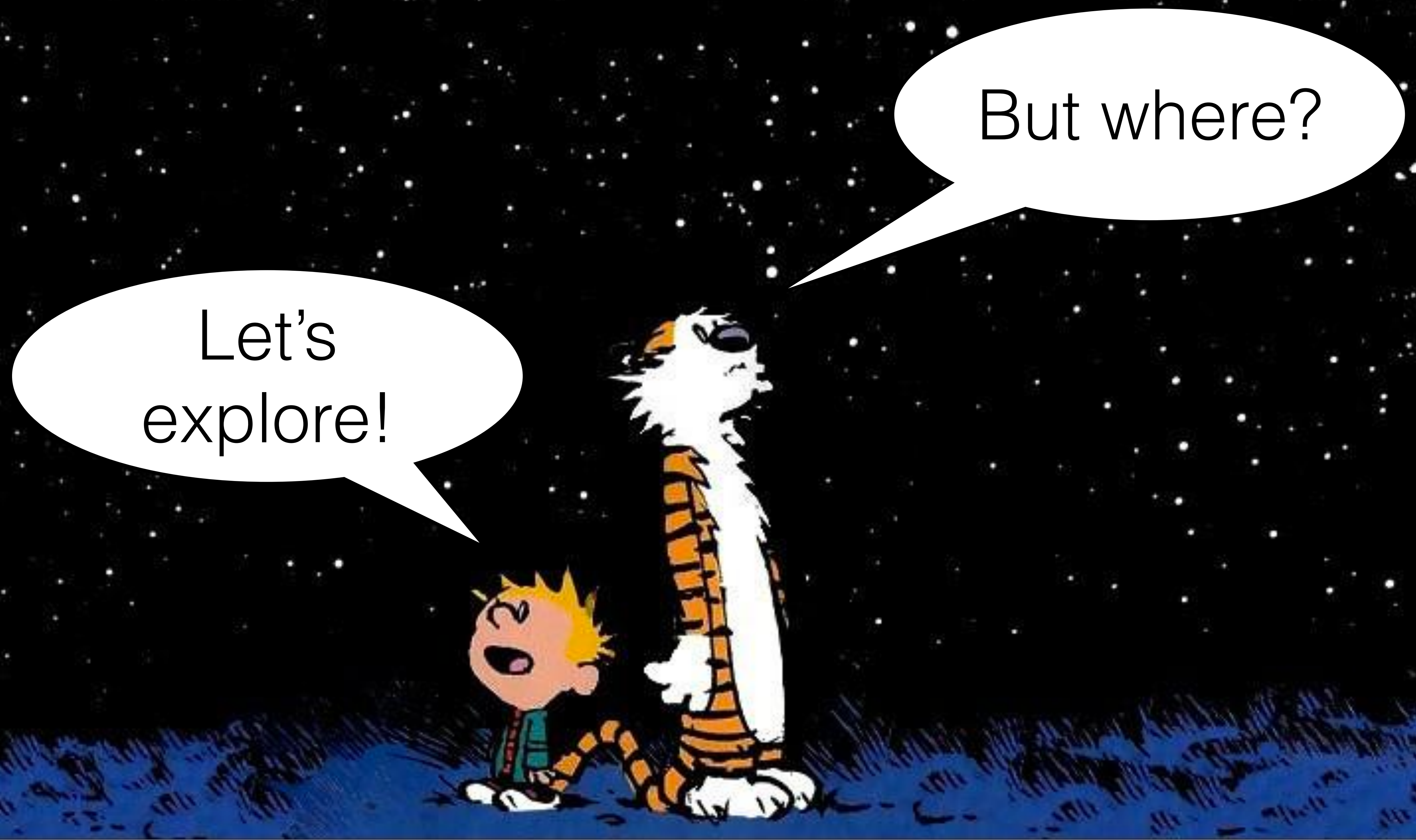
Exploration-Exploitation Dilemma



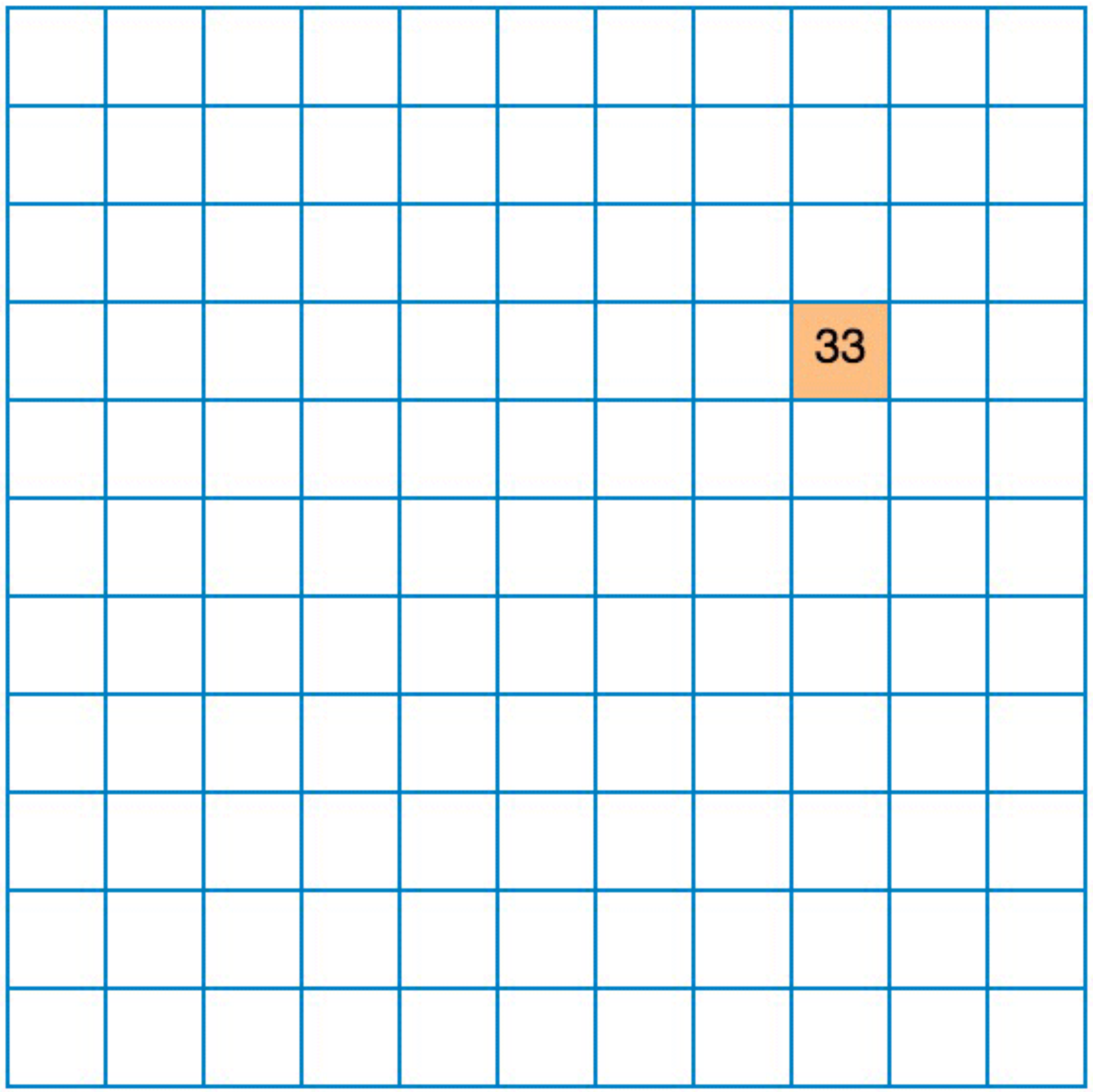
Herzfeld & Shadmehr (*Nat Neuro* 2014)


Let's explore!
But where?

How do people navigate vast environments when we cannot explore all possibilities?



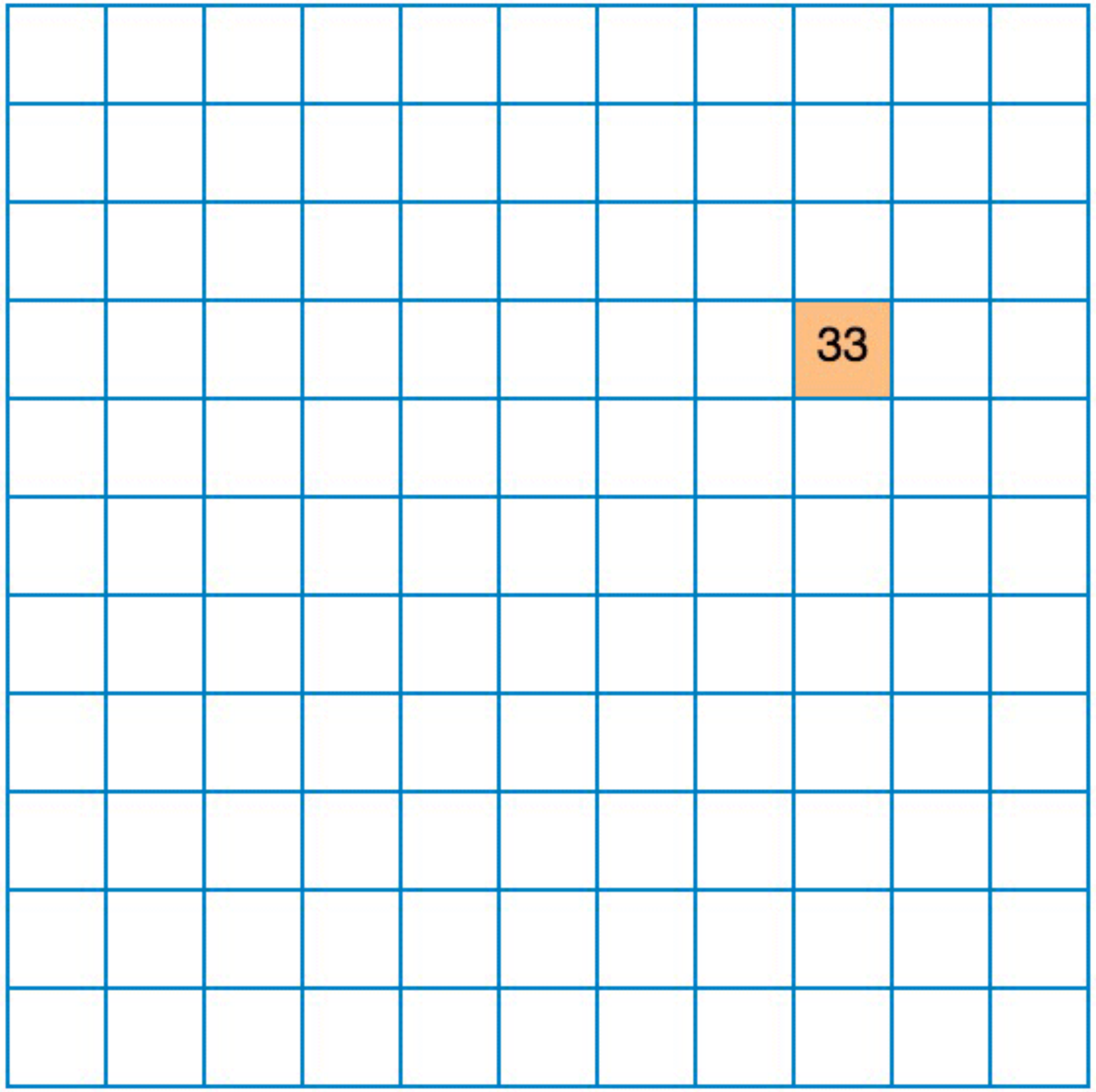
Spatially Correlated Bandit



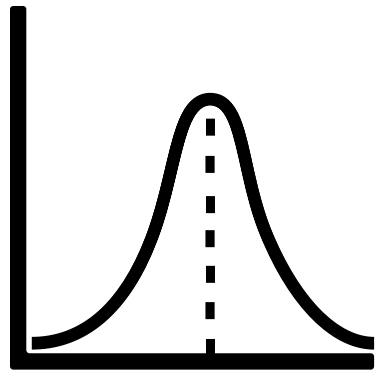
Wu et al., (*Nature Human Behaviour* 2018)

- click tiles on the grid
- maximize reward
- each tile has normally distributed rewards
- nearby tiles have similar rewards
- limited search horizon privileges good generalization & efficient exploration

Spatially Correlated Bandit



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

-  click tiles on the grid
-  maximize reward
-  each tile has normally distributed rewards
-  nearby tiles have similar rewards
-  limited search horizon privileges good generalization & efficient exploration

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

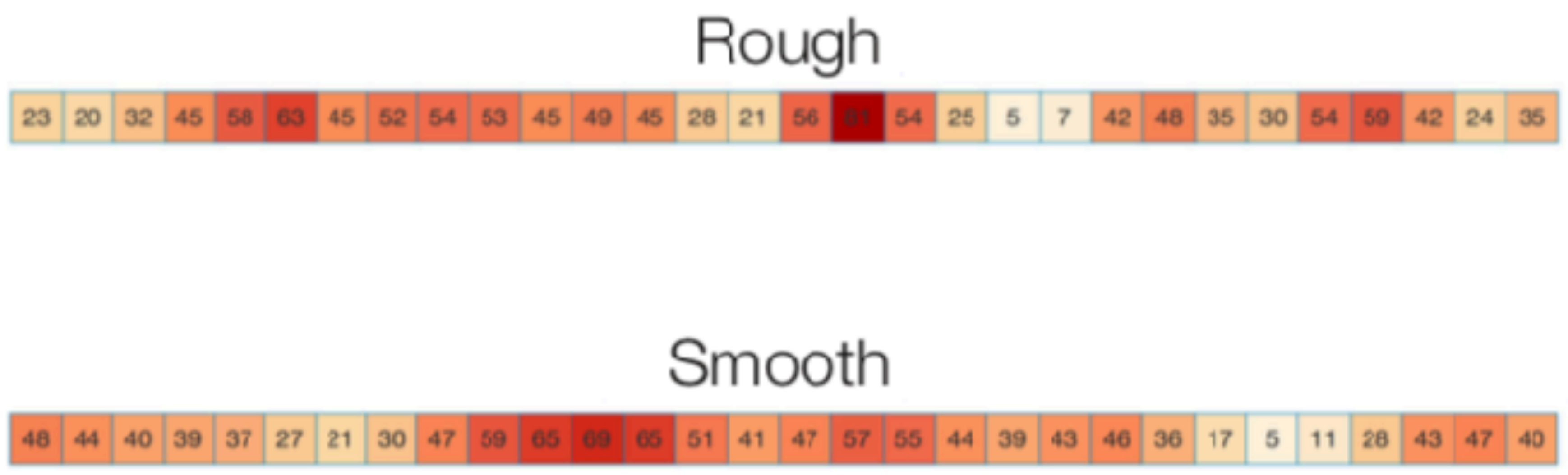
Wu et al., (*Nature Human Behaviour* 2018)

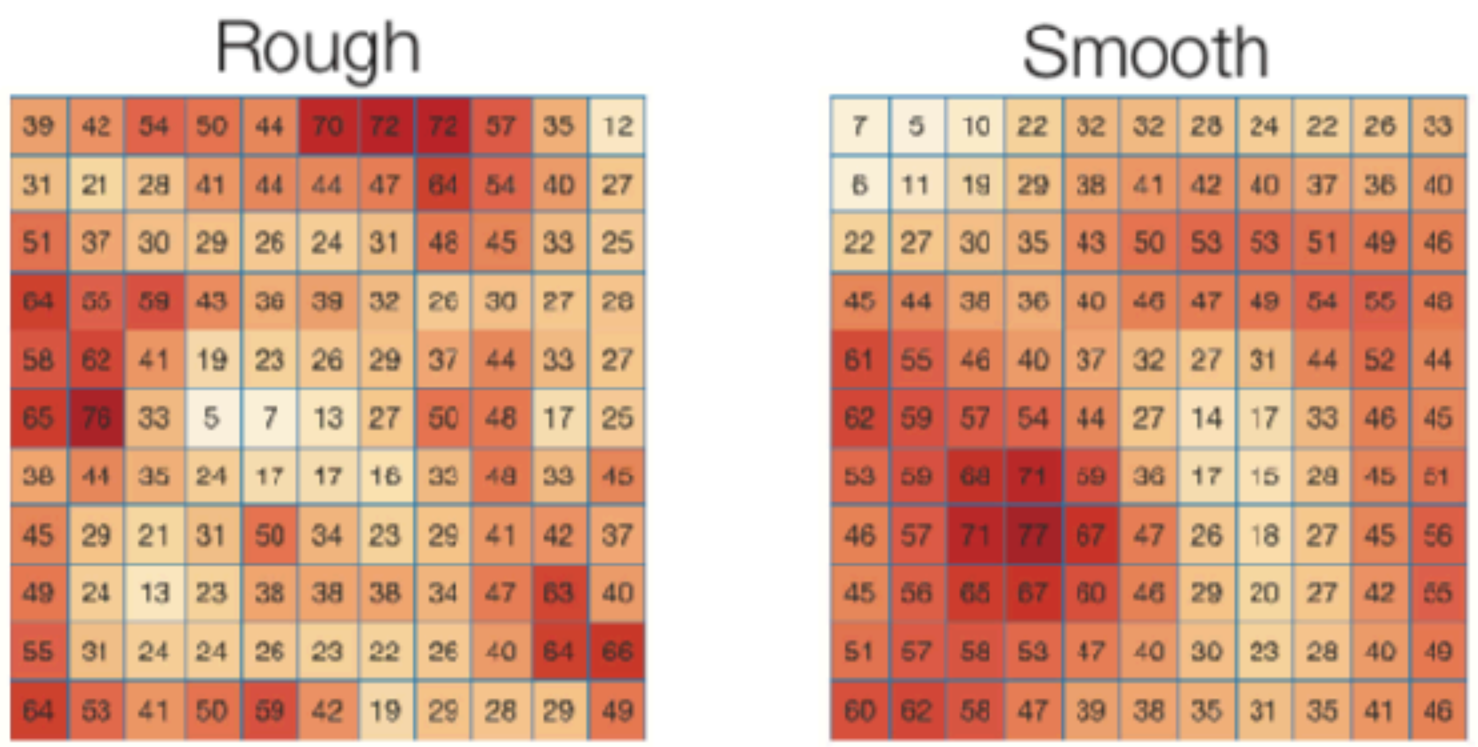
- click tiles on the grid
- maximize reward
- each tile has normally distributed rewards
- nearby tiles have similar rewards
- limited search horizon privileges good generalization & efficient exploration

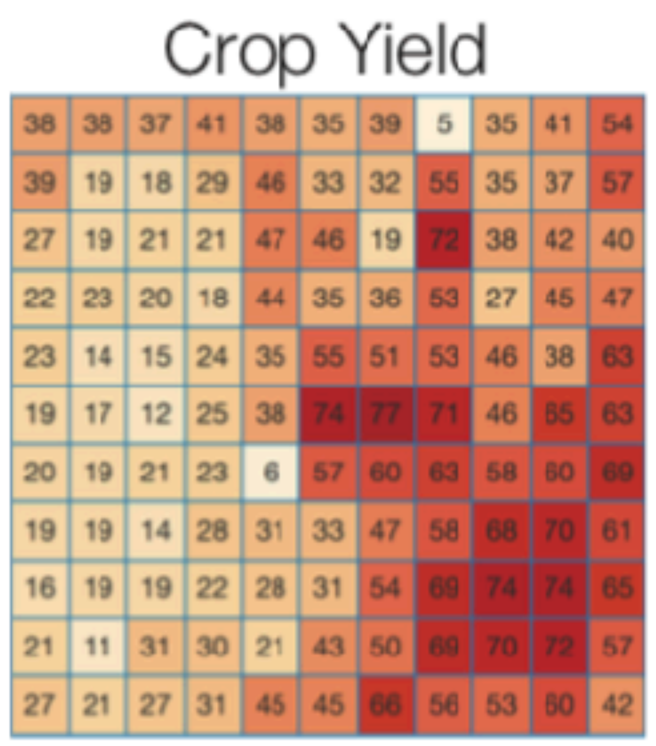
Experiment 1
30-Armed Bandit (Univariate)

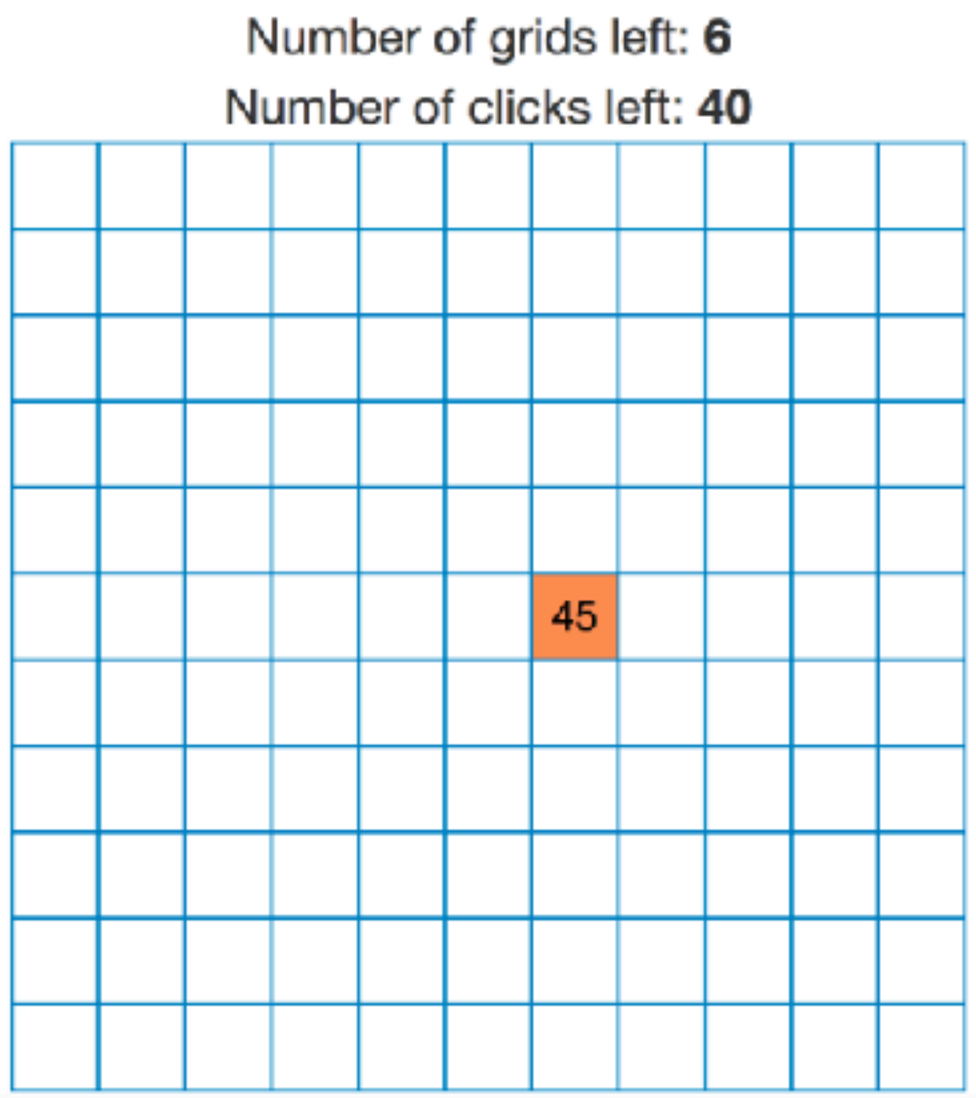
Experiment 2
121-Armed Bandit (Bivariate)

Experiment 3
121-Armed Bandit (Natural)

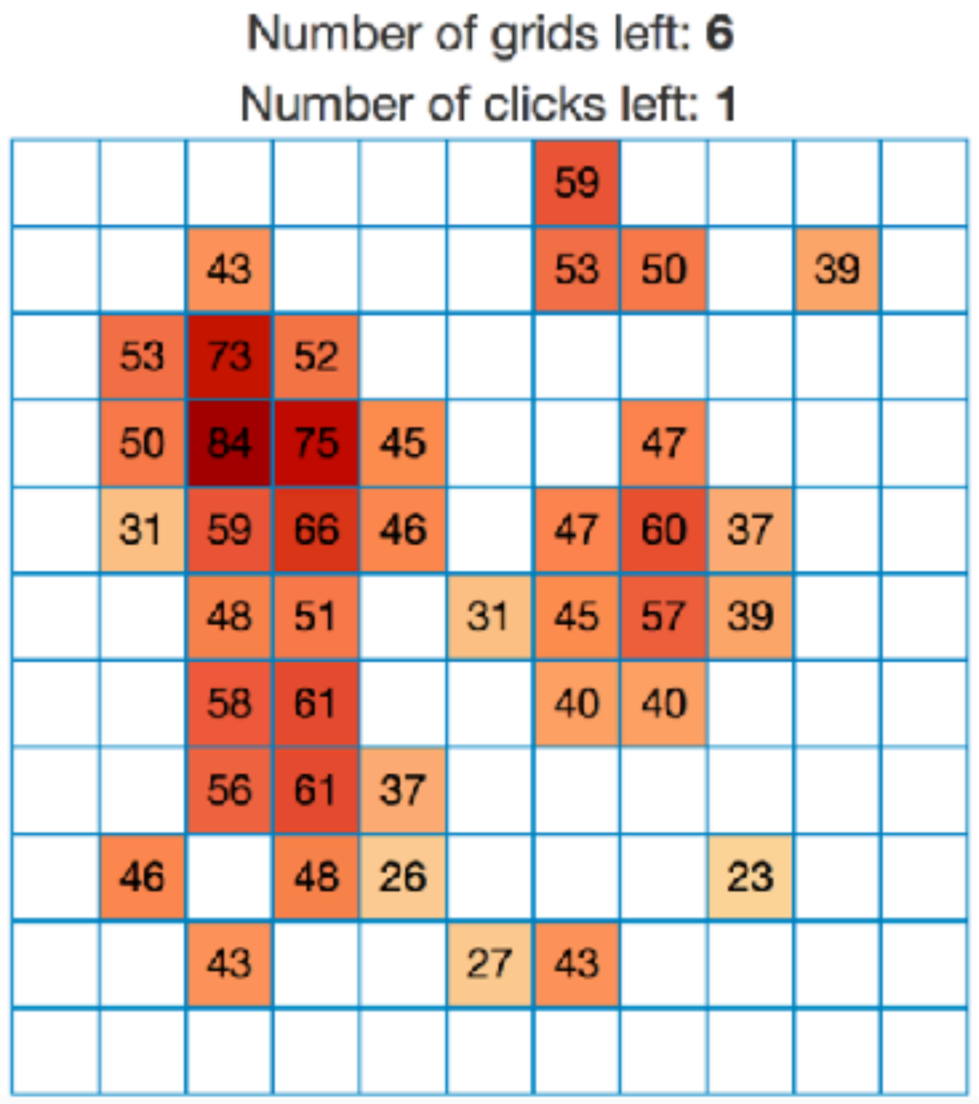






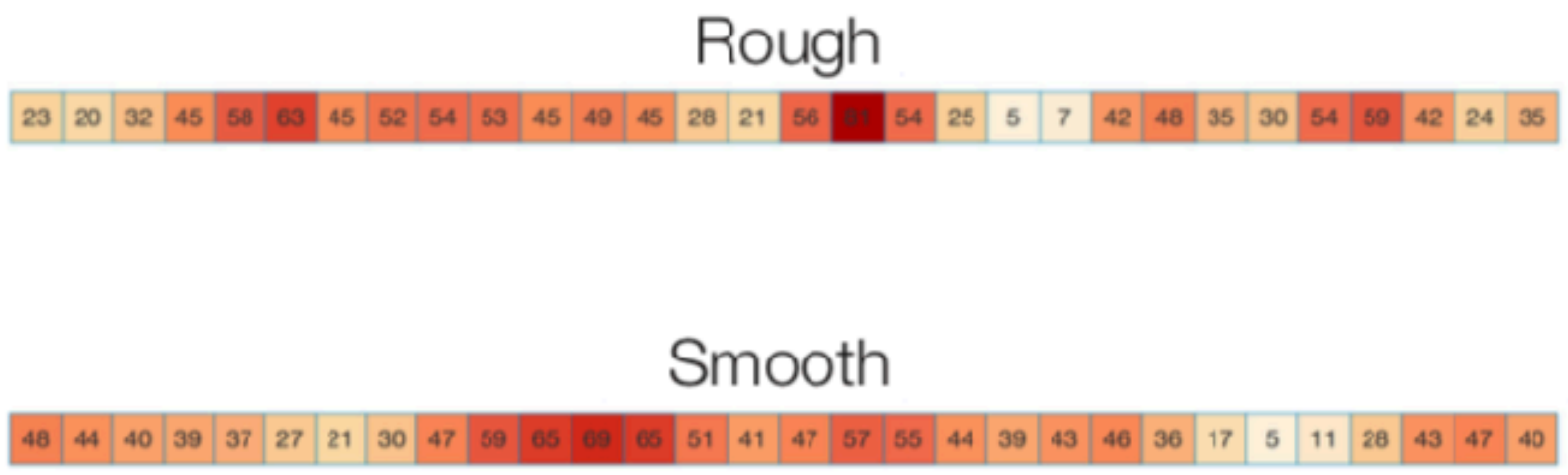


Participants acquired rewards by clicking on new or previously revealed tiles

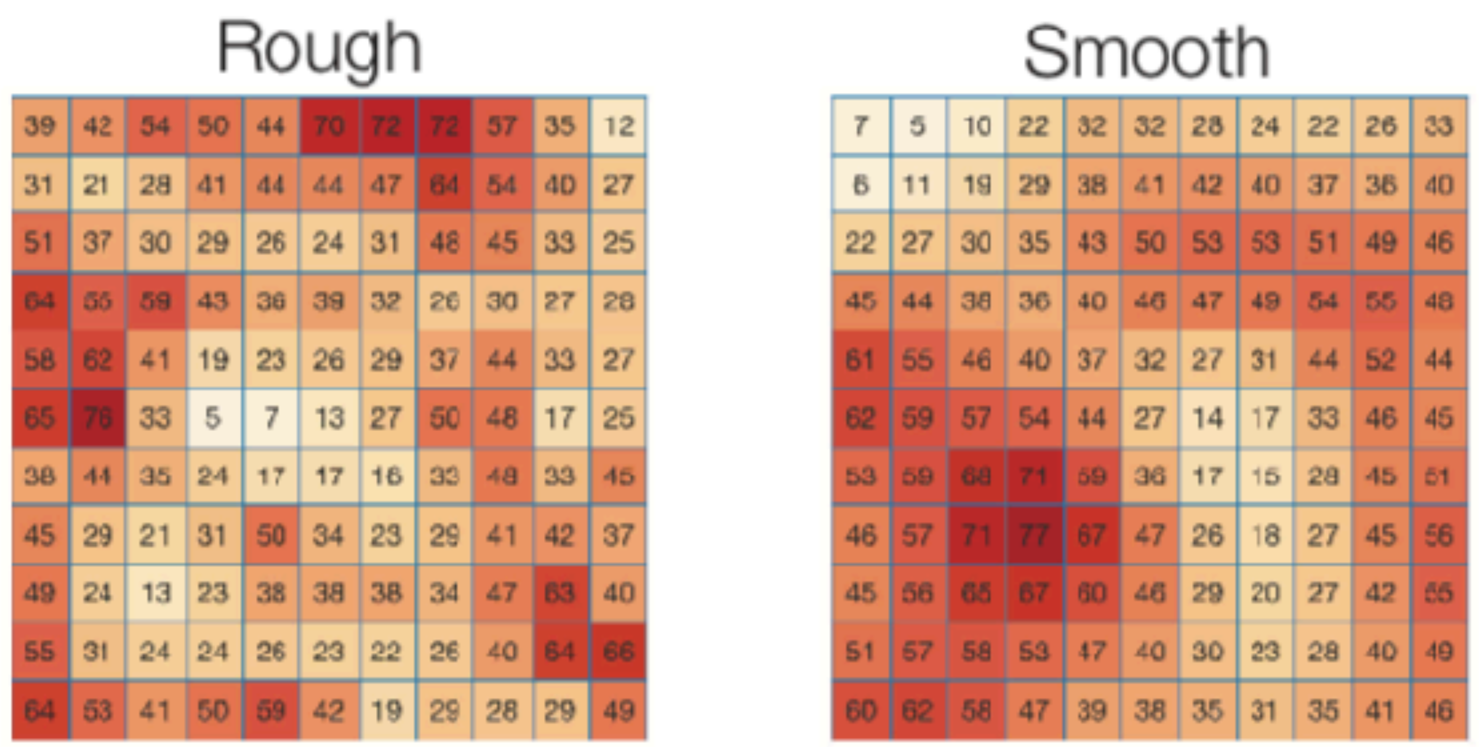


Wu et al., (*Nature Human Behaviour* 2018)

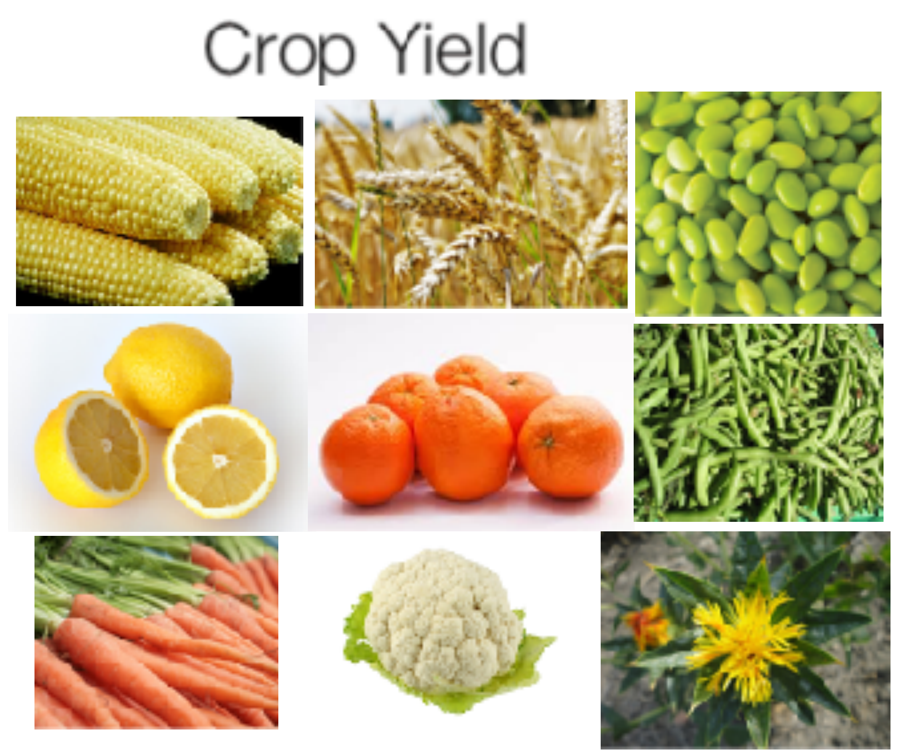
Experiment 1
30-Armed Bandit (Univariate)

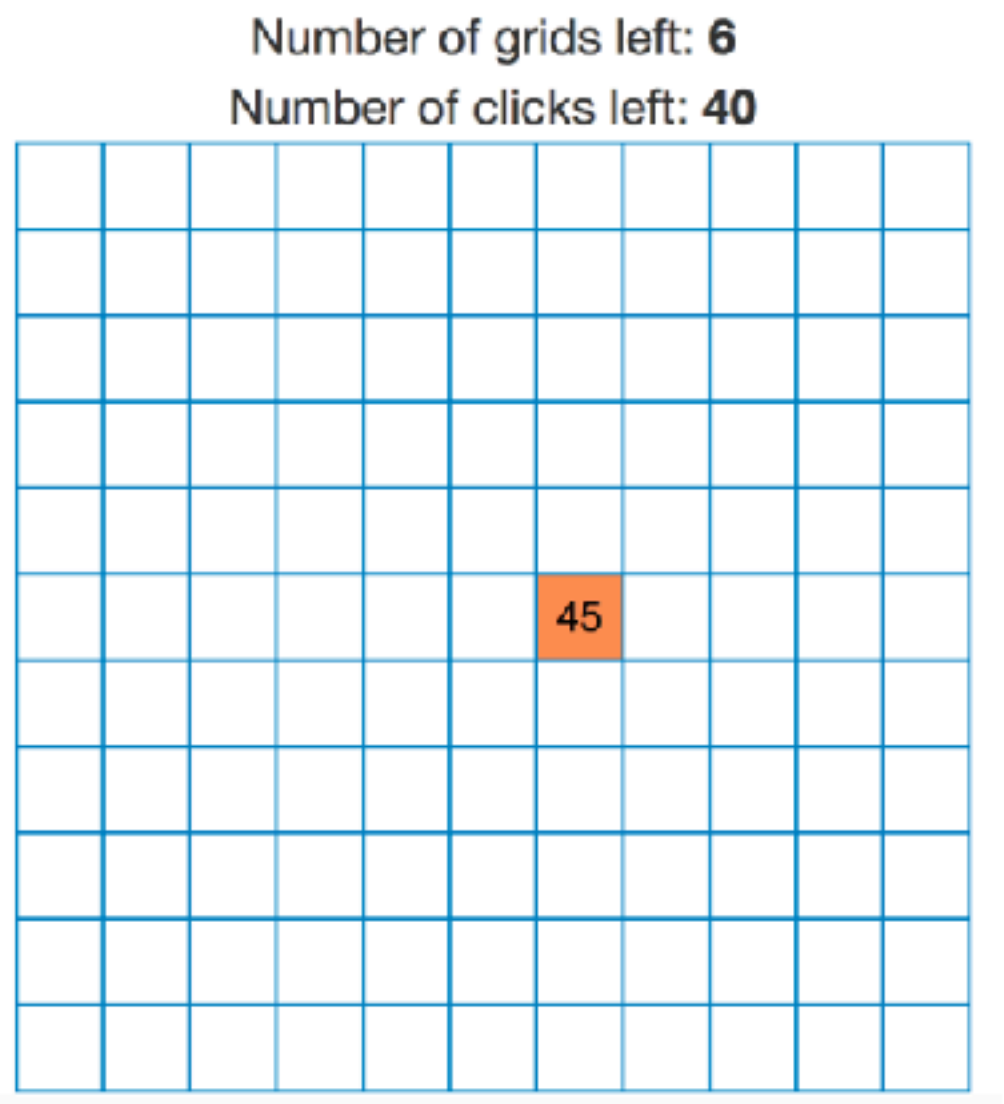


Experiment 2
121-Armed Bandit (Bivariate)

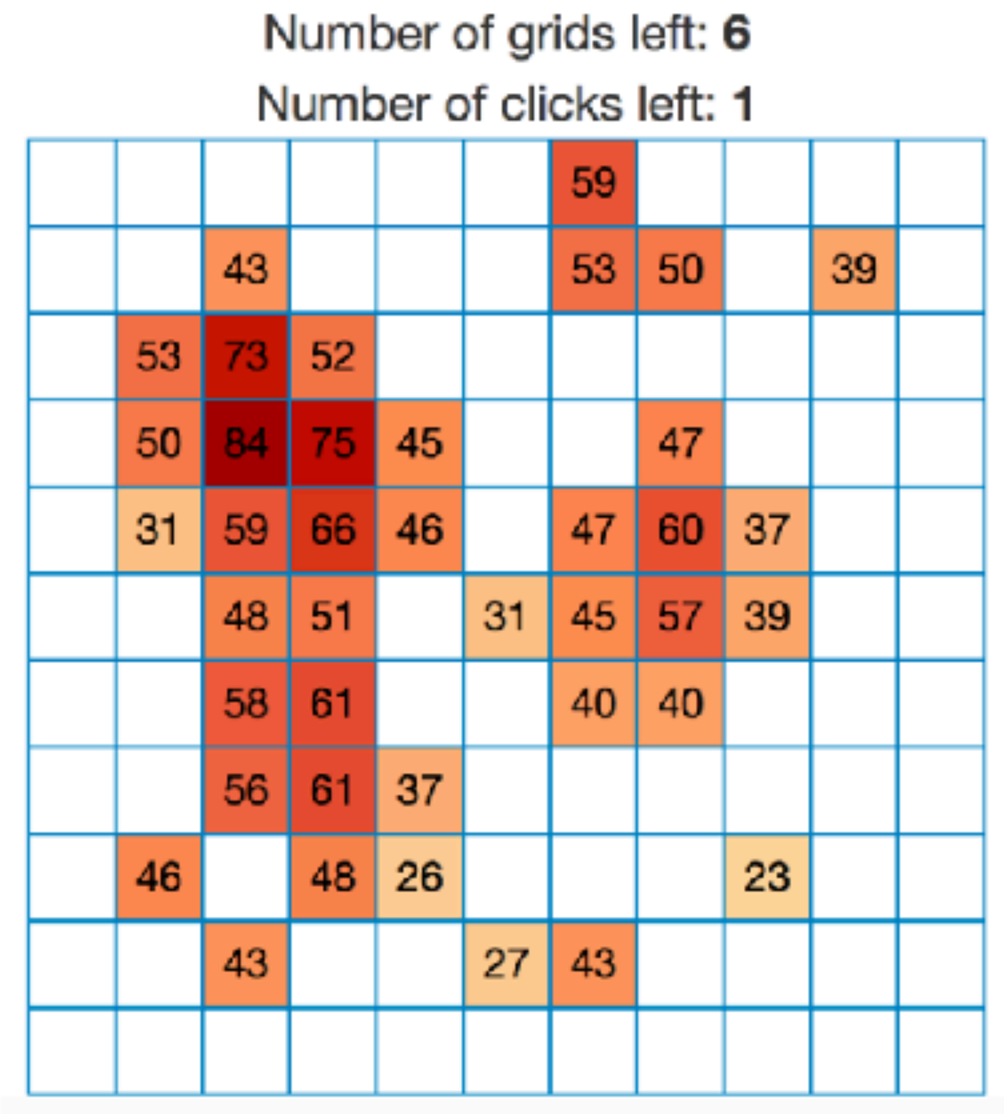


Experiment 3
121-Armed Bandit (Natural)





Participants acquired rewards by clicking on new or previously revealed tiles



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

Modeling Human Search

Modeling Human Search

- Exploration is not performed blindly 

Modeling Human Search

- Exploration is not performed blindly
- Search is guided by generalization

Modeling Human Search

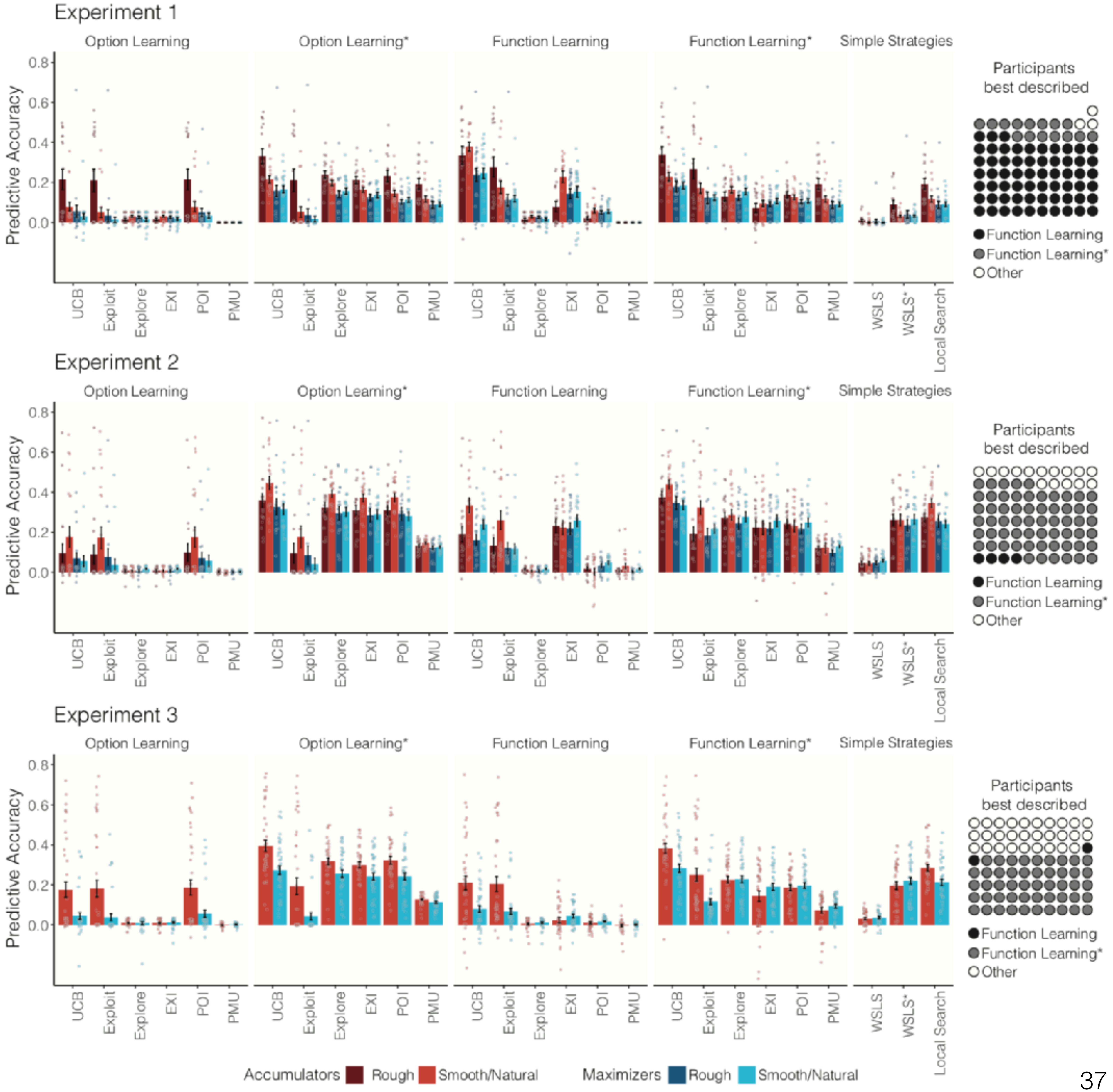
- Exploration is not performed blindly
- Search is guided by generalization
- Generalization as Bayesian inference about novel options:
  - Expected reward
  - Uncertainty

Modeling Human Search

- Exploration is not performed blindly
- Search is guided by generalization
- Generalization as Bayesian inference about novel options:
    - Expected reward
    - Uncertainty
- Human search is directed towards both ingredients

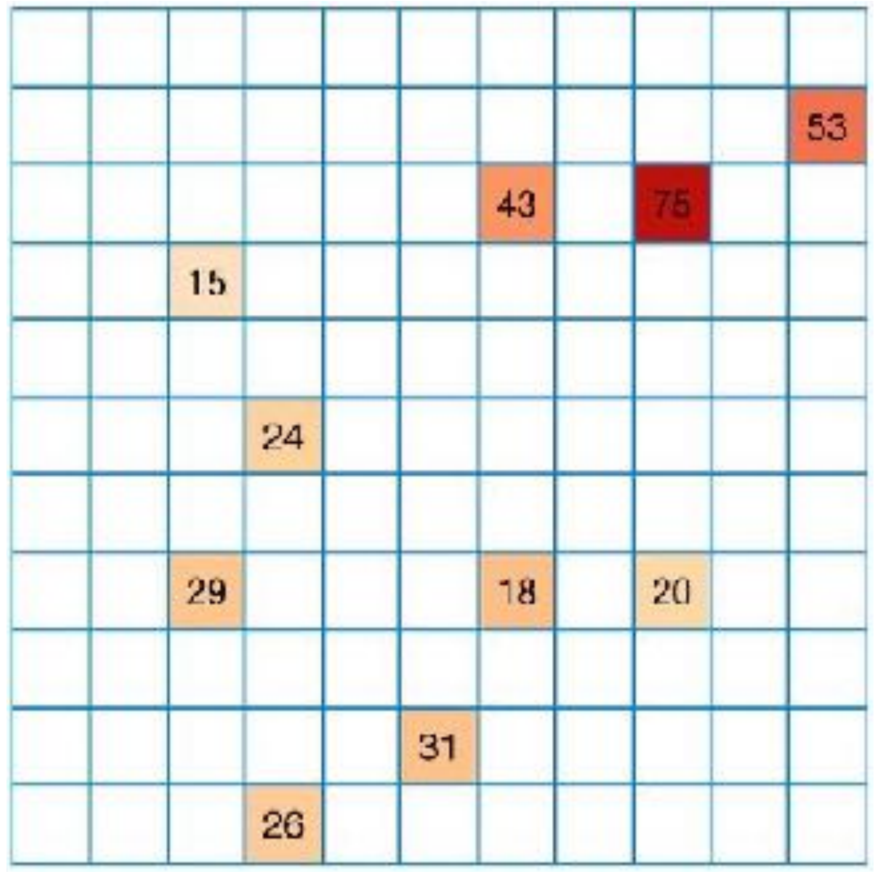
Model comparison

- We performed a large-scale comparison of 27 different models using cross-validated (out-of-sample) predictions
- Some heuristic models but mostly reinforcement learning models * sampling strategies
- … here, I focus on the best model, which consistently outperformed all others across a variety of manipulation checks

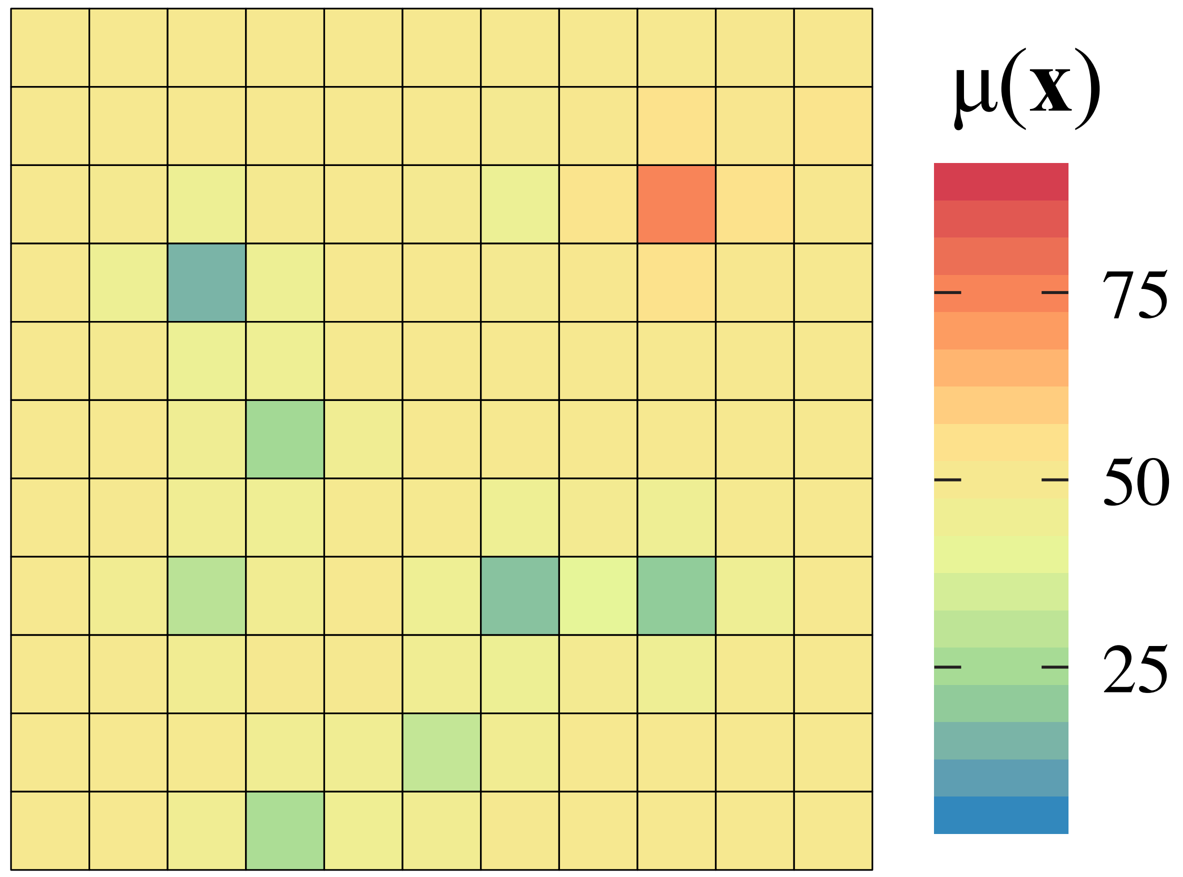


GP-UCB Model

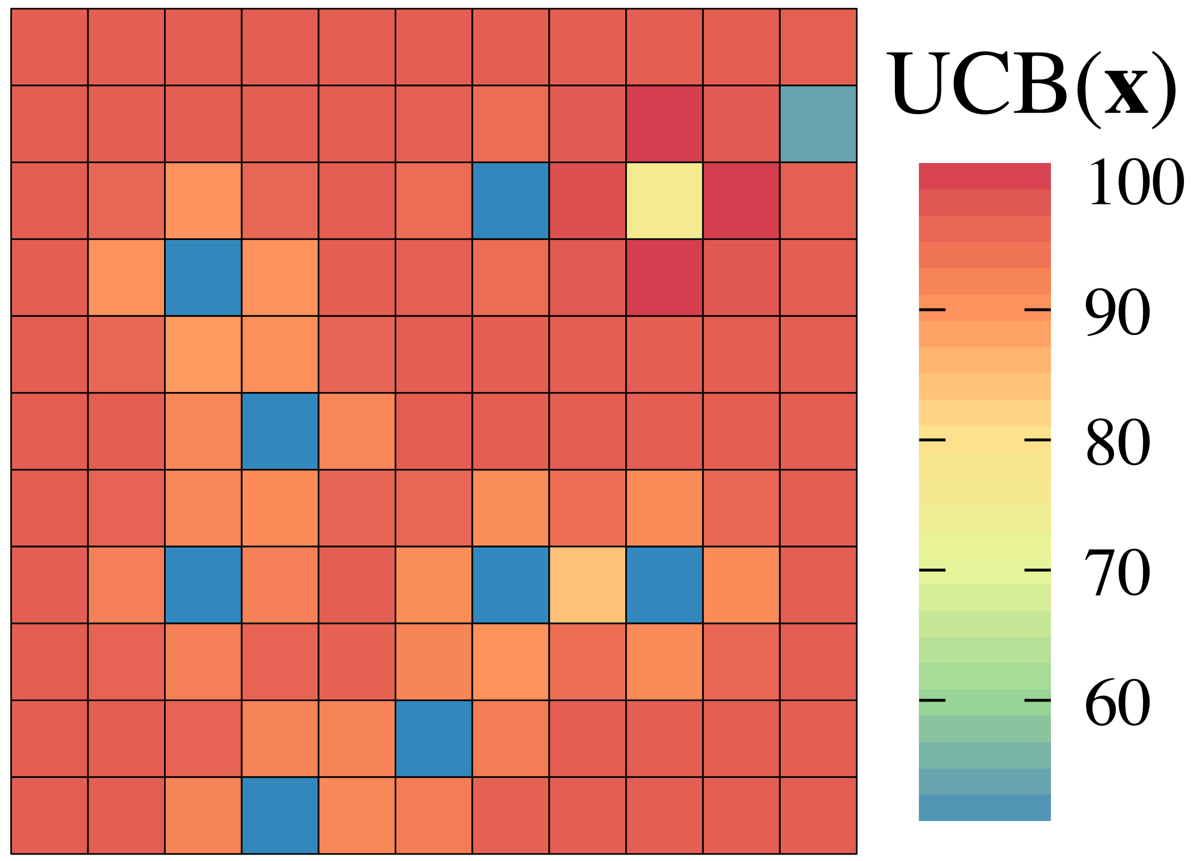
Observations



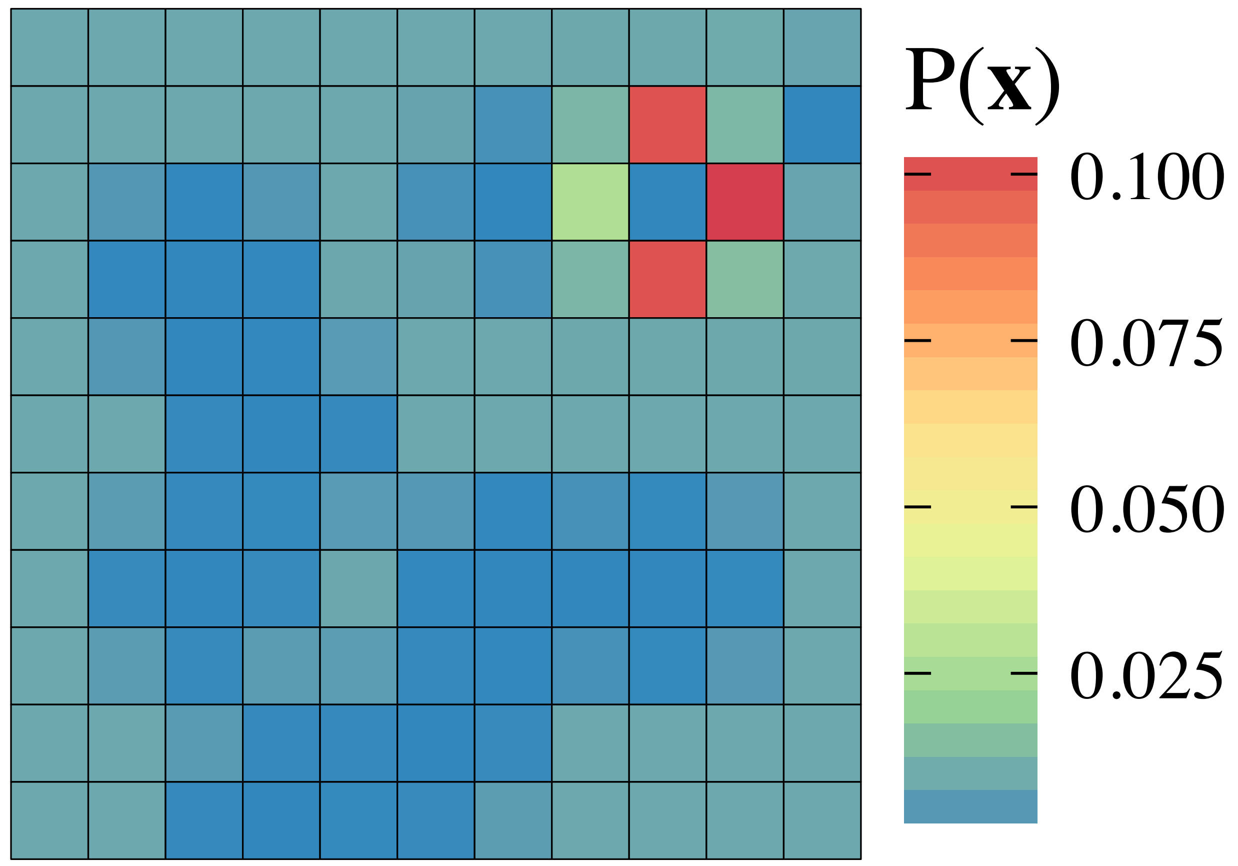
Gaussian Process (GP)



Upper Confidence Bound (UCB) Sampling

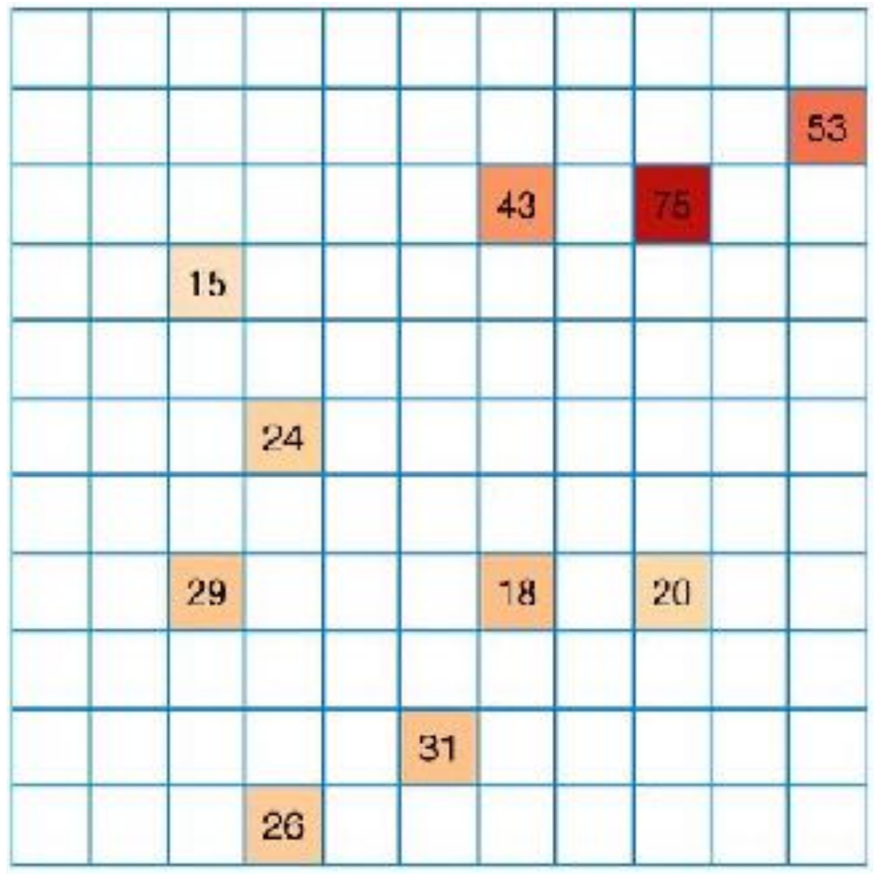


Softmax Choice Rule

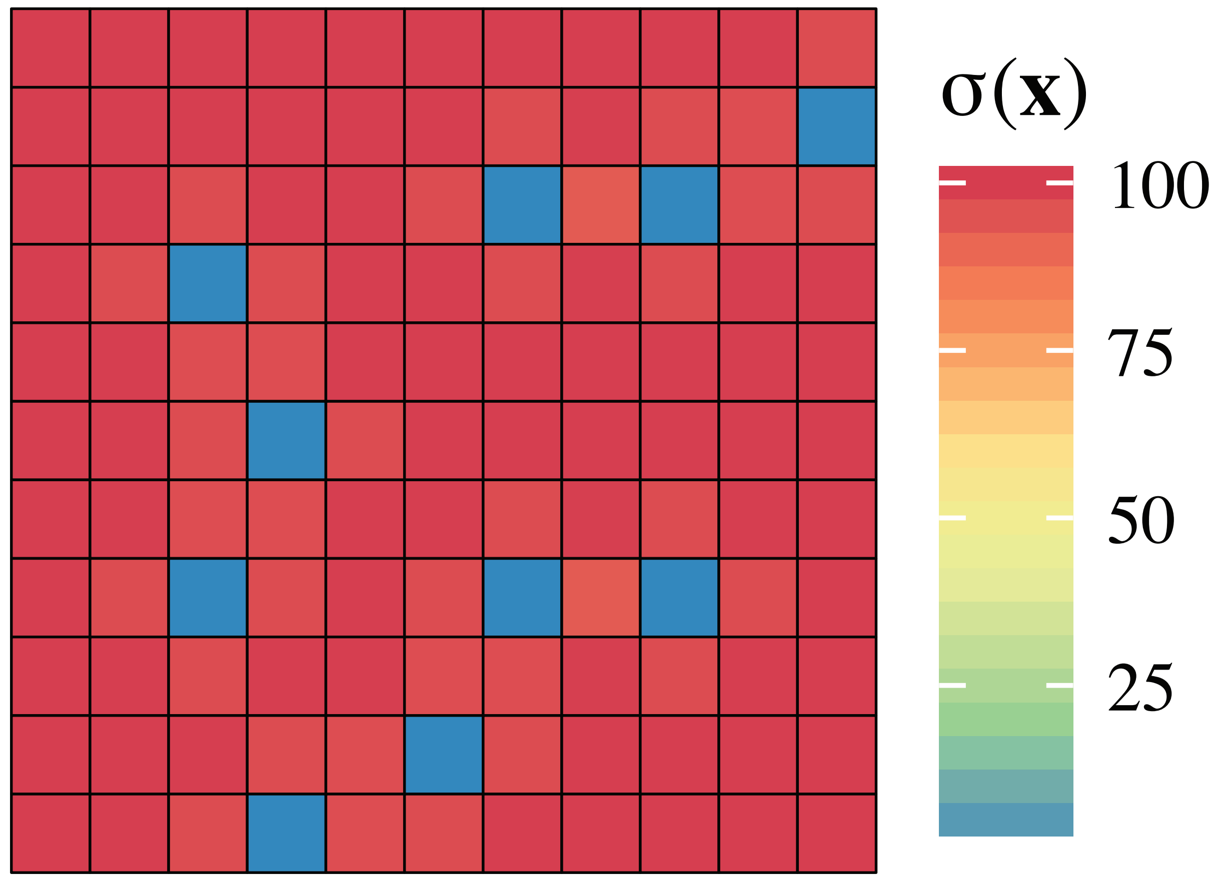


GP-UCB Model

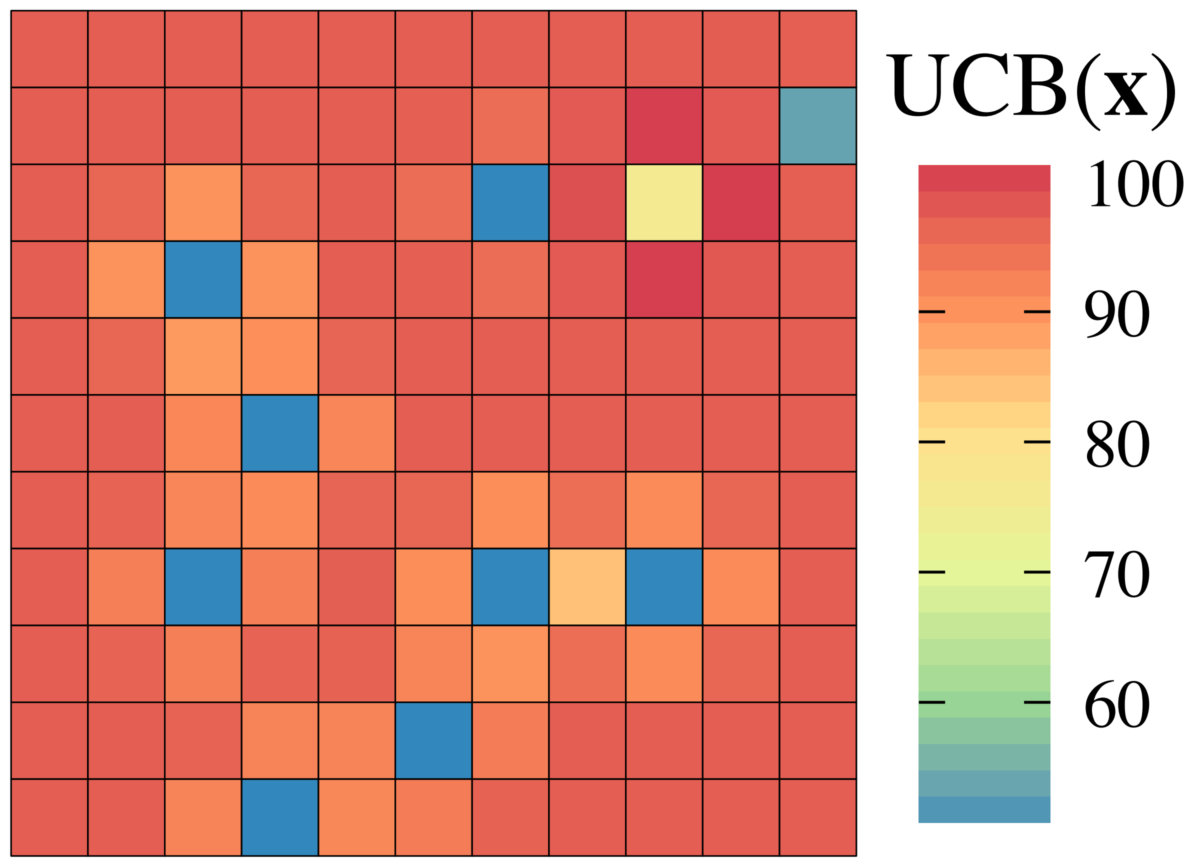
Observations



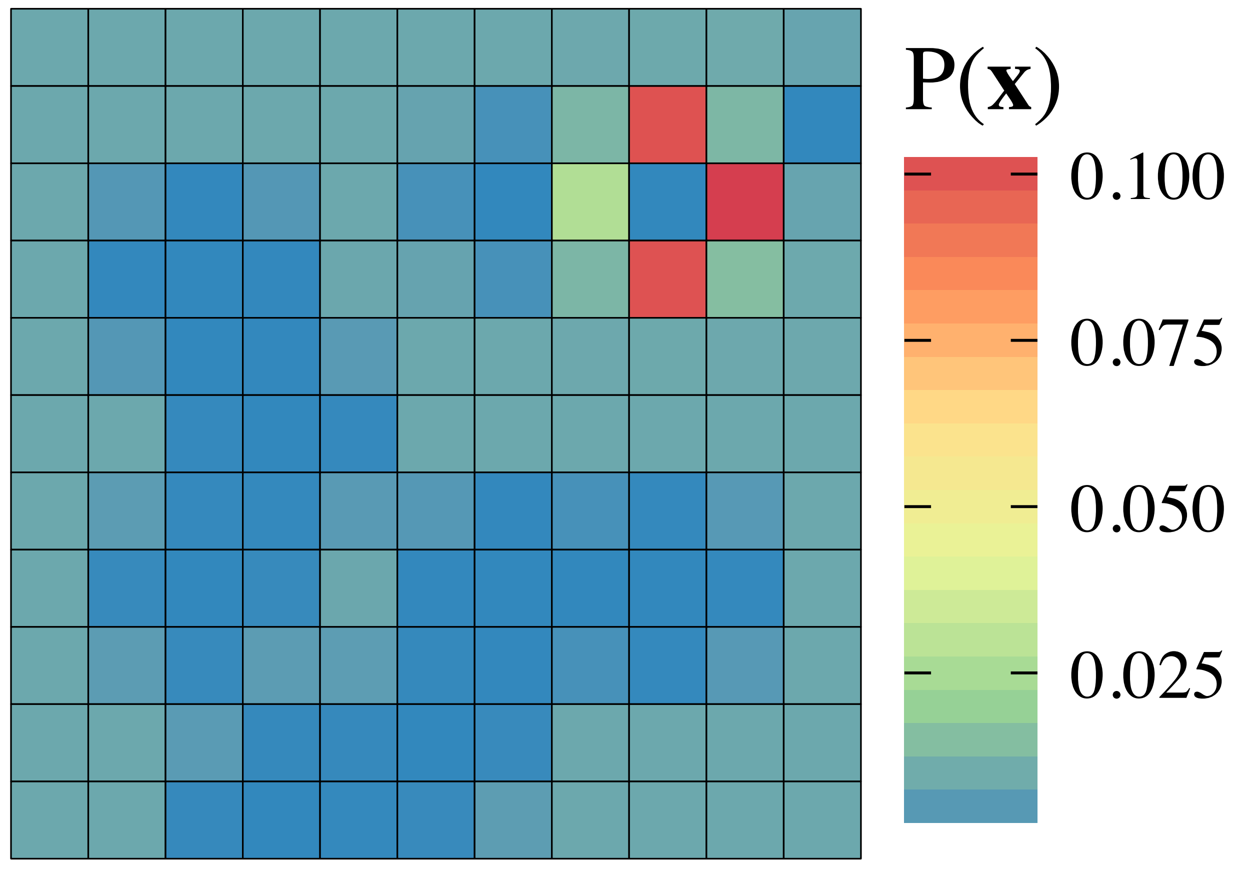
Gaussian Process (GP)



Upper Confidence Bound (UCB) Sampling

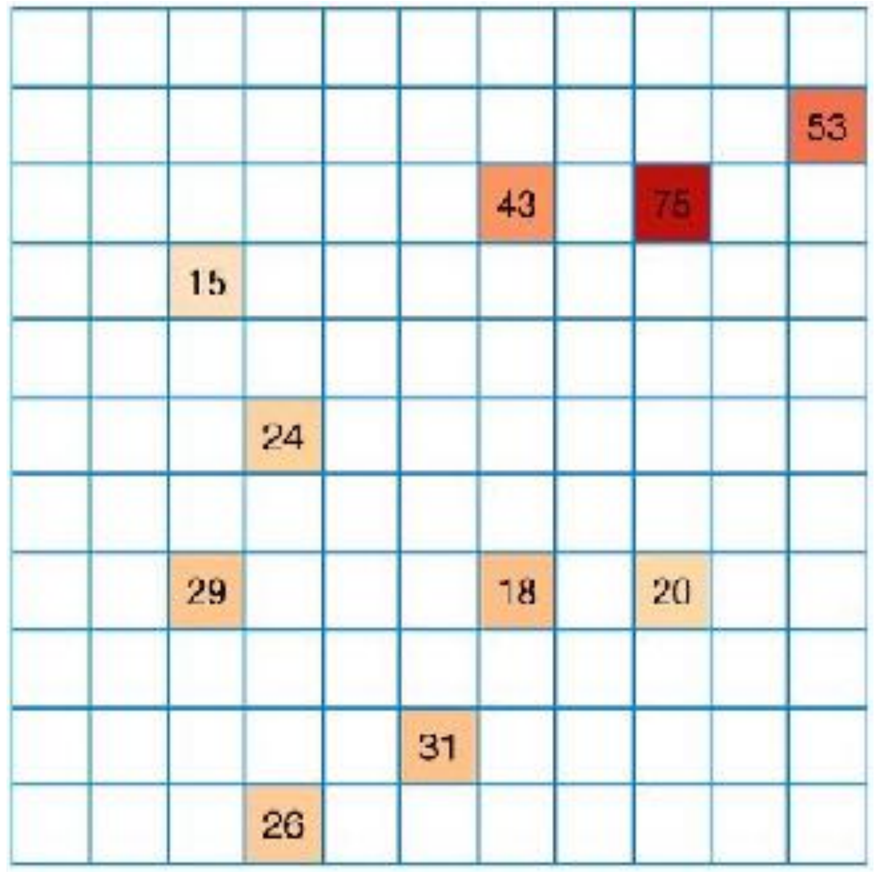


Softmax Choice Rule

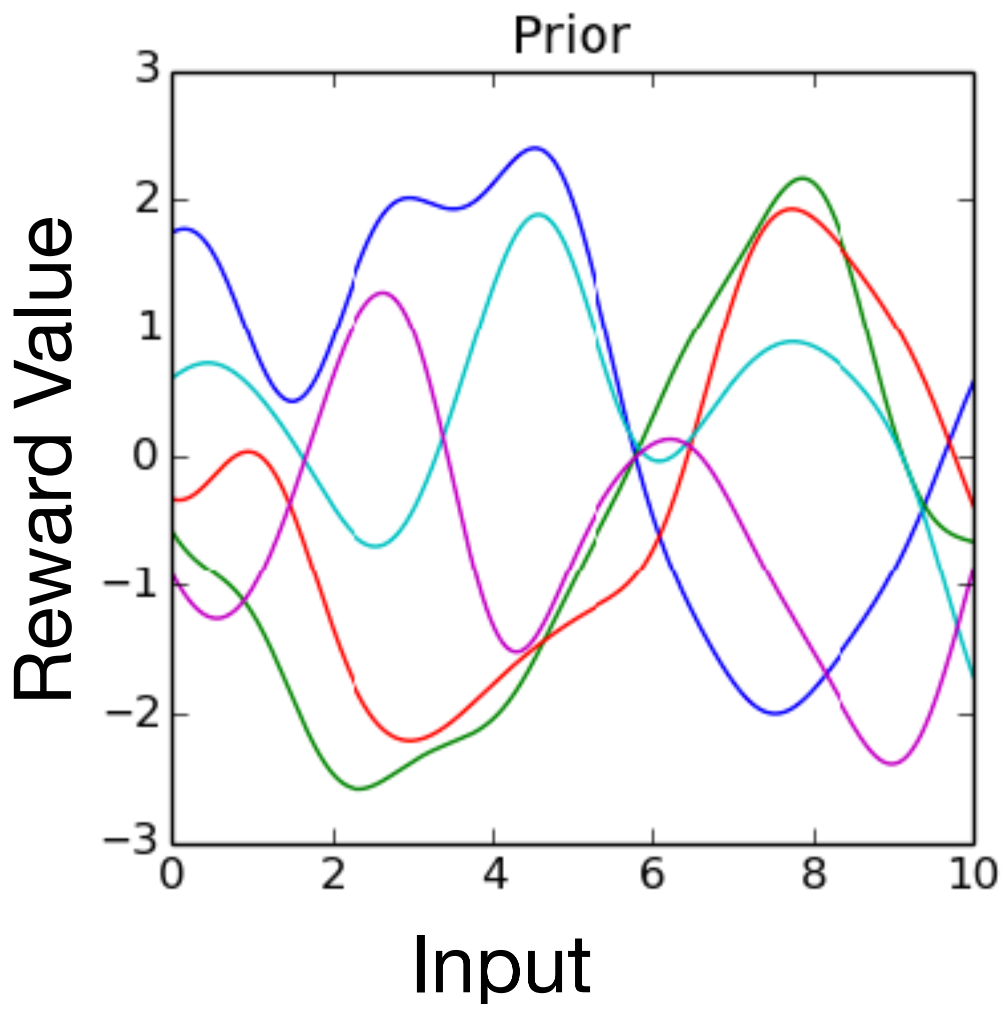


GP-UCB Model

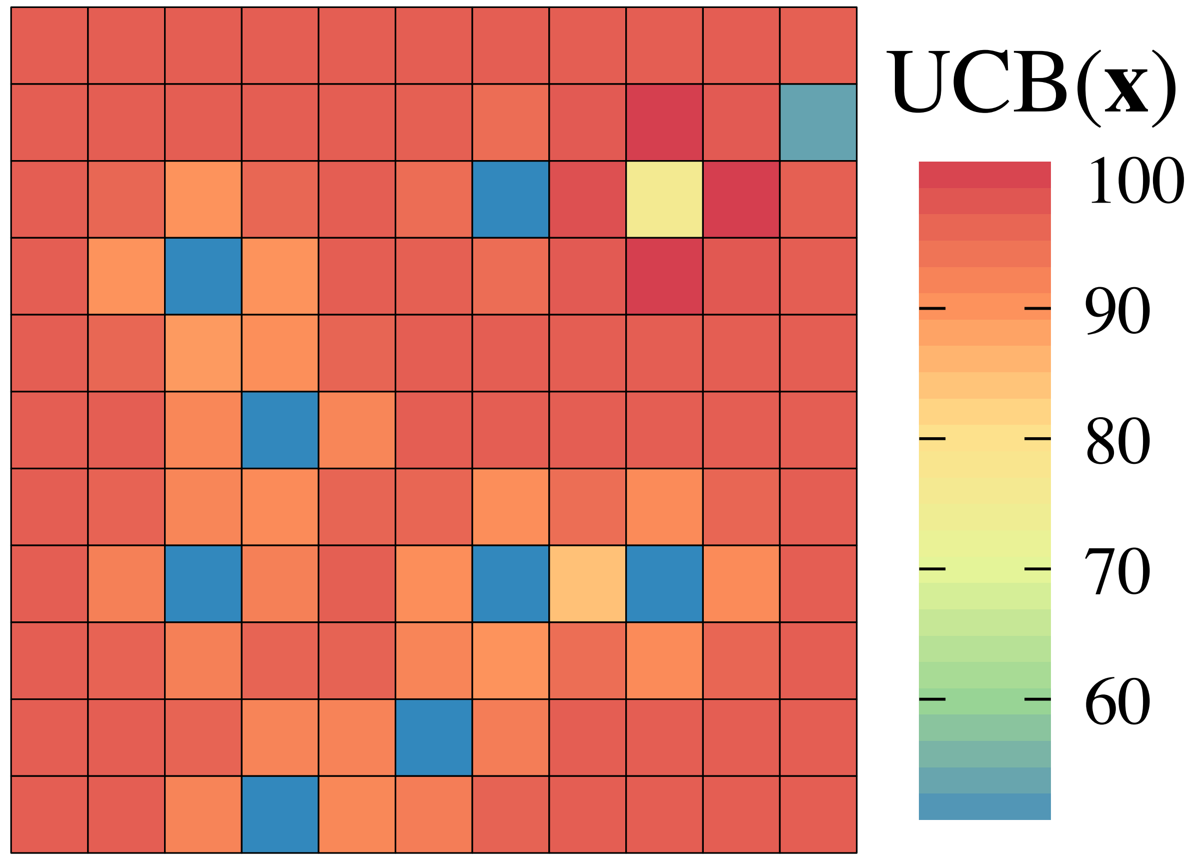
Observations



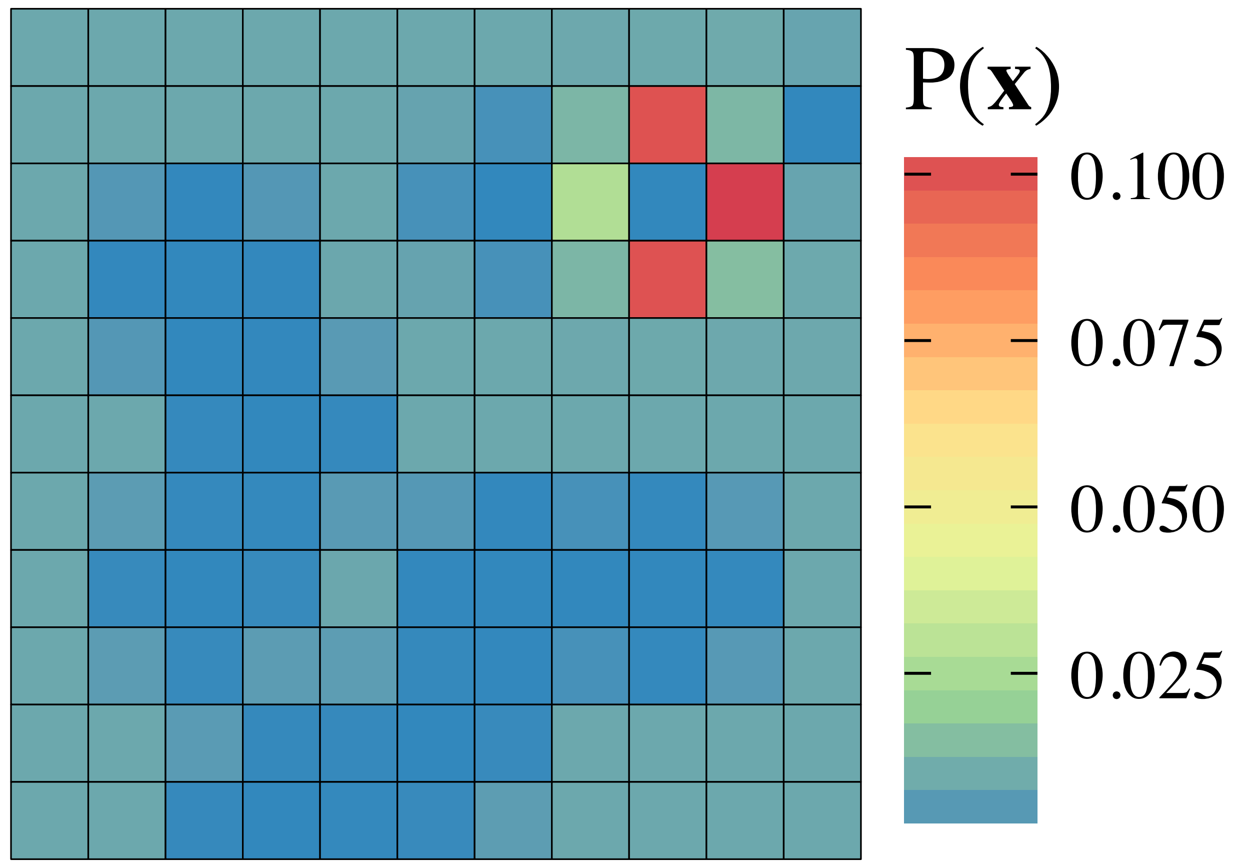
Gaussian Process (GP)



Upper Confidence Bound (UCB) Sampling

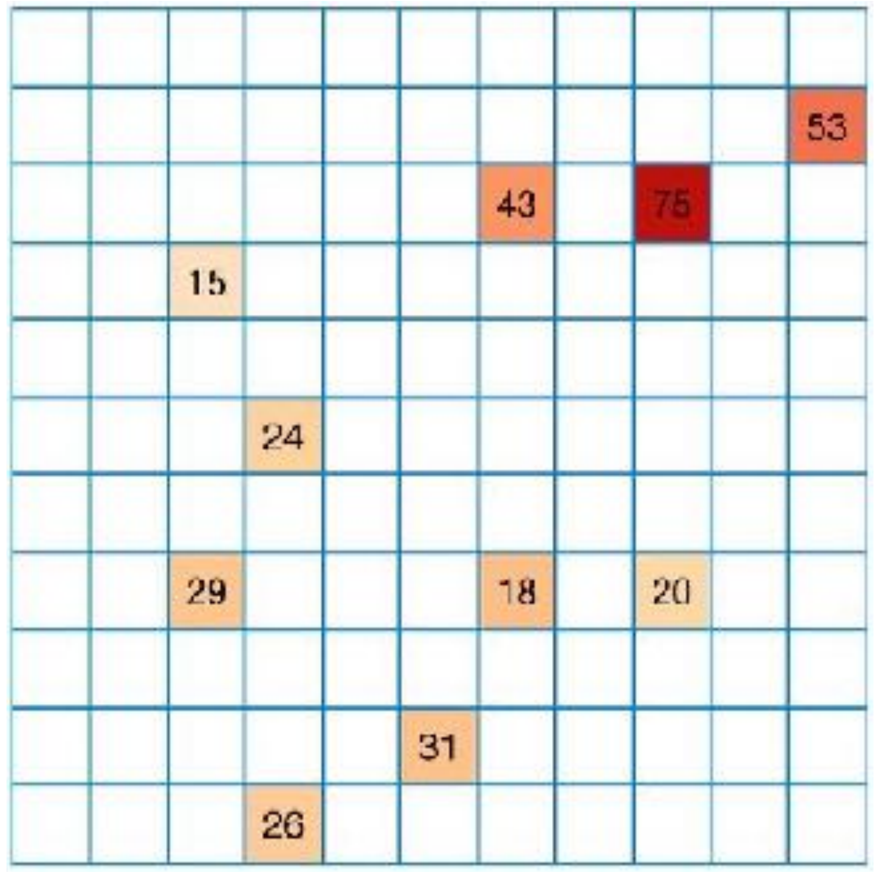


Softmax Choice Rule

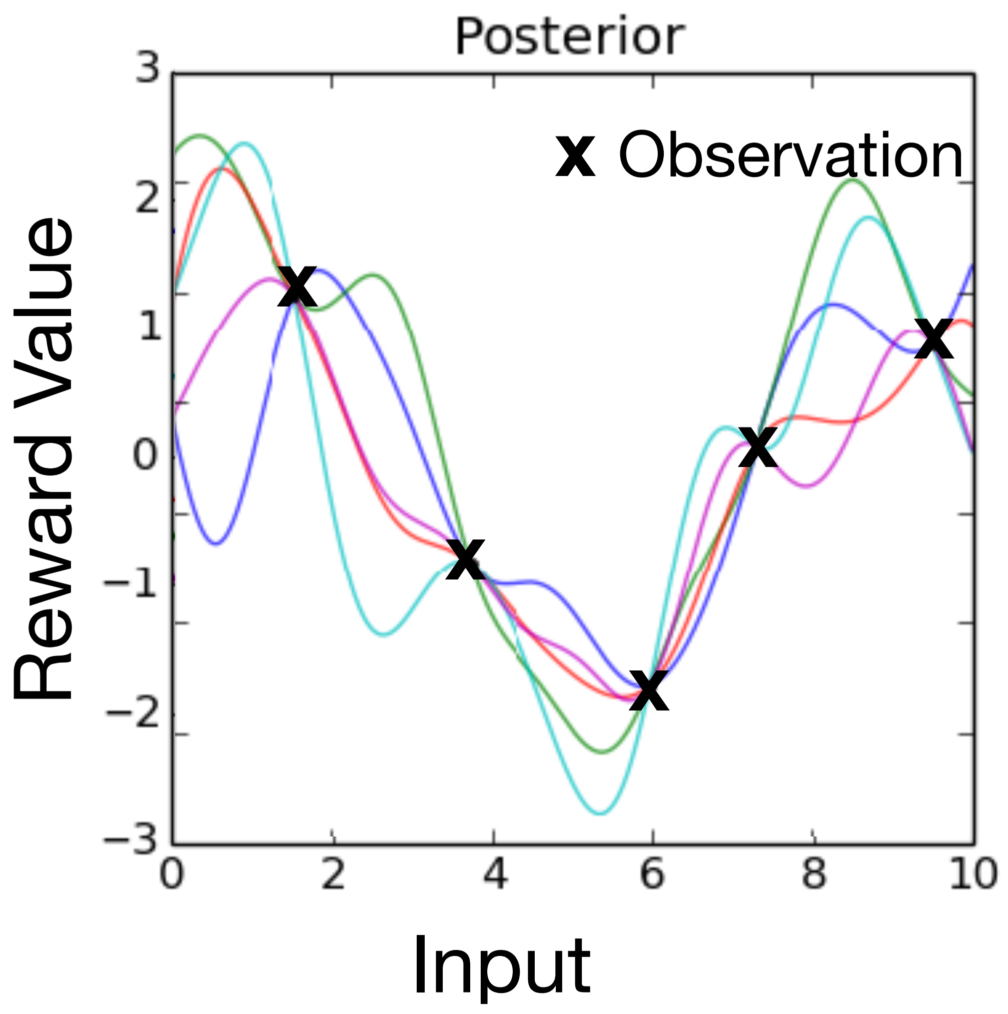


GP-UCB Model

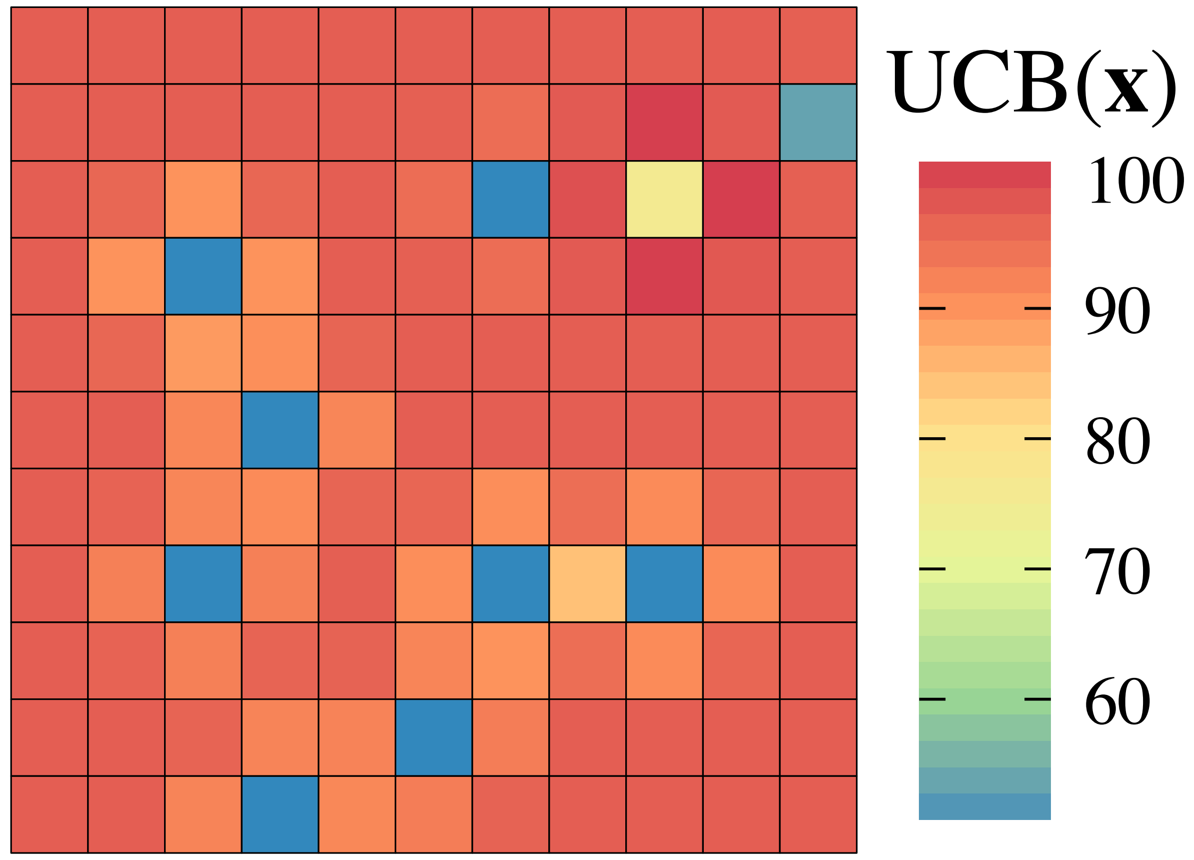
Observations



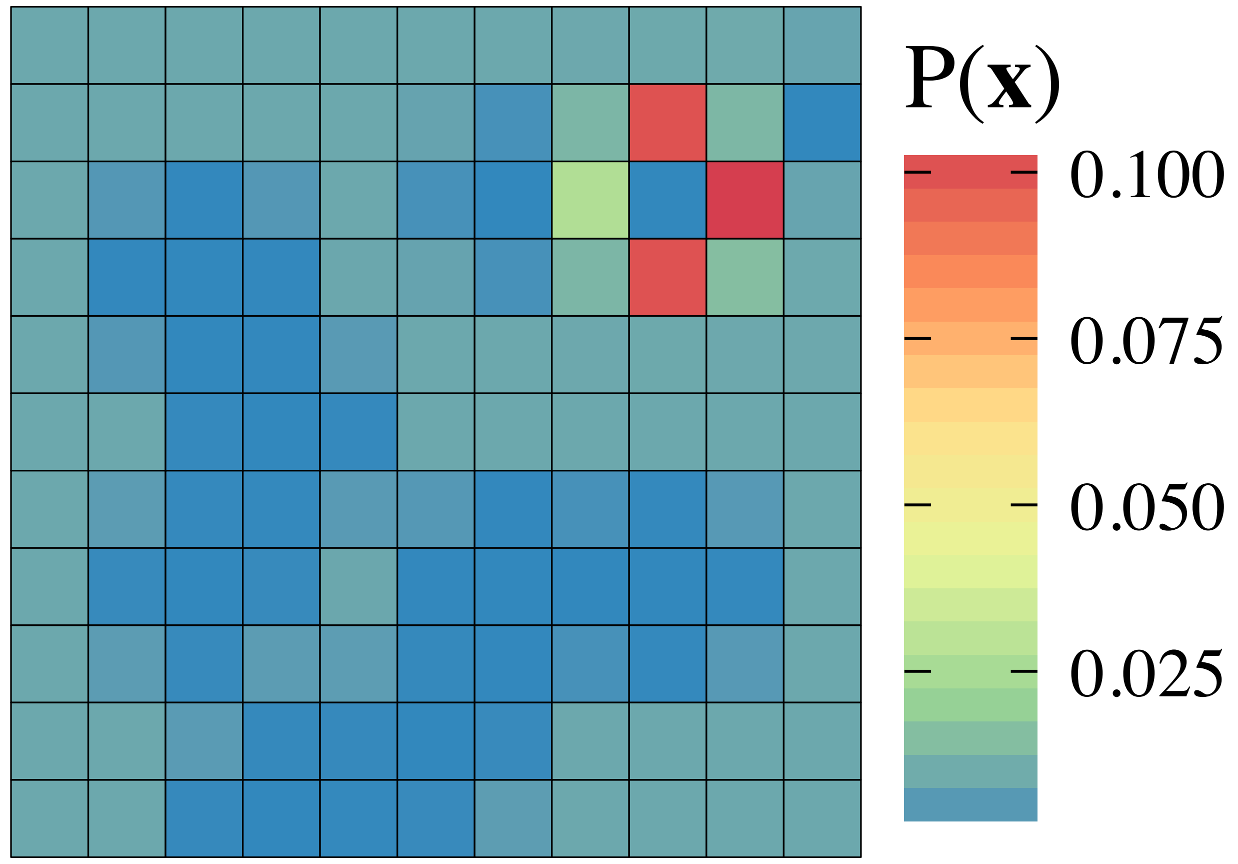
Gaussian Process (GP)



Upper Confidence Bound (UCB) Sampling

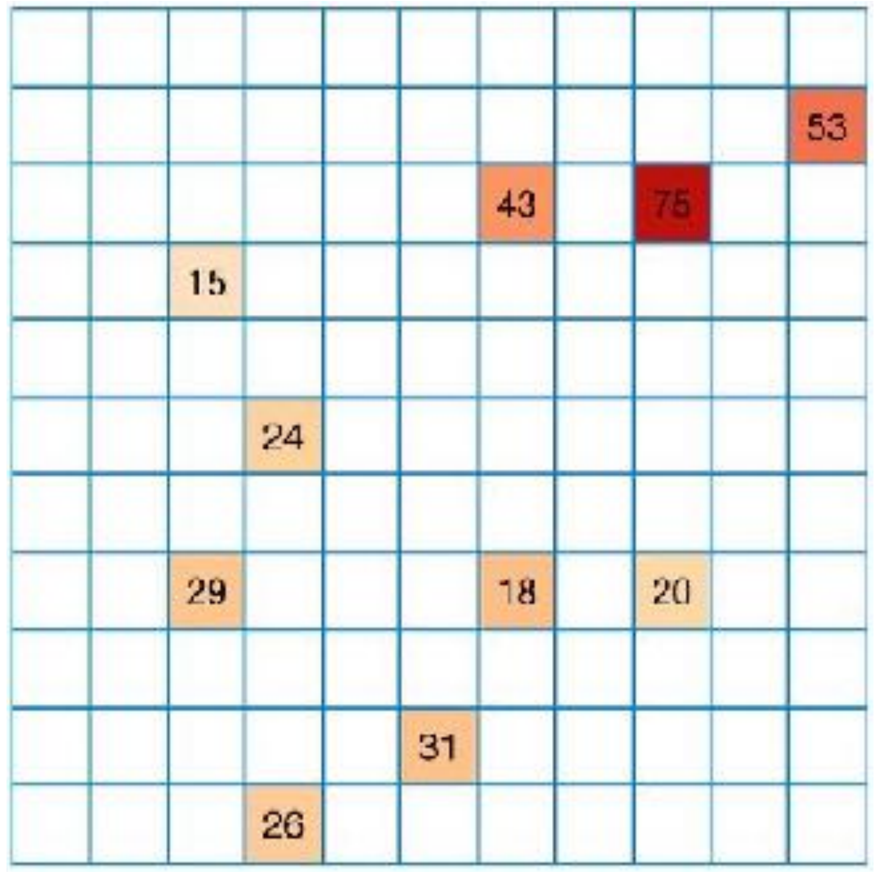


Softmax Choice Rule

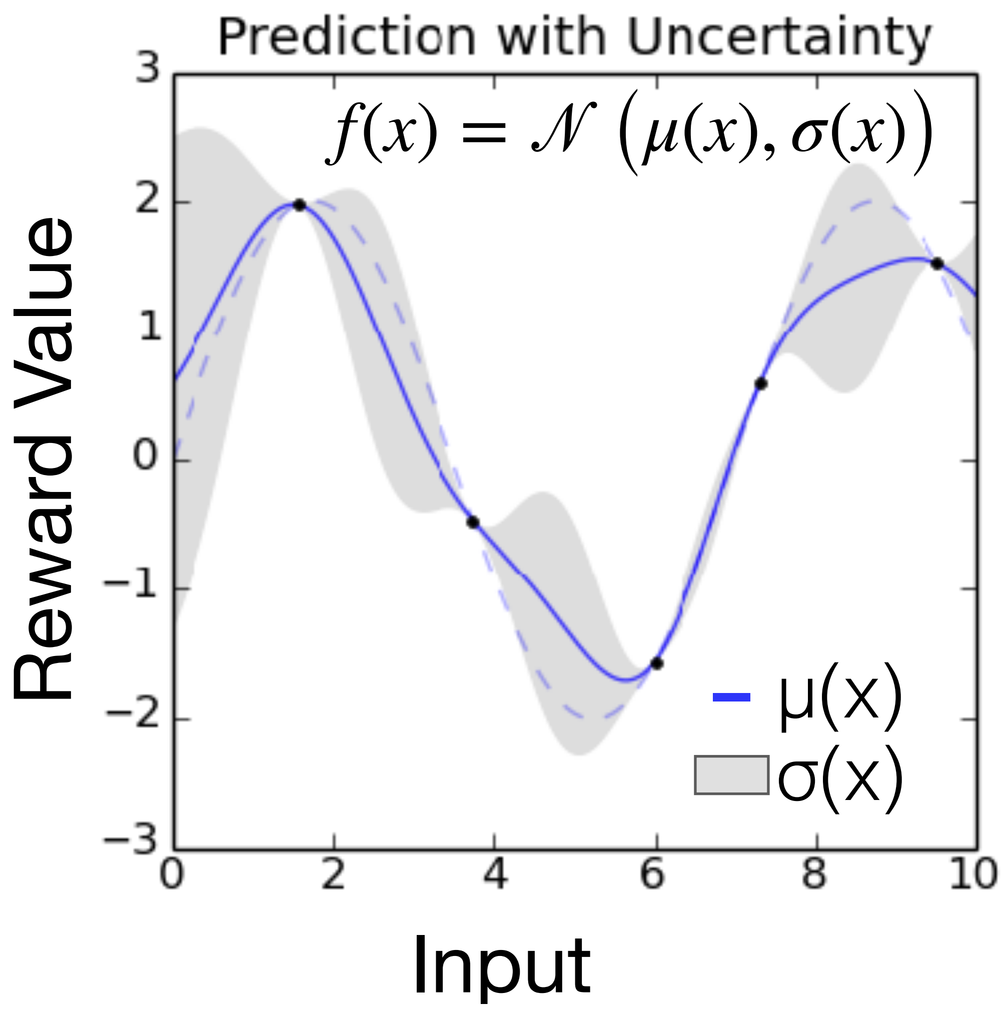


GP-UCB Model

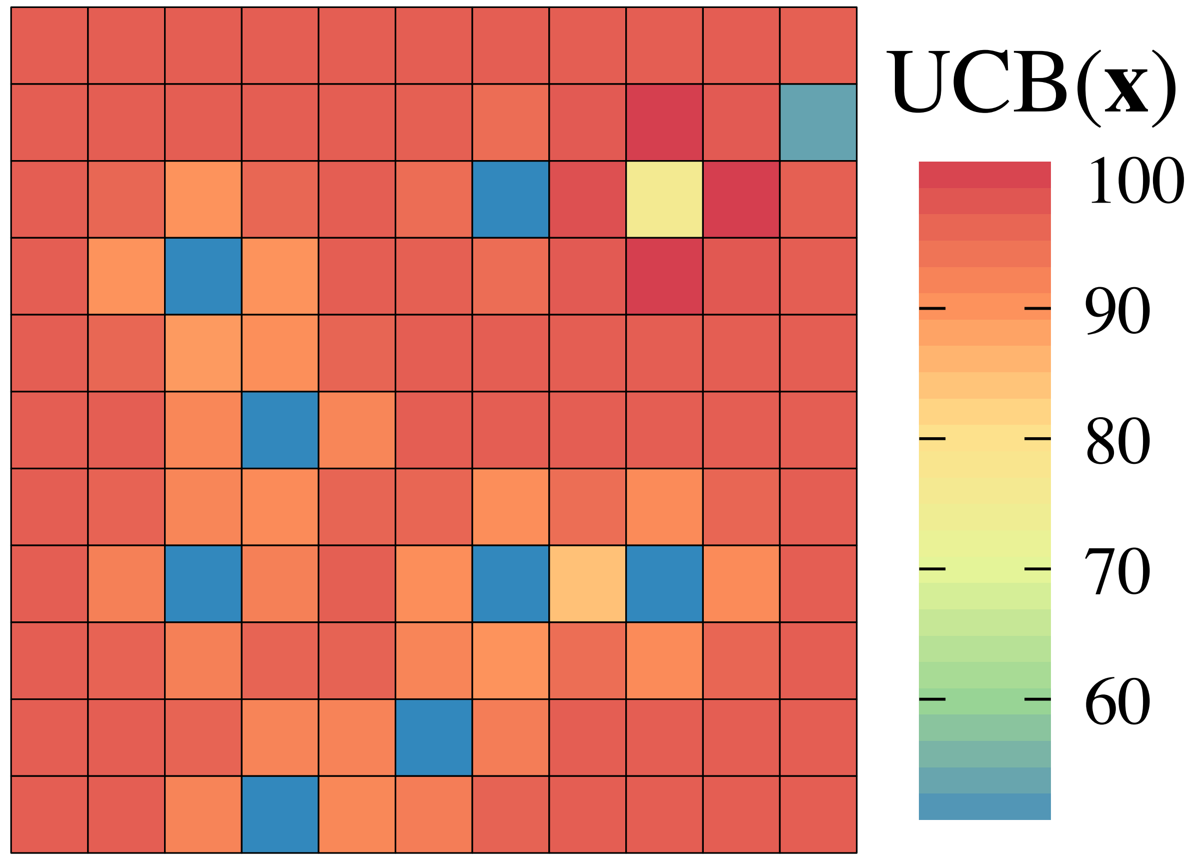
Observations



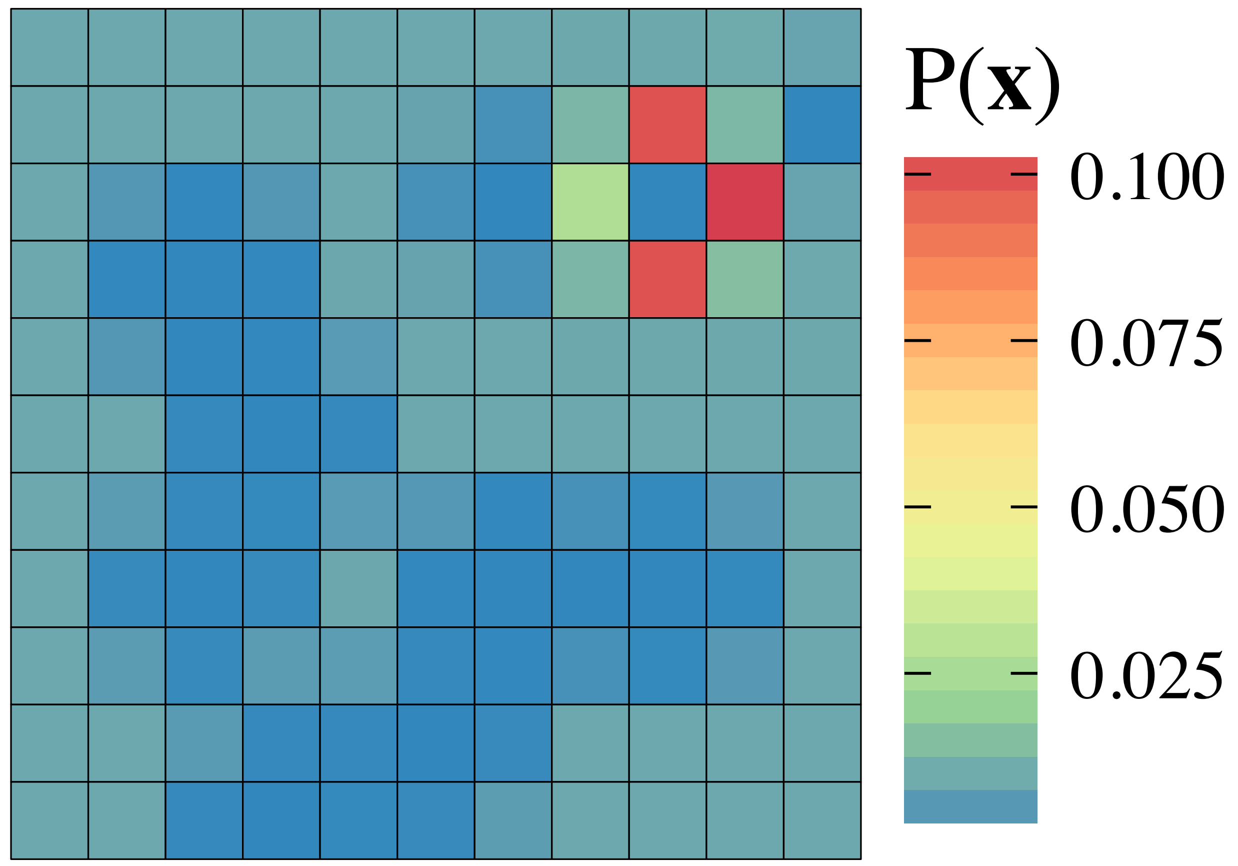
Gaussian Process (GP)



