

# REINFORCEMENT LEARNING

Francesco Pupillo,  
Tilburg University, Netherlands

A stylized white Greek letter delta ( $\delta$ ) is centered within a rounded square frame that has a glowing blue border.

CIMCYC Workshop  
Computational modelling of  
behavioral data

Granada, 9<sup>th</sup> June 2025

# Workshop structure

- Part 1 – Basic concepts
  - What is a computational model?
  - Why do we need it?
  - Reinforcement learning model
  - Rescorla-Wagner Model – Pavlovian and Instrumental learning
- Part 2 – Model Fitting
  - Parameter recovery
  - Model recovery
  - Model comparison



# Part 1 – Basic Concepts

CIMCYC Workshop  
Computational modelling of  
behavioral data

Granada, 9th June 2025

# What is a computational model?

- It is a mathematical model that defines internal variables
- These unobservable variables are parameterized and change according to the cognitive operations required to solve a task
- E.g., deciding what to eat



# What is a computational model?

- Our choice depends on the value that we assign on each option

$$v^{tapas} = 0.50$$



$$v^{arancini} = 0.50$$



# Why do we need a computational model?

- Instead of relying on verbal theories, which might be vague, computational models allow precise mathematical formulation of the theories and specification of assumptions and their implications.
- Computational models make us think deeply about the variables involved and their relationship
- They allow to compare different models based on different assumptions or theories formally
- They allow to estimate trial-level quantities that are not immediately observable

$$v^{tapas} = 0.50$$



$$v^{arancini} = 0.50$$





# How do we learn these values?

## Trial and error – Values – Action Selection-Prediction error- Value update

- We start from some expectations about the options (**Action values**)
- We compare the expectations of both options and decide on the better (**Action selection**)
- After action selection, we experience the outcome and compare it with our expectations to see whether they have been met or violated (**Prediction error**)
- **We use the prediction error to update expectations and make a better choice in the future**



This looks like a job  
for **reinforcement**  
**learning!**

$$= Q + \alpha \delta$$

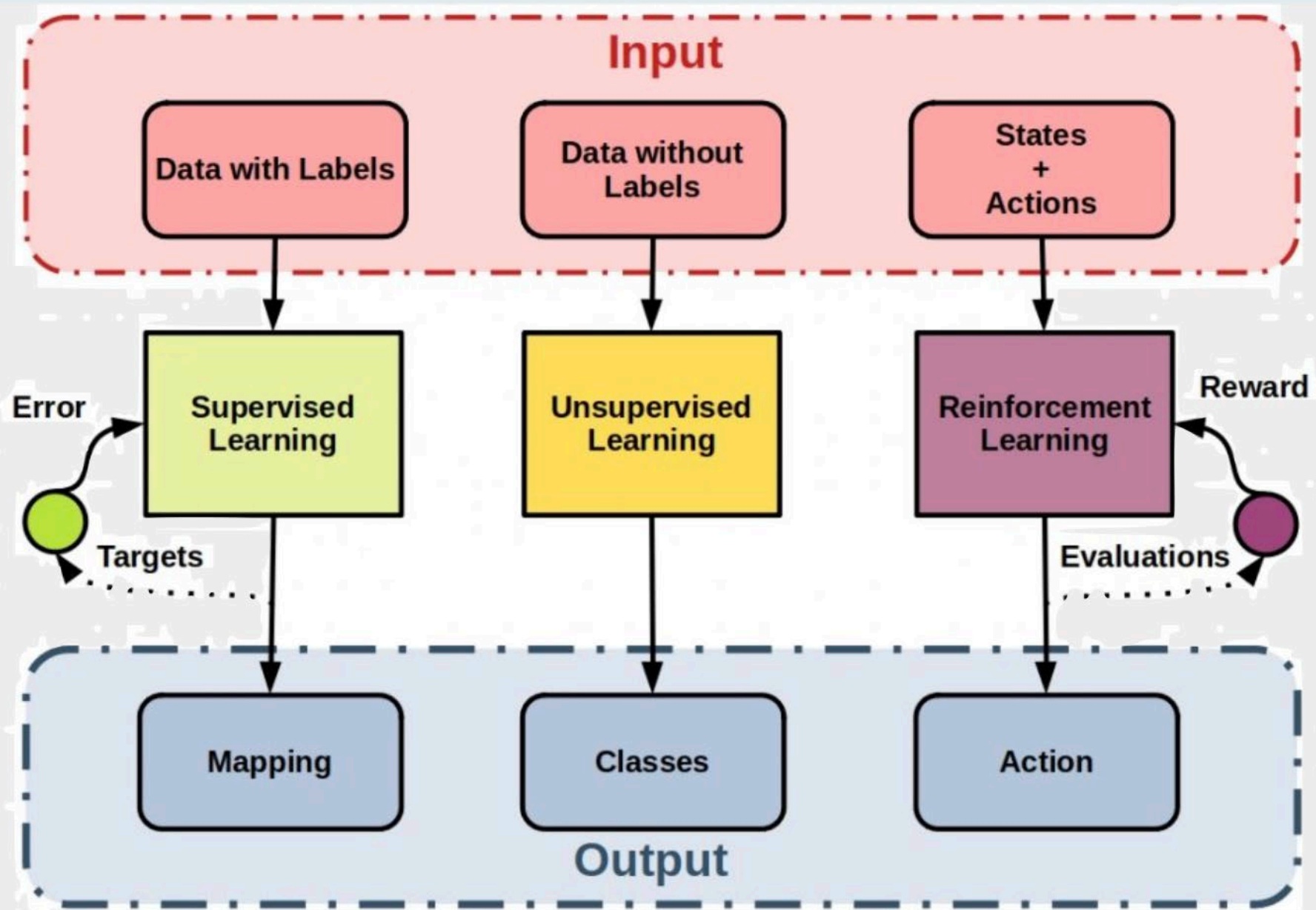
$$V^{tapas} = 0.50$$



$$V^{arancini} = 0.50$$



# Types of Machine Learning





# Types of Machine Learning

## Supervised Learning



These are  
arancinis!

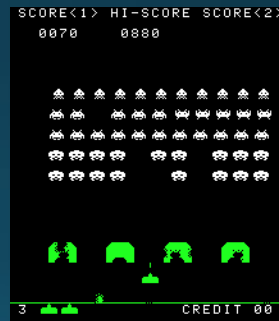
### Image classification

- After being trained with a great amount of data, a function learns to map an input (image) to an output (class)

