General Principles of Human and Machine Learning

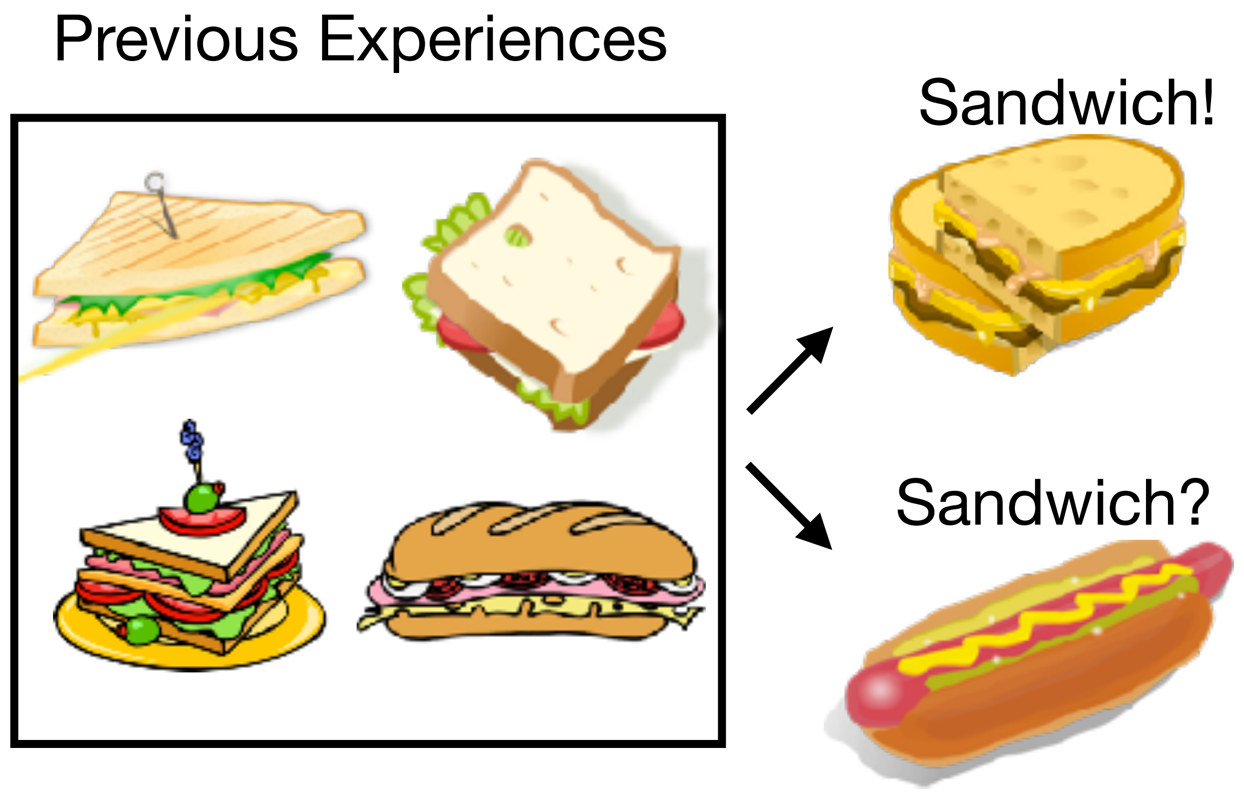
Lecture 8: Function Learning

Dr. Charley Wu

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

The story so far …

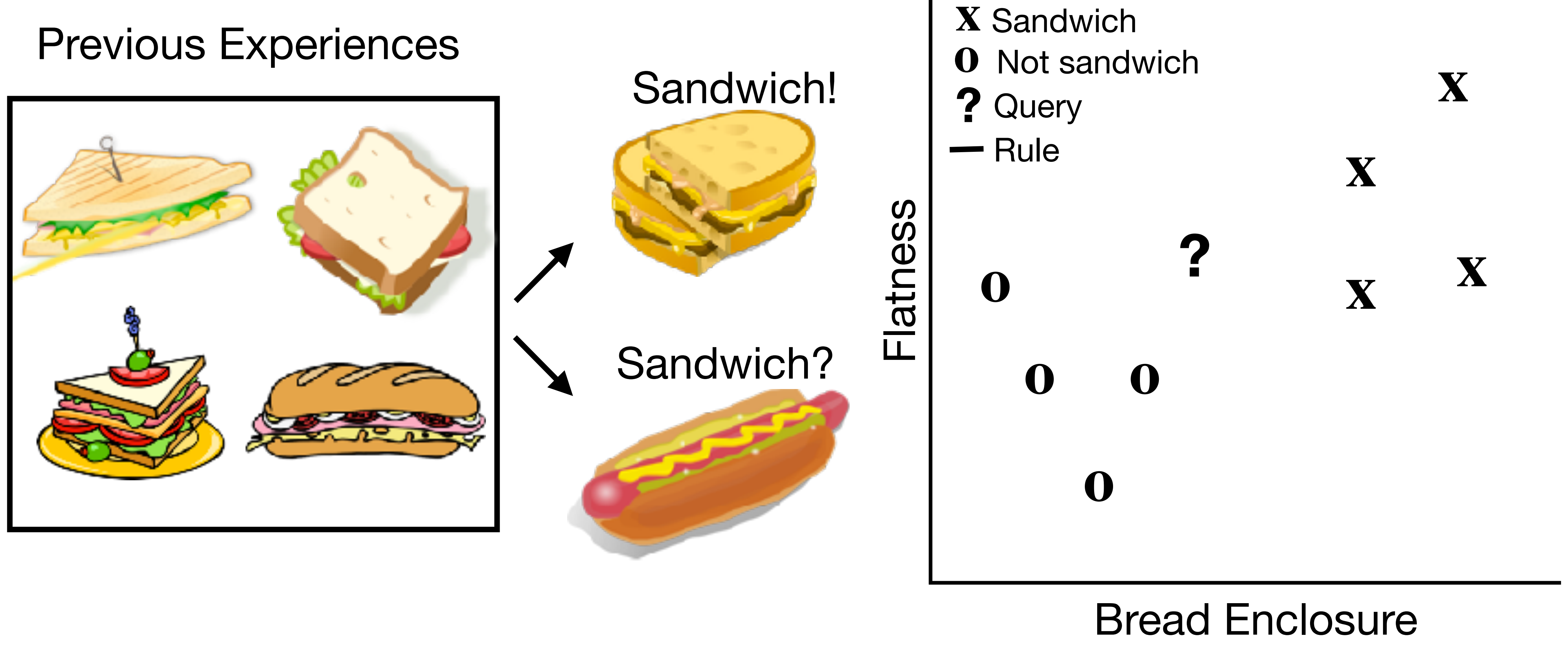
Concept learning as classification



Wu, Meder & Schulz (*AnnRevPsych* forthcoming)