Upper Confidence Bound (UCB) Sampling

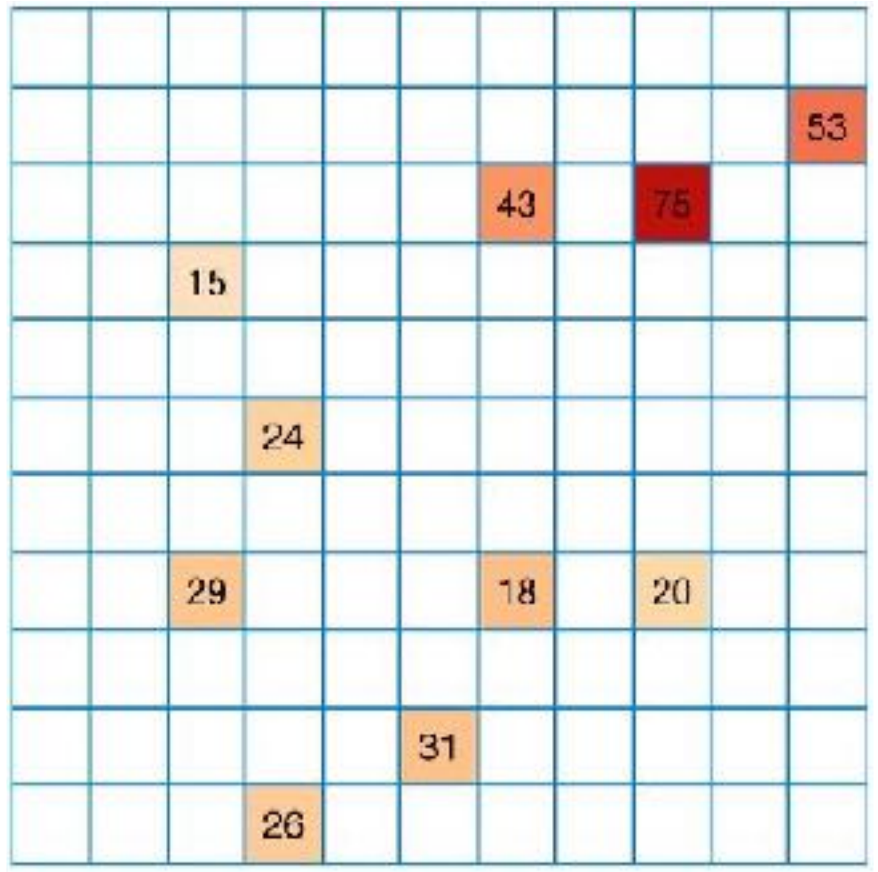


Softmax Choice Rule

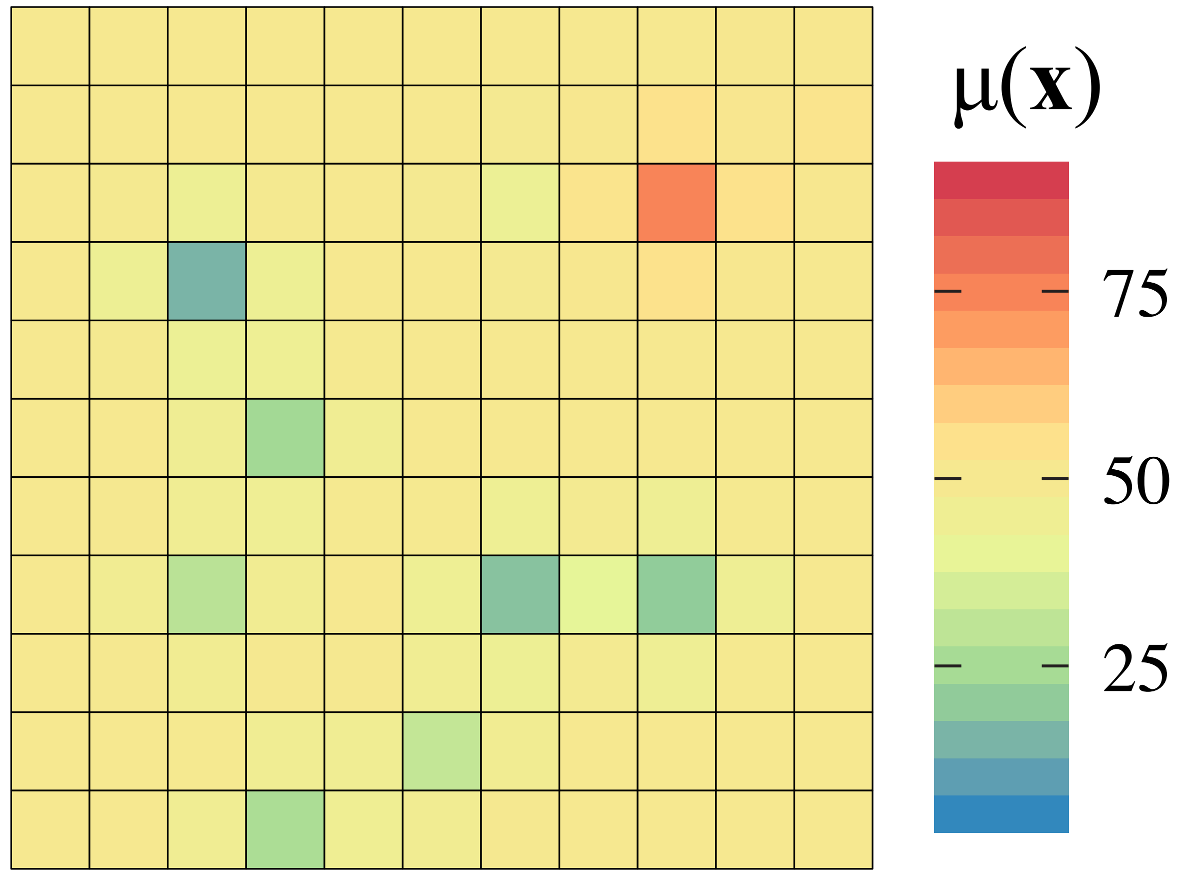


GP-UCB Model

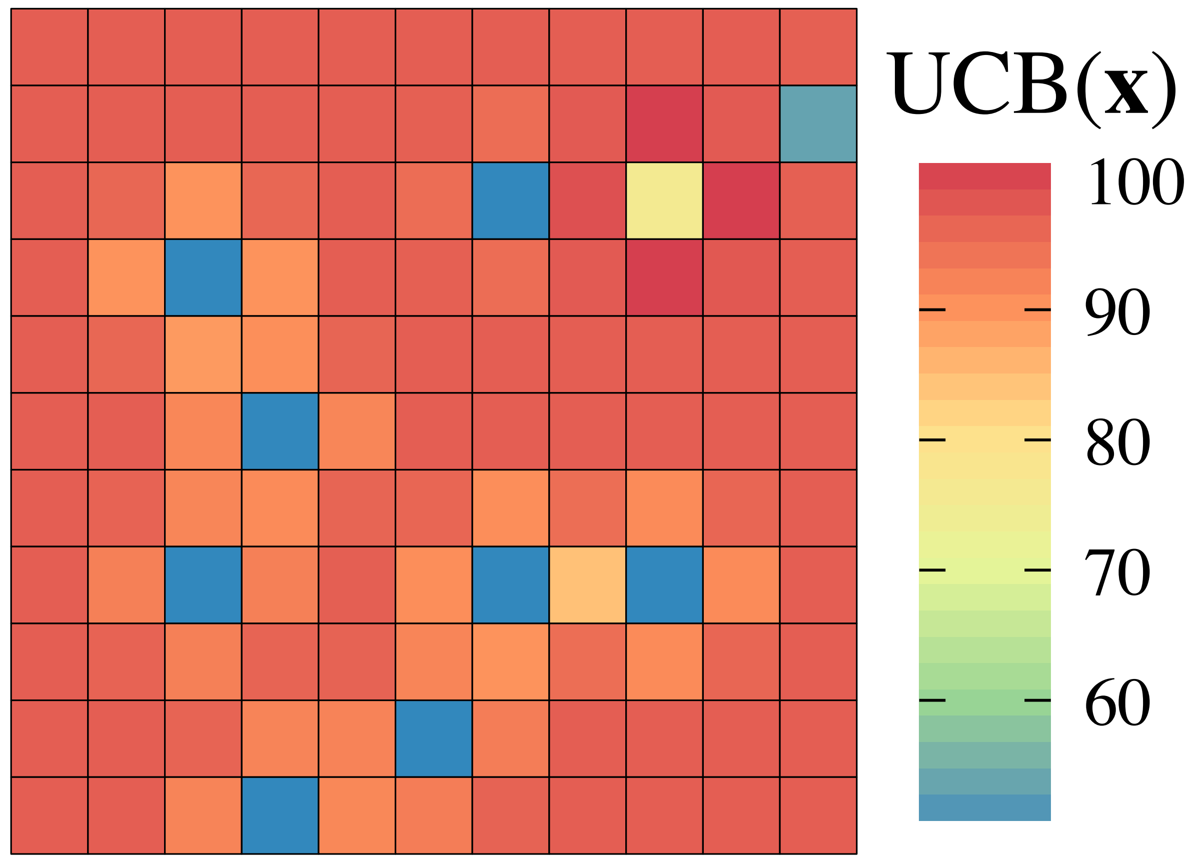
Observations



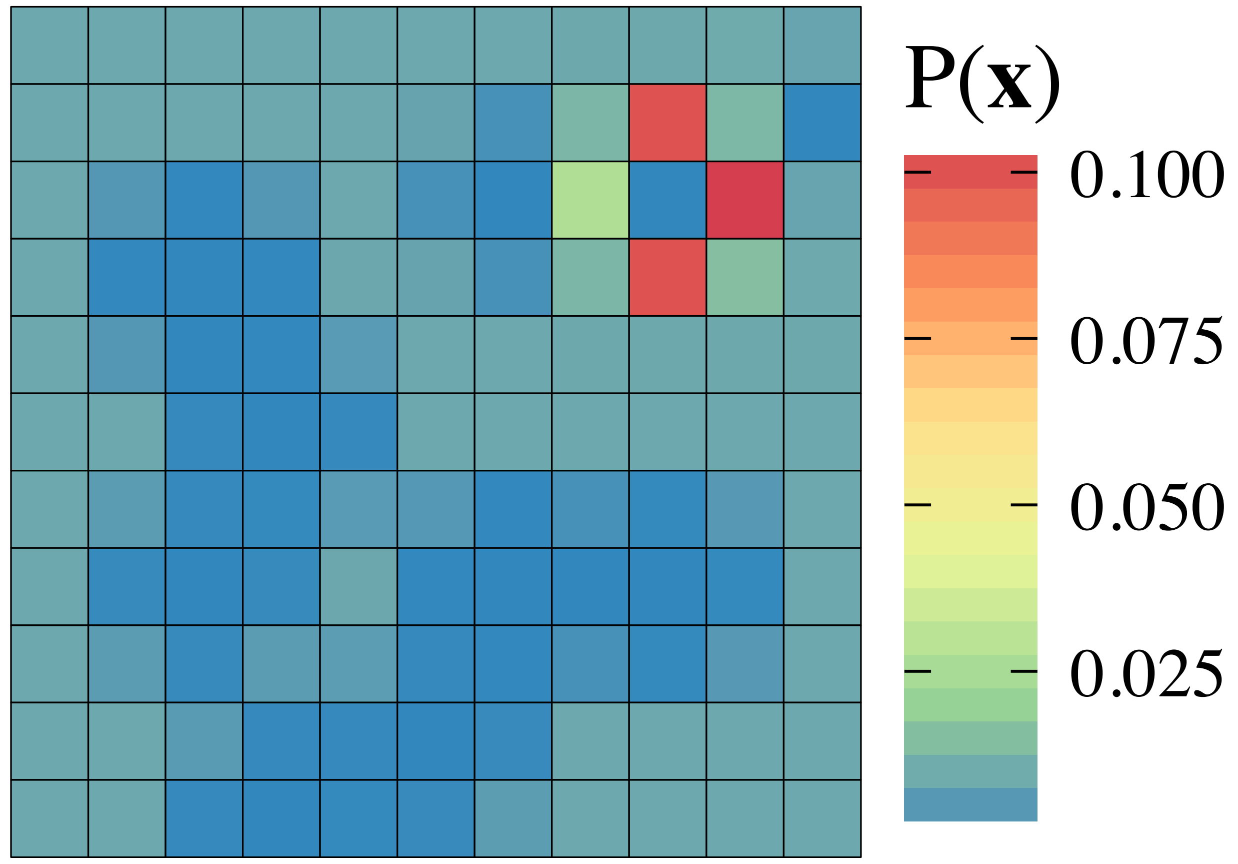
Gaussian Process (GP)



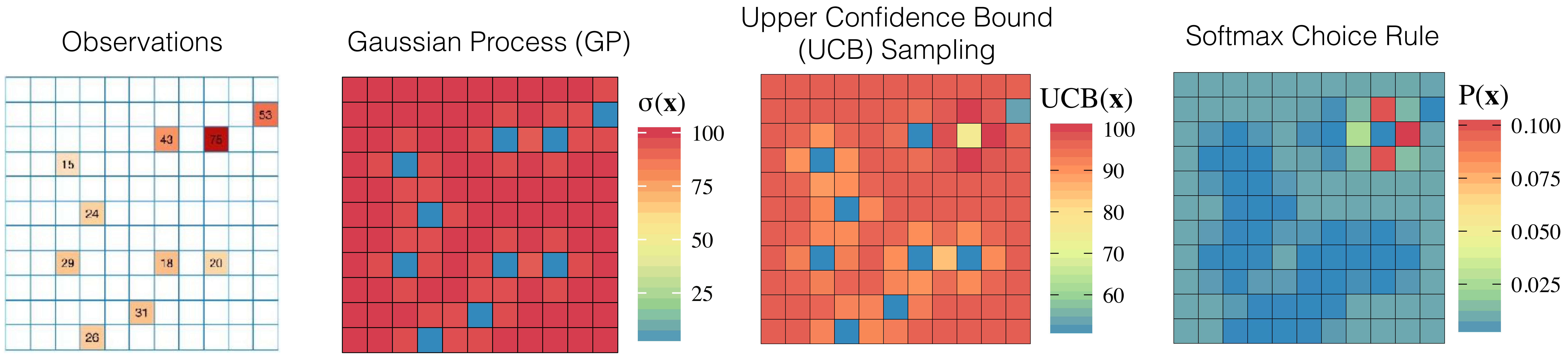
Upper Confidence Bound (UCB) Sampling



Softmax Choice Rule

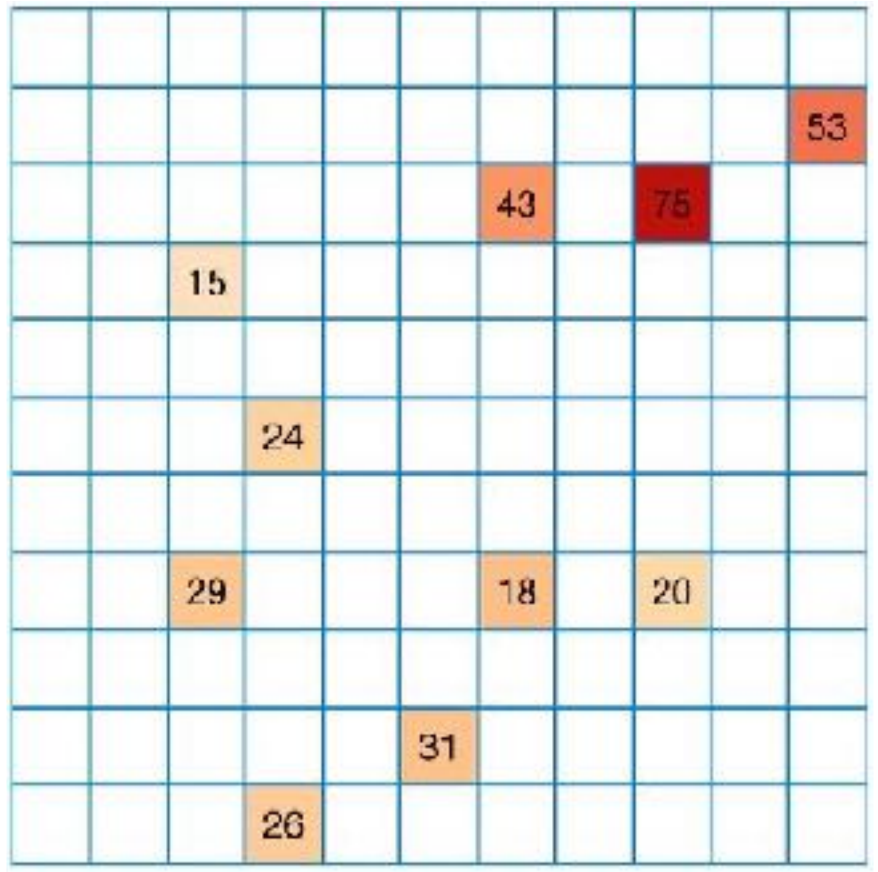


GP-UCB Model

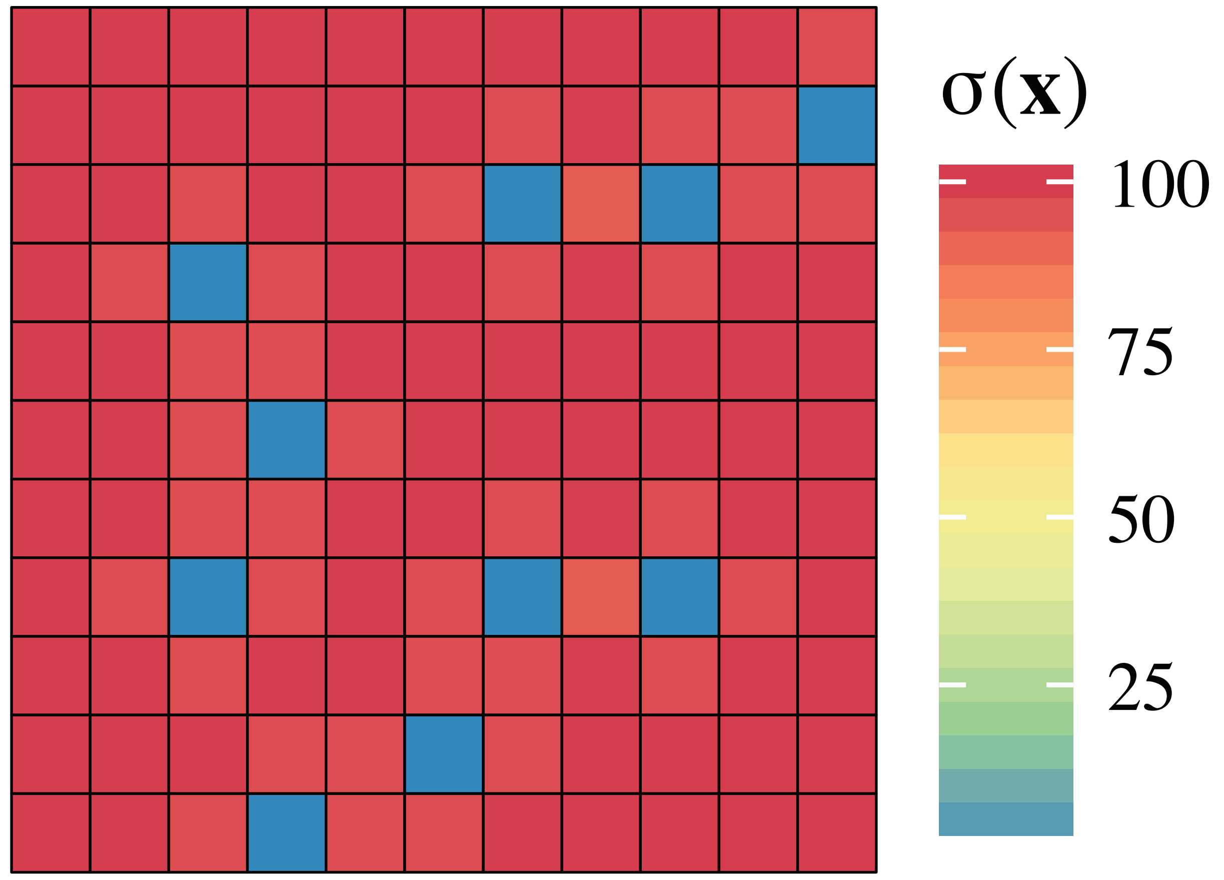

Observations
53
43
75
15
24
29
18
20
31
26
Gaussian Process (GP)
σ(x)
100
75
50
25
Upper Confidence Bound (UCB) Sampling
UCB(x)
100
90
80
70
60
Softmax Choice Rule
P(x)
0.100
0.075
0.050
0.025

GP-UCB Model

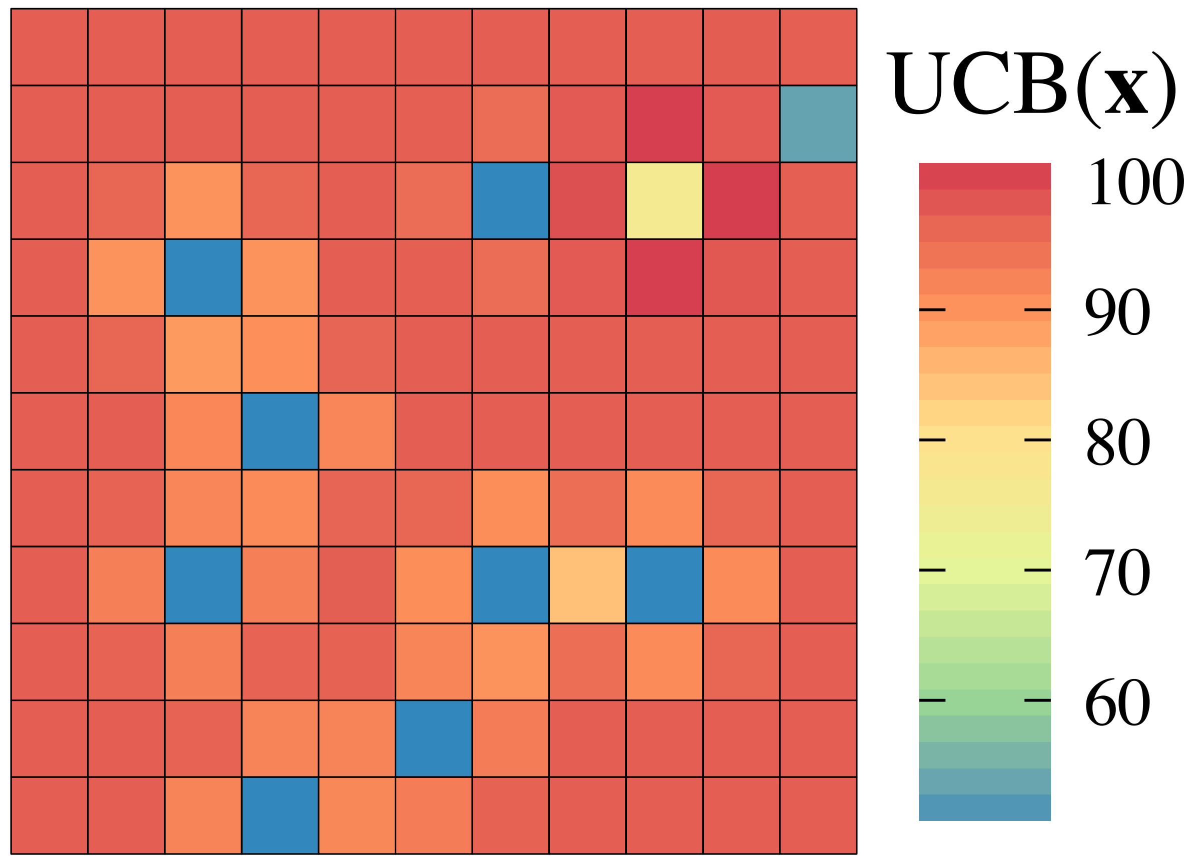
Observations



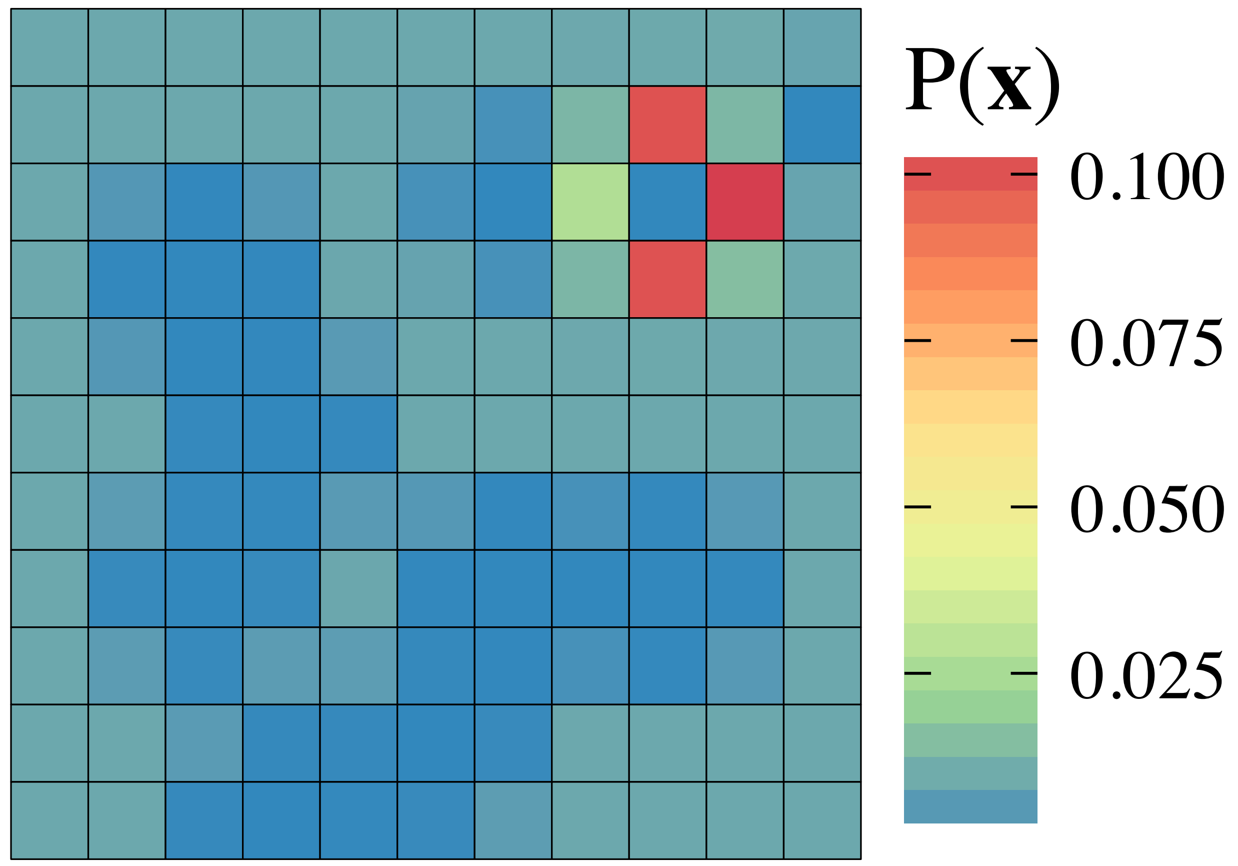
Gaussian Process (GP)



Upper Confidence Bound (UCB) Sampling

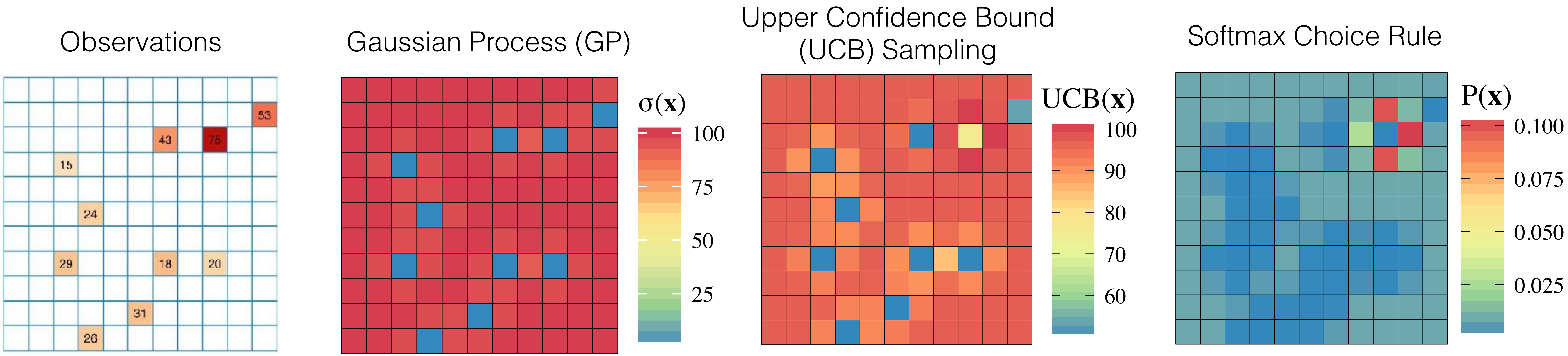


Softmax Choice Rule

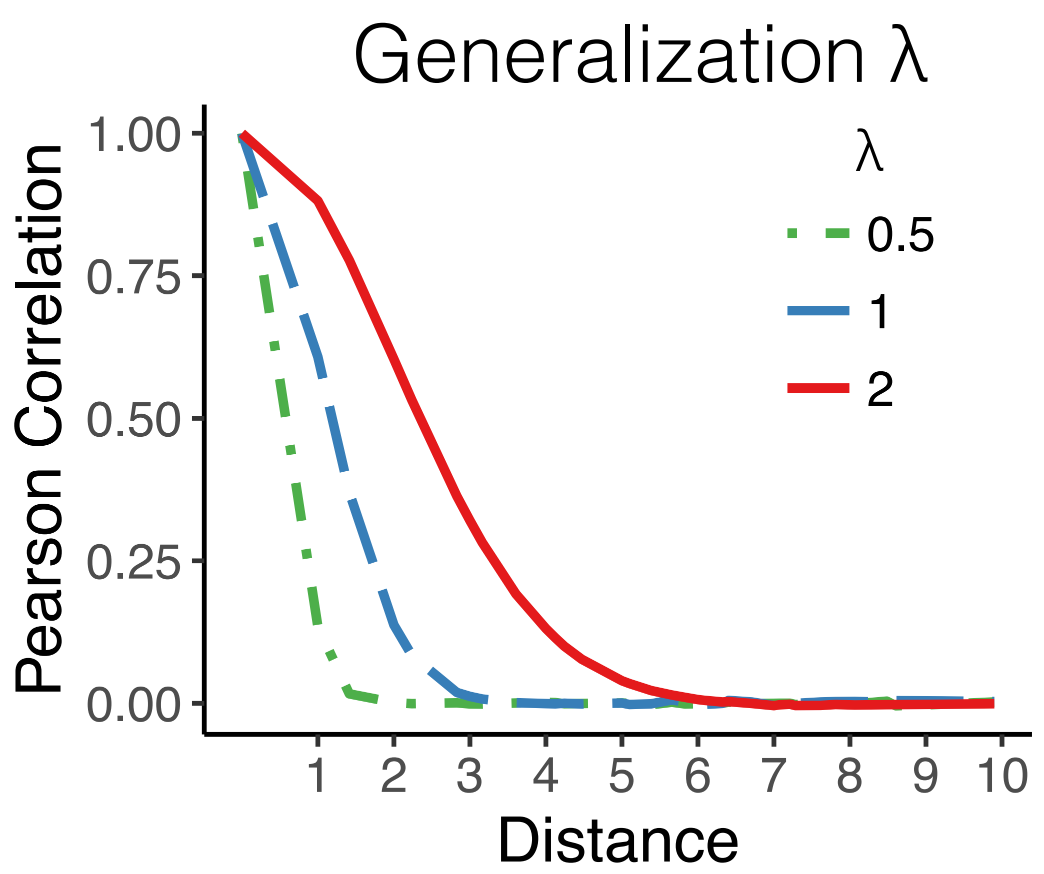


$$k_{RBF}(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{||\mathbf{x} - \mathbf{x}'||^2}{2\lambda^2}\right)$$

GP-UCB Model

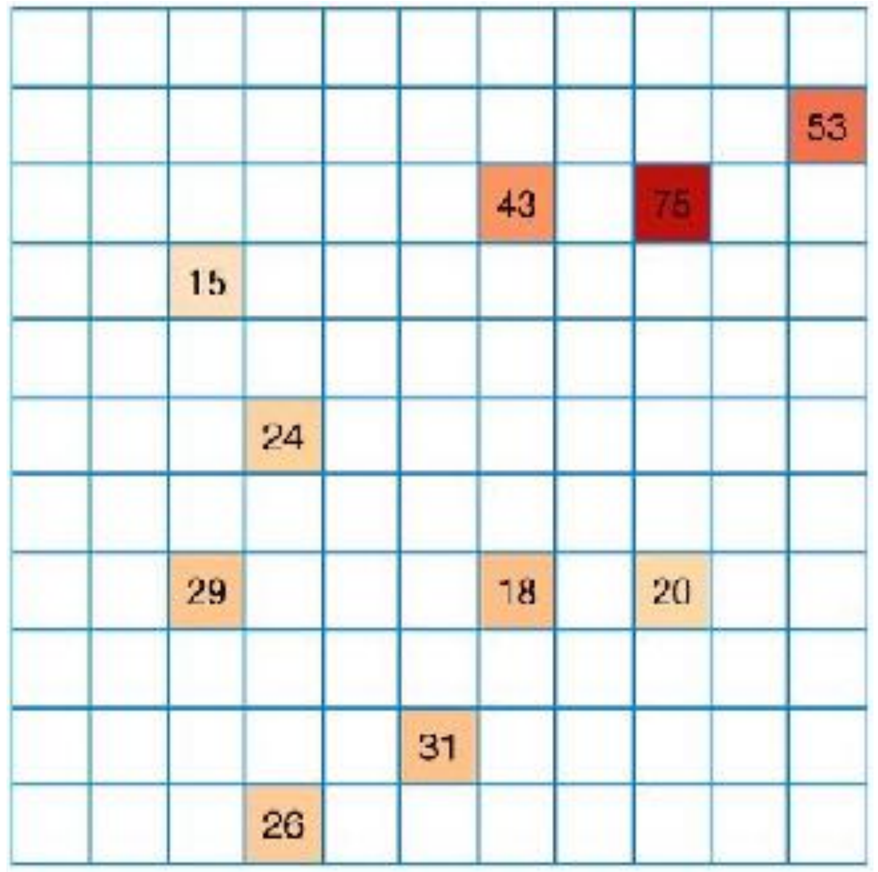


$$k_{RBF}(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{||\mathbf{x}-\mathbf{x}'||^2}{2\lambda^2}\right)$$

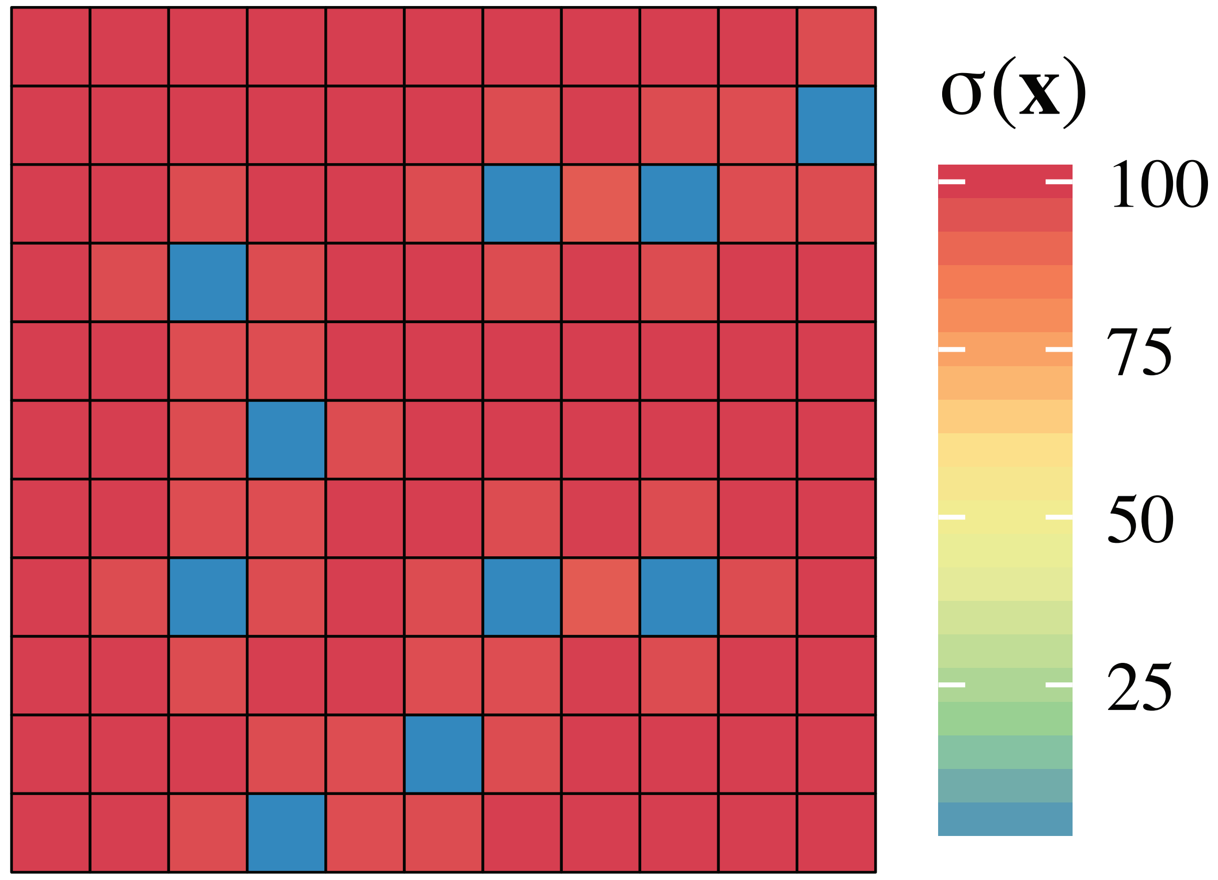


GP-UCB Model

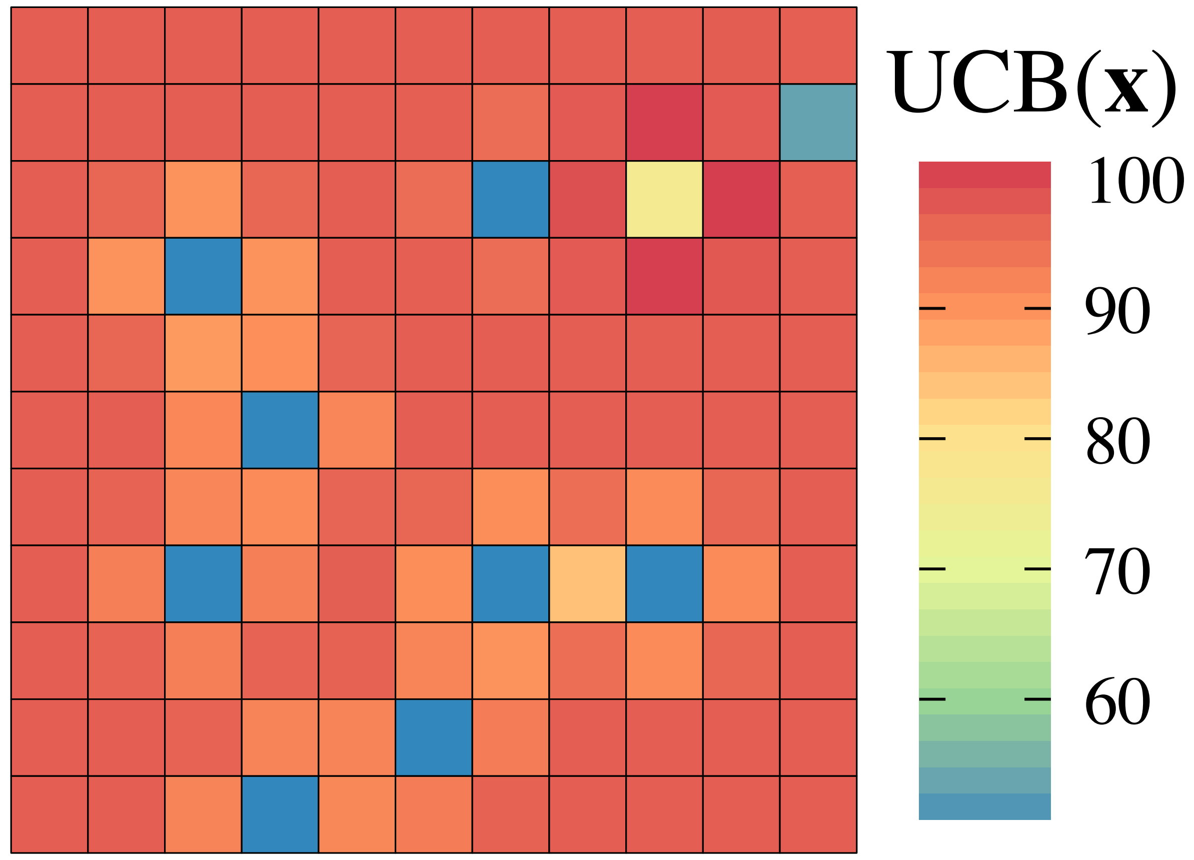
Observations



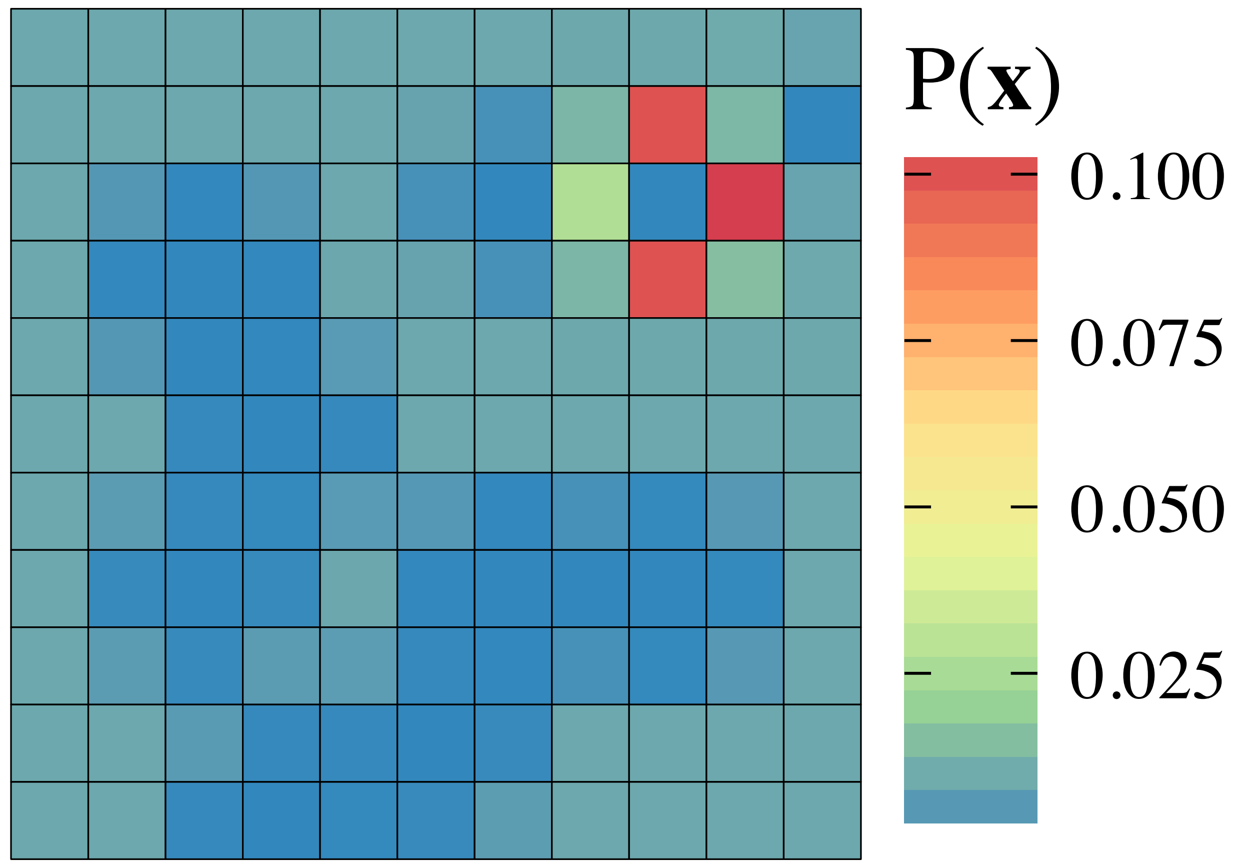
Gaussian Process (GP)

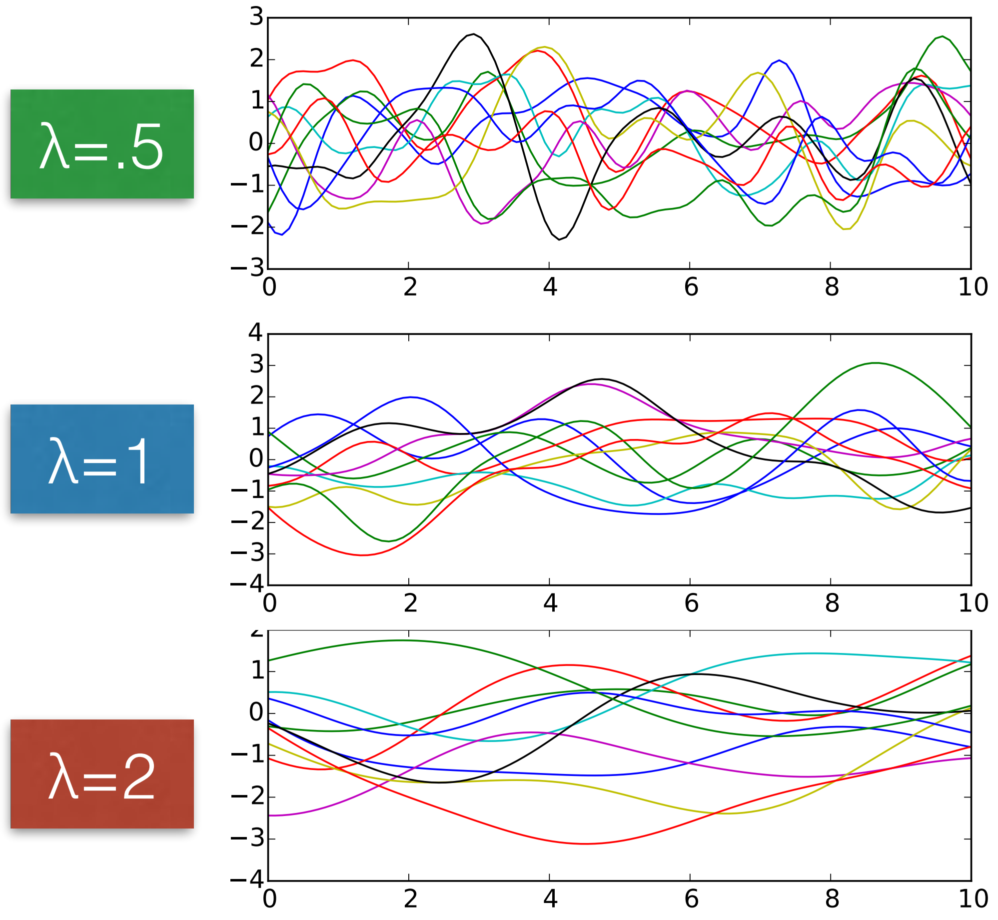


Upper Confidence Bound (UCB) Sampling



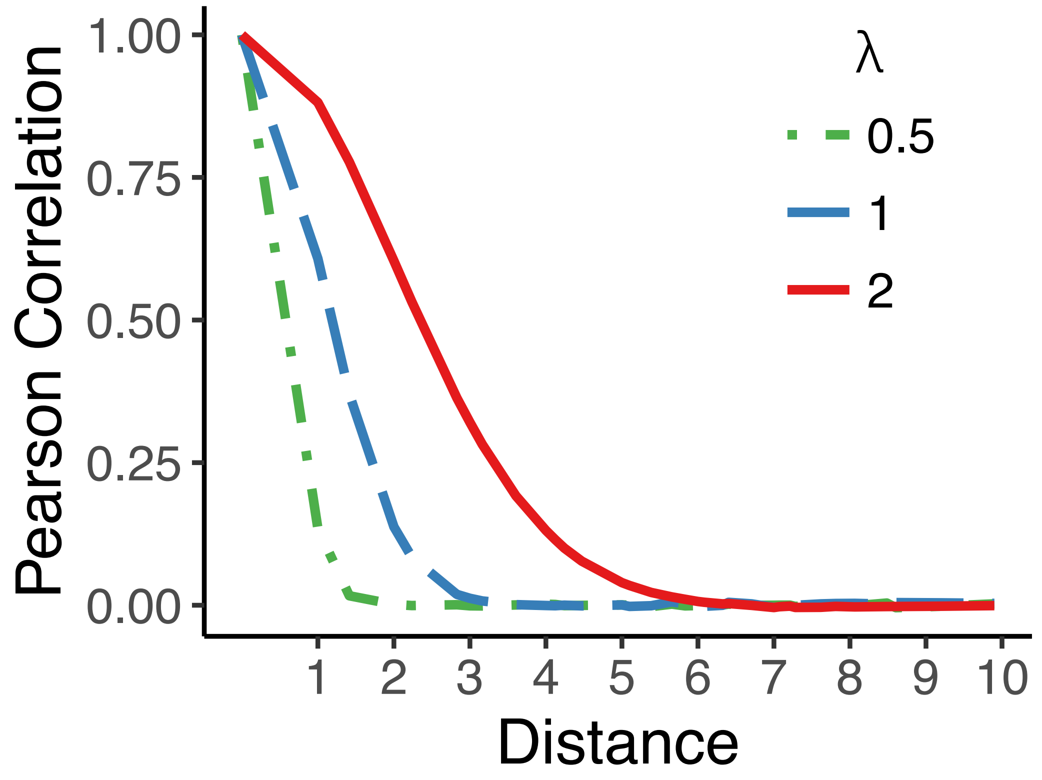
Softmax Choice Rule





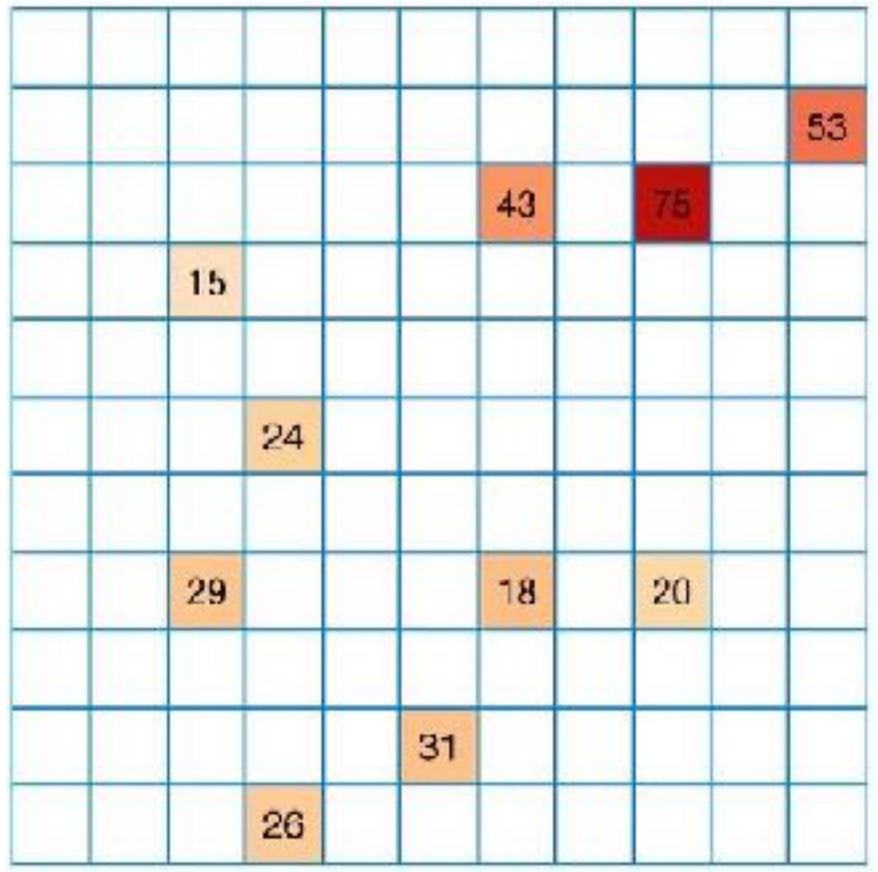
$$k_{RBF}(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{||\mathbf{x} - \mathbf{x}'||^2}{2\lambda^2}\right)$$

Generalization λ

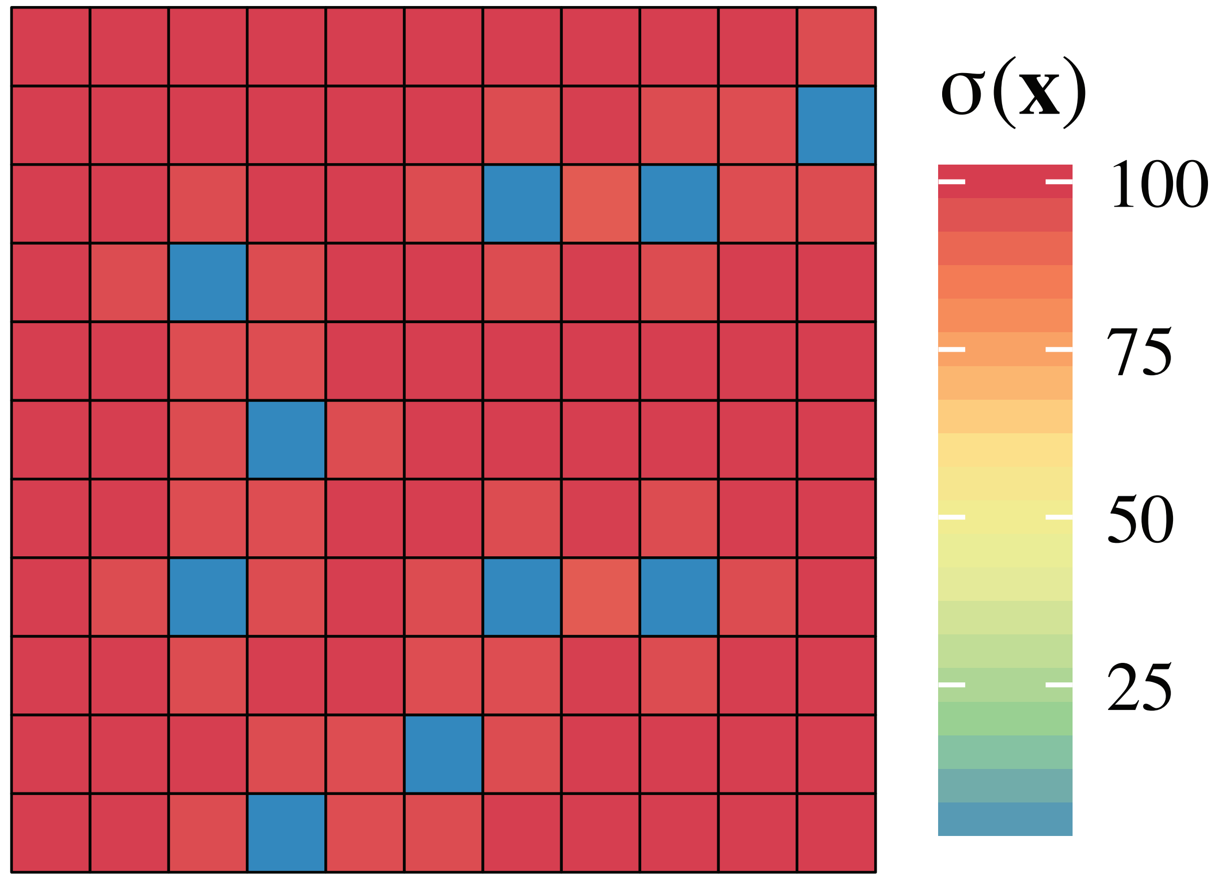


GP-UCB Model

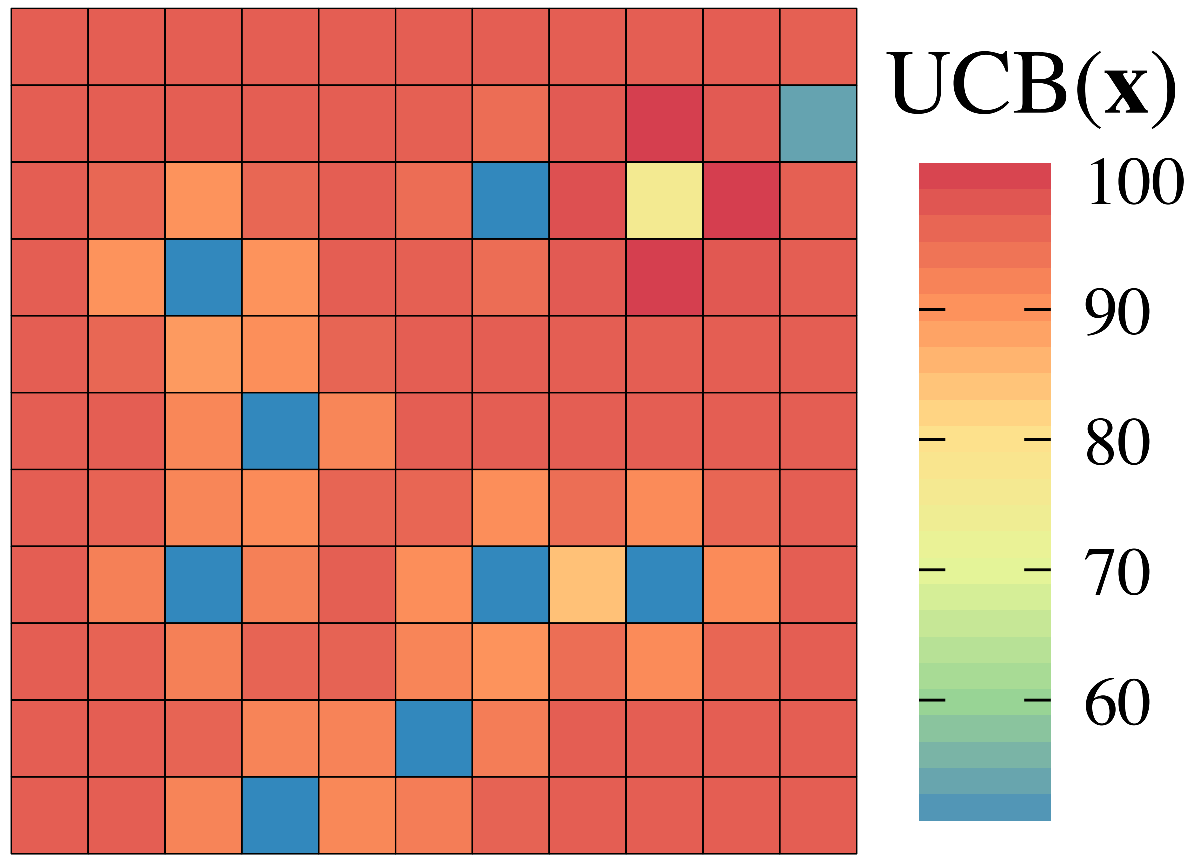
Observations



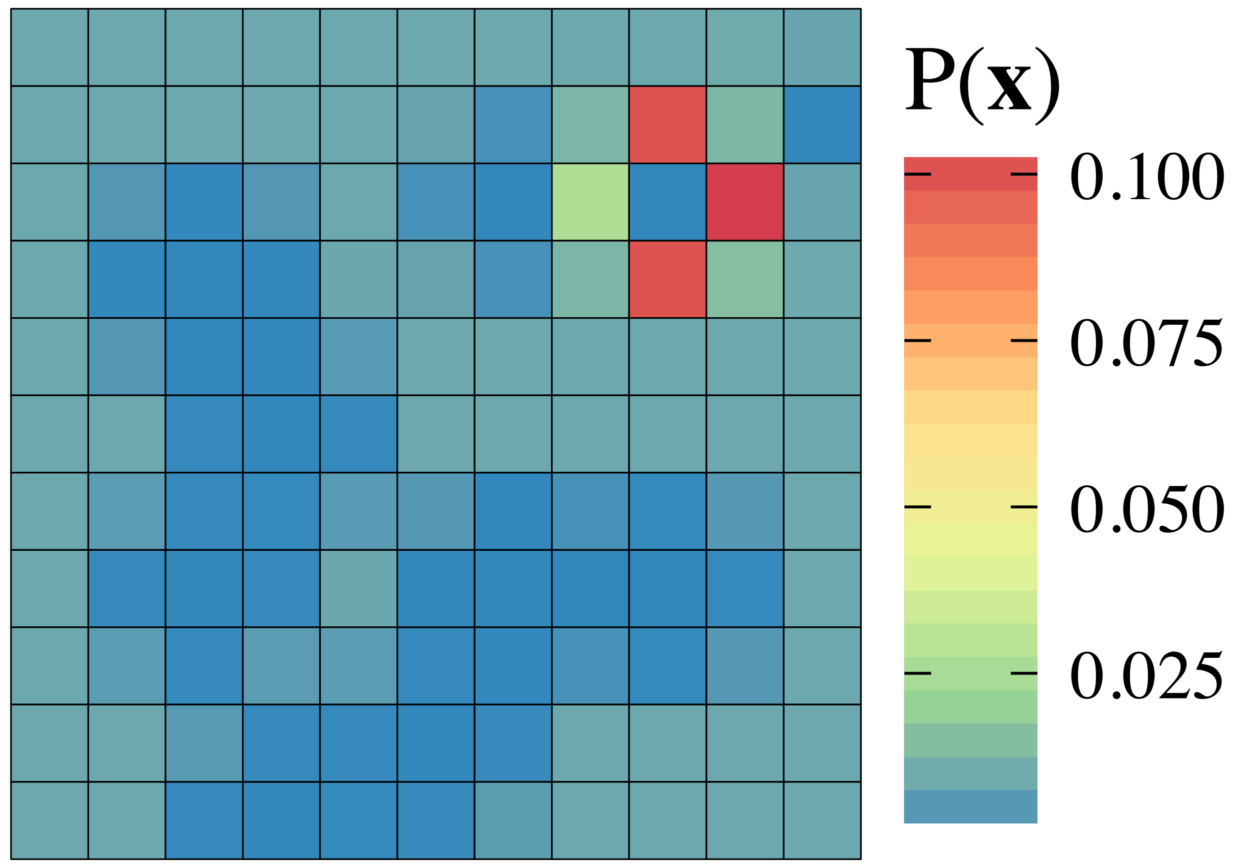
Gaussian Process (GP)

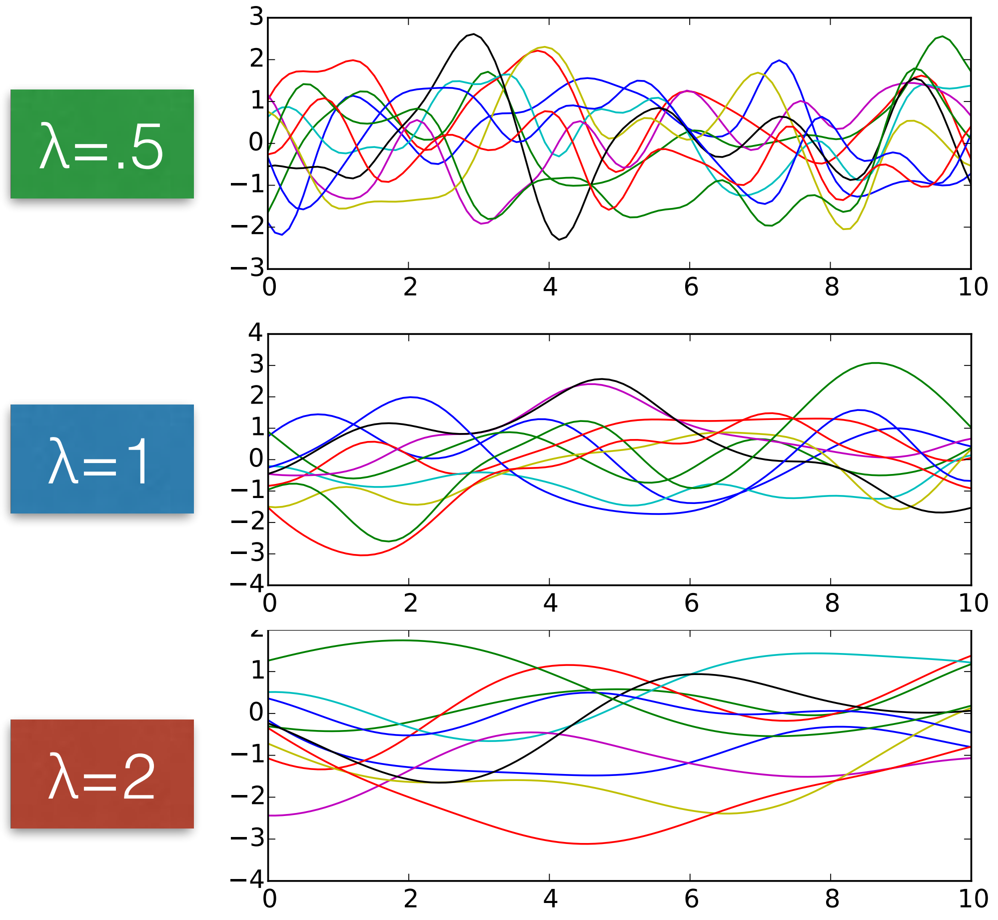


Upper Confidence Bound (UCB) Sampling



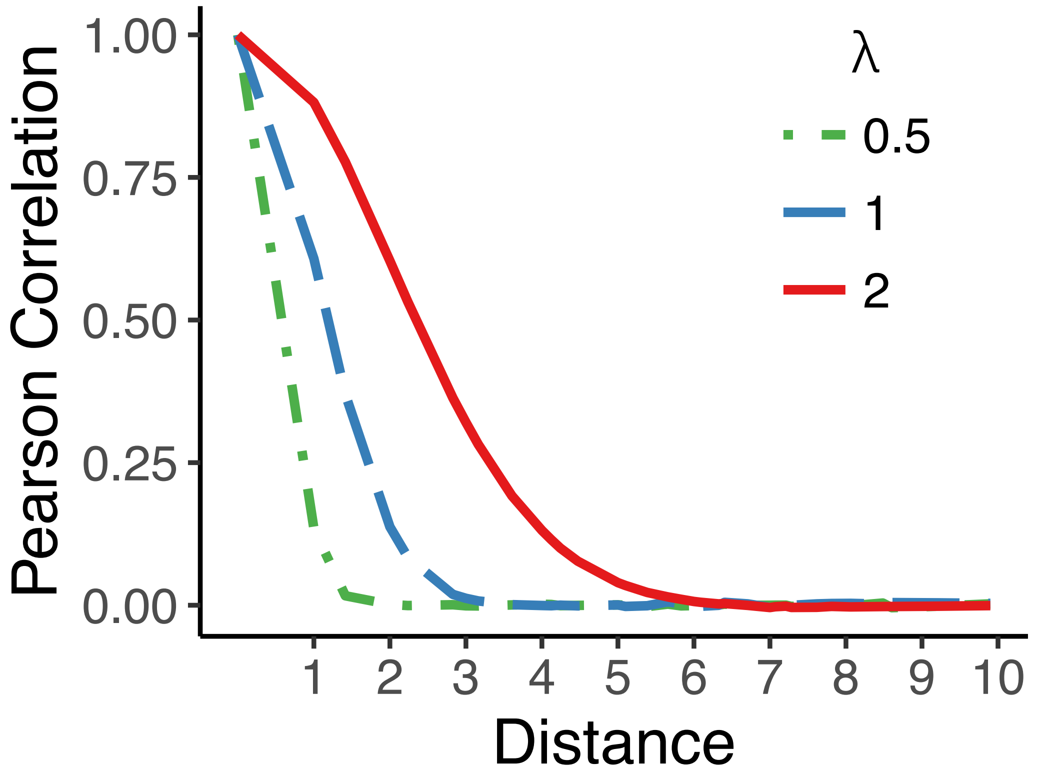
Softmax Choice Rule



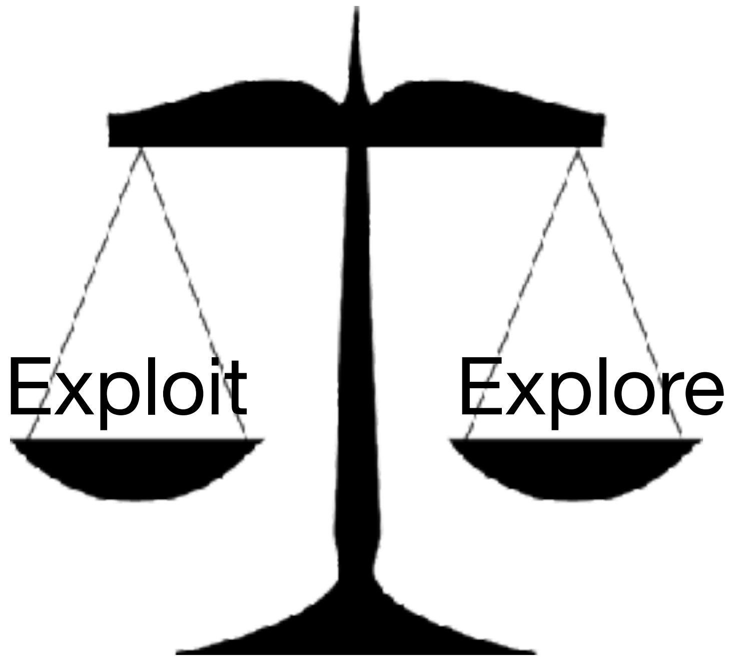


$$k_{RBF}(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{||\mathbf{x}-\mathbf{x}'||^2}{2\lambda^2}\right) \qquad UCB(\mathbf{x}_i) = \mu(\mathbf{x}_i) + \beta\sigma(\mathbf{x}_i)$$

Generalization λ

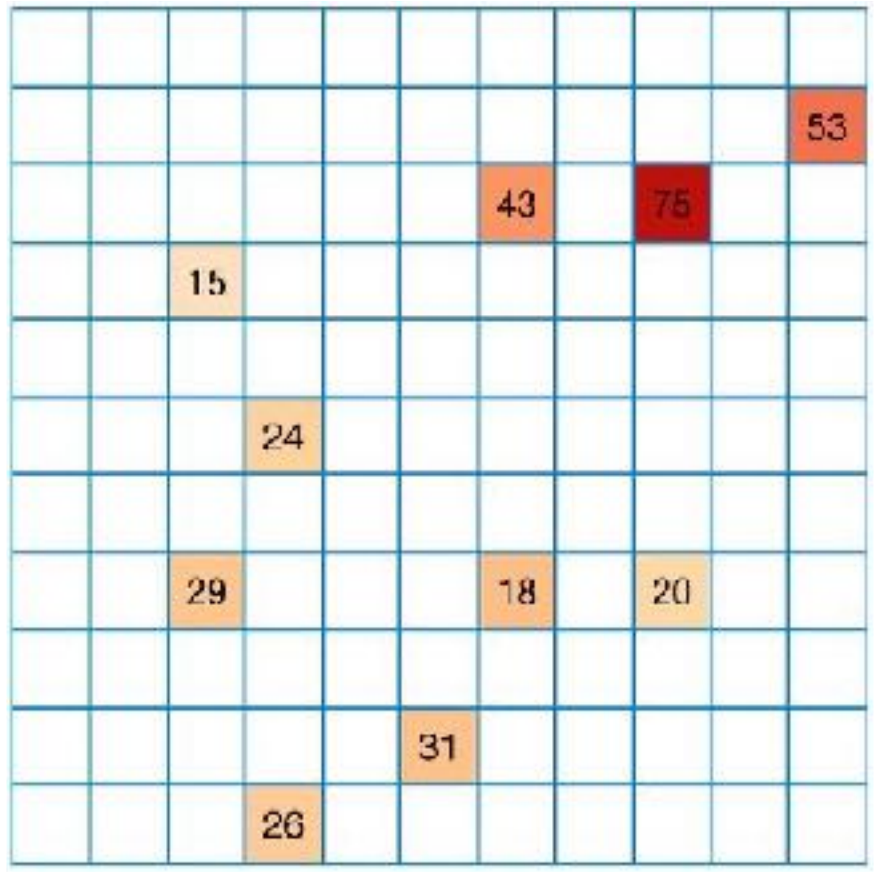


Exploration β

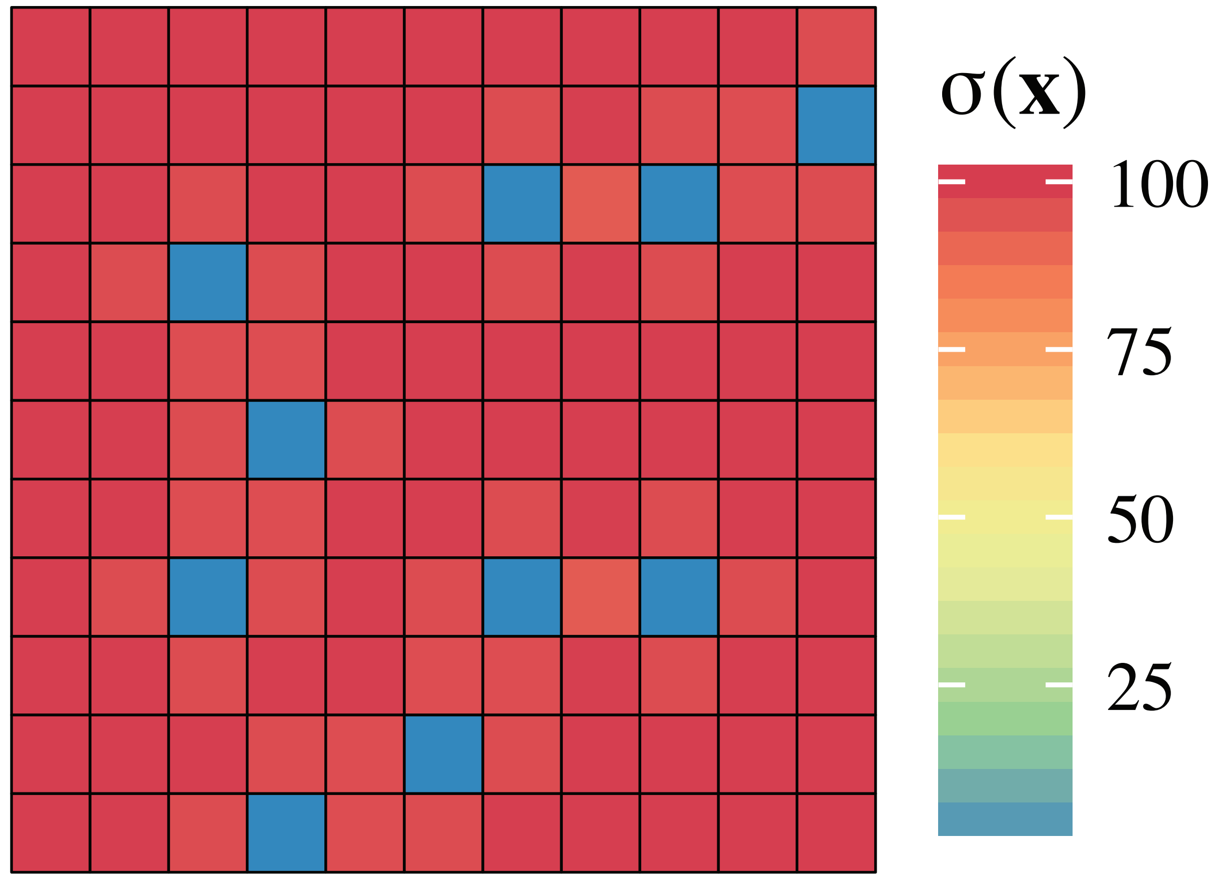


GP-UCB Model

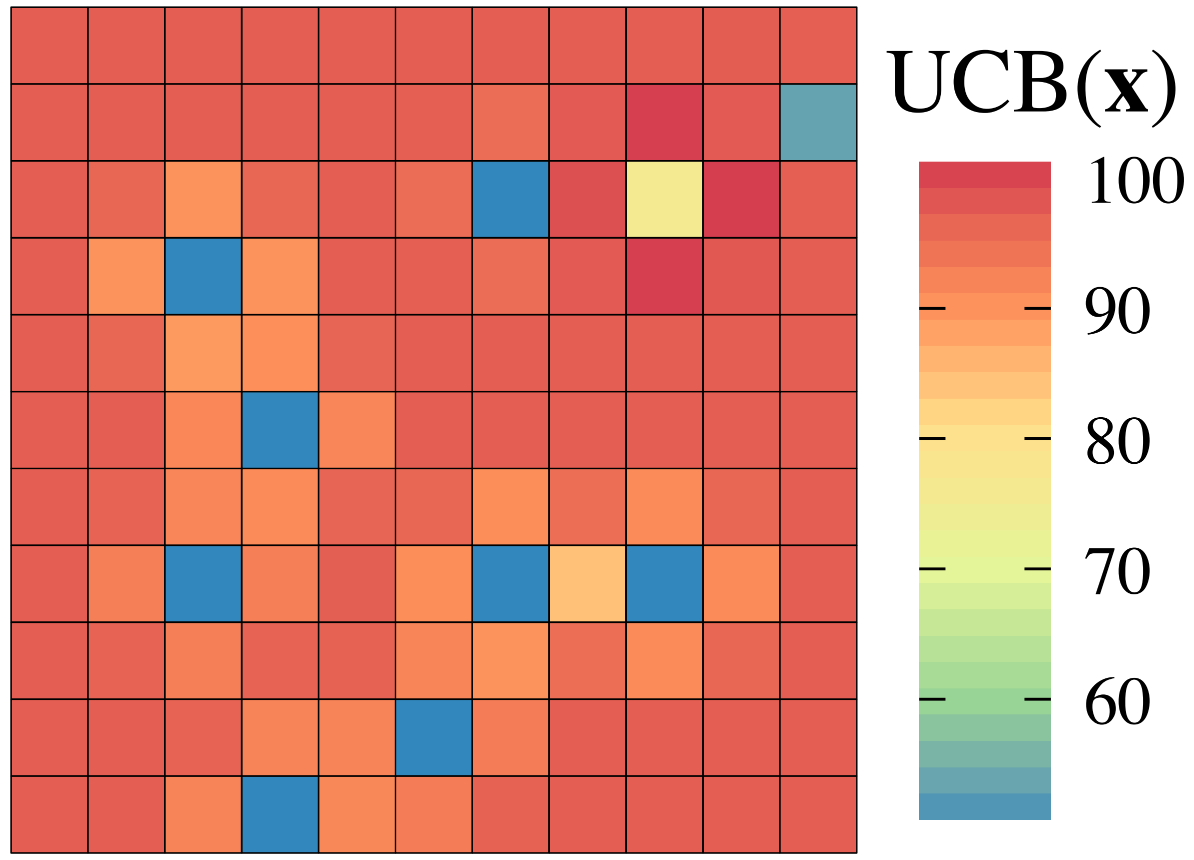
Observations



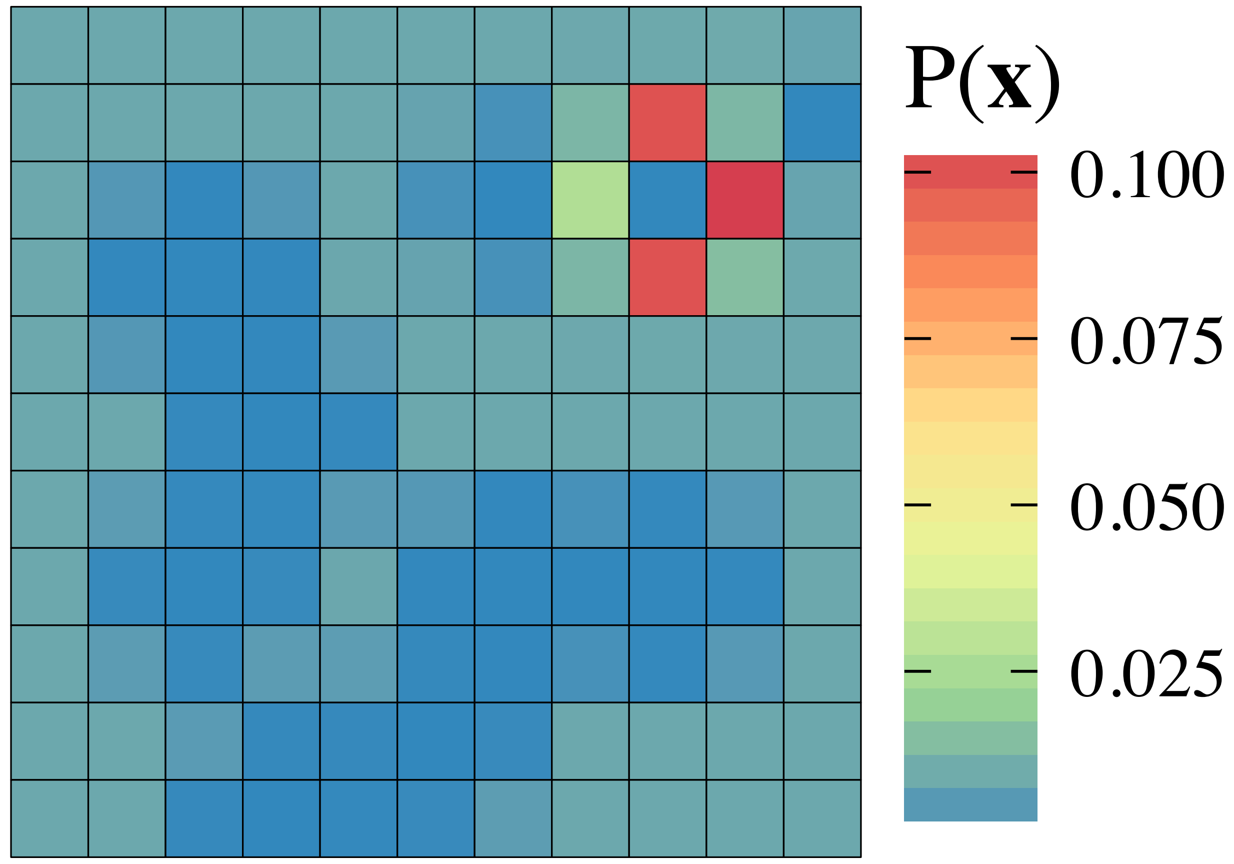
Gaussian Process (GP)



Upper Confidence Bound (UCB) Sampling



Softmax Choice Rule

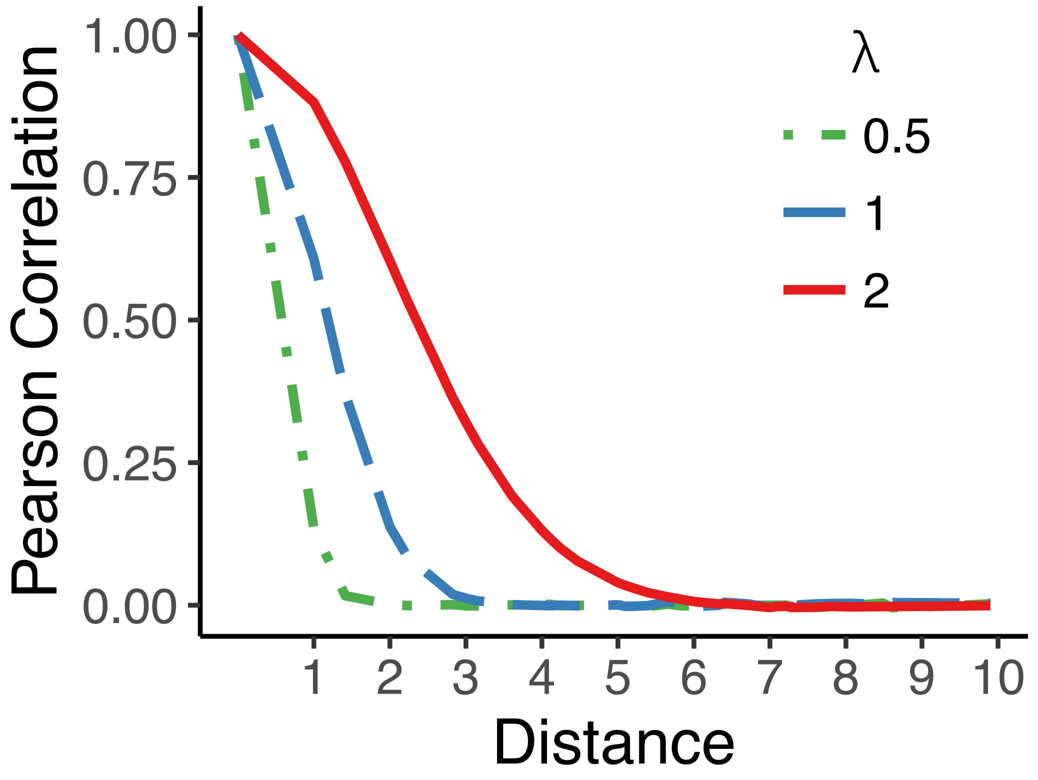


$$k_{RBF}(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{||\mathbf{x} - \mathbf{x}'||^2}{2\lambda^2}\right)$$

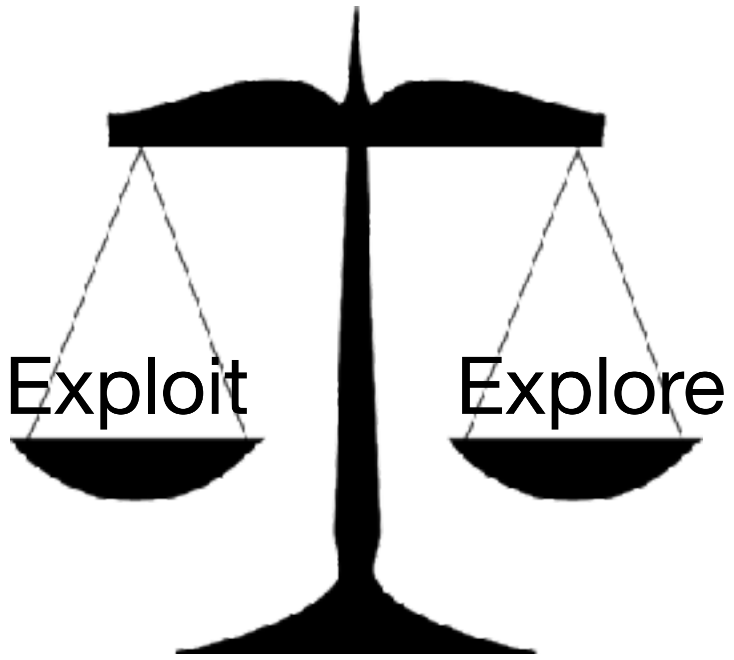
$$UCB(\mathbf{x}_i) = \mu(\mathbf{x}_i) + \beta\sigma(\mathbf{x}_i)$$

$$P(\mathbf{x}_i) \propto \exp(UCB(\mathbf{x}_i)/\tau)$$

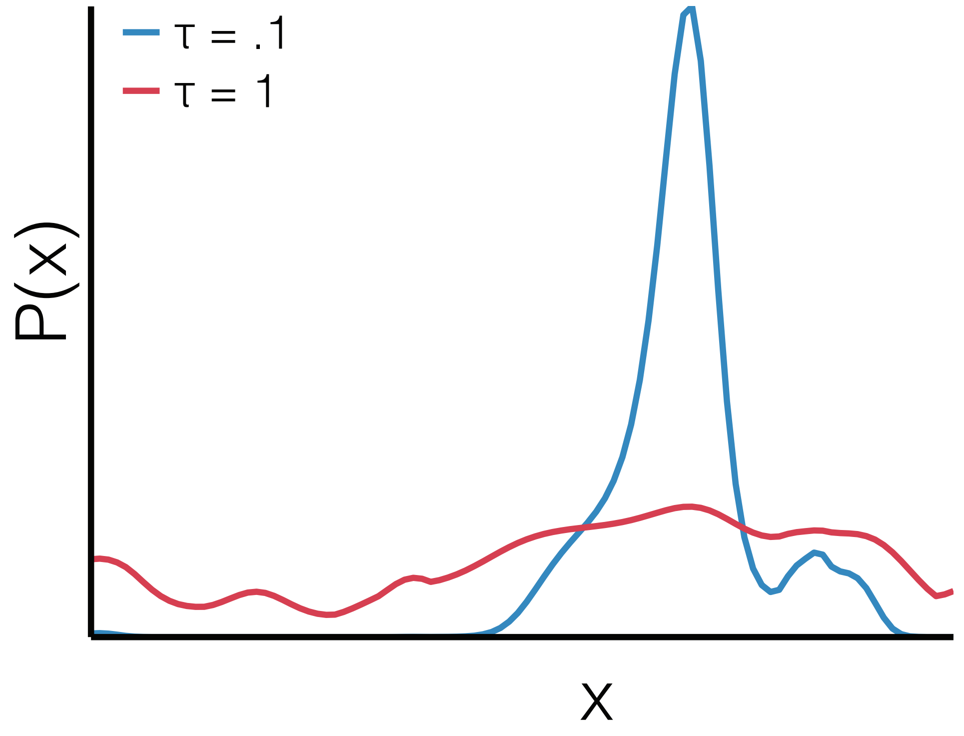
Generalization λ



Exploration β

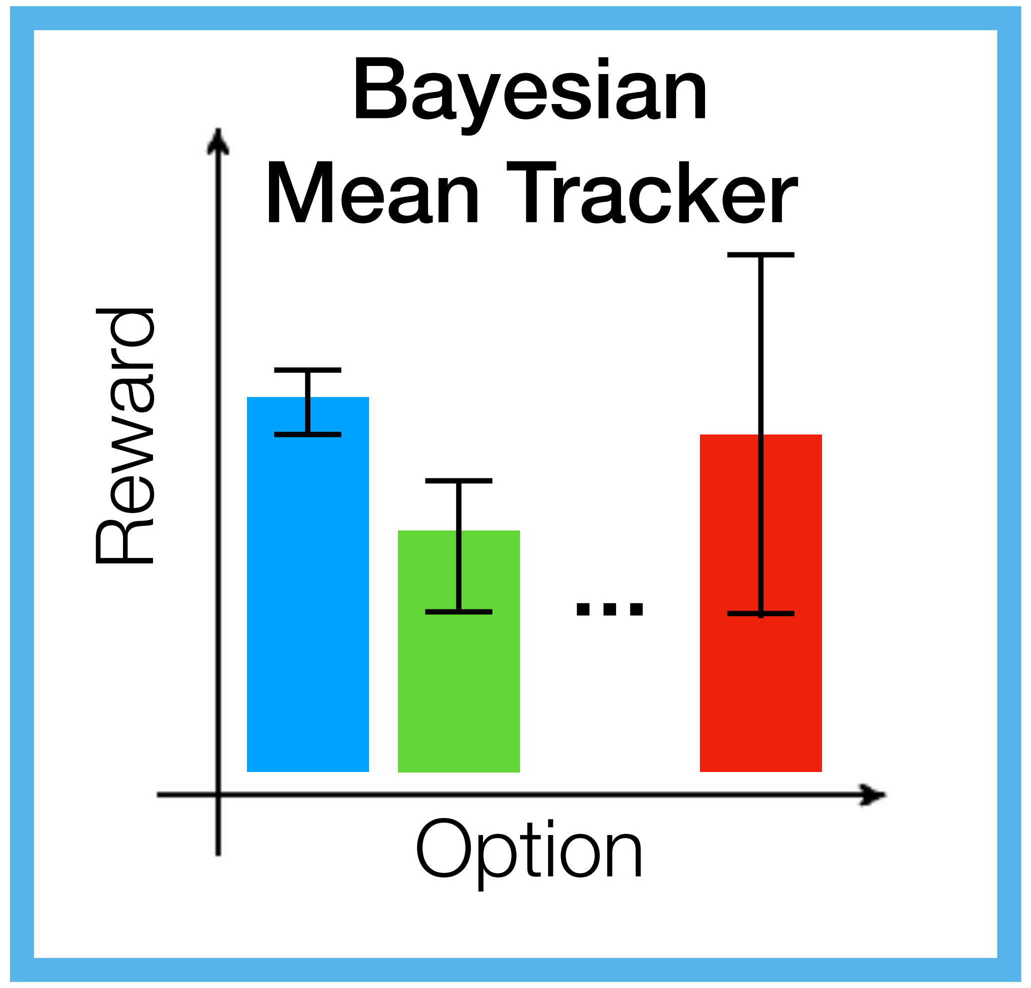


Temperature τ

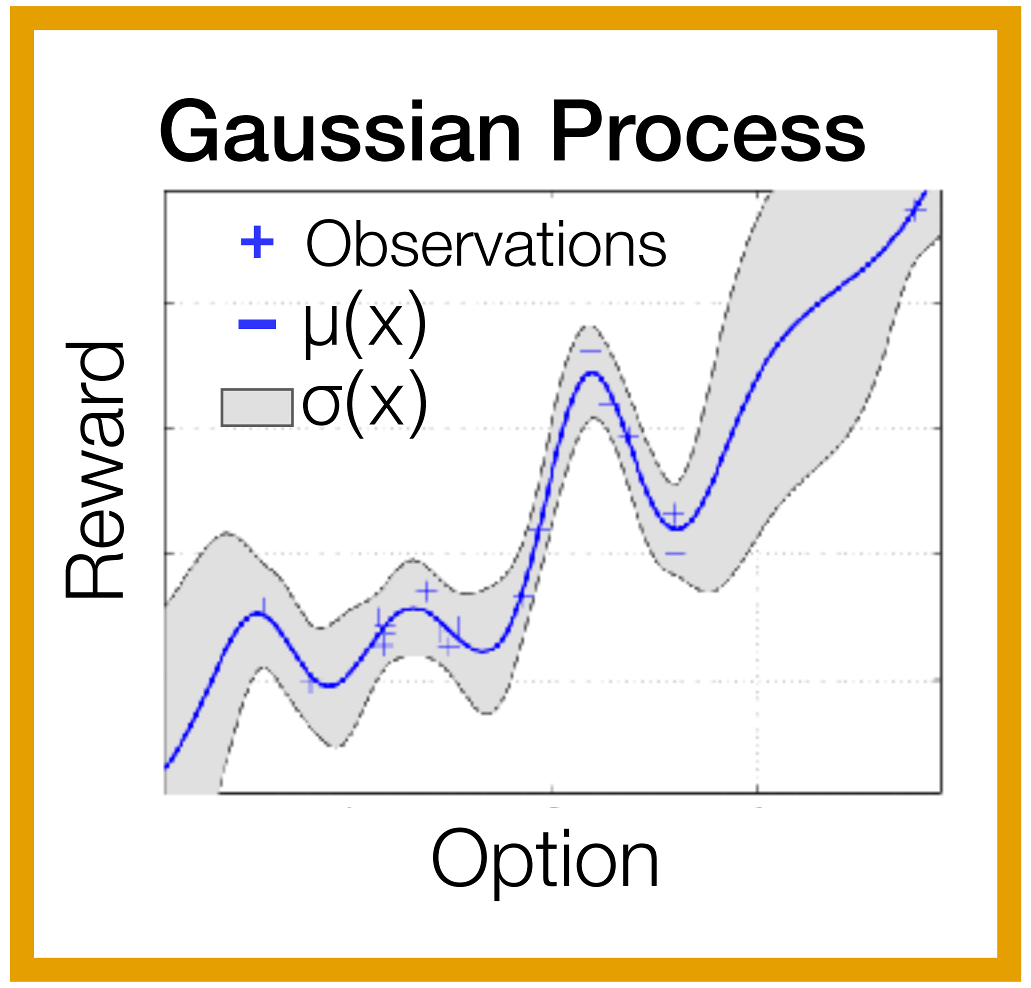


Model Comparison

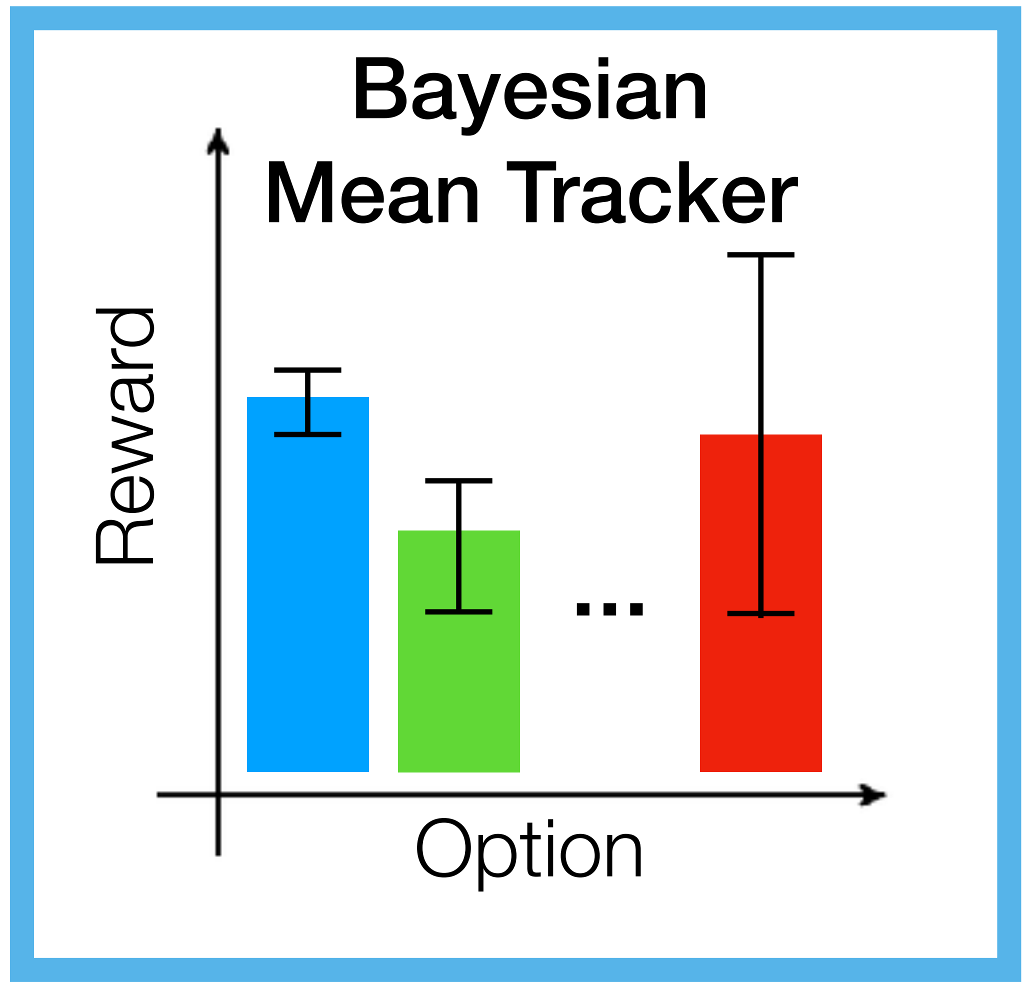
- **Traditional RL model:**
  - Learns the value of each option independently
    - e.g., Rescorla-Wagner, Q-learning, Kalman Filter, *Bayesian Mean Tracker* (BMT), etc…
  - Can balance explore-exploit dilemma using a variety of sampling strategies, but offers limited guidance about *where to explore*