## Reinforcement Learning



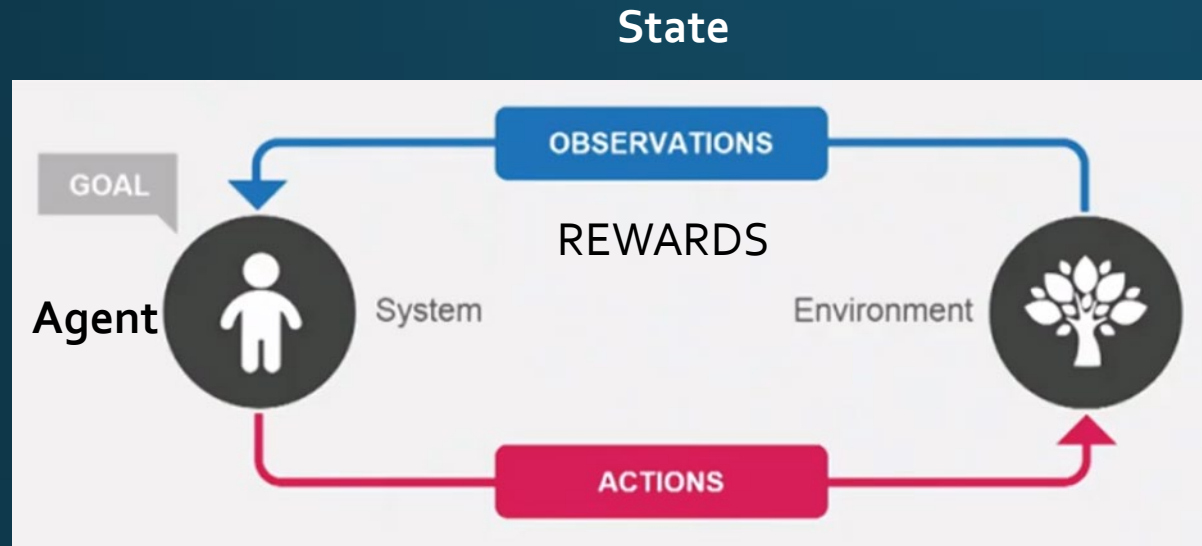
- Learning from experience to pursue goals (maximizing the rewards)
- Interact with the environment
- Breakthrough in artificial intelligence: Playing Atari games (*Mnih et al (2015) Nature*)



and self-driving cars



# Reinforcement Learning



$T = t_1, t_2, t_3, \dots t_n$

Markov decision process:  $(s, a)_{t_1} = (s, r)_{t_2}$

## Agent

The component within an animal or robot that handles reward-based learning

## Reward signals

Produced by reward neurons in the brain

Everything that has “value”



# Reinforcement Learning


## Applications:

- Modeling human behavior in tasks like reversal learning and reveal neurobiological mechanisms (e.g., Daw et al., 2006)
- Trial-by-trial modeling of human learning – extracting latent variables like prediction error (e.g., Pupillo et al., 2022)
- Computational psychiatry – exploring how symptomatology originates by alterations in specific computations (e.g., deficient updating from reward signals in depression – Reilly et al., 2020)
- Language learning (Orpella et al., 2021)
- Confirmation bias (Palminteri et al., 2017)
- Emotion (Rutledge et al., 2014)

- 
- Daw, N. D., O'doherty, J. P., Dayan, P., Seymour, B., & Dolan, R. J. (2006). Cortical substrates for exploratory decisions in humans. *Nature*, 441(7095), 876-879.
  - Orpella, J., Mas-Herrero, E., Ripollés, P., Marco-Pallarés, J., & de Diego-Balaguer, R. (2021). Language statistical learning responds to reinforcement learning principles rooted in the striatum. *PLOS Biology*, 19(9)
  - Palminteri, S., Lefebvre, G., Kilford, E. J., & Blakemore, S. J. (2017). Confirmation bias in human reinforcement learning: Evidence from counterfactual feedback processing. *PLoS Computational Biology*, 13(8), e1005684.
  - Pupillo, F., Ortiz-Tudela, J., Bruckner, R., & Shing, Y. L. (2023). The effect of prediction error on episodic memory encoding is modulated by the outcome of the predictions. *npj Science of Learning*, 8(1), 18.
  - Reilly, Erin E., et al. "Diagnostic and dimensional evaluation of implicit reward learning in social anxiety disorder and major depression." *Depression and anxiety* 37.12 (2020): 1221-1230.
  - Rutledge, R. B., Skandali, N., Dayan, P., & Dolan, R. J. (2014). A computational and neural model of momentary subjective well-being. *Proceedings of the National Academy of Sciences of the United States of America*, 111(33),

# No action: Pavlovian conditioning

Stimulus – stimulus association

e.g. associating a tone  to food



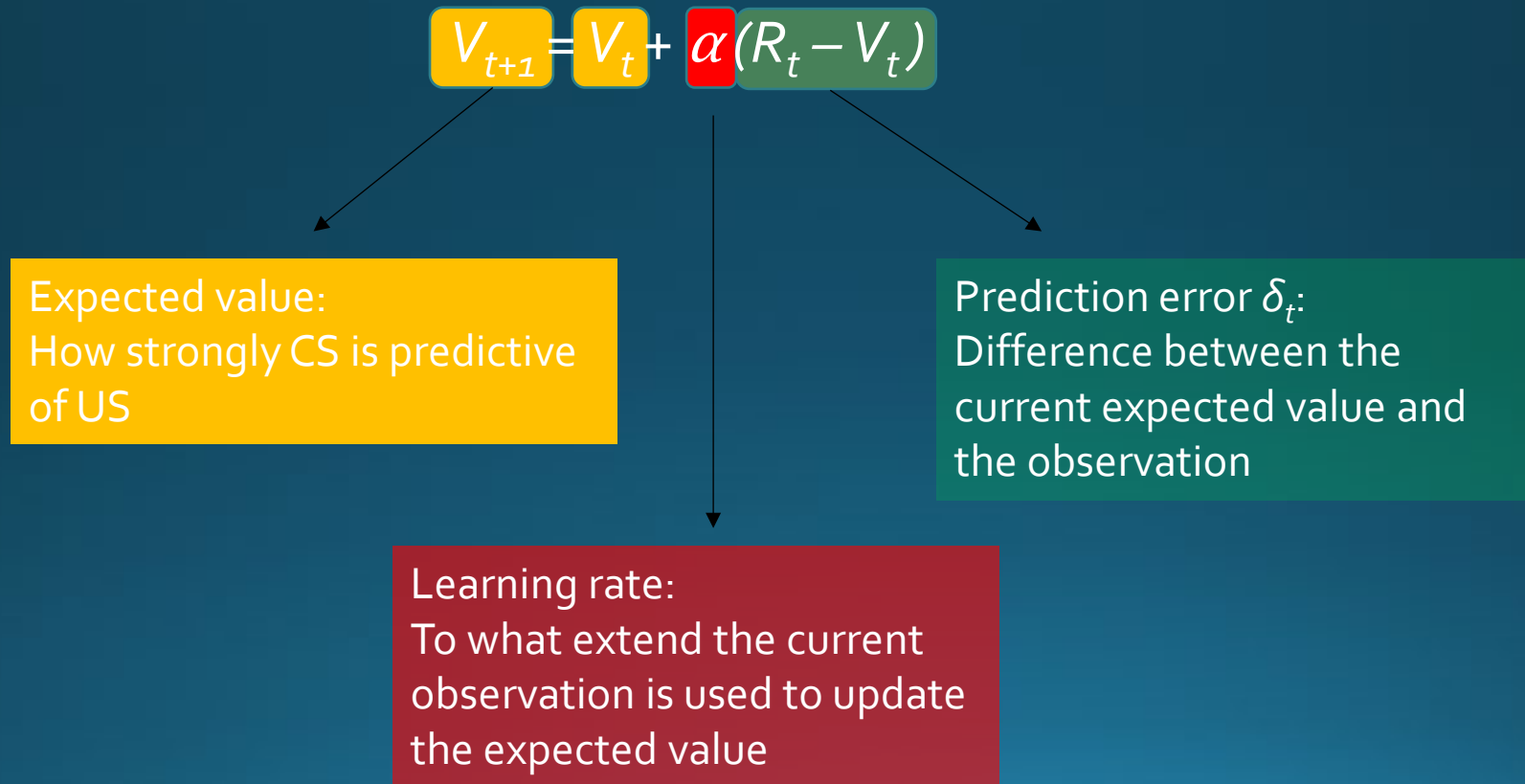
The food is delivered independently of what the agent is doing

