# COSMOS

## Computational summer school on modeling social and collective behavior - Konstanz (DE) July 4th - 7th

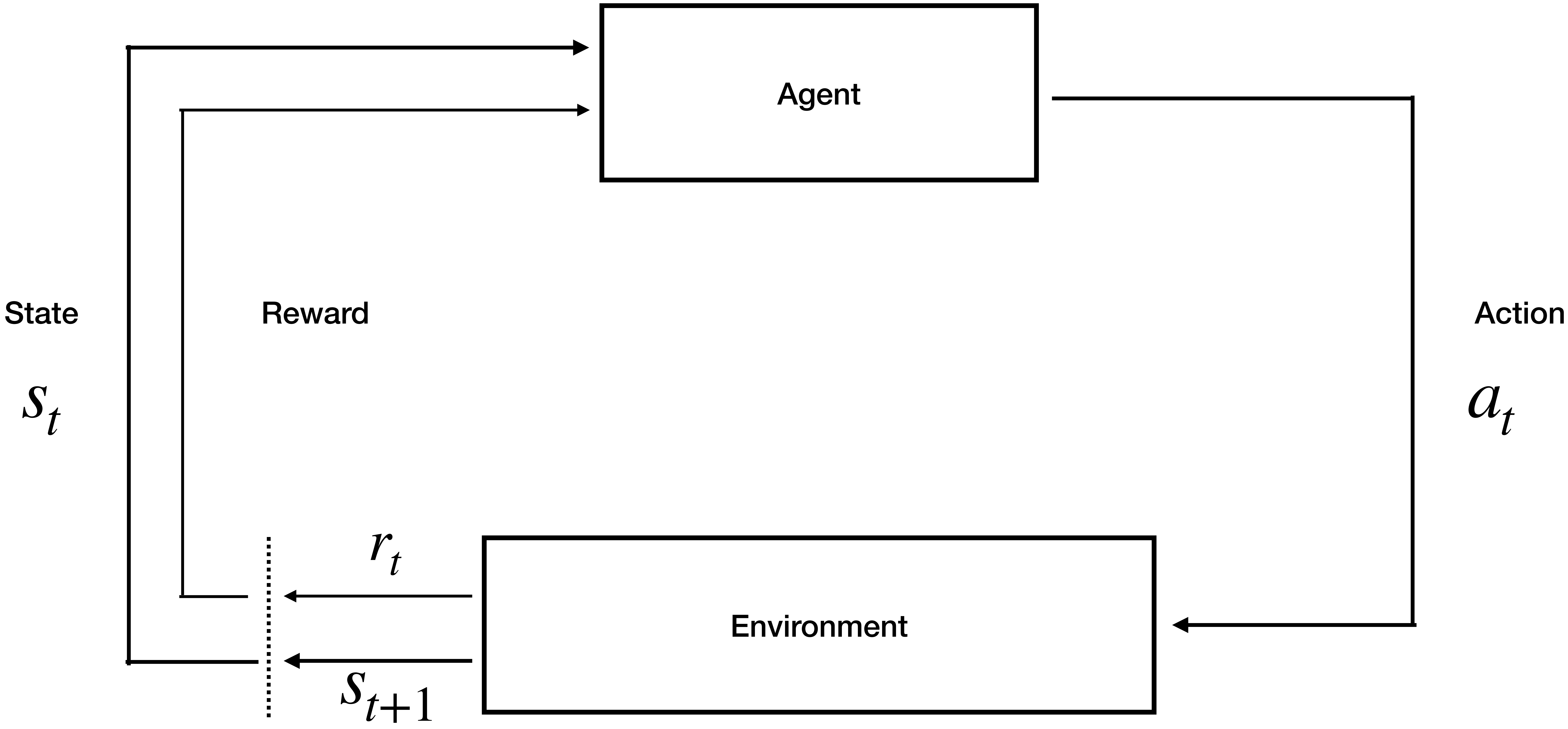
**Charley Wu & Wataru Toyokawa**

**July 5th**

# Goals of Tutorial 2:

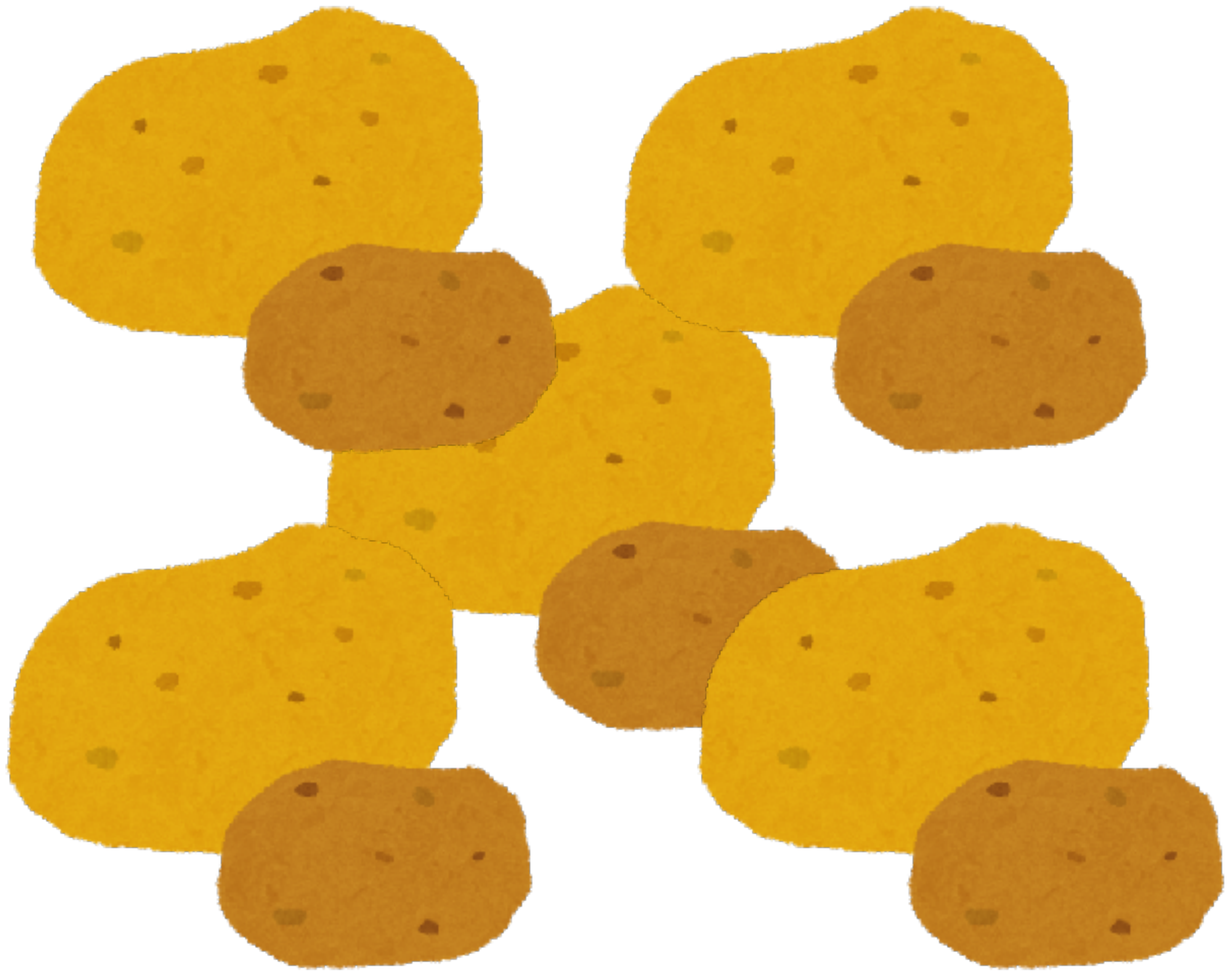
- **Brisk introduction to asocial RL**
  - Simulating data
  - Maximum likelihood estimation (MLE) of model parameters
  - Predicting choices
- **Social learning models**
  - Imitating actions
  - Combining asocial and social learning
  - Social learning hierarchy (from imitation to Theory of Mind)
- **Scaling up to more complex problems**
- **Evolutionary simulations**

# Reinforcement Learning (RL)

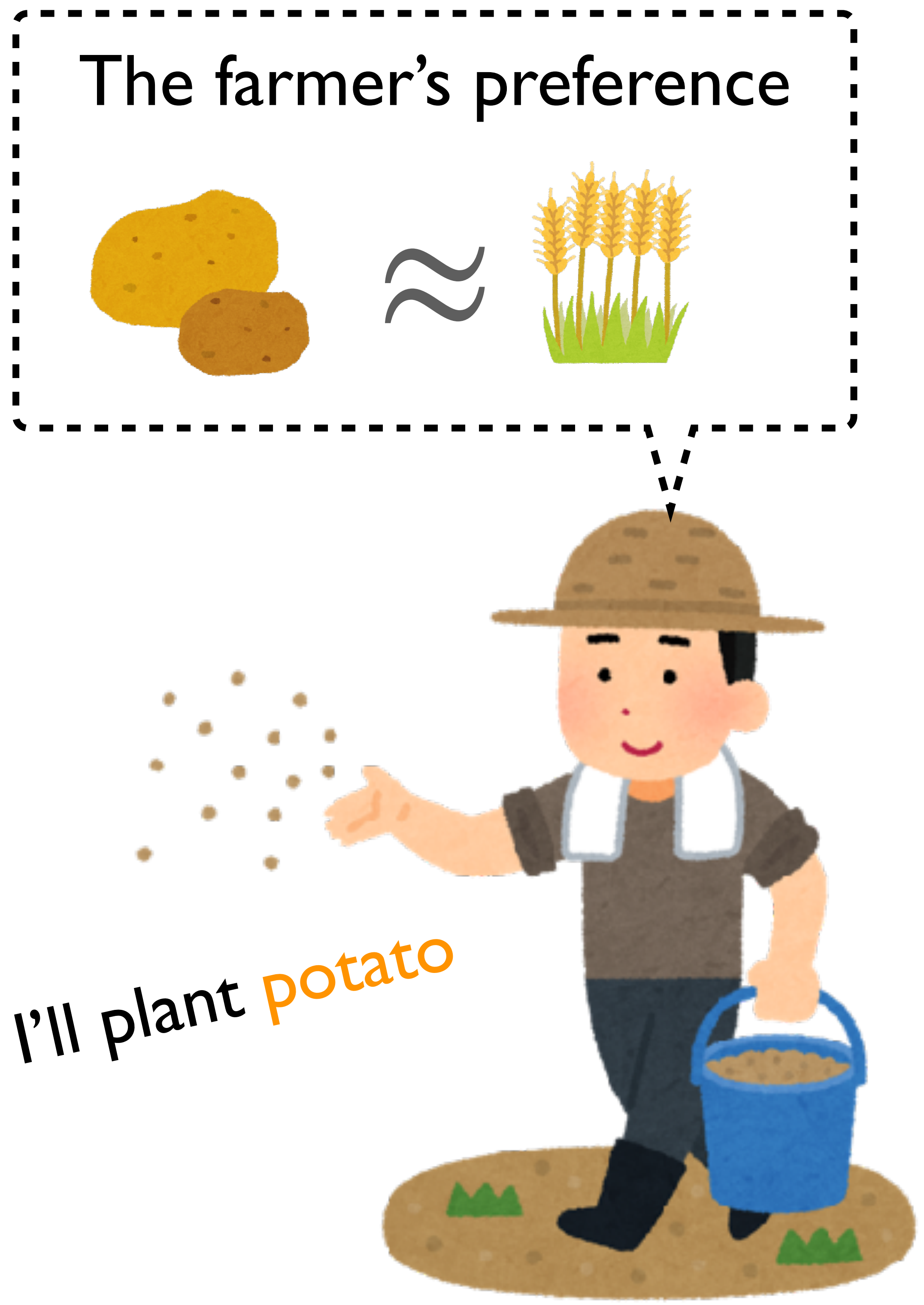



Sutton & Barto (1998)

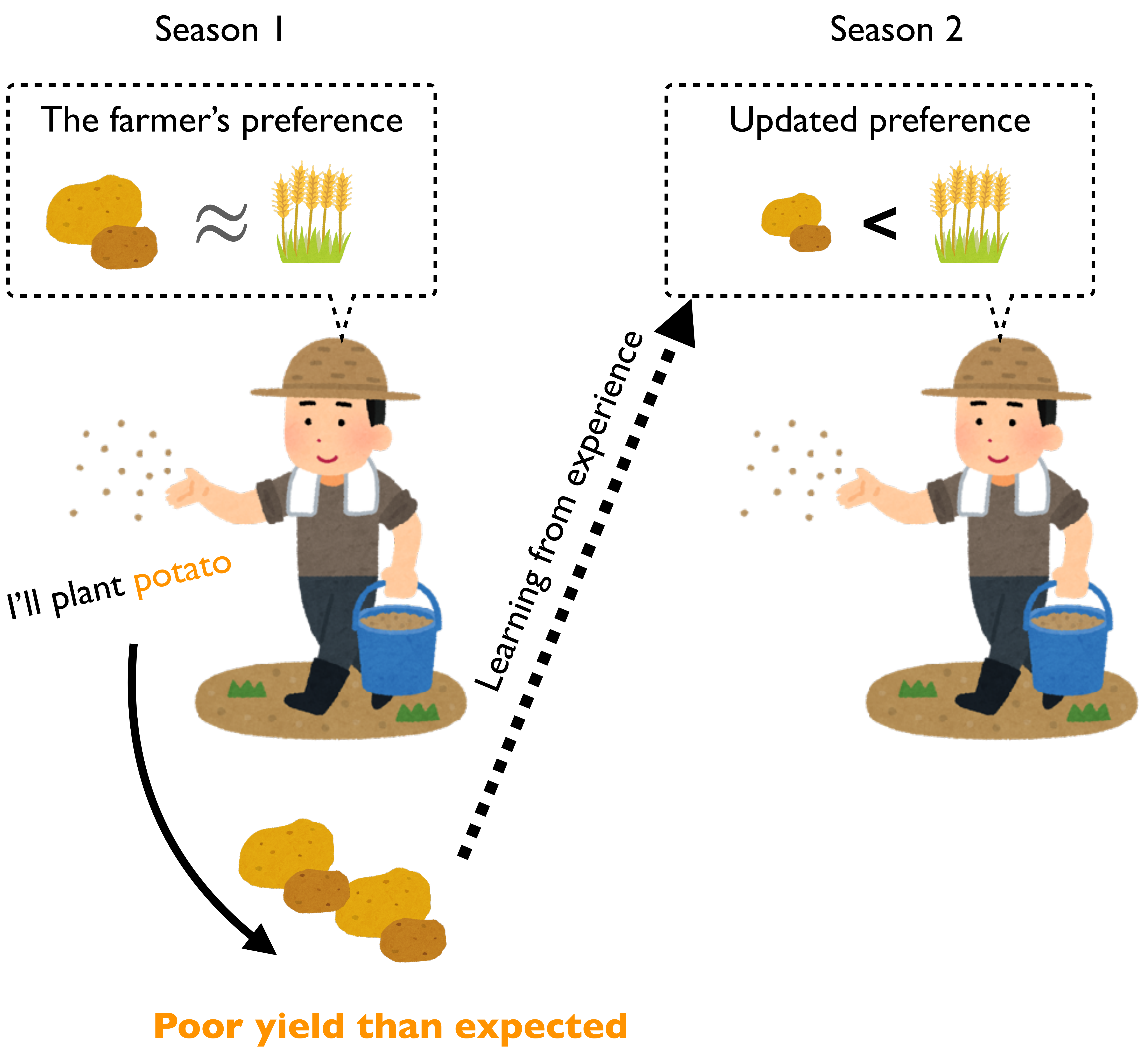
# A multi-armed bandit task

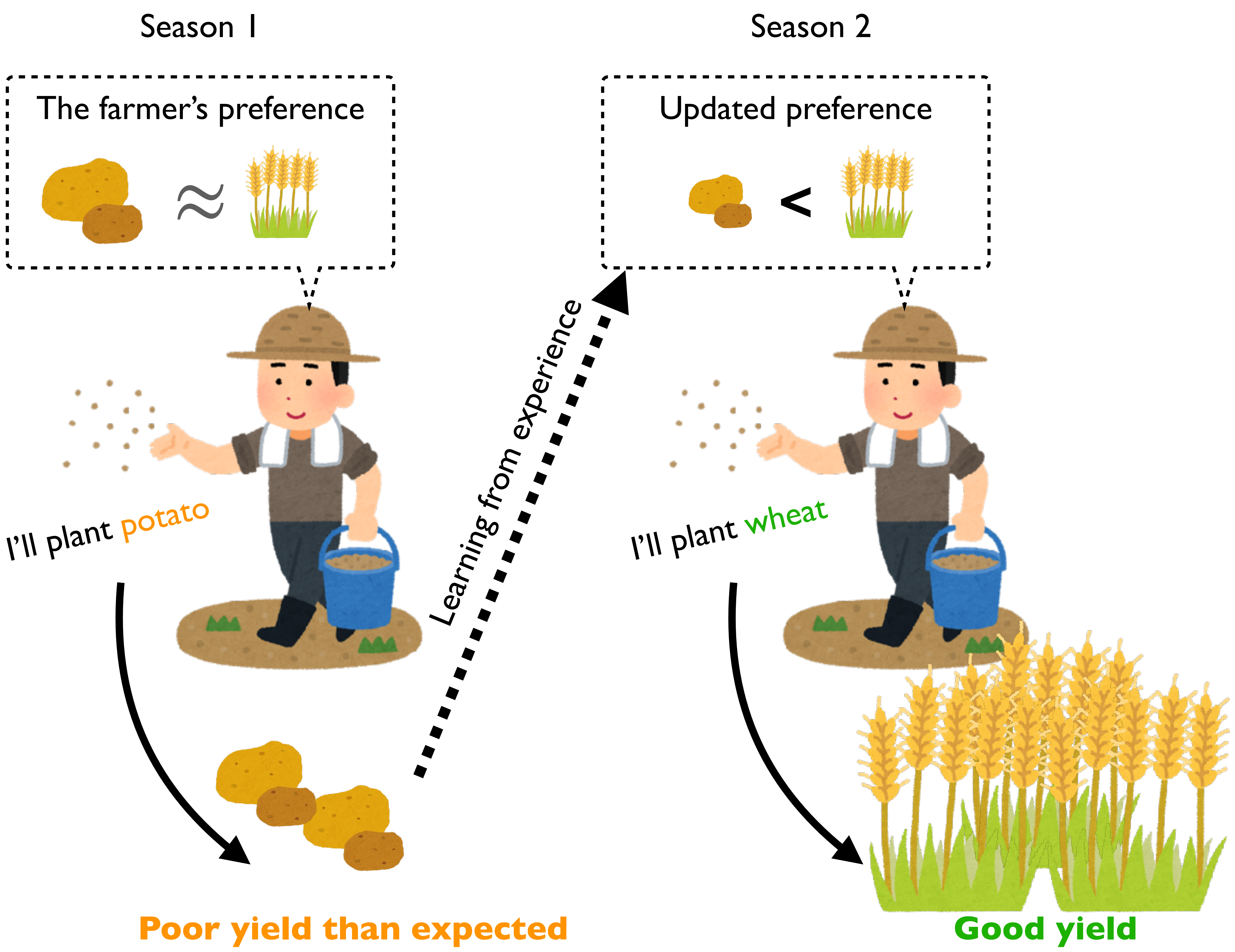


Season 1

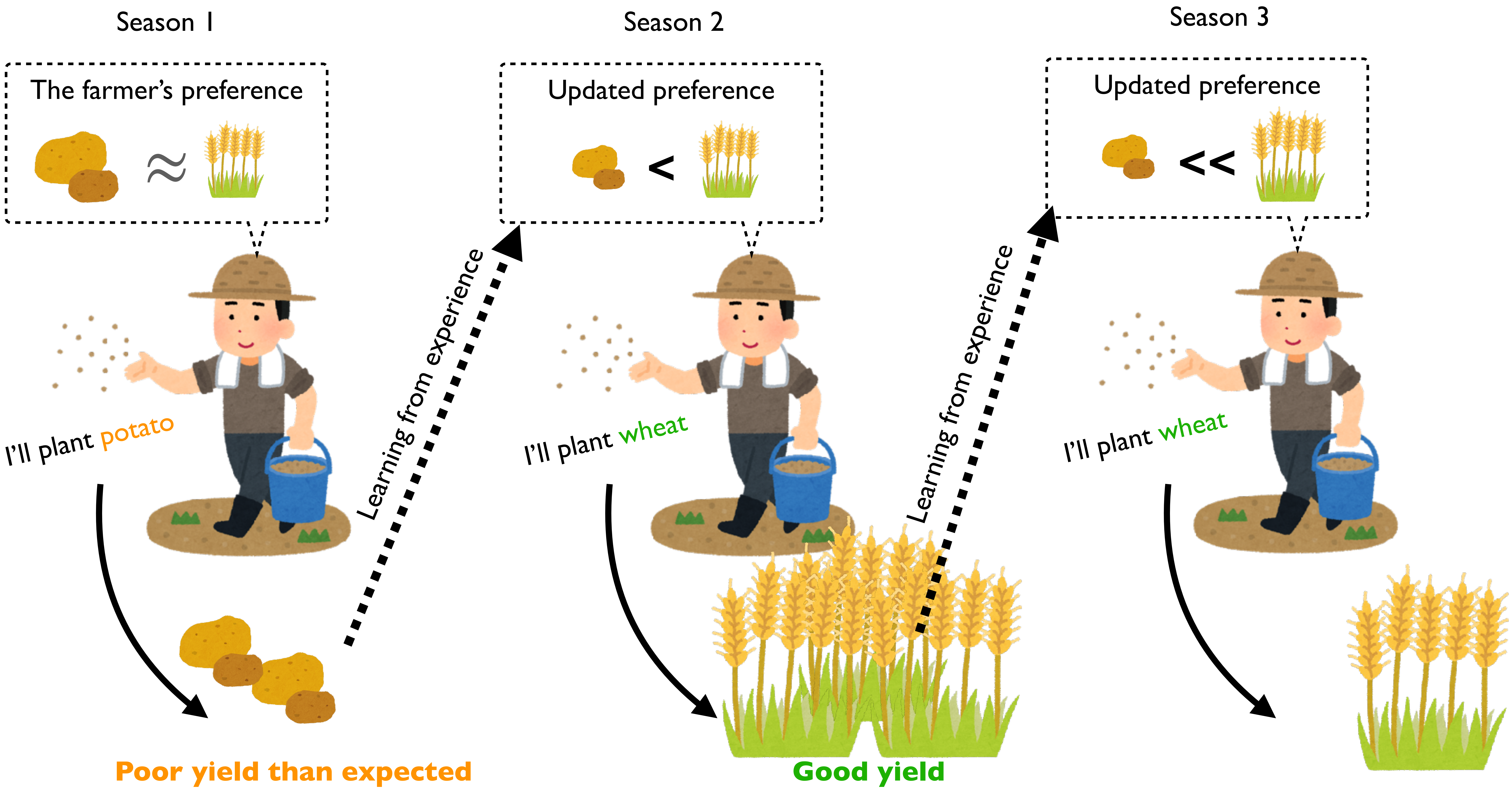


Season 1
The farmer's preference
≈
I'll plant potato
Poor yield than expected


Season 1
Season 2
The farmer's preference
≈
Updated preference
<
I'll plant potato
Learning from experience
Poor yield than expected

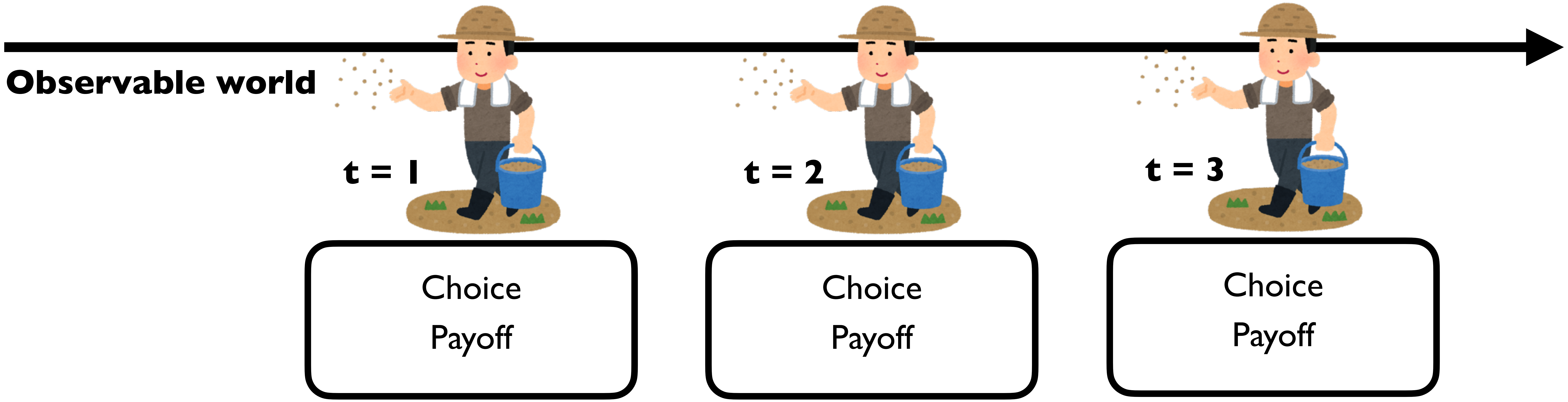


Season 1
The farmer's preference
Season 2
Updated preference
Learning from experience
I'll plant potato
I'll plant wheat
Poor yield than expected
Good yield

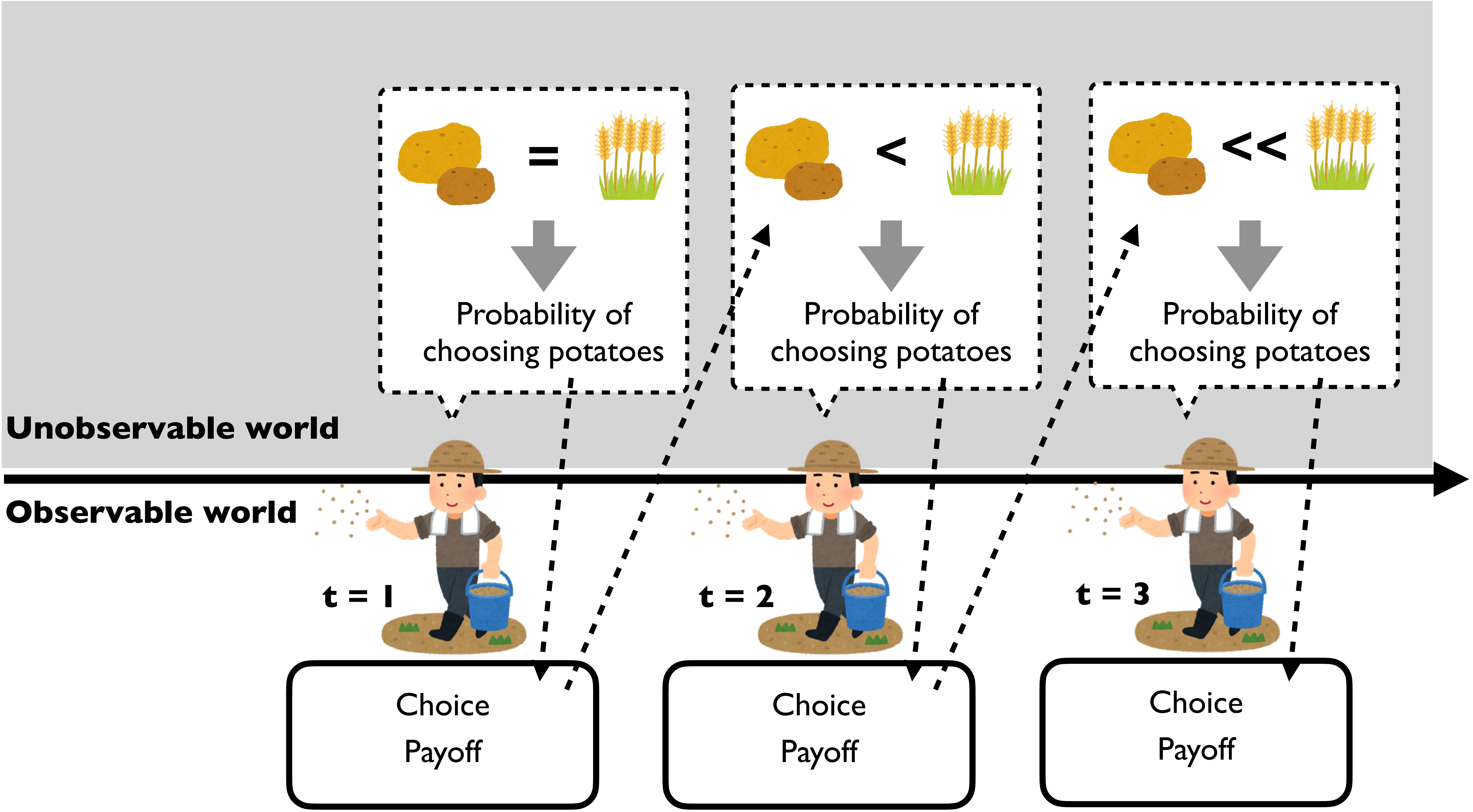


Season 1
Season 2
Season 3
The farmer's preference
≈
Updated preference
<
Updated preference
<<
I'll plant potato
Learning from experience
I'll plant wheat
Learning from experience
I'll plant wheat
Poor yield than expected
Good yield

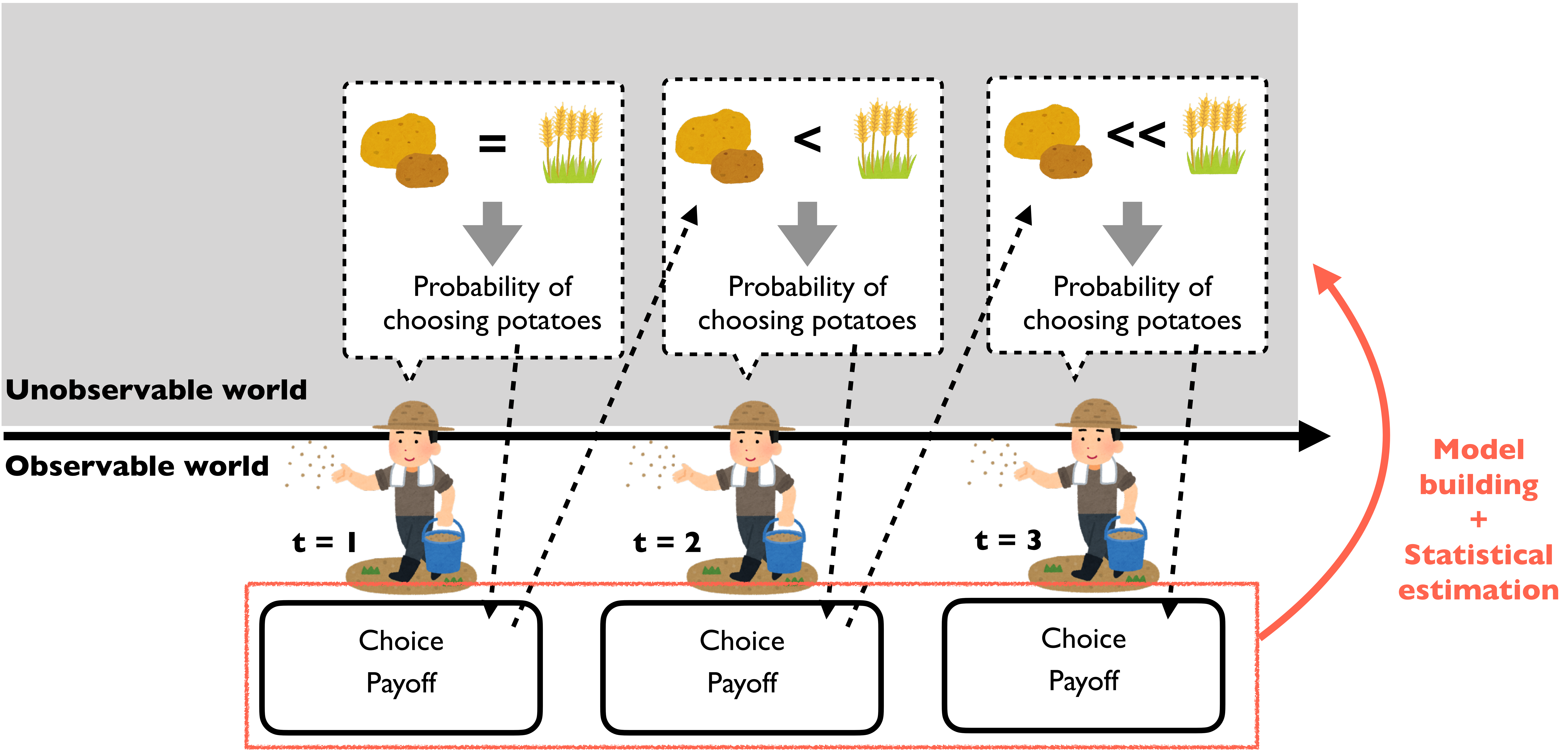
# RL process is not directly observable. It should be inferred statistically.