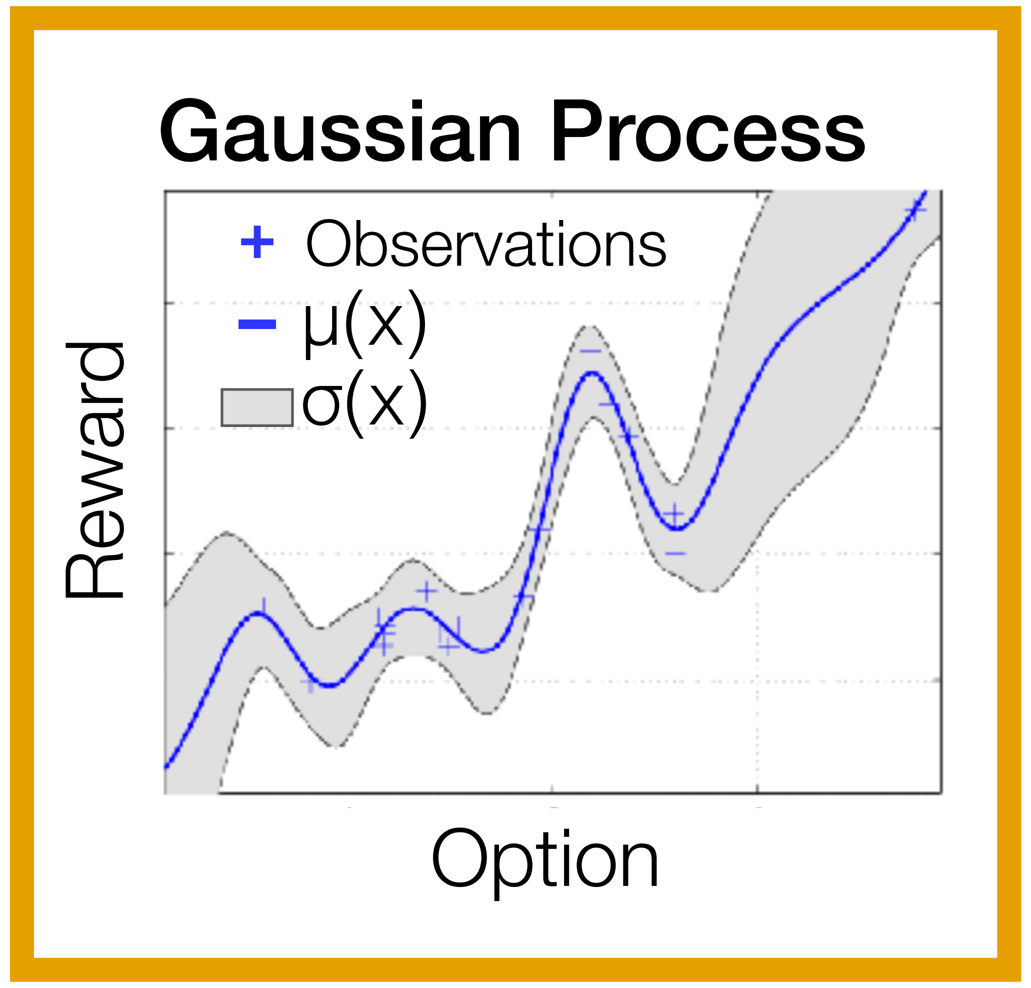
- **Function learning model:**
  - Uses function approximation to generalize about novel option
    - e.g., Neural Network function approximators, *Gaussian Process* (GP) model, etc…
  - Balances explore-exploit using the same sampling strategies as option learning models, but also makes predictions about *where* to explore through generalization



Model Comparison

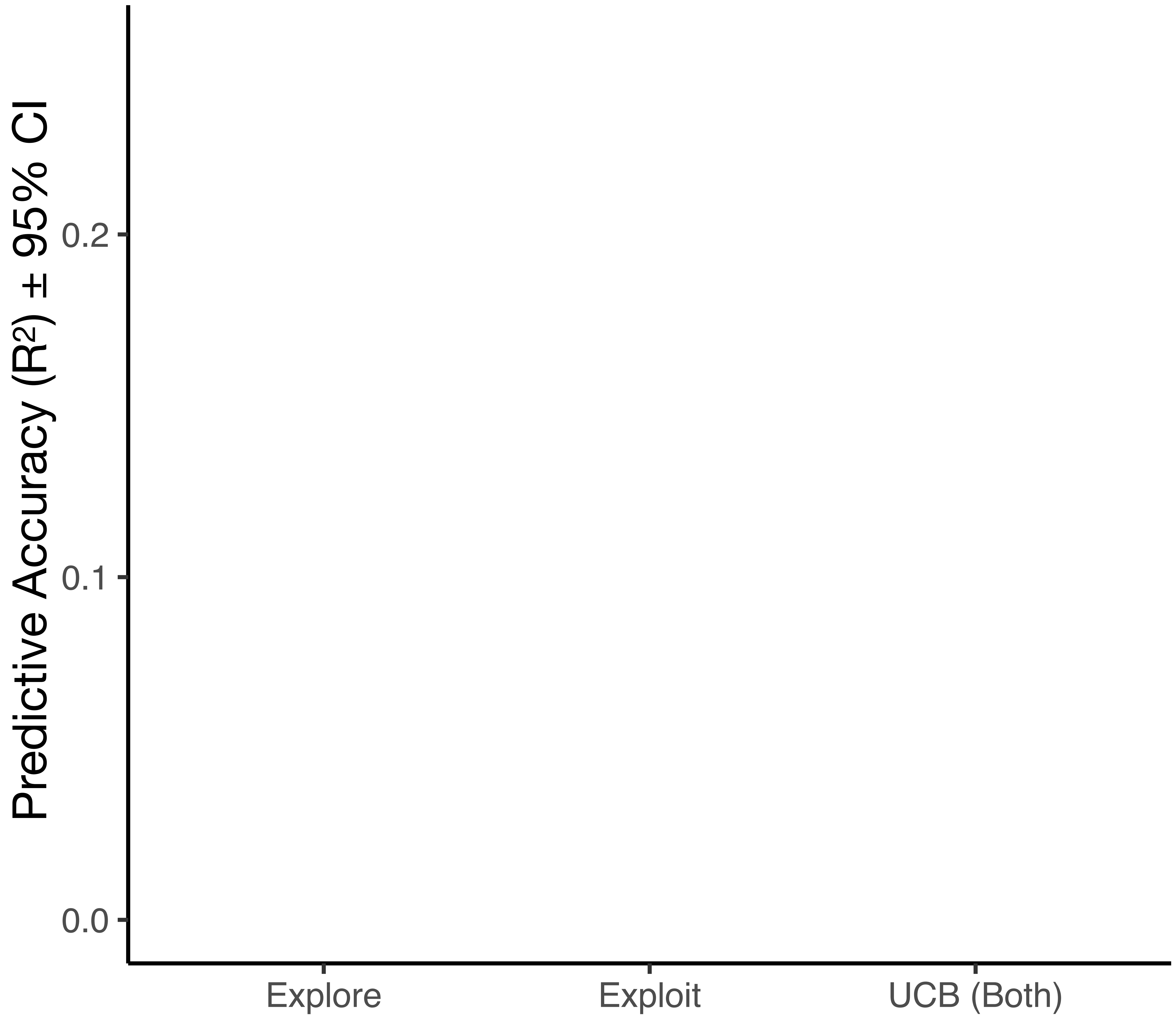


$$R^2 = 1 - \frac{\log \mathcal{L}(\mathcal{M}_k)}{\log \mathcal{L}(\mathcal{M}_{\text{rand}})}$$

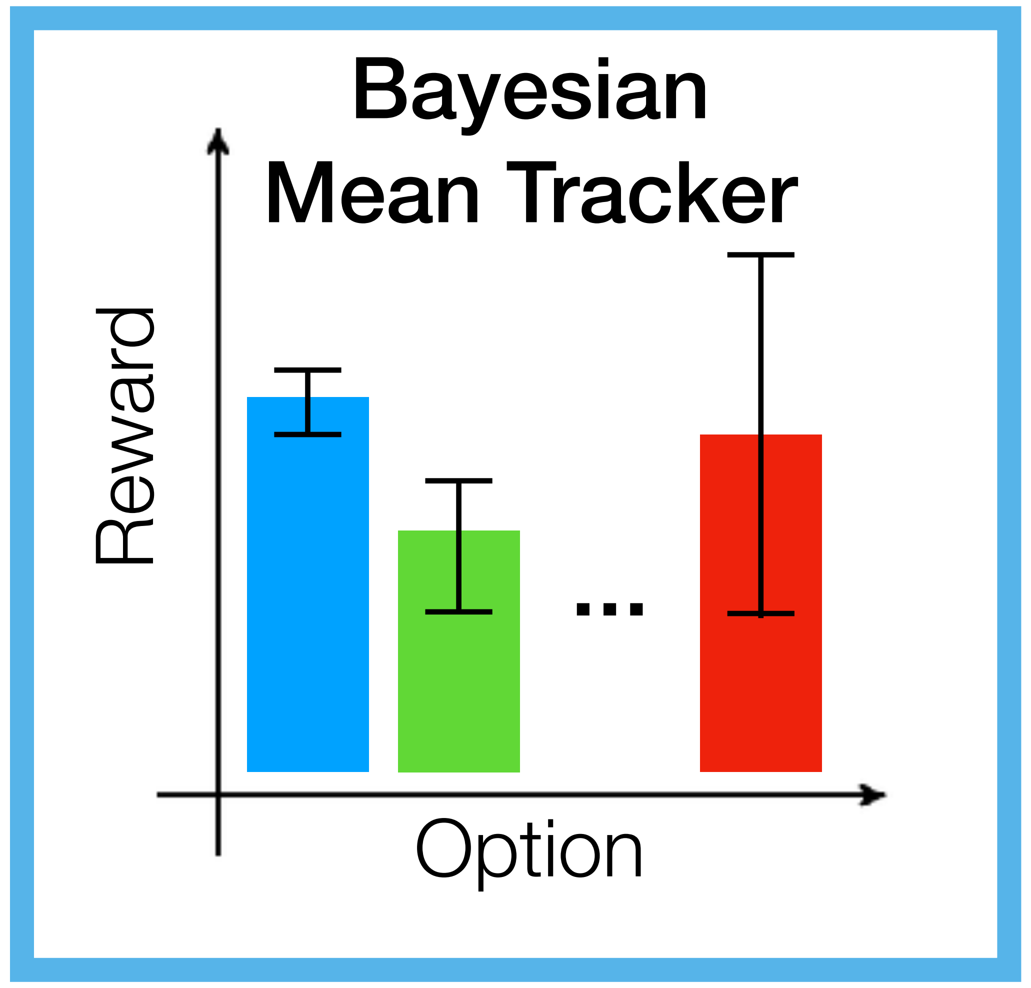


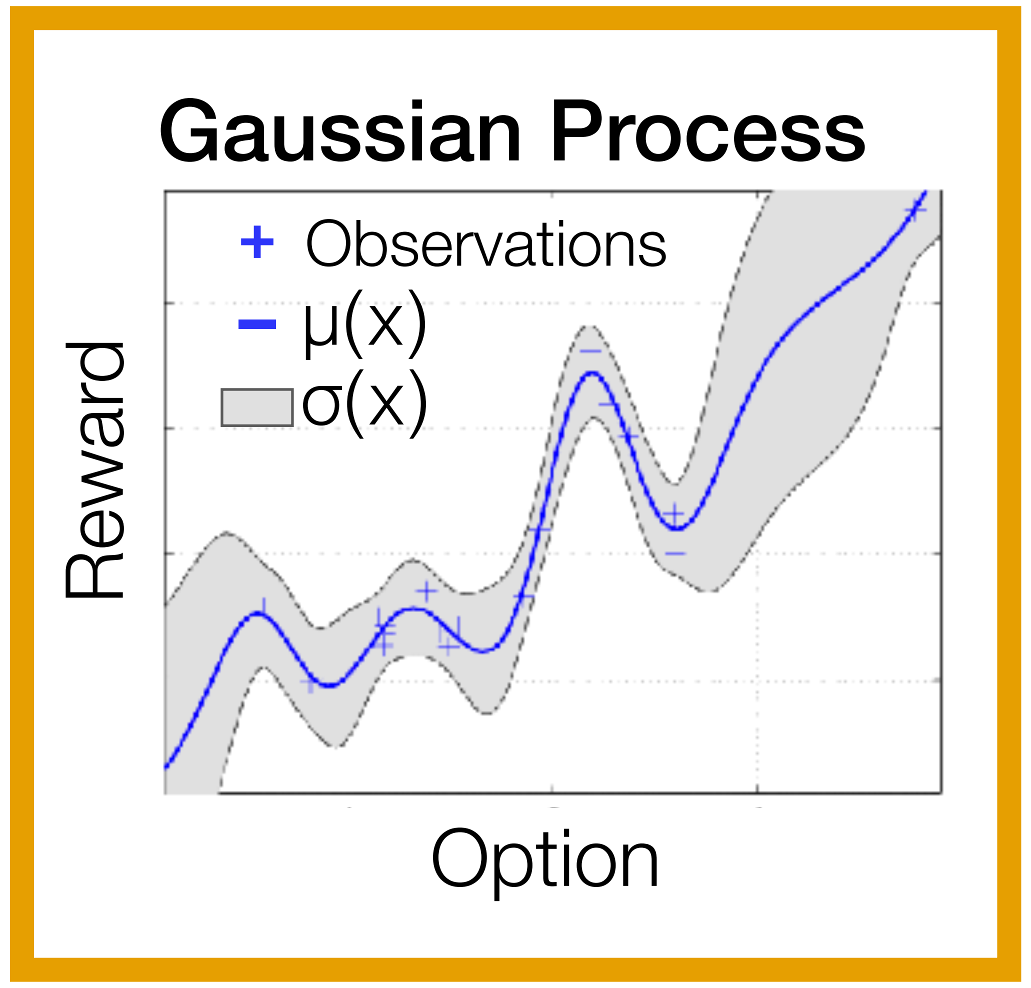
Exp. 2 from Wu et al., (*Nature Human Behaviour* 2018)

Model Comparison



$$R^2 = 1 - \frac{\log \mathcal{L}(\mathcal{M}_k)}{\log \mathcal{L}(\mathcal{M}_{\text{rand}})}$$

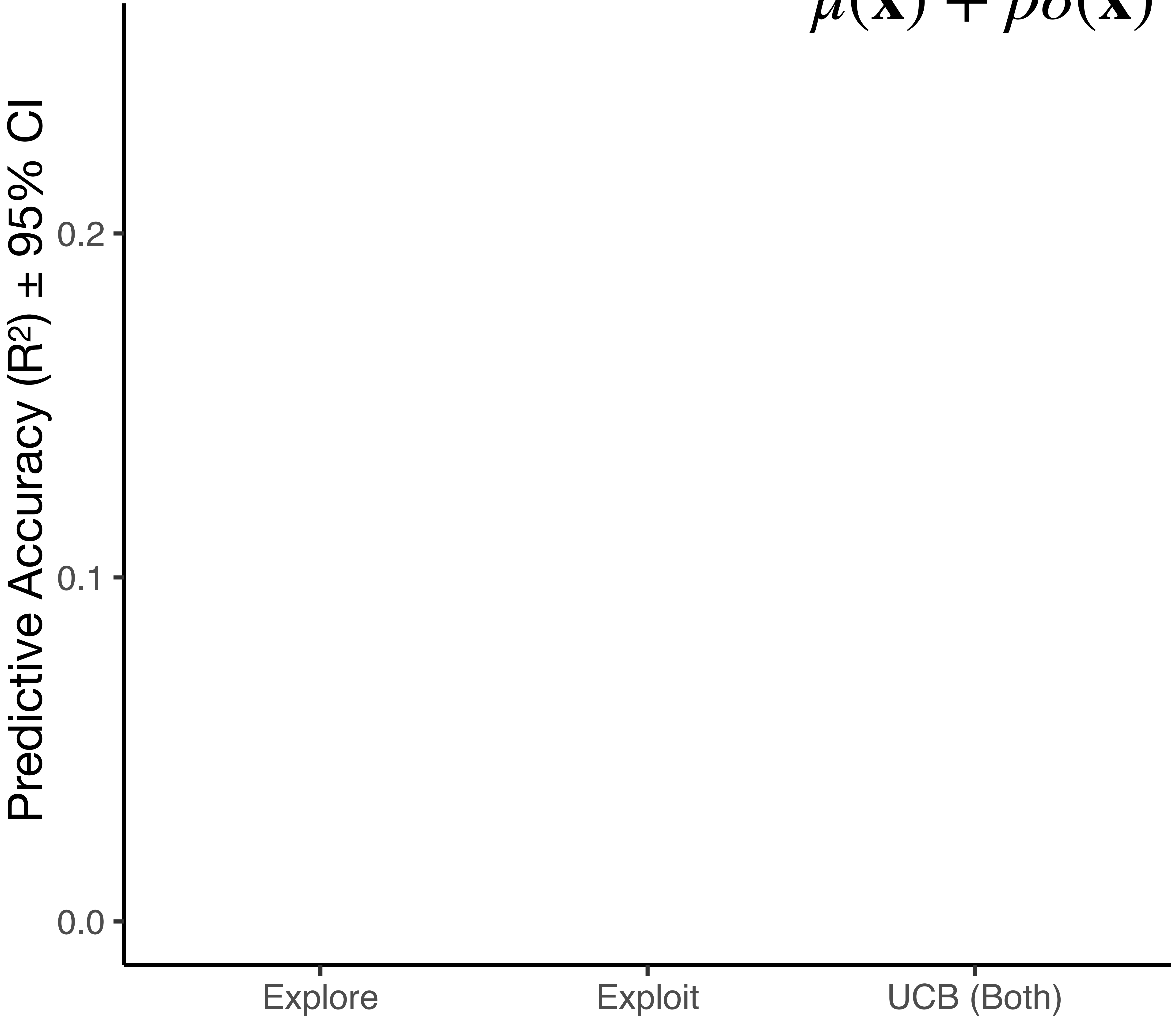




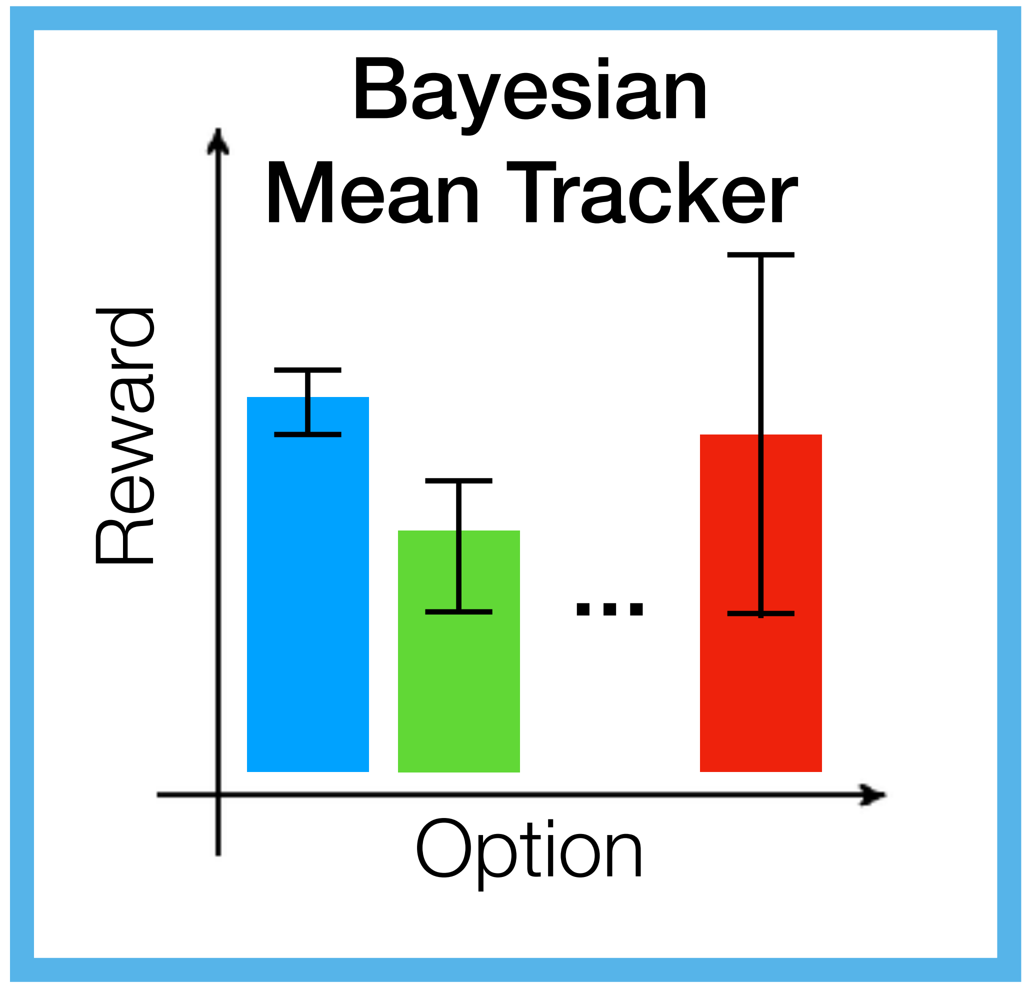
Exp. 2 from Wu et al., (*Nature Human Behaviour* 2018)

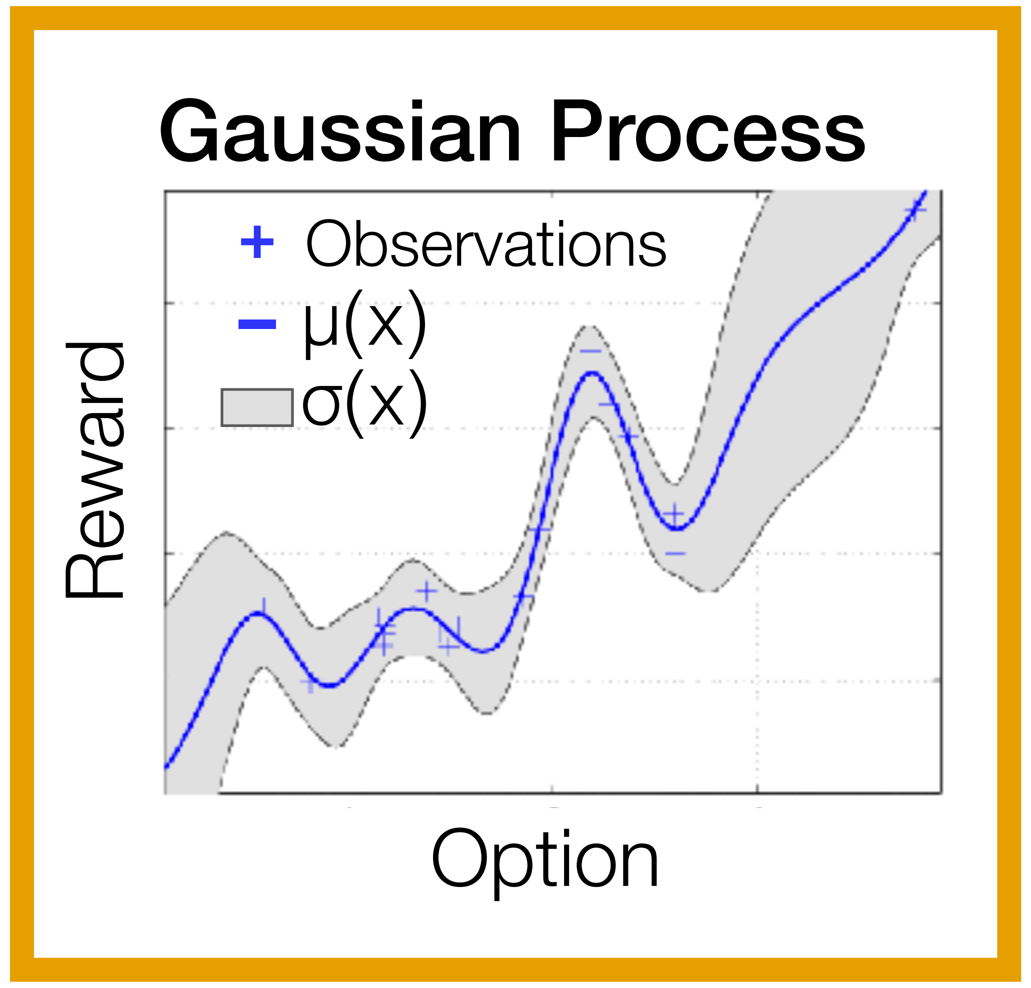
Model Comparison

$$\mu(\mathbf{x}) + \beta\sigma(\mathbf{x})$$



$$R^2 = 1 - \frac{\log \mathcal{L}(\mathcal{M}_k)}{\log \mathcal{L}(\mathcal{M}_{\text{rand}})}$$

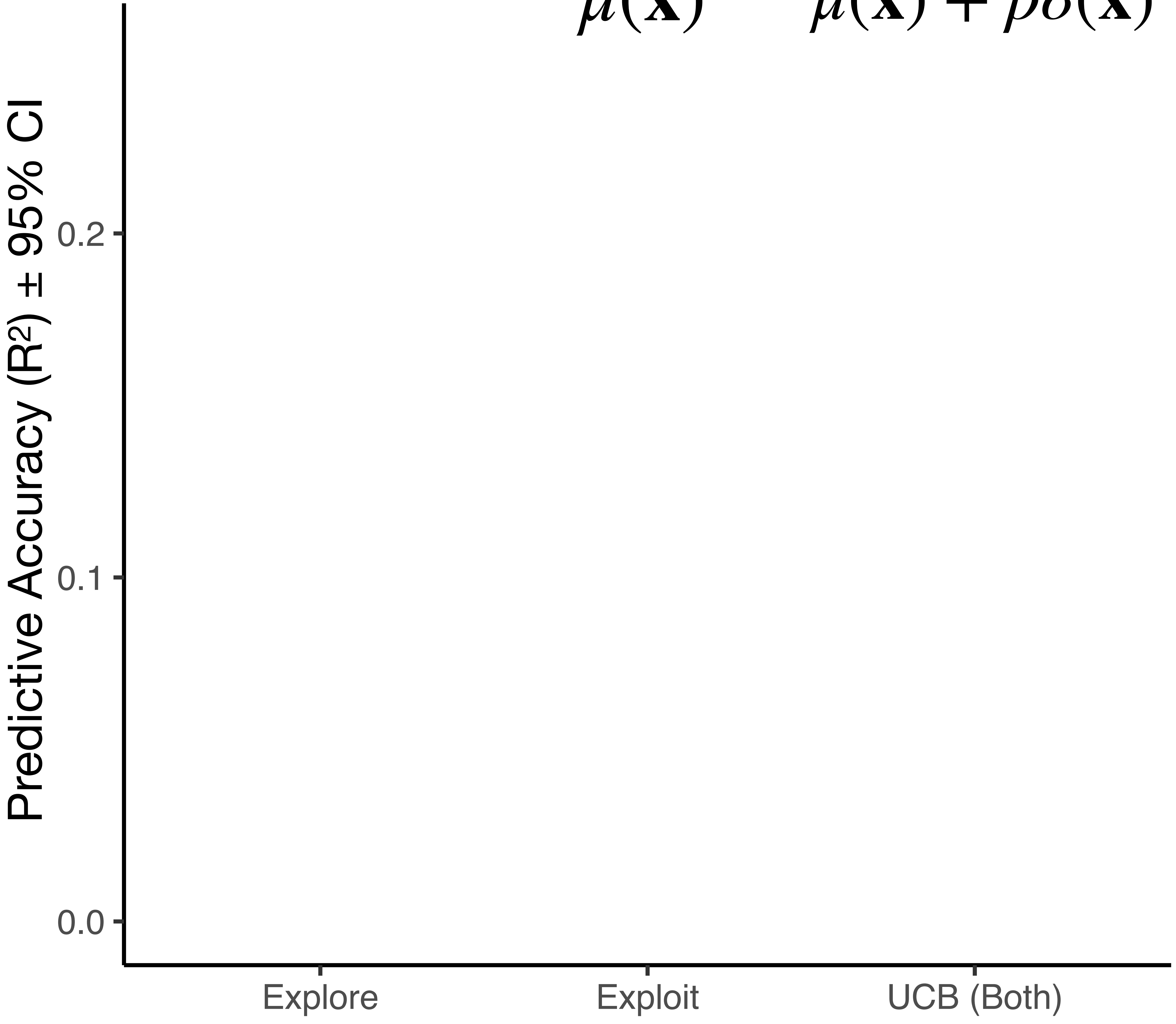




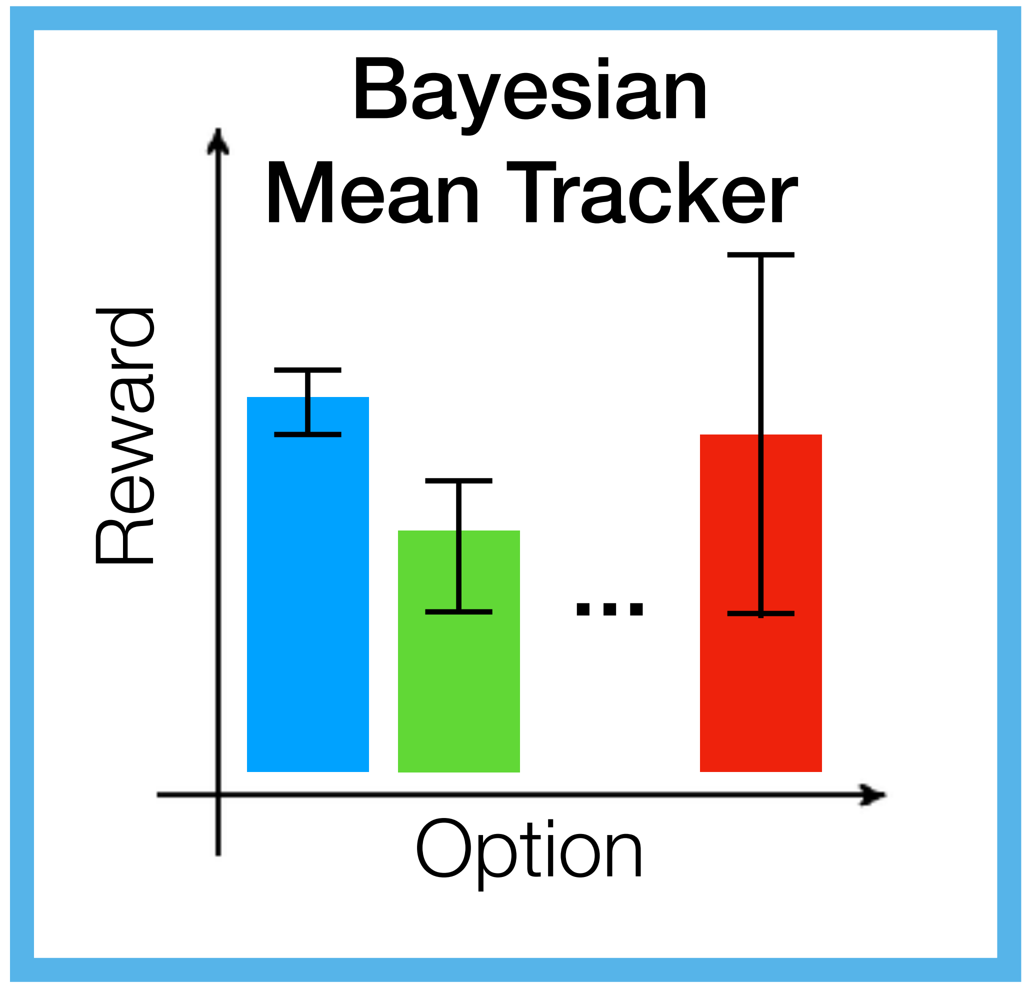
Exp. 2 from Wu et al., (*Nature Human Behaviour* 2018)

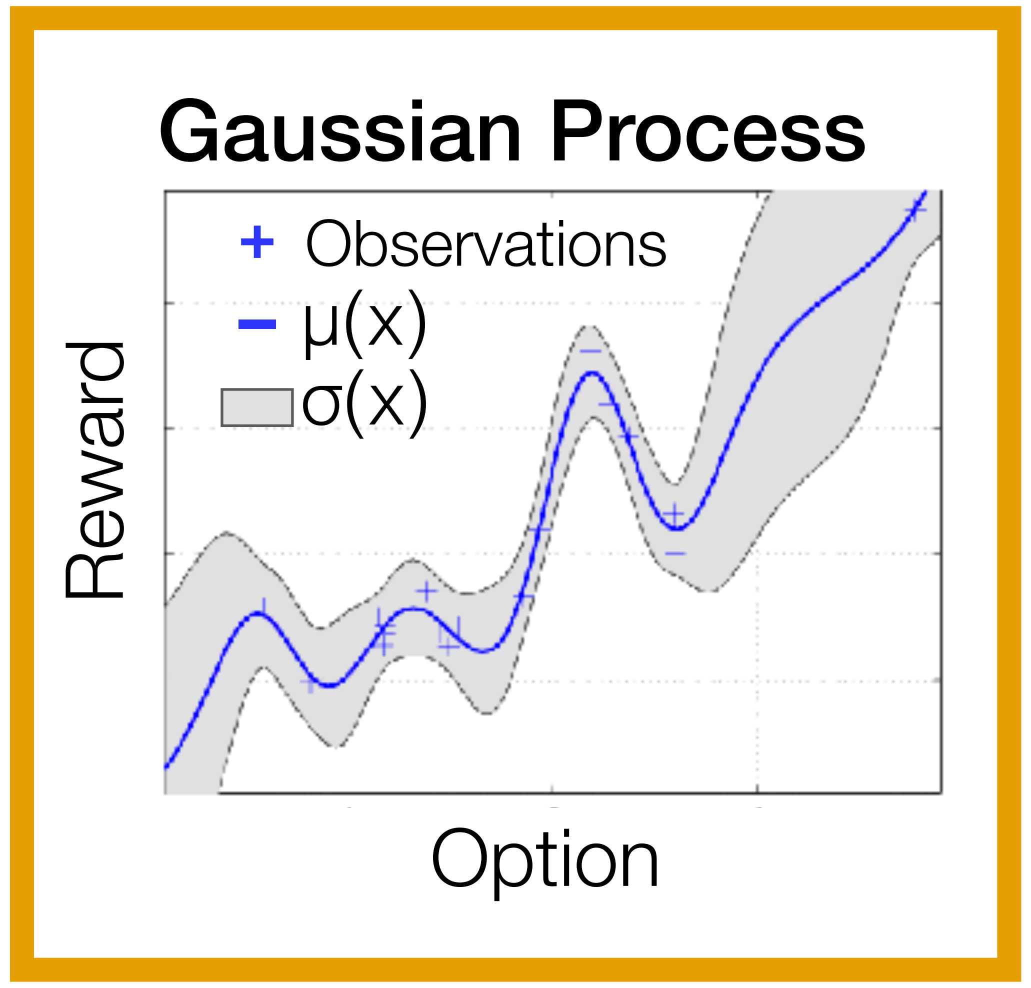
Model Comparison

$\mu(\mathbf{x}) \qquad \mu(\mathbf{x}) + \beta\sigma(\mathbf{x})$



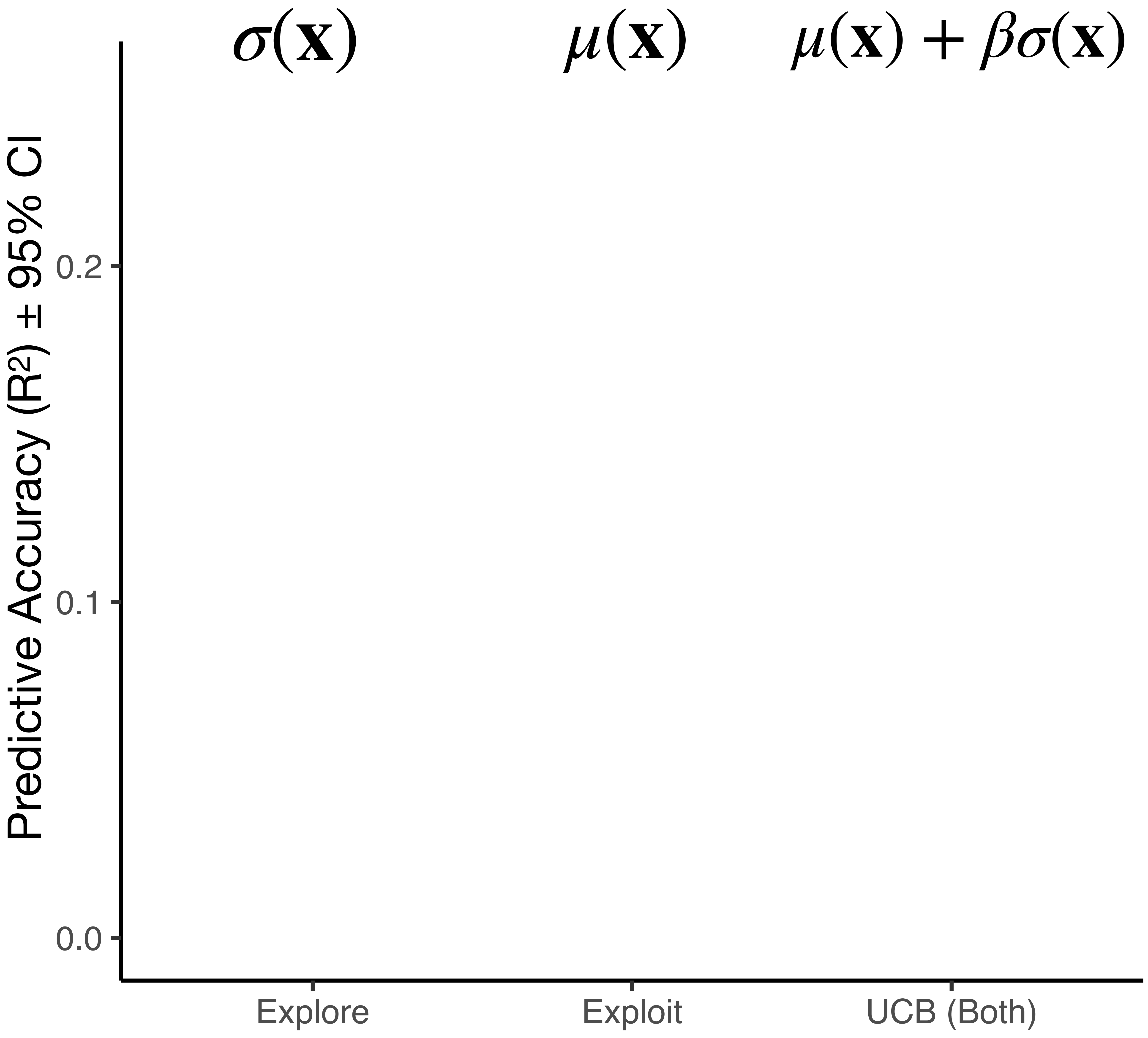
$$R^2 = 1 - \frac{\log \mathcal{L}(\mathcal{M}_k)}{\log \mathcal{L}(\mathcal{M}_{\text{rand}})}$$



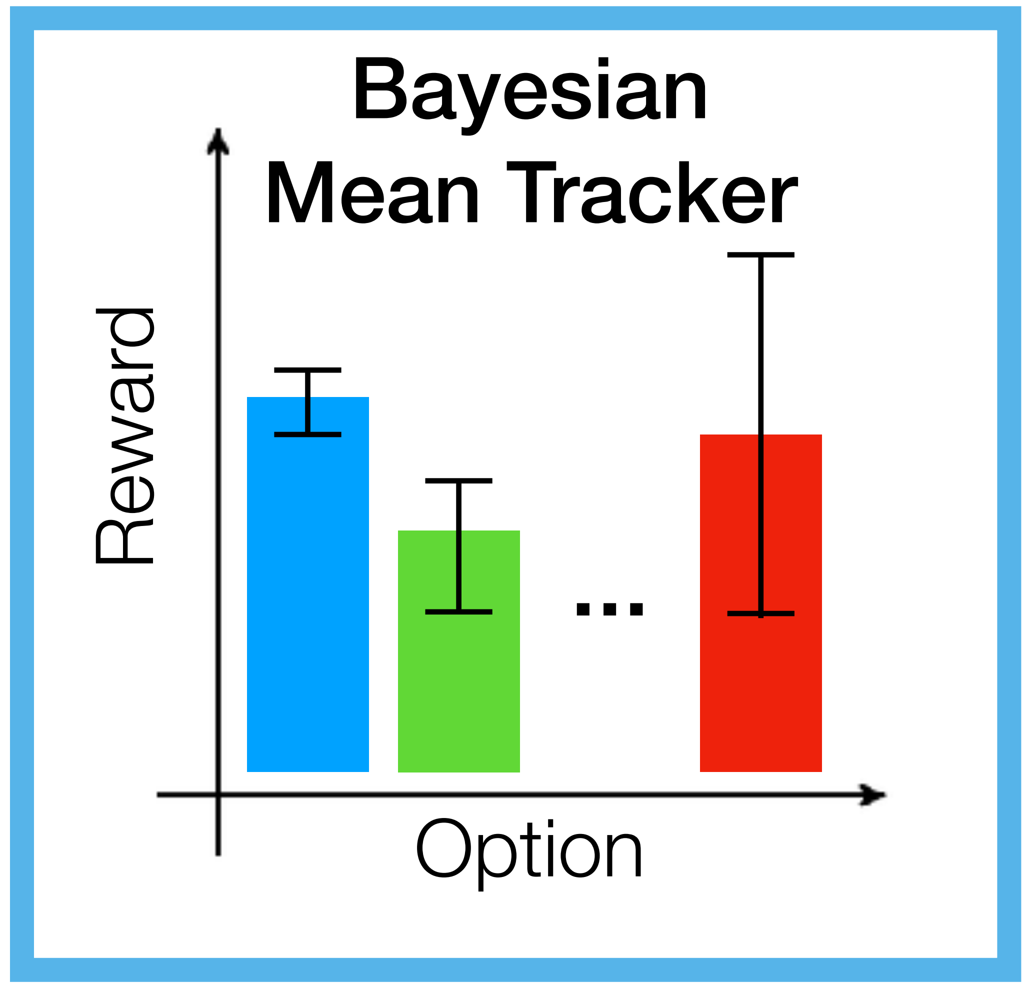


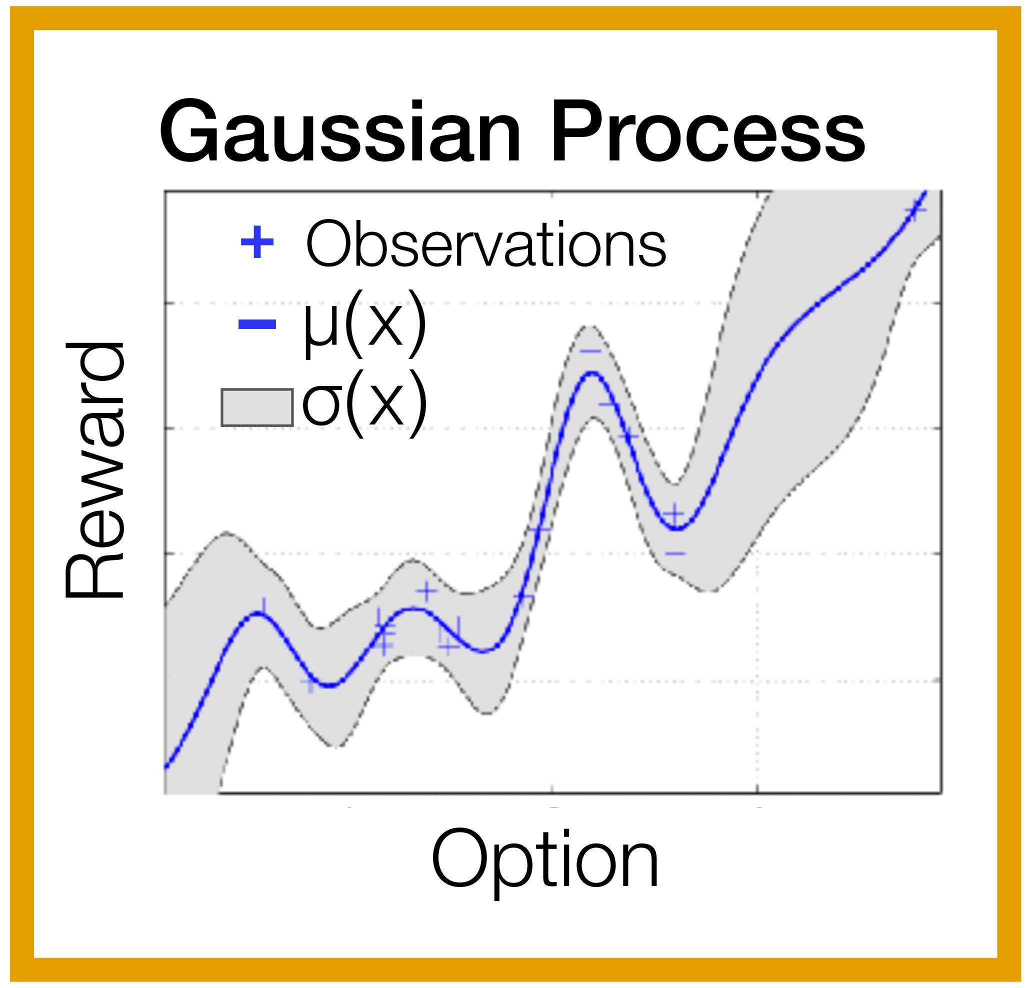
Exp. 2 from Wu et al., (*Nature Human Behaviour* 2018)

Model Comparison



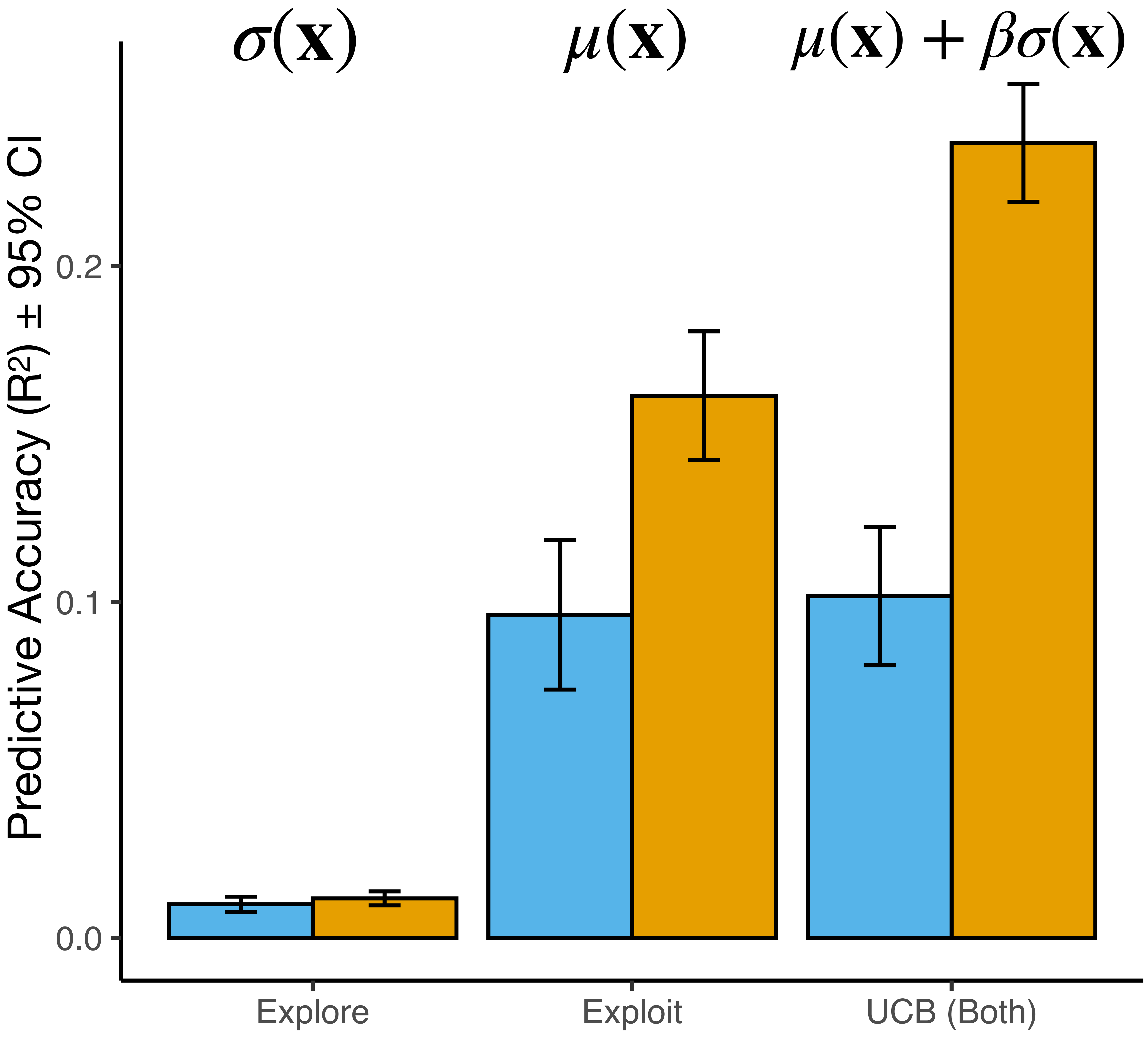
$$R^2 = 1 - \frac{\log \mathcal{L}(\mathcal{M}_k)}{\log \mathcal{L}(\mathcal{M}_{\text{rand}})}$$



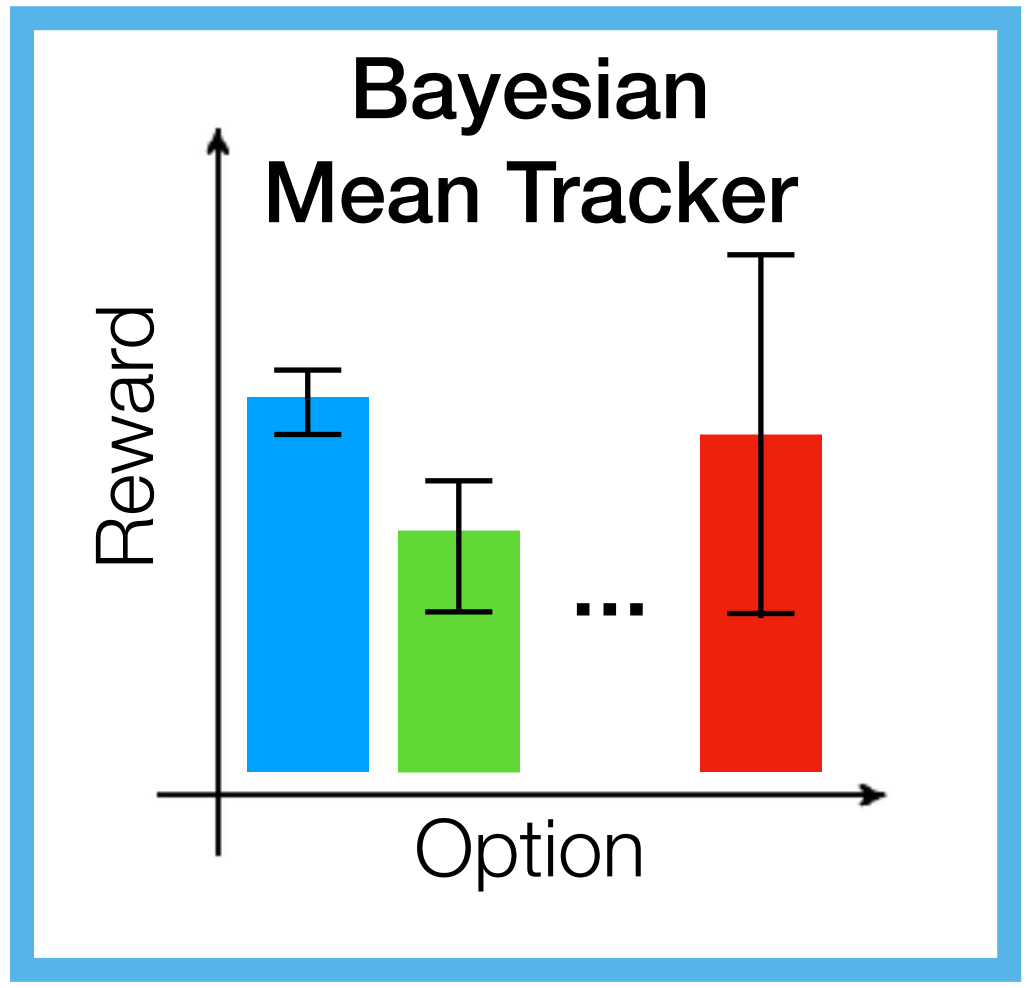


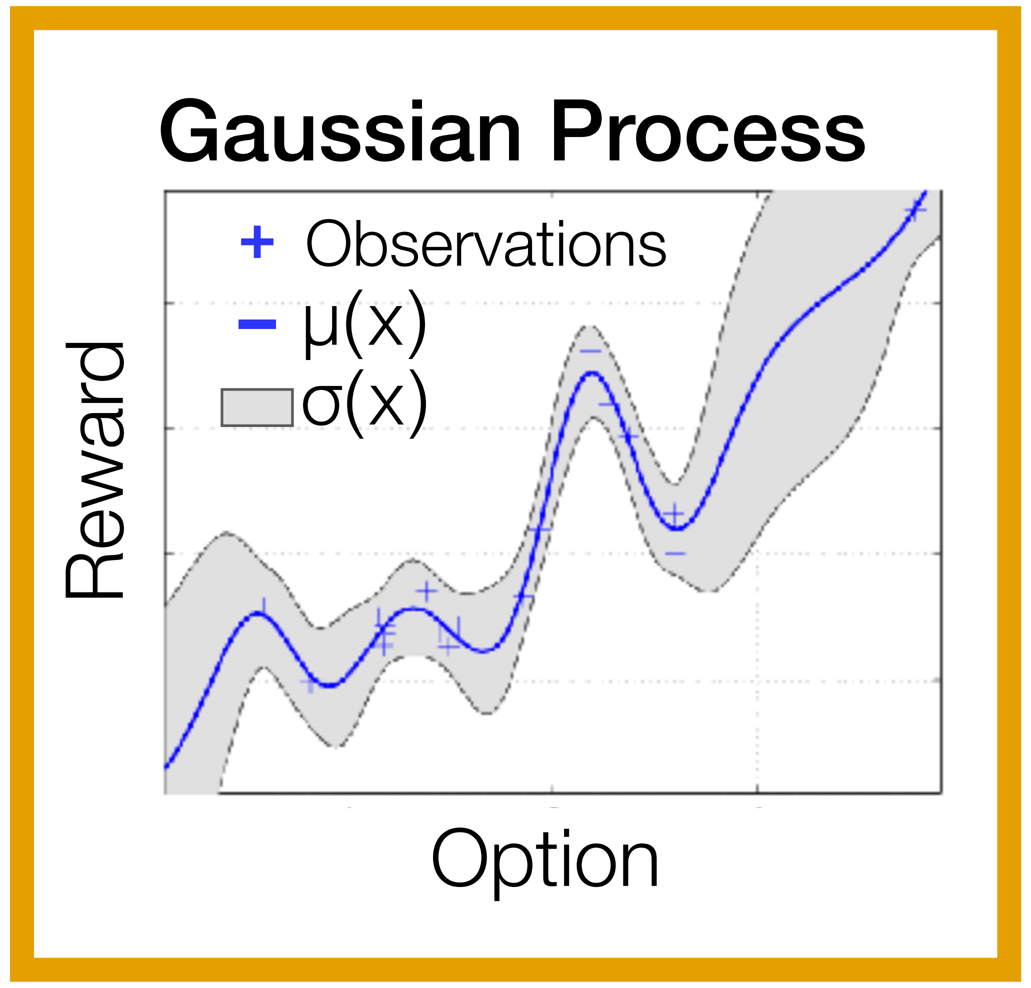
Exp. 2 from Wu et al., (*Nature Human Behaviour* 2018)

Model Comparison



$$R^2 = 1 - \frac{\log \mathcal{L}(\mathcal{M}_k)}{\log \mathcal{L}(\mathcal{M}_{\text{rand}})}$$





Exp. 2 from Wu et al., (*Nature Human Behaviour* 2018)

Establishing a new paradigm

1. Generalization guides exploration
   Wu, Schulz, Nelson, Speekenbrink & Meder (*Cogsci* 2017)
   Wu, Schulz, Nelson, Speekenbrink & Meder (*Nature Human Behaviour* 2018)

2. Learning like a child
   Schulz, Wu, Ruggeri & Meder (*PsychSci* 2019)
   Meder, Wu, Schulz & Ruggeri (*Dev Sci* in press)

3. Graph-structured Generalization
   Wu, Schulz & Gershman (*Cogsci* 2019)
   Wu, Schulz & Gershman (*CCN* 2019)
   Wu, Schulz & Gershman (*Comput Brain Behav* 2020)

4. Search in abstract conceptual spaces
   Wu, Schulz, Garvert, Meder & Schuck (*Cogsci* 2018)
   Wu, Schulz, Garvert, Meder & Schuck (*PLOS Comp Bio* 2020)

5. Safe exploration
   Schulz, Wu, Huys, Krause & Speekenbrink (*Cognitive Science* 2018)

6. Clinically depressed populations
   Schefft, Wu, Meder, Köhler & Schulz (*in prep*)

7. Social search in VR
   Wu, Ho, Kahl, Leuker, Meder & Kurvers (*bioRxiv* 2021)

Establishing a new paradigm

1. Generalization guides exploration
   Wu, Schulz, Nelson, Speekenbrink & Meder (*Cogsci* 2017)
   Wu, Schulz, Nelson, Speekenbrink & Meder (*Nature Human Behaviour* 2018)

2. Learning like a child
   Schulz, Wu, Ruggeri & Meder (*PsychSci* 2019)
   Meder, Wu, Schulz & Ruggeri (*Dev Sci* in press)

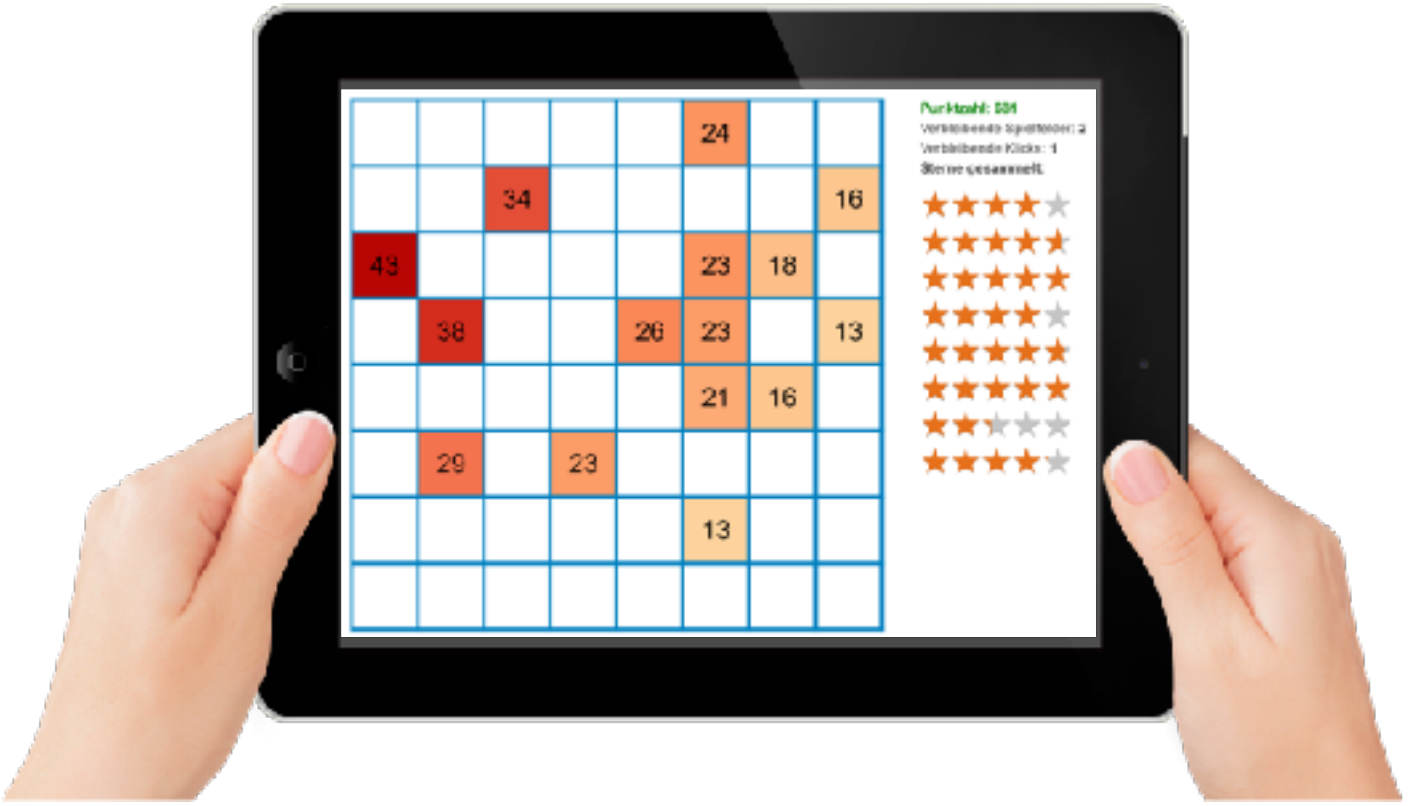
3. Graph-structured Generalization
   Wu, Schulz & Gershman (*Cogsci* 2019)
   Wu, Schulz & Gershman (*CCN* 2019)
   Wu, Schulz & Gershman (*Comput Brain Behav* 2020)

4. Search in abstract conceptual spaces
   Wu, Schulz, Garvert, Meder & Schuck (*Cogsci* 2018)
   Wu, Schulz, Garvert, Meder & Schuck (*PLOS Comp Bio* 2020)

5. Safe exploration
   Schulz, Wu, Huys, Krause & Speekenbrink (*Cognitive Science* 2018)

6. Clinically depressed populations
   Schefft, Wu, Meder, Köhler & Schulz (*in prep*)

7. Social search in VR
   Wu, Ho, Kahl, Leuker, Meder & Kurvers (*bioRxiv* 2021)



Schulz, Wu, Ruggeri & Meder (*PsychSci* 2019)

Establishing a new paradigm

1. Generalization guides exploration
   Wu, Schulz, Nelson, Speekenbrink & Meder (*Cogsci* 2017)
   Wu, Schulz, Nelson, Speekenbrink & Meder (*Nature Human Behaviour* 2018)

2. Learning like a child
   Schulz, Wu, Ruggeri & Meder (*PsychSci* 2019)
   Meder, Wu, Schulz & Ruggeri (*Dev Sci* in press)

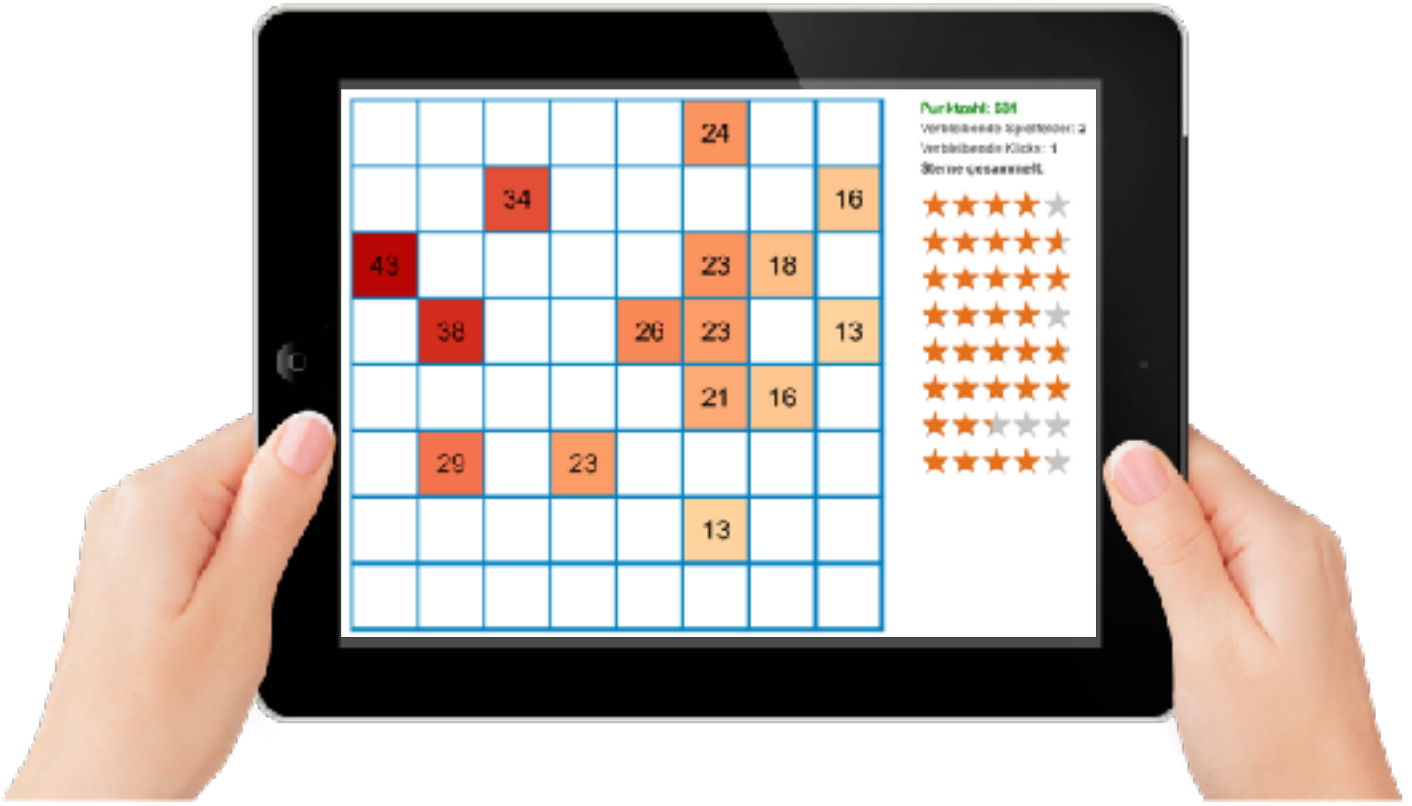
3. Graph-structured Generalization
   Wu, Schulz & Gershman (*Cogsci* 2019)
   Wu, Schulz & Gershman (*CCN* 2019)
   Wu, Schulz & Gershman (*Comput Brain Behav* 2020)

4. Search in abstract conceptual spaces
   Wu, Schulz, Garvert, Meder & Schuck (*Cogsci* 2018)
   Wu, Schulz, Garvert, Meder & Schuck (*PLOS Comp Bio* 2020)

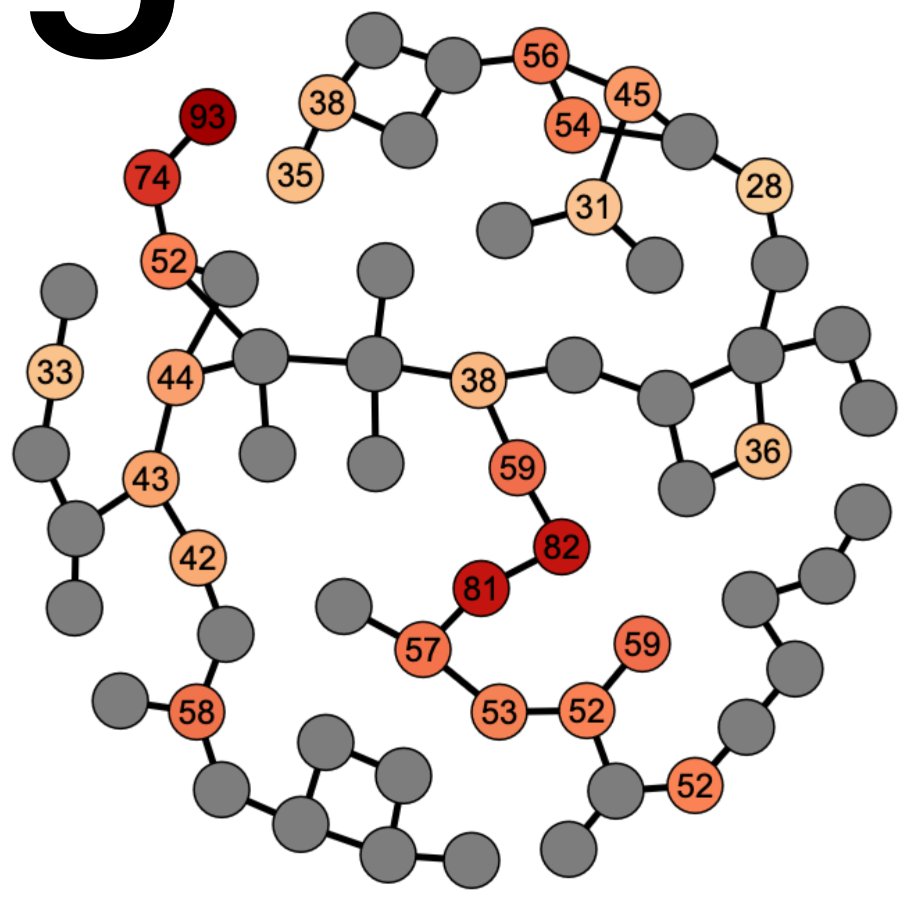
5. Safe exploration
   Schulz, Wu, Huys, Krause & Speekenbrink (*Cognitive Science* 2018)

6. Clinically depressed populations
   Schefft, Wu, Meder, Köhler & Schulz (*in prep*)

7. Social search in VR
   Wu, Ho, Kahl, Leuker, Meder & Kurvers (*bioRxiv* 2021)



Schulz, Wu, Ruggeri & Meder (*PsychSci* 2019)



Wu, Schulz & Gershman (*CBB* 2020)

Establishing a new paradigm

1. Generalization guides exploration
   Wu, Schulz, Nelson, Speekenbrink & Meder (*Cogsci* 2017)
   Wu, Schulz, Nelson, Speekenbrink & Meder (*Nature Human Behaviour* 2018)

2. Learning like a child
   Schulz, Wu, Ruggeri & Meder (*PsychSci* 2019)
   Meder, Wu, Schulz & Ruggeri (*Dev Sci* in press)

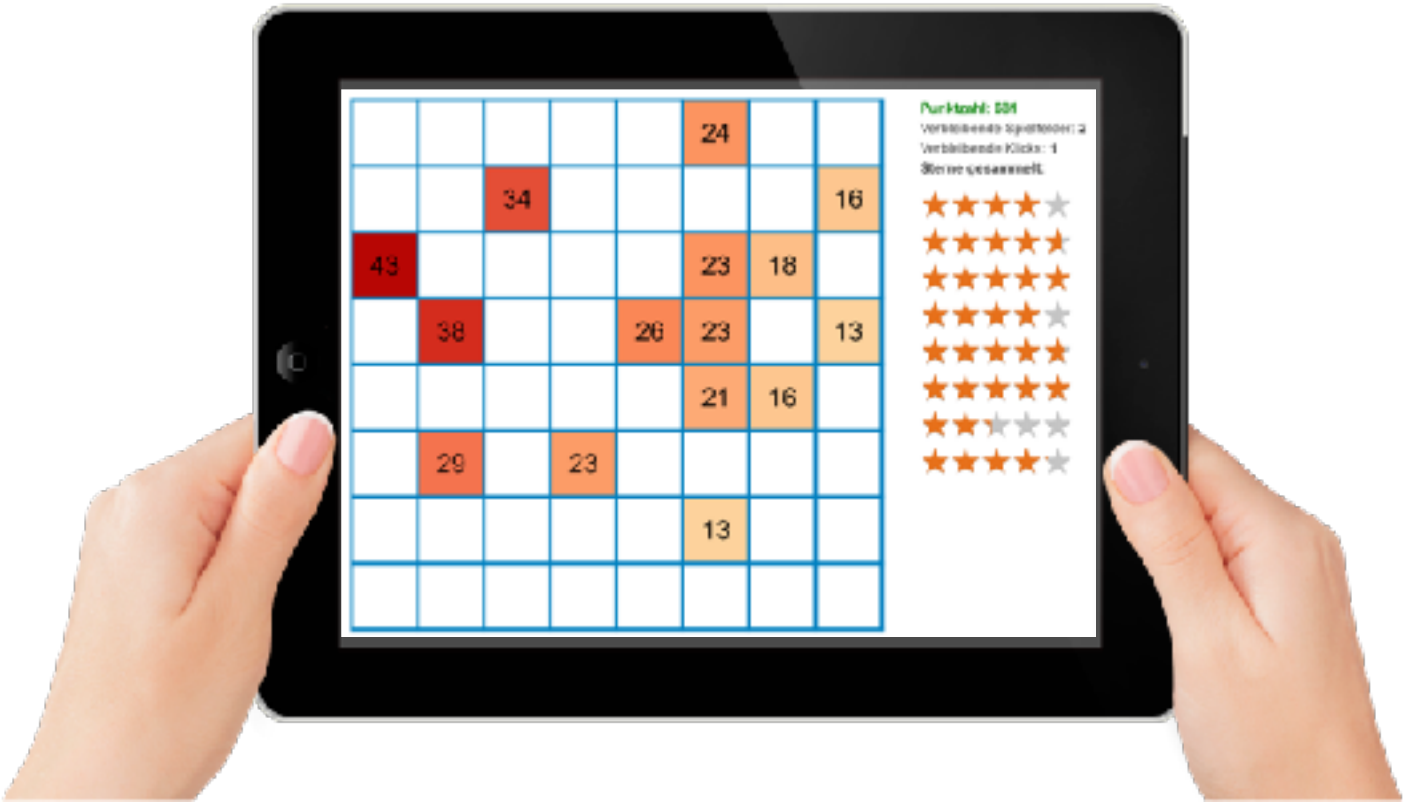
3. Graph-structured Generalization
   Wu, Schulz & Gershman (*Cogsci* 2019)
   Wu, Schulz & Gershman (*CCN* 2019)
   Wu, Schulz & Gershman (*Comput Brain Behav* 2020)

4. Search in abstract conceptual spaces
   Wu, Schulz, Garvert, Meder & Schuck (*Cogsci* 2018)
   Wu, Schulz, Garvert, Meder & Schuck (*PLOS Comp Bio* 2020)

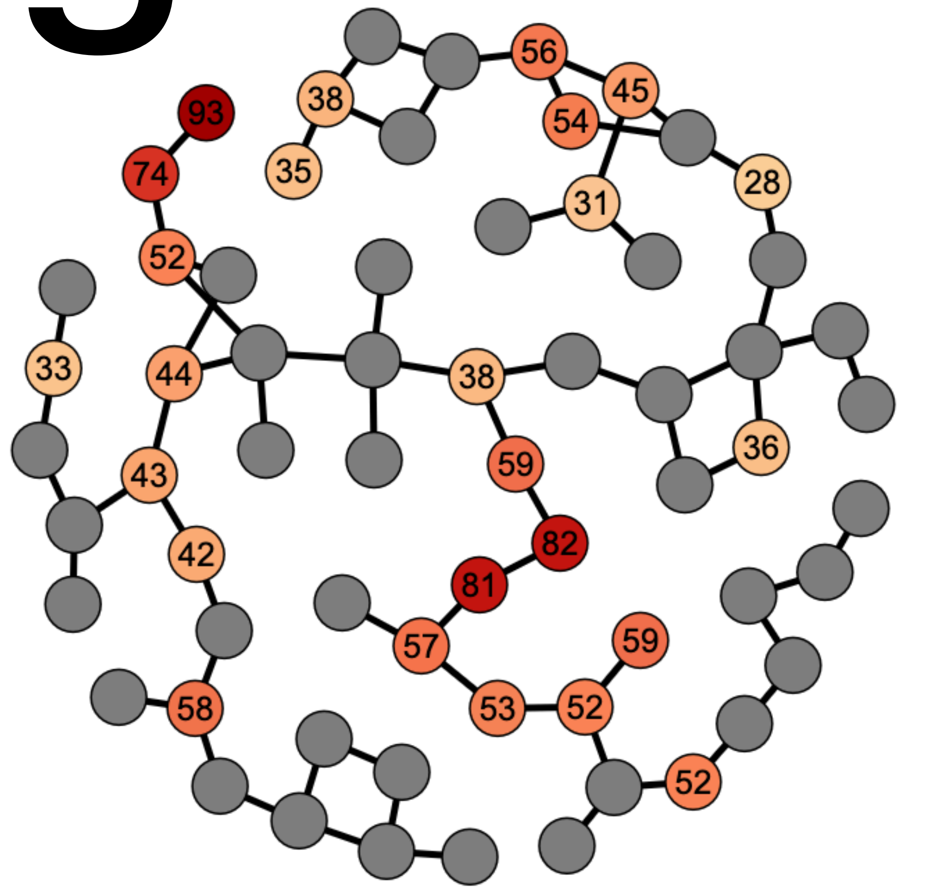
5. Safe exploration
   Schulz, Wu, Huys, Krause & Speekenbrink (*Cognitive Science* 2018)

6. Clinically depressed populations
   Schefft, Wu, Meder, Köhler & Schulz (*in prep*)

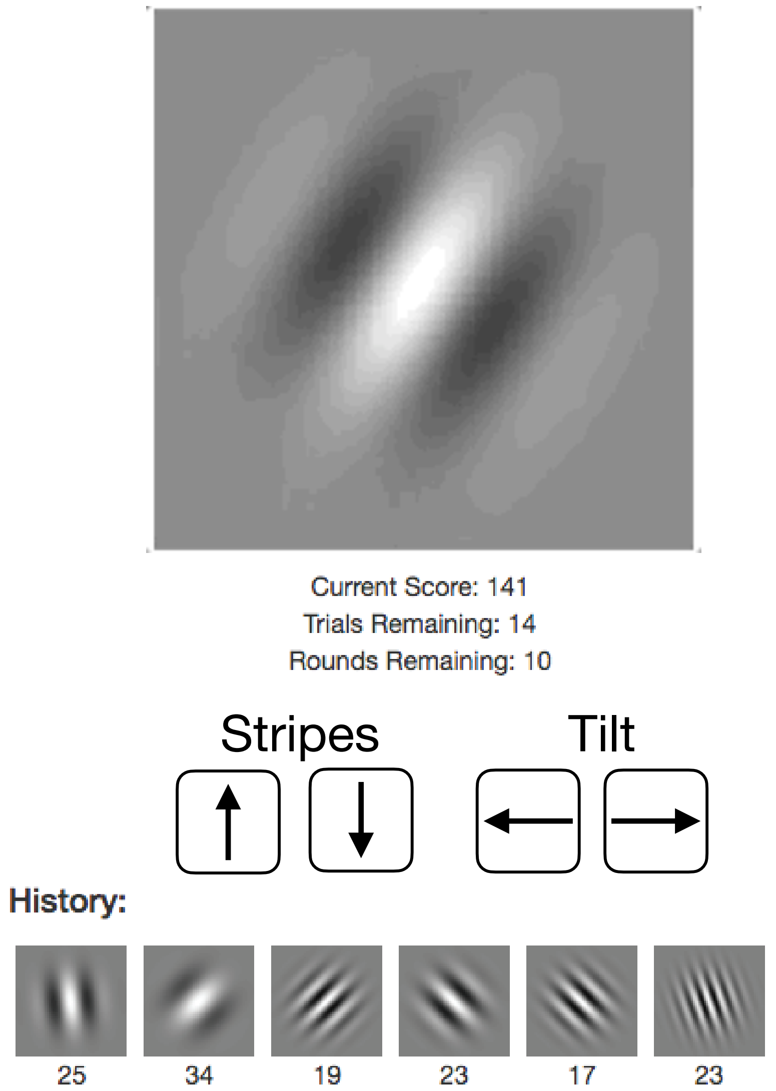
7. Social search in VR
   Wu, Ho, Kahl, Leuker, Meder & Kurvers (*bioRxiv* 2021)



Schulz, Wu, Ruggeri & Meder (*PsychSci* 2019)



Wu, Schulz & Gershman (*CBB* 2020)



Wu, Schulz, Garvert, Meder & Schuck (*PLOS Comp Bio* 2020)

Establishing a new paradigm

1. Generalization guides exploration
   Wu, Schulz, Nelson, Speekenbrink & Meder (*Cogsci* 2017)
   Wu, Schulz, Nelson, Speekenbrink & Meder (*Nature Human Behaviour* 2018)

2. Learning like a child
   Schulz, Wu, Ruggeri & Meder (*PsychSci* 2019)
   Meder, Wu, Schulz & Ruggeri (*Dev Sci* in press)

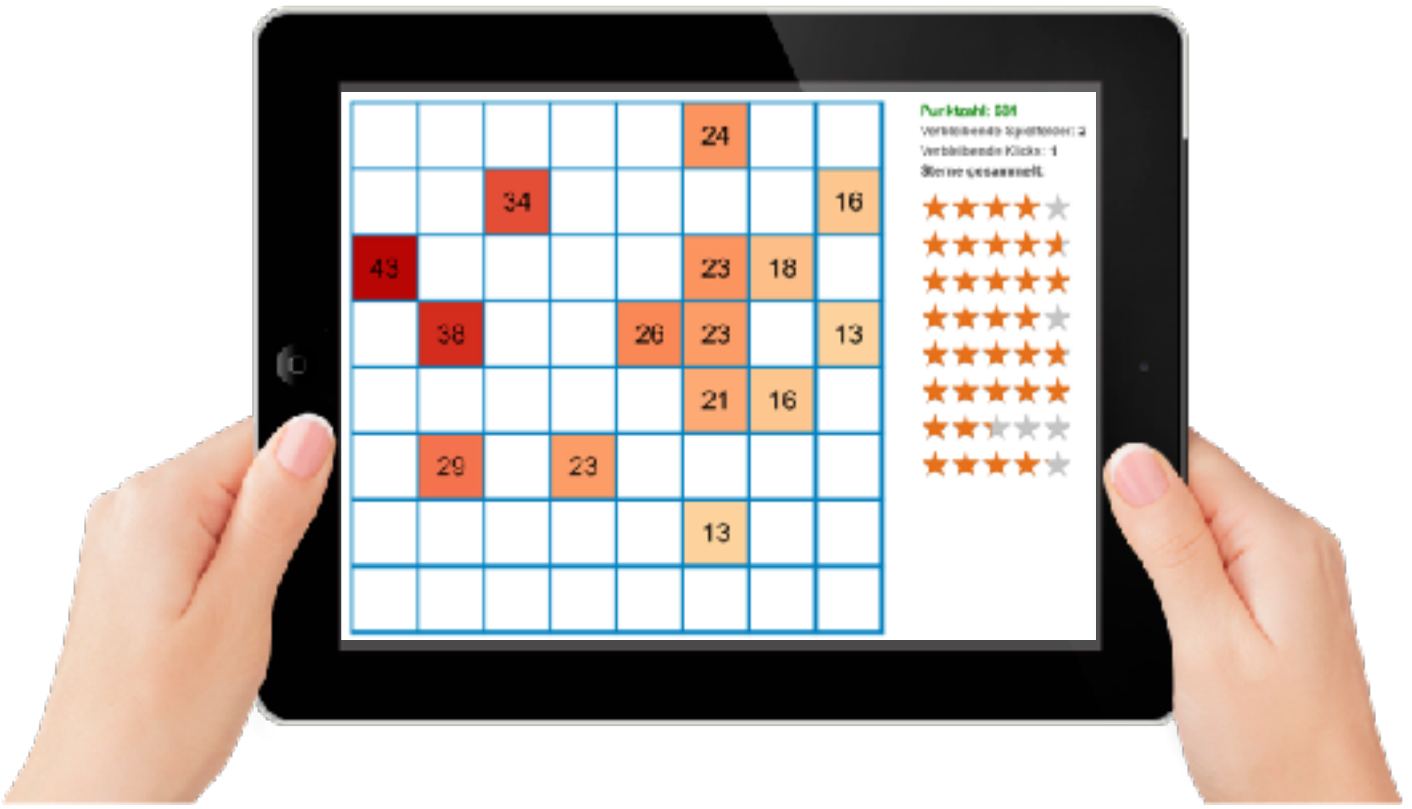
3. Graph-structured Generalization
   Wu, Schulz & Gershman (*Cogsci* 2019)
   Wu, Schulz & Gershman (*CCN* 2019)
   Wu, Schulz & Gershman (*Comput Brain Behav* 2020)

4. Search in abstract conceptual spaces
   Wu, Schulz, Garvert, Meder & Schuck (*Cogsci* 2018)
   Wu, Schulz, Garvert, Meder & Schuck (*PLOS Comp Bio* 2020)

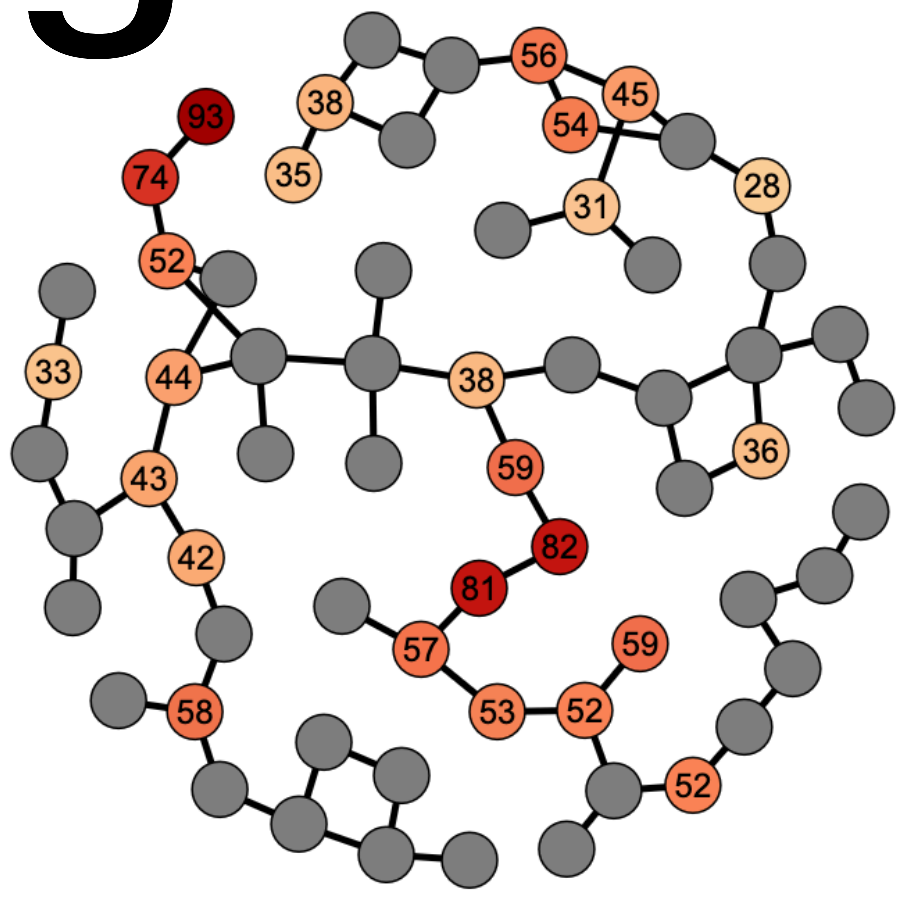
5. Safe exploration
   Schulz, Wu, Huys, Krause & Speekenbrink (*Cognitive Science* 2018)

6. Clinically depressed populations
   Schefft, Wu, Meder, Köhler & Schulz (*in prep*)

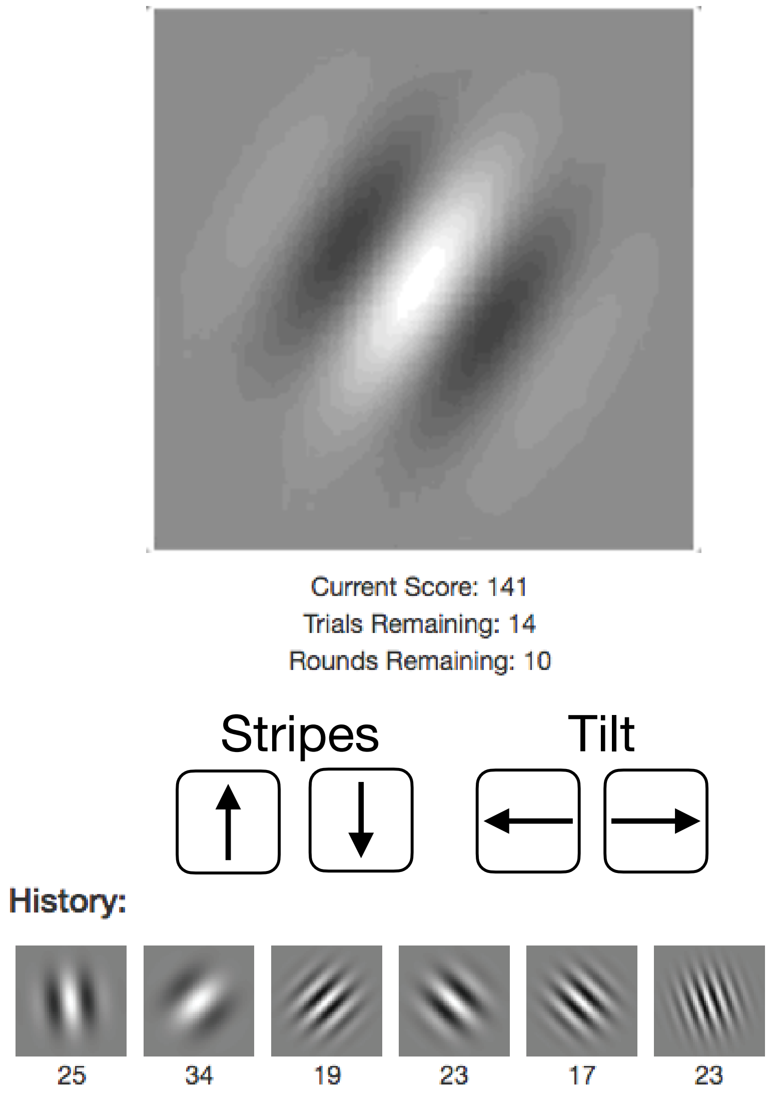
7. Social search in VR
   Wu, Ho, Kahl, Leuker, Meder & Kurvers (*bioRxiv* 2021)



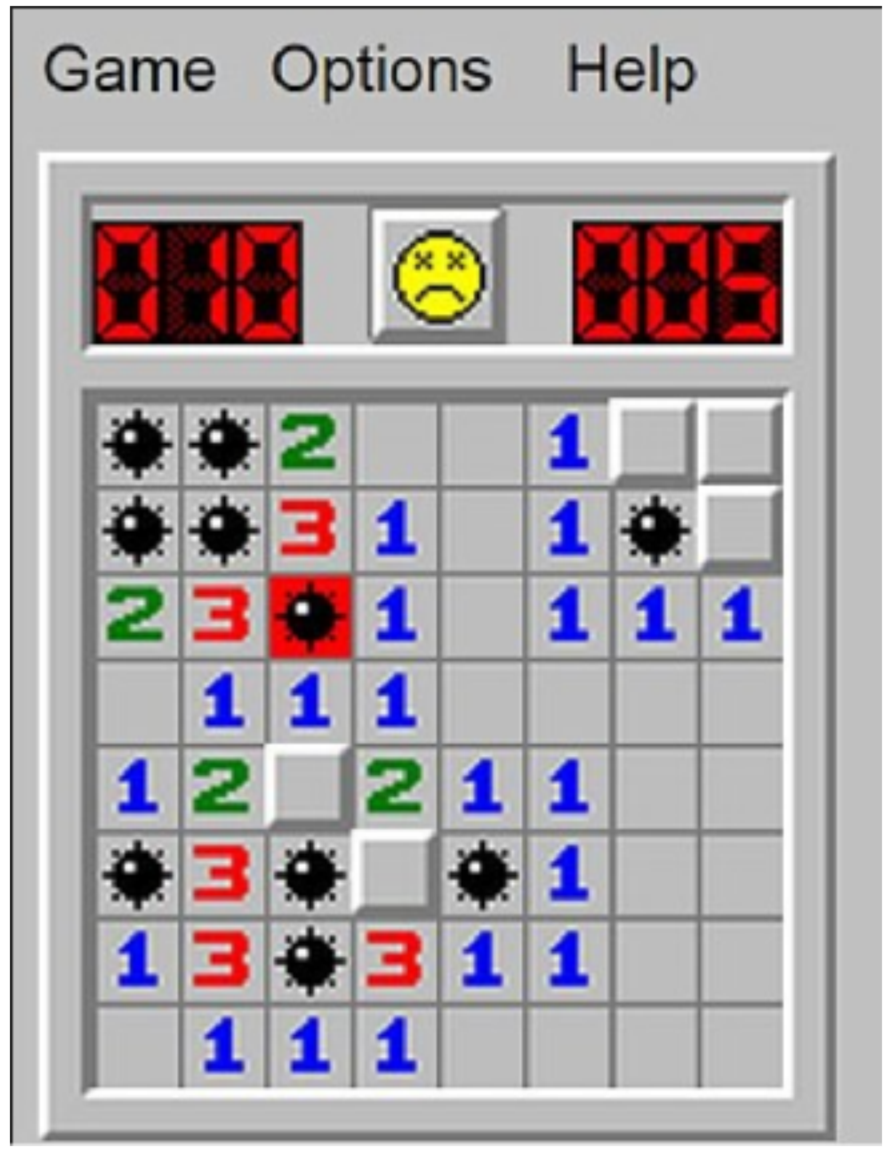
Schulz, Wu, Ruggeri & Meder (*PsychSci* 2019)



Wu, Schulz & Gershman (*CBB* 2020)



Wu, Schulz, Garvert, Meder & Schuck (*PLOS Comp Bio* 2020)



Schulz, Wu, Huys, Krause & Speekenbrink (*Cognitive Science* 2018)

https://github.com/charleywu/gridsearch

Establishing a new paradigm

1. Generalization guides exploration
   Wu, Schulz, Nelson, Speekenbrink & Meder (*Cogsci* 2017)
   Wu, Schulz, Nelson, Speekenbrink & Meder (*Nature Human Behaviour* 2018)

2. Learning like a child
   Schulz, Wu, Ruggeri & Meder (*PsychSci* 2019)
   Meder, Wu, Schulz & Ruggeri (*Dev Sci* in press)

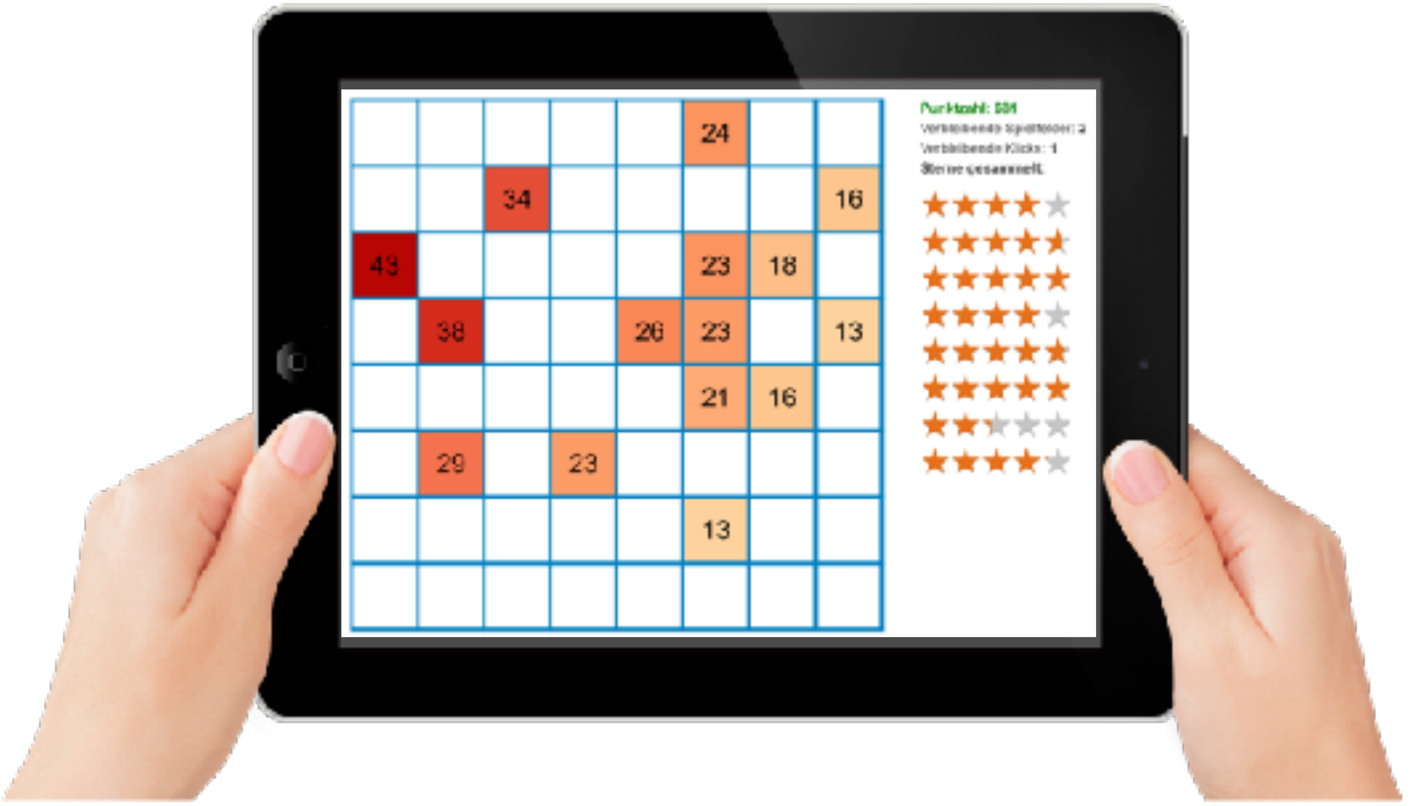
3. Graph-structured Generalization
   Wu, Schulz & Gershman (*Cogsci* 2019)
   Wu, Schulz & Gershman (*CCN* 2019)
   Wu, Schulz & Gershman (*Comput Brain Behav* 2020)

4. Search in abstract conceptual spaces
   Wu, Schulz, Garvert, Meder & Schuck (*Cogsci* 2018)
   Wu, Schulz, Garvert, Meder & Schuck (*PLOS Comp Bio* 2020)

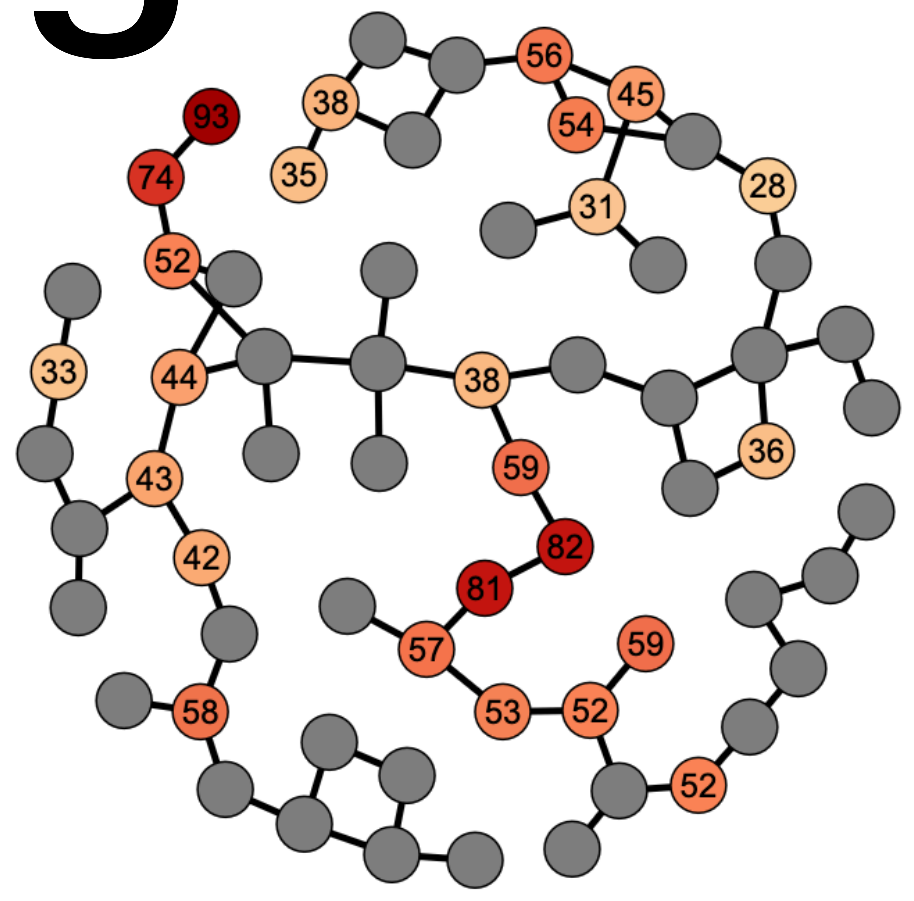
5. Safe exploration
   Schulz, Wu, Huys, Krause & Speekenbrink (*Cognitive Science* 2018)

6. Clinically depressed populations
   Schefft, Wu, Meder, Köhler & Schulz (*in prep*)

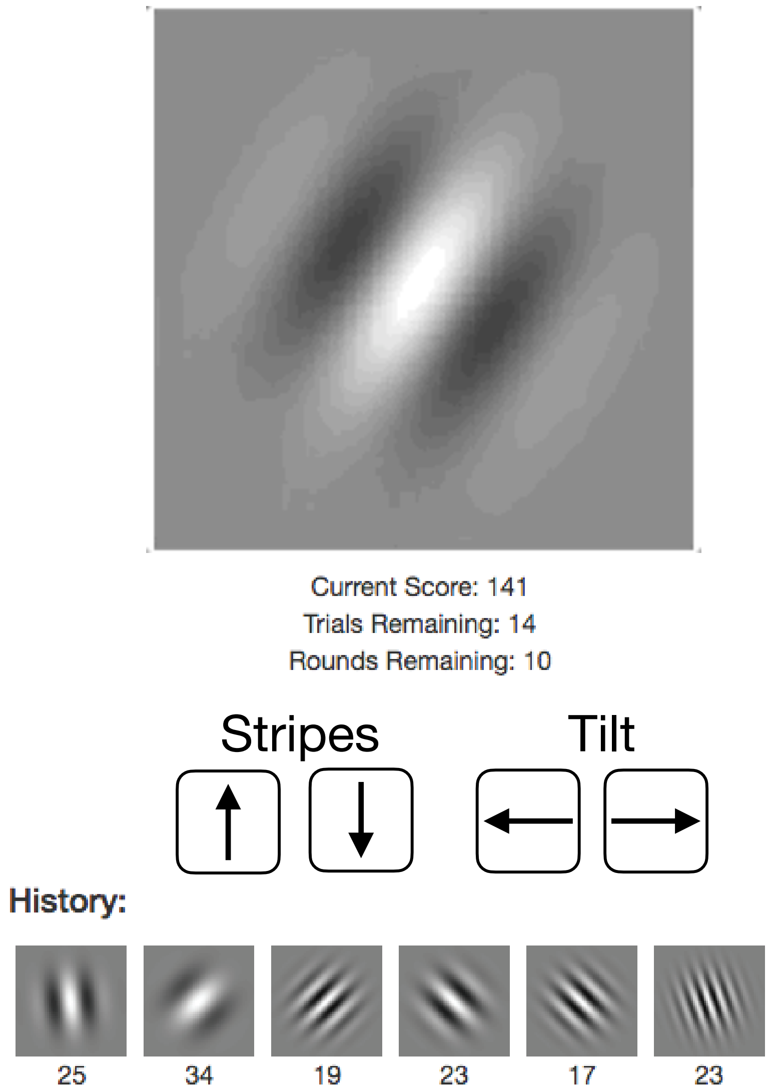
7. Social search in VR
   Wu, Ho, Kahl, Leuker, Meder & Kurvers (*bioRxiv* 2021)