The animal responds to expectations formed during learning through a conditioned response (e.g., salivation),  
And increasing the expectations of food

# Rescorla-Wagner model

It started as a simple model of how US expectations were learned

It was developed to explain many phenomena





# Rescorla-Wagner model



**Question:** What is the main implication of this model?  
When does learning occur, and when does it not?

$$V_{t+1} = V_t + \alpha(R_t - V_t)$$

Now Simulate it!

```
``{r, Pavlovian}
```

```
``
```



# Rescorla-Wagner model

Check what happens if we add probabilistic reward

$$V_{t+1} = V_t + \alpha(R_t - V_t)$$

Simulate it with different learning rates

```
```{r, Pavlovian}
```

```
```
```

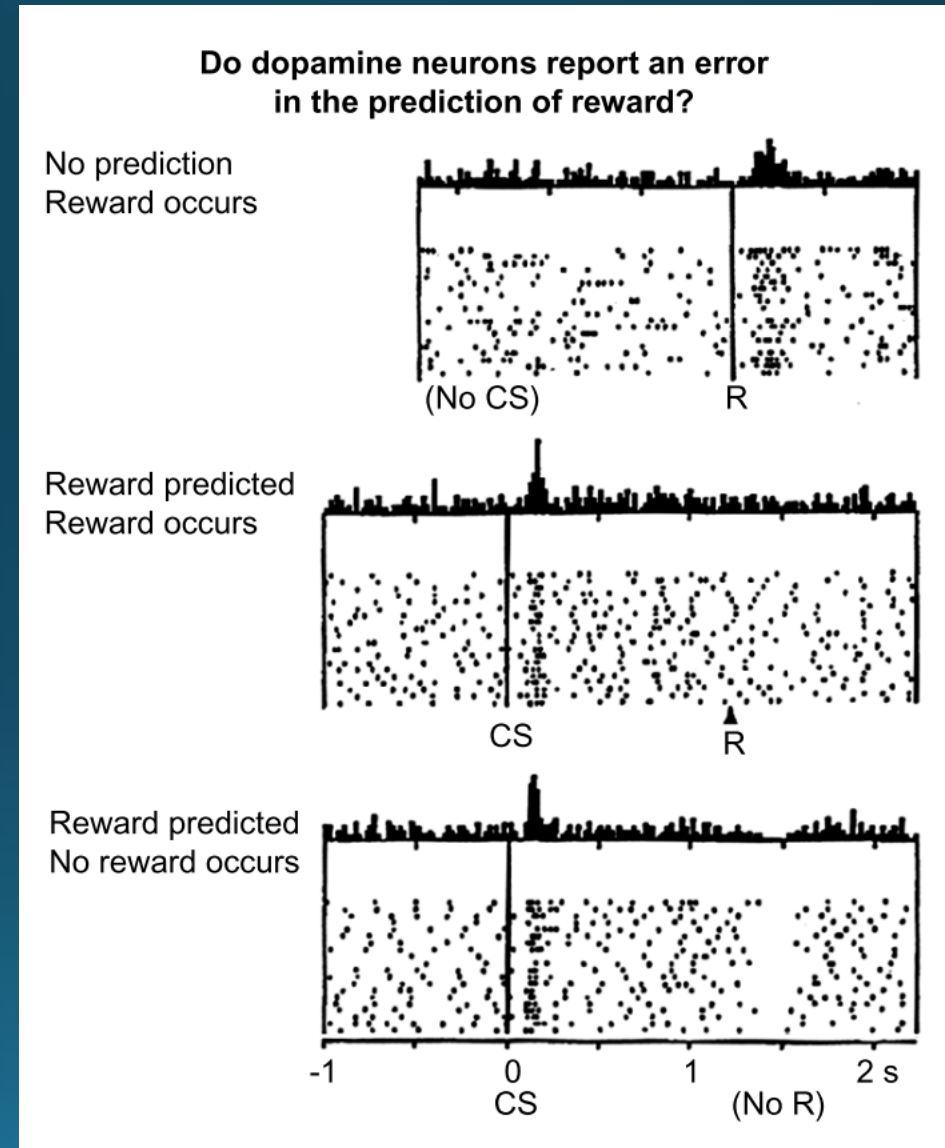
**CODING  
TIME**

# Rescorla-Wagner model

Besides explaining several aspects related to classical conditioning, a Rescorla-Wagner model was used to derived prediction error which explained **firing of dopaminergic neurons in the midbrain** (Schultz, W. et al. (1997), *Science*)

## QUESTION TIME

What would happen if there were no CS – no stimulus predicting the reward – and no reward?



$$V_t = 0$$
$$R_t = 1$$

$$\delta_t = R_t - V_t =$$
$$1 - 0 = 1$$

$$V_t = 1$$
$$R_t = 1$$
$$\delta_t = R_t - V_t =$$
$$1 - 1 = 0$$

$$V_t = 1$$
$$R_t = 0$$
$$\delta_t = R_t - V_t =$$
$$0 - 1 = -1$$

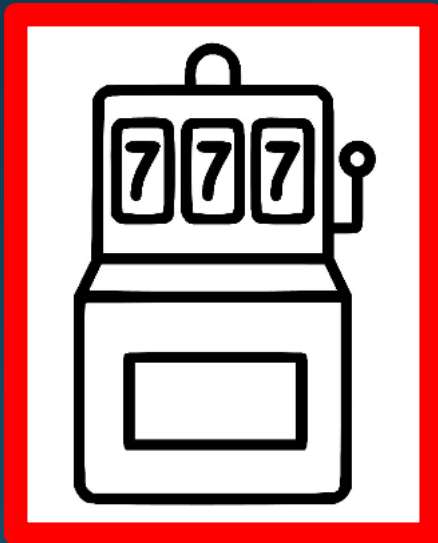
# RL and Marr's Three Levels

| Level               | Description  | RL  |
|---------------------|--|---|
| 1. Computational    | What is the goal of the computation? What problem is being solved?     | Learn associations between stimuli and outcomes (Pavlovian)-<br>Maximize cumulative reward over time (Instrumental) |
| 2. Algorithmic      | What representations and processes are used to solve the problem?      | Value functions, prediction errors, learning rate   |
| 3. Implementational | How is this physically realized in the system (e.g., brain, computer)? | Firing of dopamine circuits in the VTA-striatum   |

# Modeling actions- Instrumental Learning

In instrumental learning, the agent interacts with the environment to learn how to make the best decisions, i.e. the ones that maximize the rewards

Two-armed bandit task



- Two machine with two different reward probabilities (e.g., 70%, 30%)
- The agent learns the reward probabilities by trial and error
- Makes the choice depending on a value functions with the goal of maximizing the reward



# Instrumental Learning

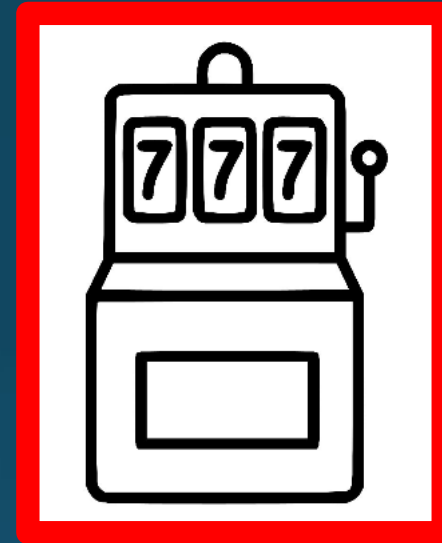
How does the value function look like?