# RL process is not directly observable. It should be inferred statistically.

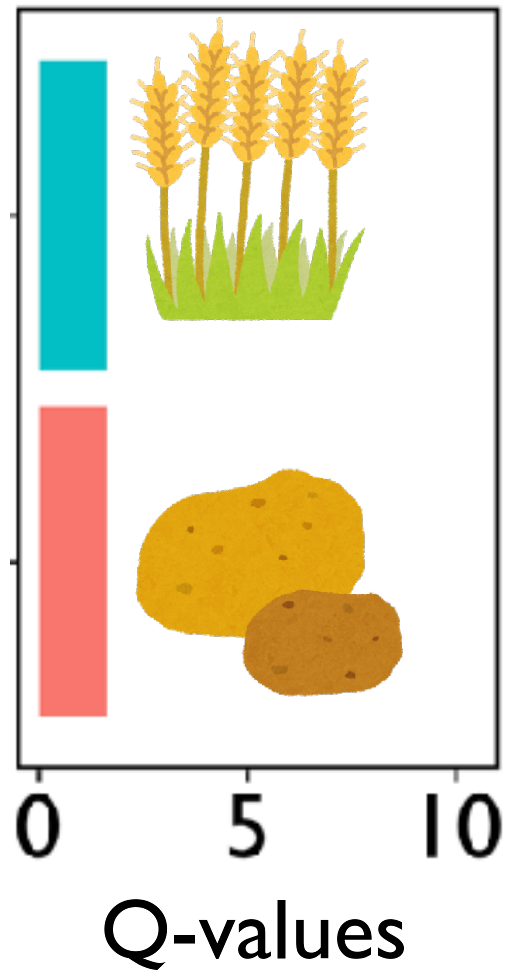


# RL process is not directly observable. It should be inferred statistically.



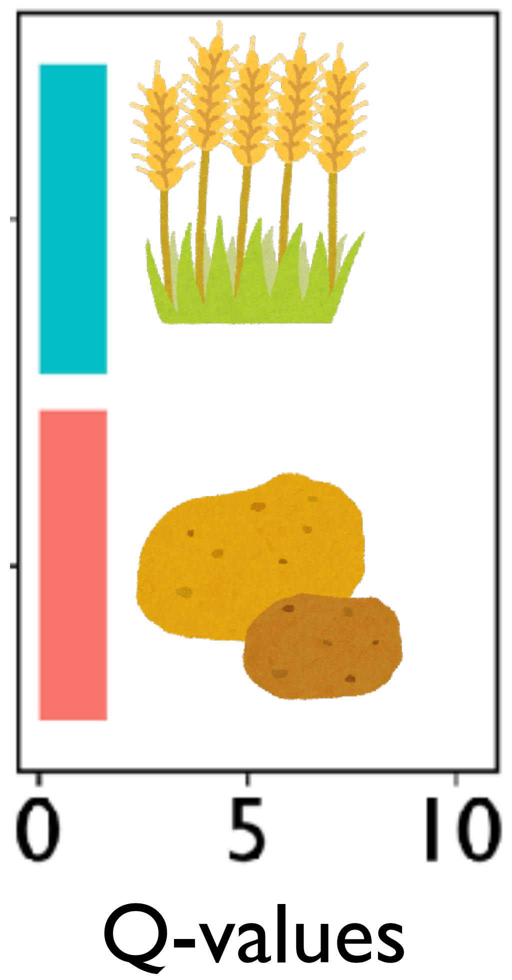
# Decision value updating: Q-Learning

**t = 1**



# Decision value updating:
# Q-Learning

**t = 1**




**Rewards**

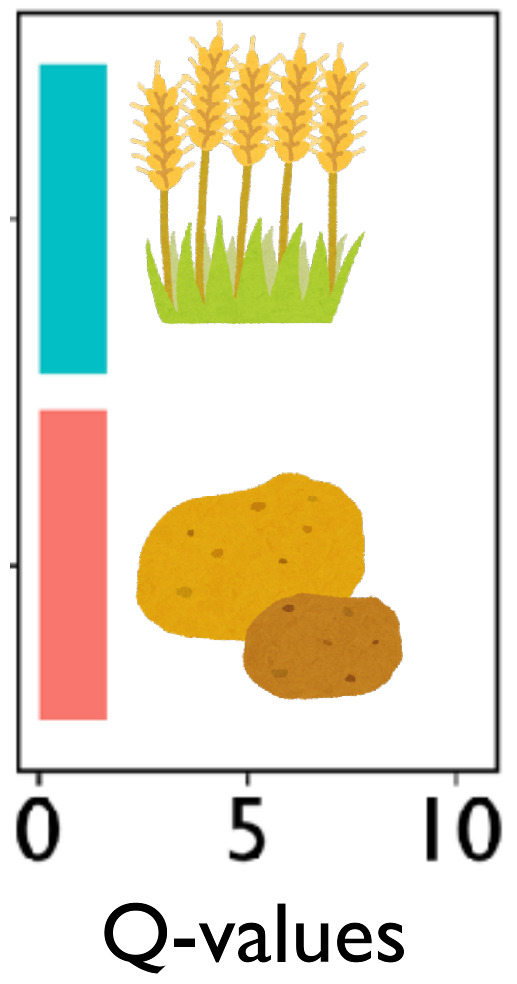
I'll plant wheat

# Decision value updating: Q-Learning

Learning rate
(memory parameter; step size)

t = 1

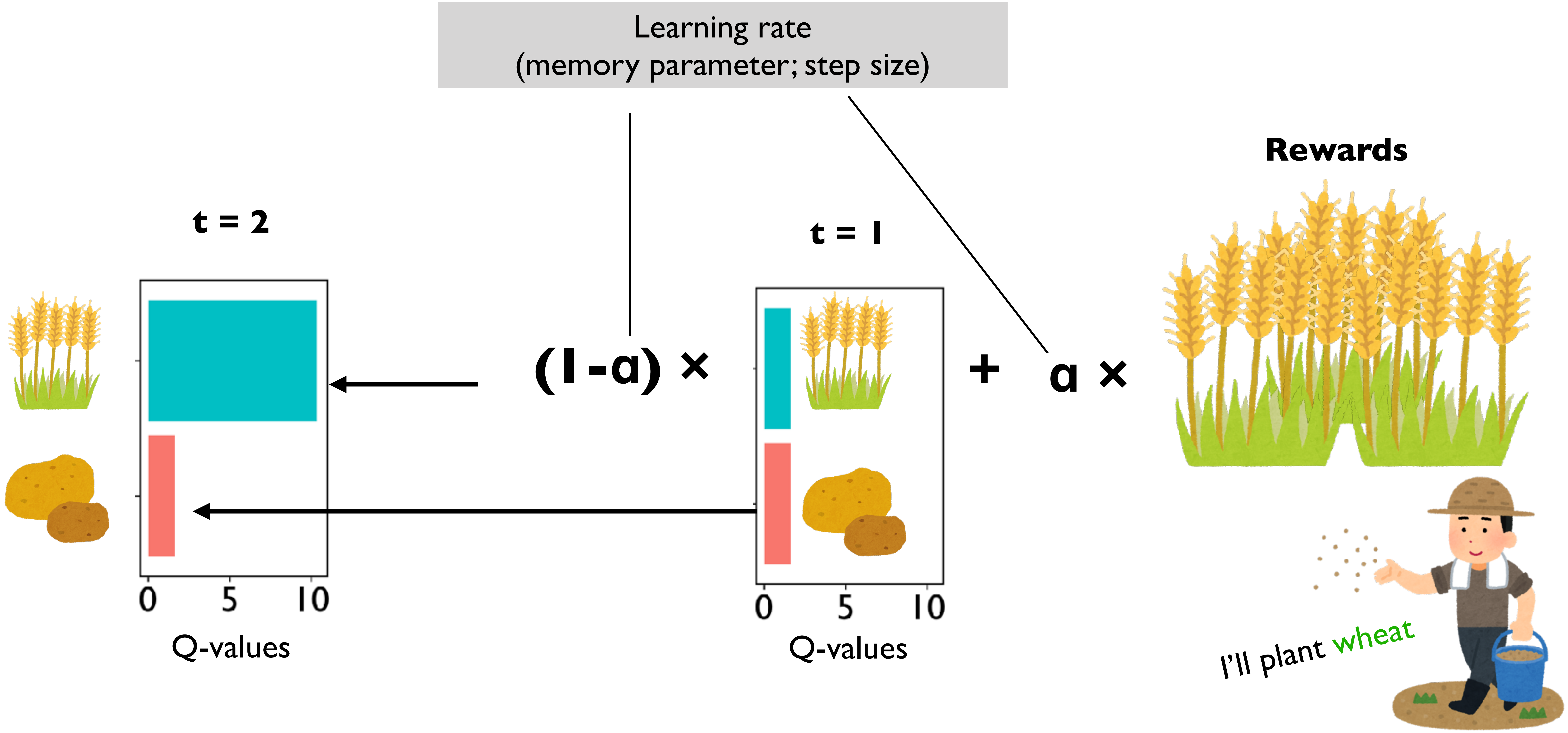
$(1-\alpha)\ \times$



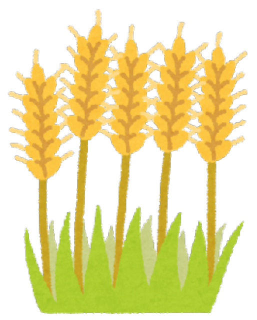

$+\ \alpha\ \times$

**Rewards**

# Decision value updating: Q-Learning

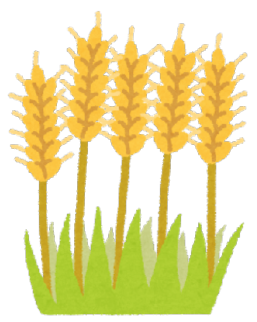


# Reward prediction error



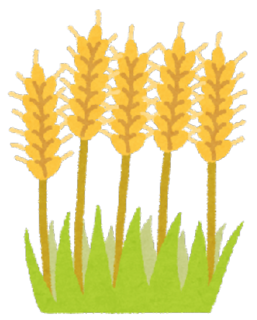
$$Q_t(a)$$

## Reward prediction error



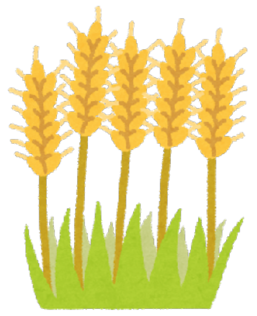
$$(1 - \alpha)Q_t(a) + \alpha r_t(a)$$

# Reward prediction error


$$Q_{t+1}(a) \leftarrow (1 - \alpha)Q_t(a) + \alpha r_t(a)$$

## Reward prediction error

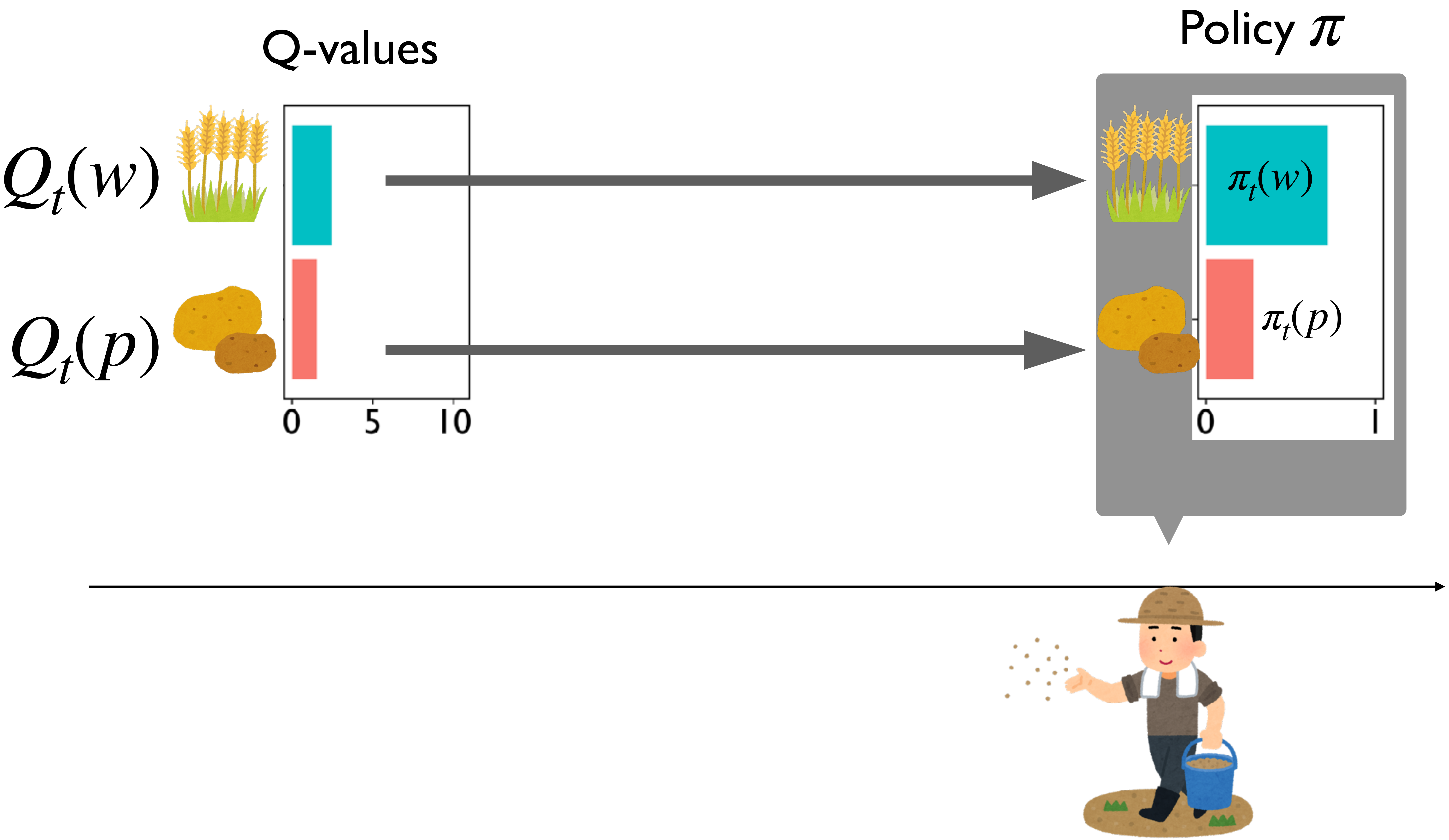




$$Q_{t+1}(a) \leftarrow (1-\alpha)Q_t(a) + \alpha r_t(a)$$

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

**reward prediction error (RPE)**

# Converting value to actions:
# Softmax policy (i.e. multinomial logistic function)

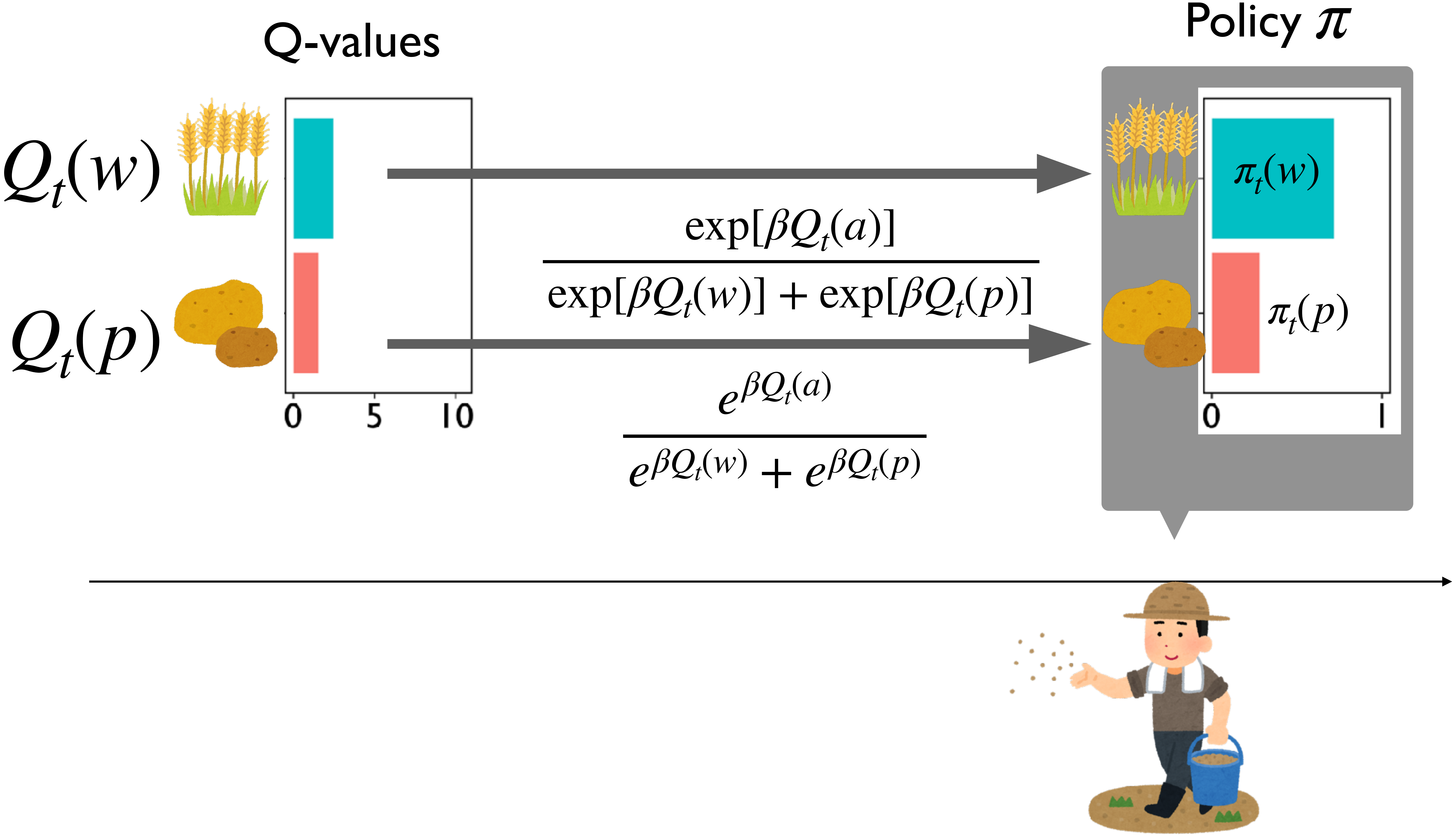



We want policy to satisfy:

$$\sum_a \pi(a) = 1$$

$$Q_1 > Q_2 \Leftrightarrow \pi_1 > \pi_2$$

# Converting value to actions:
# Softmax policy (i.e. multinomial logistic function)




$$\frac{\exp[\beta Q_t(a)]}{\exp[\beta Q_t(w)] + \exp[\beta Q_t(p)]}$$

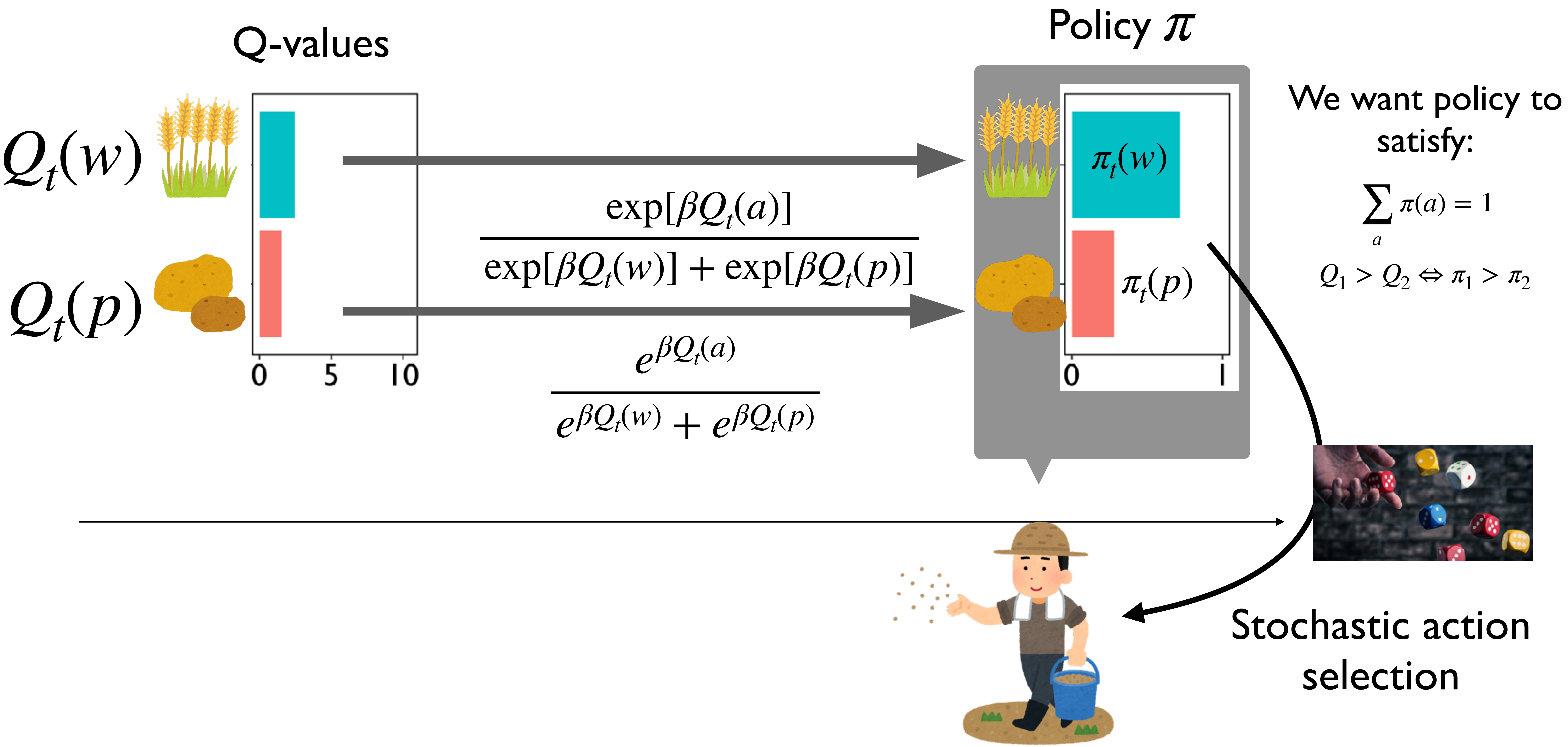
$$\frac{e^{\beta Q_t(a)}}{e^{\beta Q_t(w)} + e^{\beta Q_t(p)}}$$

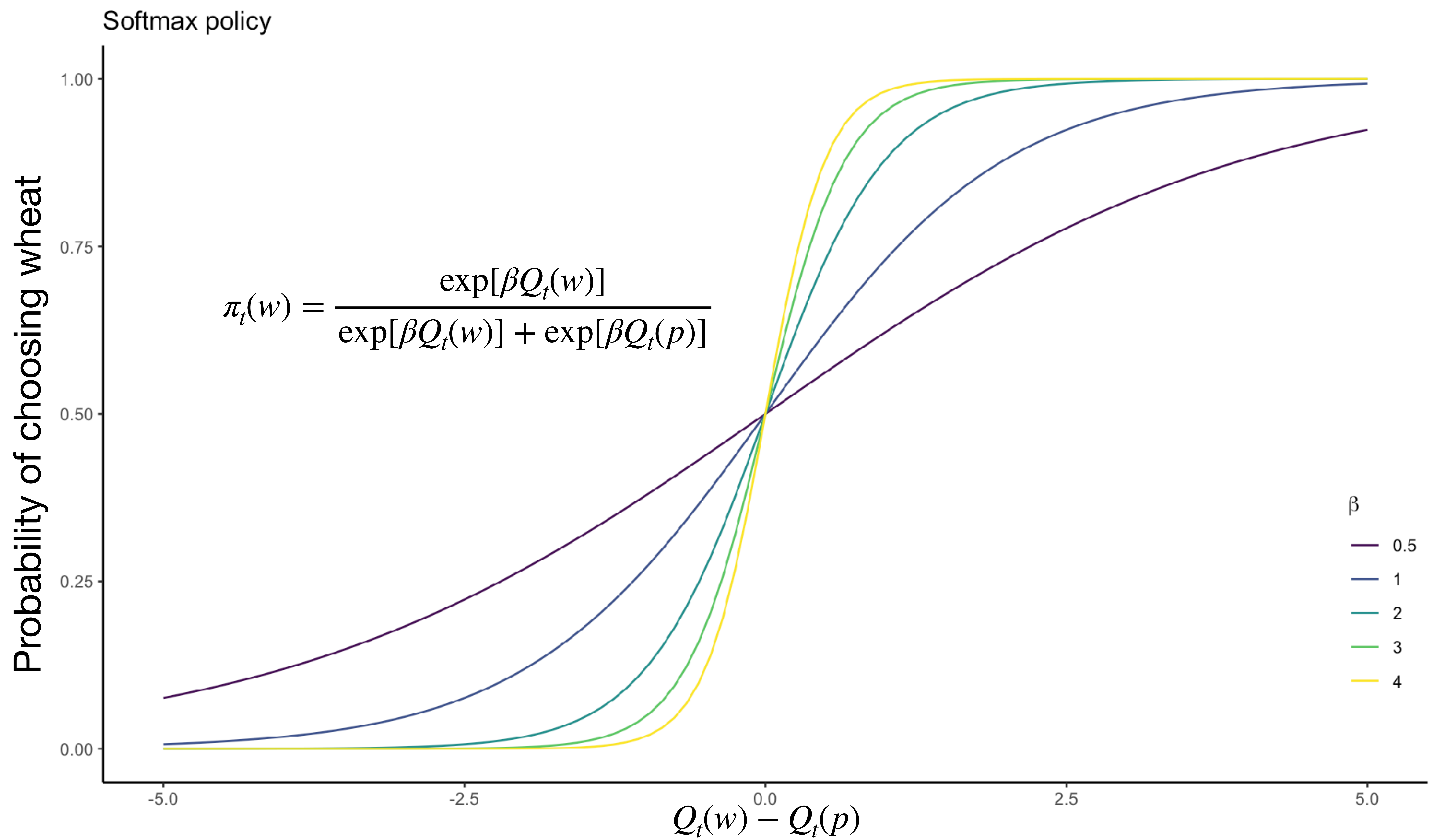
We want policy to satisfy:

$$\sum_a \pi(a) = 1$$

$$Q_1 > Q_2 \Leftrightarrow \pi_1 > \pi_2$$

# Converting value to actions:
# Softmax policy (i.e. multinomial logistic function)

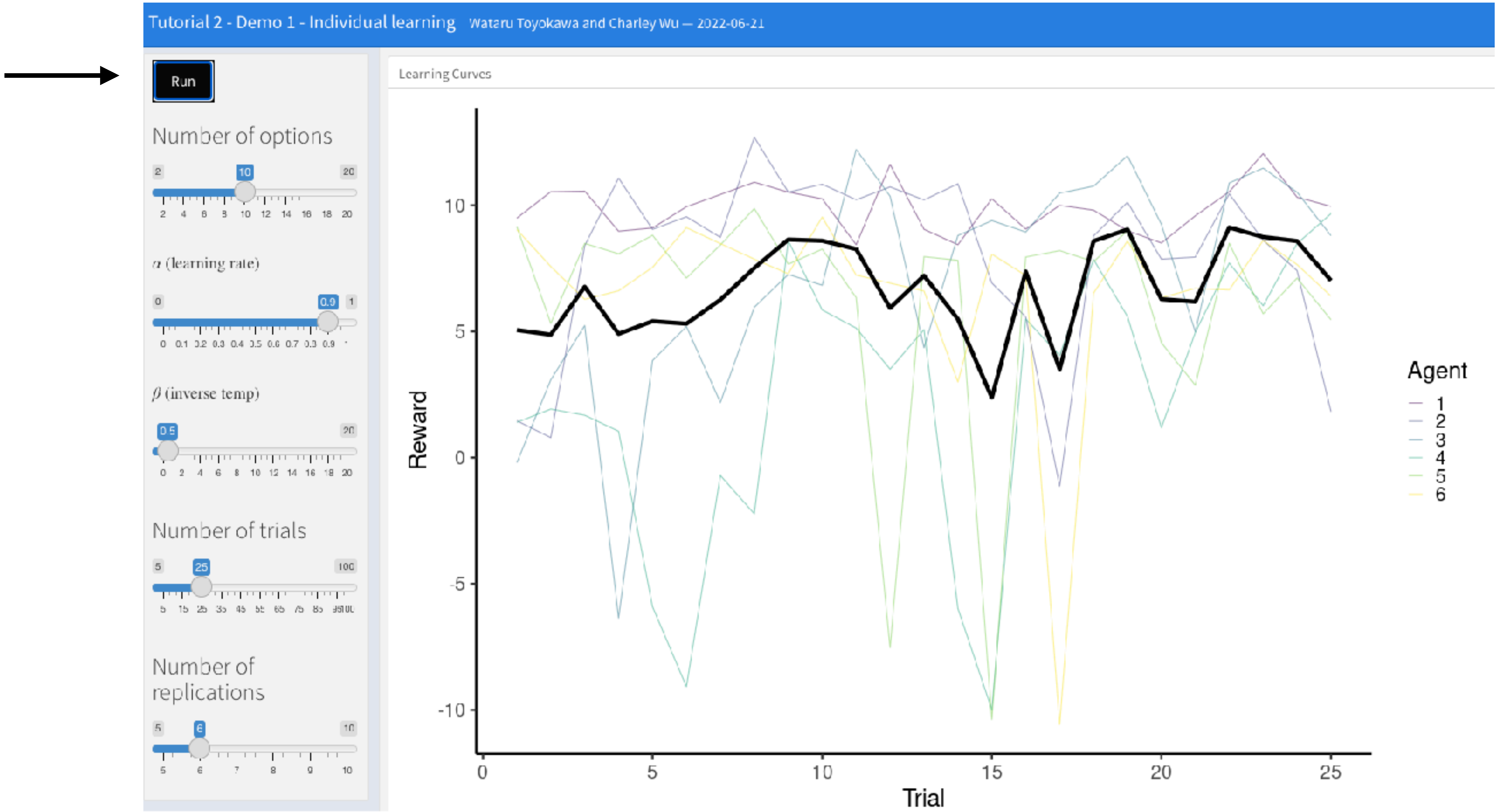



Softmax policy
Probability of choosing wheat
$\pi_t(w) = \frac{\exp[\beta Q_t(w)]}{\exp[\beta Q_t(w)] + \exp[\beta Q_t(p)]}$
1.00
0.75
0.50
0.25
0.00
-5.0
-2.5
0.0
2.5
5.0
$Q_t(w) - Q_t(p)$
β
0.5
1
2
3
4

# Notebook

https://cosmos-konstanz.github.io/notebooks/tutorial-2-models-of-learning.html#simulating-data

# Demo 1: Tweaking individual learning parameters

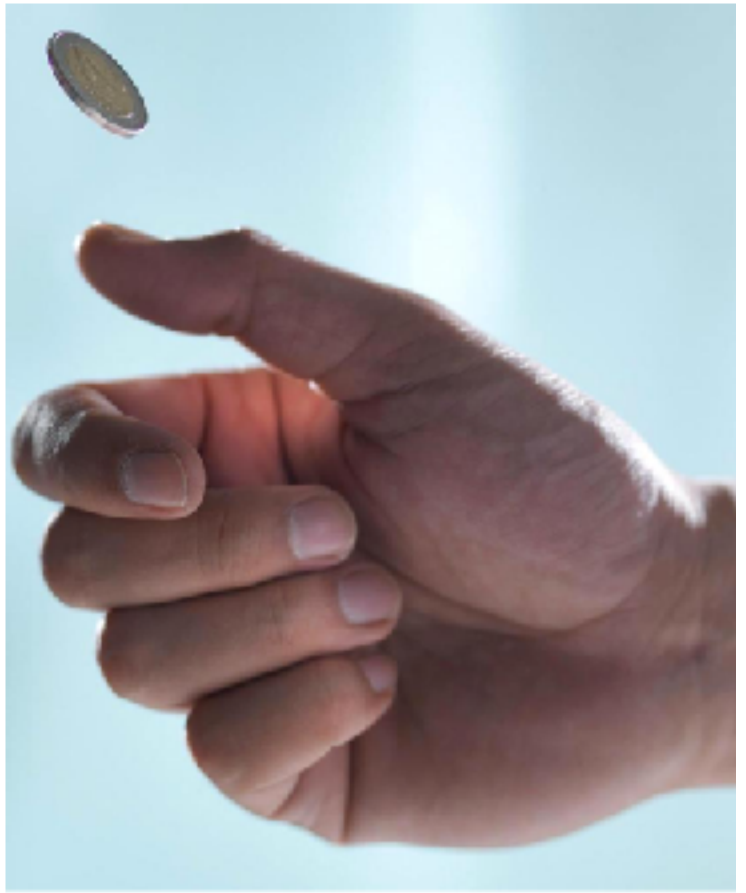



Which learning parameters ($\alpha, \beta$) typically produce the best results?