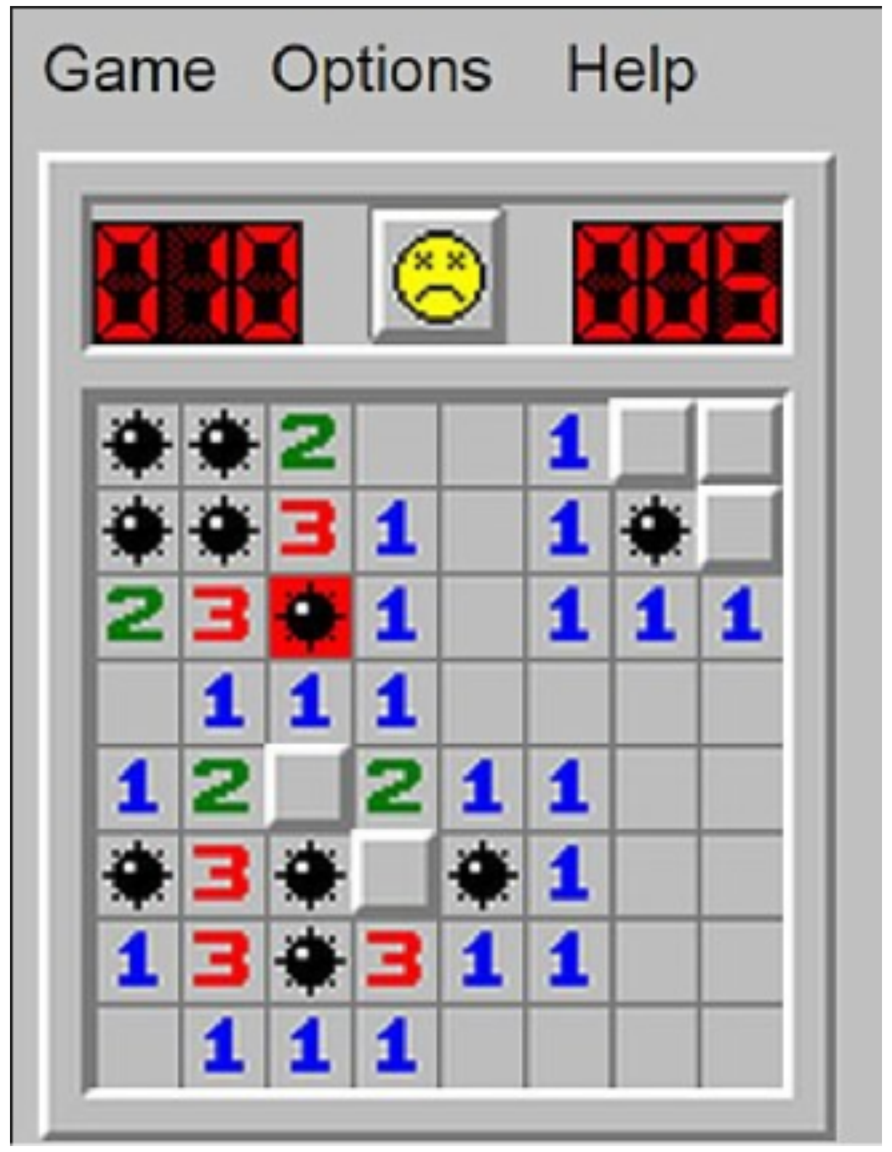
Schulz, Wu, Ruggeri & Meder (*PsychSci* 2019)



Wu, Schulz & Gershman (*CBB* 2020)



Wu, Schulz, Garvert, Meder & Schuck (*PLOS Comp Bio* 2020)



Schulz, Wu, Huys, Krause & Speekenbrink (*Cognitive Science* 2018)

https://github.com/charleywu/gridsearch

Establishing a new paradigm

1. Generalization guides exploration
   Wu, Schulz, Nelson, Speekenbrink & Meder (*Cogsci* 2017)
   Wu, Schulz, Nelson, Speekenbrink & Meder (*Nature Human Behaviour* 2018)

2. Learning like a child
   Schulz, Wu, Ruggeri & Meder (*PsychSci* 2019)
   Meder, Wu, Schulz & Ruggeri (*Dev Sci* in press)

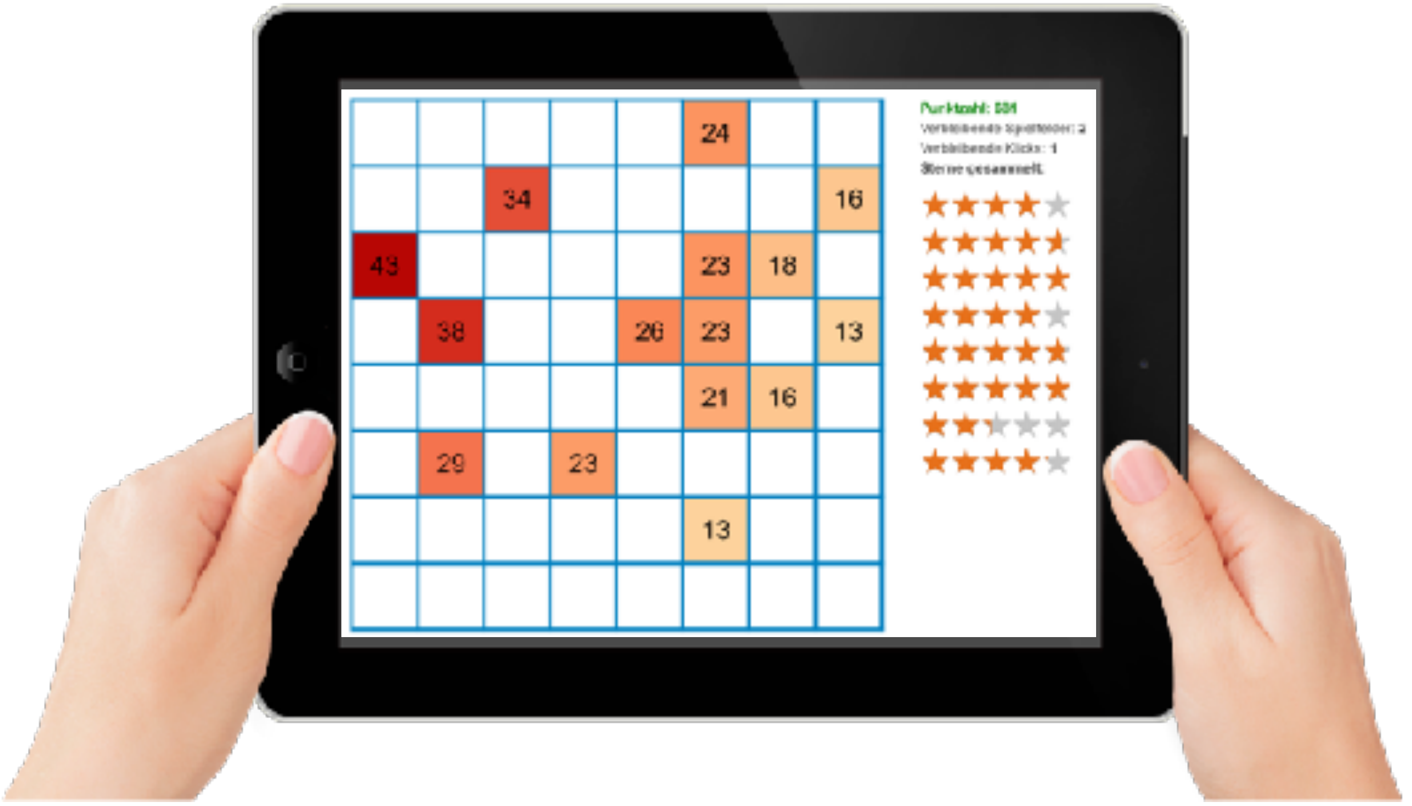
3. Graph-structured Generalization
   Wu, Schulz & Gershman (*Cogsci* 2019)
   Wu, Schulz & Gershman (*CCN* 2019)
   Wu, Schulz & Gershman (*Comput Brain Behav* 2020)

4. Search in abstract conceptual spaces
   Wu, Schulz, Garvert, Meder & Schuck (*Cogsci* 2018)
   Wu, Schulz, Garvert, Meder & Schuck (*PLOS Comp Bio* 2020)

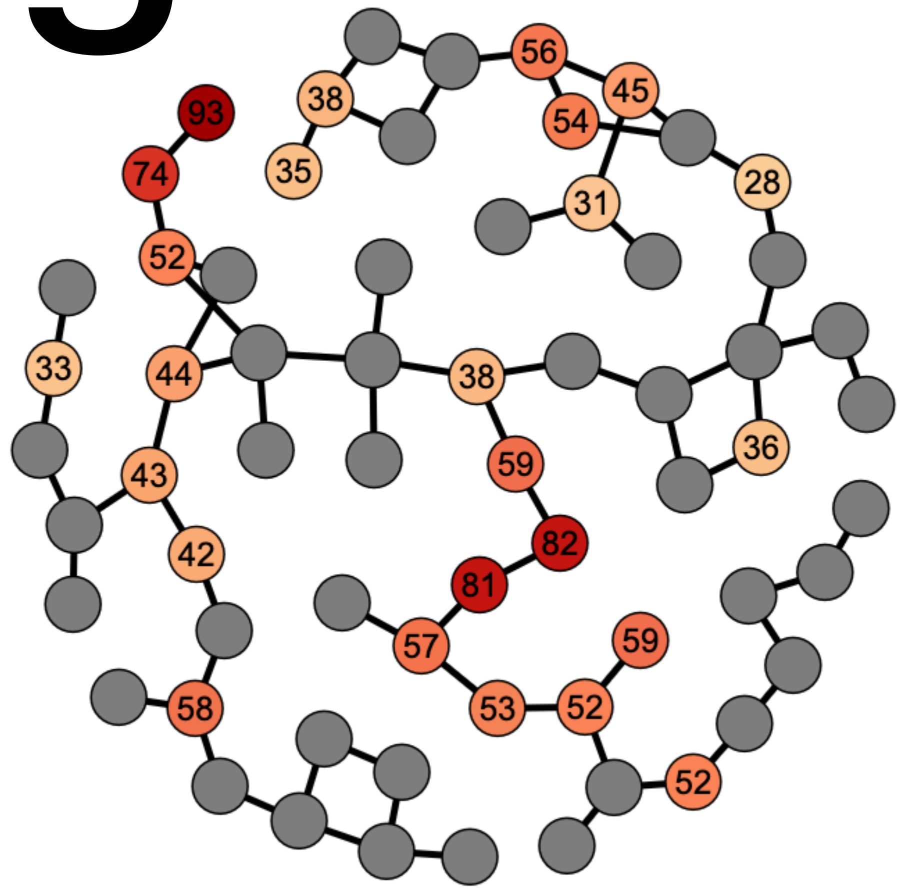
5. Safe exploration
   Schulz, Wu, Huys, Krause & Speekenbrink (*Cognitive Science* 2018)

6. Clinically depressed populations
   Schefft, Wu, Meder, Köhler & Schulz (*in prep*)

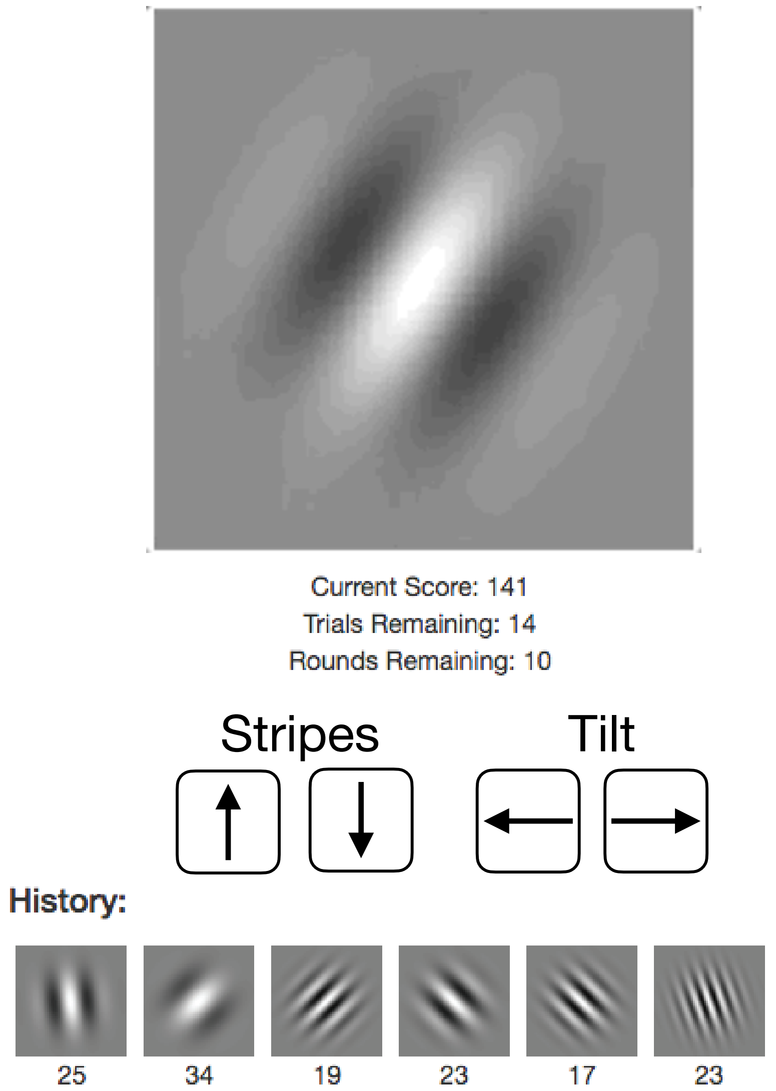
7. Social search in VR
   Wu, Ho, Kahl, Leuker, Meder & Kurvers (*bioRxiv* 2021)



Schulz, Wu, Ruggeri & Meder (*PsychSci* 2019)



Wu, Schulz & Gershman (*CBB* 2020)



Wu, Schulz, Garvert, Meder & Schuck (*PLOS Comp Bio* 2020)

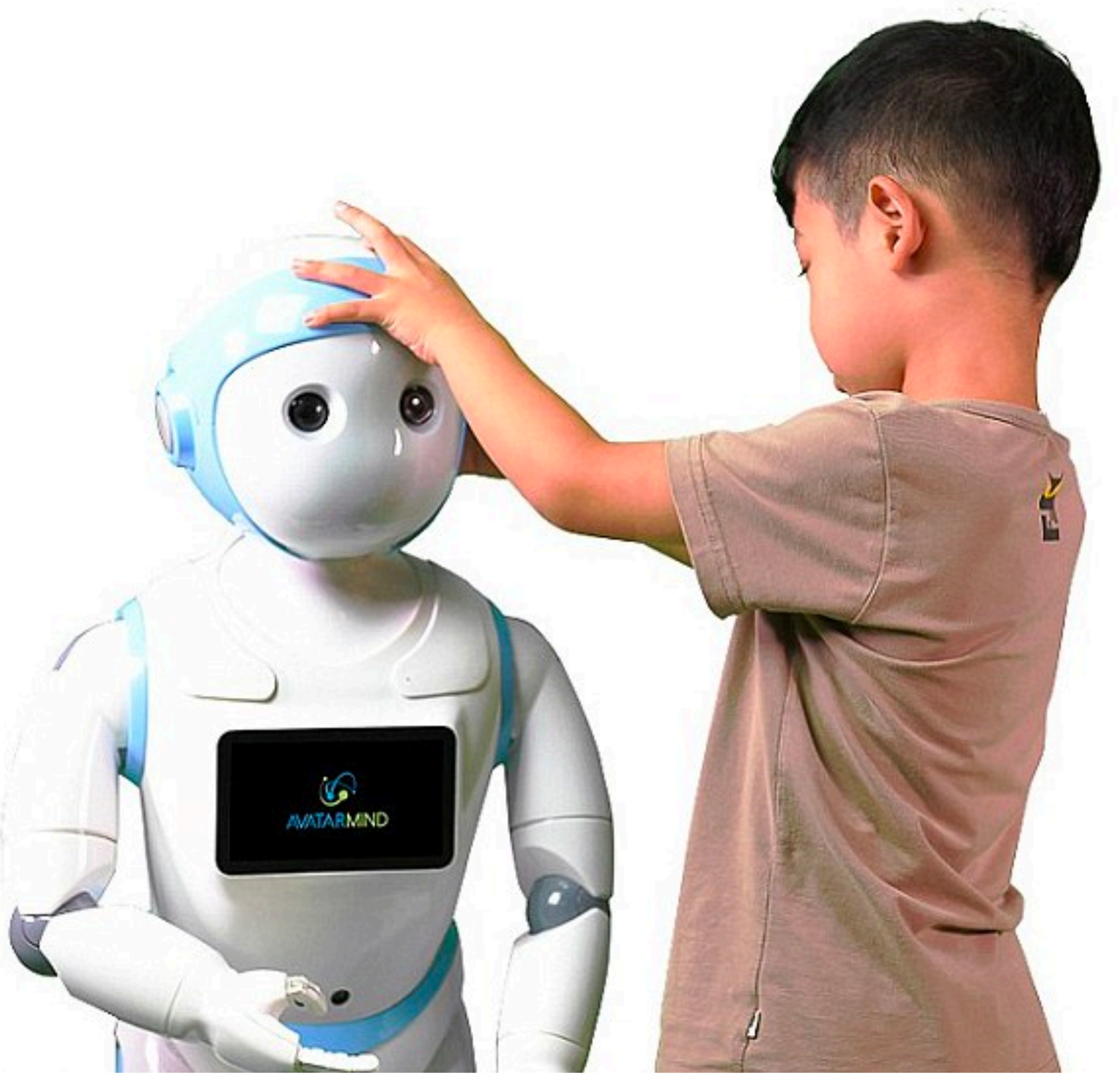
Schulz, Wu, Huys, Krause & Speekenbrink (*Cognitive Science* 2018)

Part 2

How to learn like a child

What AI can learn from Children

- Josh Tenenbaum (MIT): Children are the only known information processing system that demonstrably and reproducibly develop into intelligent systems
- Turing (1950) suggested we should build AI that learns like a child
- How do children learn differently from adults?



What AI can learn from Children

- Josh Tenenbaum (MIT): Children are the only known information processing system that demonstrably and reproducibly develop into intelligent systems

- Turing (1950) suggested we should build AI that learns like a child

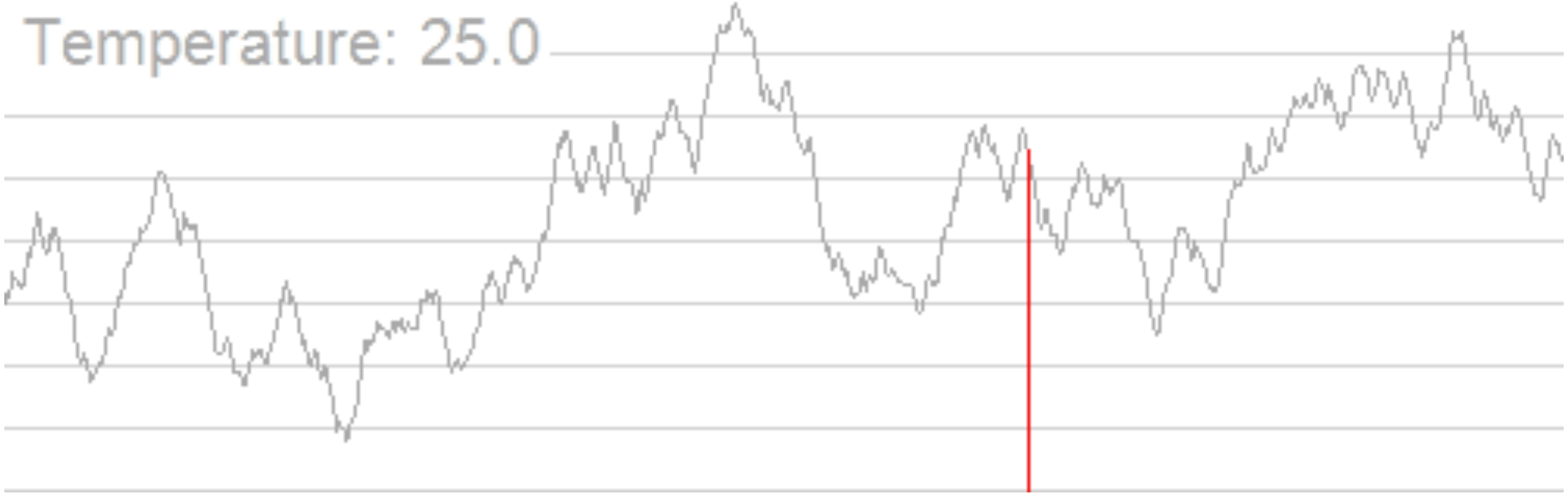
- How do children learn differently from adults?

- One robust finding is they are highly variable!

video by Francis Vachon

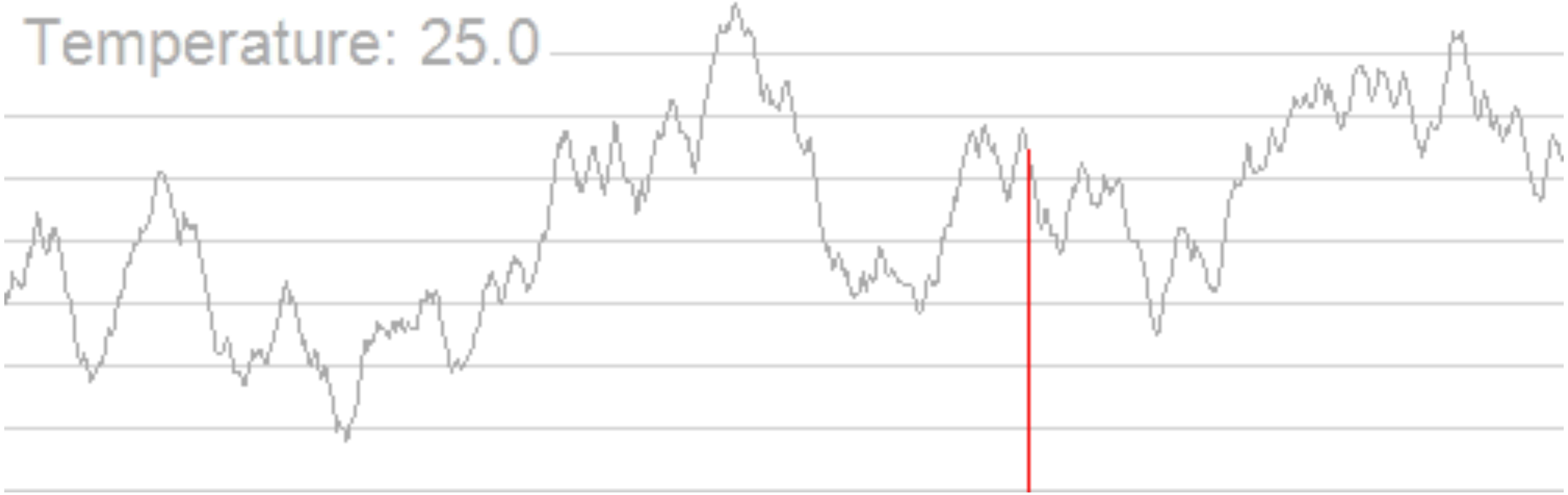
What explains the extensive variability found in children's search behavior?

- High temperature sampling hypothesis:
  - Children initially perform high-temperature search, which gradually "cool offs" as they grow older (Gopnik et al., *Curr Dir Psych Sci* 2017)



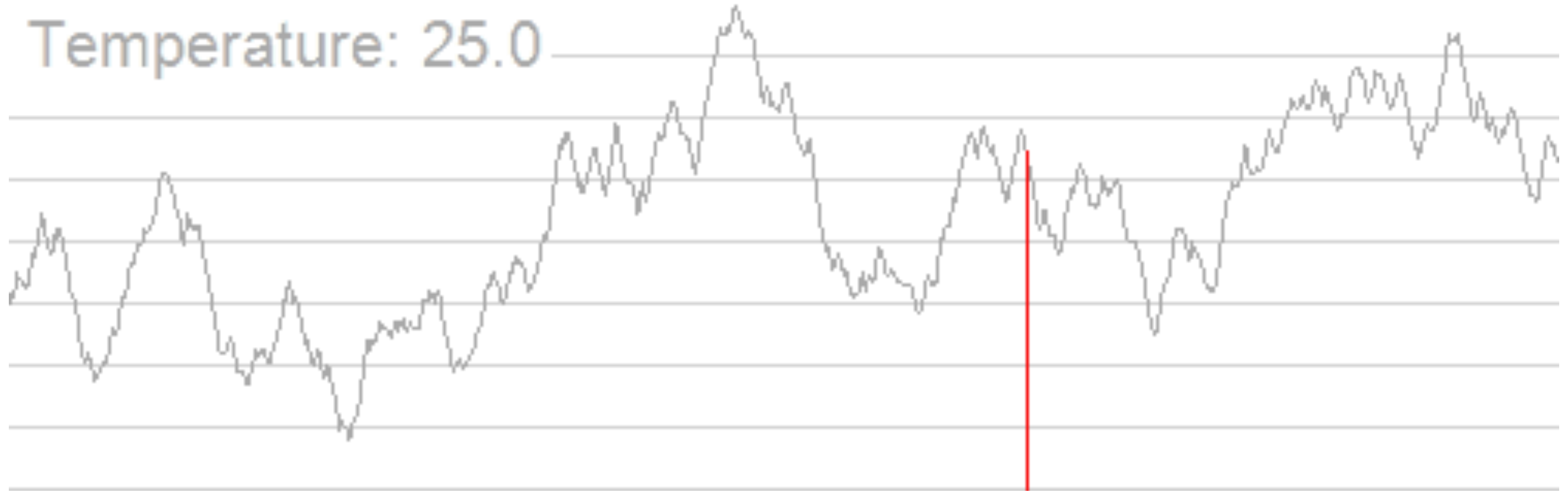
What explains the extensive variability found in children's search behavior?

- High temperature sampling hypothesis:
  - Children initially perform high-temperature search, which gradually "cool offs" as they grow older (Gopnik et al., *Curr Dir Psych Sci* 2017)



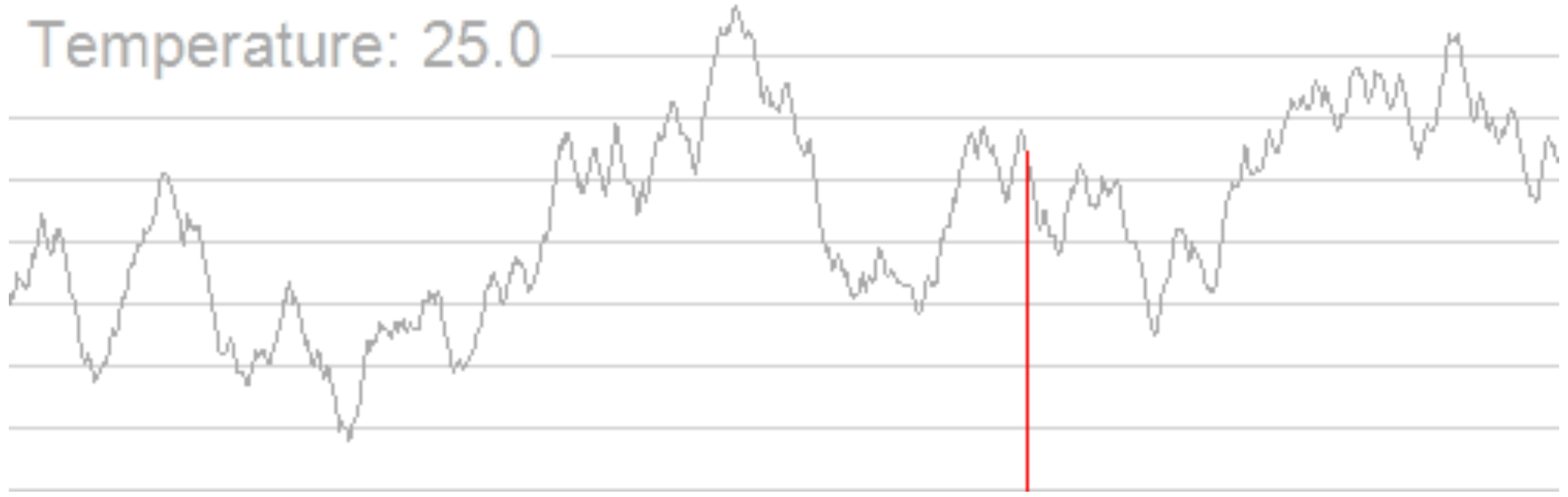
Sources of Developmental Differences

- Higher **temperature** sampling that cools off over age? (Gopnik et al., 2017)



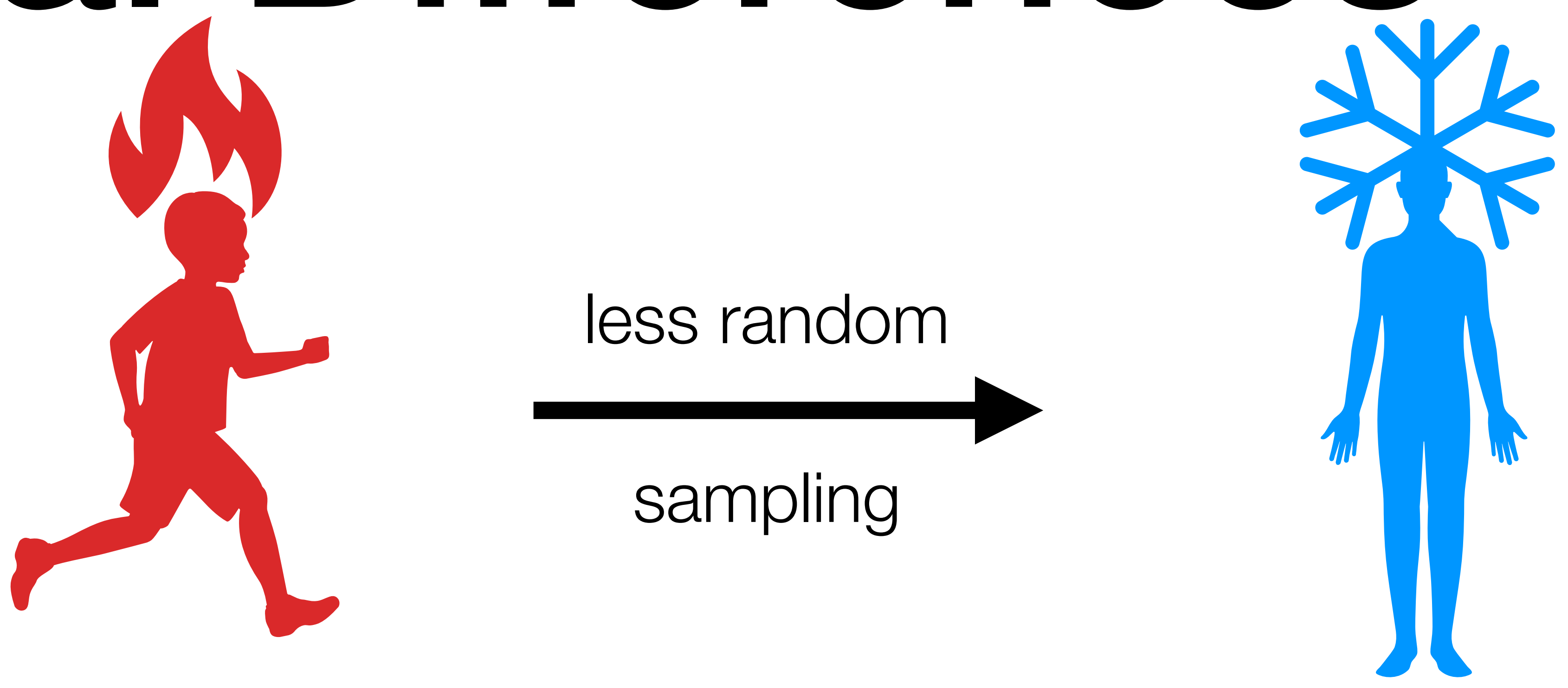
Sources of Developmental Differences

- Higher **temperature** sampling that cools off over age? (Gopnik et al., 2017)



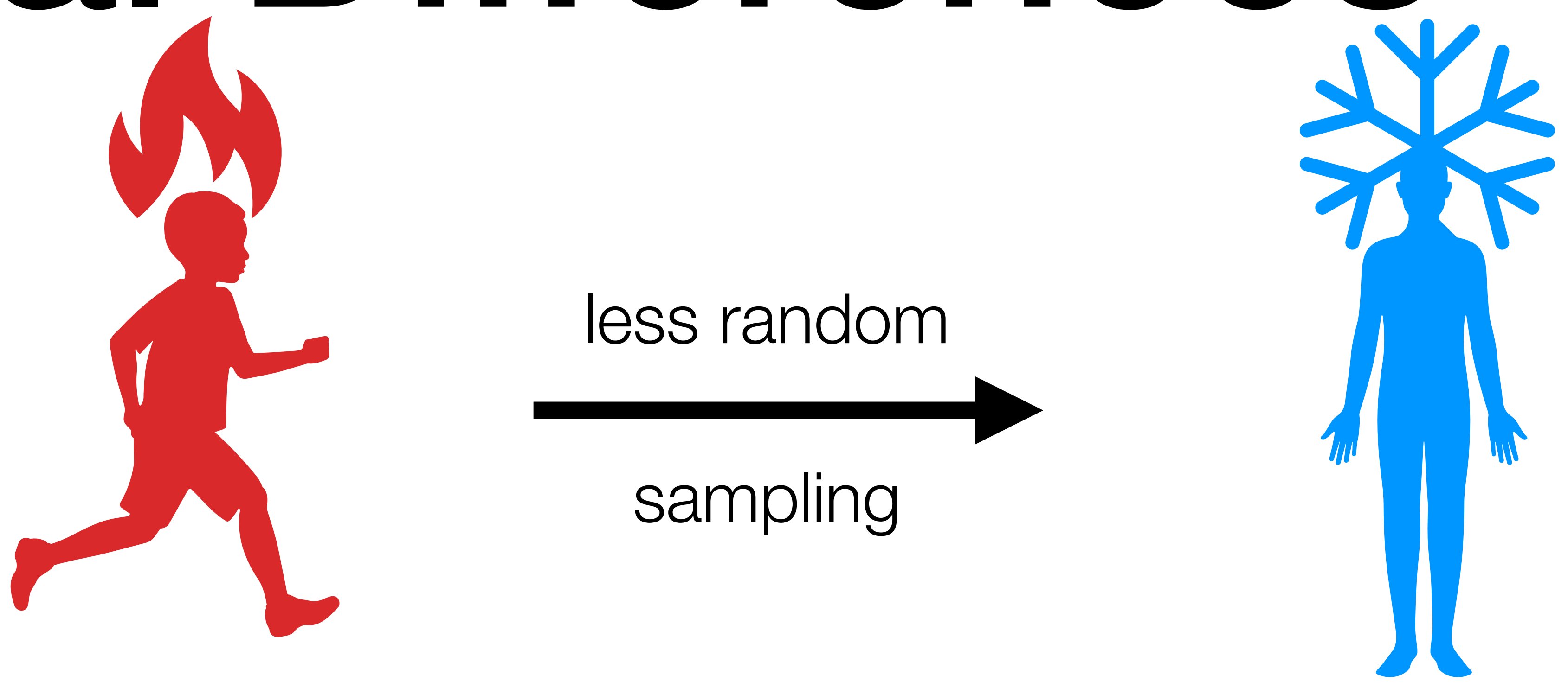
Sources of Developmental Differences

- Higher **temperature** sampling that cools off over age? (Gopnik et al., 2017)

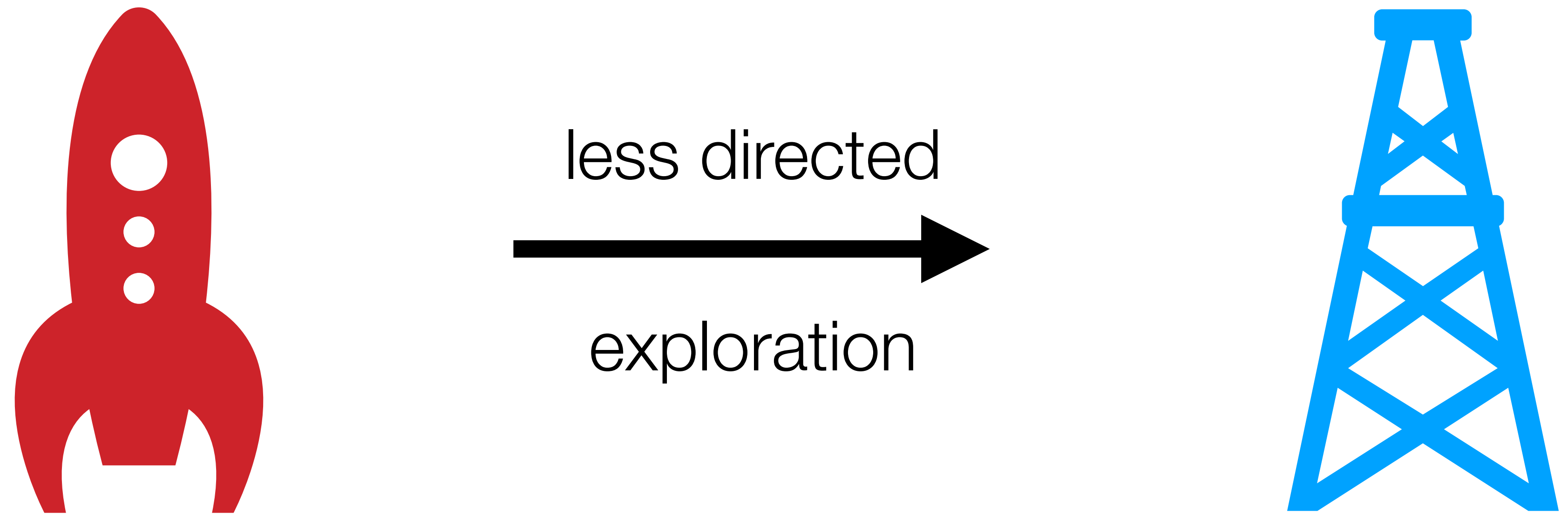


Sources of Developmental Differences

- Higher **temperature** sampling that cools off over age? (Gopnik et al., 2017)

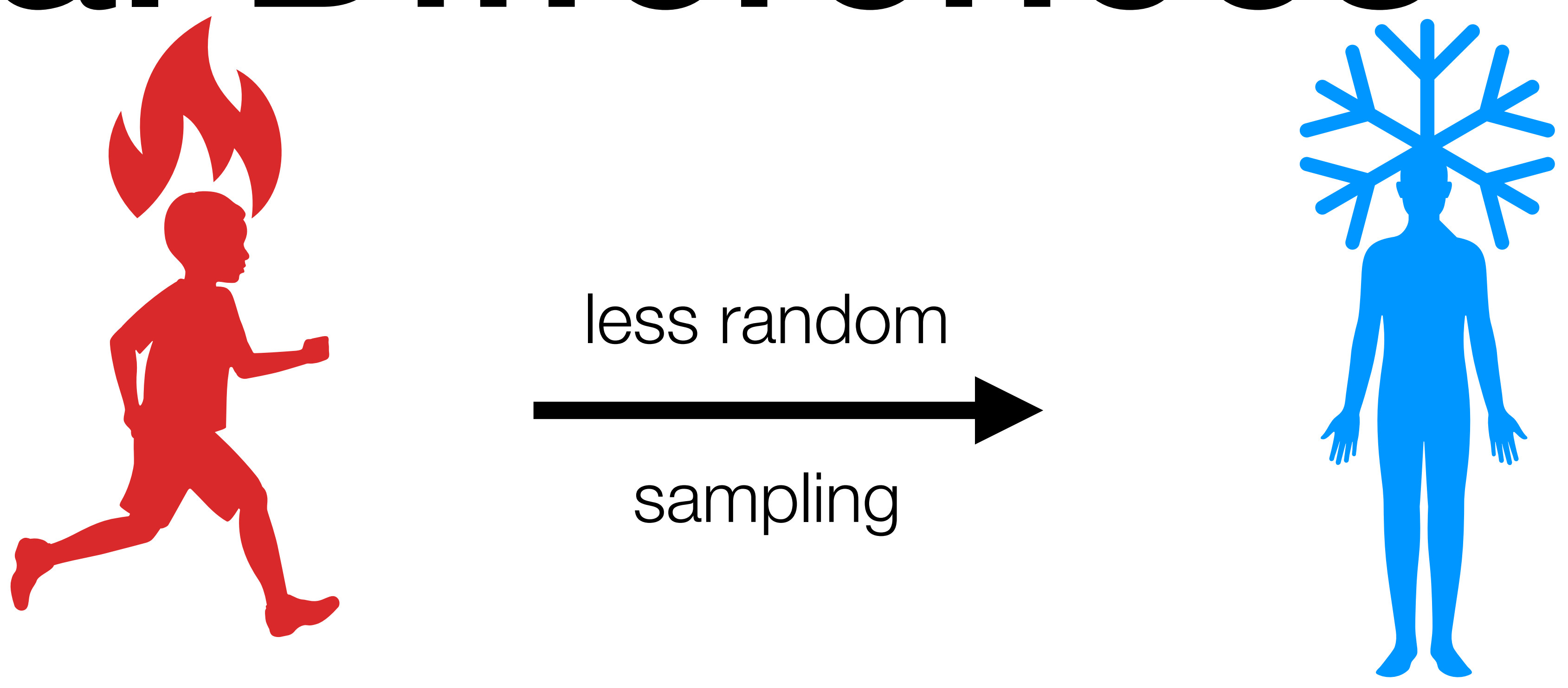


- Changes in **directed exploration** rather than random exploration? (Wilson et al., 2014)

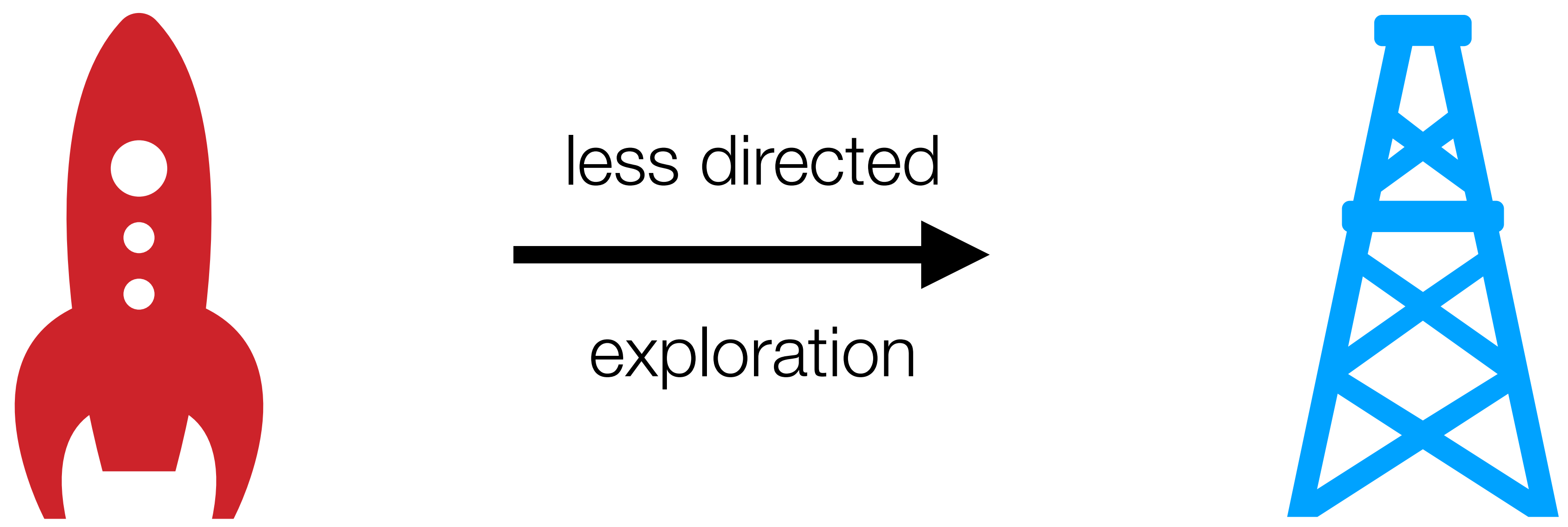


Sources of Developmental Differences

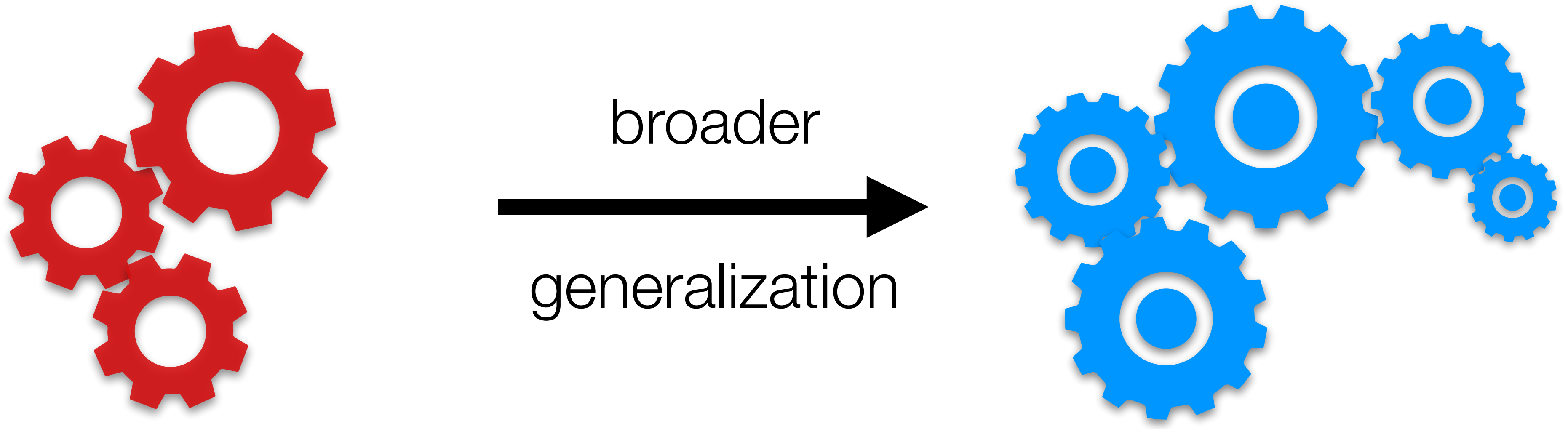
- Higher **temperature** sampling that cools off over age? (Gopnik et al., 2017)



- Changes in **directed exploration** rather than random exploration? (Wilson et al., 2014)

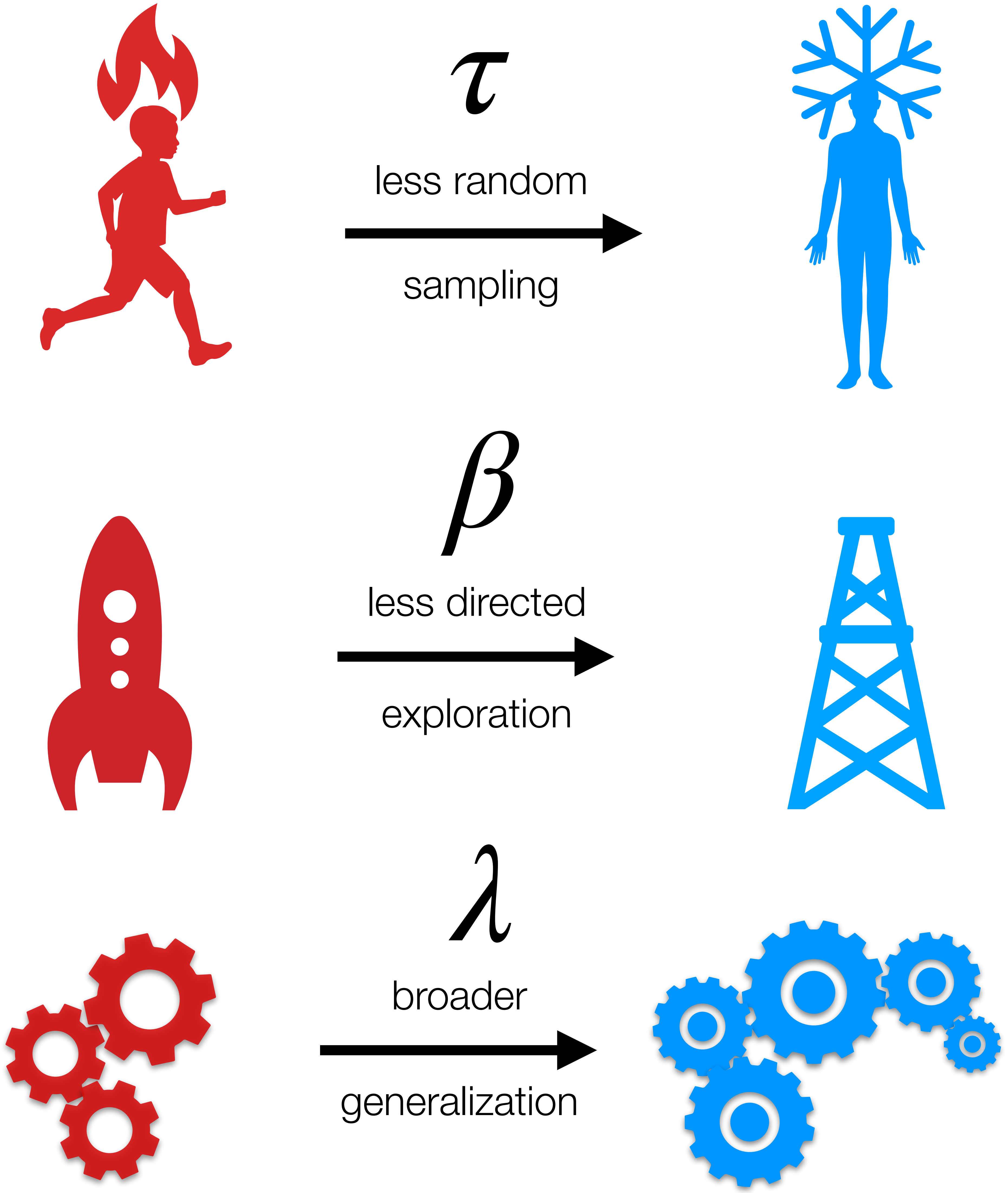


- Refinement of cognitive representations and processes supporting **generalization**? (Blanco et al., 2016)

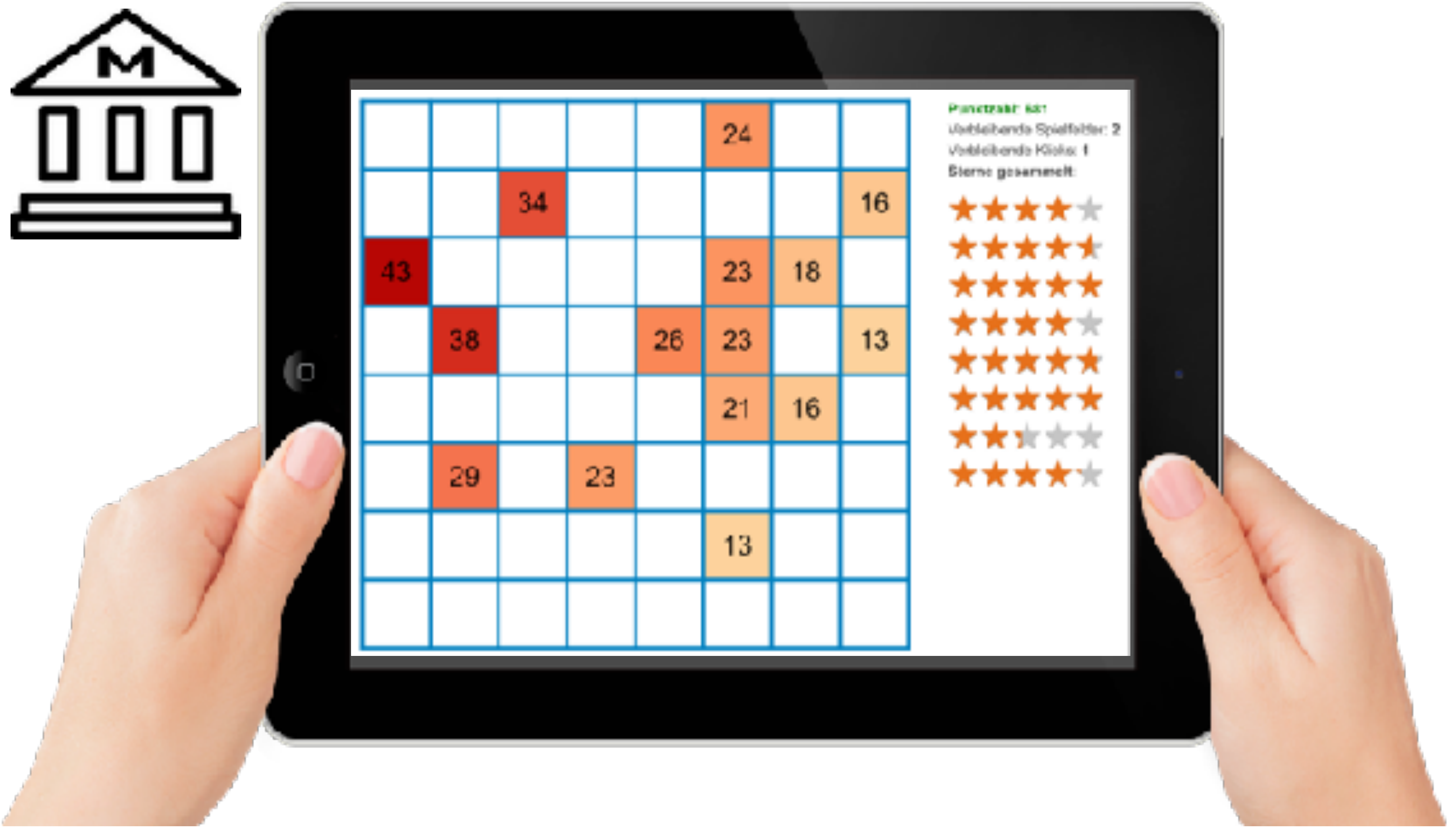


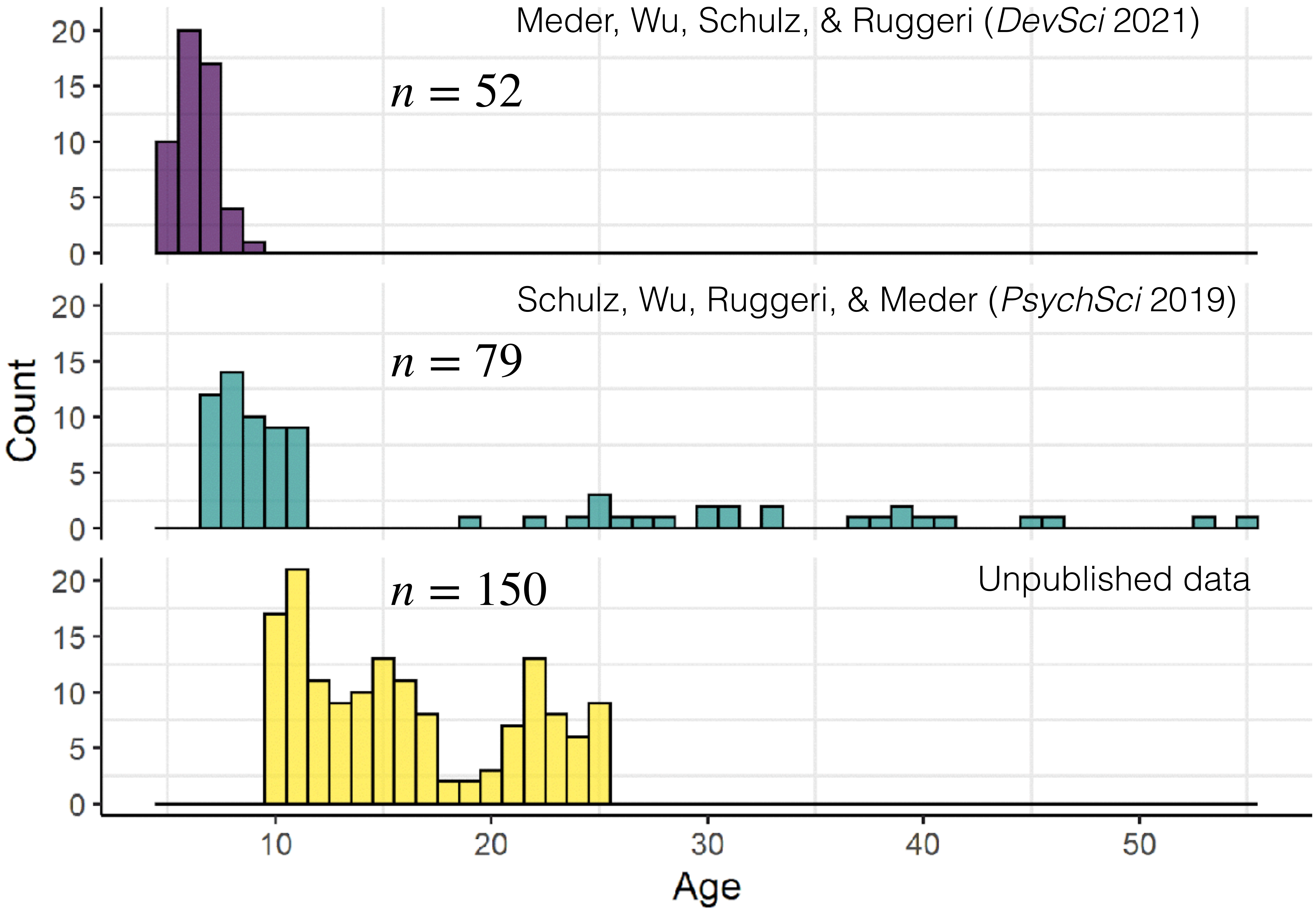
Sources of Developmental Differences

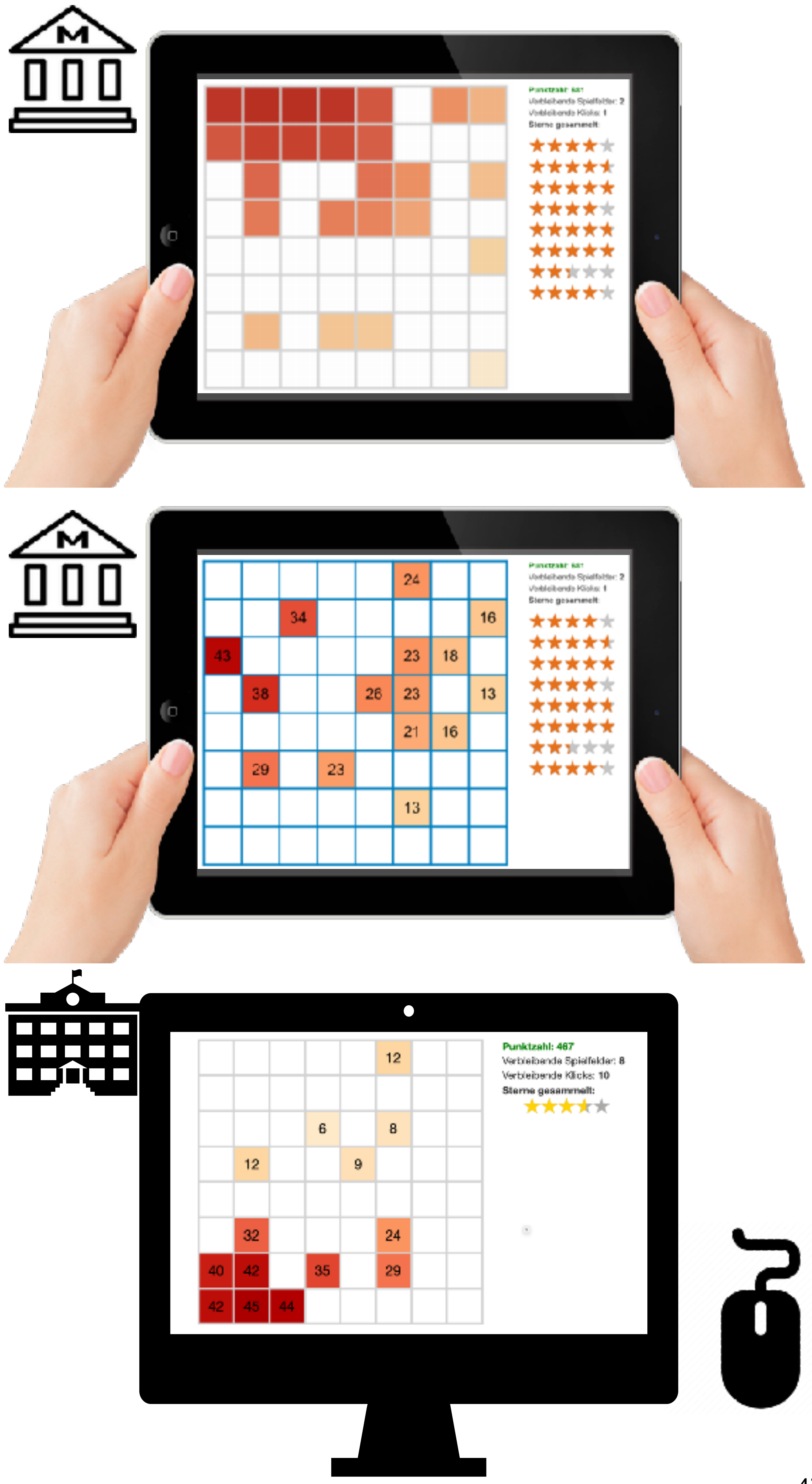
- Higher **temperature** sampling that cools off over age? (Gopnik et al., 2017)
- Changes in **directed exploration** rather than random exploration? (Wilson et al., 2014)
- Refinement of cognitive representations and processes supporting **generalization**? (Blanco et al., 2016)



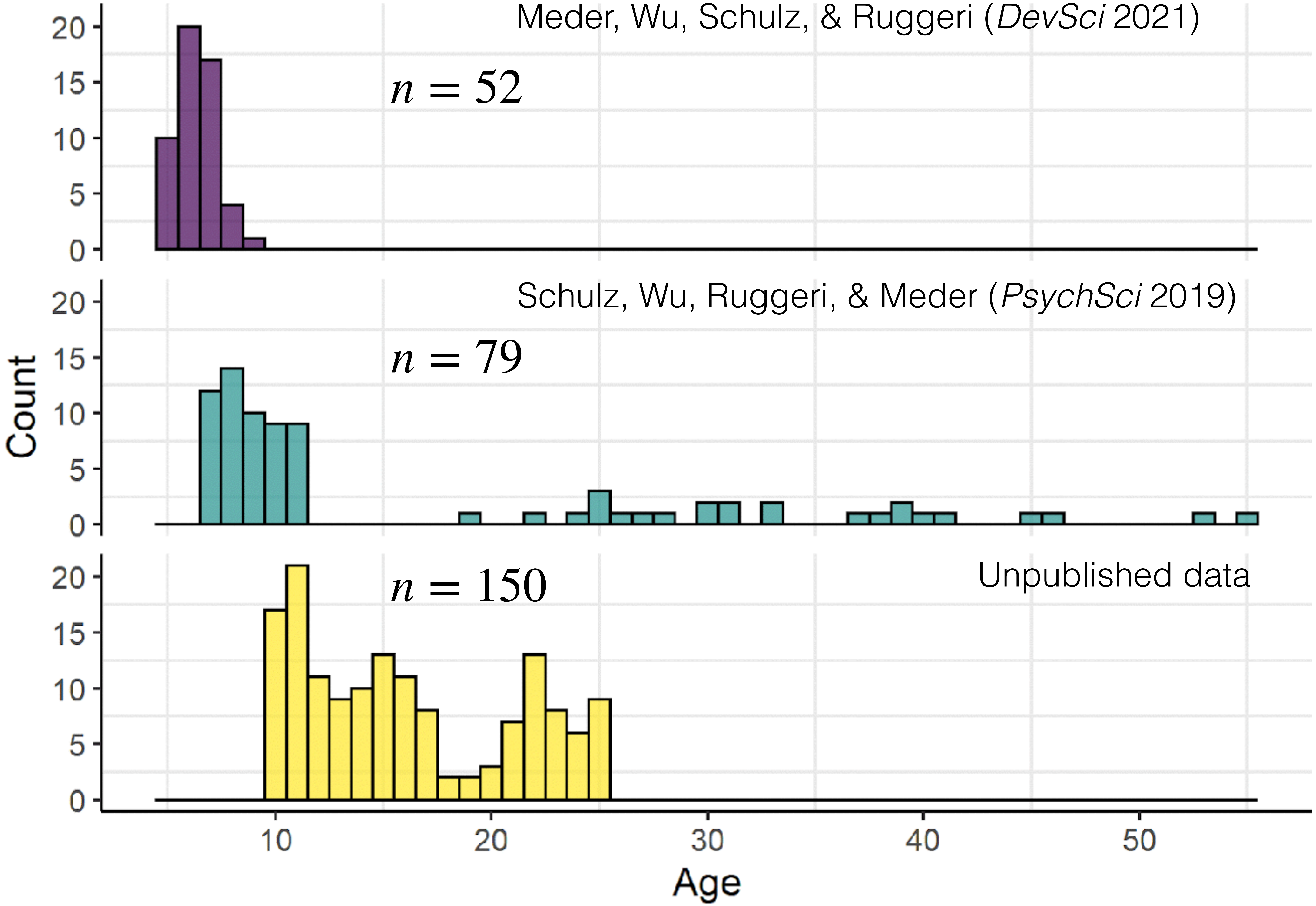
Schulz, Wu, Ruggeri, & Meder (*PsychSci* 2019)

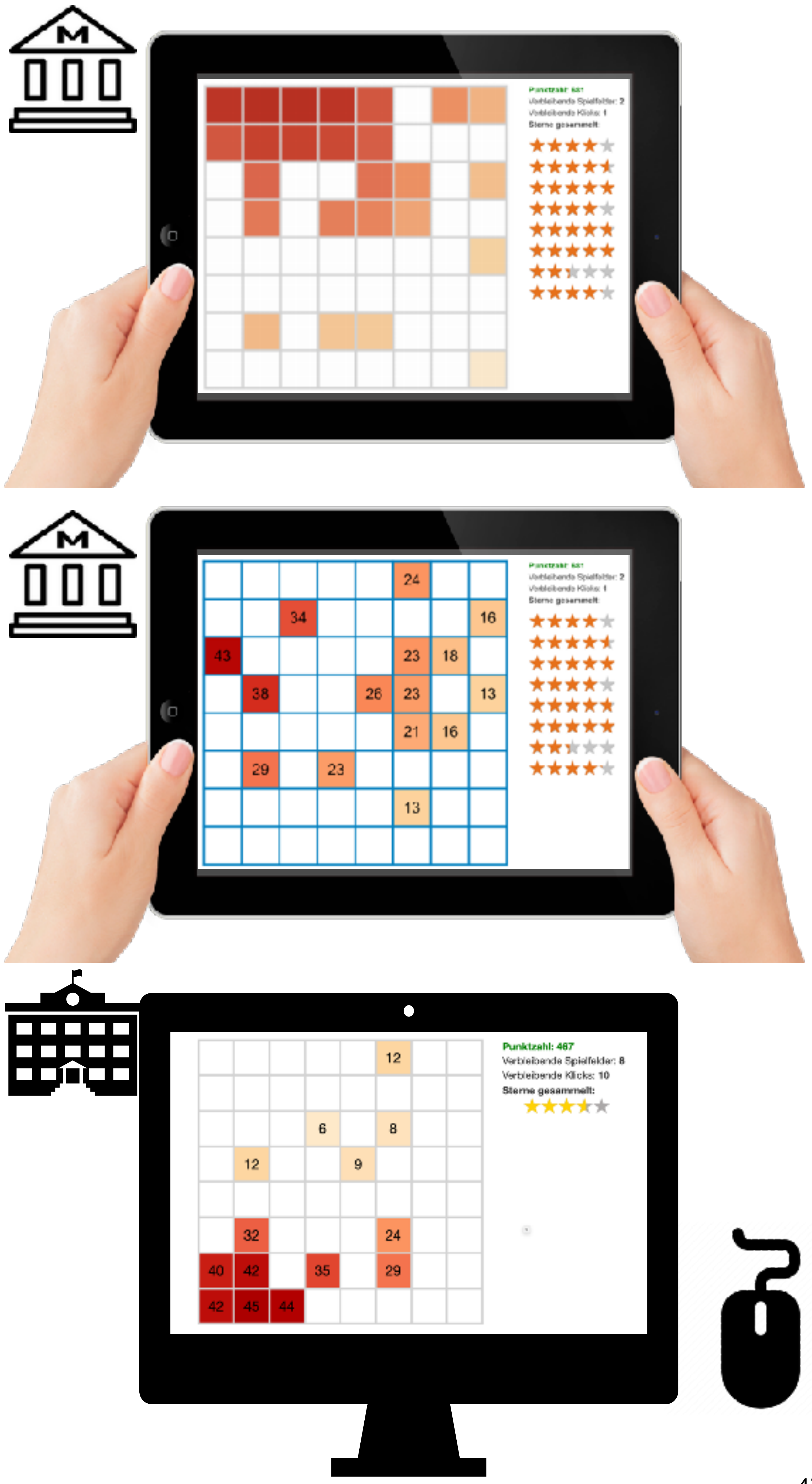



Meder, Wu, Schulz, & Ruggeri (DevSci 2021)
n = 52
Schulz, Wu, Ruggeri, & Meder (PsychSci 2019)
n = 79
Unpublished data
n = 150
Count
Age
0
5
10
15
20
10
20
30
40
50



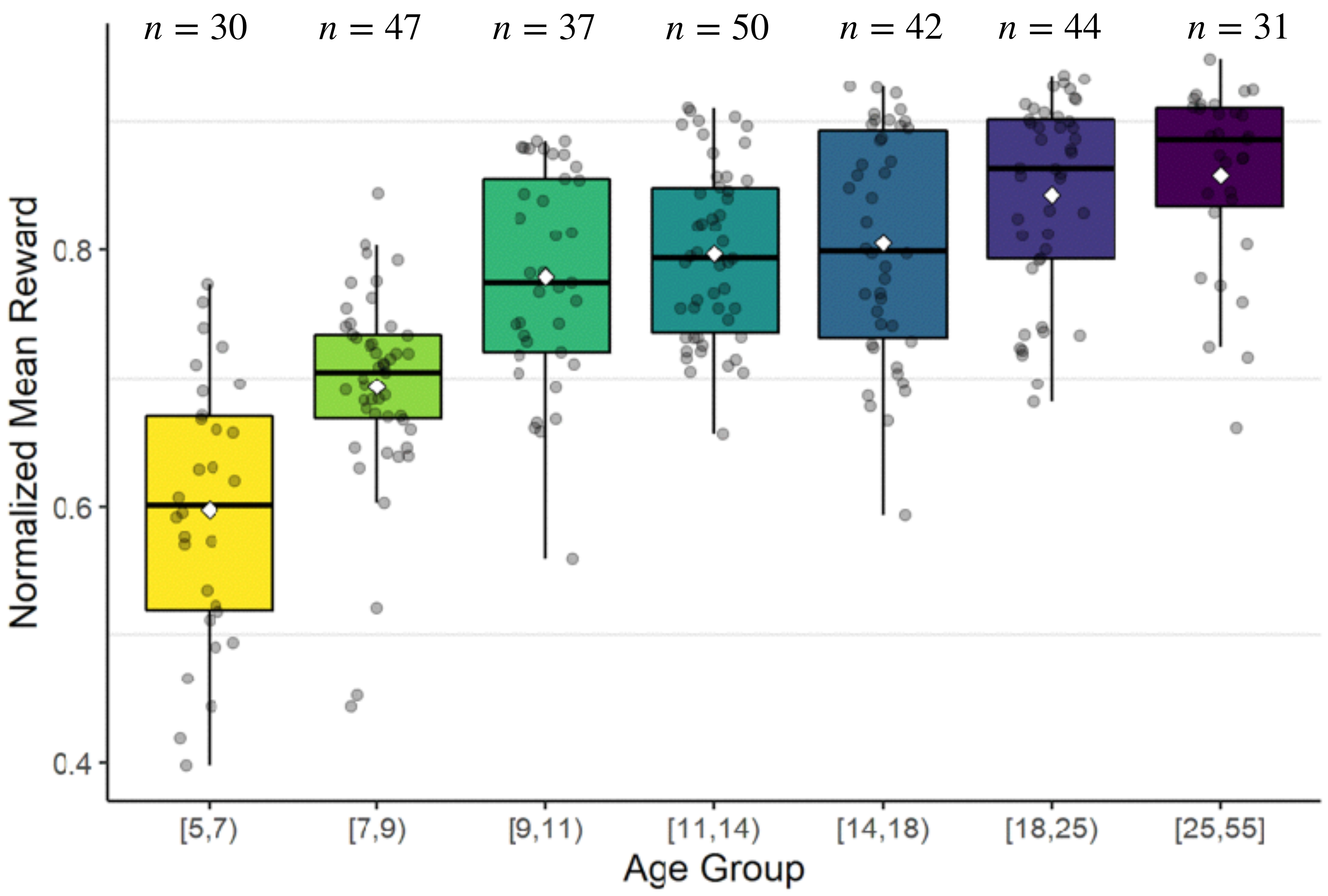
Filtered to use the same sets of environments, same grid size, and same number of trials per round





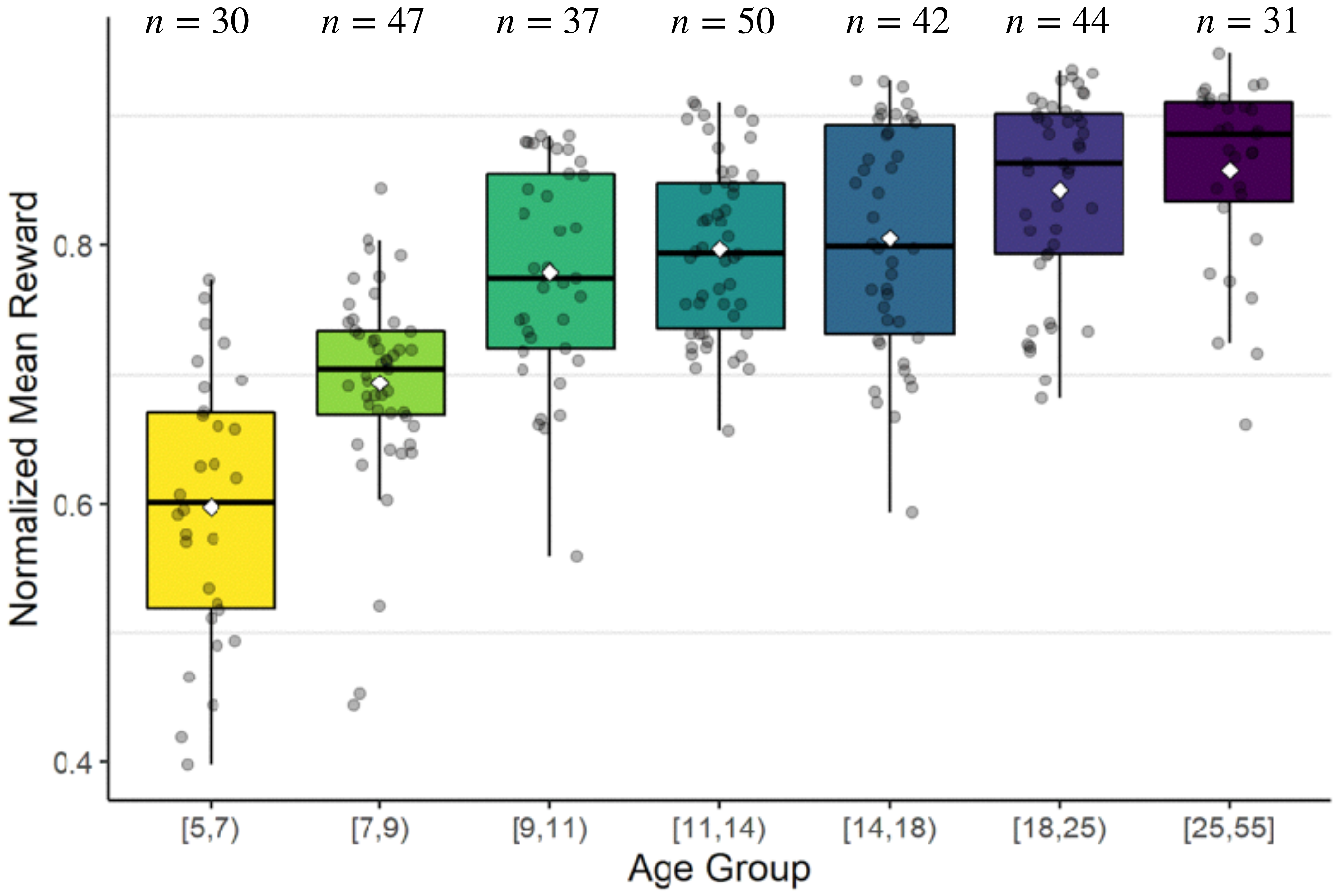
Behavioral results

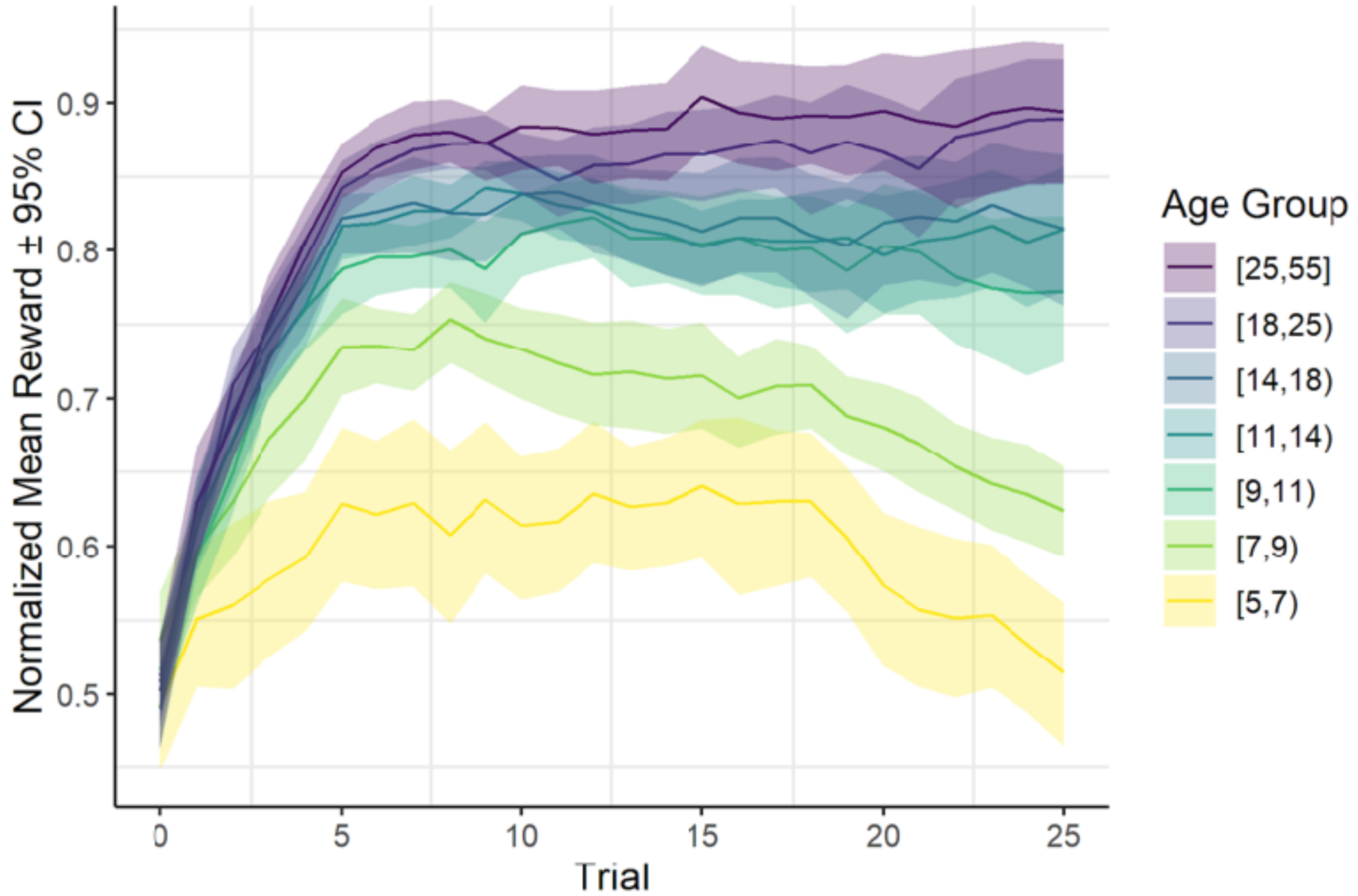
- Performance increases over age



Behavioral results

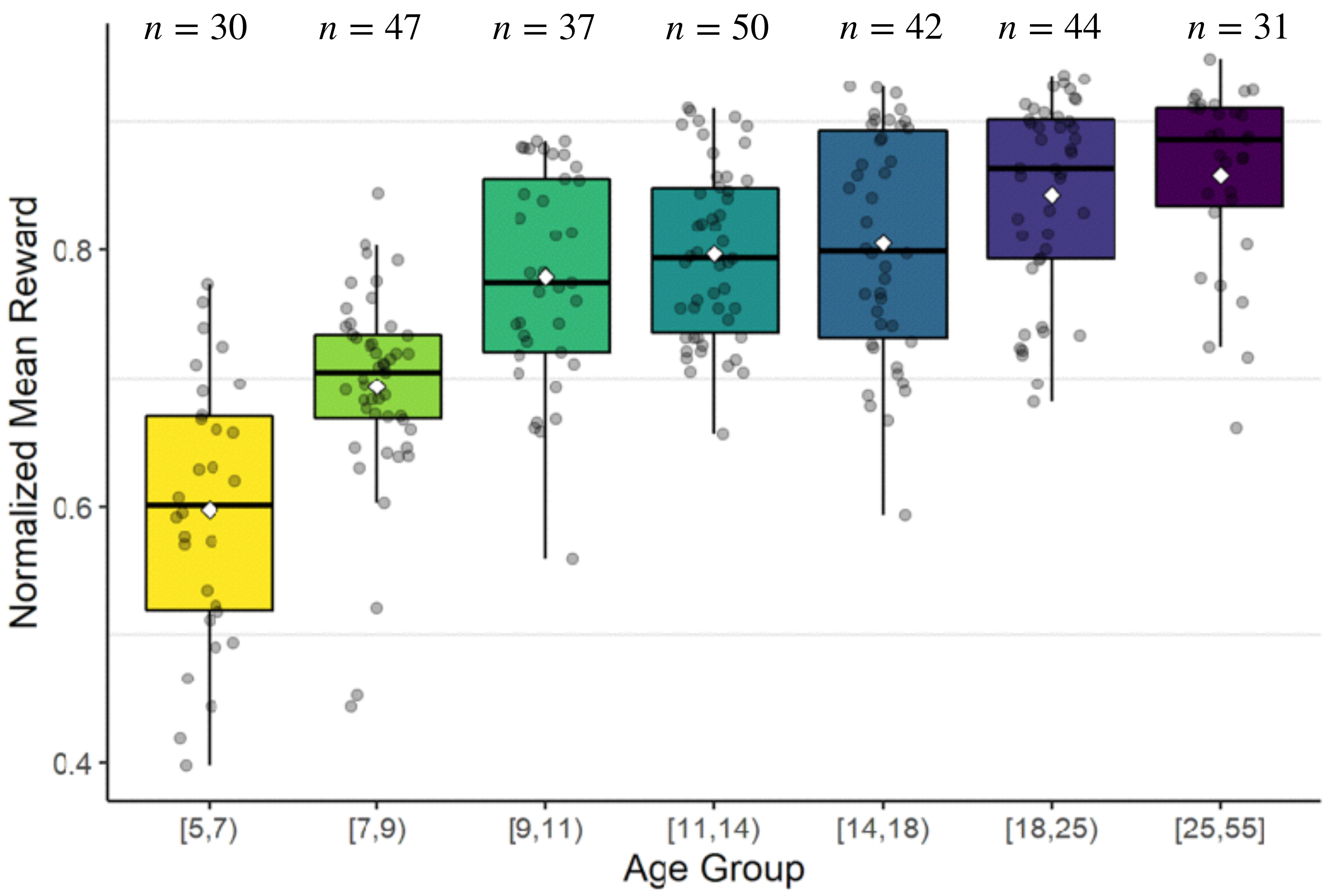
- Performance increases over age
- Age-related differences are already evident in the first few trials
  - Older subjects have steeper learning curves
  - Younger children have decaying learning curves, consistent with over-exploration, i.e., more unique options and higher entropy of choices

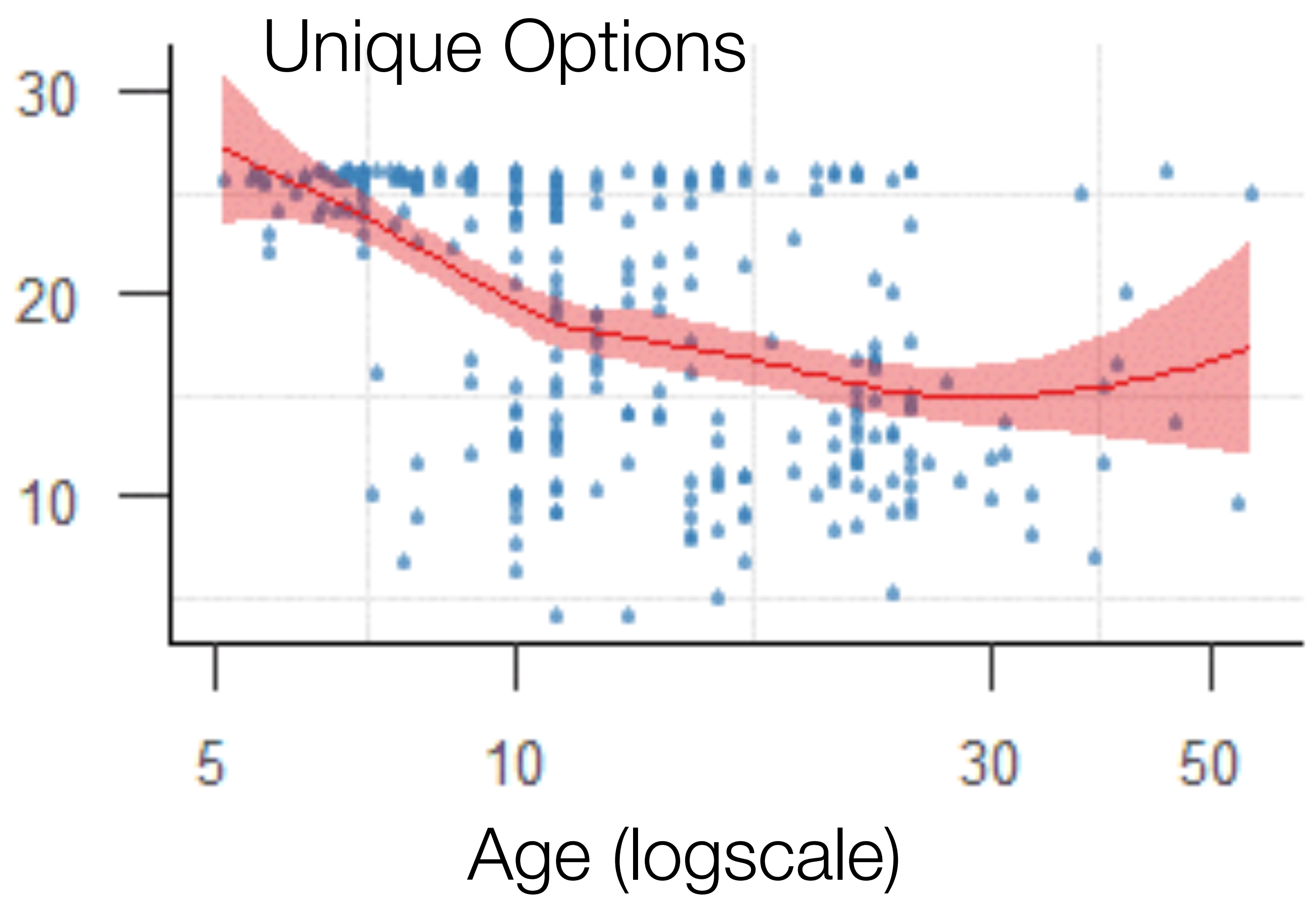


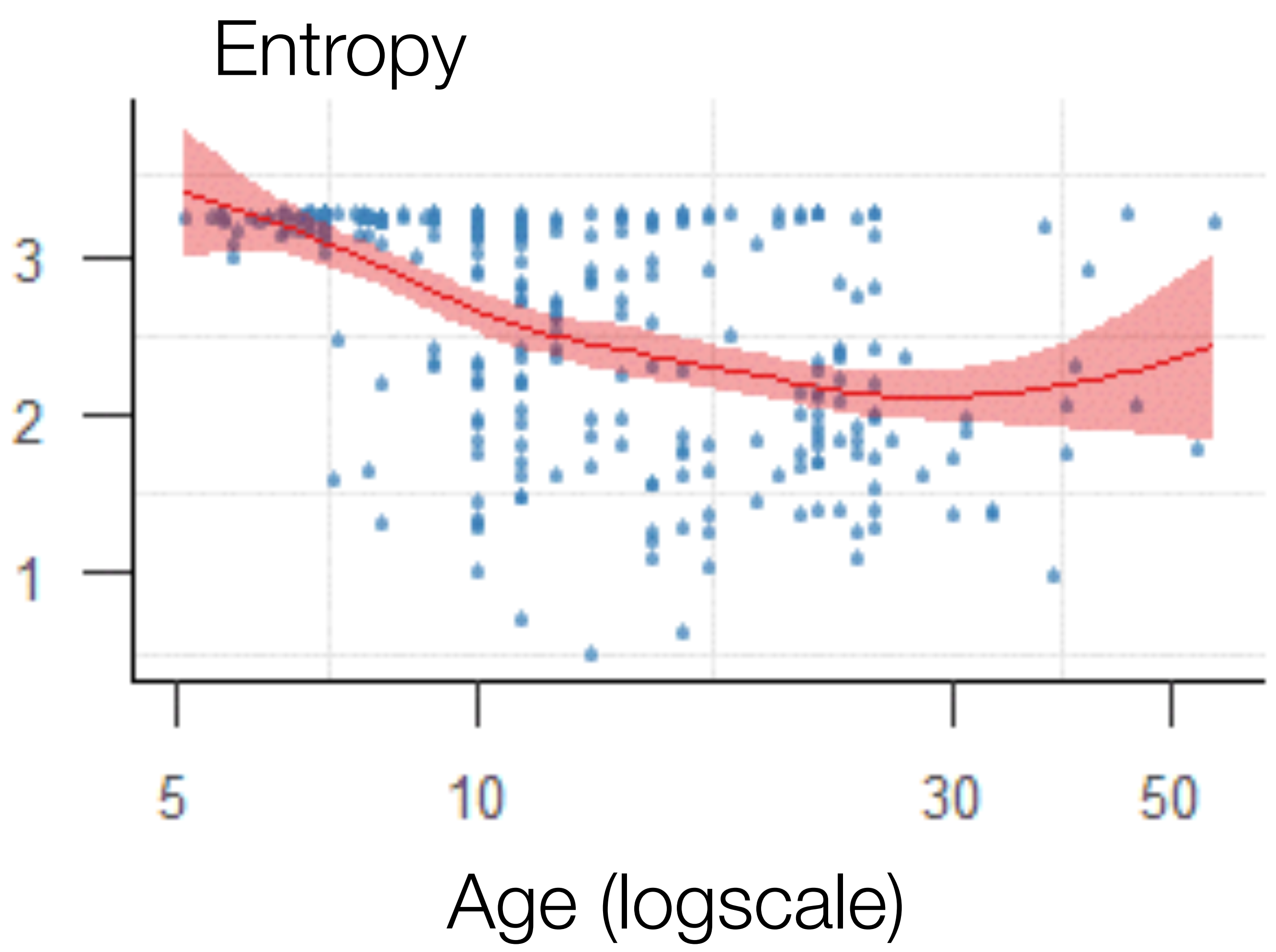


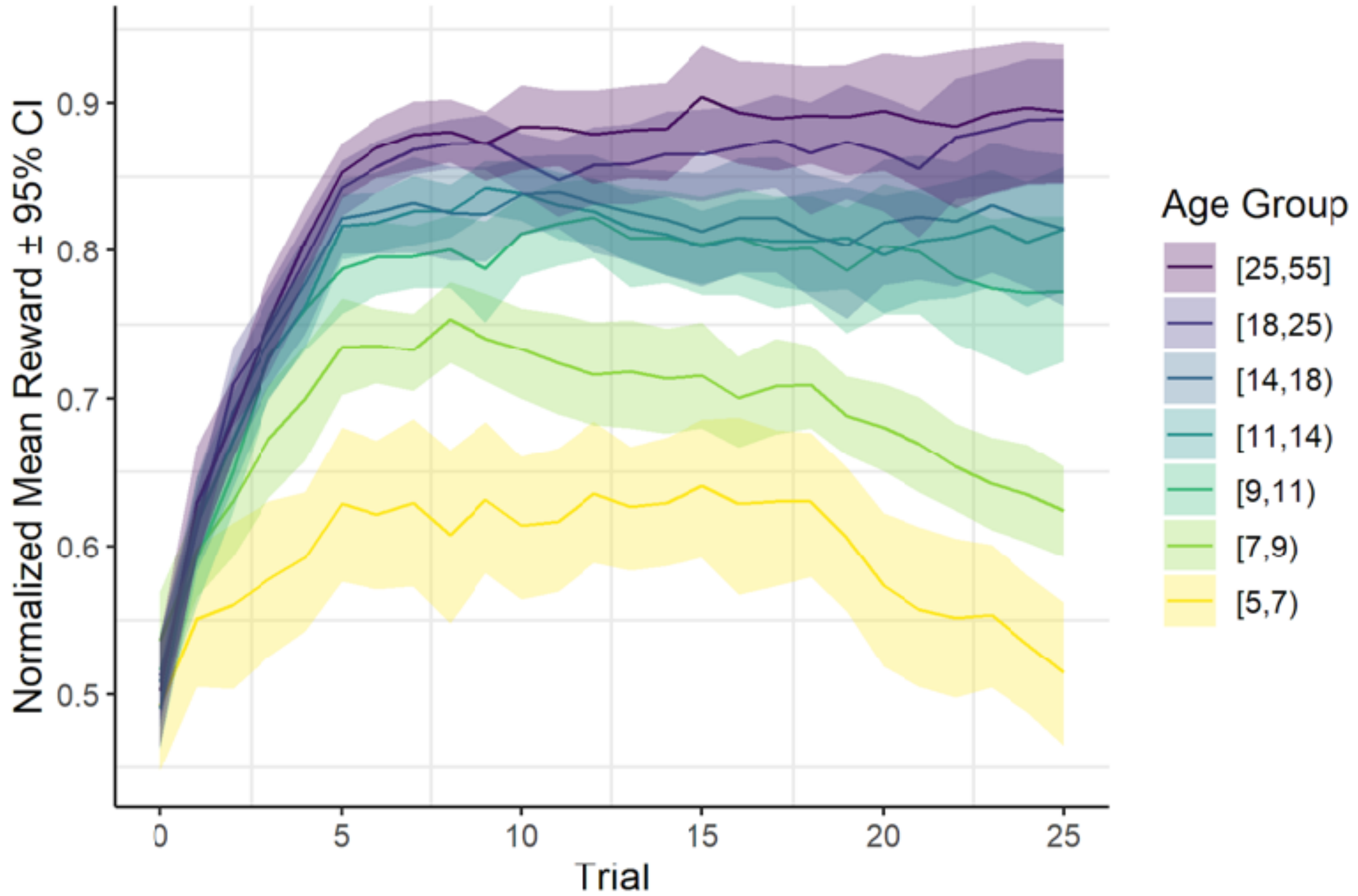
Behavioral results

- Performance increases over age
- Age-related differences are already evident in the first few trials
  - Older subjects have steeper learning curves
  - Younger children have decaying learning curves, consistent with over-exploration, i.e., more unique options and higher entropy of choices



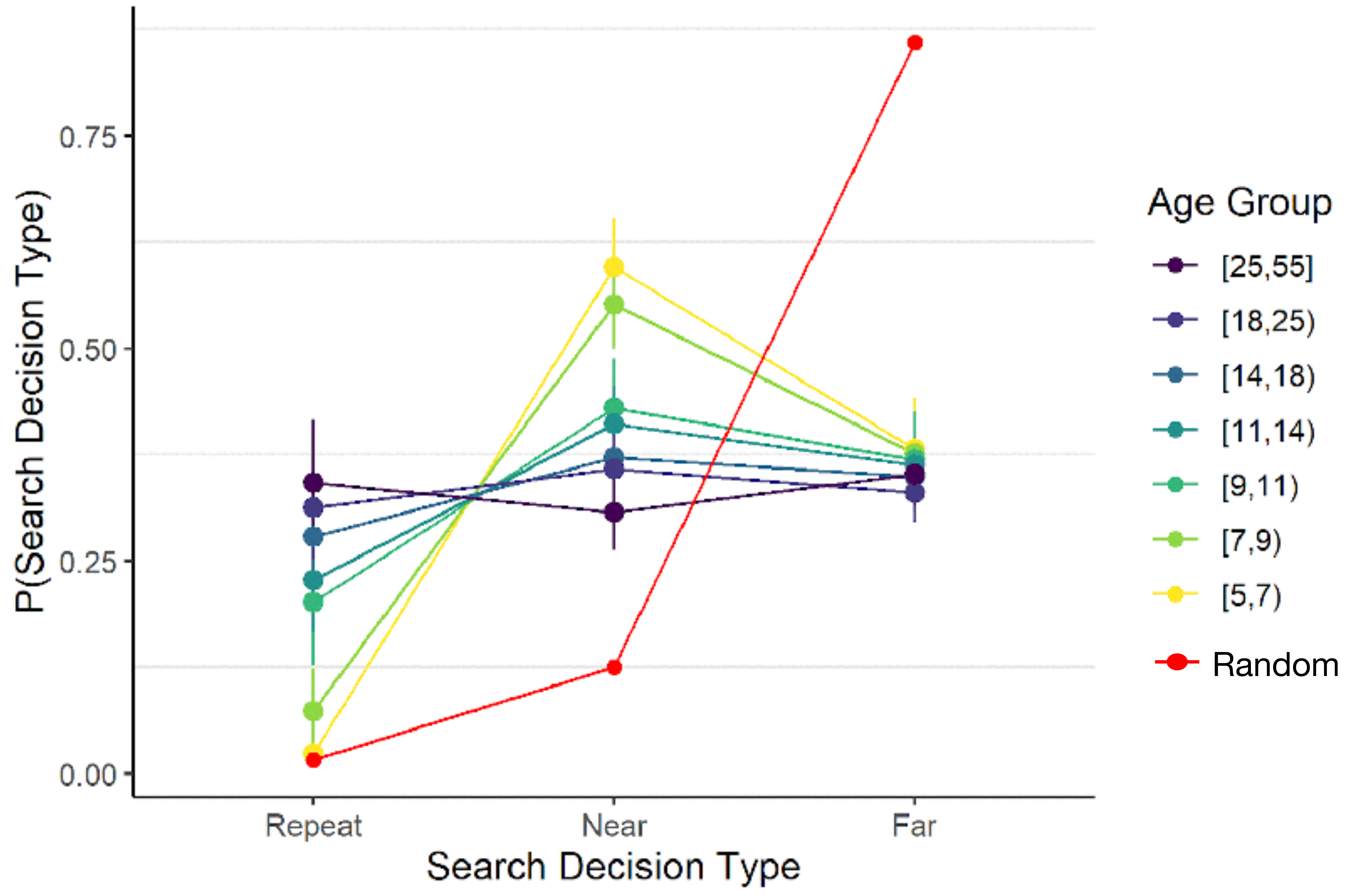






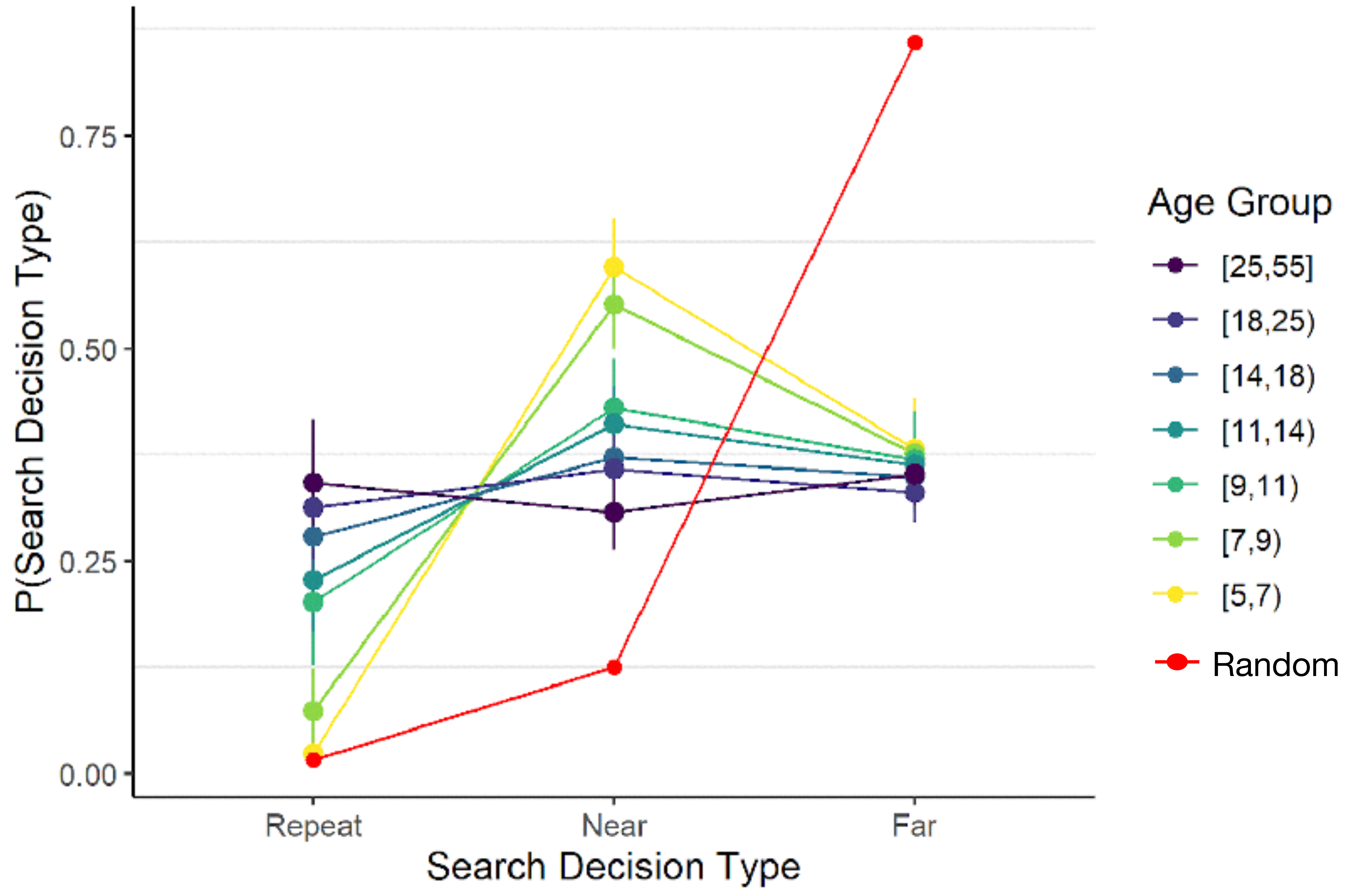
Behavioral results

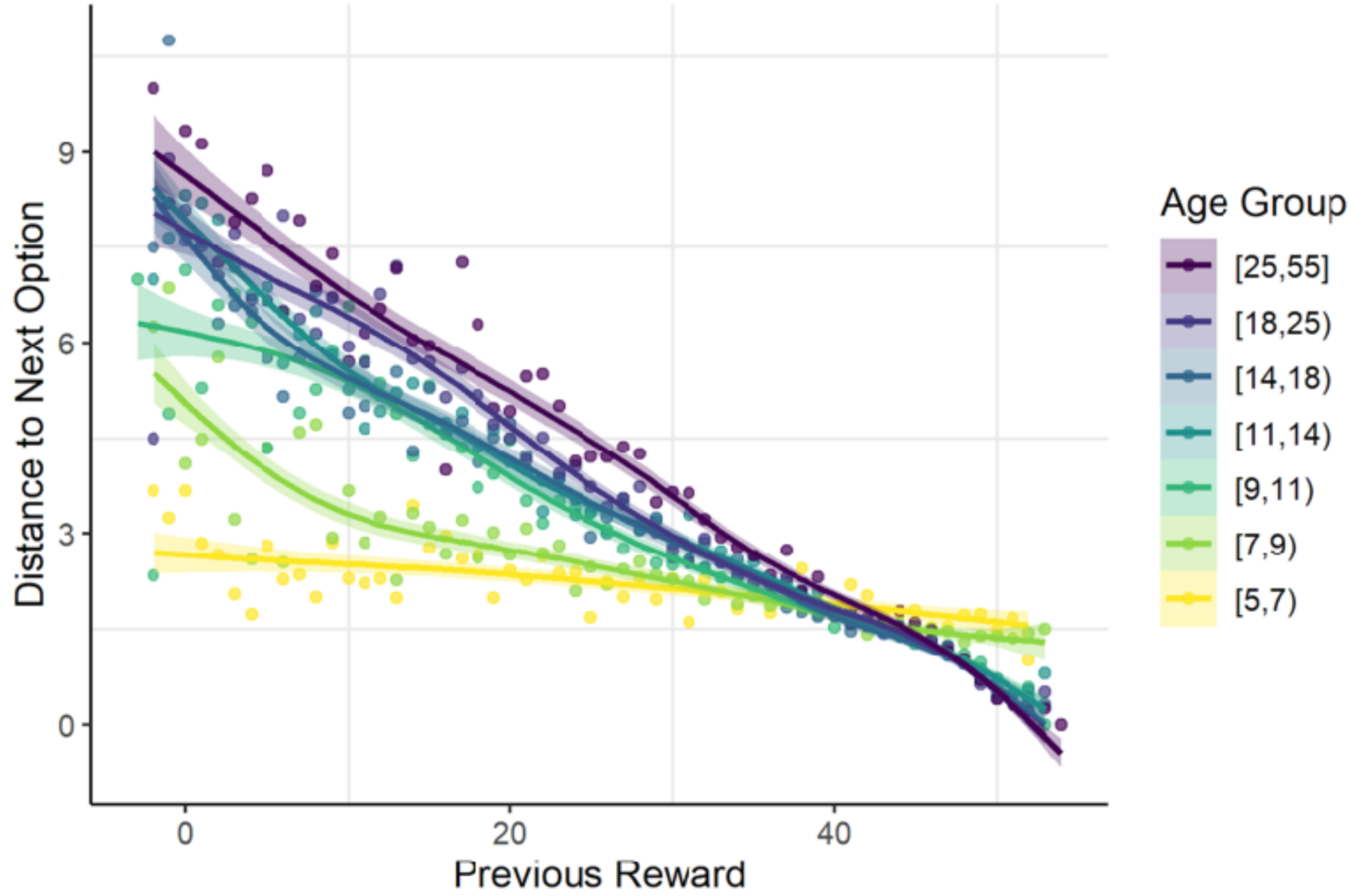
- Categorized decisions as either:
  - Repeat (same as last choice)
  - Near (neighboring option)
  - Far (any other choice)
- P(near) > P(repeat) for younger children, but reaches parity in adults