$$Q^i_{t+1} = Q^i_t + \alpha(R_t - Q^i_t)$$

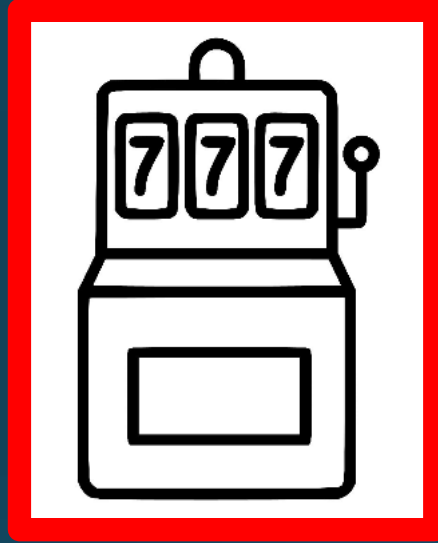
$$Q^{red}_{t+1} = Q^{red}_t + \alpha(R_t - Q^{red}_t)$$

$$Q^{yellow}_{t+1} = Q^{yellow}_t + \alpha(R_t - Q^{yellow}_t)$$

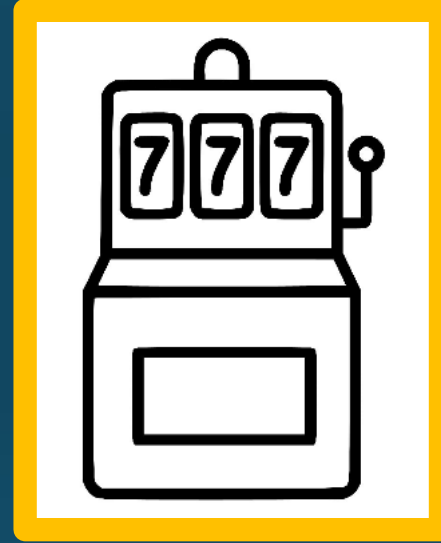
Two-armed bandit task



# Instrumental Learning



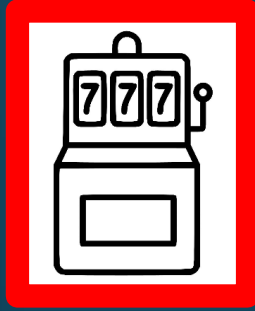
$Q = 0.51$



$Q = 0.49$

*How should the agent choose?*

# Instrumental Learning



$Q = 0.51$



$Q = 0.49$

1. Always pick the choice with the highest value (exploitation)

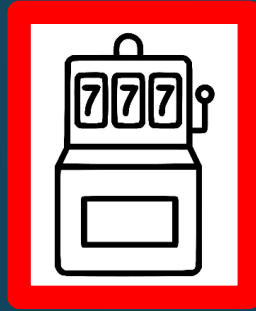
```
``{r, greedy}
```

```
...
```

$$p_t^i = \begin{cases} 1, & \text{if } Q_t^i = \operatorname{argmax} Q_t, \\ 0, & \text{otherwise} \end{cases}$$