# Likelihood function

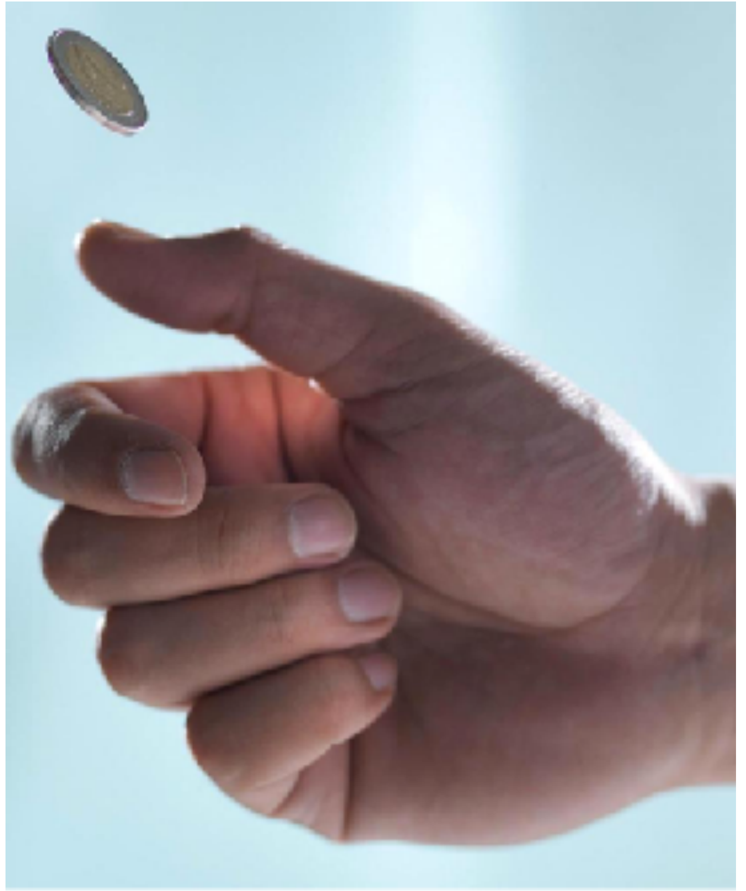
# Likelihood function

Coin Flip Model



# Likelihood function

Coin Flip Model



Observed Data:

$$D = \{d_1 = h, d_2 = t, \ldots d_n = t\}$$

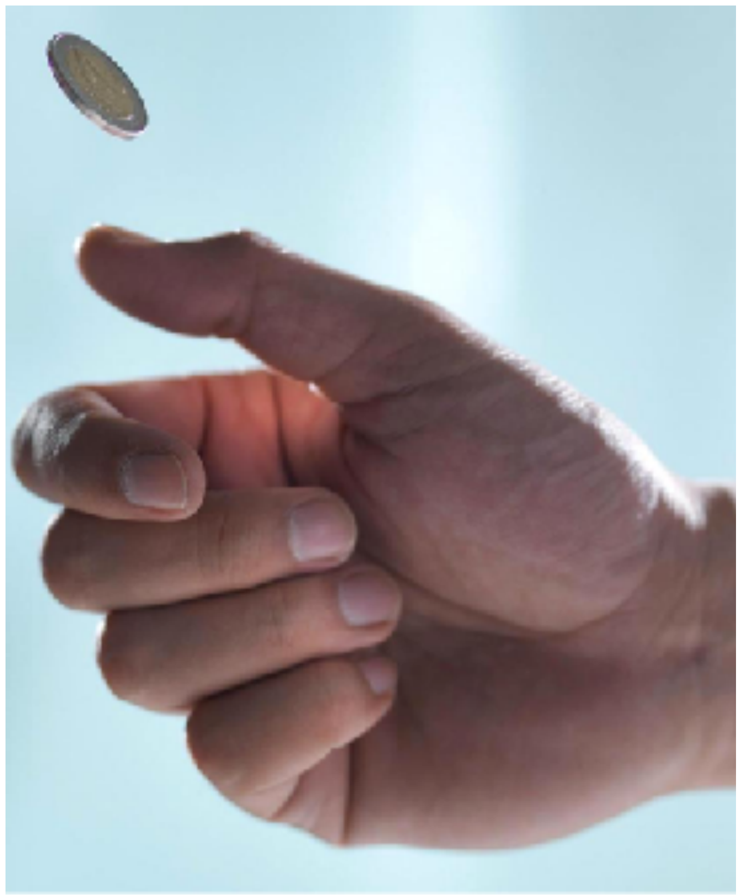
Model:

$$D \sim \textbf{Binomial}(n, \theta)$$

$$\theta = P(h)$$

# Likelihood function

Coin Flip Model



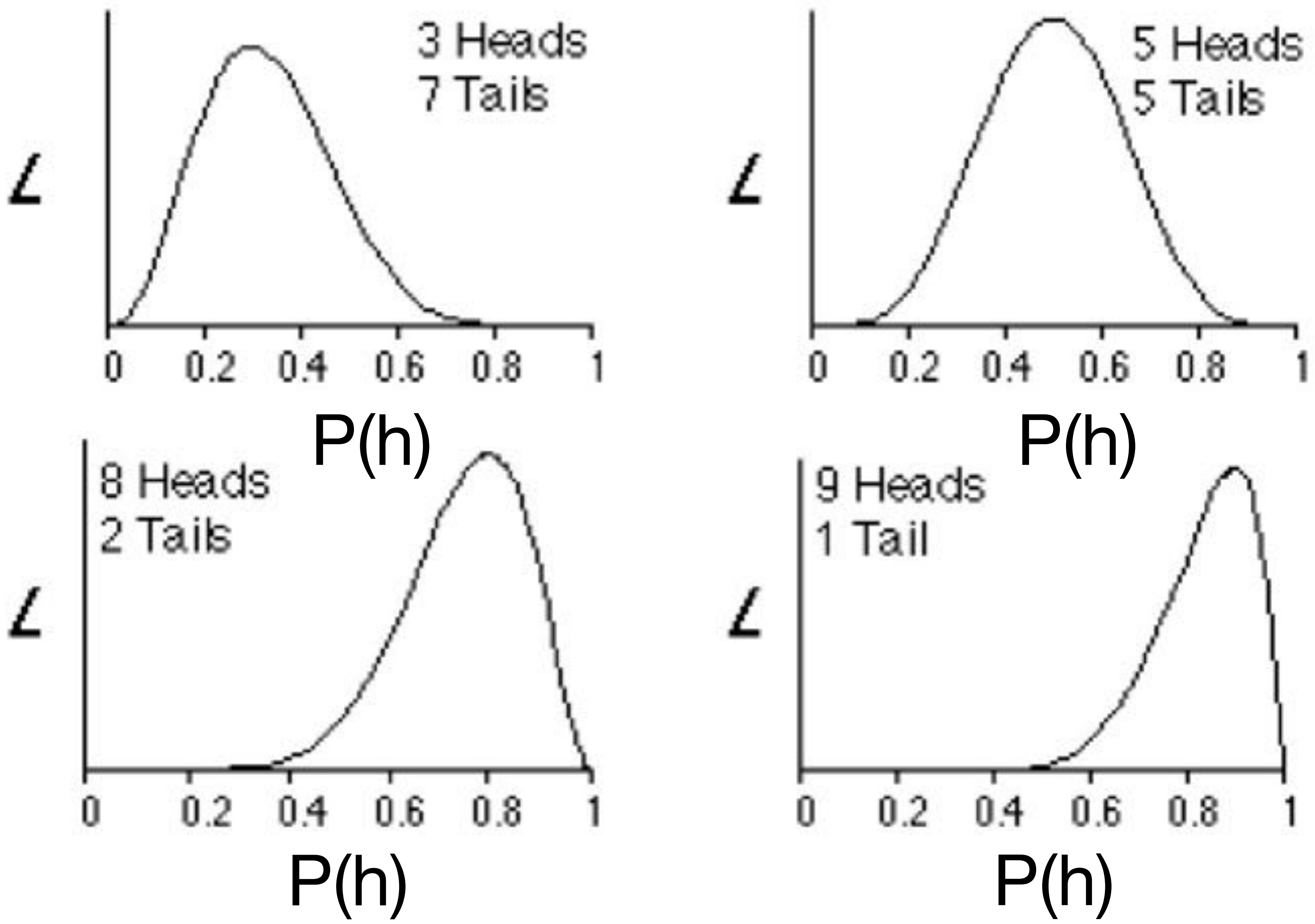
Observed Data:

$$D = \{d_1 = h, d_2 = t, \ldots d_n = t\}$$

Model:

$D \sim$ **Binomial**$(n, \theta)$

$$\theta = P(h)$$



# Likelihood function

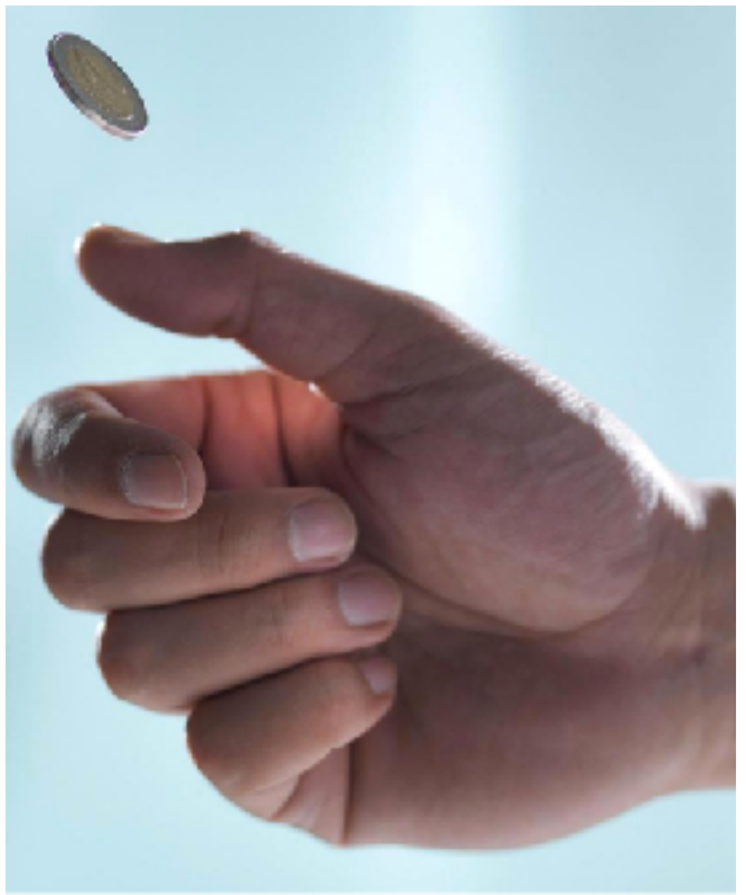
Beyond only simulating data, we also want to use models to describe experimental data.

To fit a model to data, we first need to define a **Likelihood Function:**

$$P(D \mid \theta)$$

describing the probability that the observed data $D$ was generated based on model parameters $\theta$

Coin Flip Model



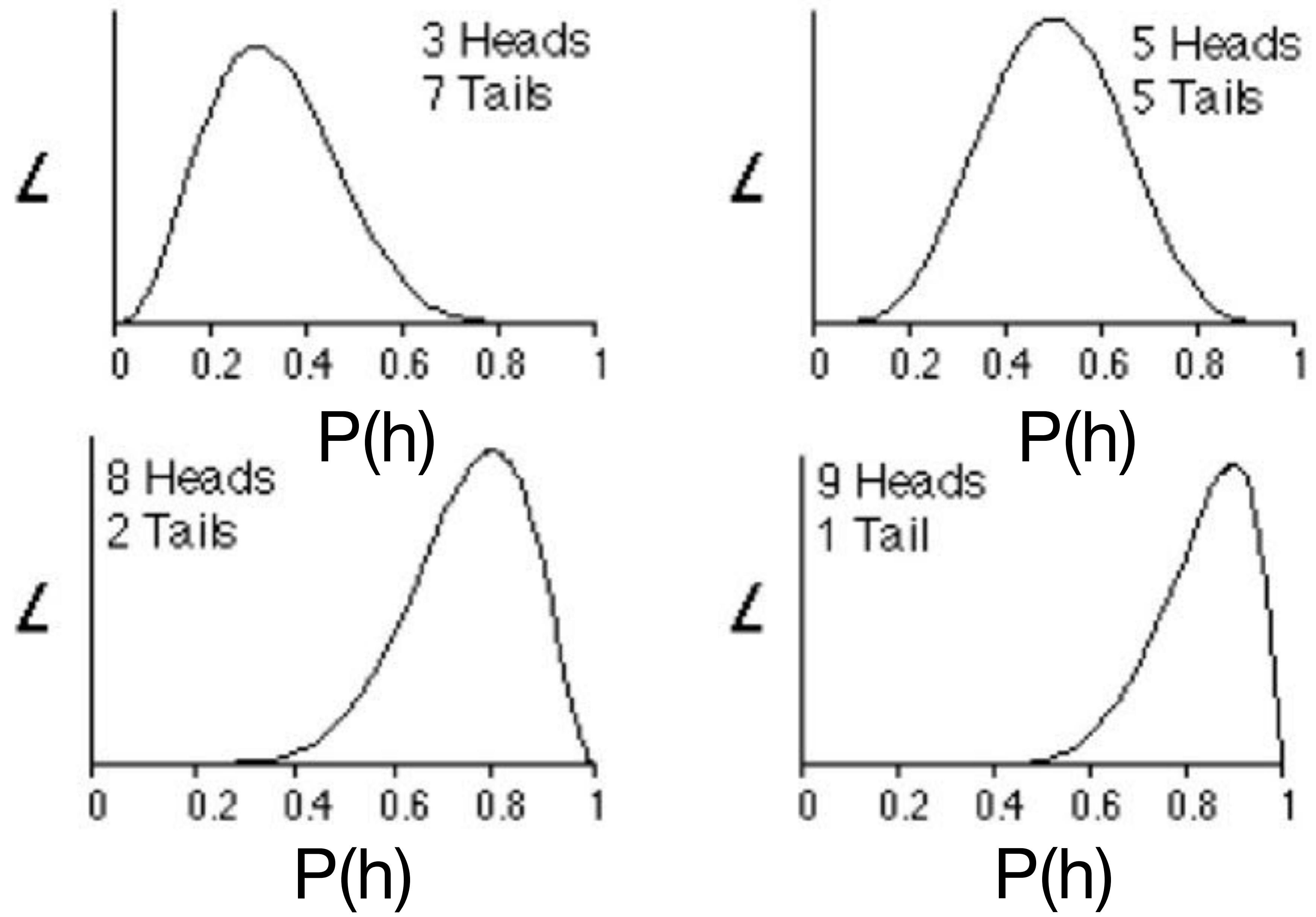
Observed Data:

$$D = \{d_1 = h, d_2 = t, \ldots d_n = t\}$$

Model:

$D \sim$ **Binomial**$(n, \theta)$

$$\theta = P(h)$$



*Note that natural logs are used by default (sometimes written as $\ln$) rather than base 10 logarithms $\log_{10}$

# Log Likelihoods

Since we are usually modeling multiple data points, we need to describe the **joint likelihood** over all observations:

$$P(D \mid \theta) = \prod_i P(d_i \mid \theta)$$

This is much easier using logarithms, since we can replace multiplication with summation in log space to compute the **log likelihood**

$$\log P(D \mid \theta) = \sum_t \log P(d_t \mid \theta)$$

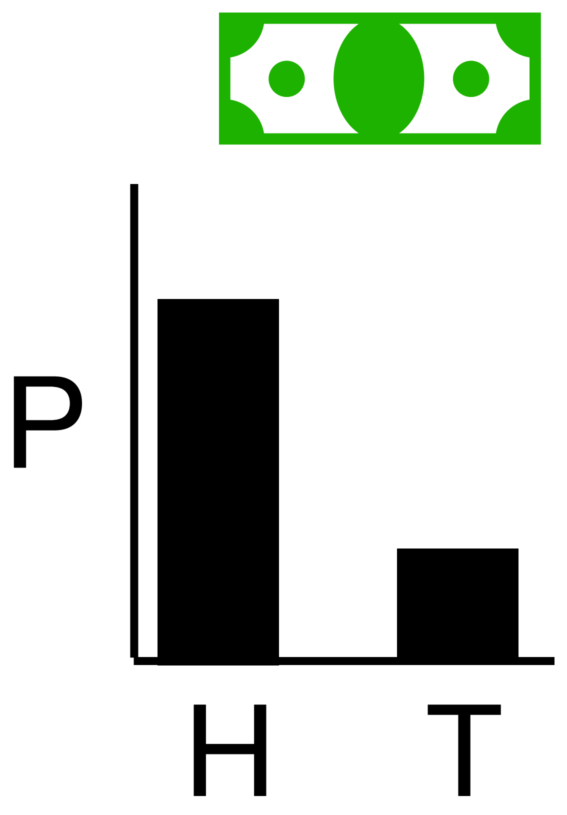
Since probabilities are always <1, the log likelihood will always be negative. Thus, it's more convenient to express the fit of a model using the **negative log likelihood** (nLL) by inverting the sign:

$$nLL = -\log P(D \mid \theta)$$

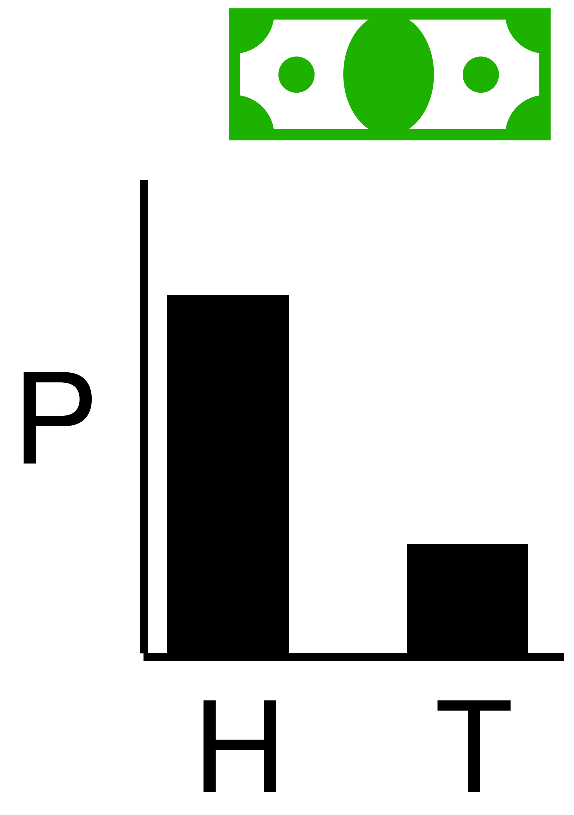
The nLL expresses the amount of error or loss (aka 'log loss') and will always be greater than zero. Smaller values thus describe better model fits.

# Likelihoods as Goodness of Fit

| Measure | Formula | Heads | Tails |
|---|---|---|---|
| Likelihood | P(D\|θ) | 80% | 20% |
| Log likelihood | log P(D\|θ) | –0.22 | –1.61 |
| Negative Log Likelihood (nLL) | – log P(D\|θ) | 0.22 | 1.61 |
| Deviance | –2 log P(D\|θ) | 0.44 | 3.22 |



# Likelihoods as Goodness of Fit

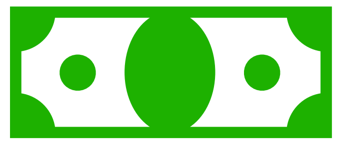


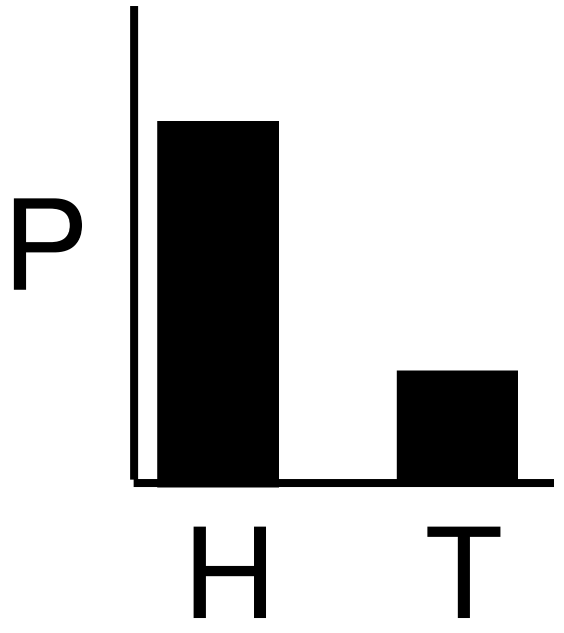

| Measure | Formula | Heads | Tails |
|---|---|---|---|
| Likelihood | P(D\|θ) | 80% | 20% |
| Log likelihood | log P(D\|θ) | –0.22 | –1.61 |
| Negative Log Likelihood (nLL) | – log P(D\|θ) | 0.22 | 1.61 |
| Deviance | –2 log P(D\|θ) | 0.44 | 3.22 |

**aka Log Loss**

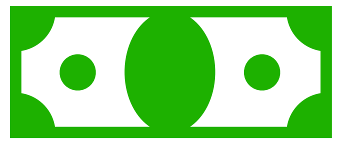
# Likelihoods as Goodness of Fit

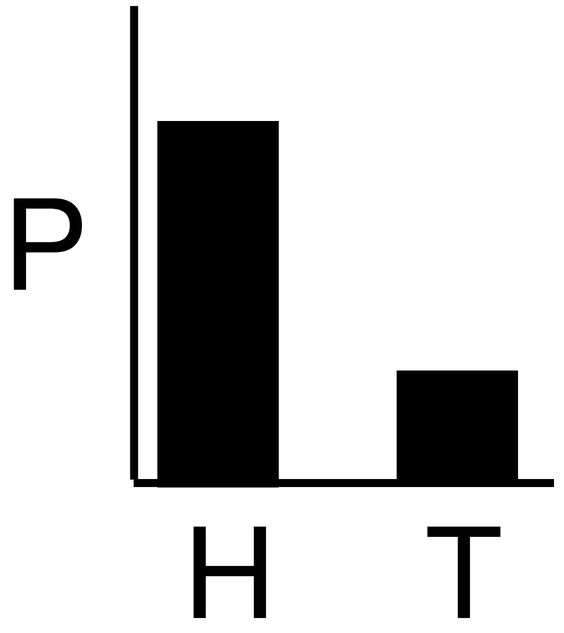
| | Measure | Formula | Heads | Tails |
|---|---|---|---|---|
| | Likelihood | $P(D\|\theta)$ | 80% | 20% |
| | Log likelihood | $\log P(D\|\theta)$ | –0.22 | –1.61 |
| **aka Log Loss** | Negative Log Likelihood (nLL) | $-\log P(D\|\theta)$ | 0.22 | 1.61 |
| | Deviance | $-2 \log P(D\|\theta)$ | 0.44 | 3.22 |






# Likelihoods as Goodness of Fit






| Measure | Formula | Heads | Tails |
|---|---|---|---|
| Likelihood | P(D\|θ) | 80% | 20% |
| Log likelihood | log P(D\|θ) | –0.22 | –1.61 |
| Negative Log Likelihood (nLL) | – log P(D\|θ) | 0.22 | 1.61 |
| Deviance | –2 log P(D\|θ) | 0.44 | 3.22 |

**aka Log Loss**

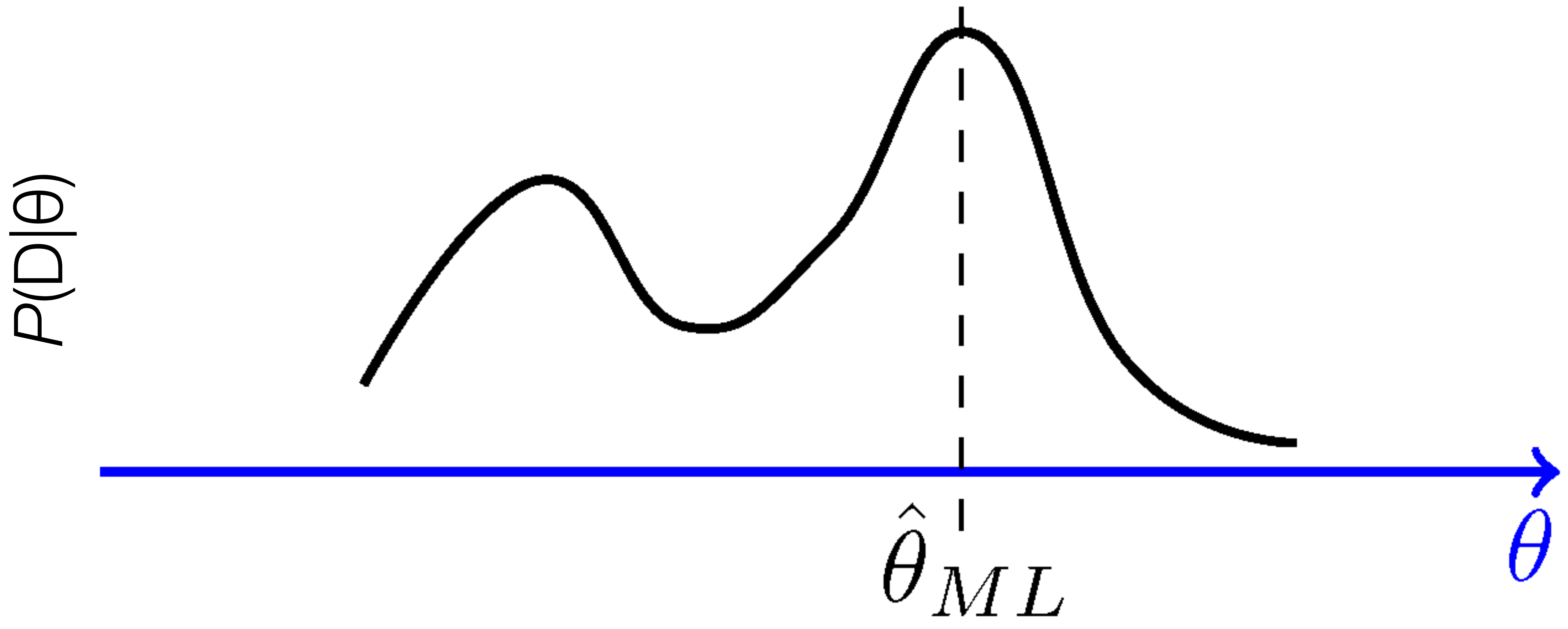
# From model simulation to likelihood functions

In practice, we can use code very simular to our model simulations to create a likelihood function

```
likelihood <- function(params, data){
   nLL <- 0 #initialize negative log likelihood
   for (d in data){ #loop through data
     predictions <- model(params) #make predictions
     observedAction <- d #define true outcome
     nLL <- nLL -log(predictions[observedAction]) #Update nLL
   }
   return(nLL)
}
```

# Maximum Likelihood Estimates (MLE)

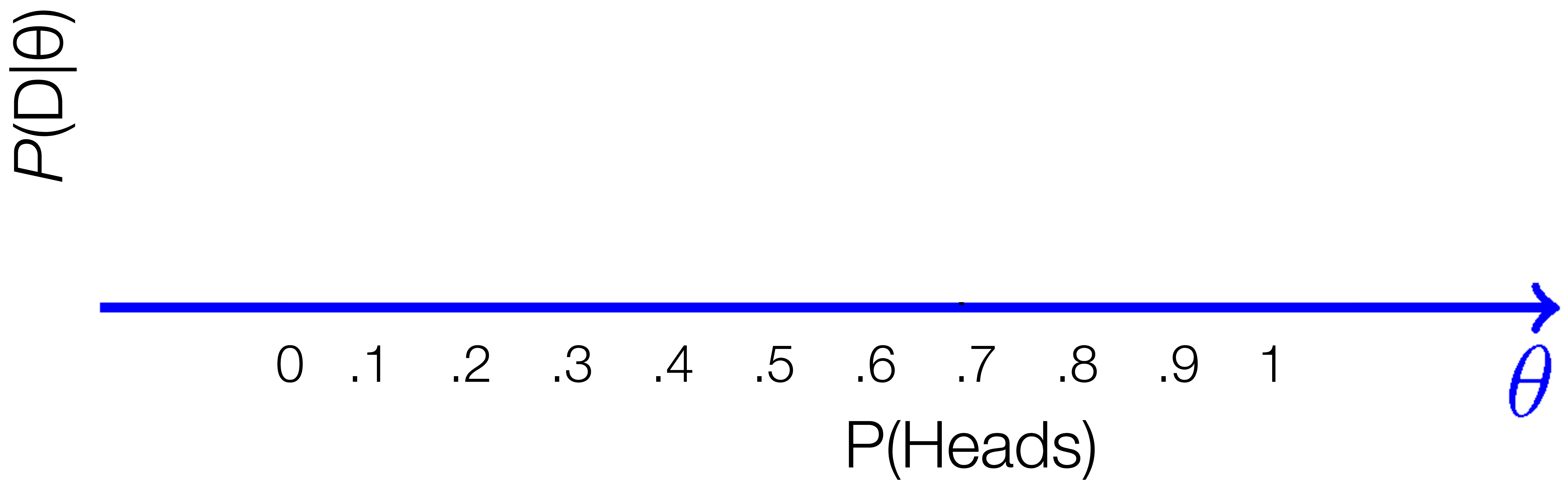
Use the likelihood function to find the parameters $\hat{\theta}$ where $P(D\,|\,\theta)$ is largest



# Maximum Likelihood Estimates (MLE)

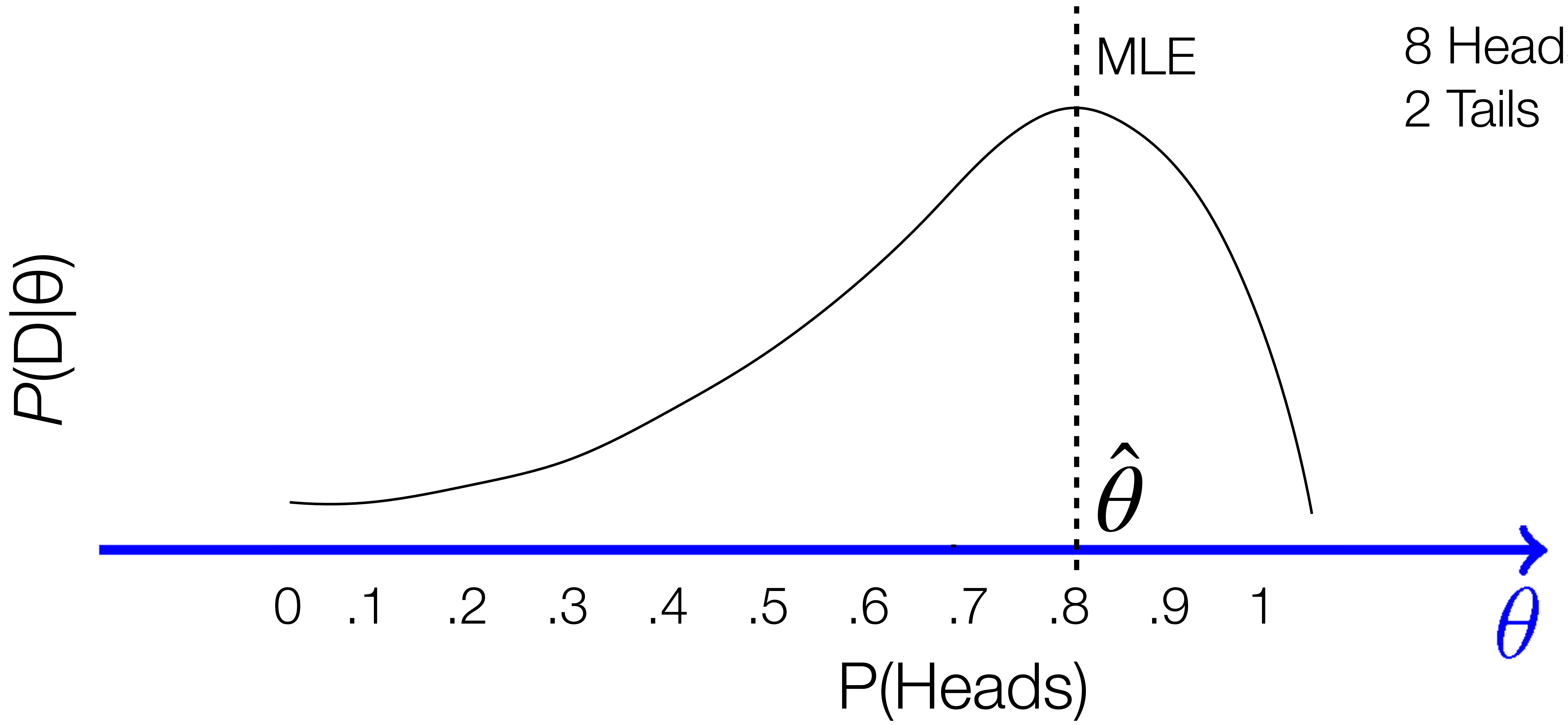
Use the likelihood function to find the parameters $\hat{\theta}$ where $P(D\,|\,\theta)$ is largest