Behavioral results

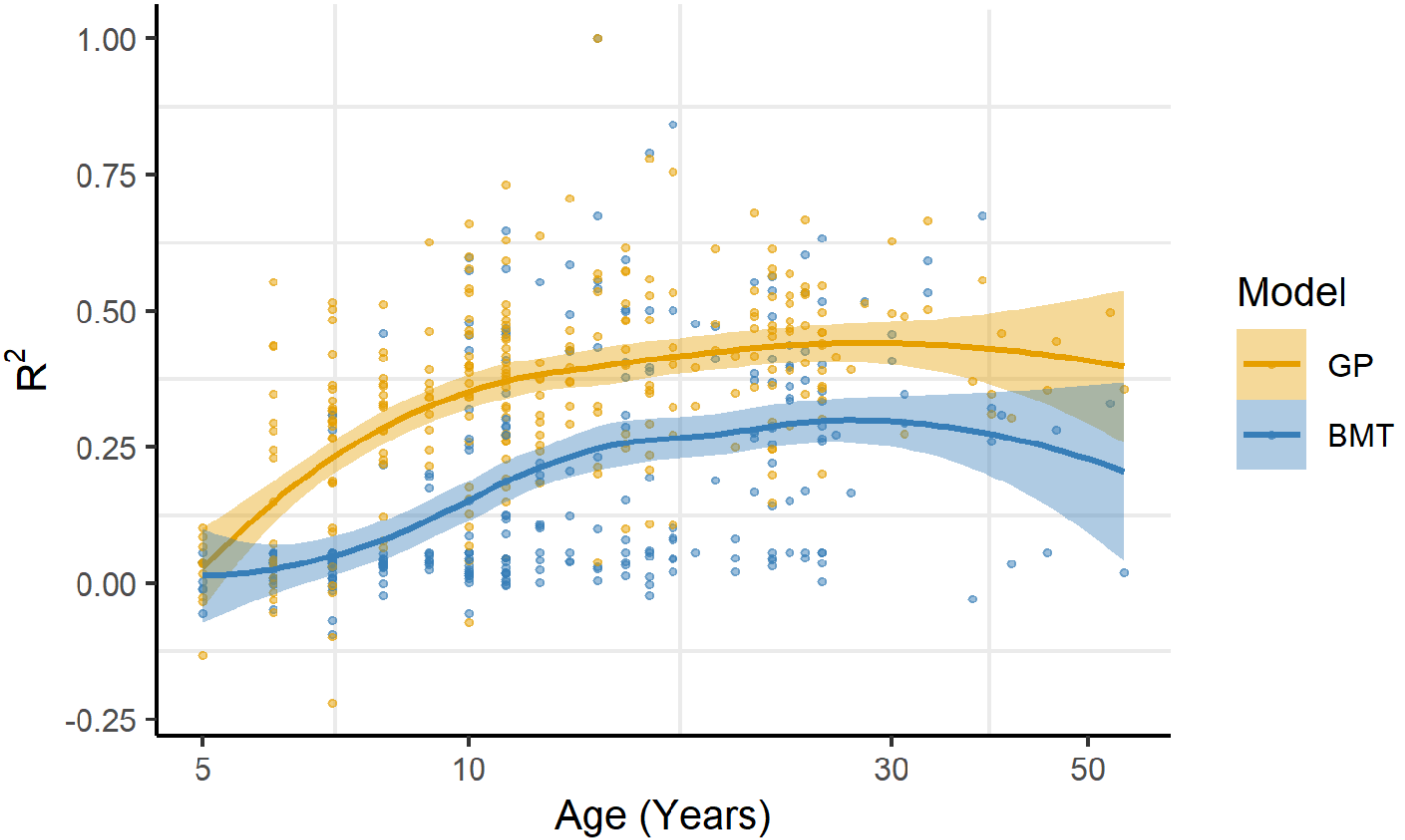
- Categorized decisions as either:
  - Repeat (same as last choice)
  - Near (neighboring option)
  - Far (any other choice)
- P(near) > P(repeat) for younger children, but reaches parity in adults
- Younger children are also less responsive in adapting search distance to reward outcomes
  - Over the lifespan, this develops into a linear relationship, resembling a gradual form of win-stay lose-shift

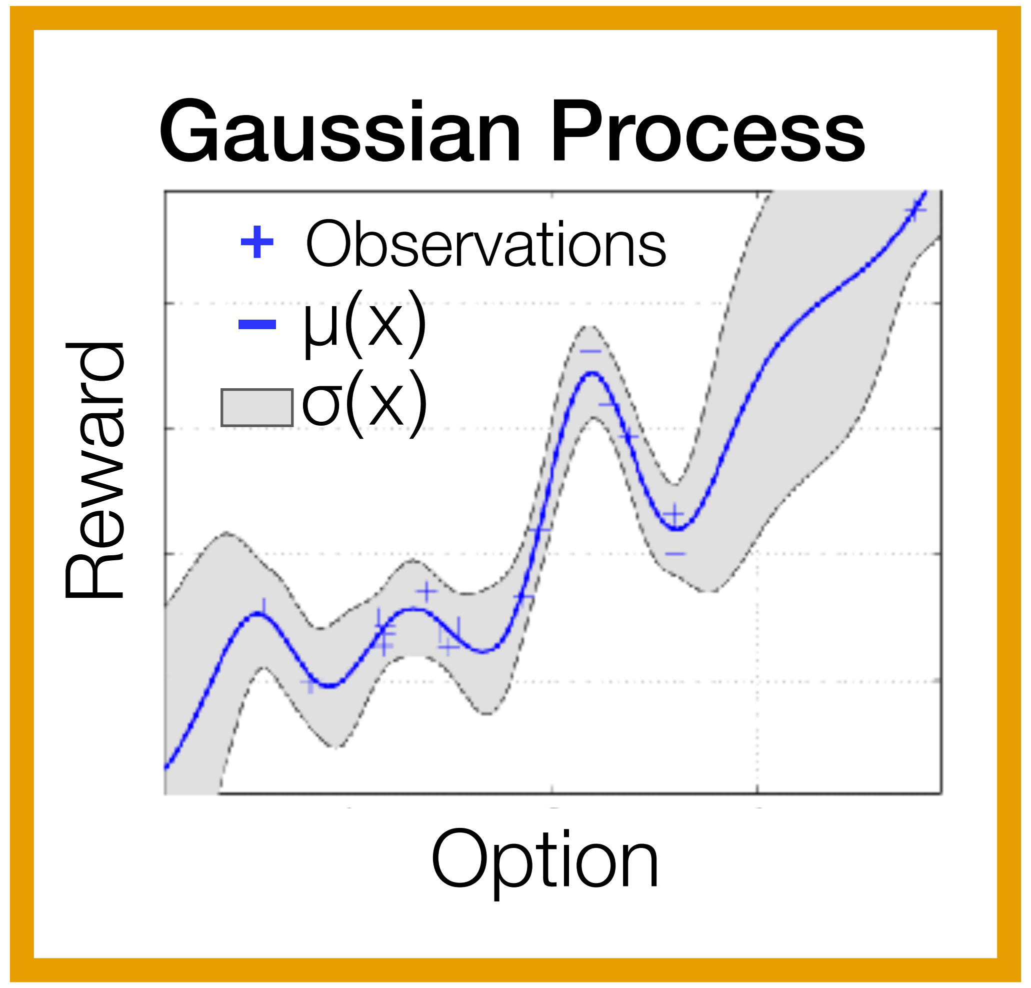


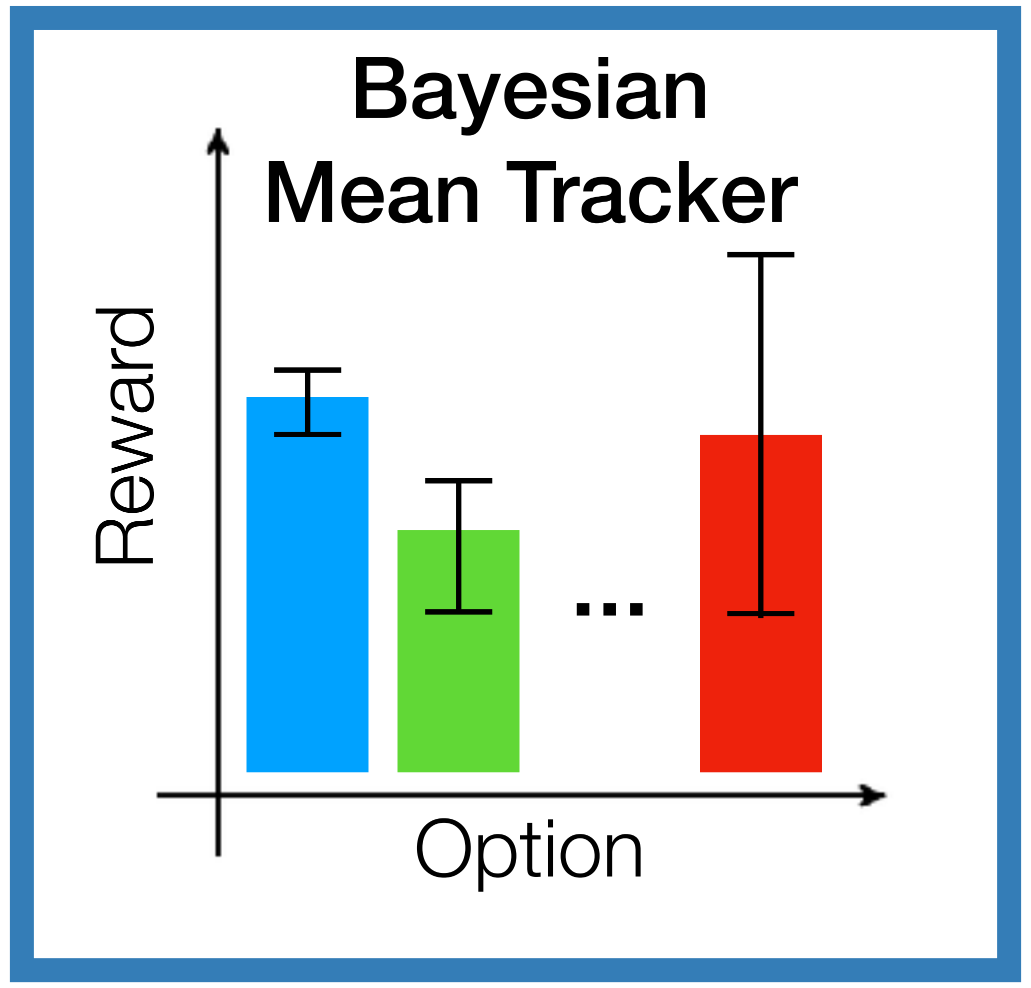


Model results

$$R^2 = 1 - \frac{\log \mathcal{L}(\mathcal{M}_k)}{\log \mathcal{L}(\mathcal{M}_{\text{rand}})}$$



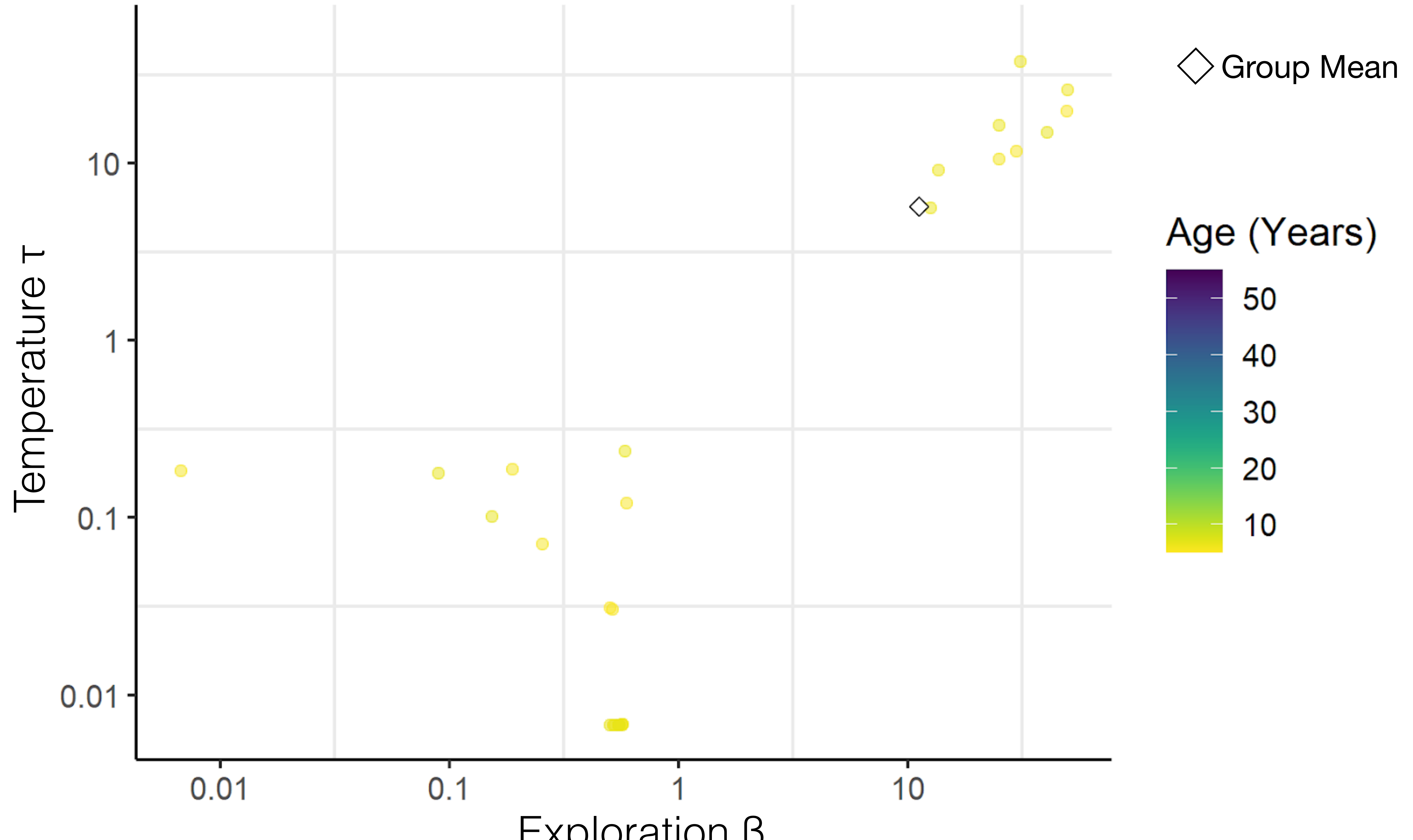




Parameter estimates

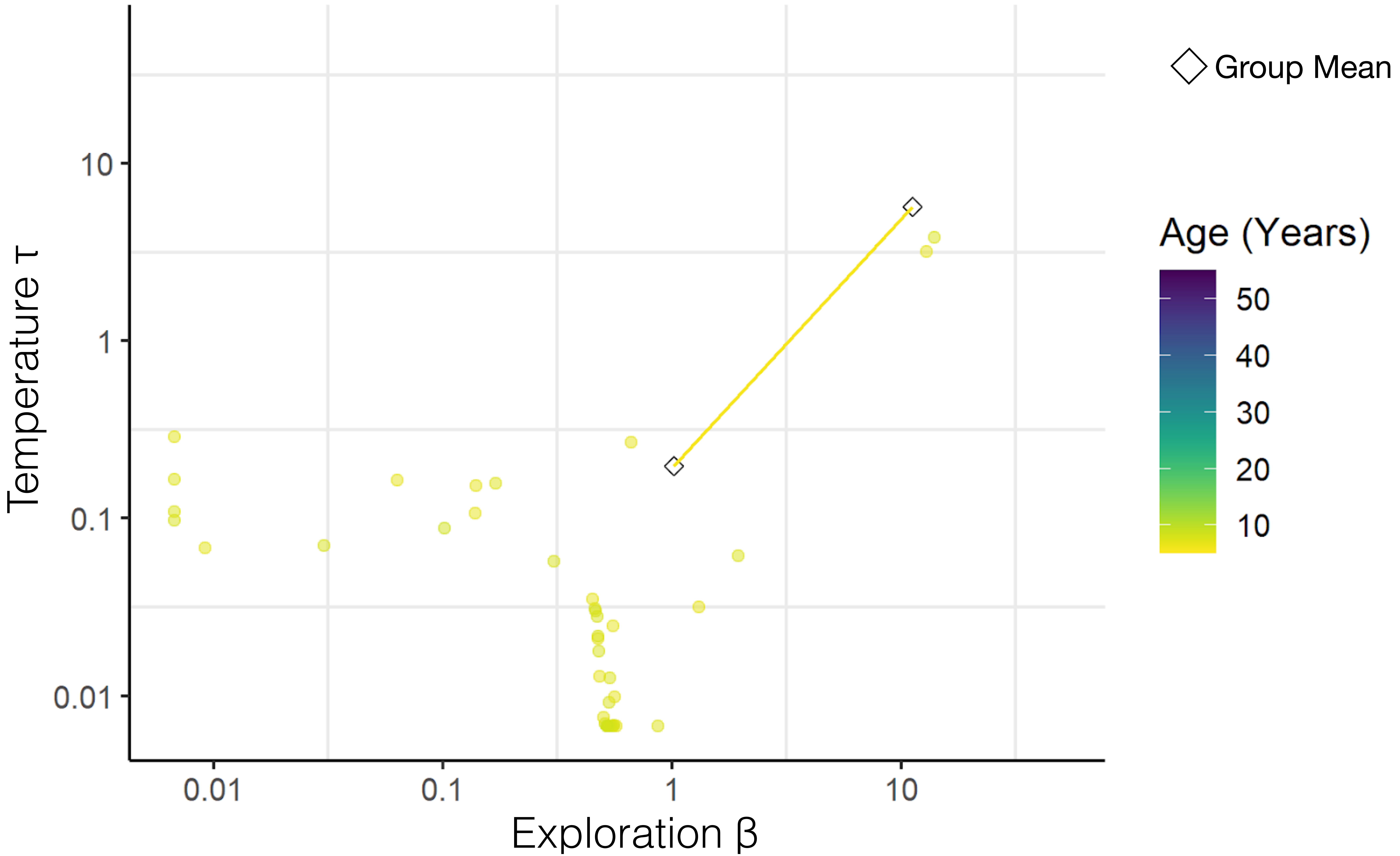
- Small uptick in generalization λ
- Large decrease in both uncertainty-directed exploration β and temperature τ

Age Group [5,7)

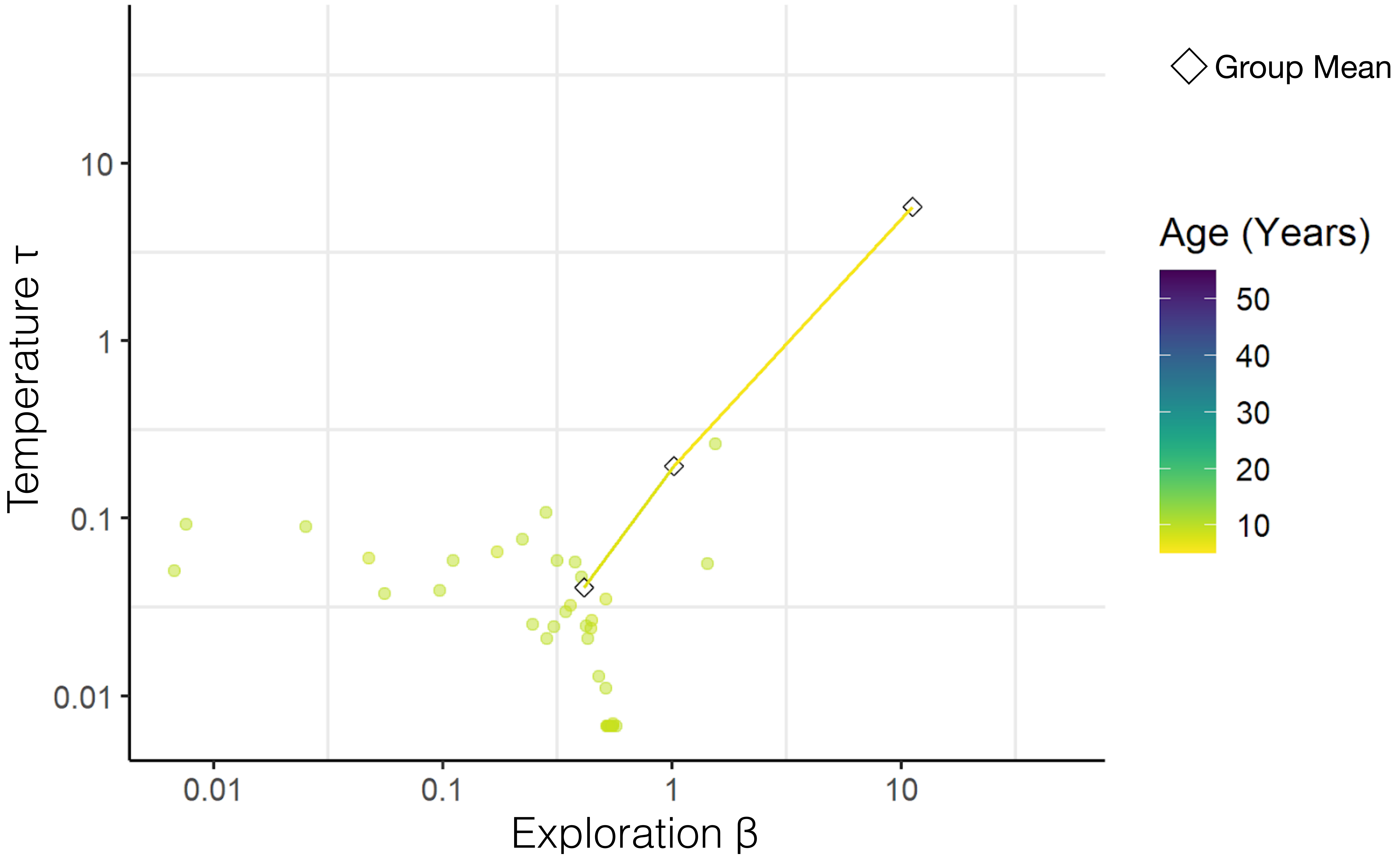


Group Mean
Age (Years)
50
40
30
20
10
10
1
0.1
0.01
Temperature τ
0.01
0.1
1
10
Exploration β

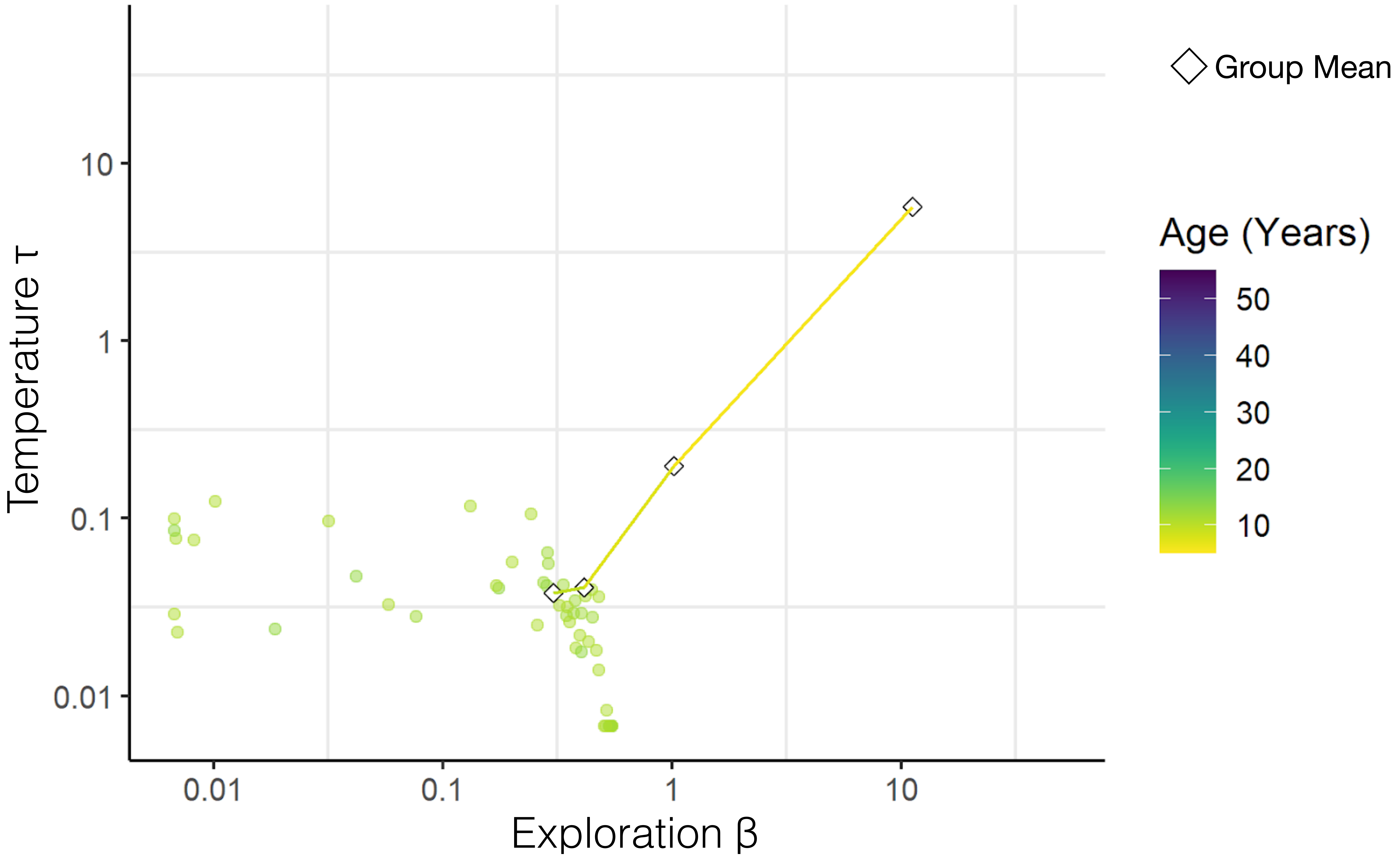
Age Group [7,9)


Group Mean
Temperature τ
Exploration β
10
1
0.1
0.01
0.01
0.1
1
10
Age (Years)
50
40
30
20
10

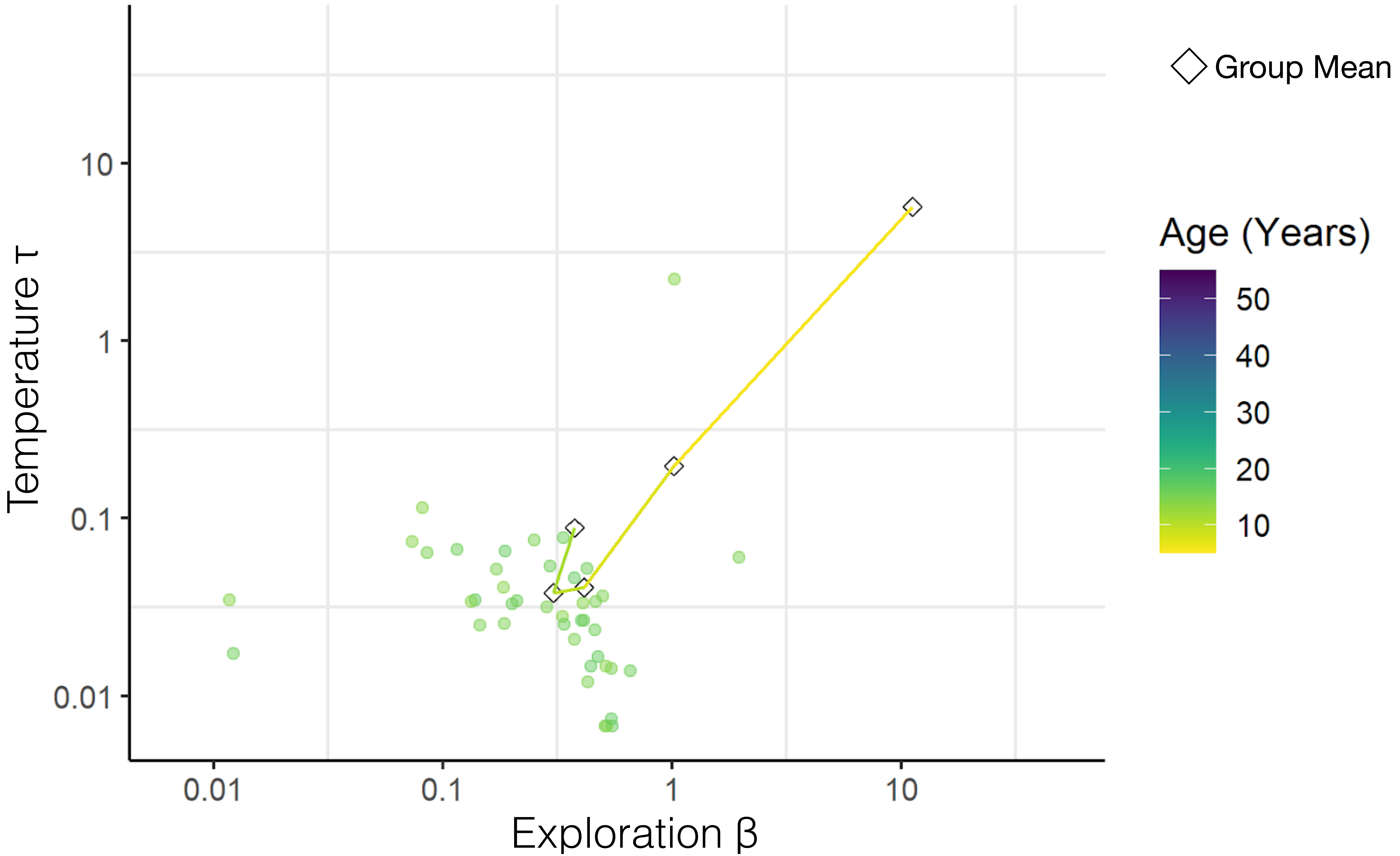
Age Group [9,11)



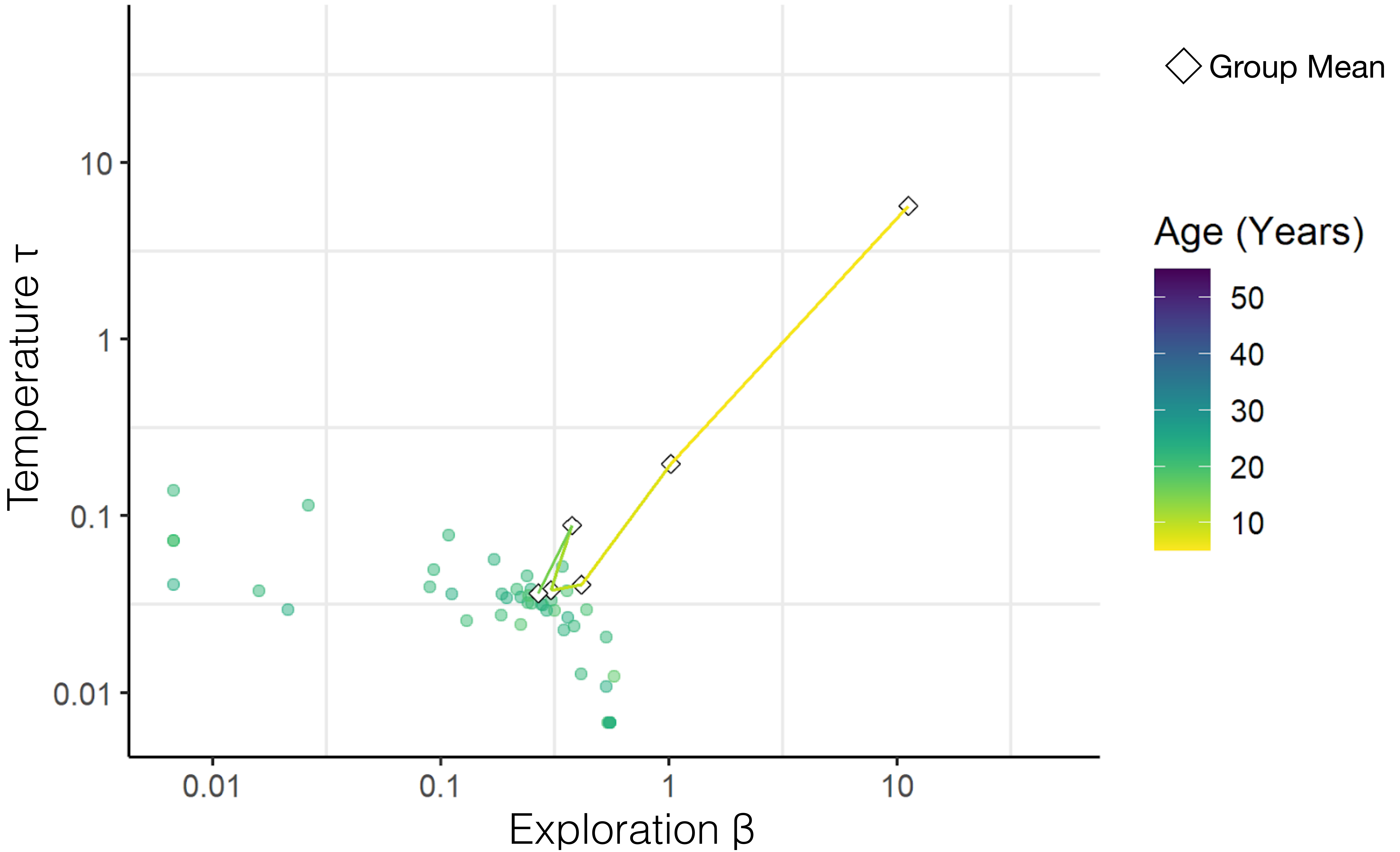
Age Group [11,14)



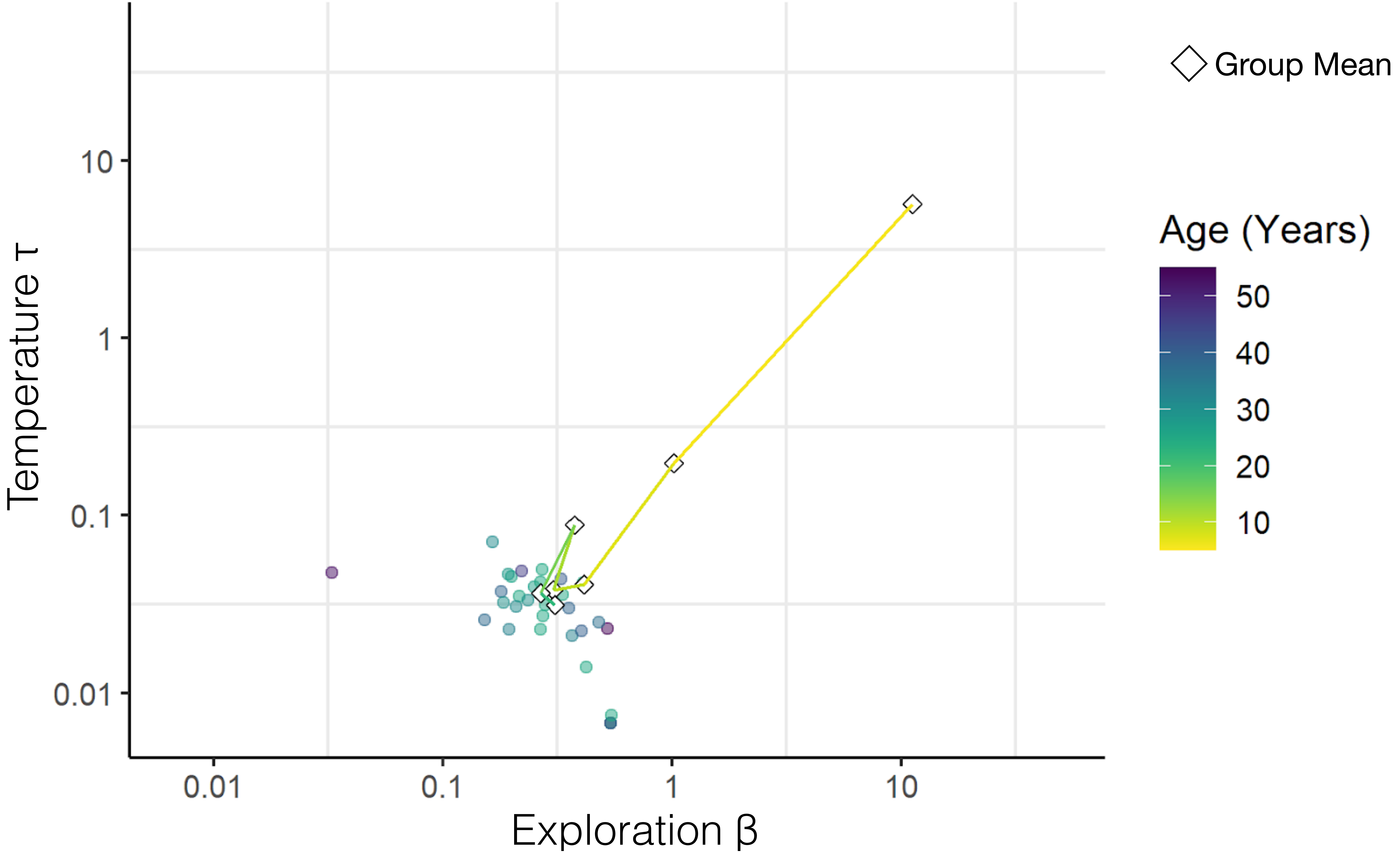
Age Group [14,18)



Age Group [18,25)



Age Group [25,55]

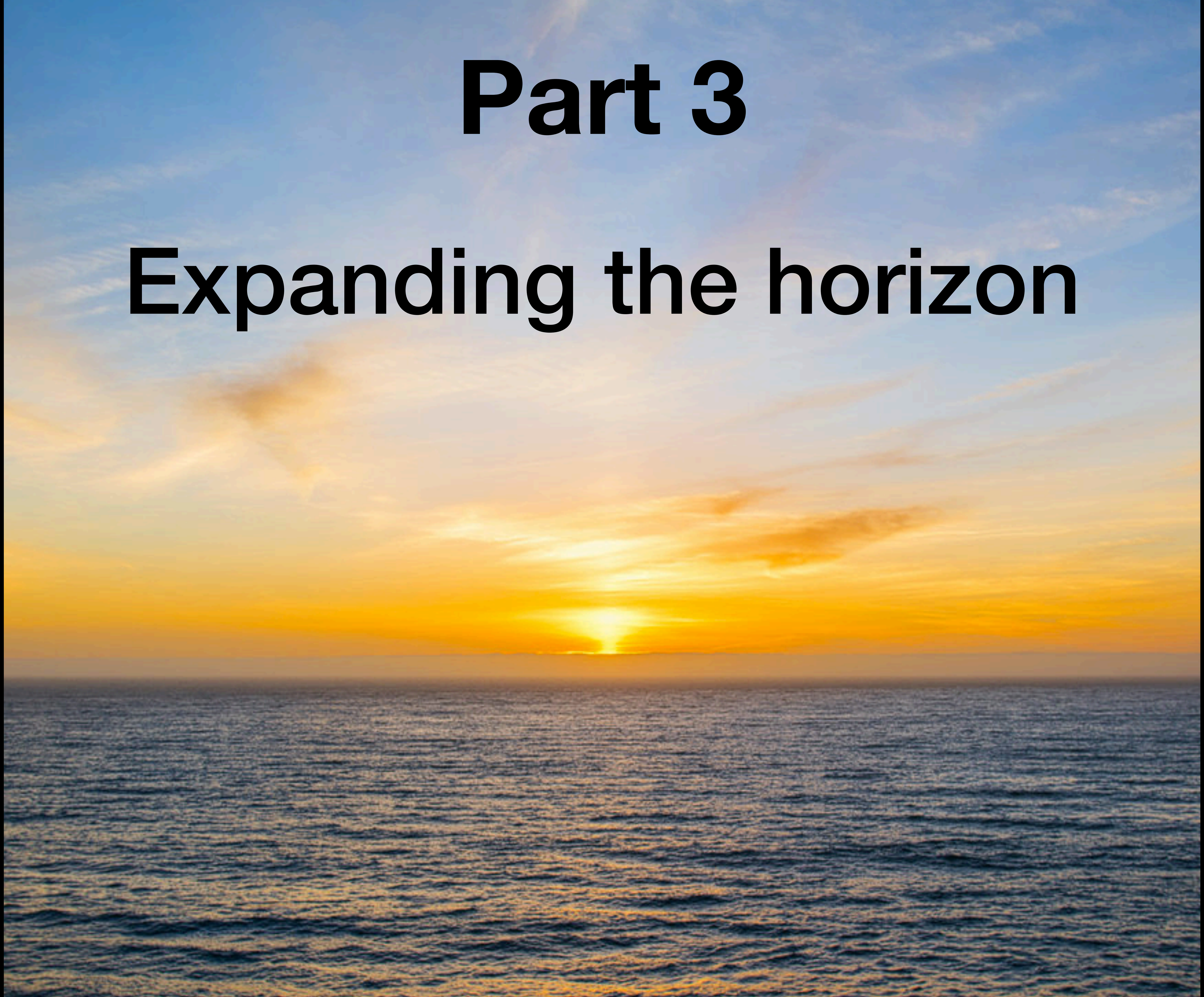


Summary and Future Directions

- The strategic use of uncertainty-directed exploration and predictive generalization comes online already at a very young age (5-7 year olds)
  - Simultaneous reduction in both directed and random exploration in early childhood
  - Children are not just more random, but also hungrier for information
- While there is an uptick in random exploration during adolescence, this is relatively minor compared to changes in childhood
  - Consistent with theories that increased exploration in adolescence is largely driven by social rather than cognitive factors
- Future work can use model simulations to examine which is the best normative developmental trajectory through model space

Part 3

Expanding the horizon



Part 3

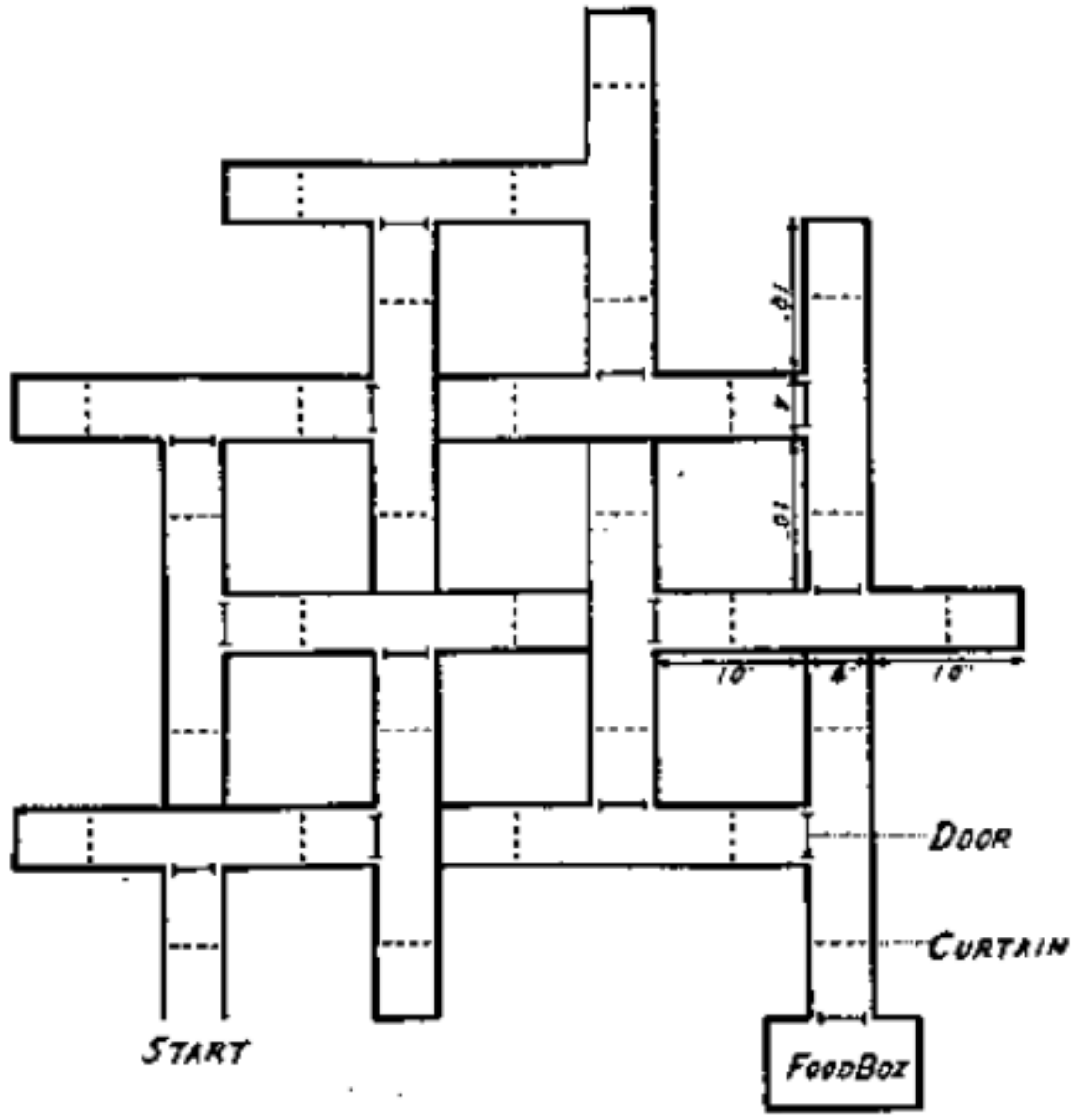
Expanding the horizon

Generalization in Conceptual and Structured domains

Cognitive Maps for Navigation

Cognitive Maps

Tolman (Psych Rev, 1948)



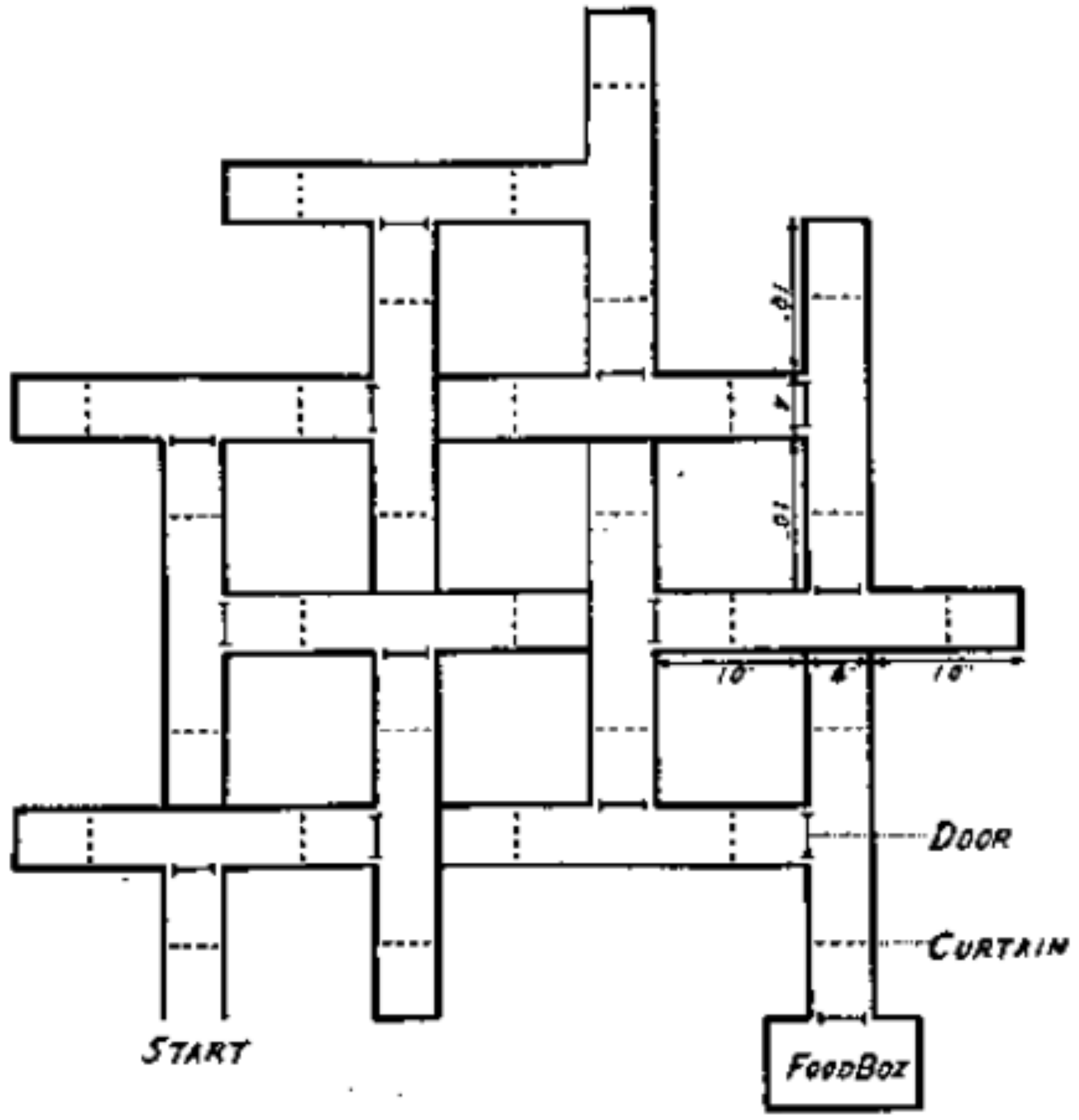
Plan of maze
14-Unit T-Alley Maze
FIG. 1
(From M. H. Elliott, The effect of change of reward on the maze performance of rats. *Univ. Calif. Publ. Psychol.*, 1928, 4, p. 20.)

… "in the course of learning something like a field map of the environment gets established in the rat's brain"

Cognitive Maps for Navigation

Cognitive Maps

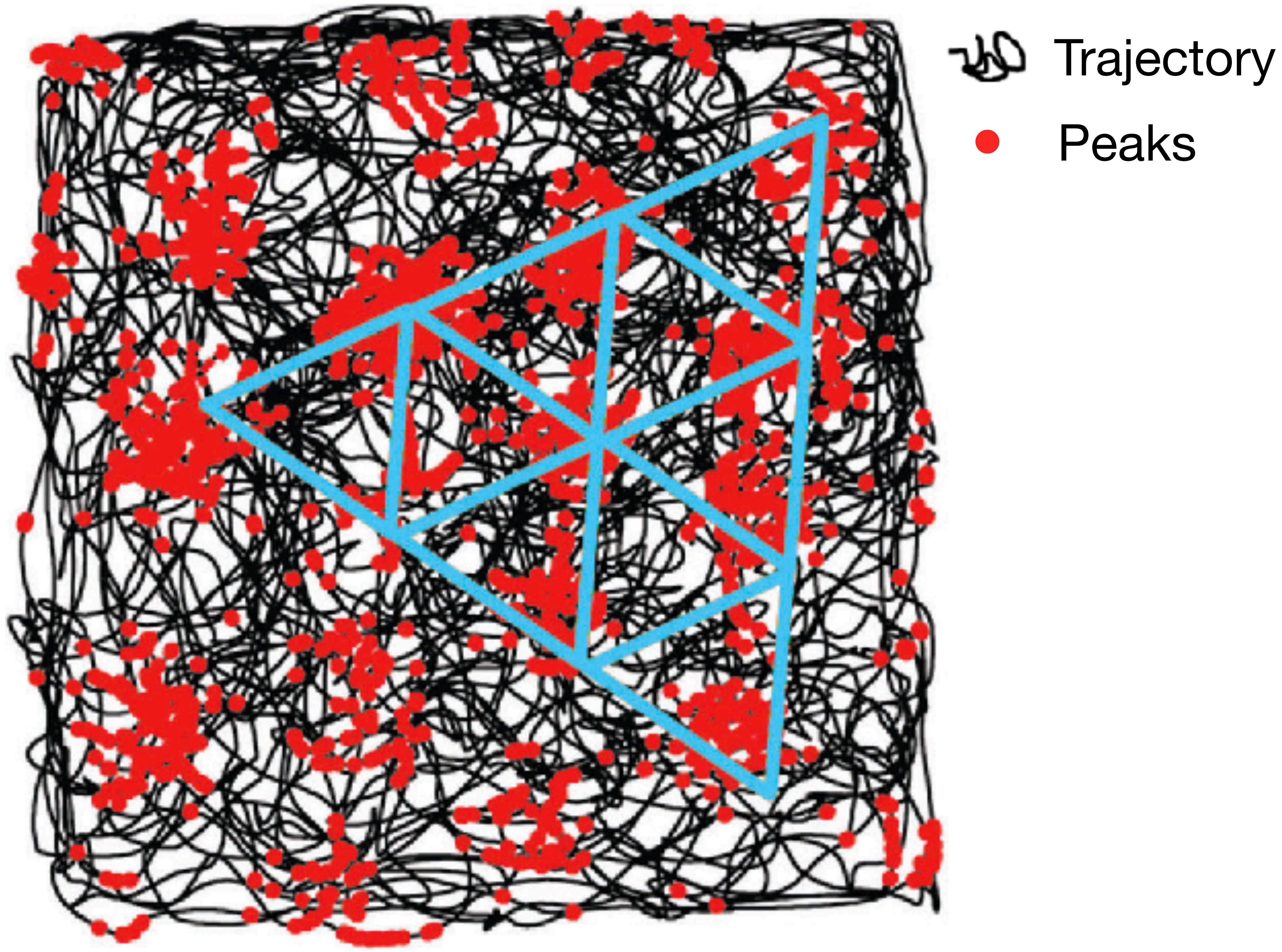
Tolman (Psych Rev, 1948)



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.)

… "in the course of learning something like a field map of the environment gets established in the rat's brain"

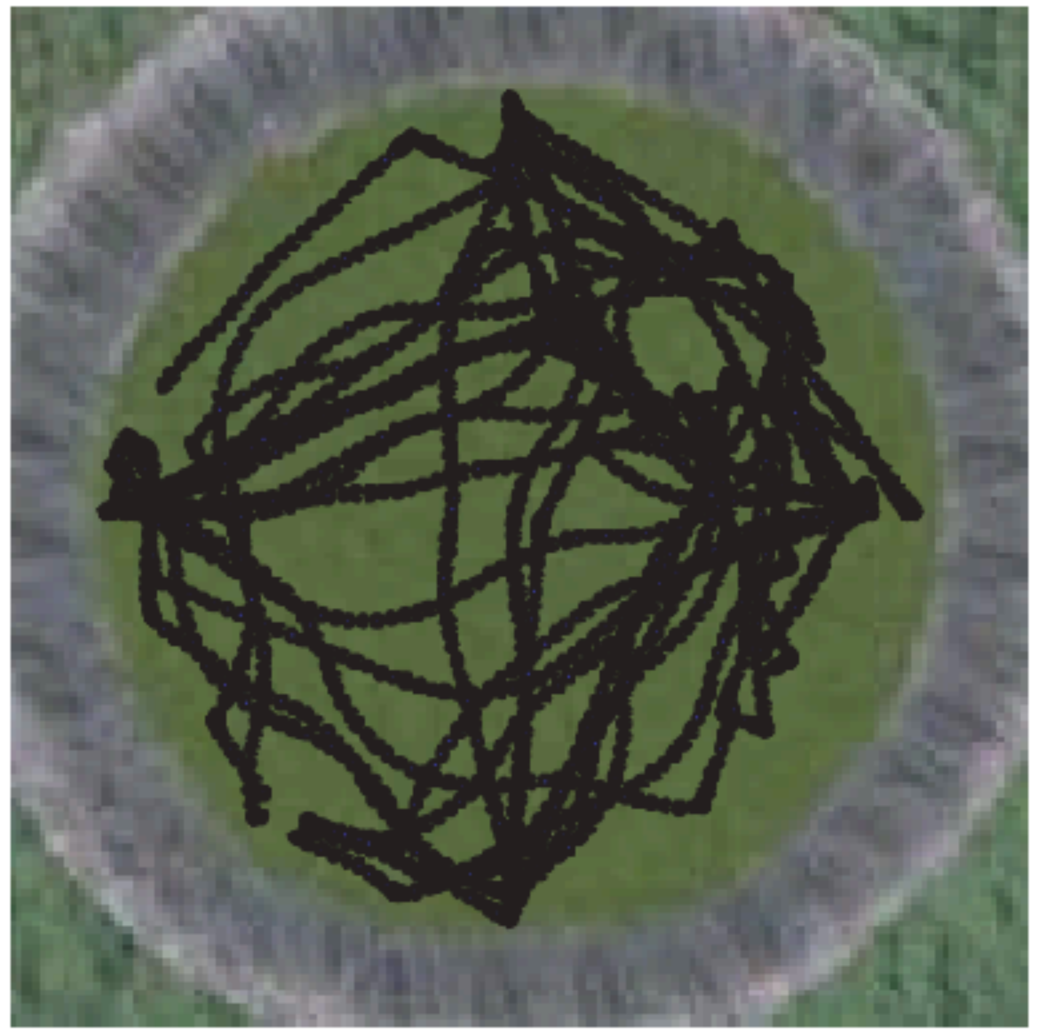
Grid Cells

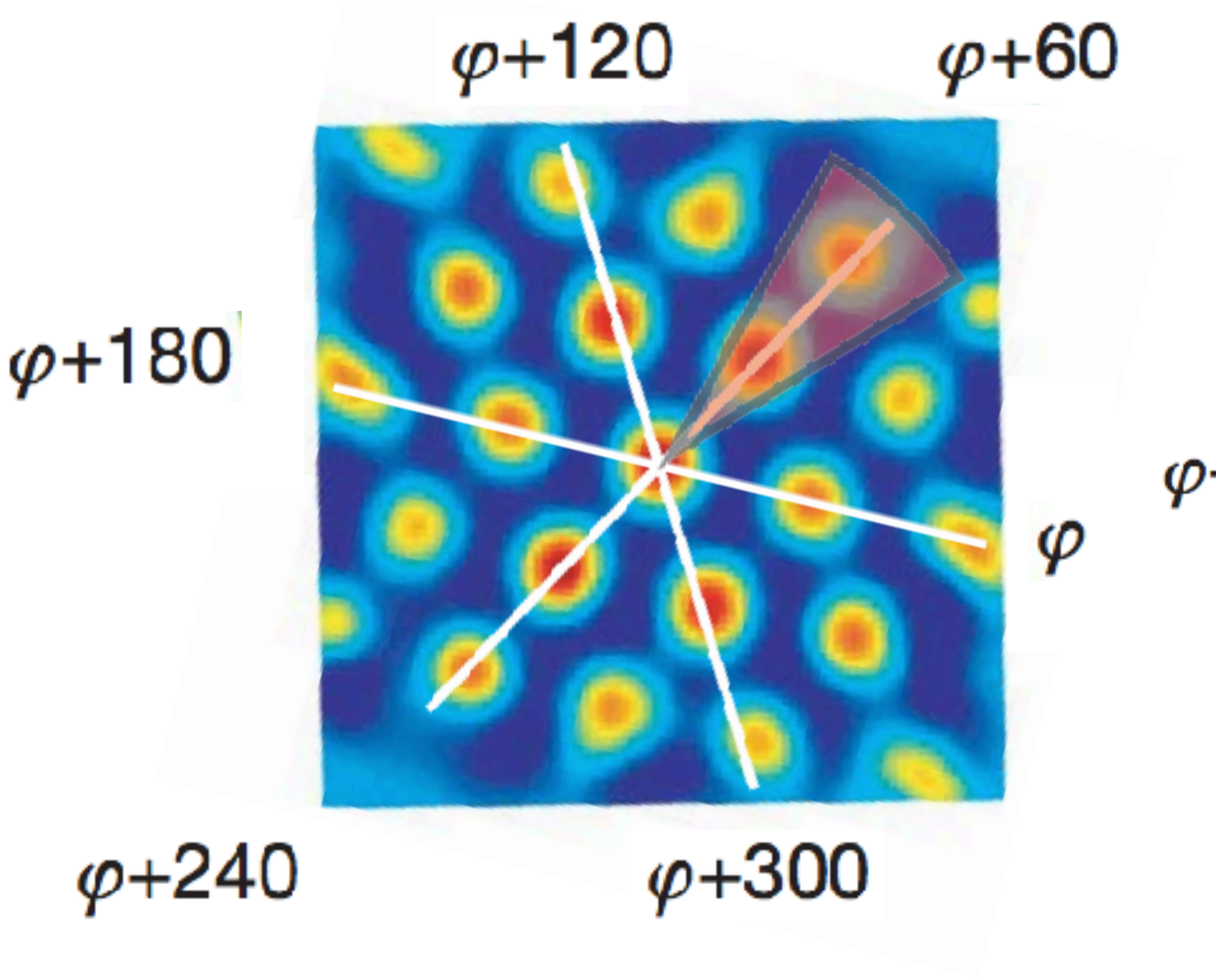


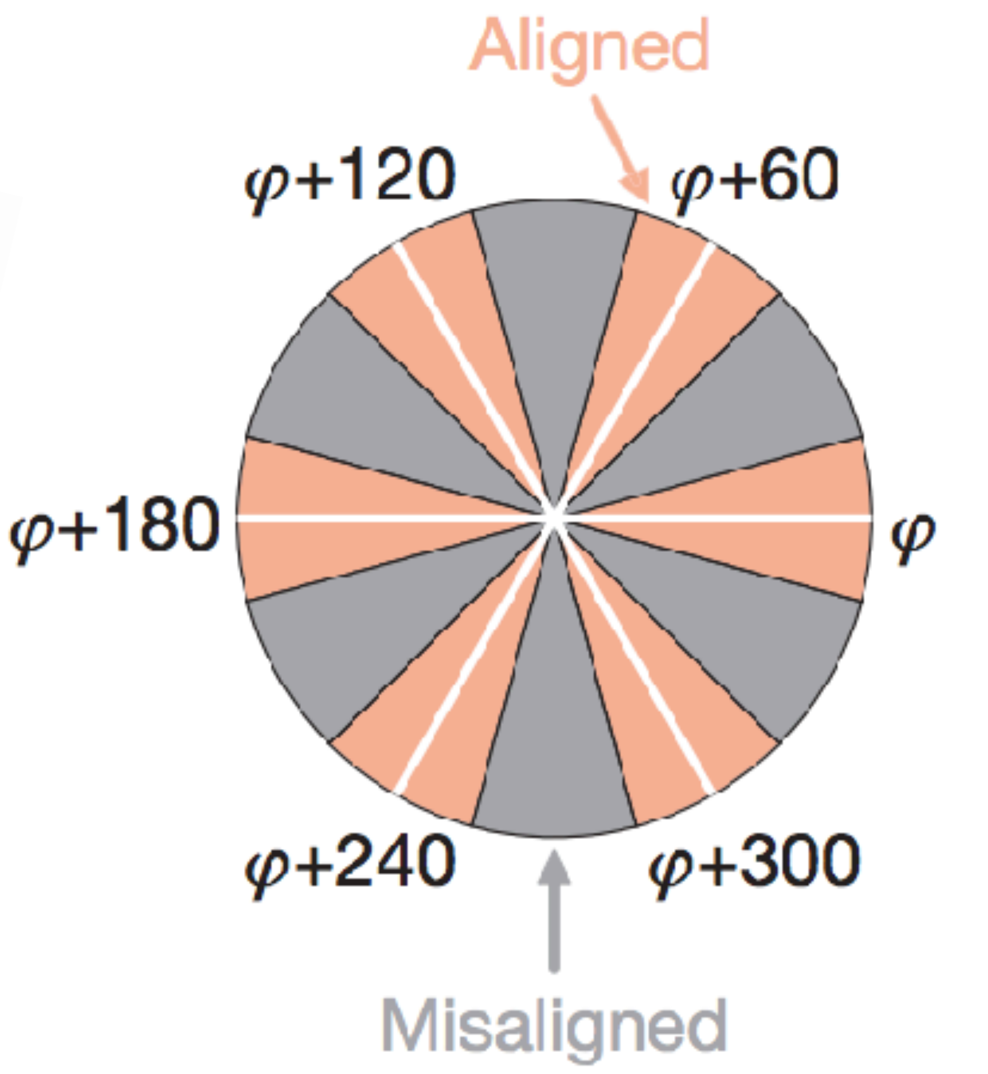
Hafting et al., (2005)
Moser, Rowland, & Moser (2015)

We can measure trajectories in human navigation

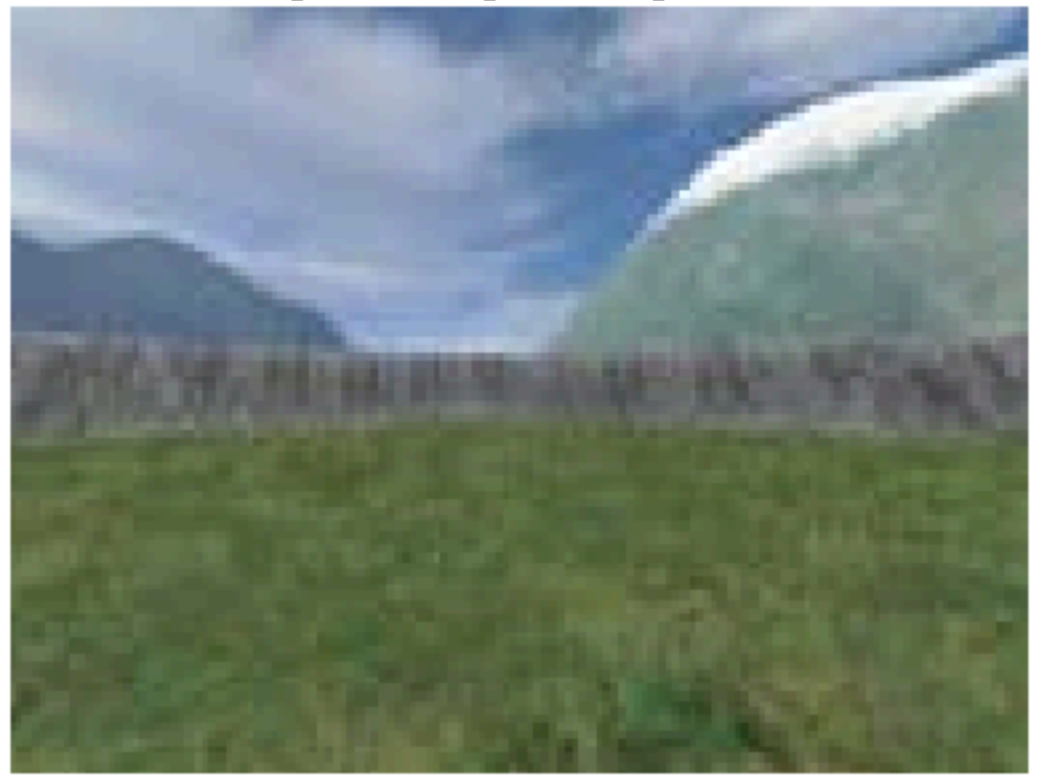
Spatial trajectory (Birds eye)

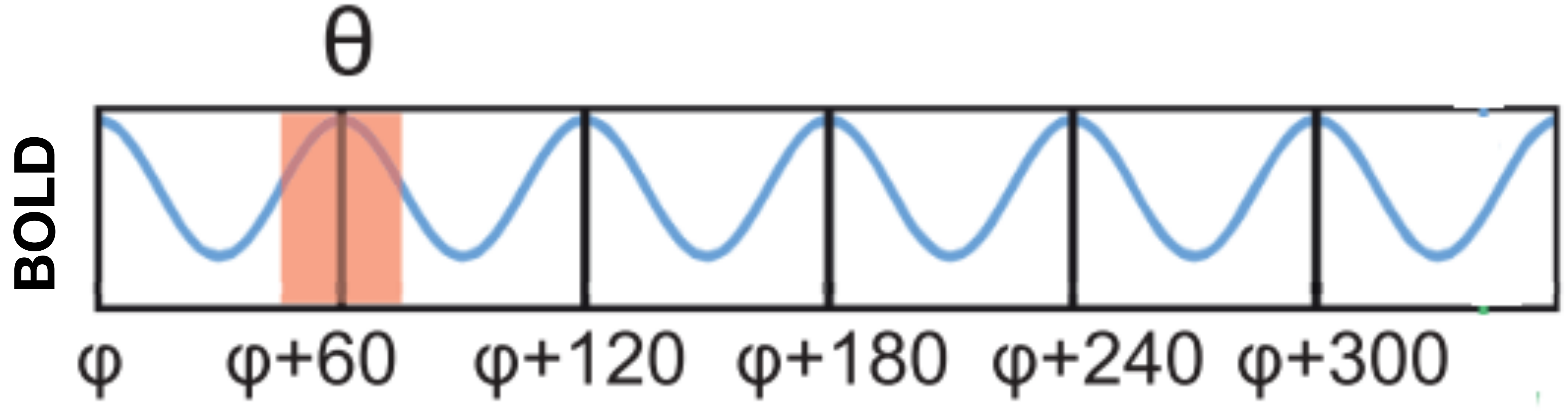






Participant perspective

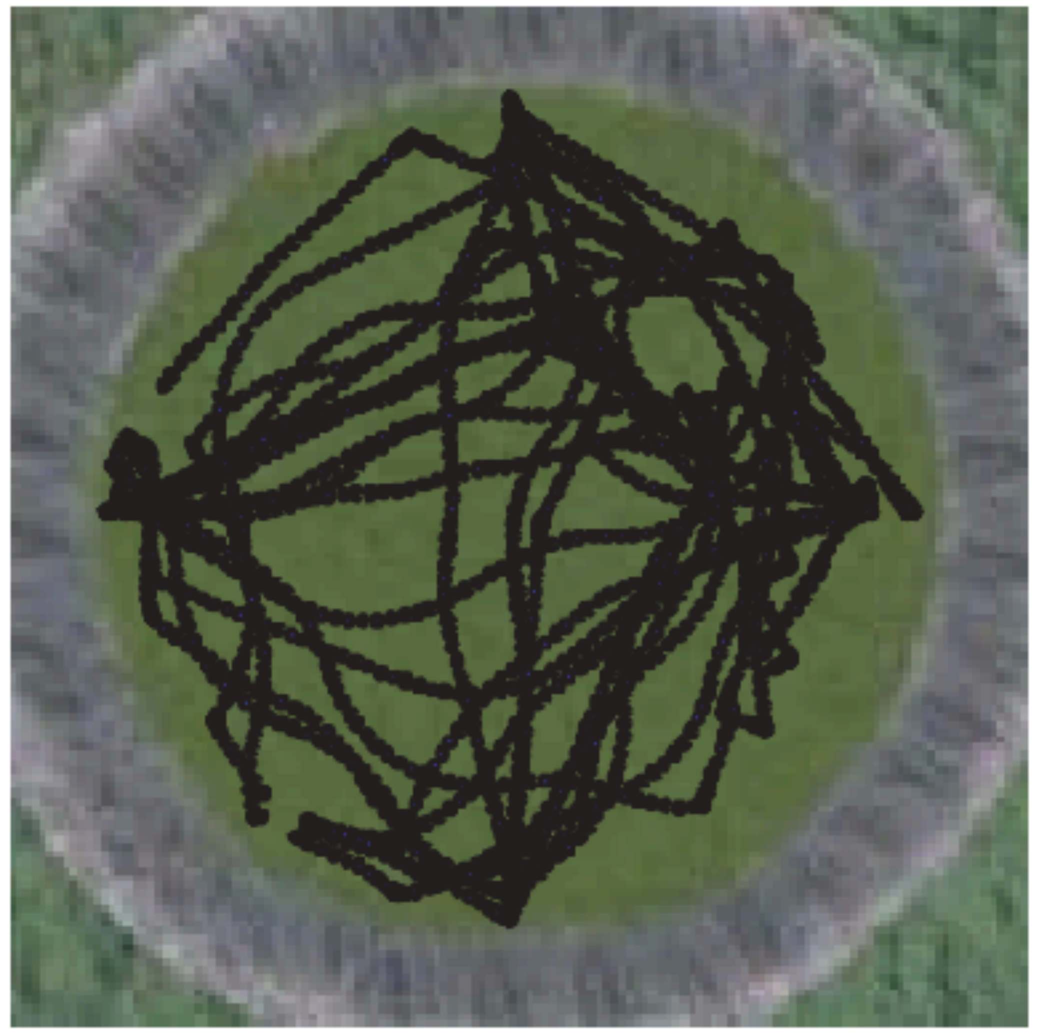


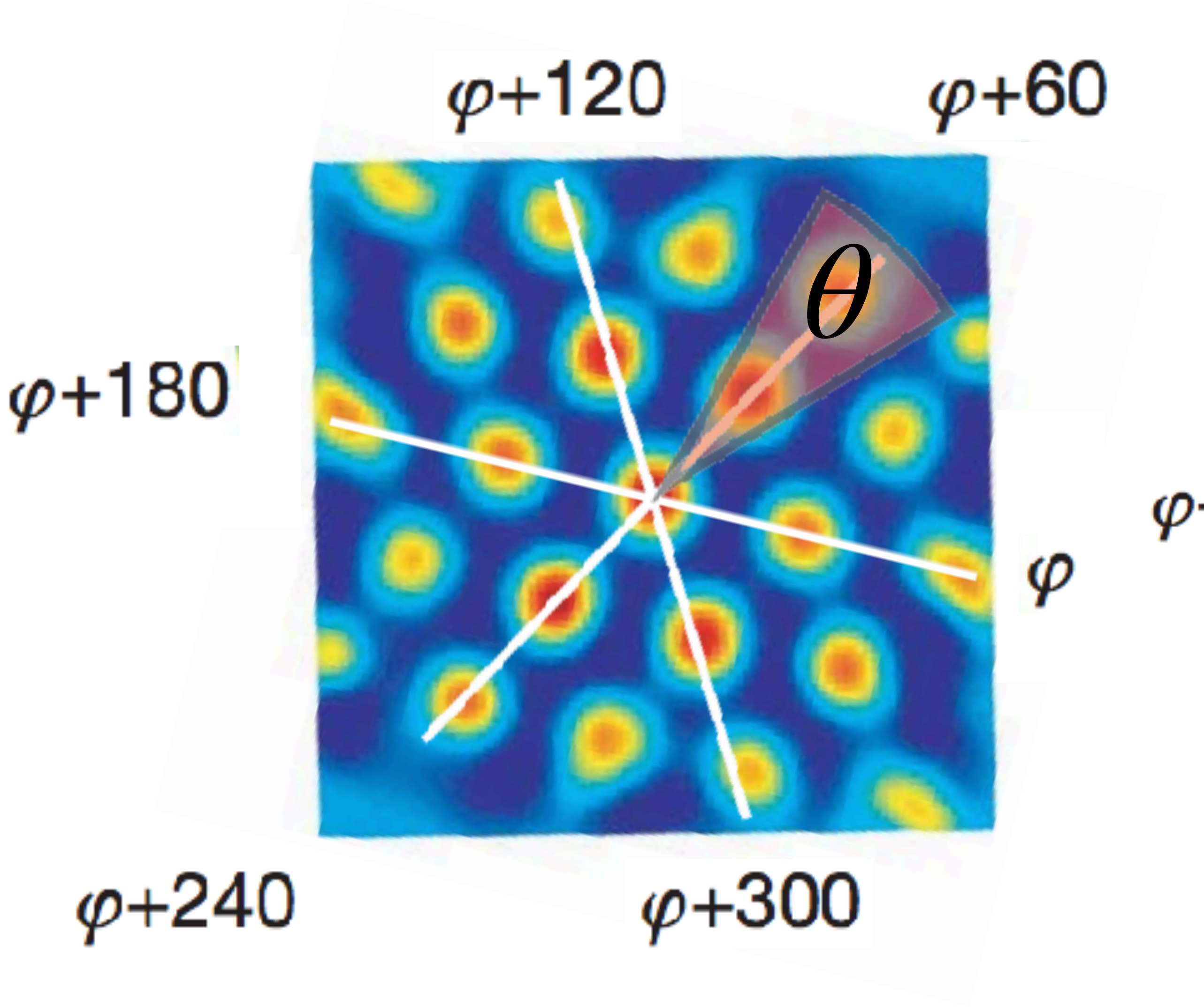


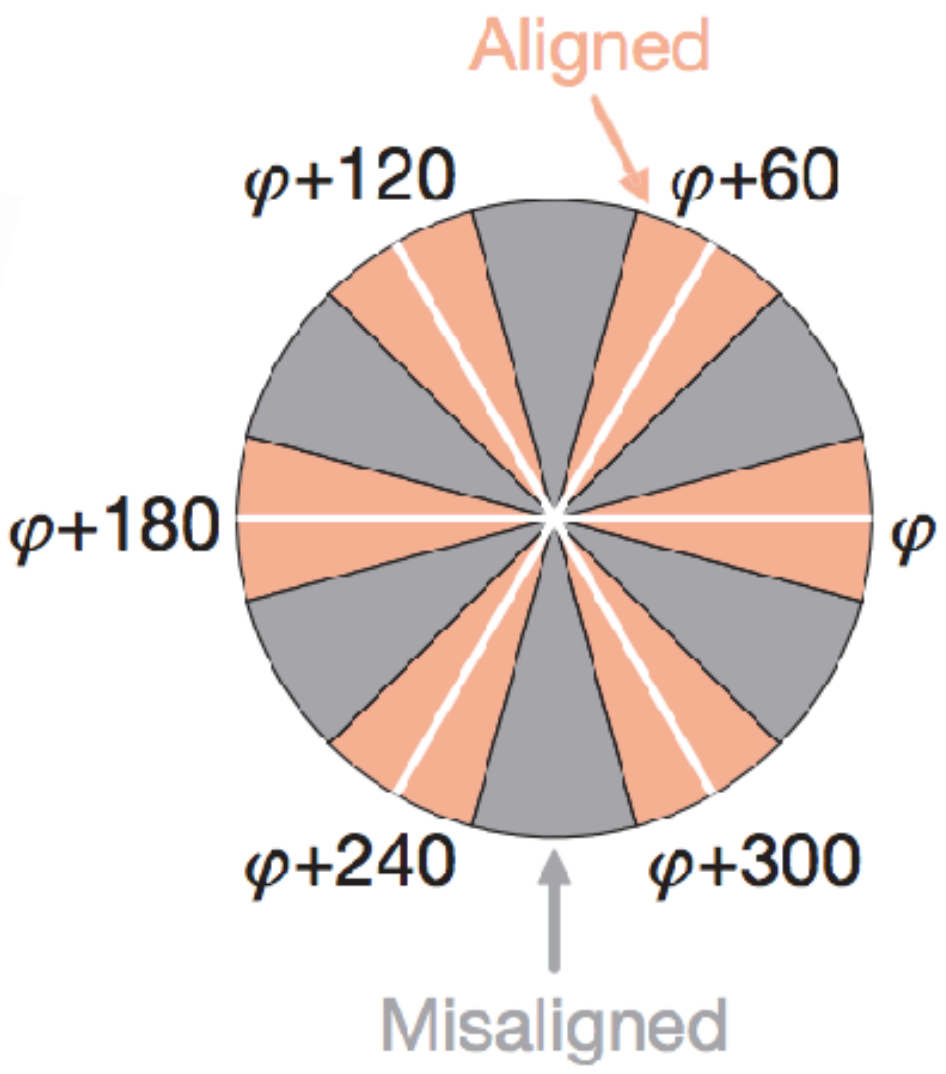
Doeller, Barry, & Burgess (*Nature*, 2010)

We can measure trajectories in human navigation

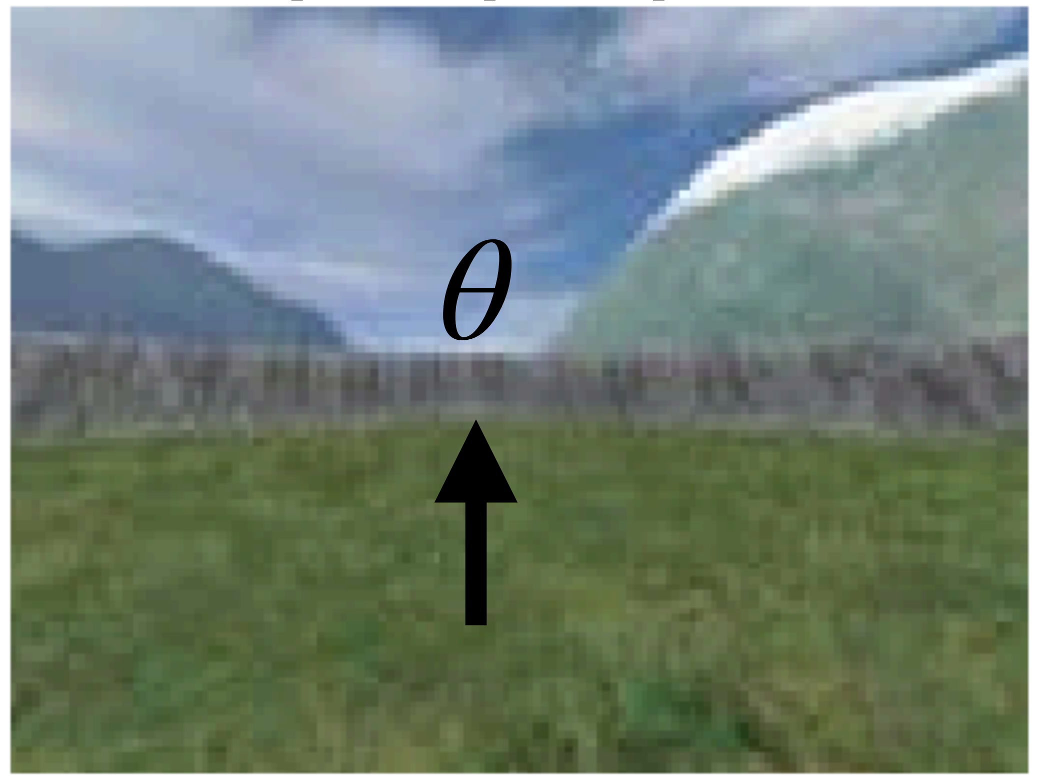
Spatial trajectory (Birds eye)

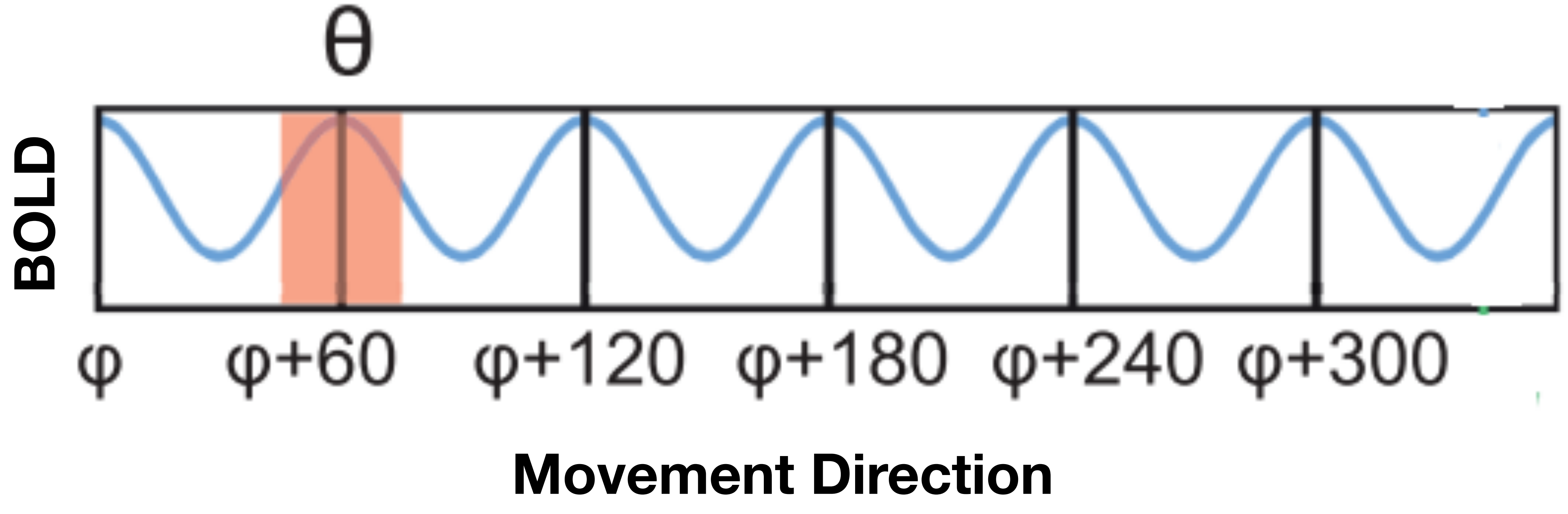






Participant perspective

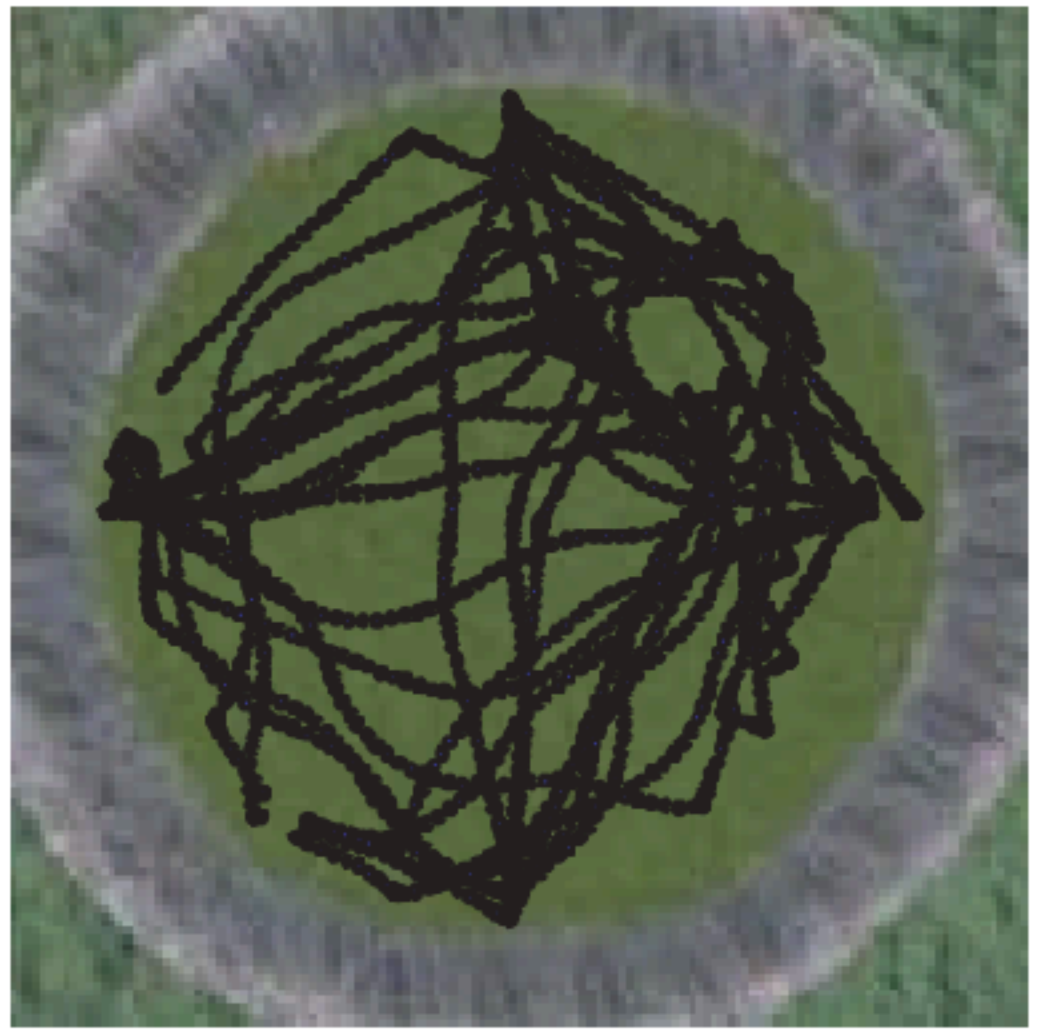




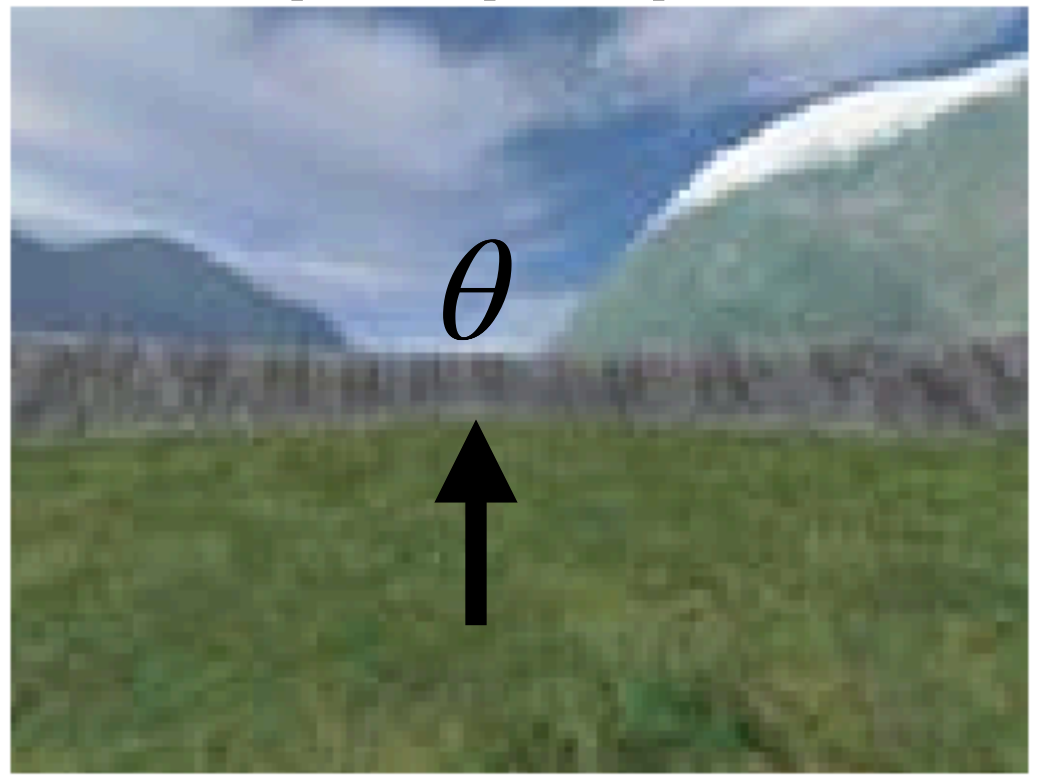
Doeller, Barry, & Burgess (*Nature*, 2010)

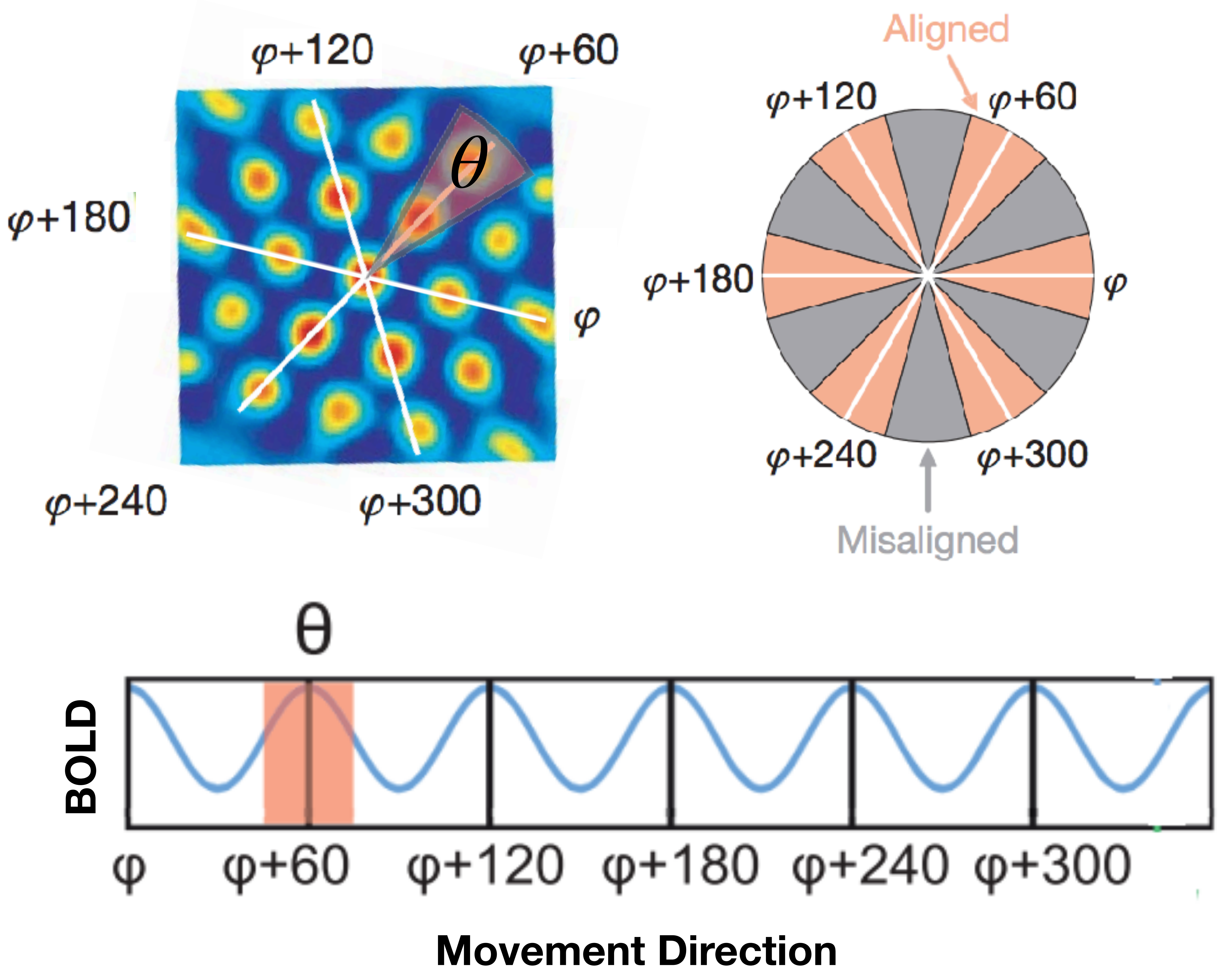
We can measure trajectories in human navigation

Spatial trajectory (Birds eye)



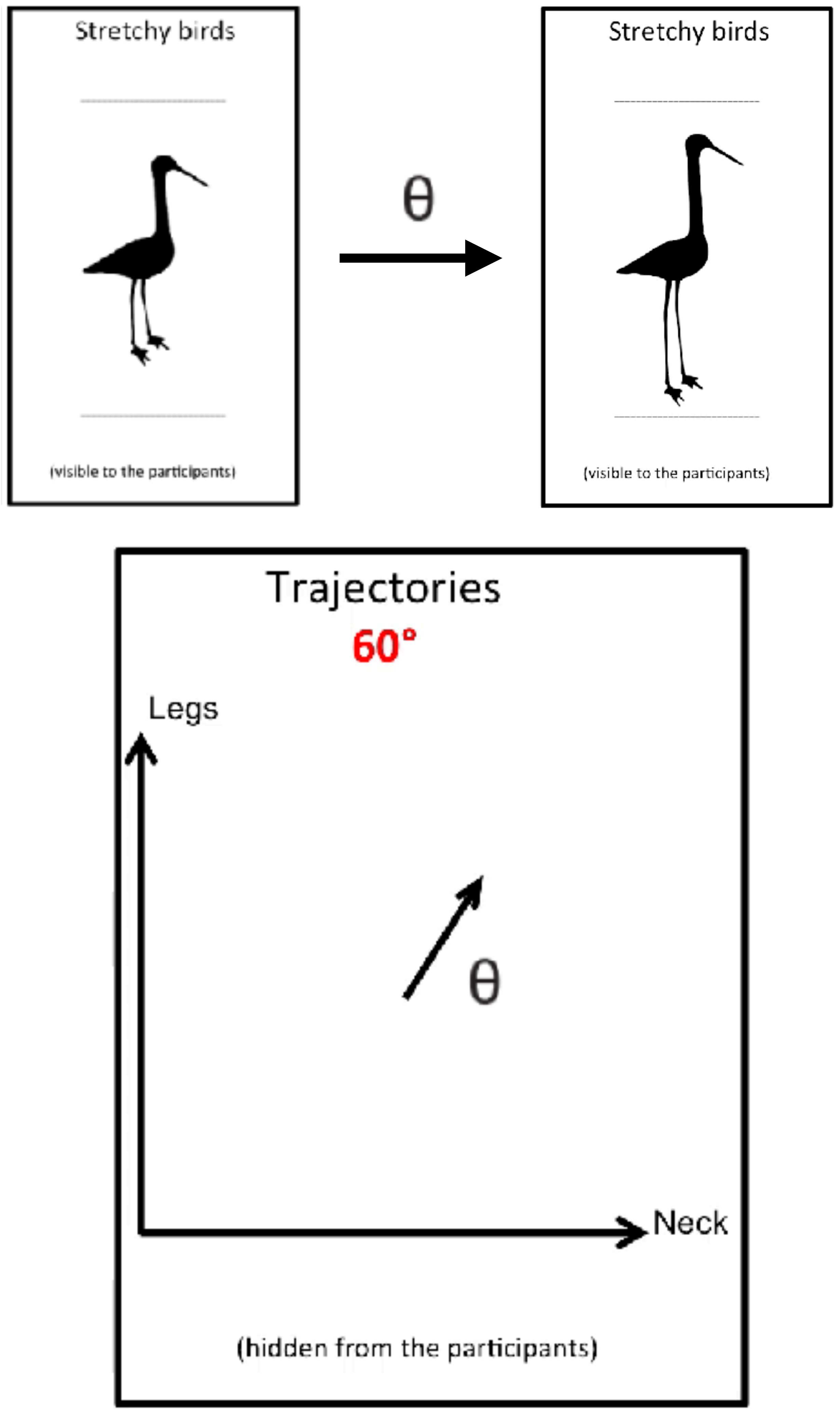
Participant perspective





Doeller, Barry, & Burgess (*Nature*, 2010)

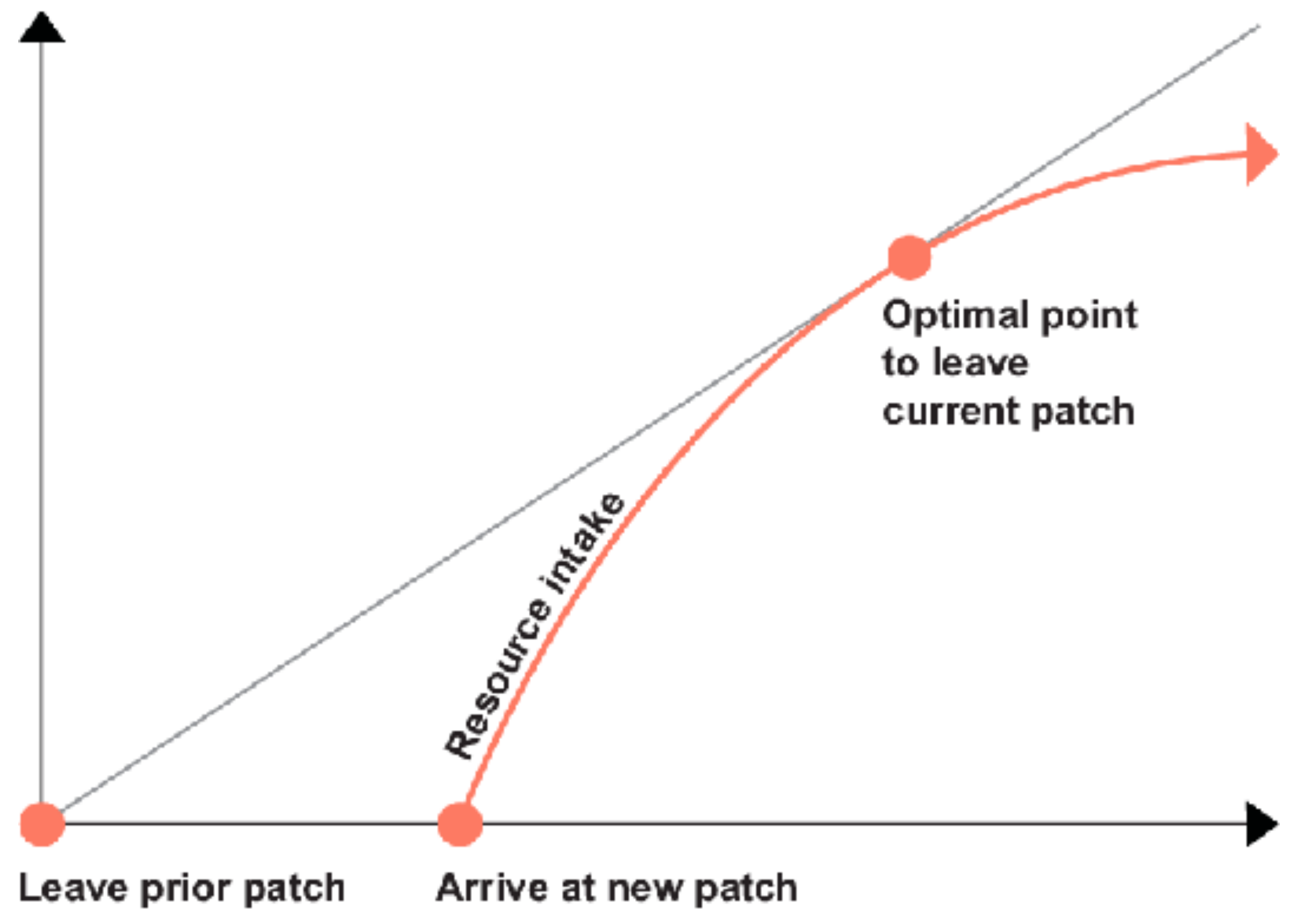
Also in non-spatial domains!