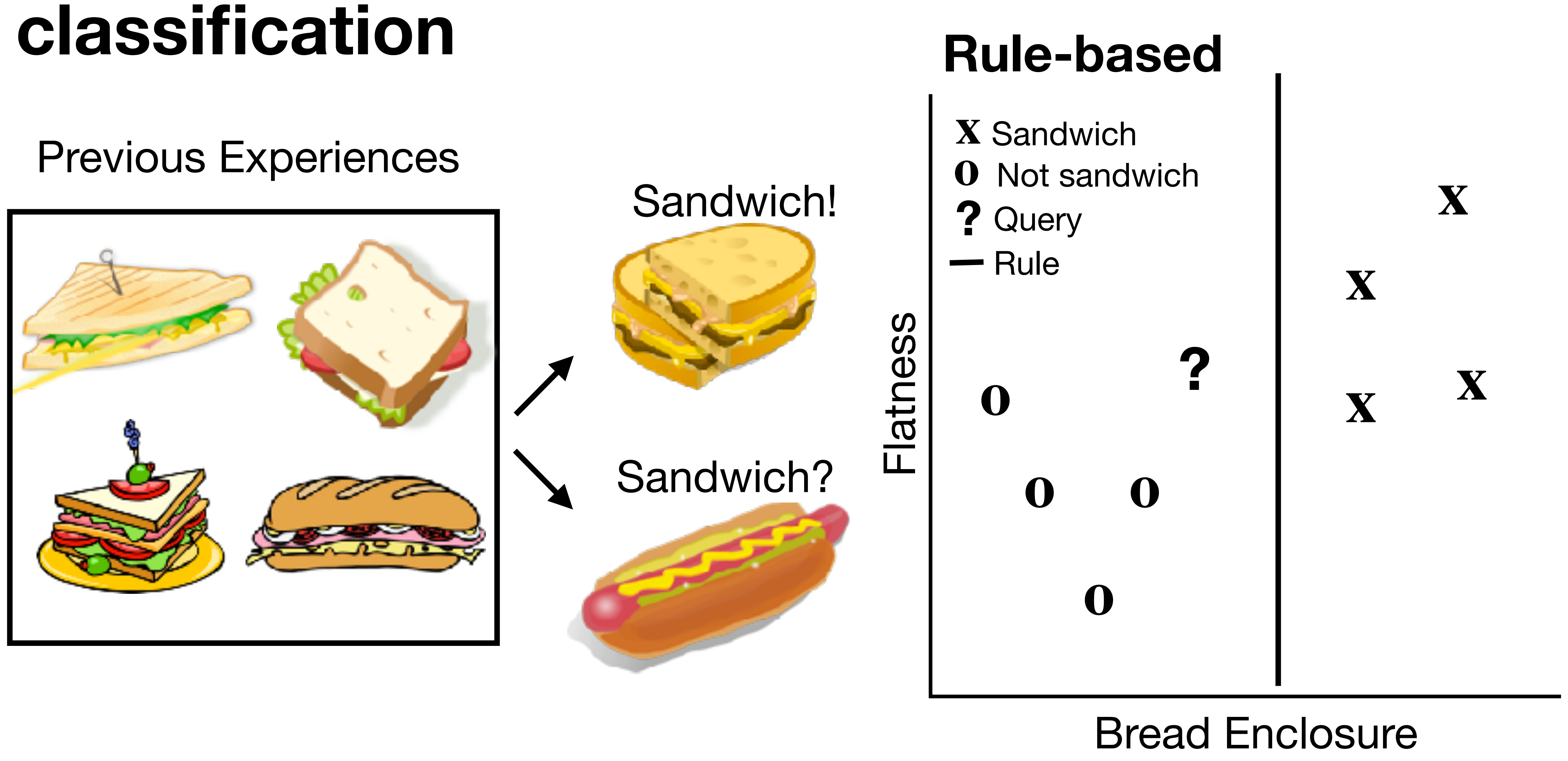
The story so far …

Concept learning as classification



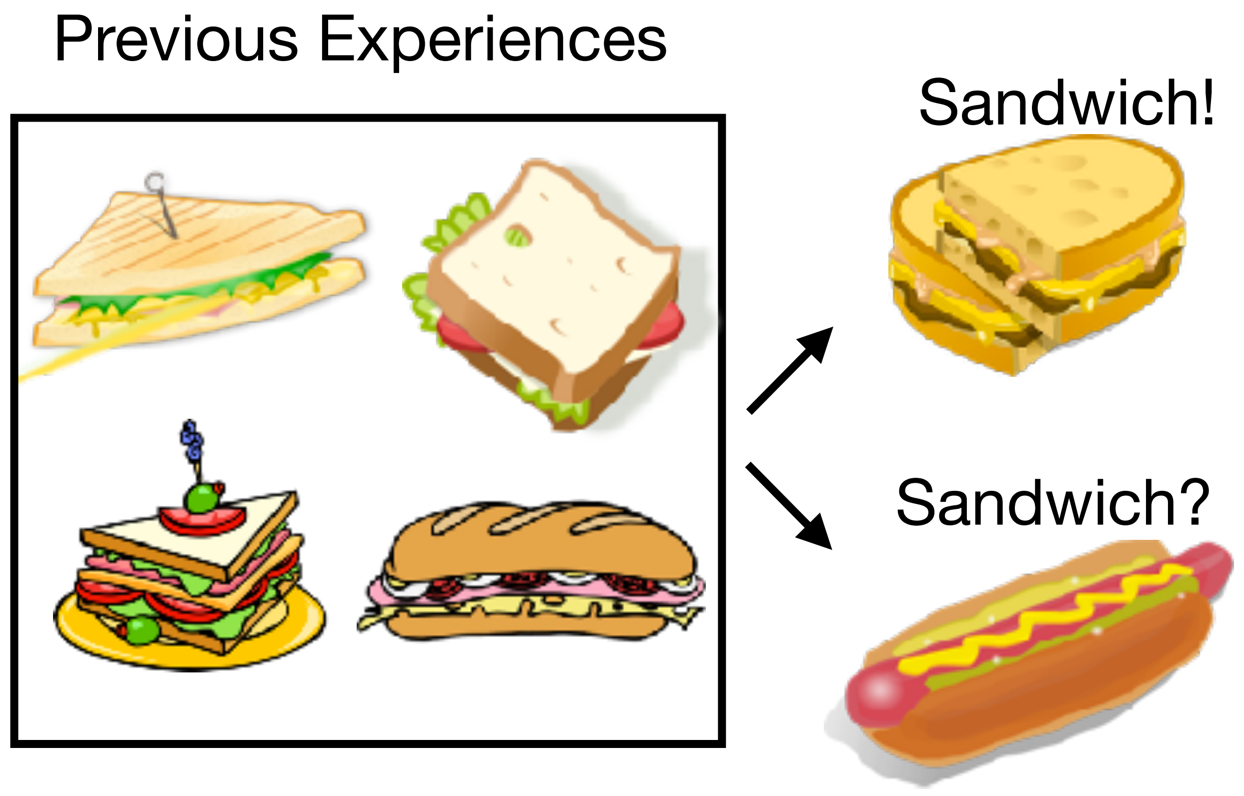
The story so far …

Concept learning as classification

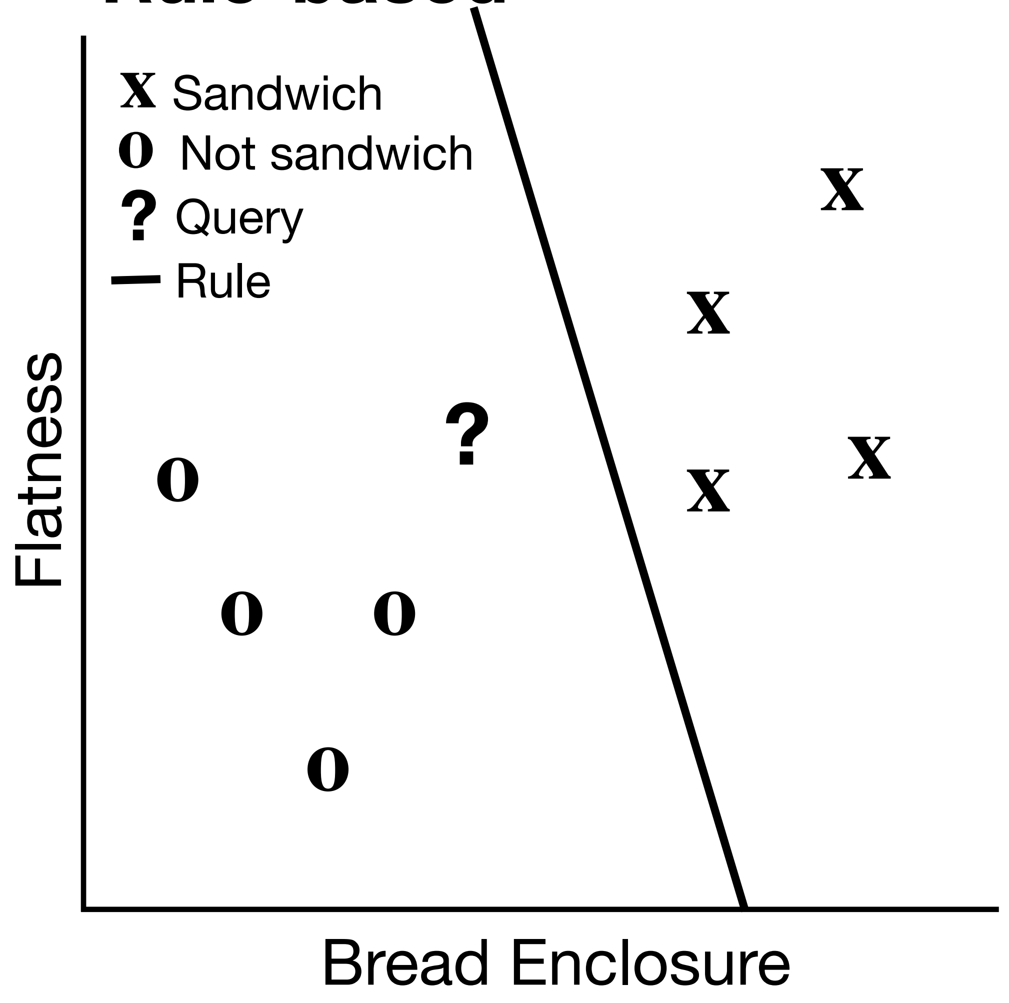


The story so far …

Concept learning as classification

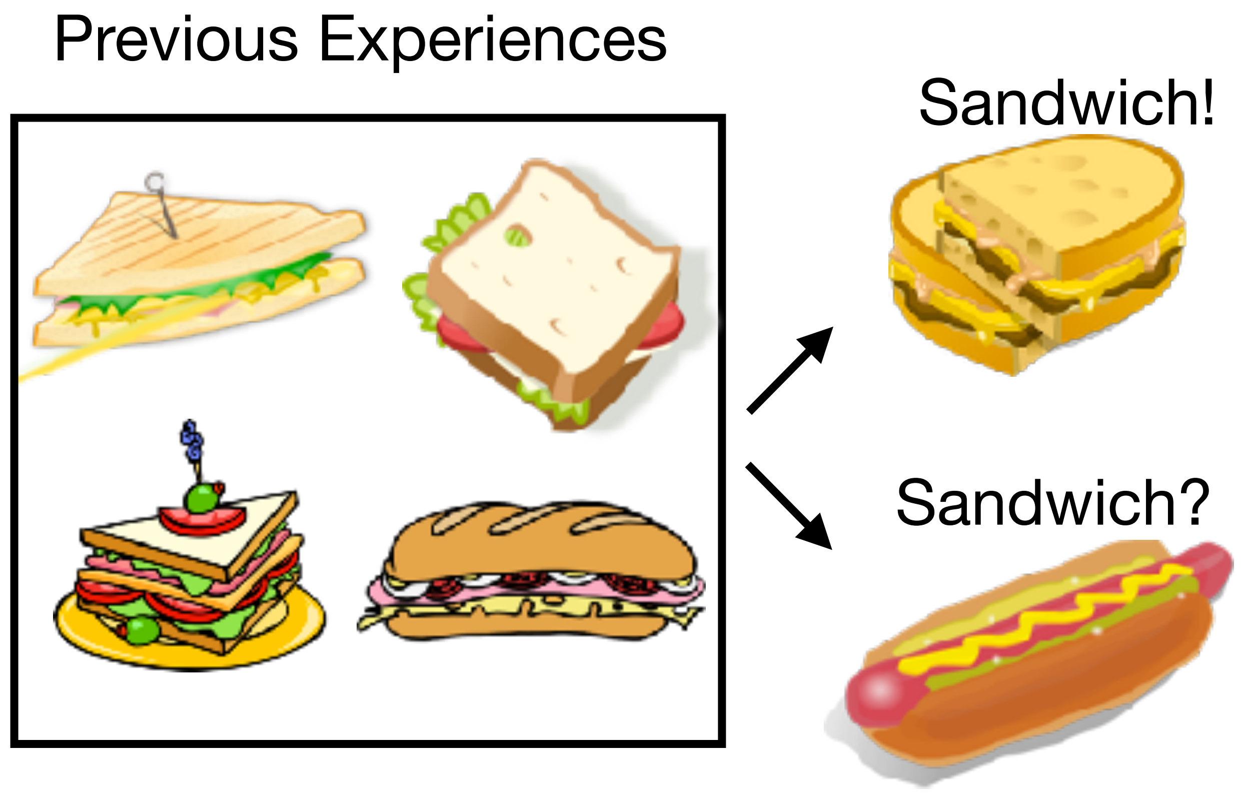


Rule-based

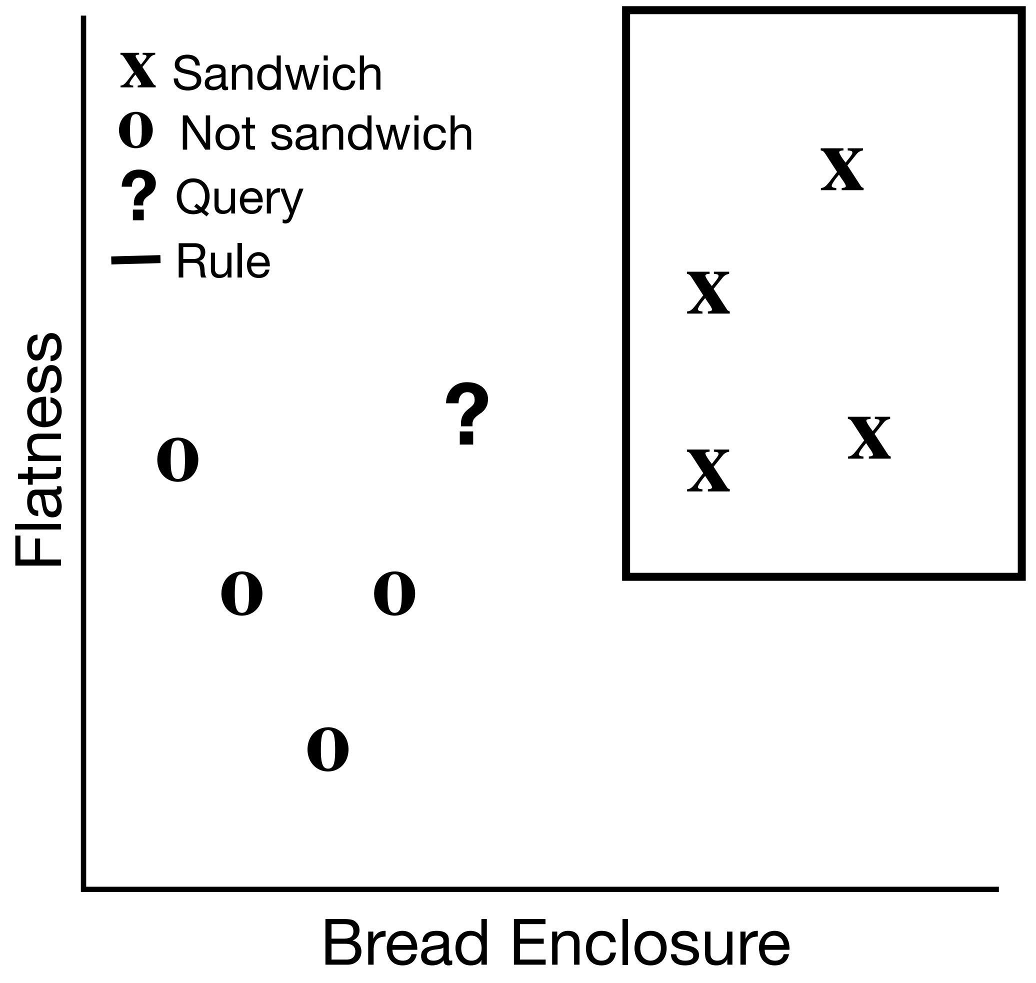


The story so far …

Concept learning as classification



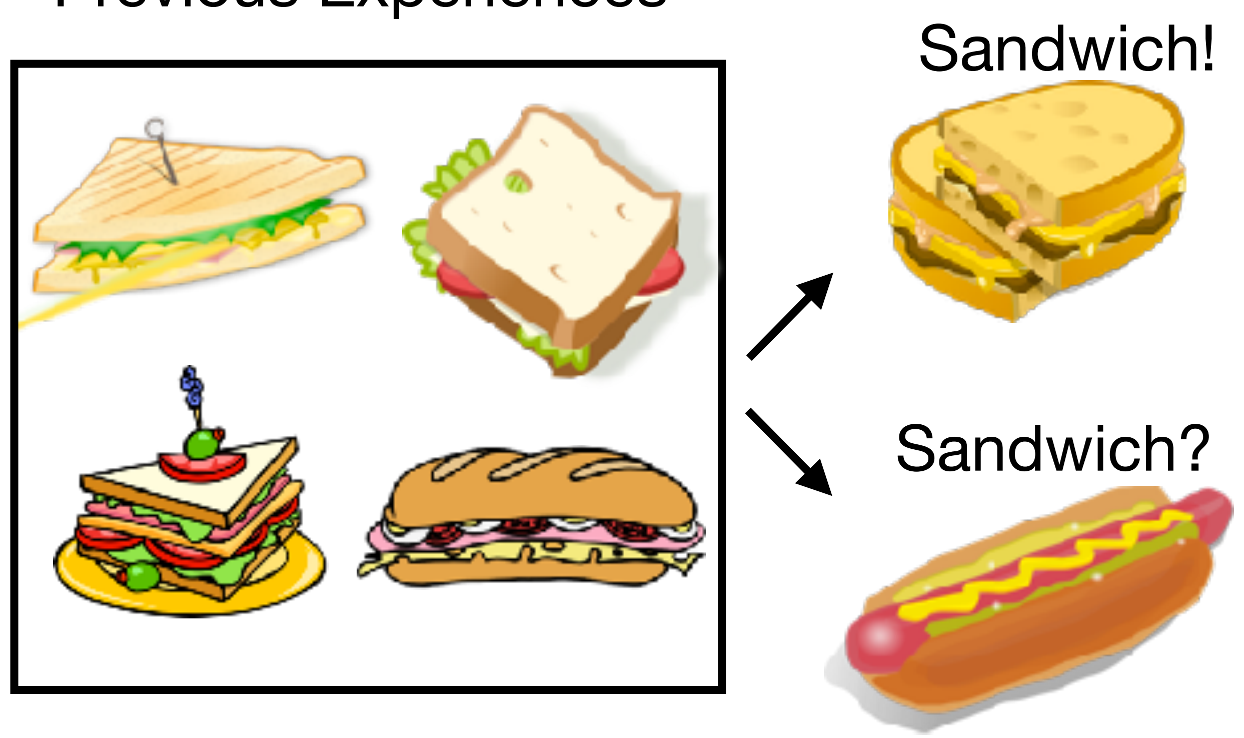
Rule-based



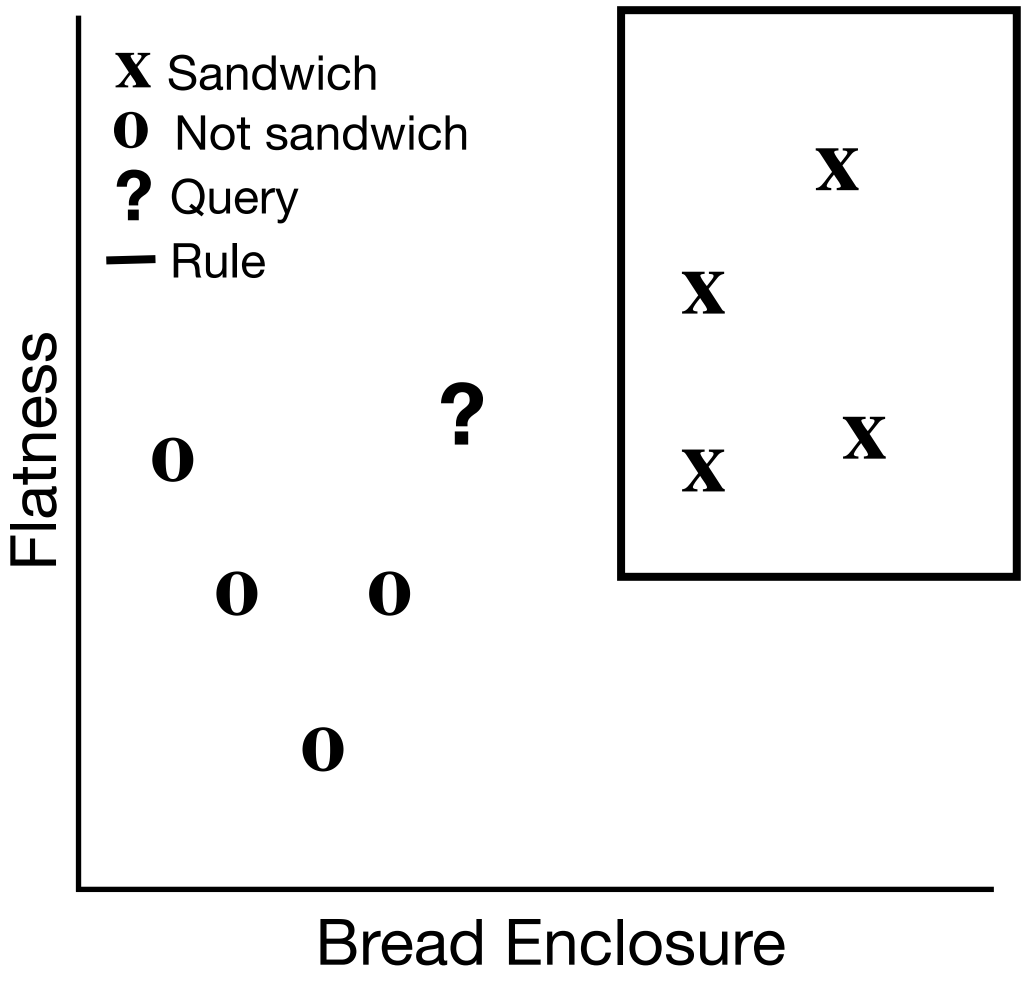
The story so far …

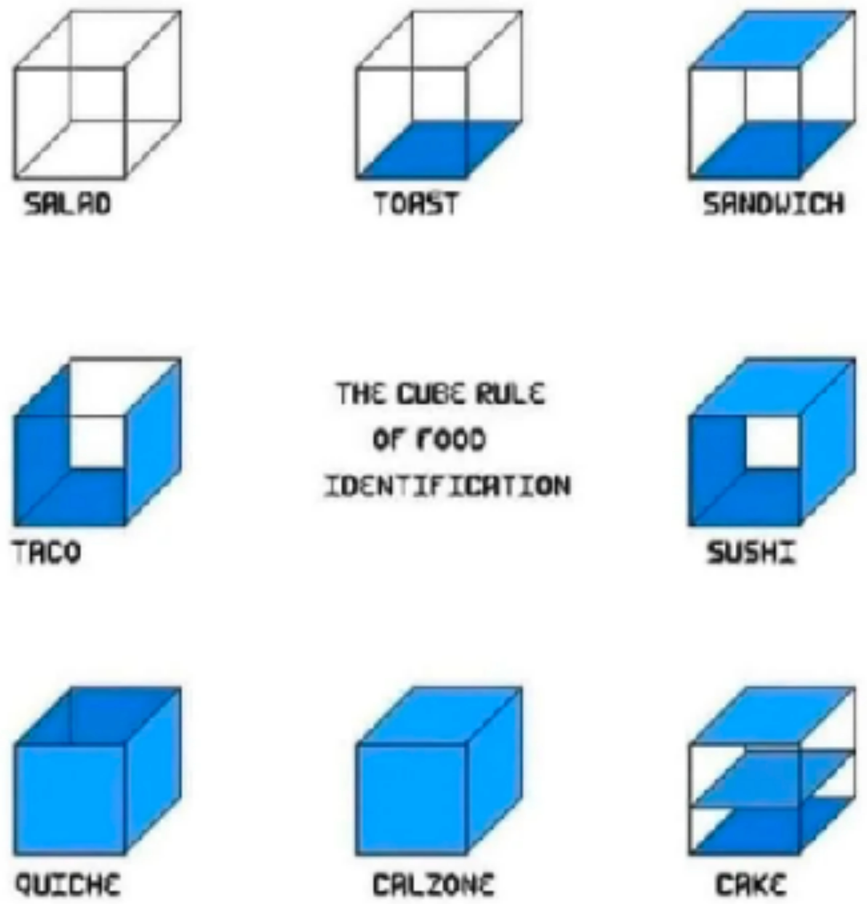
Concept learning as classification

Previous Experiences



Rule-based

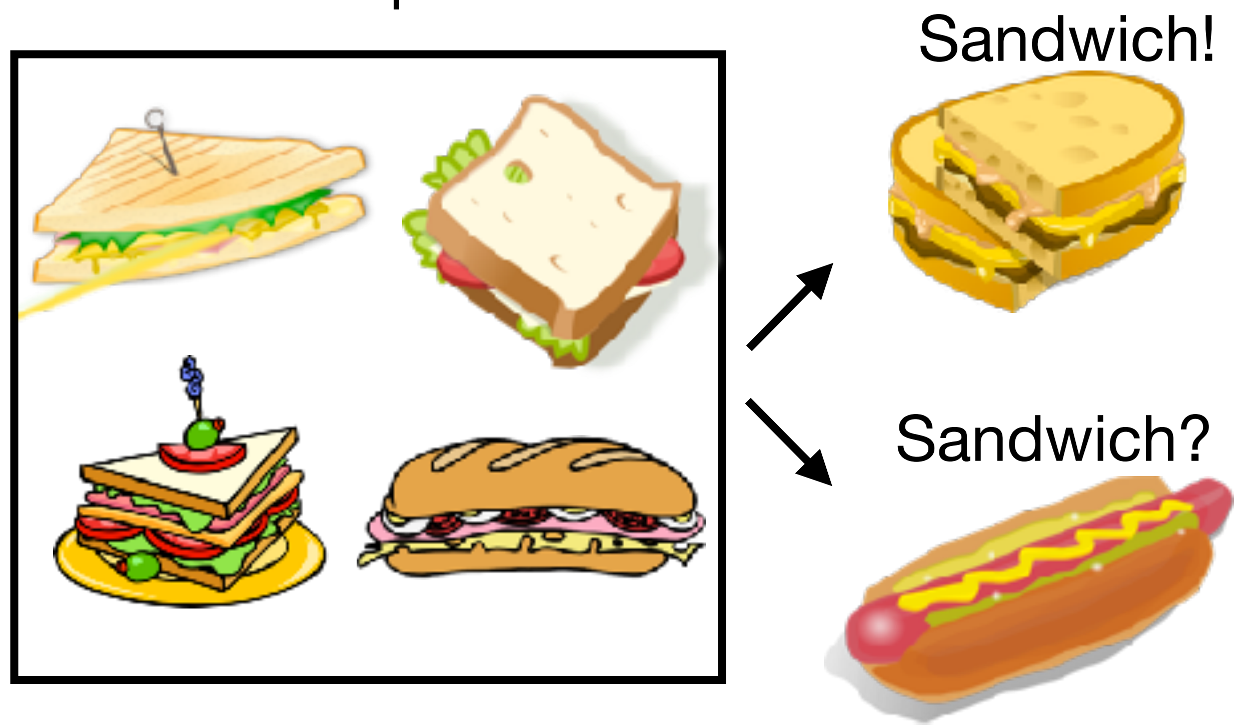




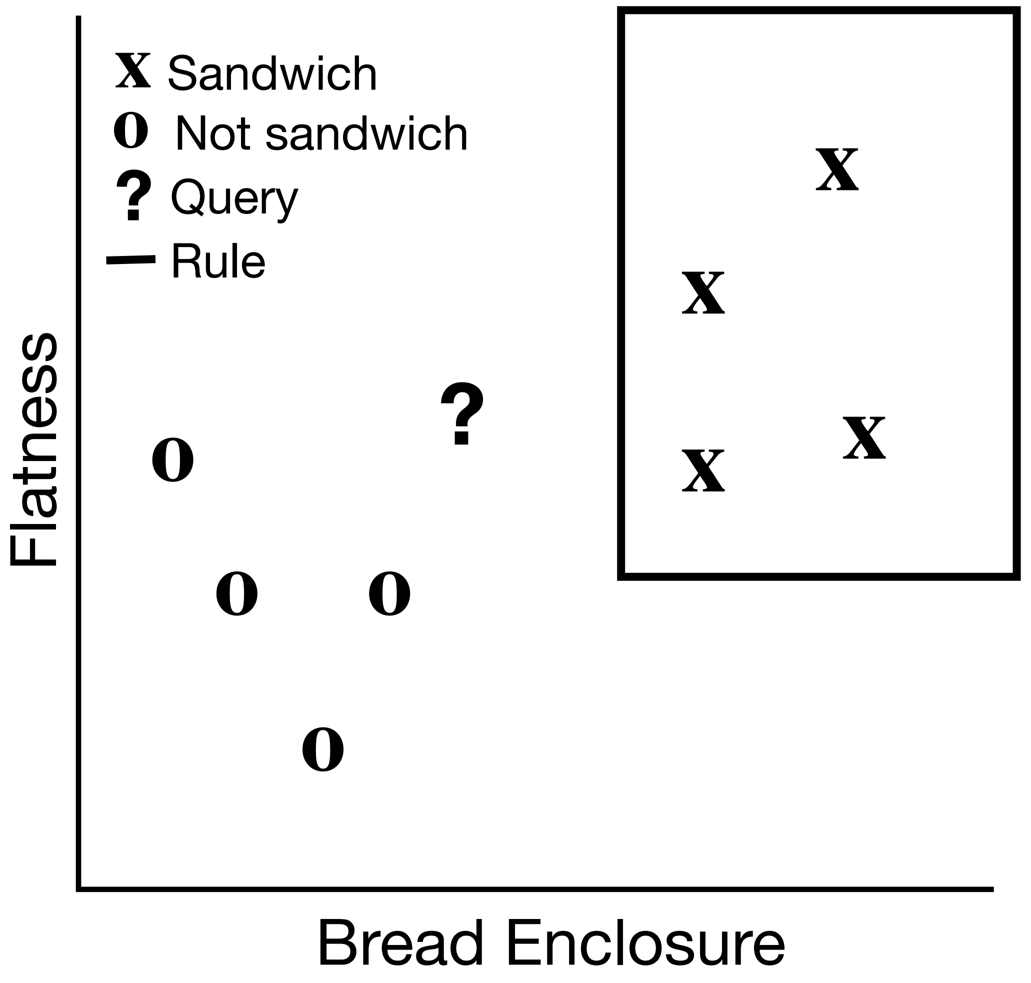
The story so far …

Concept learning as classification

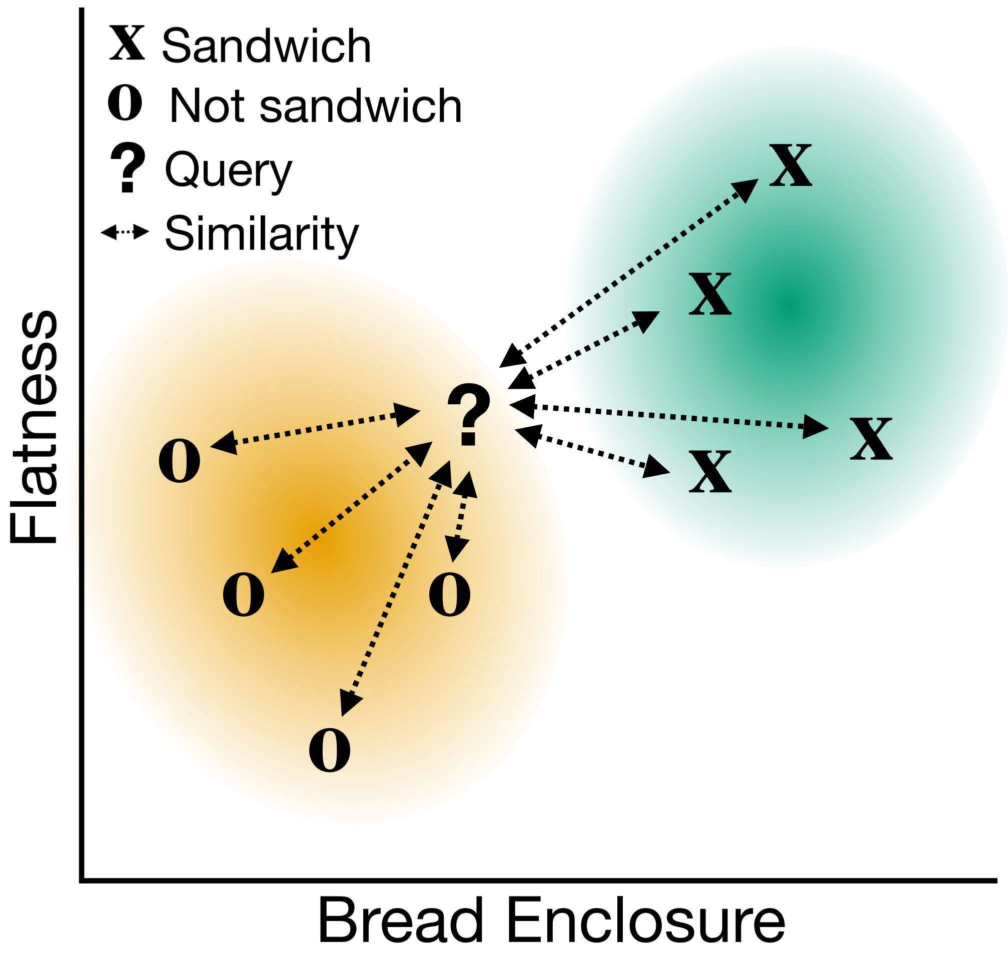
Previous Experiences

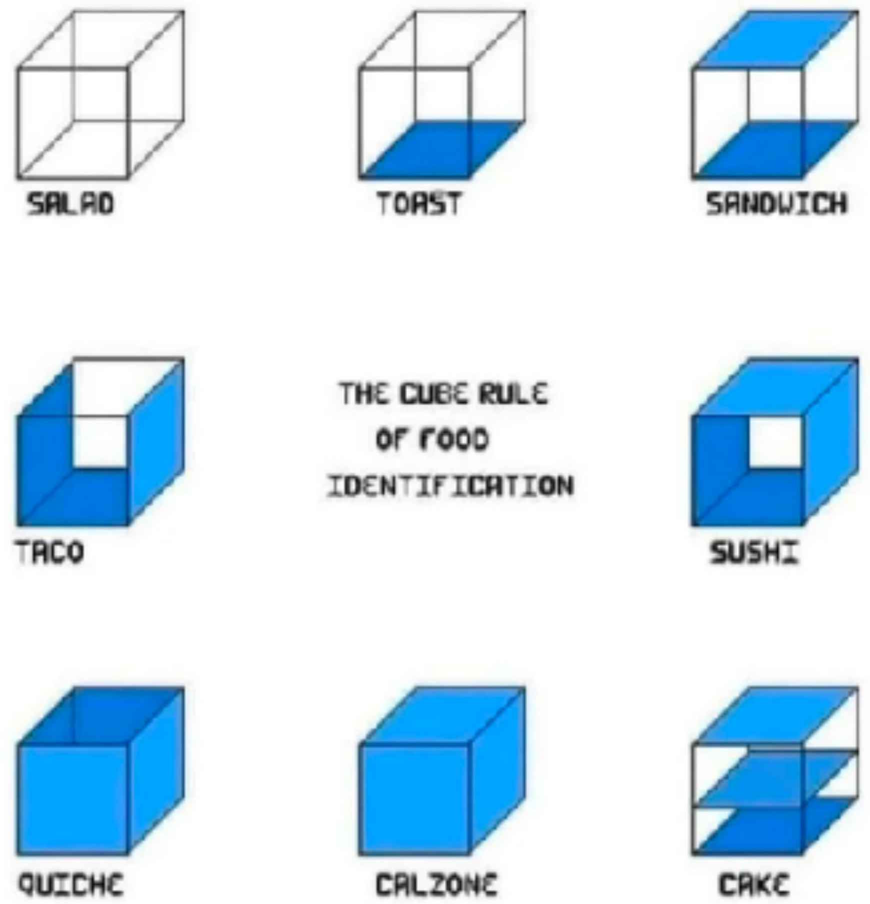


Rule-based



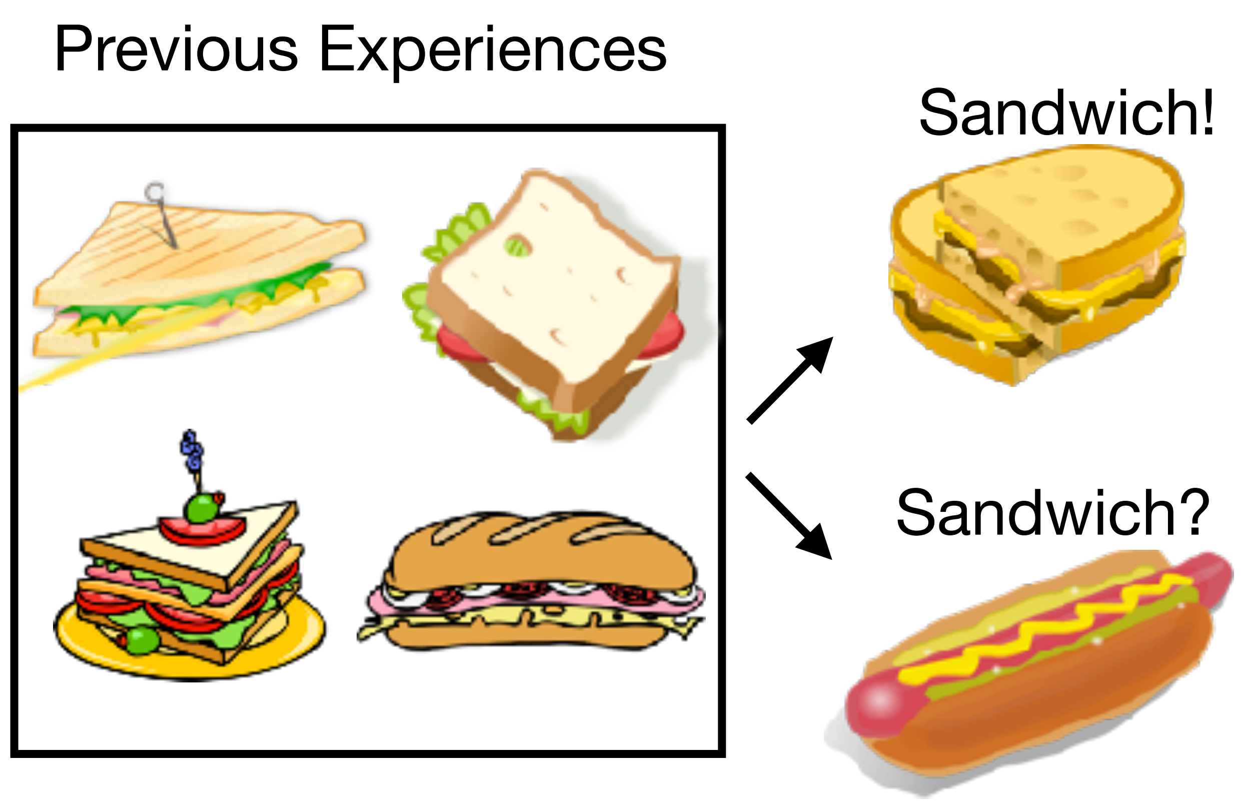
Similarity-based



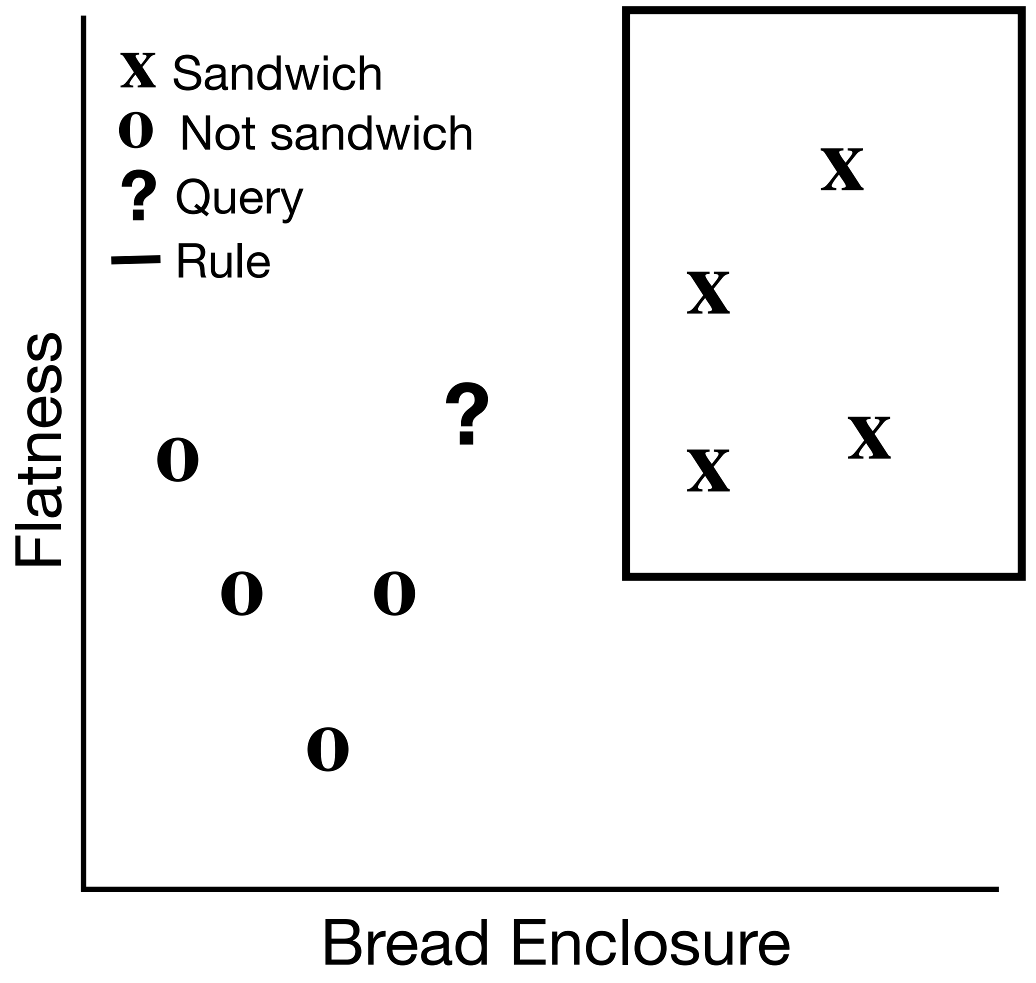


The story so far …

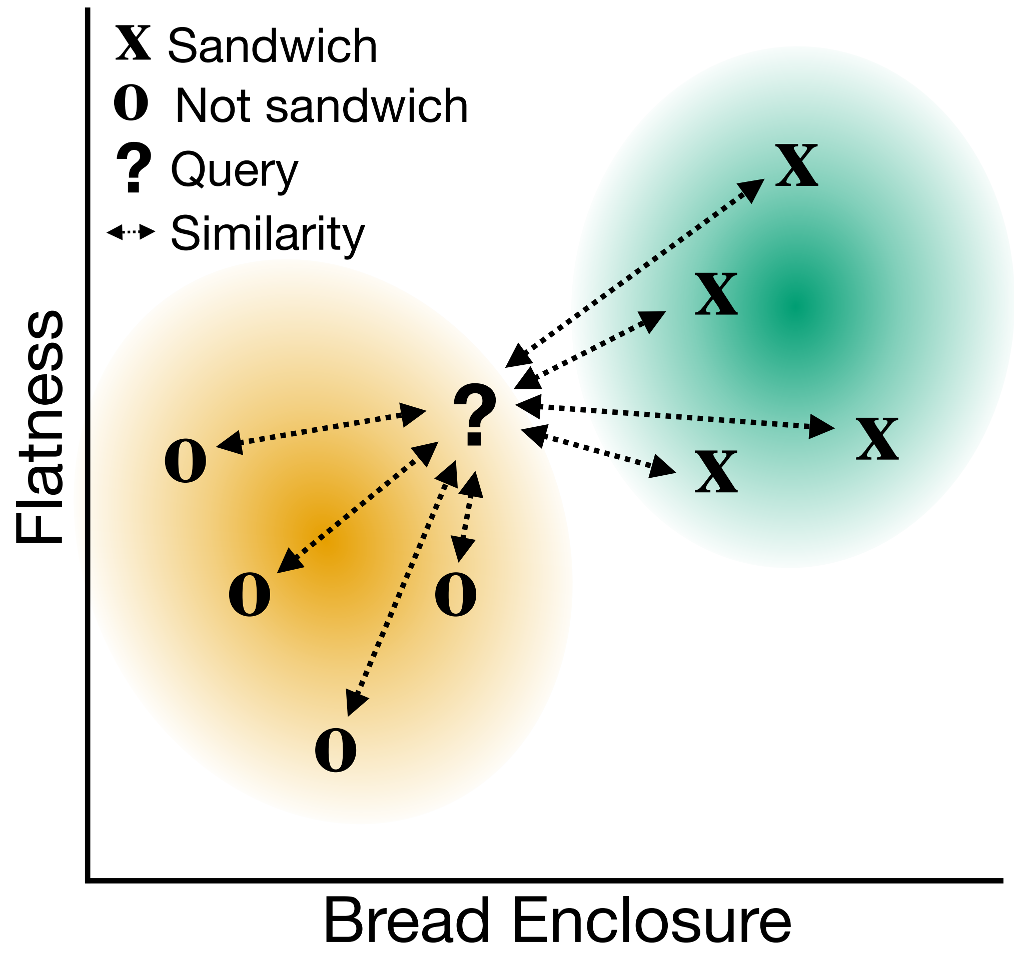
Concept learning as classification

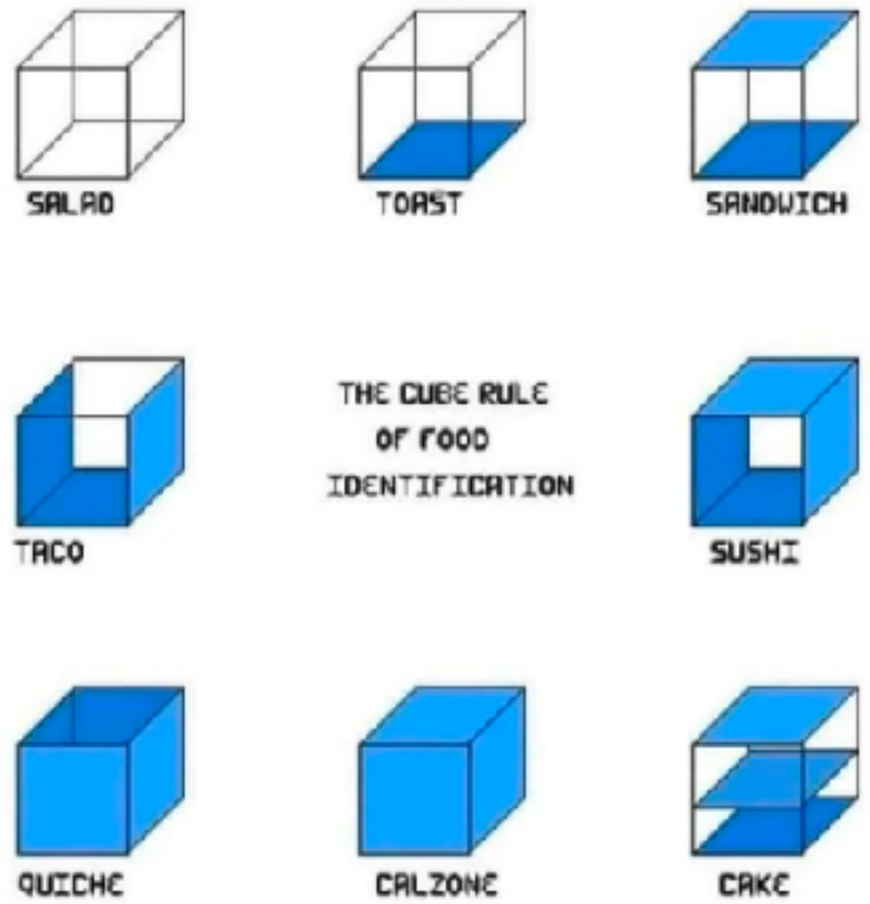


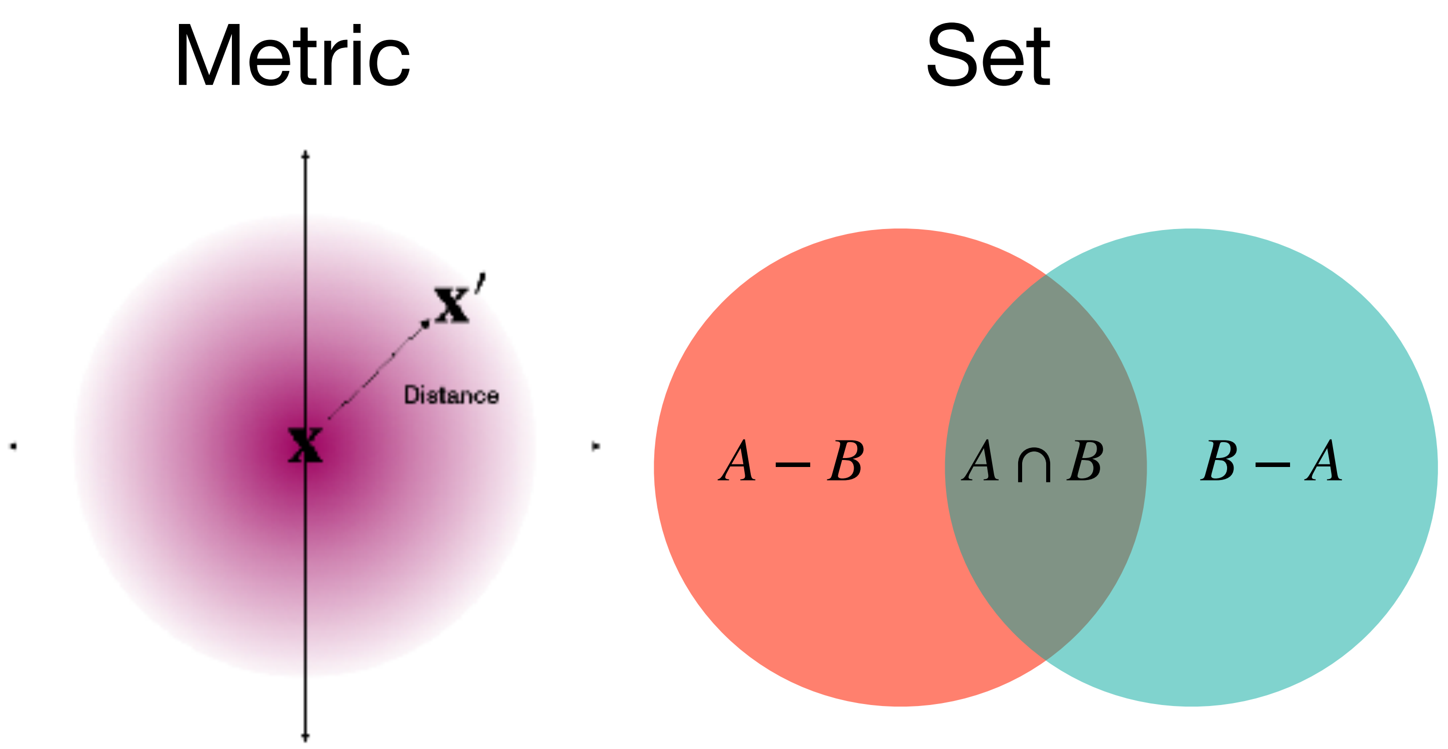
Rule-based



Similarity-based



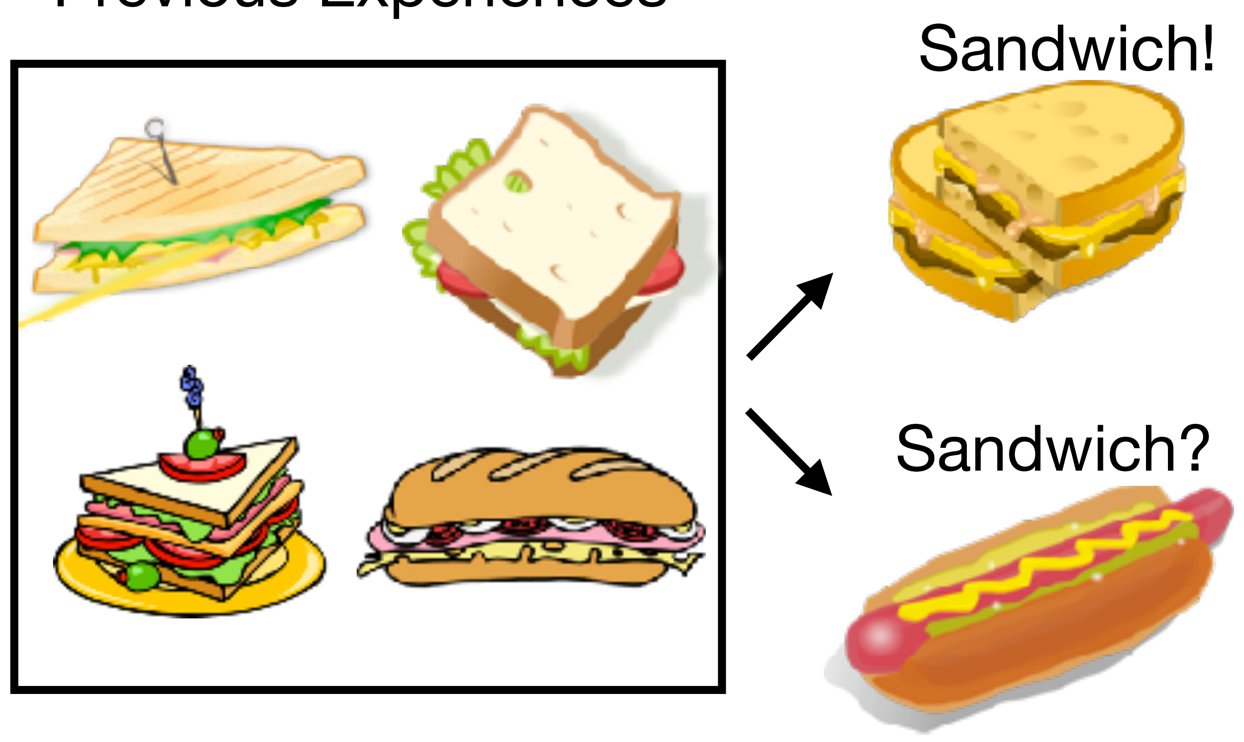




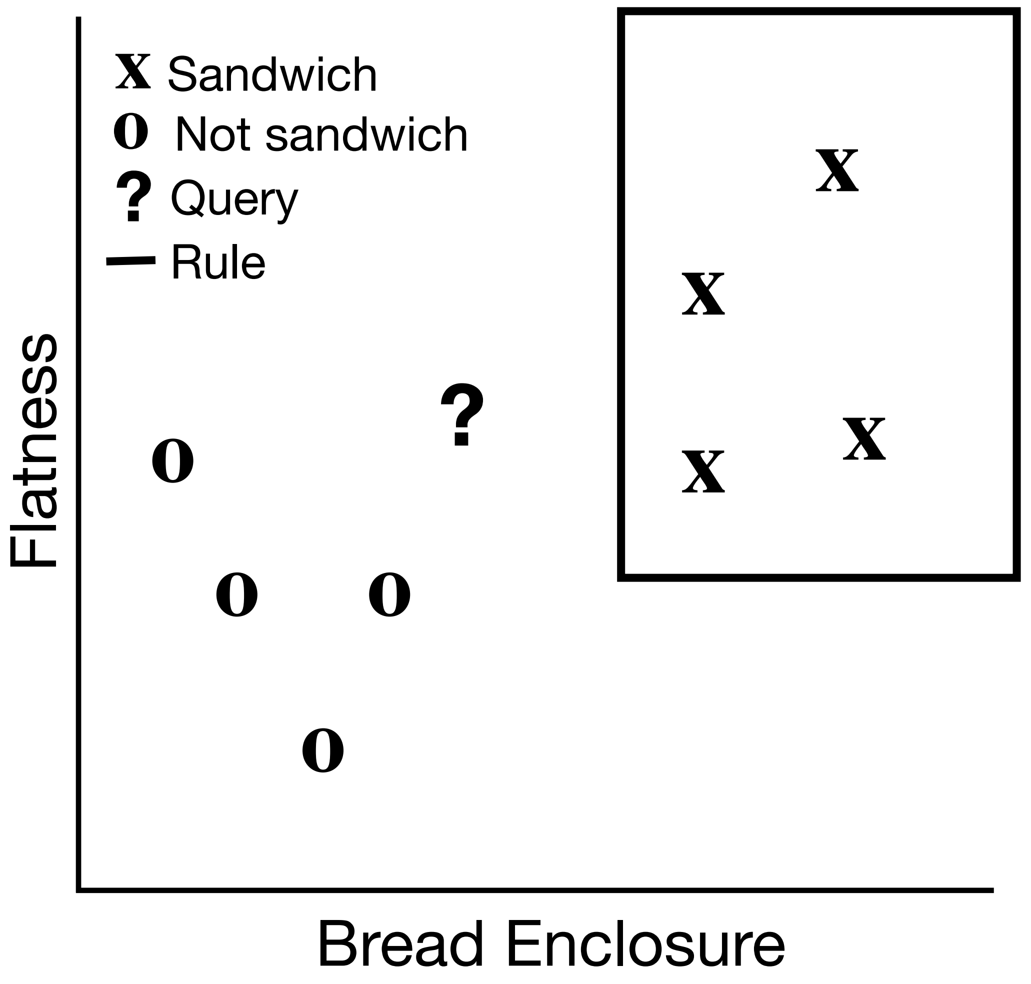
The story so far …

Concept learning as classification

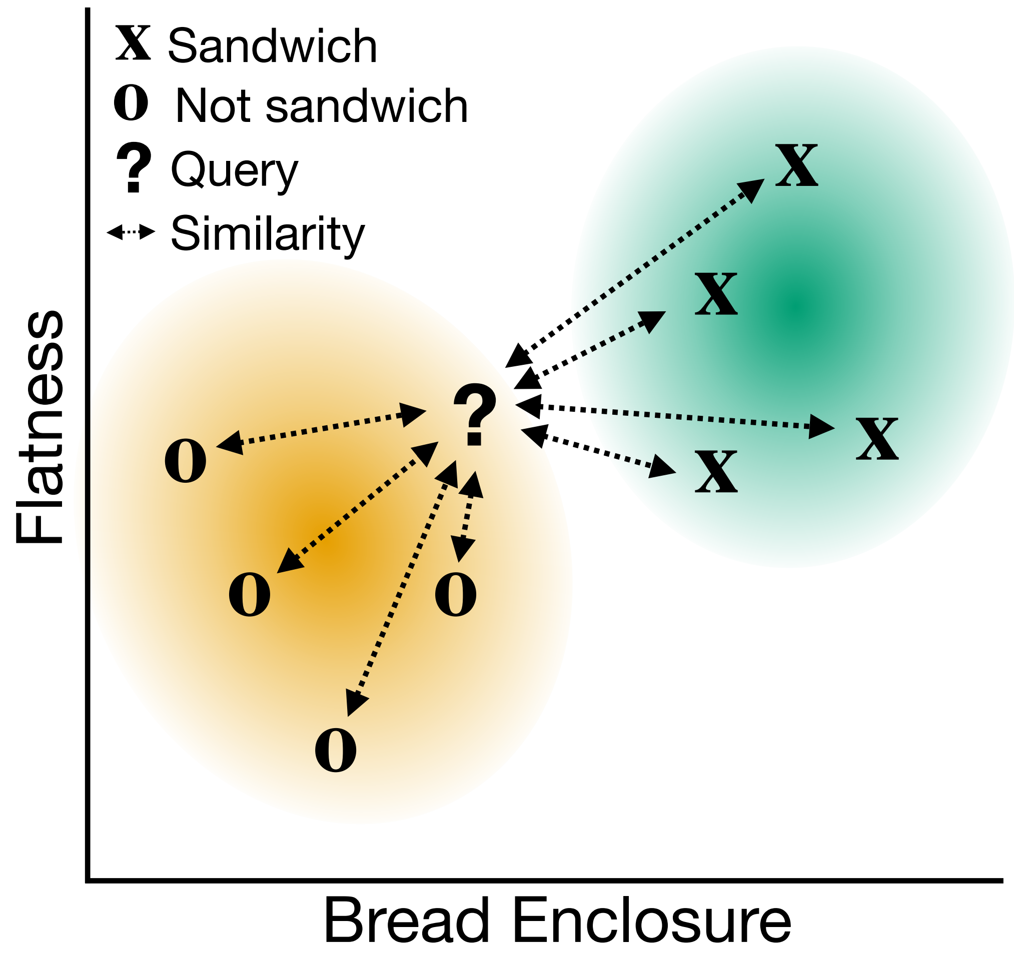
Previous Experiences



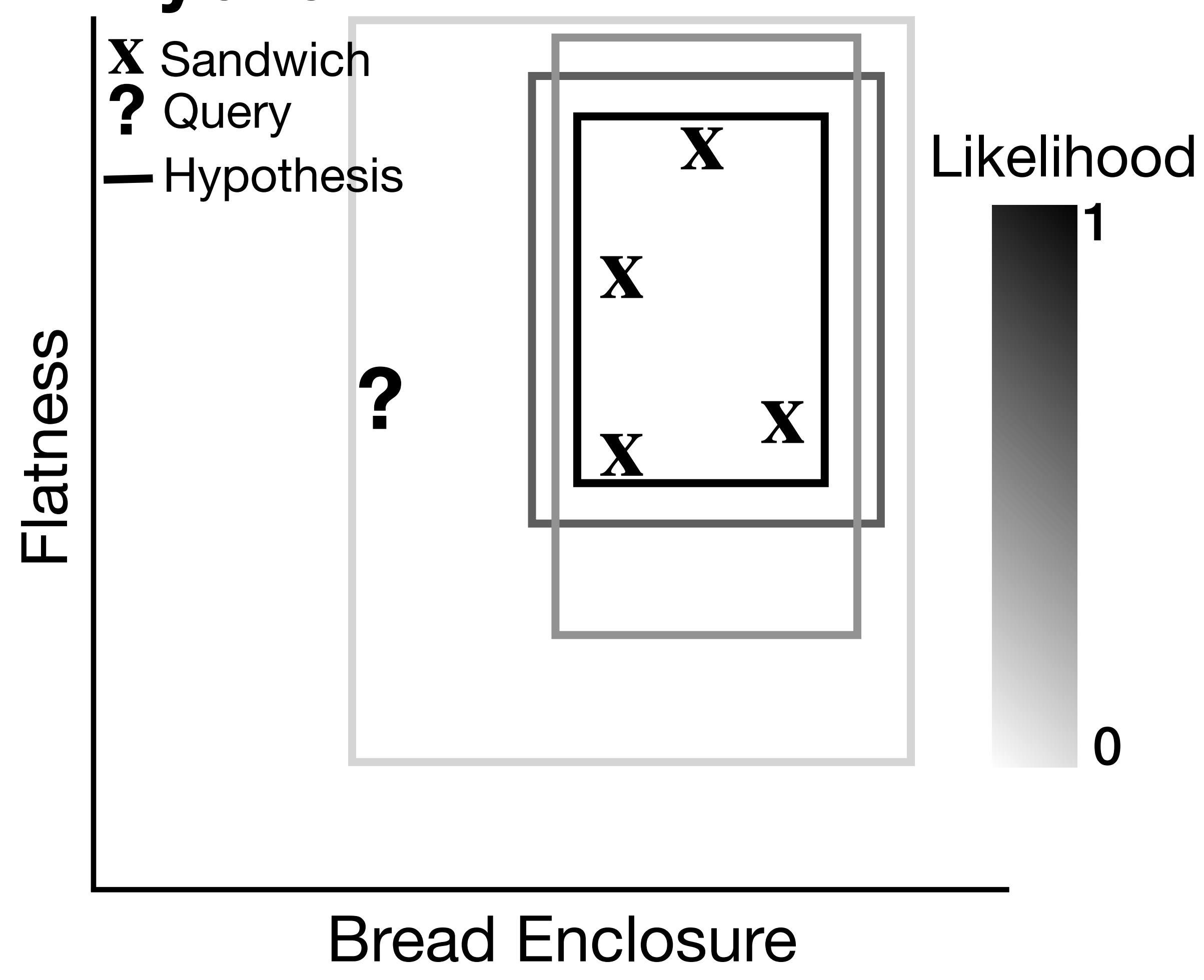
Rule-based

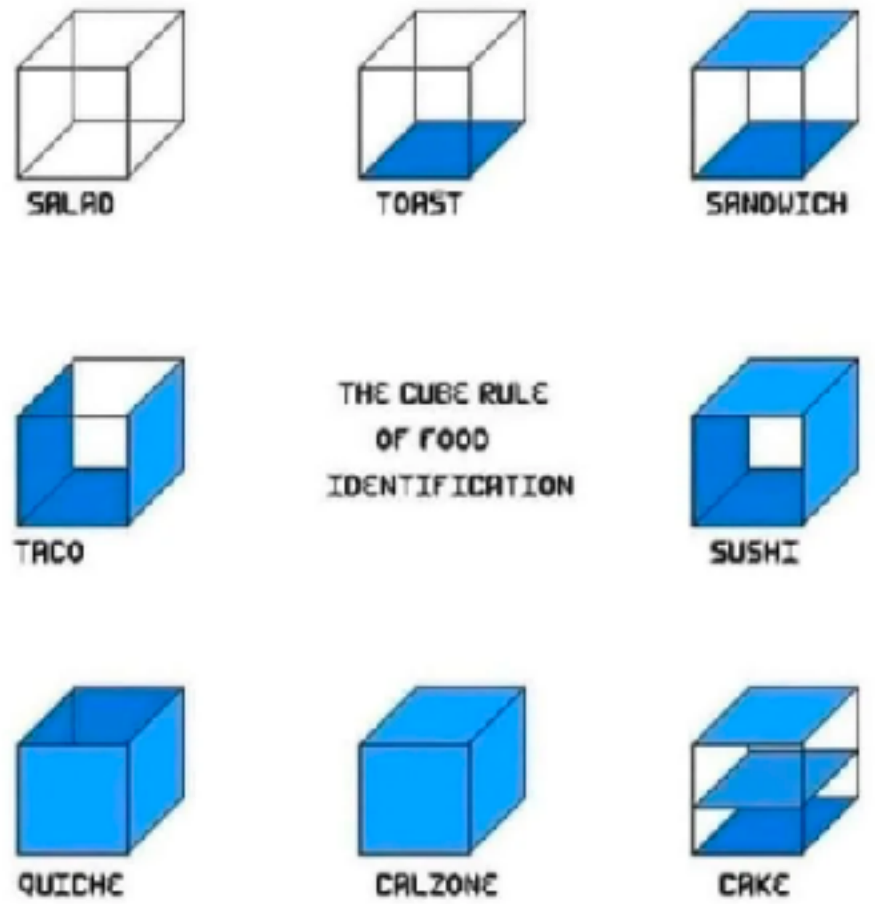


Similarity-based

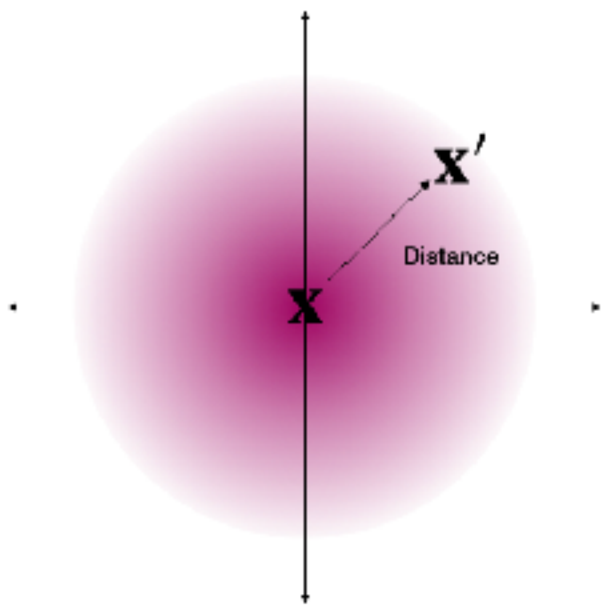


Hybrid

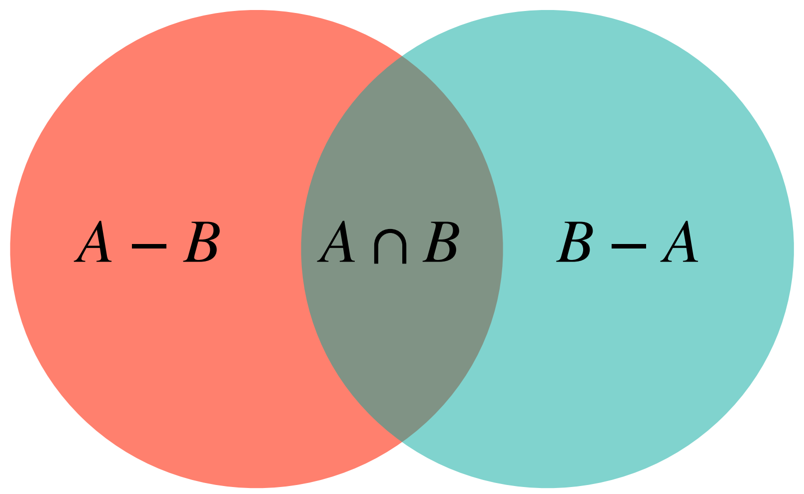




Metric

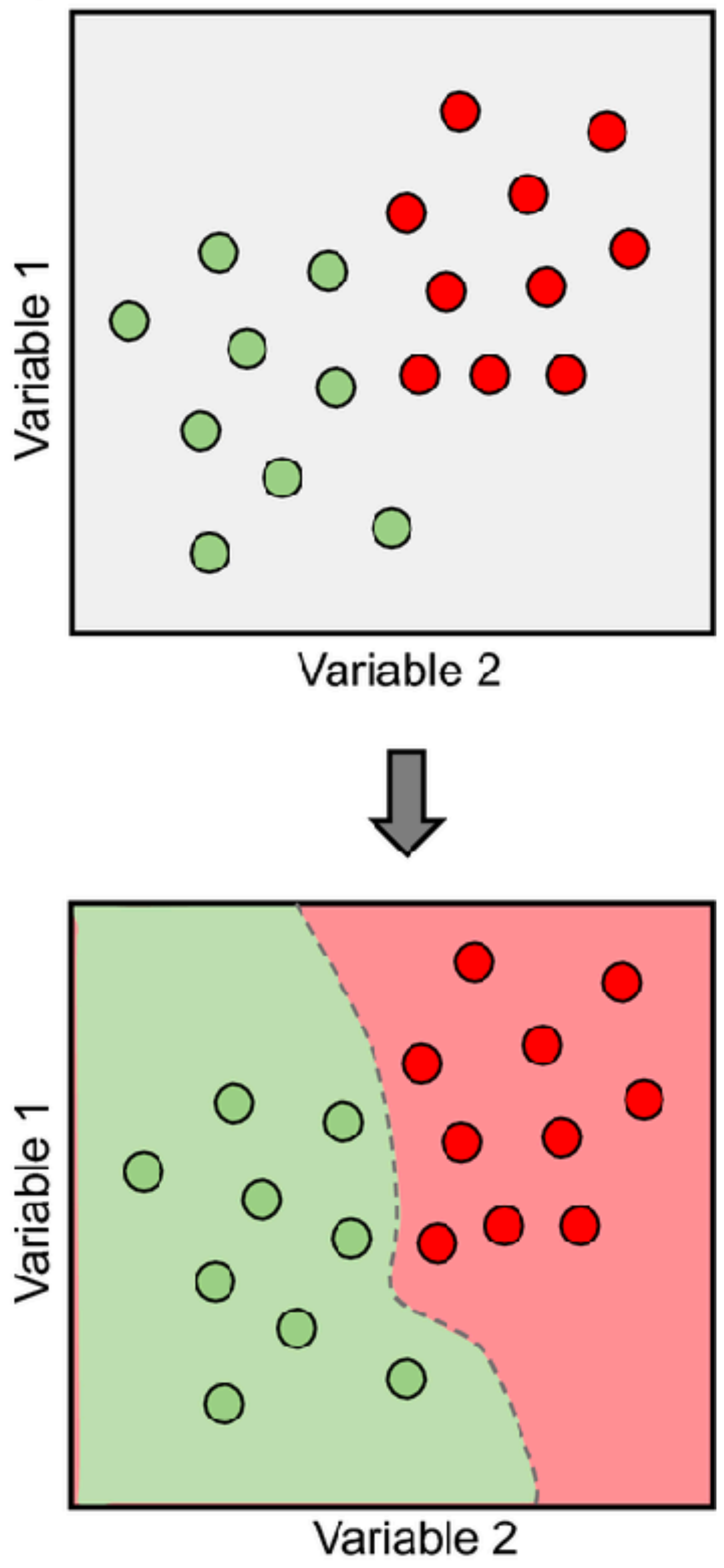


Set

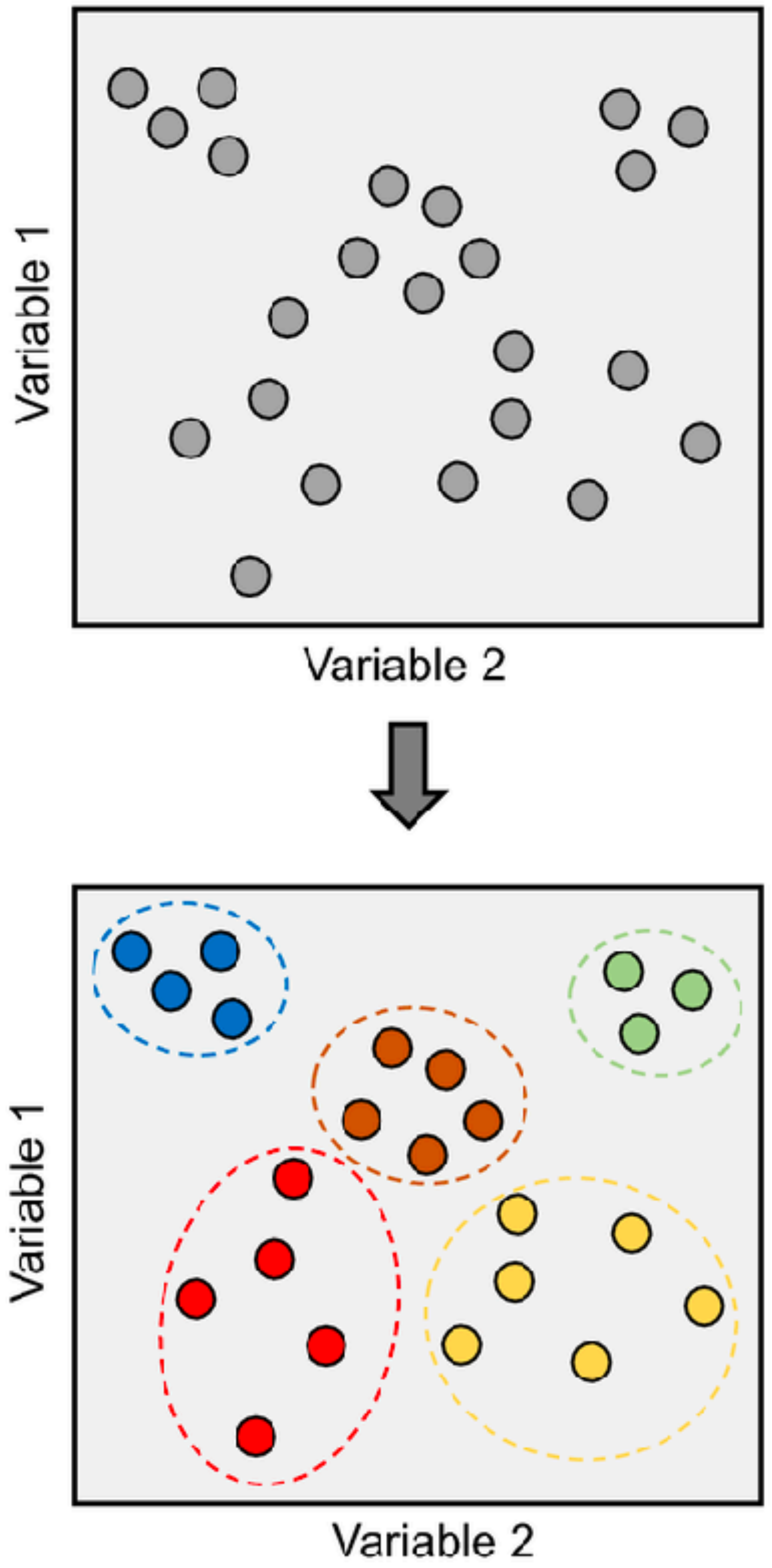


The story so far …

Supervised

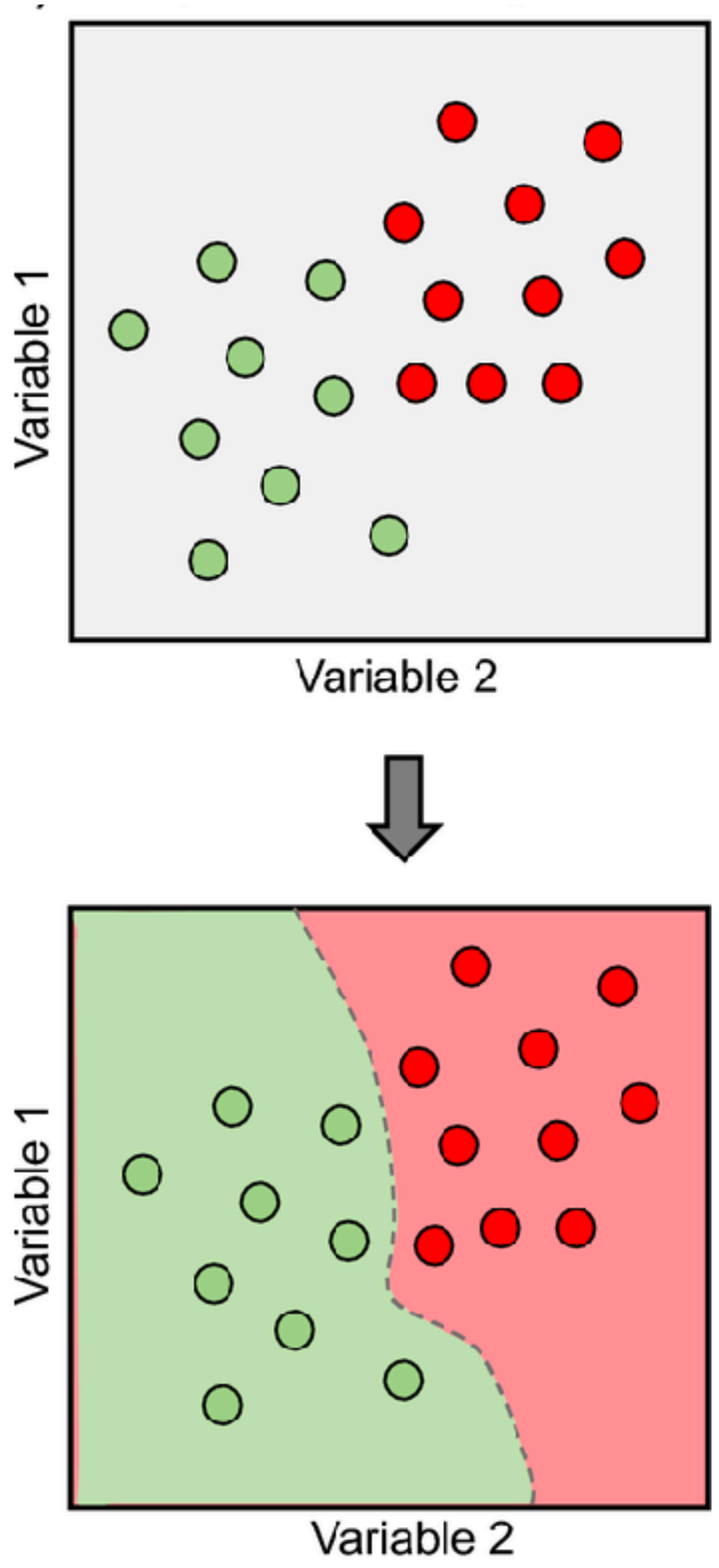


Unsupervised

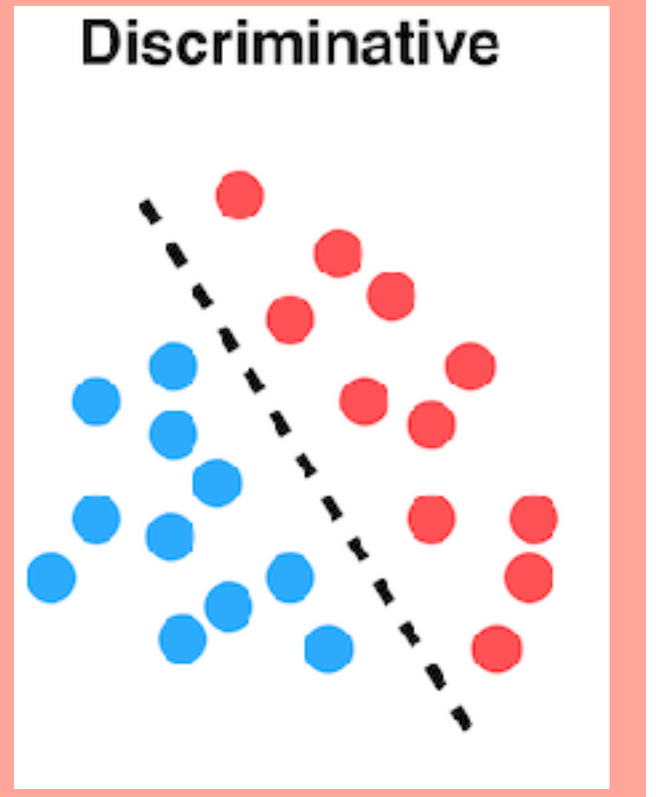


The story so far …

Supervised



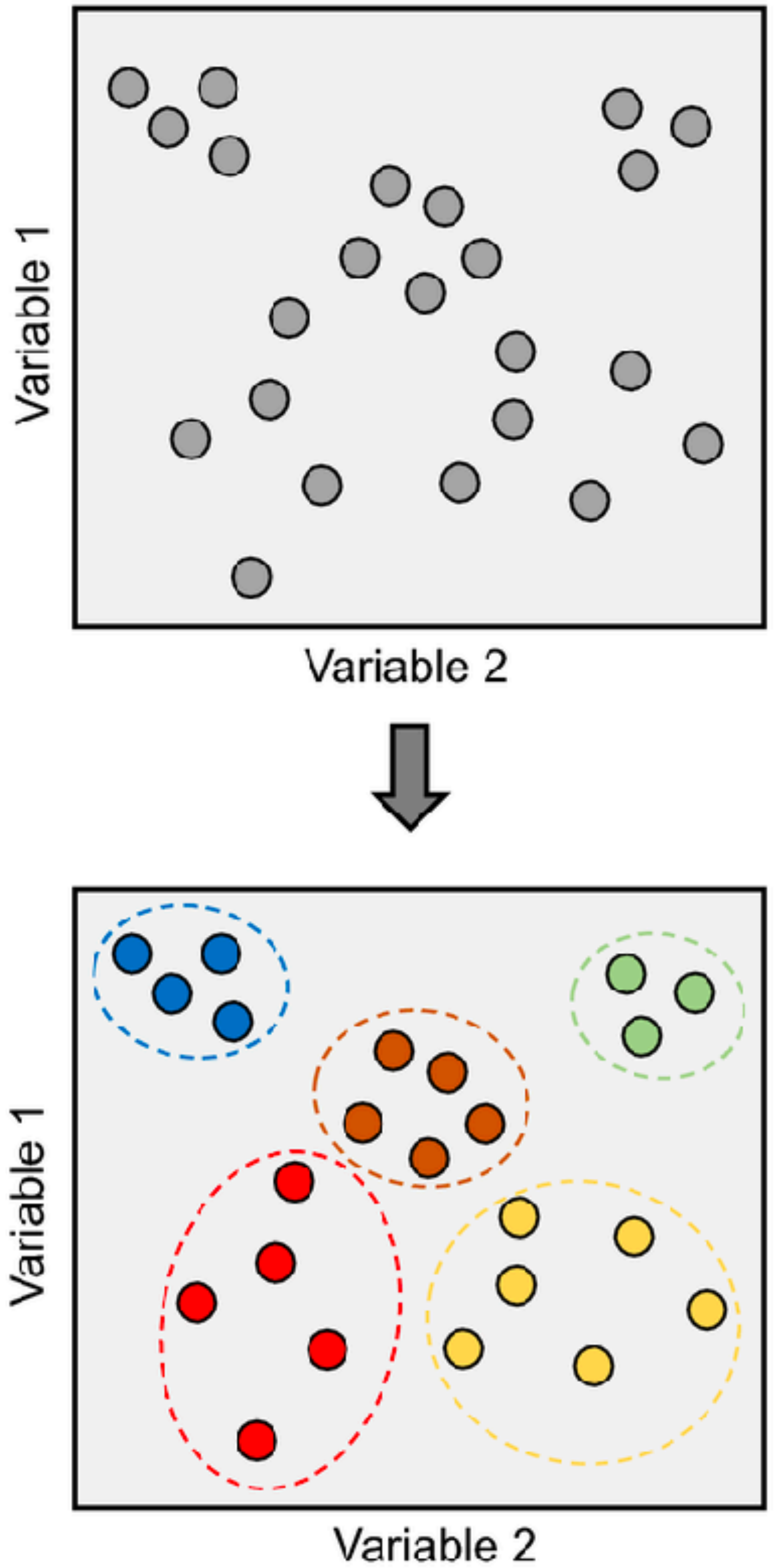
MLPs



Decision trees and random forests

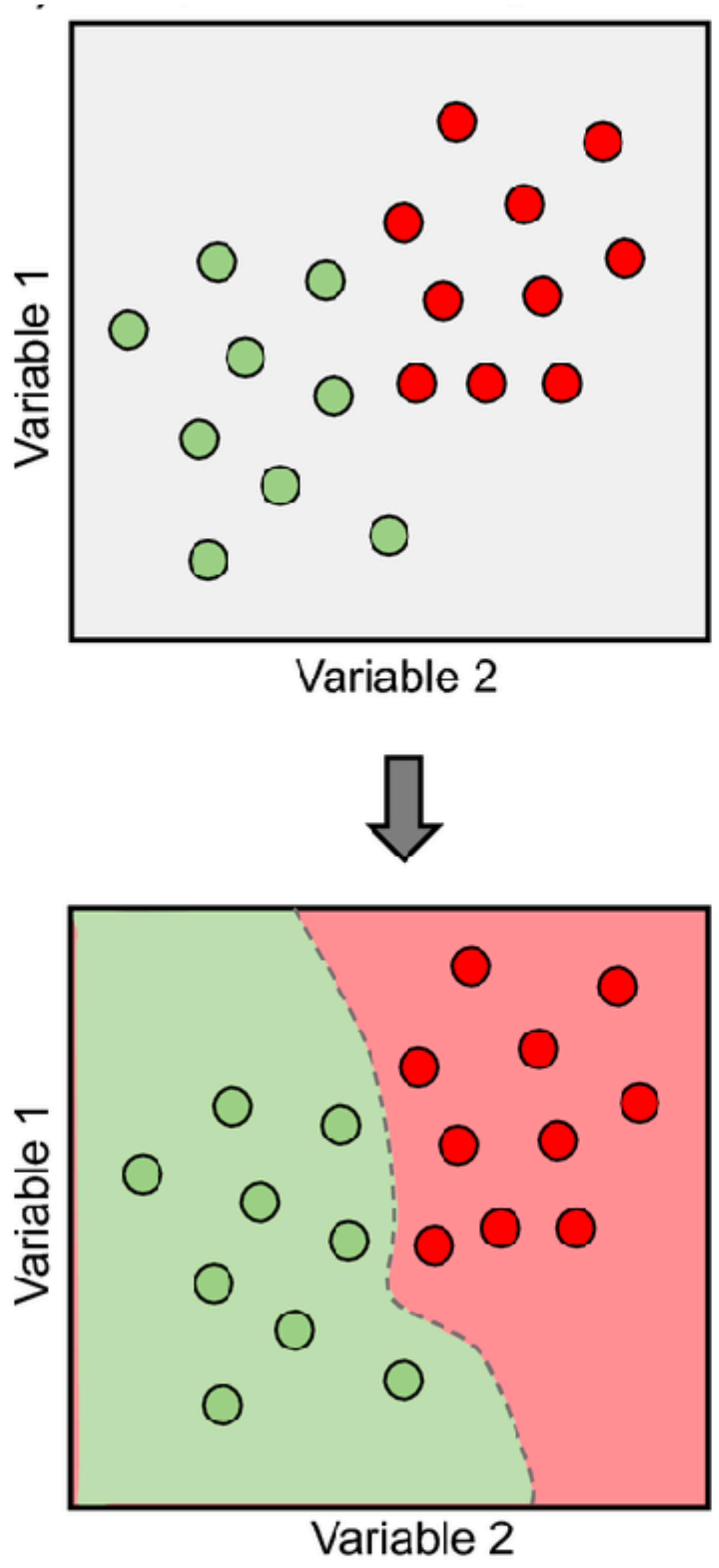
SVMs

Unsupervised

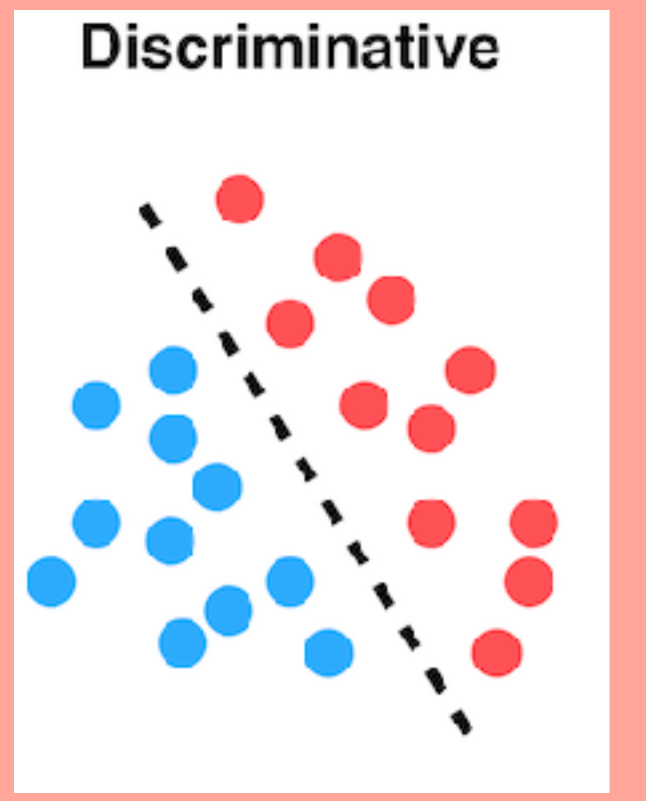


The story so far …

Supervised



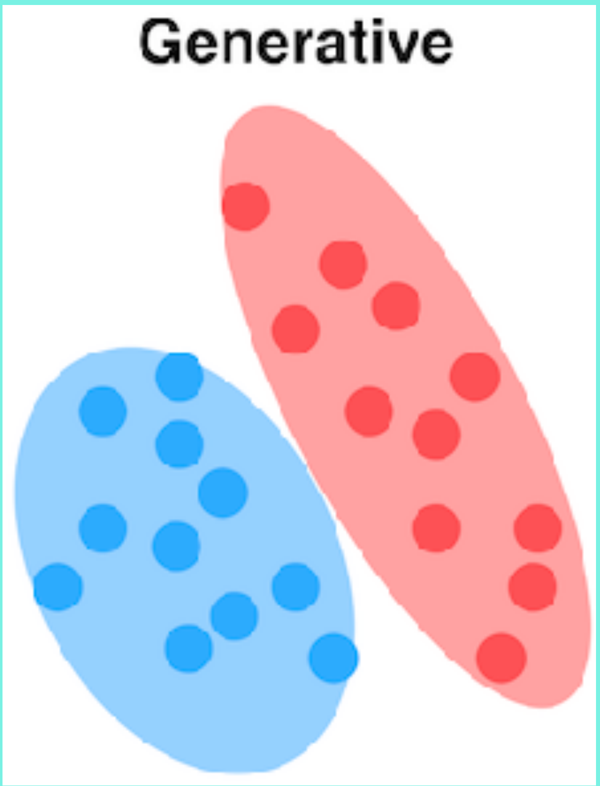
MLPs



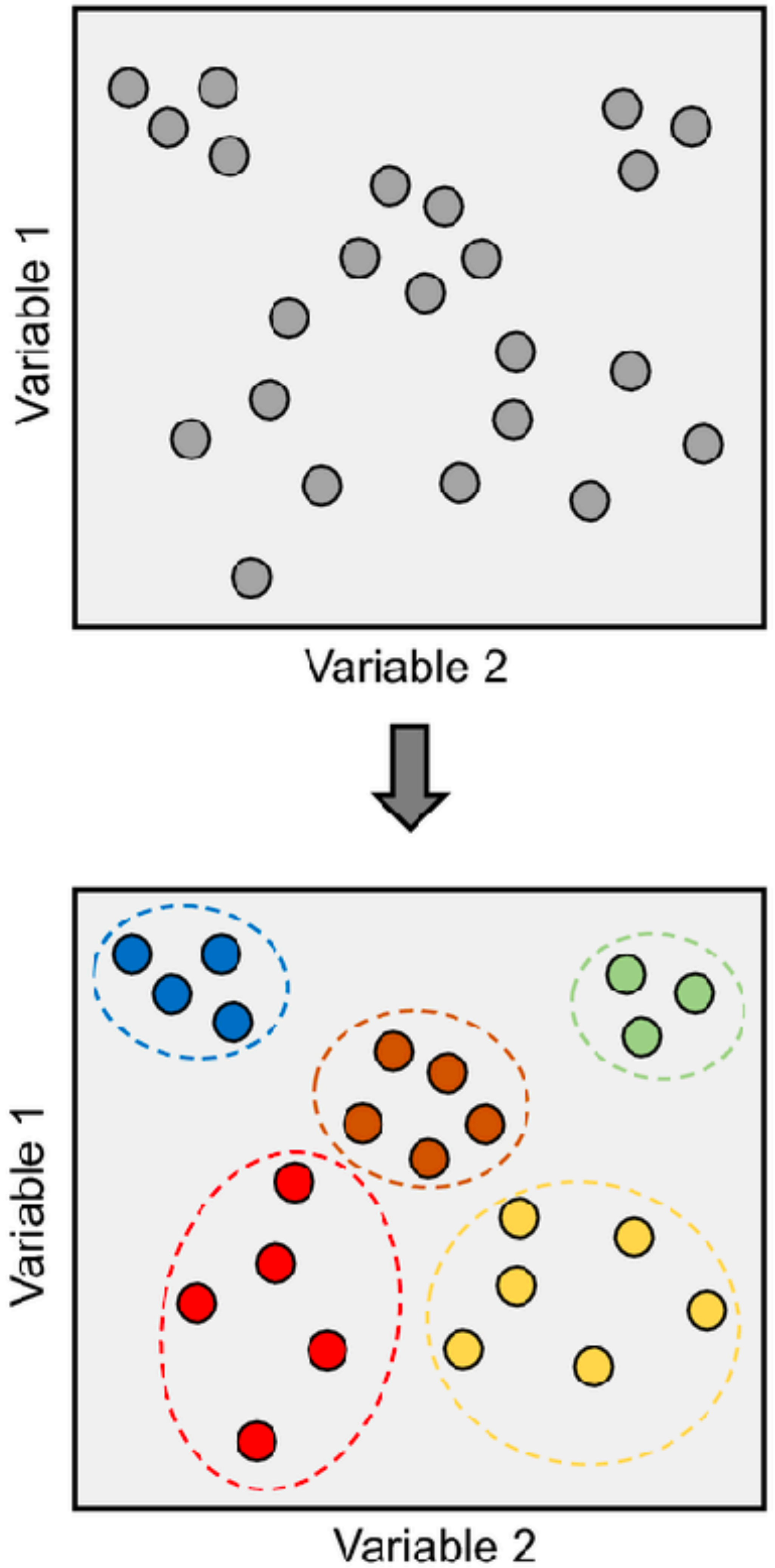
Decision trees and random forests

SVMs

Naïve Bayes

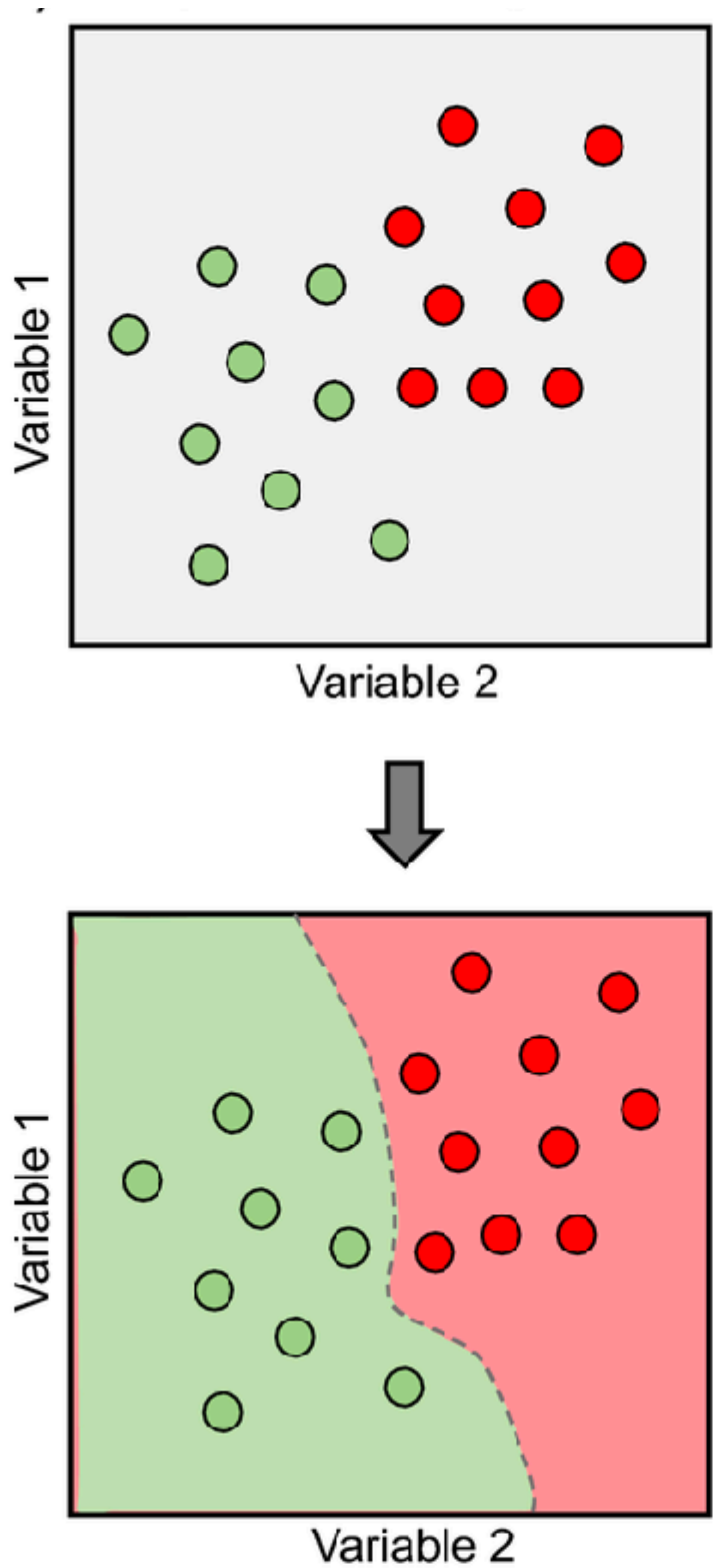


Unsupervised

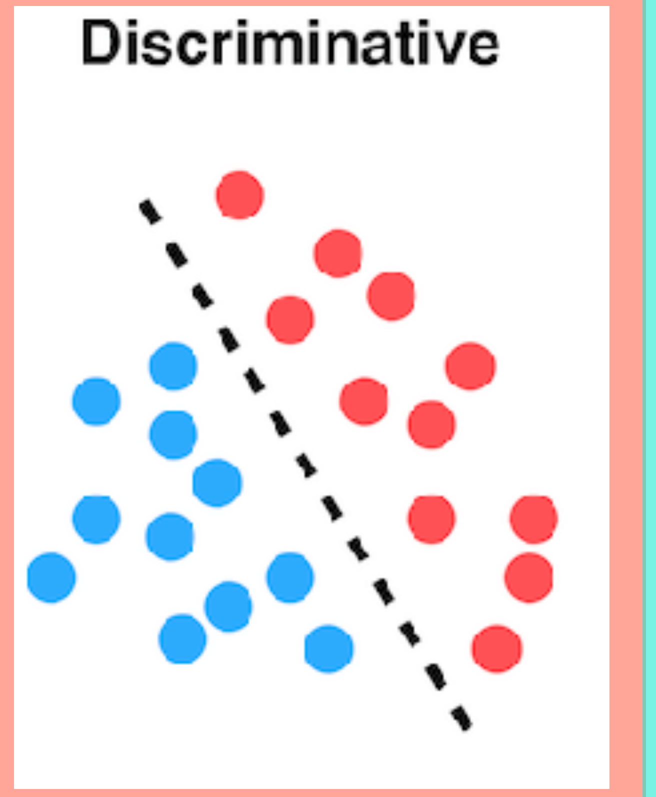


The story so far …

Supervised



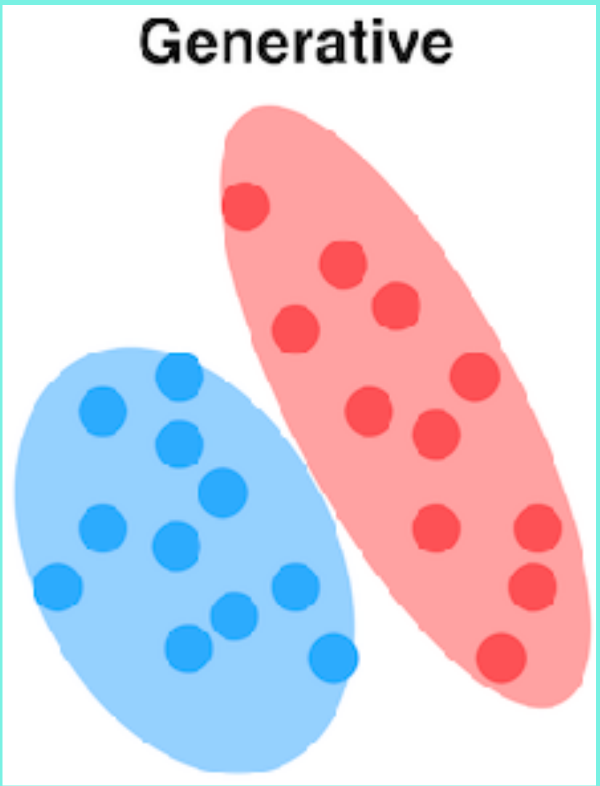
MLPs



Decision trees and random forests

SVMs

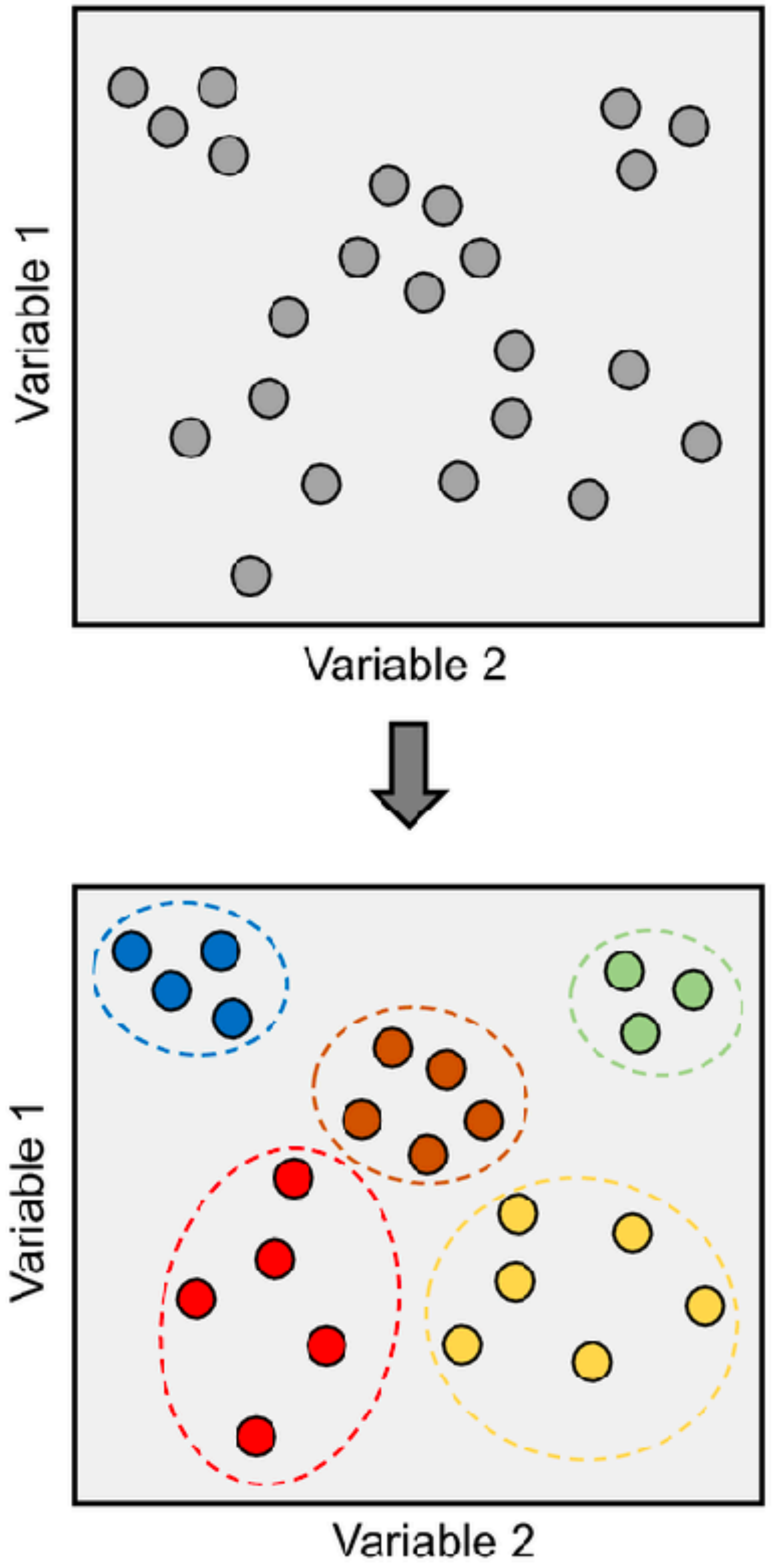
Naïve Bayes



k-Means

GMMs

Unsupervised

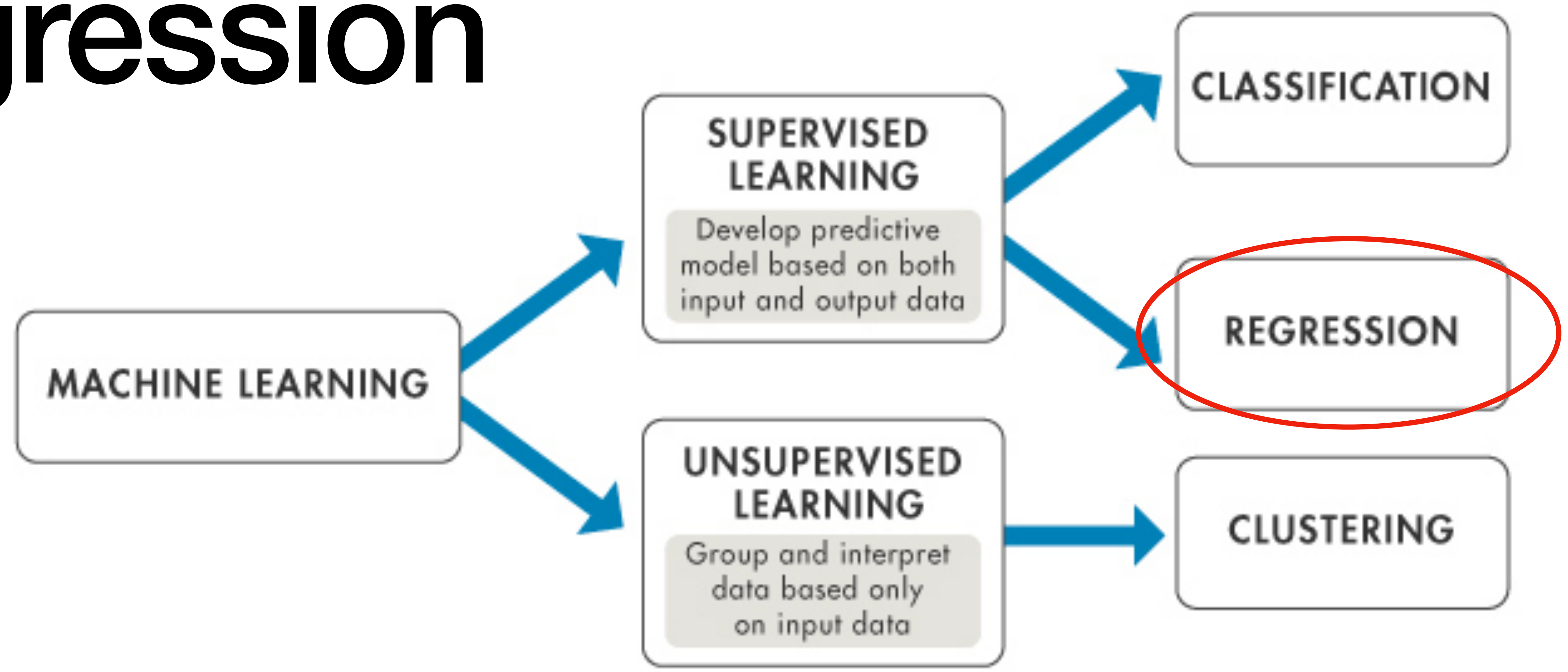


Today's agenda

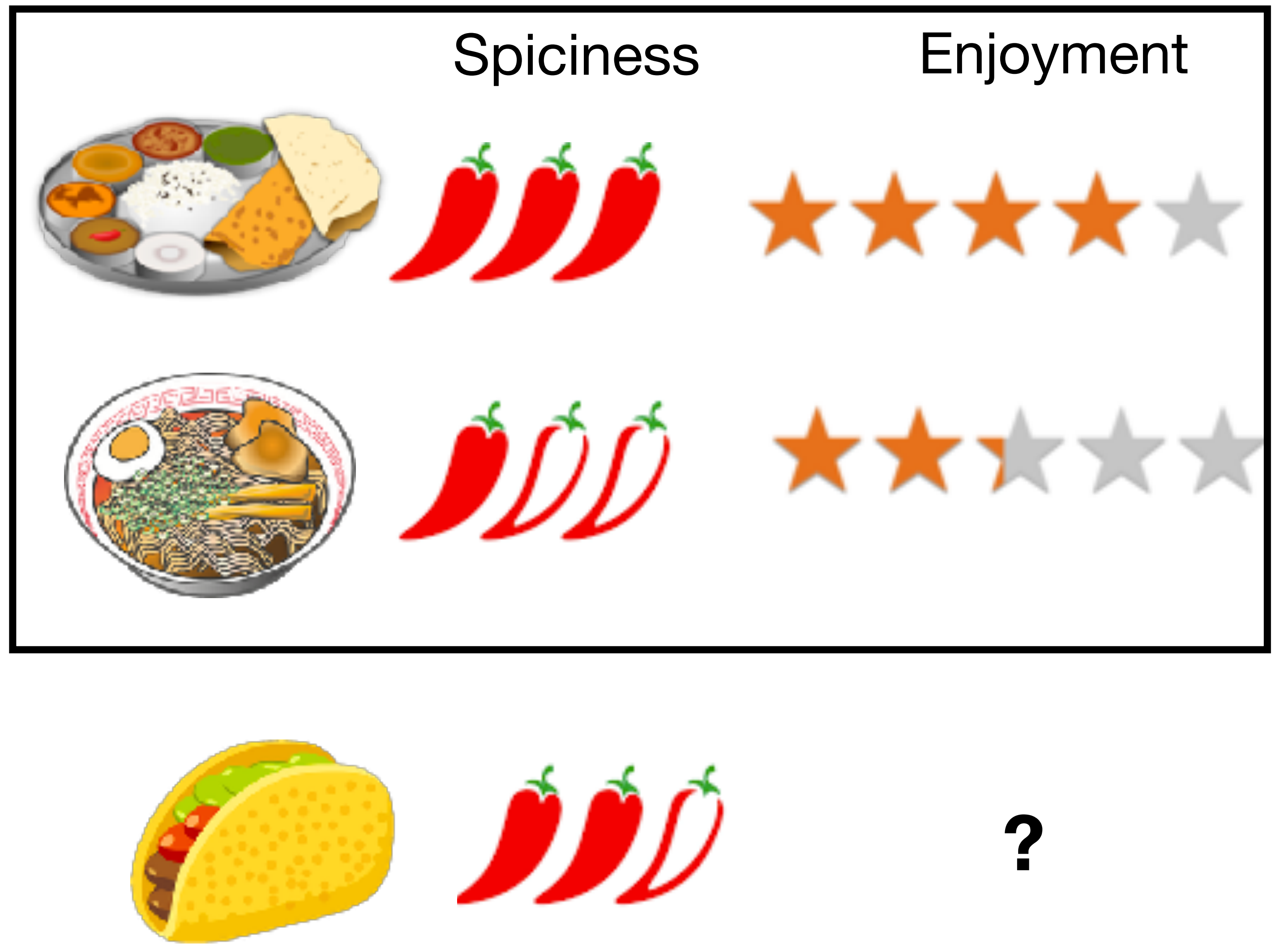
- From categories to functions
    - Early Psychological research on how people learn explicit functions
        - Rule-based
        - Similarity-based
        - Hybrid using Bayesian function learning
    - Implicit function learning as a key part of generalization in RL
- Modeling human generalization and exploration in RL
    - Spatially correlated bandit (Wu et al,. 2018)
    - Generalization to abstract (Wu et al., 2020) and graph-structured domains (Wu et al,. 2021)
    - Open challenges

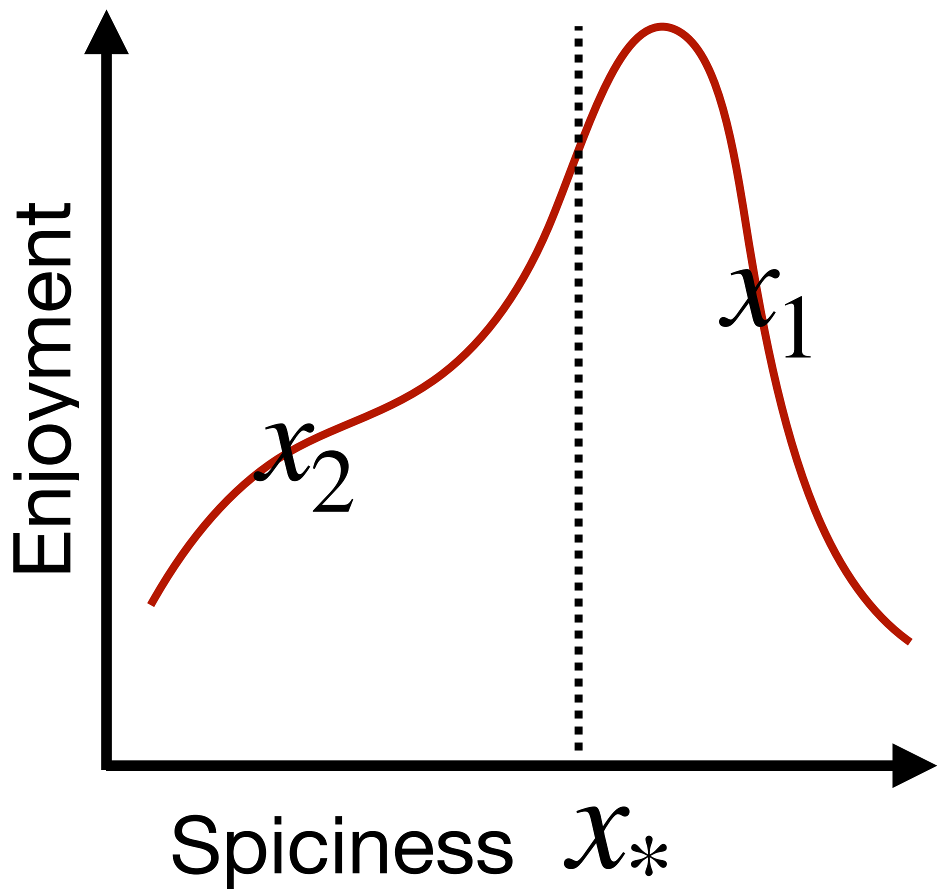
Function learning as regression

- Regression is that other branch of supervised learning problems we previously skipped over
- Rather than predicting *discrete* categories, we want to learn to predict a *continuous* real-valued variable
- We do so by learning a function mapping the input space X to the target variable Y
  $f : X \rightarrow Y$ where $y = f(x)$
- To make a prediction about so new situation x_*, we simply evaluate the function:
  $y_* = f(x_*)$
- *But how do we learn this function*? For any set of datapoints, there are an infinite number of functions that pass through them



Previous Experiences

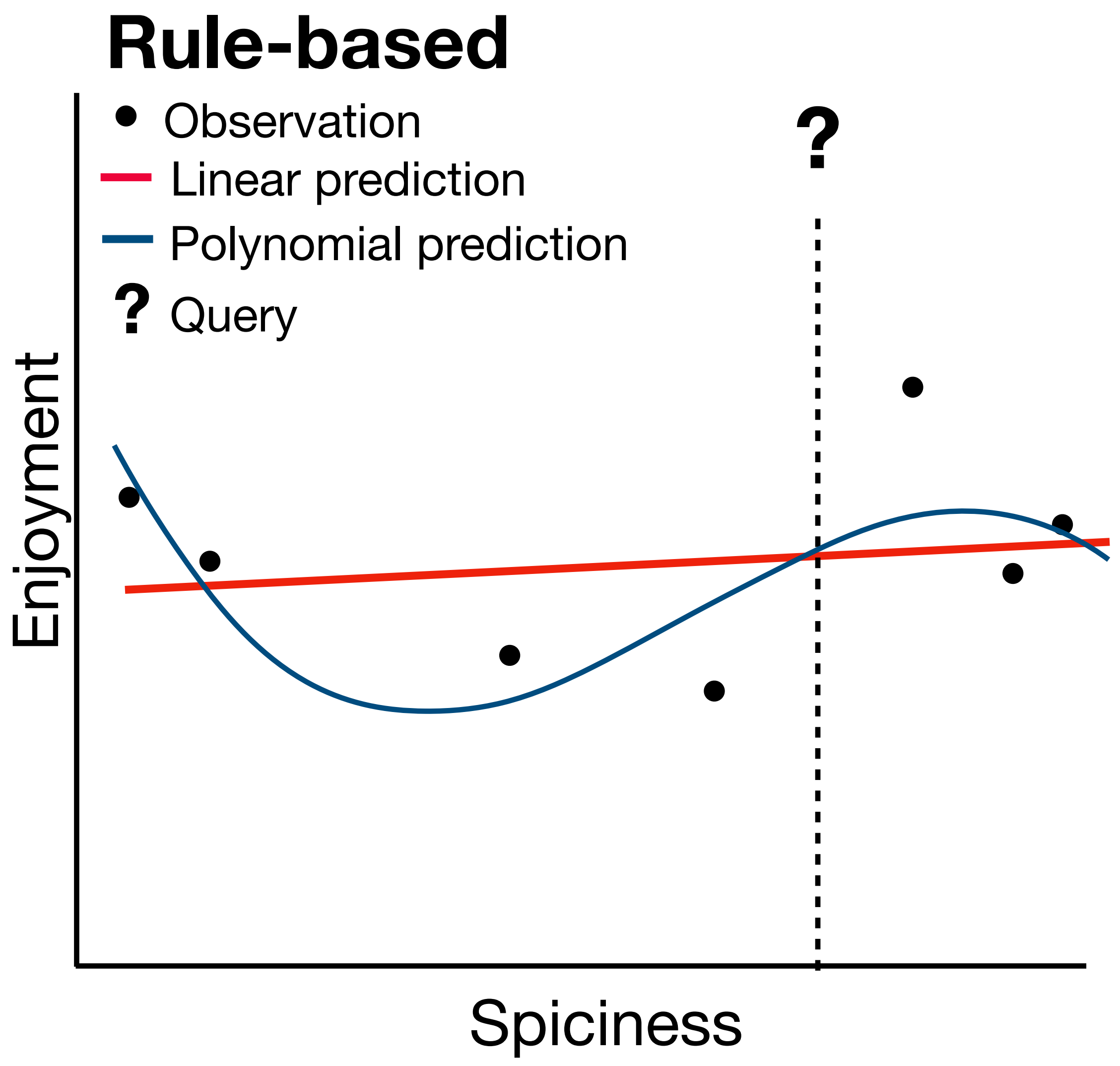




Theories of Function Learning

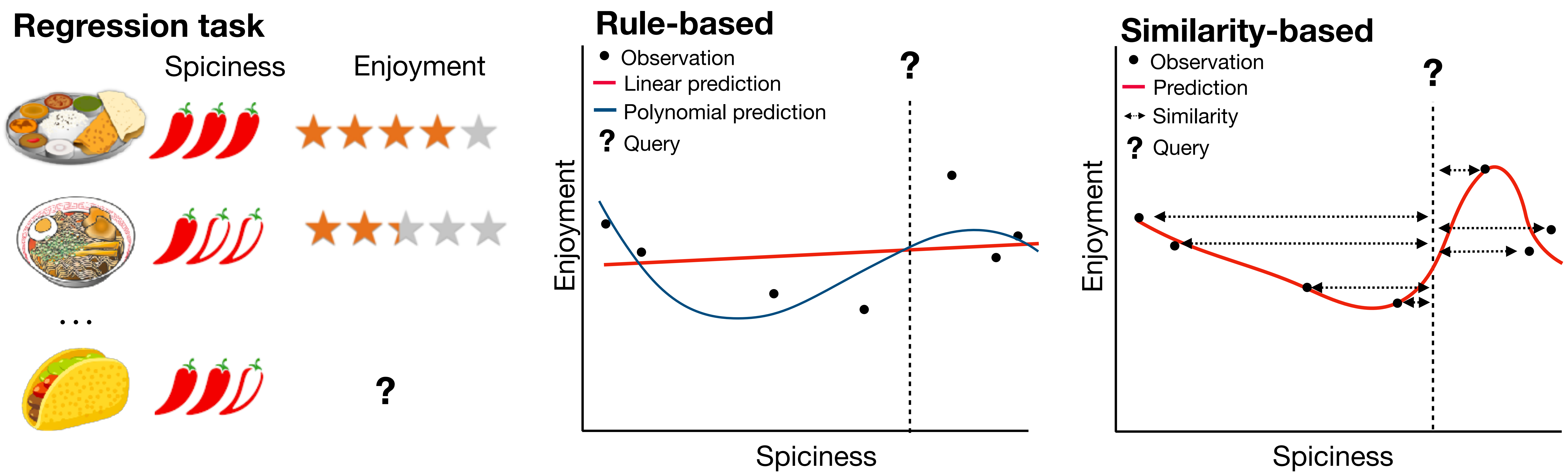
Regression task

Theories of Function Learning



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

Theories of Function Learning

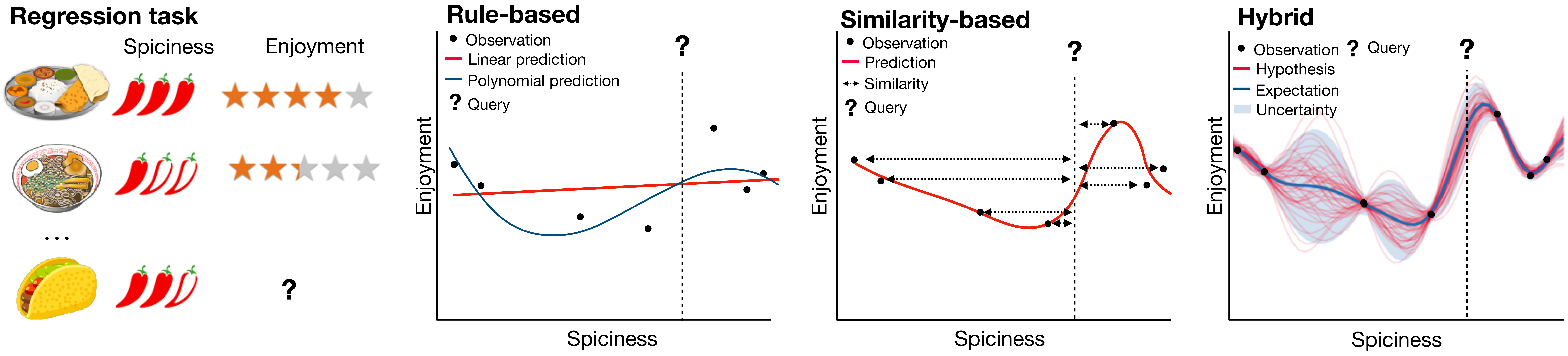


- *Rules* describe an explicit parametric family of candidate functions (e.g., linear or polynomial)
  (Carroll, 1963; Brehmer, 1976)

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

Wu, Meder & Schulz (*AnnRevPsych* forthcoming)

Theories of Function Learning



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

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

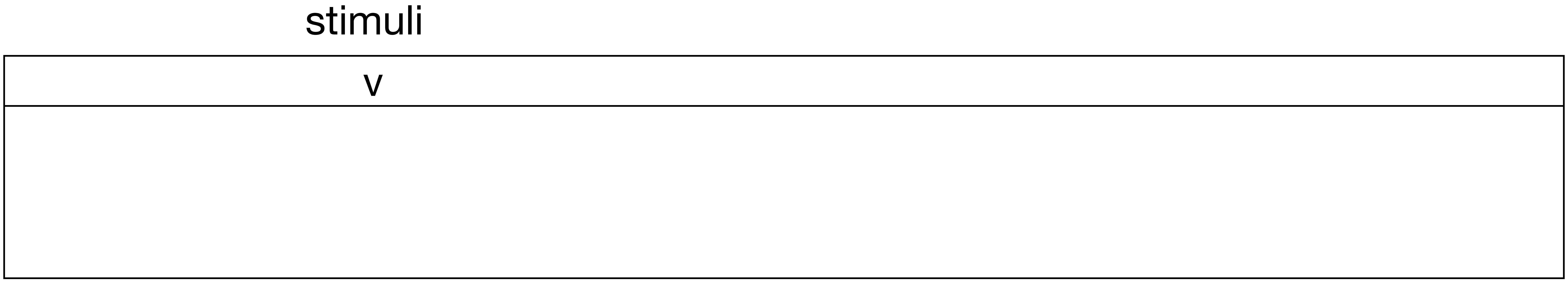
- *Hybrids* combine elements of both: Gaussian process (GP) regression uses kernel similarity to learn a distribution over functions, and can compositionally combine kernels like we can combine multiple rules
  (Rasmussen & Williams, 2005; Mercer, PhilTransRoySoc 1909; Lucas et al., PBR 2015)

Rule-based theories of function learning

- Carroll (1963) was one of the first to study how people learned continuous mappings between stimuli and responses
  - Rather than learning discrete S-R associations, people learn functions
  - Functions are not just a response, but correspond to a set of rules or programs, allowing for interpolation and extrapolation
- Experiment using relationships such as $y = 1.22x + 1.0$ or $y = -5.1x + 0.2x^2 + 32.60$

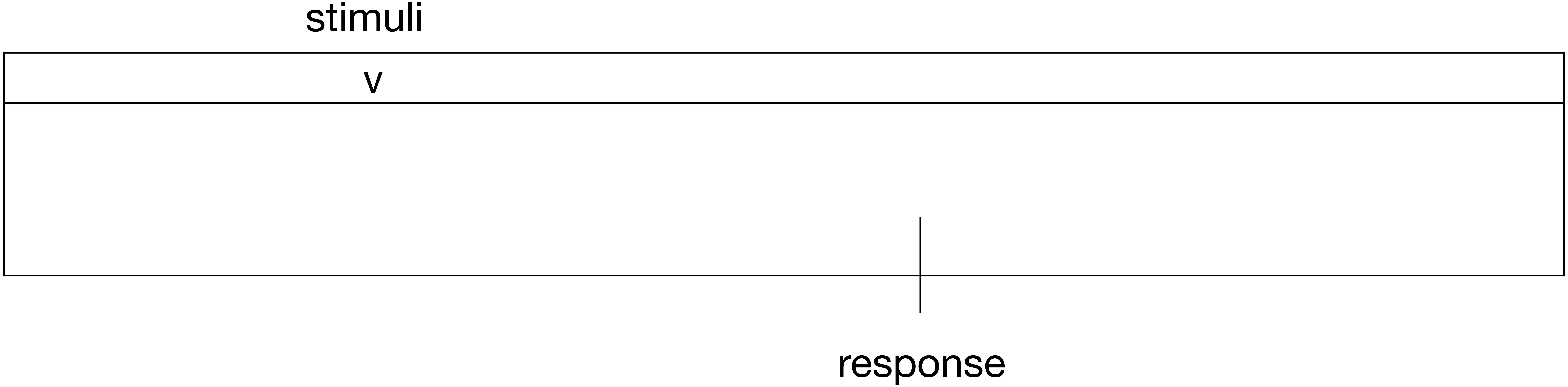
Rule-based theories of function learning

- Carroll (1963) was one of the first to study how people learned continuous mappings between stimuli and responses
  - Rather than learning discrete S-R associations, people learn functions
  - Functions are not just a response, but correspond to a set of rules or programs, allowing for interpolation and extrapolation
- Experiment using relationships such as $y = 1.22x + 1.0$ or $y = -5.1x + 0.2x^2 + 32.60$



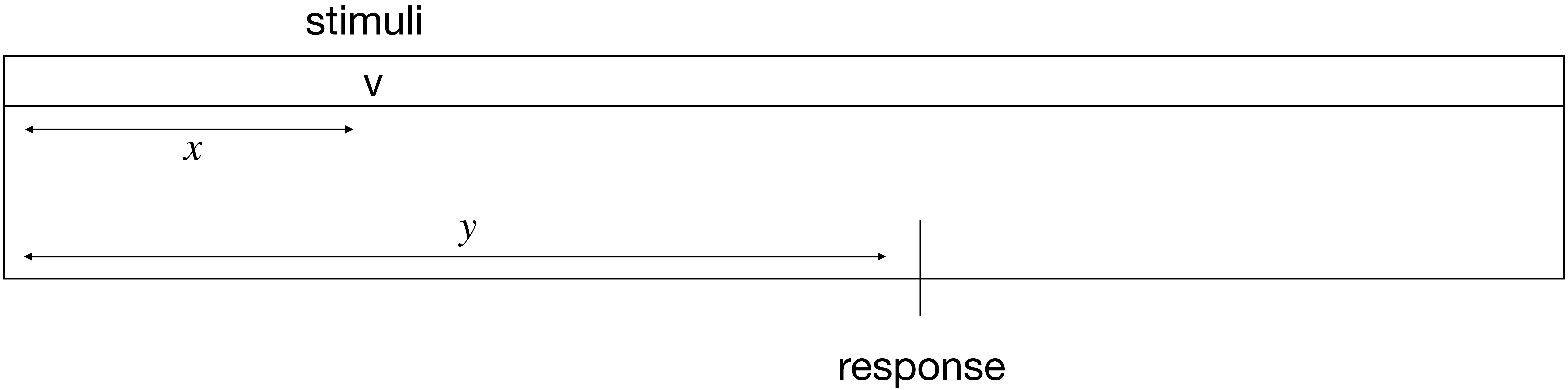
Rule-based theories of function learning

- Carroll (1963) was one of the first to study how people learned continuous mappings between stimuli and responses
  - Rather than learning discrete S-R associations, people learn functions
  - Functions are not just a response, but correspond to a set of rules or programs, allowing for interpolation and extrapolation
- Experiment using relationships such as $y = 1.22x + 1.0$ or $y = -5.1x + 0.2x^2 + 32.60$



Rule-based theories of function learning

- Carroll (1963) was one of the first to study how people learned continuous mappings between stimuli and responses
  - Rather than learning discrete S-R associations, people learn functions
  - Functions are not just a response, but correspond to a set of rules or programs, allowing for interpolation and extrapolation
- Experiment using relationships such as $y = 1.22x + 1.0$ or $y = -5.1x + 0.2x^2 + 32.60$



Results and interpretation

- Participants are learning functions rather than just discrete associations because they can interprolate and extrapolate from training data
- Participants learned simpler functions better than more complex ones and displayed inductive biases for simpler functions even when shown arbitrary relationships between x and y (no more than 4th degree polynomials)
- Carroll (1967) along with later work (Brehmer 1974; Koh & Meyer, 1991) believed that people learn functions by estimating the parameters for a class of functions (e.g., polynomials) using a process equivalent to regression
    - The class of function corresponds to a hypothesized **rule** about the relationship between variables (e.g., law of gravity: $F = G\frac{m_1 m_2}{r}$)

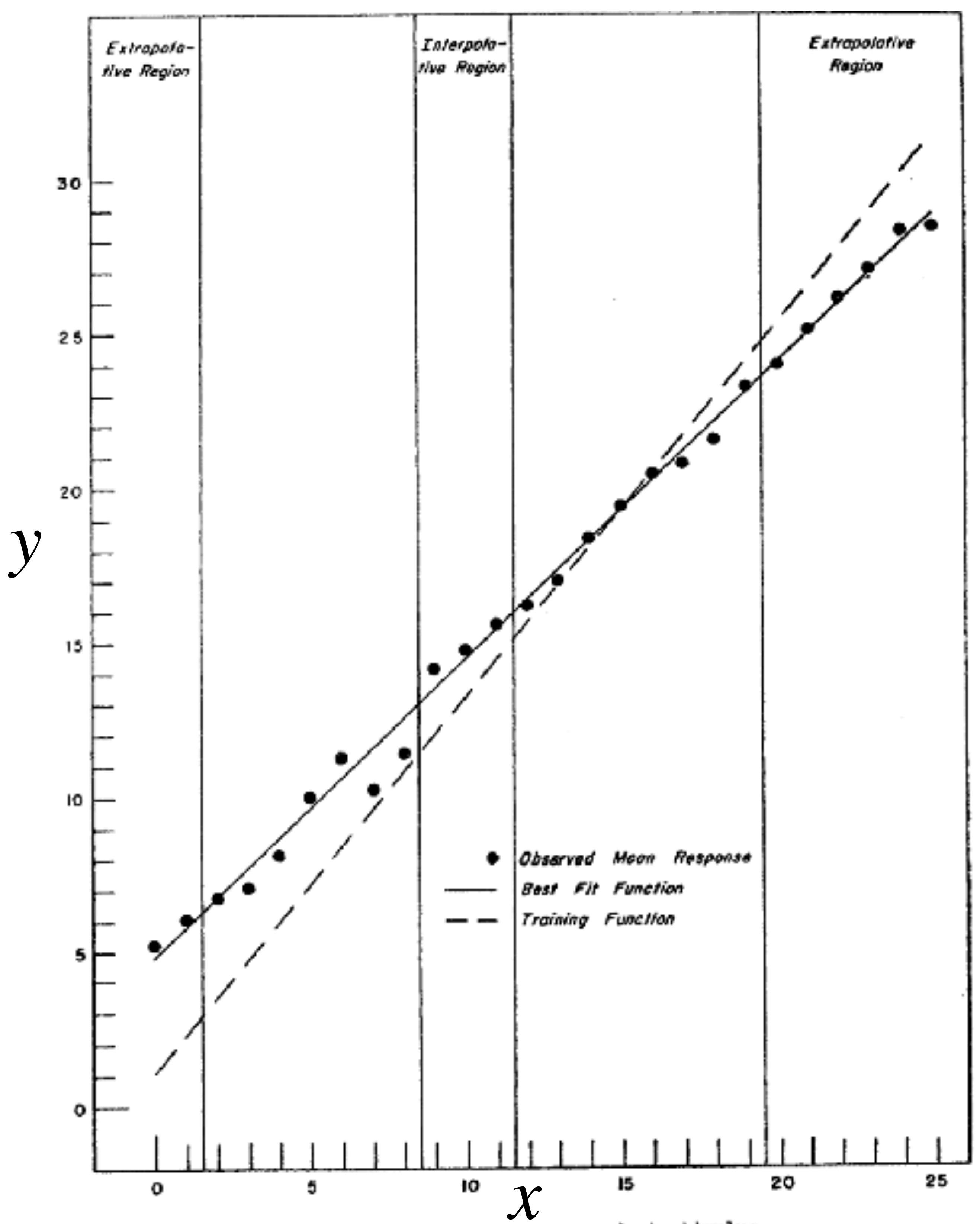


Fig. 3. Plot of mean response against stimulus for subject #4, condition I.