8 Heads
2 Tails



# Maximum Likelihood Estimates (MLE)

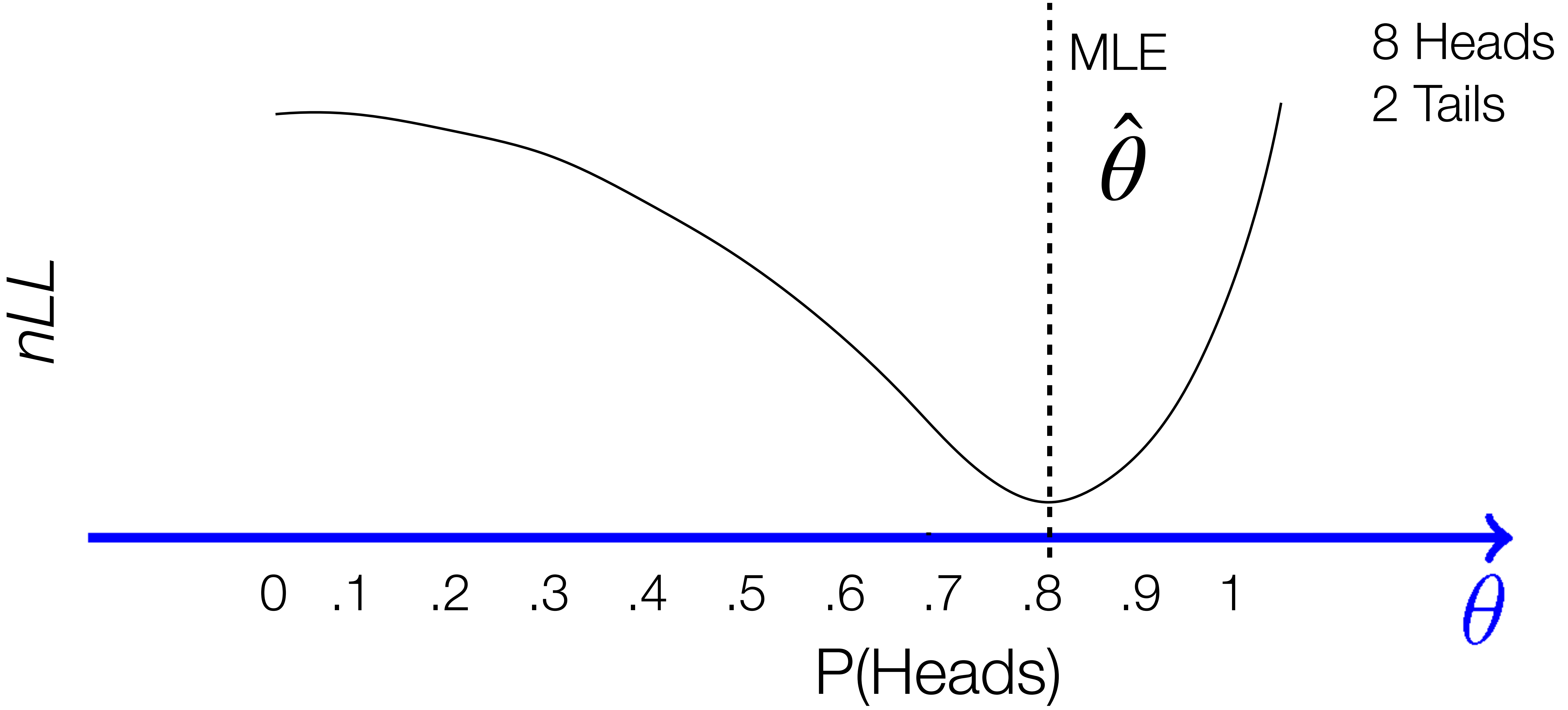
Use the likelihood function to find the parameters $\hat{\theta}$ where $P(D \mid \theta)$ is largest



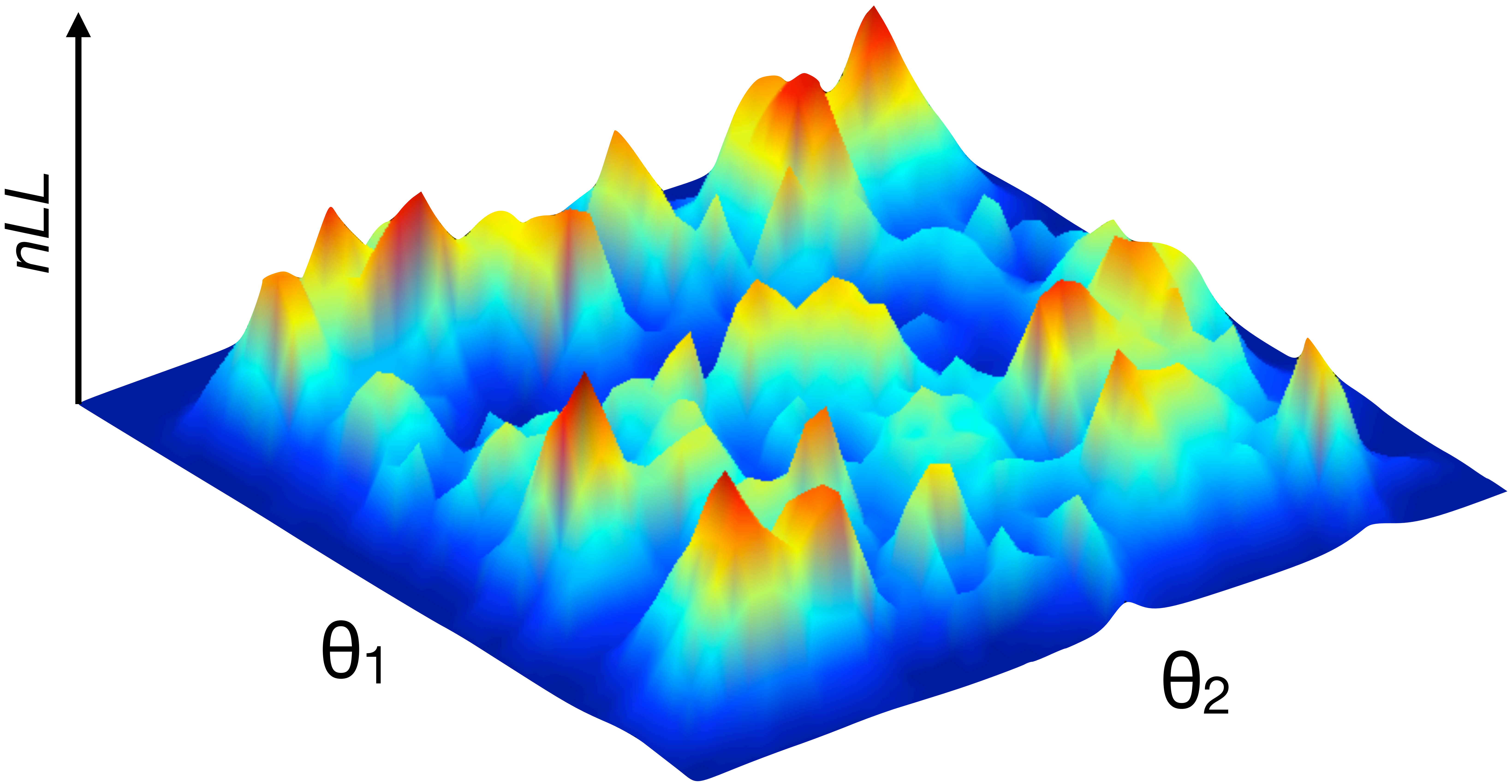
# Maximum Likelihood Estimates (MLE)

Use the likelihood function to find the parameters $\hat{\theta}$ where $P(D \mid \theta)$ is largest

…. or where nLL is lowest



# Computing the MLE




Optimization function

```
likelihood <-
function(params
, data)
```

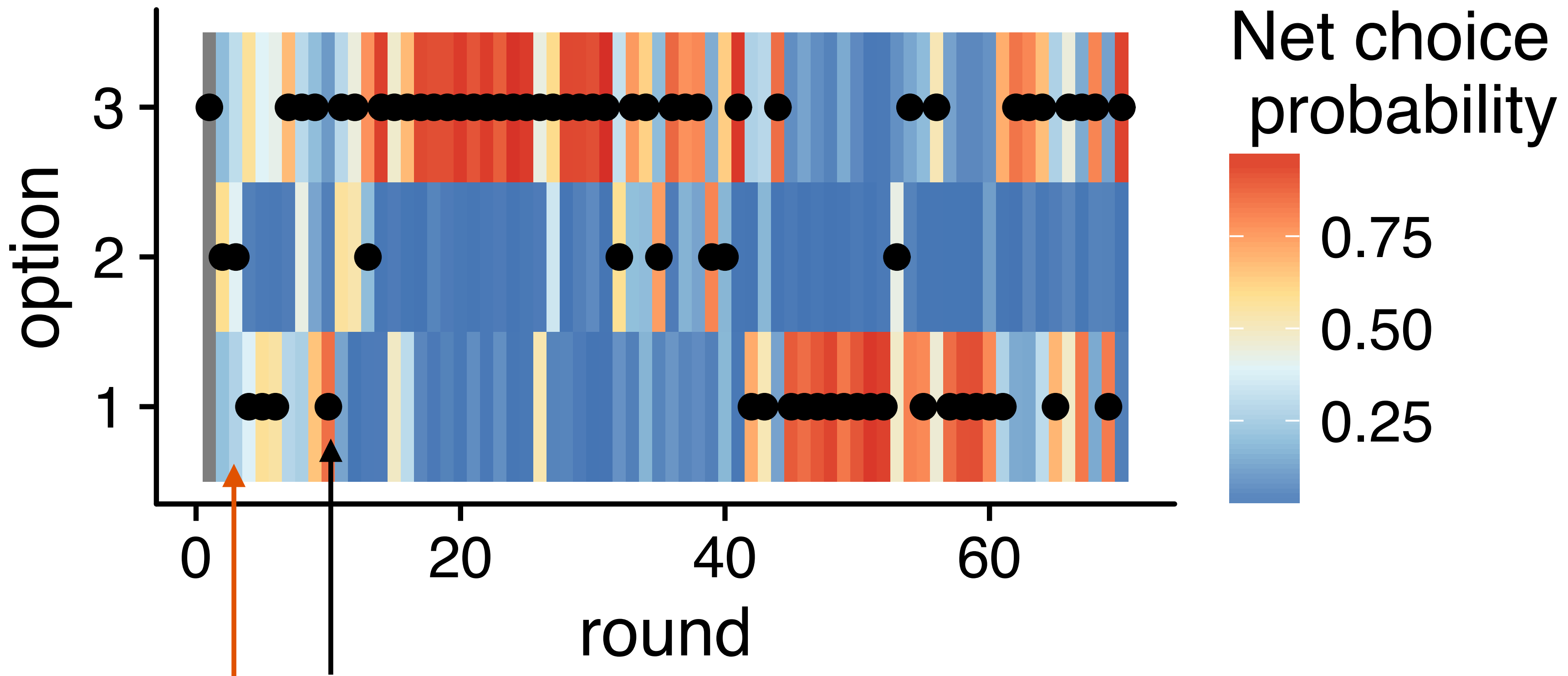
Minimize nLL

Types of optimization algorithms

- Gradient descent
- Simplex methods
- Differential evolution

# MLE for a RL model

Simulated data

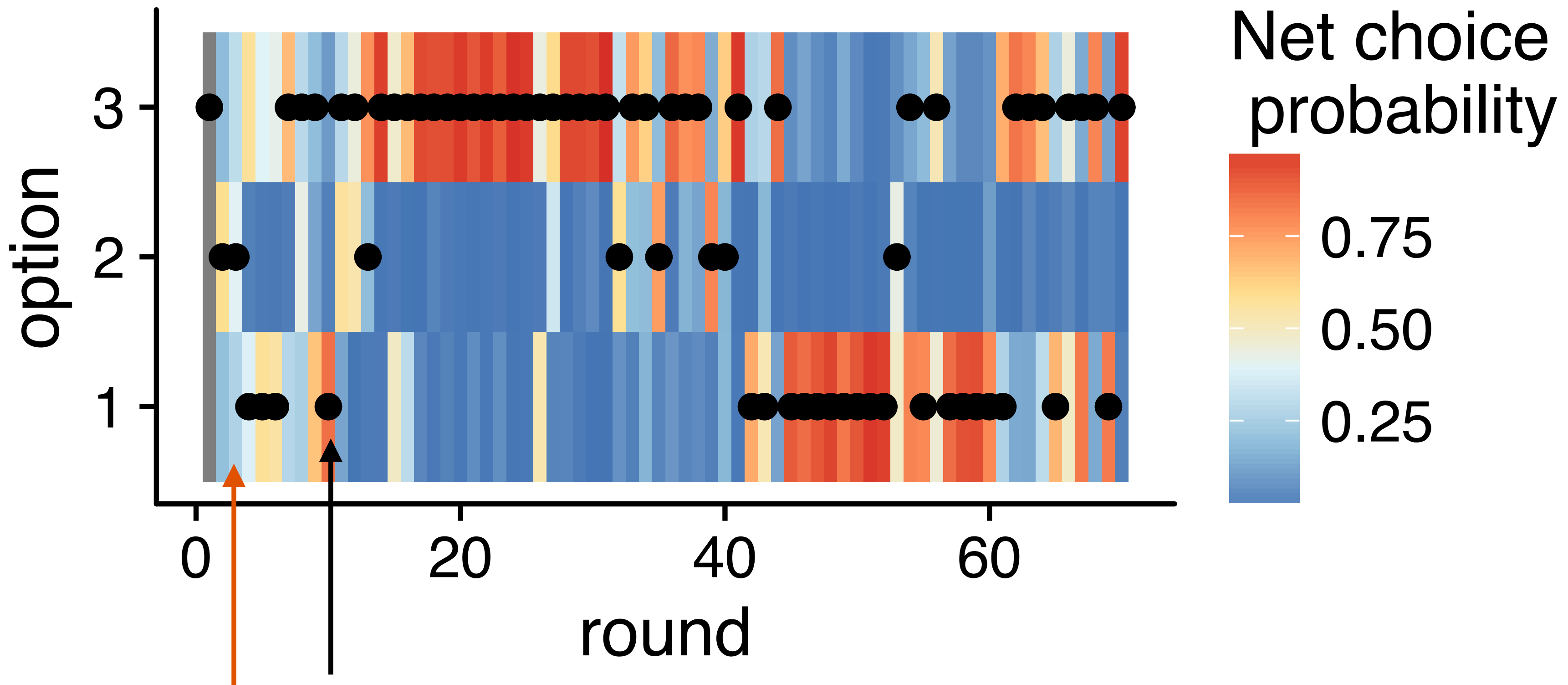



Dots: Choice (observable)

Background:
Choice probability
(unobservable latent state)

# MLE for a RL model

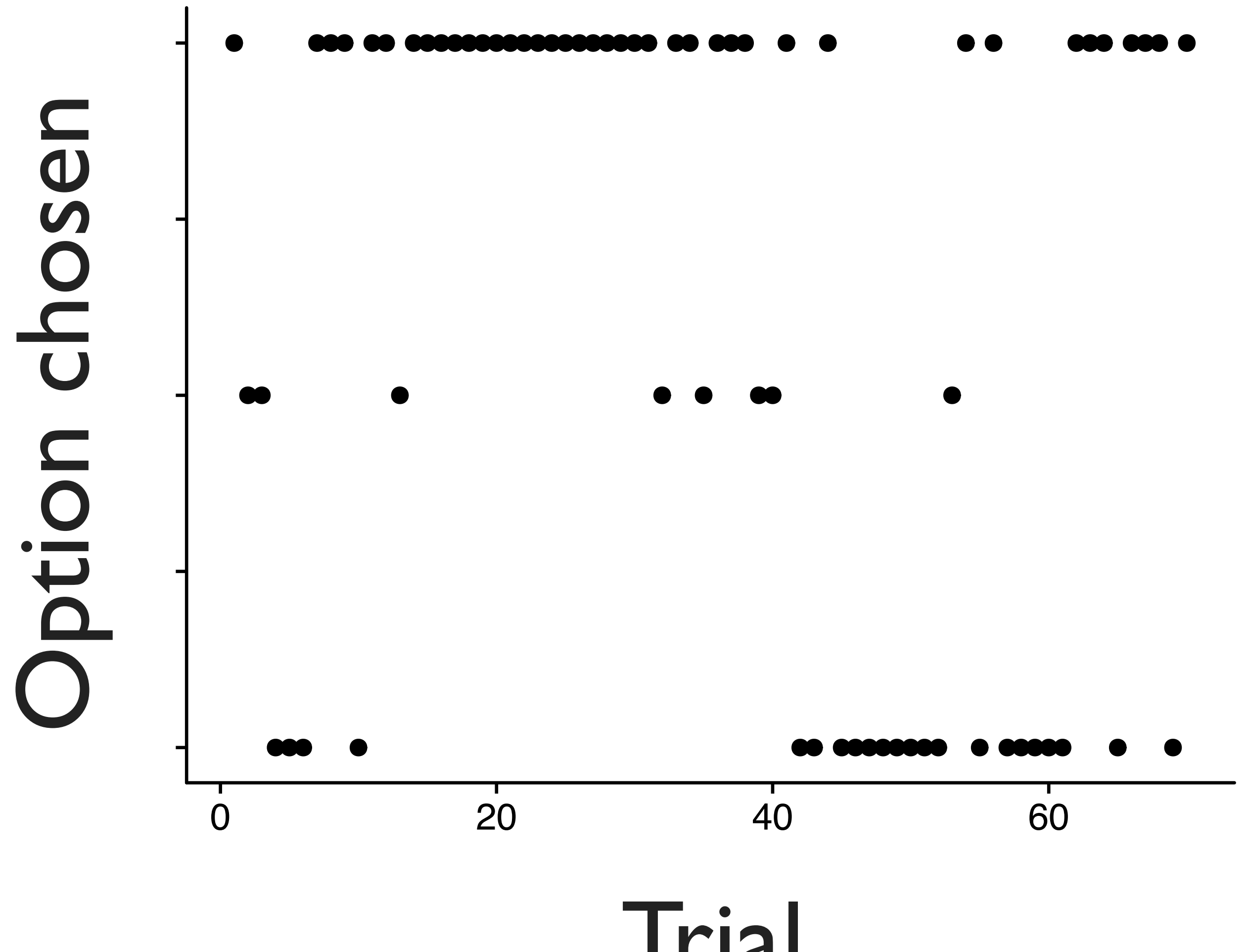
Simulated data




Dots: Choice (observable)

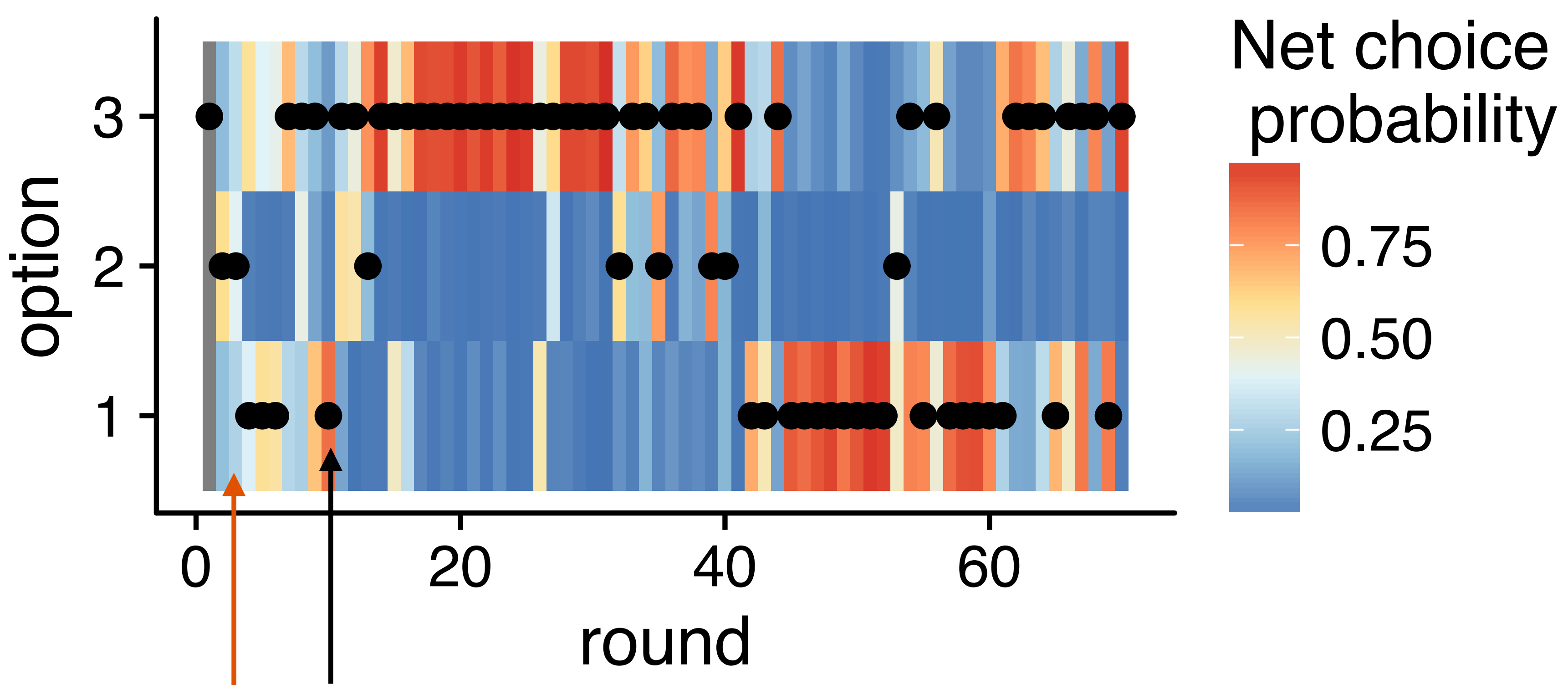
Background:
Choice probability
(unobservable latent state)

Choice data from which we can calculate likelihood of the RL model



# MLE for a RL model

Simulated data

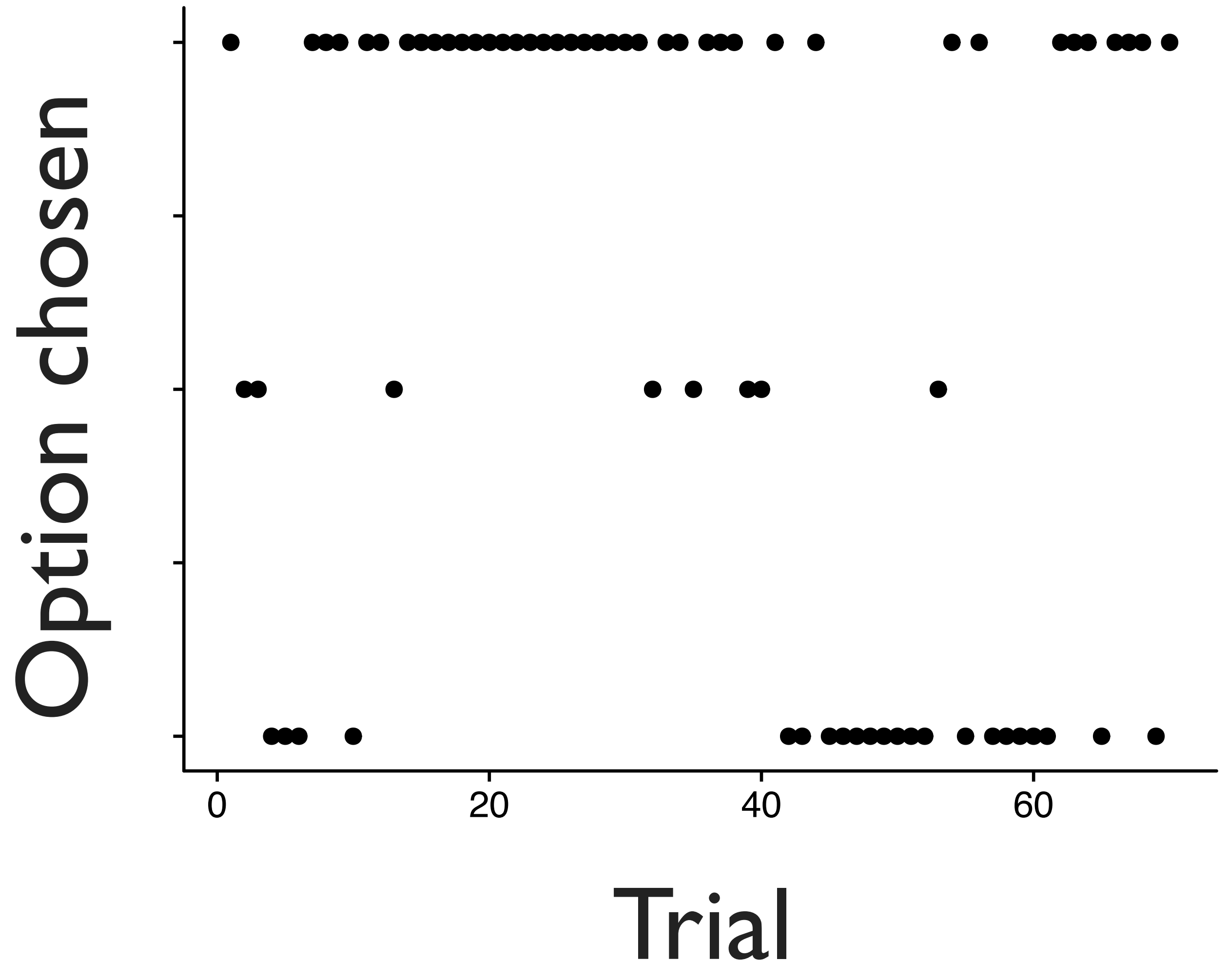



Dots: Choice (observable)

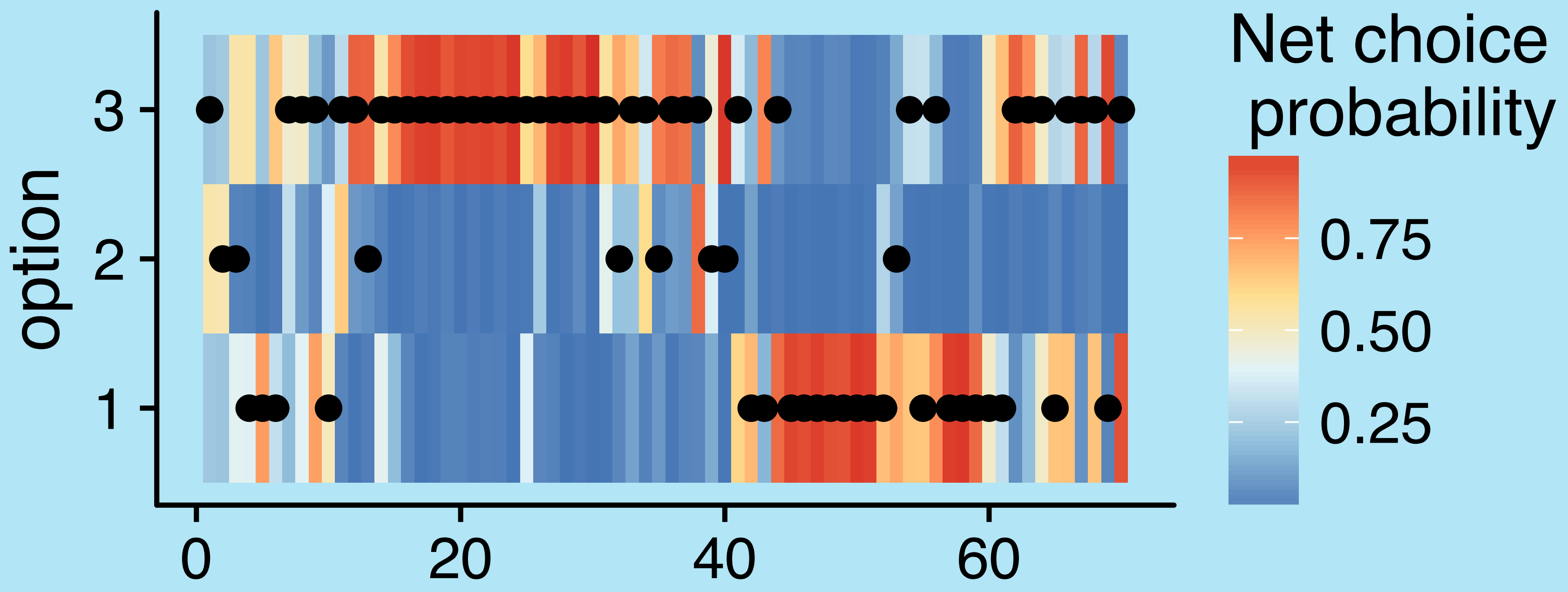
Background:
Choice probability
(unobservable latent state)

Search for parameters that maximise likelihood

Choice data from which we can calculate likelihood of the RL model

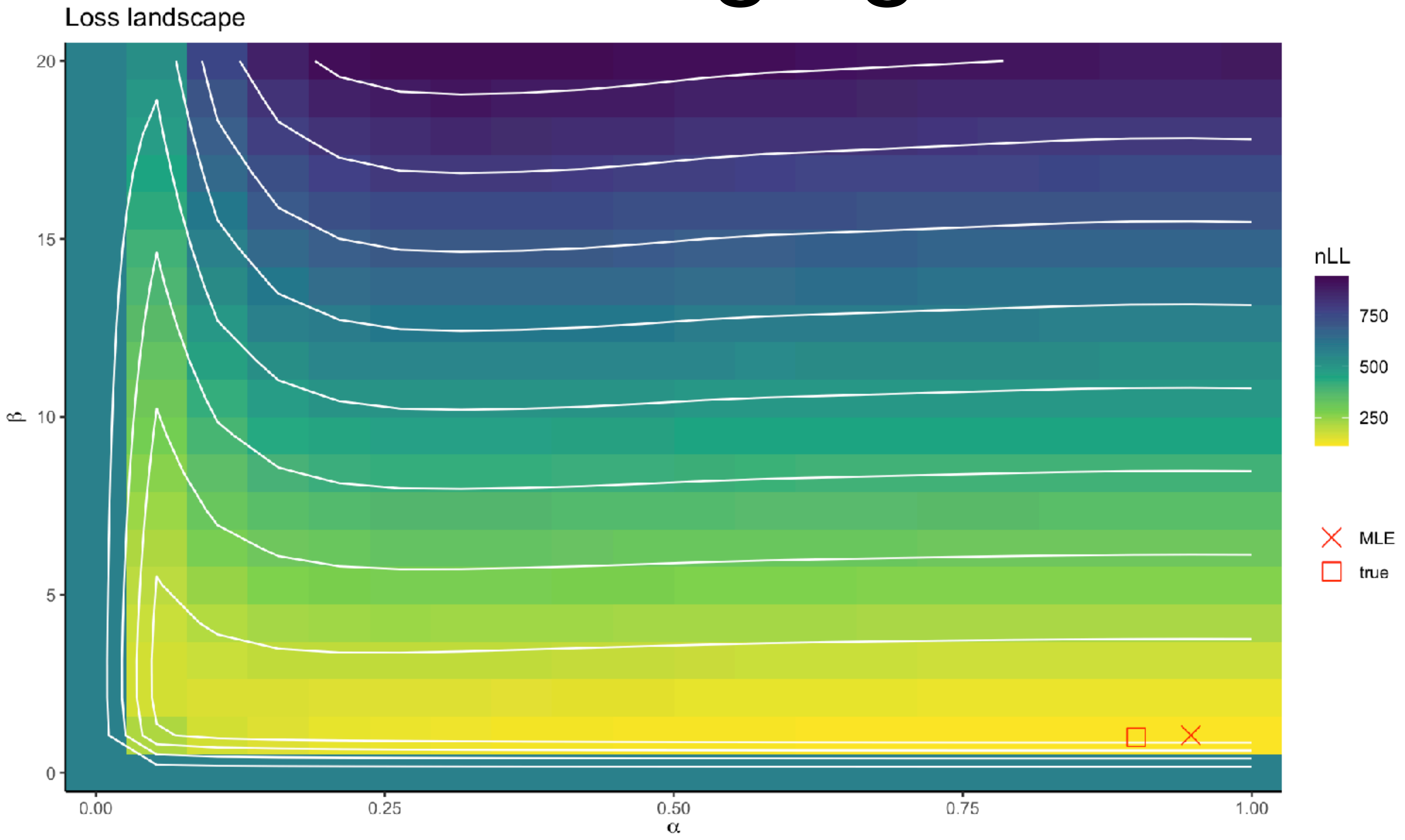



Estimated choice probability



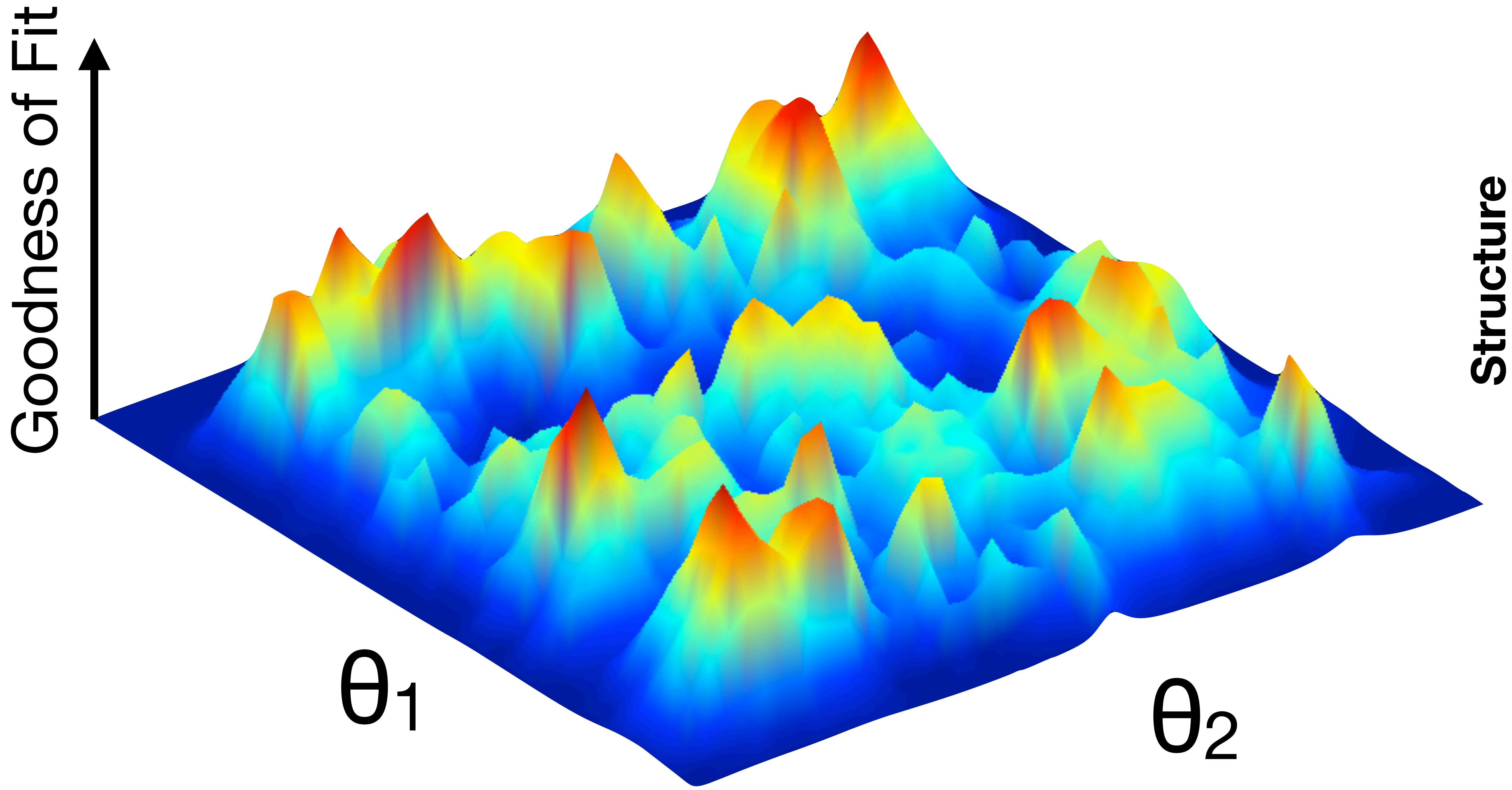

Maximum Likelihood Estimate (MLE)