**CODING  
TIME**

# Instrumental Learning



$Q = 0.51$



$Q = 0.49$

*2. Always explore*

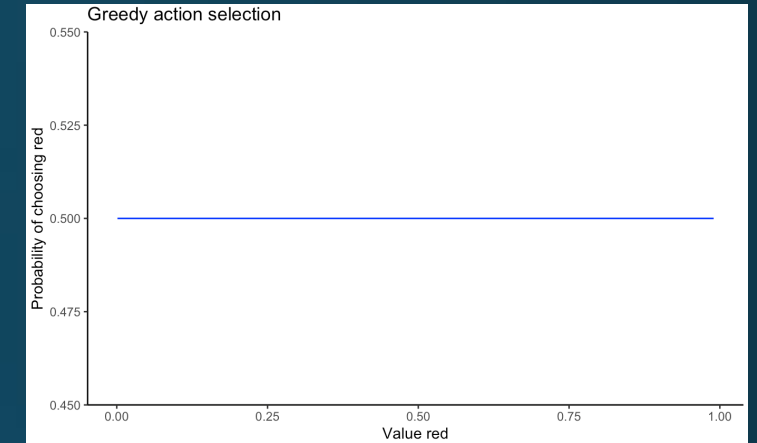
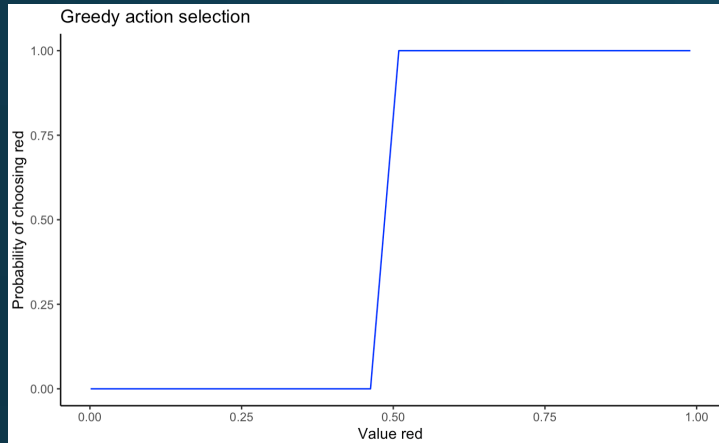
```
```${r, explore}  
...`
```

$$p_t^i = 1/k$$

$k$  = number of choices

**CODING  
TIME**

# Instrumental Learning



Exploitation vs Exploration dilemma

We can leave it as a free parameter!

(question: can you think of another parameter we have already talked about that can be considered as "free"?)



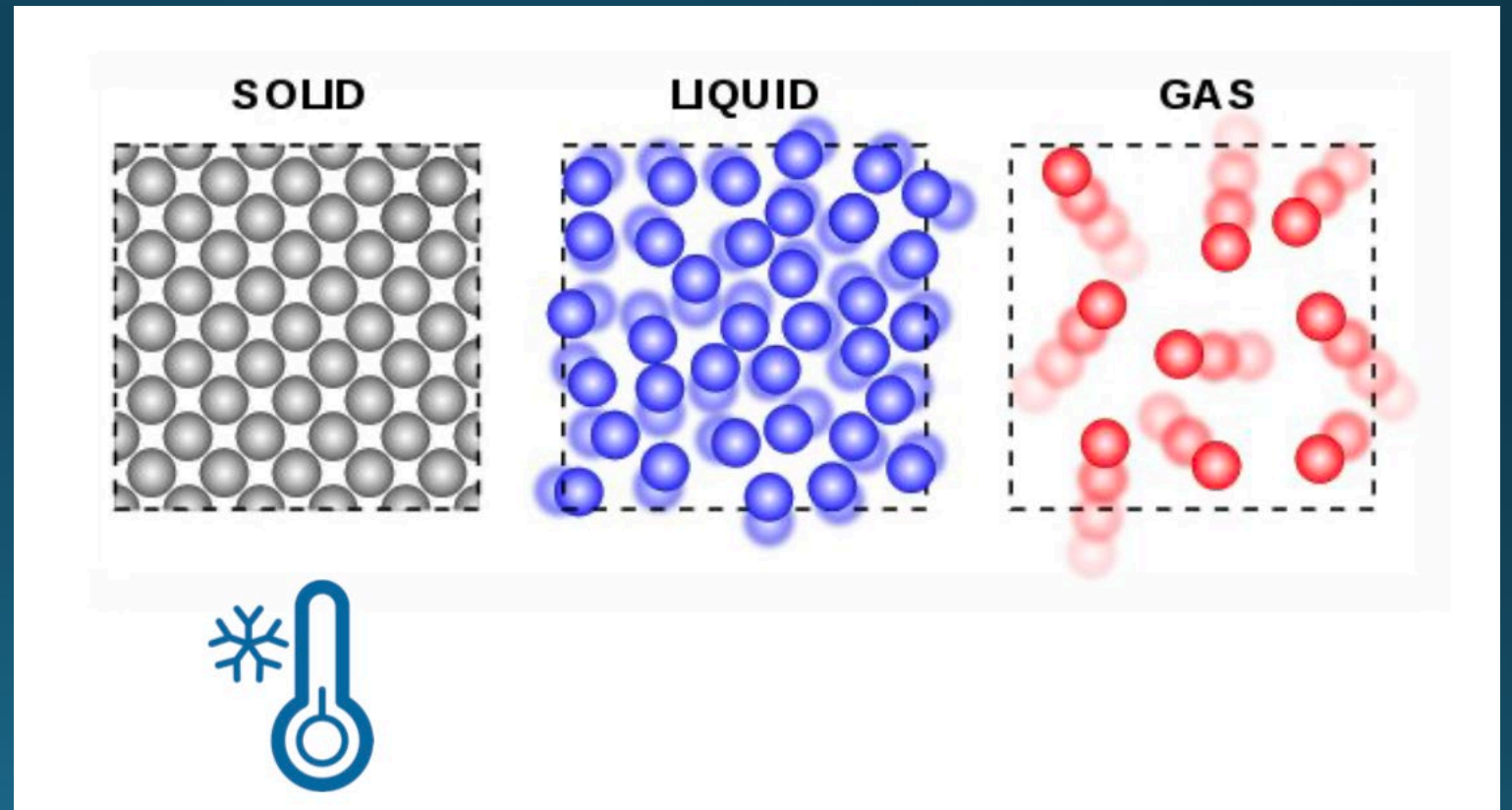


# Instrumental Learning

$\tau$  = temperature parameter

Low temperature

- Choices are less noisy
- More affected by value
- More deterministic

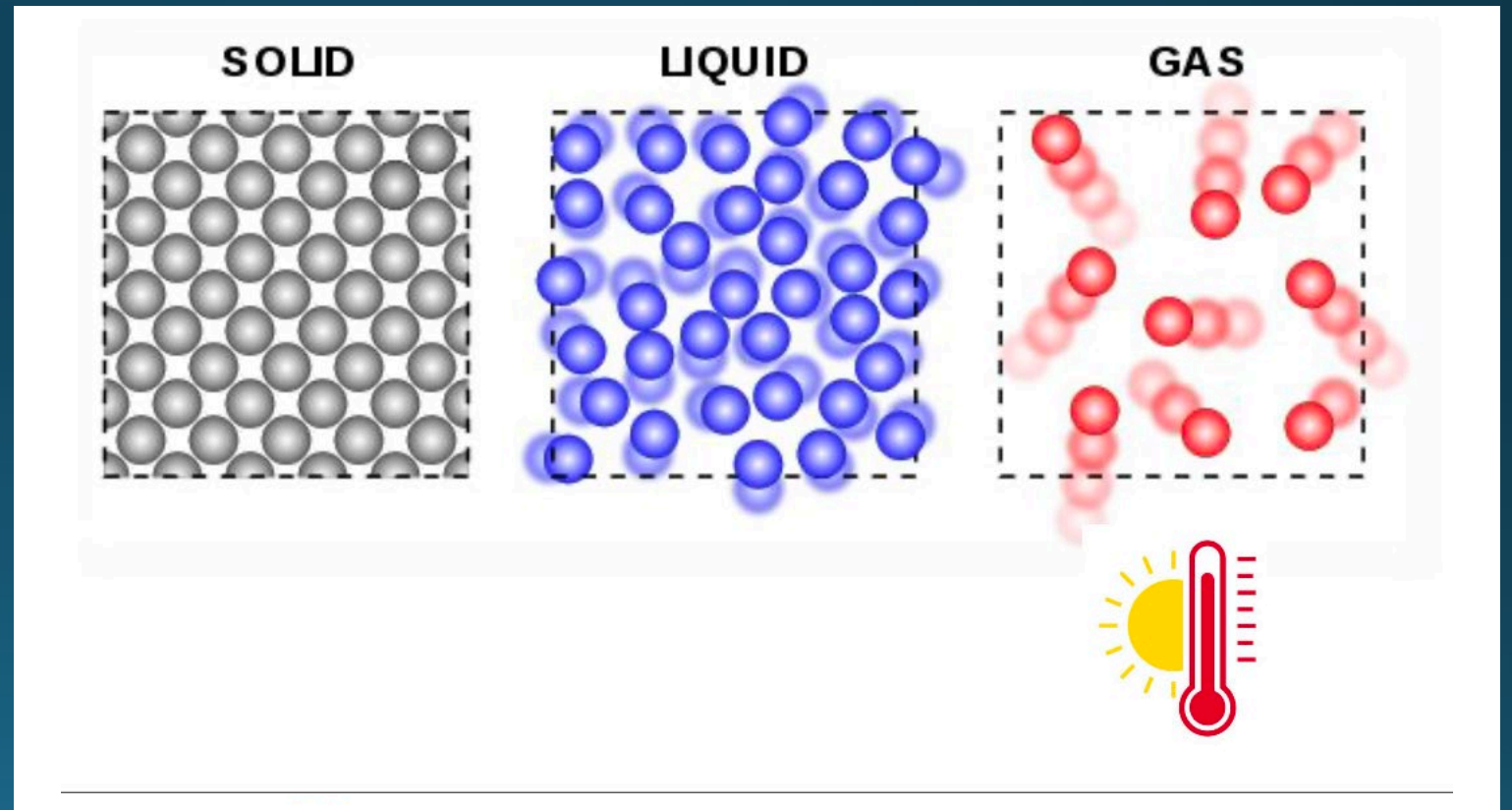


# Instrumental Learning

$\tau$  = temperature parameter

High temperature

- Choices are more noisy
- Less affected by value
- Less deterministic (more stochastic)



# Instrumental Learning

Inverse Temperature :  $\beta = 1/\tau$

Used in a **softmax** function,  $\beta$  determines the extent to which value estimates influence choice behaviour.

$$p_t^i = \frac{\exp(\beta Q_t^i)}{\sum_{i=1}^k \exp(\beta Q_t^i)}$$

The higher the  $\beta$ , the more deterministic the choice will be.

Let's try to understand it better!

And it also normalizes the Qs

**CODING  
TIME**

```
``{r, softmax}
```

```
...
```

# Instrumental Learning

Let's simulate our first instrumental model!