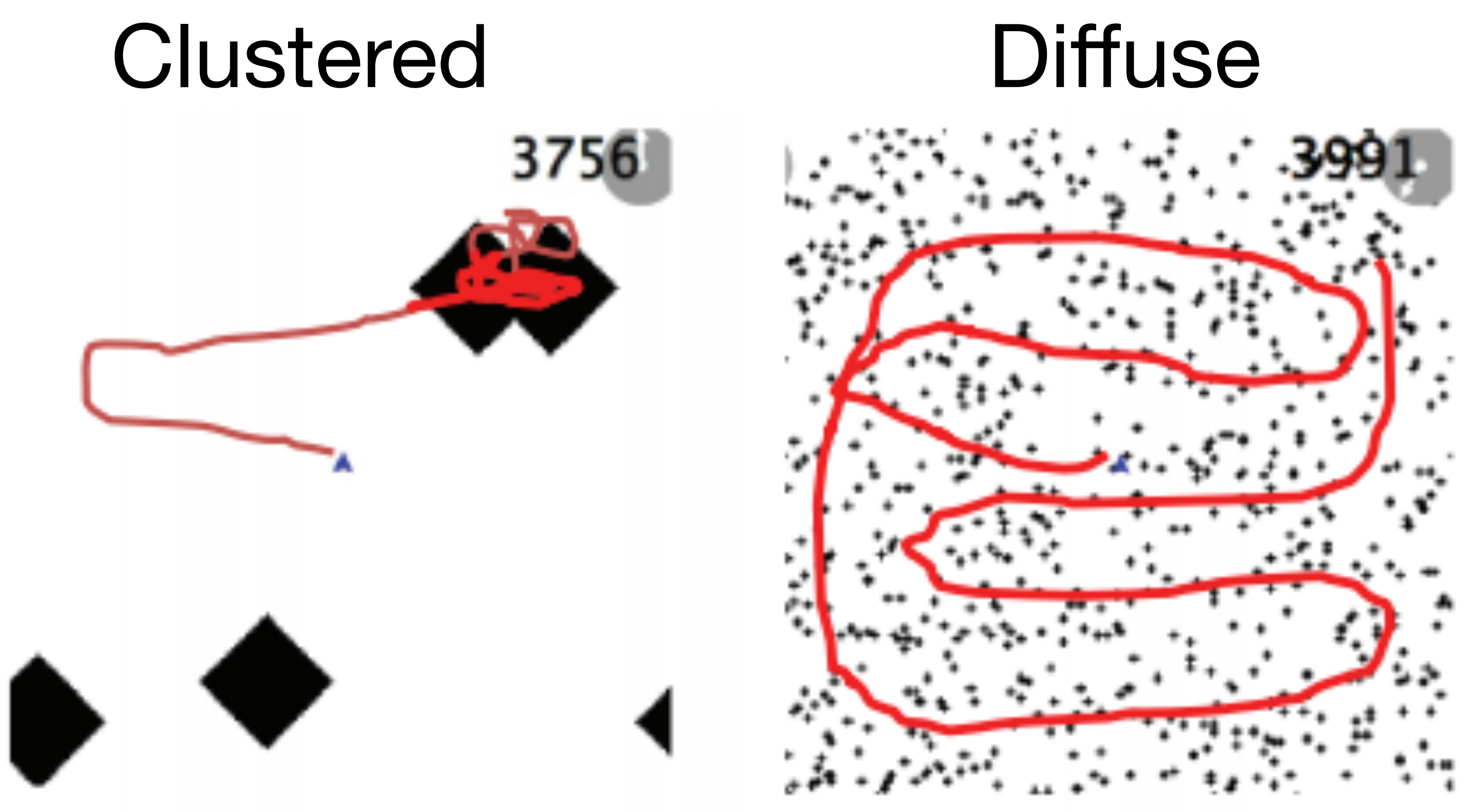


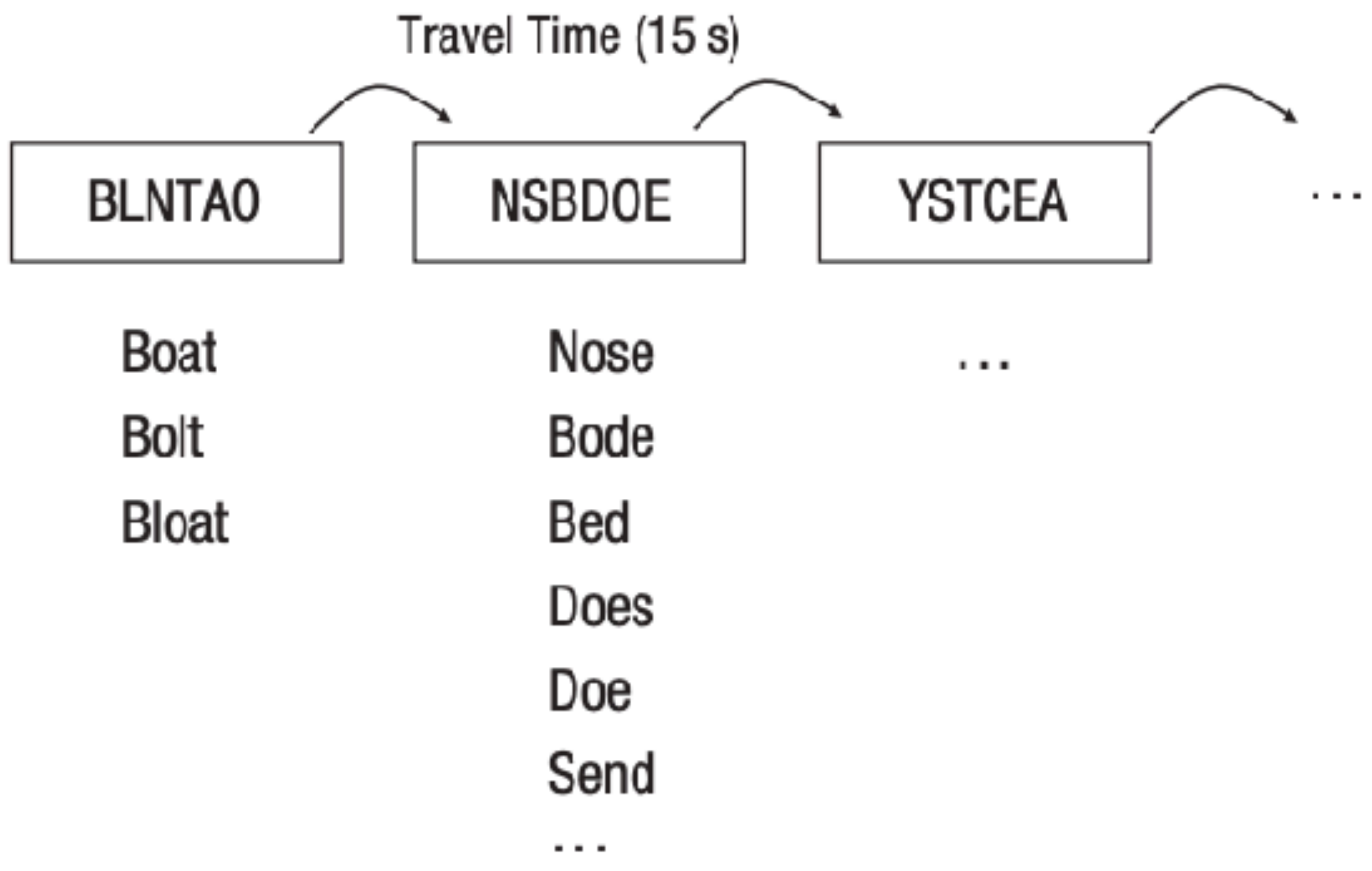
Constantinescu, O'Reilly, & Behrens (*Science*, 2016)

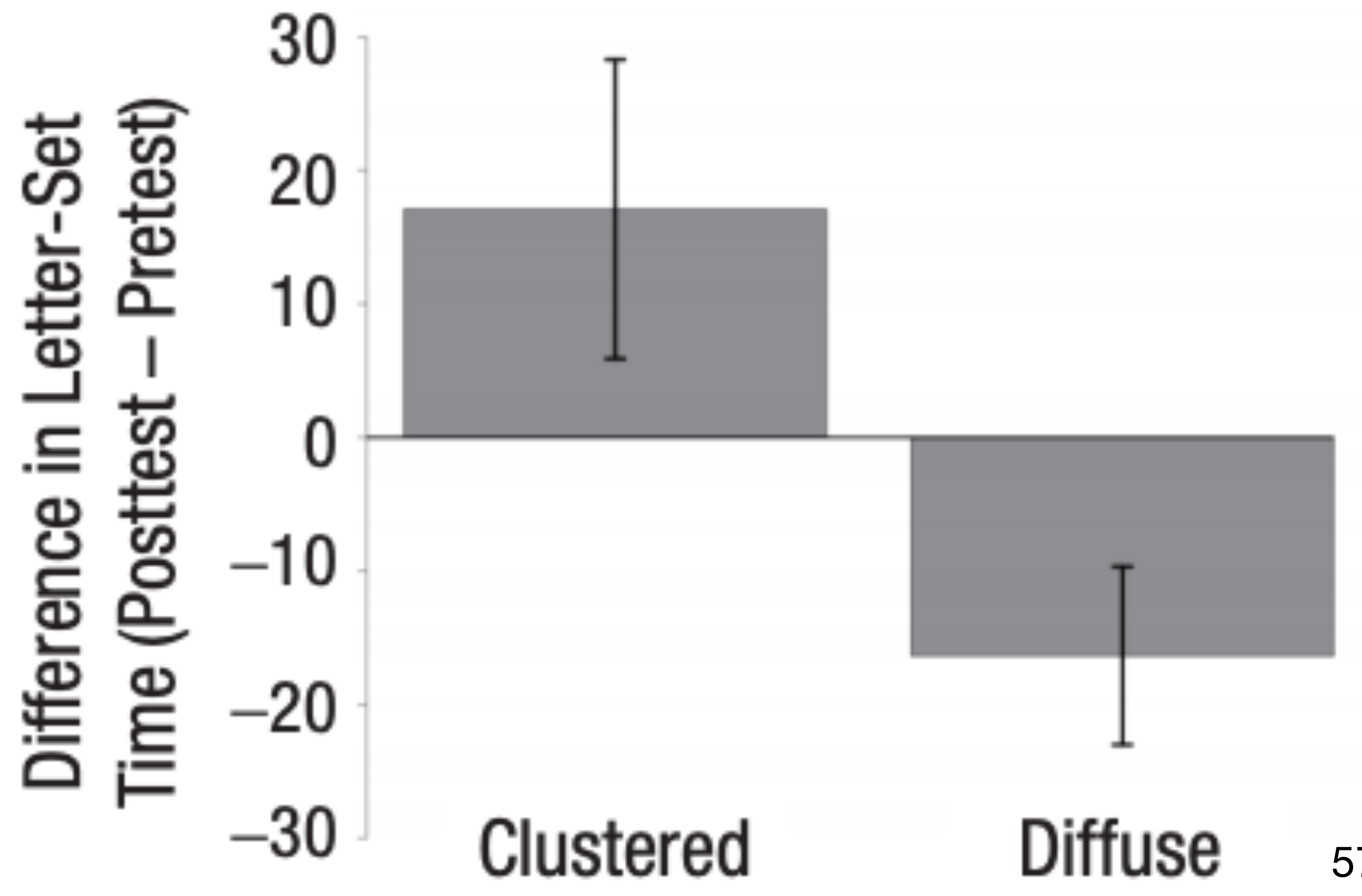
Spatial Rewards Influence Semantic Foraging

- Search in external and internal spaces follow similar principles of optimal foraging
  Charnov (1976); Pirolli & Card (1999)
- The distribution of resources in a spatial foraging task can influence semantic search patterns in a word generation task
  Hills, Todd, & Goldstone (2008)
- “Exaptation” of spatial cognition to other domains
  Hills (2006); Hills, Todd, & Goldstone (2008)







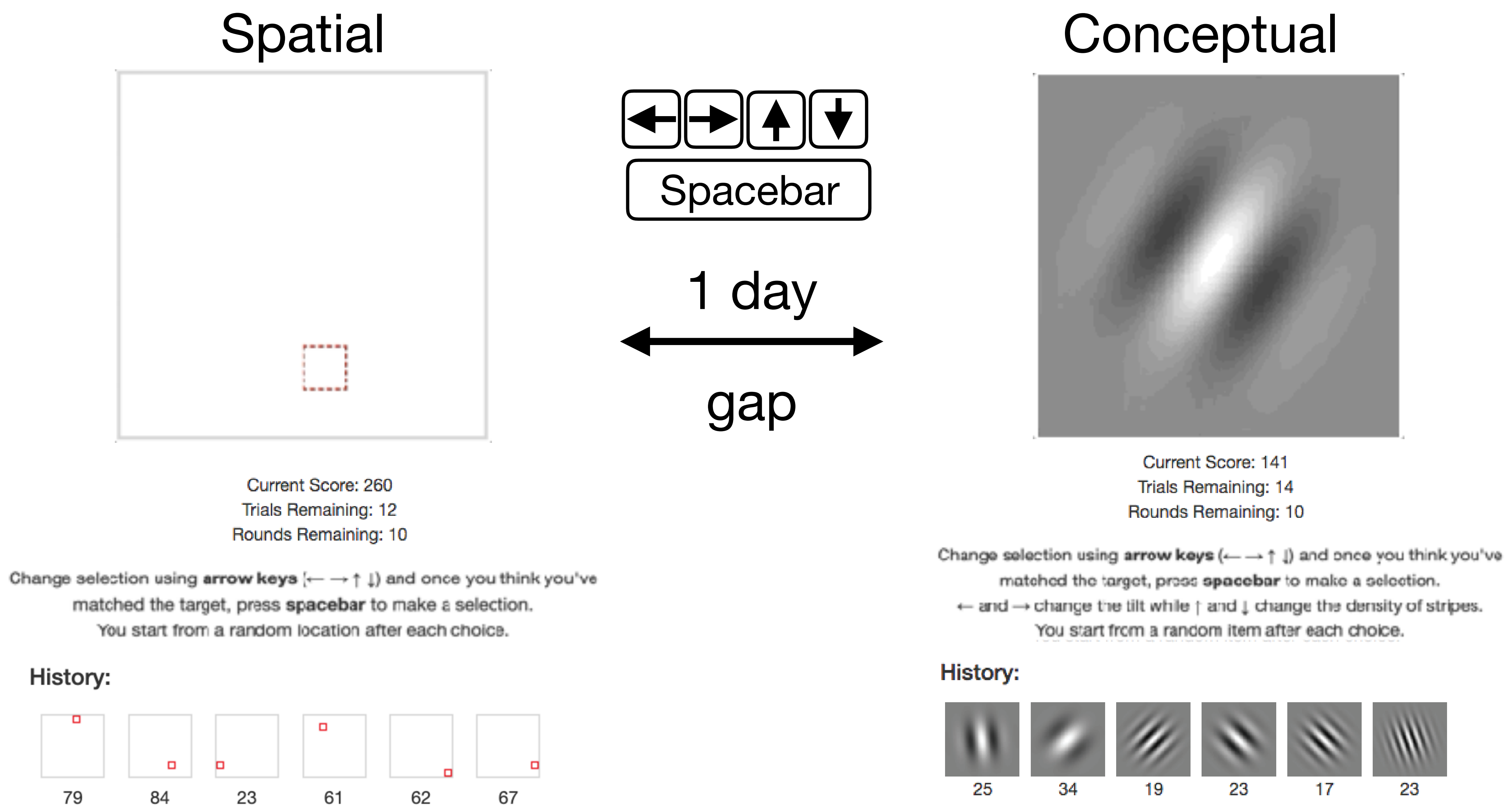


Connecting Spatial and Conceptual Search

- Since there is evidence for a common neural representation for both spatial and conceptual navigation, what are the downstream implications for behavior?

- Are there domain general principles for generalization (about novel stimuli) and exploration (in new environments)?

- Within-subject experiment, where participants used either spatial or conceptual features to guide the search for rewards



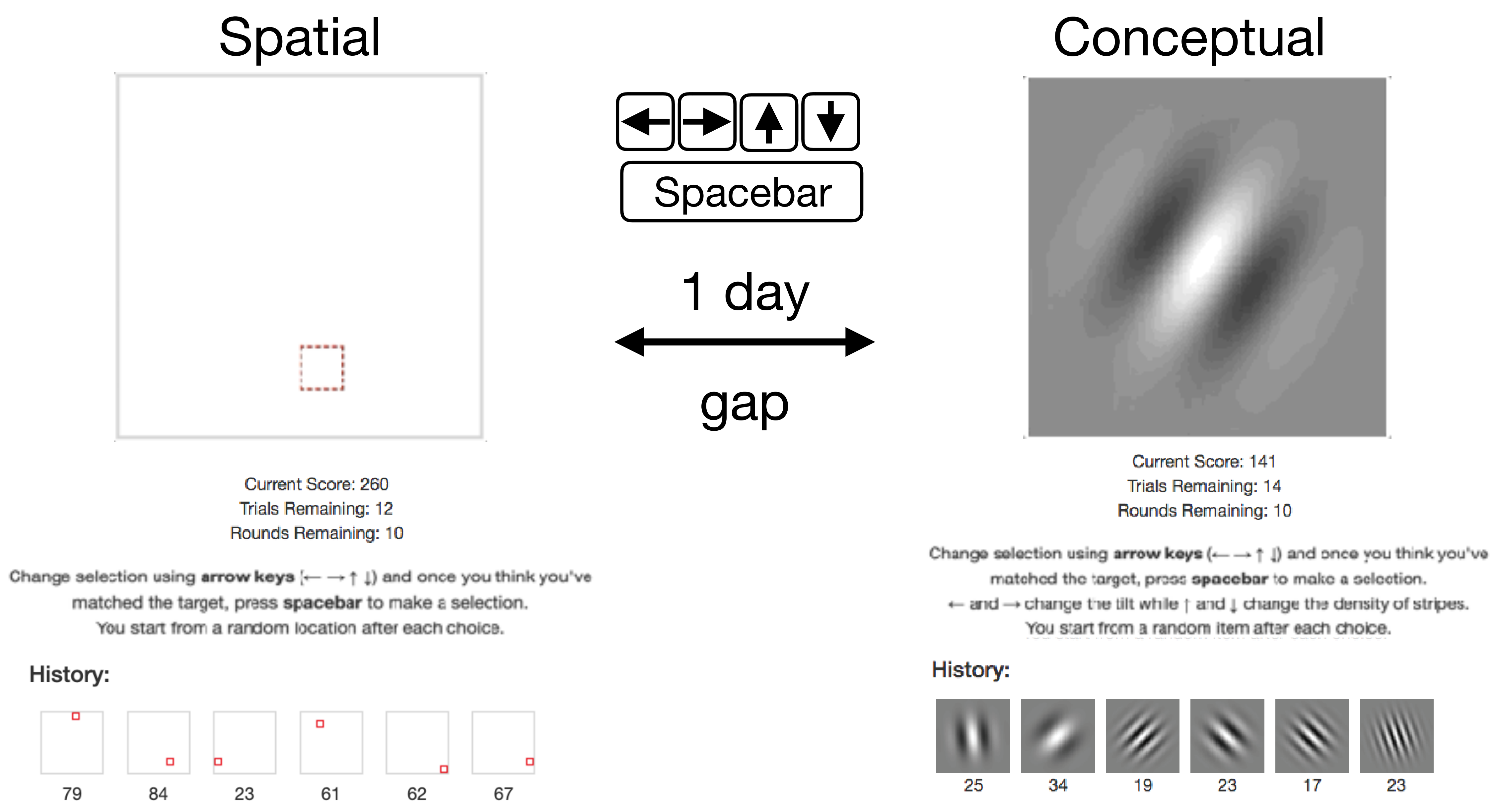
Wu, Schulz, Garvet, Meder & Schuck (*PLOS Comp Bio* 2020)

Connecting Spatial and Conceptual Search

- Since there is evidence for a common neural representation for both spatial and conceptual navigation, what are the downstream implications for behavior?

- Are there domain general principles for generalization (about novel stimuli) and exploration (in new environments)?

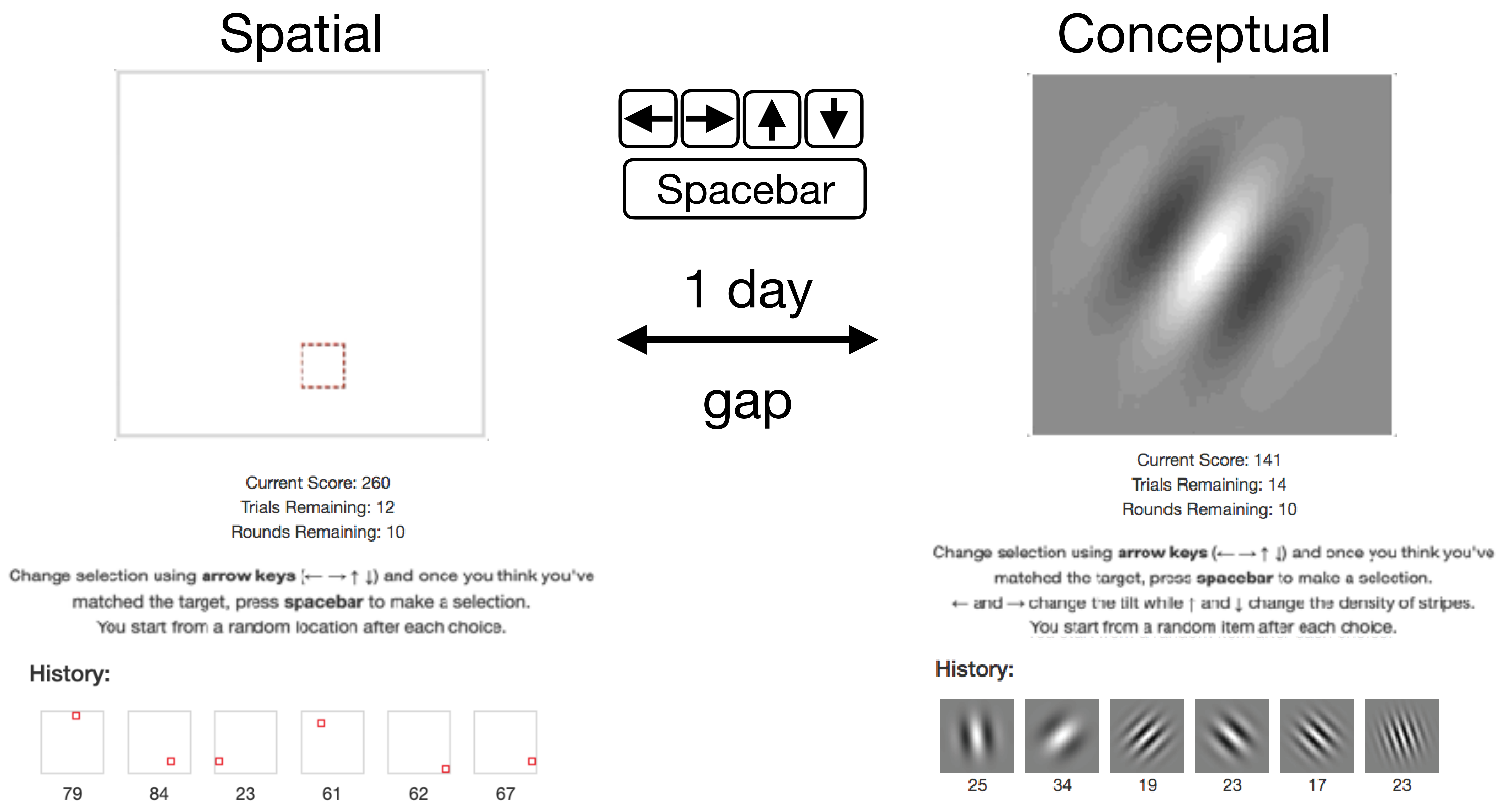
- Within-subject experiment, where participants used either spatial or conceptual features to guide the search for rewards



Wu, Schulz, Garvet, Meder & Schuck (*PLOS Comp Bio* 2020)

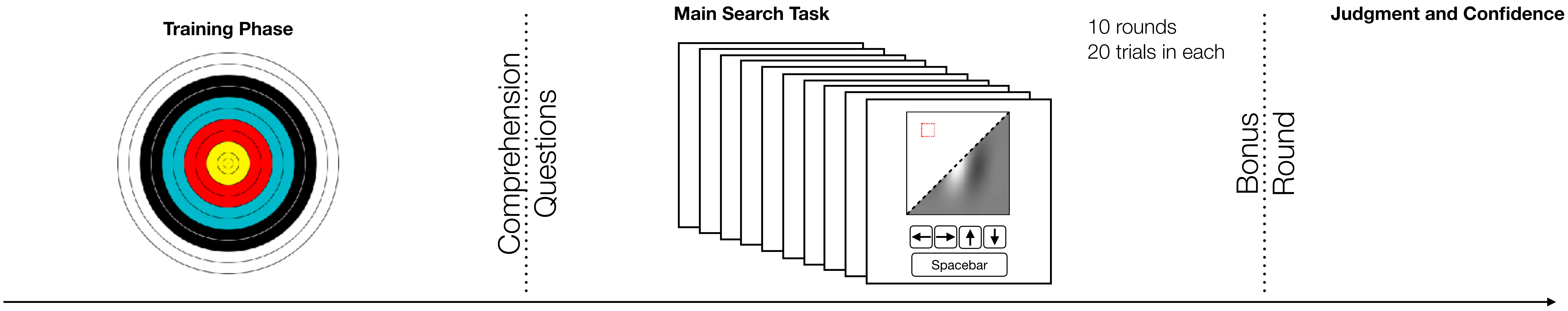
Connecting Spatial and Conceptual Search

- Since there is evidence for a common neural representation for both spatial and conceptual navigation, what are the downstream implications for behavior?
- Are there domain general principles for generalization (about novel stimuli) and exploration (in new environments)?
- Within-subject experiment, where participants used either spatial or conceptual features to guide the search for rewards

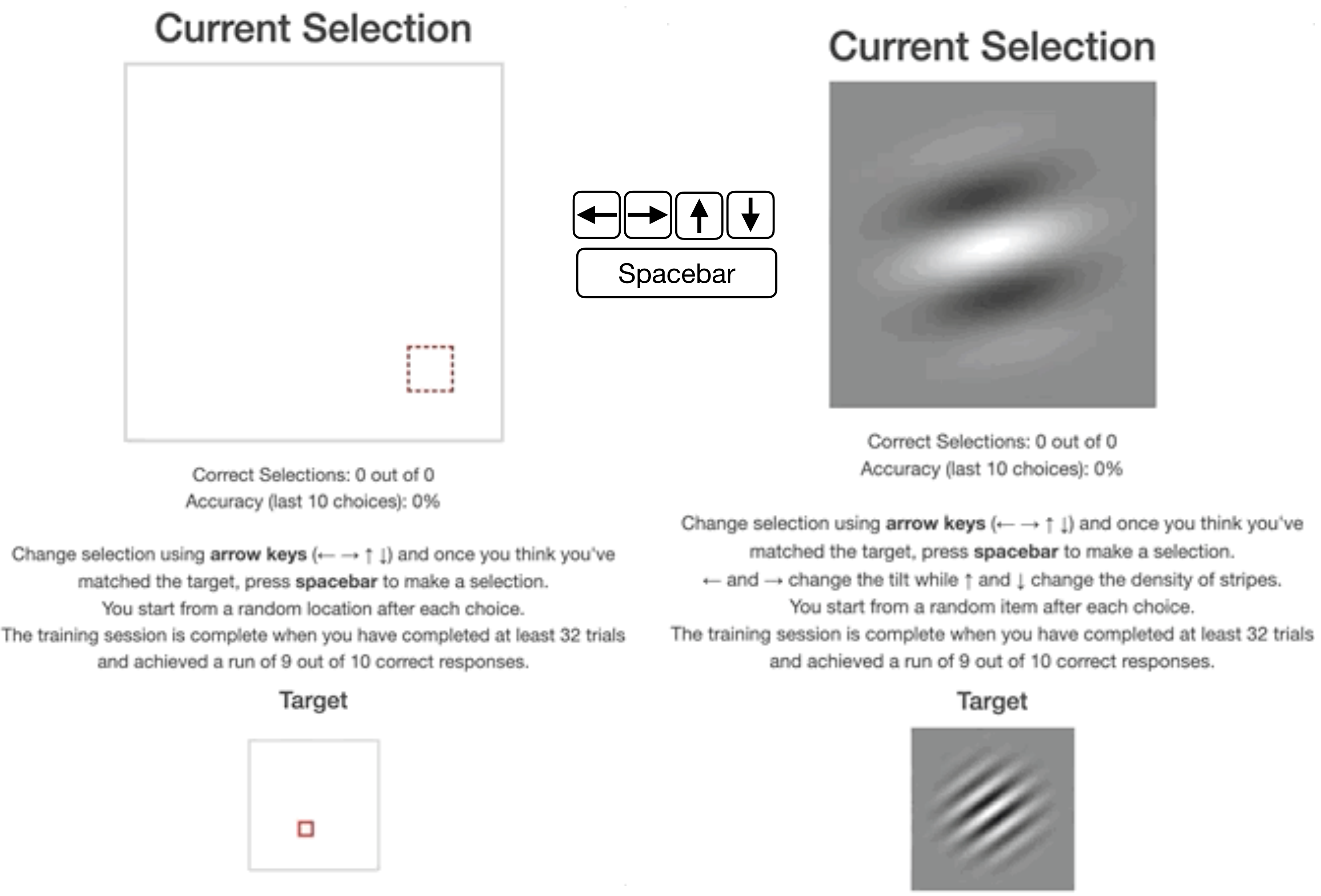


Wu, Schulz, Garvet, Meder & Schuck (*PLOS Comp Bio* 2020)

Task Design

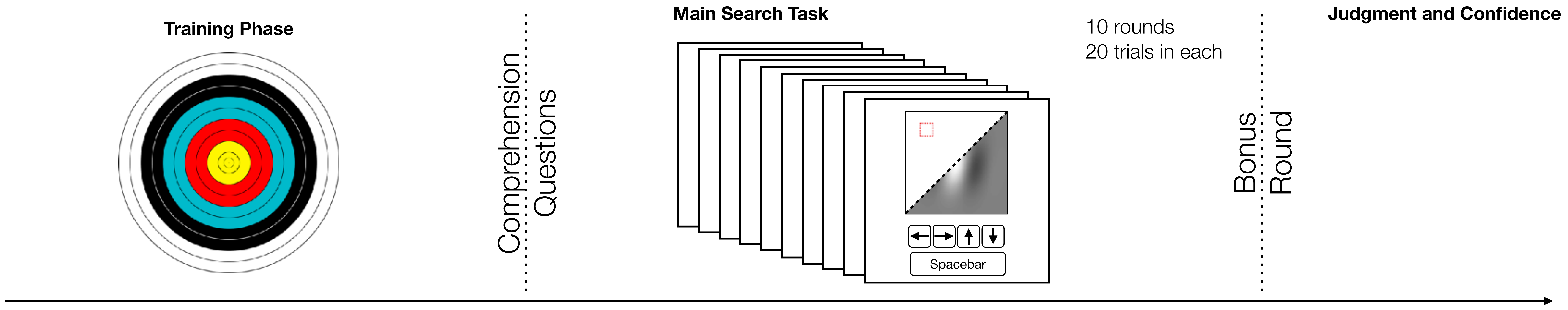


Match target stimuli until learning criterion reached

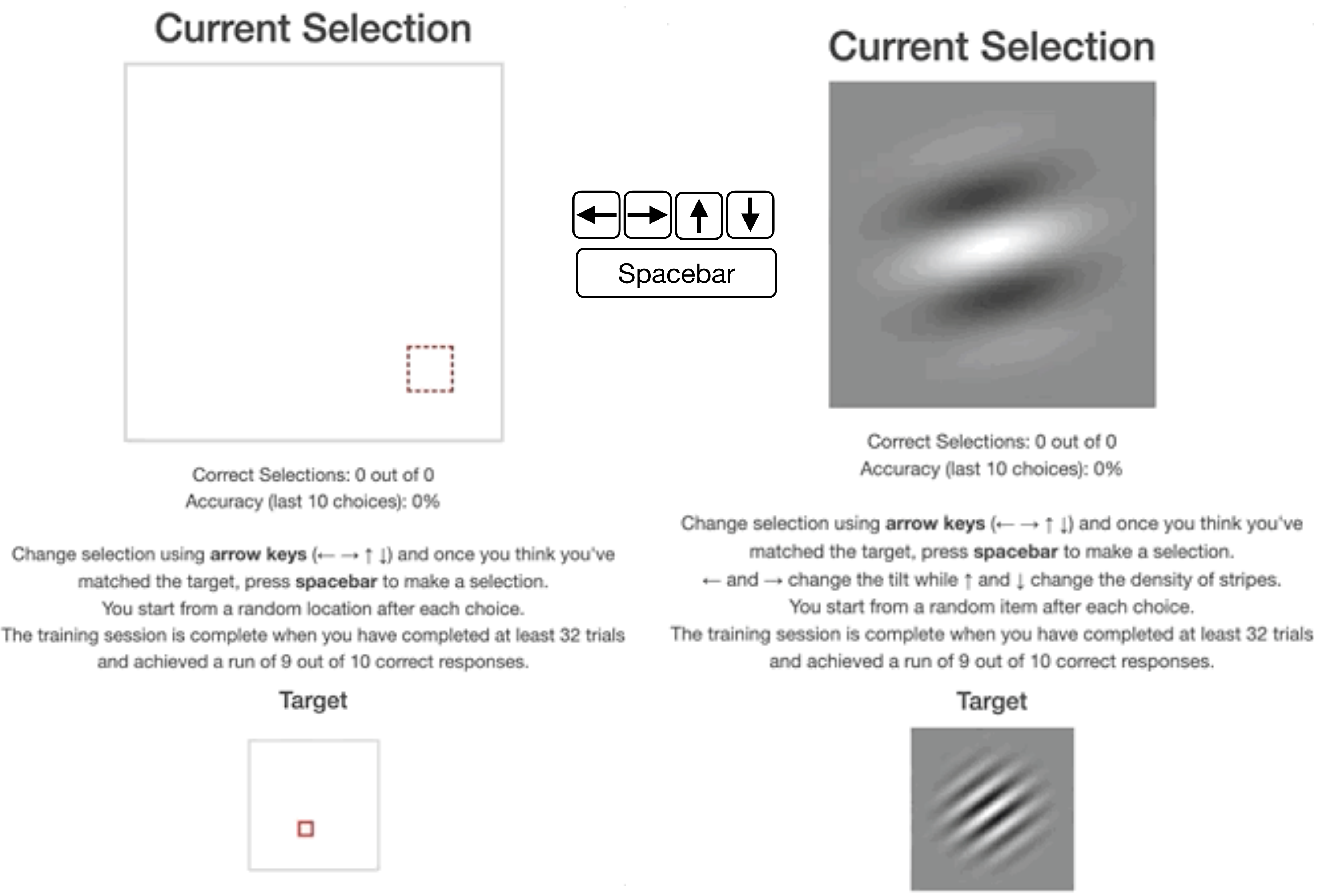


At least 32 trials AND a run of 9 out of 10 correct

Task Design

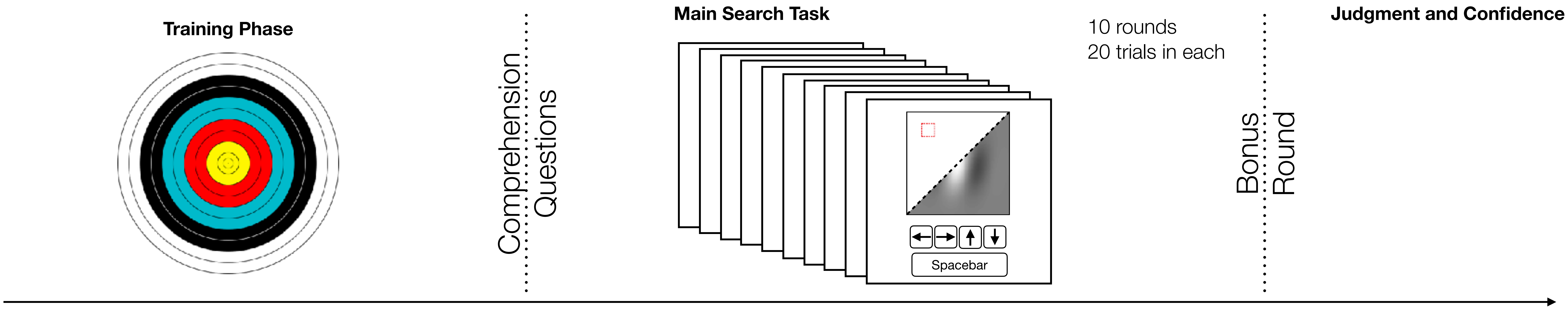


Match target stimuli until learning criterion reached

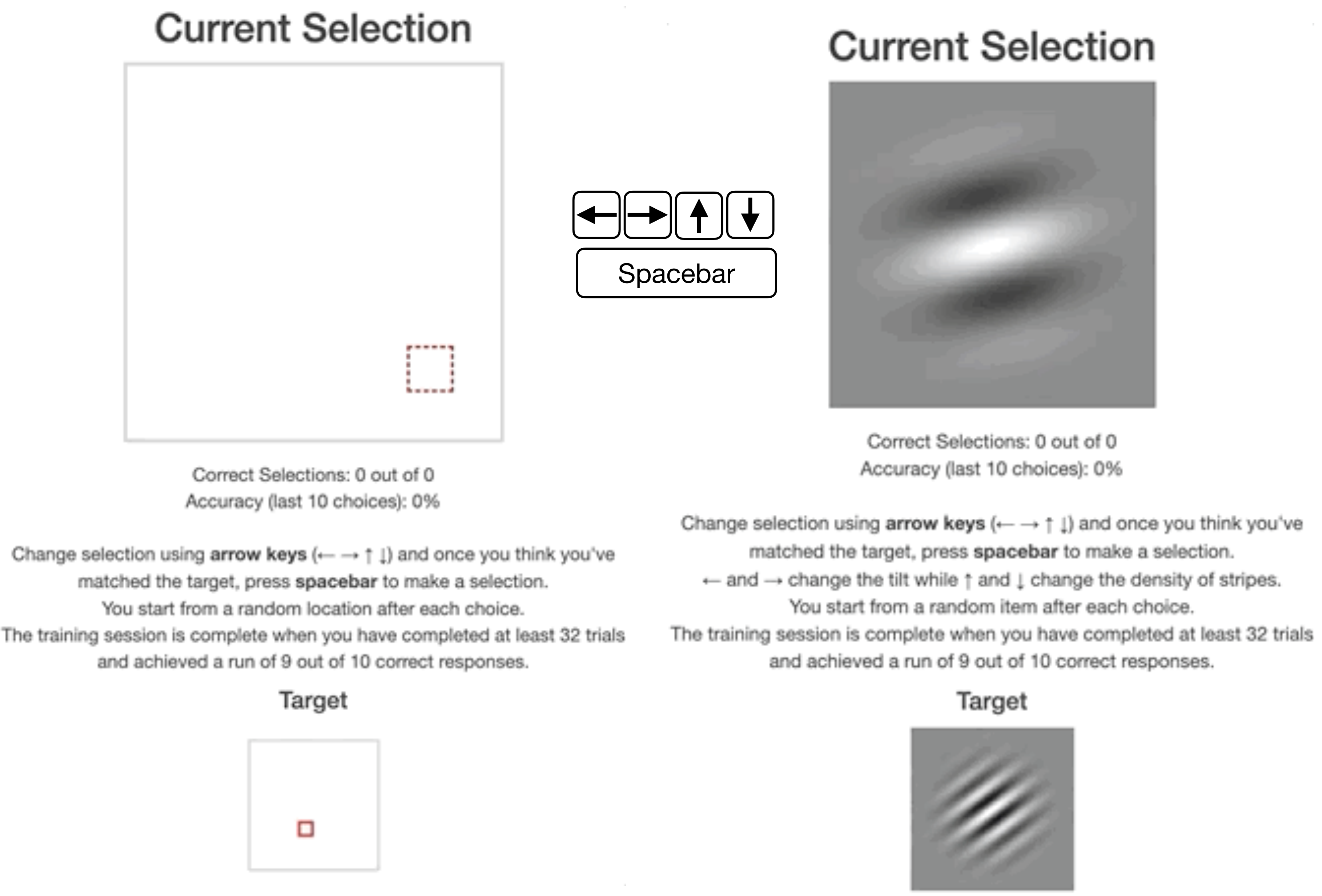


At least 32 trials AND a run of 9 out of 10 correct

Task Design



Match target stimuli until learning criterion reached



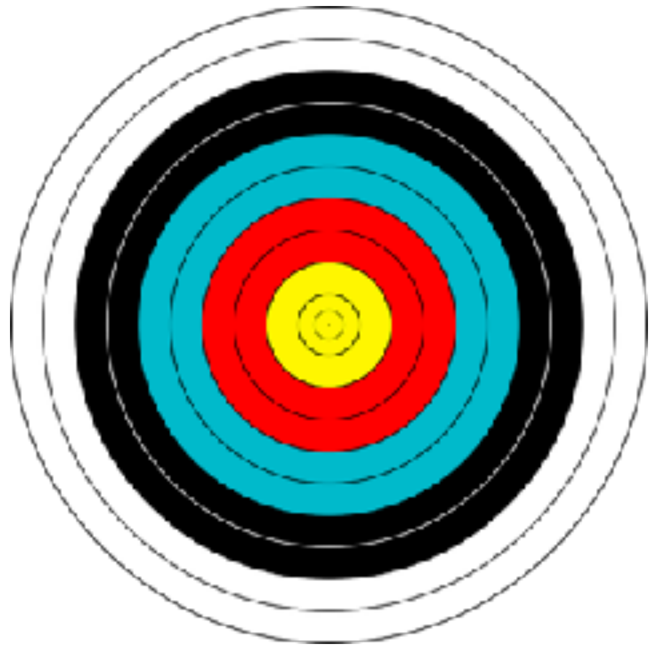
At least 32 trials AND a run of 9 out of 10 correct

Bandit Task:

1. Select stimuli using ← → ↑ ↓
2. Make selection using Spacebar
3. Reward is displayed and then added to history
4. Start at a random stimuli

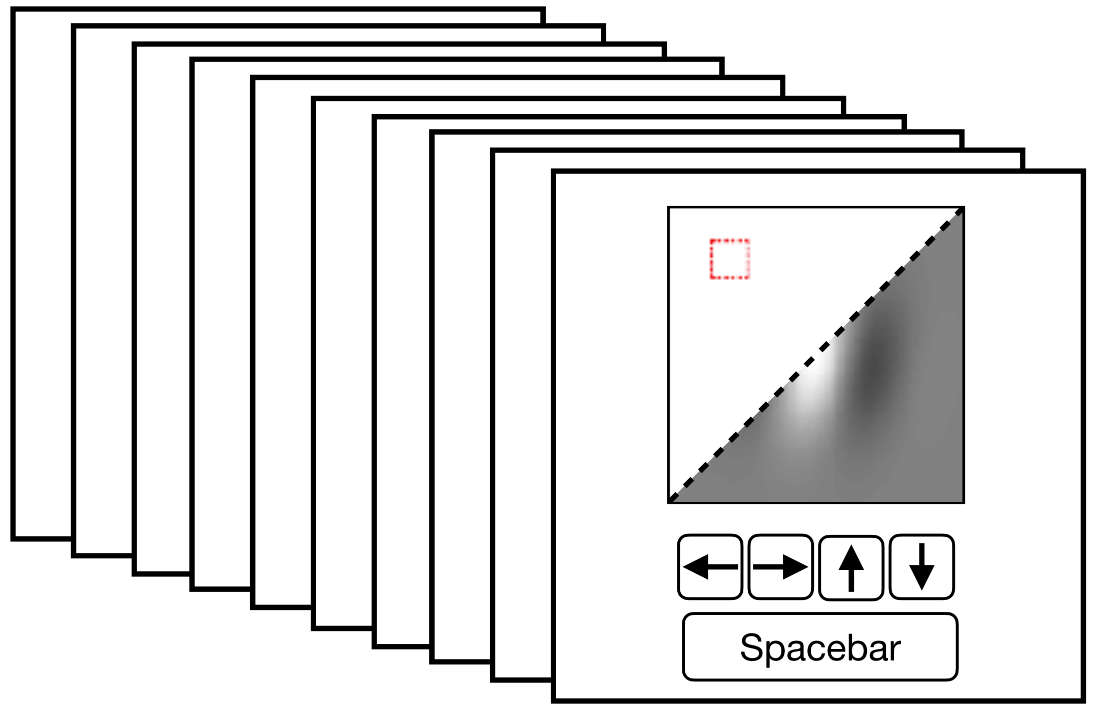
Task Design

Training Phase



Comprehension Questions

Main Search Task

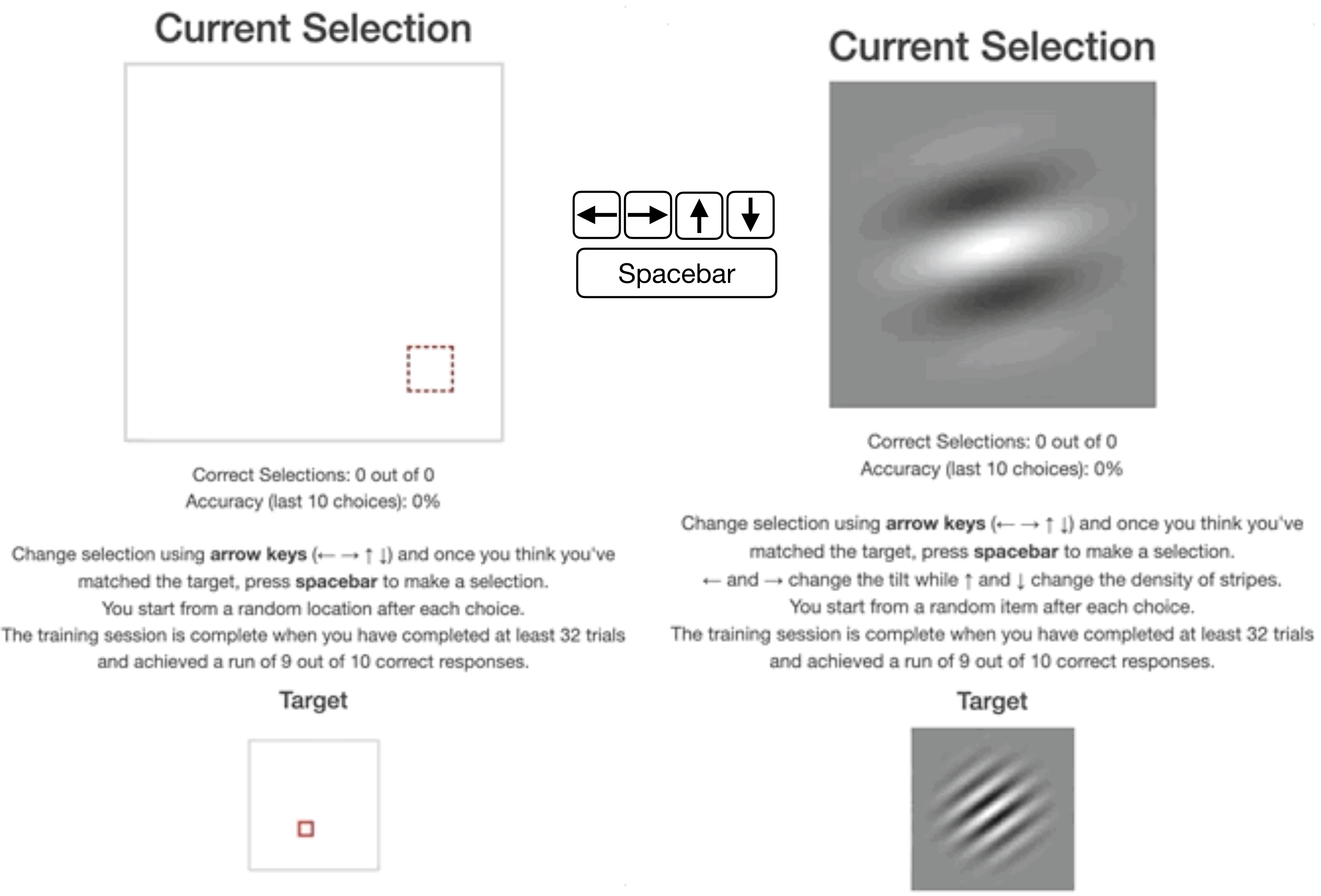


10 rounds
20 trials in each

Bonus Round

Judgment and Confidence

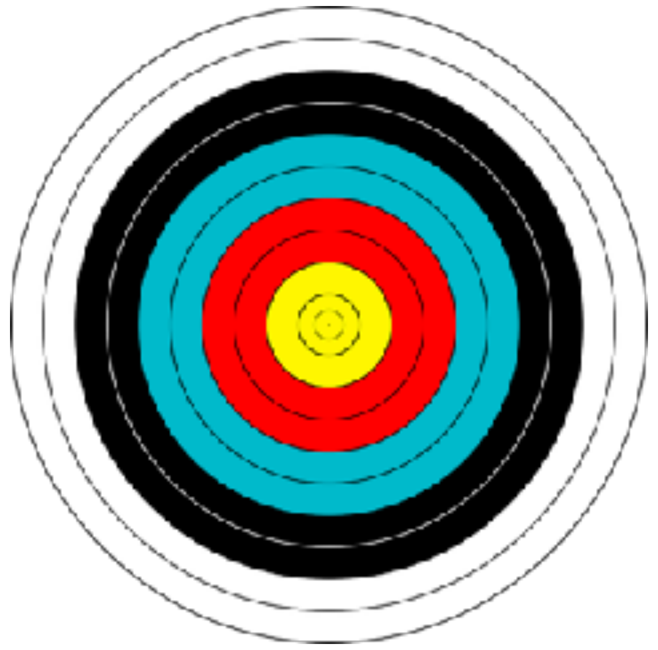
Match target stimuli until learning criterion reached



At least 32 trials AND a run of 9 out of 10 correct

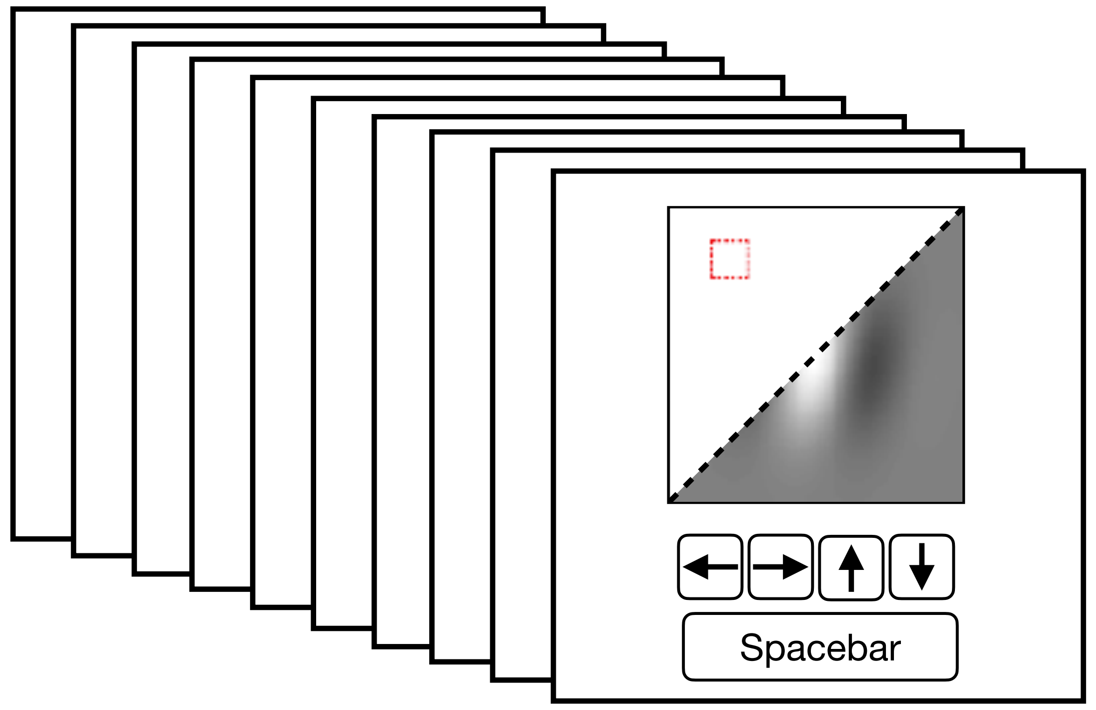
Task Design

Training Phase



Comprehension Questions

Main Search Task

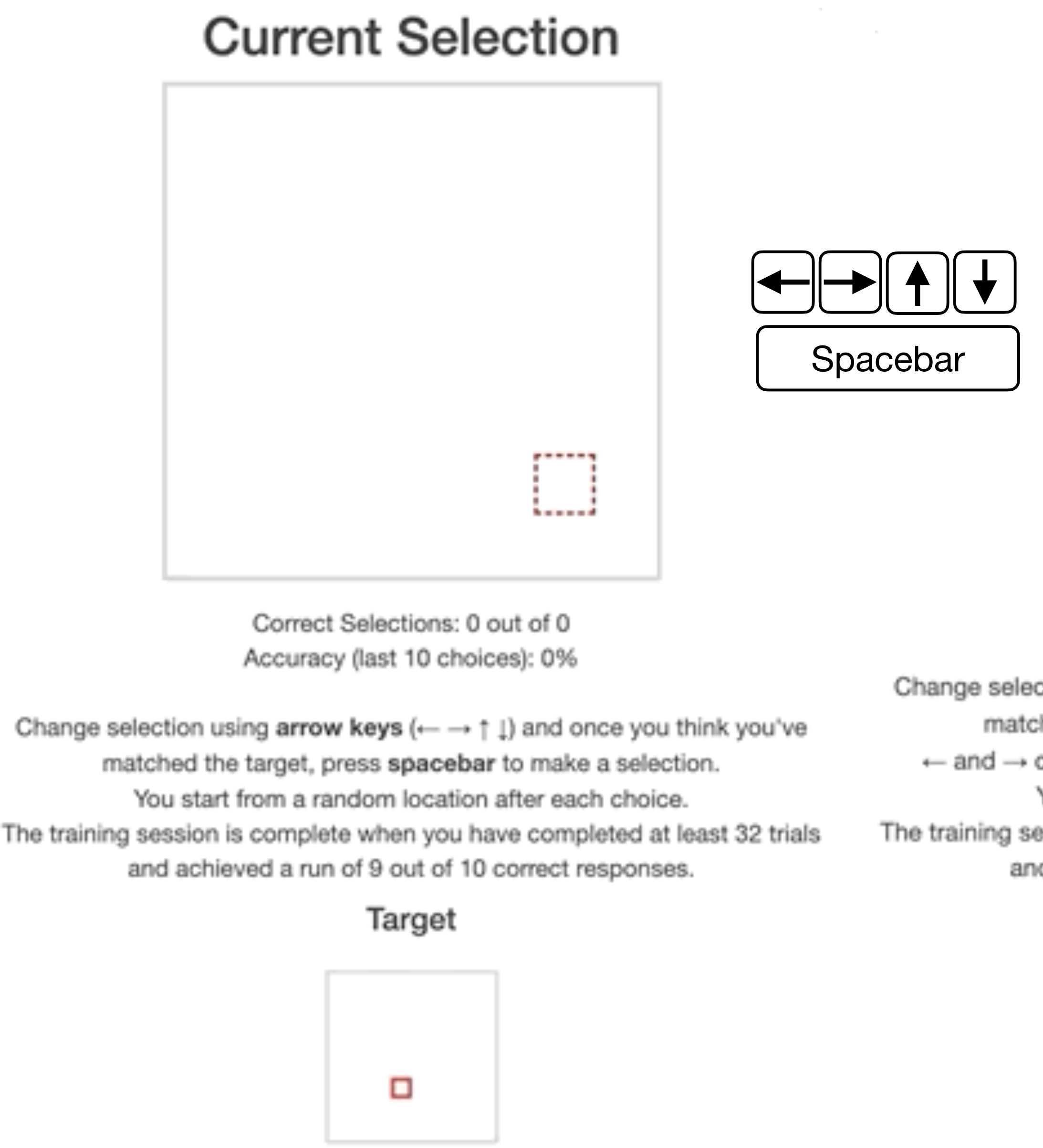


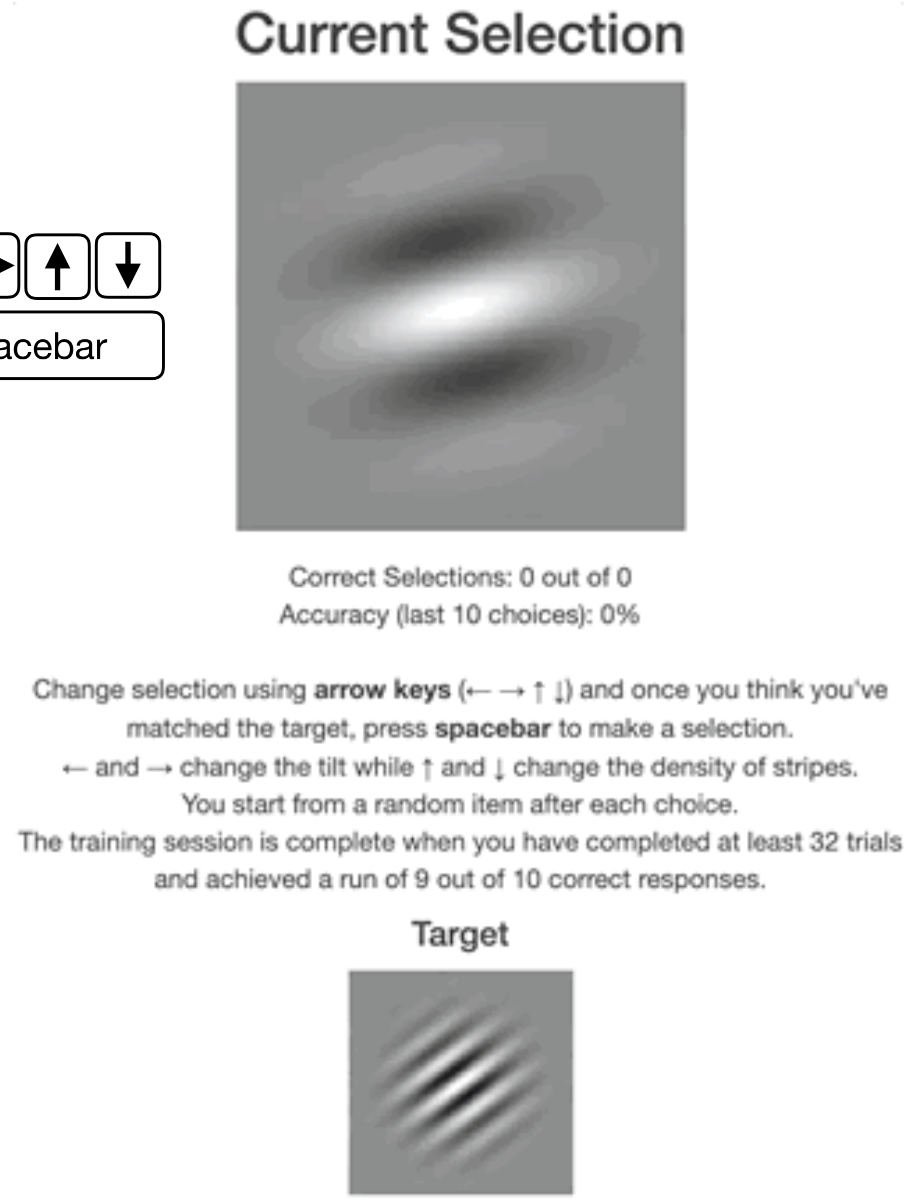
10 rounds
20 trials in each

Bonus Round

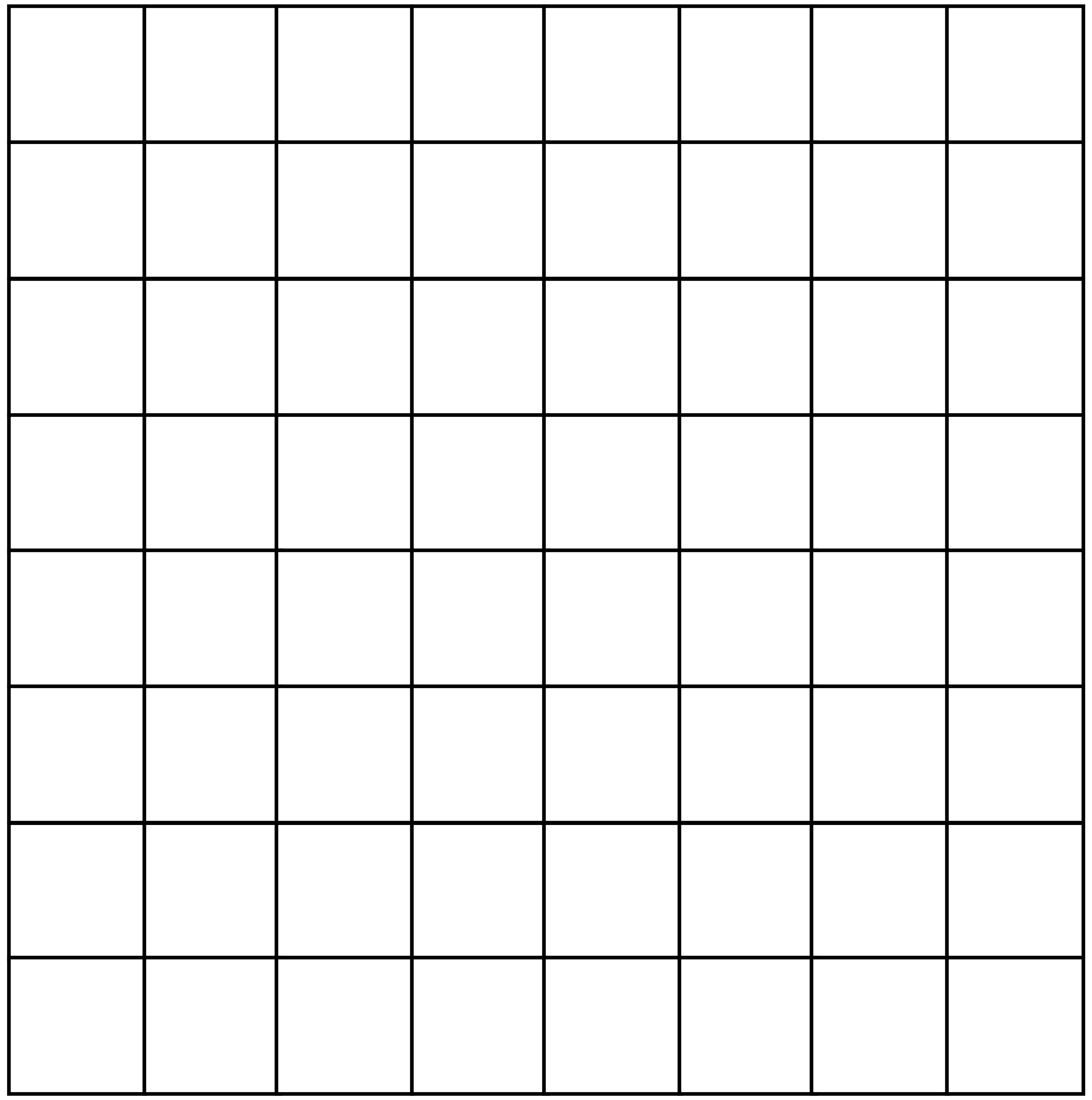
Judgment and Confidence

Match target stimuli until learning criterion reached

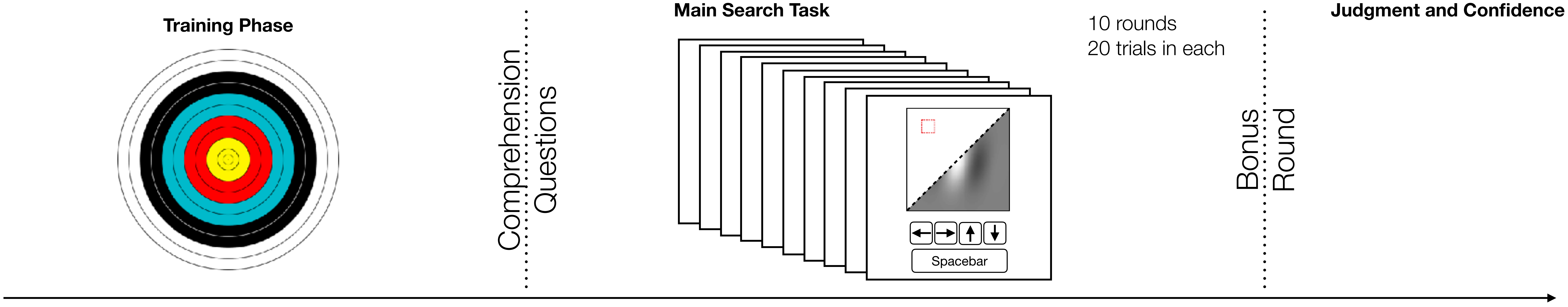




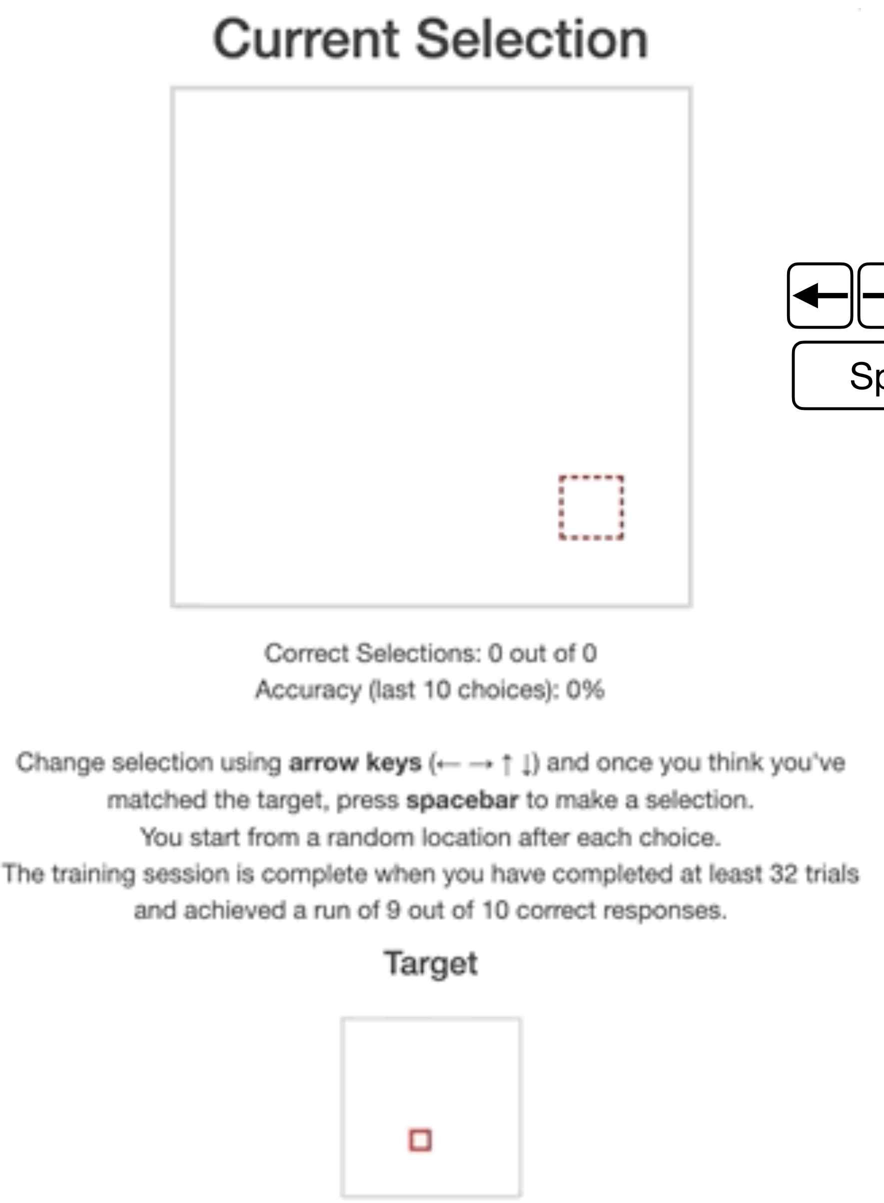
At least 32 trials AND a run of 9 out of 10 correct

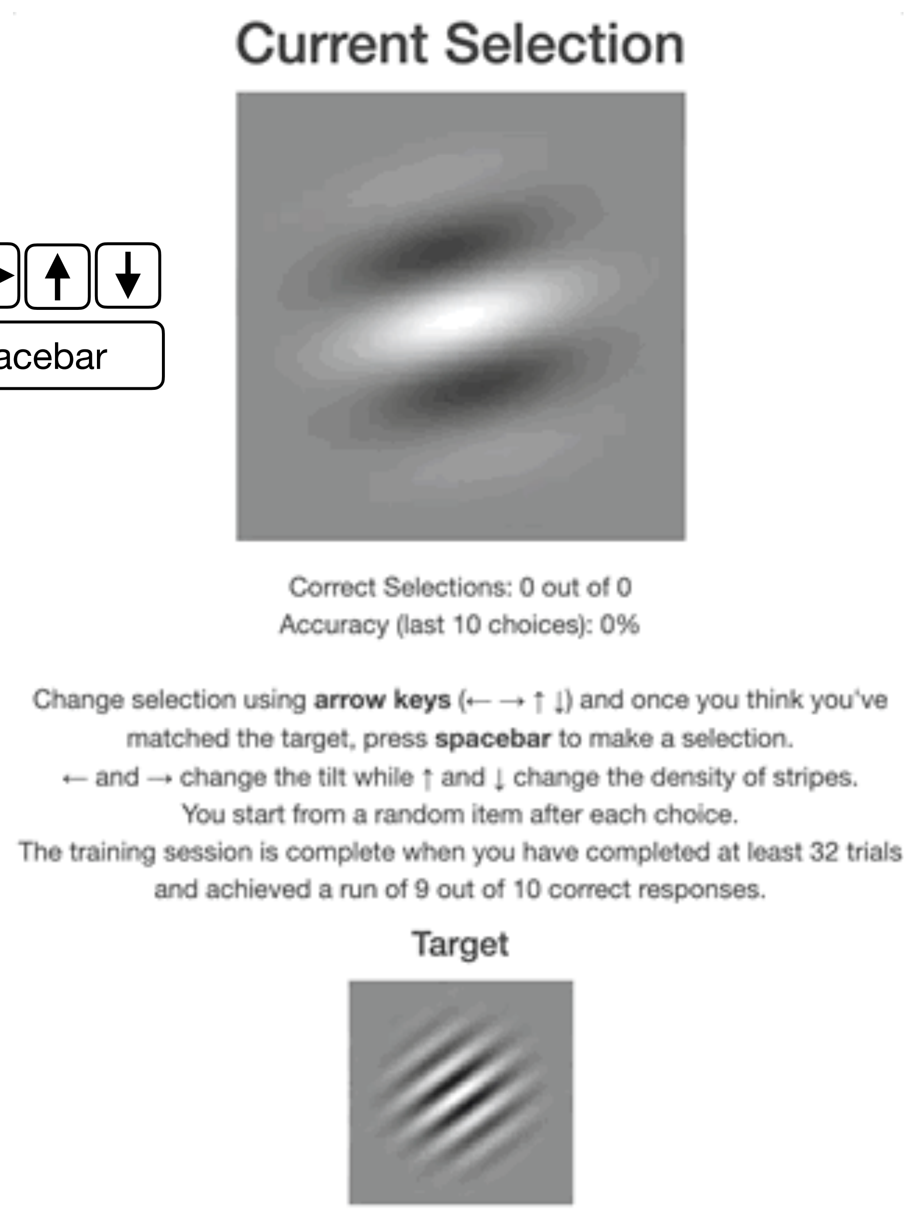


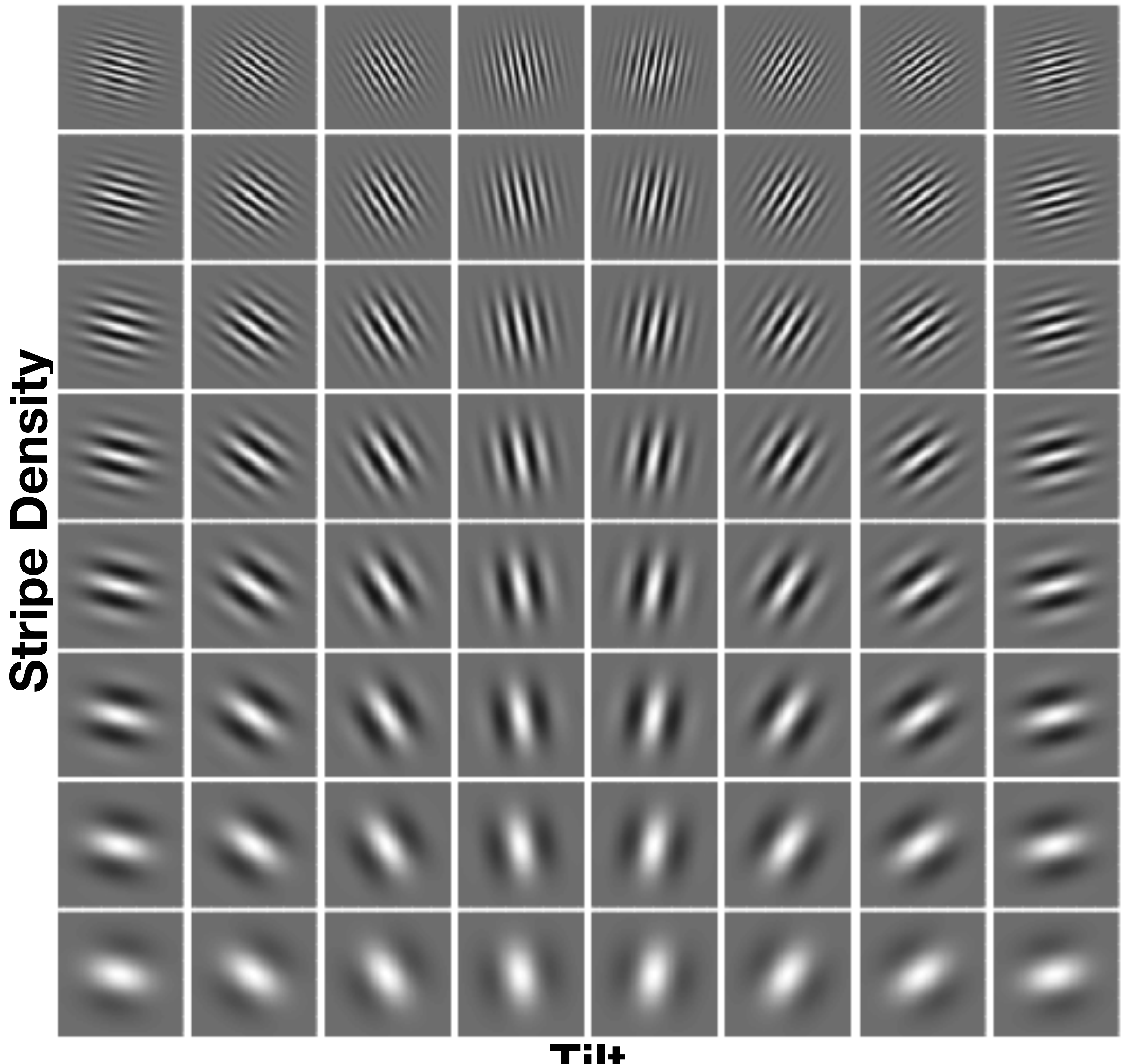
Task Design



Match target stimuli until learning criterion reached



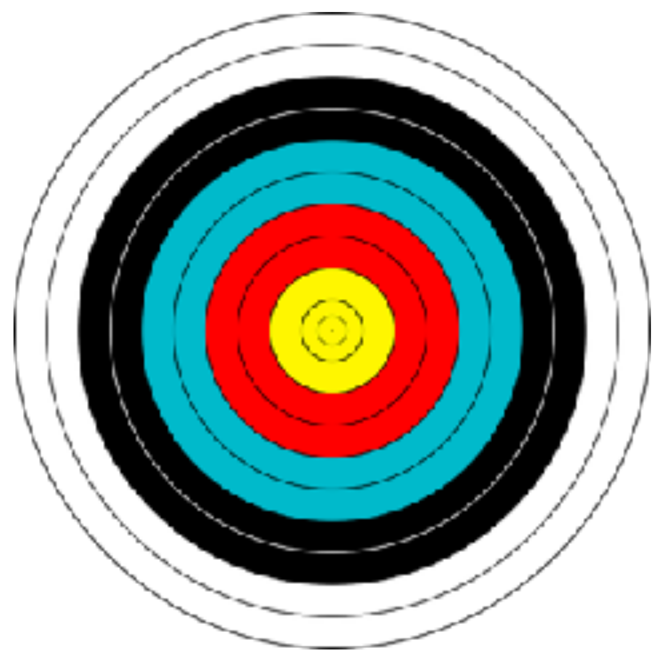




At least 32 trials AND a run of 9 out of 10 correct

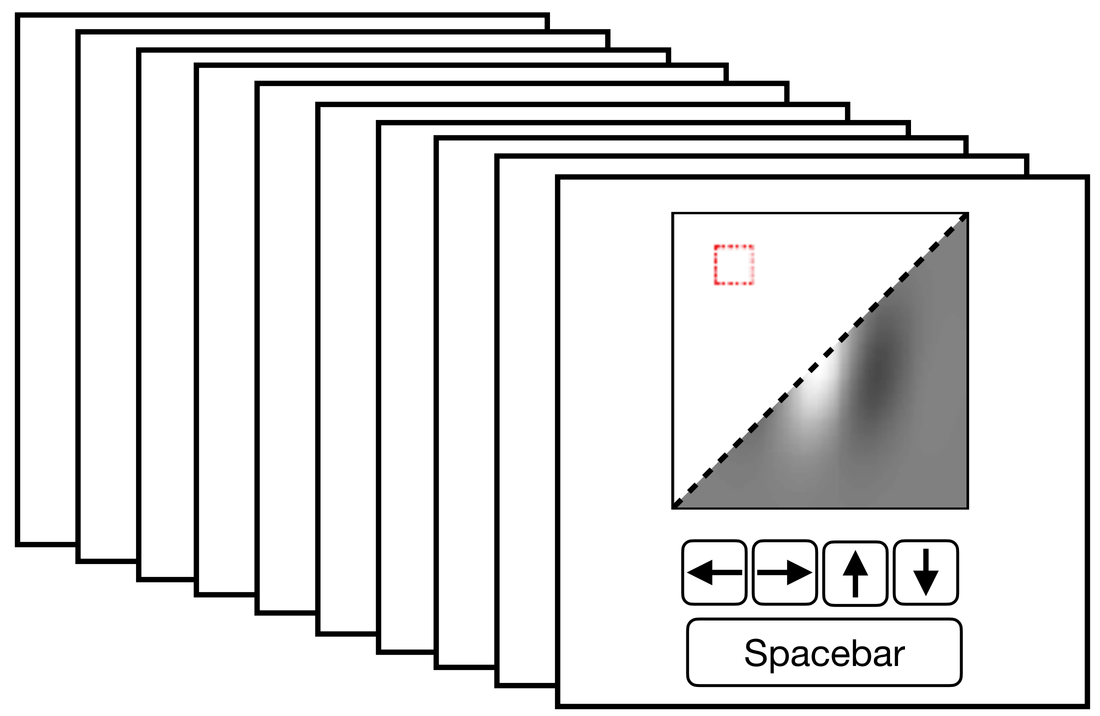
Task Design

Training Phase



Comprehension Questions

Main Search Task

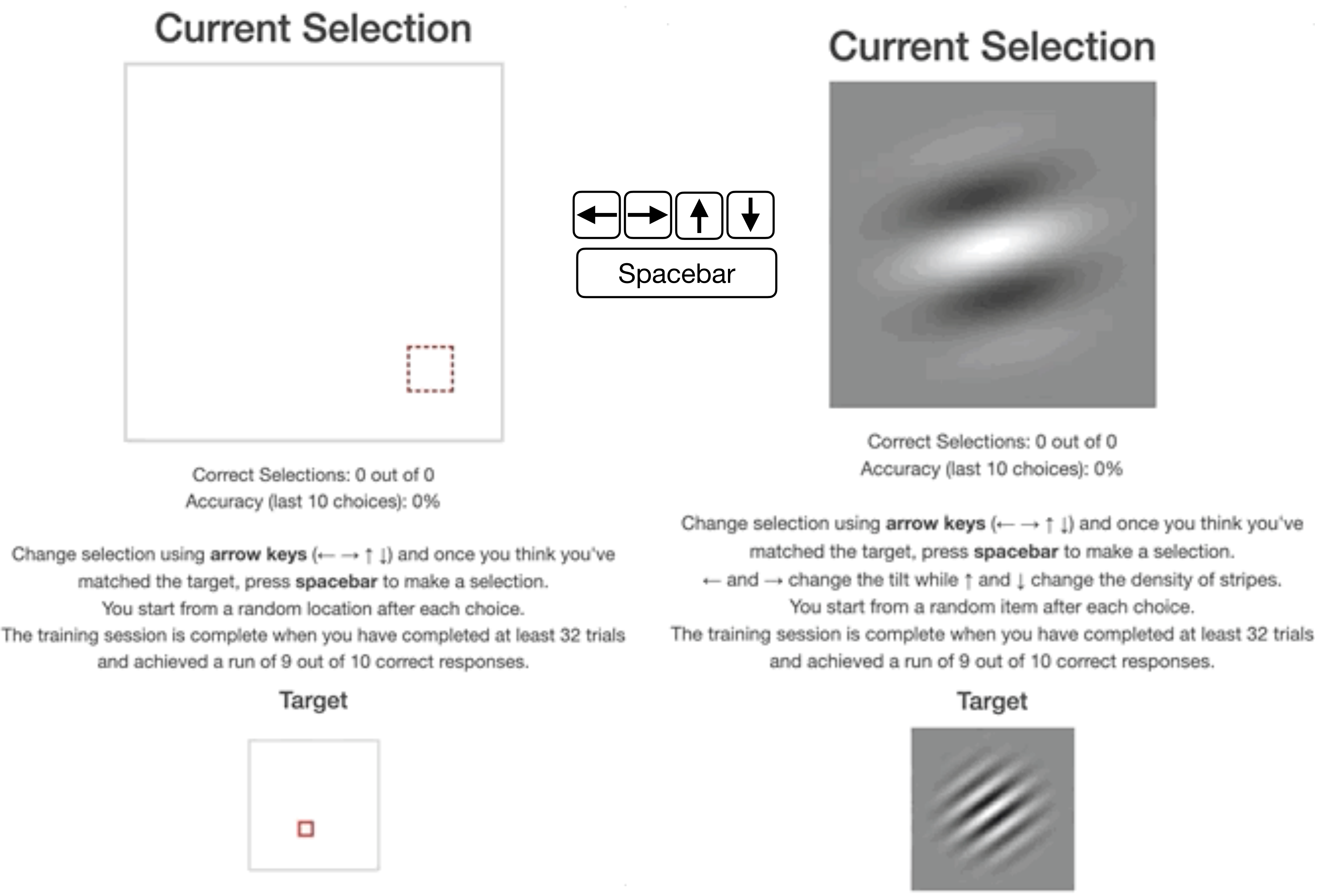


10 rounds
20 trials in each

Bonus Round

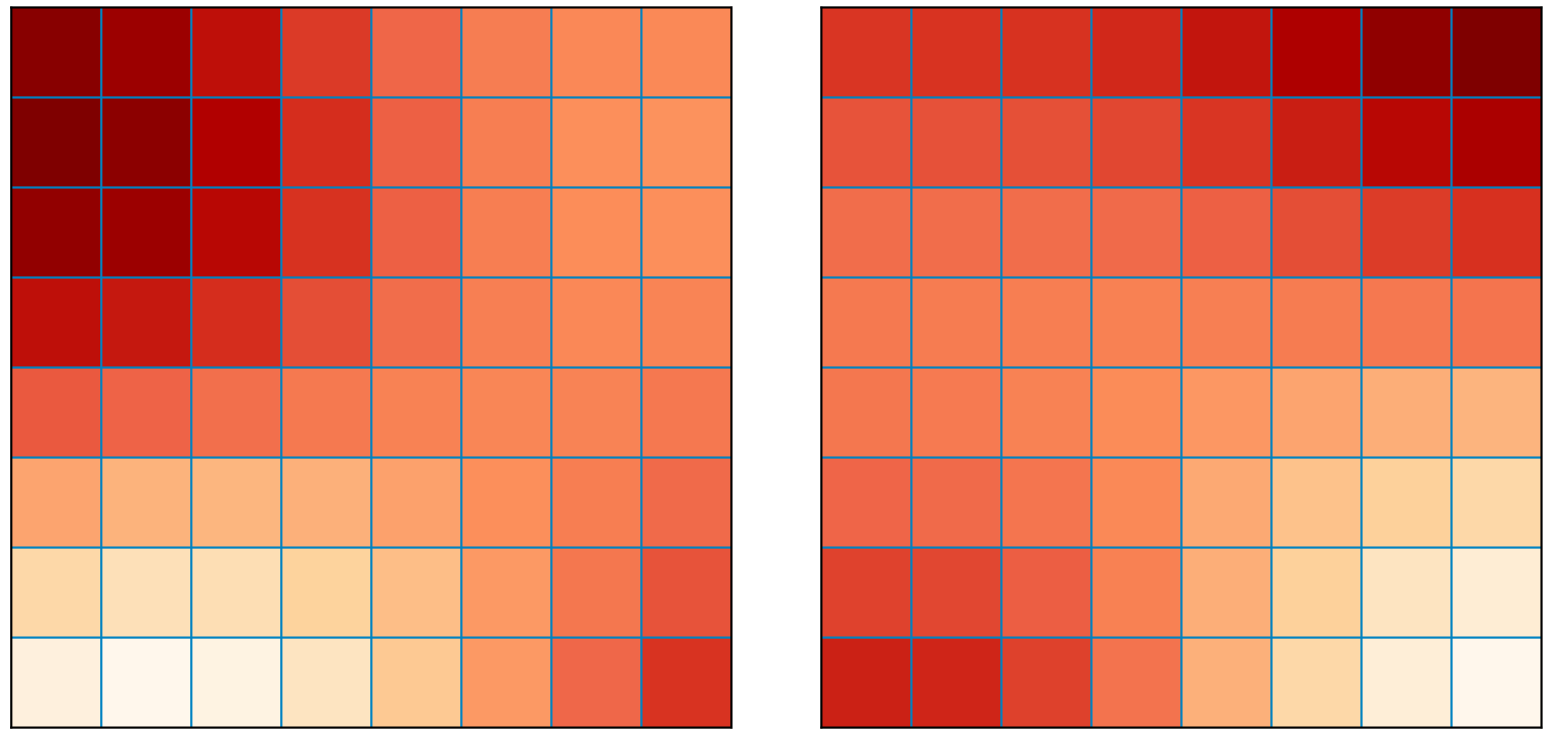
Judgment and Confidence

Match target stimuli until learning criterion reached

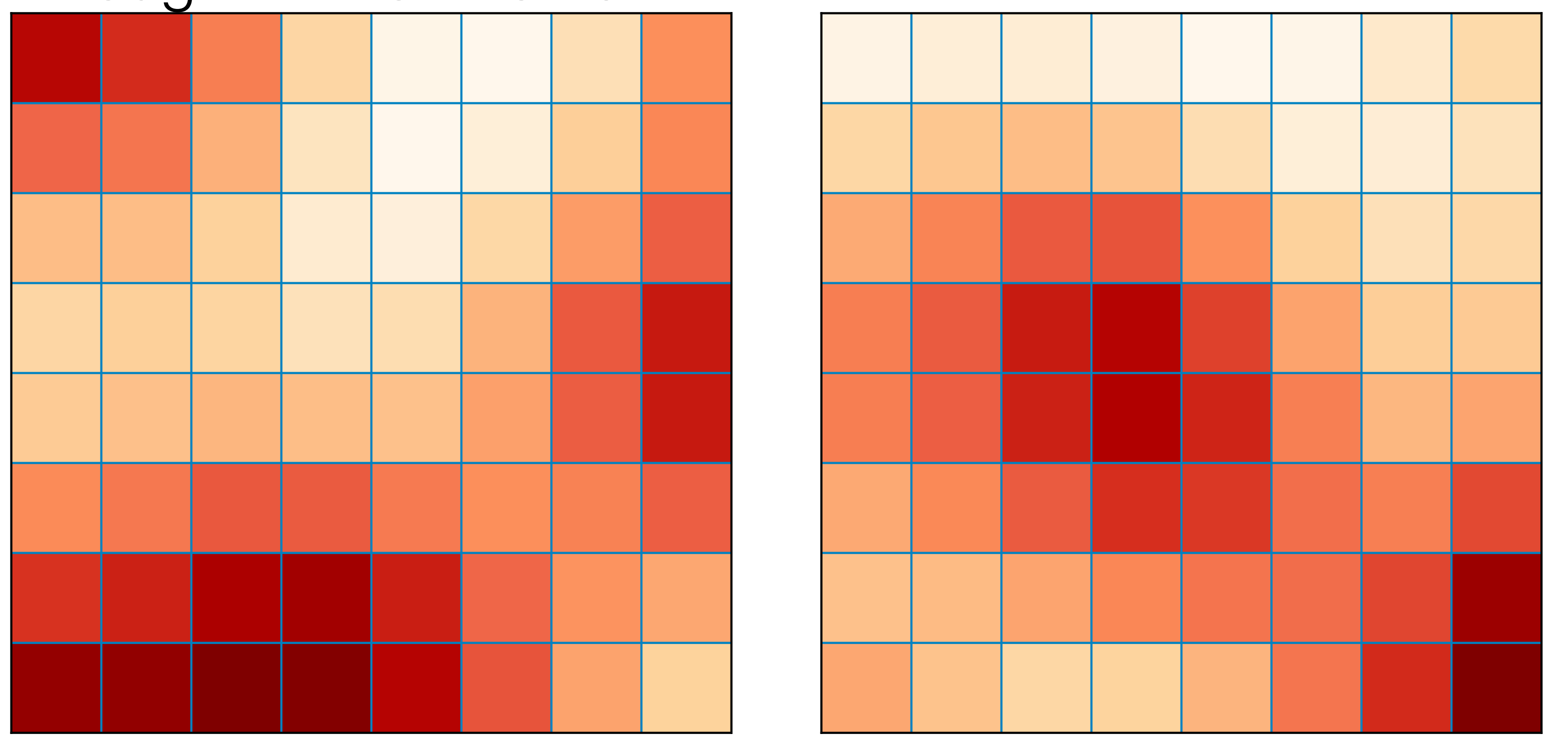


At least 32 trials AND a run of 9 out of 10 correct

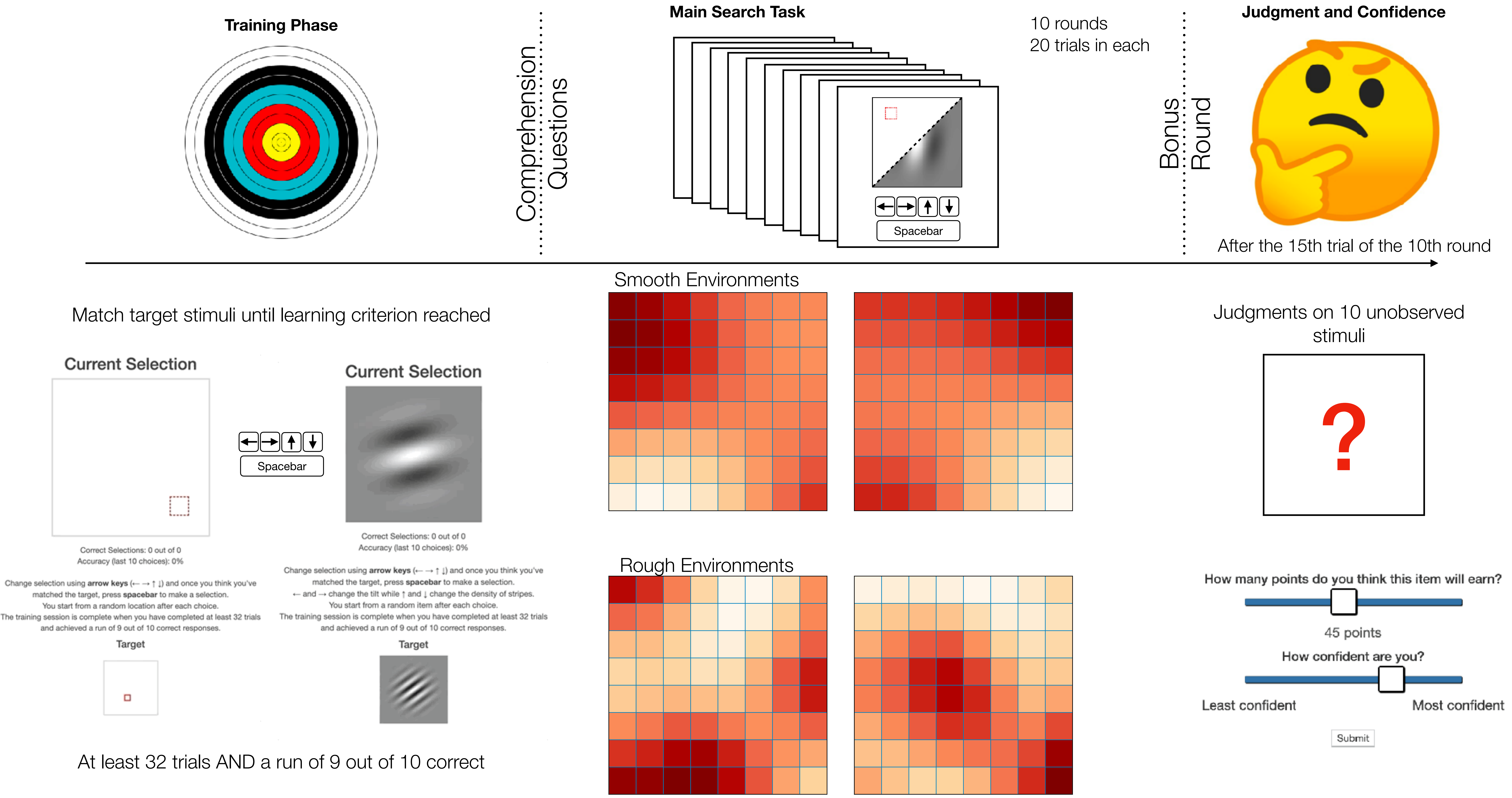
Smooth Environments



Rough Environments



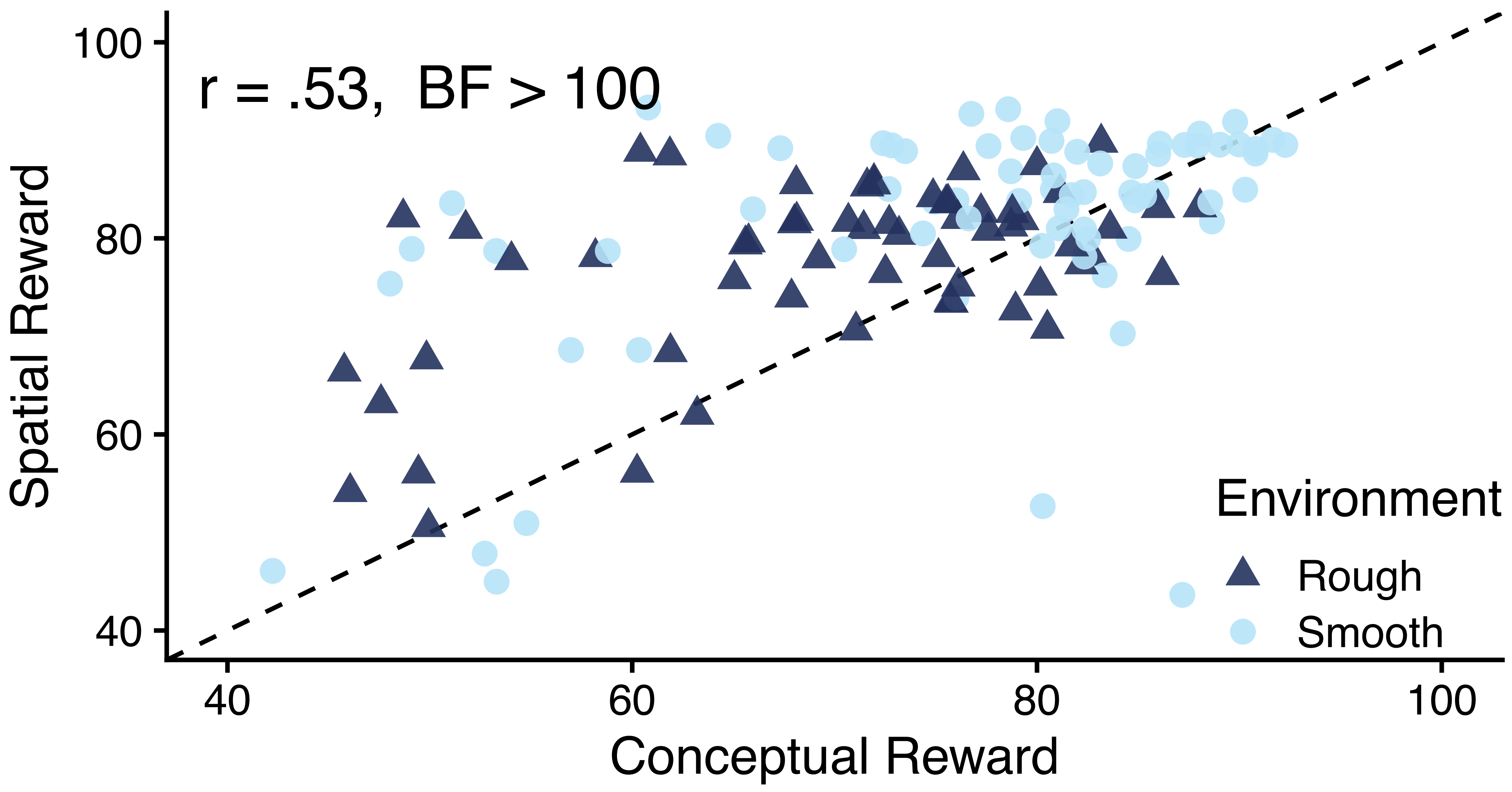
Task Design

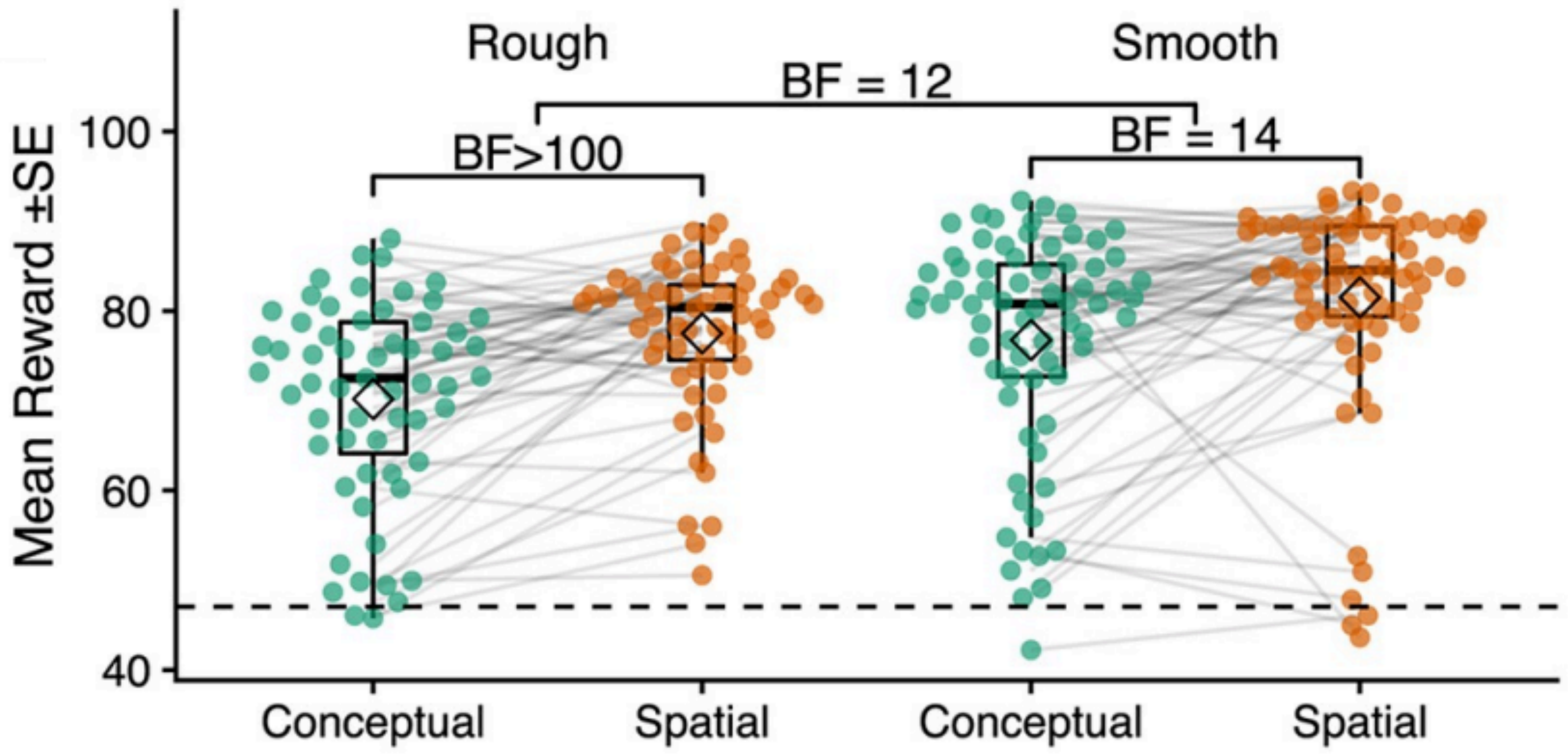


Training Phase
Comprehension Questions
Main Search Task
10 rounds
20 trials in each
Spacebar
Bonus Round
Judgment and Confidence
After the 15th trial of the 10th round
Match target stimuli until learning criterion reached
Current Selection
Correct Selections: 0 out of 0
Accuracy (last 10 choices): 0%
Change selection using arrow keys (← → ↑ ↓) and once you think you've matched the target, press spacebar to make a selection.
You start from a random location after each choice.
The training session is complete when you have completed at least 32 trials and achieved a run of 9 out of 10 correct responses.
Target
Spacebar
Current Selection
Correct Selections: 0 out of 0
Accuracy (last 10 choices): 0%
Change selection using arrow keys (← → ↑ ↓) and once you think you've matched the target, press spacebar to make a selection.
← and → change the tilt while ↑ and ↓ change the density of stripes.
You start from a random item after each choice.
The training session is complete when you have completed at least 32 trials and achieved a run of 9 out of 10 correct responses.
Target
At least 32 trials AND a run of 9 out of 10 correct
Smooth Environments
Rough Environments
Judgments on 10 unobserved stimuli
?
How many points do you think this item will earn?
45 points
How confident are you?
Least confident
Most confident
Submit