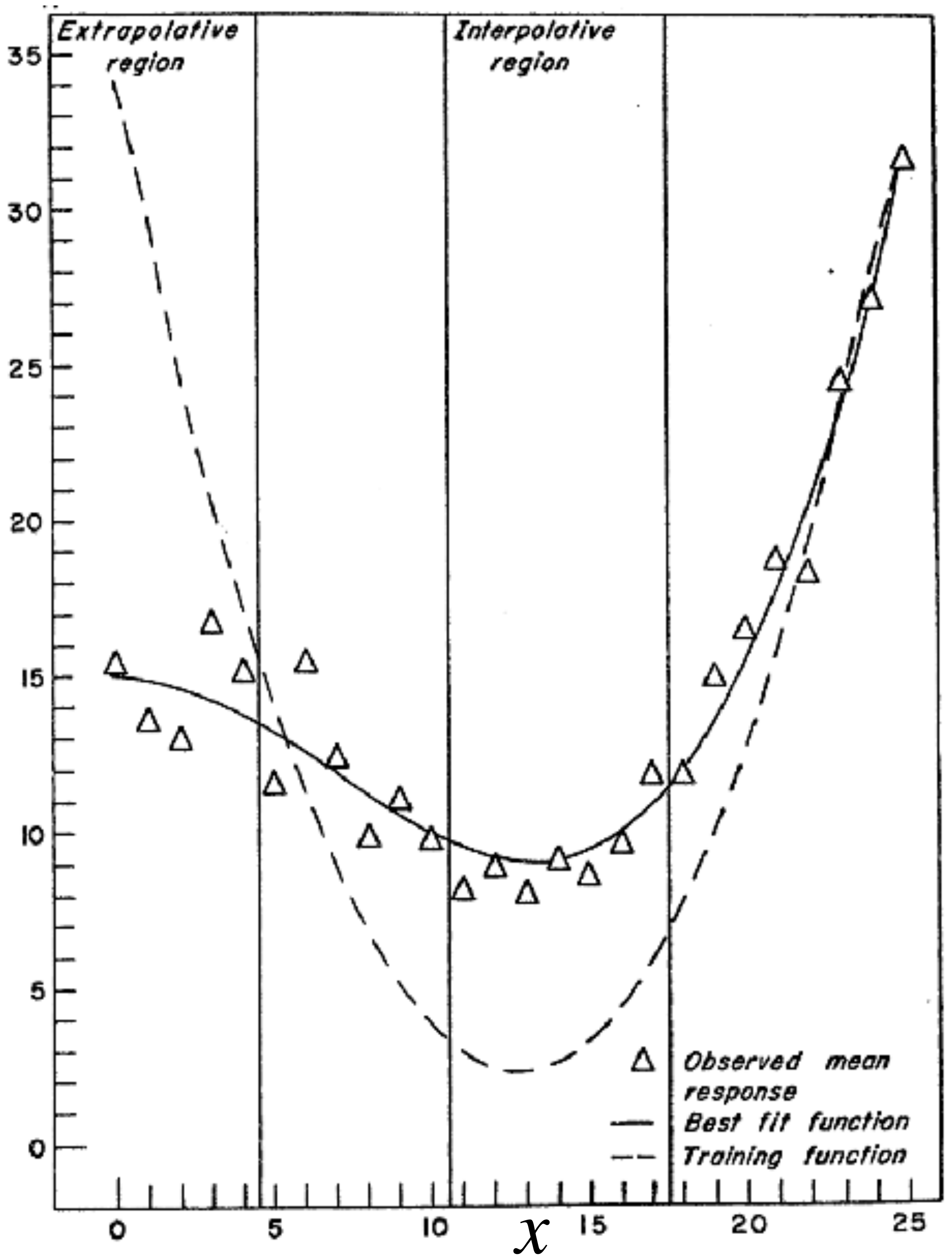


Fig. 4. Plot of mean response against stimulus for subject #20, condition III.

Linear regression

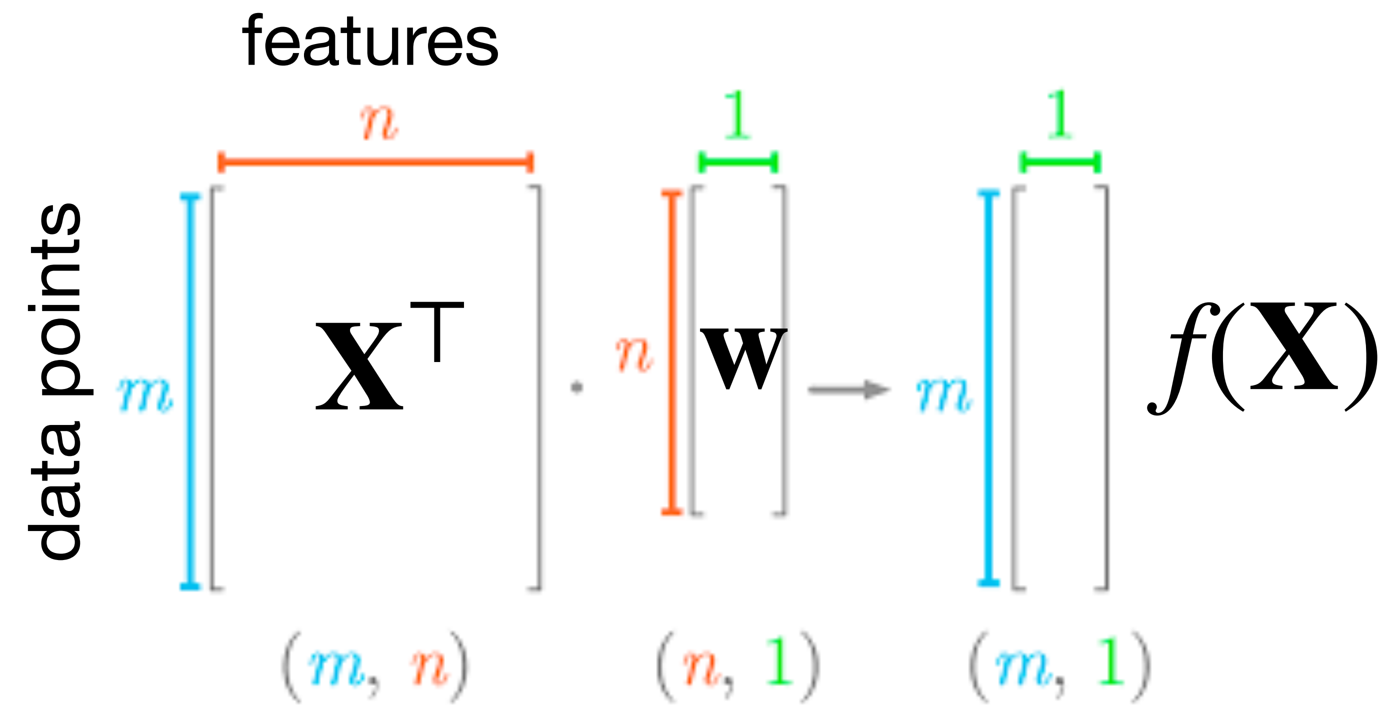
- Standard formulation assuming noisy observations
  $\mathbf{y} = f(\mathbf{X}) + \epsilon$ where $\epsilon \sim \mathcal{N}(0, \sigma_n^2)$ is i.i.d. noise
- **Linear assumption** — we can model the function as features x weights: $f(\mathbf{X}) = \mathbf{X}^\top \mathbf{w}$
  (this simplified notations appends a 1 to each $\mathbf{x}$ so that one of the weights in $\mathbf{w}$ is the intercept)
- **Maximum Likelihood Estimation (MLE)**
  - MLE of weights can be found by minimizing the Residual Sum of Squares (RSS):

    $$RSS(\mathbf{w}) = \sum_{i}^{n} (y_i - \hat{y}_i)^2 = \|\mathbf{y} - \mathbf{X}^\top \mathbf{w}\|^2$$

  - An analytic solution is available through the Moore-Penrose psuedoinverse (Penrose, 1955): $\hat{\mathbf{w}} = \left(\mathbf{X}^\top \mathbf{X}\right)^{-1} \mathbf{X}^\top \mathbf{y}$

Linear regression

- Standard formulation assuming noisy observations
  $\mathbf{y} = f(\mathbf{X}) + \epsilon$ where $\epsilon \sim \mathcal{N}(0, \sigma_n^2)$ is i.i.d. noise
- **Linear assumption** — we can model the function as features x weights: $f(\mathbf{X}) = \mathbf{X}^\top \mathbf{w}$
  (this simplified notations appends a 1 to each $\mathbf{x}$ so that one of the weights in $\mathbf{w}$ is the intercept)
- **Maximum Likelihood Estimation (MLE)**
  - MLE of weights can be found by minimizing the Residual Sum of Squares (RSS):
    $$RSS(\mathbf{w}) = \sum_{i}^{n} (y_i - \hat{y}_i)^2 = ||\mathbf{y} - \mathbf{X}^\top \mathbf{w}||^2$$
  - An analytic solution is available through the Moore-Penrose psuedoinverse (Penrose, 1955): $\hat{\mathbf{w}} = \left(\mathbf{X}^\top \mathbf{X}\right)^{-1} \mathbf{X}^\top \mathbf{y}$

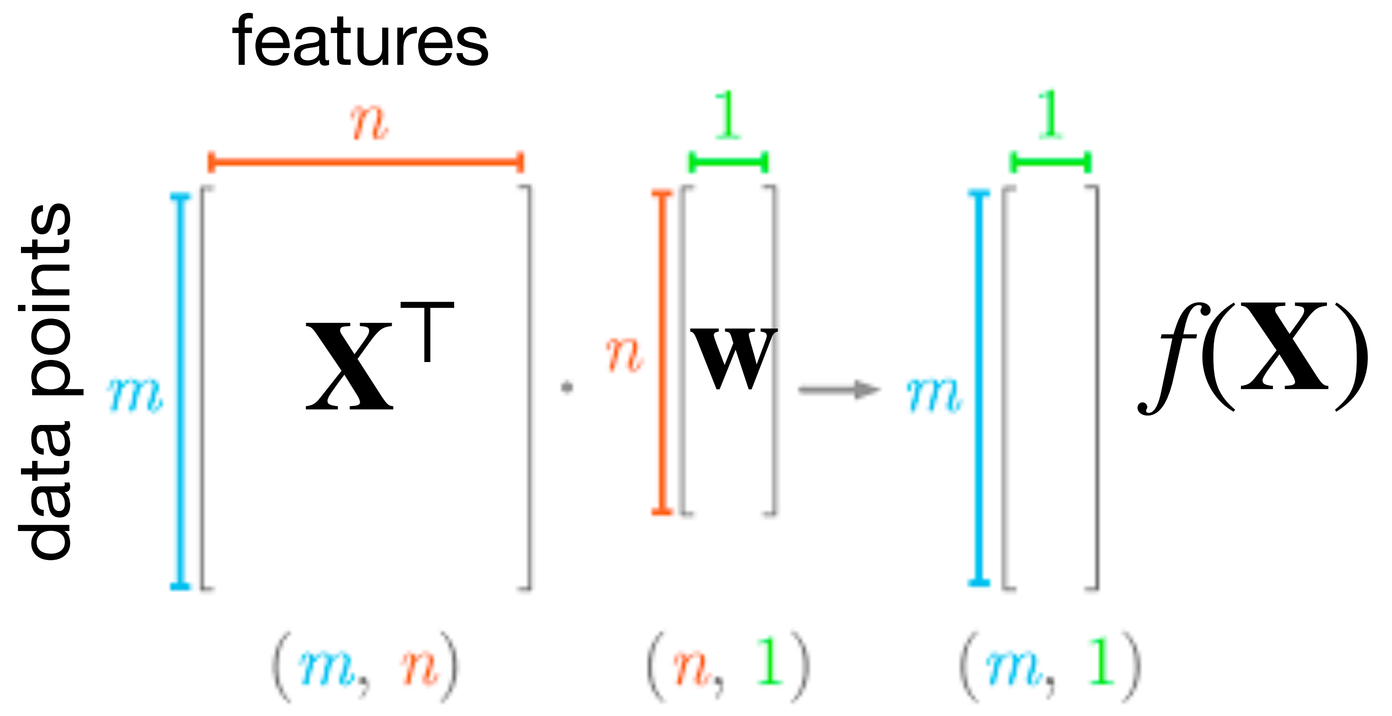


Linear regression

- Standard formulation assuming noisy observations
  $\mathbf{y} = f(\mathbf{X}) + \epsilon$ where $\epsilon \sim \mathcal{N}(0, \sigma_n^2)$ is i.i.d. noise
- **Linear assumption** — we can model the function as features x weights: $f(\mathbf{X}) = \mathbf{X}^\top \mathbf{w}$
  (this simplified notations appends a 1 to each $\mathbf{x}$ so that one of the weights in $\mathbf{w}$ is the intercept)
- **Maximum Likelihood Estimation (MLE)**
  - MLE of weights can be found by minimizing the Residual Sum of Squares (RSS):
    $$RSS(\mathbf{w}) = \sum_i^n (y_i - \hat{y}_i)^2 = \|\mathbf{y} - \mathbf{X}^\top \mathbf{w}\|^2$$
  - An analytic solution is available through the Moore-Penrose psuedoinverse (Penrose, 1955): $\hat{\mathbf{w}} = \left(\mathbf{X}^\top \mathbf{X}\right)^{-1} \mathbf{X}^\top \mathbf{y}$

Extension to Bayesian framework (not on the exam!)

- Gaussian prior on weights: $P(\mathbf{w}) \sim \mathcal{N}(0, \Sigma)$
- Gaussian likelihood: $P(\mathbf{y} | \mathbf{X}, \mathbf{w}) = \prod_i^n P(y_i | \mathbf{x}_i, \mathbf{w})$
  $= \mathcal{N}(\mathbf{y} | \mathbf{X}^\top \mathbf{w}, \sigma_n^2 \mathbf{I})$
- Apply Bayes theorem to estimate the weights:
  $P(\mathbf{w} | \mathbf{X}, \mathbf{y}) \propto P(\mathbf{y} | \mathbf{X}, \mathbf{w}) P(\mathbf{w})$
  $\sim \mathcal{N}(\hat{\mathbf{w}} | \frac{1}{\sigma_n^2} \mathbf{A}^{-1} \mathbf{X} \mathbf{y}, \mathbf{A}^{-1})$
  where $\mathbf{A} = \sigma_n^{-2} \mathbf{X} \mathbf{X}^\top + \Sigma^{-1}$



Linear regression

- Standard formulation assuming noisy observations
  $\mathbf{y} = f(\mathbf{X}) + \epsilon$ where $\epsilon \sim \mathcal{N}(0,\sigma_n^2)$ is i.i.d. noise
- **Linear assumption** — we can model the function as features x weights: $f(\mathbf{X}) = \mathbf{X}^\top \mathbf{w}$
  (this simplified notations appends a 1 to each $\mathbf{x}$ so that one of the weights in $\mathbf{w}$ is the intercept)
- **Maximum Likelihood Estimation (MLE)**
  - MLE of weights can be found by minimizing the Residual Sum of Squares (RSS):

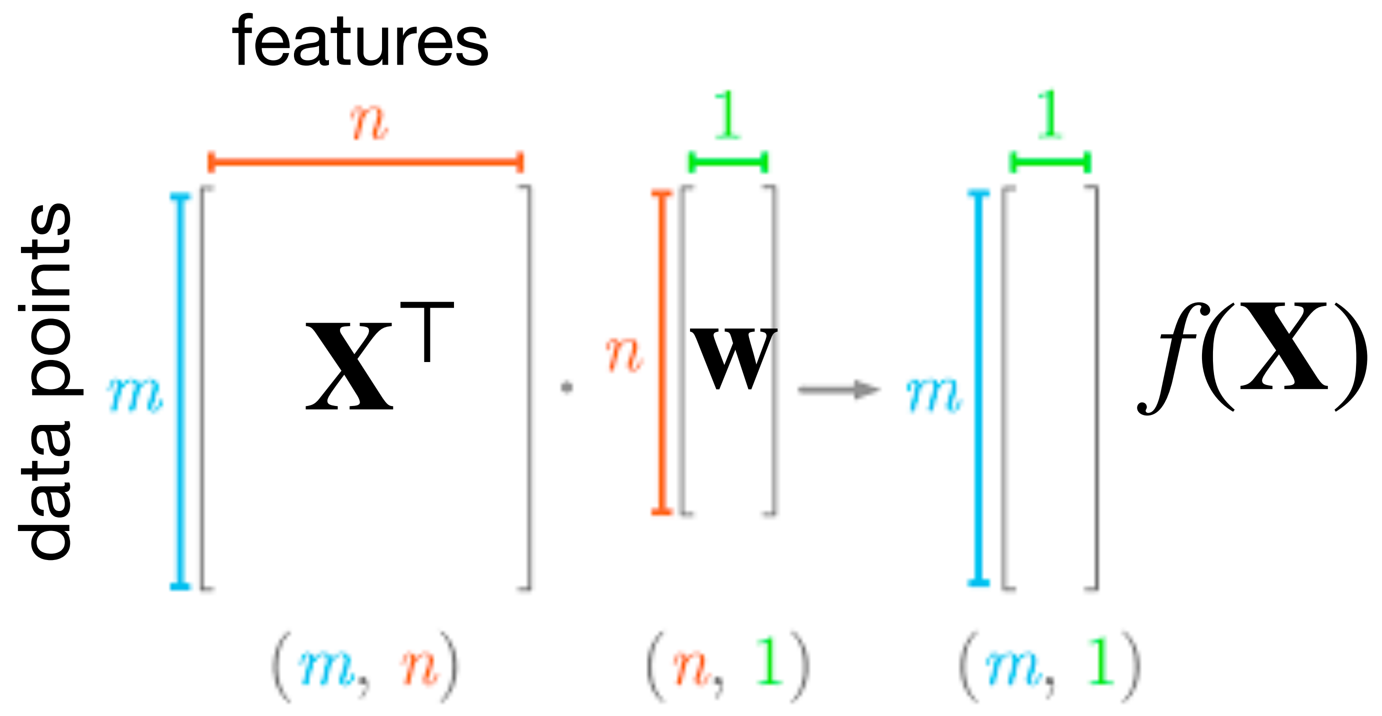
$$RSS(\mathbf{w}) = \sum_i^n (y_i - \hat{y}_i)^2 = \|\mathbf{y} - \mathbf{X}^\top \mathbf{w}\|^2$$

estimate ($\hat{y}_i$), observed (y_i)

  - An analytic solution is available through the Moore-Penrose psuedoinverse (Penrose, 1955): $\hat{\mathbf{w}} = \left(\mathbf{X}^\top \mathbf{X}\right)^{-1} \mathbf{X}^\top \mathbf{y}$

Extension to Bayesian framework (not on the exam!)

- Gaussian prior on weights: $P(\mathbf{w}) \sim \mathcal{N}(0,\Sigma)$
- Gaussian likelihood: $P(\mathbf{y} \mid \mathbf{X}, \mathbf{w}) = \prod_i^n P(y_i \mid \mathbf{x}_i, \mathbf{w})$
  $= \mathcal{N}(\mathbf{y} \mid \mathbf{X}^\top \mathbf{w}, \sigma_n^2 \mathbf{I})$
- Apply Bayes theorem to estimate the weights:
  $P(\mathbf{w} \mid \mathbf{X}, \mathbf{y}) \propto P(\mathbf{y} \mid \mathbf{X}, \mathbf{w}) P(\mathbf{w})$
  $\sim \mathcal{N}(\hat{\mathbf{w}} \mid \frac{1}{\sigma_n^2} \mathbf{A}^{-1} \mathbf{X}\mathbf{y}, \mathbf{A}^{-1})$
  where $\mathbf{A} = \sigma_n^{-2} \mathbf{X}\mathbf{X}^\top + \Sigma^{-1}$



Linear regression

- Standard formulation assuming noisy observations
  $\mathbf{y} = f(\mathbf{X}) + \epsilon$ where $\epsilon \sim \mathcal{N}(0, \sigma_n^2)$ is i.i.d. noise
- **Linear assumption** — we can model the function as features x weights: $f(\mathbf{X}) = \mathbf{X}^\top \mathbf{w}$
  (this simplified notations appends a 1 to each $\mathbf{x}$ so that one of the weights in $\mathbf{w}$ is the intercept)
- **Maximum Likelihood Estimation (MLE)**
  - MLE of weights can be found by minimizing the Residual Sum of Squares (RSS):

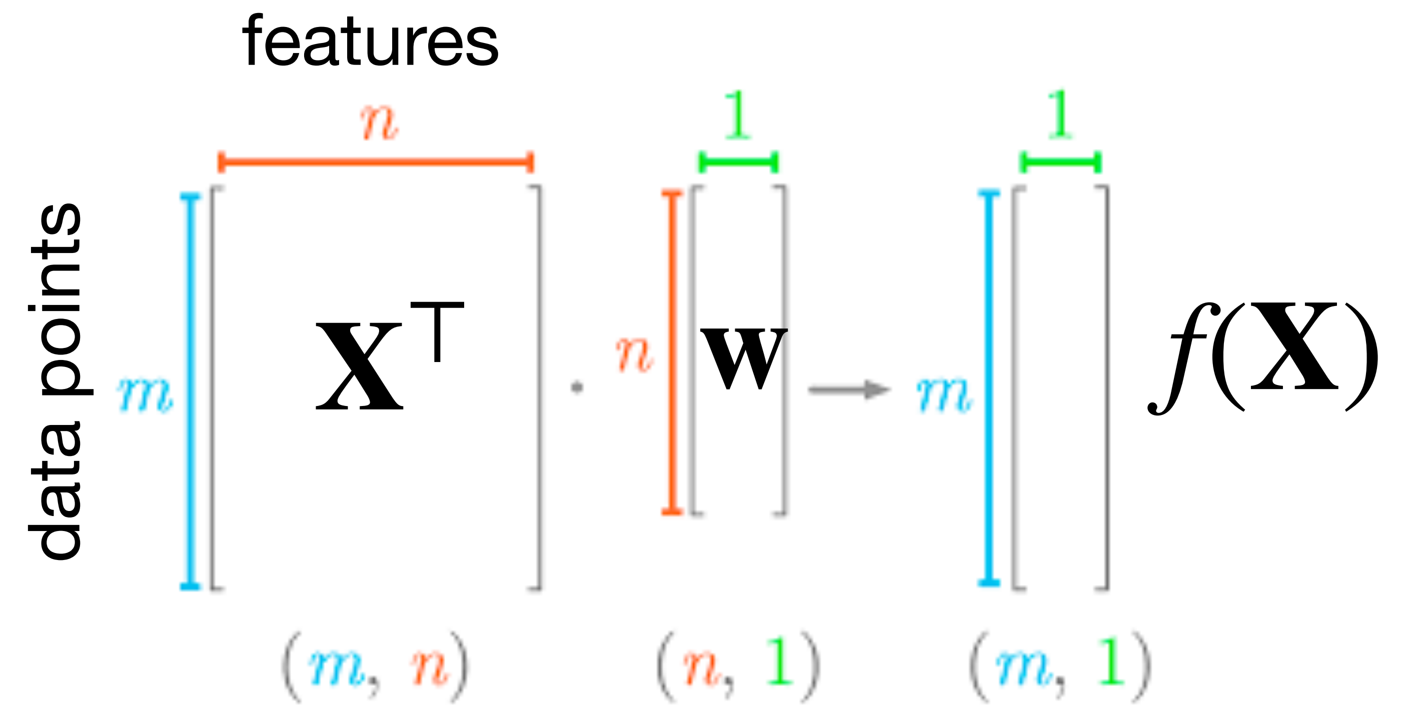
$$RSS(\mathbf{w}) = \sum_i^n (y_i - \hat{y}_i)^2 = ||\mathbf{y} - \mathbf{X}^\top \mathbf{w}||^2$$

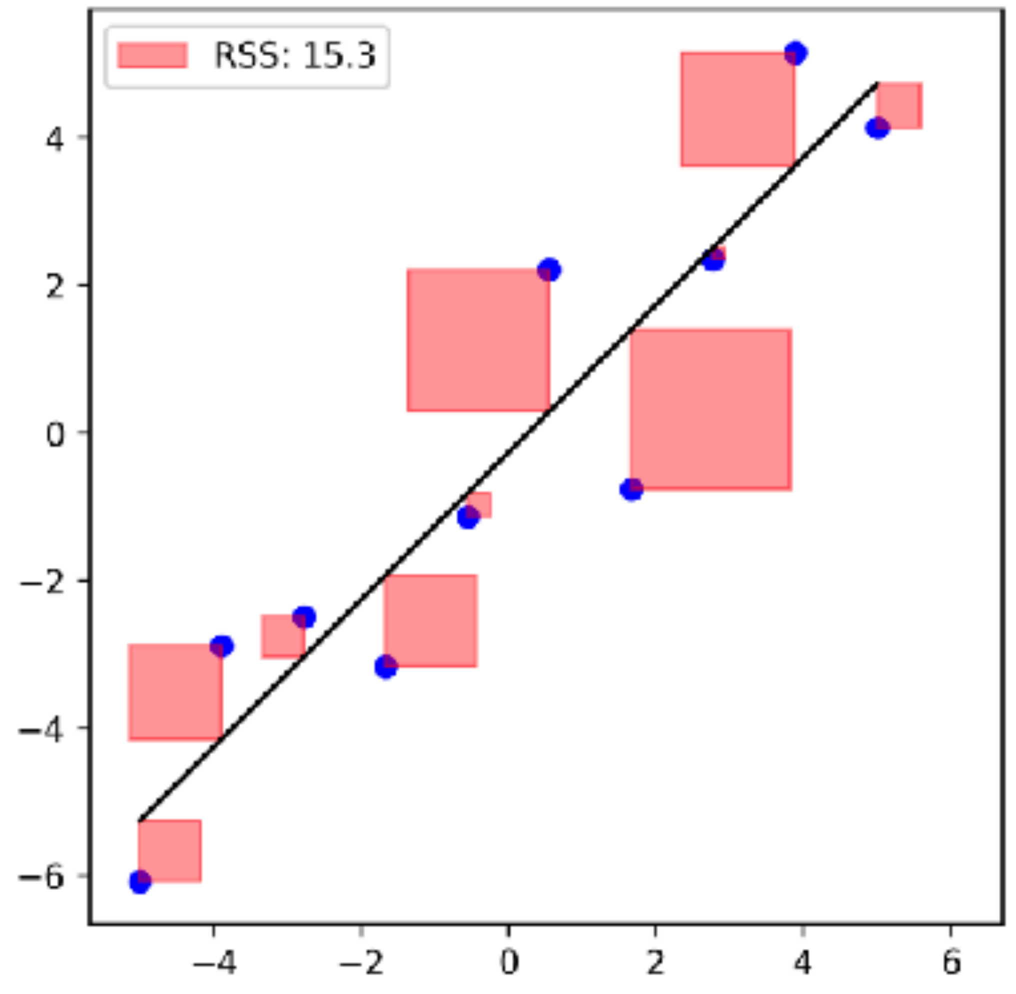
estimate ($\hat{y}_i$), observed (y_i)

  - An analytic solution is available through the Moore-Penrose psuedoinverse (Penrose, 1955): $\hat{\mathbf{w}} = \left(\mathbf{X}^\top \mathbf{X}\right)^{-1} \mathbf{X}^\top \mathbf{y}$

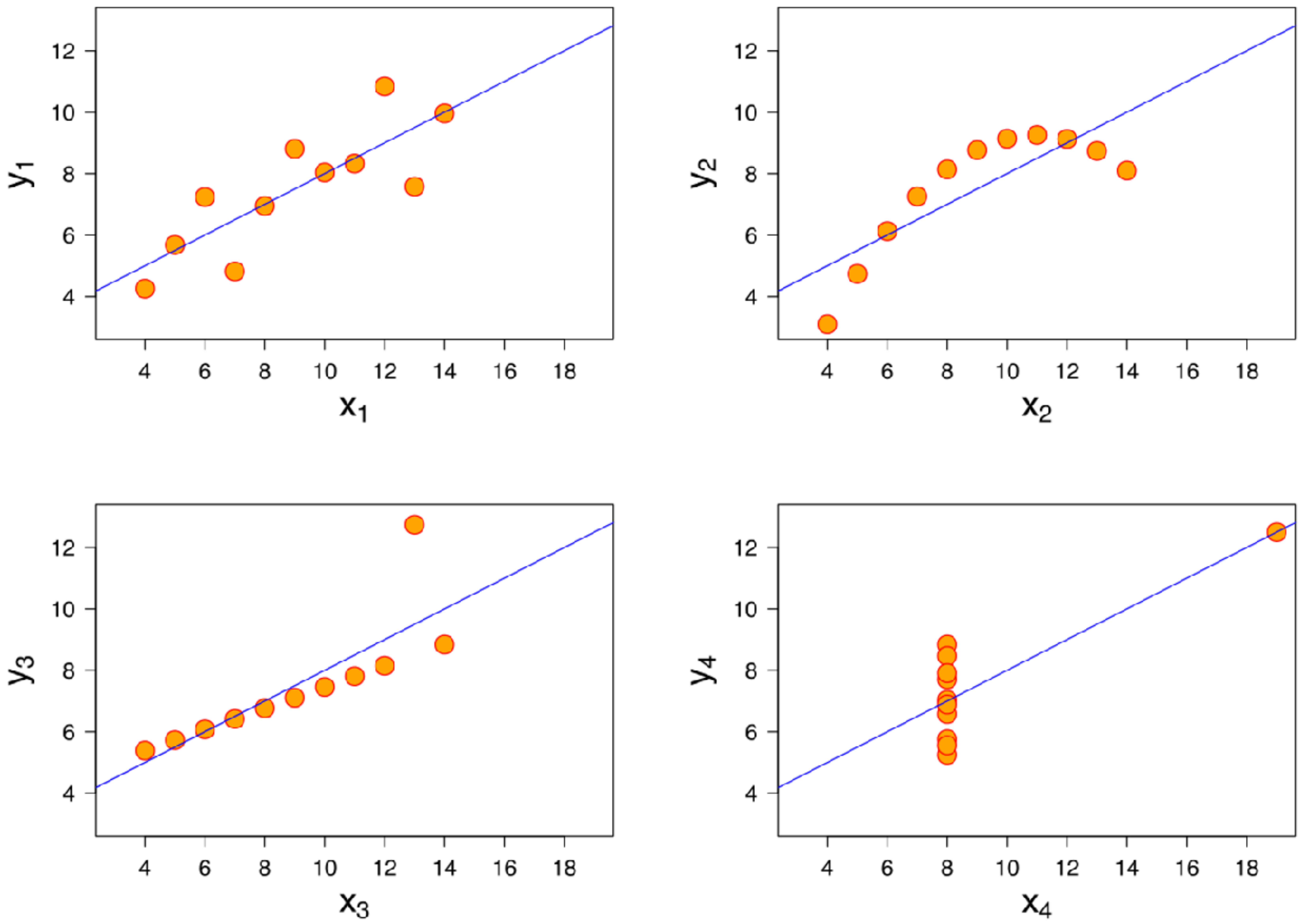
Extension to Bayesian framework (not on the exam!)

- Gaussian prior on weights: $P(\mathbf{w}) \sim \mathcal{N}(0, \Sigma)$
- Gaussian likelihood: $P(\mathbf{y} | \mathbf{X}, \mathbf{w}) = \prod_i^n P(y_i | \mathbf{x}_i, \mathbf{w})$
  $= \mathcal{N}(\mathbf{y} | \mathbf{X}^\top \mathbf{w}, \sigma_n^2 \mathbf{I})$
- Apply Bayes theorem to estimate the weights:
  $P(\mathbf{w} | \mathbf{X}, \mathbf{y}) \propto P(\mathbf{y} | \mathbf{X}, \mathbf{w}) P(\mathbf{w})$
  $\sim \mathcal{N}(\hat{\mathbf{w}} | \frac{1}{\sigma_n^2} \mathbf{A}^{-1} \mathbf{X} \mathbf{y}, \mathbf{A}^{-1})$
  where $\mathbf{A} = \sigma_n^{-2} \mathbf{X} \mathbf{X}^\top + \Sigma^{-1}$

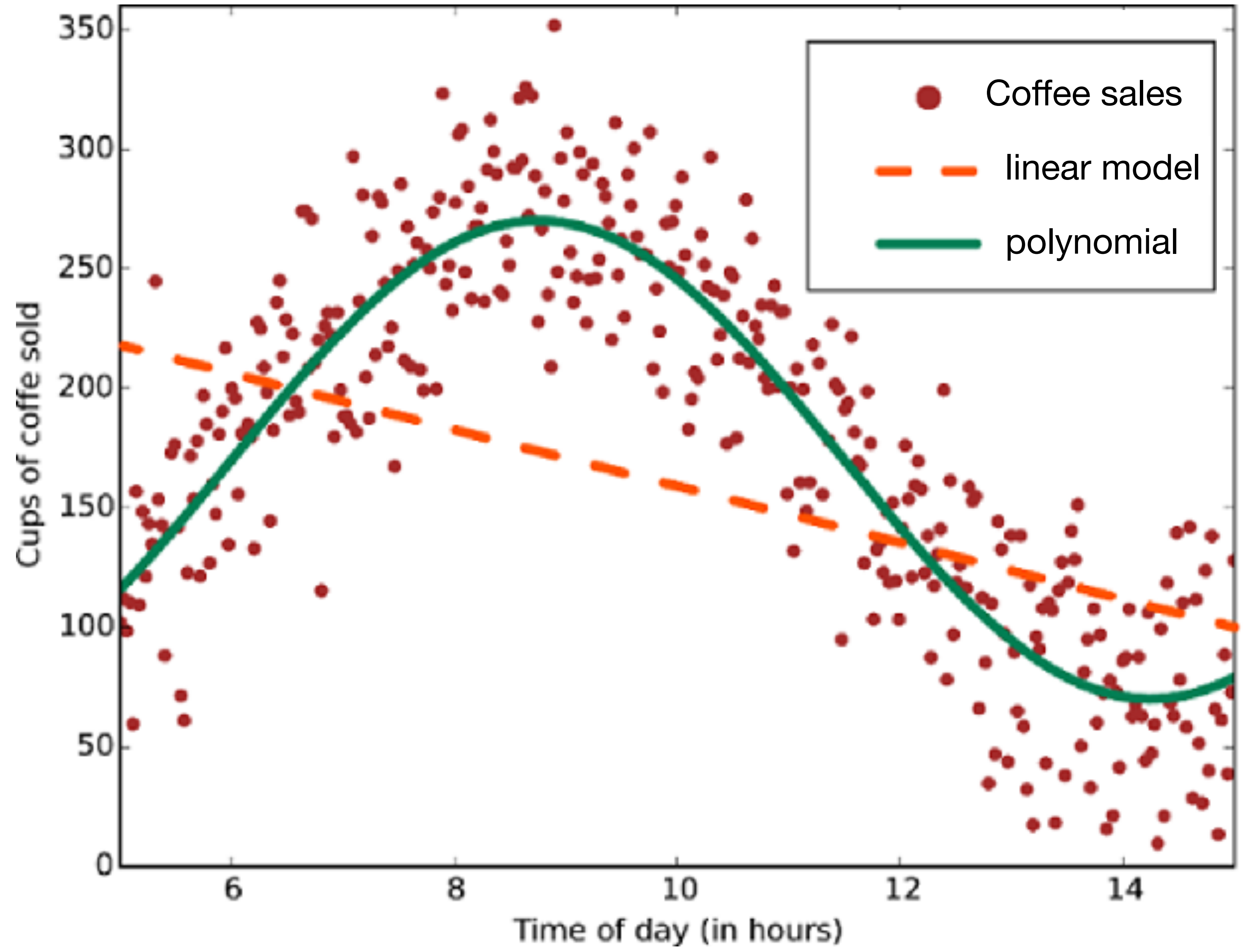




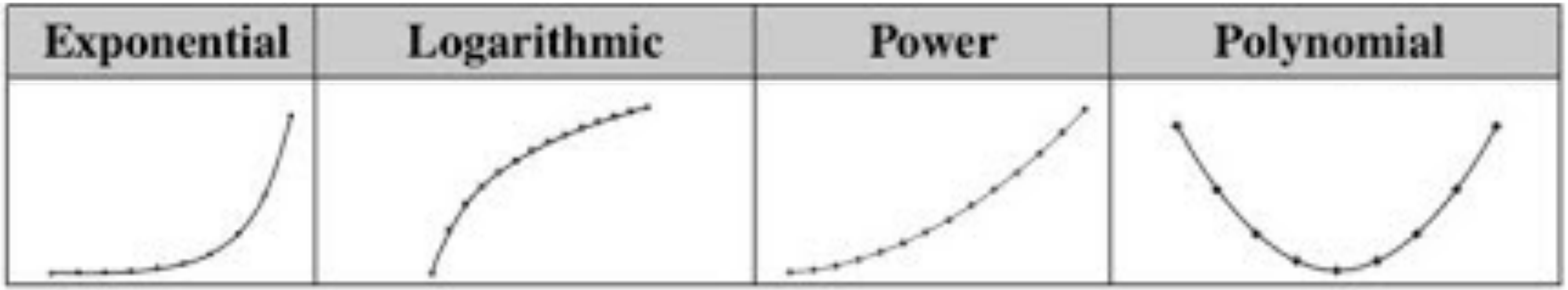
Linear assumptions don't always work



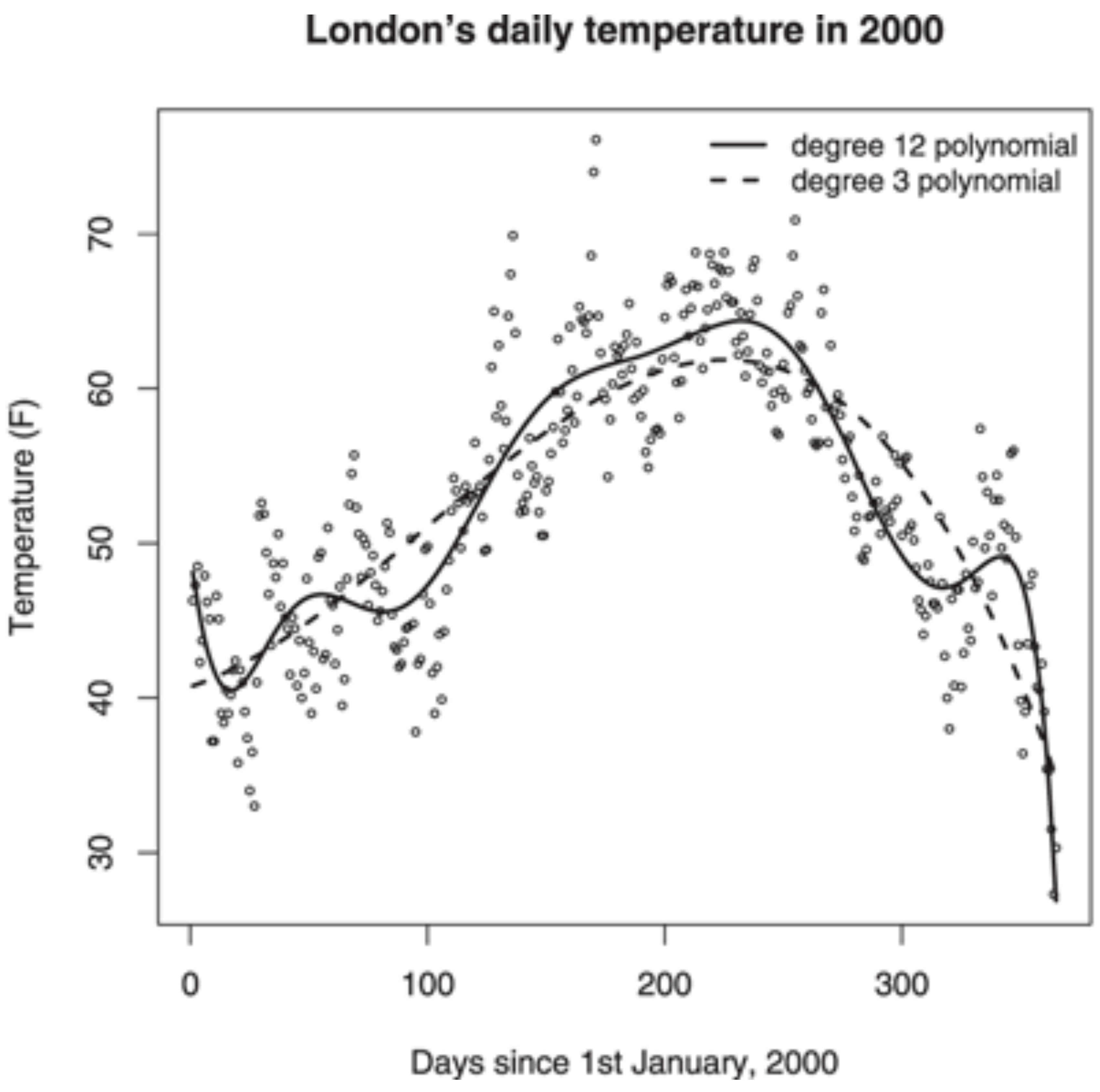
Parametric regression

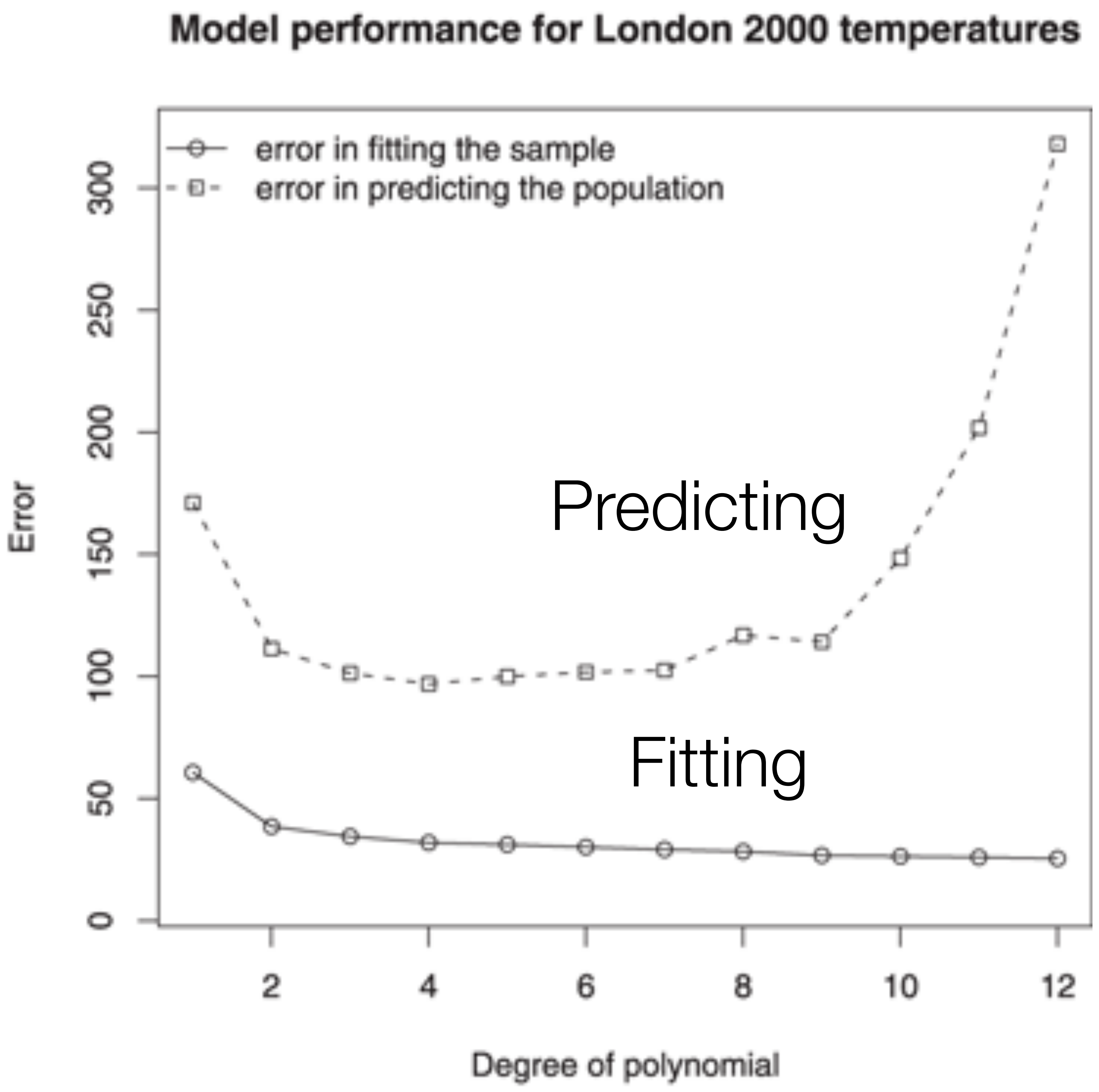


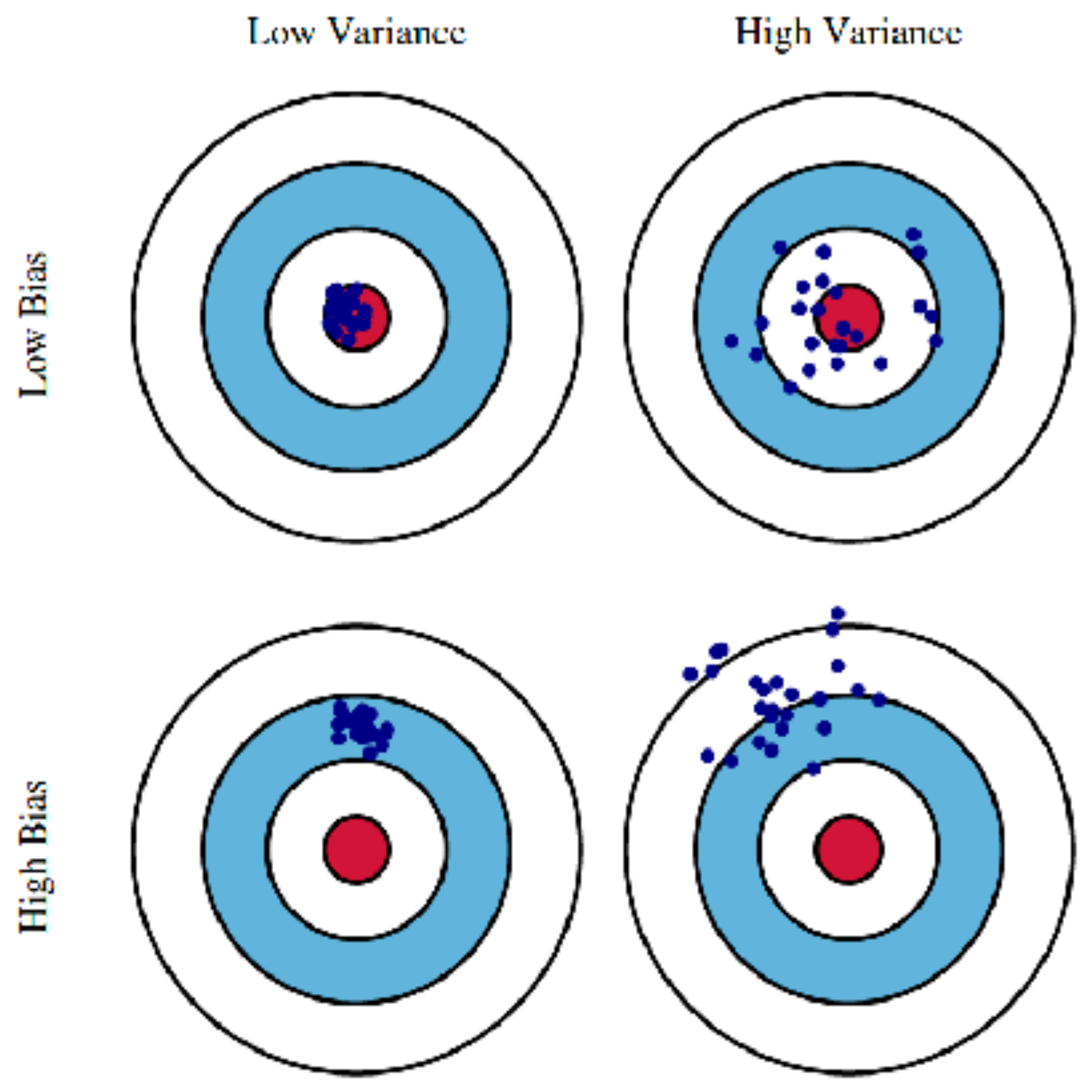
- Rather than assuming a linear relationship, assume a different functional form
  - Exponential: $f(\mathbf{x}) = \mathbf{w}^{\mathbf{x}}$
  - Logarithmic: $f(\mathbf{x}) = \mathbf{w}\log(\mathbf{x})$
  - Power: $f(\mathbf{x}) = \mathbf{x}^{\mathbf{w}}$
  - Polynomial: $f(x) = w_i x^i + w_{i-1} x^{i-1} + \ldots + w_i x$ (switching to univariate x for simplicity)

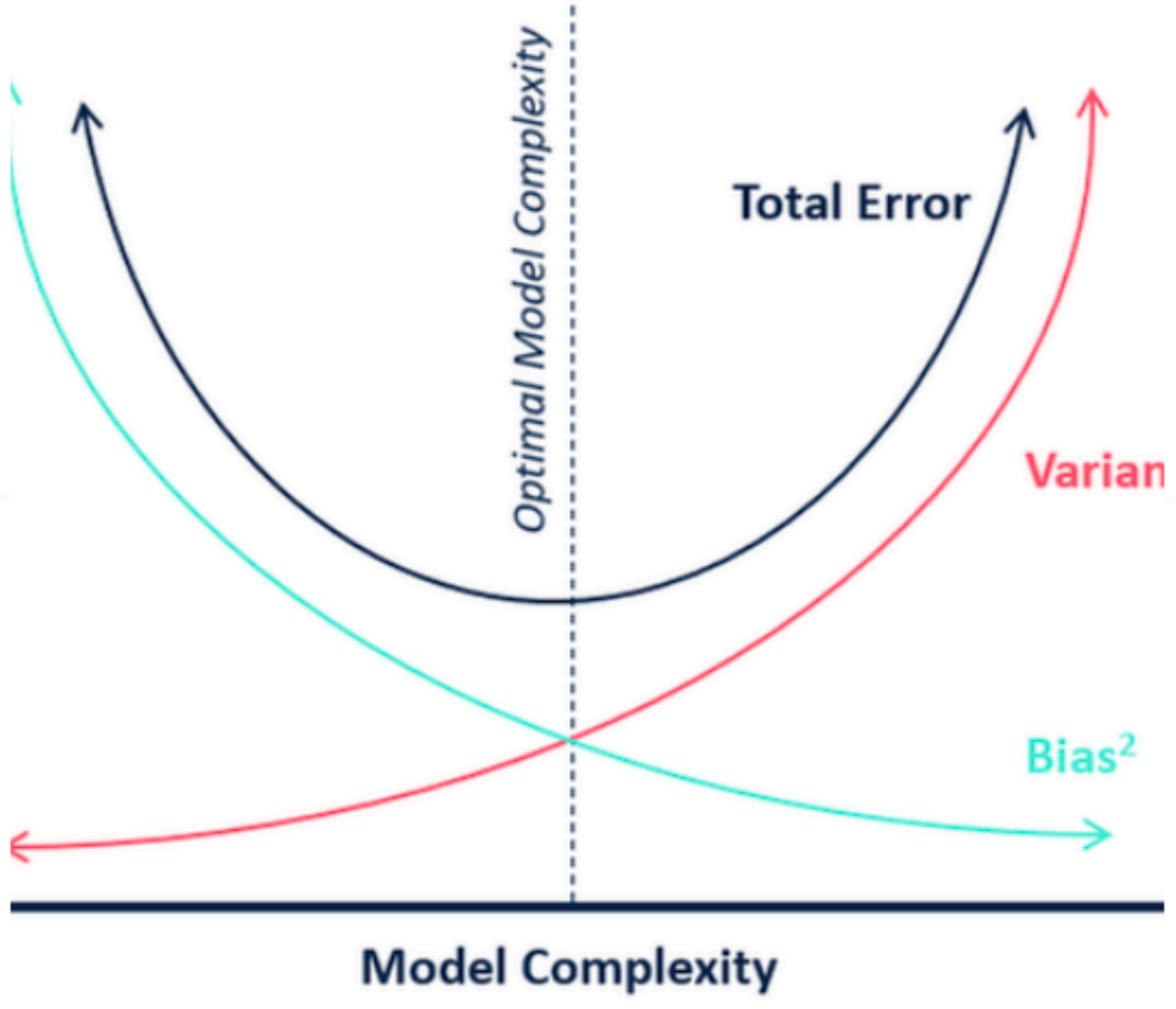


Bias-Variance trade-off









Gigerener & Brighton (*TopiCS*, 2009)

Similarity-based theories of function learning

- Associative learning model (**ALM**; Buseymeyer et al., 1997) used neural networks to encode the generic principle that *similar inputs produce similar outputs*
  - When stimuli x_* is presented, it activates all input nodes according to their similarity: $a_i(x_*) = \exp\left[-\gamma(x_* - x_i)^2\right]$ where γ is a sensitivity parameter
  - Output node y_j is activated according to learned weights:

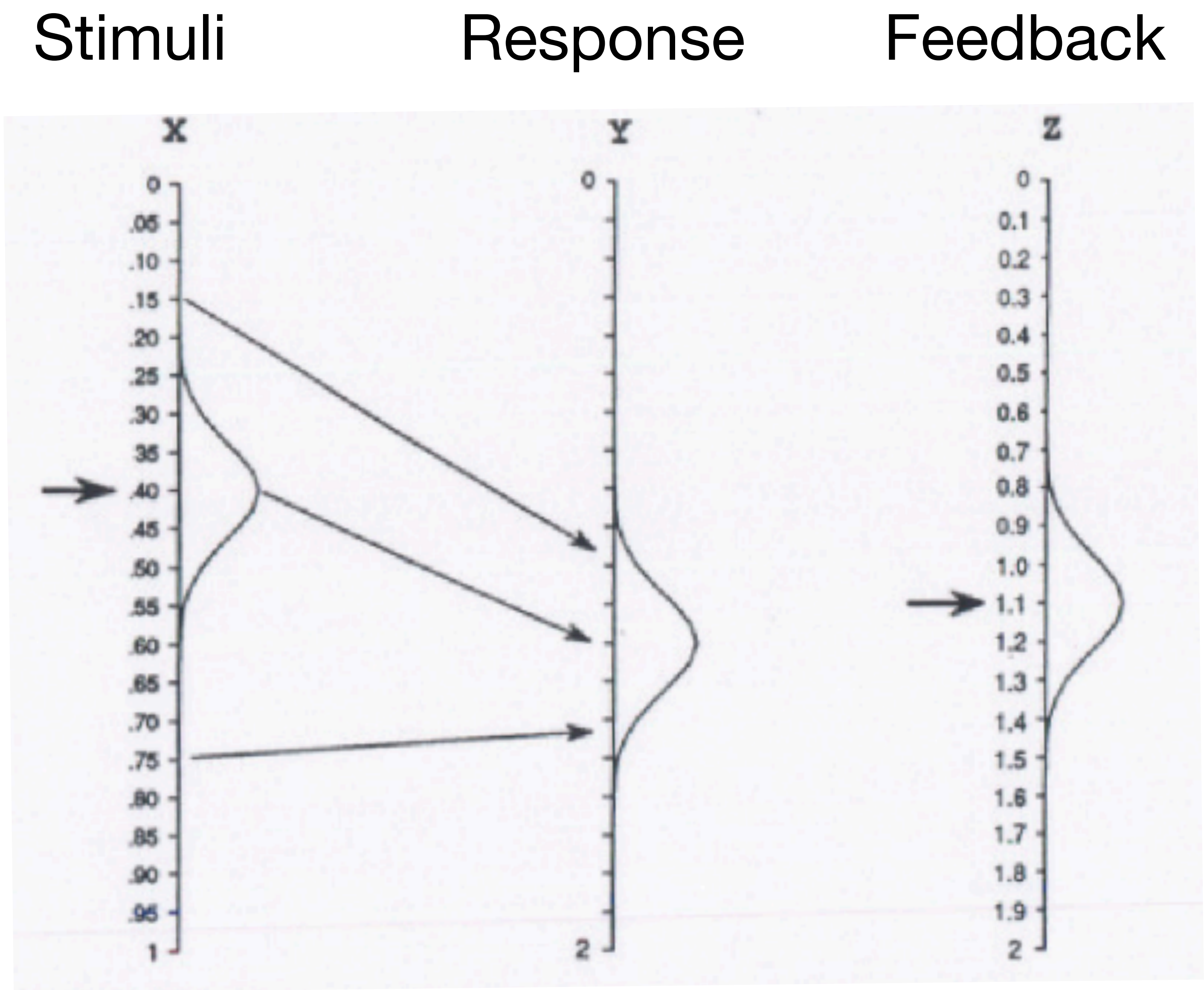
$$y_j(x_*) = \sum_{i}^{M} w_{ji} \cdot a_i(x_*)$$

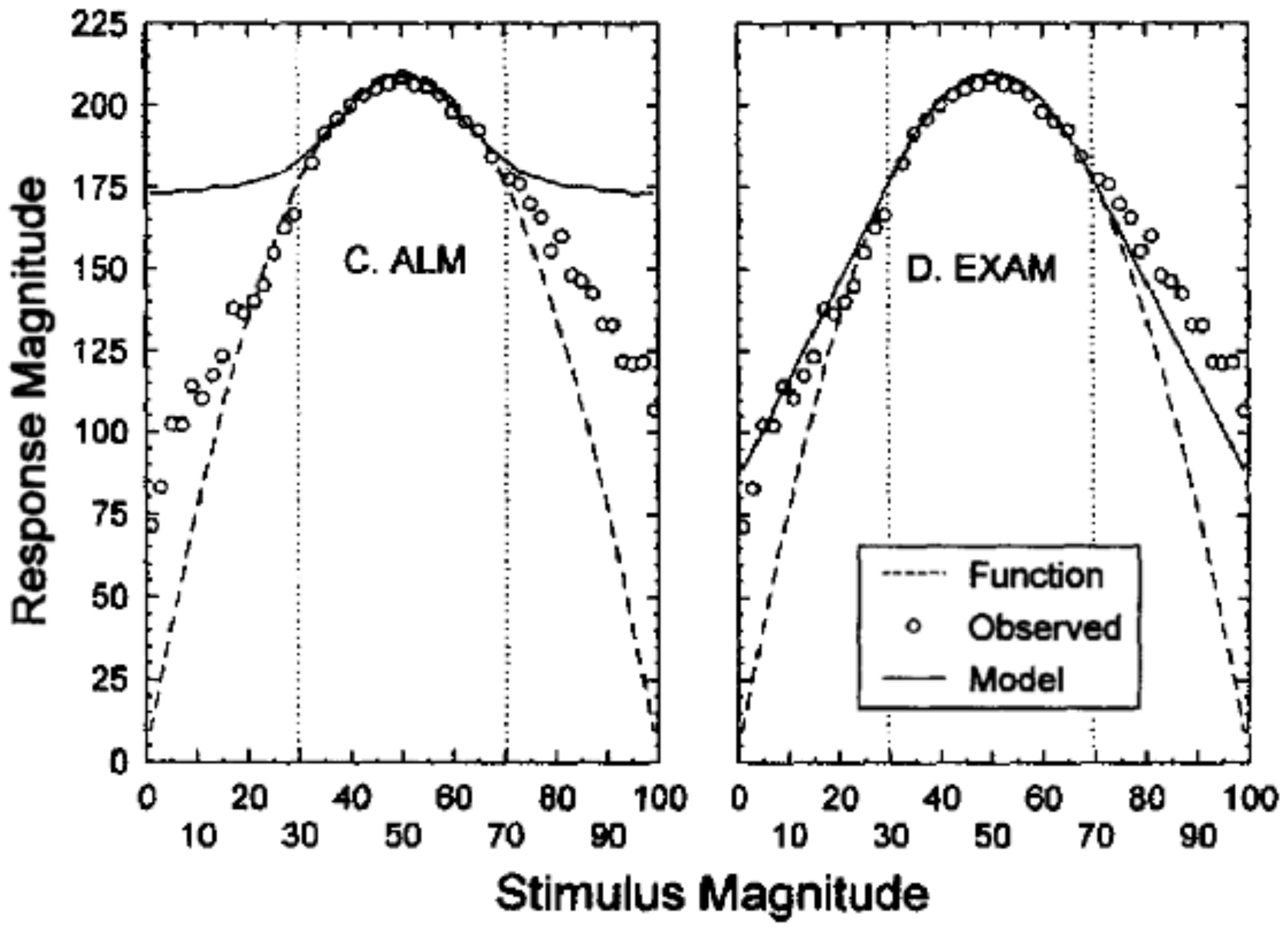
  - Weights are updated using the delta-rule based on feedback signal z:

$$w_{ji} \leftarrow w_{ji} + \alpha \left[f_j(z) - y_j(x_*)\right] a_i(x_*)$$

where $f_j(z) = \exp\left[-\gamma(z - y_j)^2\right]$

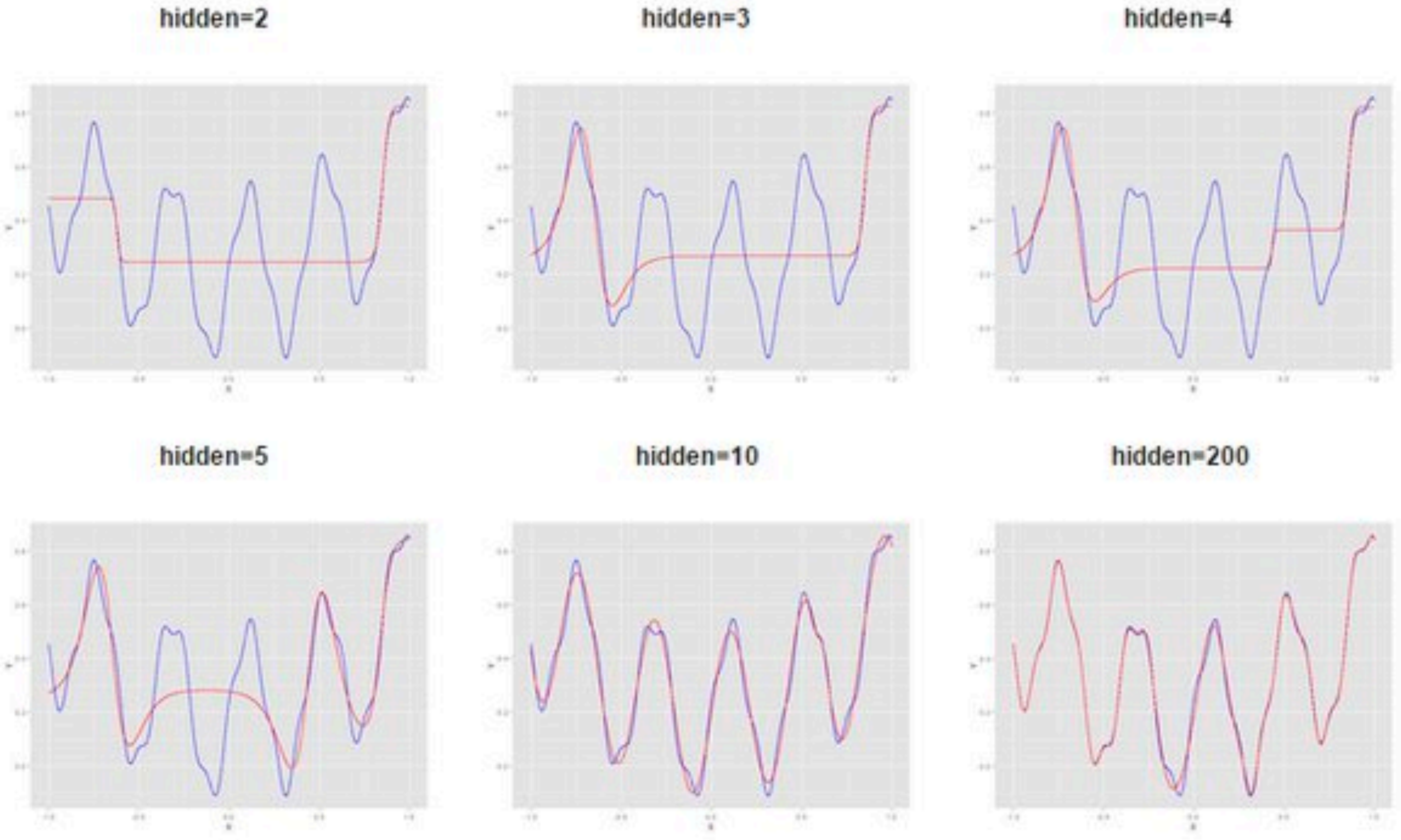
  - Limitation: fails to capture human extrapolation patterns —>
- Extrapolation-Association Model (**EXAM**; Delosh et al,. 1997) extends ALM by adding a linear approximation of ALM outputs to account for more linear extrapolation patterns in humans
  - But humans also sometimes extrapolate in a non-linear fashion (Bott & Heit, 2004)





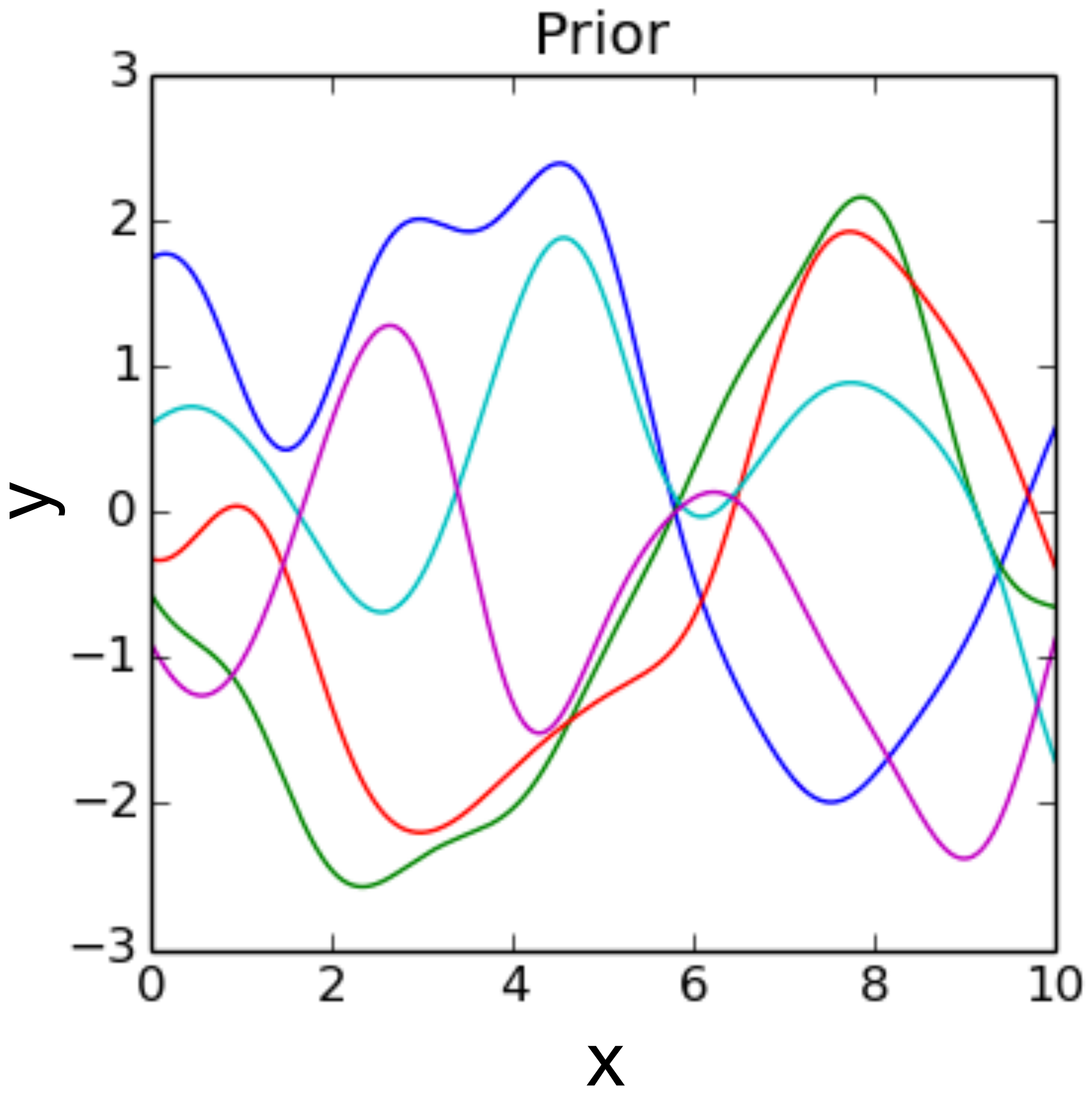
Neural networks as Universal Function Approximators