```
``{r, instrumental simulate}  
``
```

Try to simulate for different parameters

End of the first part!



# Part 2 – Model Fitting

CIMCYC Workshop  
Computational modelling of  
behavioral data

Granada, 9th June 2025



# Model Fitting

What we created is a **Generative Model**

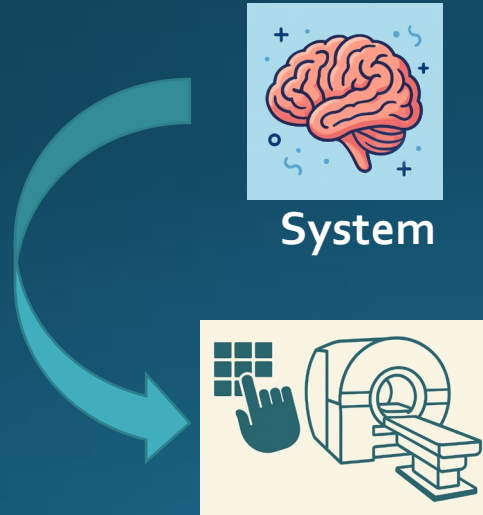
$$Q_{t+1}^i = Q_t^i + \alpha(R_t - Q_t^i)$$

$$p_t^i = \frac{\exp(\beta Q_t^i)}{\sum_{i=1}^k \exp(\beta Q_t^i)}$$



$$\theta = \{ \alpha, \beta \}$$

$$P(D|\theta, M)$$



# Model Fitting

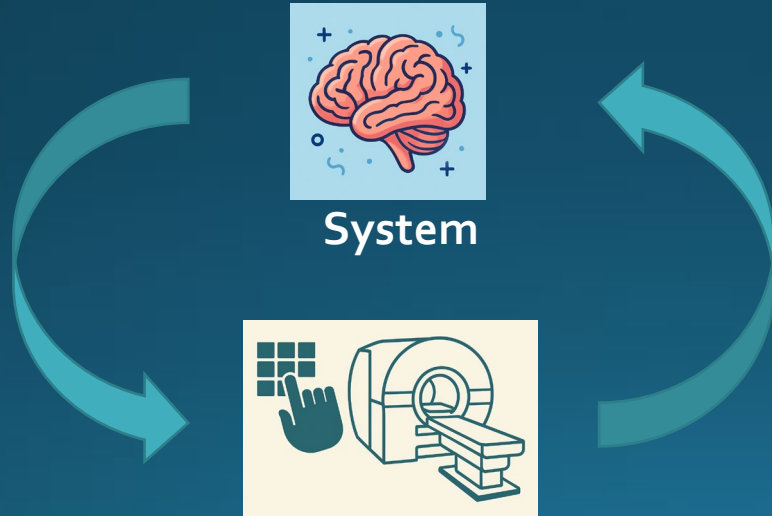
What we created is a **Generative Model**

$$Q_{t+1}^i = Q_t^i + \alpha(R_t - Q_t^i)$$

$$p_t^i = \frac{\exp(\beta Q_t^i)}{\sum_{i=1}^k \exp(\beta Q_t^i)}$$

$$\theta = \{ \alpha, \beta \}$$

$$P(D|\theta, M)$$



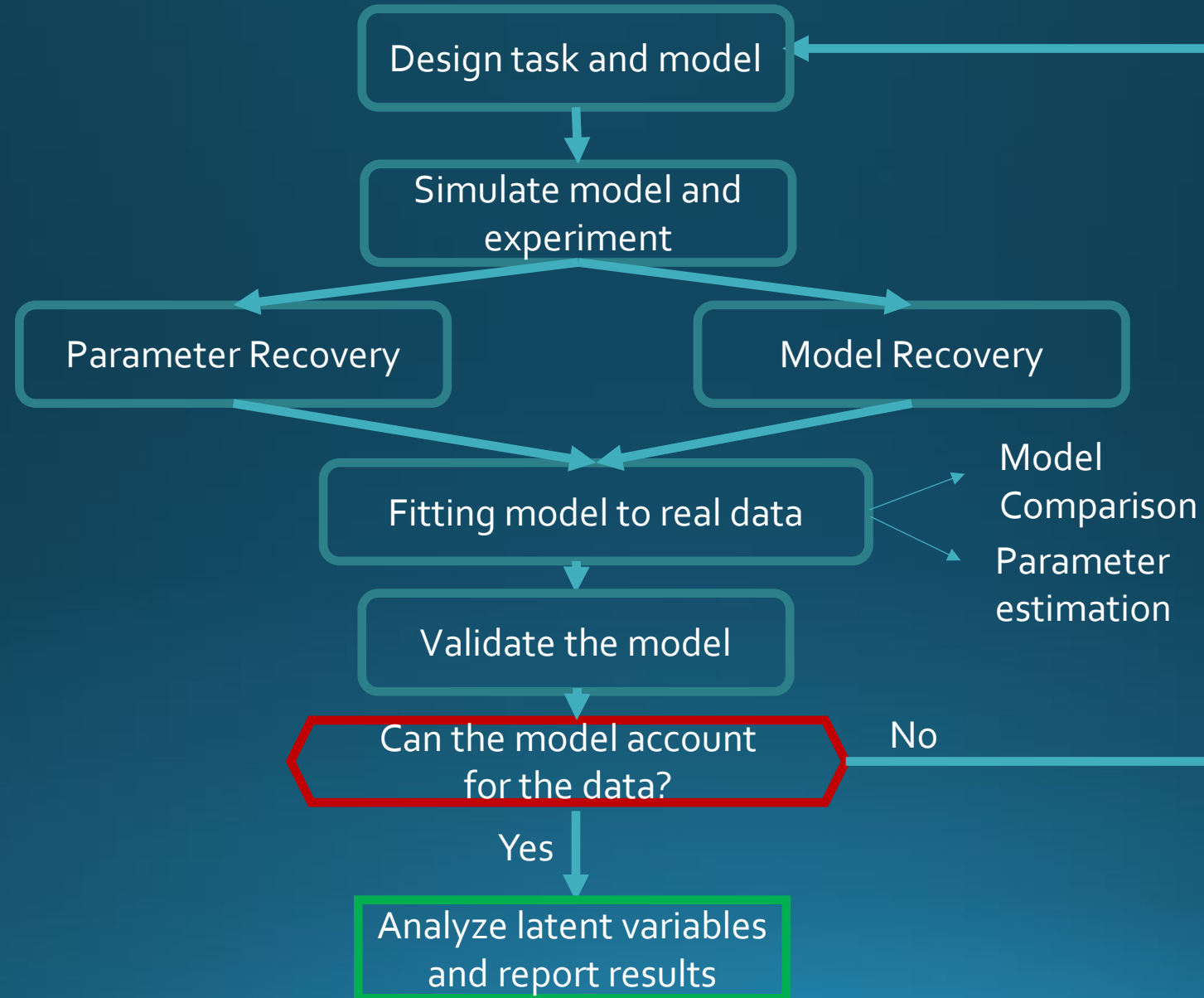
*Model Inversion*

$$P(\theta|D, M)$$

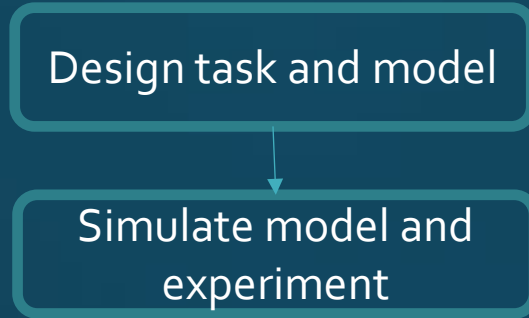
# Model Fitting

Fitting model to real data

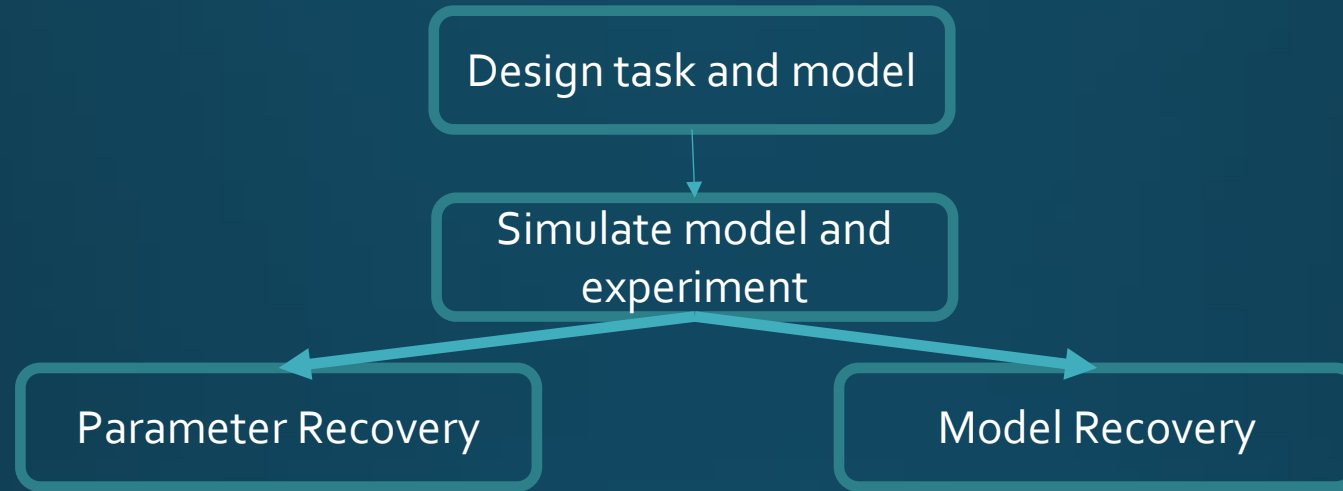
# Model Fitting



# Model Fitting



# Model Fitting





# Parameter Recovery

in order to check whether the fitting procedure for each model gave meaningful parameters.

1. Simulate data with parameters randomly sampled from our parameter space.
2. Fit the model to the simulated data.
3. Compared parameters used to simulate the data to the fitted data

Run the two sections now, as they take quite some time