# Q-learning agent

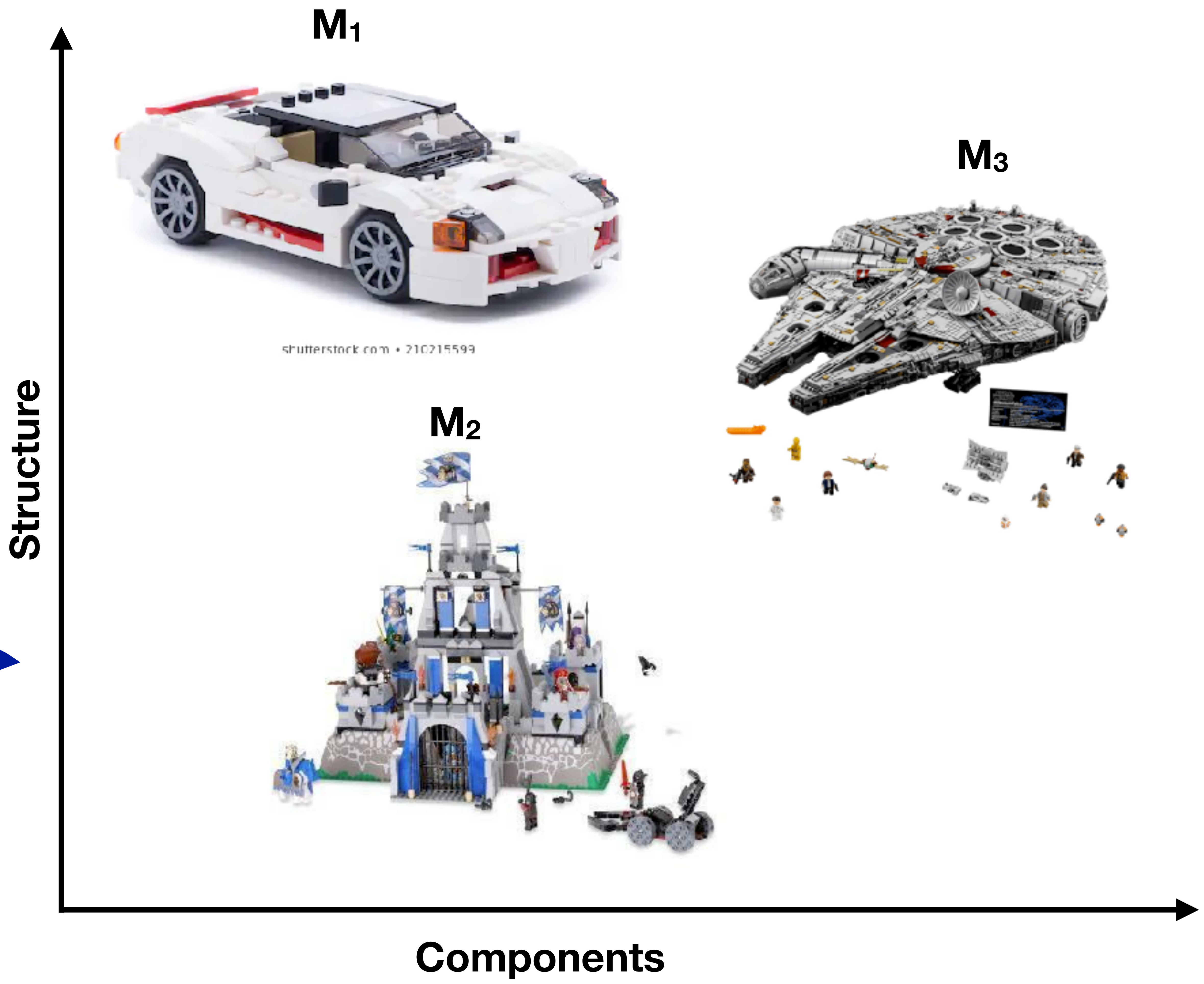


# Parameter Space and Model Space

**Parameter Space**



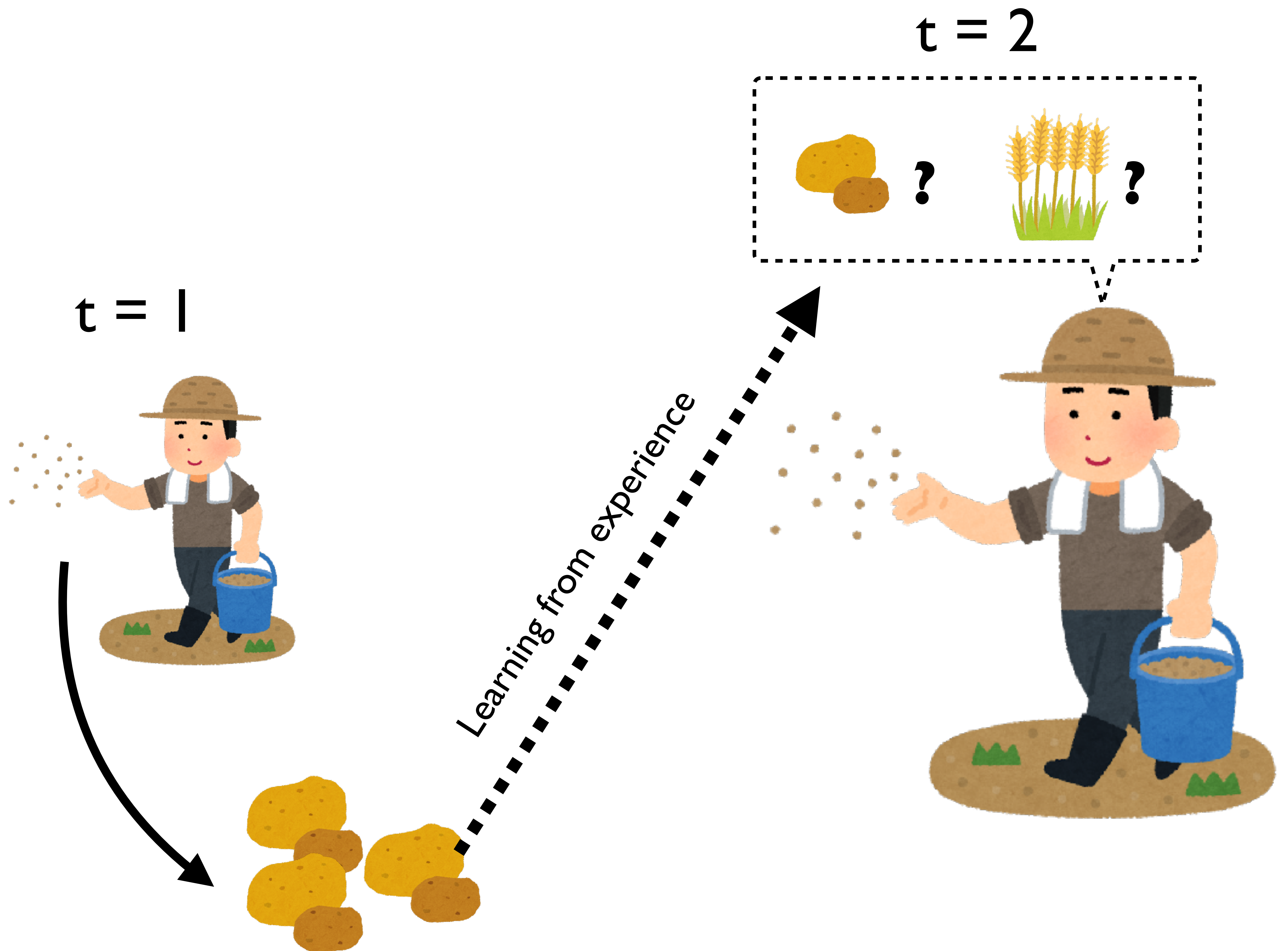

**Model Space**



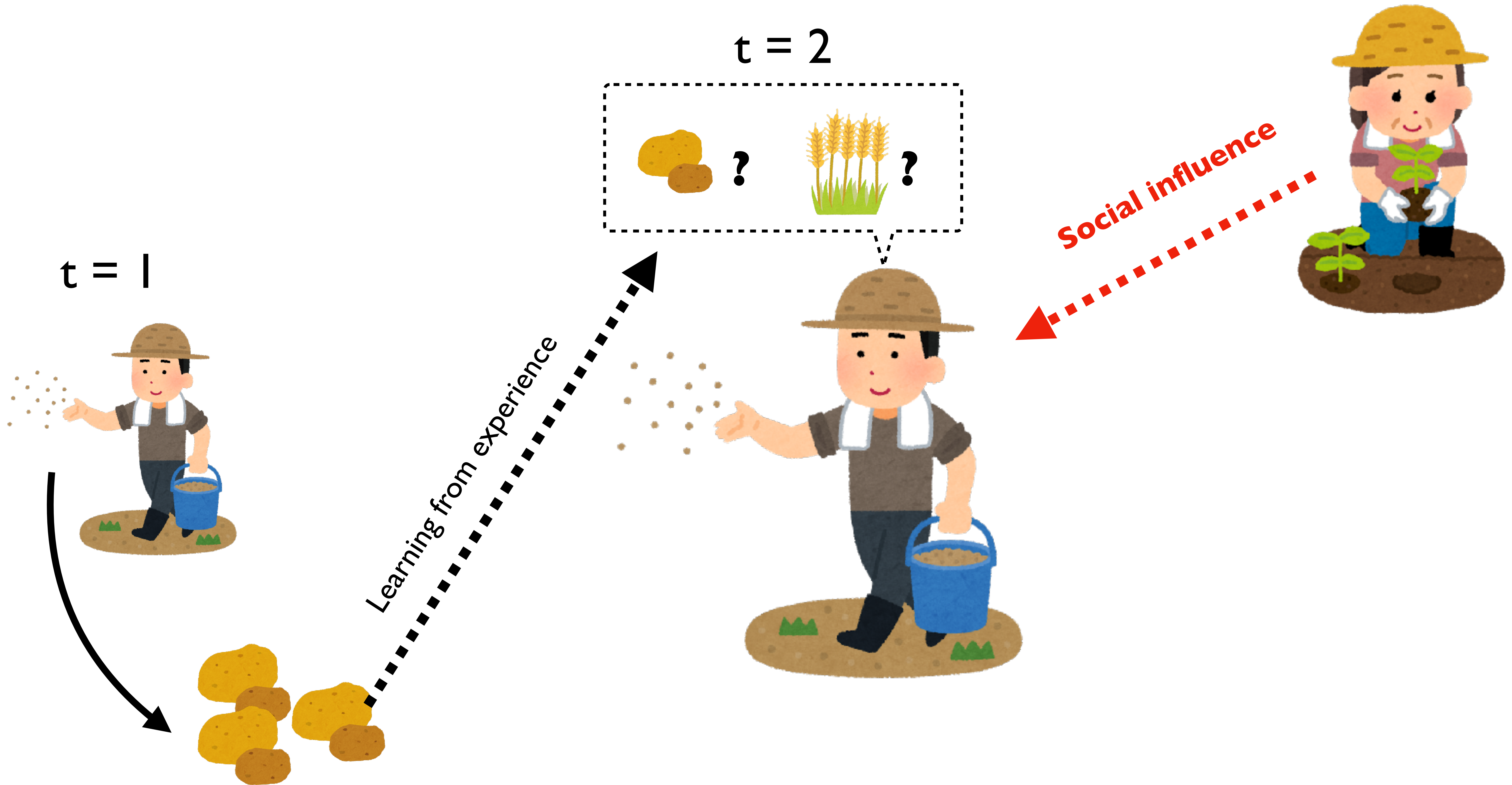
# Learning from social information

(5 minute break)

# Learning from social information



# Learning from social information



# Imitating actions

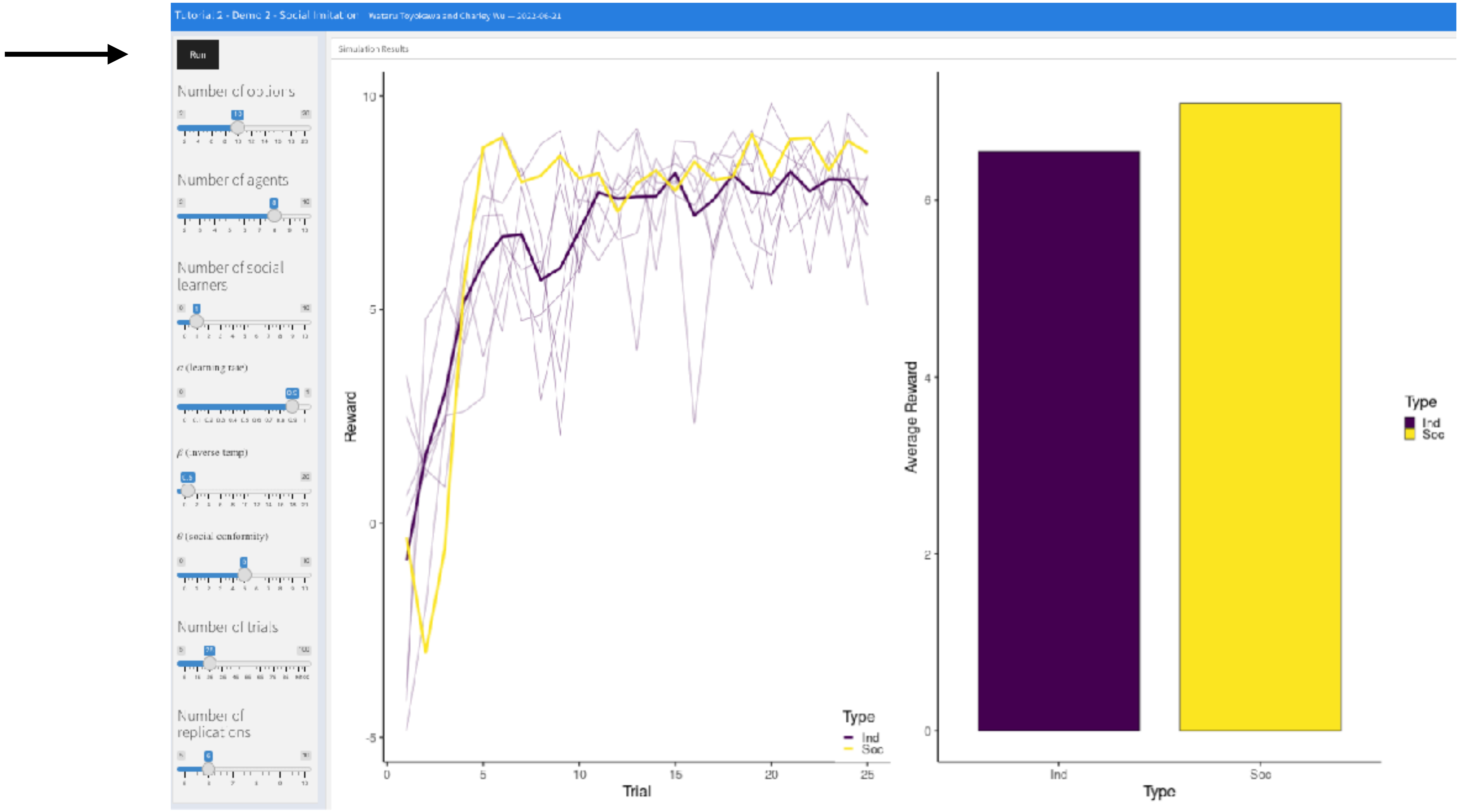
Frequency-dependent copying

Probability of choosing option a $\propto$ frequency of other agents performing the same action

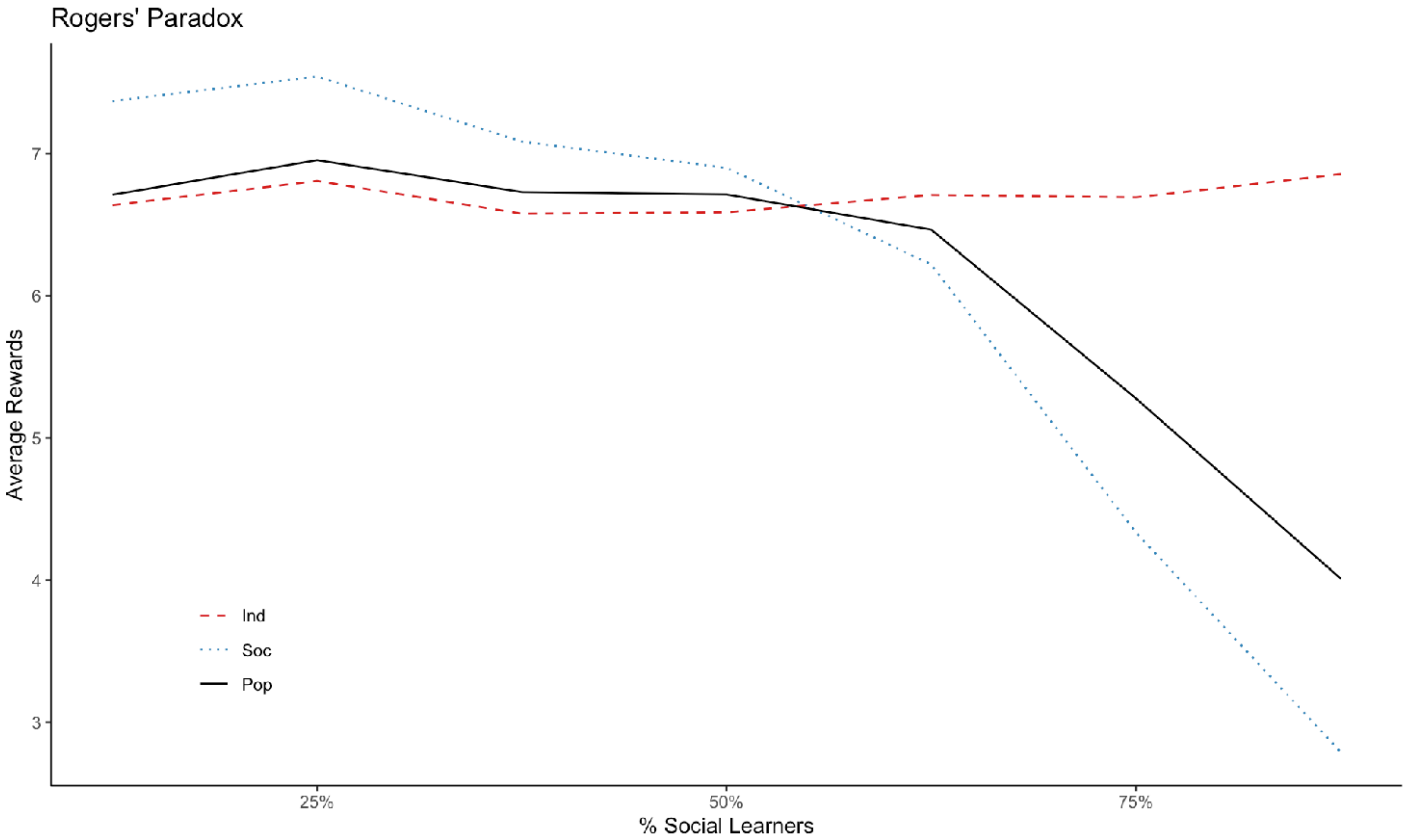
$$\pi_{\text{FDC}}(a) \propto \frac{f(a)^\theta}{\sum f(k)^\theta}$$

Notebook https://cosmos-konstanz.github.io/notebooks/tutorial-2-models-of-learning.html#imitating-actions

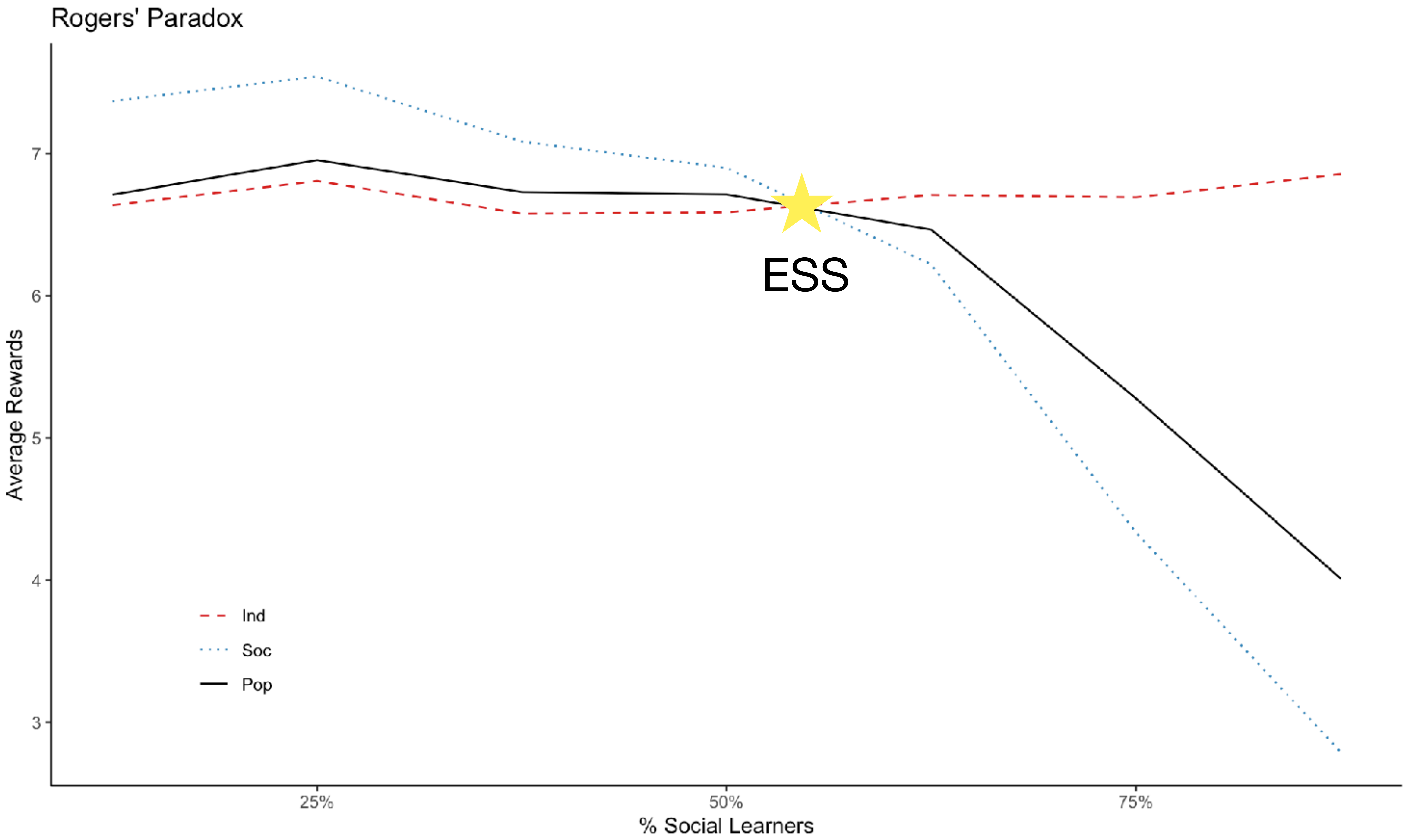
# Demo 2: Imitation and Rogers' paradox

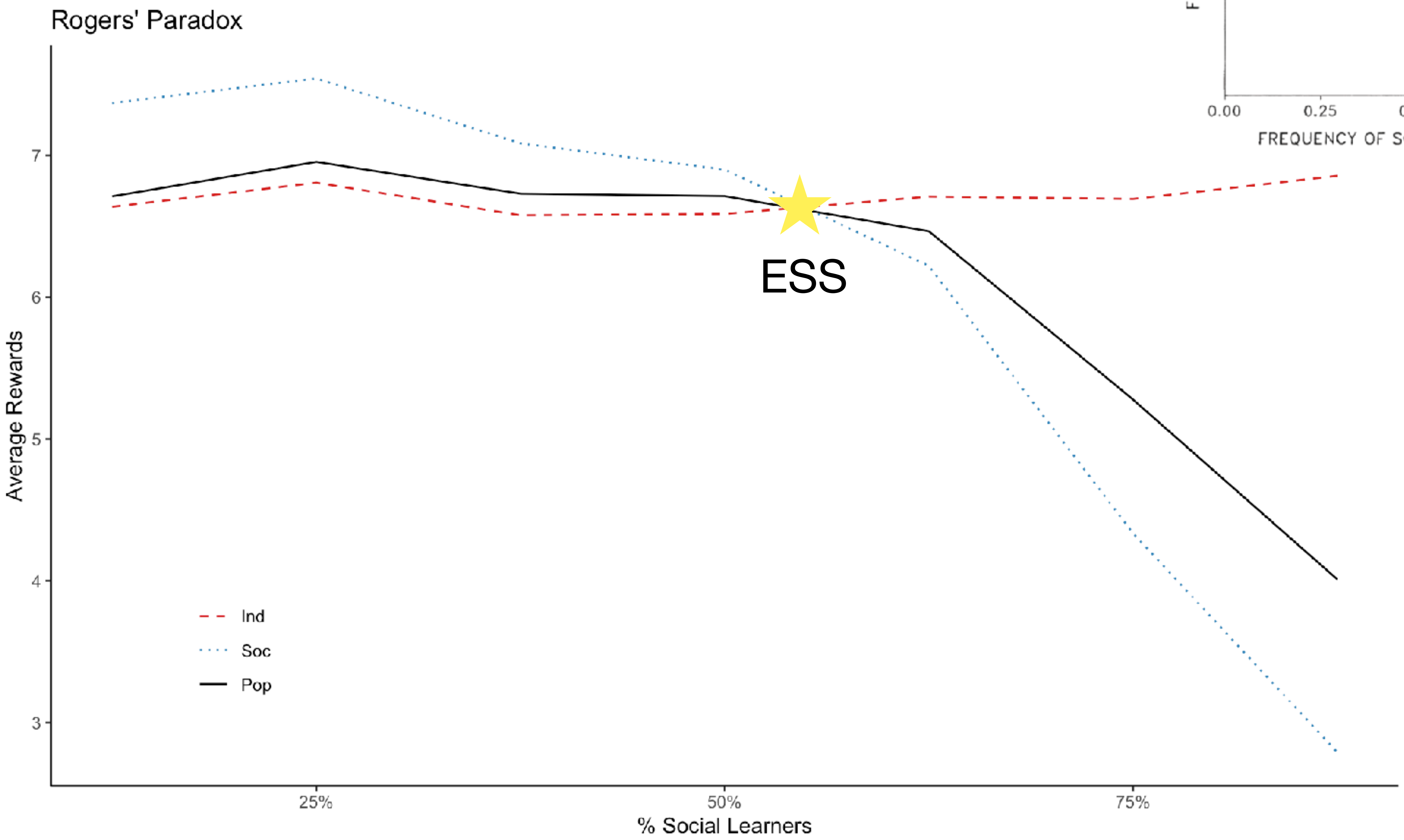



How do different ratios of individual vs. social learners change the performance of each agent type?