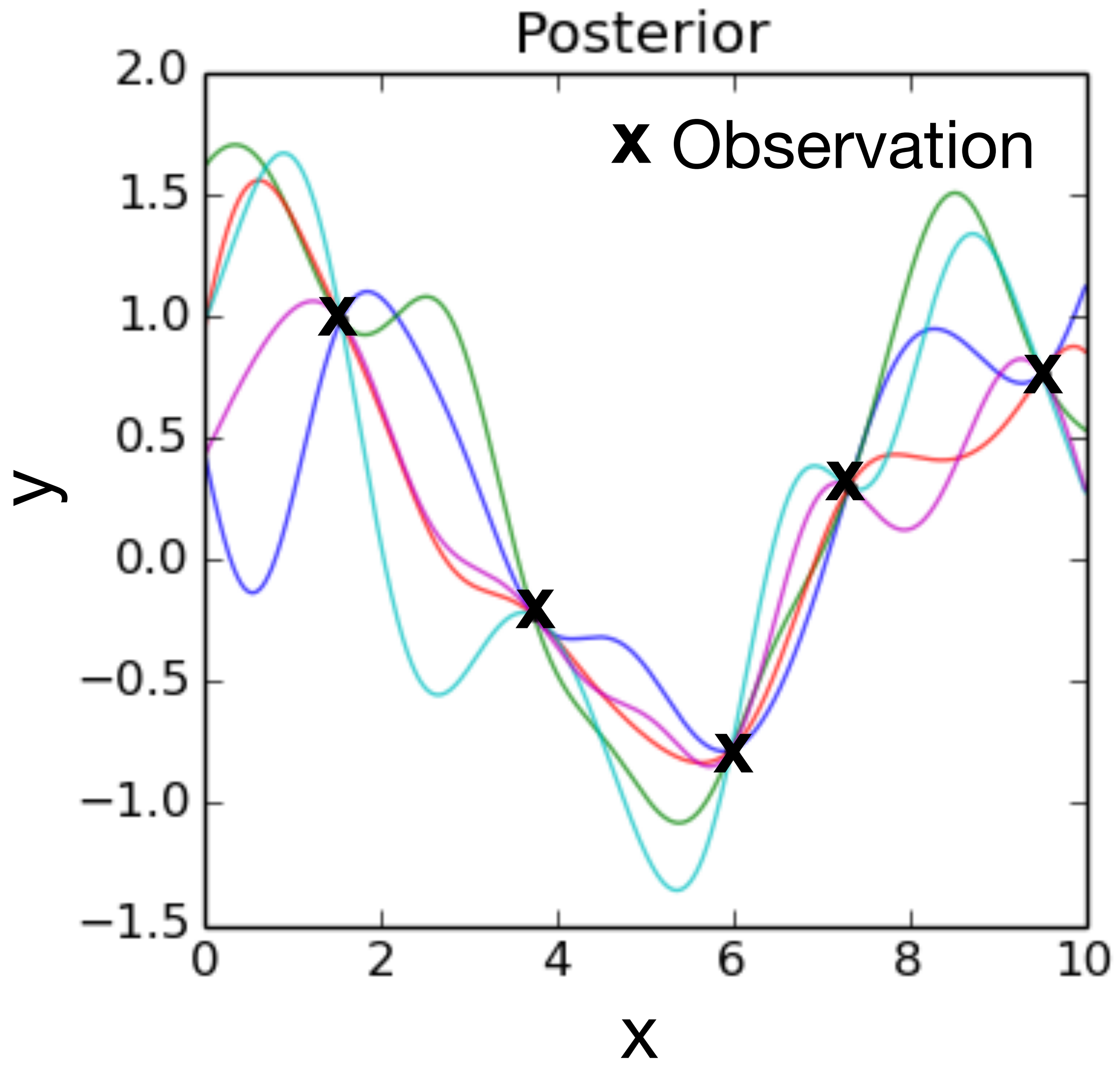
- Recall Cybenko (1989): Every continuous function can be approximated arbitrarily closely by an MLP with just a single hidden layer
  - adding more nodes in the hidden layer increases the representational capacity of the network
- But fitting is not the same as predicting
- As we see from ALM, extrapolation patterns of NNs don't always match the inductive biases of humans learners

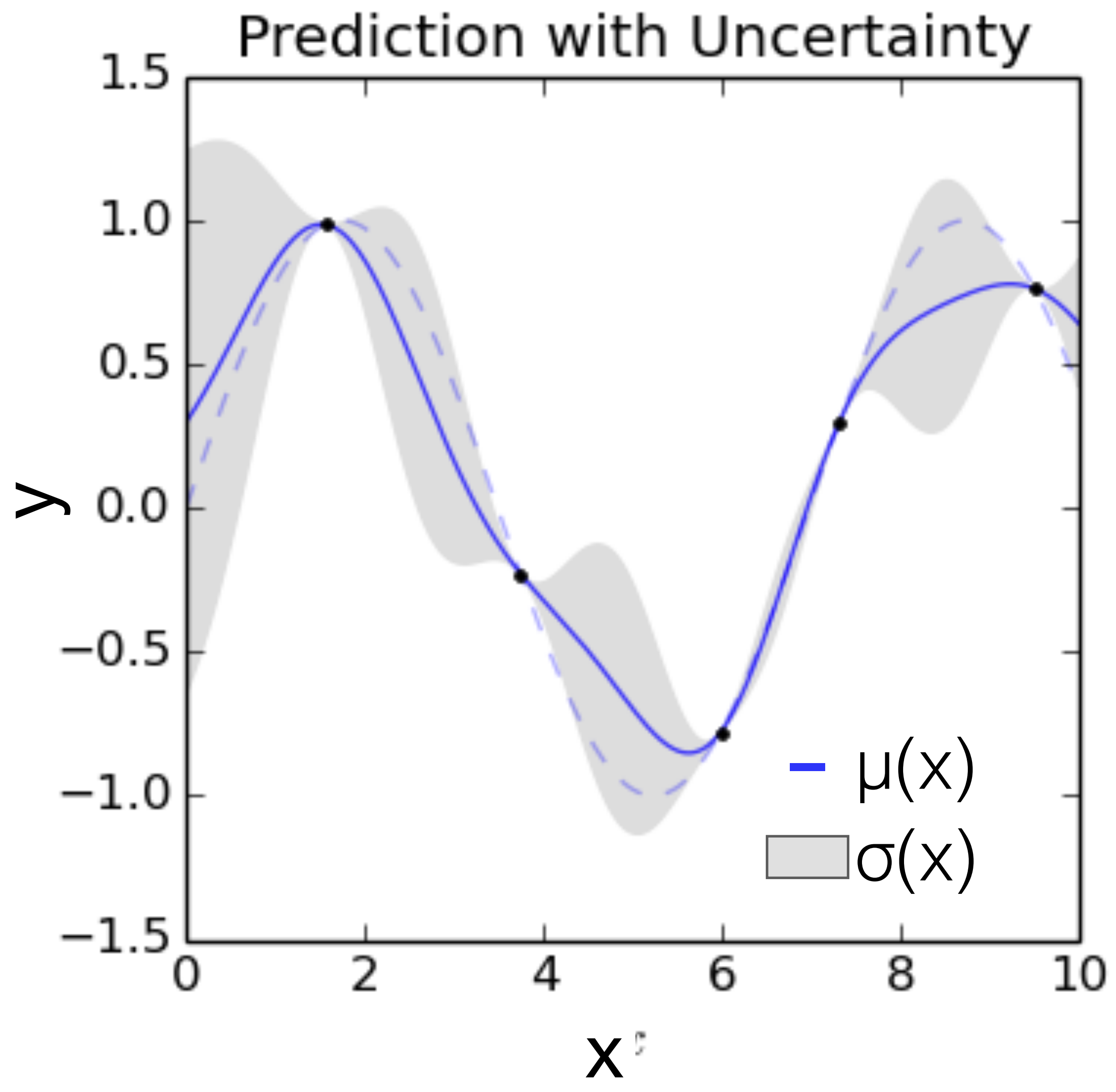


Gaussian Process (GP) regression as a hybrid model

- Bayesian framework for function learning
- Assumes a distribution over functions, where each function corresponds to a hypothesis about the relationship between x and y
- After conditioning on observations, we can make predictions (with uncertainty) about any point along the input space
- Called Gaussian process, because we make Gaussian assumptions
  - the posterior at each point is defined by a mean and variance (details on the next slide)
- GPs are a *non-parametric* model, meaning the complexity is defined by the data not the number of parameters in the chosen functional class (i.e., *parametric models*)

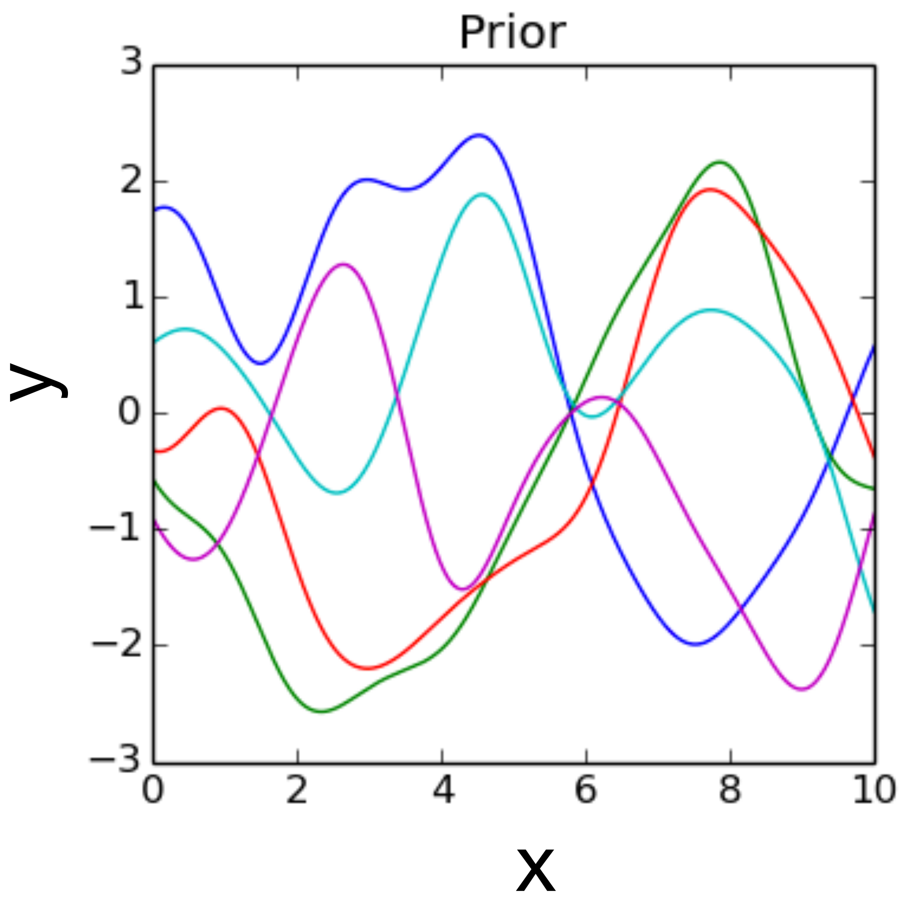






see Rasmussen & Williams (2005) for further details

Gaussian Process (GP) regression in detail

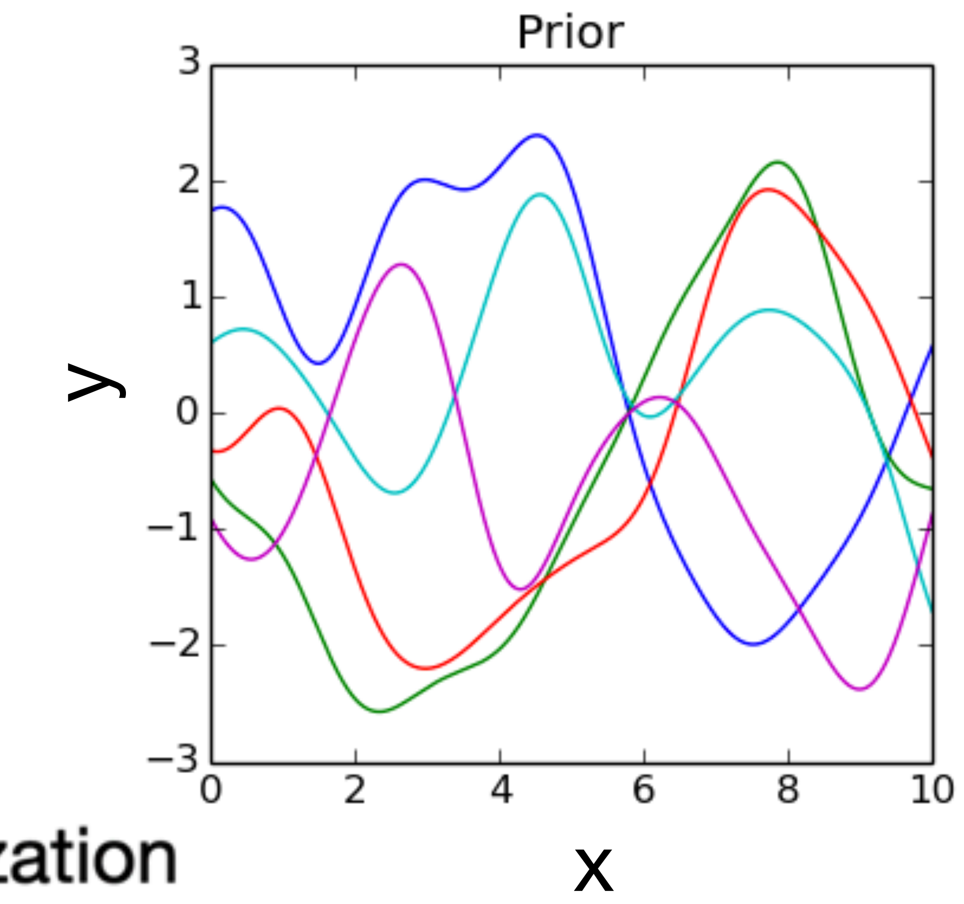


- Prior over functions (i.e., hypotheses) is a multivariate Gaussian:
  $P(f) \sim \mathcal{GP}\left(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')\right)$
  - prior mean $m(\mathbf{x})$ is typically set to 0 without loss of generalization
  - Covariance $k(\mathbf{x}, \mathbf{x}')$ is defined by a choice of kernel
    e.g., RBF kernel:
    $$k(\mathbf{x}, \mathbf{x}') = \exp\left(\frac{-||\mathbf{x} - \mathbf{x}'||^2}{2\lambda^2}\right)$$
    where λ defines the expected smoothness of the function
- Once we acqire some data $\mathcal{D} = \{\mathbf{X}, \mathbf{y}\}$, we can compute a posterior prediction about any new datapoint $\mathbf{x}_*$ that is also Gaussian with mean and variance defined as

$$m(\mathbf{x}_* | \mathcal{D}) = \mathbf{k}_*^\top(\mathbf{K} + \sigma^2\mathbf{I})^{-1}\mathbf{y}$$
$$v(\mathbf{x}_* | \mathcal{D}) = k(\mathbf{x}_*, \mathbf{x}_*) - \mathbf{k}_*^\top(\mathbf{K} + \sigma^2\mathbf{I})^{-1}\mathbf{k}_*$$

see Rasmussen & Williams (2005) for further details

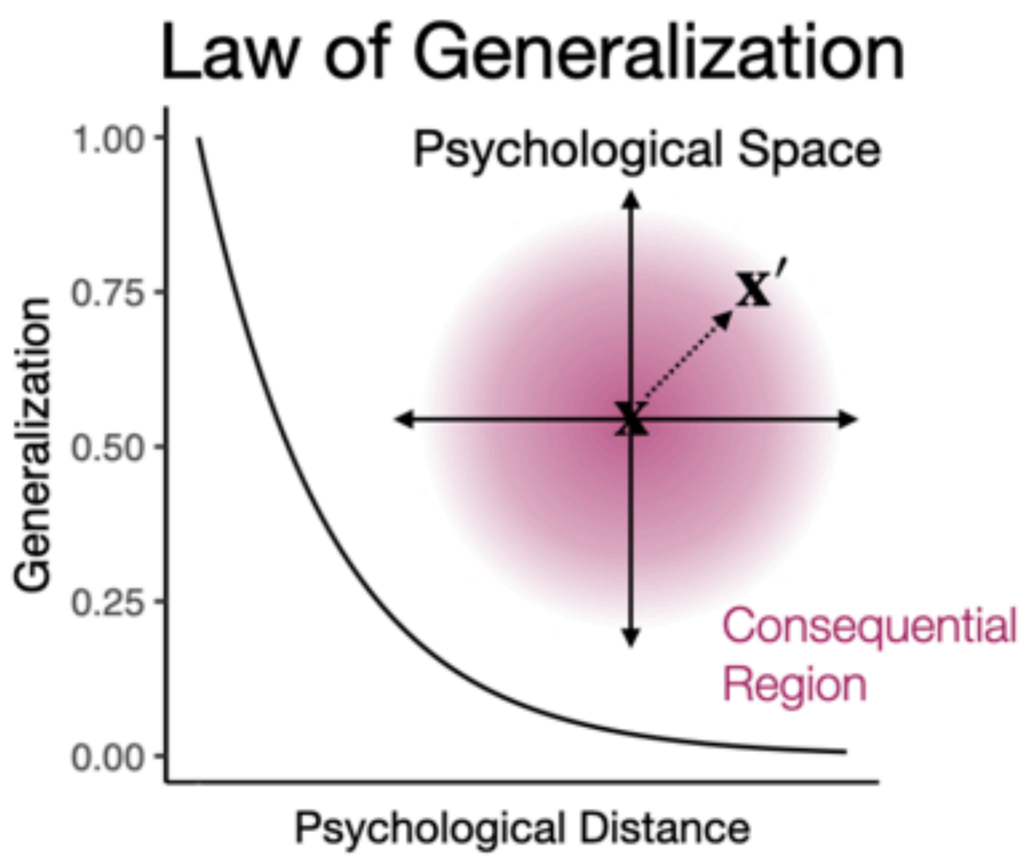
Gaussian Process (GP) regression in detail



- Prior over functions (i.e., hypotheses) is a multivariate Gaussian: $P(f) \sim \mathcal{GP}\left(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')\right)$
  - prior mean $m(\mathbf{x})$ is typically set to 0 without loss of generalization
  - Covariance $k(\mathbf{x}, \mathbf{x}')$ is defined by a choice of kernel e.g., RBF kernel:

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

    where λ defines the expected smoothness of the function



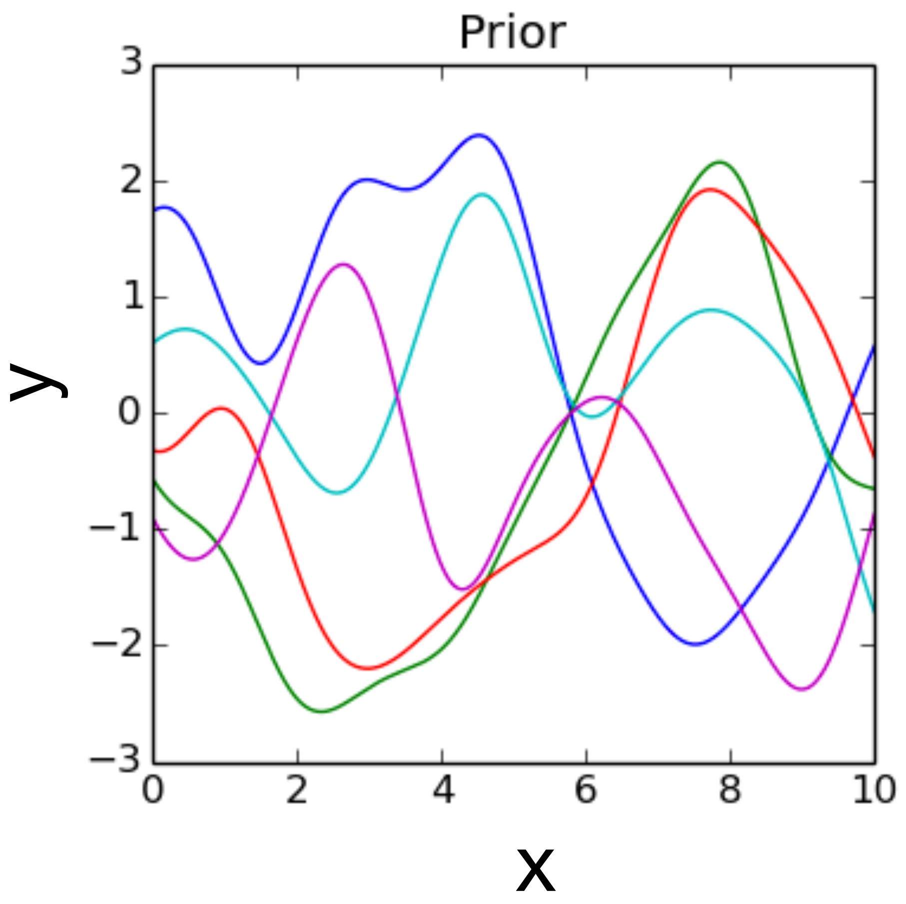
- Once we acqire some data $\mathcal{D} = \{\mathbf{X}, \mathbf{y}\}$, we can compute a posterior prediction about any new datapoint $\mathbf{x}_*$ that is also Gaussian with mean and variance defined as

$$m(\mathbf{x}_* | \mathcal{D}) = \mathbf{k}_*^\top (\mathbf{K} + \sigma^2 \mathbf{I})^{-1} \mathbf{y}$$

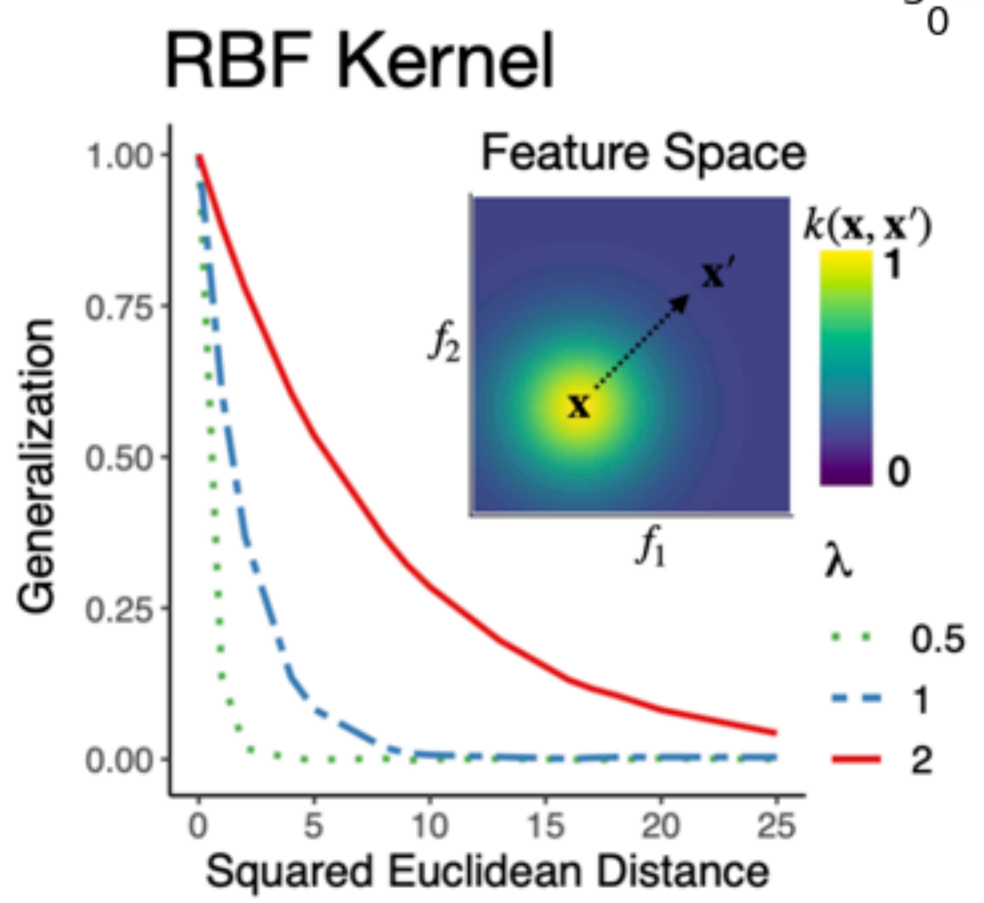
$$v(\mathbf{x}_* | \mathcal{D}) = k(\mathbf{x}_*, \mathbf{x}_*) - \mathbf{k}_*^\top (\mathbf{K} + \sigma^2 \mathbf{I})^{-1} \mathbf{k}_*$$

see Rasmussen & Williams (2005) for further details

Gaussian Process (GP) regression in detail



- Prior over functions (i.e., hypotheses) is a multivariate Gaussian:
  $P(f) \sim \mathcal{GP}\left(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')\right)$
  - prior mean $m(\mathbf{x})$ is typically set to 0 without loss of generalization
  - Covariance $k(\mathbf{x}, \mathbf{x}')$ is defined by a choice of kernel e.g., RBF kernel:
    $$k(\mathbf{x}, \mathbf{x}') = \exp\left(\frac{-||\mathbf{x} - \mathbf{x}'||^2}{2\lambda^2}\right)$$
    where λ defines the expected smoothness of the function



- Once we acqire some data $\mathcal{D} = \{\mathbf{X}, \mathbf{y}\}$, we can compute a posterior prediction about any new datapoint $\mathbf{x}_*$ that is also Gaussian with mean and variance defined as

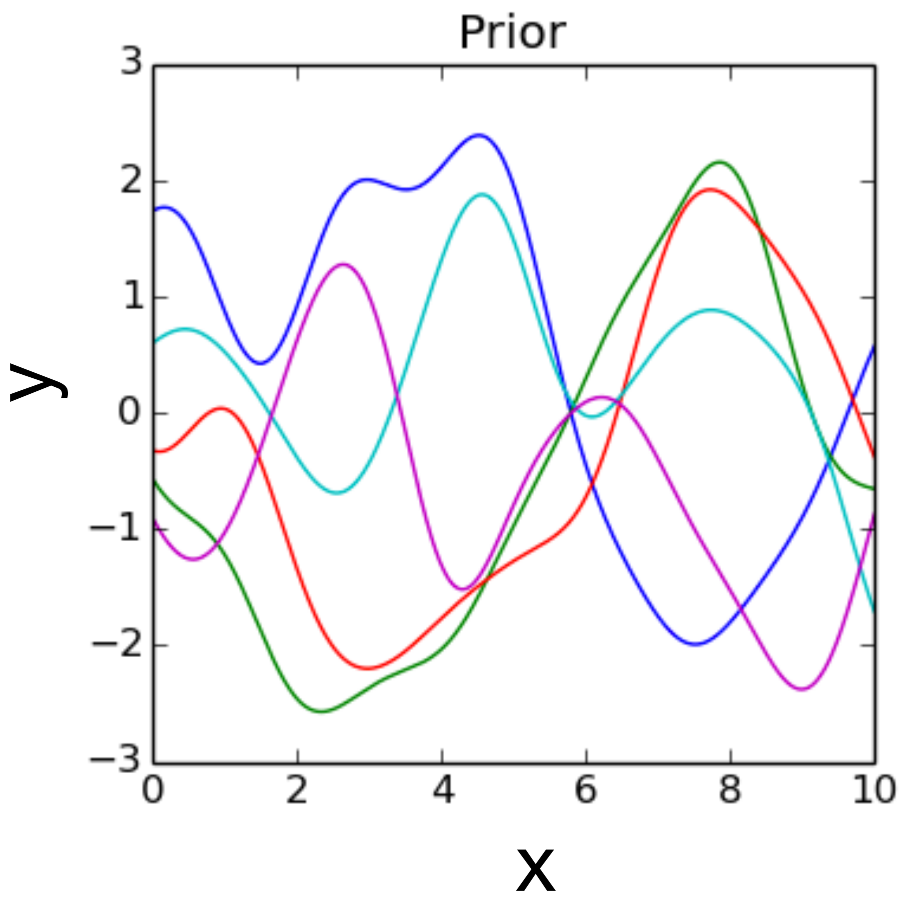
$$m(\mathbf{x}_* | \mathcal{D}) = \mathbf{k}_*^\top(\mathbf{K} + \sigma^2\mathbf{I})^{-1}\mathbf{y}$$
$$v(\mathbf{x}_* | \mathcal{D}) = k(\mathbf{x}_*, \mathbf{x}_*) - \mathbf{k}_*^\top(\mathbf{K} + \sigma^2\mathbf{I})^{-1}\mathbf{k}_*$$

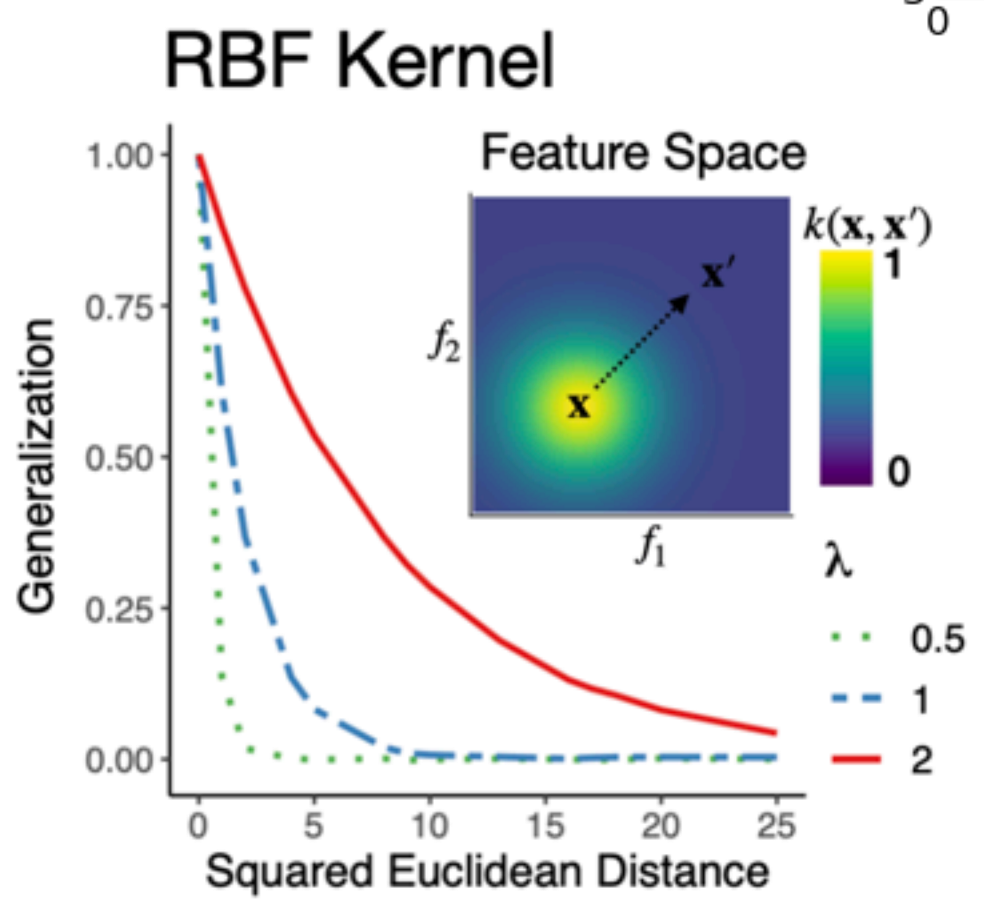
see Rasmussen & Williams (2005) for further details

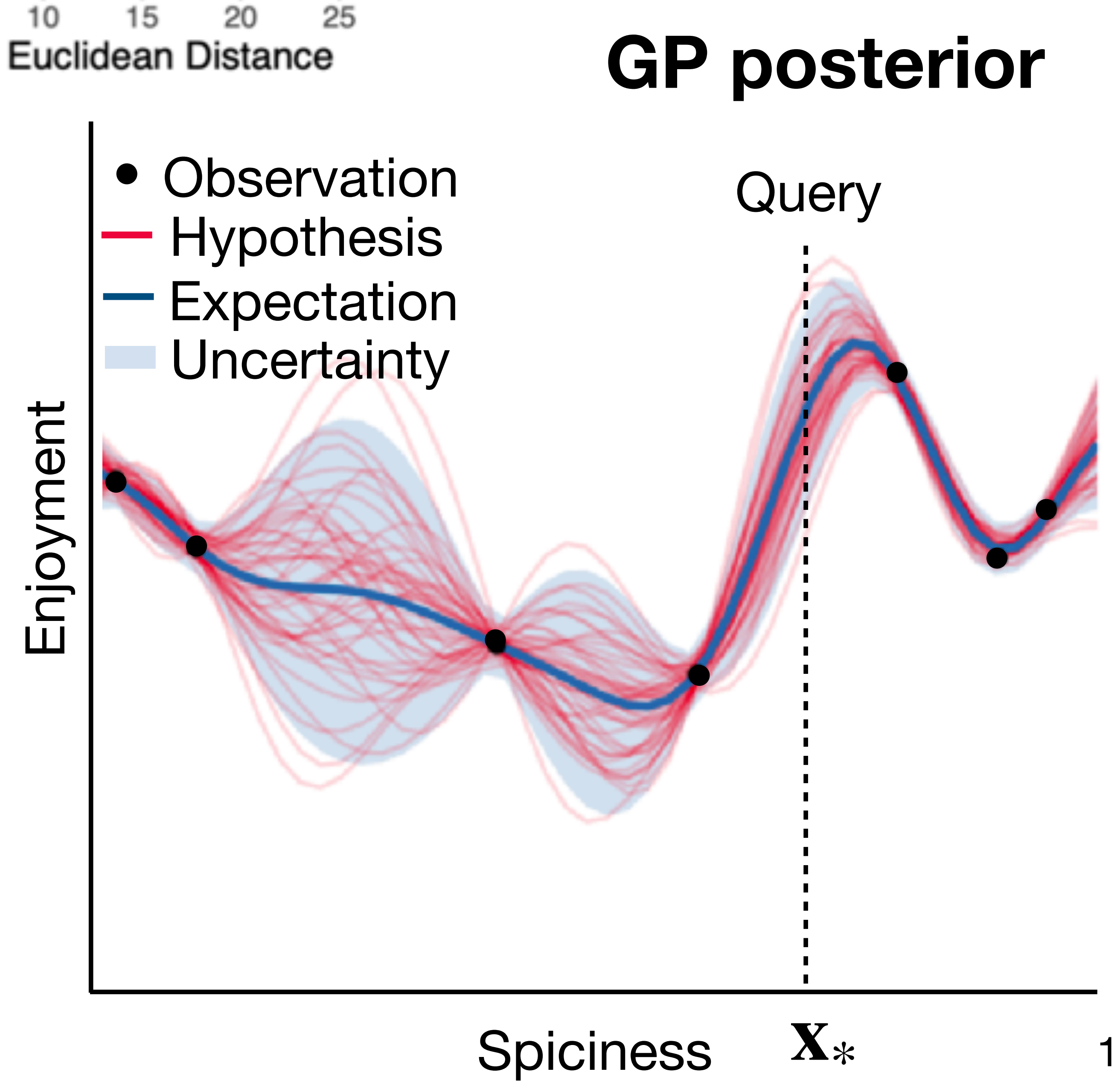
Gaussian Process (GP) regression in detail

- Prior over functions (i.e., hypotheses) is a multivariate Gaussian:
  $P(f) \sim \mathcal{GP}\left(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')\right)$
  - prior mean $m(\mathbf{x})$ is typically set to 0 without loss of generalization
  - Covariance $k(\mathbf{x}, \mathbf{x}')$ is defined by a choice of kernel
    e.g., RBF kernel:
    $$k(\mathbf{x}, \mathbf{x}') = \exp\left(\frac{-||\mathbf{x} - \mathbf{x}'||^2}{2\lambda^2}\right)$$
    where λ defines the expected smoothness of the function
- Once we acqire some data $\mathcal{D} = \{\mathbf{X}, \mathbf{y}\}$, we can compute a posterior prediction about any new datapoint $\mathbf{x}_*$ that is also Gaussian with mean and variance defined as

$$m(\mathbf{x}_* | \mathcal{D}) = \mathbf{k}_*^\top (\mathbf{K} + \sigma^2 \mathbf{I})^{-1} \mathbf{y}$$
$$v(\mathbf{x}_* | \mathcal{D}) = k(\mathbf{x}_*, \mathbf{x}_*) - \mathbf{k}_*^\top (\mathbf{K} + \sigma^2 \mathbf{I})^{-1} \mathbf{k}_*$$

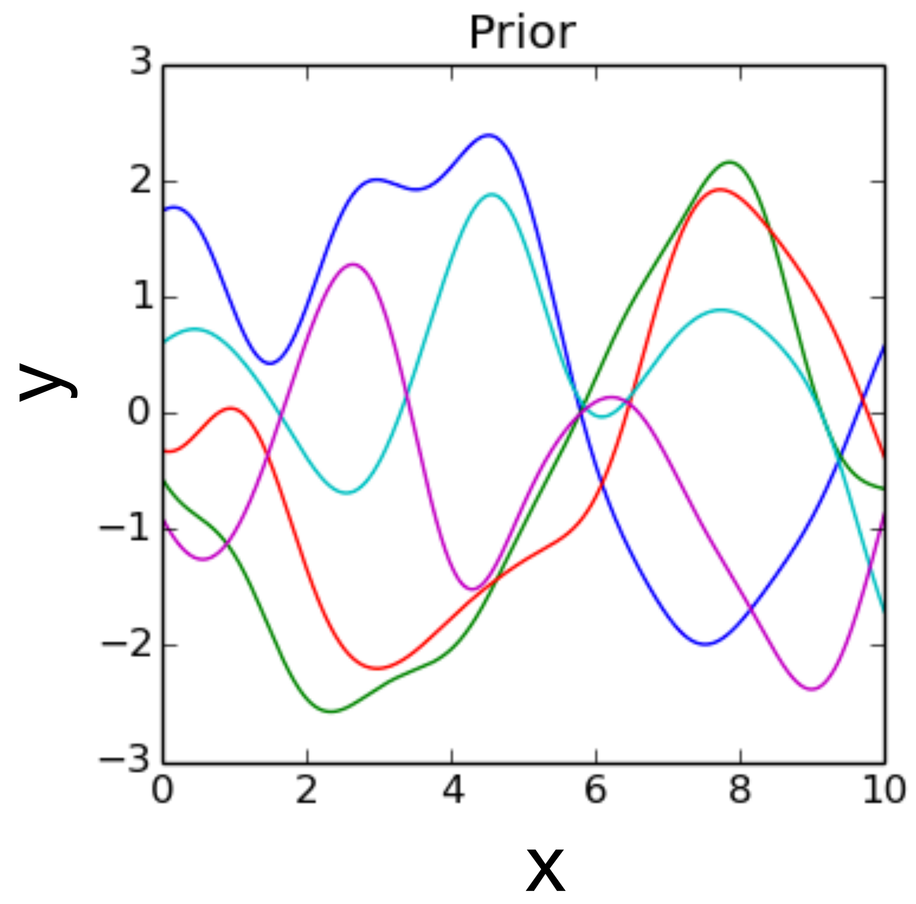






see Rasmussen & Williams (2005) for further details

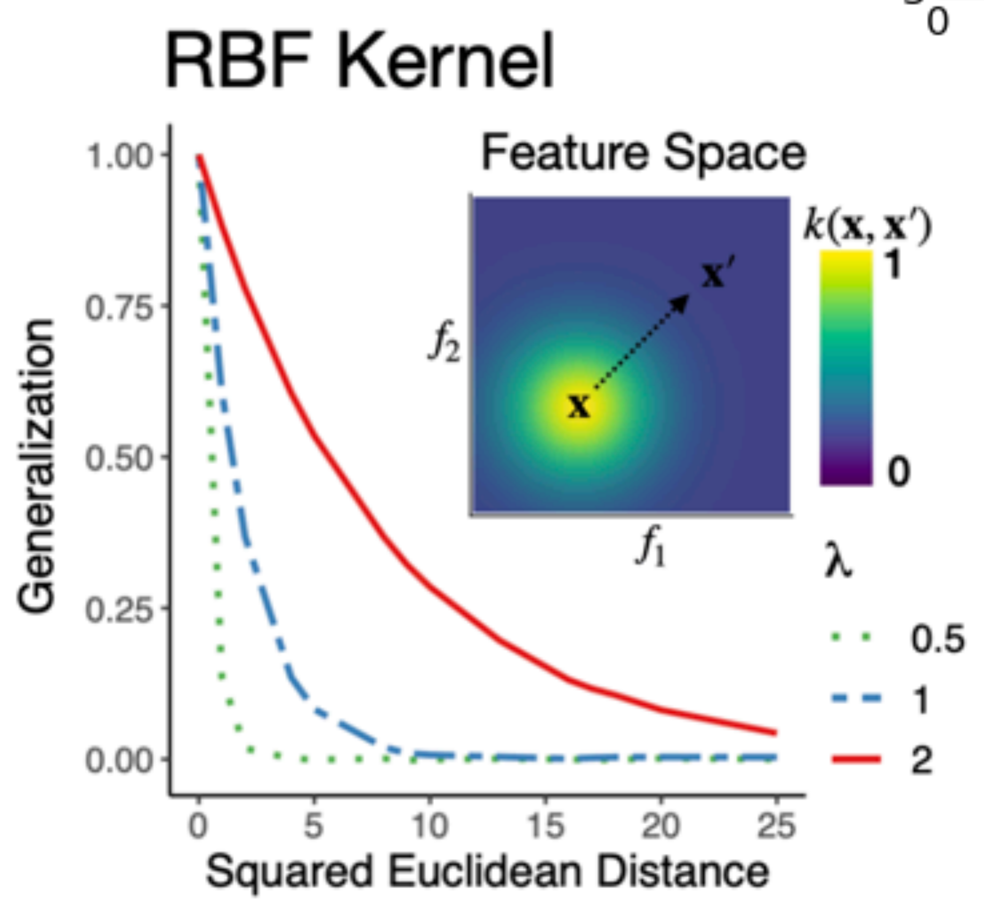
Gaussian Process (GP) regression in detail



- Prior over functions (i.e., hypotheses) is a multivariate Gaussian: $P(f) \sim \mathcal{GP}\left(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')\right)$
  - prior mean $m(\mathbf{x})$ is typically set to 0 without loss of generalization
  - Covariance $k(\mathbf{x}, \mathbf{x}')$ is defined by a choice of kernel e.g., RBF kernel:

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

where λ defines the expected smoothness of the function



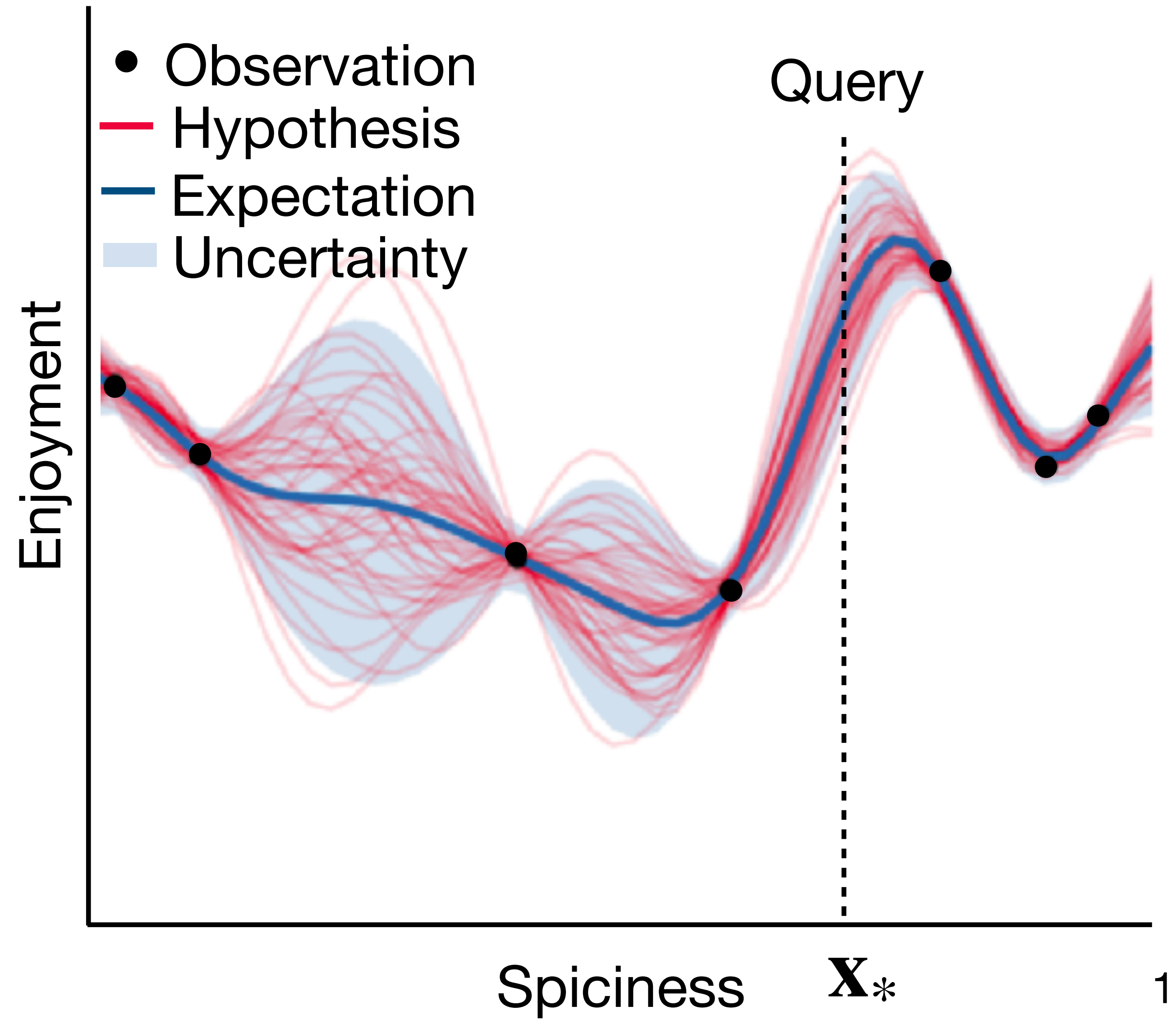
- Once we acqire some data $\mathcal{D} = \{\mathbf{X}, \mathbf{y}\}$, we can compute a posterior prediction about any new datapoint $\mathbf{x}_*$ that is also Gaussian with mean and variance defined as

$$m(\mathbf{x}_* | \mathcal{D}) = \mathbf{k}_*^\top (\mathbf{K} + \sigma^2 \mathbf{I})^{-1} \mathbf{y}$$

$$v(\mathbf{x}_* | \mathcal{D}) = k(\mathbf{x}_*, \mathbf{x}_*) - \mathbf{k}_*^\top (\mathbf{K} + \sigma^2 \mathbf{I})^{-1} \mathbf{k}_*$$

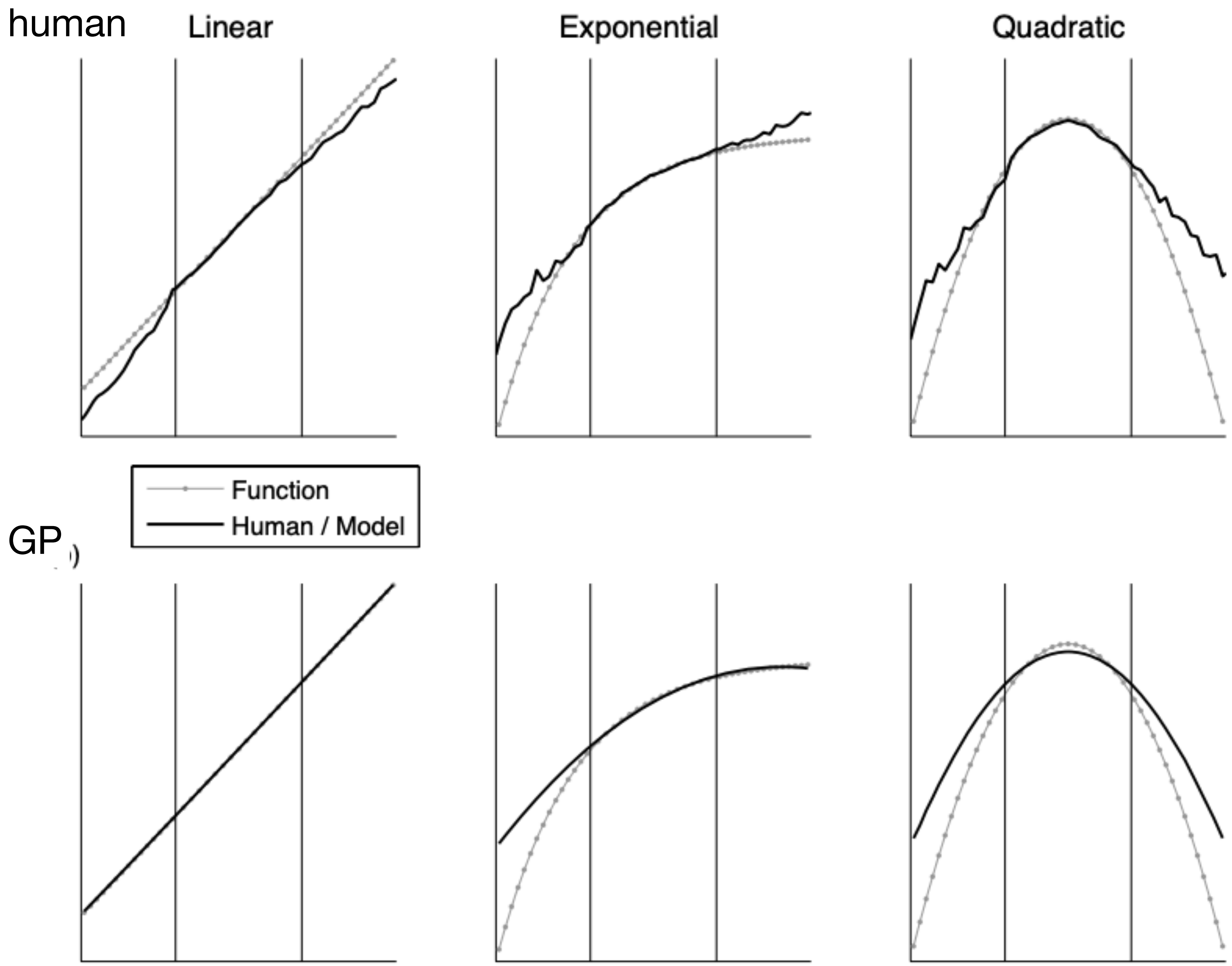
*Don't worry too much about what these equations mean for now; I will provide some intuitions later

GP posterior



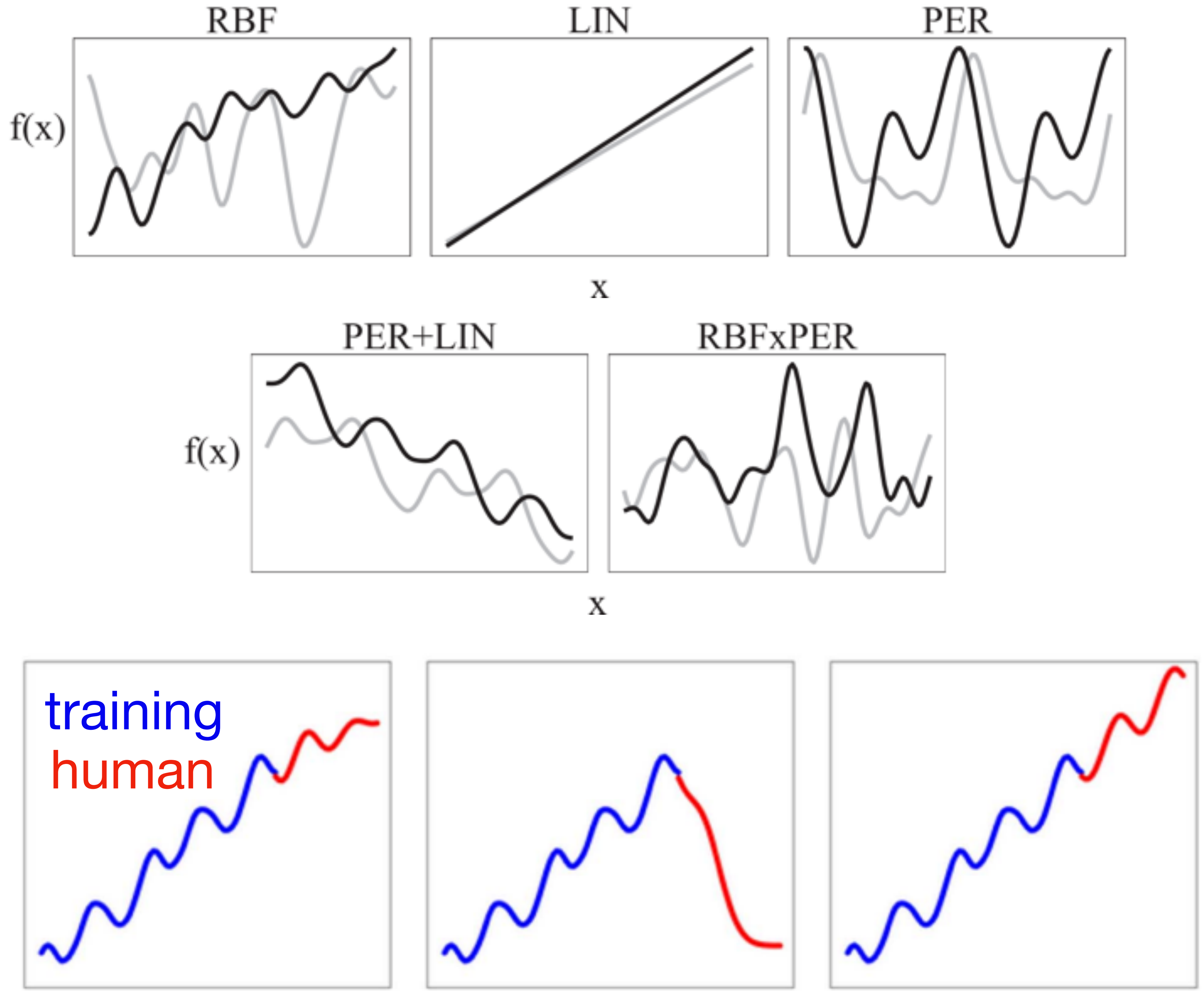
GPs provide the best predictions for human function learning

Extrapolation



Griffiths, Lucas, & Williams, (*Neurips* 2008)

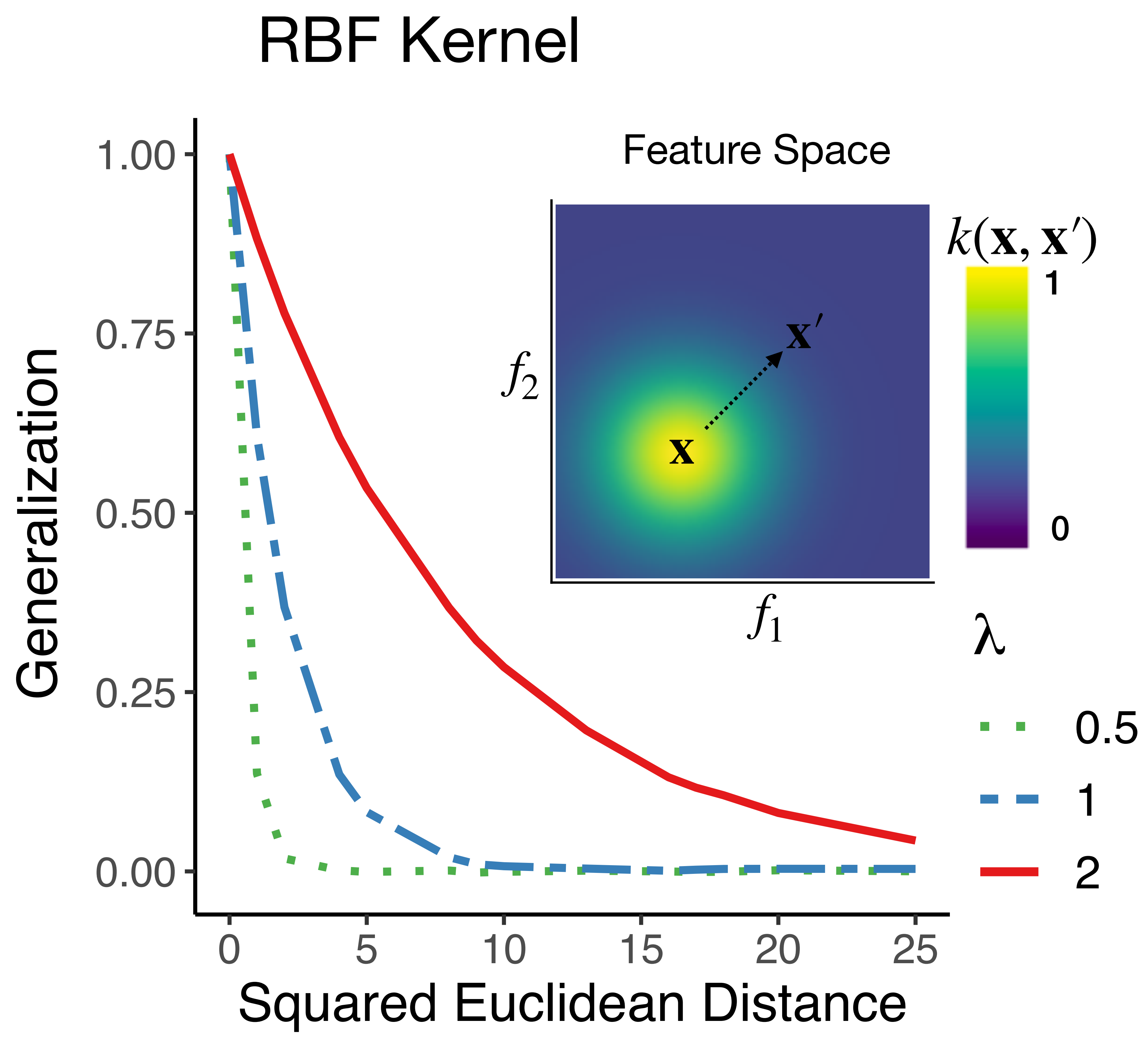
Compositional functions



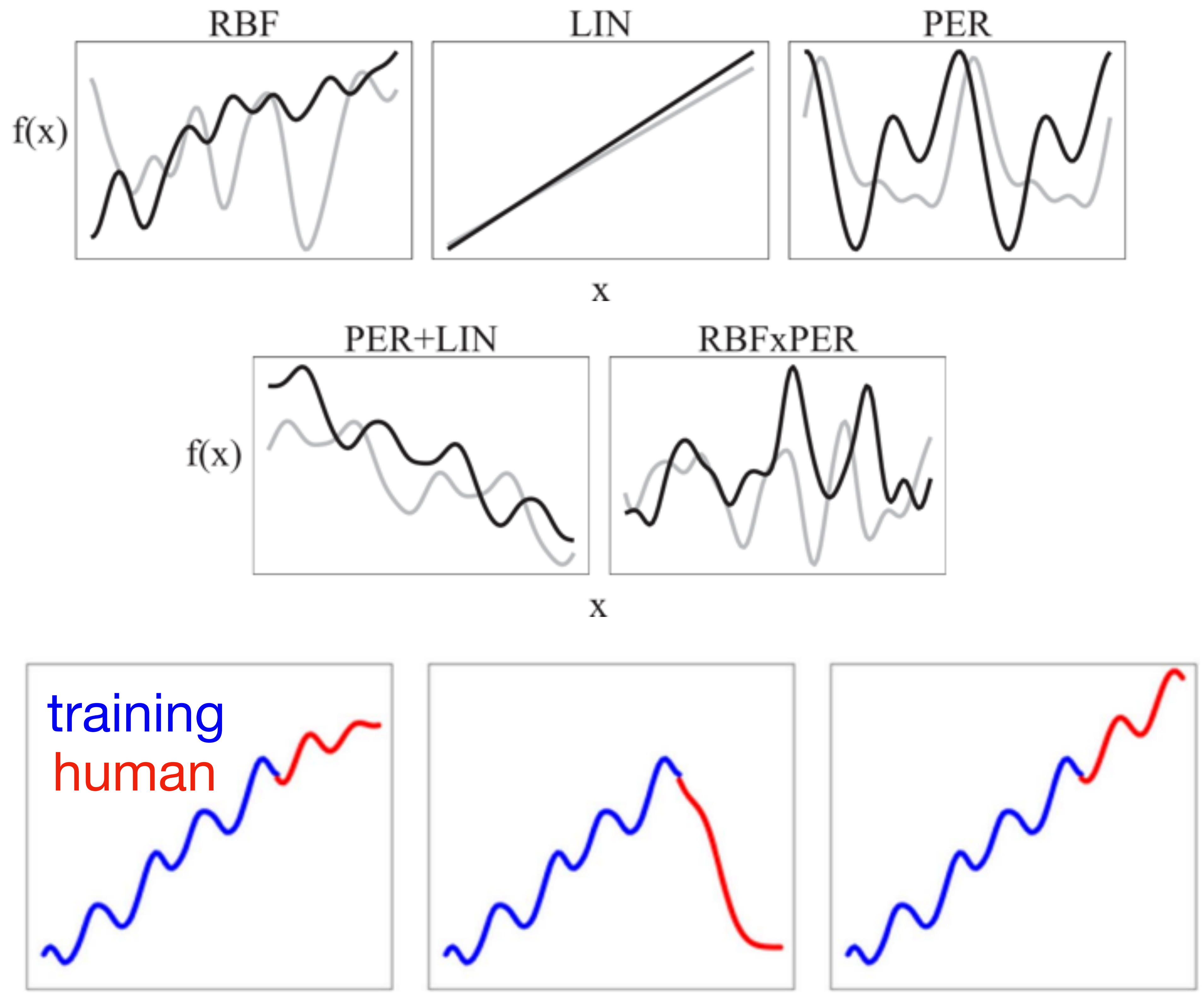
Schulz et al., (*CogPsych* 2017)

Duality of GP function learning

Kernel provides an explicit **similarity** metric



Kernels can be compositionally combined, similar to how we can combine **rules** to create new ones



Wu et al,. (*AnnRevPsych* forthcoming)

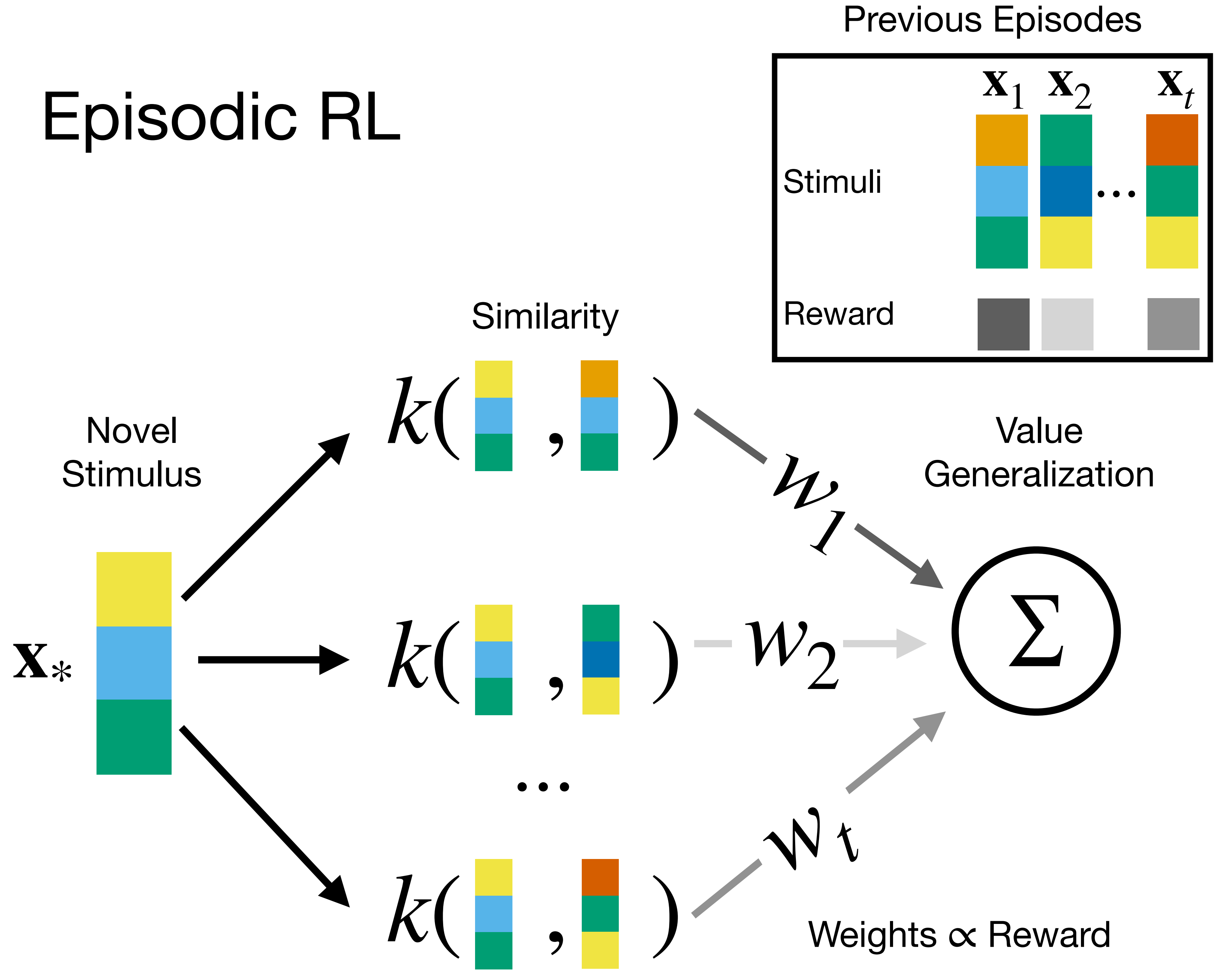
Schulz et al., (*CogPsych* 2017)

Connection to RL

Connection to RL

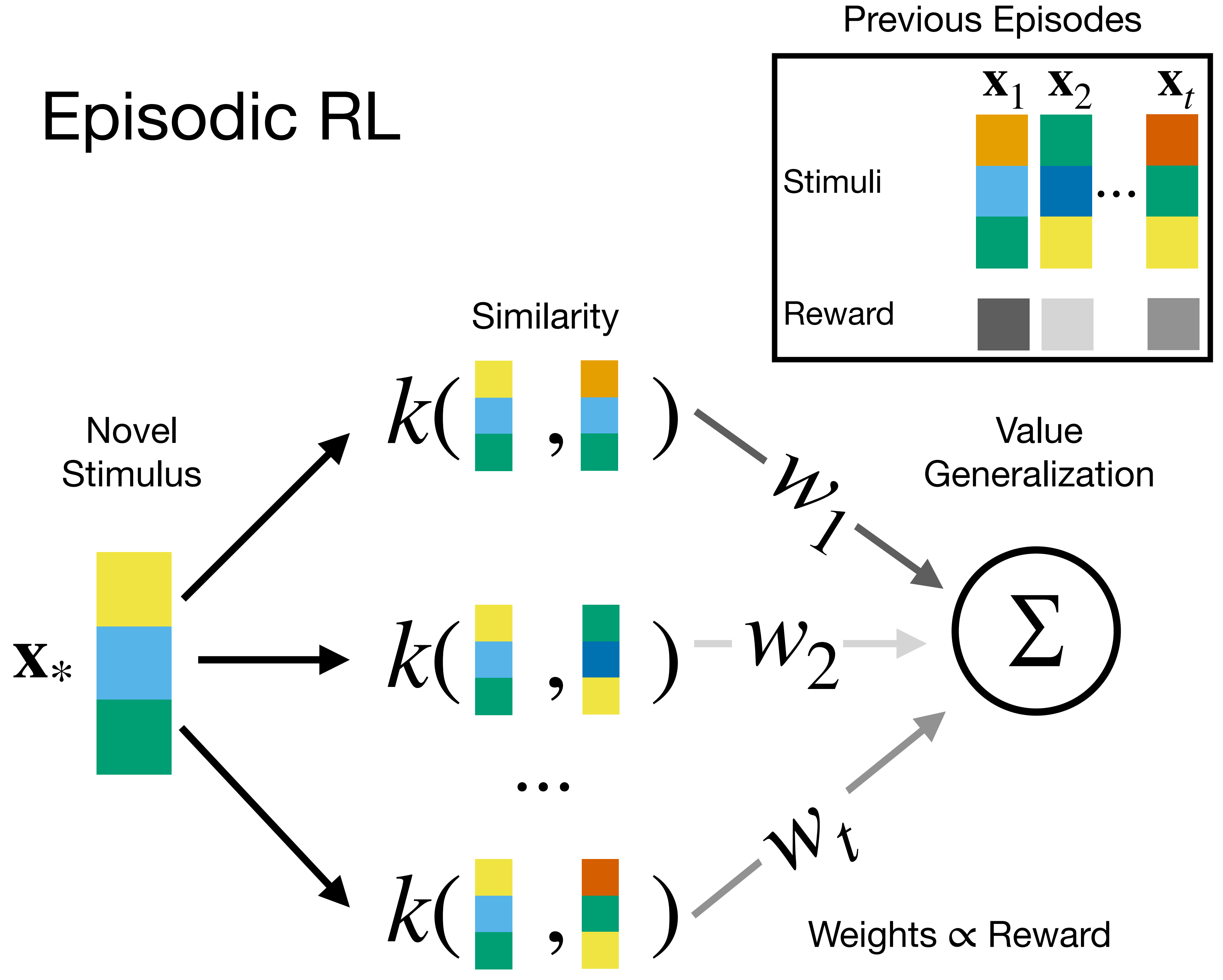
- Episodic RL for generalization in new settings
(Gershman & Daw, *AnnRevPsych* 2017; Bottvinick et al., *TICS* 2019)
  - Store a memory for each previously encountered stimuli $\mathbf{x}$ and it's reward y
  - Predict the value of new stimuli based on a similarity-weighted sum of past episodes

Episodic RL



Connection to RL

- Episodic RL for generalization in new settings
  (Gershman & Daw, *AnnRevPsych* 2017; Bottvinick et al., *TICS* 2019)
  - Store a memory for each previously encountered stimuli $\mathbf{x}$ and it's reward y
  - Predict the value of new stimuli based on a similarity-weighted sum of past episodes
- GPs provide a Bayesian analogue of Episodic RL
  - Using an RBF kernel as the similarity metric, Episodic RL is equivalent to the GP posterior mean
    (Poggio & Bizzi, *Nature* 2004; Sutton & Barto, 2018; Jäkel, Schölkopf, & Wichman, *J.MathPsych*, 2008)
  - Yet GPs provide uncertainty estimates, which is essential for defining which states to explore!