```
```{r, parameter recovery}
```

```
```
```

# Parameter Recovery

We know how to simulate parameters, but how can we actually fit the model to data?

When we fit a model to data we are estimating the parameters that maximize the likelihood of observing the data.

This is called **Maximum Likelihood estimation**

$p(c_t | d_{1:t-1}, \theta, m)$  = *likelihood of observing the data given the parameters and the model*

This is estimated as the product of the probabilities, and it is thus usually a small number

To make it more tractable, the logarithm is taken - Log Likelihood

$$LL = \sum_{t=1}^n \log p(c_t | d_{1:t-1}, \theta, m)$$

The *optim* function from R uses optimization algorithms to find the values that minimize each functions:  
As it looks for a minimum, we need to feed it the Negative Log Likelihood

# Parameter Recovery

We know how to simulate parameters, but how can we actually fit the model to data?

When we fit a model to data we are estimating the parameters that maximize the likelihood of observing the data.

This is called **Maximum Likelihood estimation**

$p(c_t | d_{1:t-1}, \theta, m)$  = *likelihood of observing the data given the parameters and the model*

This is estimated as the product of the probabilities, and it is thus usually a small number

To make it more tractable, the logarithm is taken - Log Likelihood

$$LL = \sum_{t=1}^n \log p(c_t | d_{1:t-1}, \theta, m)$$



The *optim* function from R uses optimization algorithms to find the values that minimize each functions:  
As it looks for a minimum, we need to feed it the Negative Log Likelihood

# Parameter Recovery

```
likelihood_RW_instr<-function( df, alpha, beta, out){
```

Alpha = 0.1  
Beta = 0.3

| t | Choice | p    | Log(p) |
|---|--------|------|--------|
| 1 | Yellow | 0.50 | -0.69  |
| 2 | Red    | 0.50 | -0.69  |
| 3 | Red    | 0.51 | -0.67  |
| 4 | Red    | 0.51 | -0.67  |
| 5 | Red    | 0.52 | -0.65  |

*Likel* = 0.03  
*-LL* = 2.02

Alpha = 0.5  
Beta = 4

| t | Choice | p    | Log(p) |
|---|--------|------|--------|
| 1 | Yellow | 0.50 | -0.69  |
| 2 | Red    | 0.73 | -0.31  |
| 3 | Red    | 0.88 | -0.13  |
| 4 | Red    | 0.92 | -0.08  |
| 5 | Red    | 0.94 | -0.06  |

*Likel* = 0.28  
*-LL* = 1.27

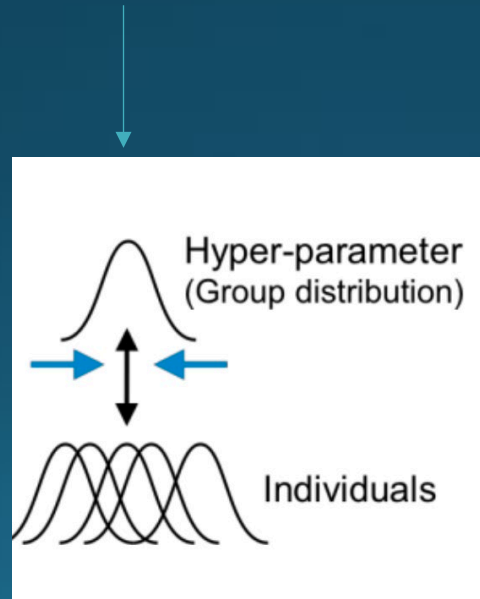


# Parameter Recovery

Alternative to Maximum Likelihood: Maximum a Posteriori estimation, Hierarchical Bayesian Estimation

$p(c_t | d_{1:t-1}, \theta, m)$  = likelihood of observing the data given the parameters and the model

$$P(\theta, m | c_t, d_{1:t-1}) = p(c_t | d_{1:t-1}, \theta, m) * p(\theta, m)$$



e.g.:

Brown et al. (2021). Reinforcement learning disruptions in individuals with depression and sensitivity to symptom change following cognitive behavioral therapy. *JAMA psychiatry*, 78(10), 1113-1122.

# Parameter Recovery