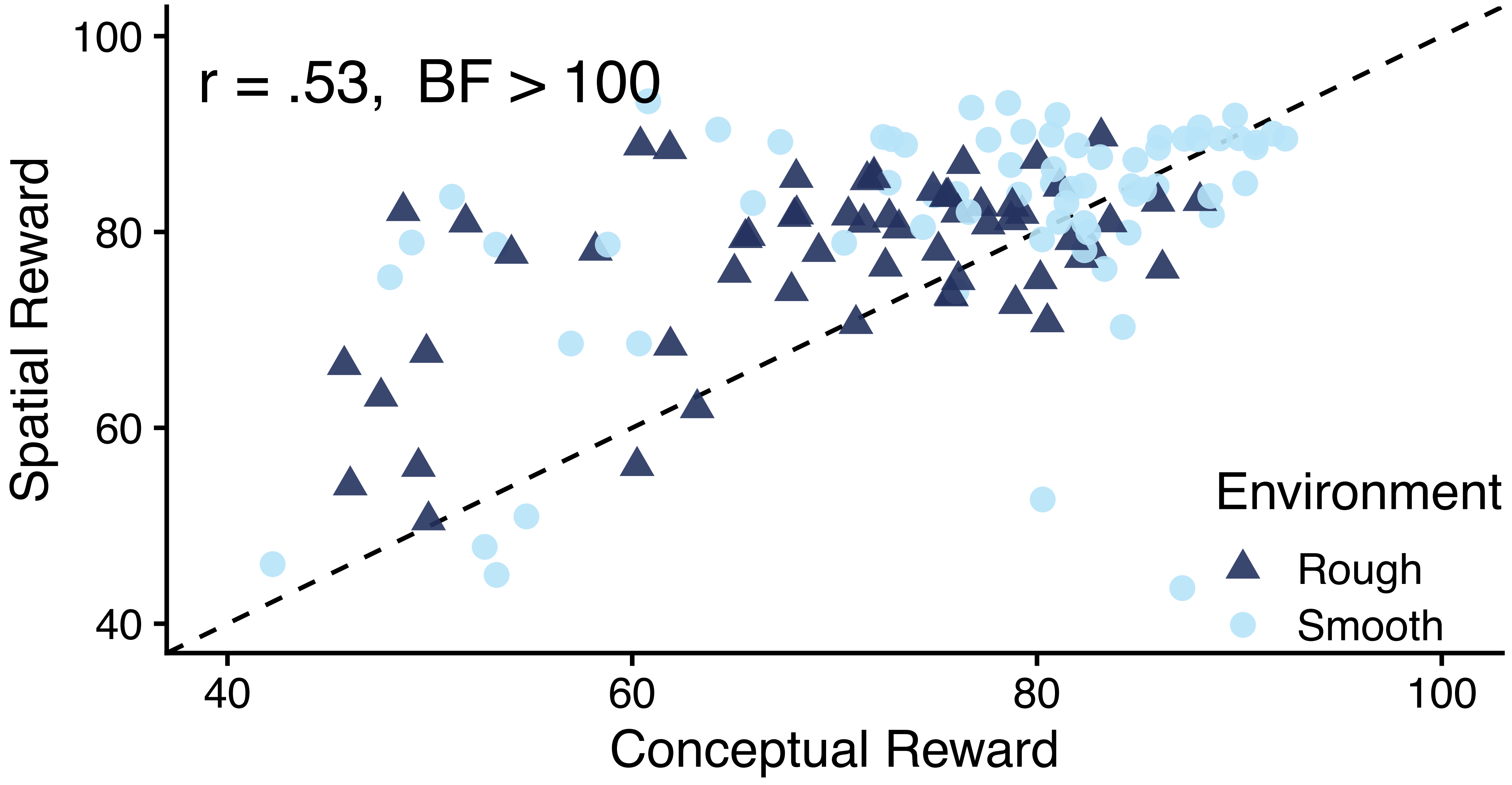
Behavioral Results

- Correlated performance, but generally better in the spatial task
- This difference can largely be explained by a one-directional transfer effect:
    - Experience with spatial search boosted performance on conceptual search, but not vice versa





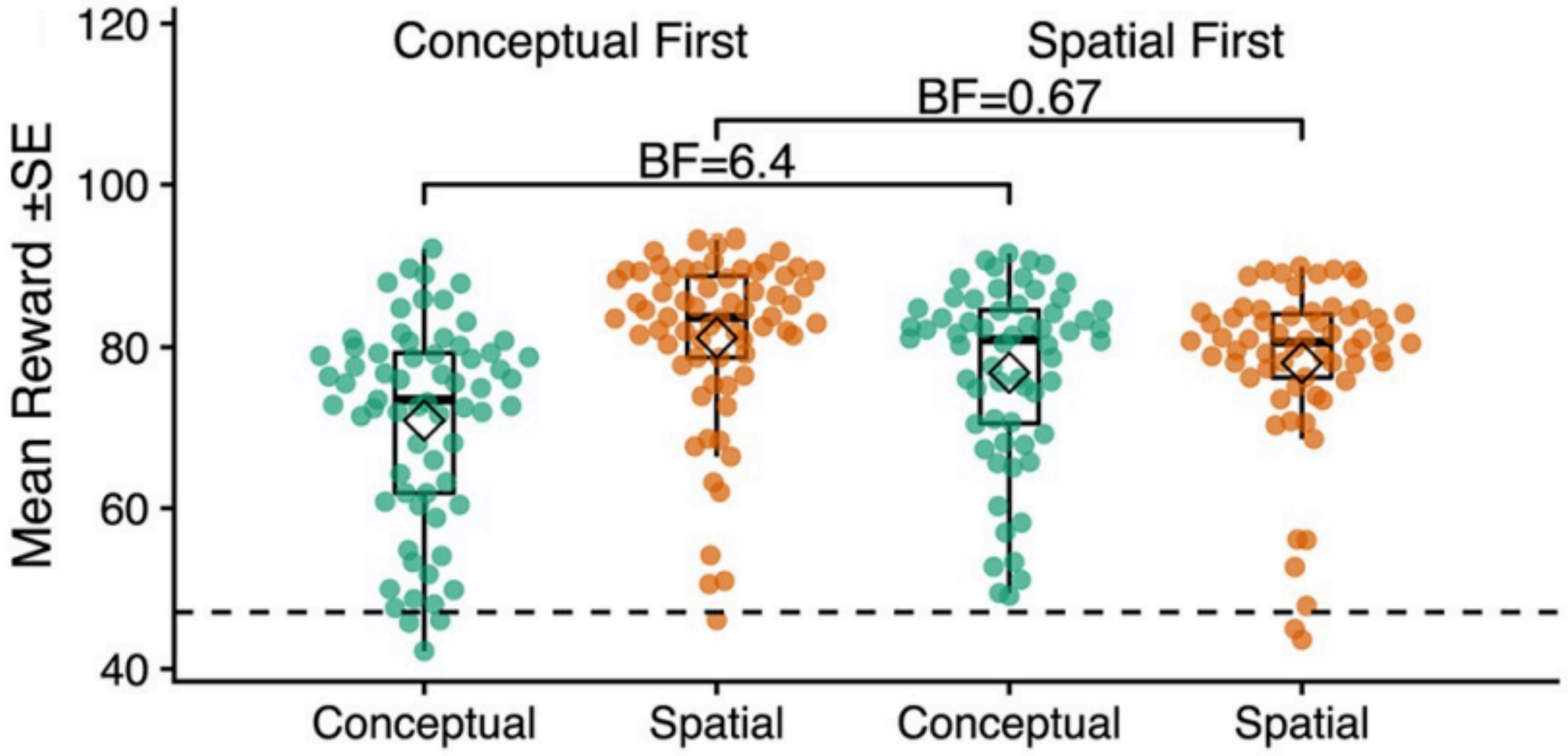
Behavioral Results



- Correlated performance, but generally better in the spatial task

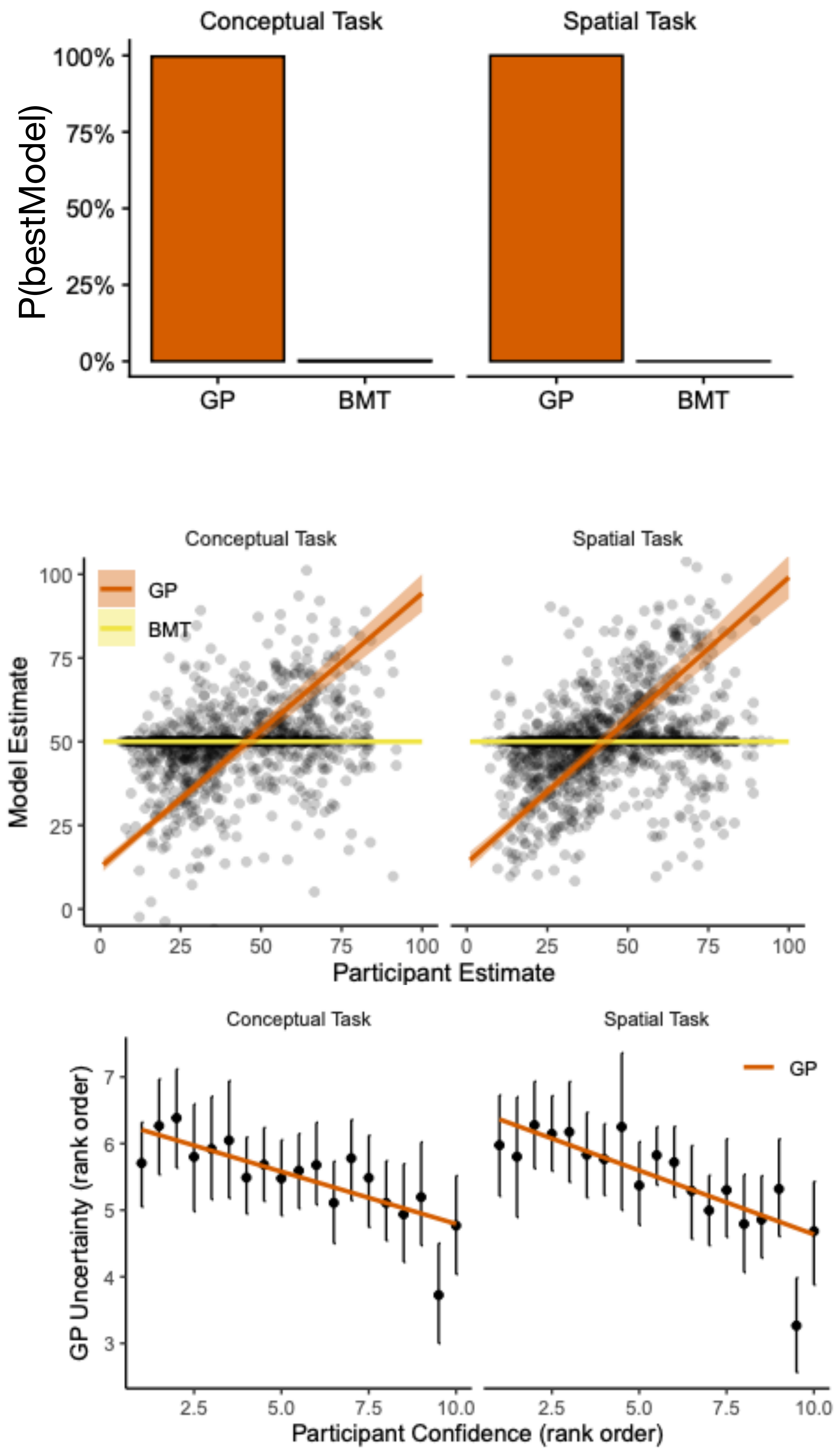
- This difference can largely be explained by a one-directional transfer effect:

  - Experience with spatial search boosted performance on conceptual search, but not vice versa



Modeling Results

- Using group-level Bayesian estimation, we find that GP-UCB is the best model of choice behavior
  - P(bestModel) estimates the most prevelant model (corrected for chance); also known as protected exceedance probability
- GP-UCB also predicts bonus round judgments about expected reward and confidence
  - using parameters estimated from rounds 1-9, we can use model simulations to predict participant judgments for unobserved stimuli in round 10
  - BMT makes invariant predictions for novel options, but the GP predictions correspond to participant judgments, where uncertainty is the opposite of confidence

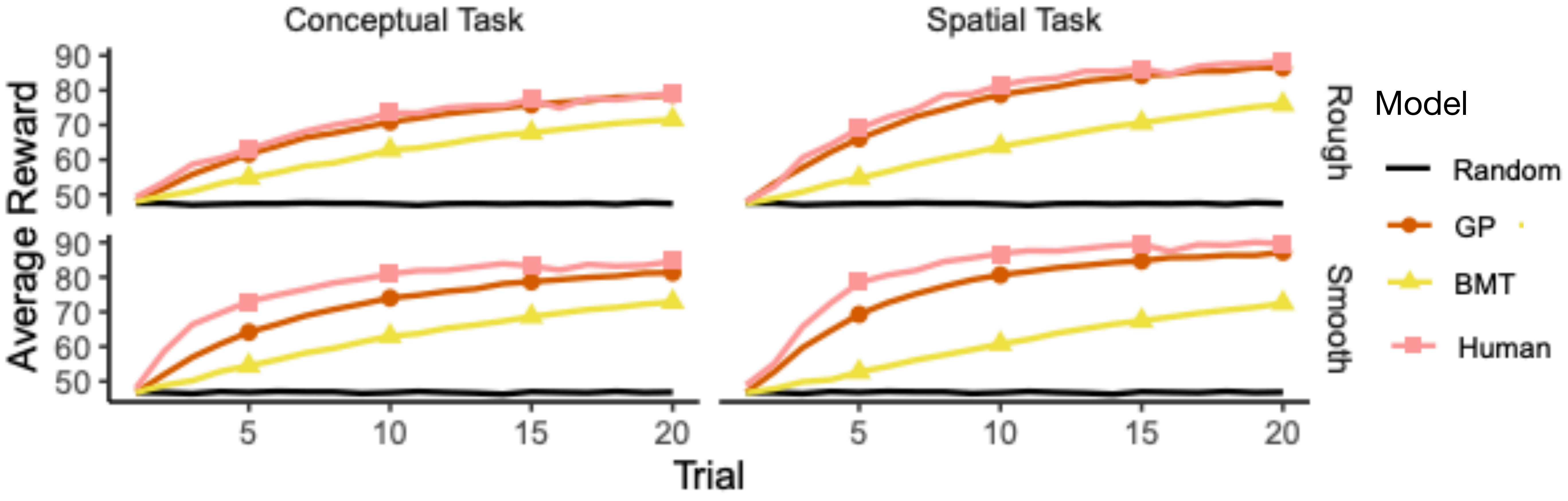


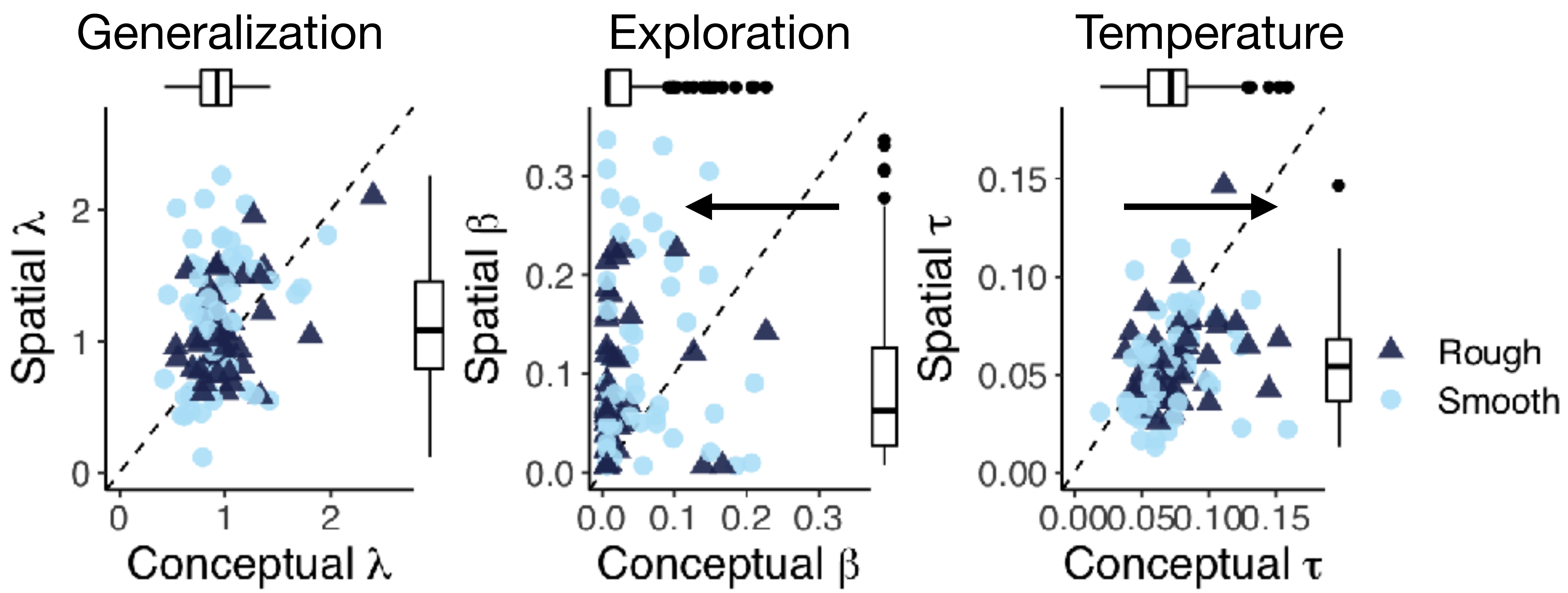
Modeling Results

- GP-UCB simulated learning curves resemble human performance

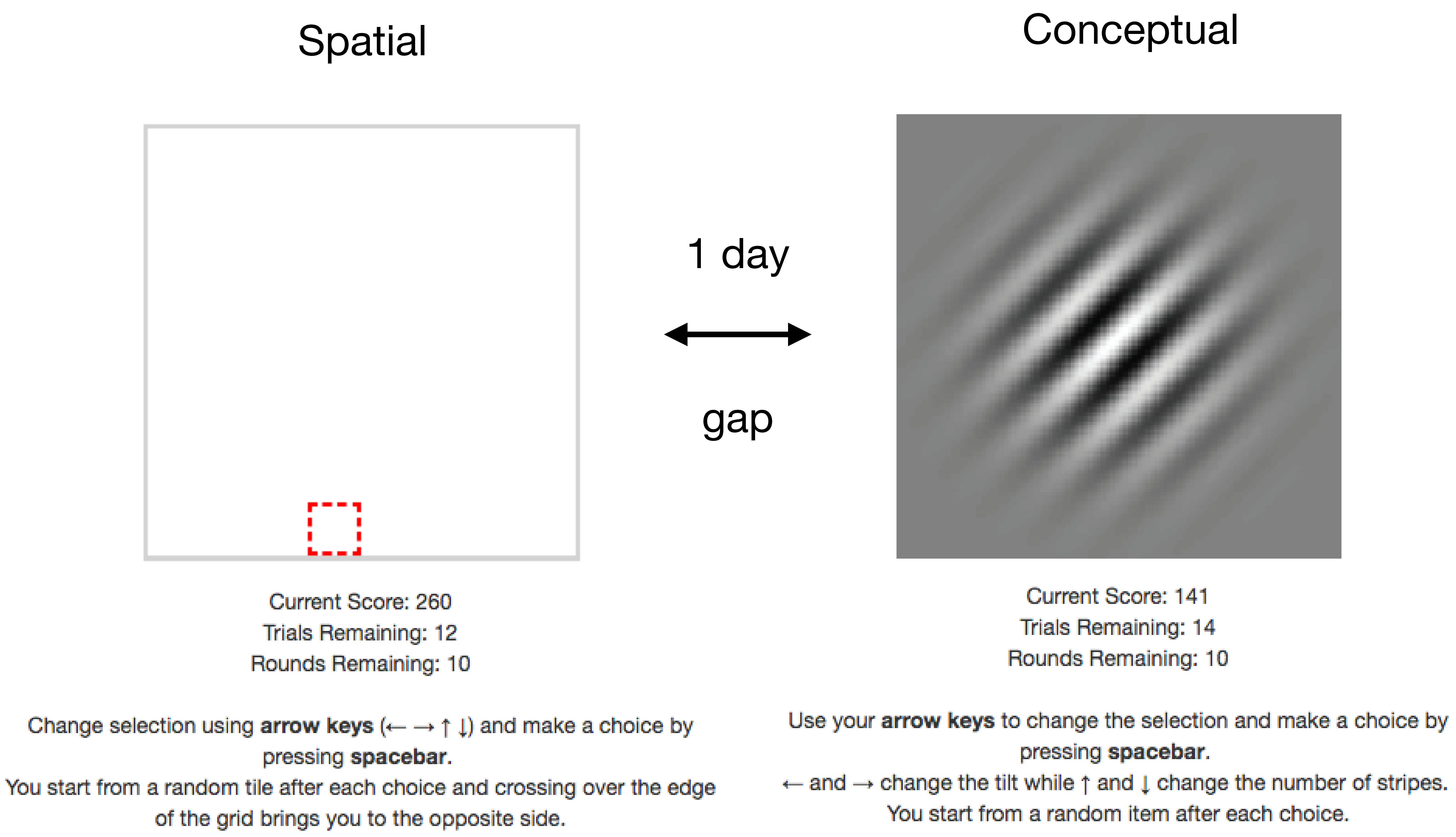
- Parameter estimates show similar levels of generalization but change in exploration

- Directed exploration vanishes in the conceptual task, replaced by higher temperature (i.e., random) sampling



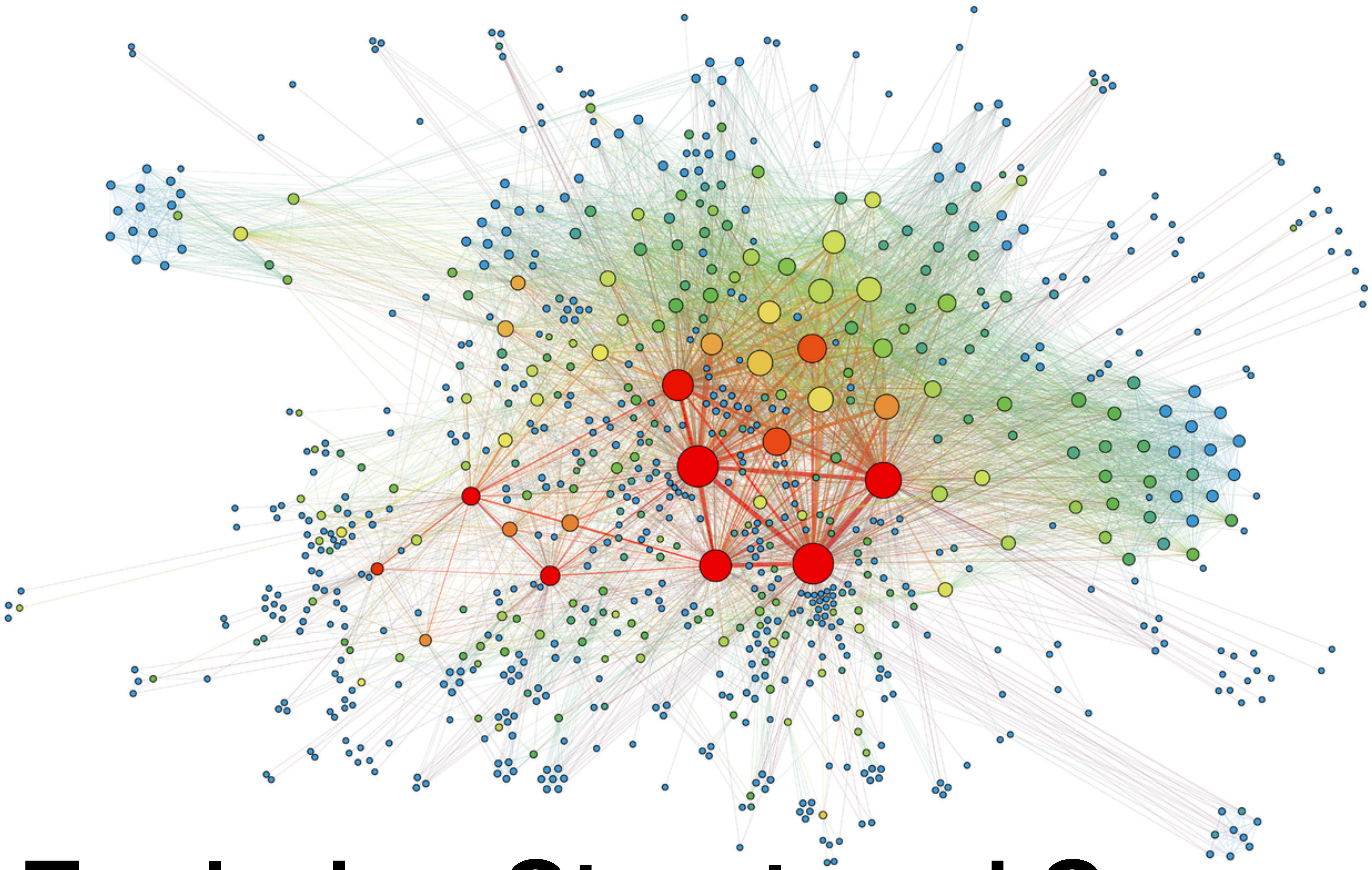


Summary



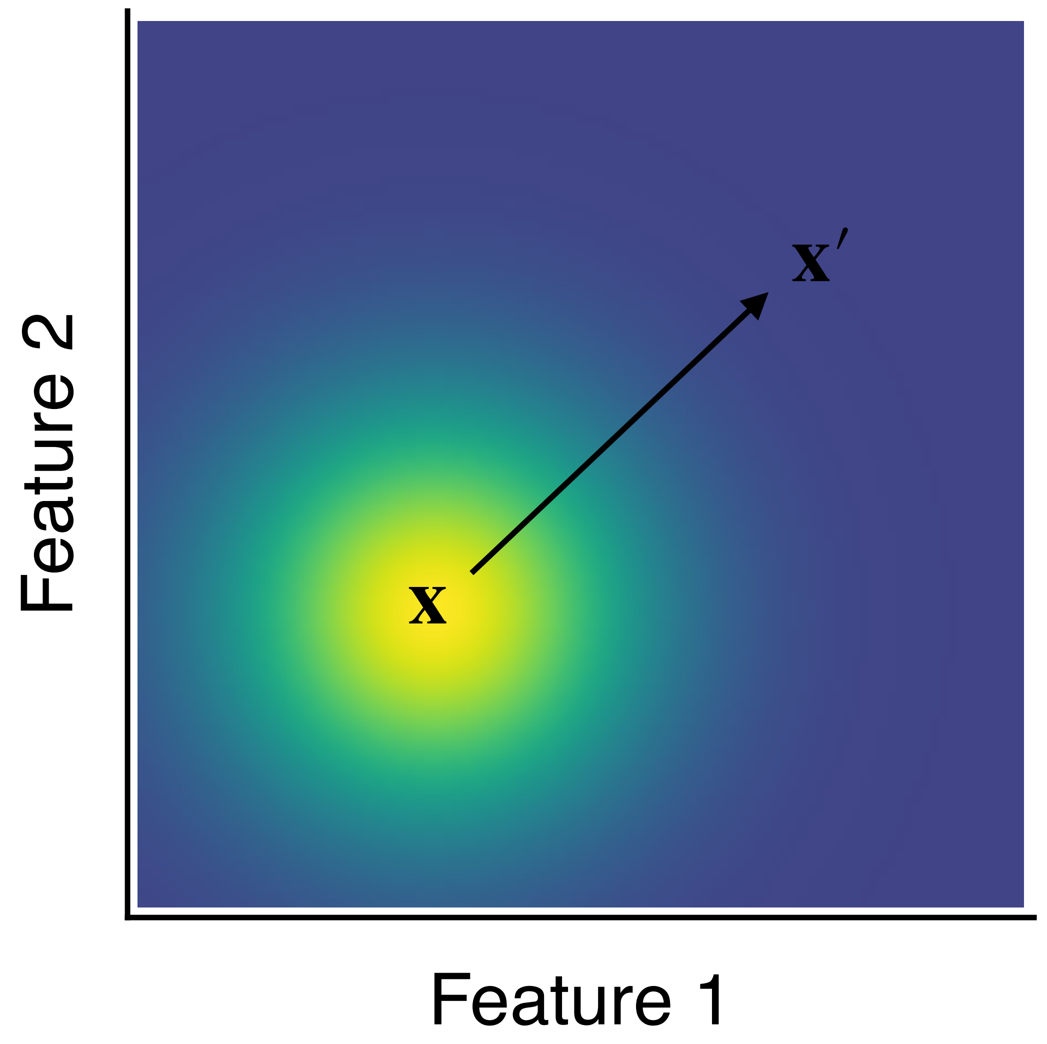
- Similar mechanisms of generalization-guided search in both domains
- But also diagnostic differences:
    - One-directional transfer effect suggest something fundamental about spatial reasoning
    - Switch from directed to random exploration
- Bonus round suggests participants have a good sense of uncertainty (confidence ratings) in both domains, but aren't able to leverage it to direct their exploration in the conceptual task

Wu, Schulz, Meder & Schuck (*PLOS Comp Bio* 2020)

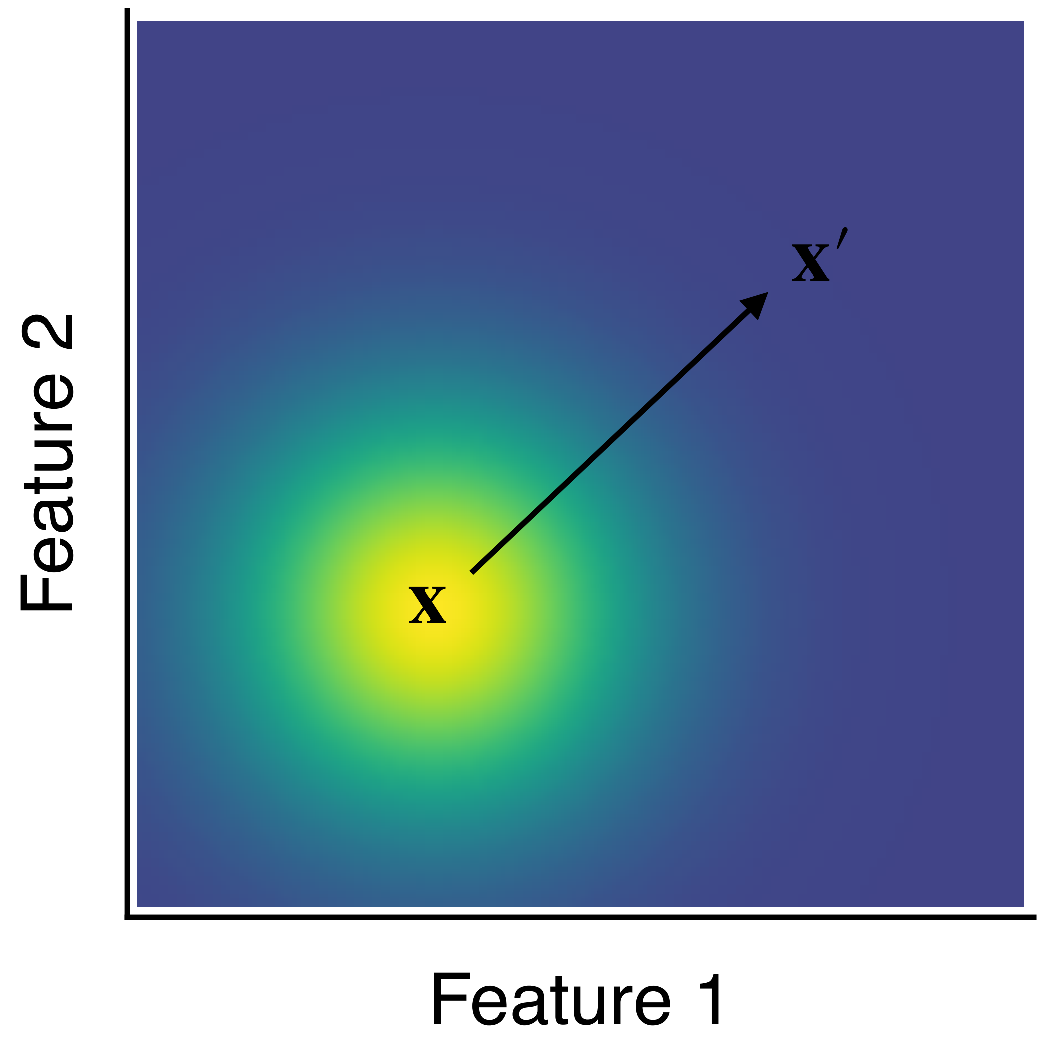


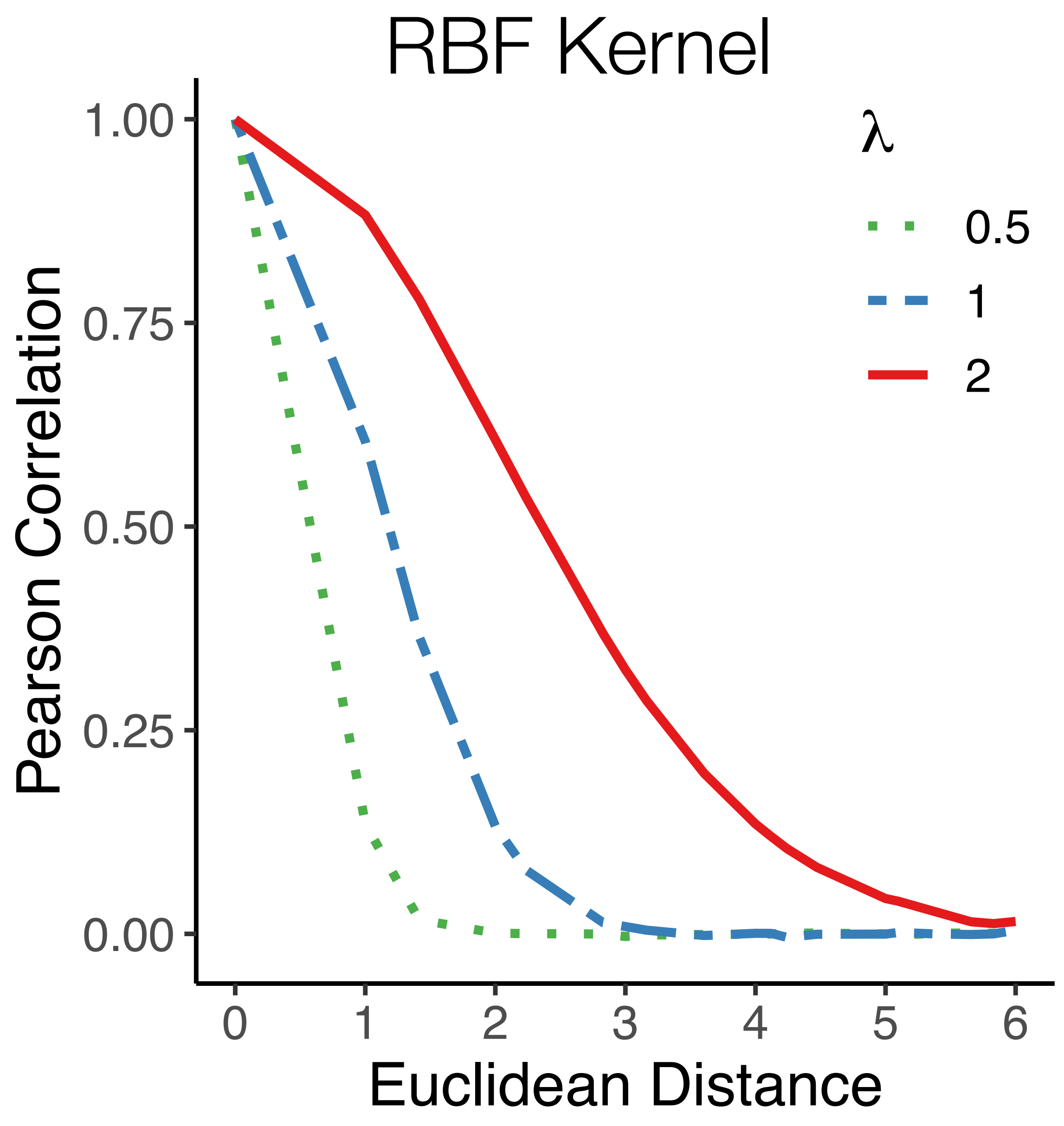
Exploring Structured Spaces

From continuous to structured spaces

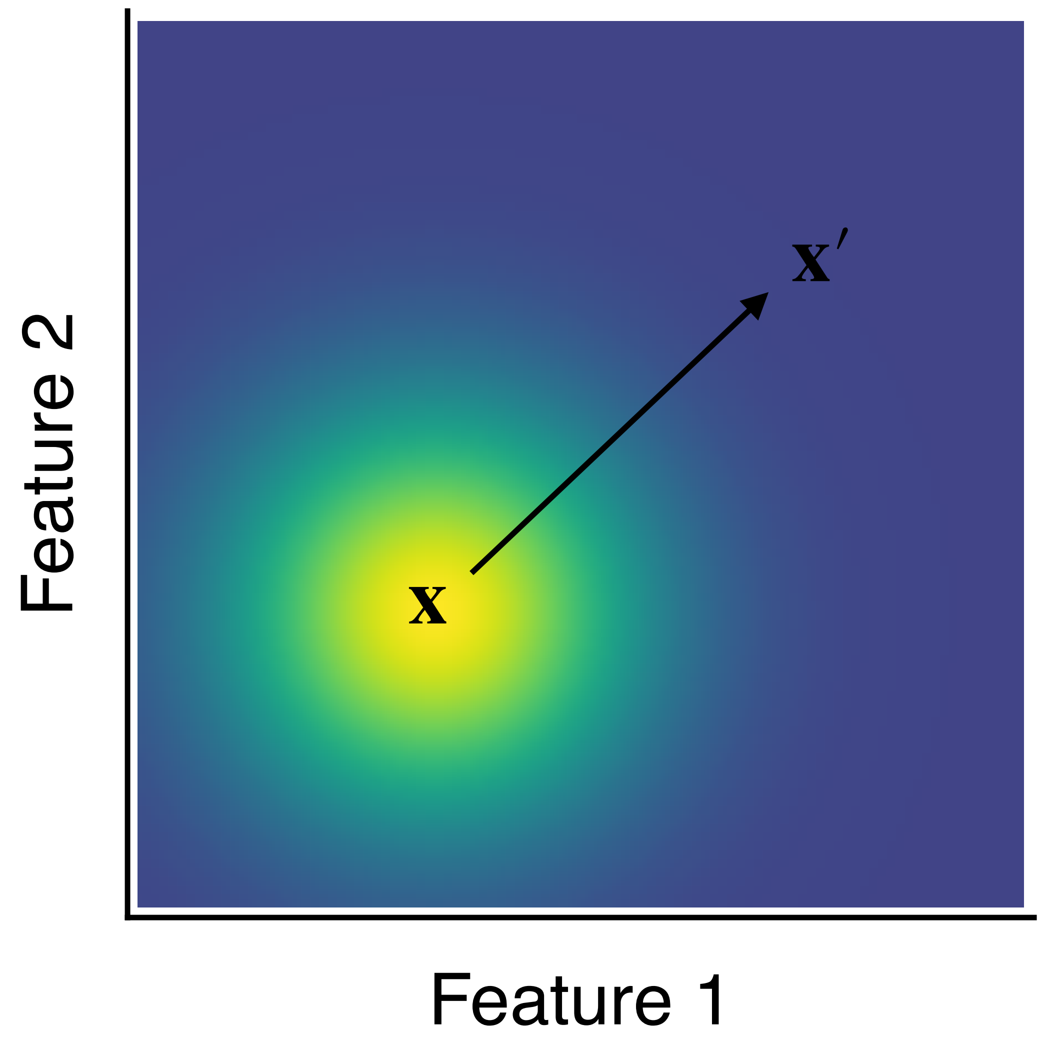


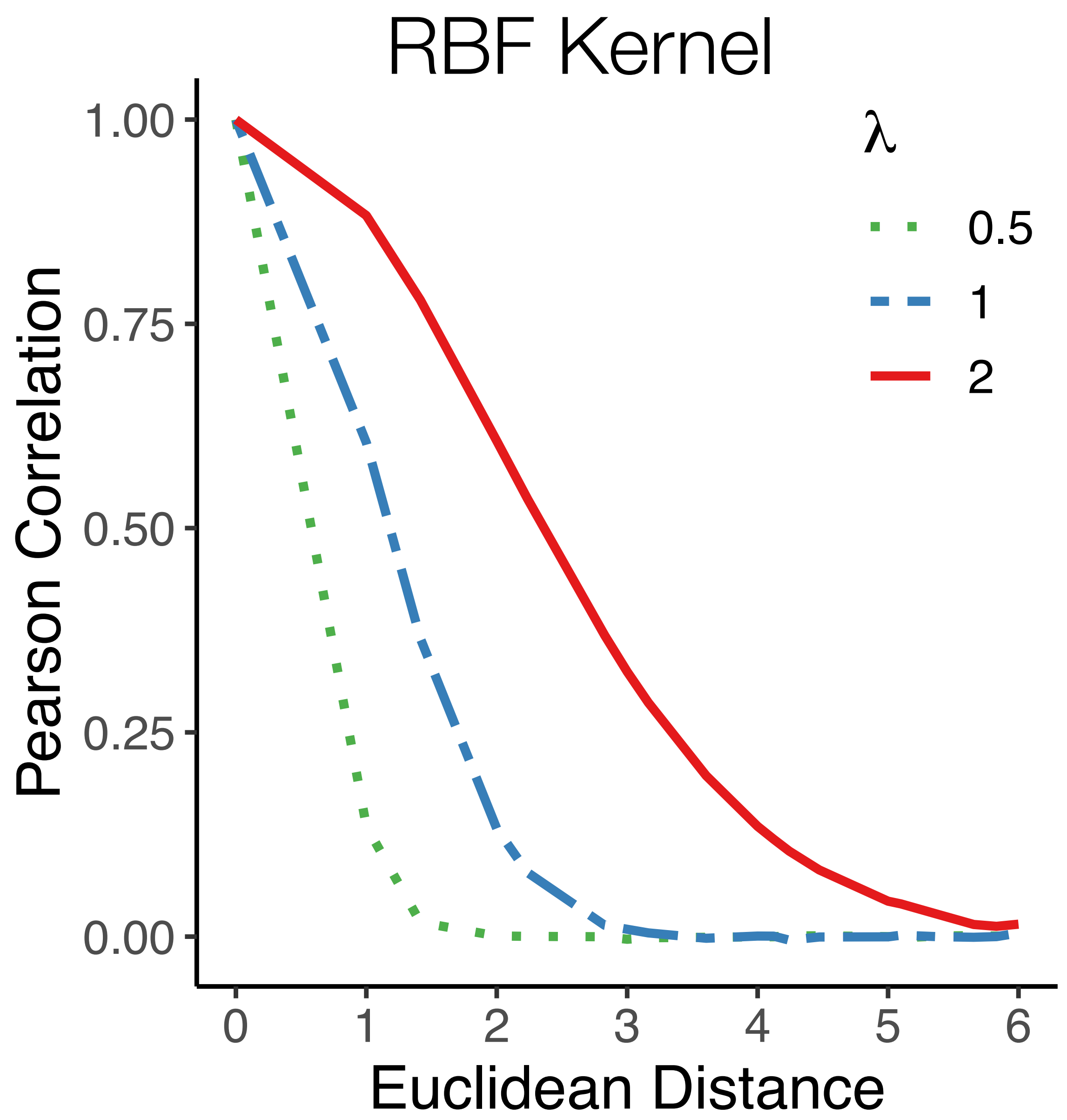
From continuous to structured spaces



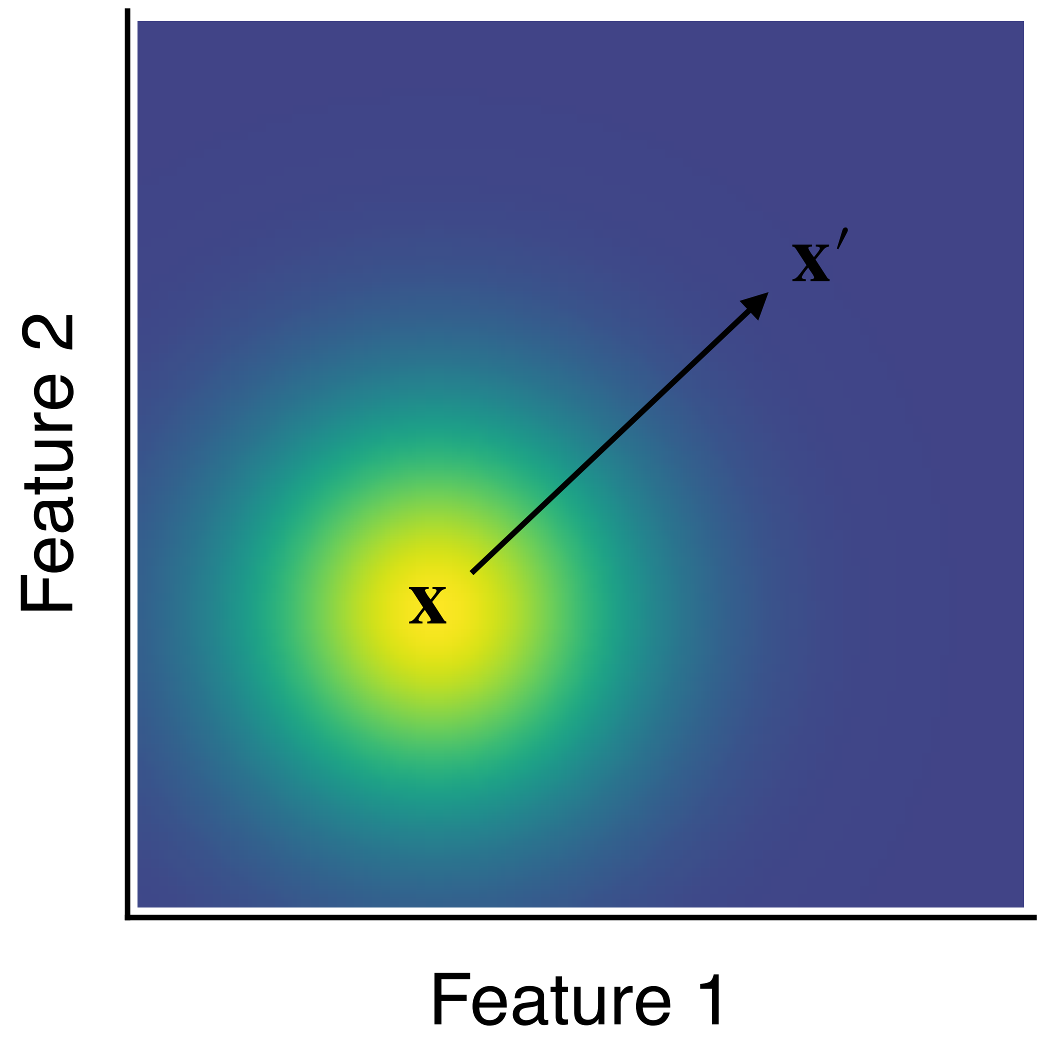


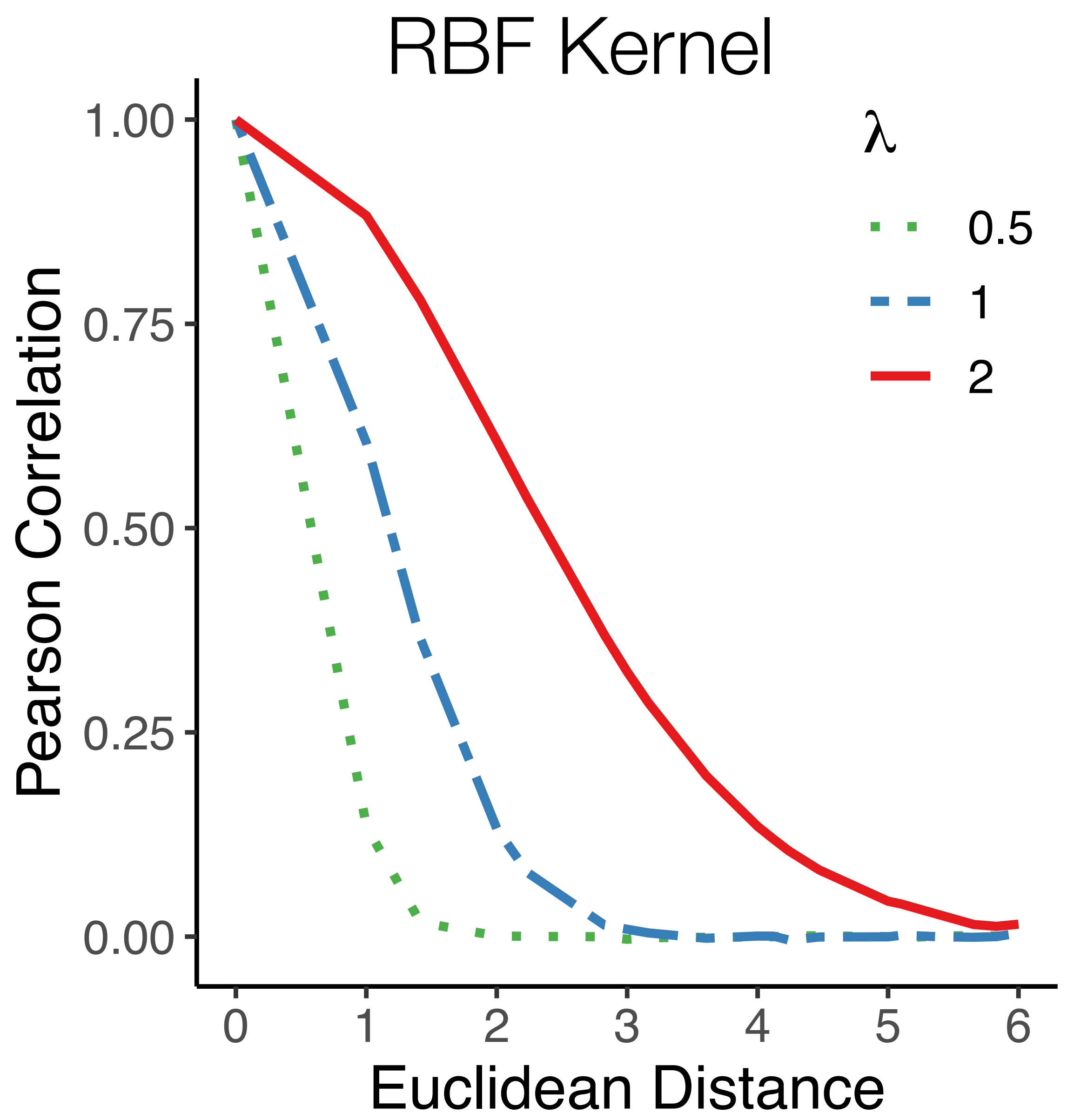
From continuous to structured spaces

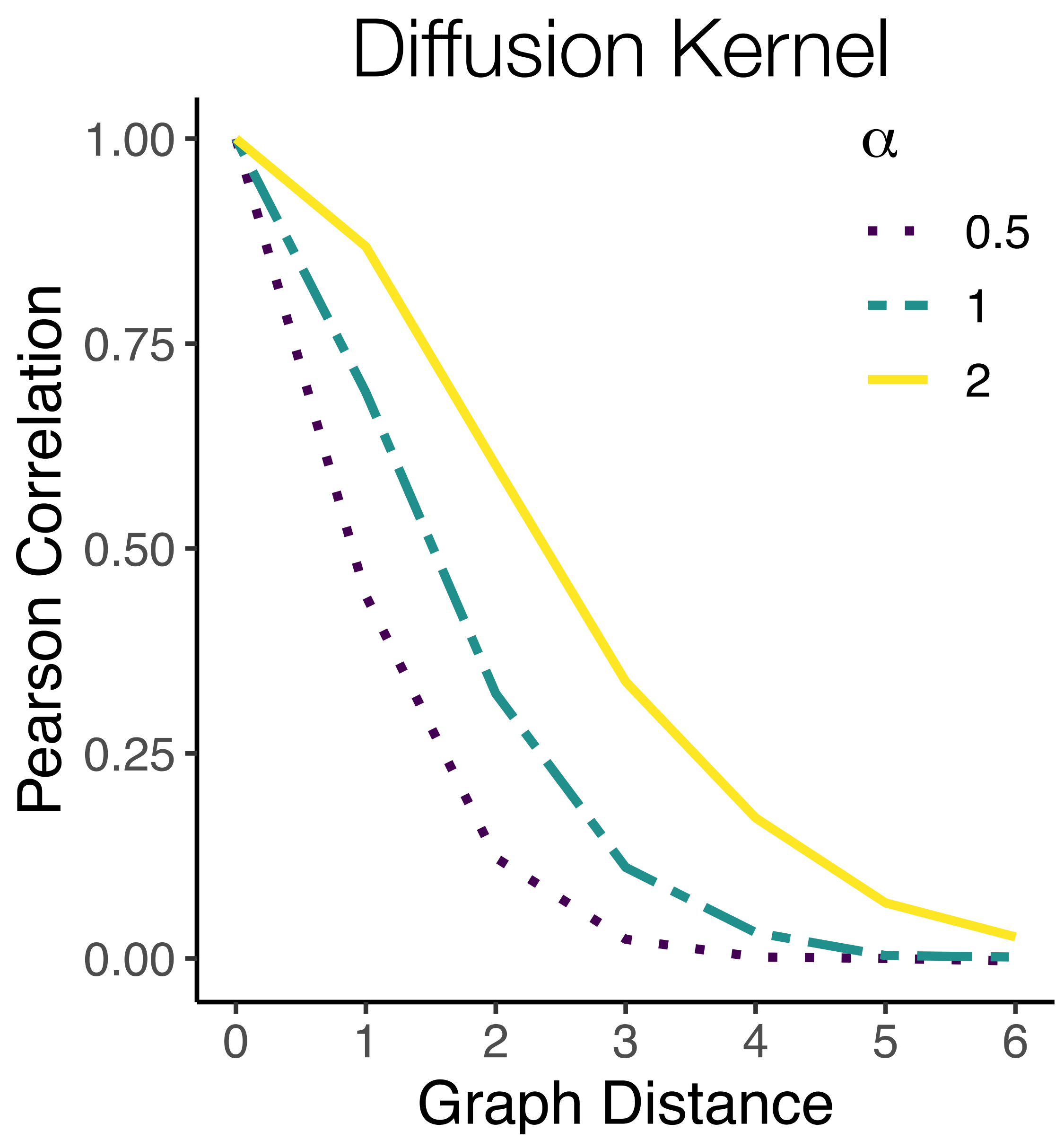




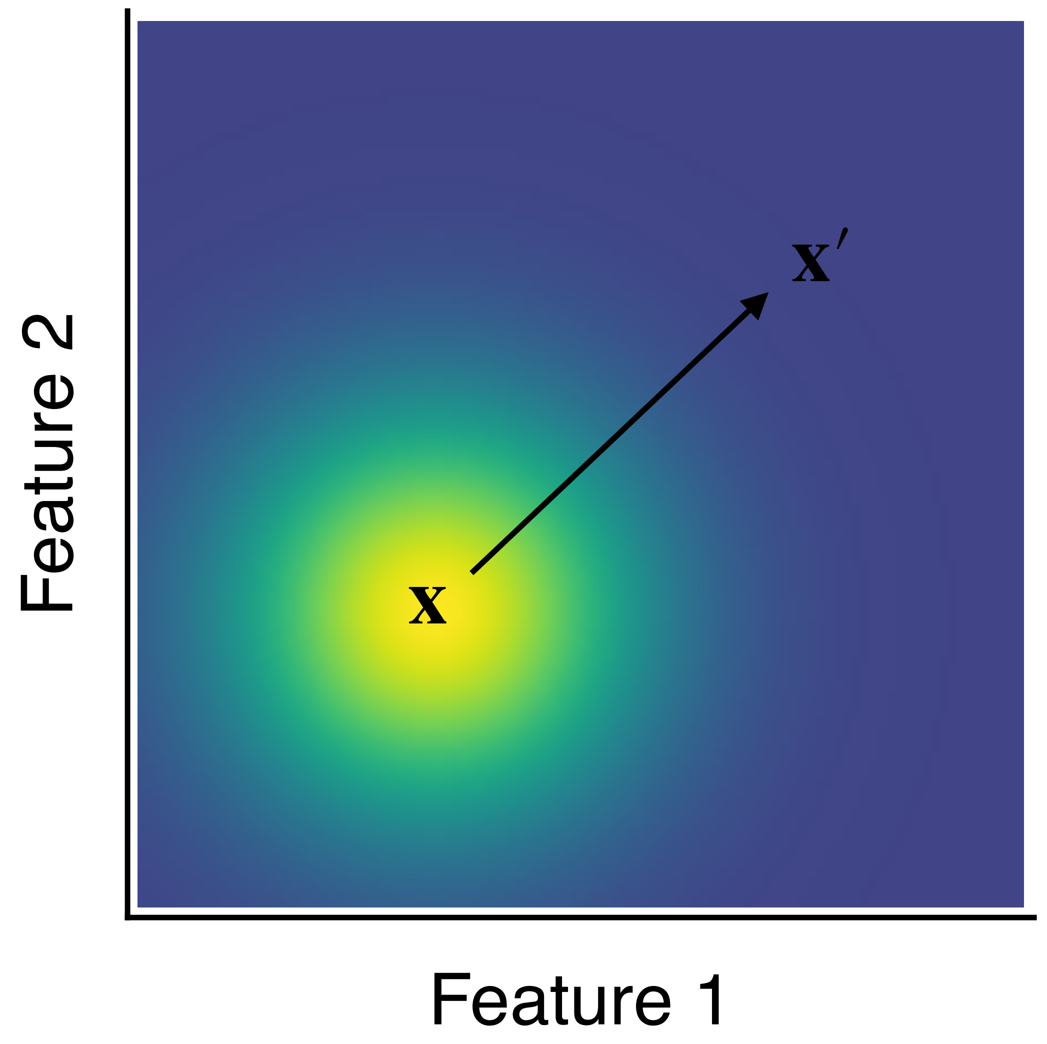
From continuous to structured spaces

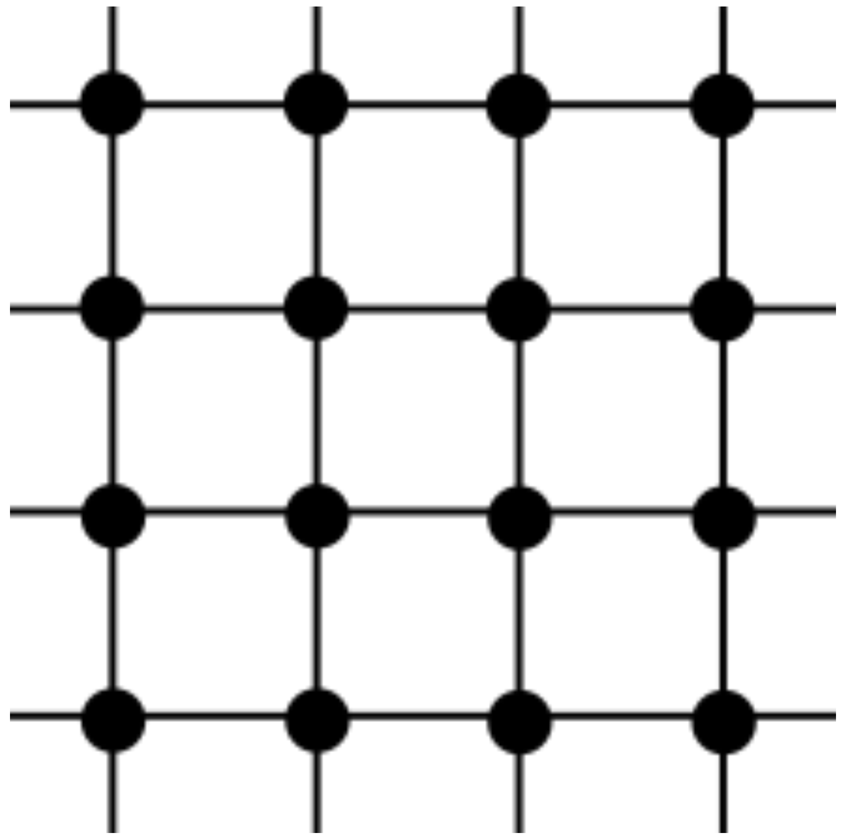


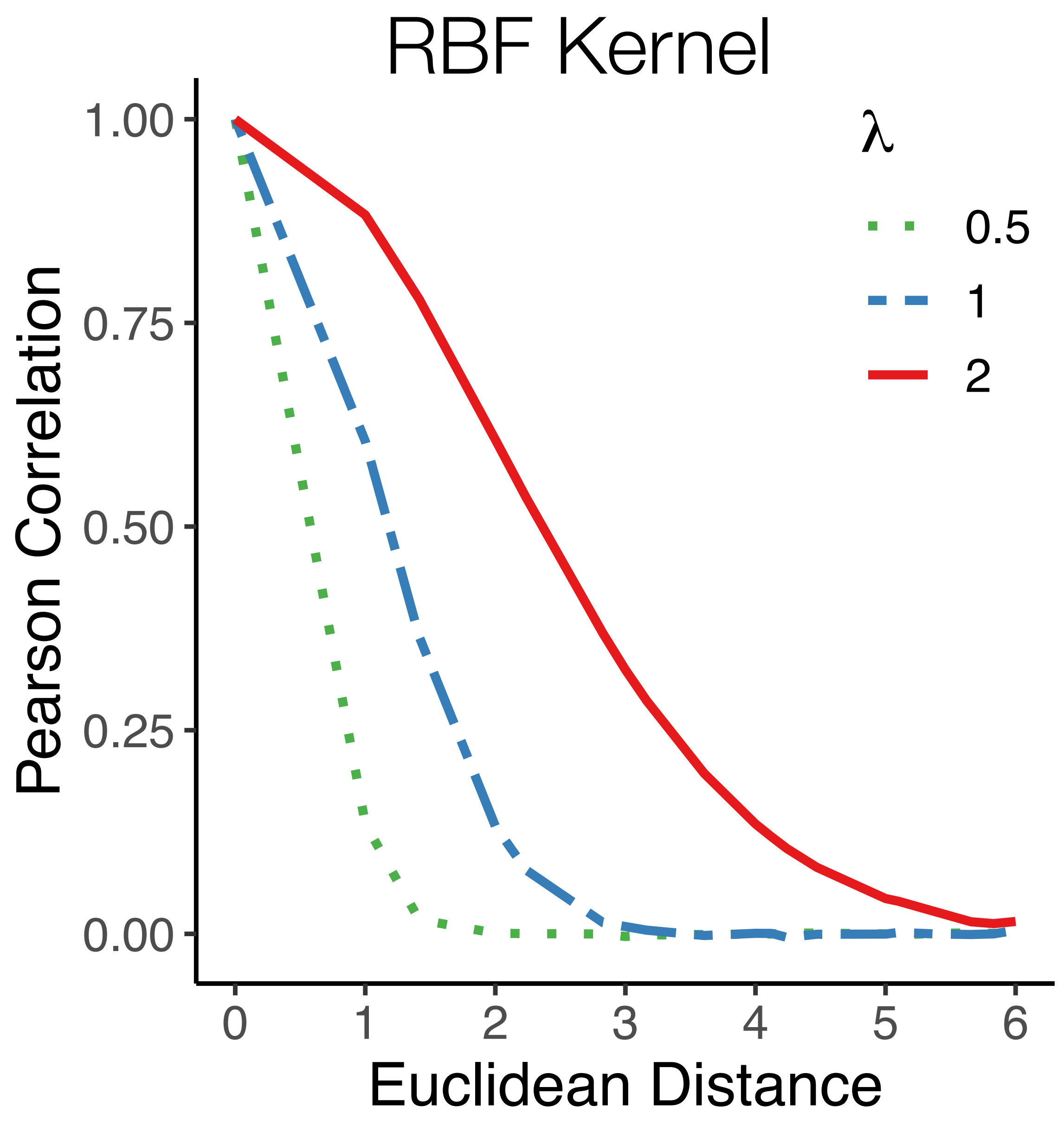




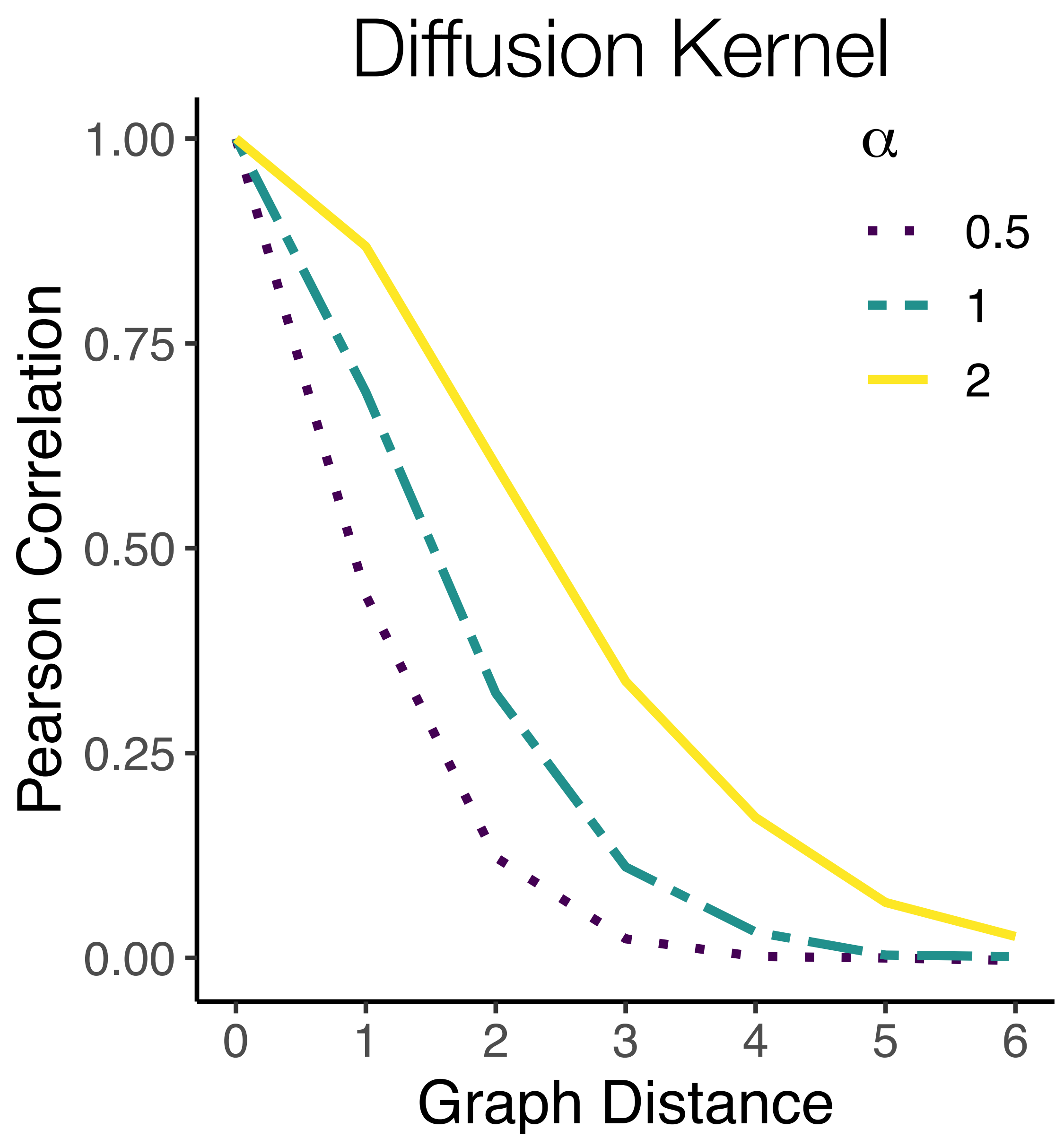
From continuous to structured spaces



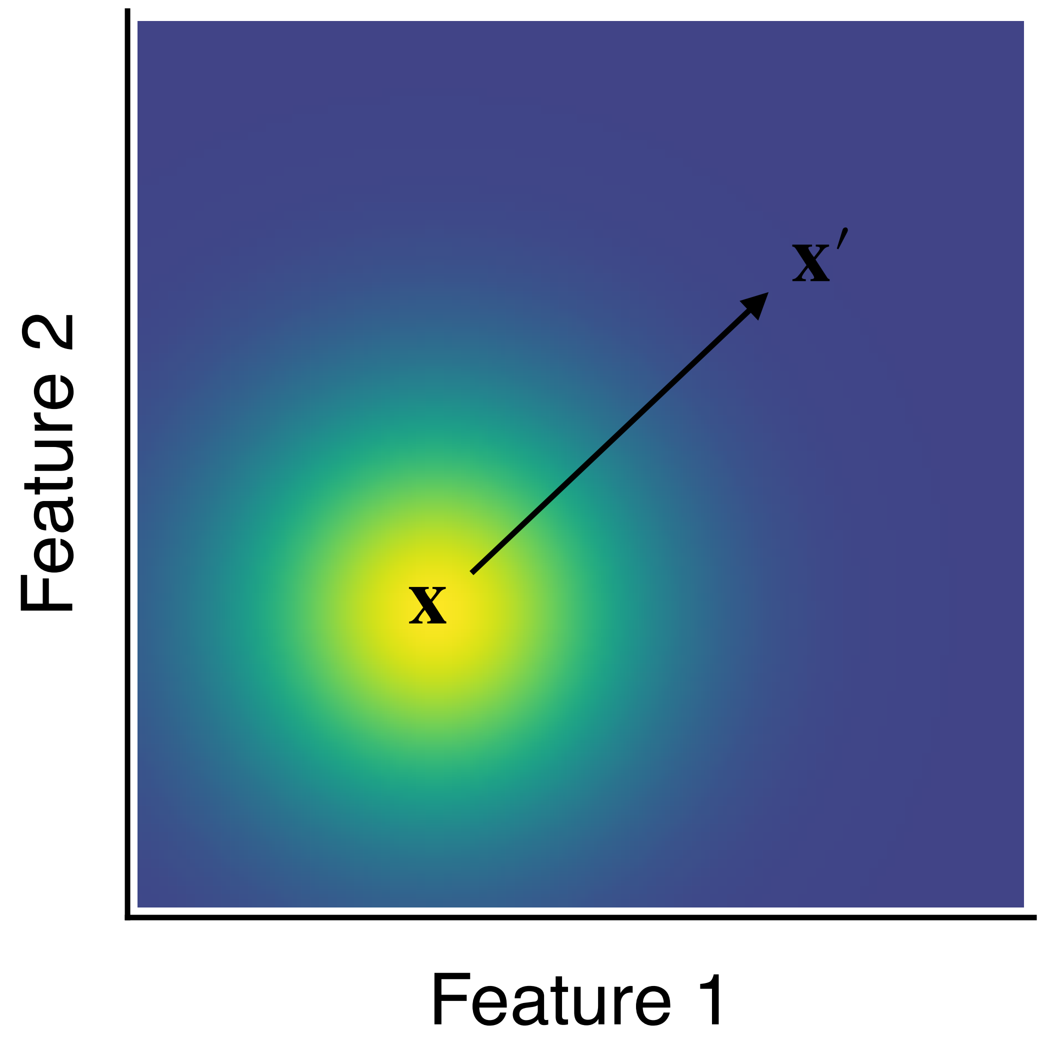


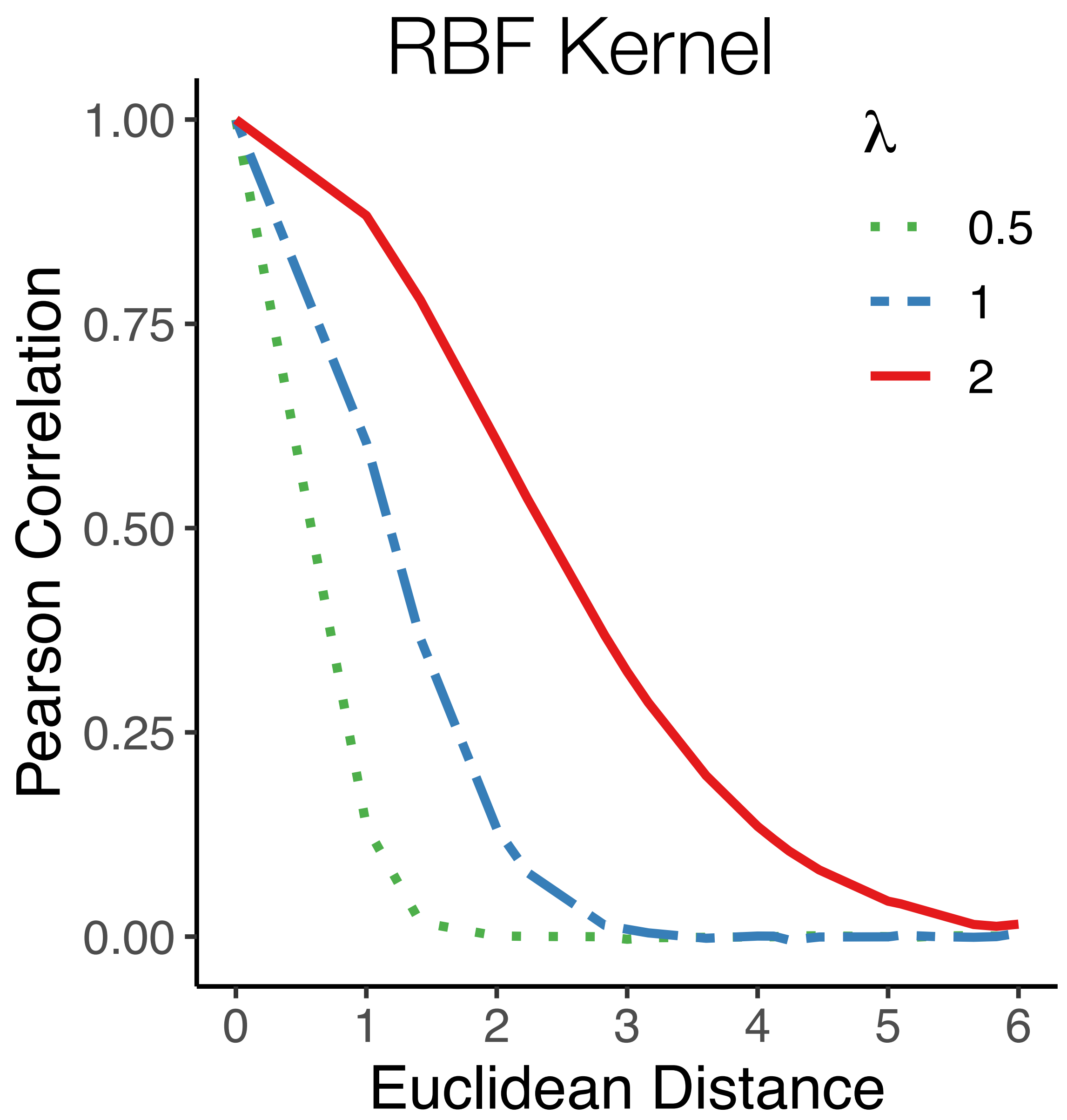


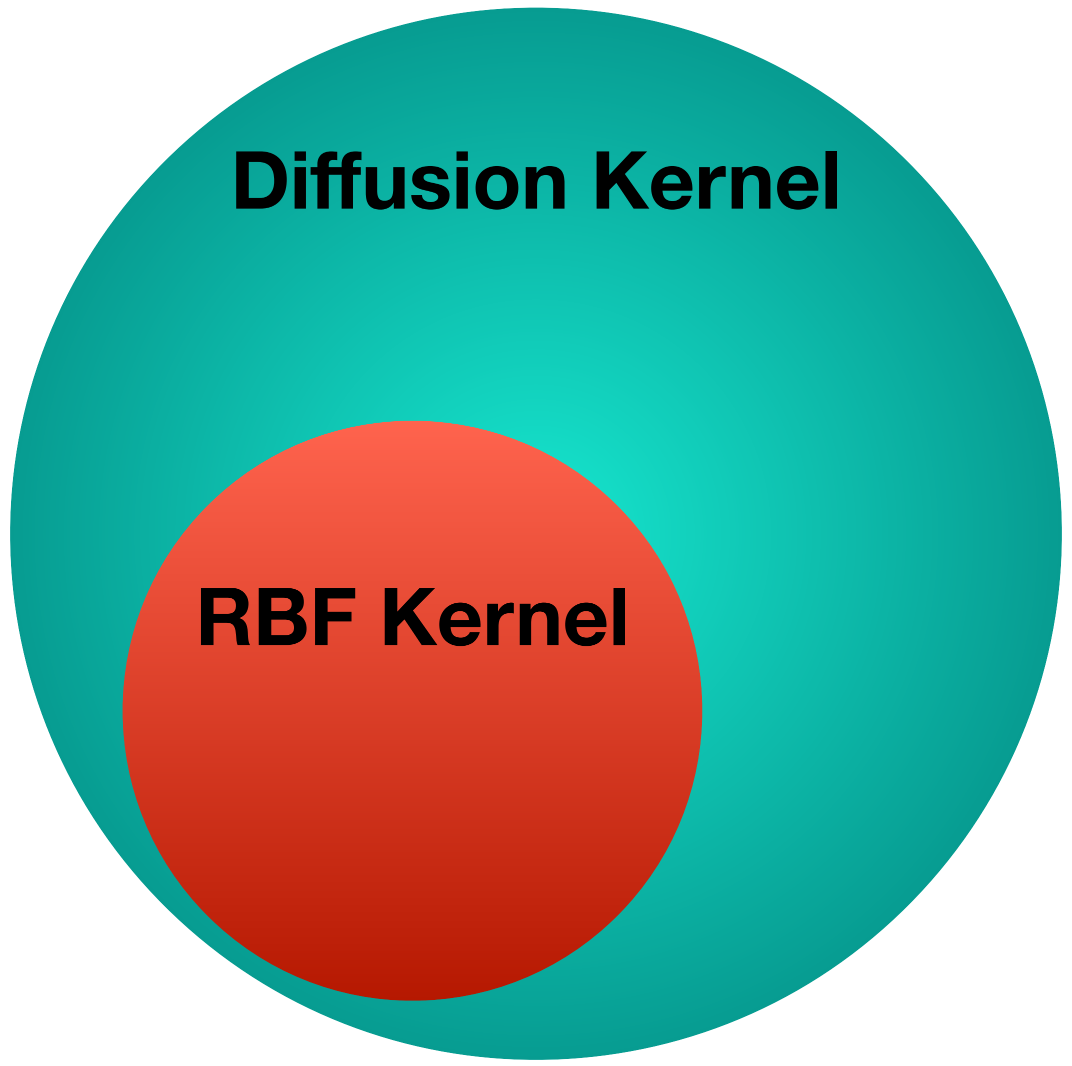
=

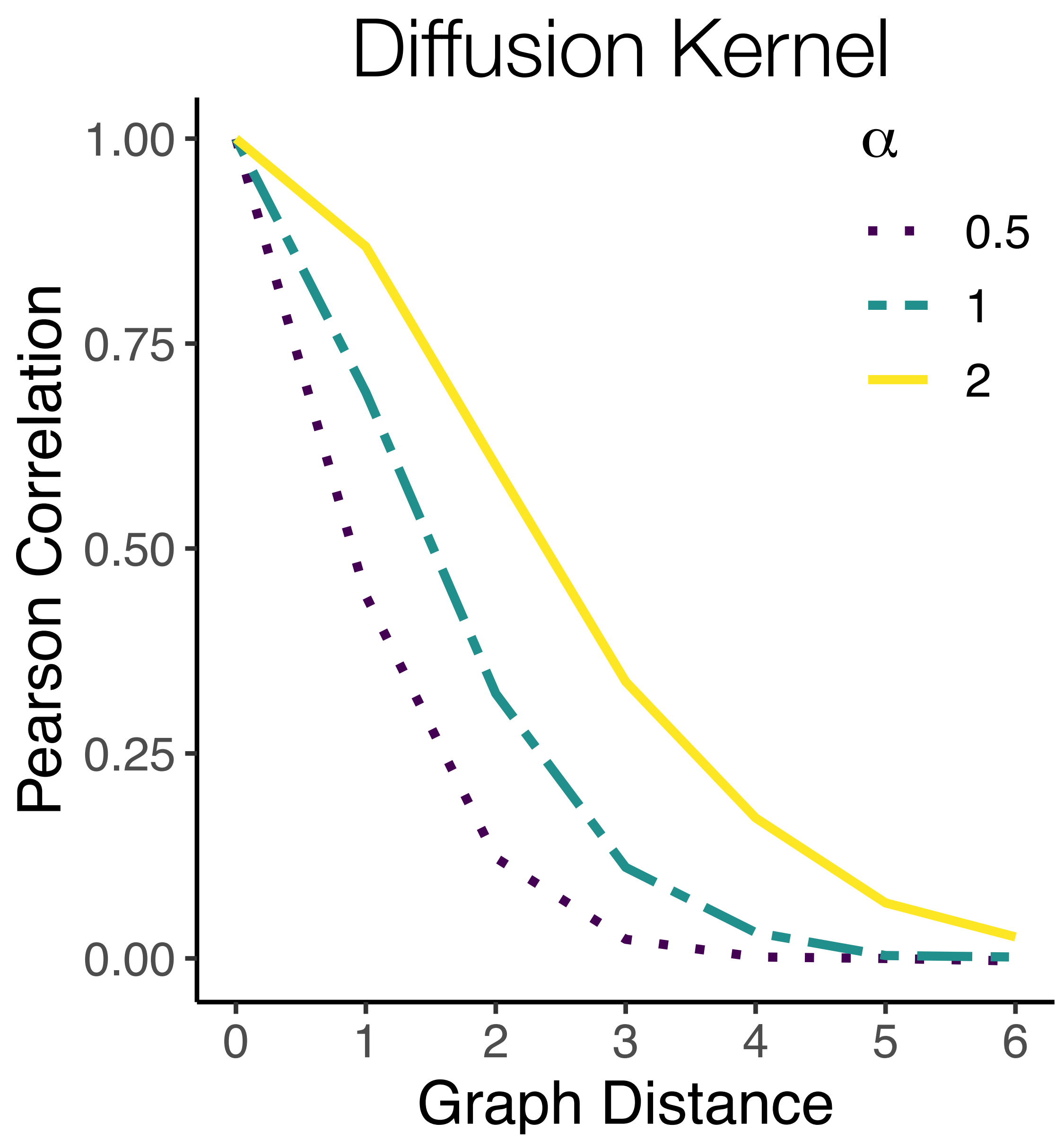


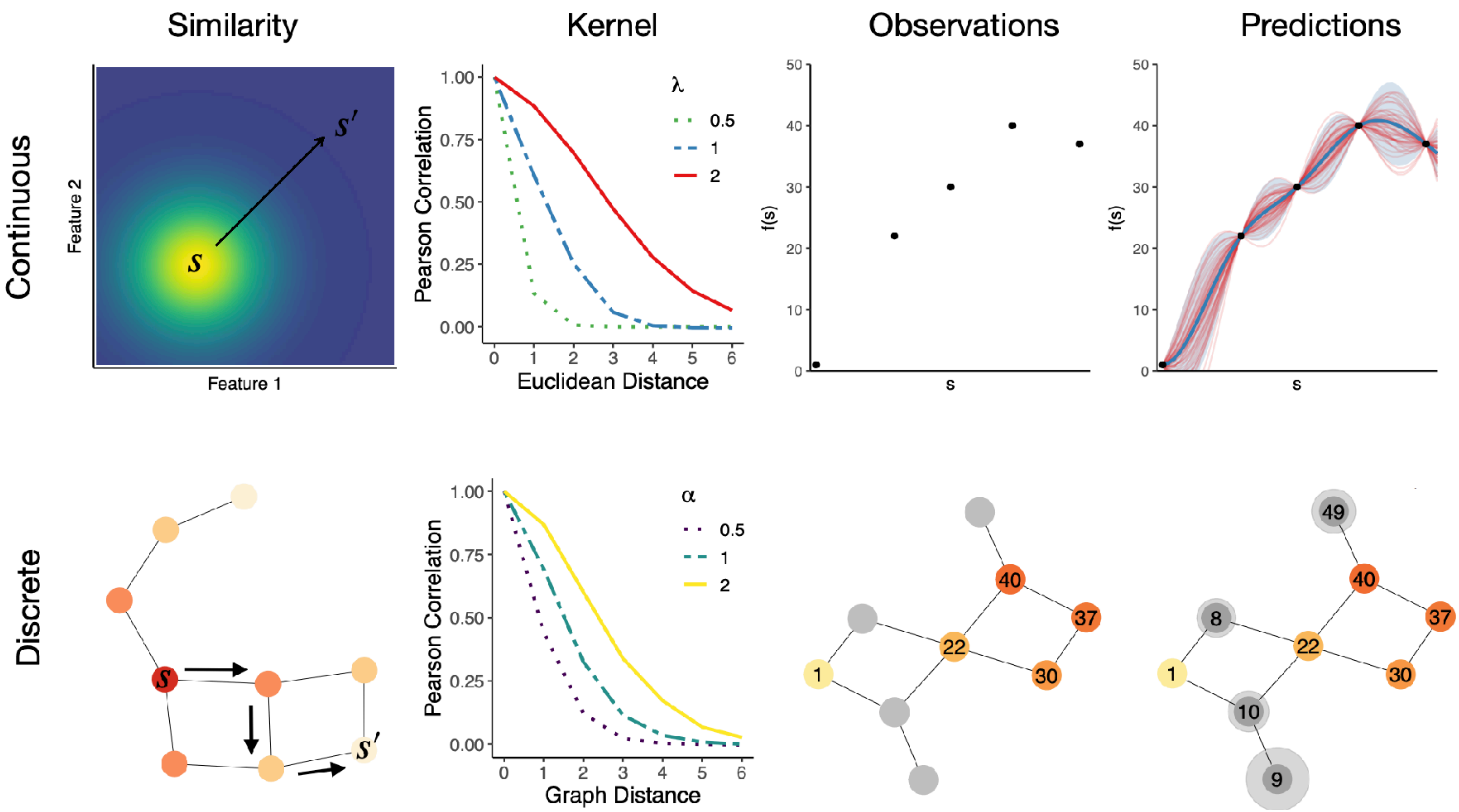
From continuous to structured spaces



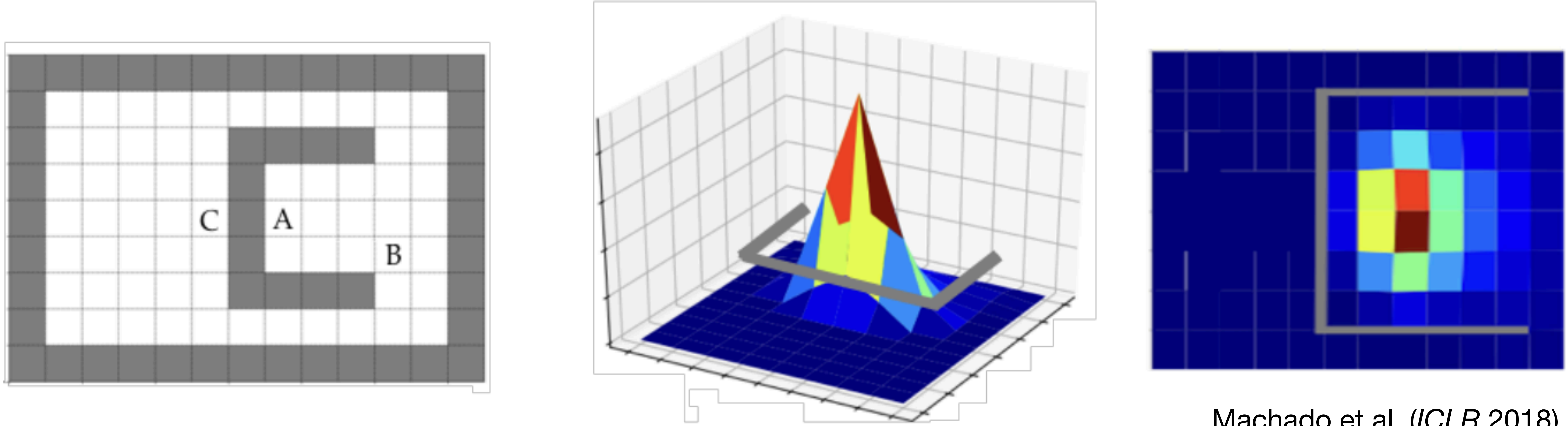







Similarity
Kernel
Observations
Predictions
Continuous
Discrete
Feature 2
Feature 1
s
s′
Pearson Correlation
Euclidean Distance
λ
0.5
1
2
f(s)
s
Graph Distance
α
0.5
1
2
40
37
22
30
1
49
8
10
9

Generalization based on transition dynamics



Machado et al. (*ICLR* 2018)

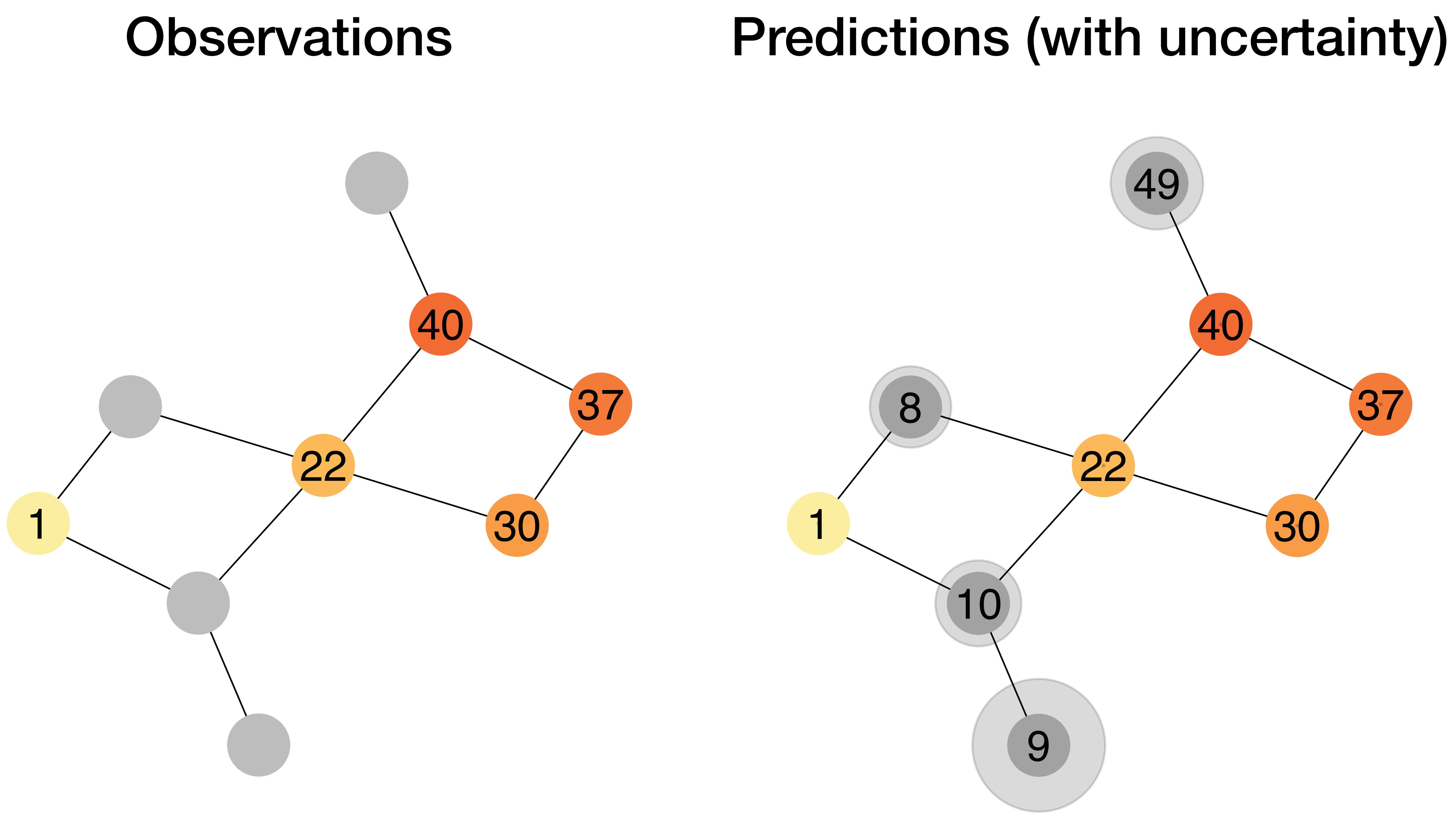
- A indicates a reward
- Even though C is closer than B, the transition dynamics of the environment make it easier for B to reach A

Diffusion Kernel

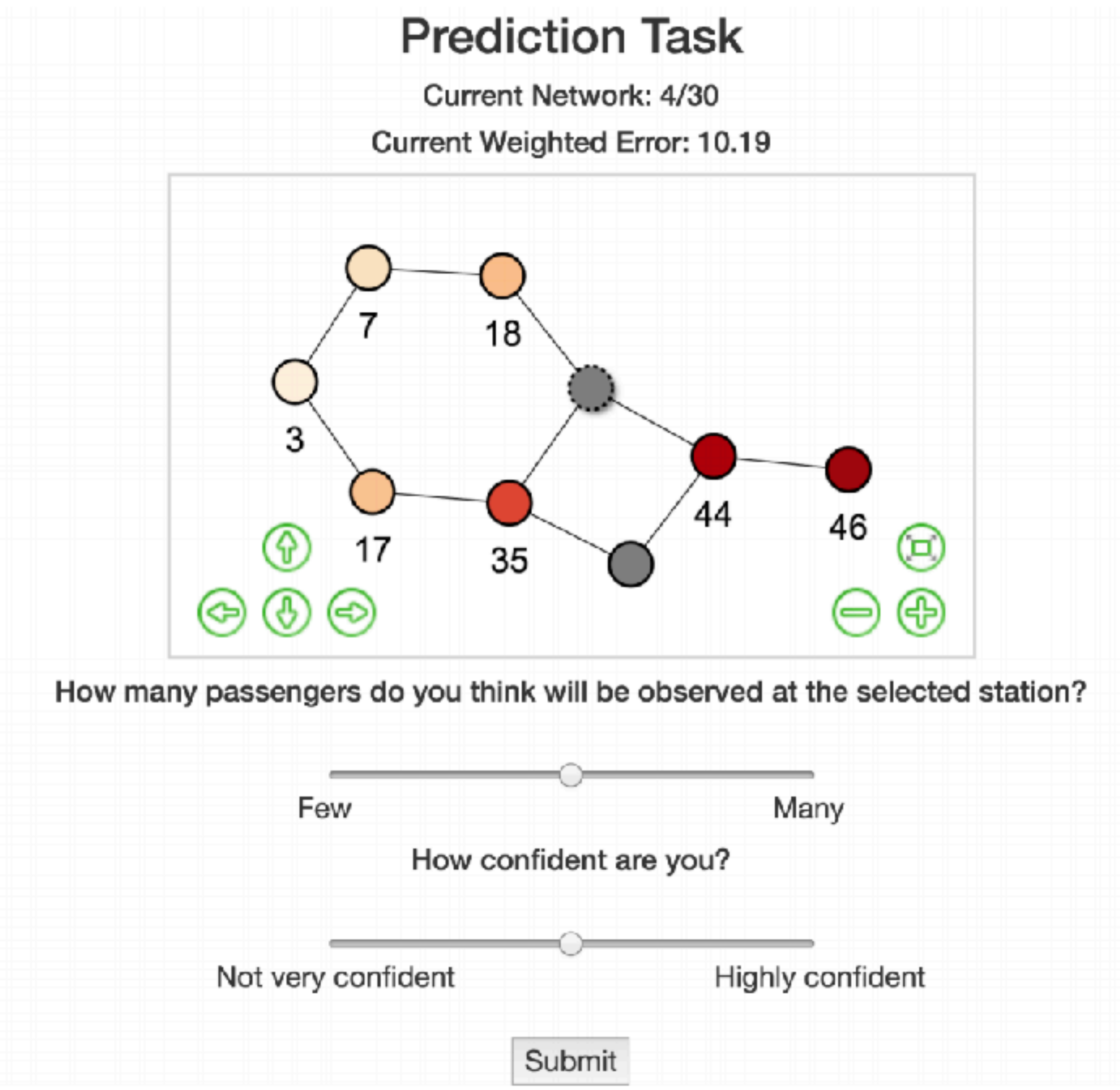
- Rather than similarity between features, we use the connectivity structure of the graph to define similarity

$$k_{DF}(s, s') = \exp(-\alpha L)$$

- Where L is the graph Laplacian
- α is a free parameter (diffusion level)
- The diffusion kernel assumes function values diffuse across the graph according to a random walk