GP posterior mean

$$m(\mathbf{x}_* \mid \mathcal{D}) = \mathbf{k}_*^\top(\mathbf{K} + \sigma^2\mathbf{I})^{-1}\mathbf{y}$$

$$= \sum_{i=1}^{N} w_i k(\mathbf{x}, \mathbf{x}')$$

where $\mathbf{w} = [\mathbf{K} + \sigma^2\mathbf{I}]^{-1}\mathbf{y}$

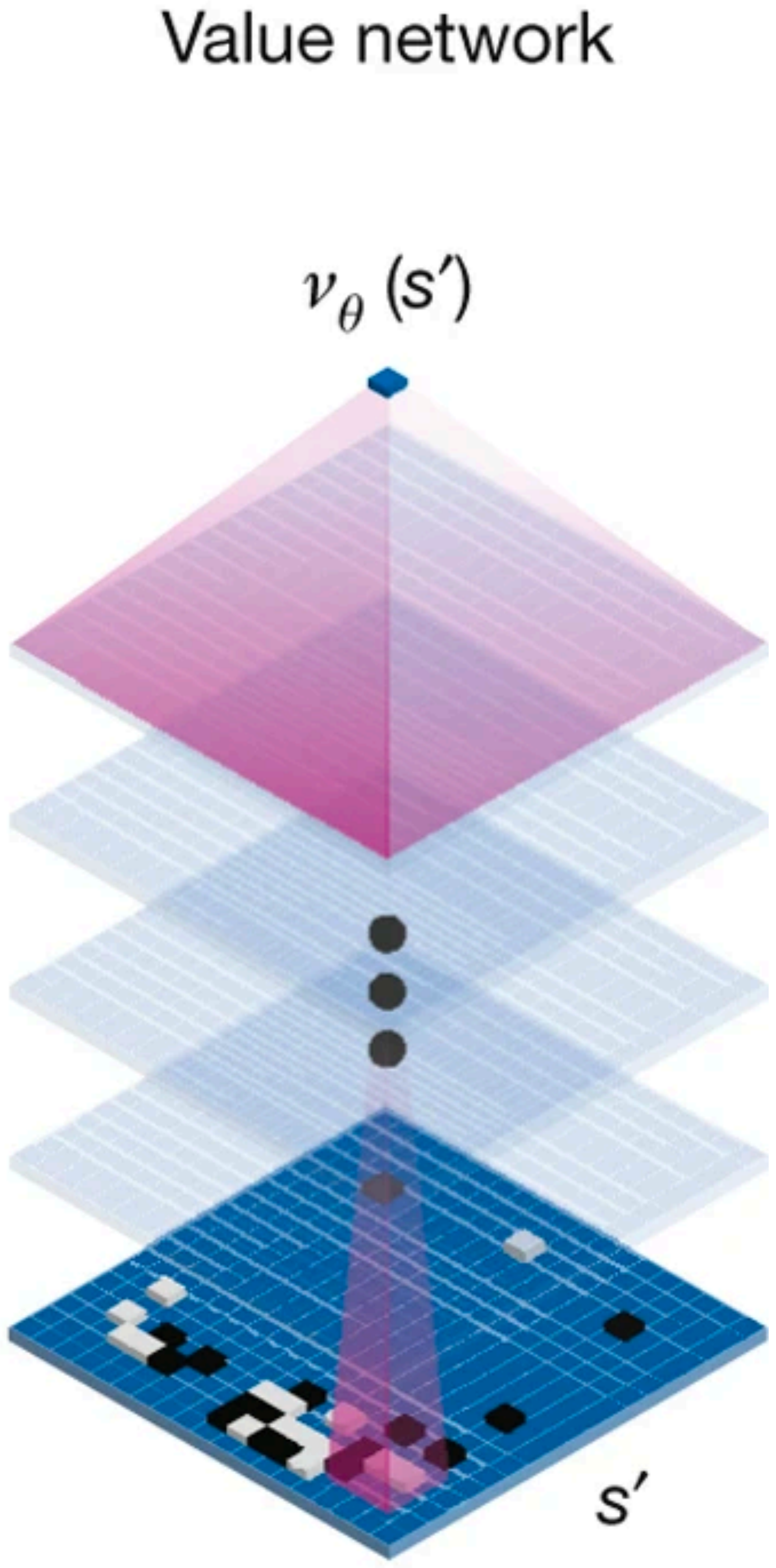
Value function approximation in RL

- Classic function learning is typically a supervised learning problem
    - Given stimulus $\mathbf{x}_*$ predict $f(\mathbf{x}_*)$
- Value function approximation is a key method for generalization in RL
    - Use function learning mechanisms for inferring *implicit* value of novel states: $V(s') = f(s')$
    - Implement a policy on the basis of value: $\pi(s') \propto \exp(V(\mathbf{s}'))$
- AlphaGo uses a deep neural network for value function approximation
    - DNNs are simply a universal function approximator (Cybenko, 1989)
    - But for understanding human behavior, GPs offer better interpretability due to psychologically meaningful parameters
    - GPs are equivalent to an infinitely wide deep neural network (Neal, 1996)
- After the break, I will present some of my research using GPs to model human generalization in RL

Silver et al., (*Nature* 2016)

Value function approximation in RL

- Classic function learning is typically a supervised learning problem
  - Given stimulus $\mathbf{x}_*$ predict $f(\mathbf{x}_*)$
- Value function approximation is a key method for generalization in RL
  - Use function learning mechanisms for inferring *implicit* value of novel states: $V(s') = f(s')$
  - Implement a policy on the basis of value: $\pi(s') \propto \exp(V(\mathbf{s}'))$
- AlphaGo uses a deep neural network for value function approximation
  - DNNs are simply a universal function approximator (Cybenko, 1989)
  - But for understanding human behavior, GPs offer better interpretability due to psychologically meaningful parameters
  - GPs are equivalent to an infinitely wide deep neural network (Neal, 1996)
- After the break, I will present some of my research using GPs to model human generalization in RL



Silver et al., (*Nature* 2016)

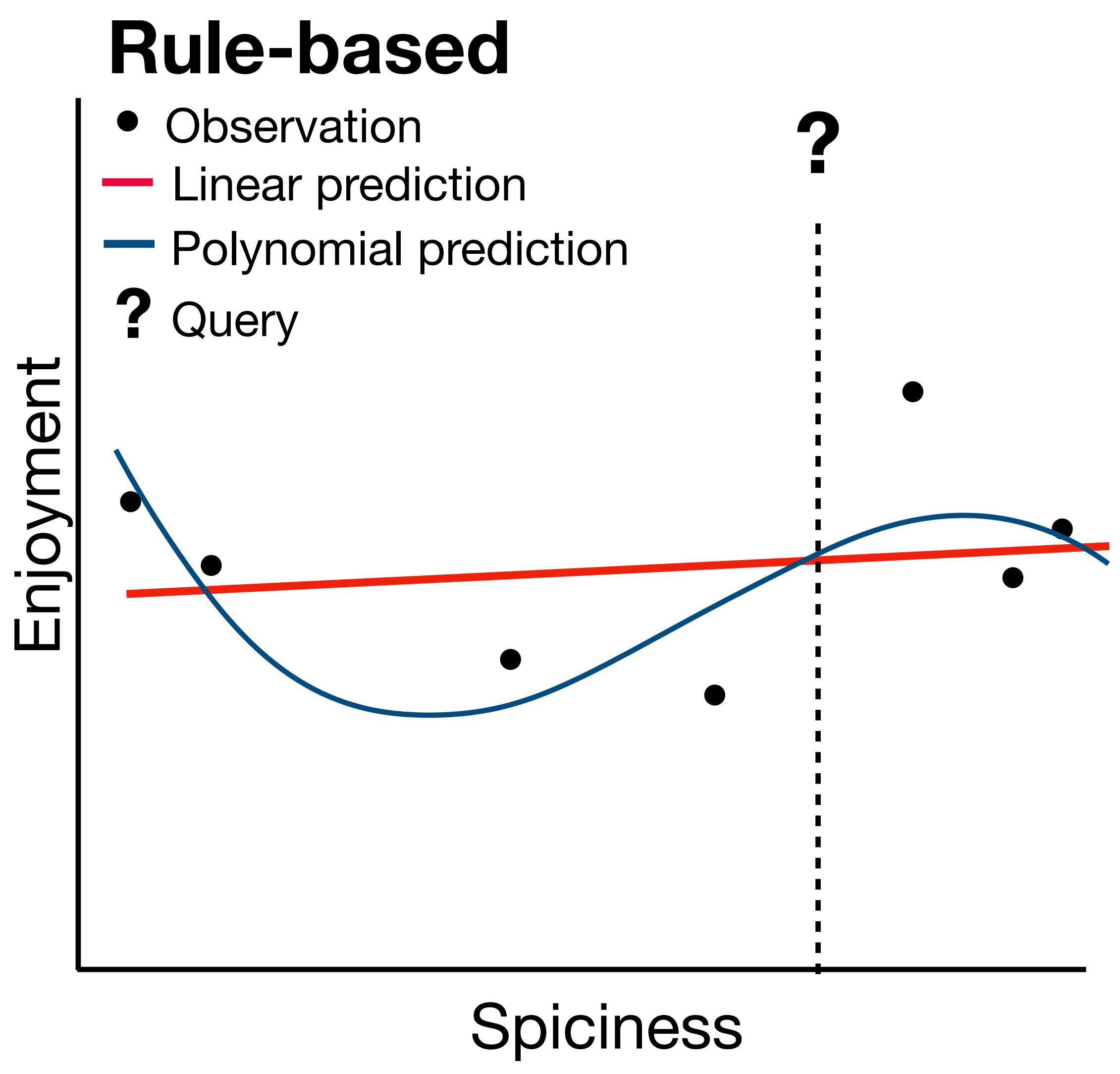
Interim summary

- Function learning is a regression problem
- Early **rule-based theories** assumed humans learn functions by picking specific class of functions and then optimizing the weights (as in linear or parametric regression) —> Brittle and lacked flexibility
- **Similarity-based methods** used ANNs to encode the generic principle that similar inputs produce similar outputs —> failed to capture systematic biases in how humans extrapolate
- **Hybrid approaches** using GP regression offer a Bayesian framework, combining kernel similarity and rule-like compositionality of kernels

Regression task

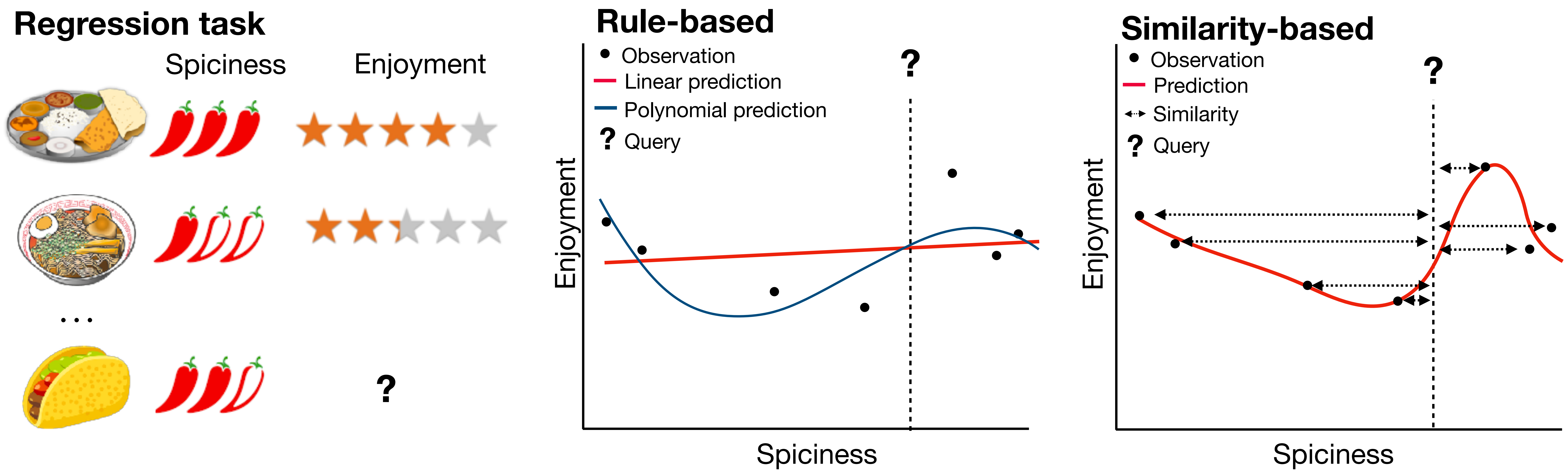
Interim summary

- Function learning is a regression problem
- Early **rule-based theories** assumed humans learn functions by picking specific class of functions and then optimizing the weights (as in linear or parametric regression) —> Brittle and lacked flexibility
- **Similarity-based methods** used ANNs to encode the generic principle that similar inputs produce similar outputs —> failed to capture systematic biases in how humans extrapolate
- **Hybrid approaches** using GP regression offer a Bayesian framework, combining kernel similarity and rule-like compositionality of kernels



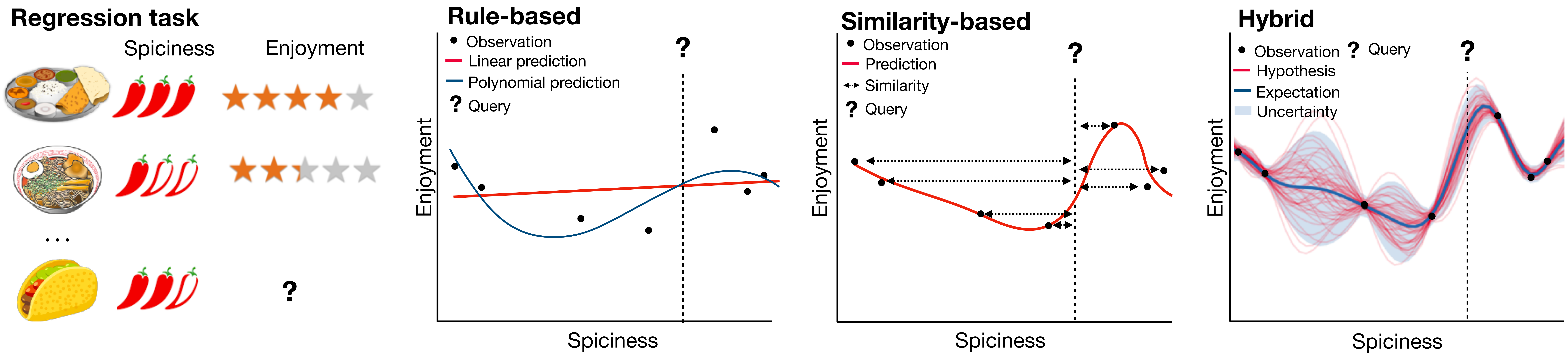
Interim summary

- Function learning is a regression problem
- Early **rule-based theories** assumed humans learn functions by picking specific class of functions and then optimizing the weights (as in linear or parametric regression) —> Brittle and lacked flexibility
- **Similarity-based methods** used ANNs to encode the generic principle that similar inputs produce similar outputs —> failed to capture systematic biases in how humans extrapolate
- **Hybrid approaches** using GP regression offer a Bayesian framework, combining kernel similarity and rule-like compositionality of kernels



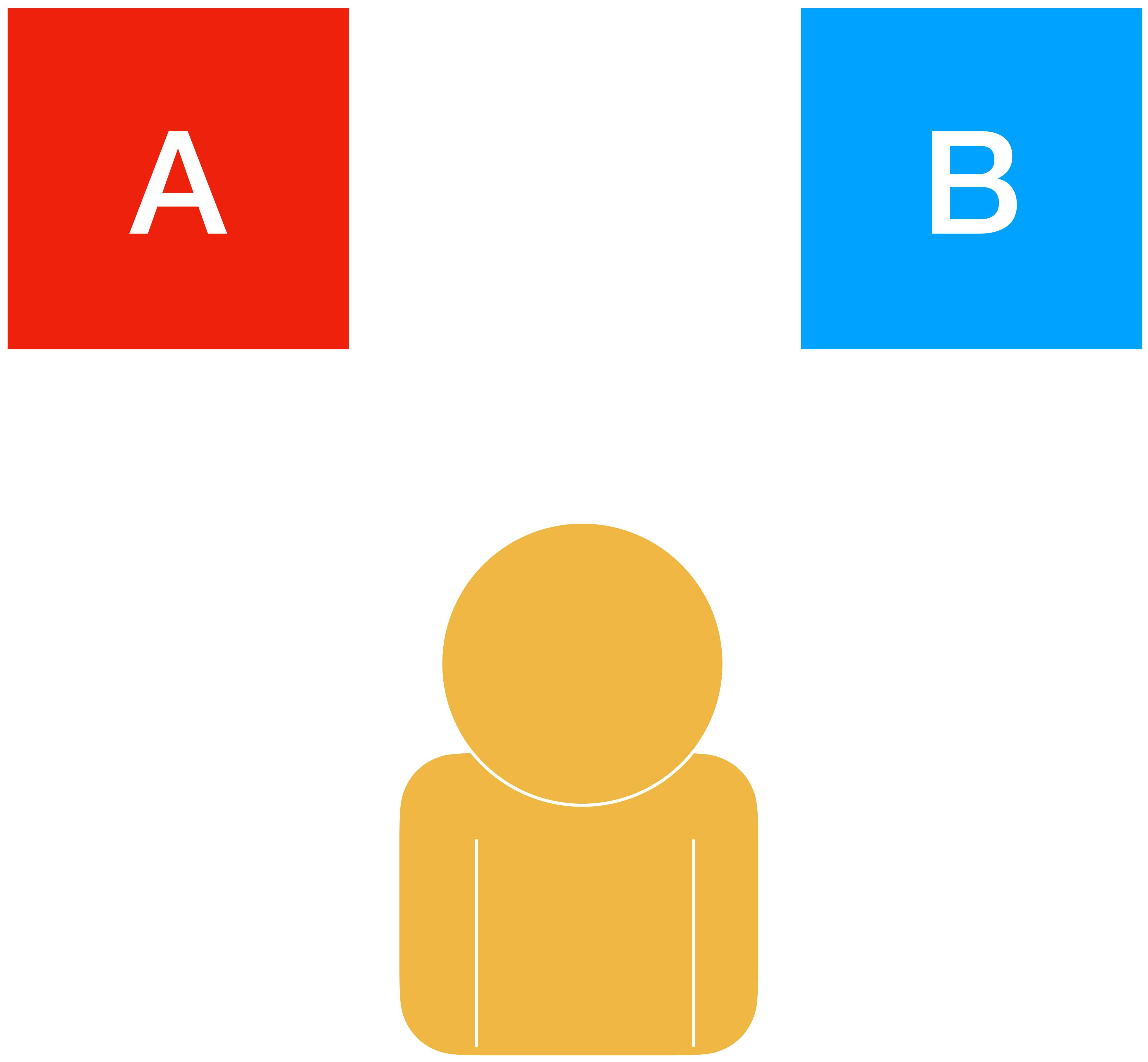
Interim summary

- Function learning is a regression problem
- Early **rule-based theories** assumed humans learn functions by picking specific class of functions and then optimizing the weights (as in linear or parametric regression) —> Brittle and lacked flexibility
- **Similarity-based methods** used ANNs to encode the generic principle that similar inputs produce similar outputs —> failed to capture systematic biases in how humans extrapolate
- **Hybrid approaches** using GP regression offer a Bayesian framework, combining kernel similarity and rule-like compositionality of kernels

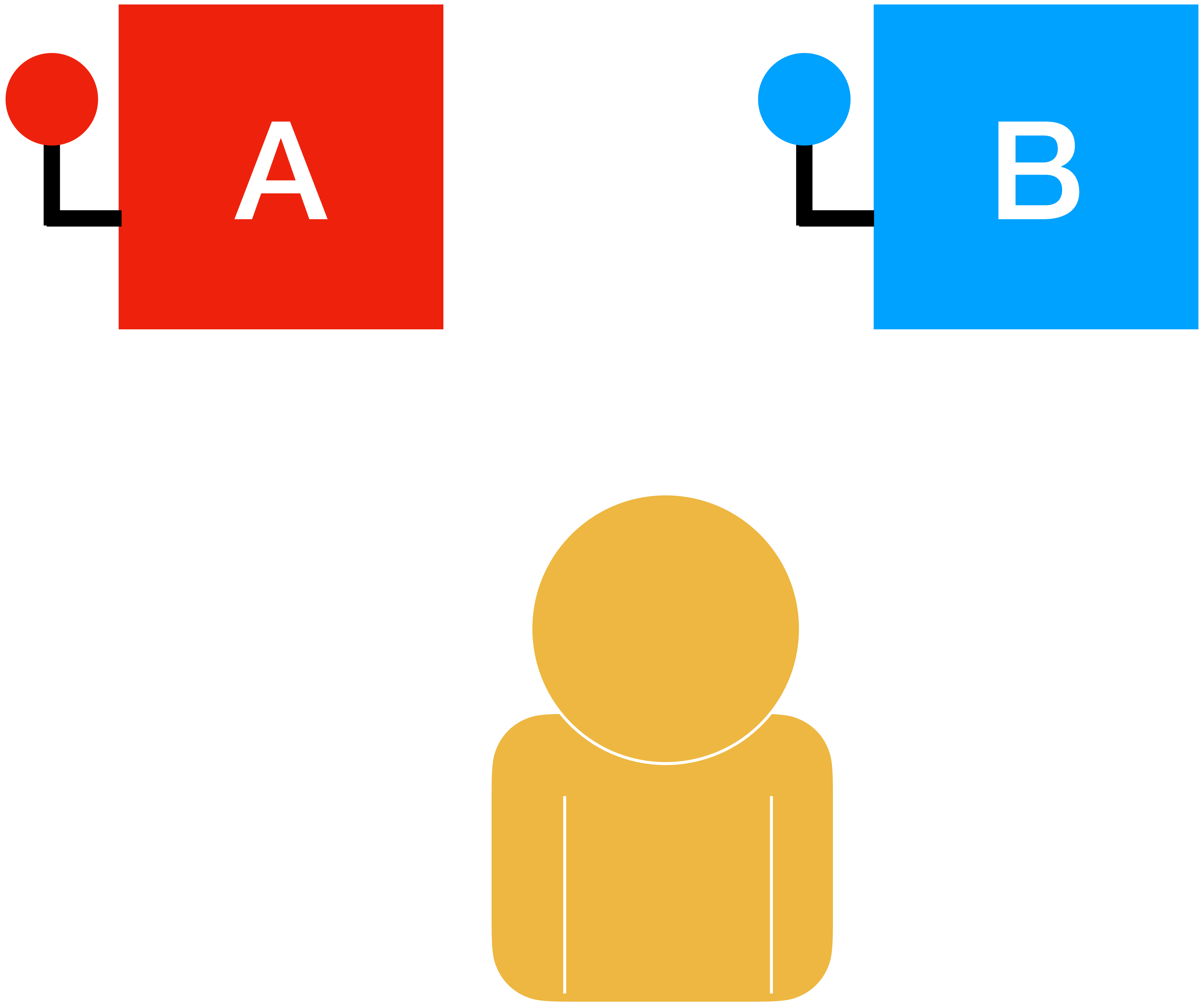


5 minute break

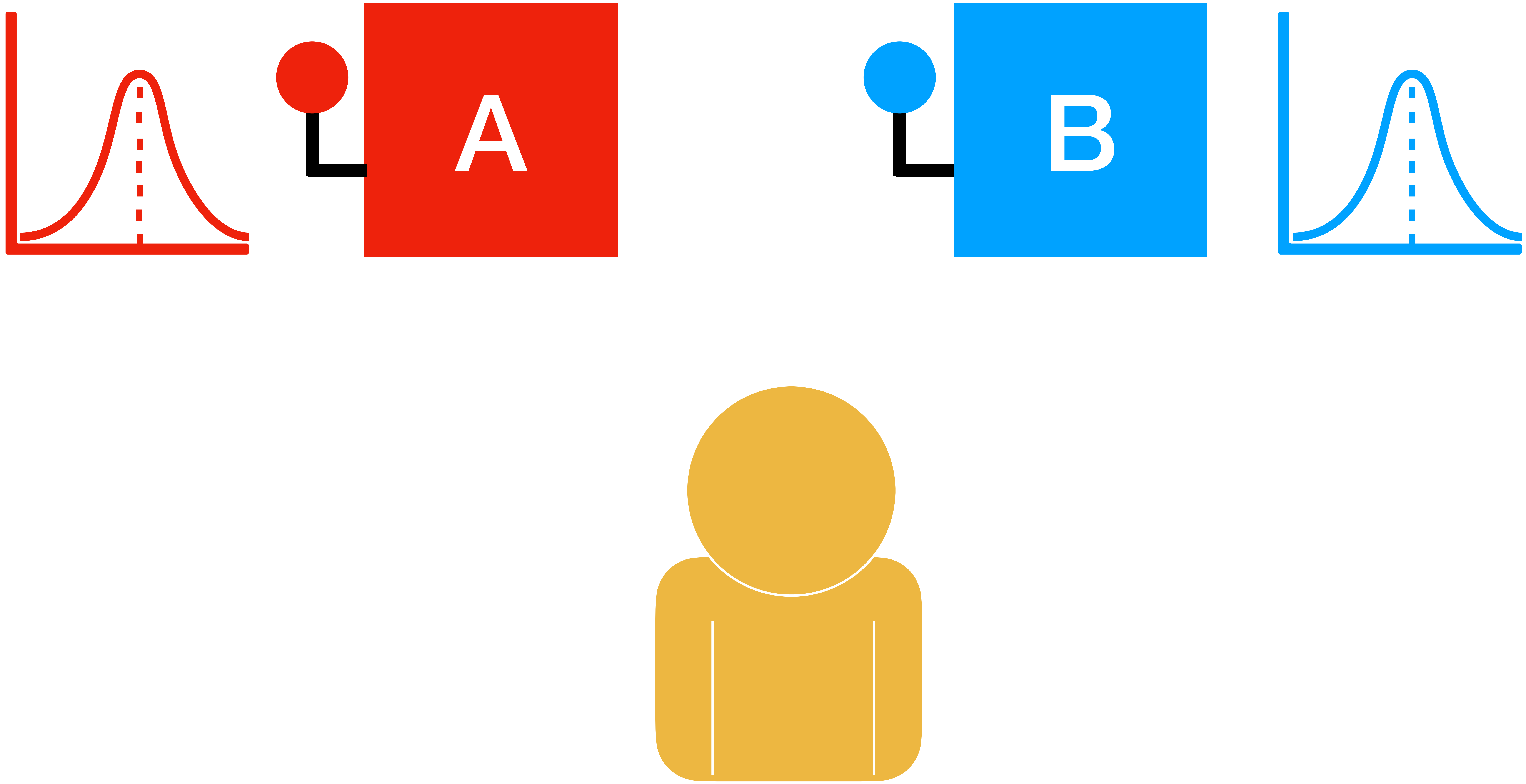
Human learning in the lab



Human learning in the lab

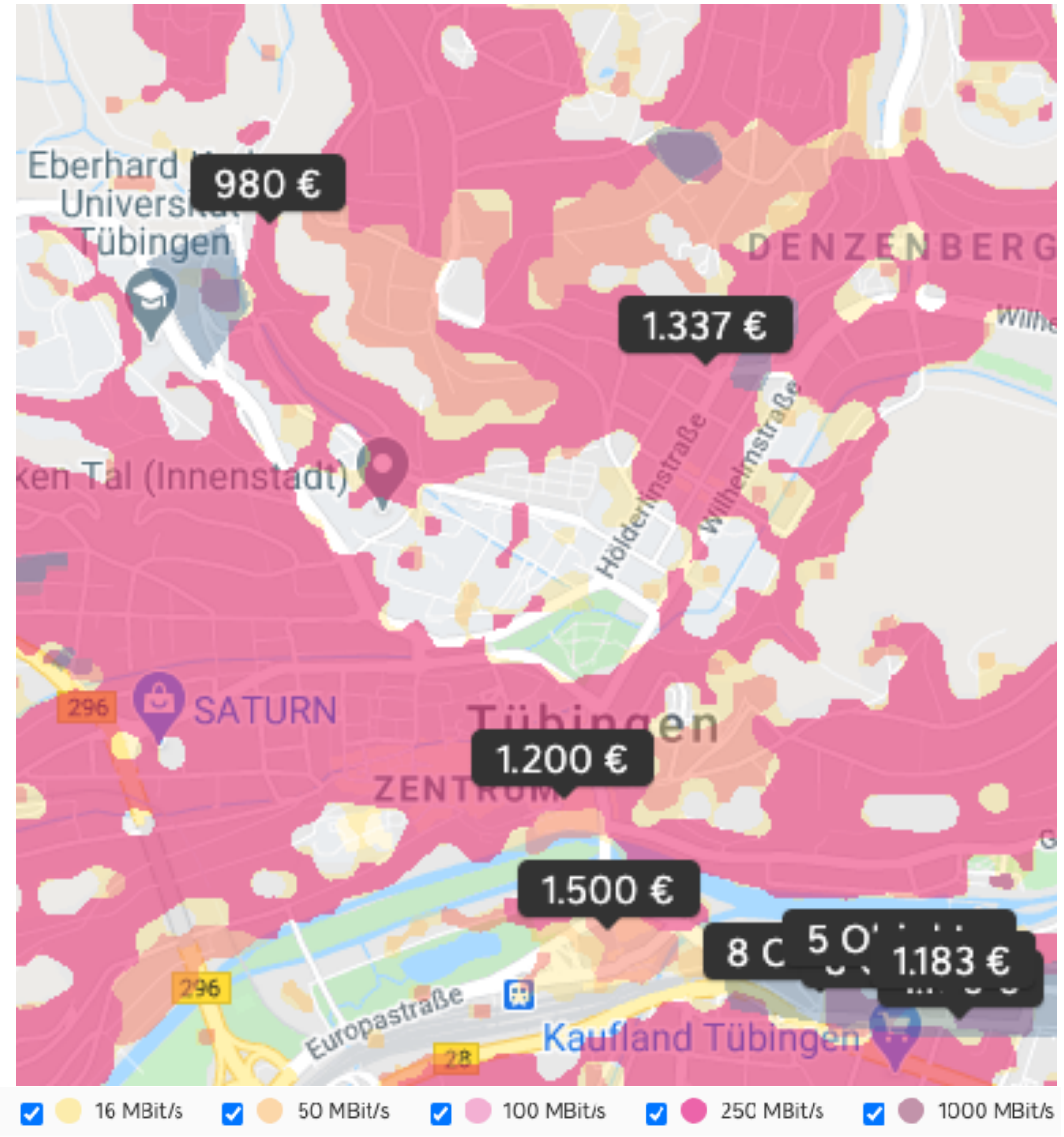


Human learning in the lab



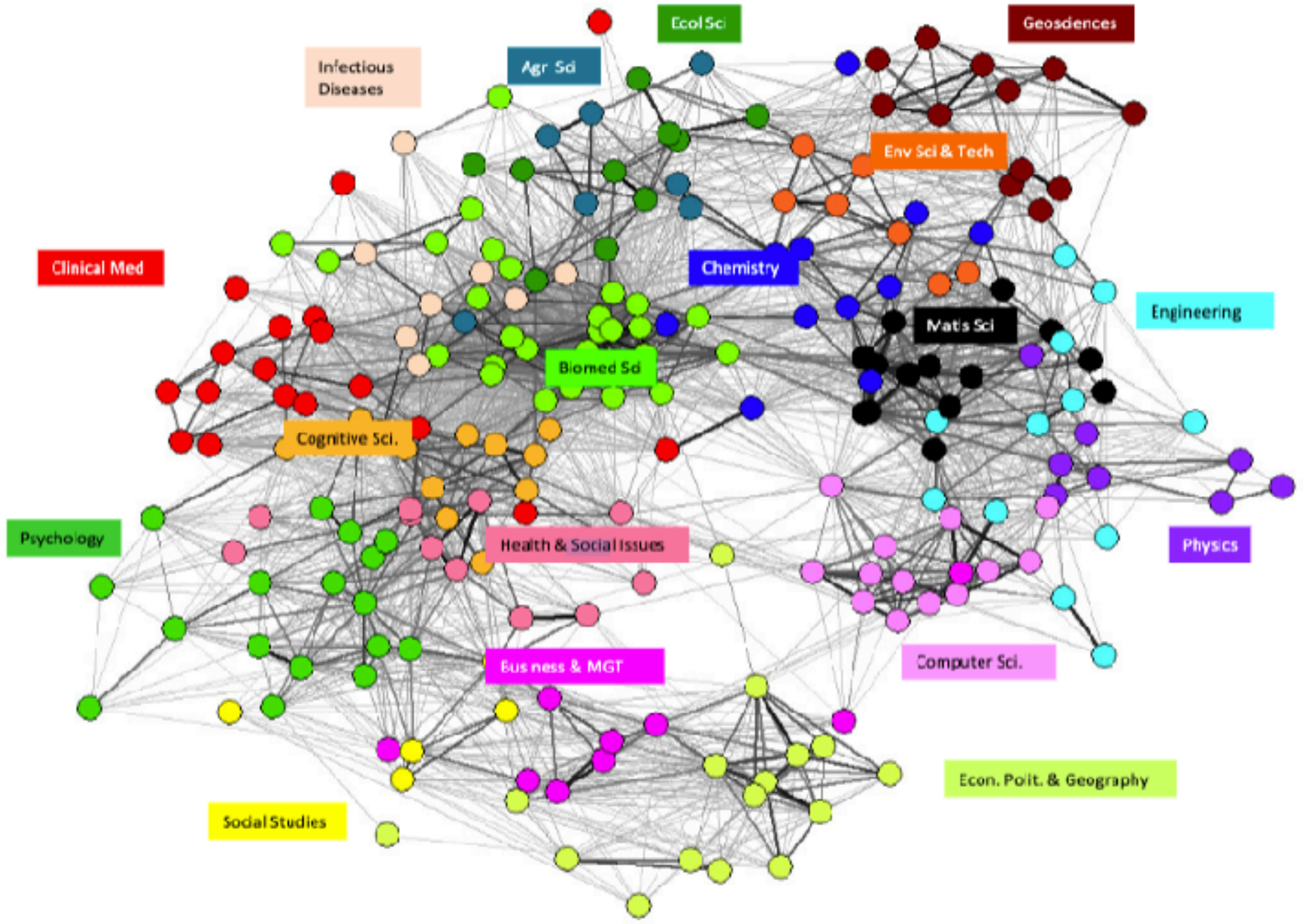
Real life problems

Finding a place to live

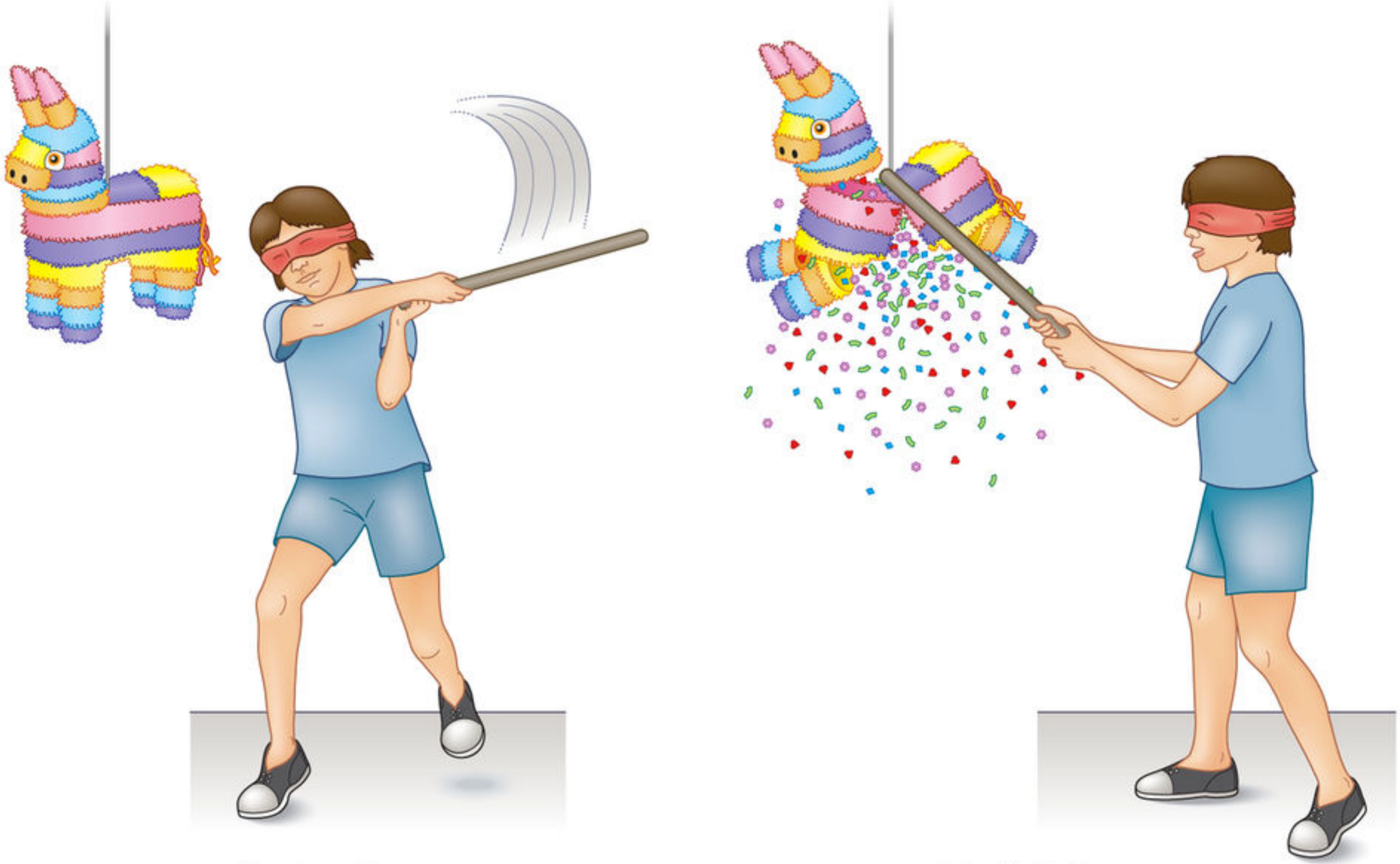


Picking what to eat

Choosing a research topic



Exploration-Exploitation Dilemma



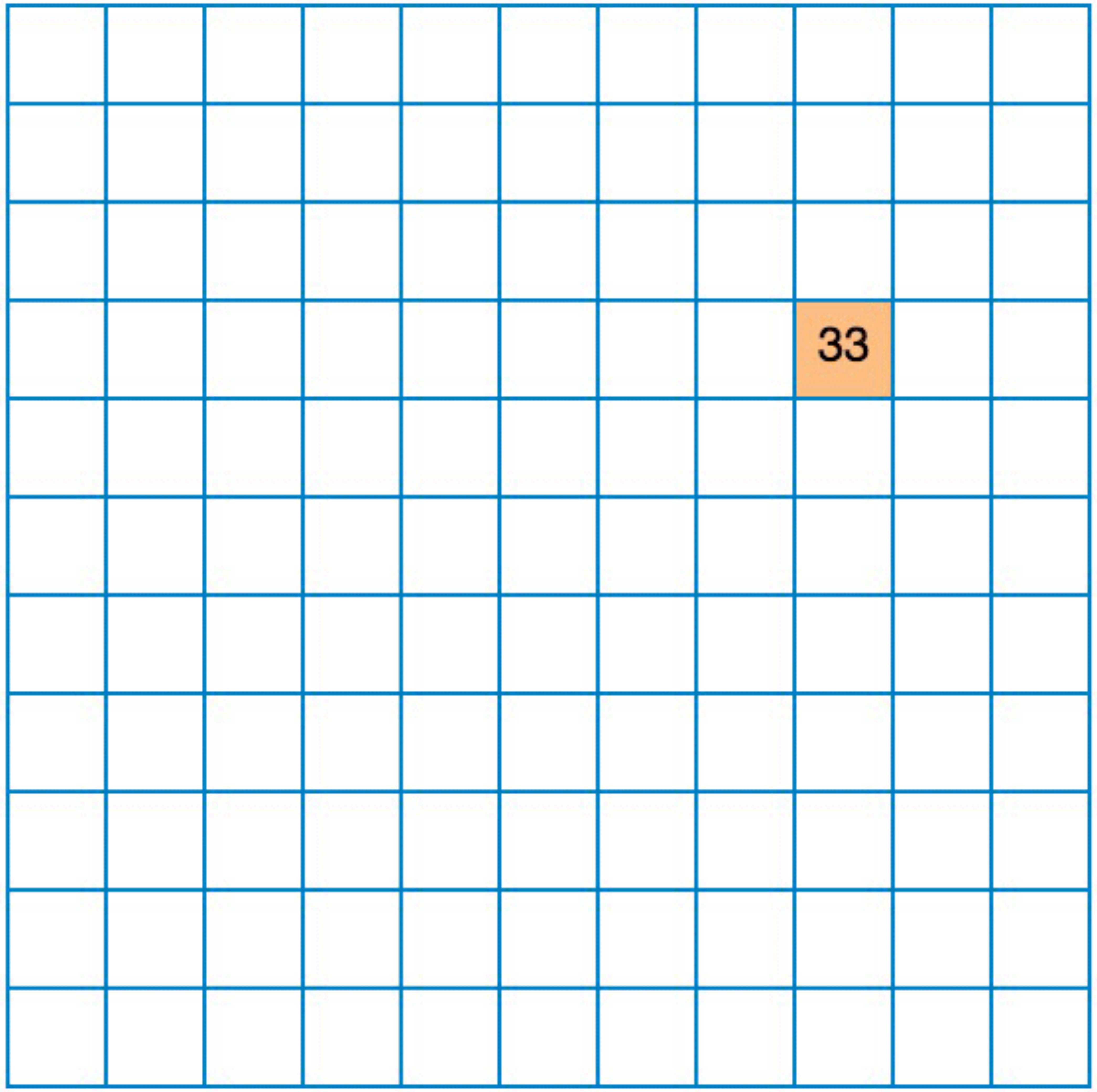
Exploration Exploitation

Herzfeld & Shadmehr (*Nat Neuro* 2014)


Let's explore!
But where?

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

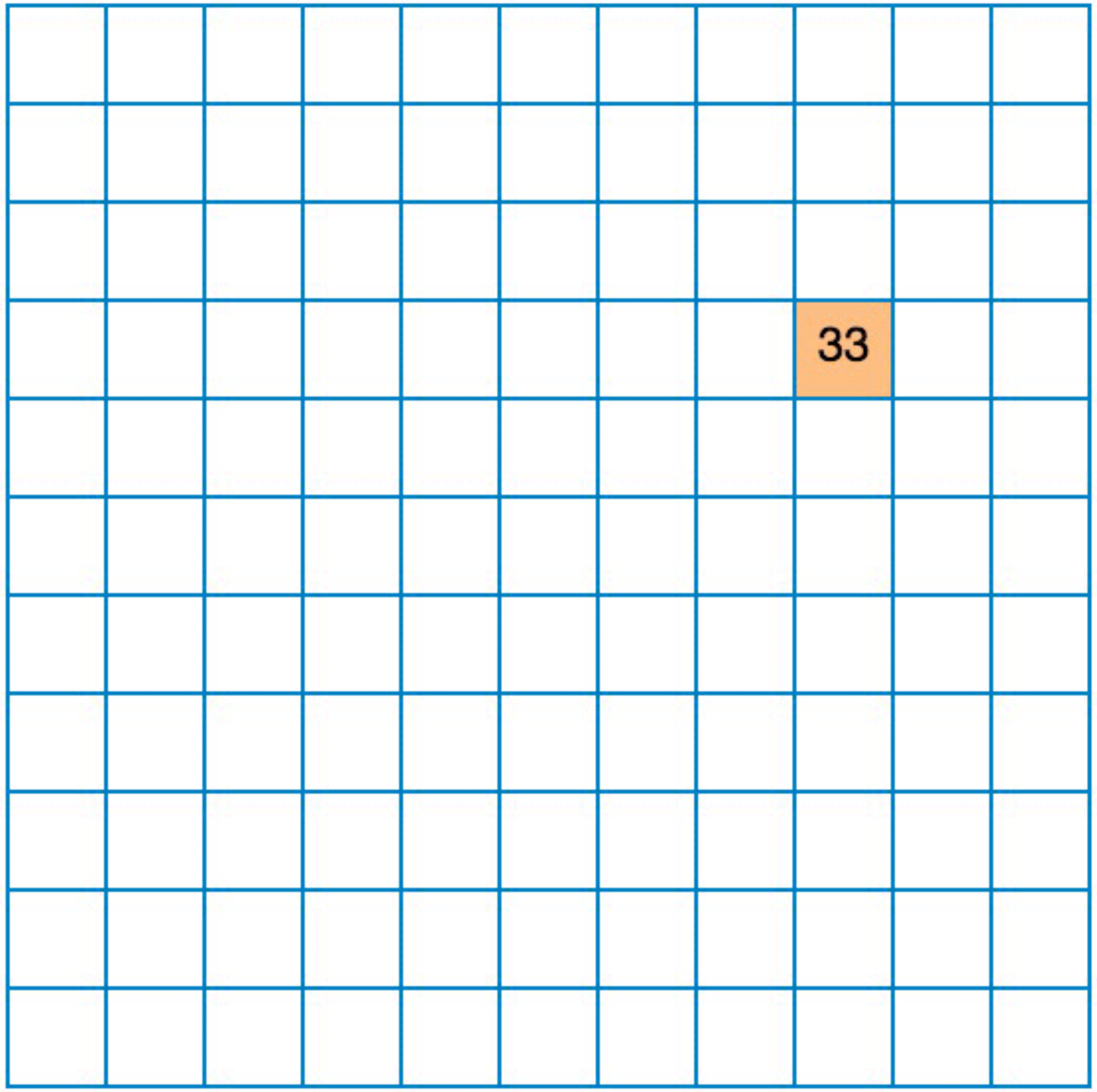
Spatially Correlated Bandit



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

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

Spatially Correlated Bandit



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

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

Spatially Correlated Bandit

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

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

click tiles on the grid

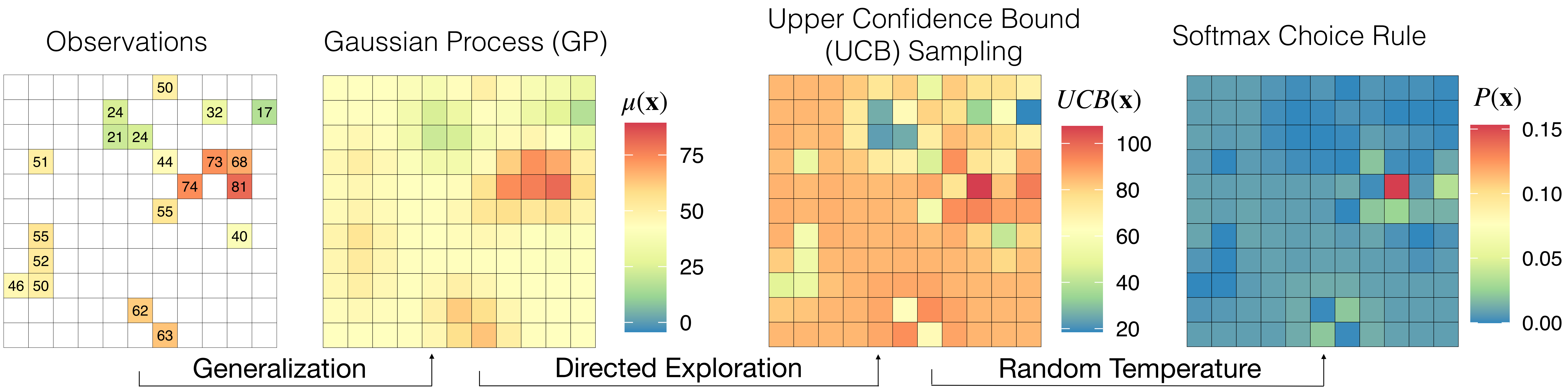
maximize reward

each tile has normally distributed rewards

limited search horizon

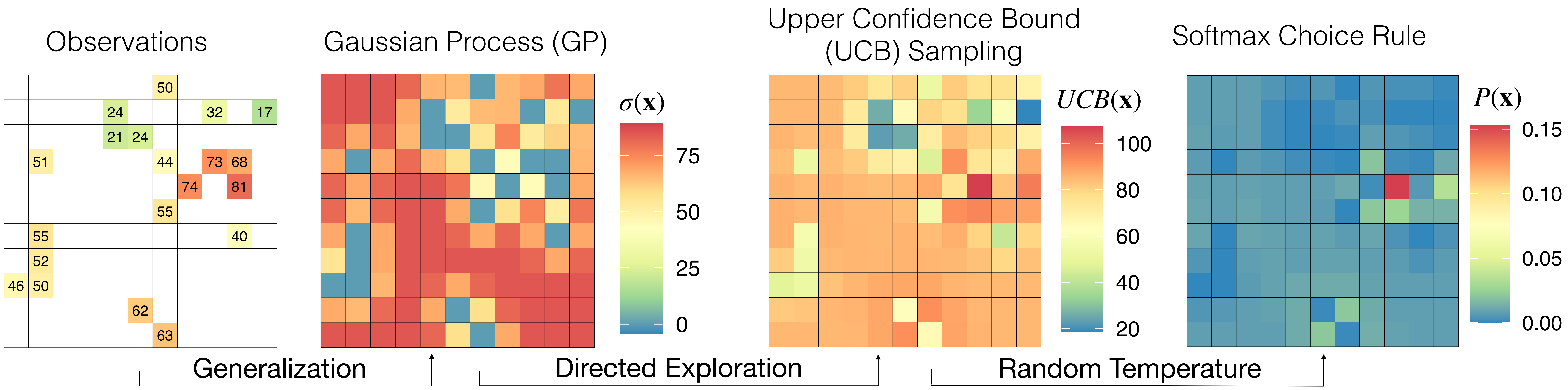
nearby tiles have similar rewards

GP-UCB Model



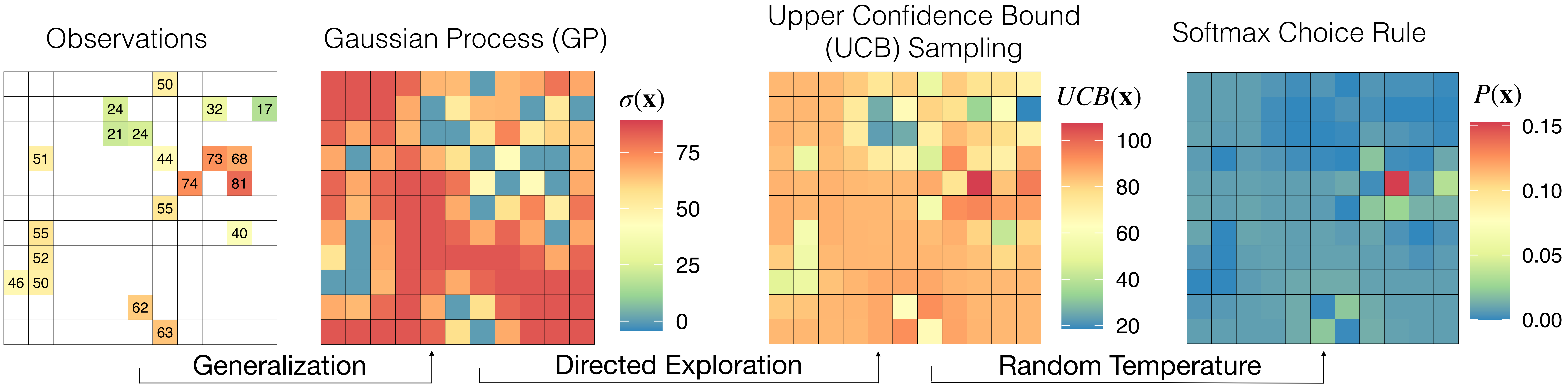
Observations
Gaussian Process (GP)
Upper Confidence Bound (UCB) Sampling
Softmax Choice Rule
50
24 32 17
21 24
51 44 73 68
74 81
55
55 40
52
46 50
62
63
μ(x)
75
50
25
0
UCB(x)
100
80
60
40
20
P(x)
0.15
0.10
0.05
0.00
Generalization
Directed Exploration
Random Temperature

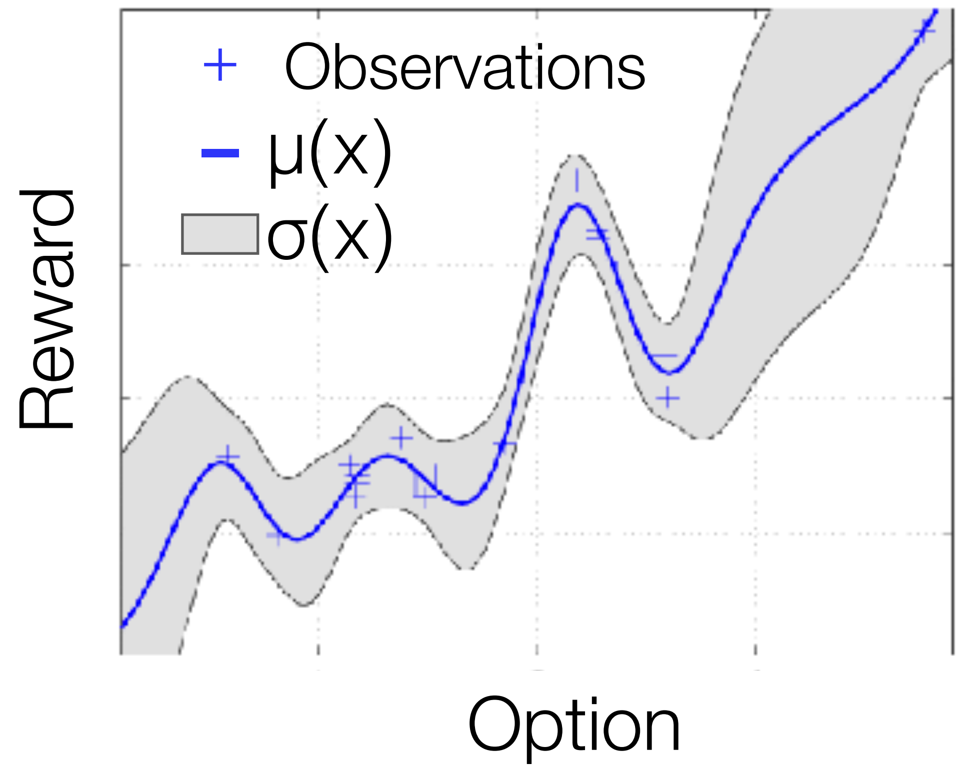
GP-UCB Model



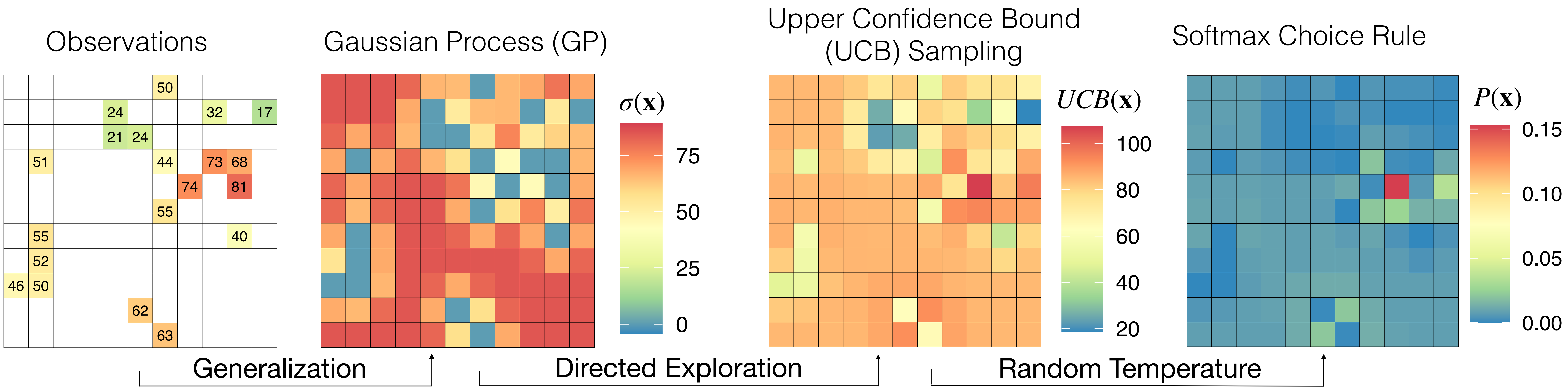
Observations
Gaussian Process (GP)
Upper Confidence Bound (UCB) Sampling
Softmax Choice Rule
50
24
32
17
21
24
51
44
73
68
74
81
55
55
40
52
46
50
62
63
$\sigma(\mathbf{x})$
75
50
25
0
$UCB(\mathbf{x})$
100
80
60
40
20
$P(\mathbf{x})$
0.15
0.10
0.05
0.00
Generalization
Directed Exploration
Random Temperature

GP-UCB Model

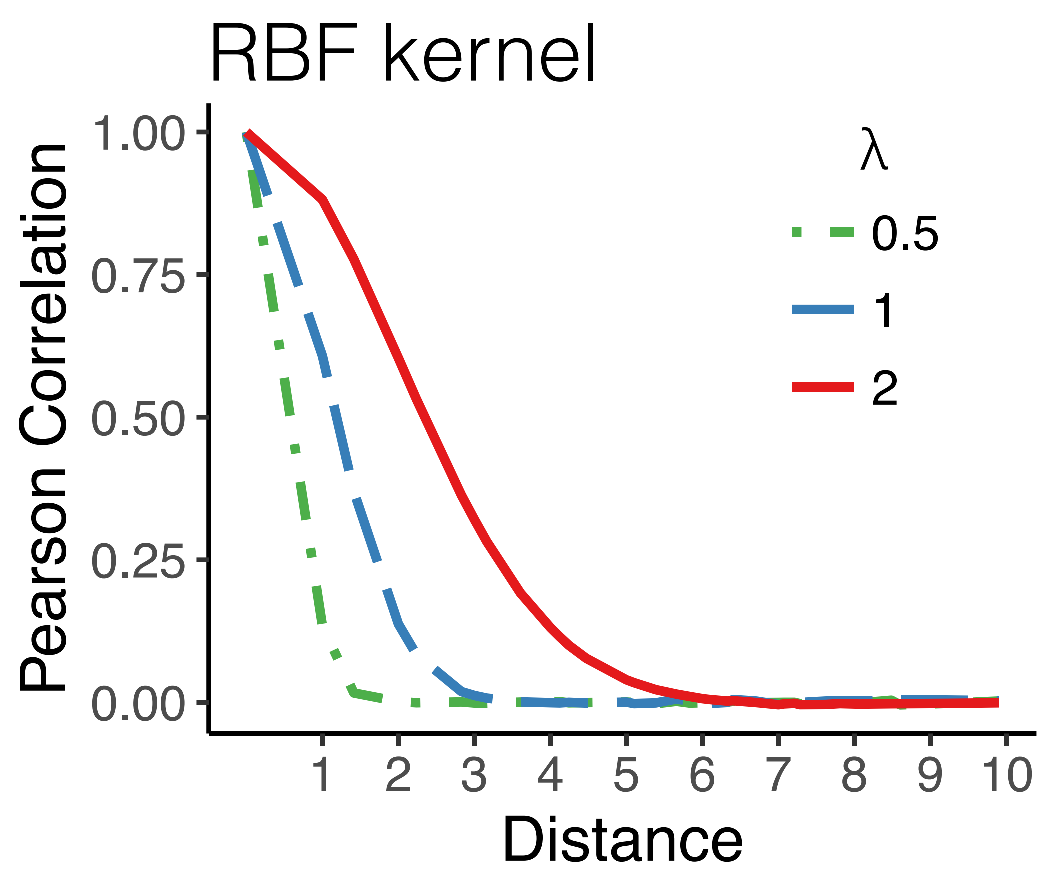

Observations
Gaussian Process (GP)
Upper Confidence Bound (UCB) Sampling
Softmax Choice Rule
50
24
32
17
21
24
51
44
73
68
74
81
55
55
40
52
46
50
62
63
σ(x)
75
50
25
0
UCB(x)
100
80
60
40
20
P(x)
0.15
0.10
0.05
0.00
Generalization
Directed Exploration
Random Temperature


Observations
μ(x)
σ(x)
Reward
Option

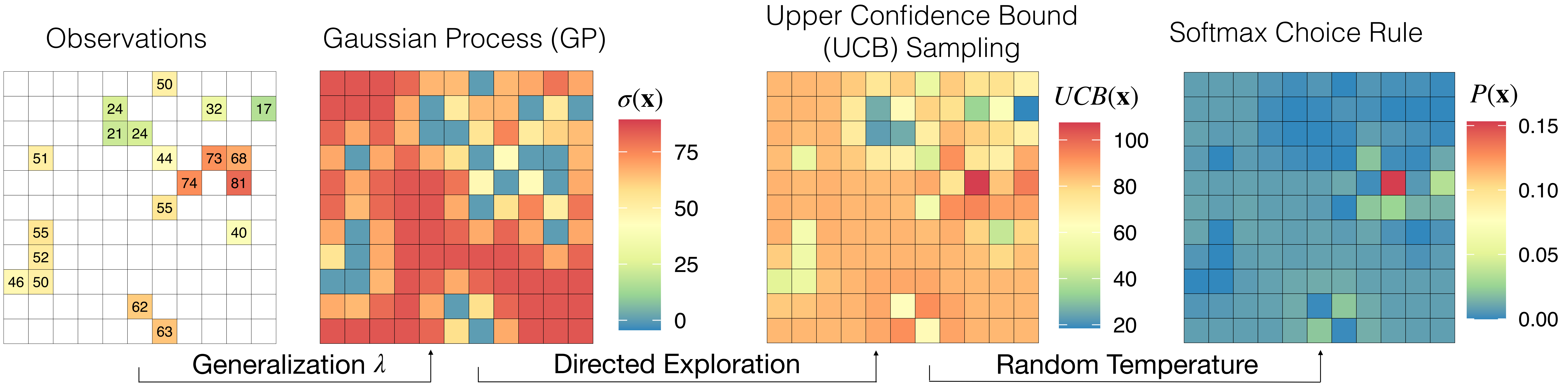
GP-UCB Model



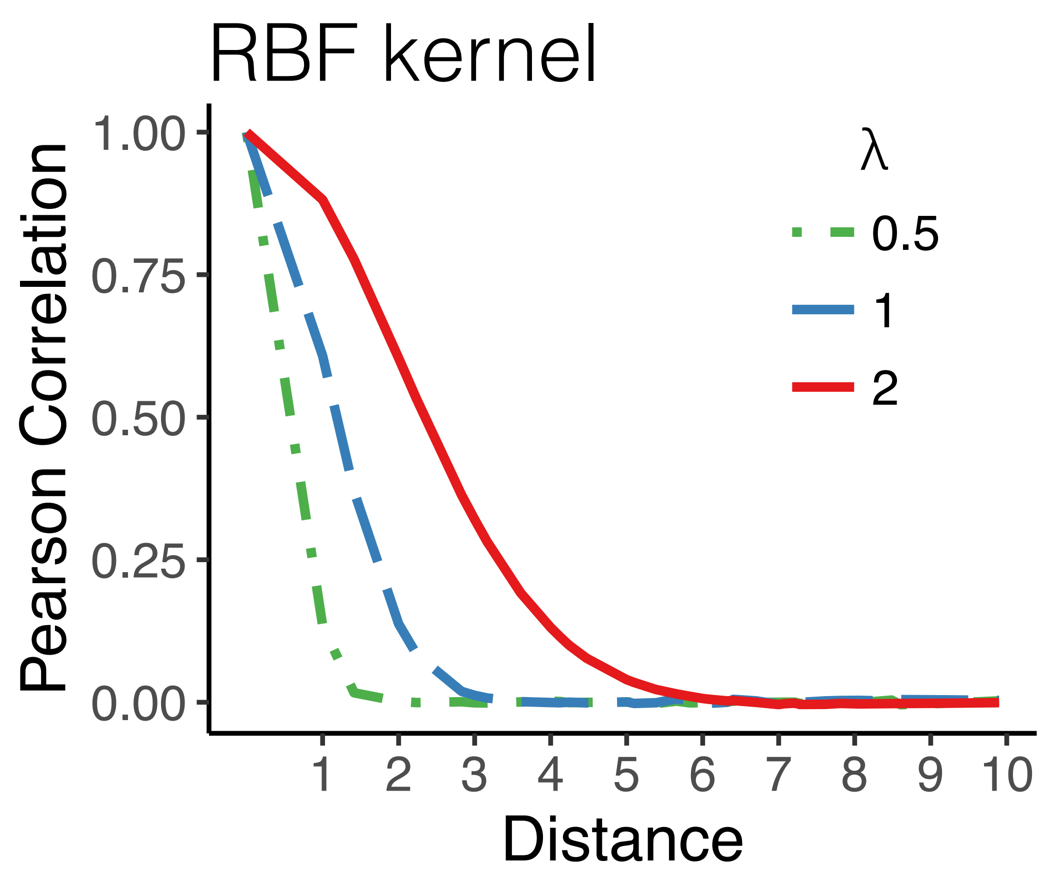
$$k_{RBF}(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{||\mathbf{x} - \mathbf{x}'||^2}{2\lambda^2}\right)$$



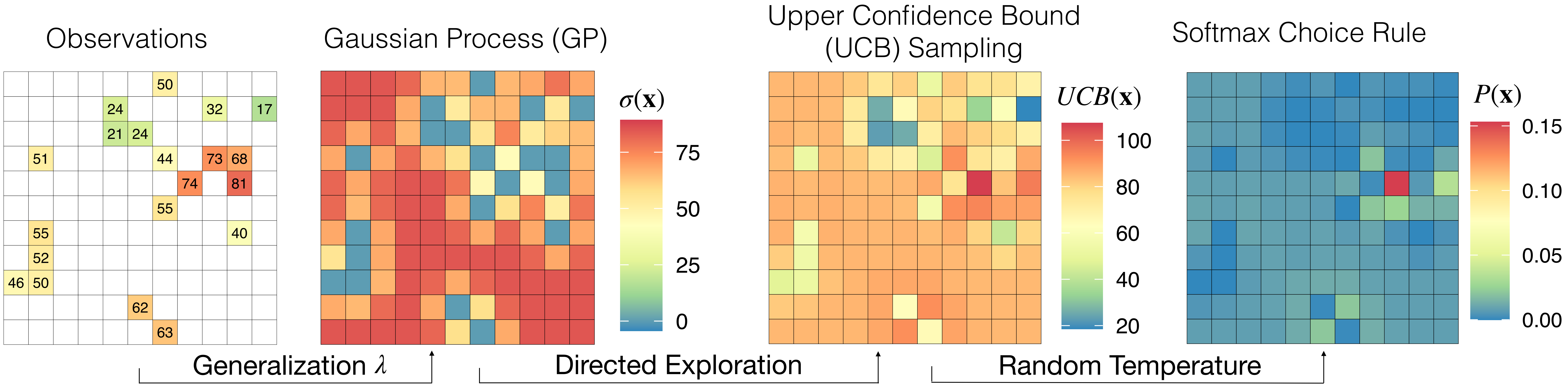
GP-UCB Model



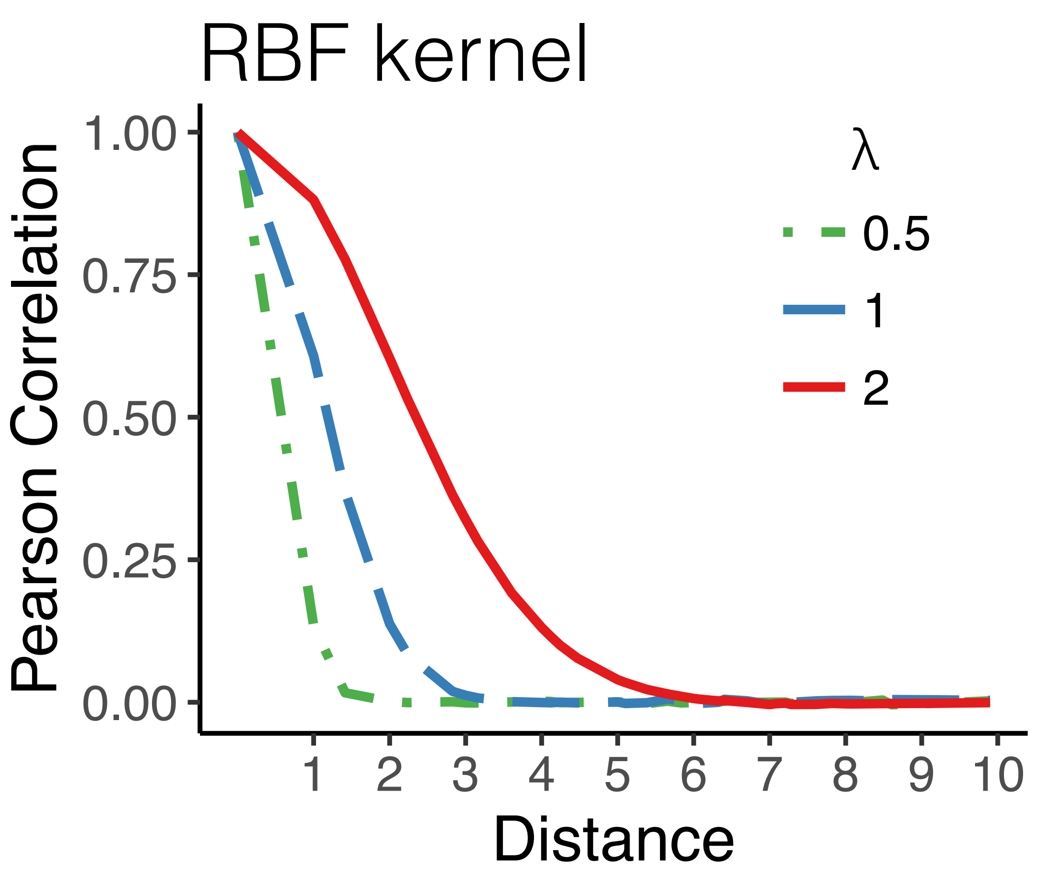
$$k_{RBF}(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{||\mathbf{x} - \mathbf{x}'||^2}{2\lambda^2}\right)$$