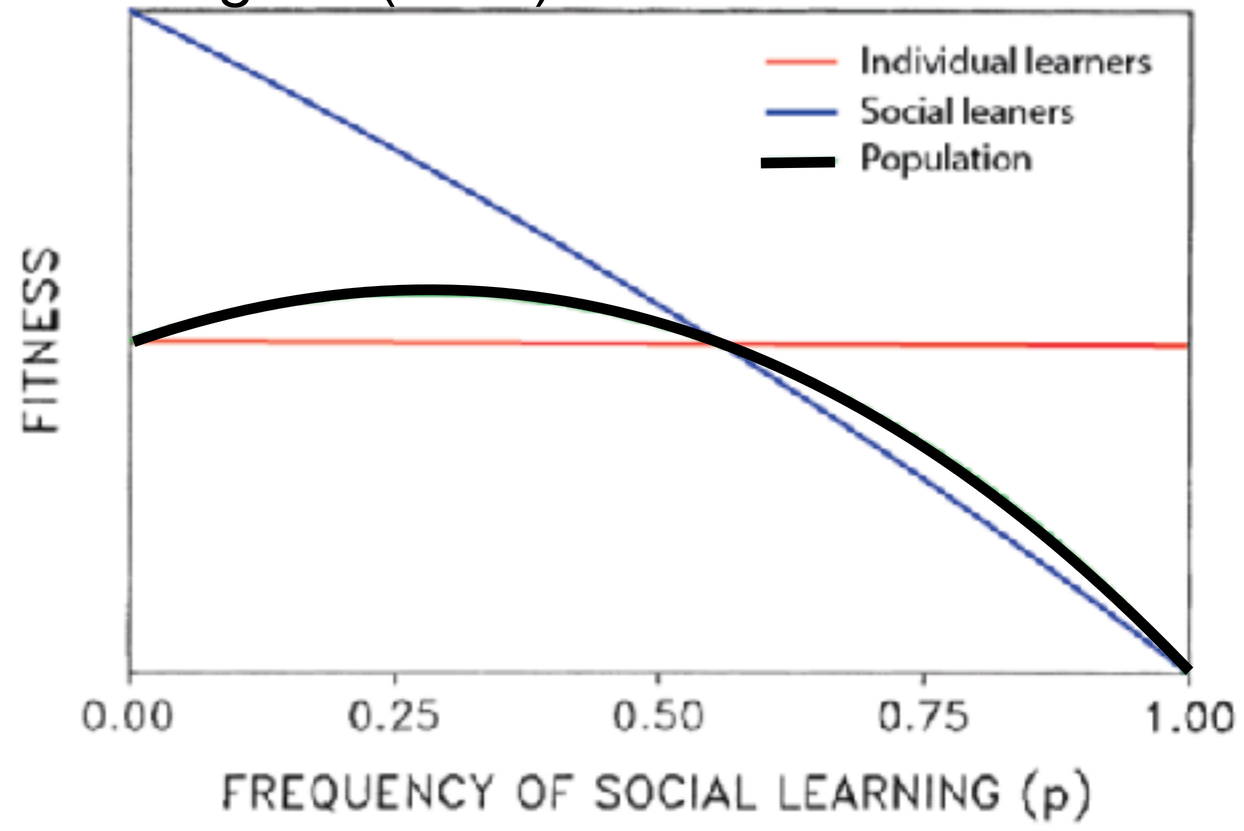


Rogers' Paradox
Average Rewards
7
6
5
4
3
25%
50%
75%
% Social Learners
Ind
Soc
Pop


Rogers' Paradox
Average Rewards
7
6
5
4
3
25%
50%
75%
% Social Learners
ESS
Ind
Soc
Pop


Rogers' Paradox
Average Rewards
% Social Learners
7
6
5
4
3
25%
50%
75%
Ind
Soc
Pop
ESS

Rogers' (1988)


Individual learners
Social leaners
Population
FITNESS
0.00
0.25
0.50
0.75
1.00
FREQUENCY OF SOCIAL LEARNING (p)

# Combining imitation and value-learning

# Decision-biasing social influence

individual social

Choice probability at t = (1 - γ) Softmax + γ FDC

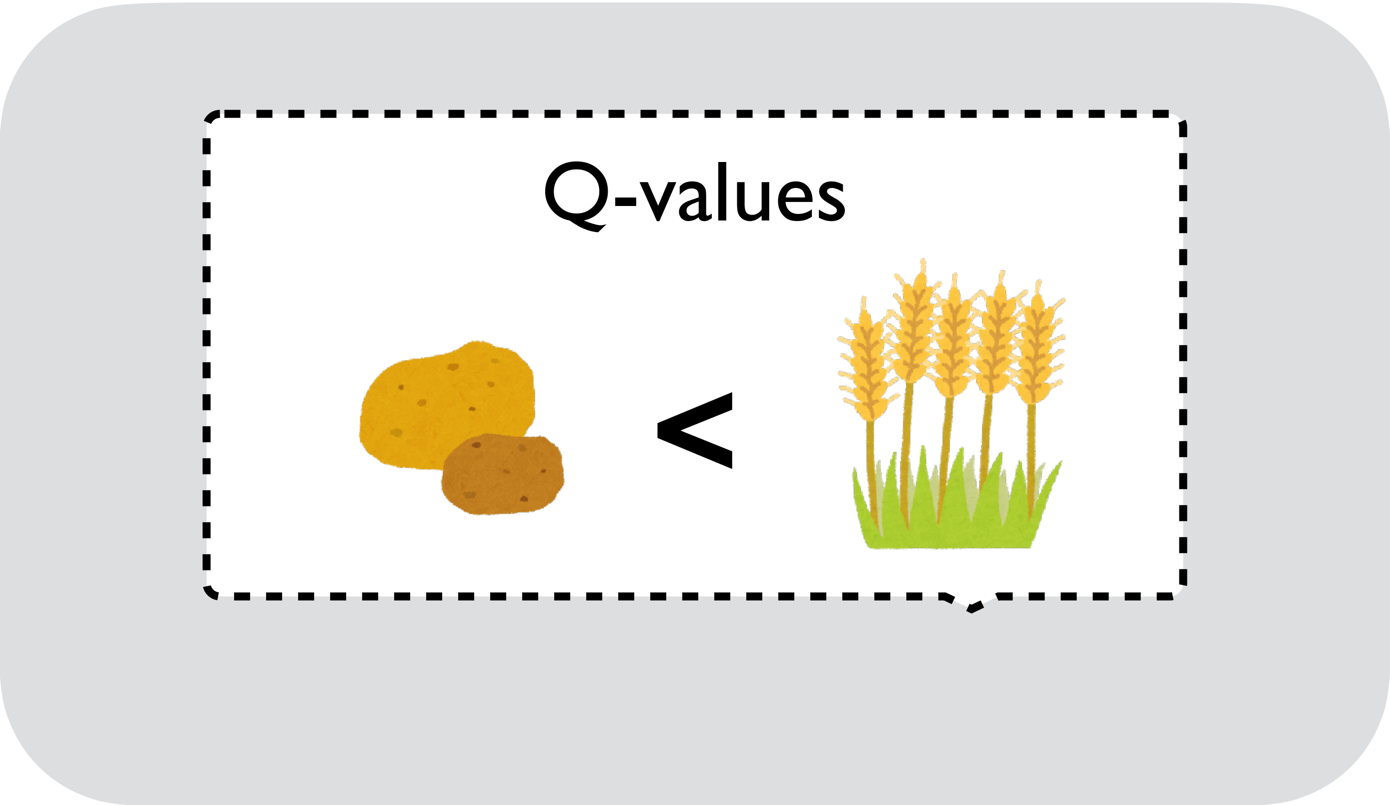
**Individual Q-learning**

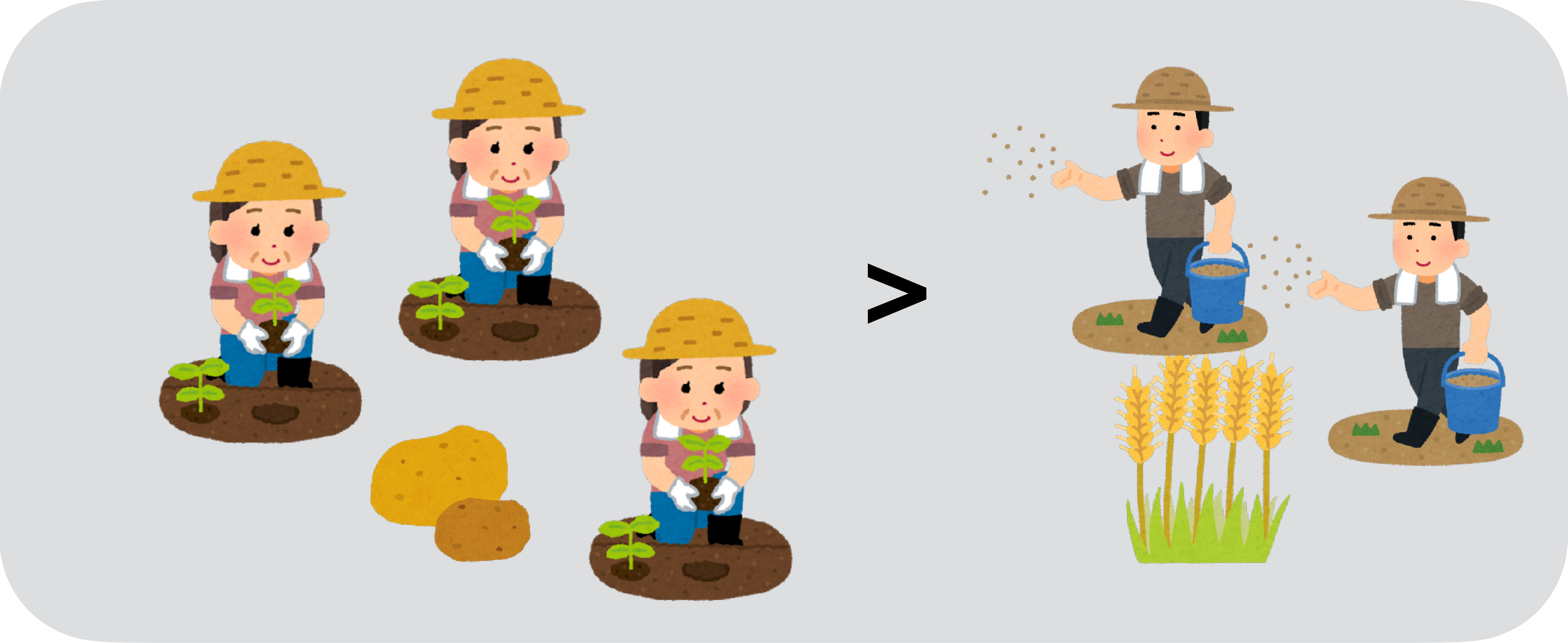
**1 - γ**

**Social frequency influence**

**γ**

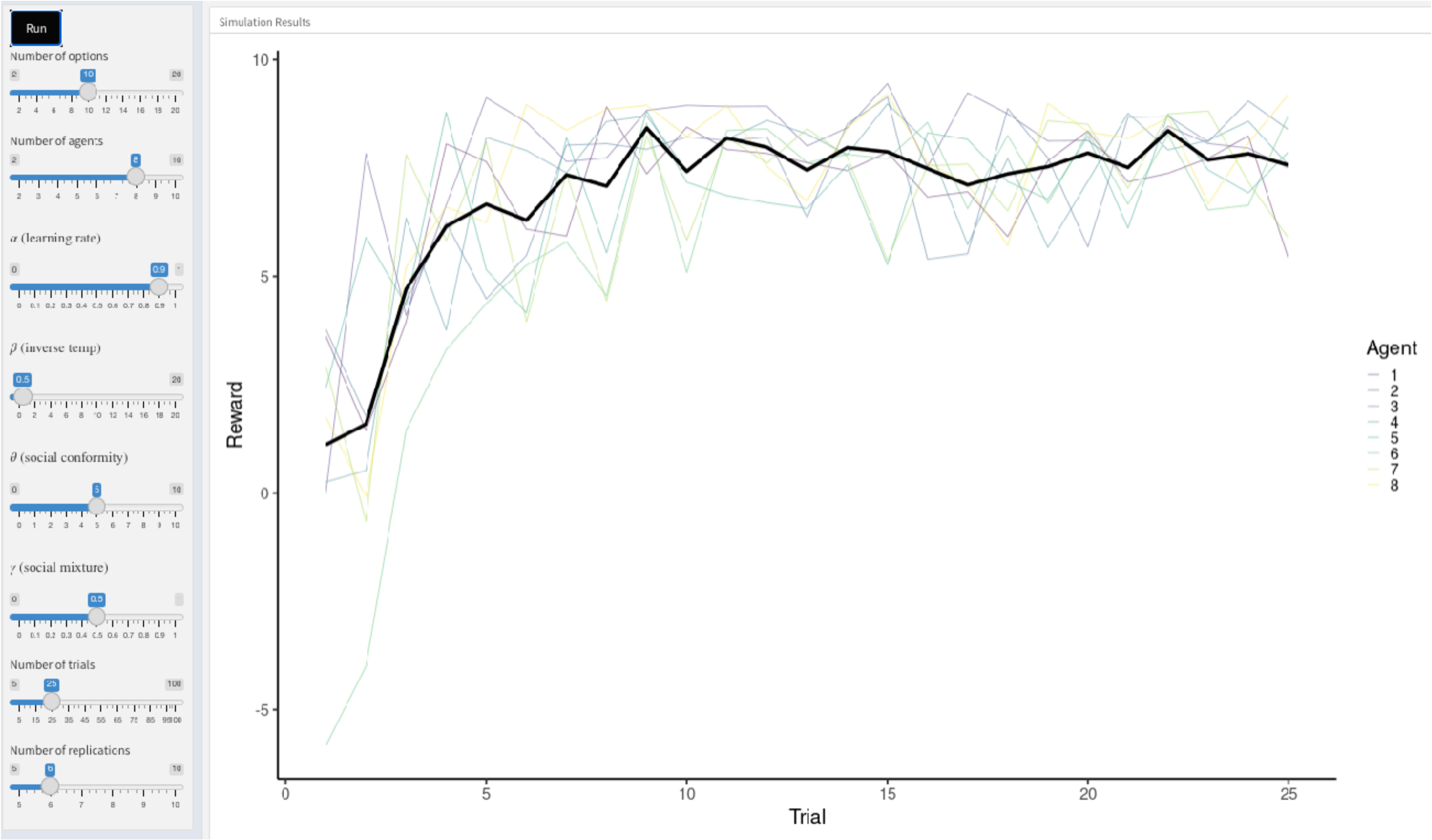
$$\frac{f(i)^{\theta}}{\sum_k f(k)^{\theta}}$$



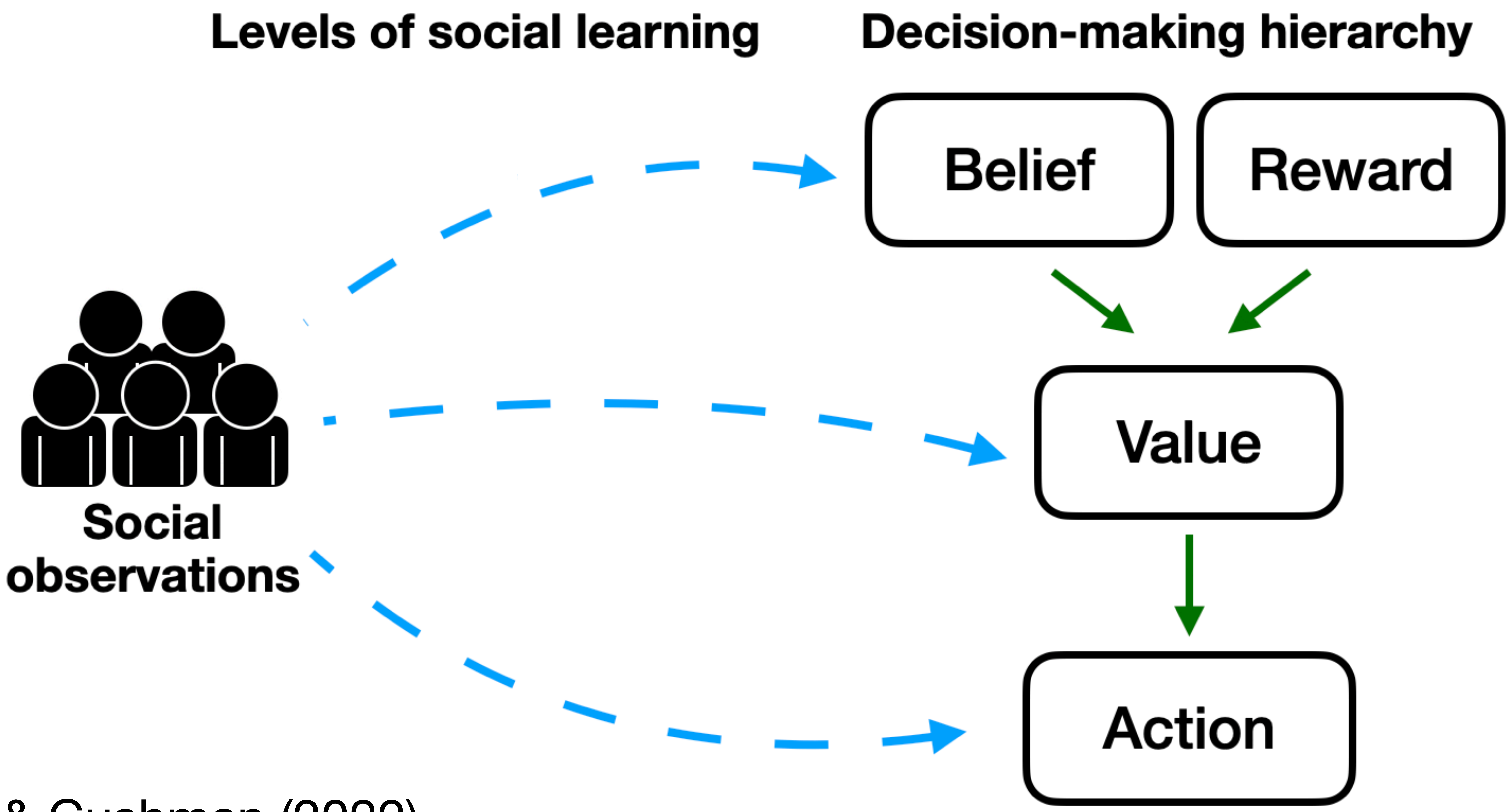
Notebook https://cosmos-konstanz.github.io/notebooks/tutorial-2-models-of-learning.html#combining-imitation-and-value-learning

# Demo 3: Decision-biasing



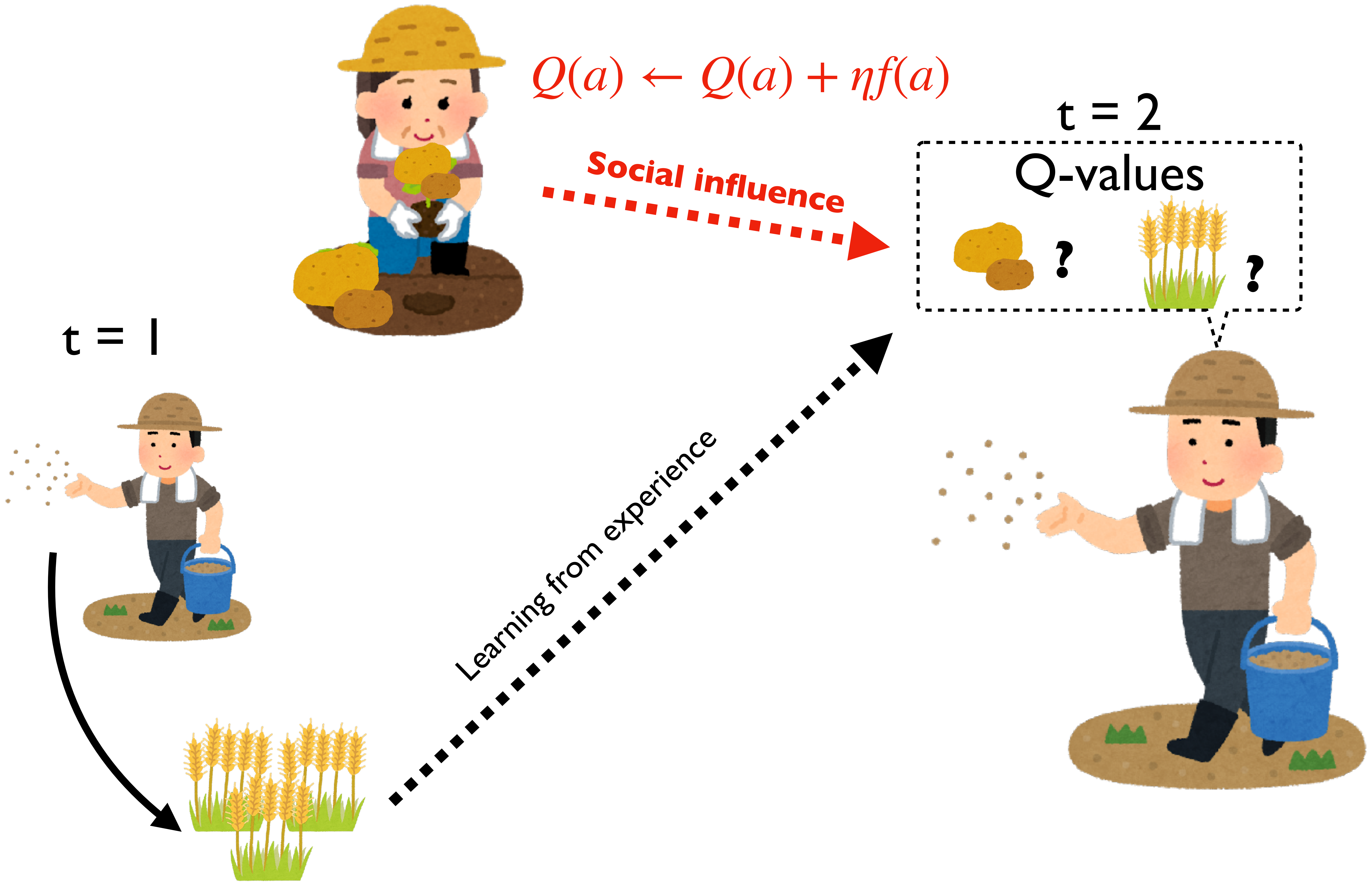

Which values of γ (social mixture) and θ (conformity exponent) typically produce the best results?

# Social influence at different levels of learning

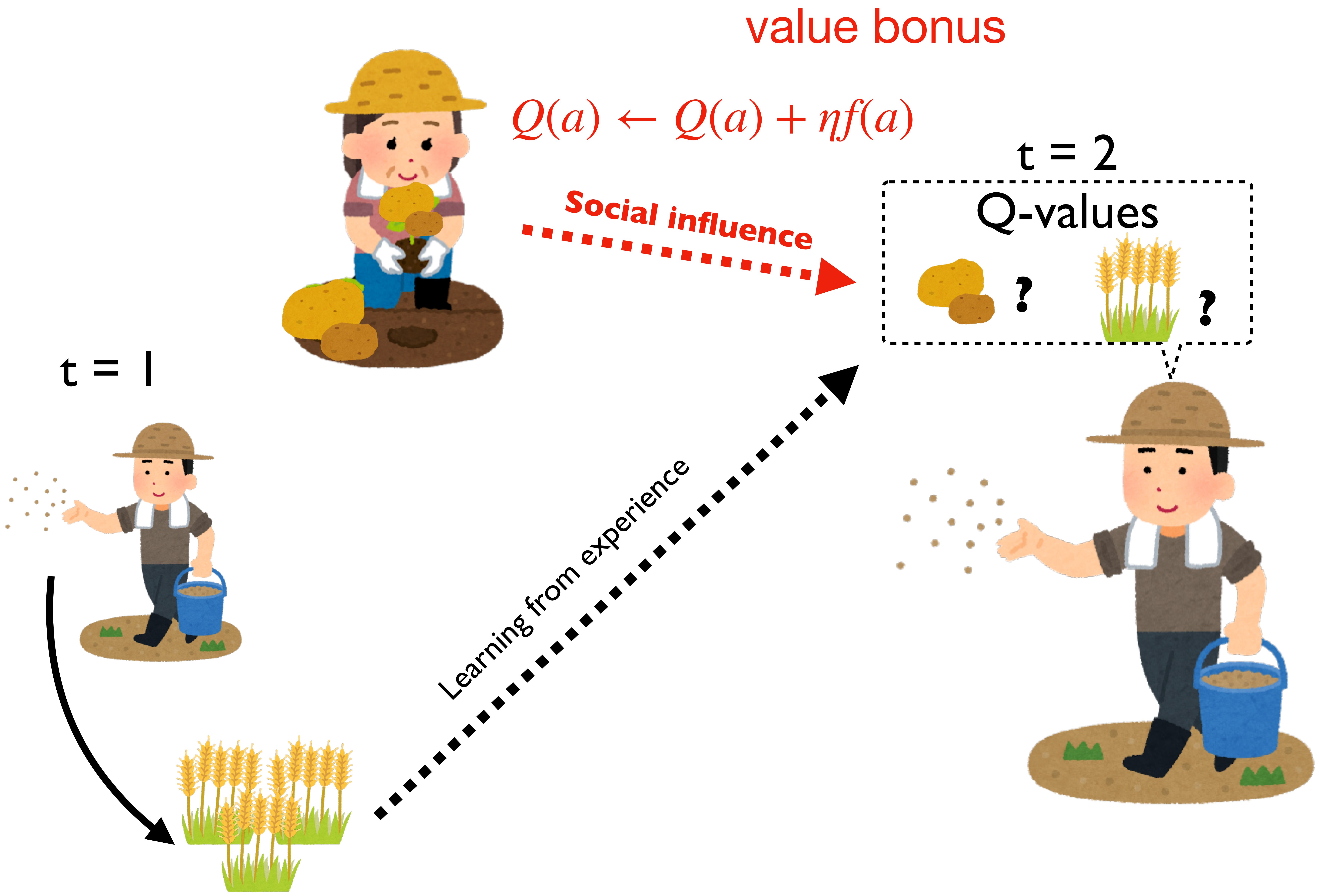



Wu, Vélez, & Cushman (2022)

# Value-shaping social influence

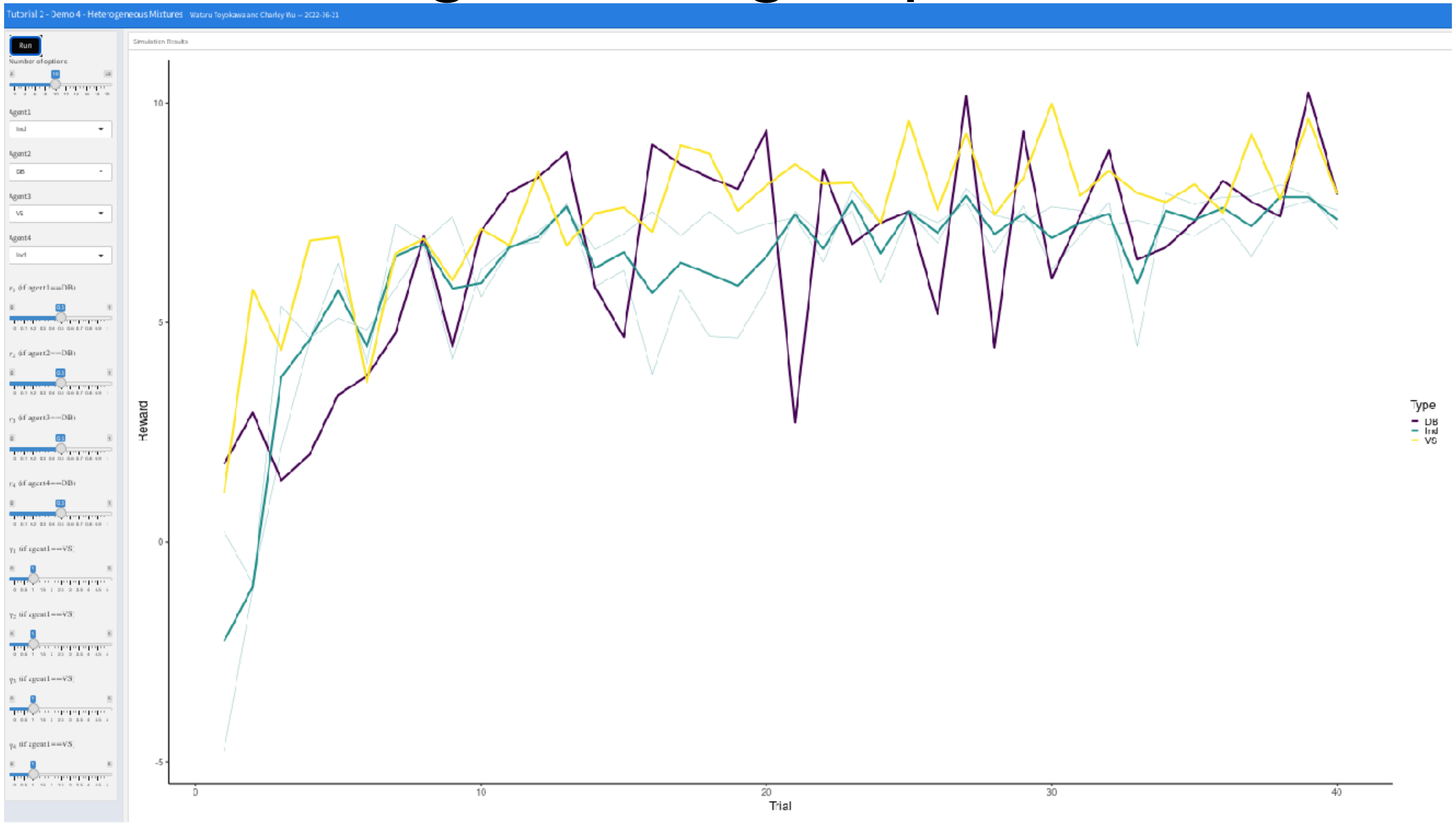


# Value-shaping social influence



Notebook https://cosmos-konstanz.github.io/notebooks/tutorial-2-models-of-learning.html#value-shaping

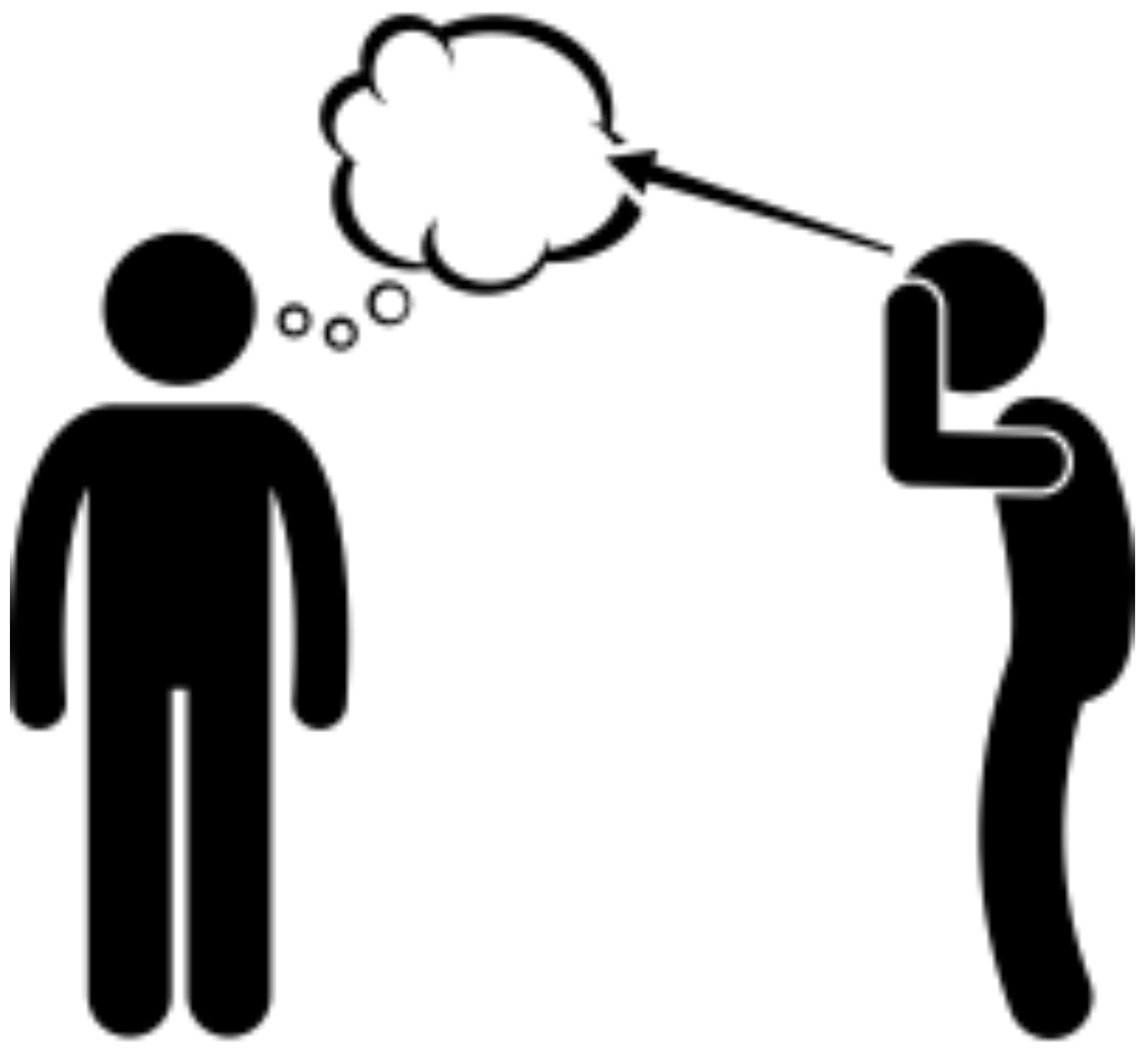
# Demo 4: Heterogeneous groups

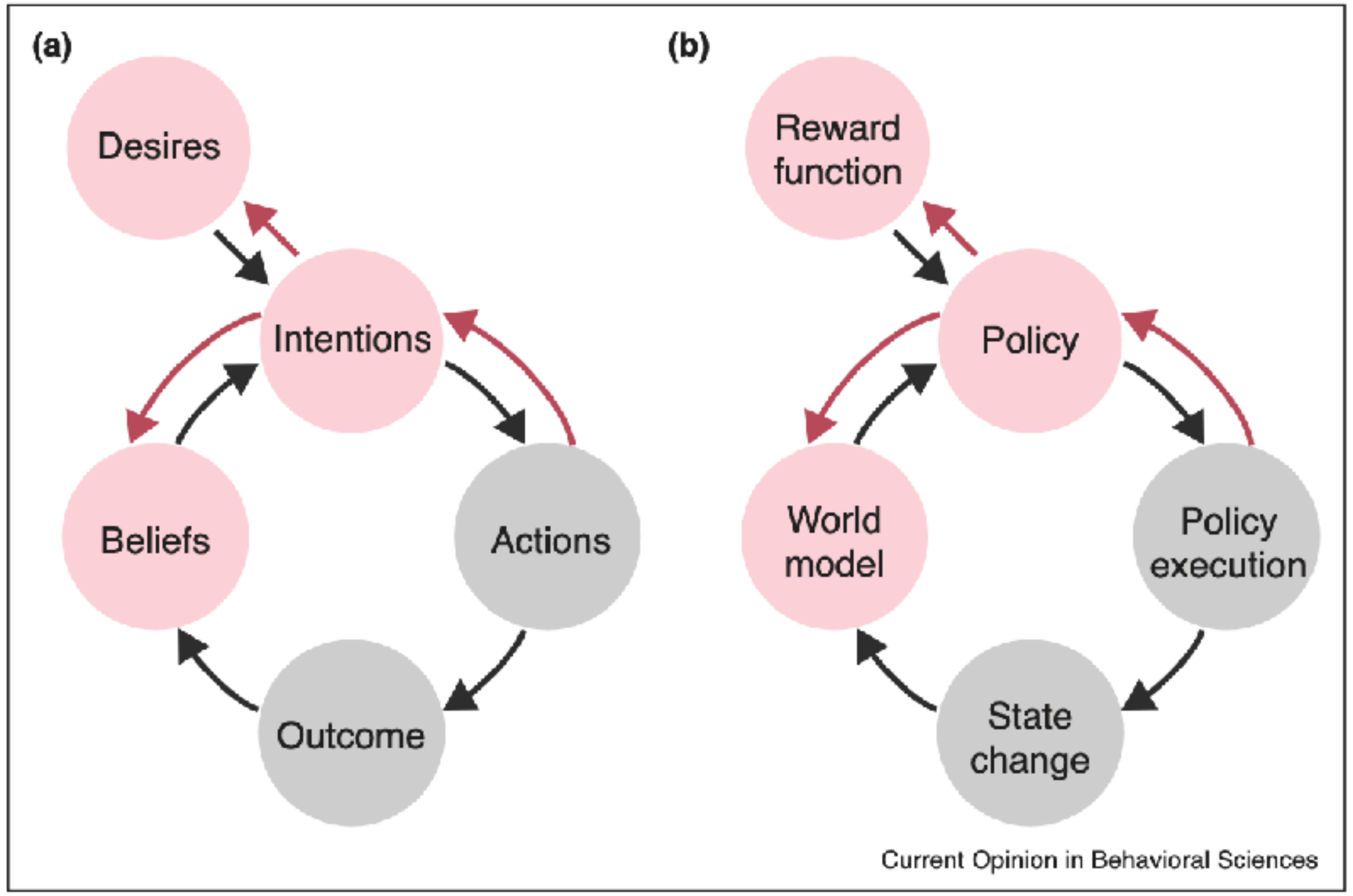



Which strategy perform better than others? Is it robust to different group compositions?