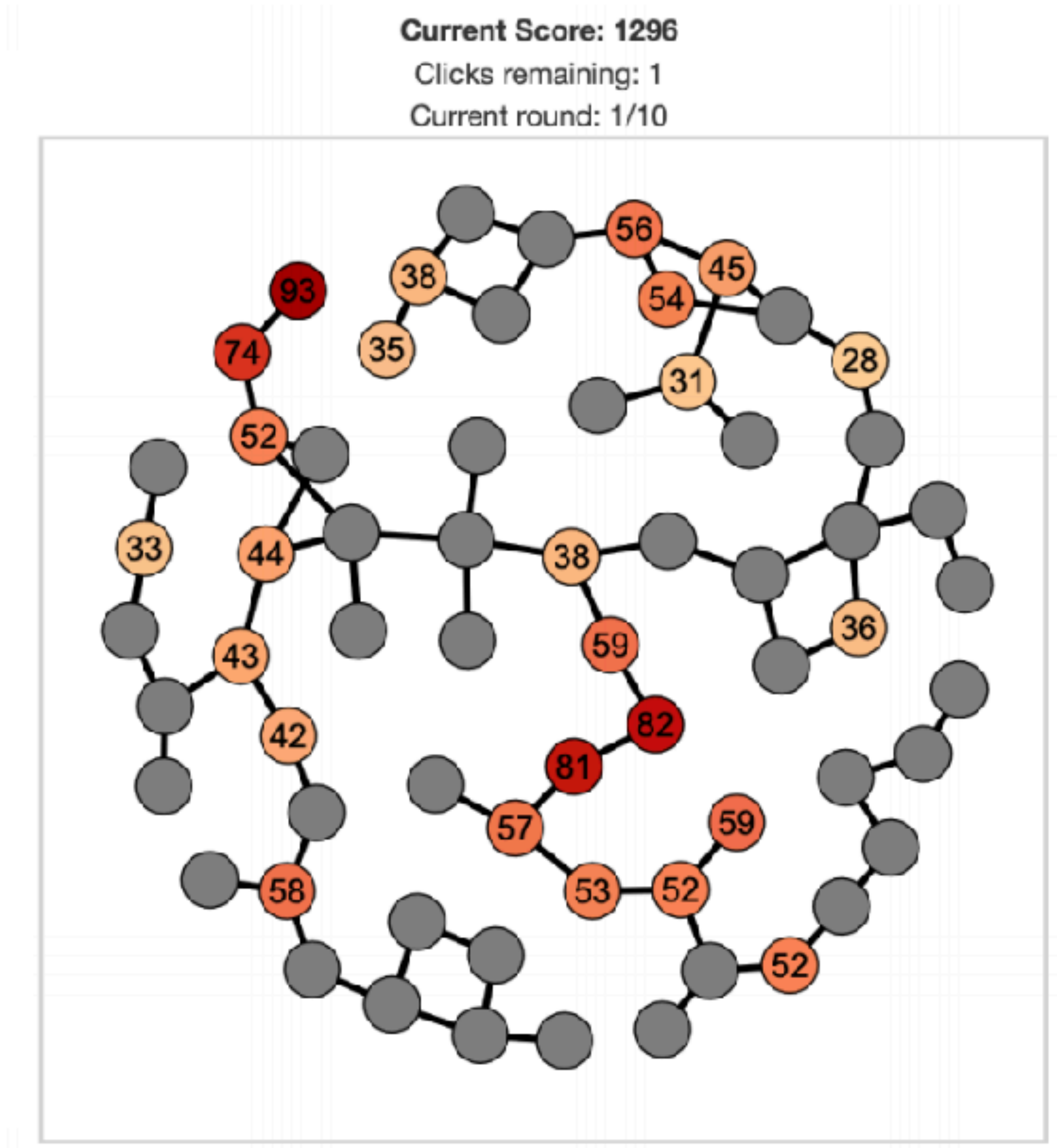


Experiment 1



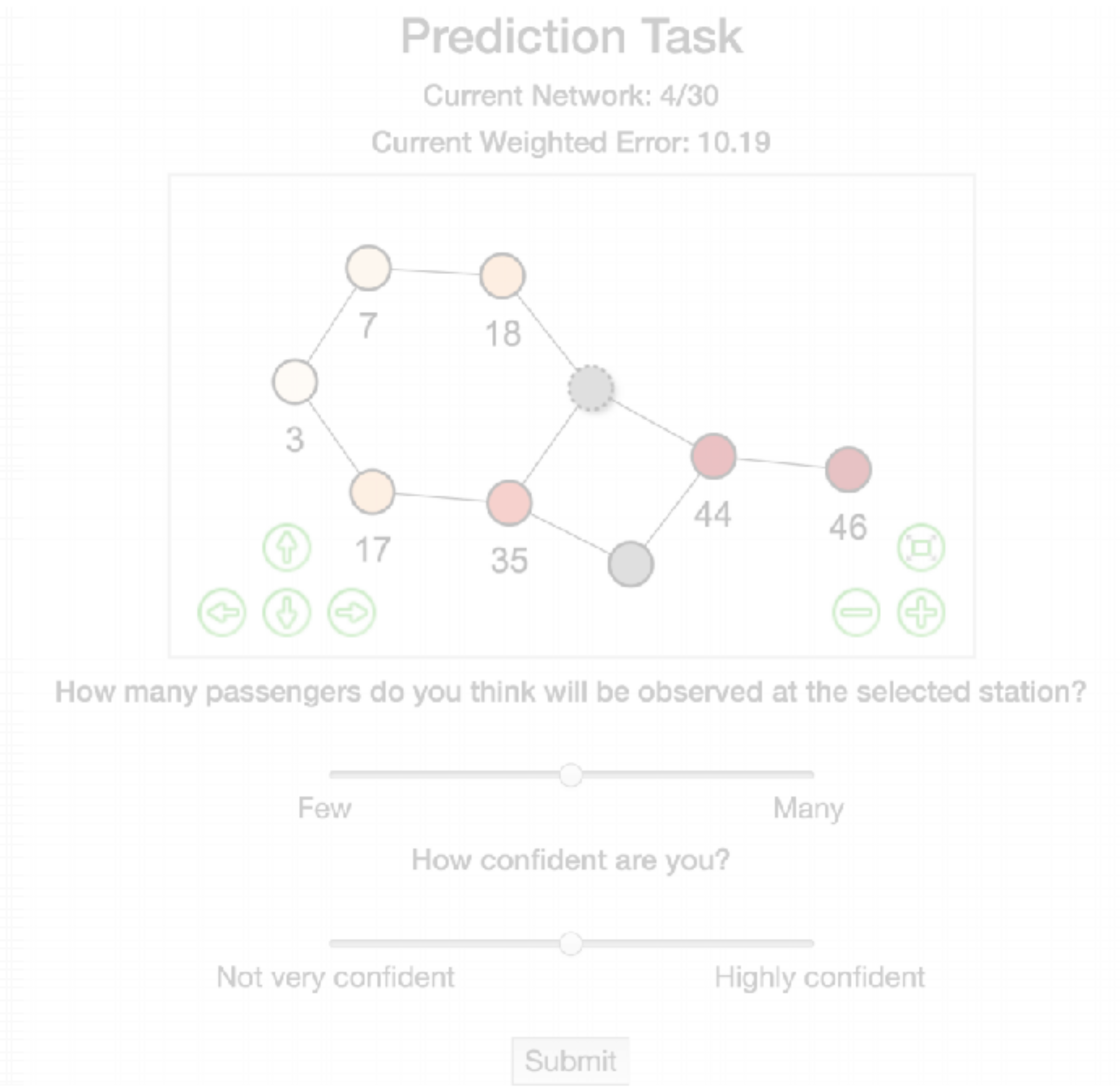
Experiment 2

Bandit Task



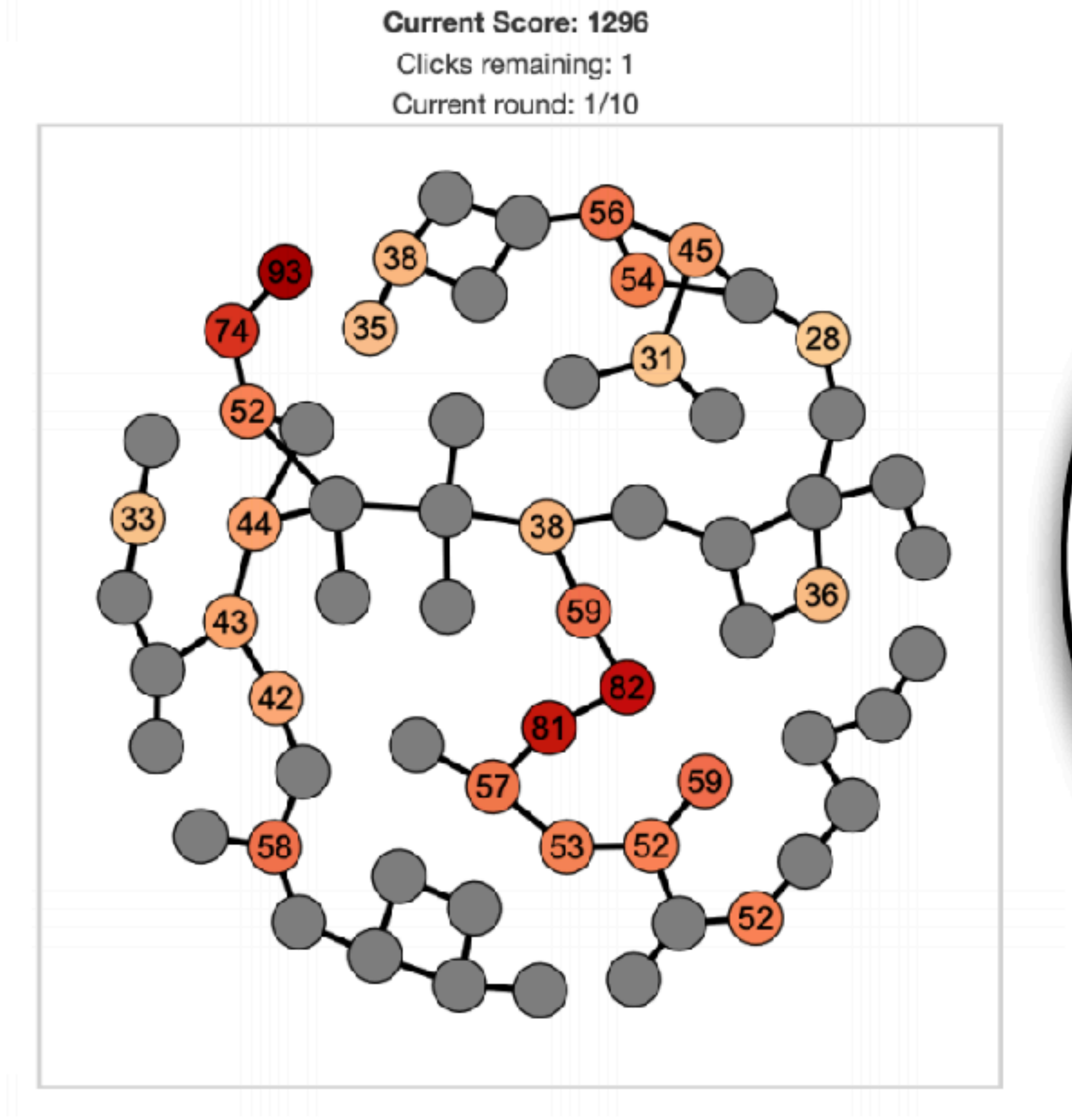
Bonus Round

Experiment 1



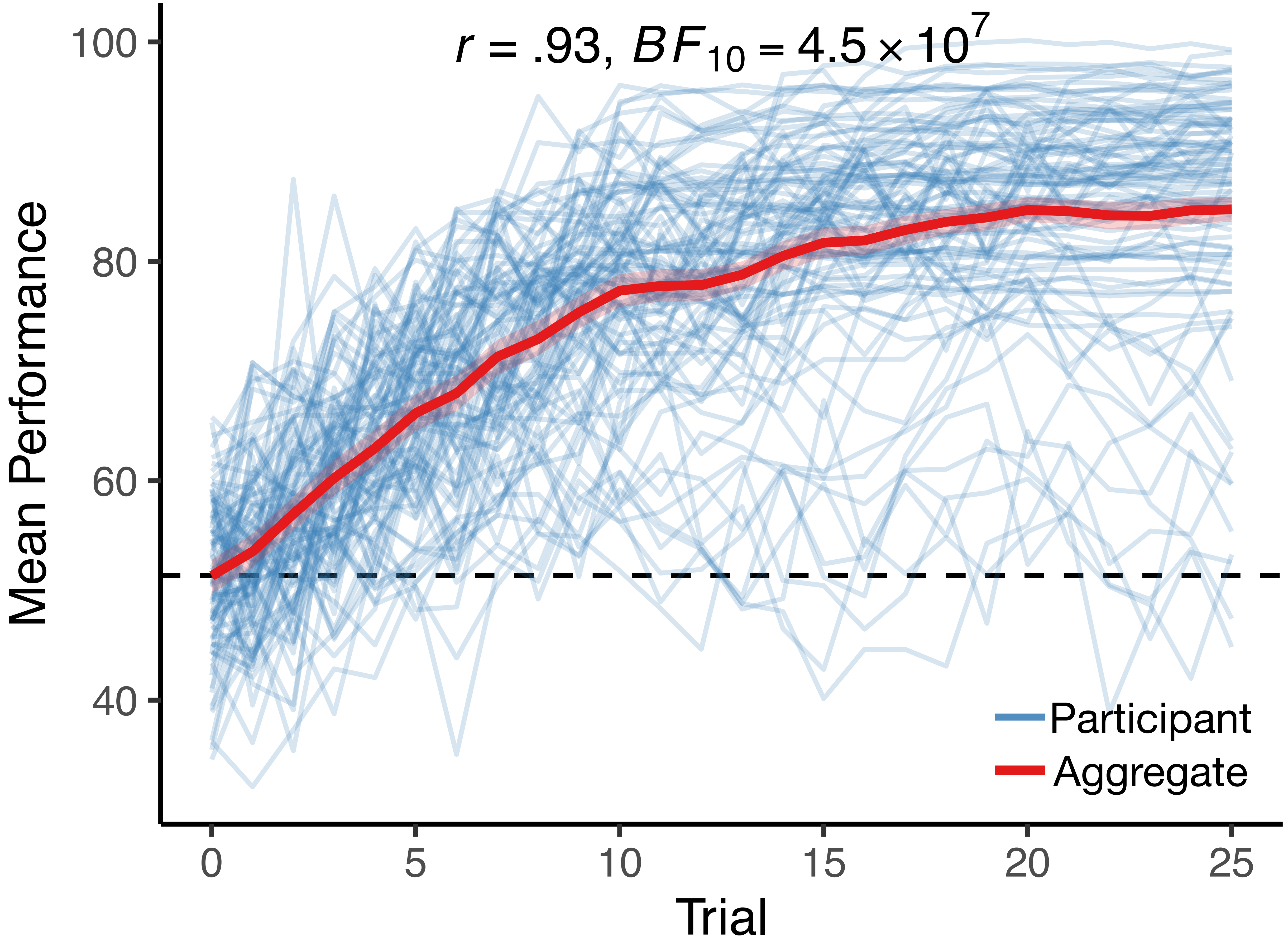
Experiment 2

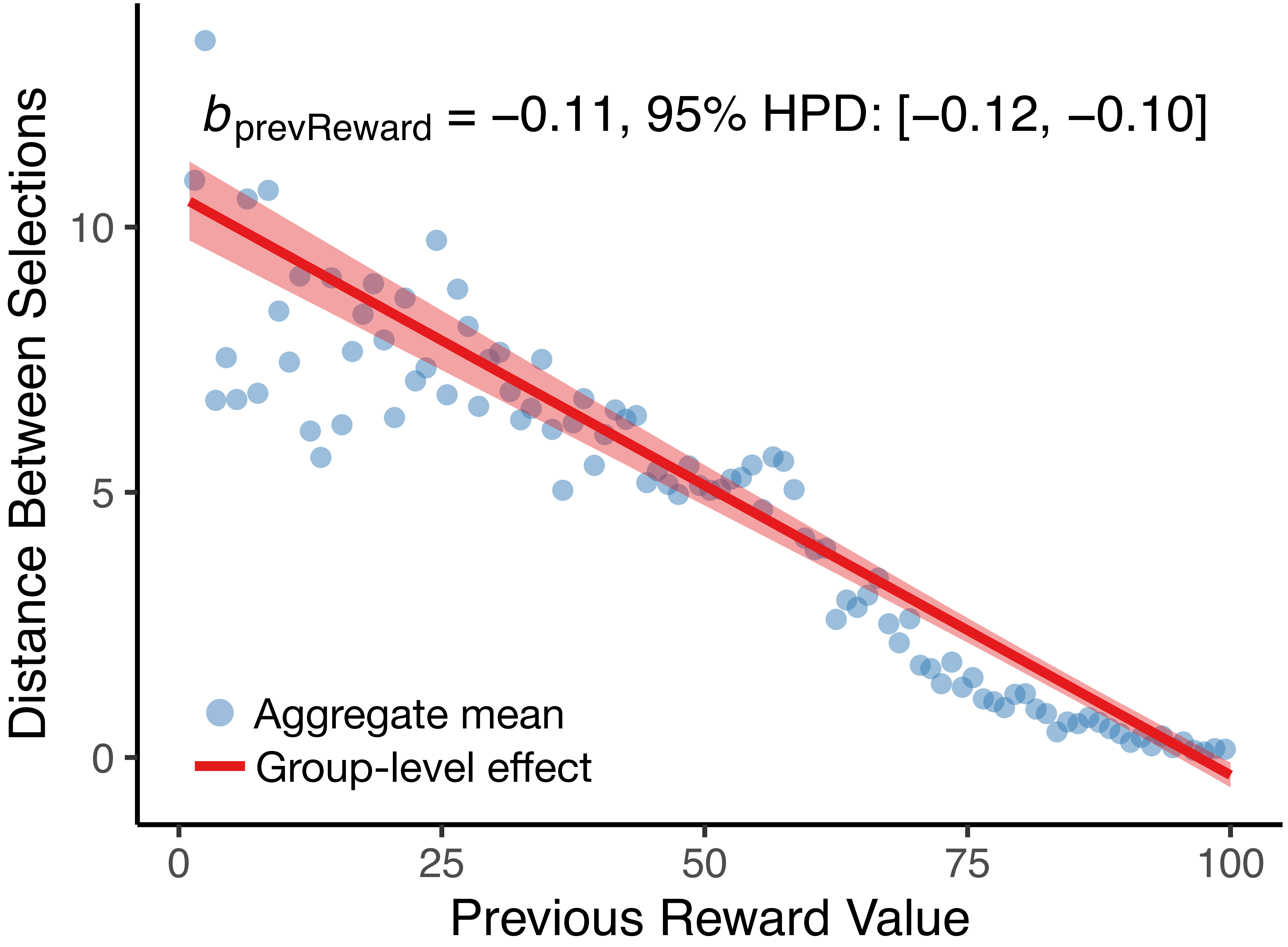
Bandit Task



Bonus Round

Behavioral Results

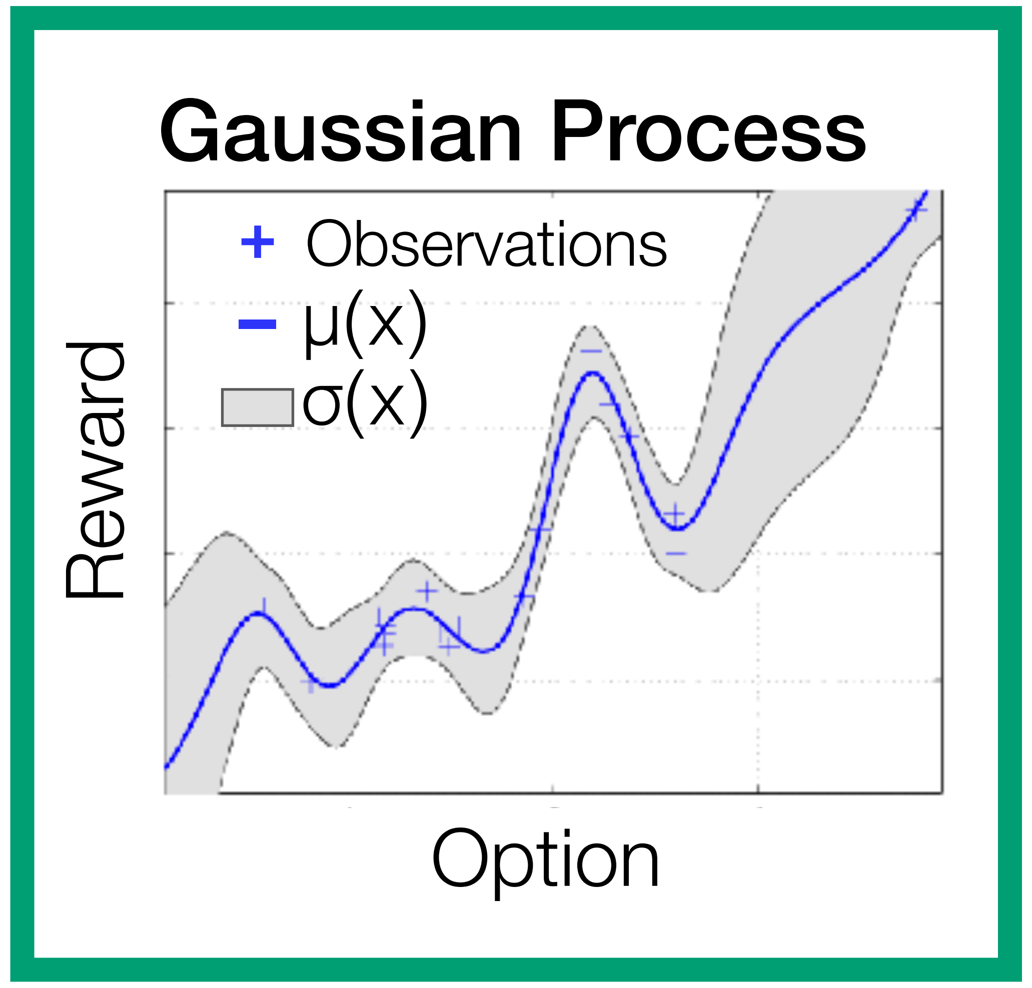




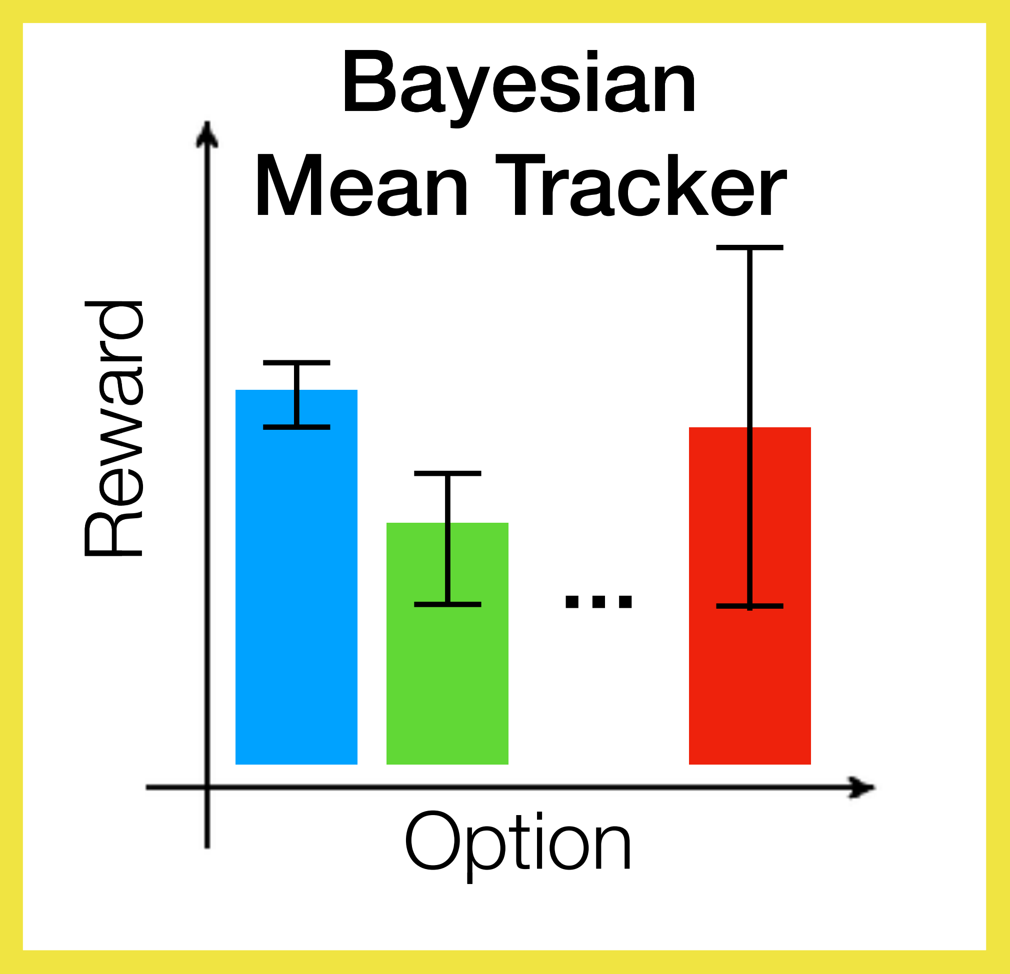
Wu, Schulz & Gershman (*CCN* 2019)

Model Results

Generalization



No generalization

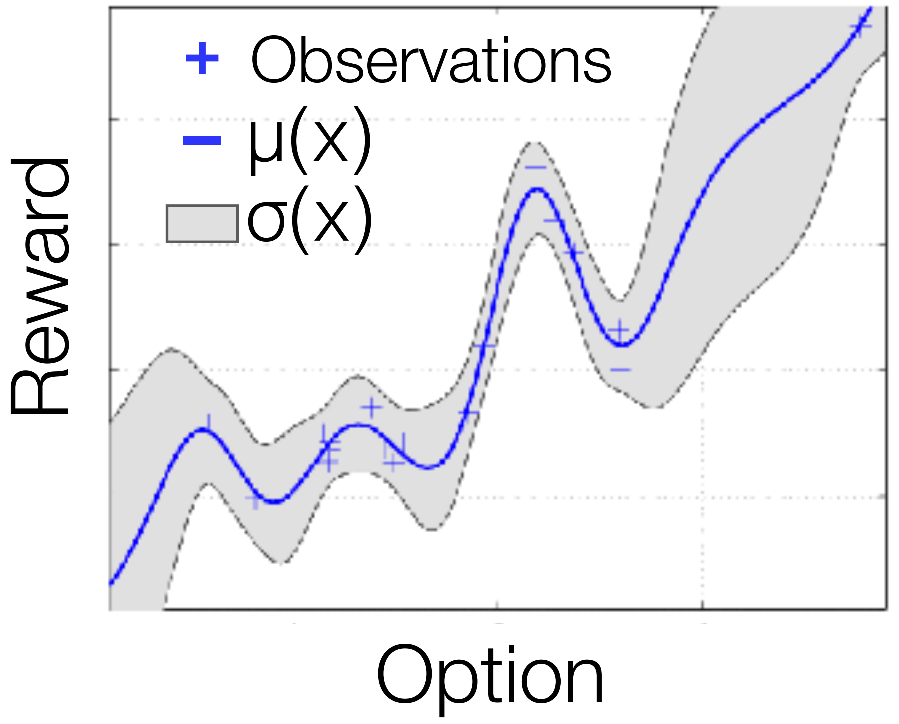


Model Results

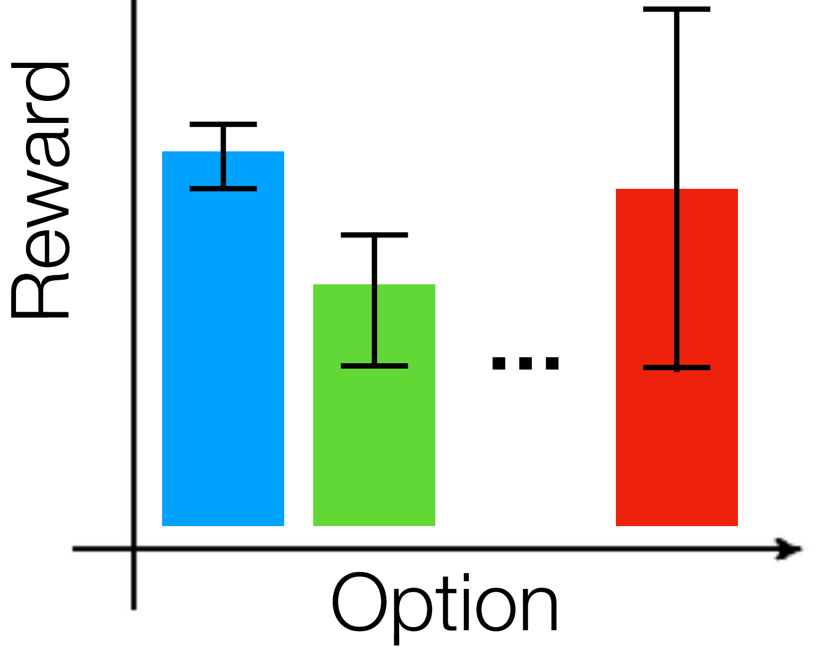
Generalization

No generalization

Gaussian Process

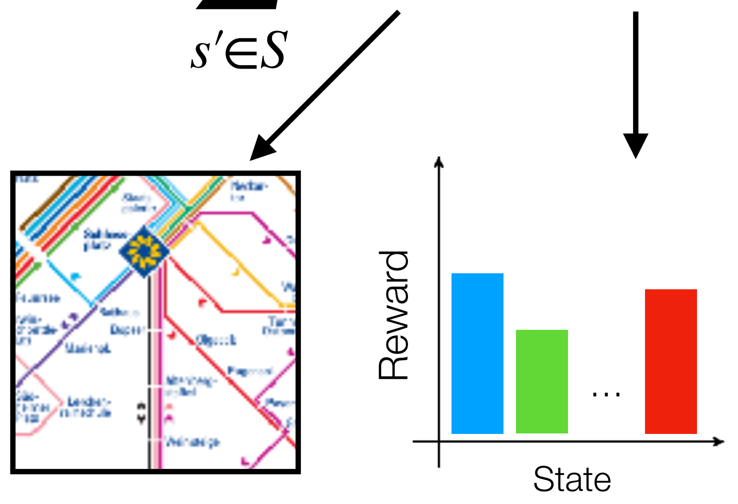


Bayesian Mean Tracker



Successor Representation

$$V^{\pi}(s,a) = \sum_{s' \in S} M(s,s',a)R(s')$$

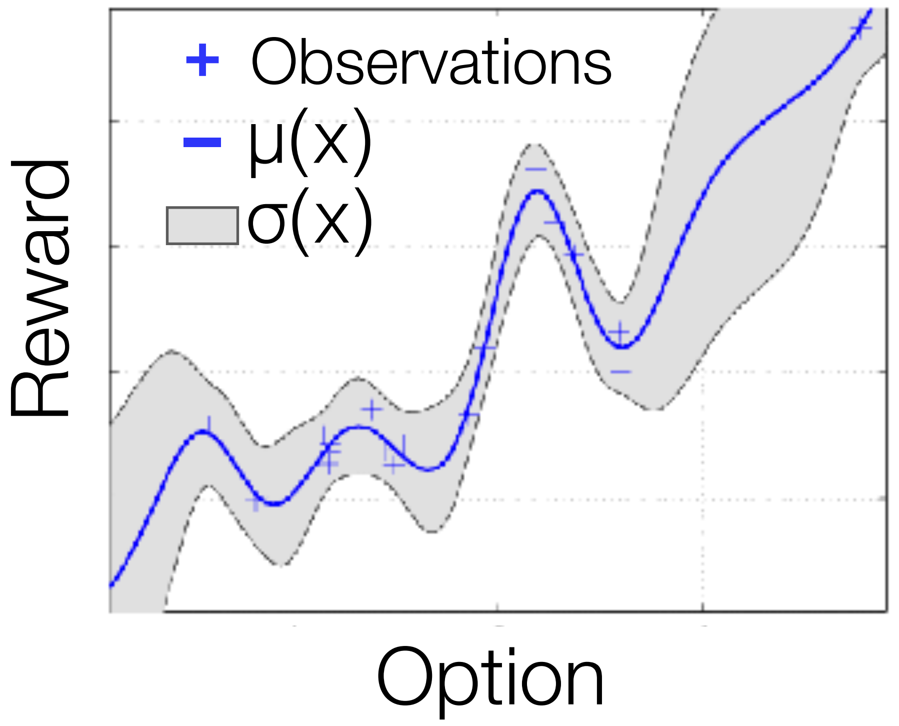


Model Results

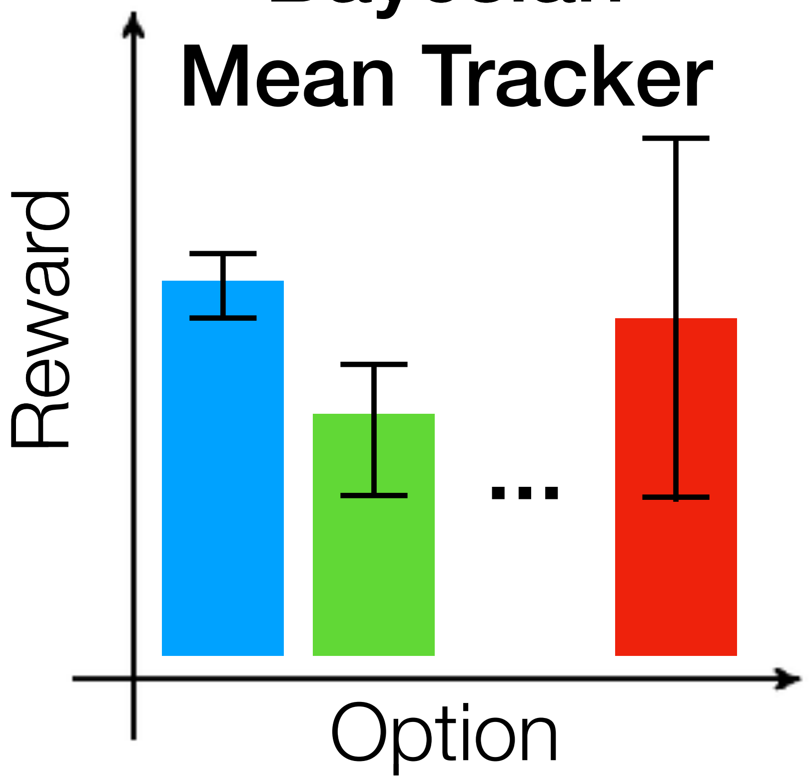
Generalization | **No generalization**

directed + random exploration

Gaussian Process



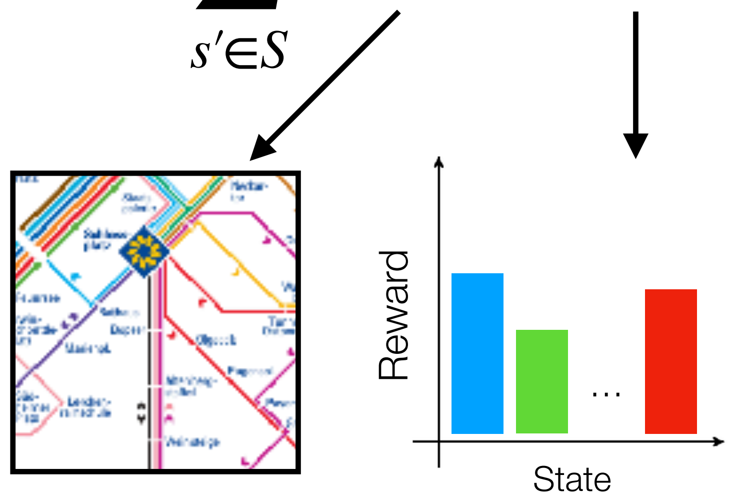
Bayesian Mean Tracker



random exploration

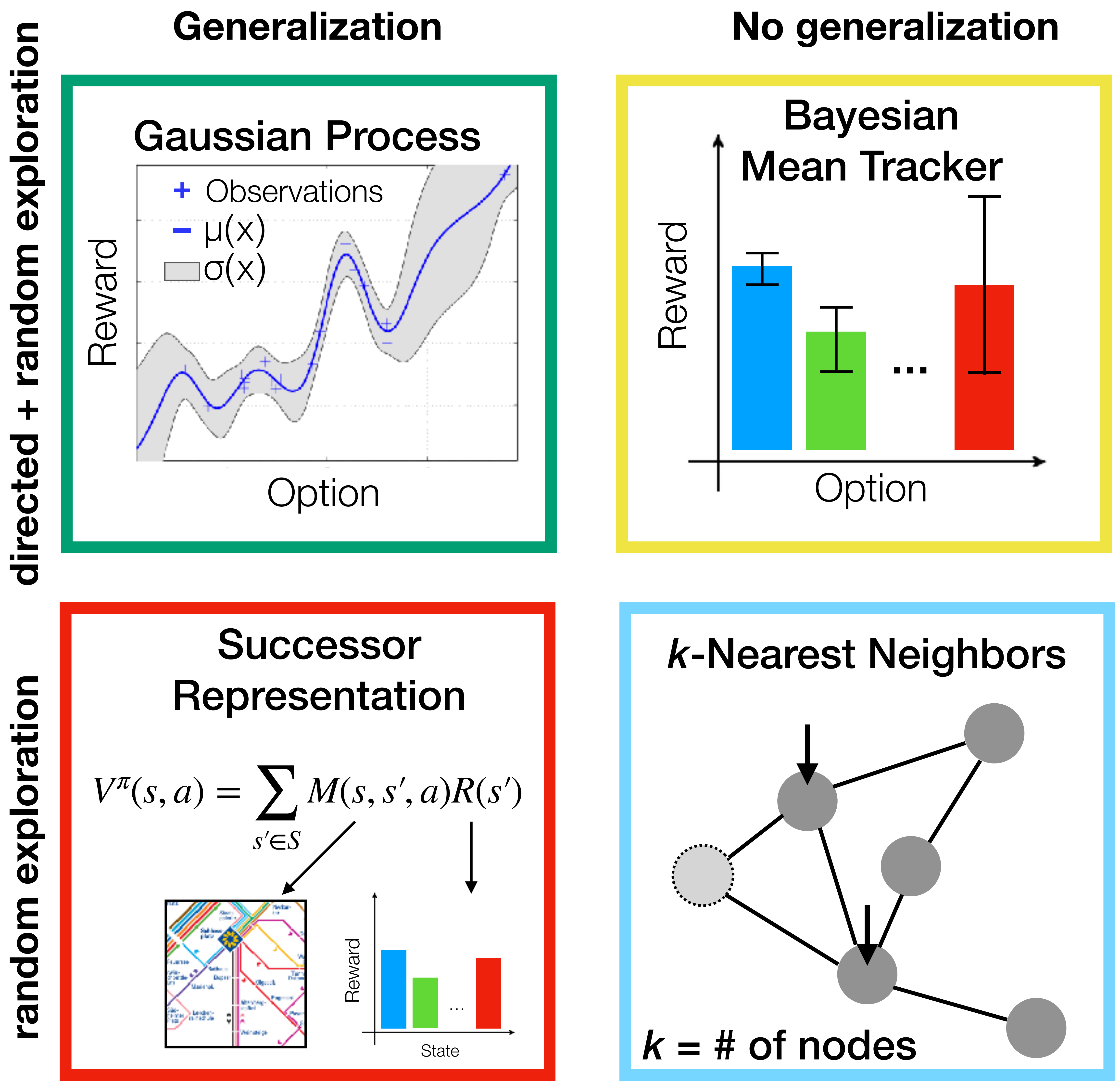
Successor Representation

$$V^{\pi}(s,a) = \sum_{s' \in S} M(s,s',a)R(s')$$



Wu, Schulz & Gershman (*CCN* 2019)

Model Results

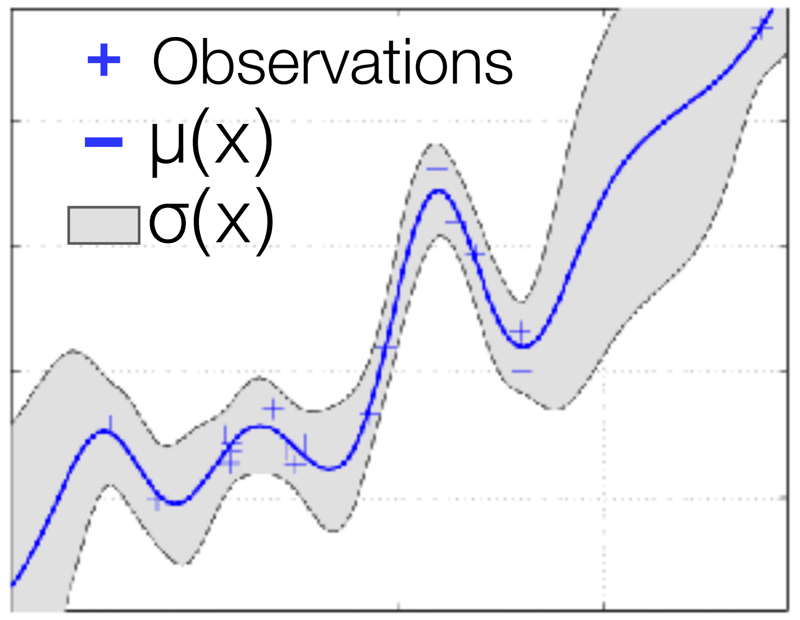


Model Results

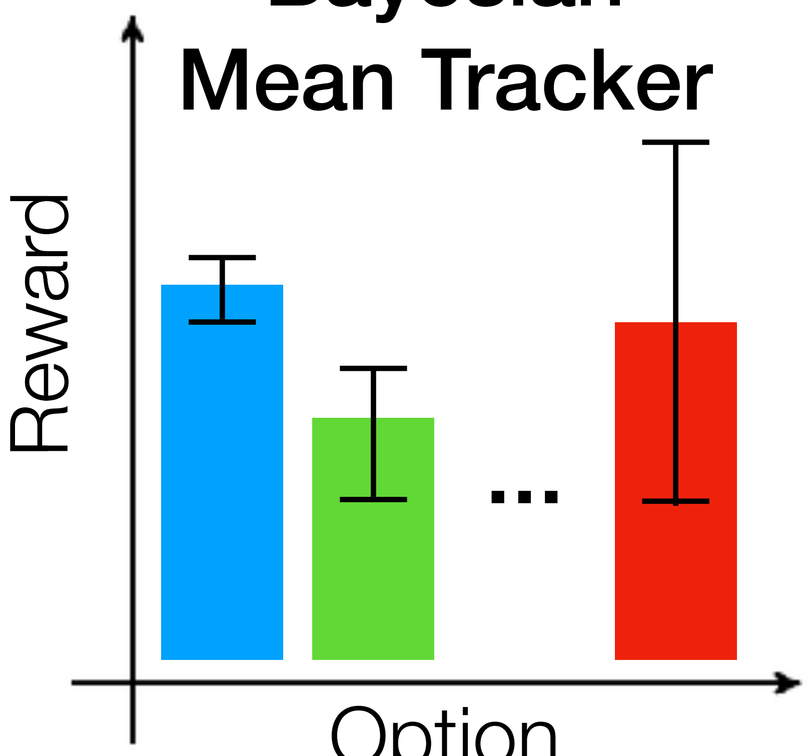
Generalization | **No generalization**

directed + random exploration

Gaussian Process



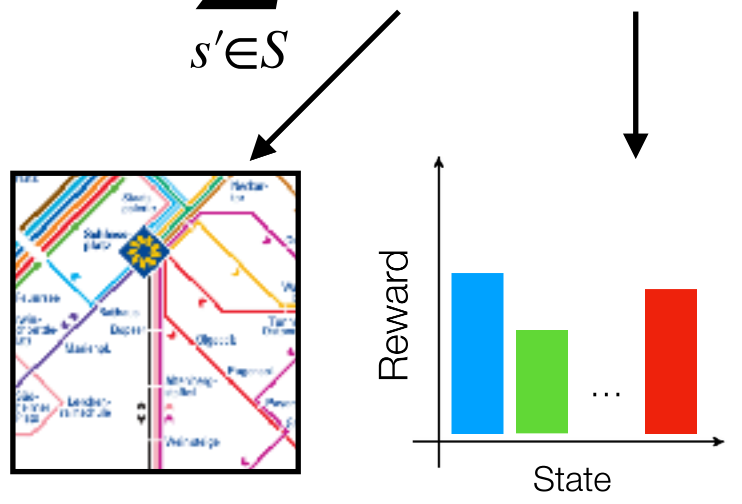
Bayesian Mean Tracker



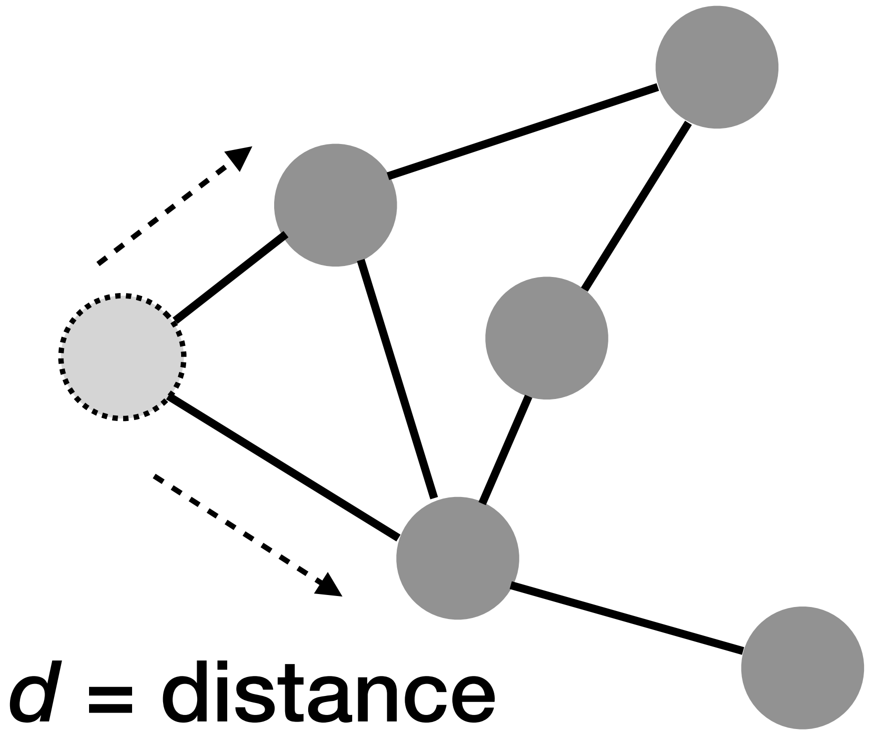
random exploration

Successor Representation

$$V^{\pi}(s,a) = \sum_{s' \in S} M(s,s',a)R(s')$$

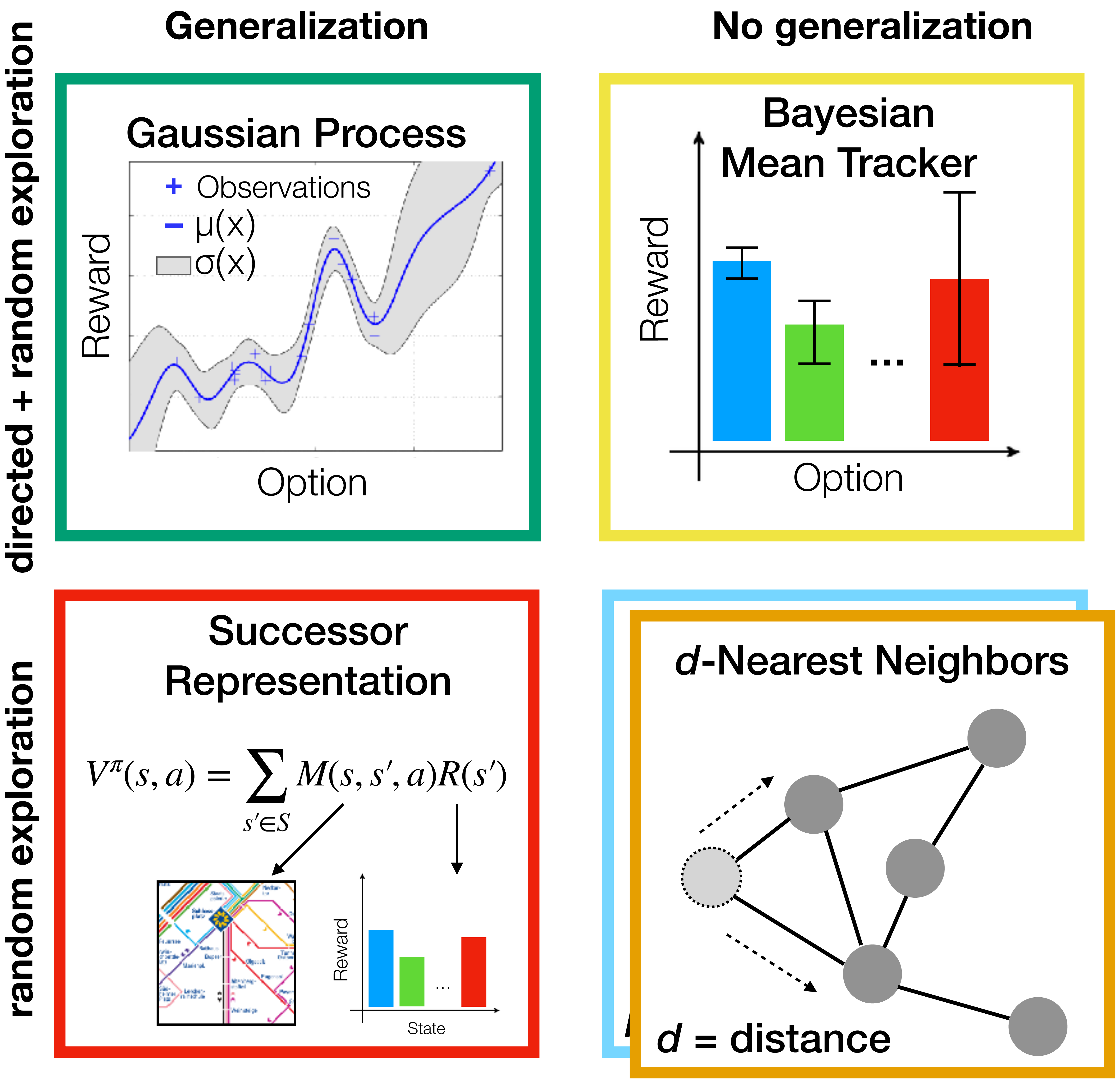


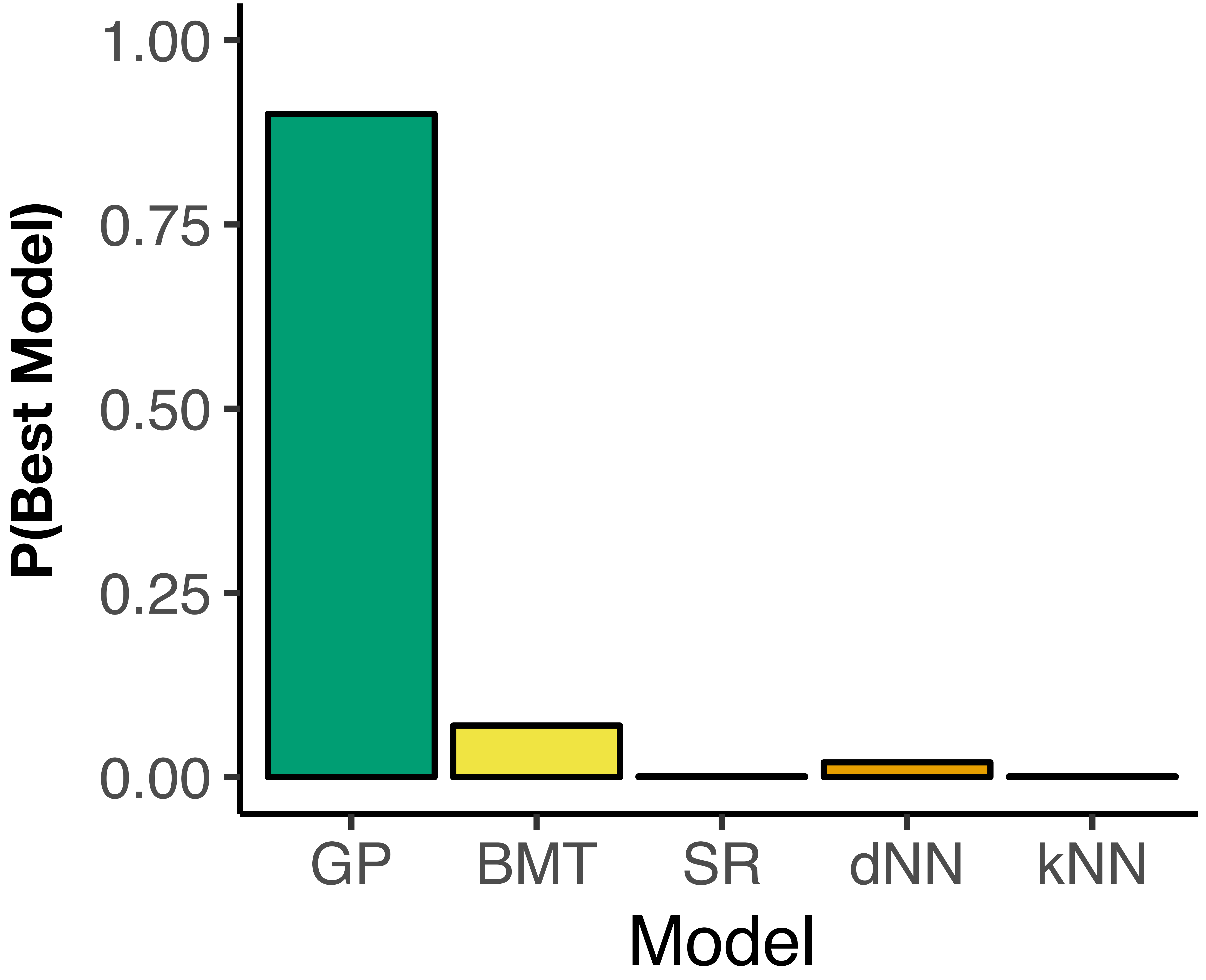
d-Nearest Neighbors



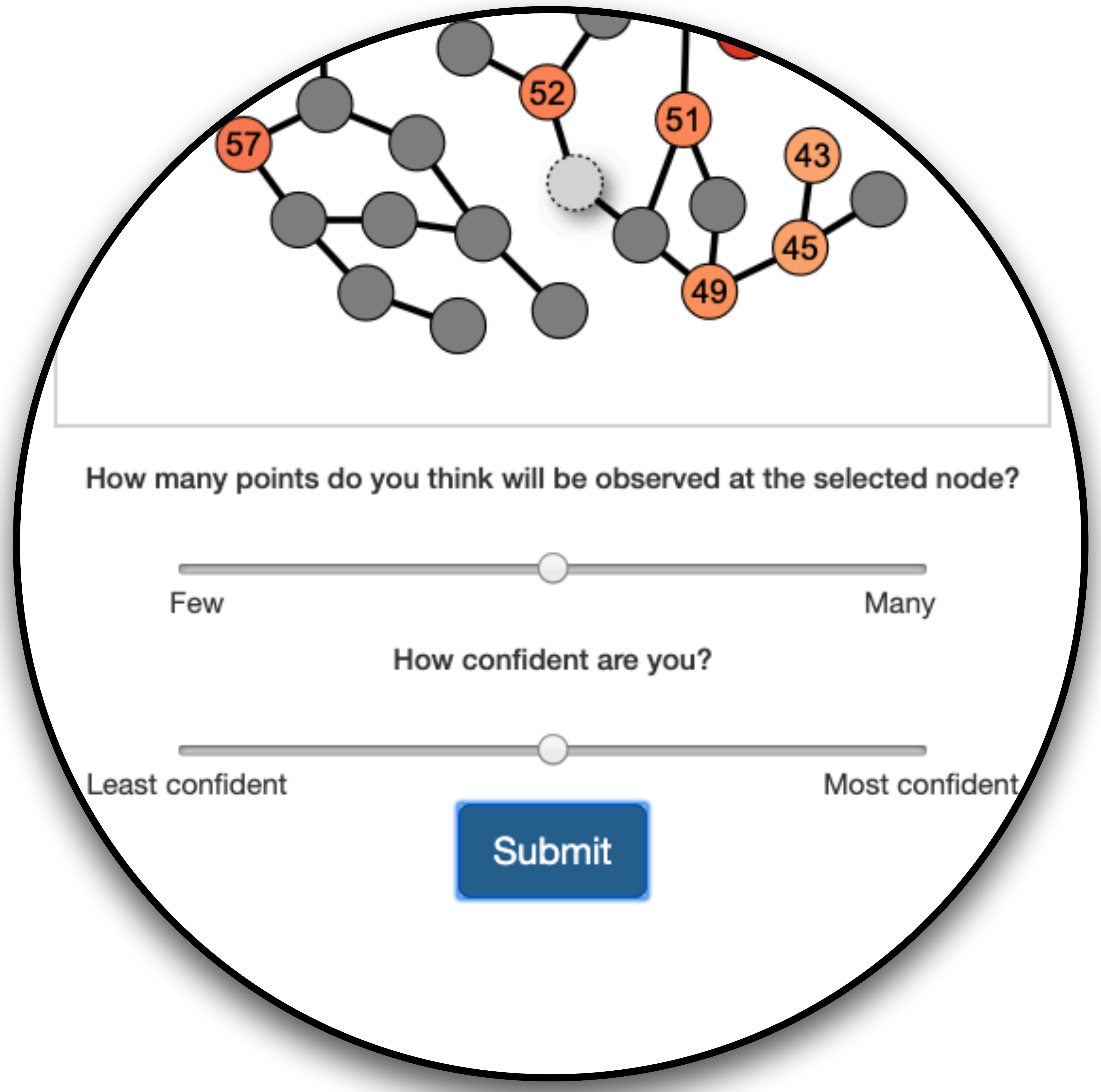
***d* = distance**

Model Results

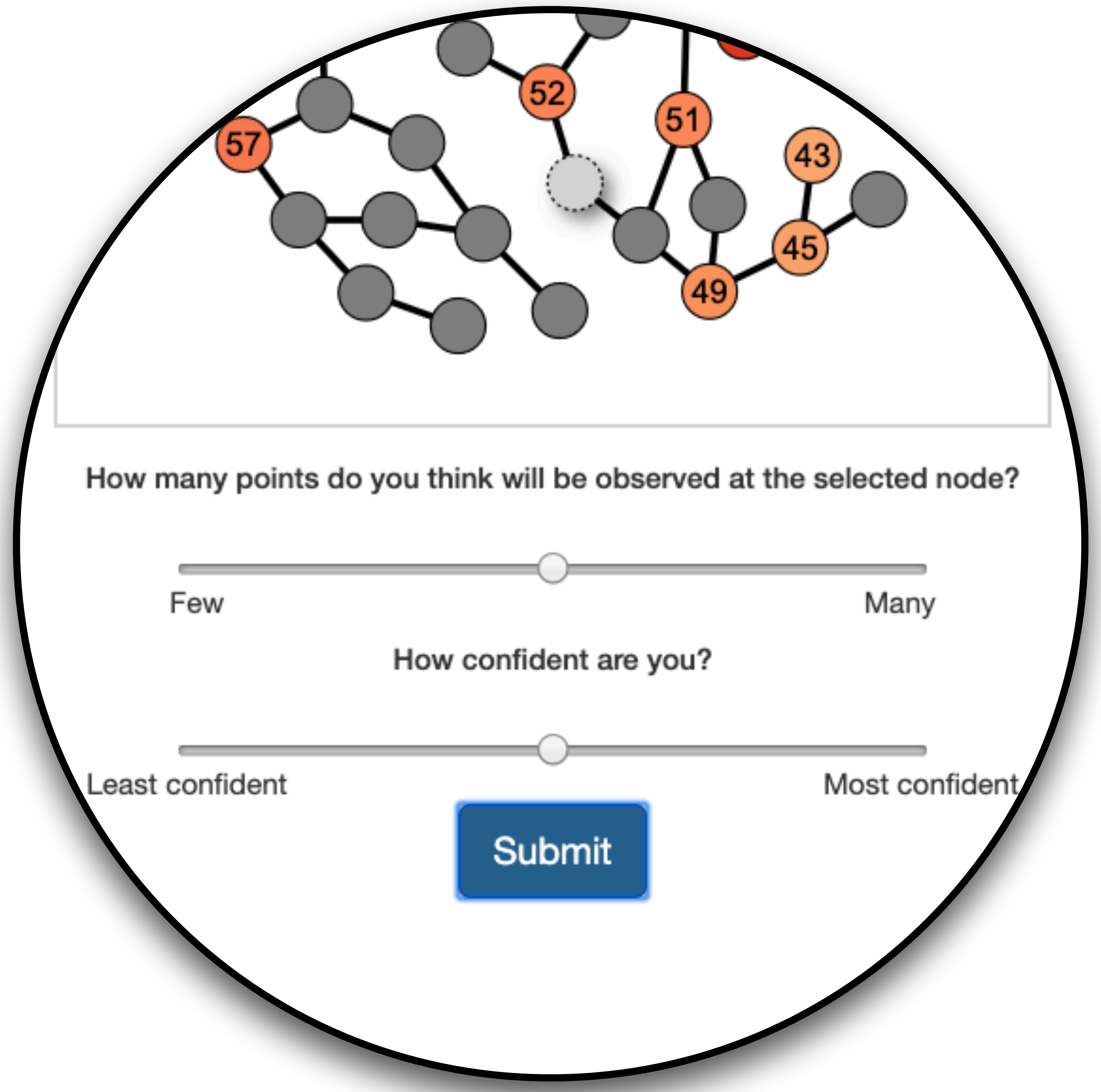


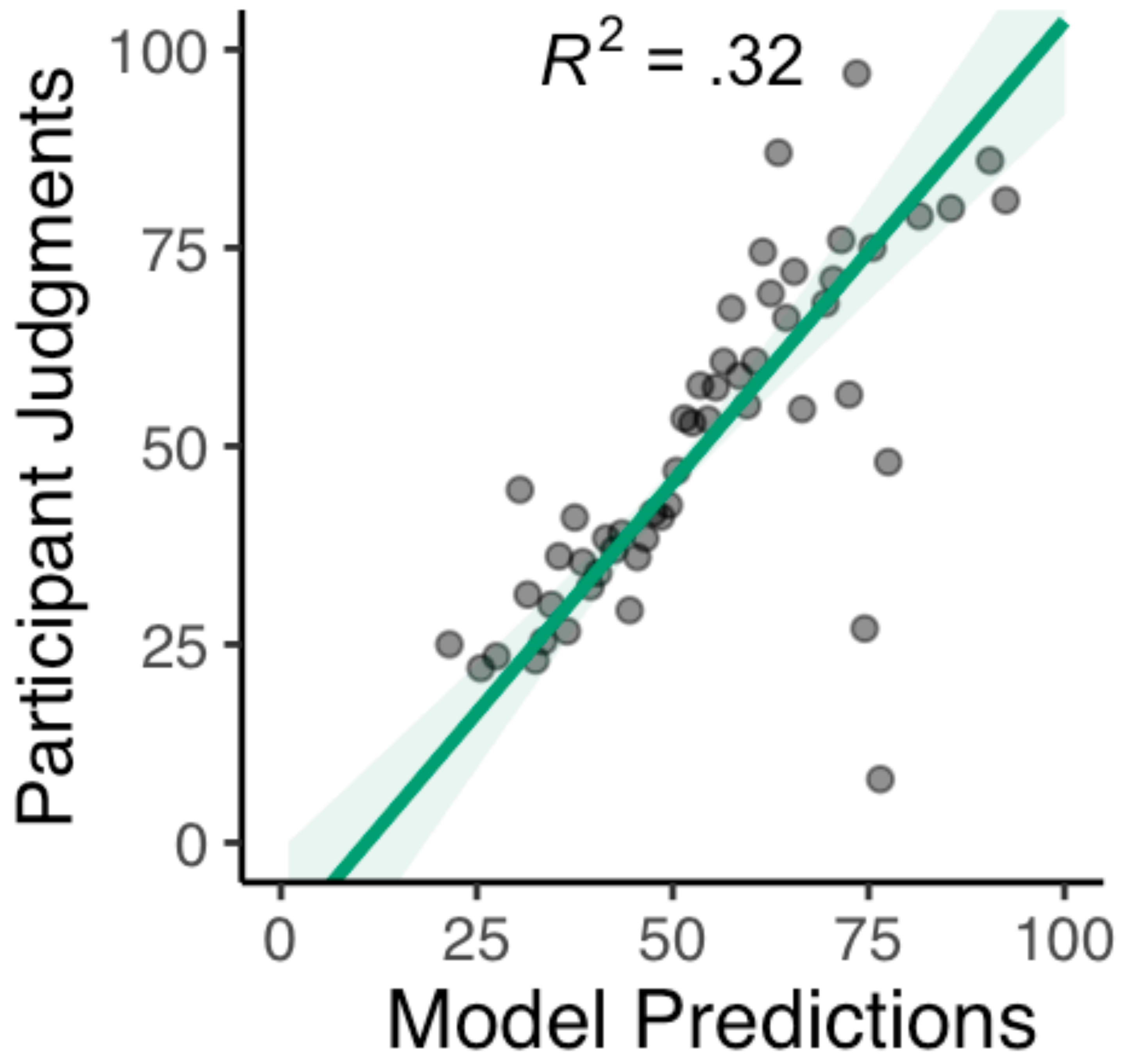


Validation on judgments

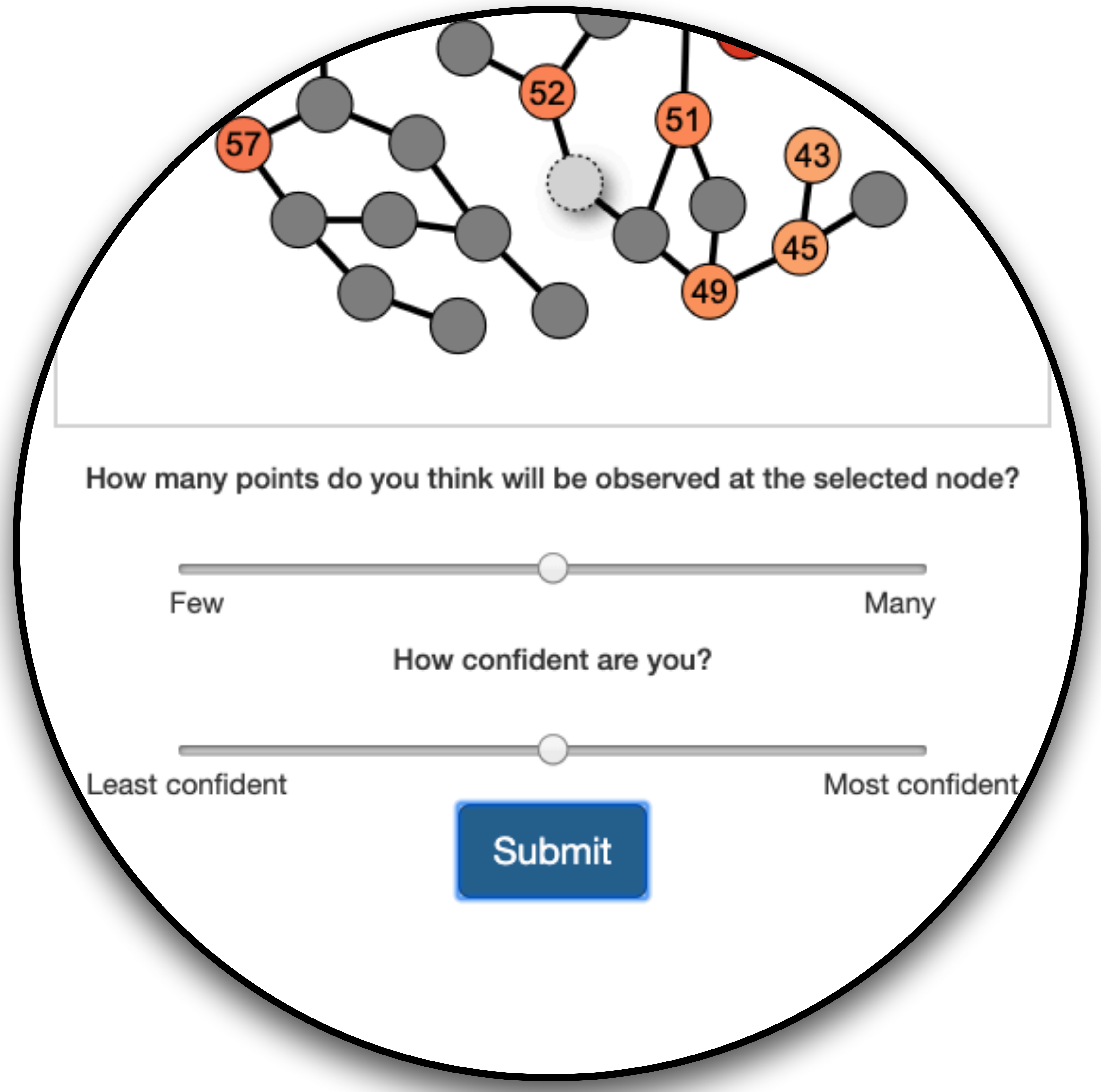


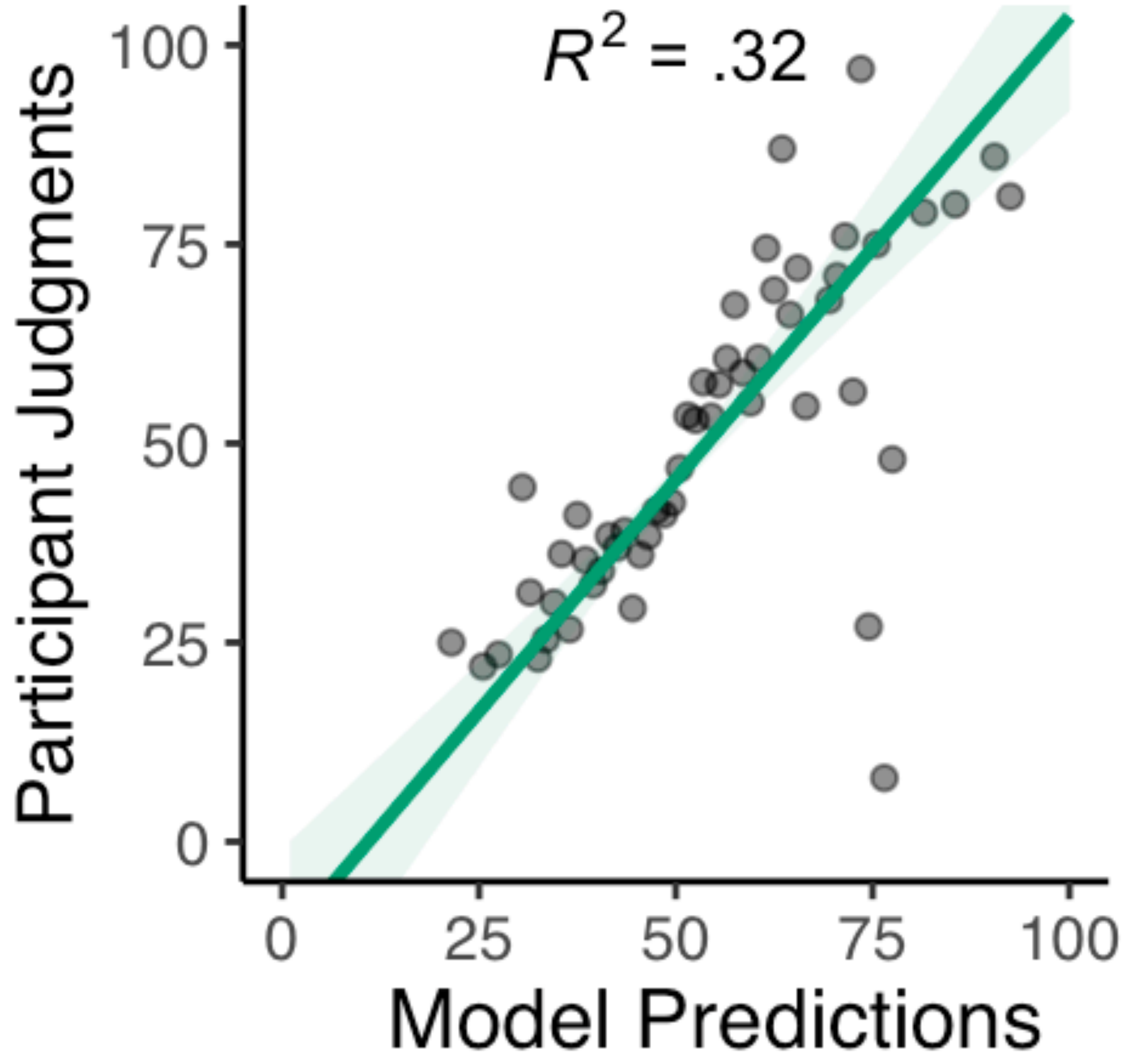
Validation on judgments

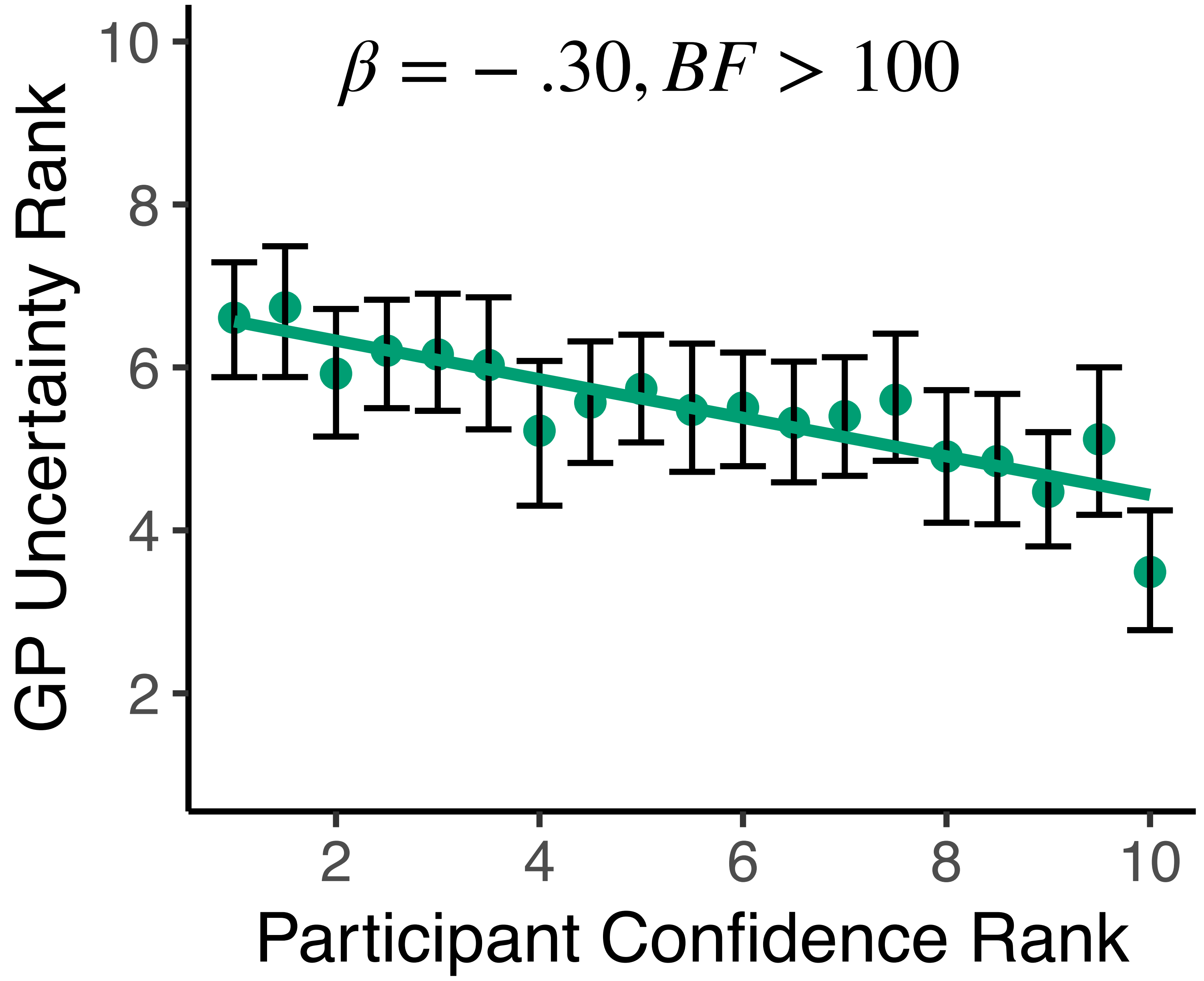




Validation on judgments







Conclusions

- How do we navigate vast problem spaces?
  - Generalization and directed exploration provide a powerful model for efficient learning across many domains
- By modeling generalization as functional inference, we can:
  - Predict search decisions
  - Simulate human-like performance
  - Predict judgments of expected reward and confidence
- Underlying mechanisms of Bayesian inference, kernel similarity, and episodic RL have deep theoretical connections to other models in Neuroscience and Computer Science

Future directions

- How is the structure of similarity learned?
  - **Successor Representation:** online prediction error about future states? (Dayan, *NeurComp*1993; Stachenfeld et al., *Nat Neuro* 2017)
  - **Tolman Eichenbaum Machine**: associative learning mechanisms (Whittington et al., *Cell* 2020)
  - **Structure induction:** Bayesian inference about hypothesized structure? (Kemp & Tenenbaum, *PNAS* 2008)
- How do humans keep functional inference tractable as we gain more experience?
  - GPs scale cubically with the size of the data $\mathcal{O}(n^3)$
  - A large part of this complexity is in computing the underlying uncertainty
  - Perhaps population codes can support flexible generalization with uncertainty (Tano, Dayan, & Pouget, *NeurIPS* 2020)

Thanks to my collaborators and funders

Eric Schulz
MPI Tübingen

Björn Meder
MPI Berlin, Uni. Potsdam

Azzurra Ruggeri
MPI Berlin

Nico Schuck
MPI Berlin

Sam Gershman
Harvard

Simon Ciranka
MPI Berlin

Anna Giron
Uni Tübingen

Mona Garvert
MPI Leipzig

Maarten Speekenbrink
UCL

Jonathan Nelson
MPI Berlin, Uni. Surrey

Wouter van den Bos
Uni Amsterdam

ML

EBERHARD KARLS UNIVERSITÄT TÜBINGEN

Robust Learning

TÜBINGEN AI CENTER
BMBF Competence Center for Machine Learning

CyberValley

Federal Ministry of Education and Research

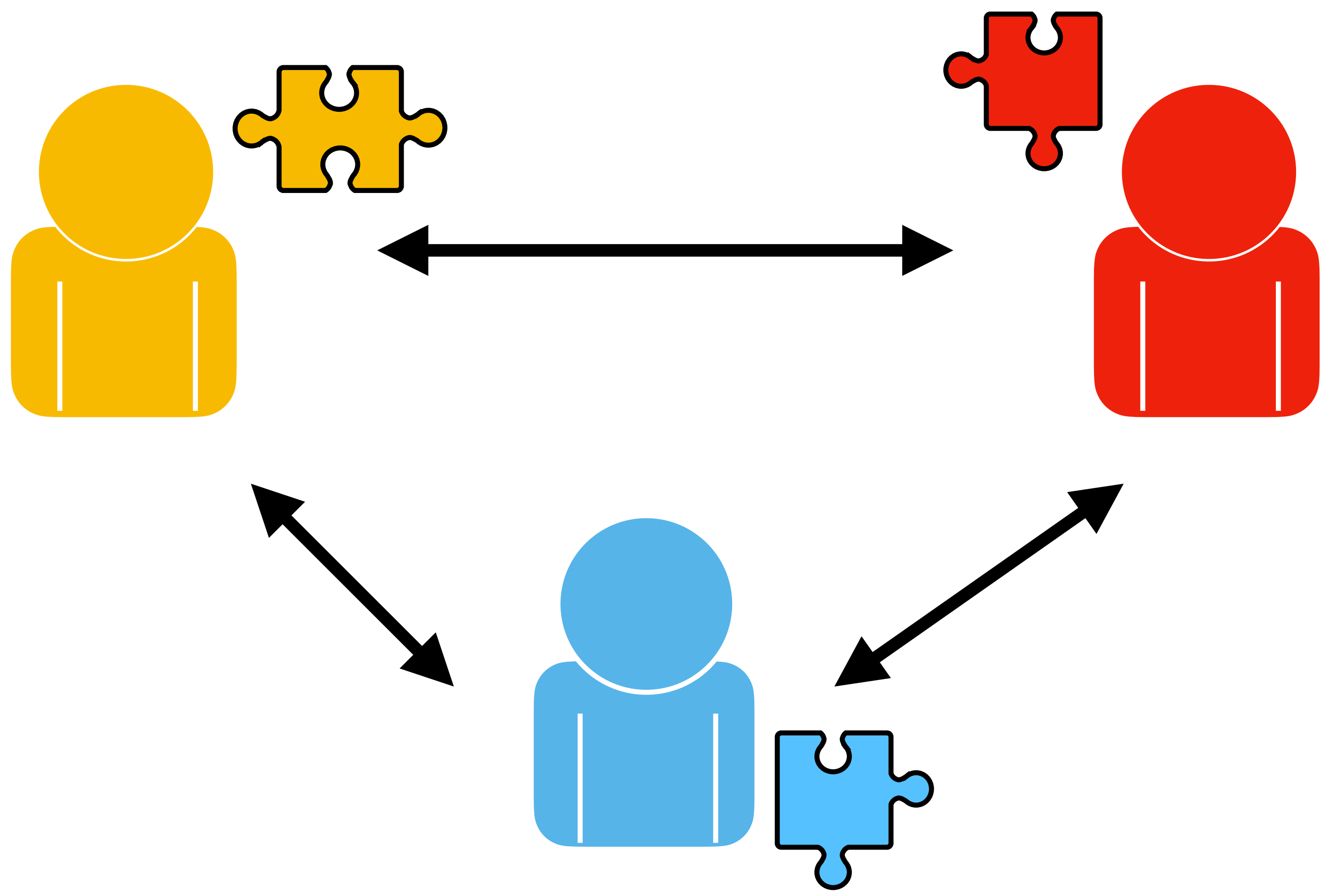
ADVANCING MACHINE INTELLIGENCE WITH ROBUST MACHINE LEARNING

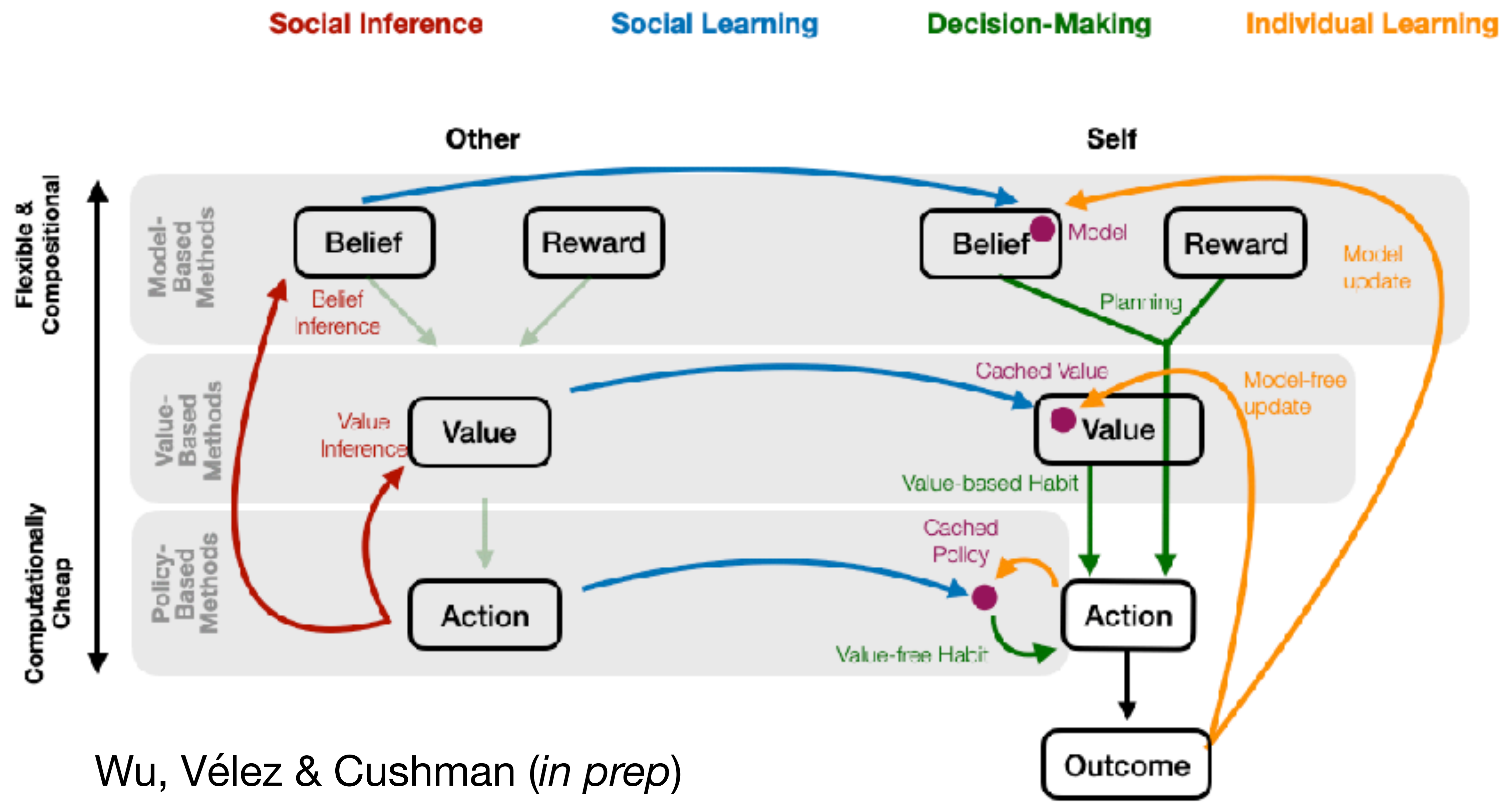
Fuly Funded PhD position: Computational mechanisms of Social Learning

Potential topics include:

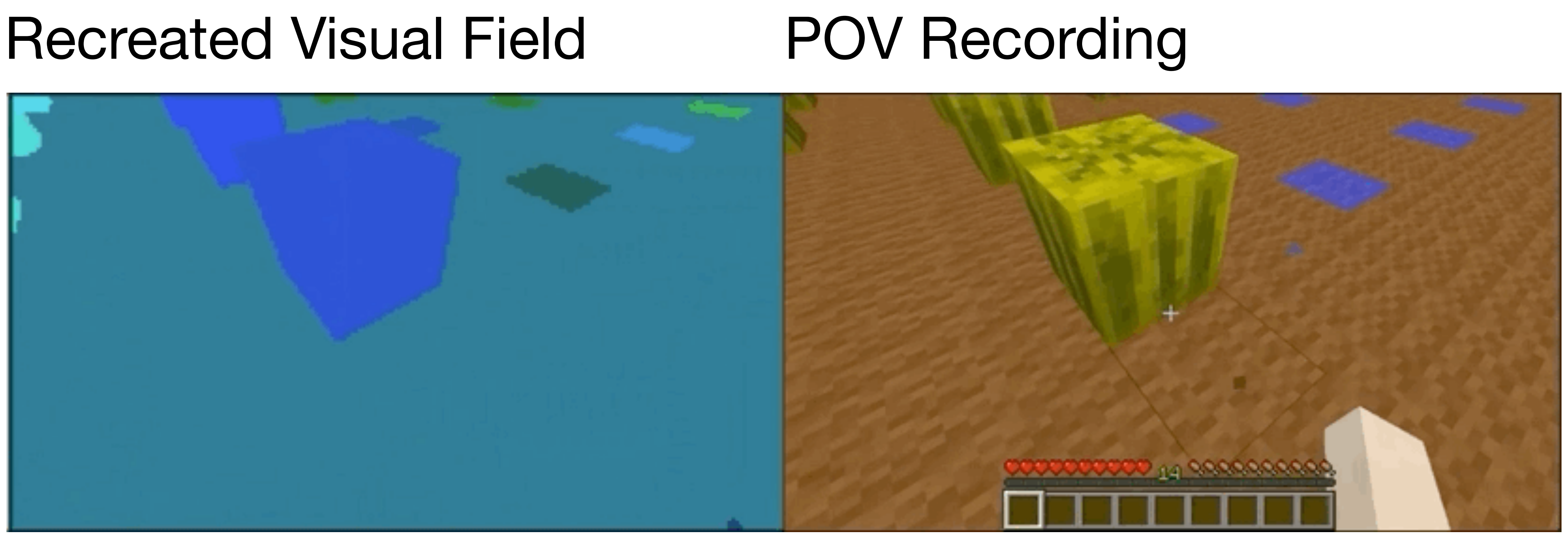
- Theory of Mind inference using inverse RL
- Selectivity and specialization in social learning using VR experiments programmed in minecraft
- Cumulative cultural evolution
- Integration of individual and social information

Visit hmc-lab.com for more info! Apply by **May 15th**





Wu, Vélez & Cushman (*in prep*)



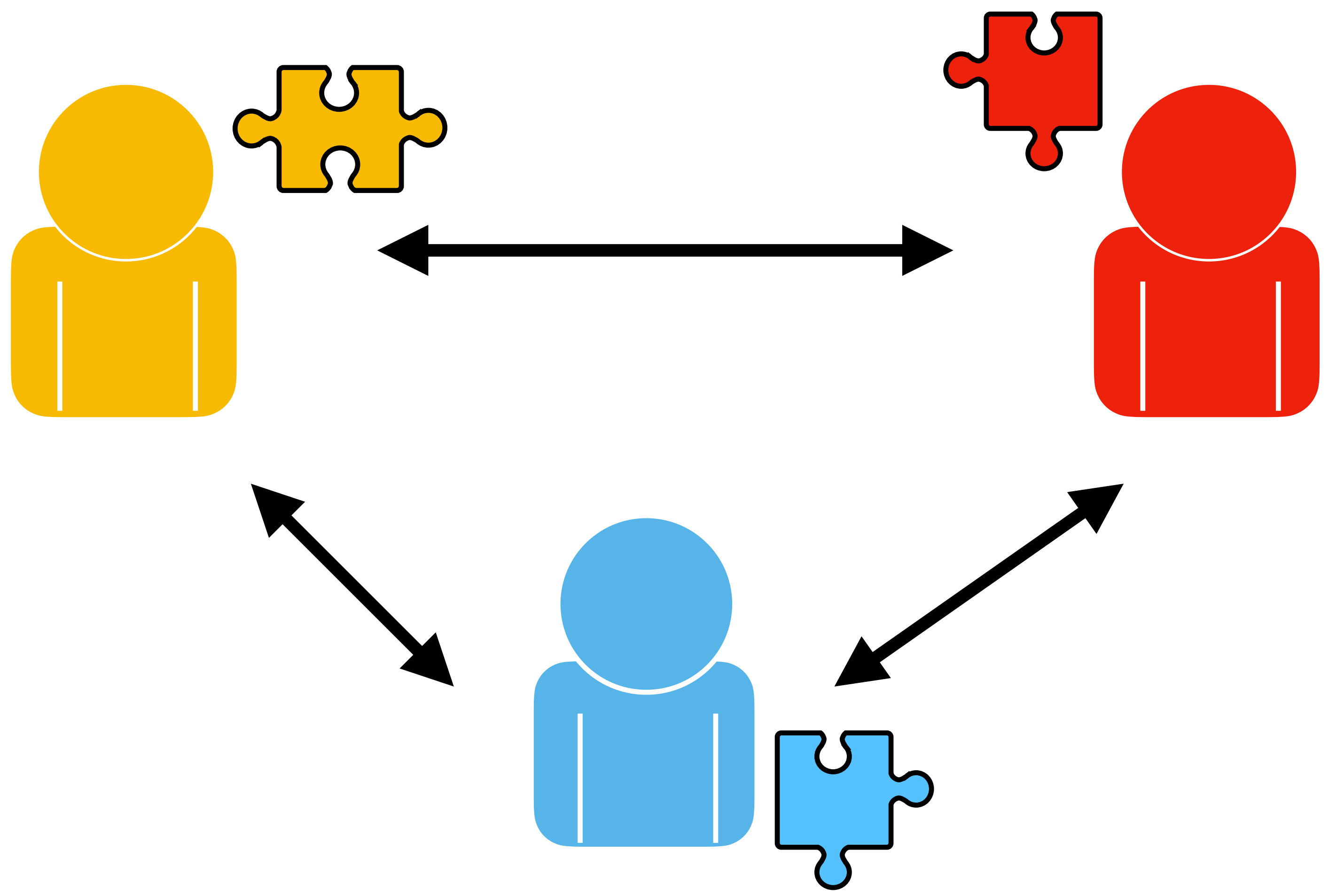
Wu et al,. (*biorxiv* 2021)

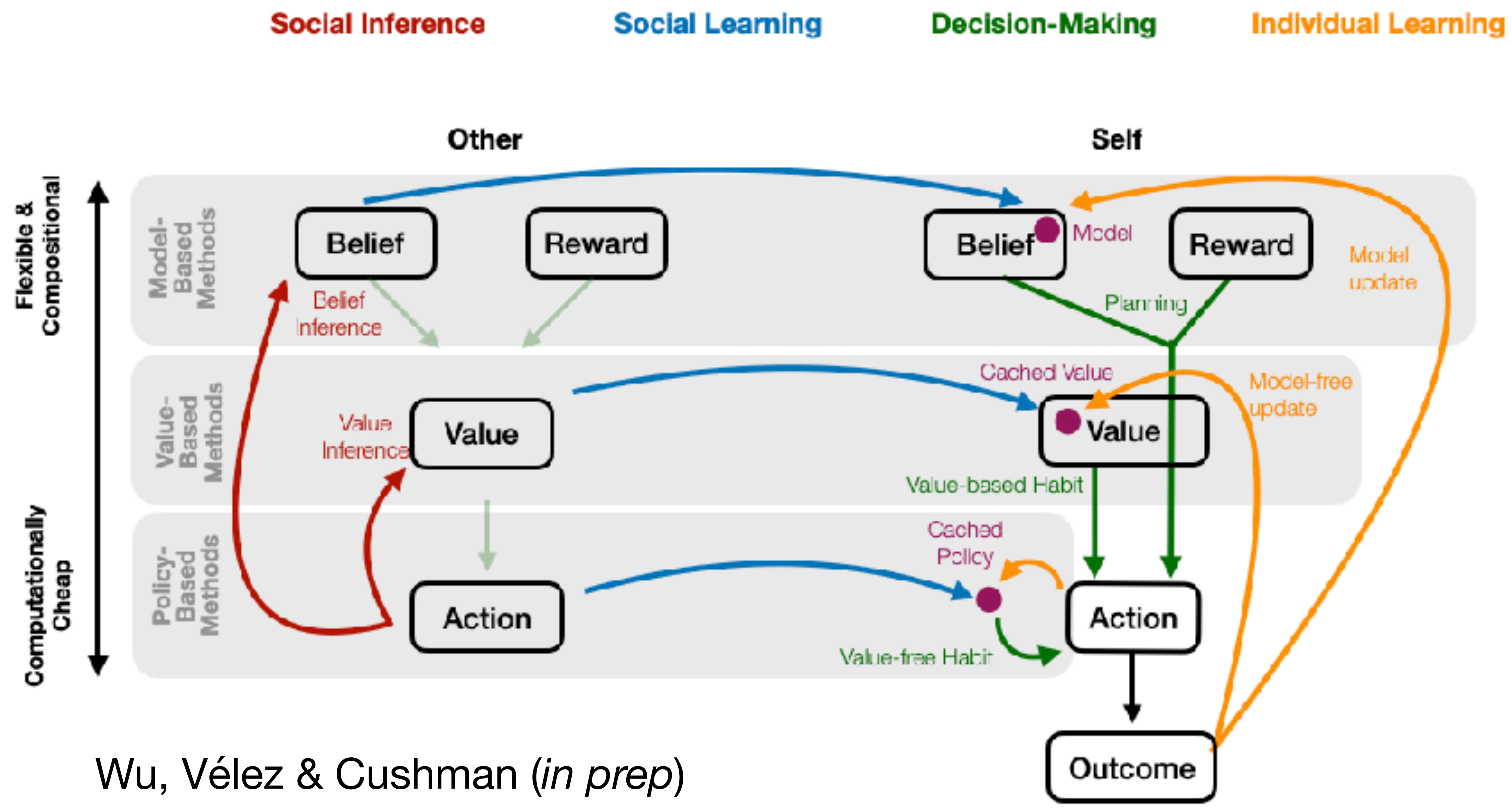
Fuly Funded PhD position: Computational mechanisms of Social Learning

Potential topics include:

- Theory of Mind inference using inverse RL
- Selectivity and specialization in social learning using VR experiments programmed in minecraft
- Cumulative cultural evolution
- Integration of individual and social information

Visit hmc-lab.com for more info! Apply by **May 15th**





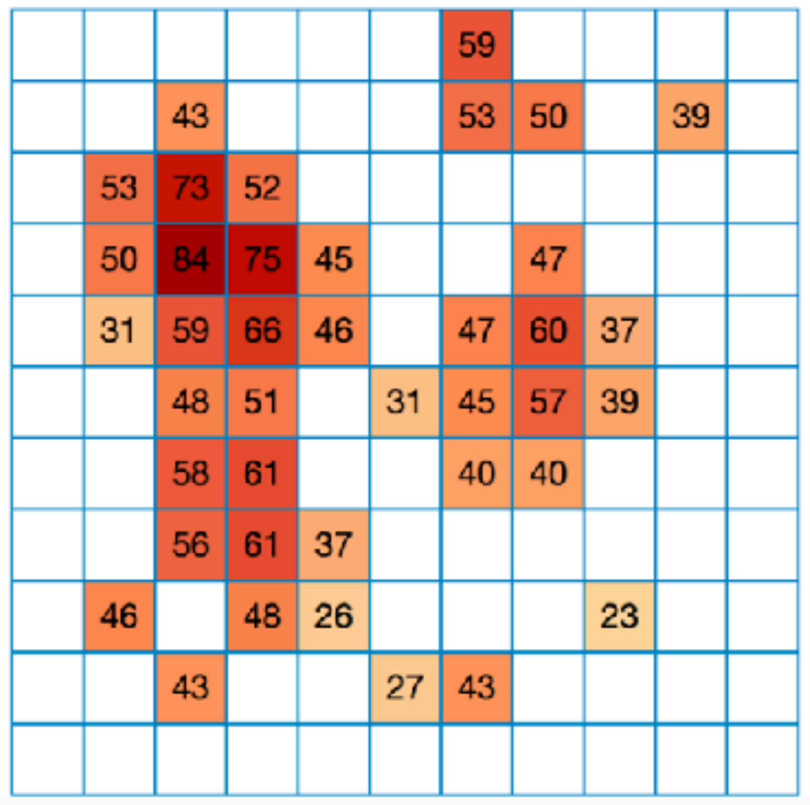
Wu, Vélez & Cushman (*in prep*)

Recreated Visual Field POV Recording

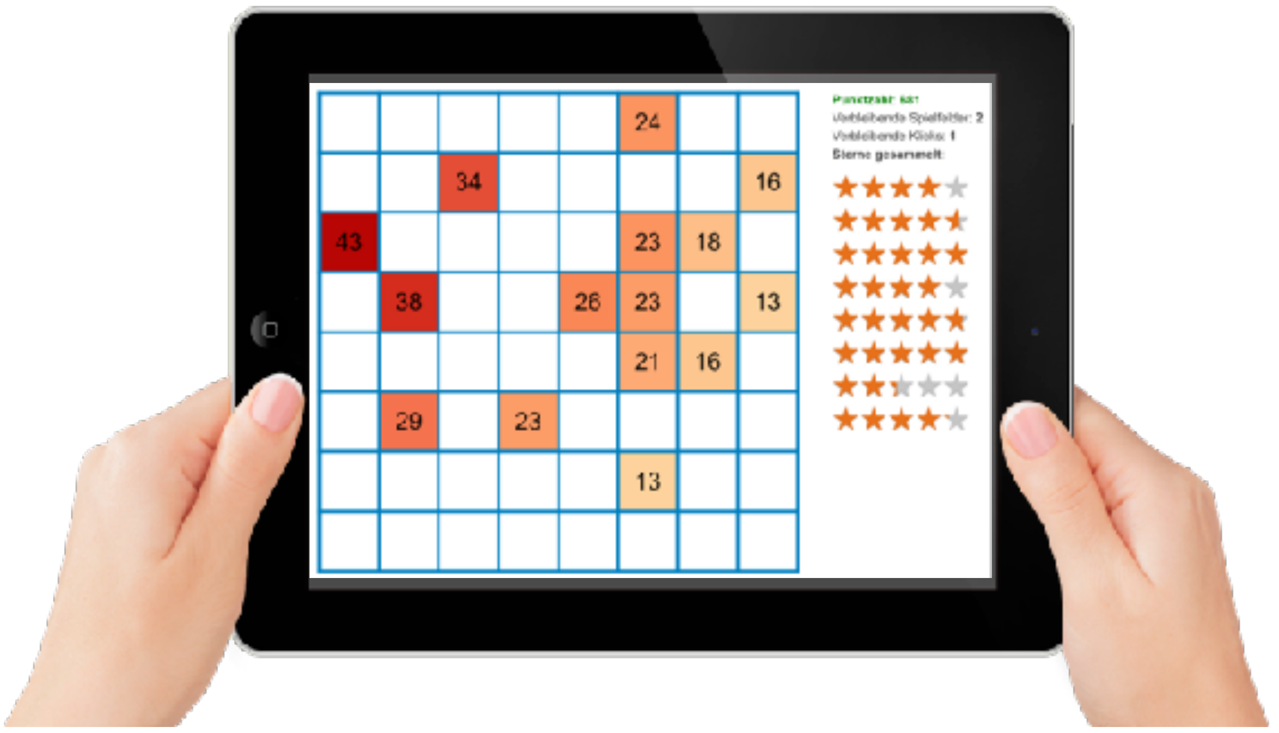
Wu et al,. (*biorxiv* 2021)

Questions?

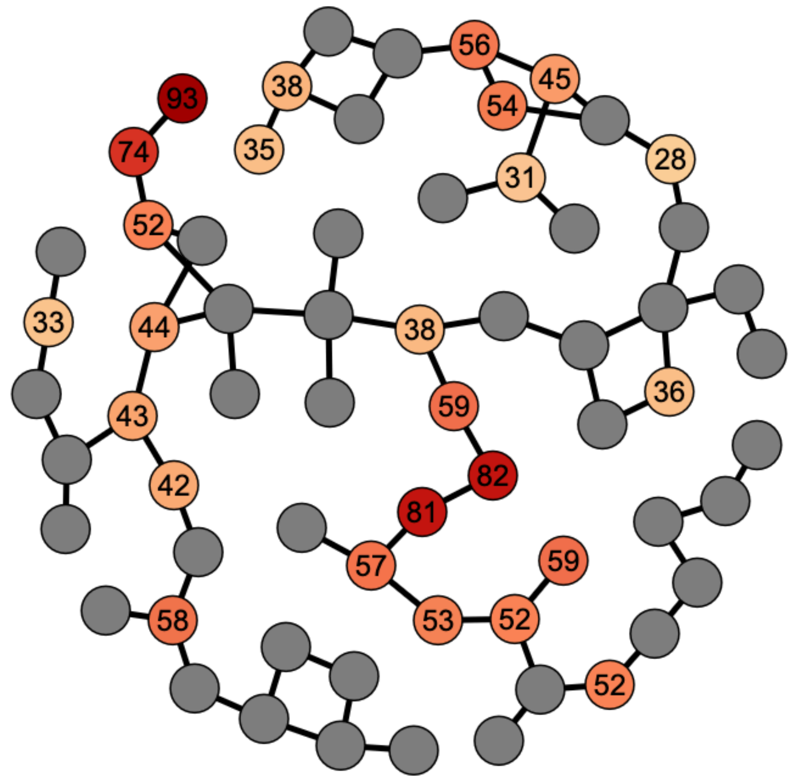
Grid Search



Learning Like a Child

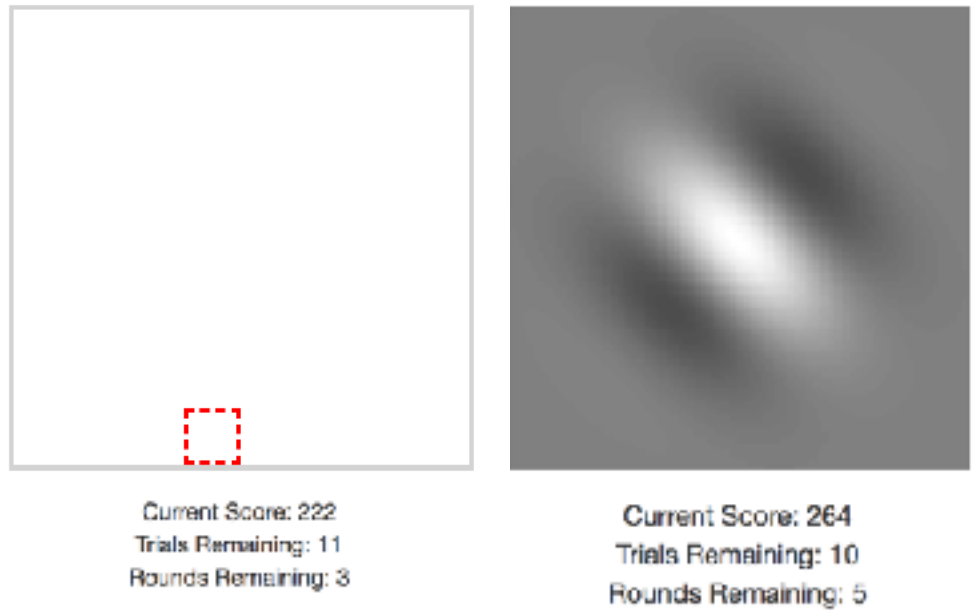


Graph Search



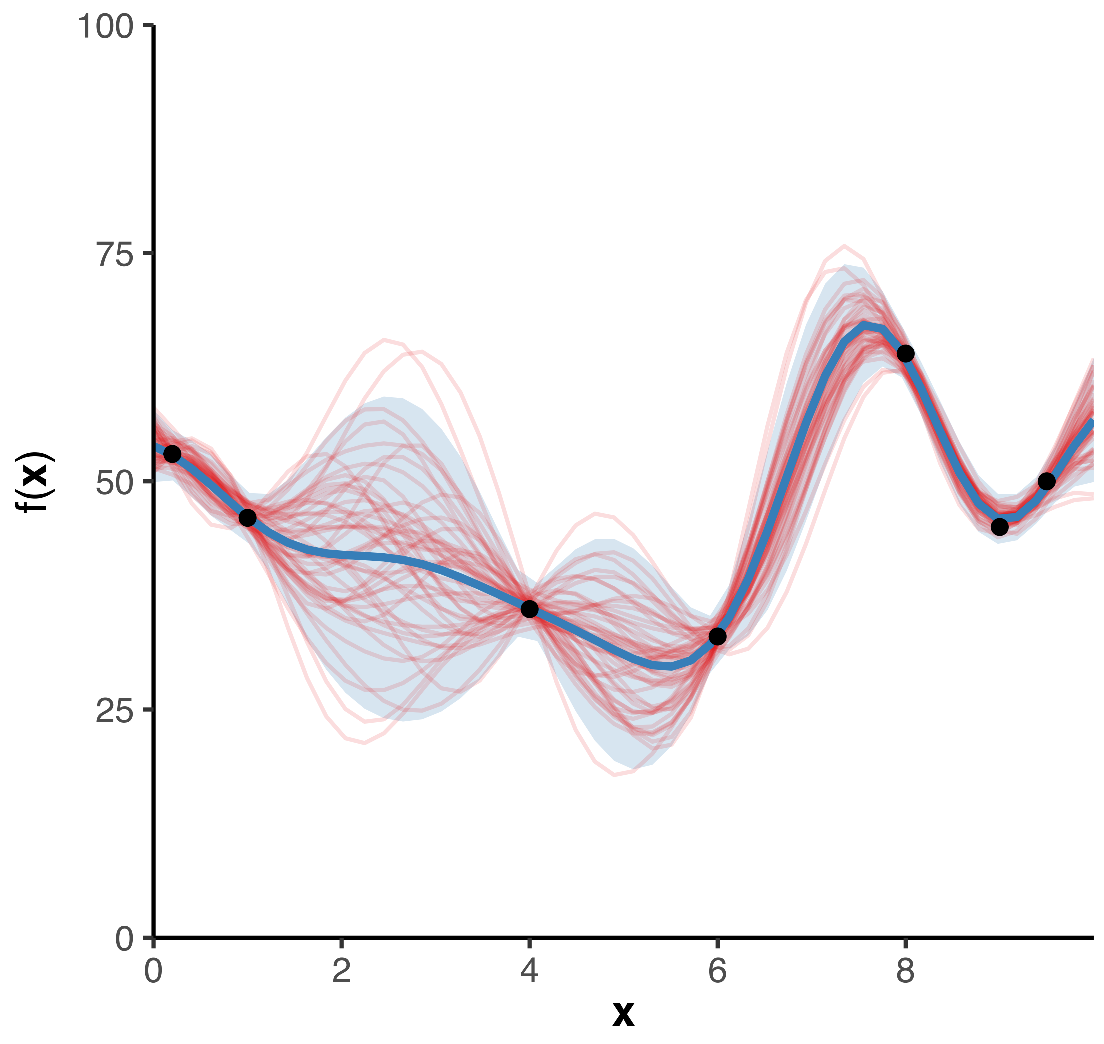
Conceptual Search

Spatial Conceptual



Safe Search

Gaussian Processes



Generalization