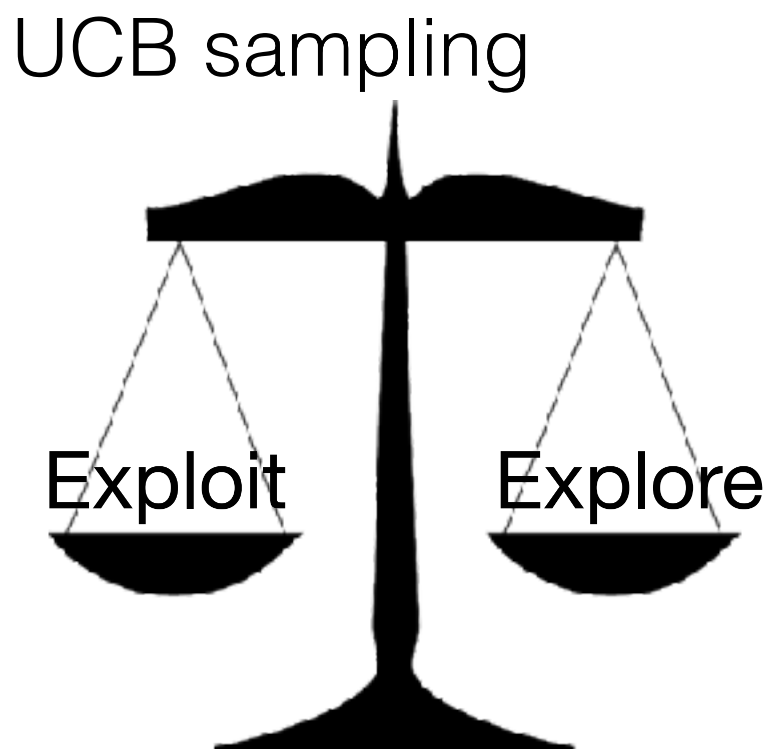


GP-UCB Model

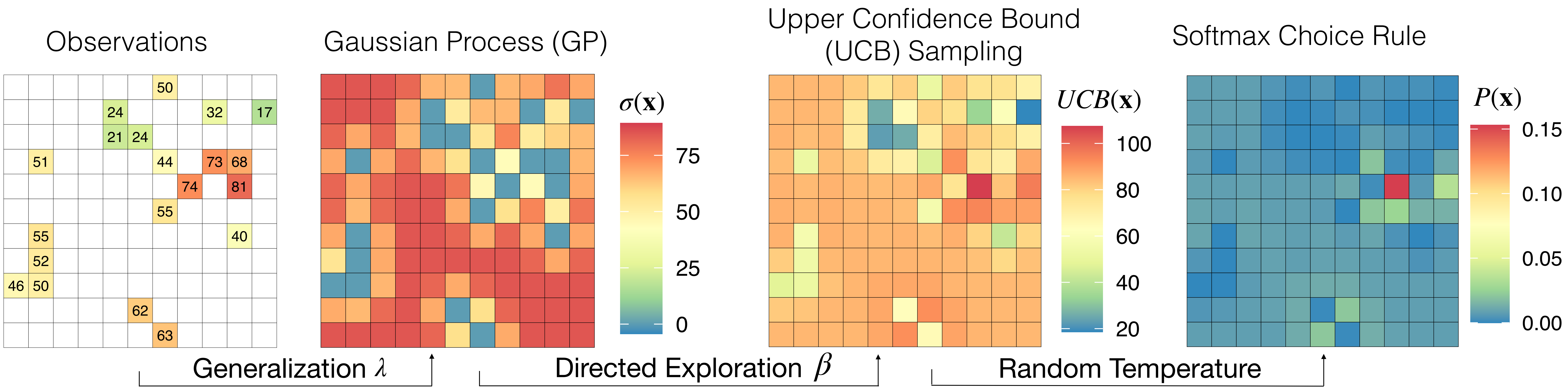


$$k_{RBF}(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{||\mathbf{x} - \mathbf{x}'||^2}{2\lambda^2}\right) \qquad UCB(\mathbf{x}) = \mu(\mathbf{x}) + \beta\sigma(\mathbf{x})$$

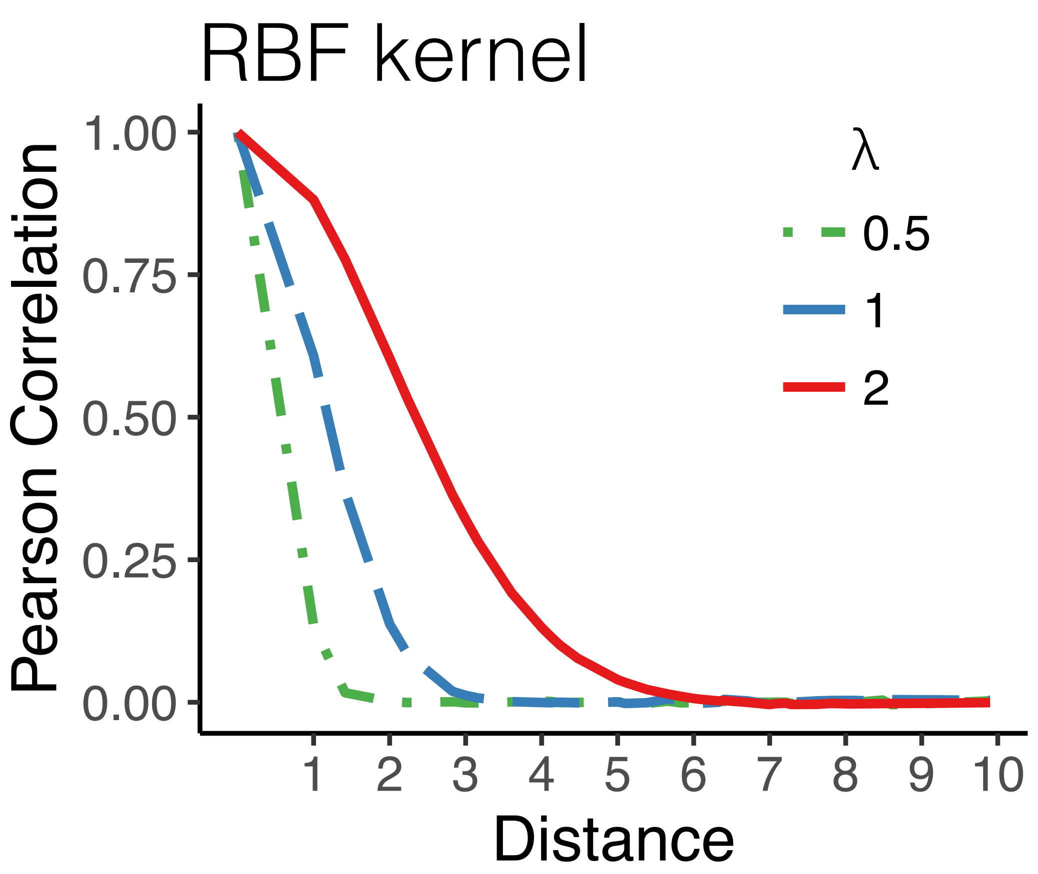


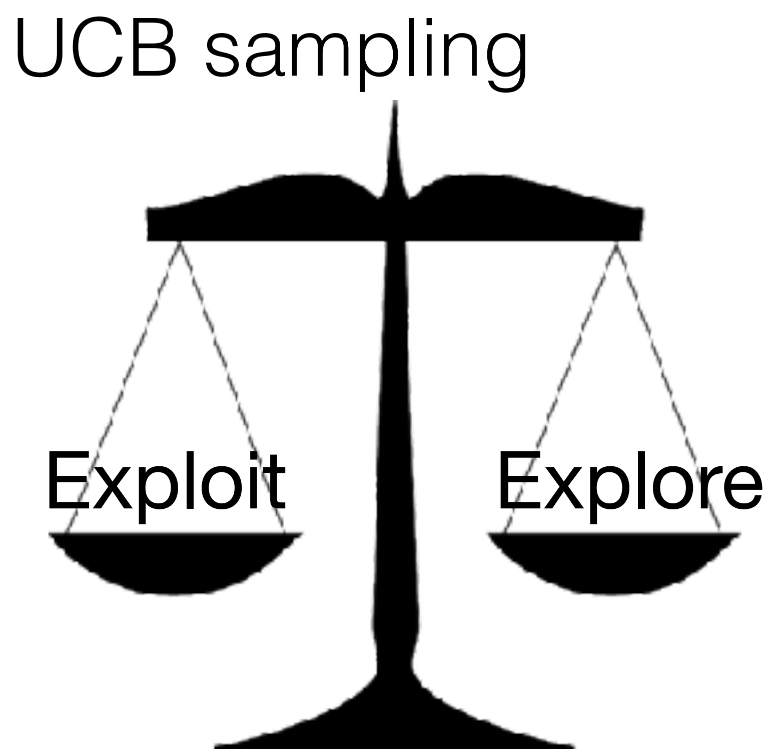


GP-UCB Model



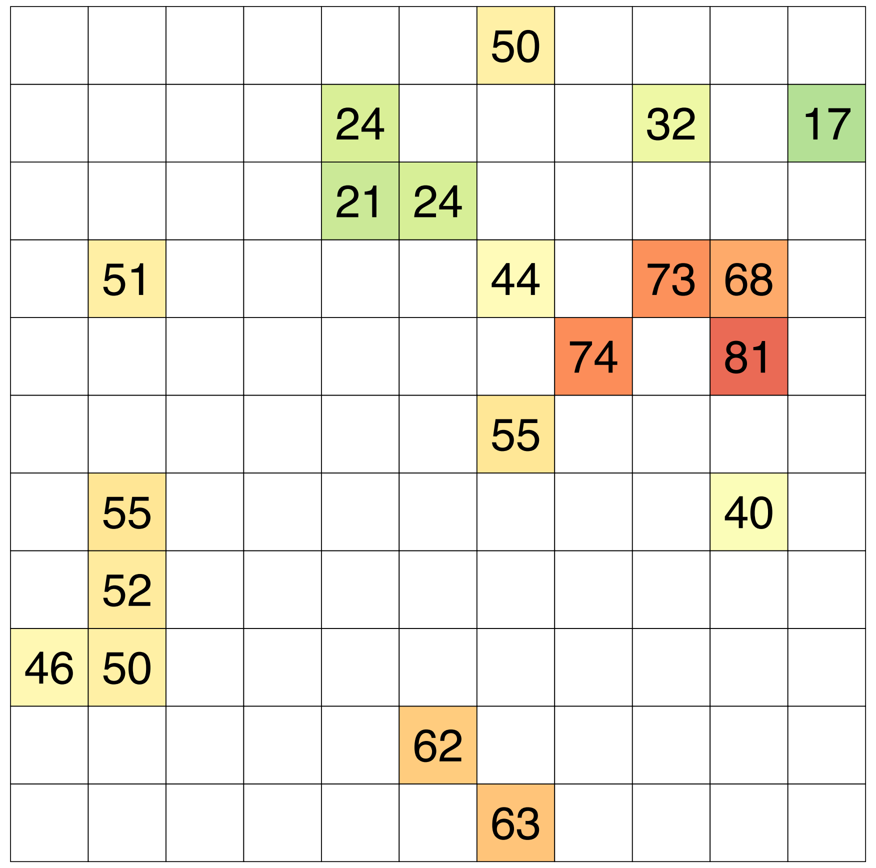
$$k_{RBF}(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{||\mathbf{x} - \mathbf{x}'||^2}{2\lambda^2}\right) \qquad UCB(\mathbf{x}) = \mu(\mathbf{x}) + \beta\sigma(\mathbf{x})$$



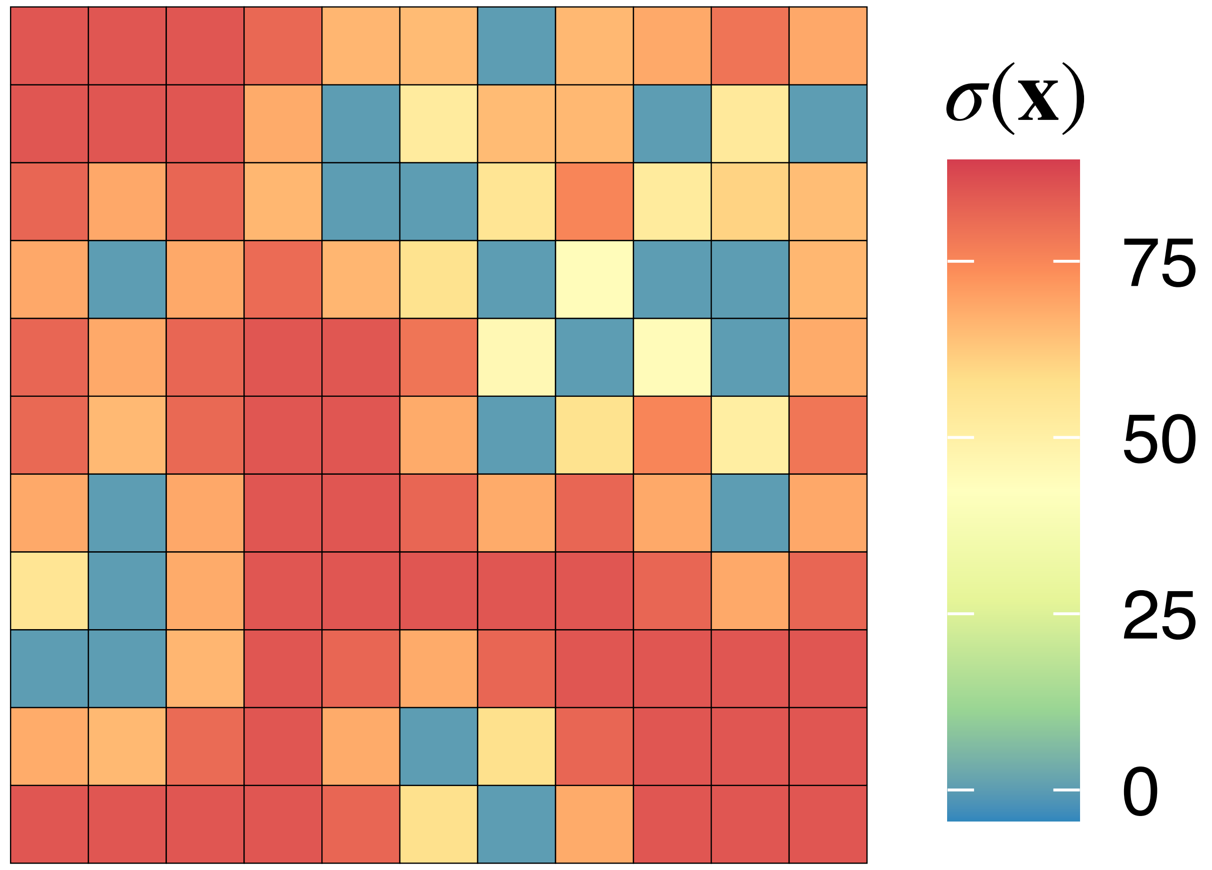


GP-UCB Model

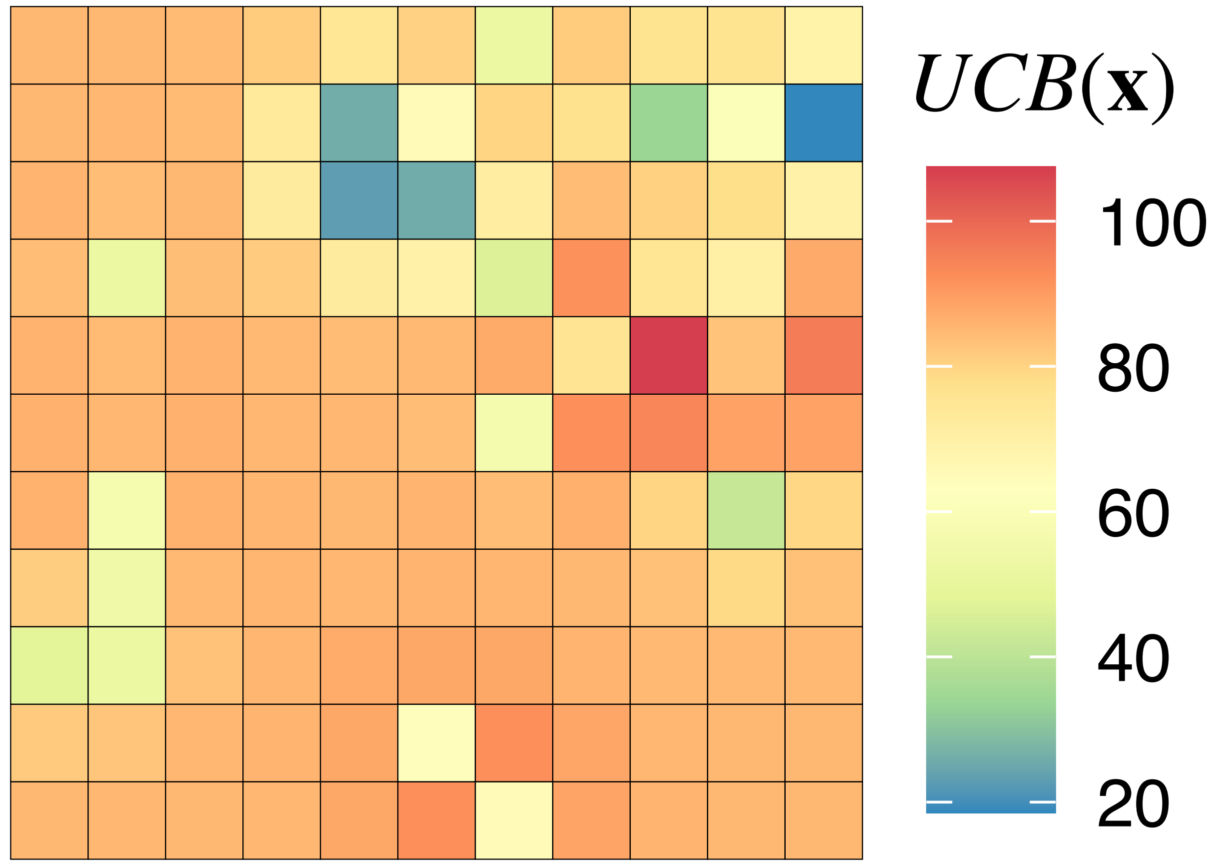
Observations



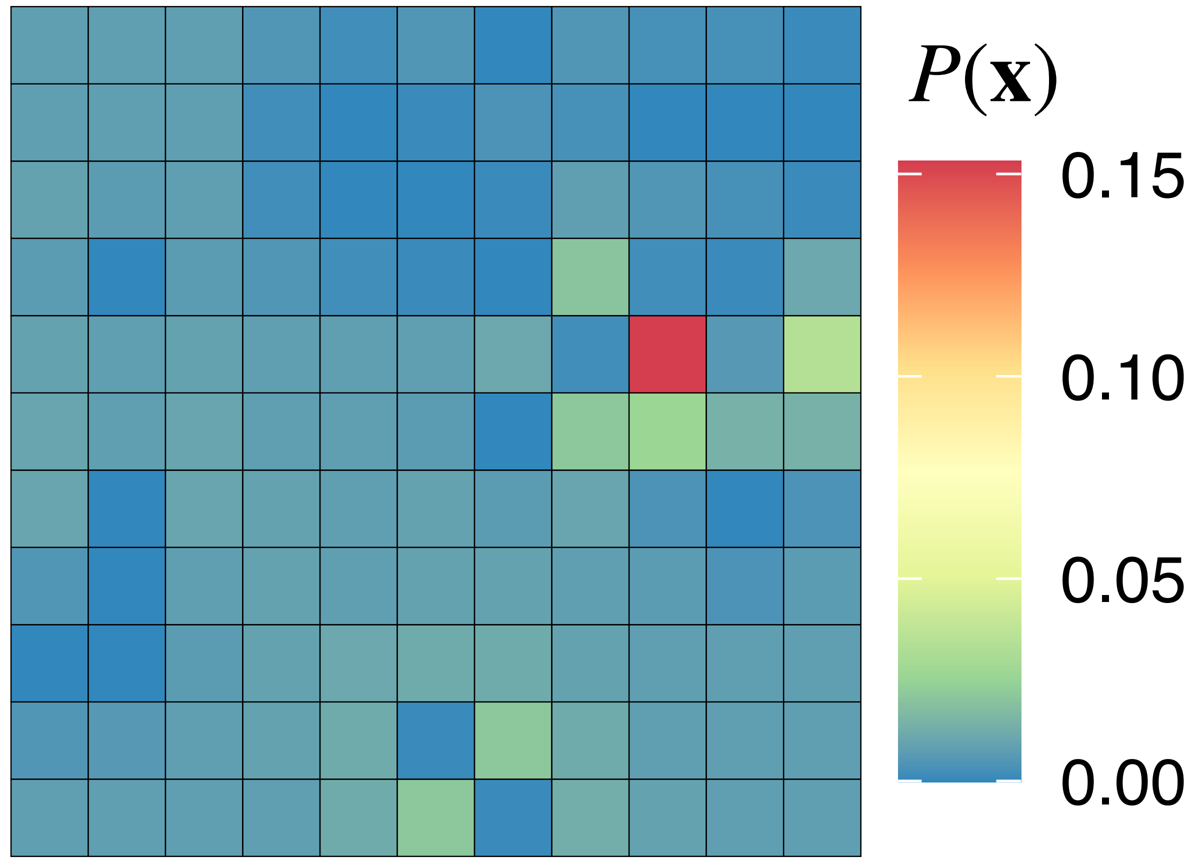
Gaussian Process (GP)



Upper Confidence Bound (UCB) Sampling



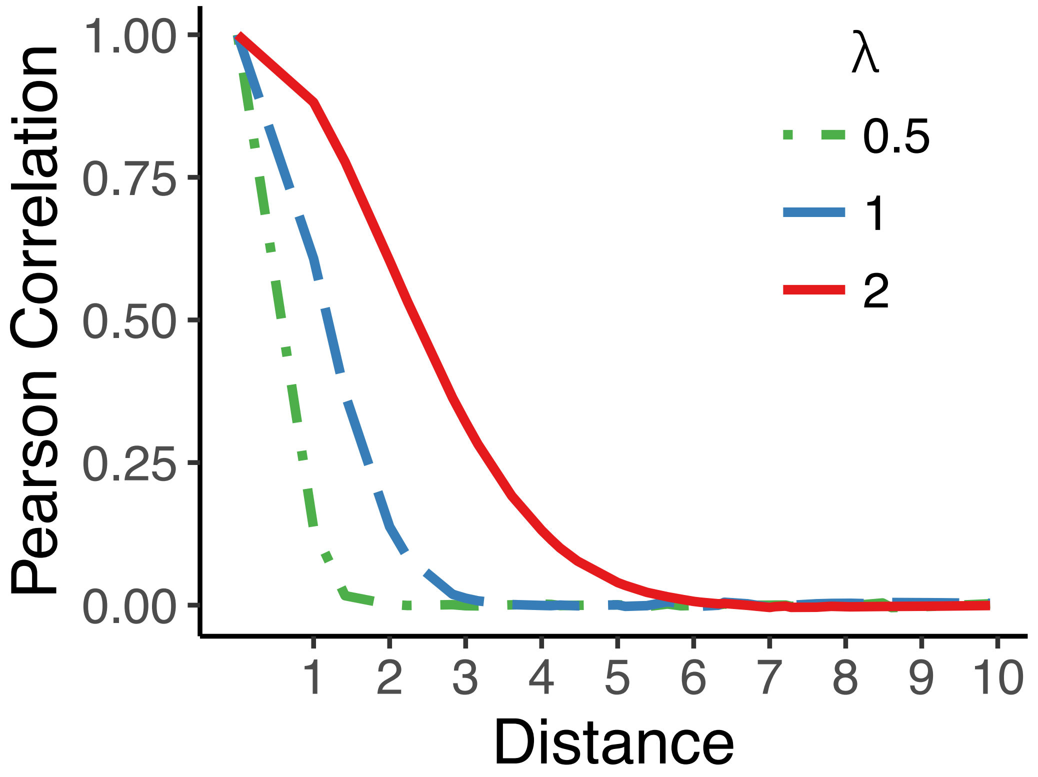
Softmax Choice Rule



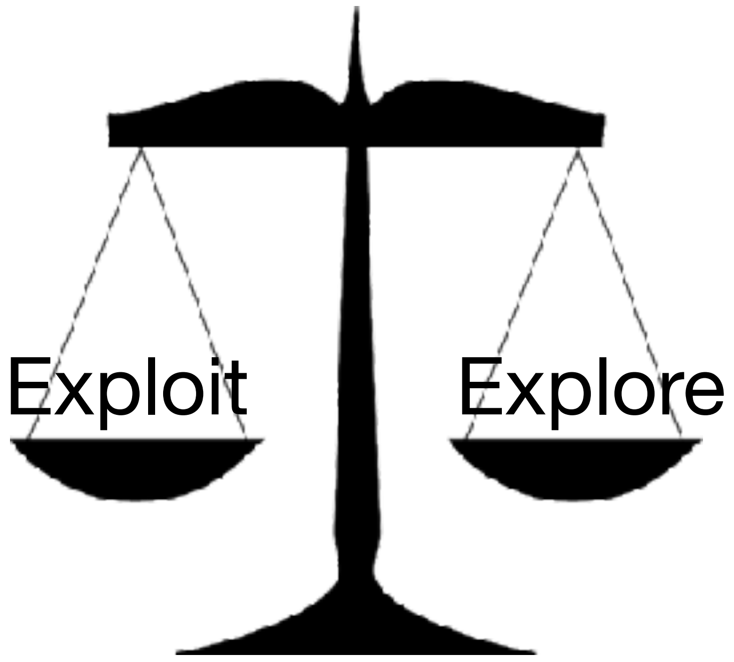
Generalization λ — Directed Exploration β — Random Temperature

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

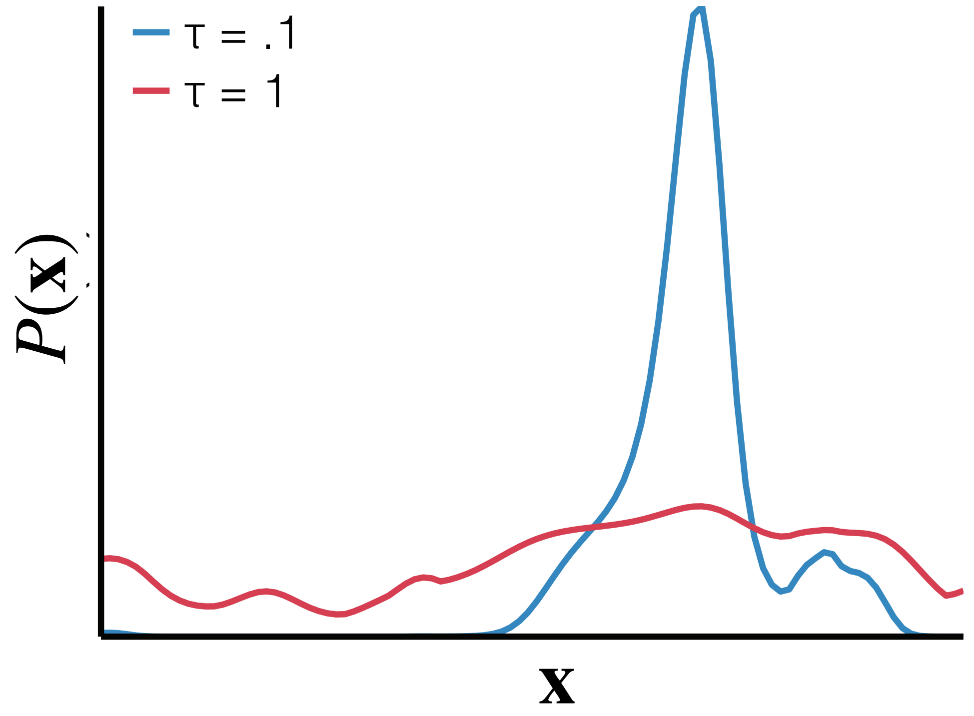
RBF kernel



UCB sampling

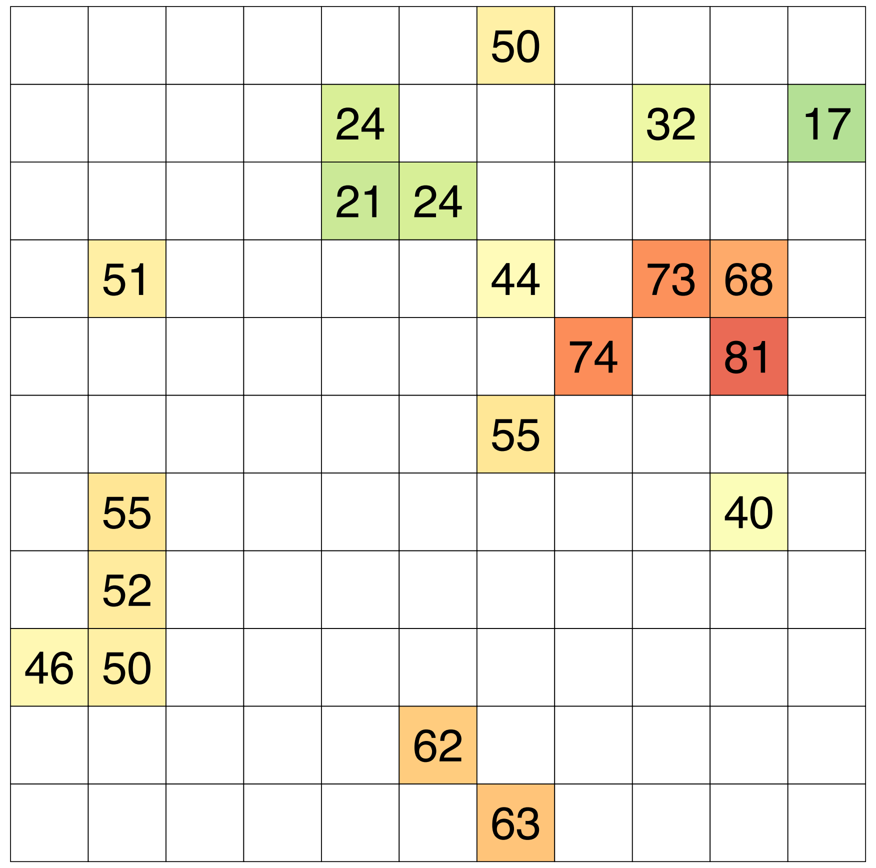


Softmax

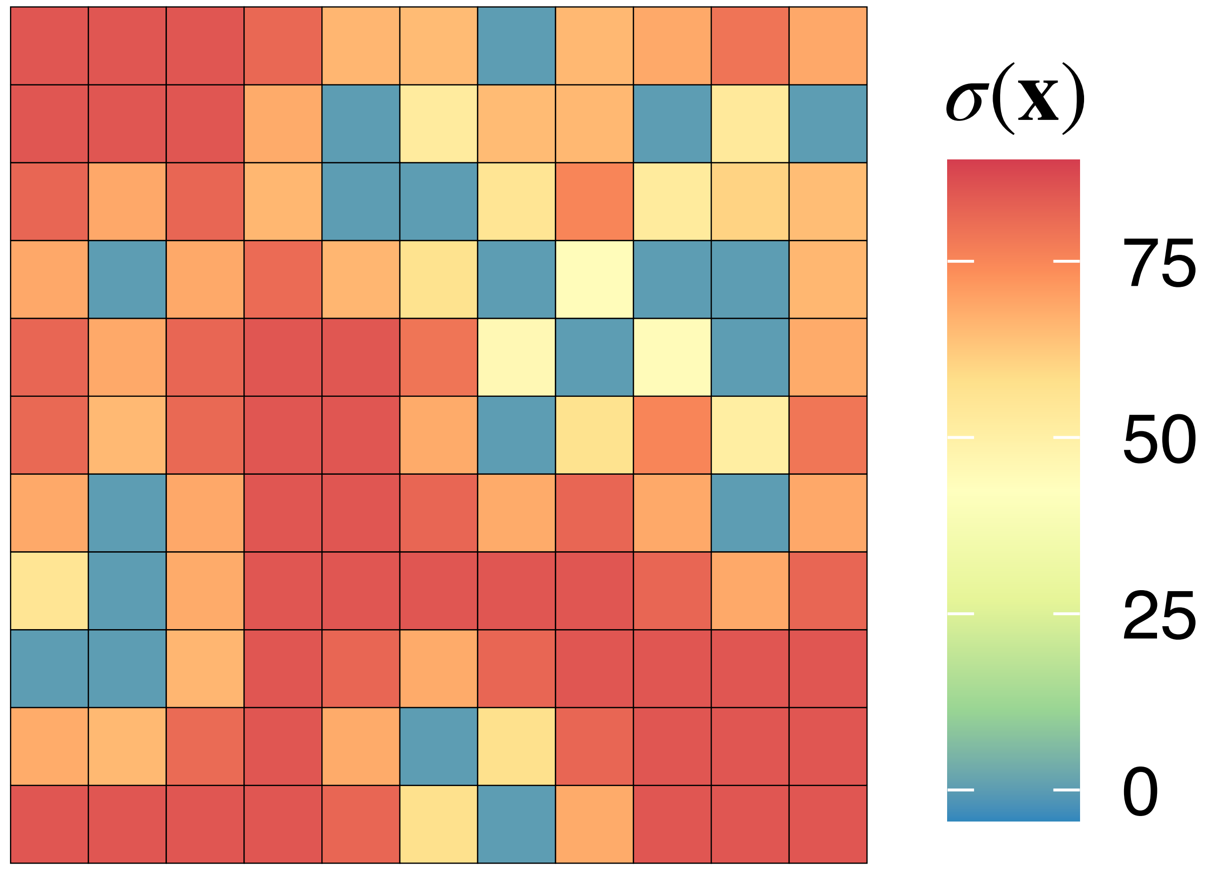


GP-UCB Model

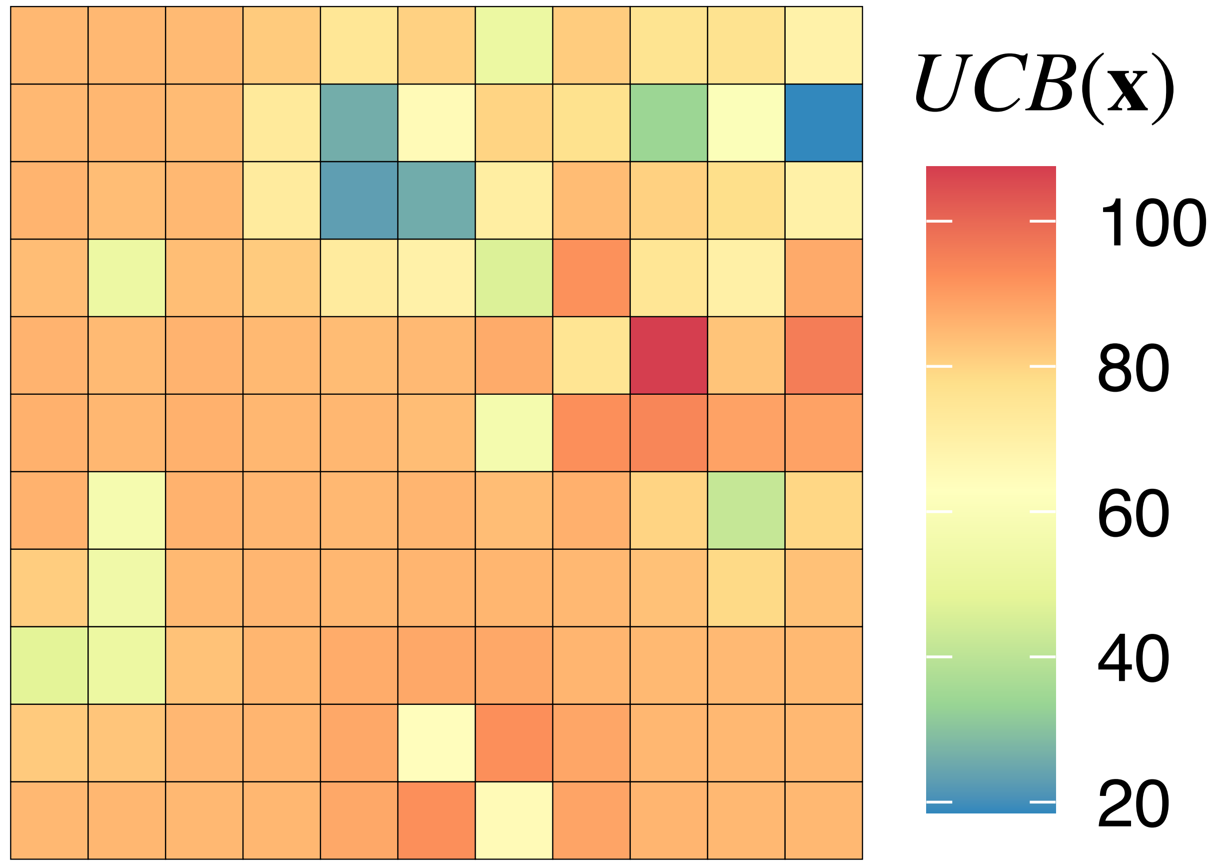
Observations



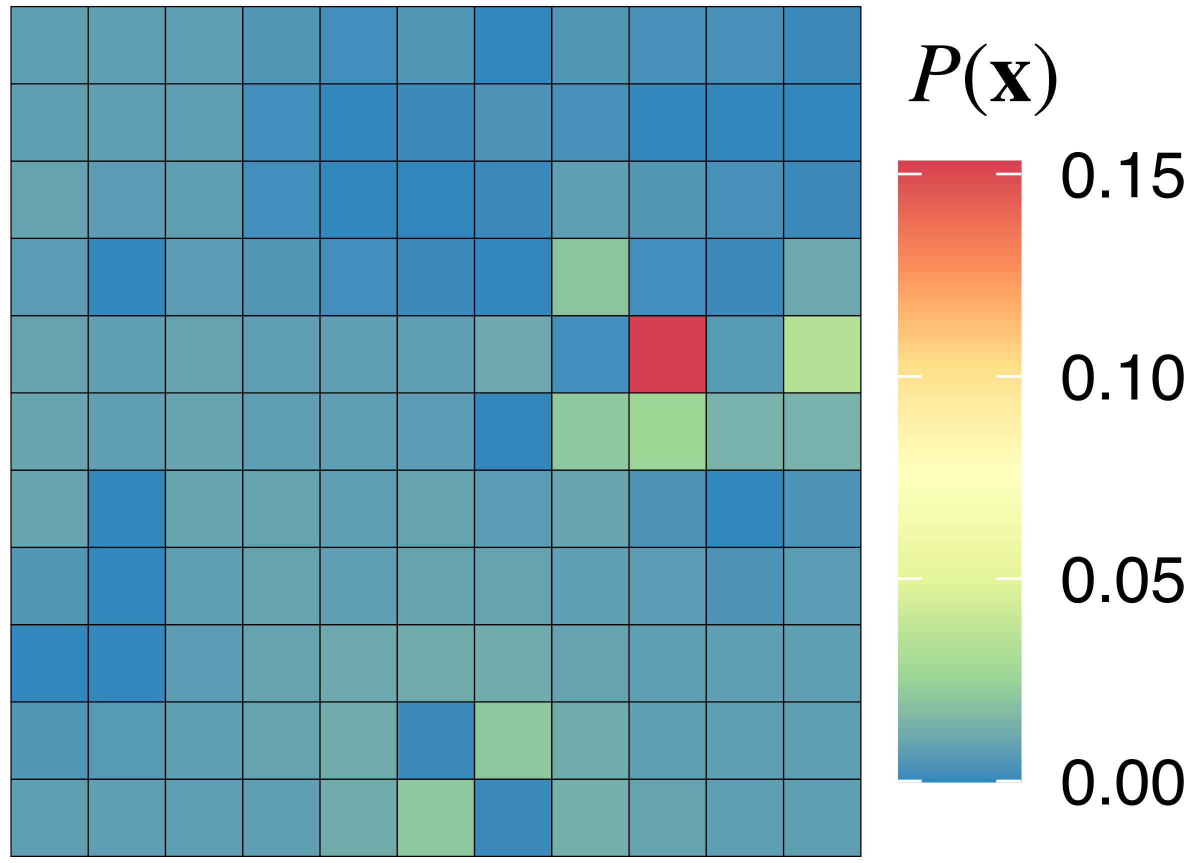
Gaussian Process (GP)



Upper Confidence Bound (UCB) Sampling



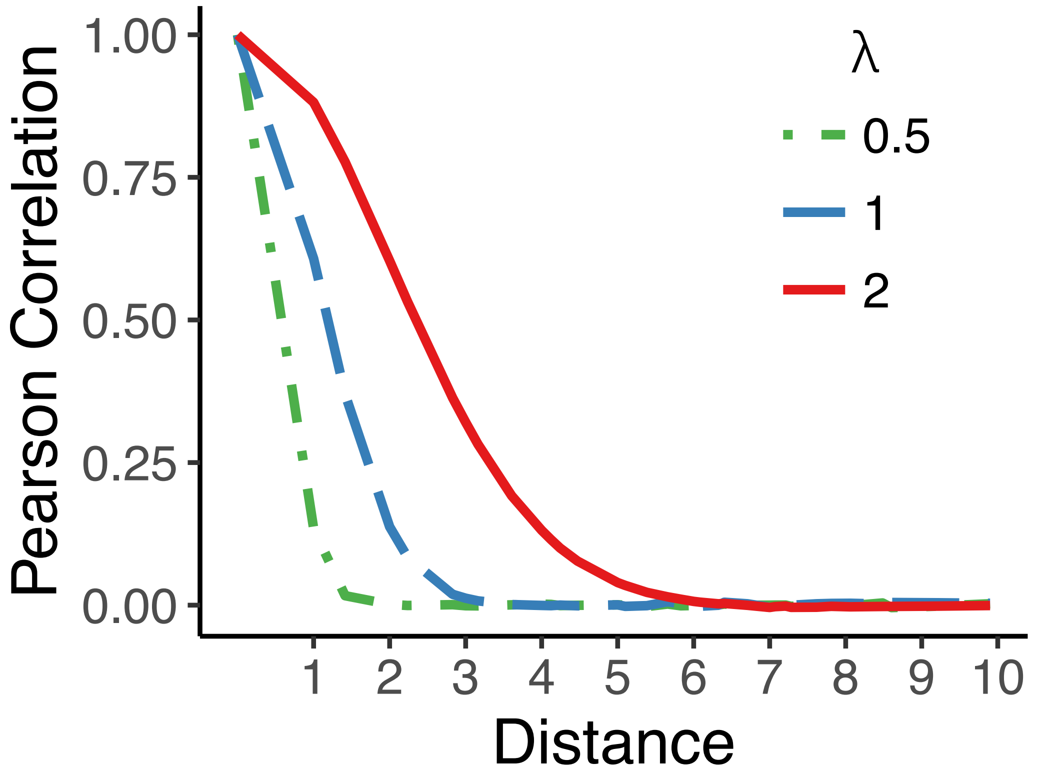
Softmax Choice Rule



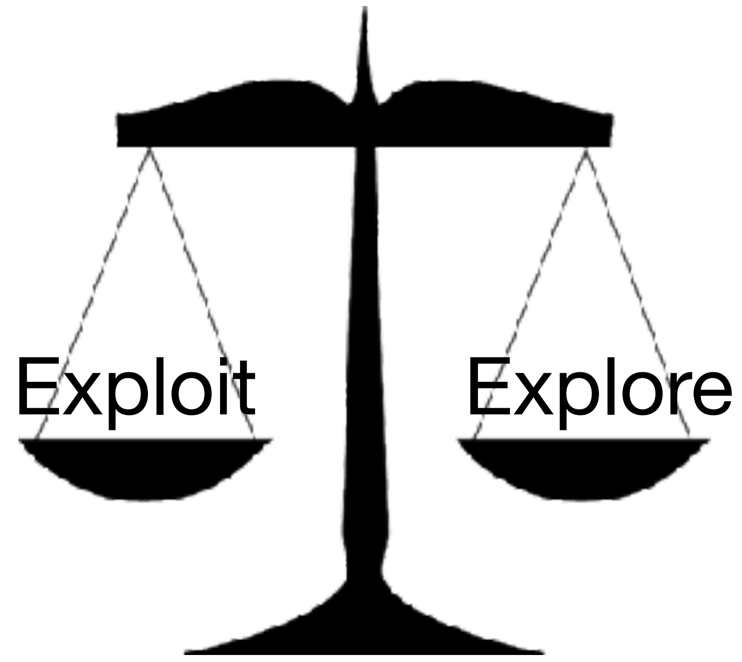
Generalization λ | Directed Exploration β | Random Temperature τ

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

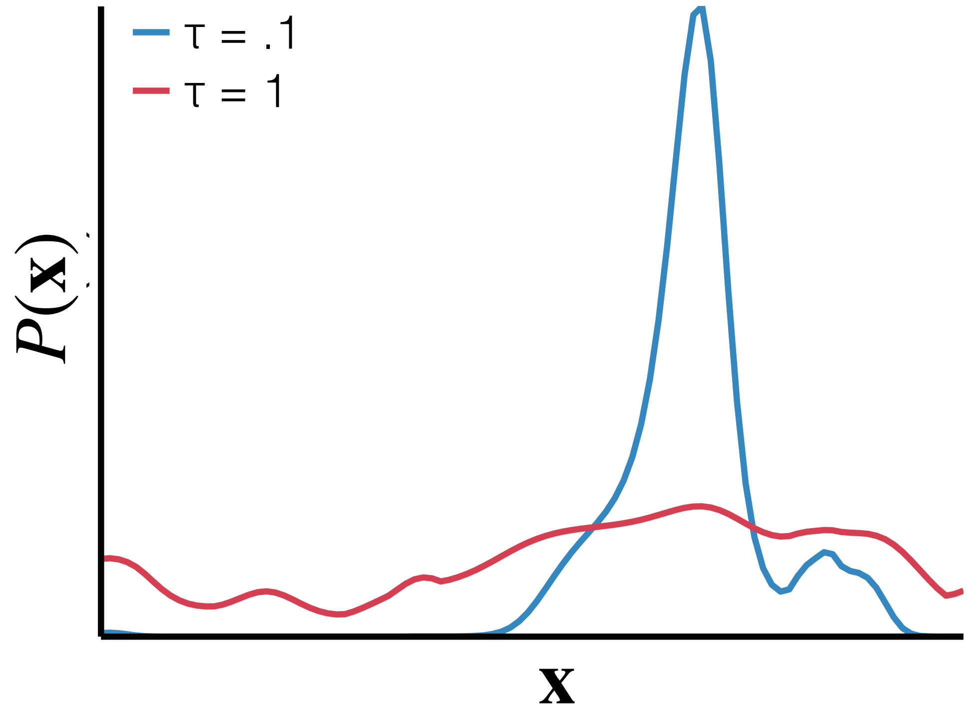
RBF kernel



UCB sampling



Softmax

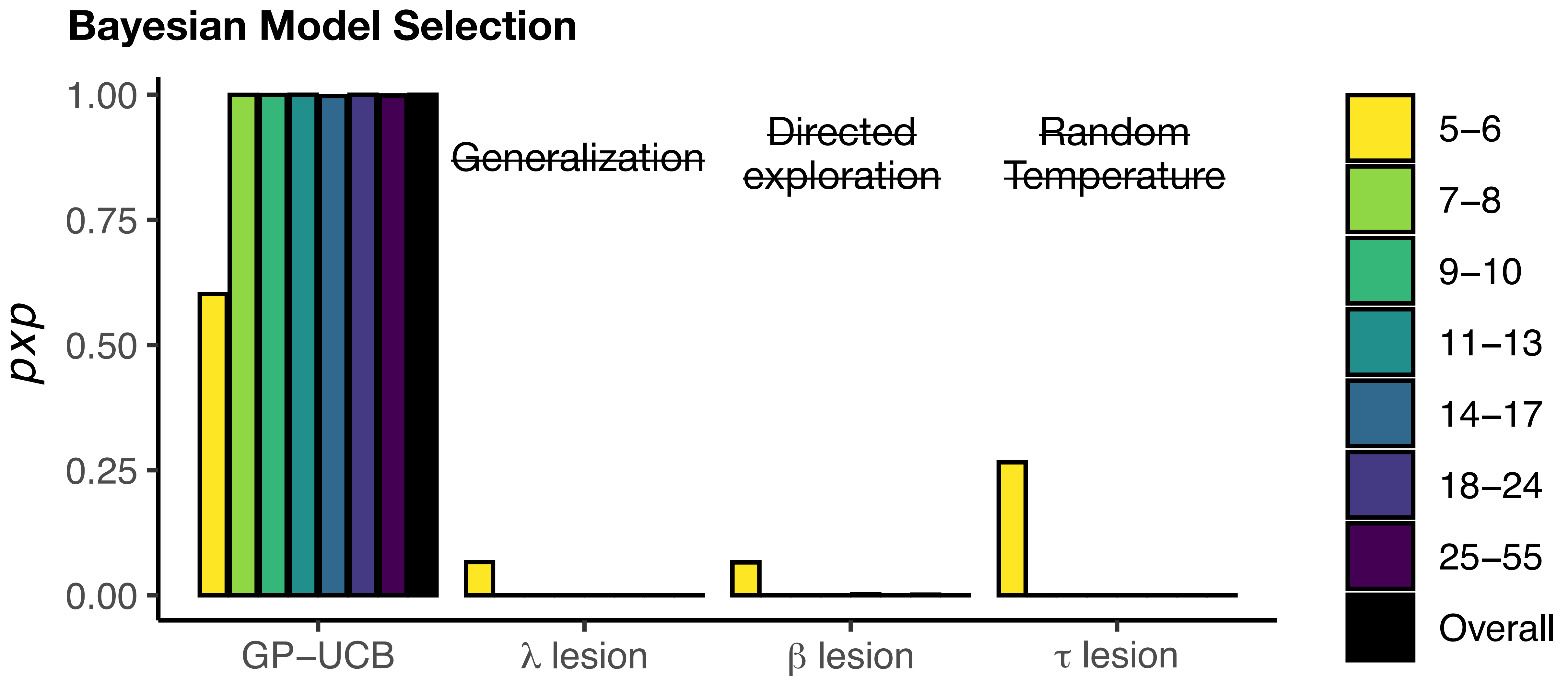


GP-UCB across the lifespan

- GP-UCB provides the best account of behavior from the ages of 5 to 55 (*n*=281)

GP-UCB across the lifespan

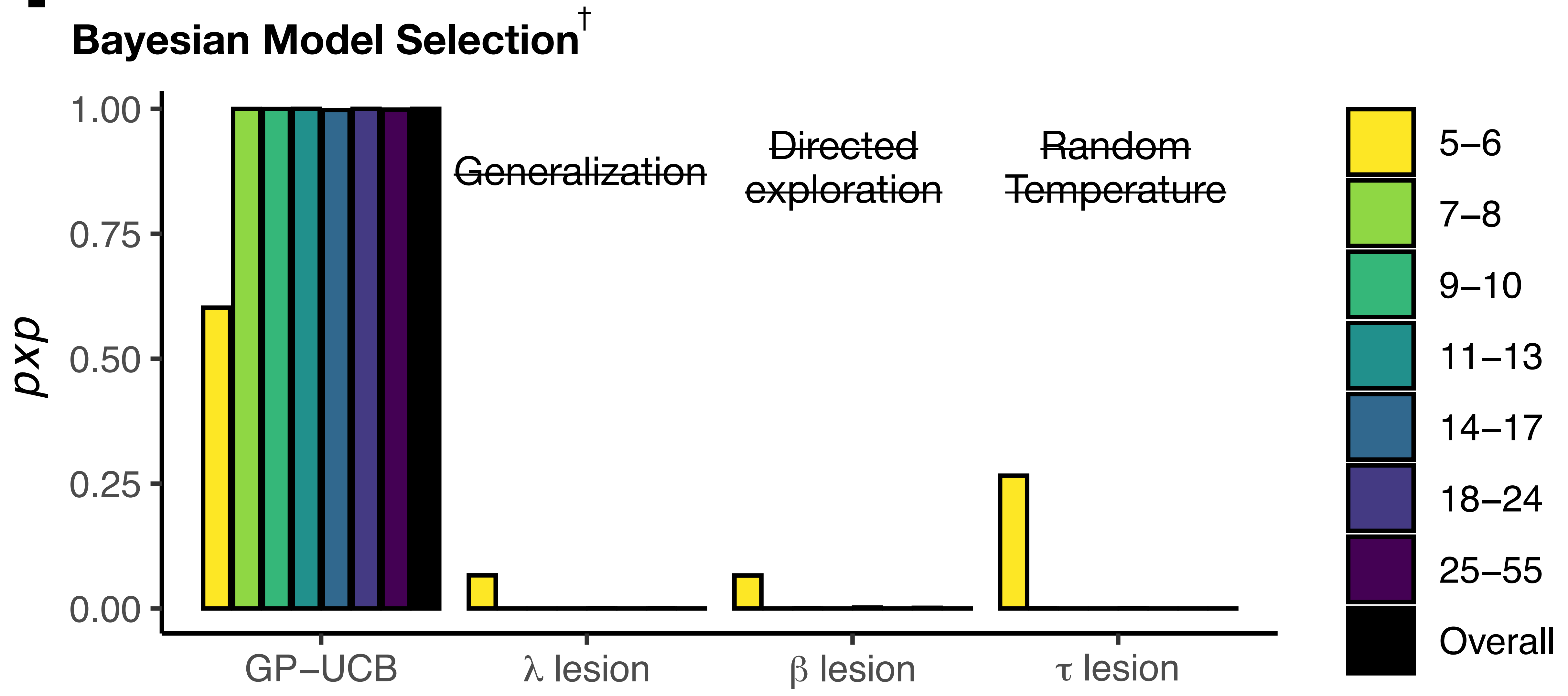
- GP-UCB provides the best account of behavior from the ages of 5 to 55 (*n*=281)
- We can lesion out each component to show that all are necessary
    - λ lesion replaces GP with a Bayesian RL model (i.e., Kalman filter) that learns the value of each option independently without generalization
    - β lesion removes uncertainty-directed exploration by setting $\beta = 0$
    - τ lesion swaps softmax for an ϵ-greedy policy



†Protected Exceedance Probability (Rigoux et al., 2014)

GP-UCB across the lifespan

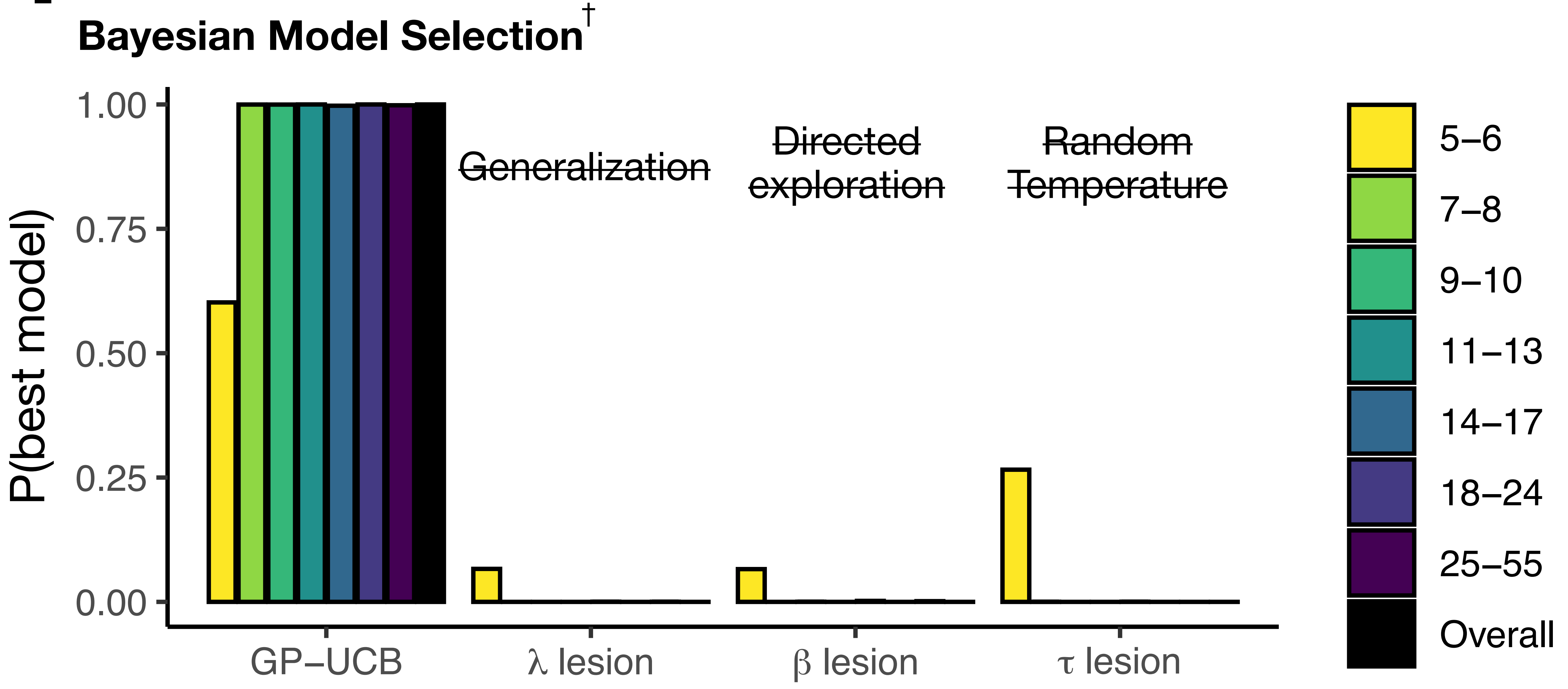
- GP-UCB provides the best account of behavior from the ages of 5 to 55 (n=281)
- We can lesion out each component to show that all are necessary
    - λ lesion replaces GP with a Bayesian RL model (i.e., Kalman filter) that learns the value of each option independently without generalization
    - β lesion removes uncertainty-directed exploration by setting $\beta = 0$
    - τ lesion swaps softmax for an ϵ-greedy policy



†Protected Exceedance Probability (Rigoux et al., 2014)

GP-UCB across the lifespan

- GP-UCB provides the best account of behavior from the ages of 5 to 55 (*n*=281)
- We can lesion out each component to show that all are necessary
  - λ lesion replaces GP with a Bayesian RL model (i.e., Kalman filter) that learns the value of each option independently without generalization
  - β lesion removes uncertainty-directed exploration by setting $\beta = 0$
  - τ lesion swaps softmax for an ϵ-greedy policy

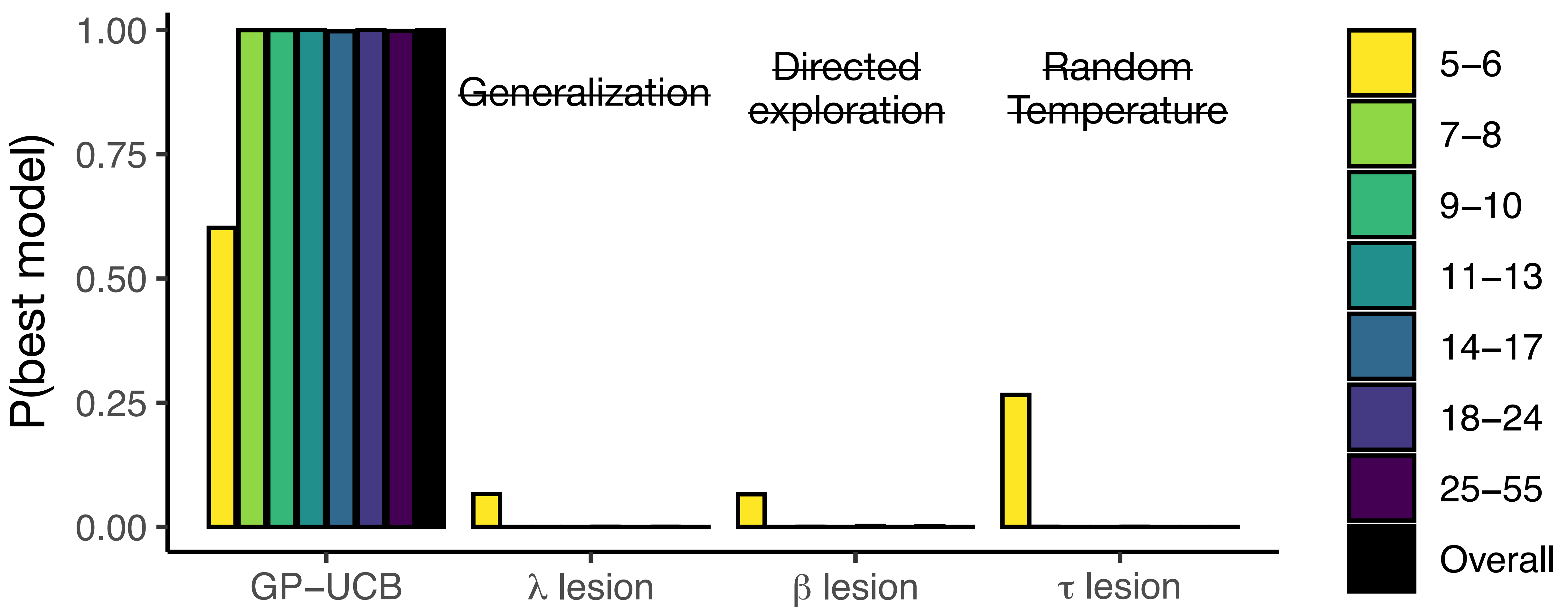


†Protected Exceedance Probability (Rigoux et al., 2014)

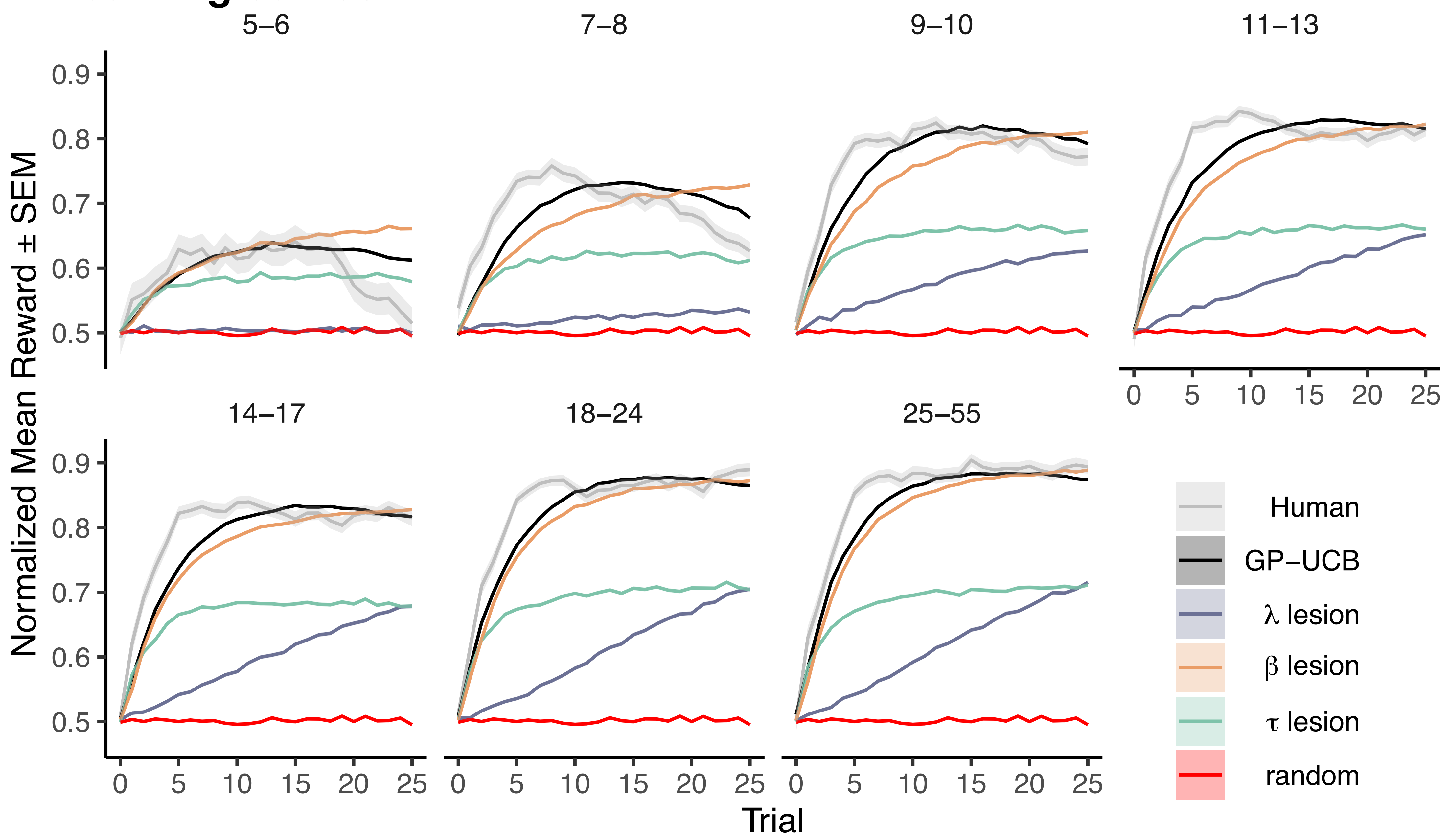
GP-UCB across the lifespan

- GP-UCB provides the best account of behavior from the ages of 5 to 55 (*n*=281)
- We can lesion out each component to show that all are necessary
  - λ lesion replaces GP with a Bayesian RL model (i.e., Kalman filter) that learns the value of each option independently without generalization
  - β lesion removes uncertainty-directed exploration by setting $\beta = 0$
  - τ lesion swaps softmax for an ϵ-greedy policy
- The **full model** reproduces the same age-related differences in learning curves
  - β-lesion is also good, but doesn't produce the same decaying learning curves that children have and generally learns slower

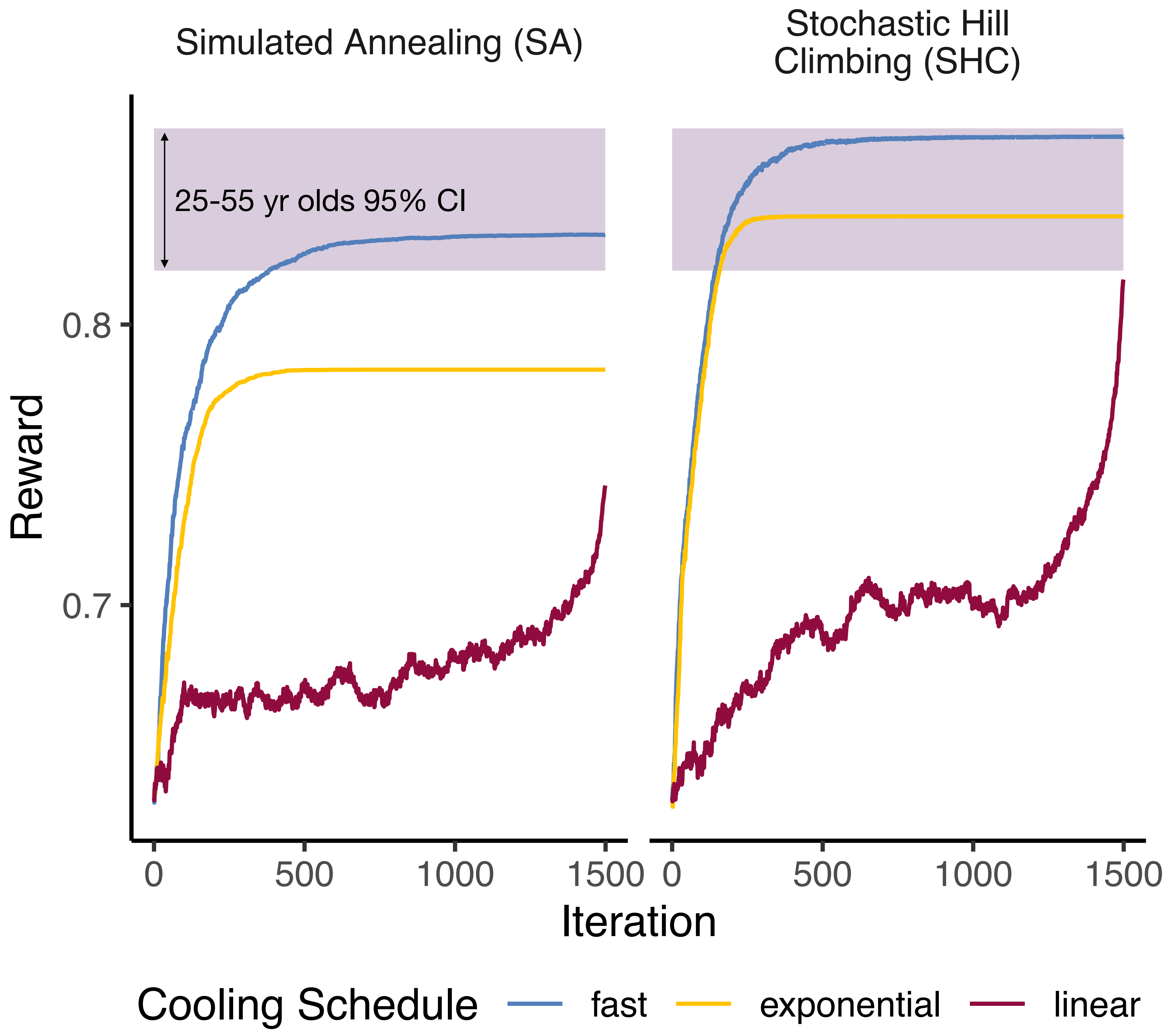
Bayesian Model Selection†



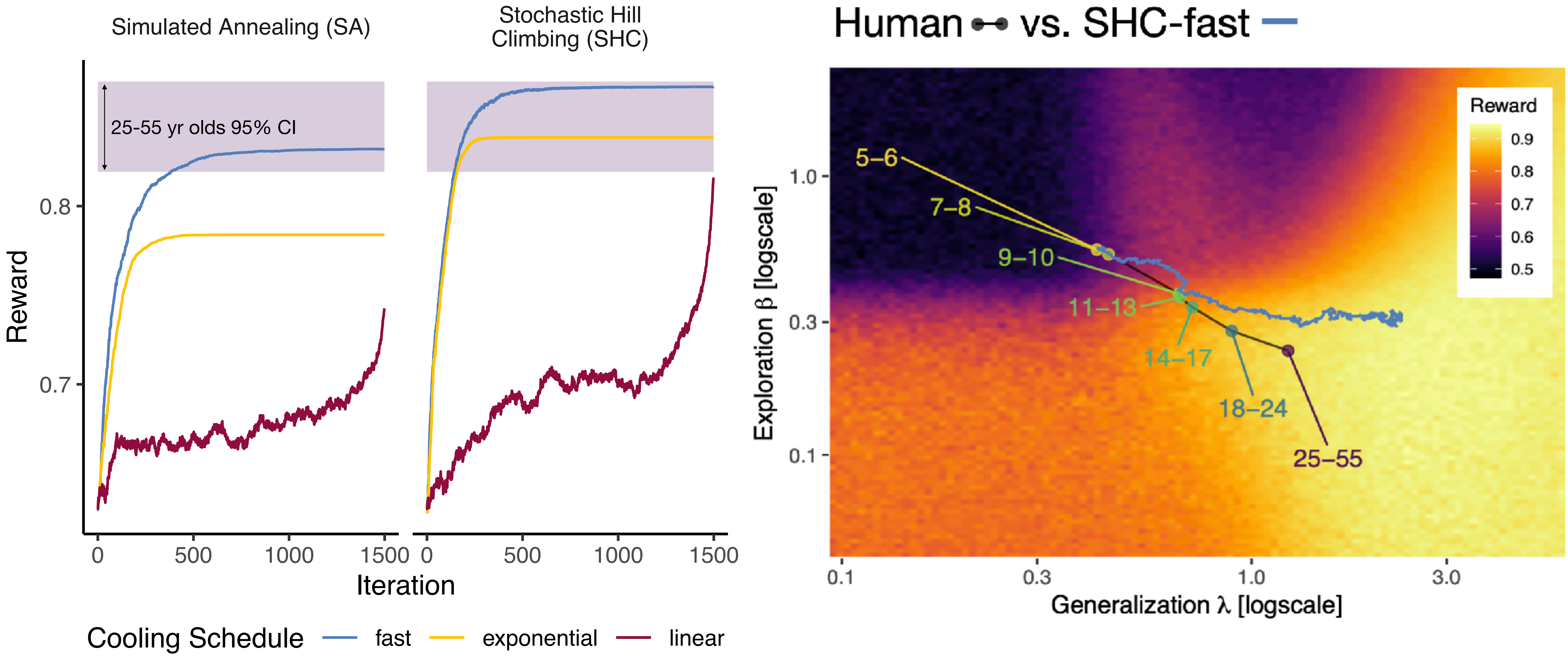
Learning curves



Human development resembles an optimization process in GP parameter space



Human development resembles an optimization process in GP parameter space



General principles of human exploration

- **Generalization**
  - assume similar stimuli will yield similar rewards
    - RBF kernel is analogous to Shepard's Law of Generalization
  - use Bayesian function learning to learn a value function
    - Distribution over functions is analogous to Bayesian Concept Learning
  - extrapolation/interpolation to make predictions about novel stimuli
- **Uncertainty-directed exploration**
  - rather than only random exploration, people direct their exploration towards regions of the search space they are most uncertain about
  - can also be viewed as an optimism bias:
    - inflating reward expectations by their uncertainty is akin to assuming the most optimistic outcome
    - this optimism pays off because, because it corresponds to also valuing *information gain*