# Theory of Mind

- So far we have described very simple social learning mechanism
- Yet an important aspect of human social learning is our ability to “unpack” observed actions into imputed mental states
    - desires and intentions
    - beliefs and one’s model of the world
- This is known as Theory of Mind (ToM) inference



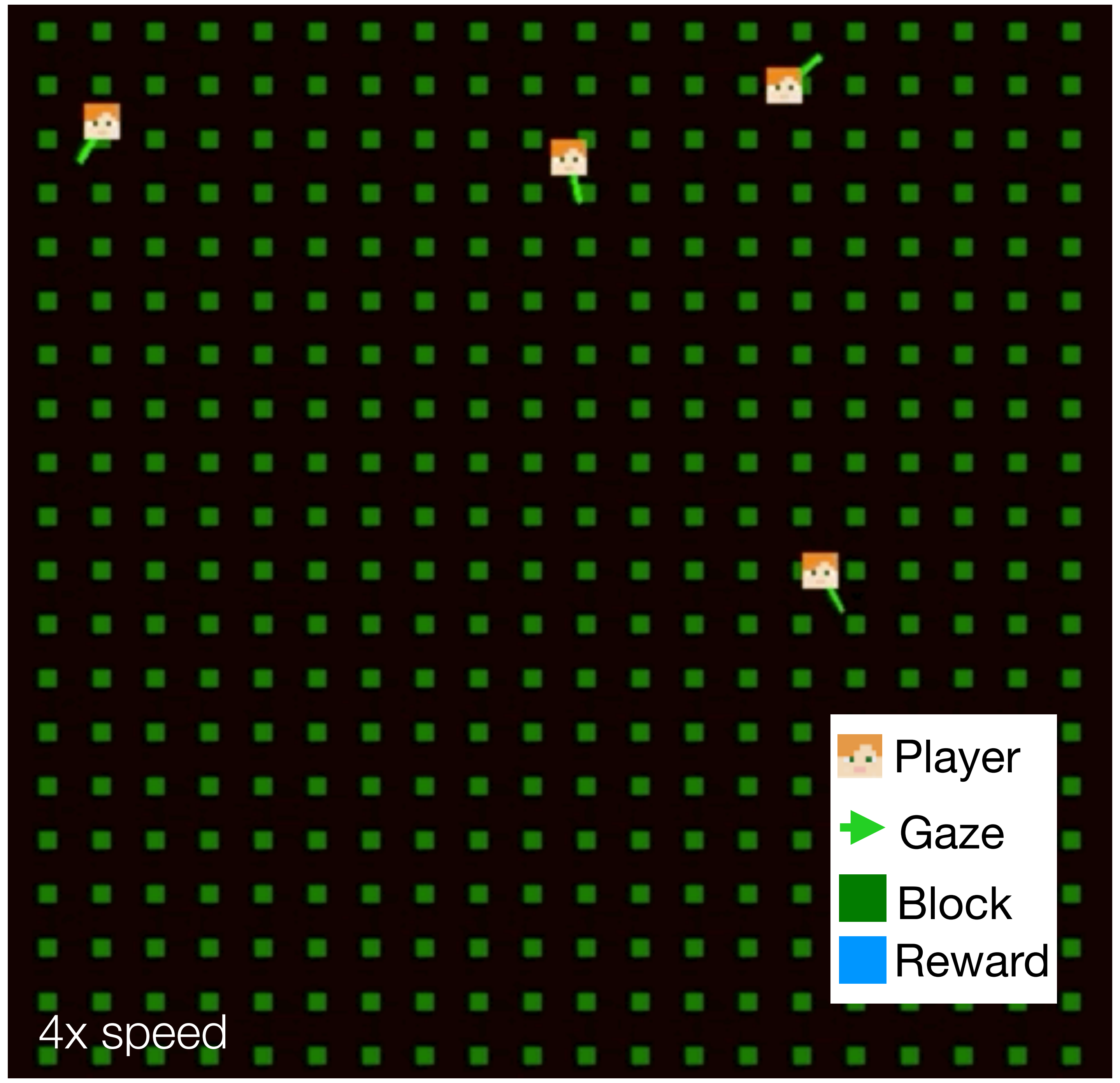



Jara-Ettinger (2019)

Wu et al,. (*bioRxiv* 2023)
Wu et al., (Cogsci 2021)

# Scaling up to more complex tasks

Collective foraging in a dynamic and immersive virtual environment

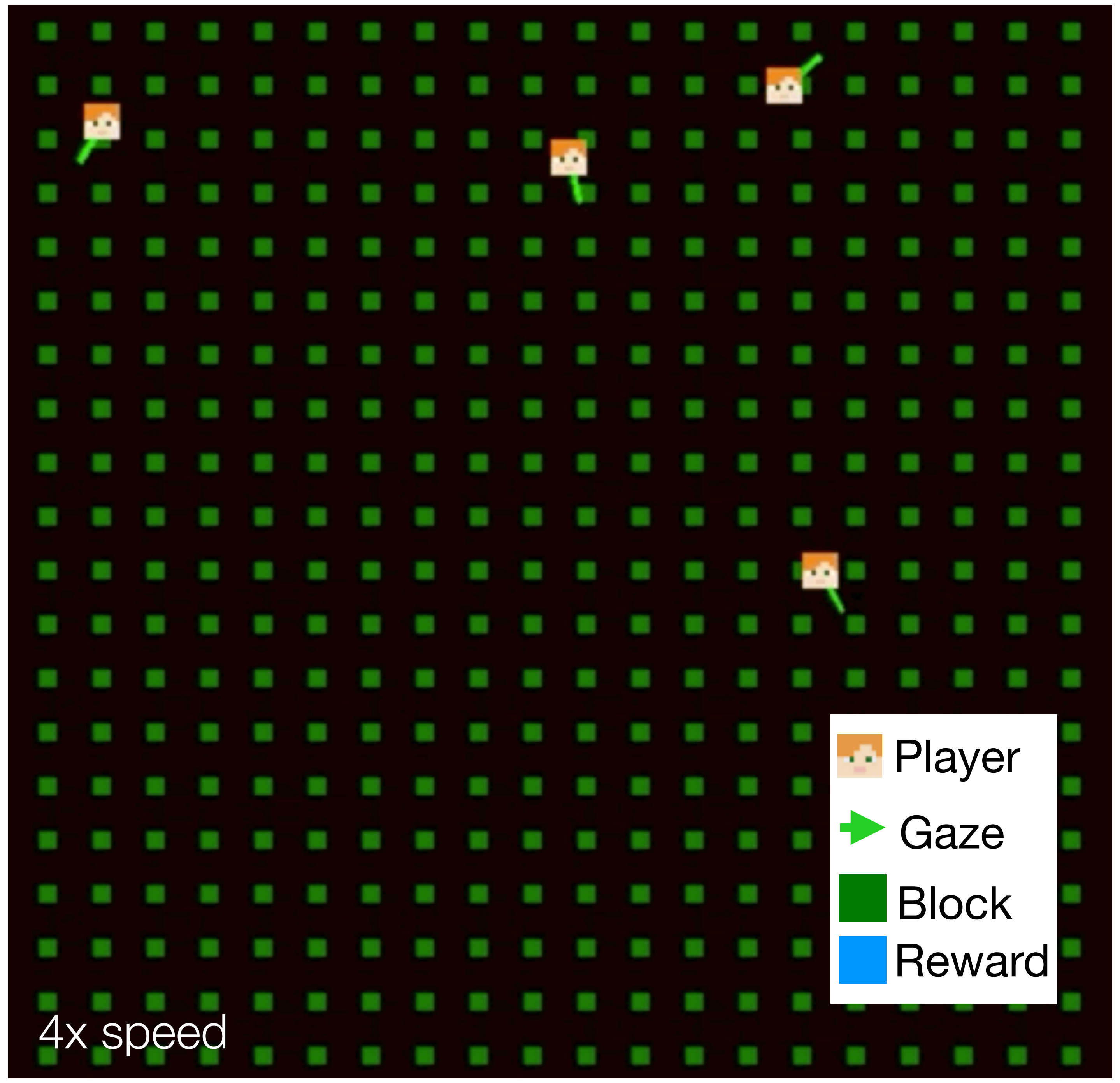



- Participants forage for hidden rewards (blue splash) on a field of melons
- Realistic field of view creates attentional trade-offs and opportunity costs for social learning:
  - Looking at other players for social imitation comes at the cost of slower individual foraging
- Rich and dynamic social interactions through spatial position and visual gaze

Wu et al,. (*bioRxiv* 2023)
Wu et al., (Cogsci 2021)

# Scaling up to more complex tasks

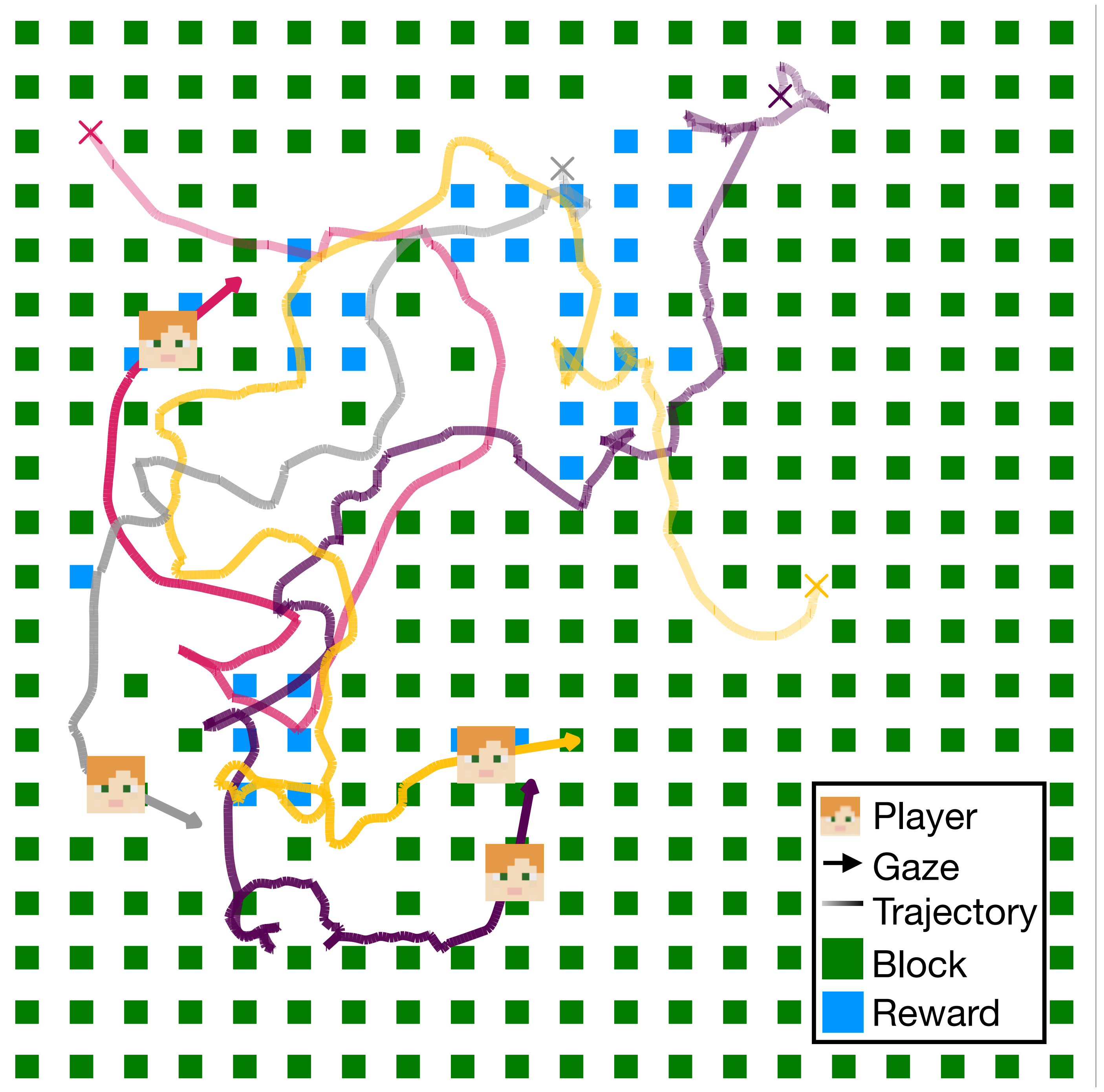
Collective foraging in a dynamic and immersive virtual environment




- Participants forage for hidden rewards (blue splash) on a field of melons
- Realistic field of view creates attentional trade-offs and opportunity costs for social learning:
    - Looking at other players for social imitation comes at the cost of slower individual foraging
- Rich and dynamic social interactions through spatial position and visual gaze

# Scaling up to more complex tasks

## Collective foraging in a dynamic and immersive virtual environment




- Participants forage for hidden rewards (blue splash) on a field of melons
- Realistic field of view creates attentional trade-offs and opportunity costs for social learning:
  - Looking at other players for social imitation comes at the cost of slower individual foraging
- Rich and dynamic social interactions through spatial position and visual gaze

# Interactive Tutorial


2x speed

- Smash blocks by clicking and holding mouse (2.25 seconds)
- Some blocks contain rewards, indicated by a blue splash, visible to other players
- Other blocks have no reward
- Participants incentivized to collect as many rewards as possible

# Interactive Tutorial

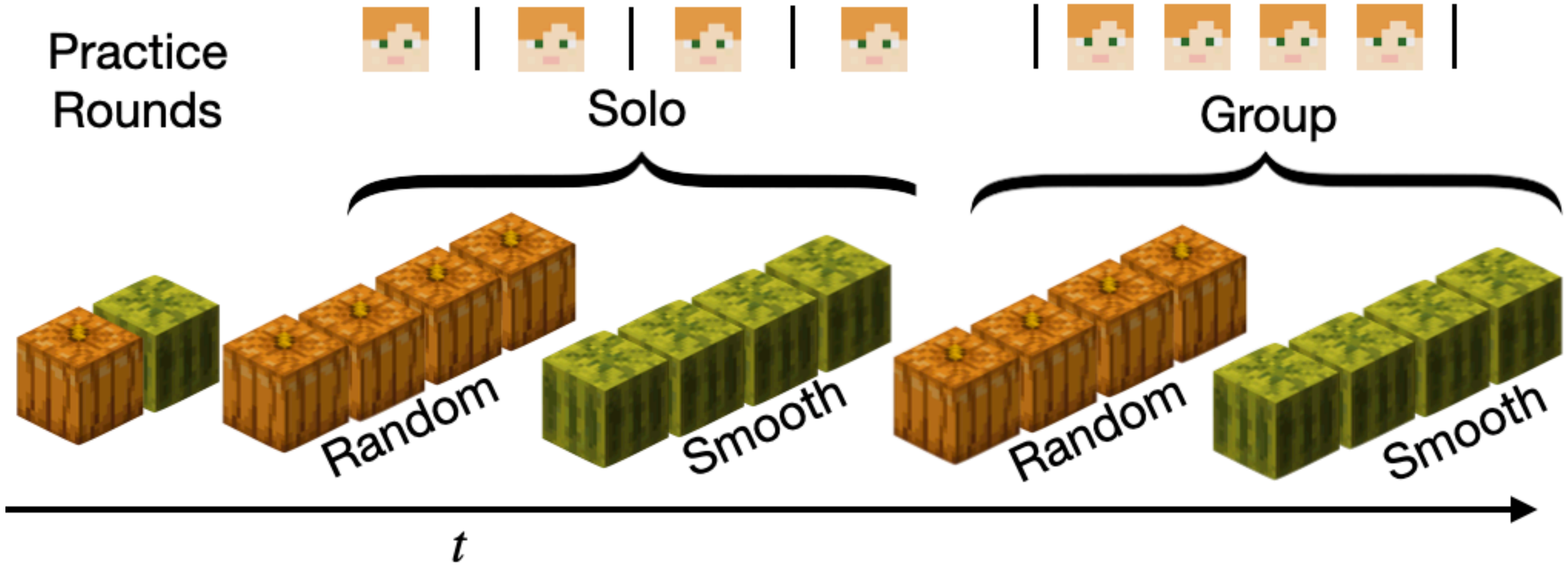

2x speed

- Smash blocks by clicking and holding mouse (2.25 seconds)
- Some blocks contain rewards, indicated by a blue splash, visible to other players
- Other blocks have no reward
- Participants incentivized to collect as many rewards as possible

# Experimental Design

**Smooth**

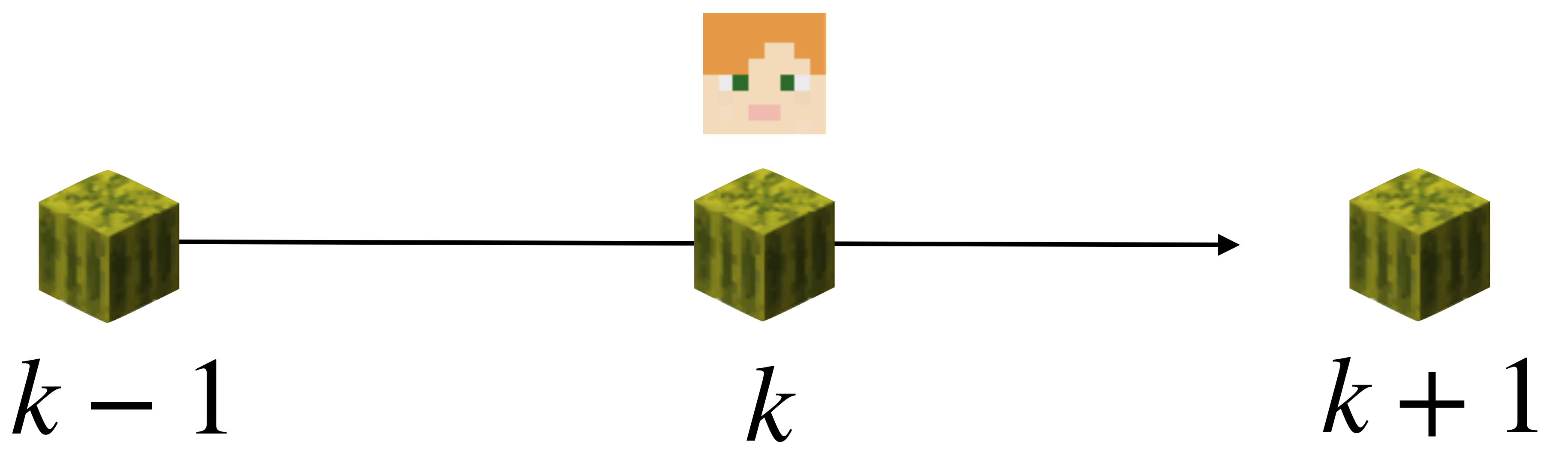
**Random**




16 rounds with a 2x2 within-subject design

- Environment: smooth vs. random
- Condition: solo vs. groups of four

# Computational models



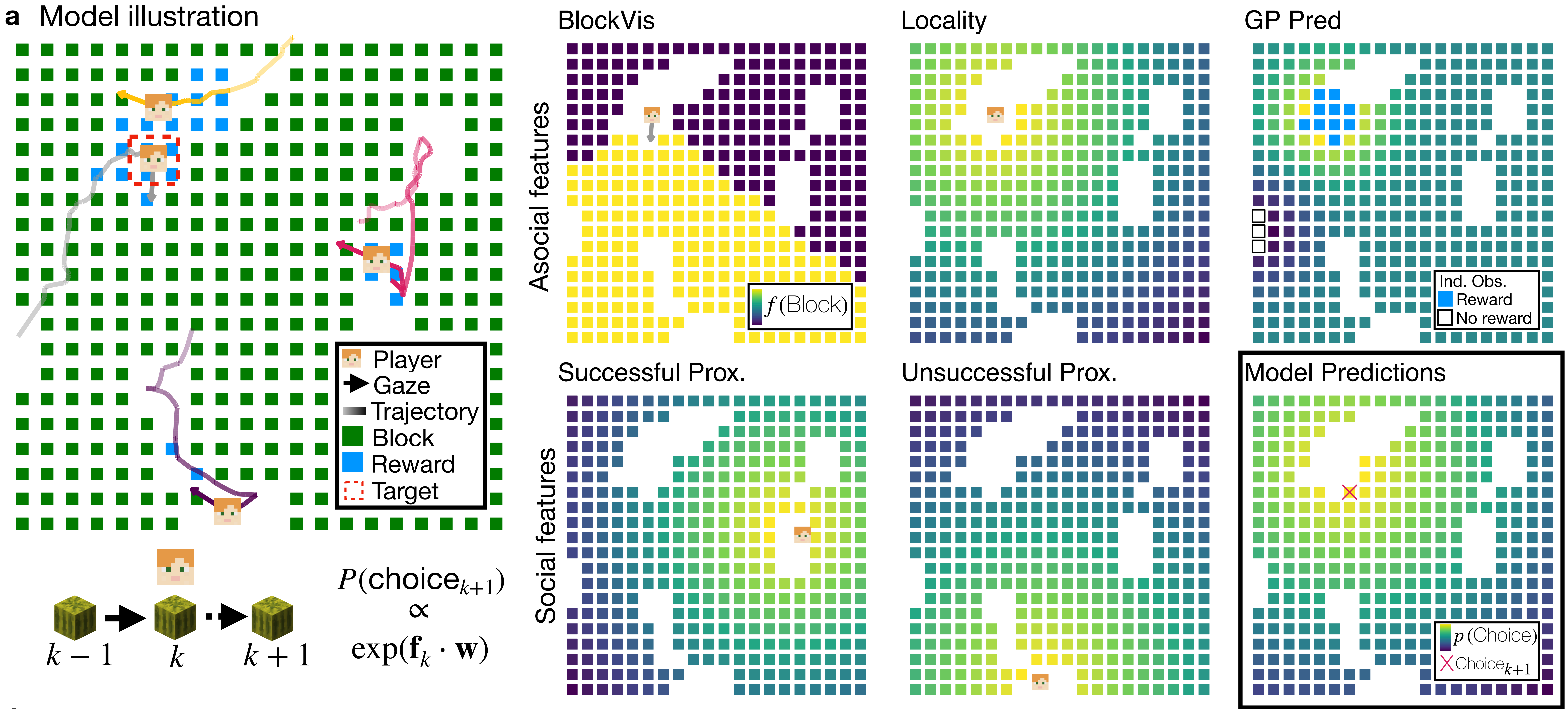

Sequentially predict each of the $k$ blocks participants destroy:

$$P(\text{Choice}_{k+1}) \propto \exp(\mathbf{f}_k \cdot \mathbf{w})$$

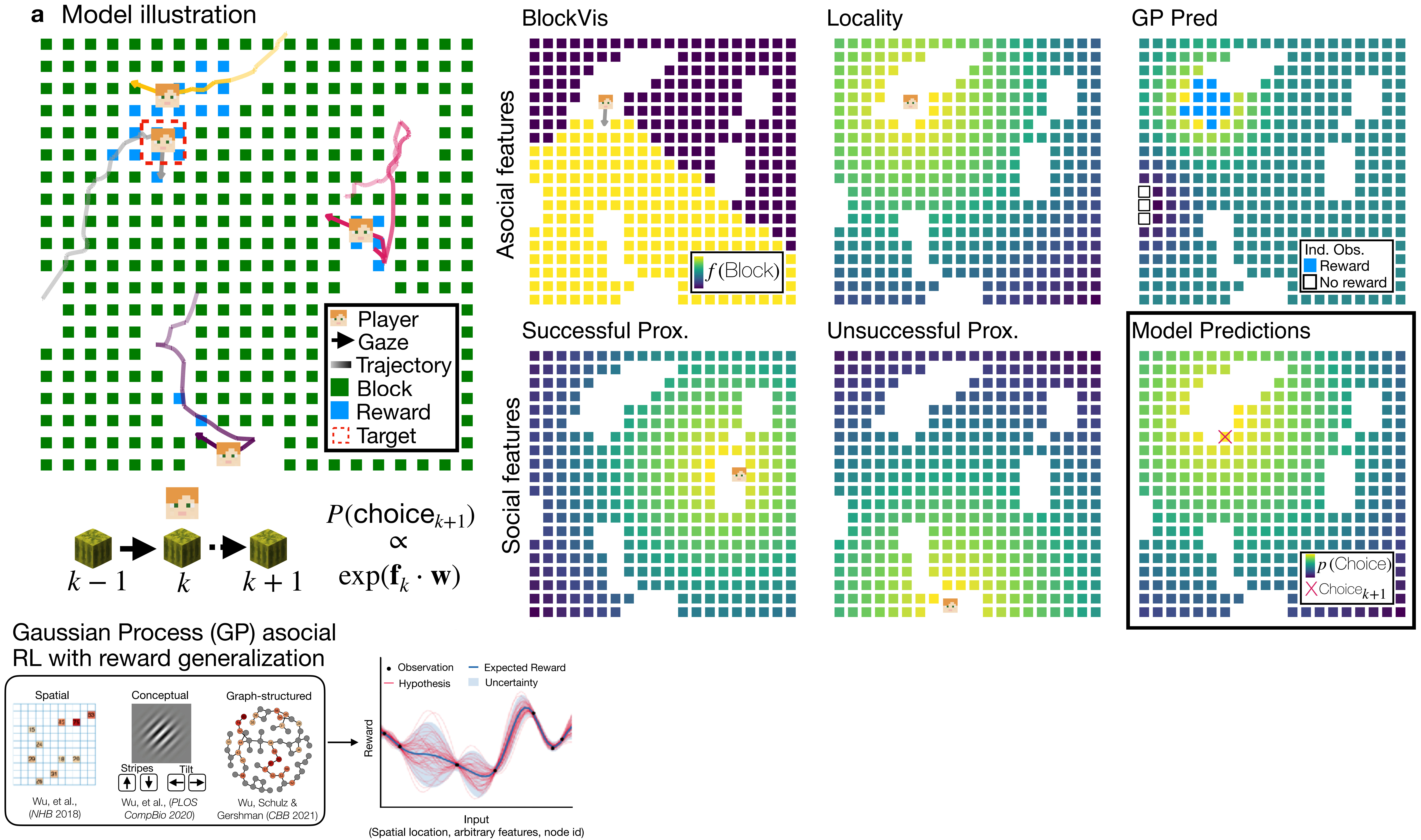
using a softmax over a set of features $\mathbf{f}$ times weights $\mathbf{w}$

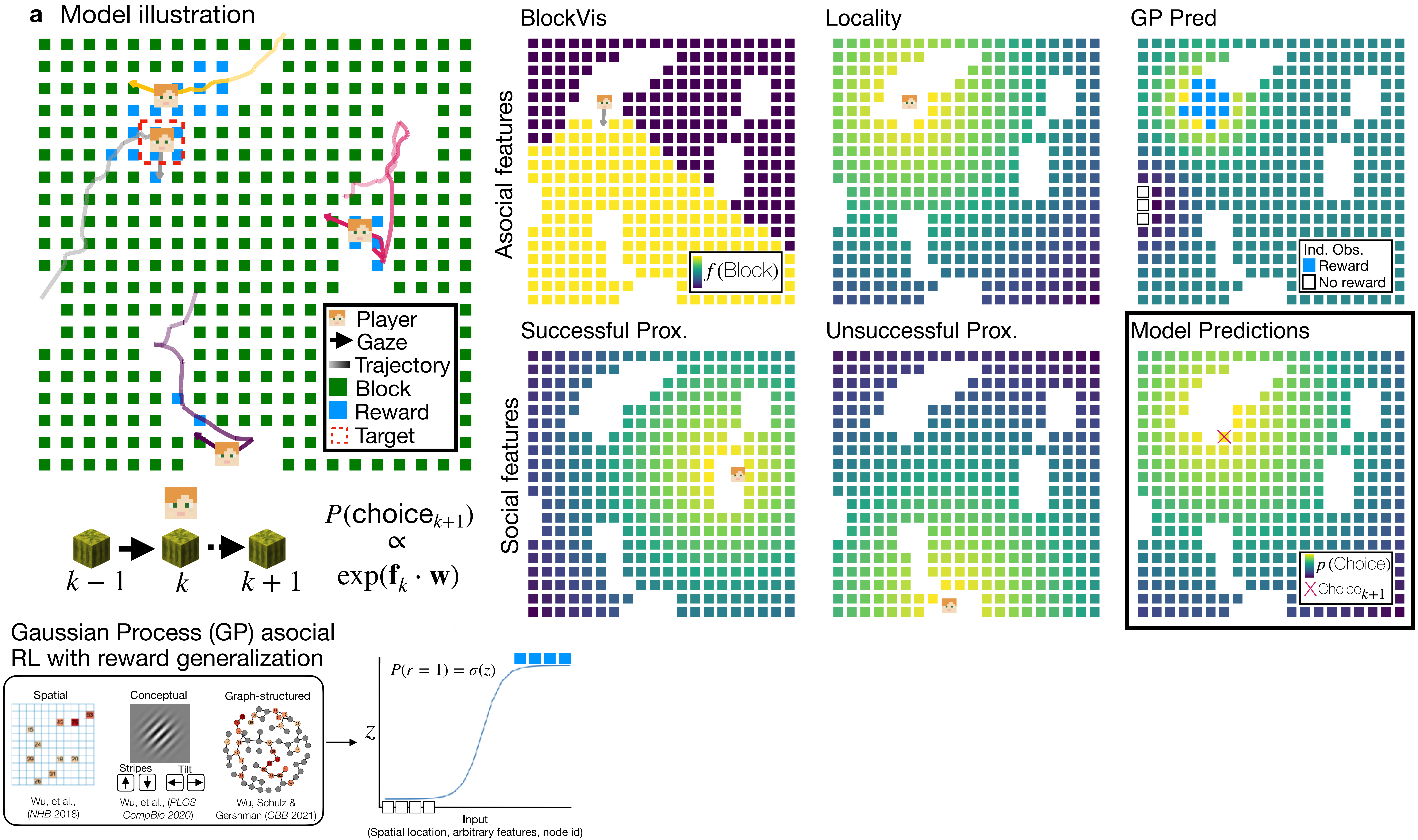
Model **f**eatures capture hypotheses about individual and social learning mechanisms (details on next slide)

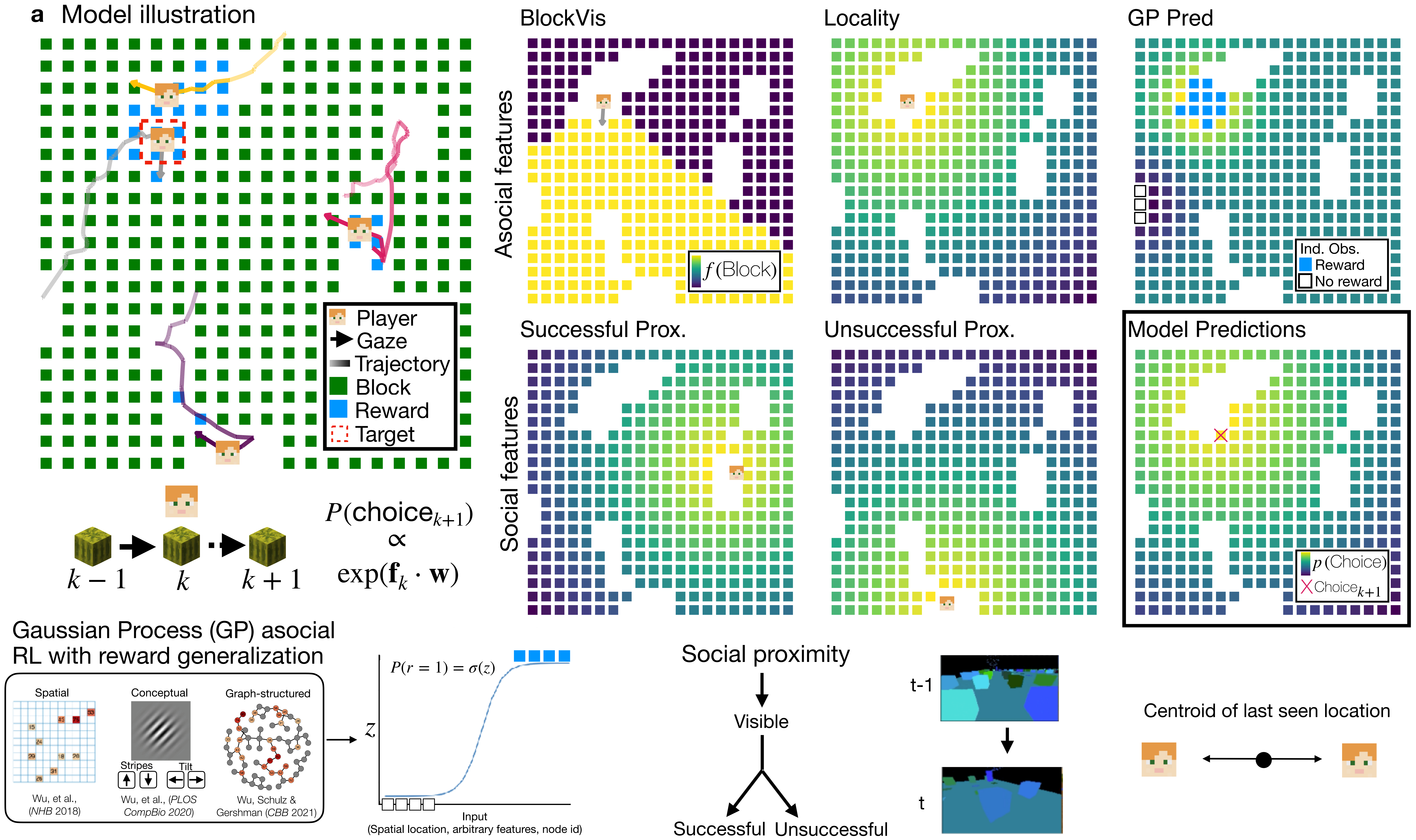
Model **w**eights are estimated using hierarchical Bayesian methods in STAN with individual and group as random effects



a Model illustration
Player
Gaze
Trajectory
Block
Reward
Target
$k-1$
$k$
$k+1$
$P(\text{choice}_{k+1}) \propto \exp(\mathbf{f}_k \cdot \mathbf{w})$
Asocial features
BlockVis
$f$(Block)
Locality
GP Pred
Ind. Obs.
Reward
No reward
Social features
Successful Prox.
Unsuccessful Prox.
Model Predictions
$p$(Choice)
Choice$_{k+1}$