General principles of human exploration

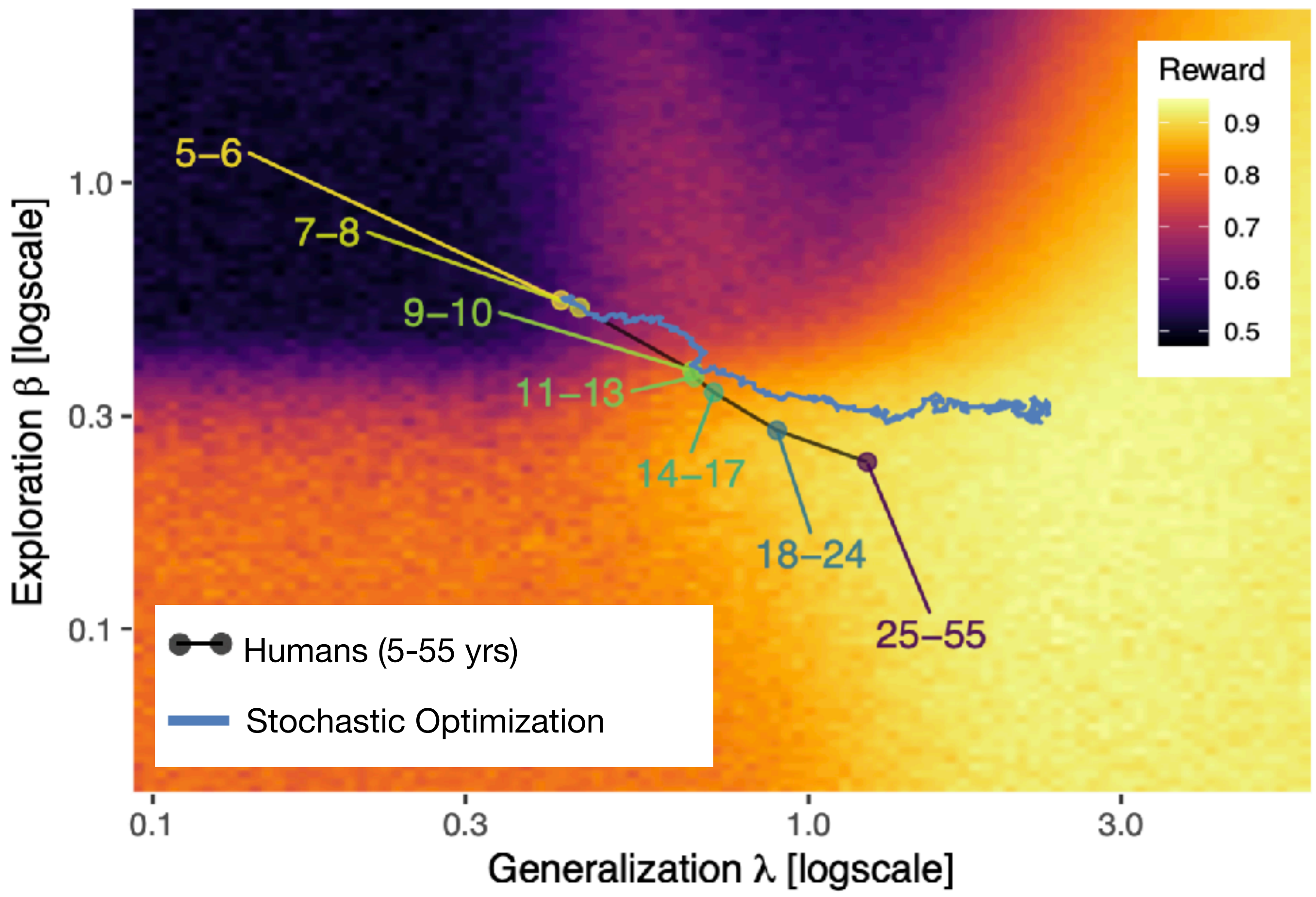
- **Generalization**
  - assume similar stimuli will yield similar rewards
    - RBF kernel is analogous to Shepard's Law of Generalization
  - use Bayesian function learning to learn a value function
    - Distribution over functions is analogous to Bayesian Concept Learning
  - extrapolation/interpolation to make predictions about novel stimuli
- **Uncertainty-directed exploration**
  - rather than only random exploration, people direct their exploration towards regions of the search space they are most uncertain about
  - can also be viewed as an optimism bias:
    - inflating reward expectations by their uncertainty is akin to assuming the most optimistic outcome
    - this optimism pays off because, because it corresponds to also valuing *information gain*

https://charleywu.github.io/downloads/driveforknowledge.pdf
pwd is "knowledge"

GP-UCB across domains

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

2. Developmental trajectory of learning
   Schulz, Wu, Ruggeri & Meder (*PsychSci* 2019)
   Meder, Wu, Schulz & Ruggeri (*DevSci 2021*)
   Giron*, Ciranka*, Schulz, van den Bos, Ruggeri, Meder & Wu (*NHB* in press)



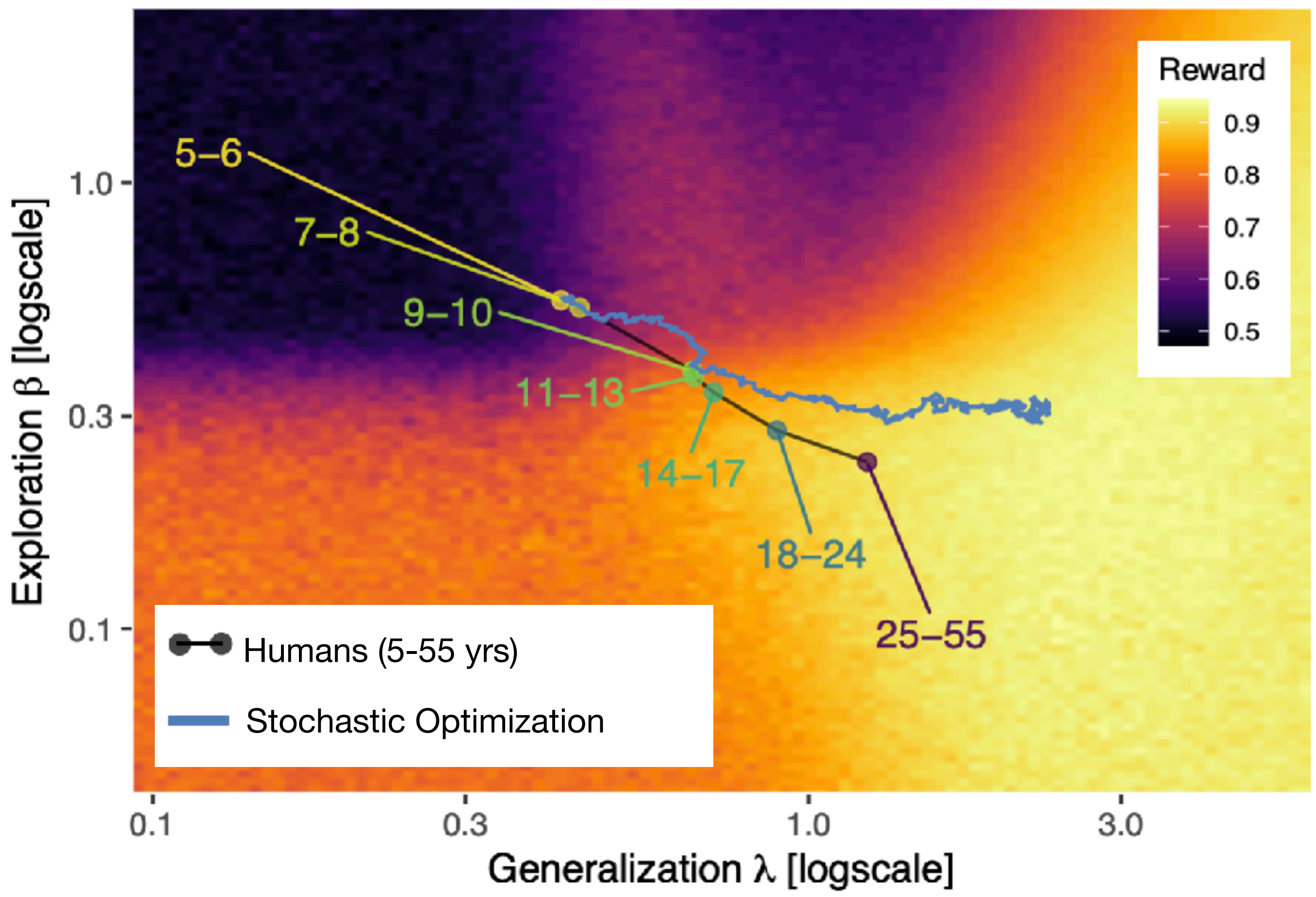
Giron*, Ciranka*,… & Wu (*NHB* in press)

GP-UCB across domains

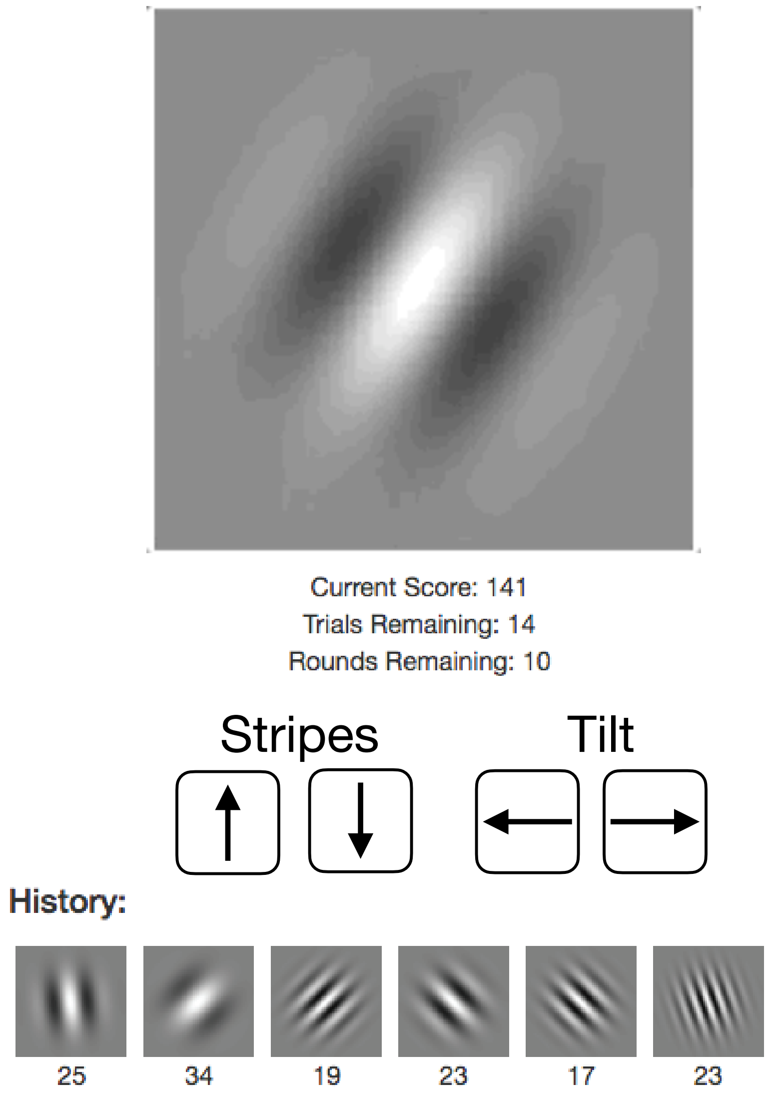
1. Generalization guides exploration
   Wu, Schulz, Nelson, Speekenbrink & Meder (*Nature Human Behaviour* 2018)

2. Developmental trajectory of learning
   Schulz, Wu, Ruggeri & Meder (*PsychSci* 2019)
   Meder, Wu, Schulz & Ruggeri (*DevSci 2021*)
   Giron*, Ciranka*, Schulz, van den Bos, Ruggeri, Meder & Wu (*NHB* in press)

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



Giron*, Ciranka*,… & Wu (*NHB* in press)



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

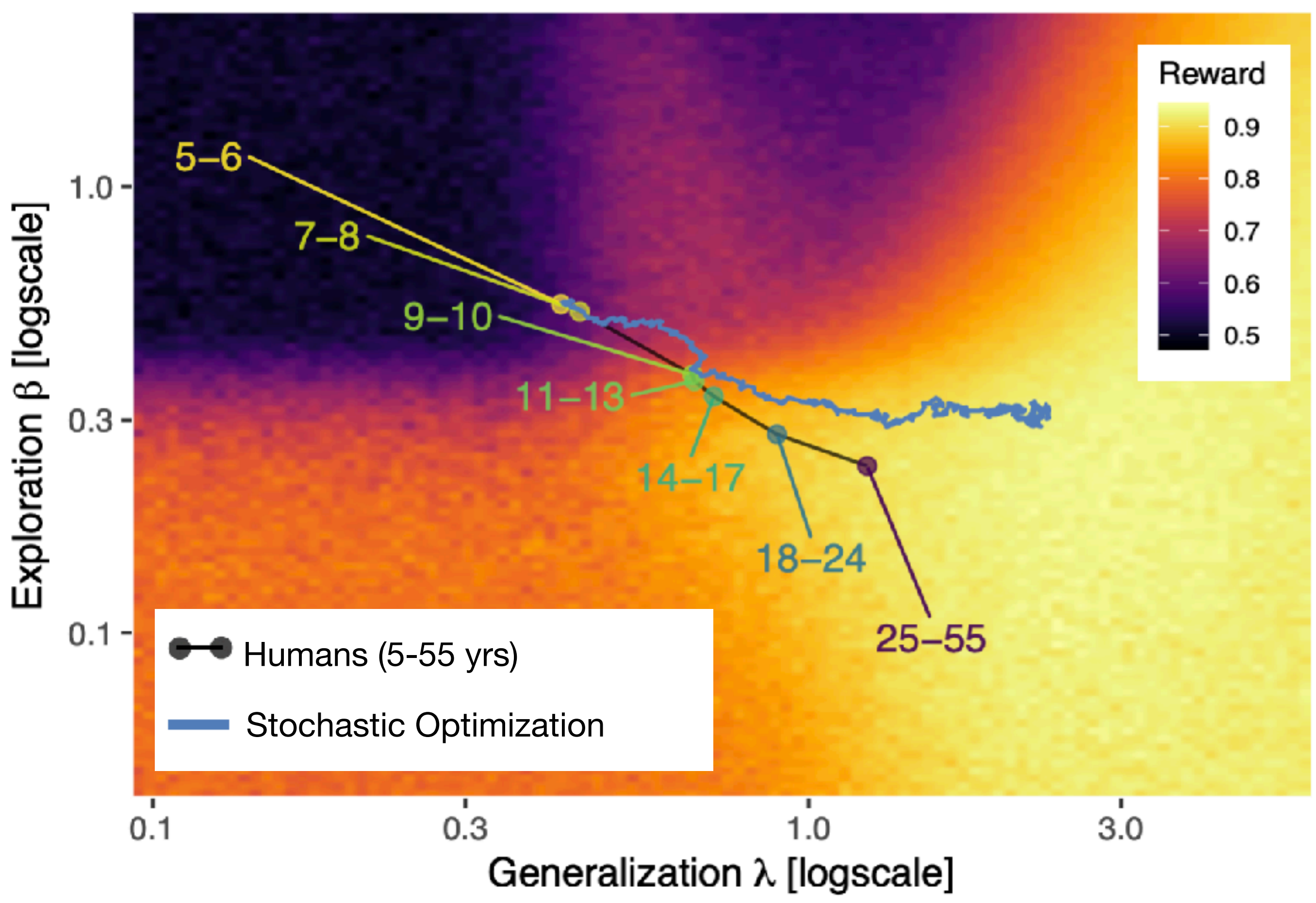
GP-UCB across domains

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

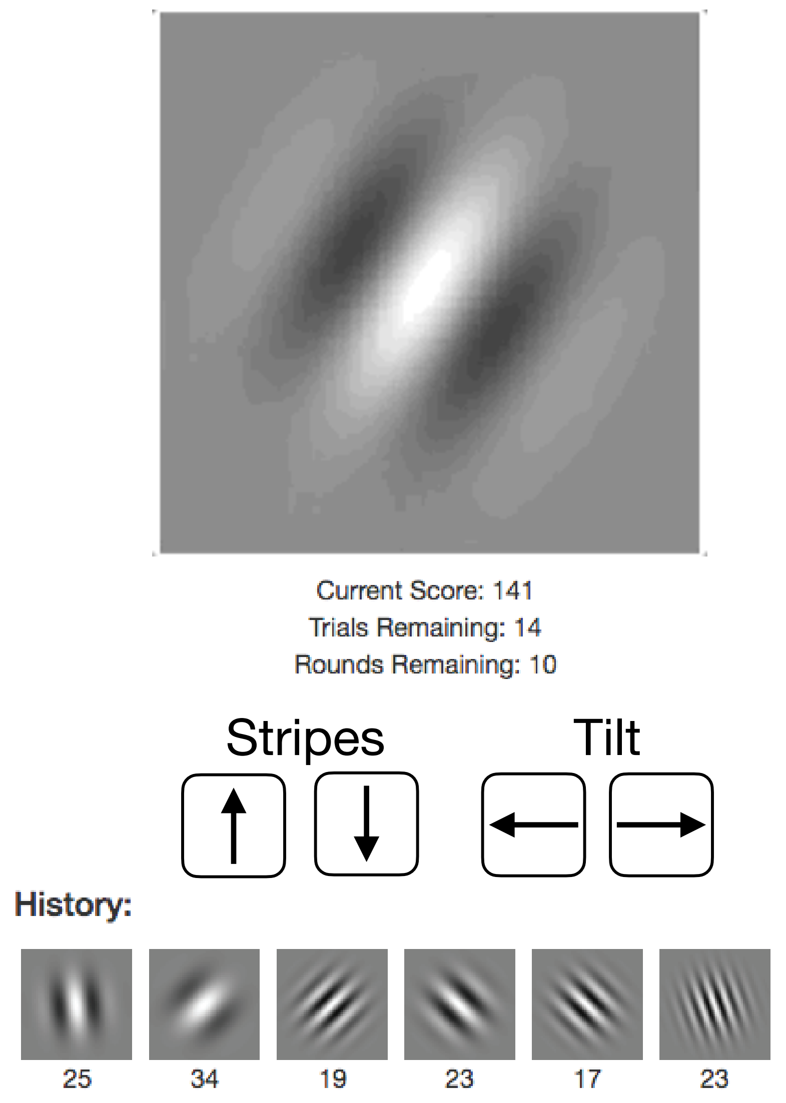
2. Developmental trajectory of learning
   Schulz, Wu, Ruggeri & Meder (*PsychSci* 2019)
   Meder, Wu, Schulz & Ruggeri (*DevSci 2021*)
   Giron*, Ciranka*, Schulz, van den Bos, Ruggeri, Meder & Wu (*NHB* in press)

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

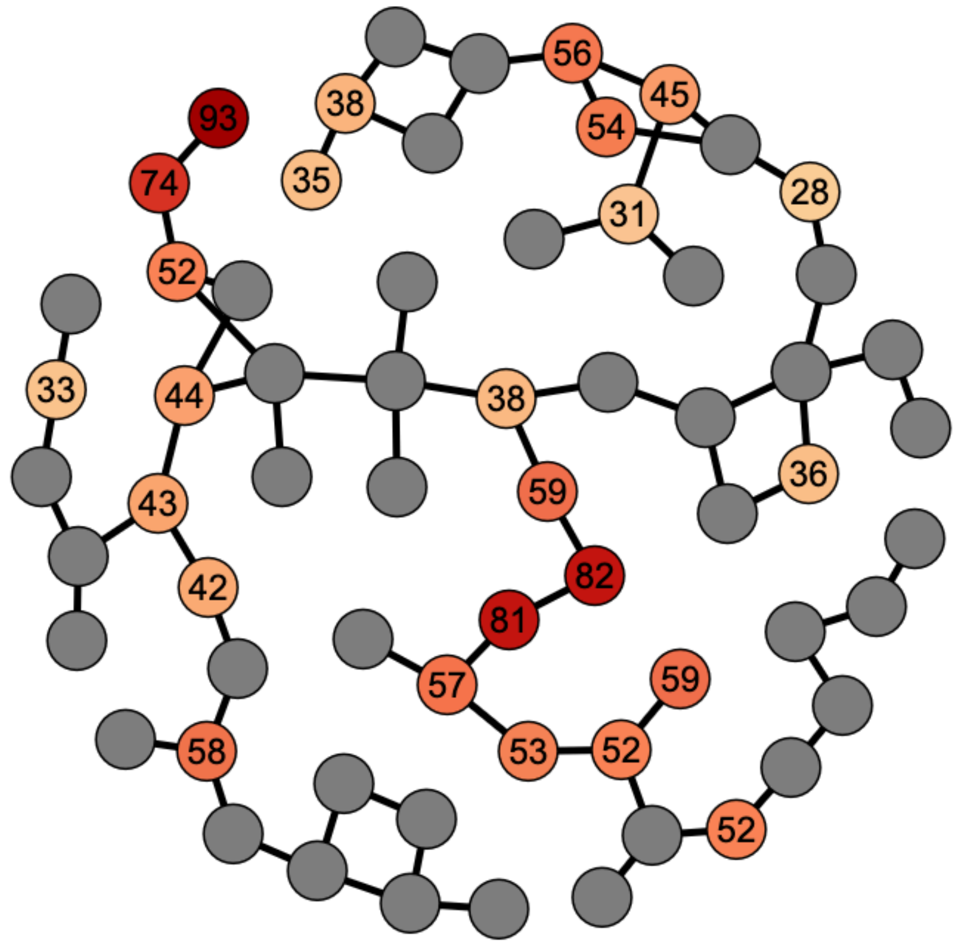
4. Graph-structured generalization
   Wu, Schulz & Gershman (*Comput Brain Behav* 2021)



Giron*, Ciranka*,… & Wu (*NHB* in press)



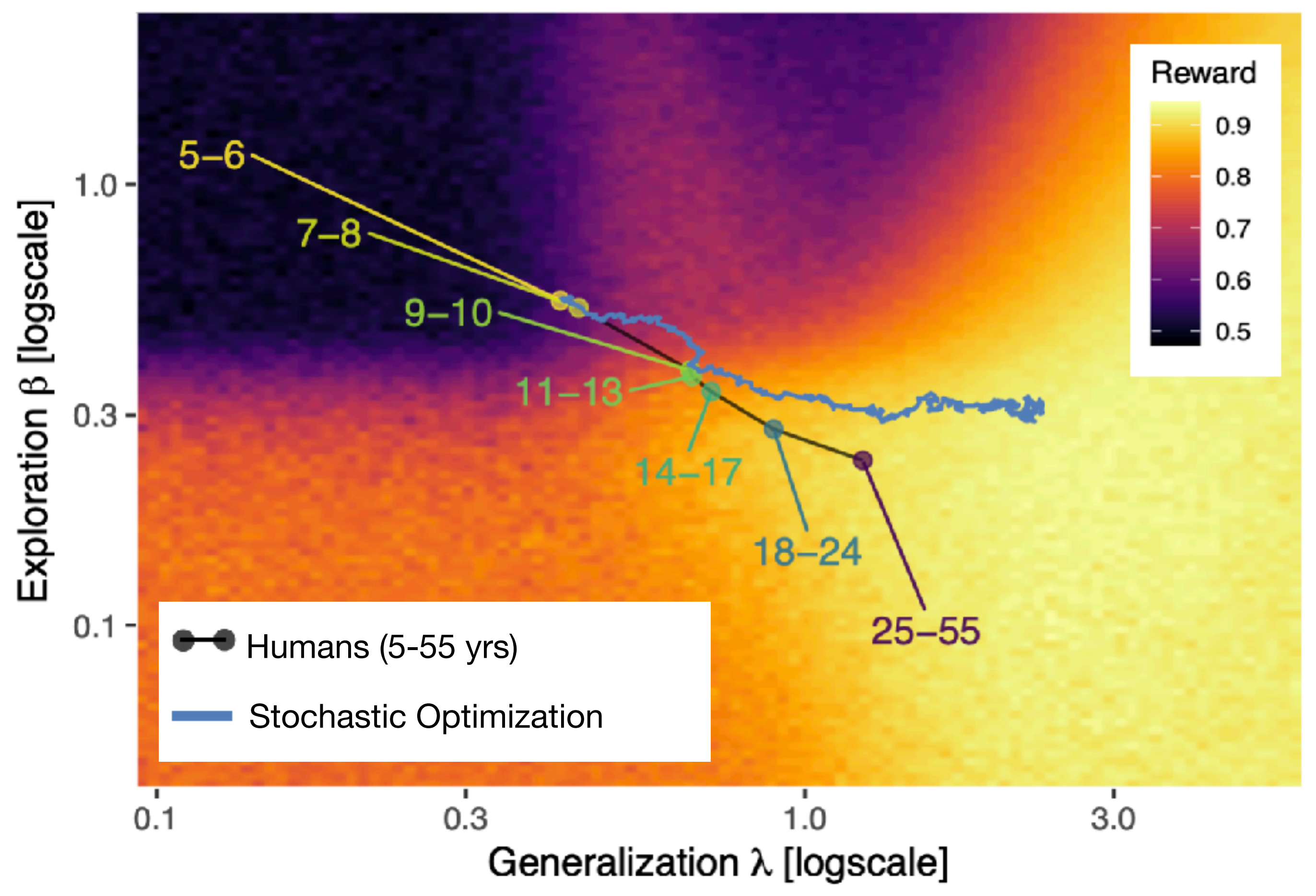
Wu, Schulz, Garvert, Meder & Schuck (*PLOS Comp Bio 2020*)



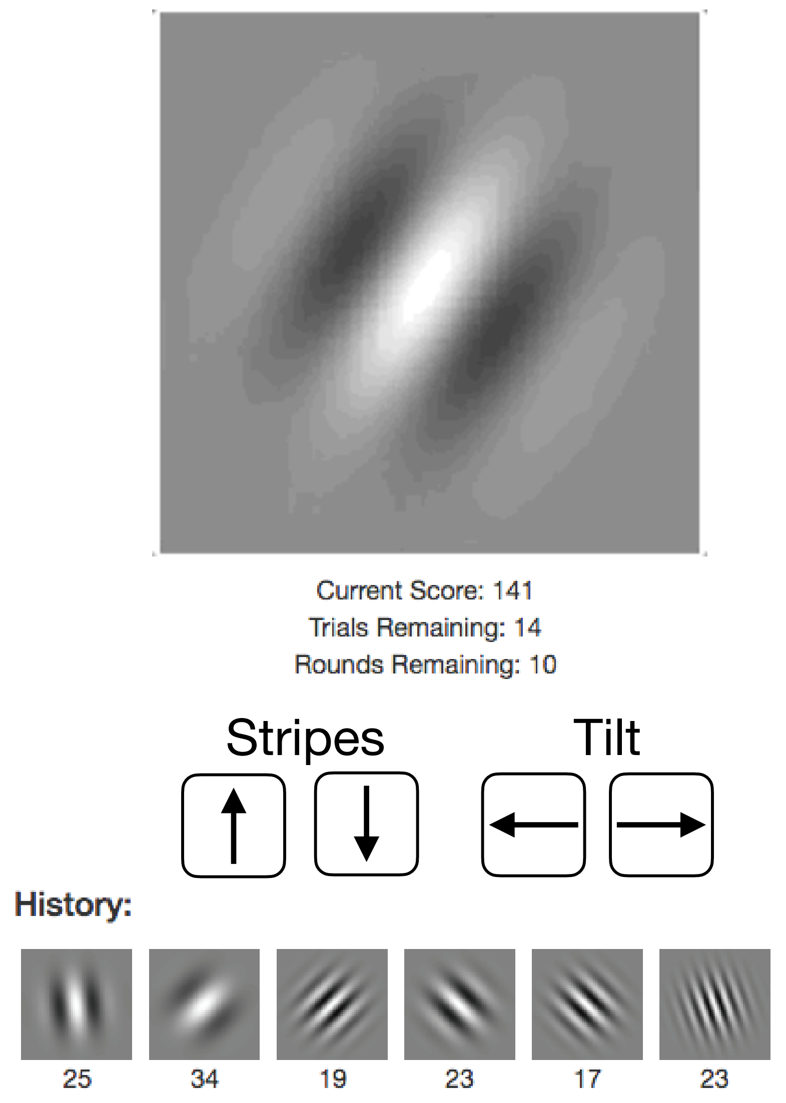
Wu, Schulz & Gershman (*CBB* 2021)

GP-UCB across domains

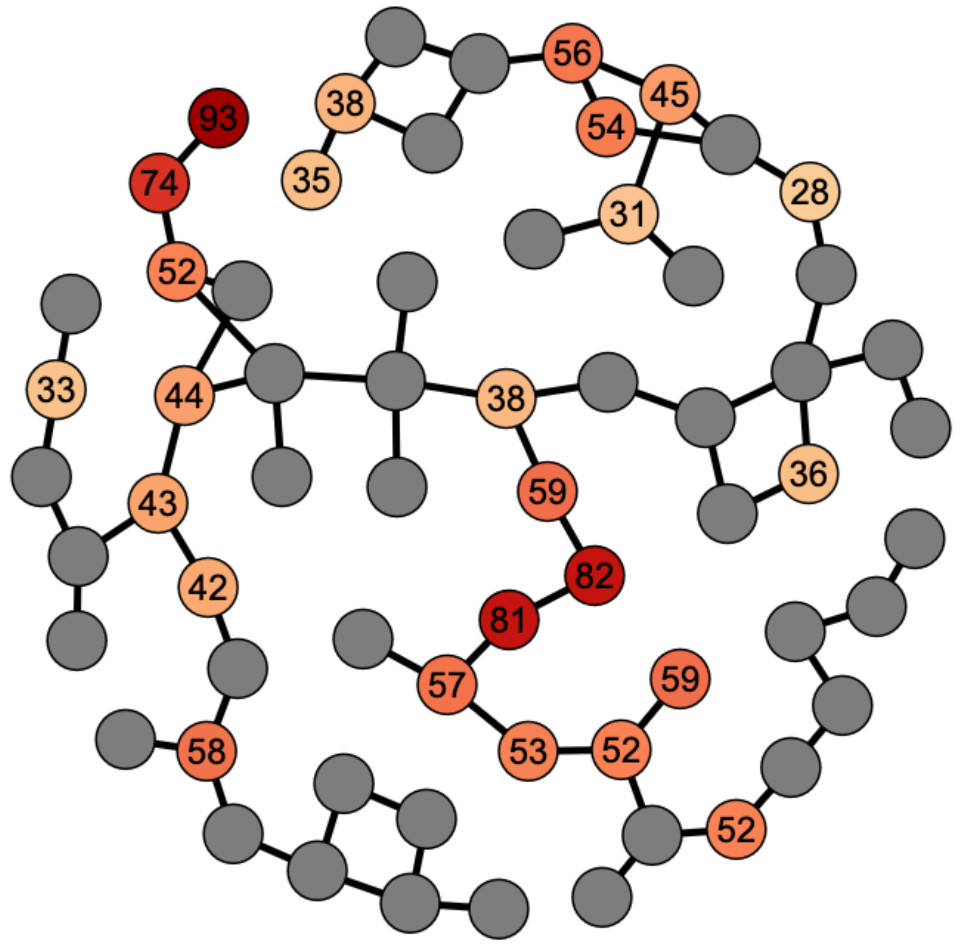
1. Generalization guides exploration
   Wu, Schulz, Nelson, Speekenbrink & Meder (*Nature Human Behaviour* 2018)
2. Developmental trajectory of learning
   Schulz, Wu, Ruggeri & Meder (*PsychSci* 2019)
   Meder, Wu, Schulz & Ruggeri (*DevSci 2021*)
   Giron*, Ciranka*, Schulz, van den Bos, Ruggeri, Meder & Wu (*NHB* in press)
3. Search in abstract conceptual spaces
   Wu, Schulz, Garvert, Meder & Schuck (*PLOS Comp Bio 2020*)
4. Graph-structured generalization
   Wu, Schulz & Gershman (*Comput Brain Behav* 2021)
5. Safe exploration in risky environments
   Schulz, Wu, Huys, Krause & Speekenbrink (*Cognitive Science* 2018)



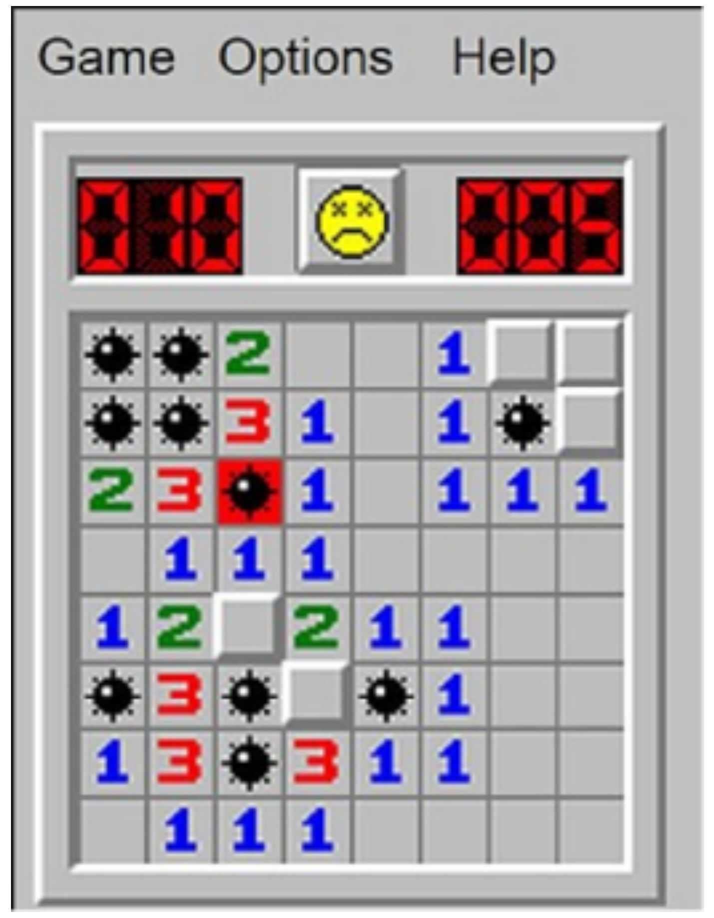
Giron*, Ciranka*,… & Wu (*NHB* in press)



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



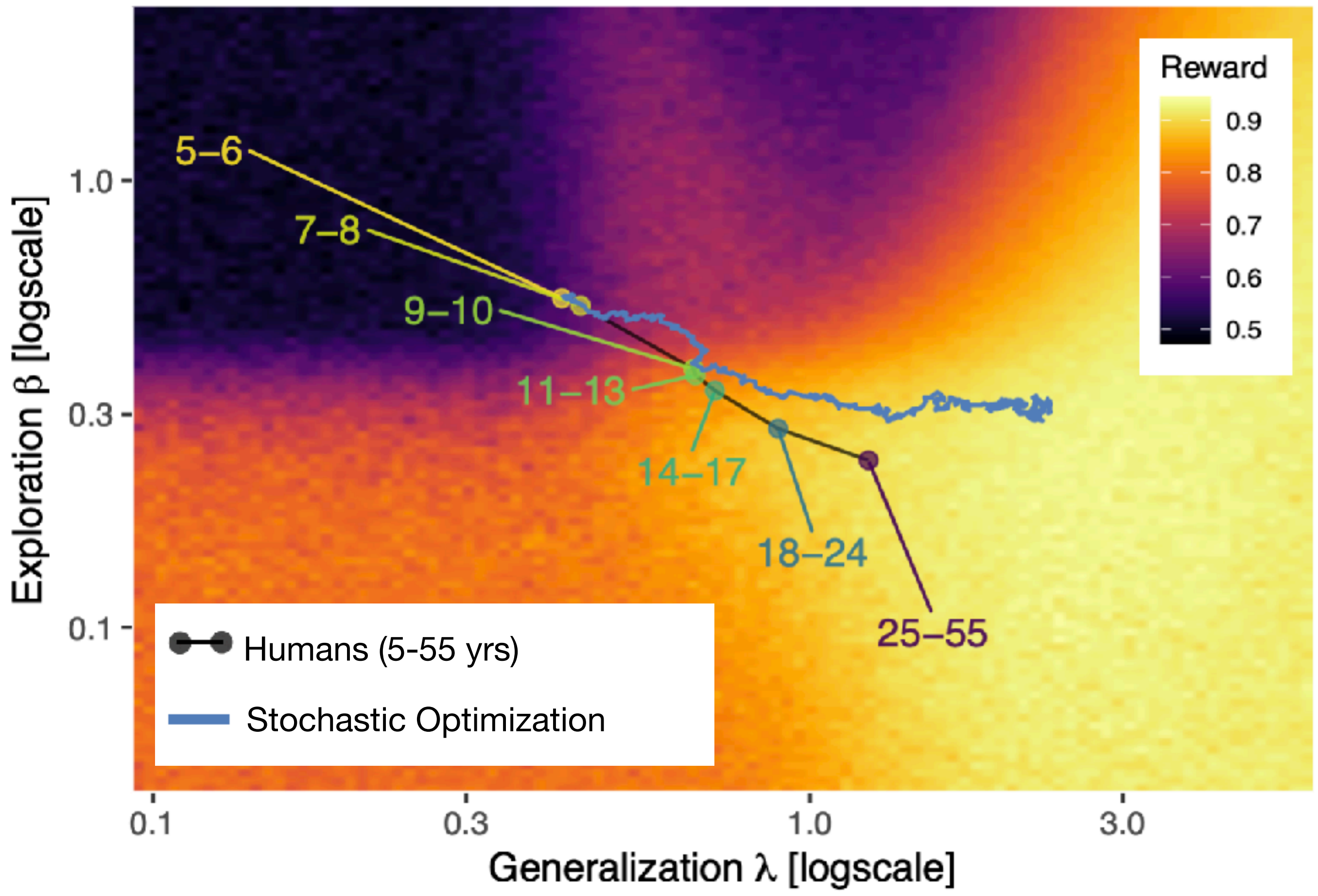
Wu, Schulz & Gershman (*CBB* 2021)



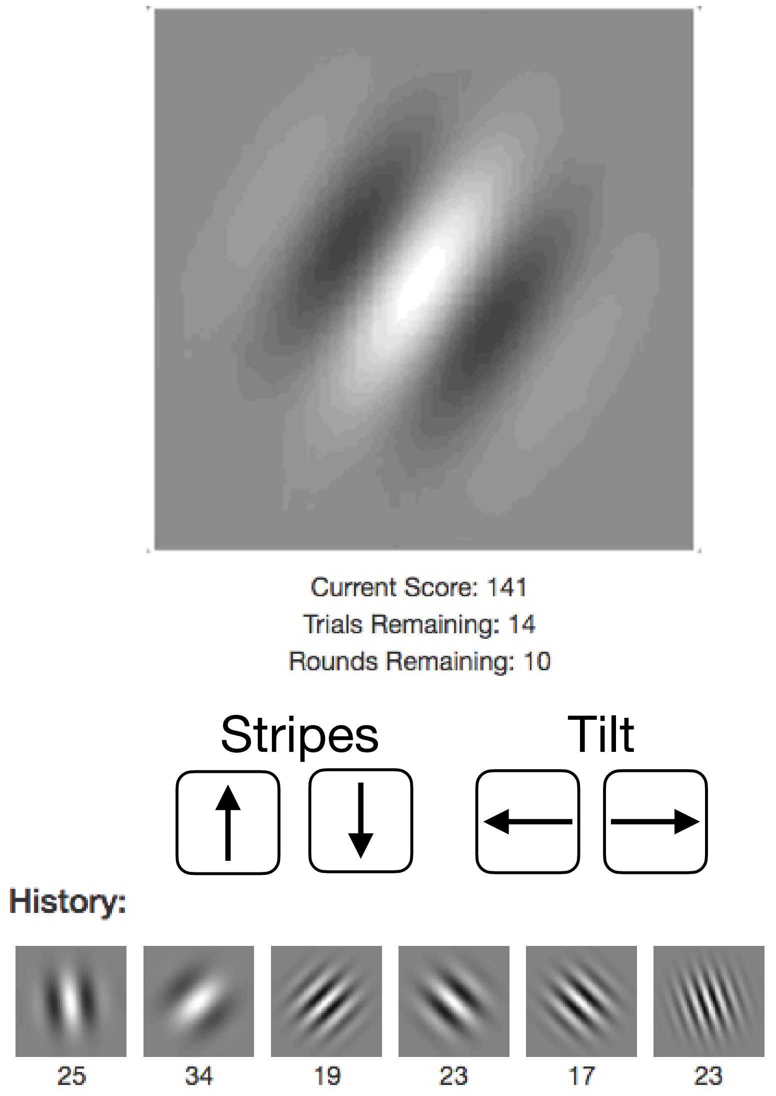
Schulz, Wu, Huys, Krause & Speekenbrink (*Cognitive Science* 2018)

GP-UCB across domains

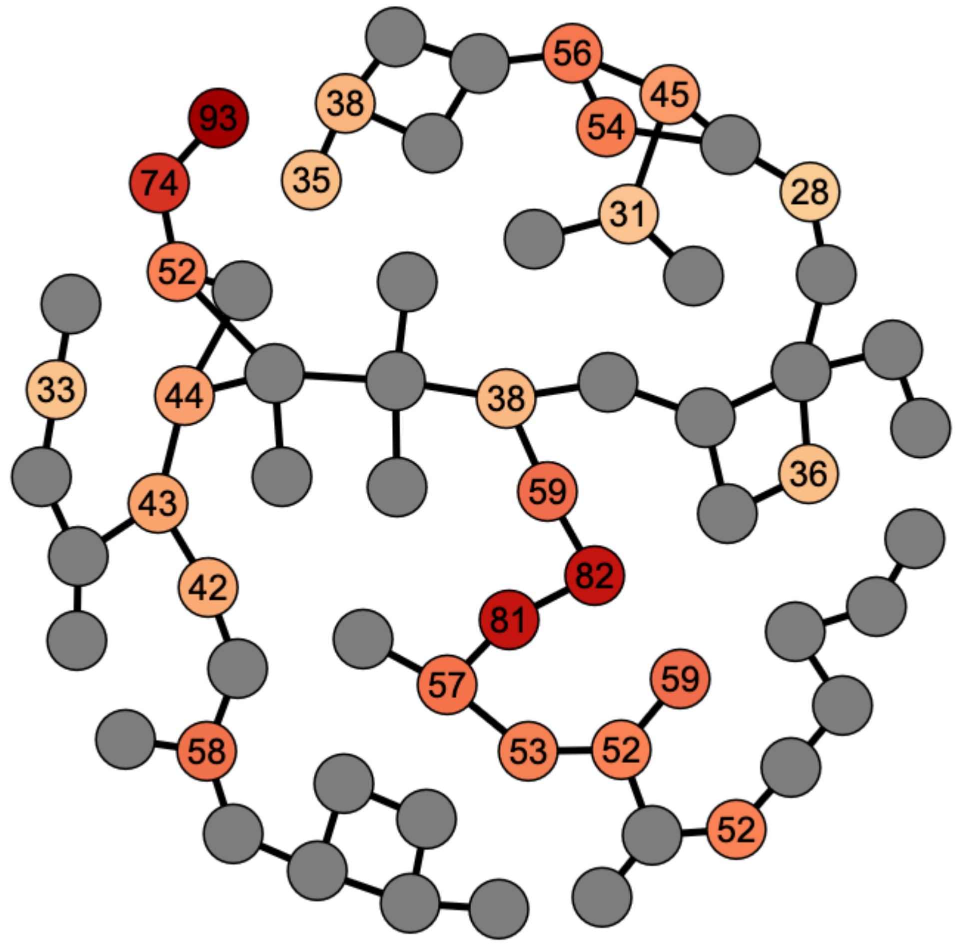
1. Generalization guides exploration
   Wu, Schulz, Nelson, Speekenbrink & Meder (*Nature Human Behaviour* 2018)
2. Developmental trajectory of learning
   Schulz, Wu, Ruggeri & Meder (*PsychSci* 2019)
   Meder, Wu, Schulz & Ruggeri (*DevSci 2021*)
   Giron*, Ciranka*, Schulz, van den Bos, Ruggeri, Meder & Wu (*NHB* in press)
3. Search in abstract conceptual spaces
   Wu, Schulz, Garvert, Meder & Schuck (*PLOS Comp Bio 2020*)
4. Graph-structured generalization
   Wu, Schulz & Gershman (*Comput Brain Behav* 2021)
5. Safe exploration in risky environments
   Schulz, Wu, Huys, Krause & Speekenbrink (*Cognitive Science* 2018)
6. Forgetful generalization with limited memory
   Breit, Ten, Sakaki, Murayama, & Wu (*in prep*)
7. Social generalization
   Wu, Deffner, Kahl, Meder, Ho & Kurvers (*bioRxiv* 2023)
   Wu, Ho, Kahl, Leuker, Meder, & Kurvers (*CogSci* 2021)
   Witt, Toyokawa, Gaissmaier, Laland, & Wu (*Cogsci* in press)
8. Neural basis for generalization and exploration
   Liebe*, Ciranka*, Spies, Lanzenburger, & Wu (*in prep*)



Giron*, Ciranka*,… & Wu (*NHB* in press)



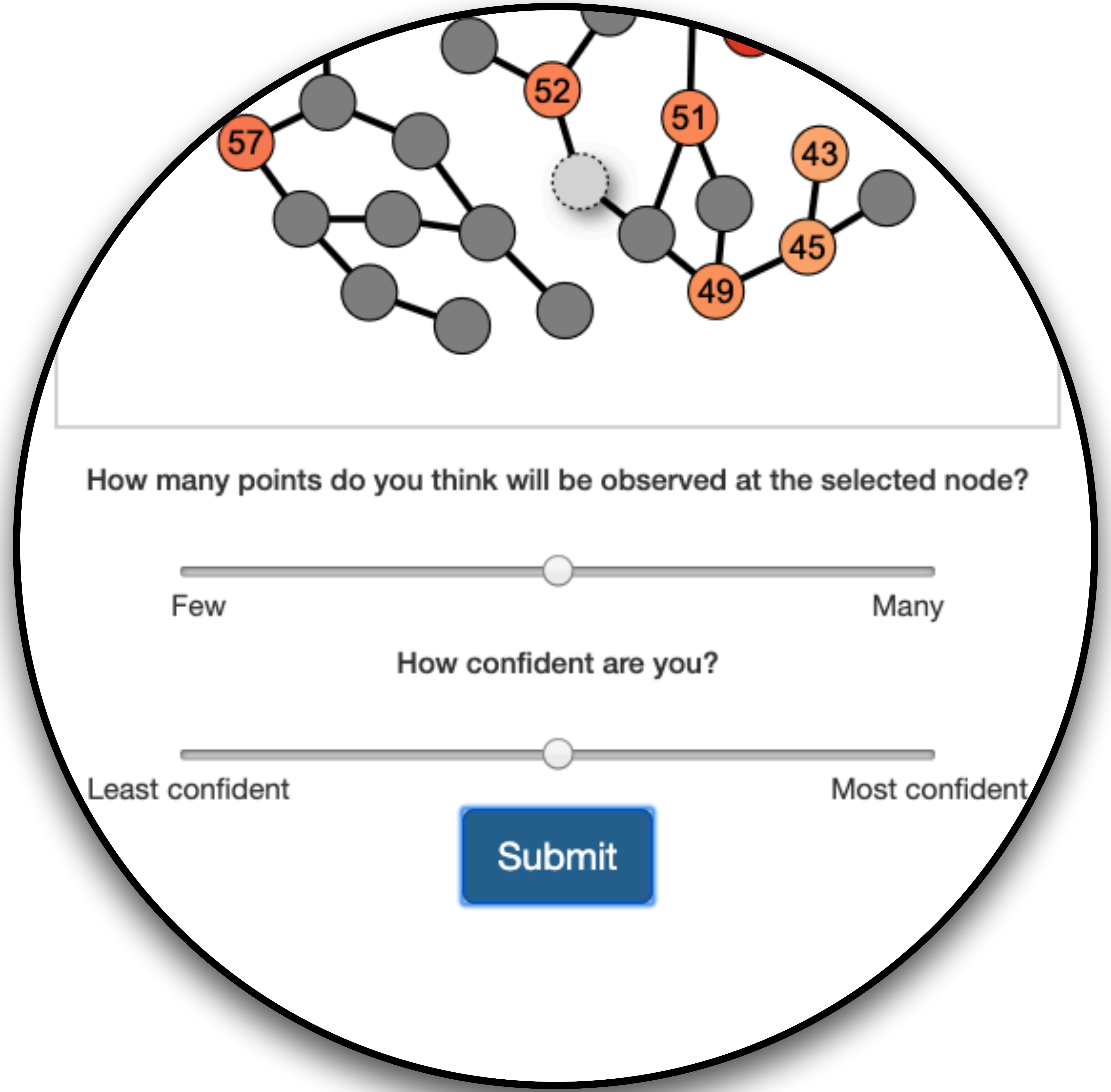
Wu, Schulz, Garvert, Meder & Schuck (*PLOS Comp Bio 2020*)



Wu, Schulz & Gershman (*CBB* 2021)

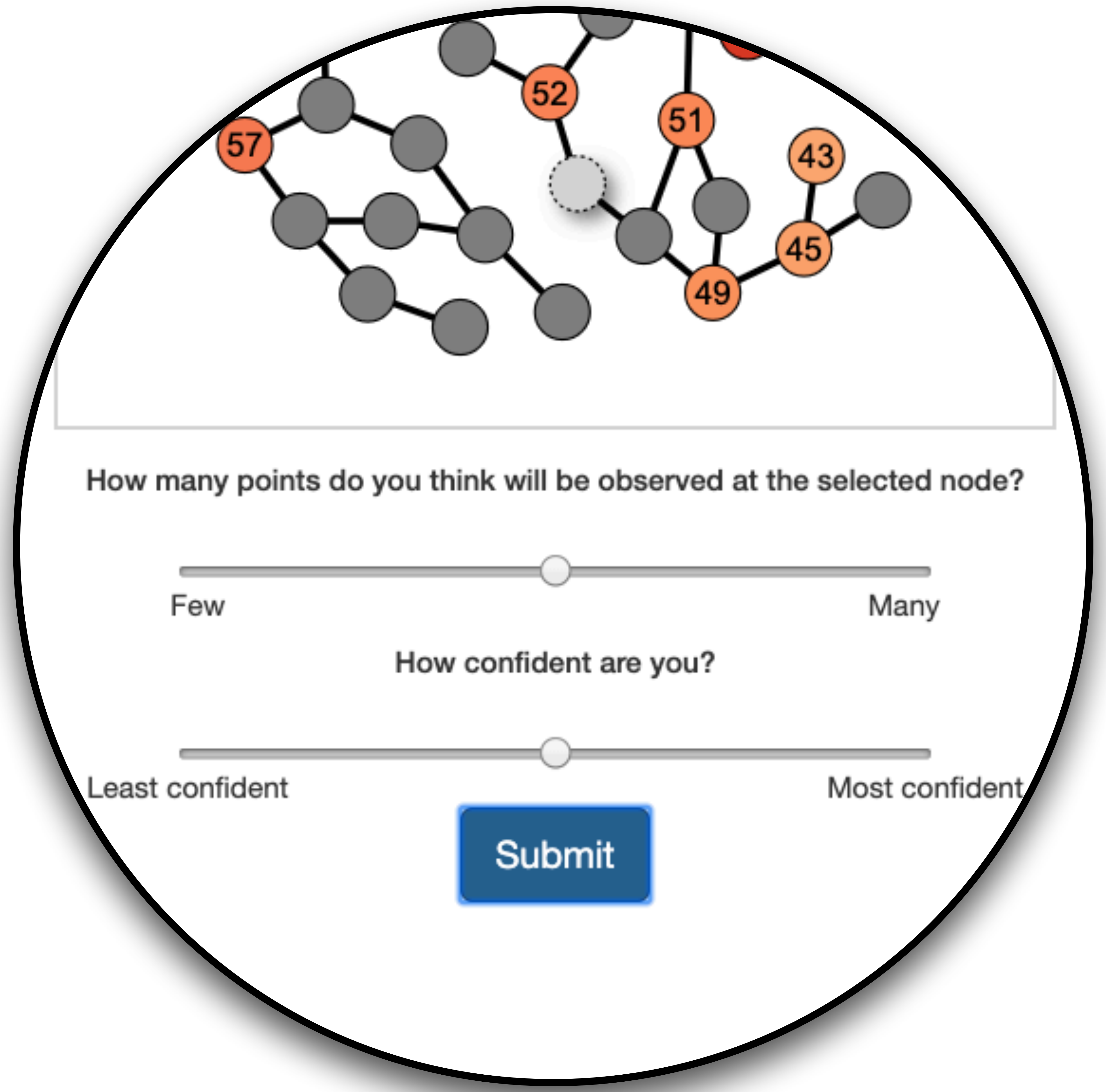
Schulz, Wu, Huys, Krause & Speekenbrink (*Cognitive Science* 2018)

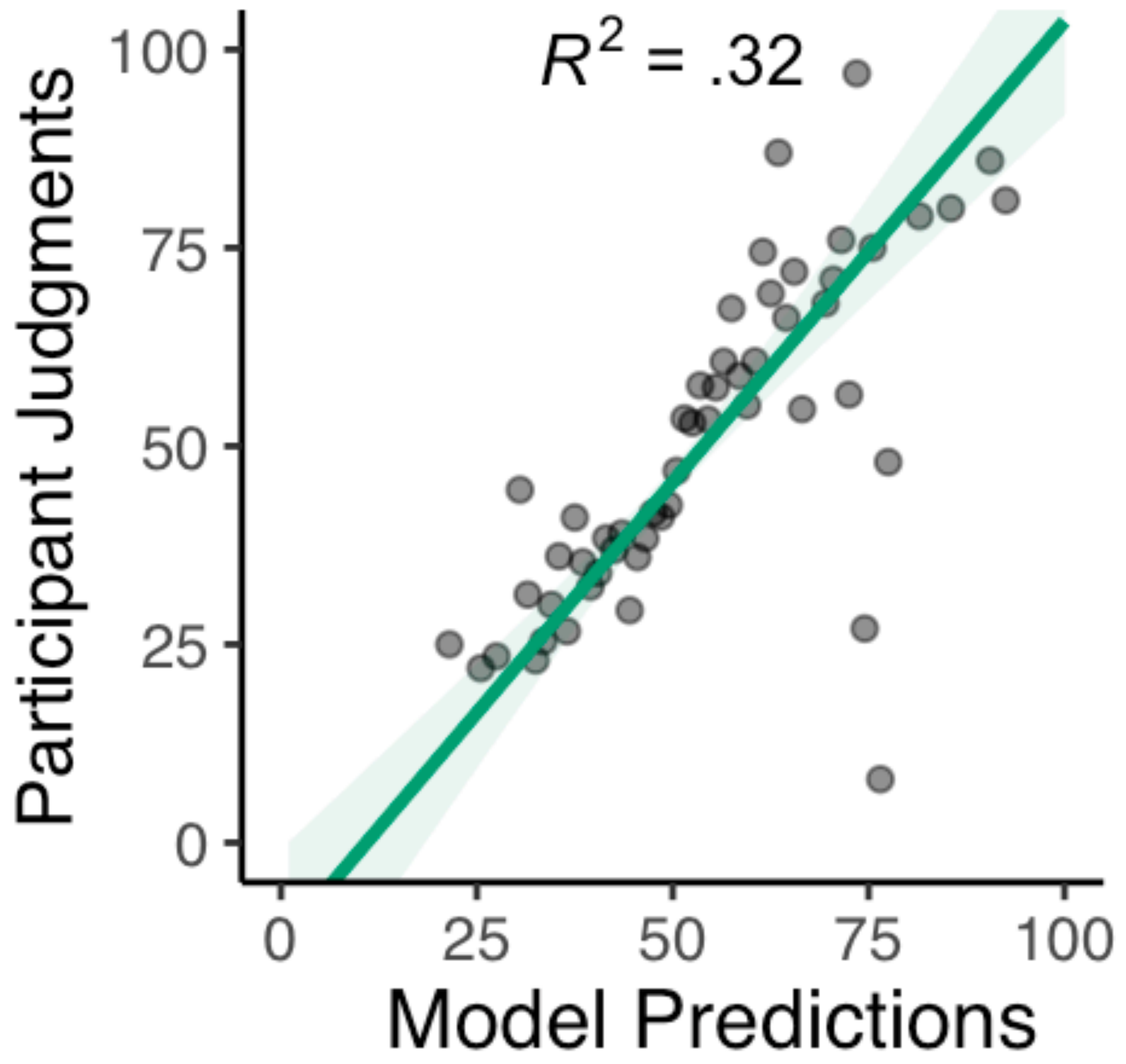
Validation on judgments



Wu, Schulz & Gershman (*CBB* 2021); see also Wu et al,. (*PlosCompBio* 2022)

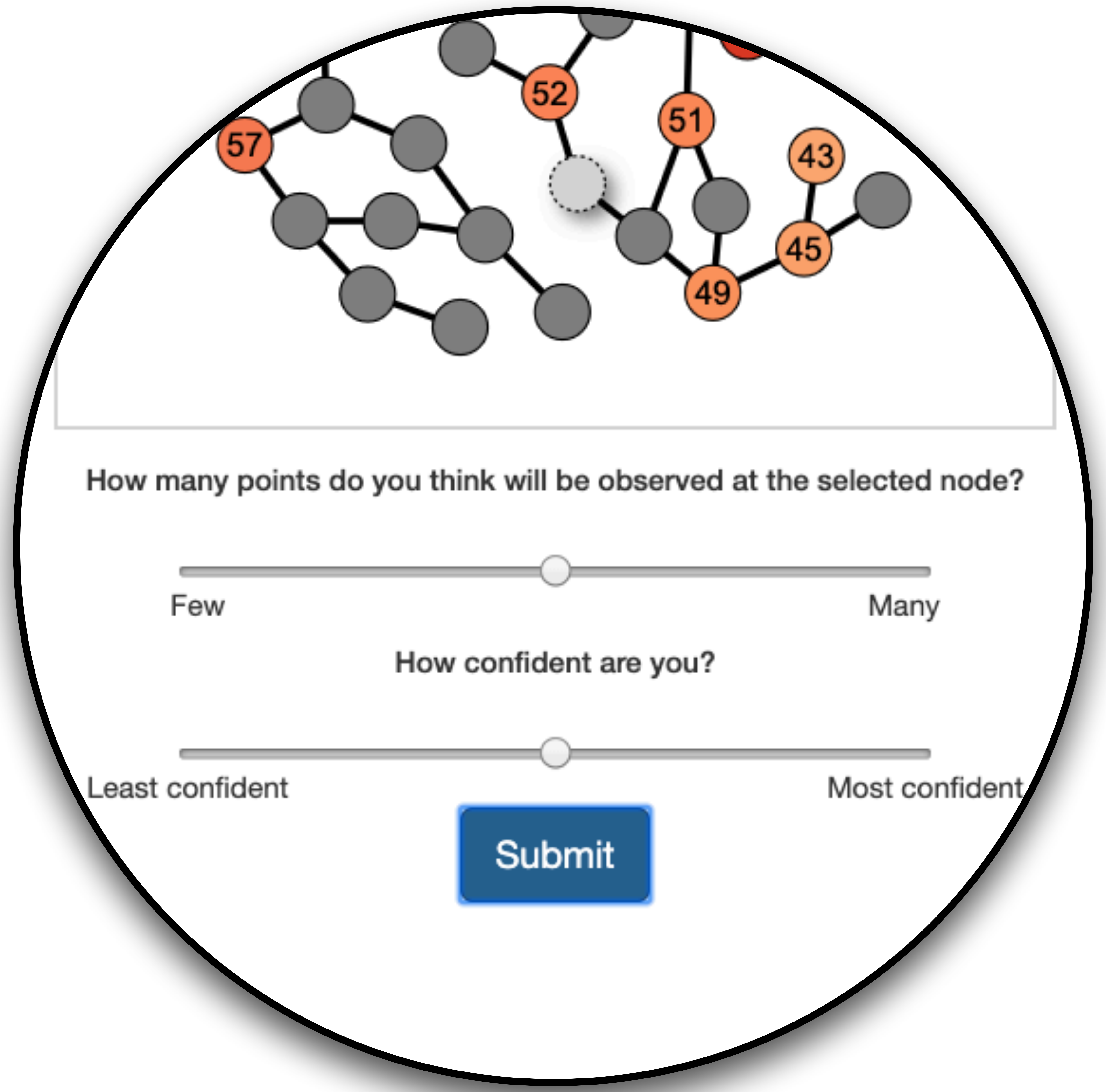
Validation on judgments

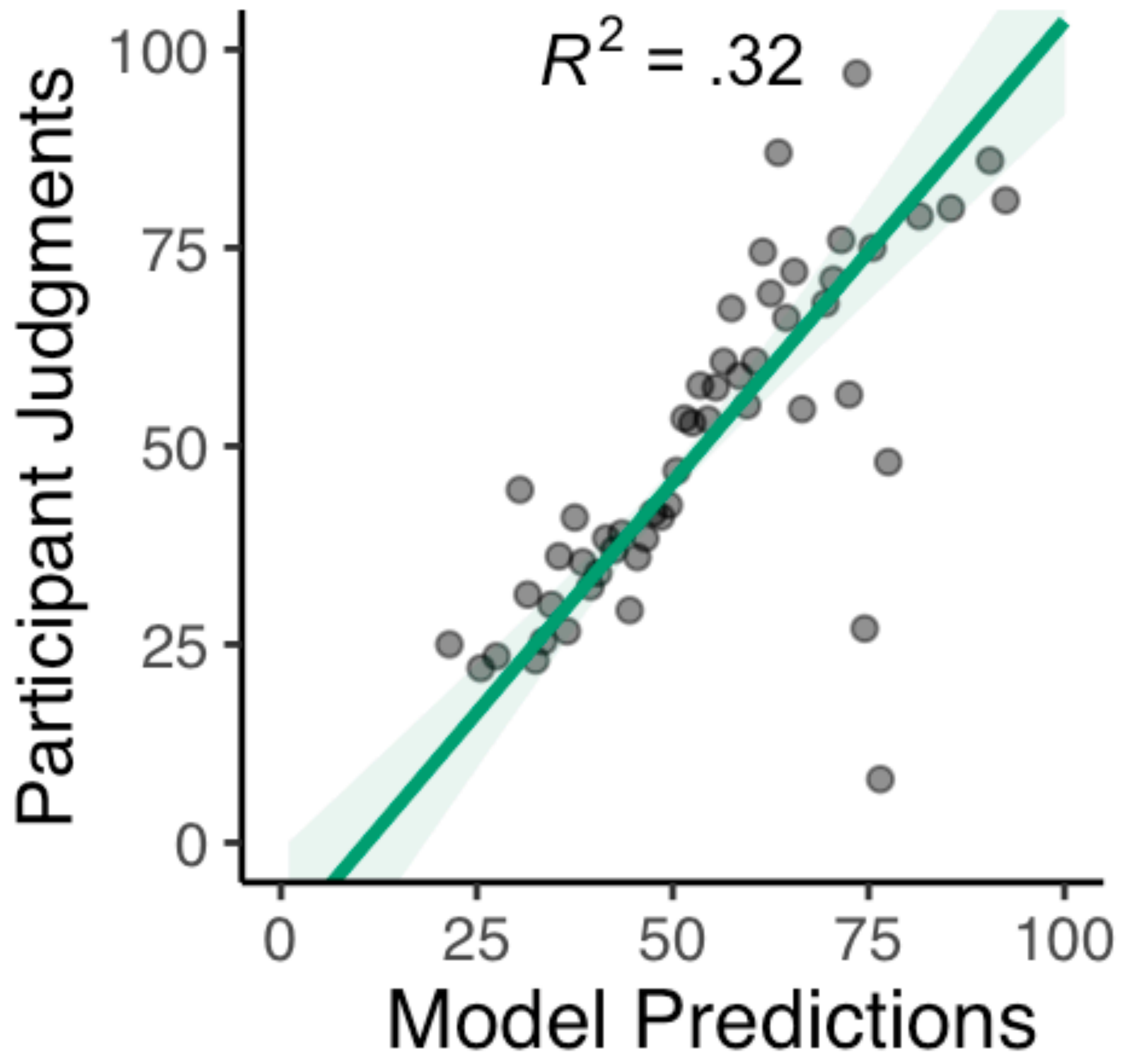


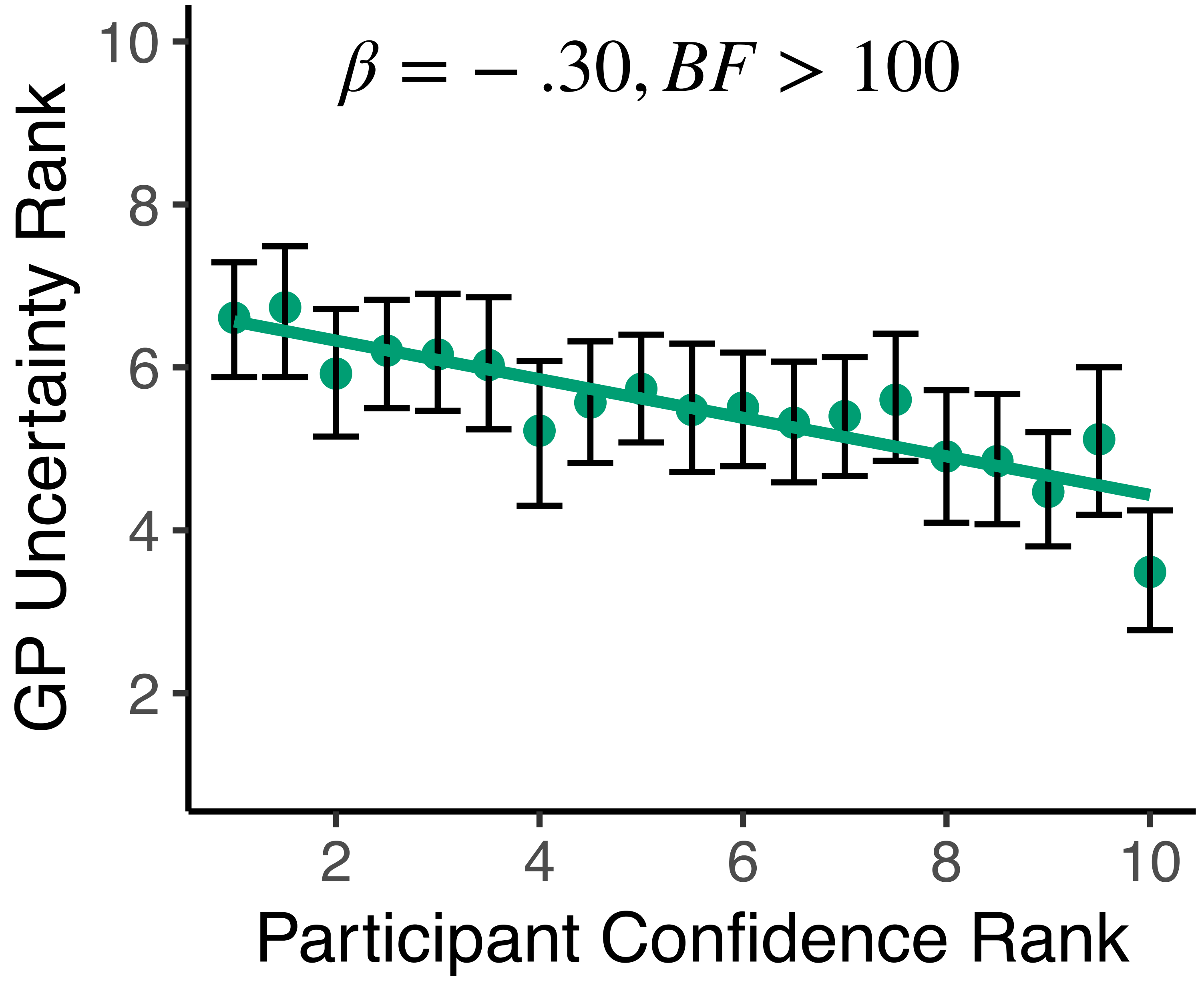


Wu, Schulz & Gershman (*CBB* 2021); see also Wu et al,. (*PlosCompBio* 2022)

Validation on judgments



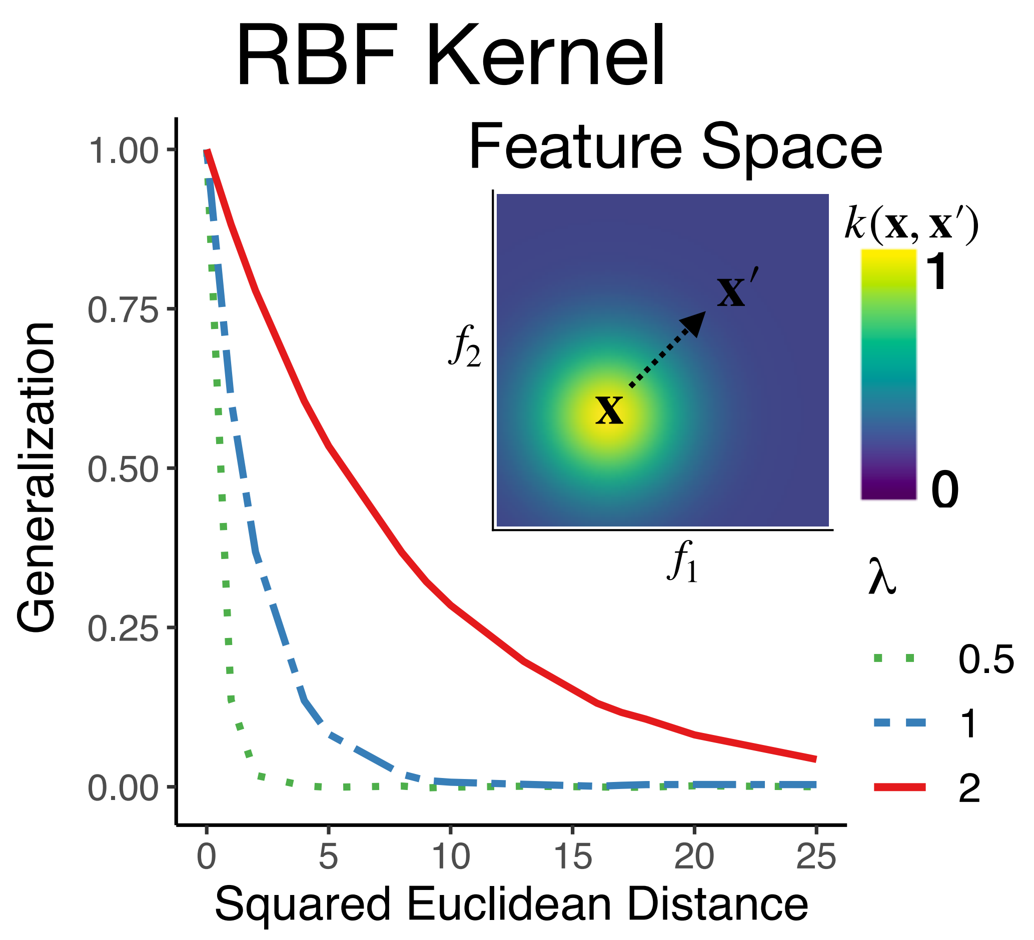




Wu, Schulz & Gershman (*CBB* 2021); see also Wu et al,. (*PlosCompBio* 2022)

Similarity can also capture relational structure

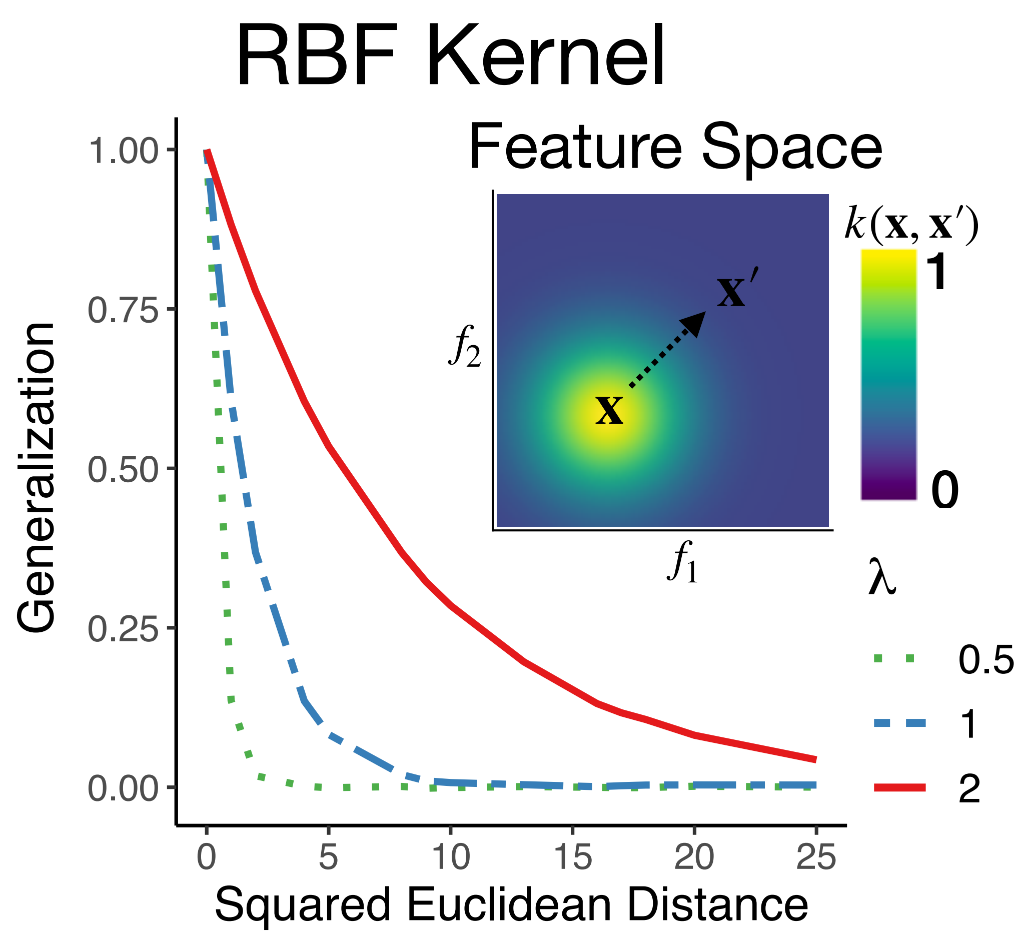
- The RBF kernel, like most classic accounts, represent similarity as distance in feature space
  - Learns smooth functions in a continuous domain

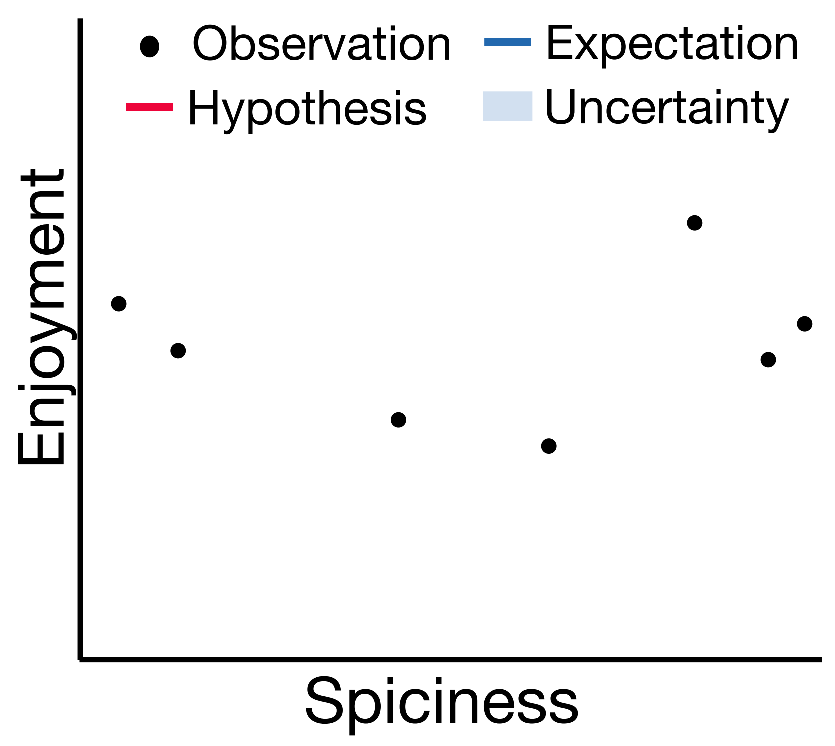


Wu et al., (*CBB* 2021)

Similarity can also capture relational structure

- The RBF kernel, like most classic accounts, represent similarity as distance in feature space
  - Learns smooth functions in a continuous domain

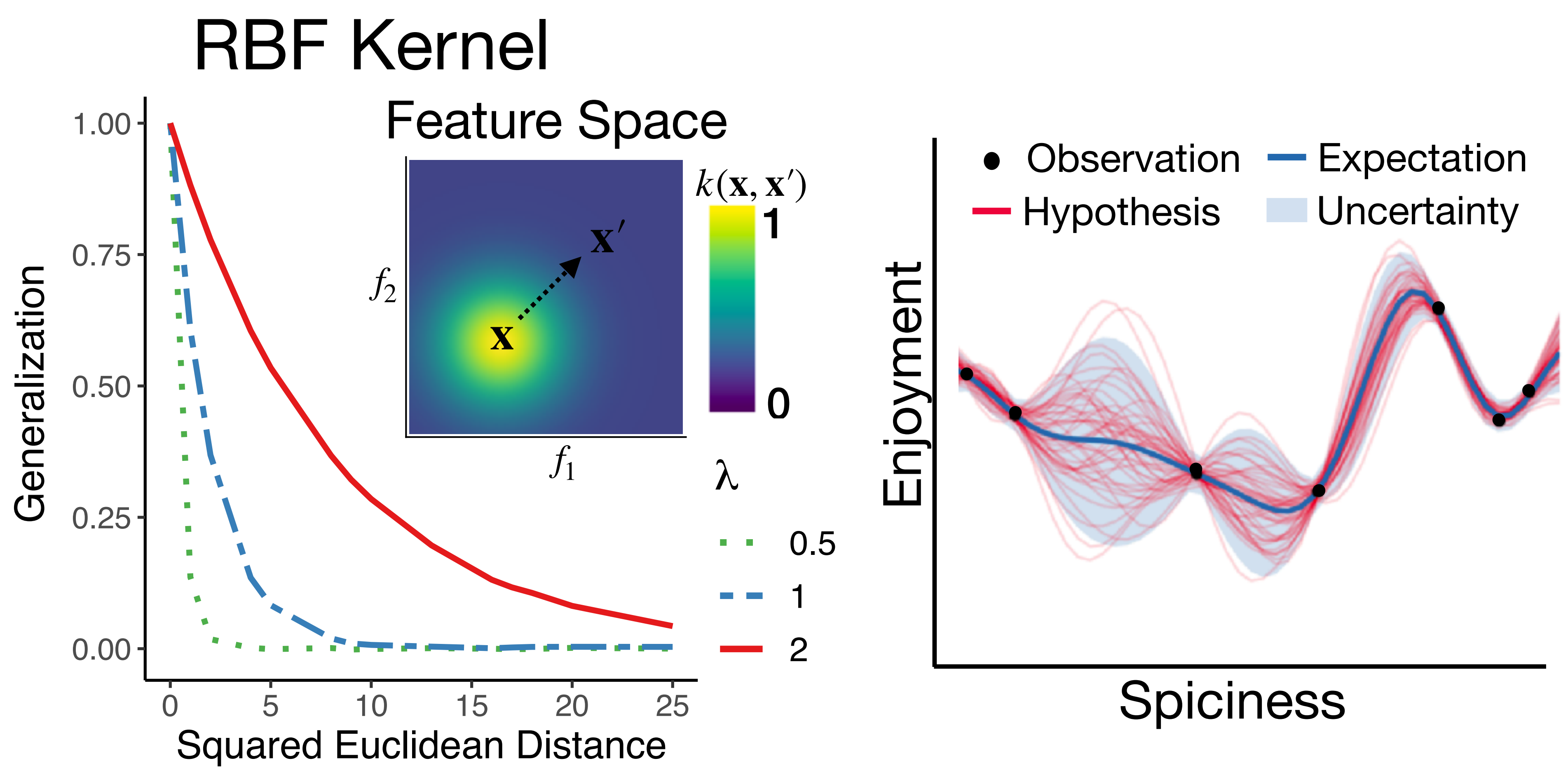




Wu et al., (*CBB* 2021)

Similarity can also capture relational structure

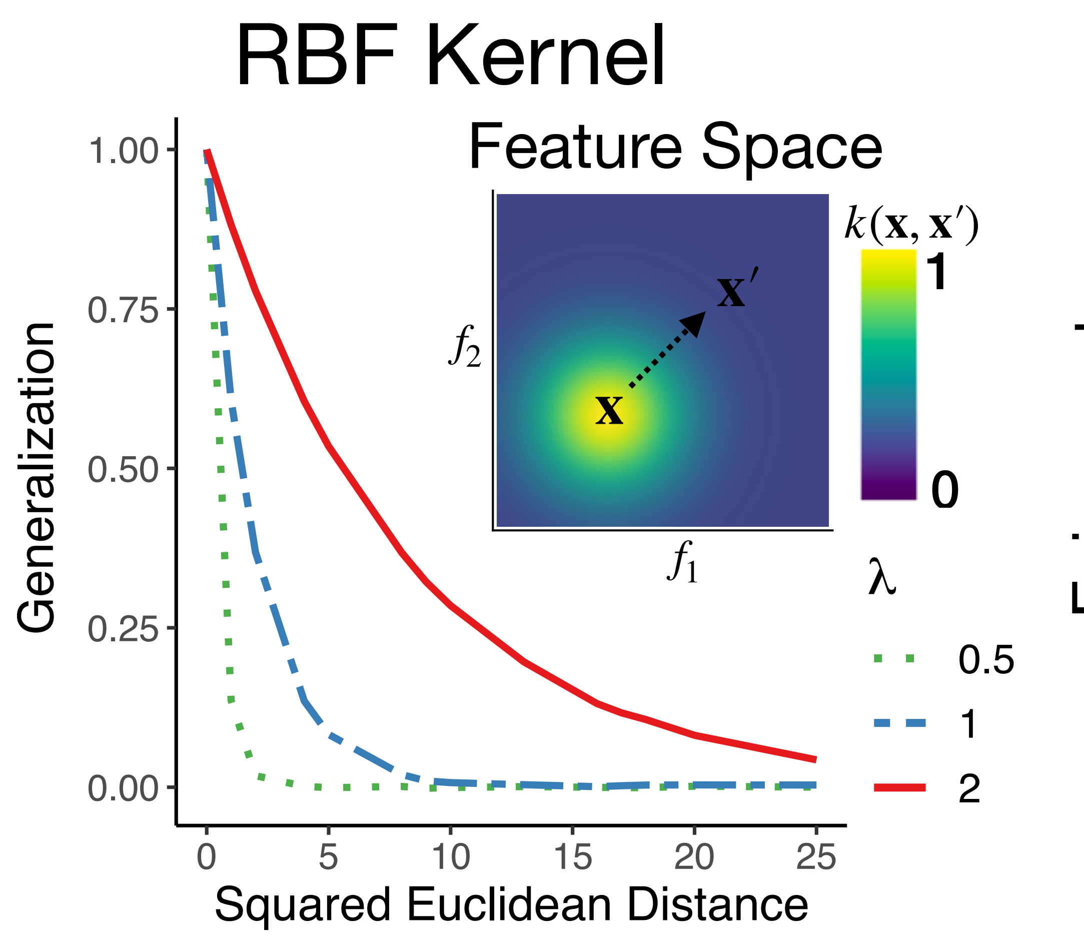
- The RBF kernel, like most classic accounts, represent similarity as distance in feature space
  - Learns smooth functions in a continuous domain

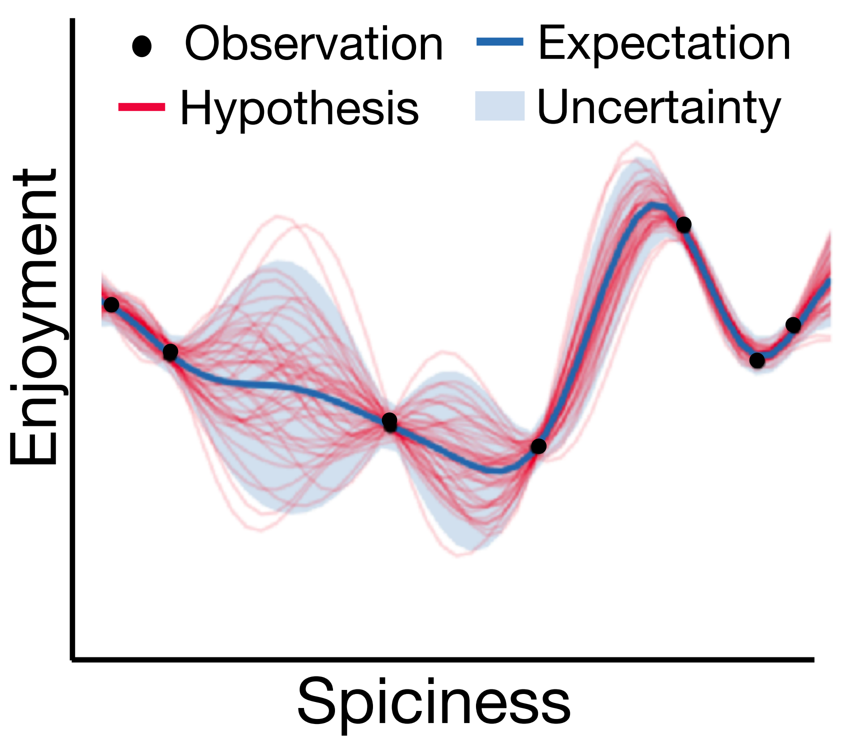


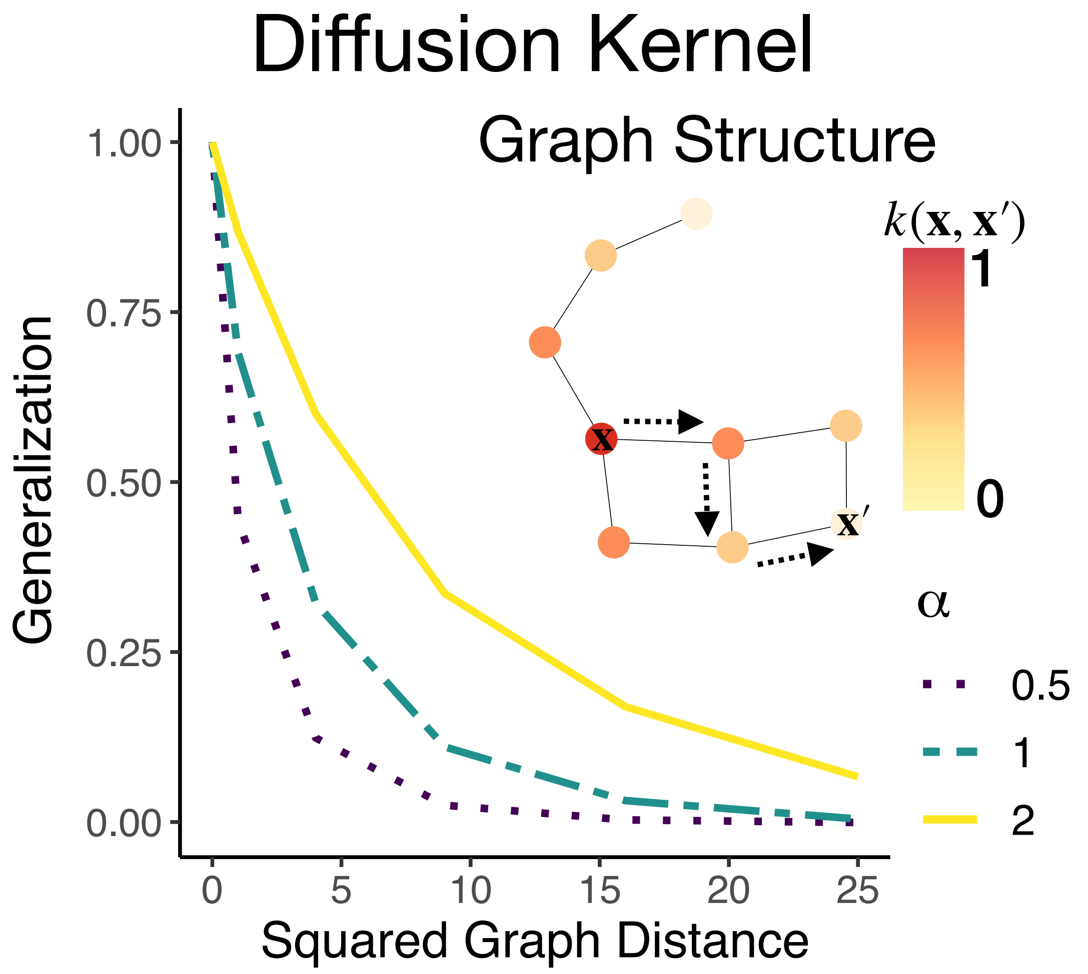
Wu et al., (*CBB* 2021)

Similarity can also capture relational structure

- The RBF kernel, like most classic accounts, represent similarity as distance in feature space
  - Learns smooth functions in a continuous domain
- A diffusion kernel represents similarity based on the connectivity of a graph
  - Learns functions on discrete graph representations



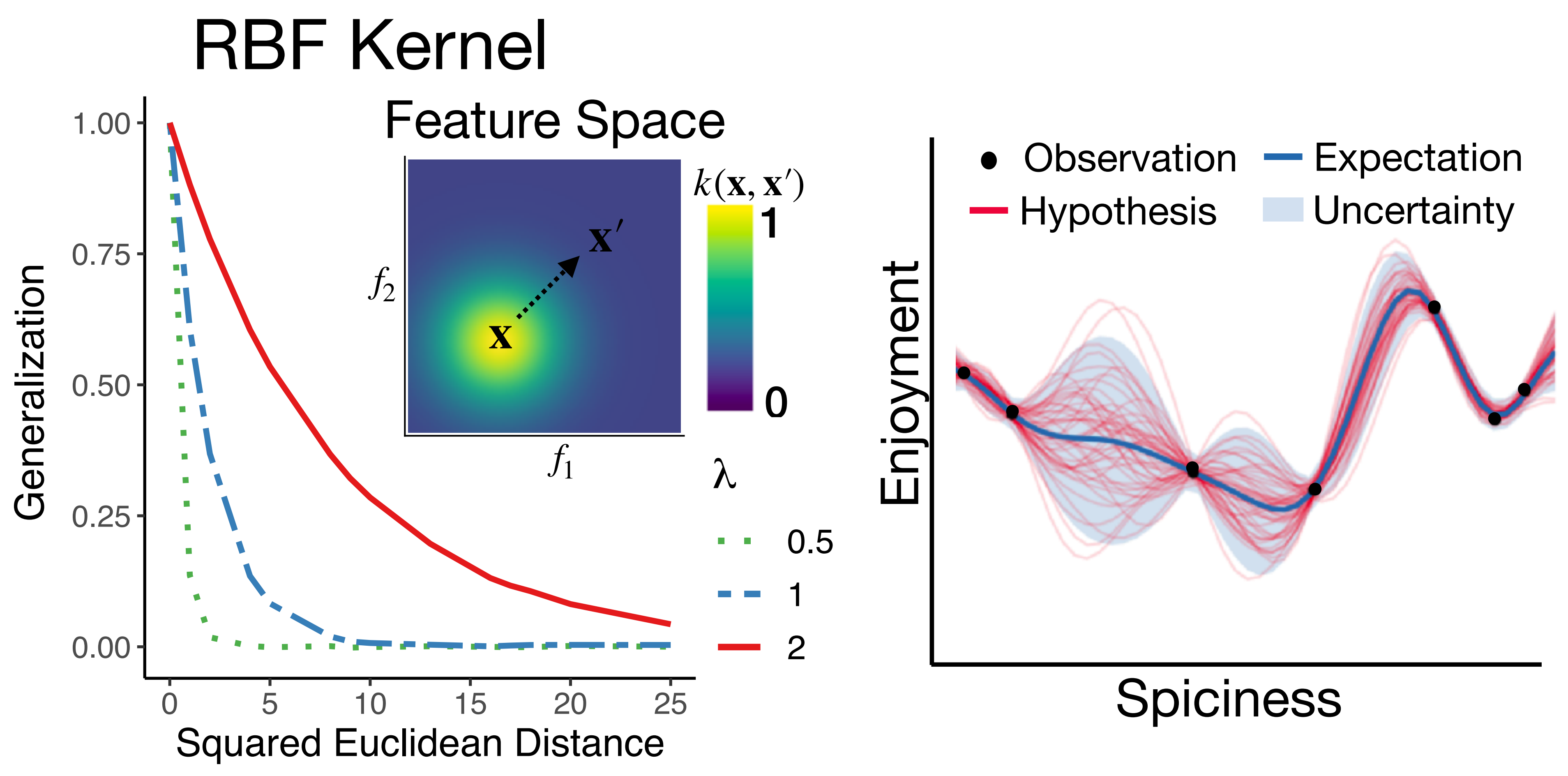


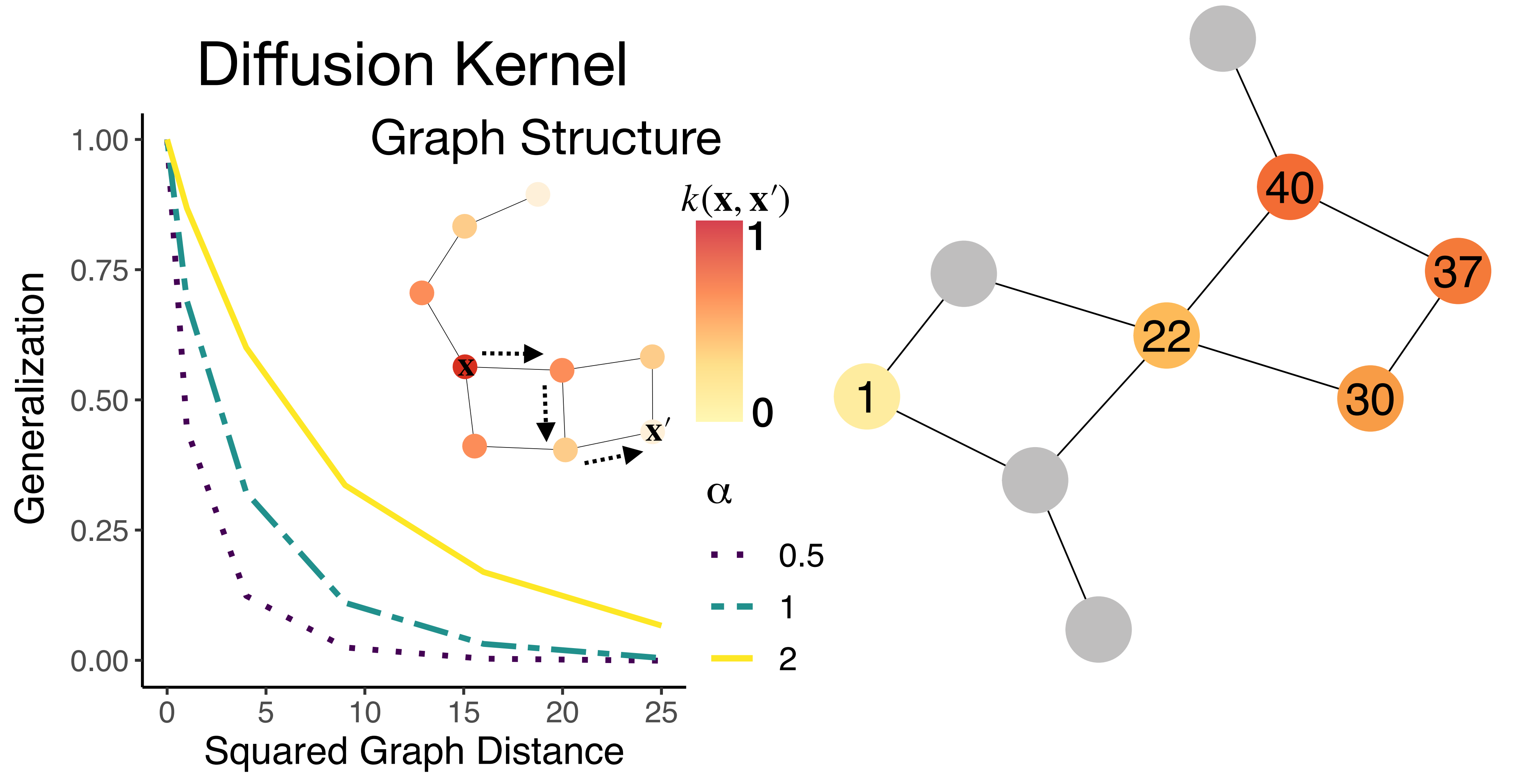


Wu et al., (*CBB* 2021)

Similarity can also capture relational structure

- The RBF kernel, like most classic accounts, represent similarity as distance in feature space
    - Learns smooth functions in a continuous domain
- A diffusion kernel represents similarity based on the connectivity of a graph
    - Learns functions on discrete graph representations

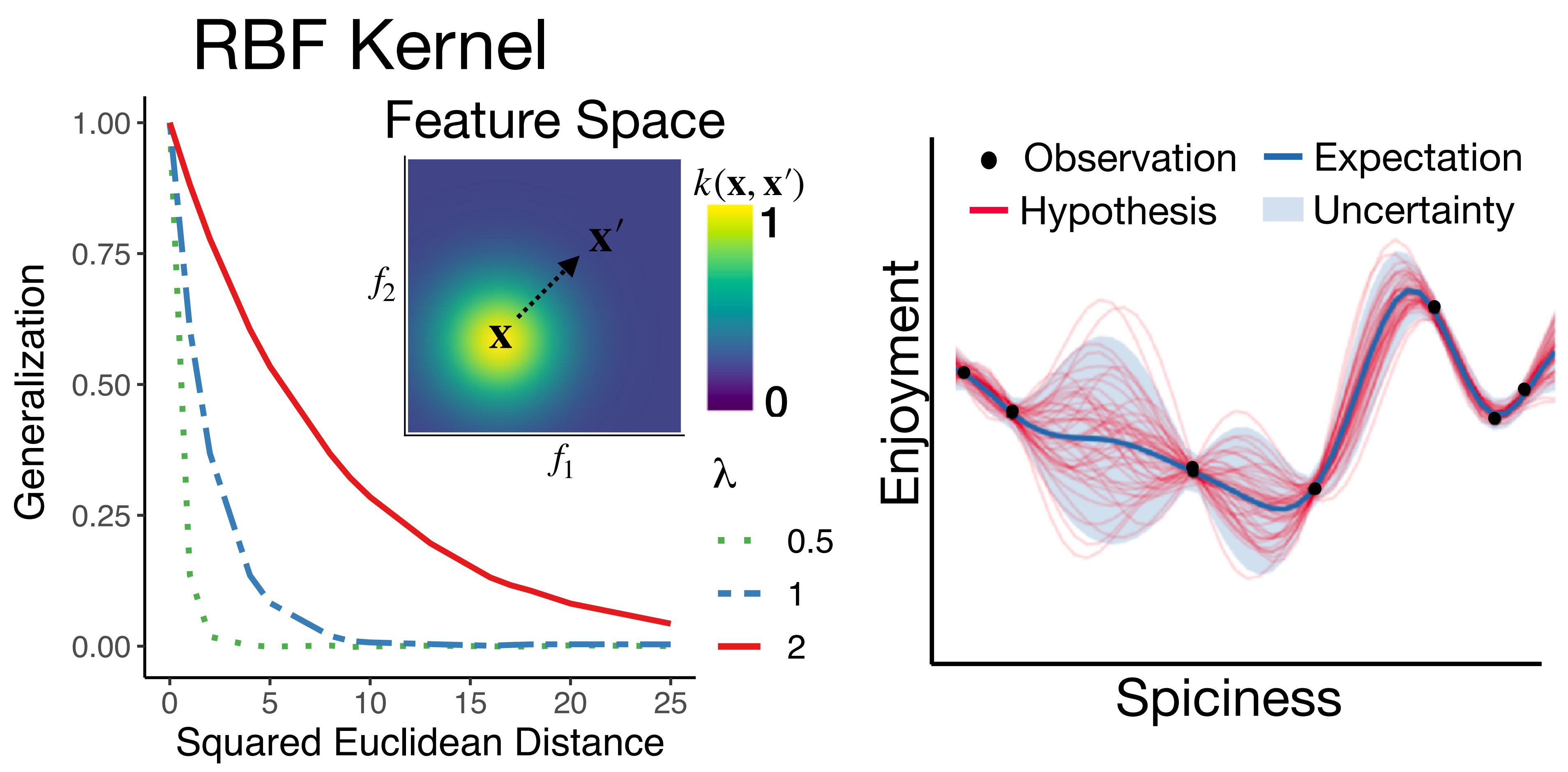


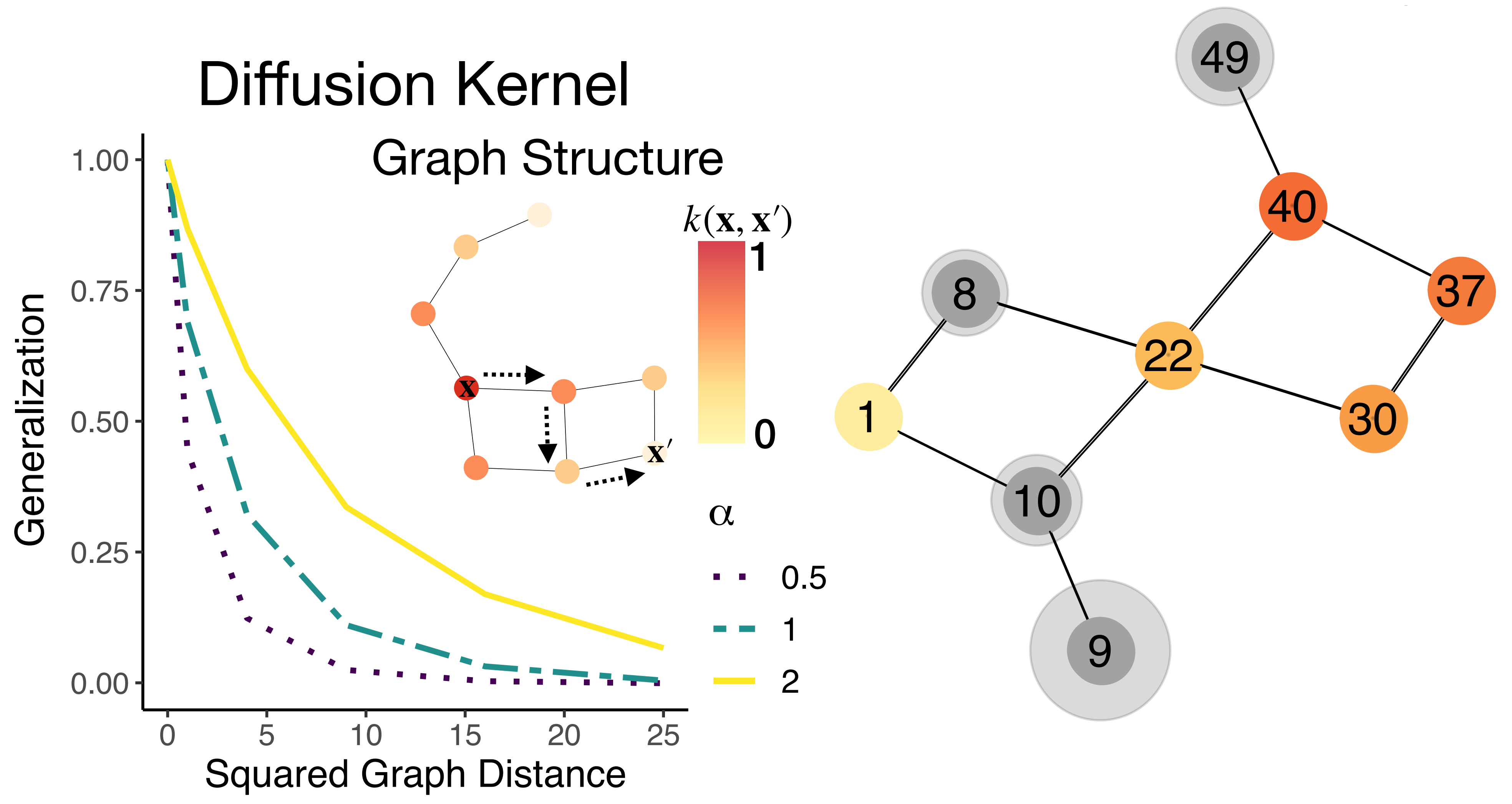


Wu et al., (*CBB* 2021)

Similarity can also capture relational structure

- The RBF kernel, like most classic accounts, represent similarity as distance in feature space
  - Learns smooth functions in a continuous domain
- A diffusion kernel represents similarity based on the connectivity of a graph
  - Learns functions on discrete graph representations

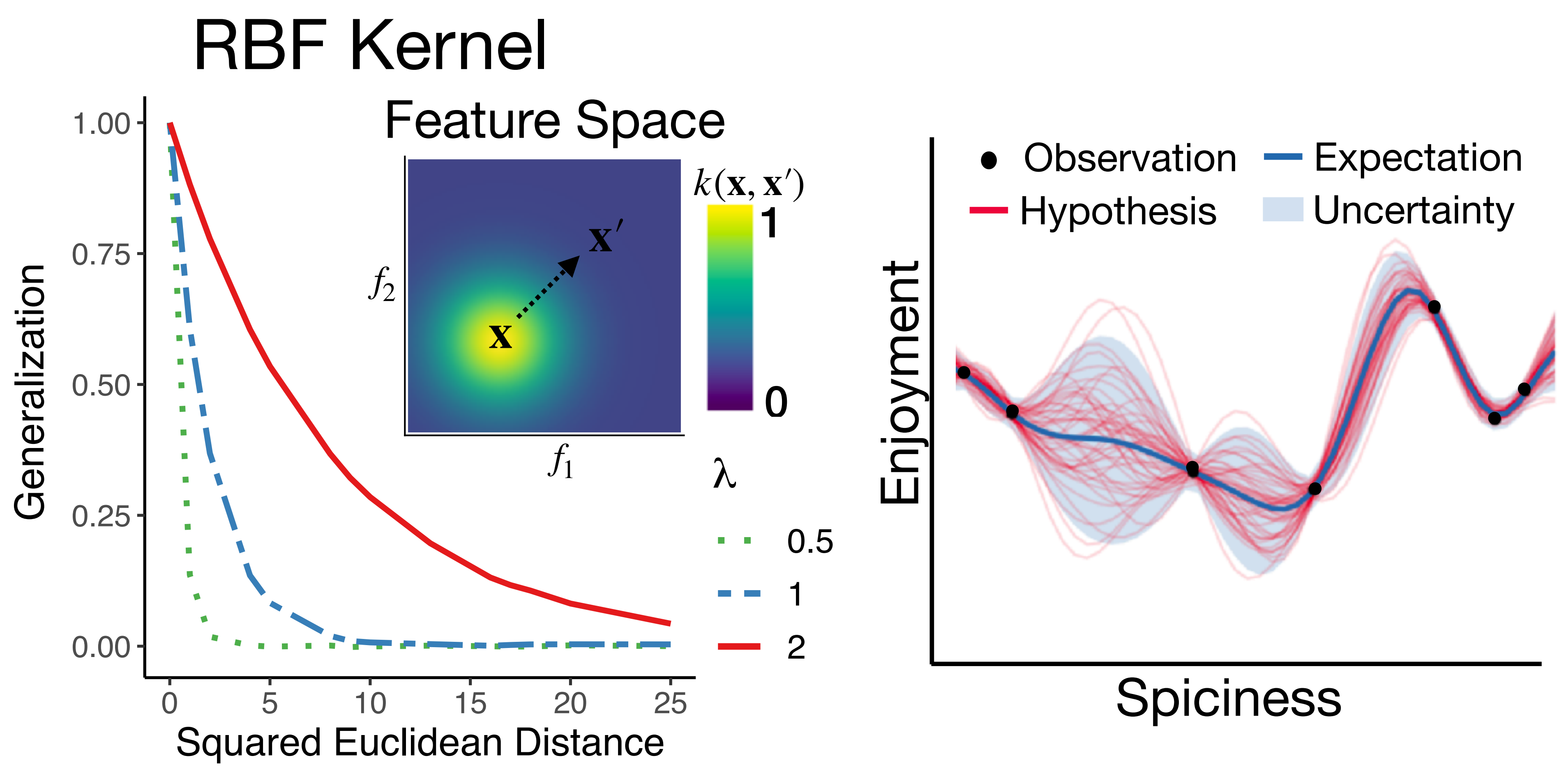


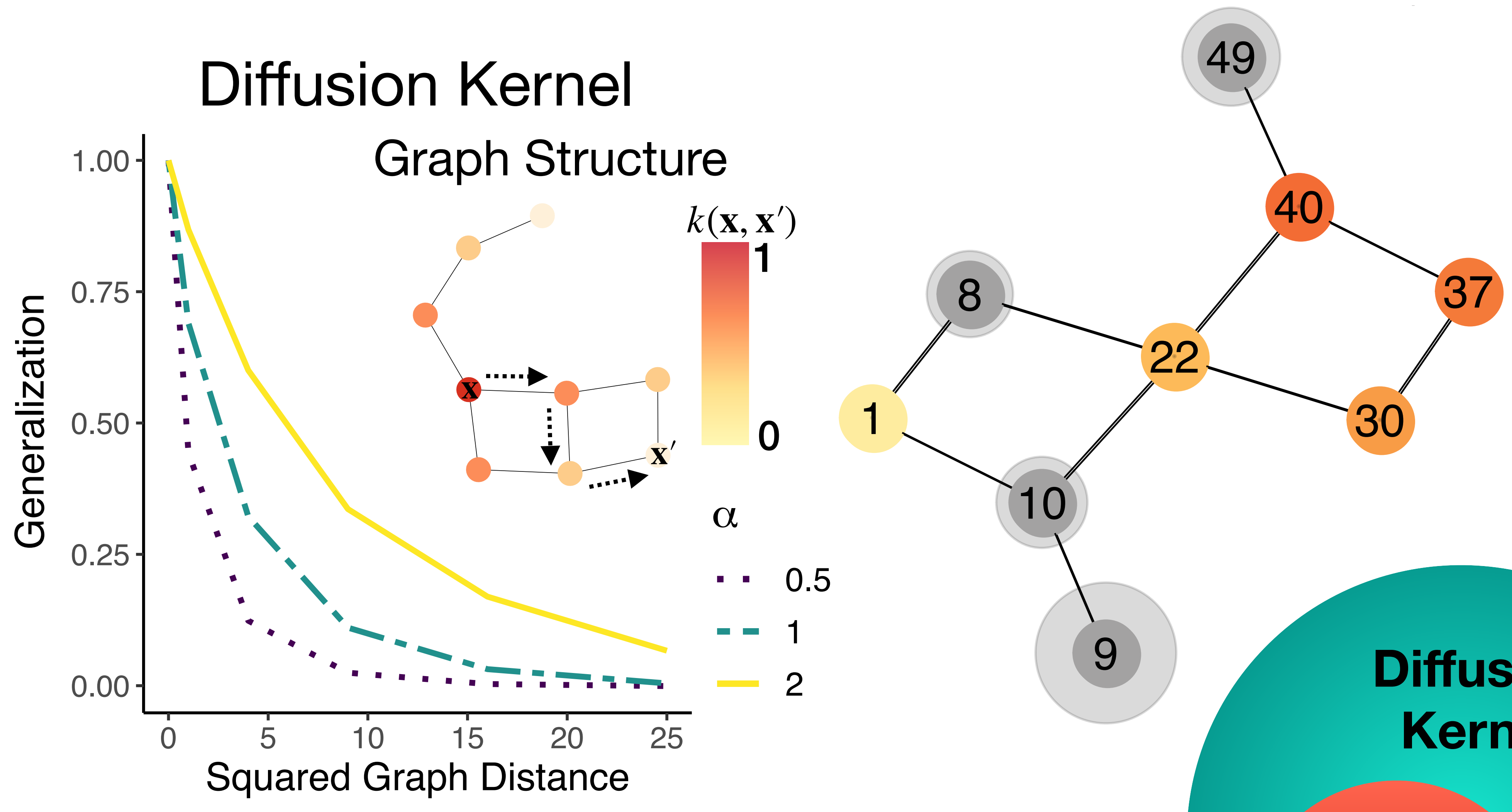


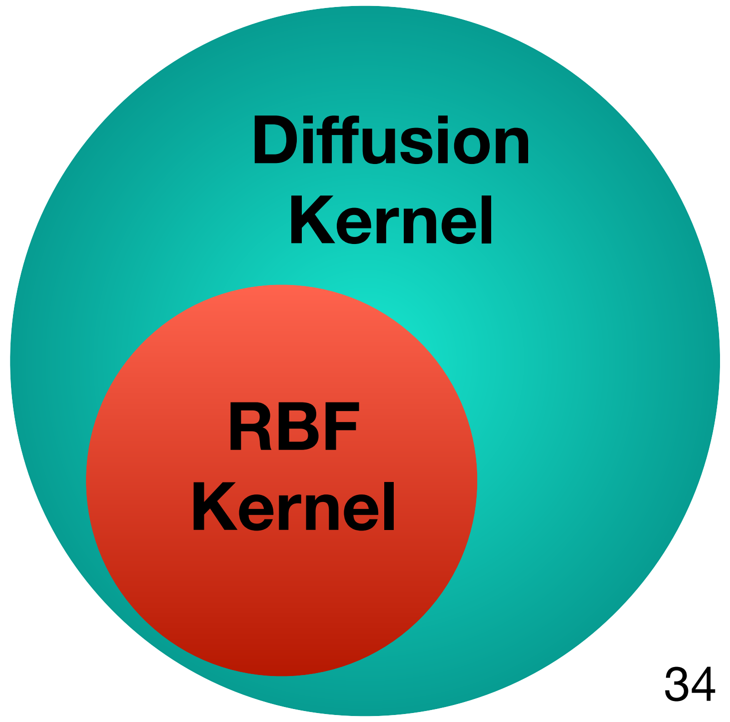
Wu et al., (*CBB* 2021)

Similarity can also capture relational structure

- The RBF kernel, like most classic accounts, represent similarity as distance in feature space
    - Learns smooth functions in a continuous domain
- A diffusion kernel represents similarity based on the connectivity of a graph
    - Learns functions on discrete graph representations
- RBF kernel = Diffusion kernel in the limit of an infinitely fine lattice graph





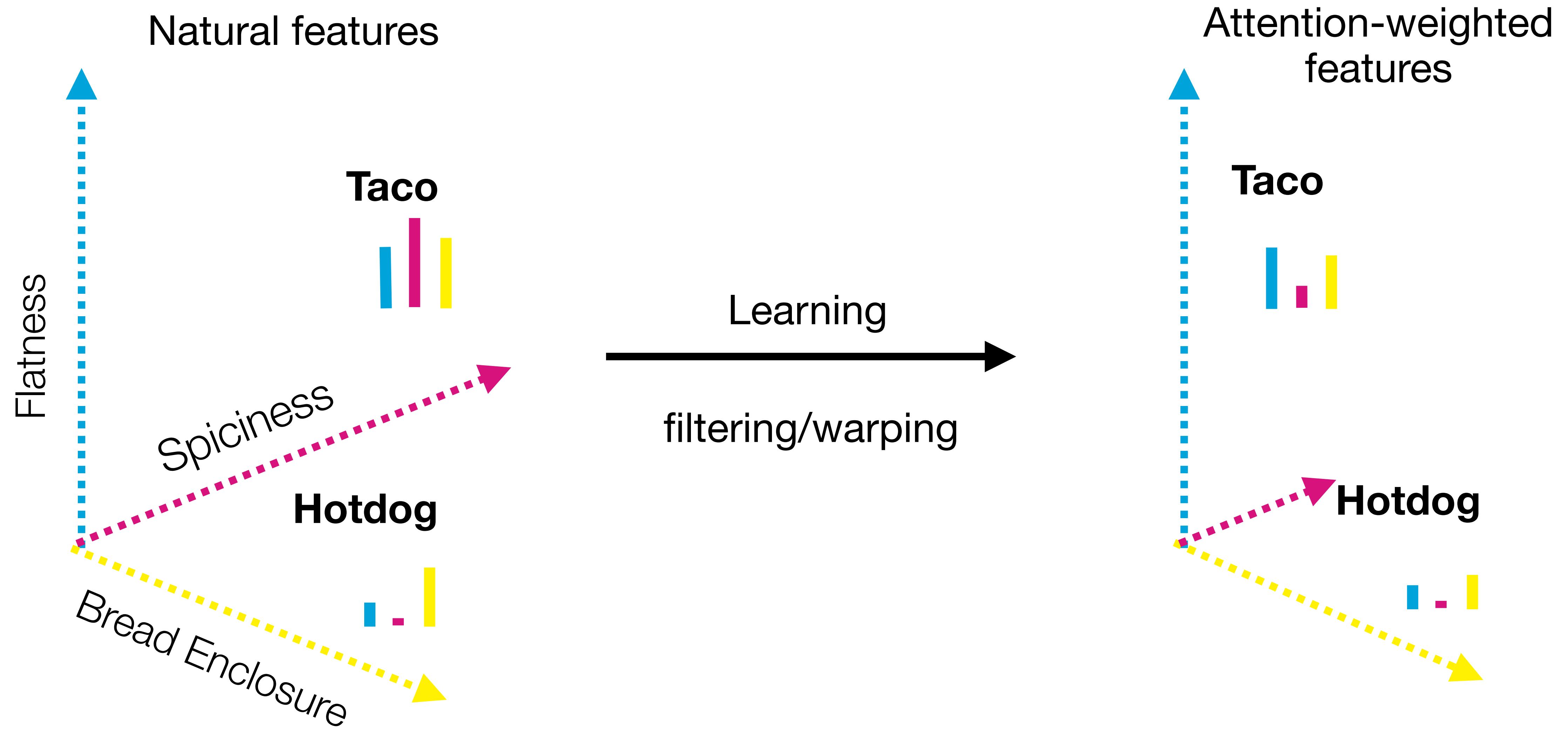


Wu et al., (*CBB* 2021)

Open challenges

Selective Attention

- Early models included **attentional weights** to prioritize similarity comparisons along relevant feature dimensions, but assumed weights were known (Nosoftsky, *JEP:G* 1986; Love et al., *PsychRev* 2004)
- Recently, theories of selective attention describe the learning process whereby irrelevant features are gradually filtered out over the course of learning (Radulescu et al., *AnnuRevNeuro* 2021)
  - These theories largely align with rational theories of attention, which balance cost of control vs. benefits of increased performance (Gottlieb et al., *CurrOpBehavSci* 2020; Dayan et al., *NatNeuro* 2000)
- While this provides a means to convert raw features into some "psychological space", it fails in natural settings (e.g., with rich visual features) —> we can't simply attend to all visual features in a scene and then learn to ignore irrelevant ones
- **Open Question:** How do we learn to attend to relevant features in real-world problems, when we cannot consider all of them?



Selective Attention

- Early models included **attentional weights** to prioritize similarity comparisons along relevant feature dimensions, but assumed weights were known (Nosoftsky, *JEP:G* 1986; Love et al., *PsychRev* 2004)
- Recently, theories of selective attention describe the learning process whereby irrelevant features are gradually filtered out over the course of learning (Radulescu et al., *AnnuRevNeuro* 2021)
    - These theories largely align with rational theories of attention, which balance cost of control vs. benefits of increased performance (Gottlieb et al., *CurrOpBehavSci* 2020; Dayan et al., *NatNeuro* 2000)
- While this provides a means to convert raw features into some "psychological space", it fails in natural settings (e.g., with rich visual features) —> we can't simply attend to all visual features in a scene and then learn to ignore irrelevant ones
- **Open Question:** How do we learn to attend to relevant features in real-world problems, when we cannot consider all of them?

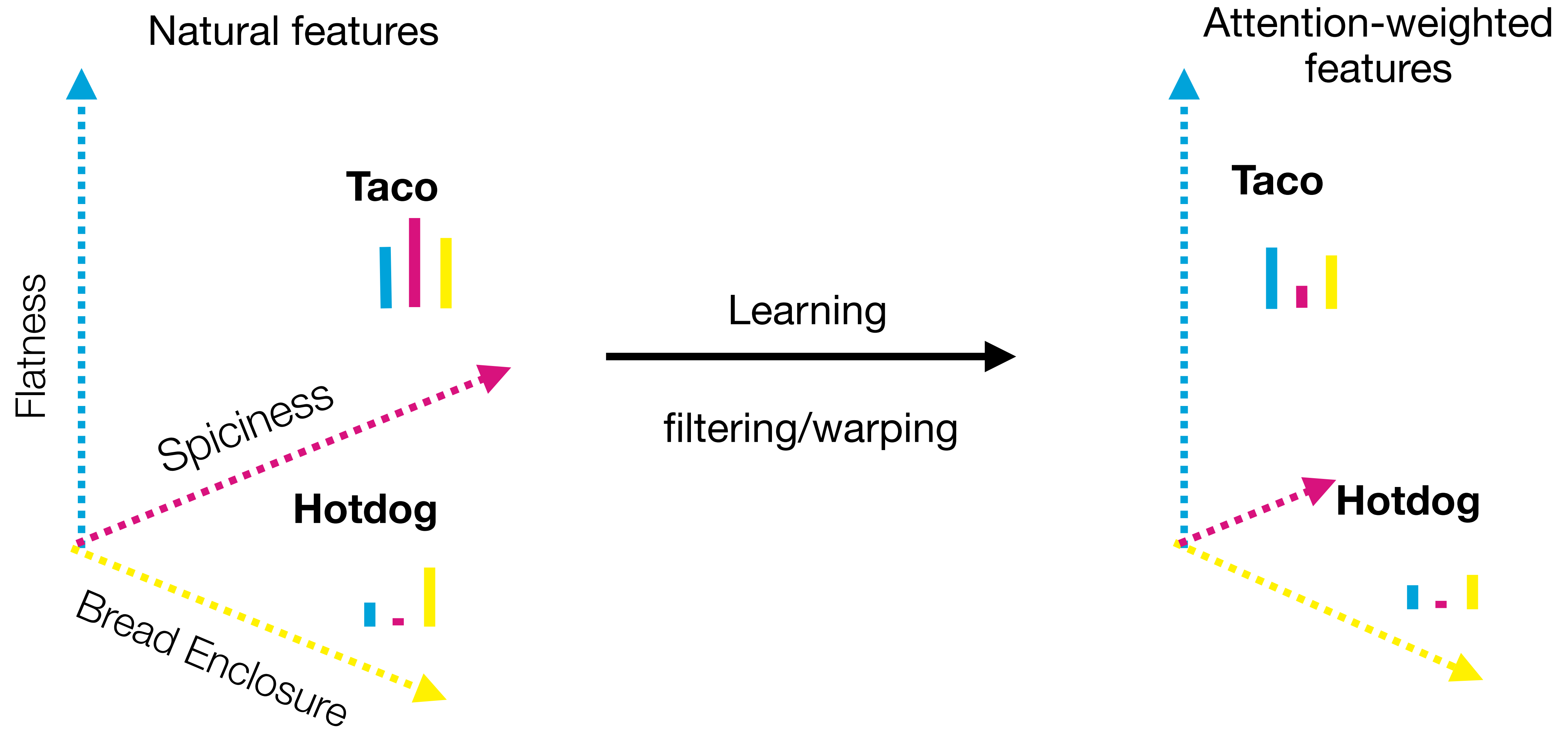
Feature RL model (Niv et al., *J.Neuro* 2015)

- assumes value is a sum of feature weights

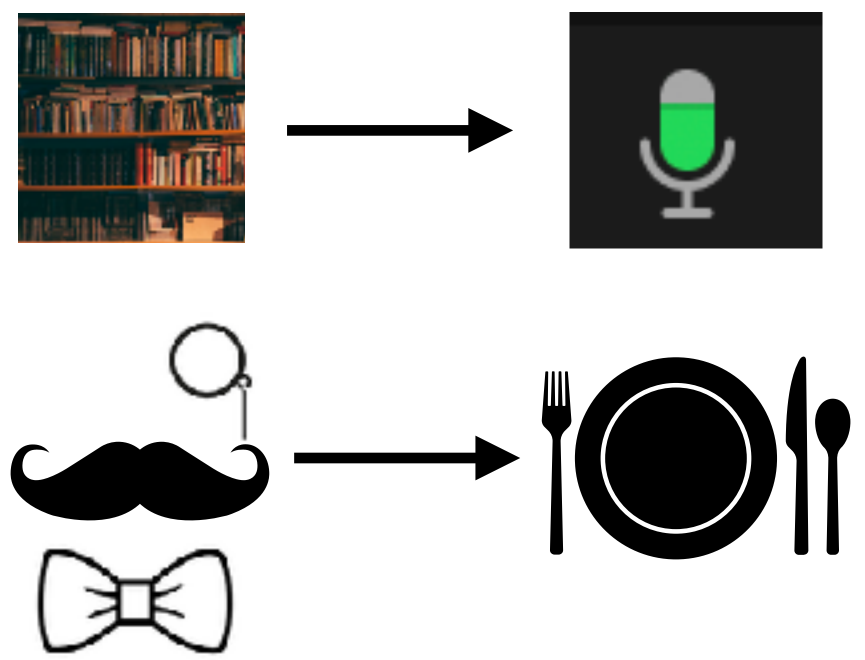
$$V(\mathbf{x}) = \sum_{\phi_i \in \Phi} \phi_i$$

- Weights are updated using the delta rule

$$\phi_i^{new} = \phi_i^{old} + \eta[R_t - V(\mathbf{x}_{chosen})], \quad \forall \phi_i \in \Phi$$

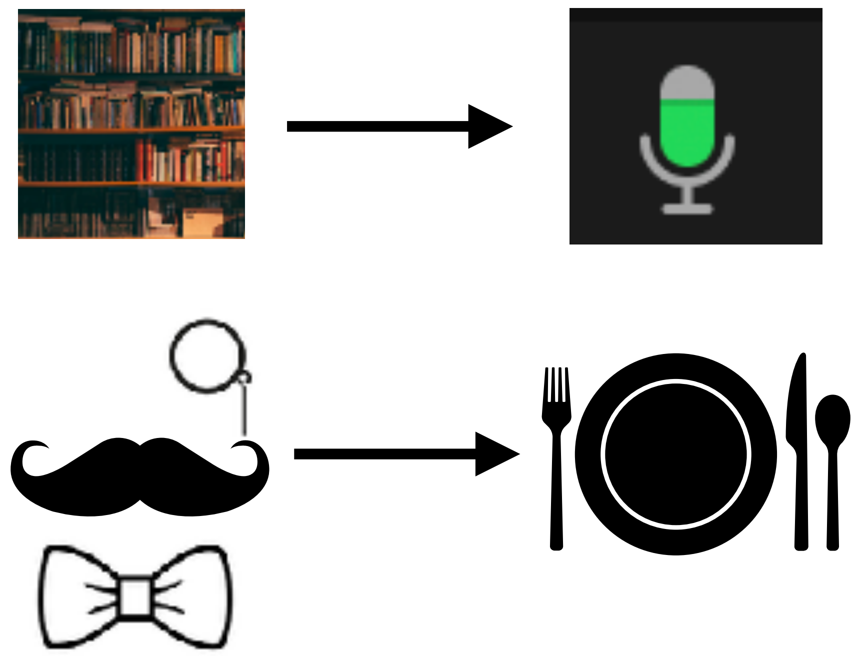


Contextual Clustering



- Different features are relevant in different contexts, which was already built into classic models of concept learning (Nosofsky, *JEP:G* 1986)
  - How do we infer which context or “event” we are in from continous streams of data?

Contextual Clustering

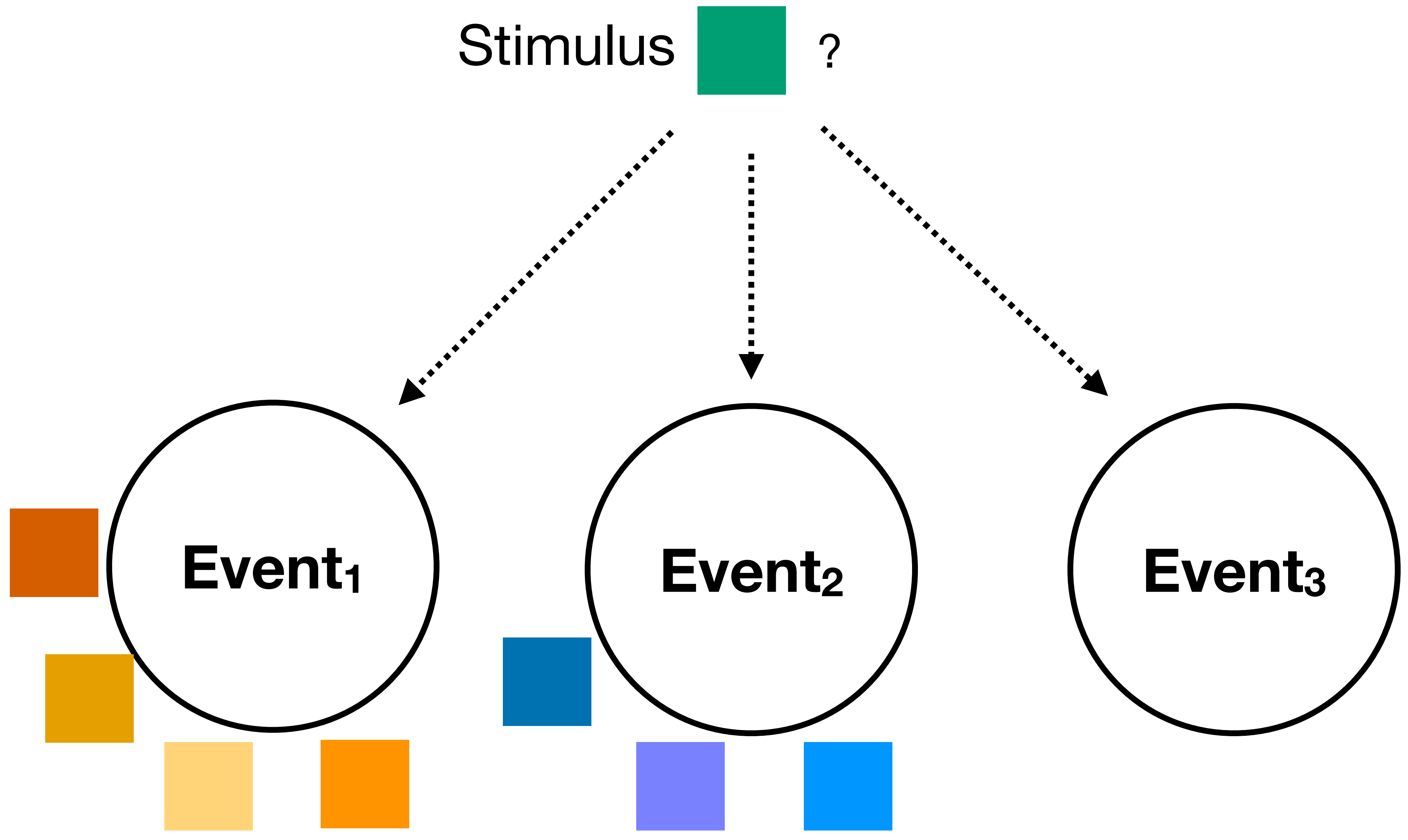


- Different features are relevant in different contexts, which was already built into classic models of concept learning (Nosofsky, *JEP:G* 1986)
  - How do we infer which context or "event" we are in from continous streams of data?
- We can frame this as a unsupervised clustering problem, and group related experiences into clusters (Franklin et al., *PsychRev* 2020; Gershman et al., *PsychRev* 2010)
  - Different event clusters can thus correspond to different attentional weights or different kernels
- **Open Question**: How do we transfer learned representations from one context to another?

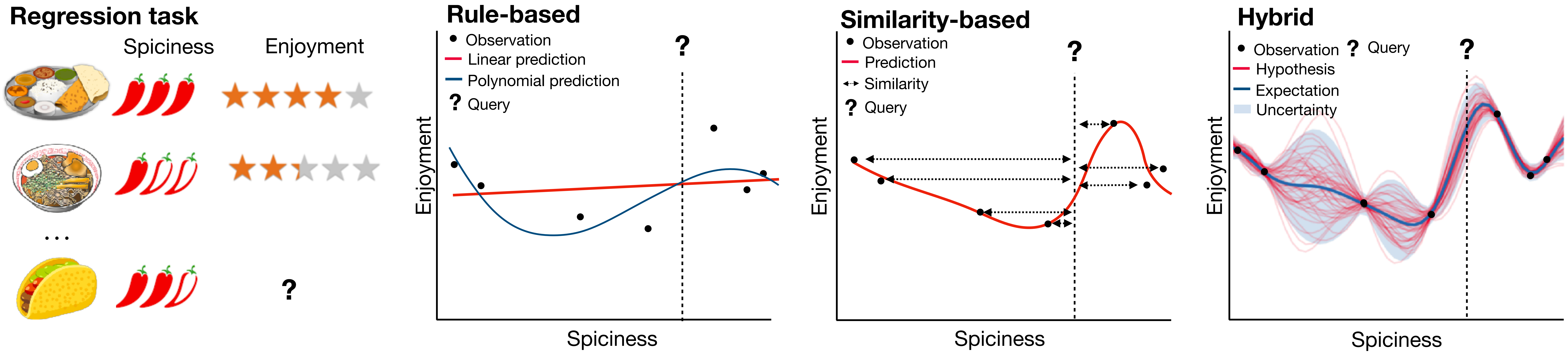
Chinese Restaurant Process (CRP)

- Similar to seating banquet guests
- We first try to seat the *i*th guest (i.e., stimulus) at one of the k existing table (i.e., event), otherwise we open a new table

$$p(\mathbf{x}_i = E_k) = \begin{cases} \frac{n_k}{n-1+\alpha} & \text{if k is an occupied table} \\ \frac{\alpha}{n-1+\alpha} & \text{otherwise (i.e., open a new table)} \end{cases}$$



Function Learning Summary



- Functions represent candidate hypotheses about the world allowing us to evaluate an infinite range of possibilities through interpolation and extrapolation
- Early **rule-based** approaches lacked flexibility, while **similarity-based** approaches didn't capture human inductive biases
- GP regression is a **hybrid** model, using the principles of Bayesian inference to compute a distribution over candidate hypotheses
- GPs not only capture how humans explicitly learn functions, but also how we implicitly learn a value function to guide our exploration in RL tasks with large search spaces
    - Originally tested in spatial environments (Wu et al,. 2018), but can also be applied to any arbitrary features (Wu et al,. 2020), or even graph-structured environments (Wu et al., 2021)

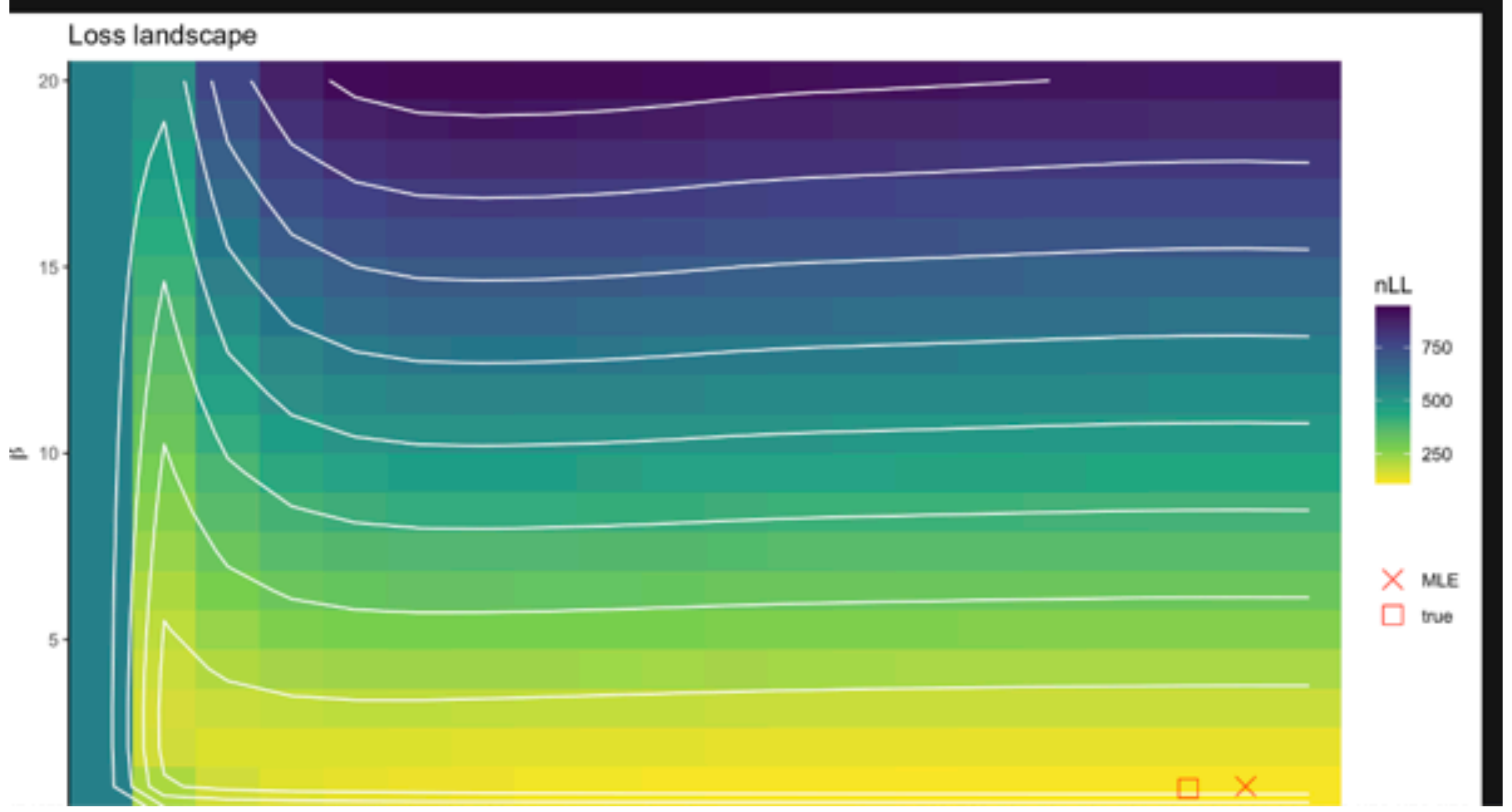
Next week


```
#Create a grid of plausible parameter combinations
paramCombs <- expand.grid(alpha = seq(0,1, length.out=20), beta = seq(0,20, lengt

#Now compute nLLs for each grid combination
nLLs <- sapply(1:nrow(paramCombs), FUN=function(i) likelihood(as.numeric(paramCom
paramCombs$nLL <- nLLs #add to dataframe

bestFit <- paramCombs[paramCombs$nLL ==  min(nLLs),] #best fitting combination

#plot data
ggplot(paramCombs)+
  geom_tile(aes(x = alpha, y = beta, fill = nLL)) +
  geom_contour(aes(x = alpha, y = beta, z = nLL), color = 'white')+
  geom_point(data=data.frame(alpha=alpha, beta=beta, type = 'true'), aes(alpha,be
  geom_point(data=data.frame(alpha=bestFit$alpha, beta=bestFit$beta, type = 'MLE'
  scale_fill_viridis_c('nLL', direction = -1)+
  labs(title = "Loss landscape",
       x = expression(alpha),
       y = expression(beta))+
  scale_shape_manual(values= c(4, 0), name = '')+
  theme_classic()+
  scale_x_continuous(expand = c(0, 0))+
  scale_y_continuous(expand = c(0, 0))
```



https://cosmos-konstanz.github.io/

Next week

Common tools for understanding brains and neural networks

- Manifold Analysis
- Representational Similarity Analysis

When things go wrong…

- Link to computational psychiatry