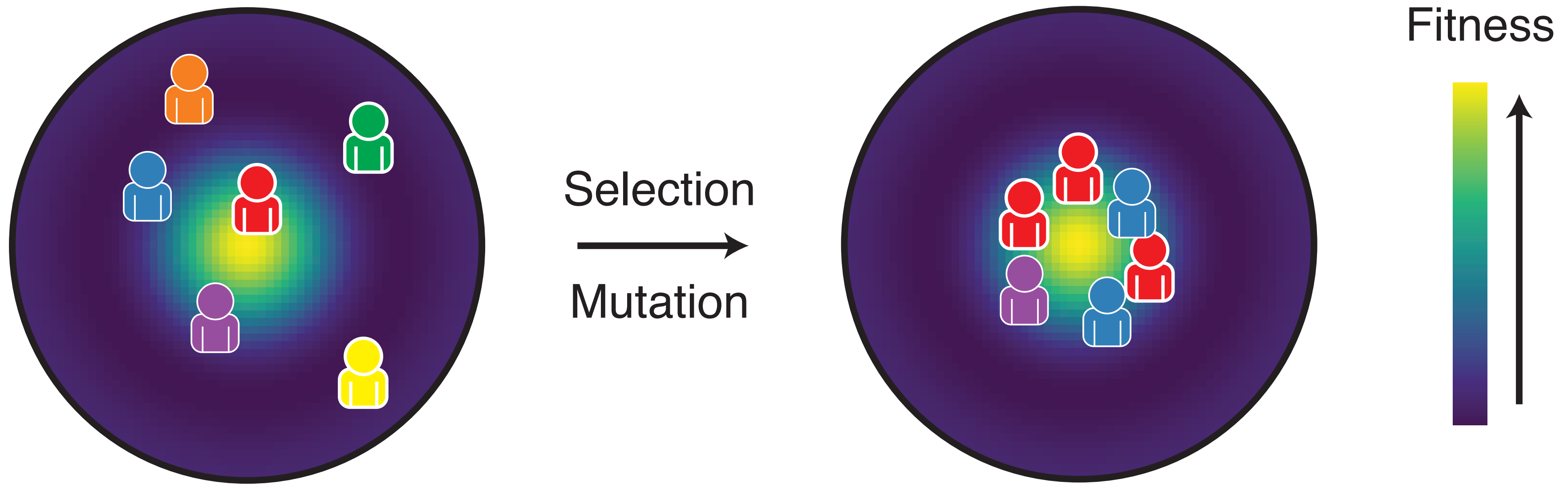

a Model illustration
Player
Gaze
Trajectory
Block
Reward
Target
k − 1
k
k + 1
P(choice_{k+1}) ∝ exp(f_k · w)
BlockVis
Locality
GP Pred
Asocial features
f(Block)
Ind. Obs.
Reward
No reward
Successful Prox.
Unsuccessful Prox.
Model Predictions
Social features
p(Choice)
Choice_{k+1}
Gaussian Process (GP) asocial RL with reward generalization
Spatial
Conceptual
Graph-structured
Stripes
Tilt
Wu, et al., (NHB 2018)
Wu, et al., (PLOS CompBio 2020)
Wu, Schulz & Gershman (CBB 2021)
Observation
Expected Reward
Hypothesis
Uncertainty
Reward
Input
(Spatial location, arbitrary features, node id)

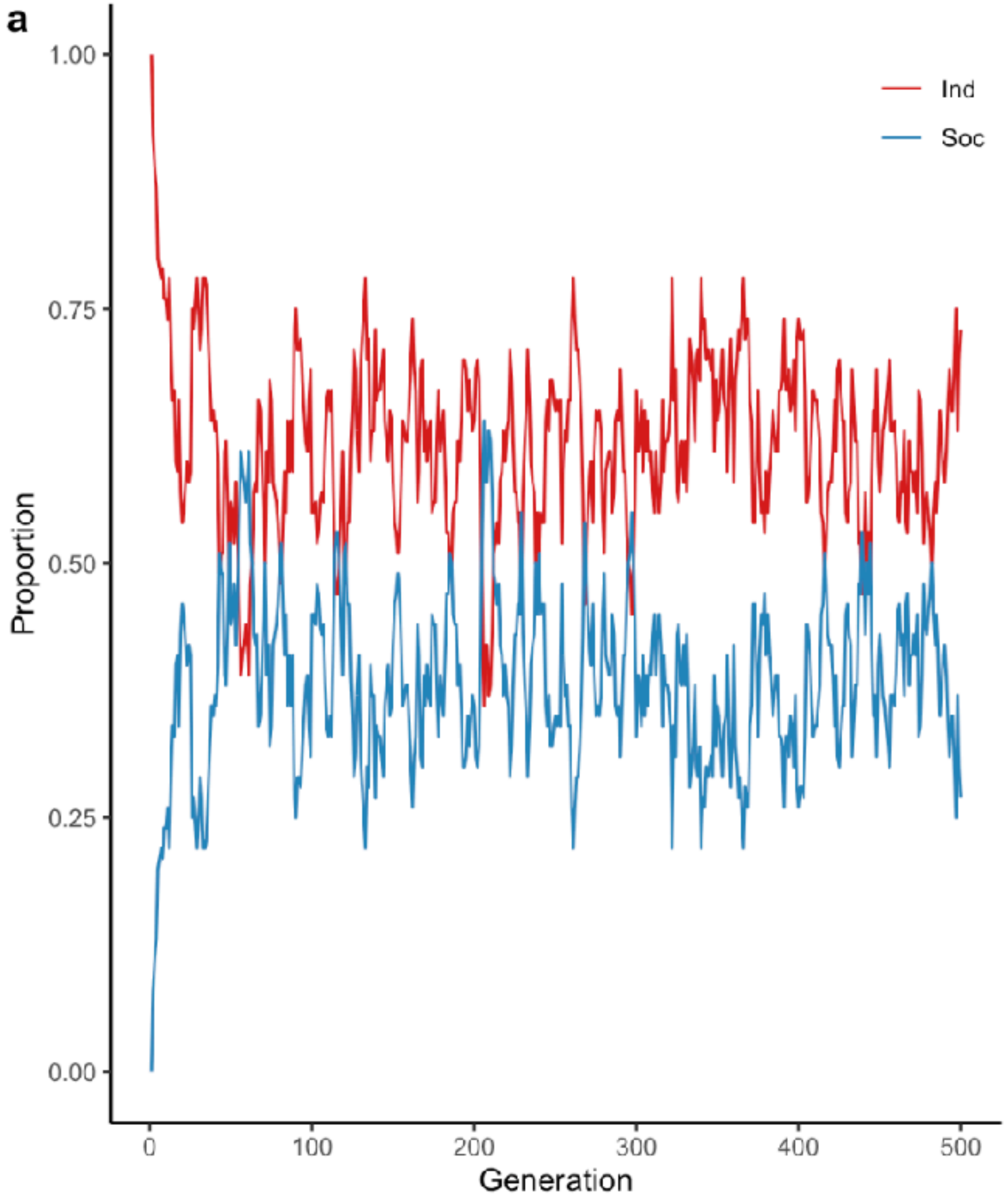

a Model illustration
Player
Gaze
Trajectory
Block
Reward
Target
$k-1$
$k$
$k+1$
$P(\text{choice}_{k+1}) \propto \exp(\mathbf{f}_k \cdot \mathbf{w})$
Asocial features
BlockVis
$f$(Block)
Locality
GP Pred
Ind. Obs.
Reward
No reward
Social features
Successful Prox.
Unsuccessful Prox.
Model Predictions
$p$(Choice)
Choice$_{k+1}$
Gaussian Process (GP) asocial
RL with reward generalization
Spatial
Conceptual
Graph-structured
Stripes
Tilt
Wu, et al., (NHB 2018)
Wu, et al., (PLOS CompBio 2020)
Wu, Schulz & Gershman (CBB 2021)
$P(r=1)=\sigma(z)$
$z$
Input
(Spatial location, arbitrary features, node id)


a Model illustration
Player
Gaze
Trajectory
Block
Reward
Target
$k-1$
$k$
$k+1$
$P(\text{choice}_{k+1}) \propto \exp(\mathbf{f}_k \cdot \mathbf{w})$
BlockVis
Locality
GP Pred
Asocial features
$f$(Block)
Ind. Obs.
Reward
No reward
Successful Prox.
Unsuccessful Prox.
Model Predictions
Social features
$p$(Choice)
Choice$_{k+1}$
Gaussian Process (GP) asocial RL with reward generalization
Spatial
Conceptual
Stripes
Tilt
Graph-structured
Wu, et al., (NHB 2018)
Wu, et al., (PLOS CompBio 2020)
Wu, Schulz & Gershman (CBB 2021)
$P(r=1) = \sigma(z)$
$z$
Input
(Spatial location, arbitrary features, node id)
Social proximity
Visible
Successful
Unsuccessful
t-1
t
Centroid of last seen location

# Evolutionary dynamics

- Social learning has *frequency-dependent fitness* (Rogers, 1988)
  - The best strategy to use depends on what others in the population are doing
- In order to determine the best *normative* strategy, it is often helpful to use evolutionary simulations:
  1. Initialize a population of agents
  2. Simulate performance on the task
  3. Select agents to seed the next generation (e.g., based on performance)
  4. Add mutation (change agent type, modify parameters)
  5. Repeat until convergence

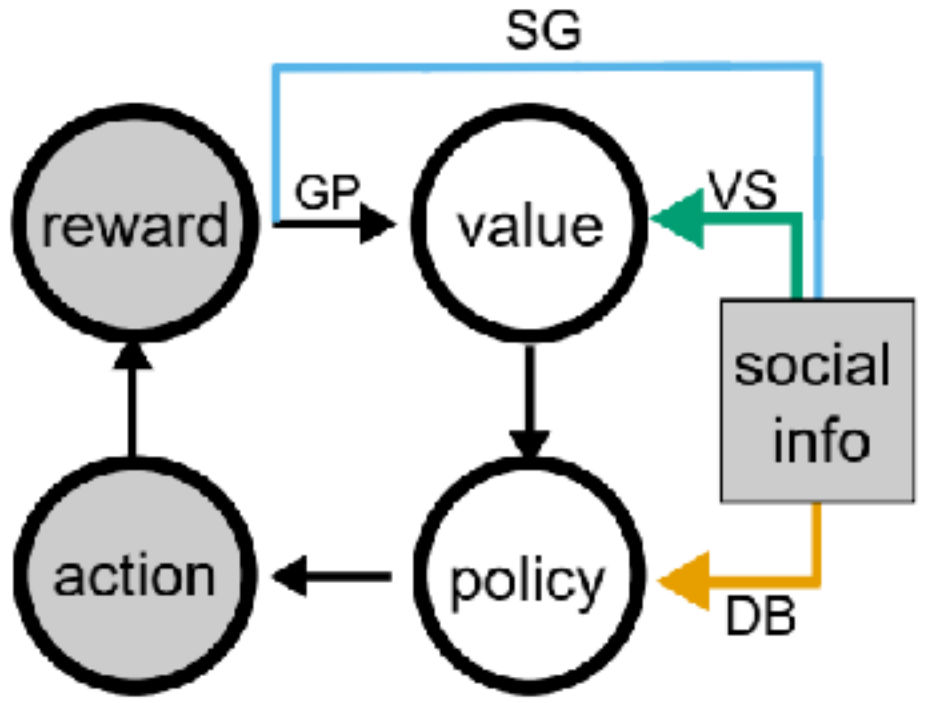


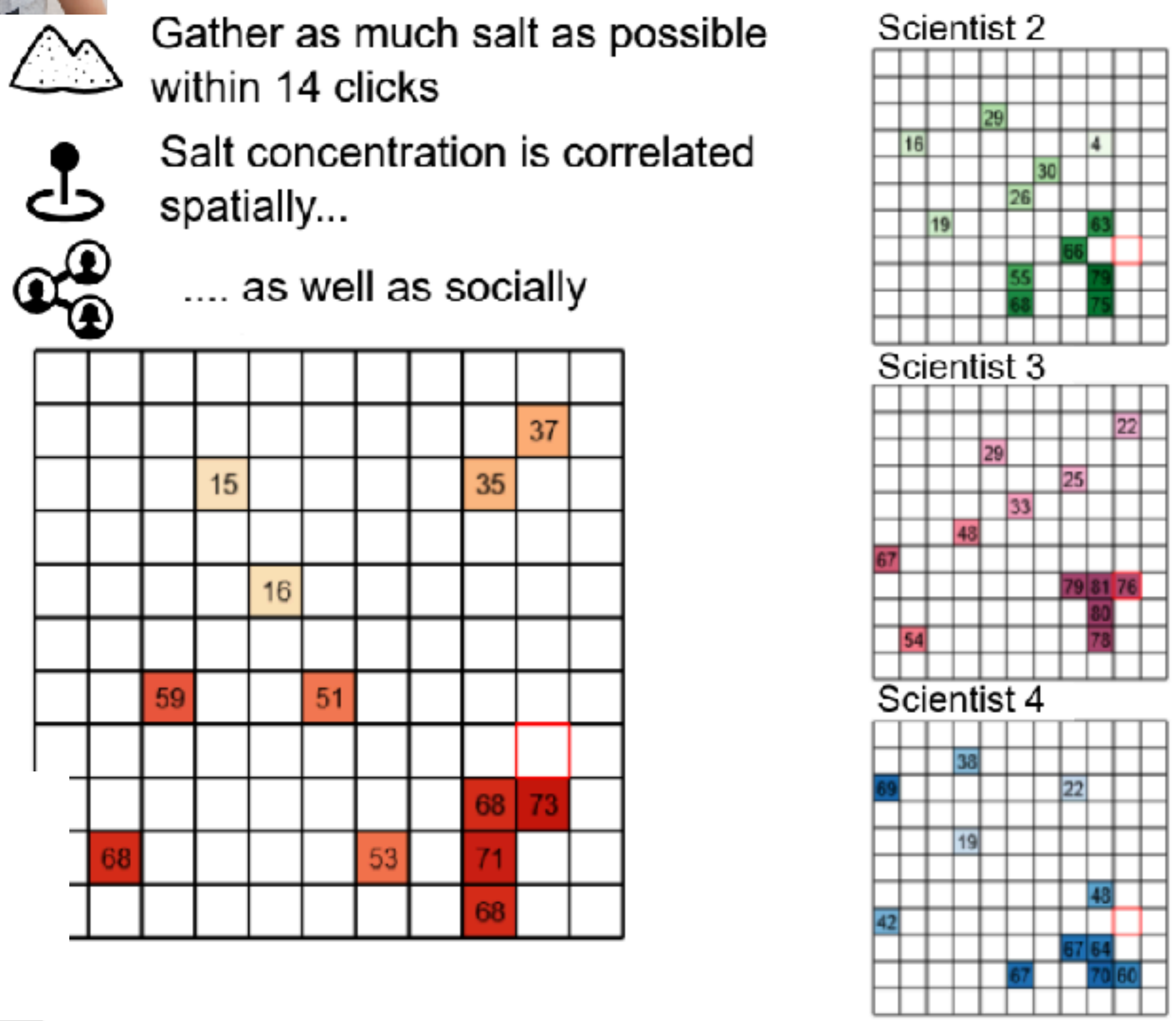



Witt, Toyokawa, Gaissmaier, Lala, & Wu (*Cogsci* 2023)

# Evolutionary simulations

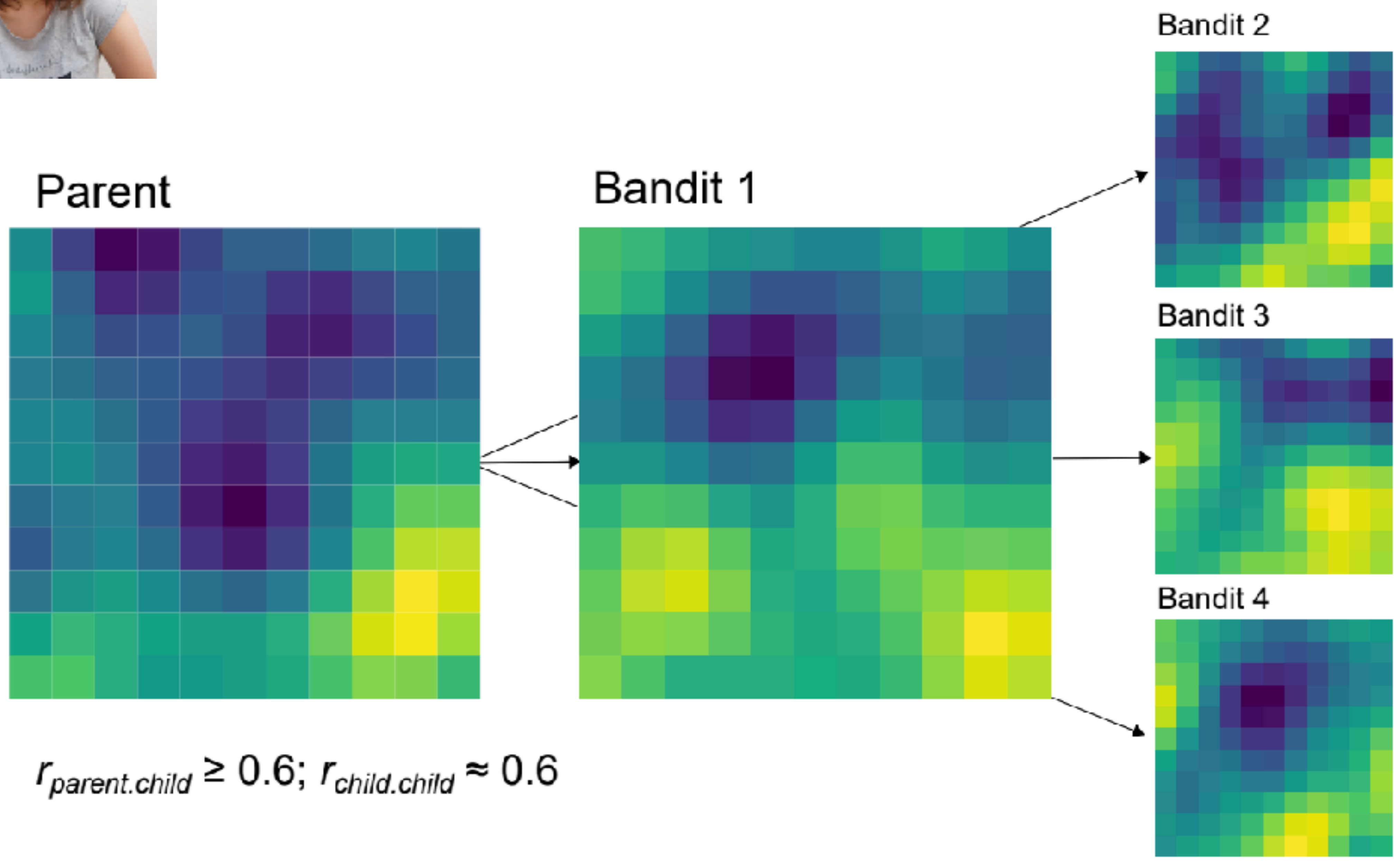
- Social learning despite individual differences (Witt et al., 2023)
  - People can use social information, but not verbatim
  - Exact imitation strategies might fail to account for social differences
- Decision-Biasing (**DB**)
- Value-Shaping (**VS**)
- Social Generalization (**SG**):
  - integration social info in the reward generalization process
  - assume social info is noisier than individual experiences



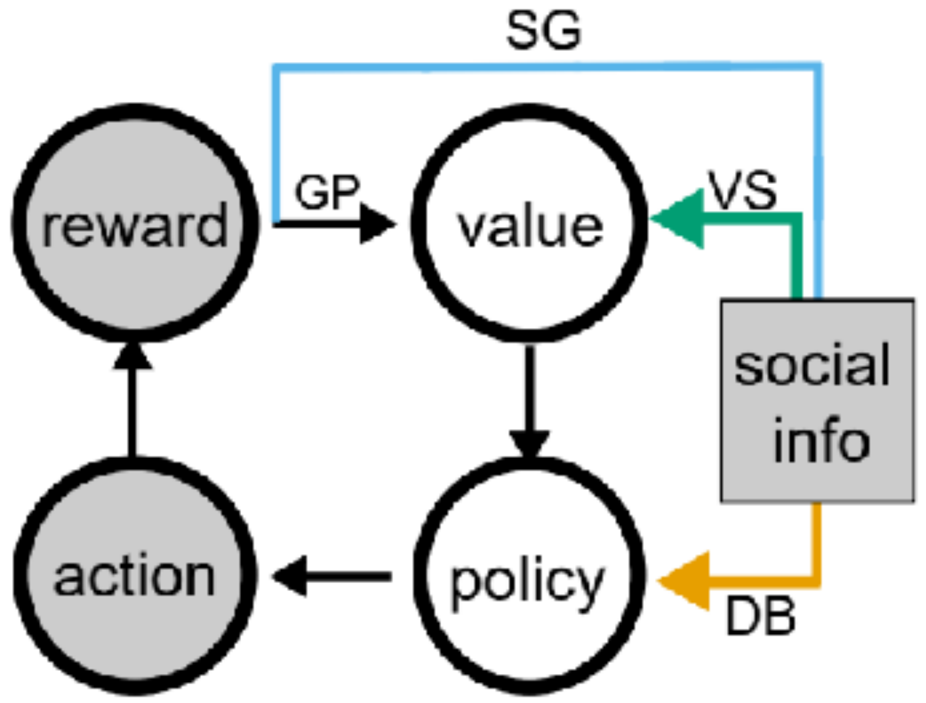



Witt, Toyokawa, Gaissmaier, Lala, & Wu (*Cogsci* 2023)

# Evolutionary simulations

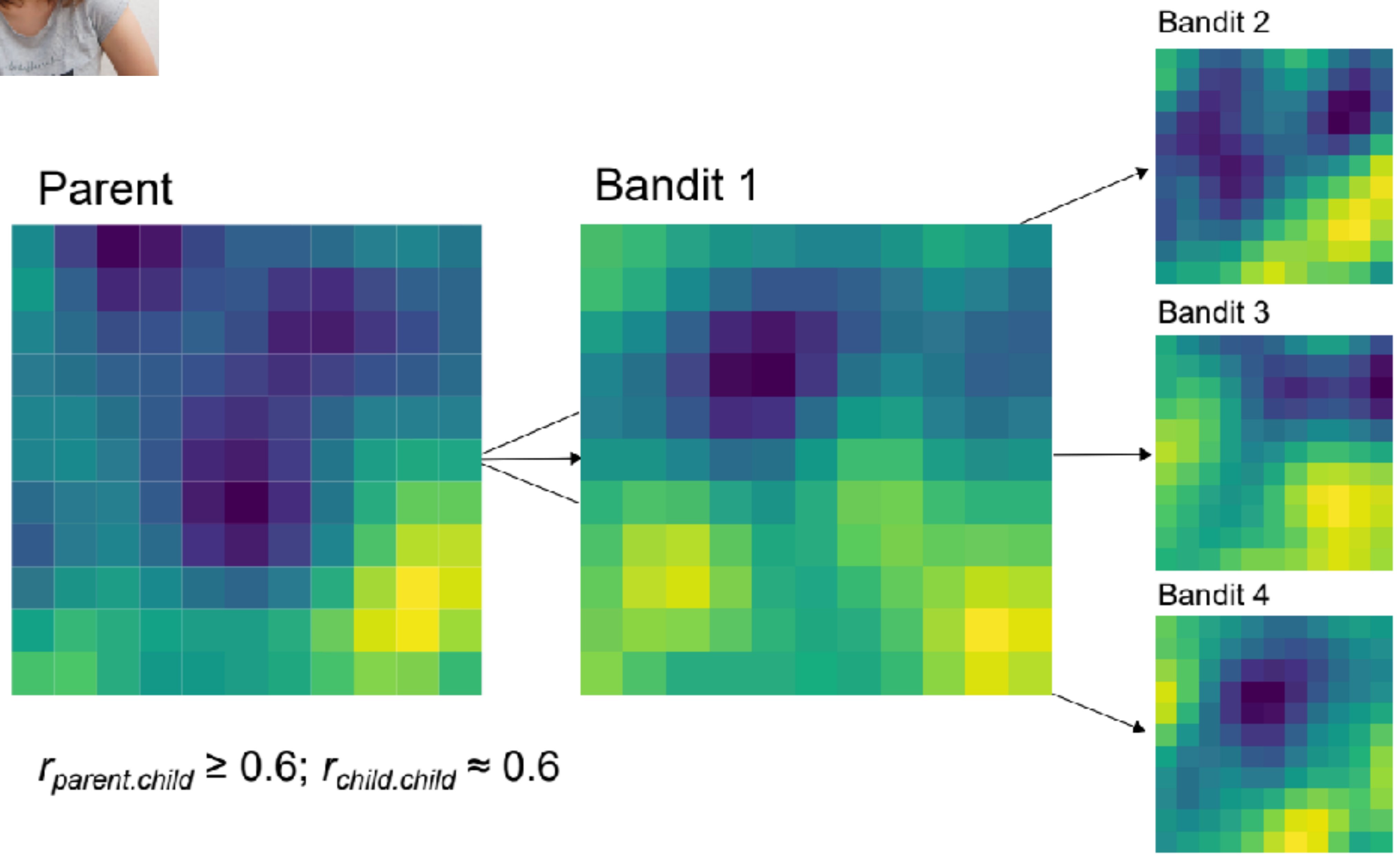



- Social learning despite individual differences (Witt et al., 2023)
  - People can use social information, but not verbatim
  - Exact imitation strategies might fail to account for social differences
- Decision-Biasing (**DB**)
- Value-Shaping (**VS**)
- Social Generalization (**SG**):
  - integration social info in the reward generalization process
  - assume social info is noisier than individual experiences

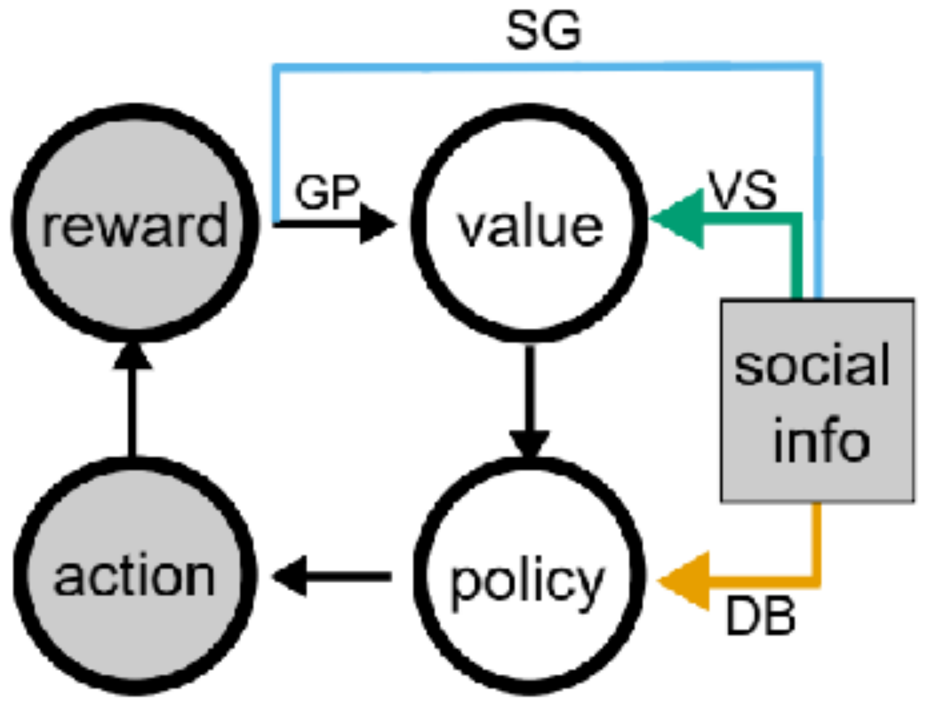


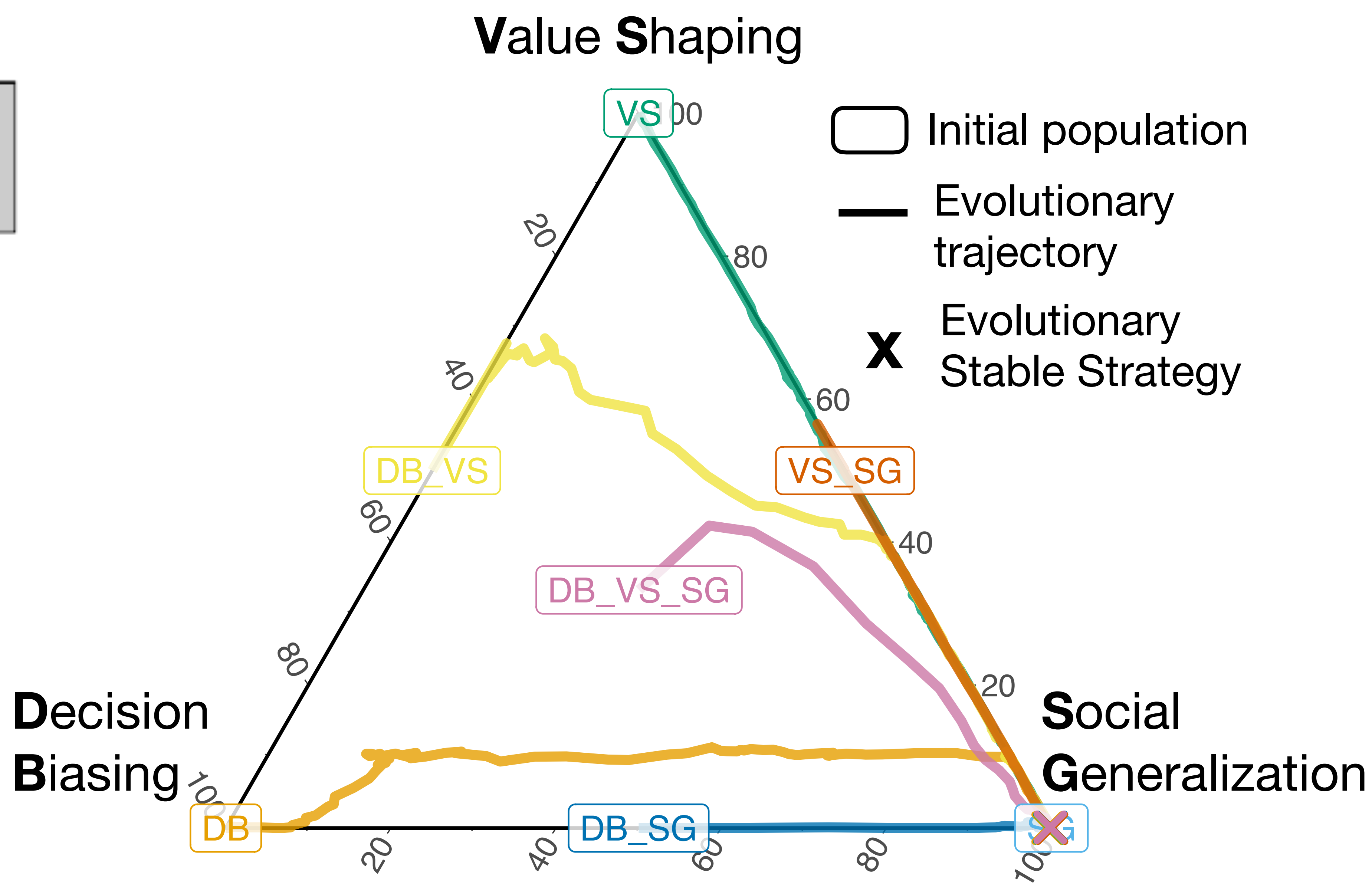
Witt, Toyokawa, Gaissmaier, Lala, & Wu (*Cogsci* 2023)

# Evolutionary simulations




- Social learning despite individual differences (Witt et al., 2023)
  - People can use social information, but not verbatim
  - Exact imitation strategies might fail to account for social differences
- Decision-Biasing (**DB**)
- Value-Shaping (**VS**)
- Social Generalization (**SG**):
  - integration social info in the reward generalization process
  - assume social info is noisier than individual experiences



# Summary and open challenges

- Social learning deploys a range of tools:
  - **imitation**: directly copy observed behaviors
  - **value-shaping:** add a heuristic bonus to observed behaviors
  - **ToM Inference**: inferring hidden value representations or hidden beliefs about the world
- However, this represents only a subset of social learning mechanisms:
  - Intelligent behavior is not only a function of each individual but also how well groups collectively solve problems
  - Over large time scales, simple innovations can cumulatively add up to produce massively complex cultural solutions
  - So far we have focused on observational learning, but social learning also involves pedagogy and explicit communication
- Yet for each mechanism we can describe verbally, we can also define a computational model that makes more precise commitments to the mechanisms of behavior
- Through experimentation and modeling, we can iteratively tweak and refine our understanding of social learning.