```
```{r, parameter recovery}
```

```
```
```

**QUESTION  
TIME**



**Question:** Can the parameters be recovered?

# Model Recovery

We can show that the model can successfully recover the parameters.

But can the model distinguish between different models that might have generated the data?

In order to answer this question, we need to check whether the model can successfully recover the model that generated the data.

1. Simulate data with different models from our model space.
2. Fit all the models to the simulated data.
3. Compare the fit of each models

Ideally, the model that generated the data should have the best fit compared to the others



# Model Recovery

But how can we compare the fit of different models?

In model comparison, our goal is to figure out which model of a set of possible models is most likely to have generated the data.

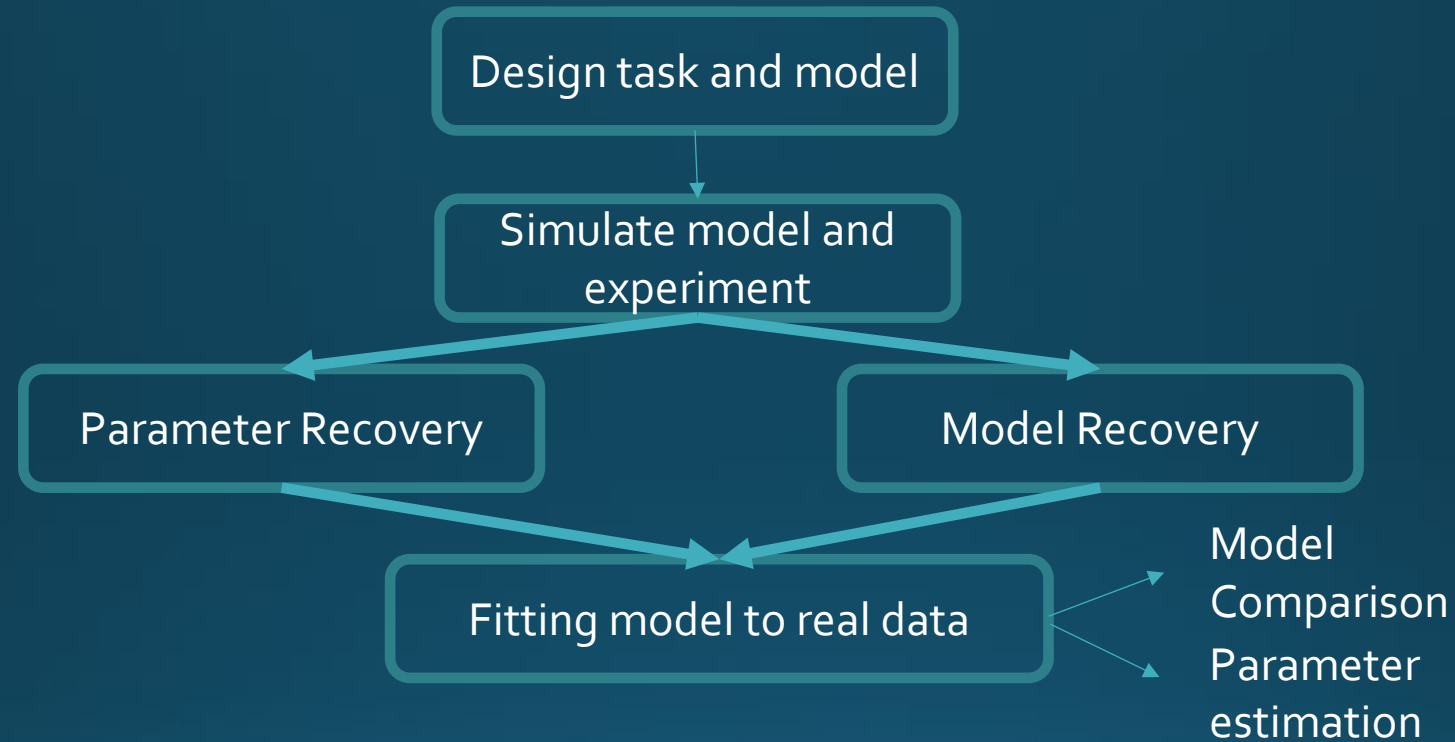
The method commonly used is the Bayesian Information Criterion :

$$BIC = -2\log\hat{L} + k_m\log(T)$$

# Model Recovery



# Model Fitting



# Model Fit

Just like what we did in parameter recovery –

When we fit model to data we are estimating the parameters that maximize the likelihood of observing those data.

``{r, Model fit one participant}

...

``{r, Model fit all participant}

...



Which model is the best one?

# Model Comparison

After calculating the BIC, we can count the number of participants for which each model was the best fit.

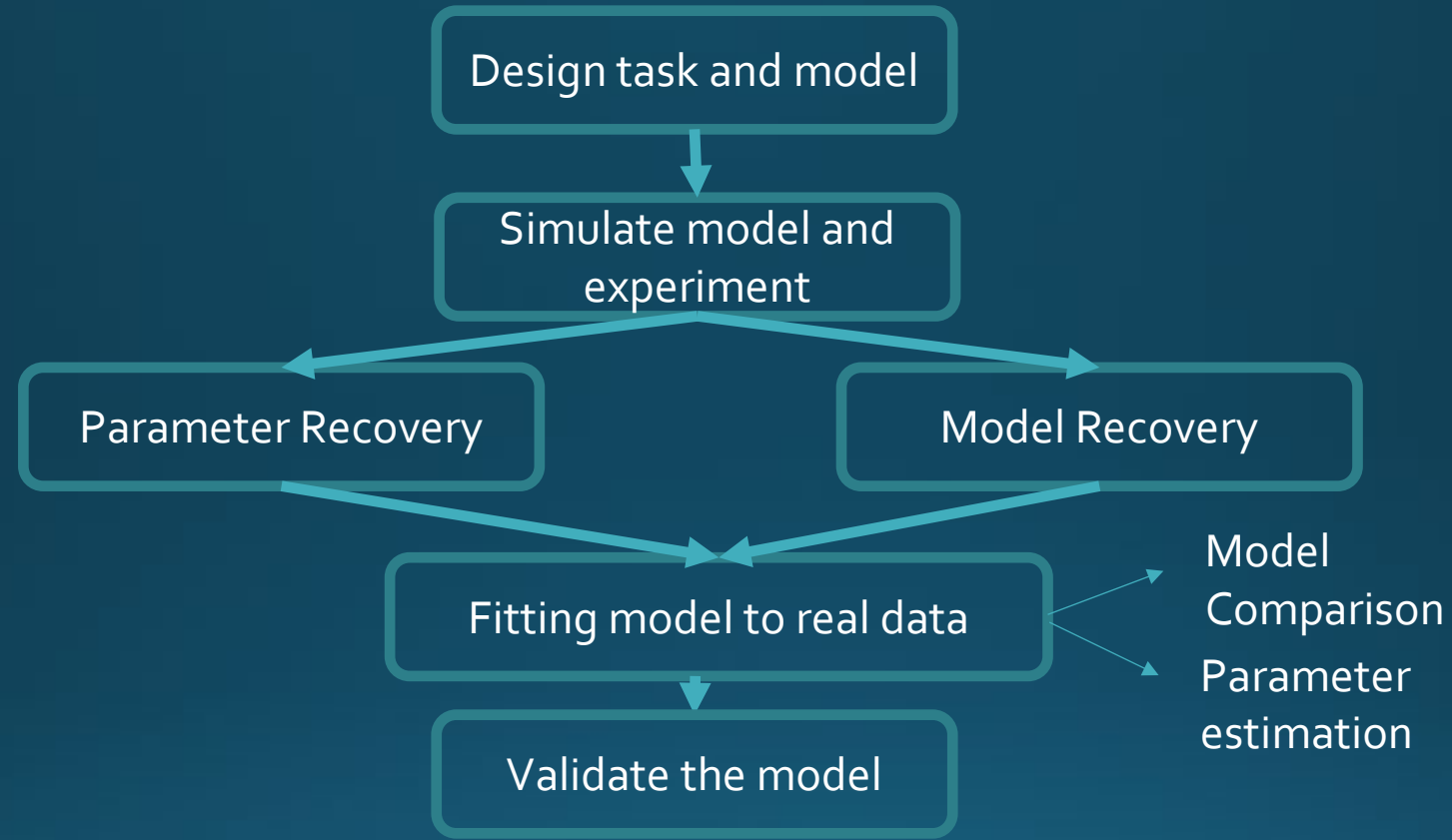
In addition, We can calculate the “model evidence”, following Gluth et al. (2017), depending on the difference between the best and the second-best model for each participant.

```{r, Model comparison}

```

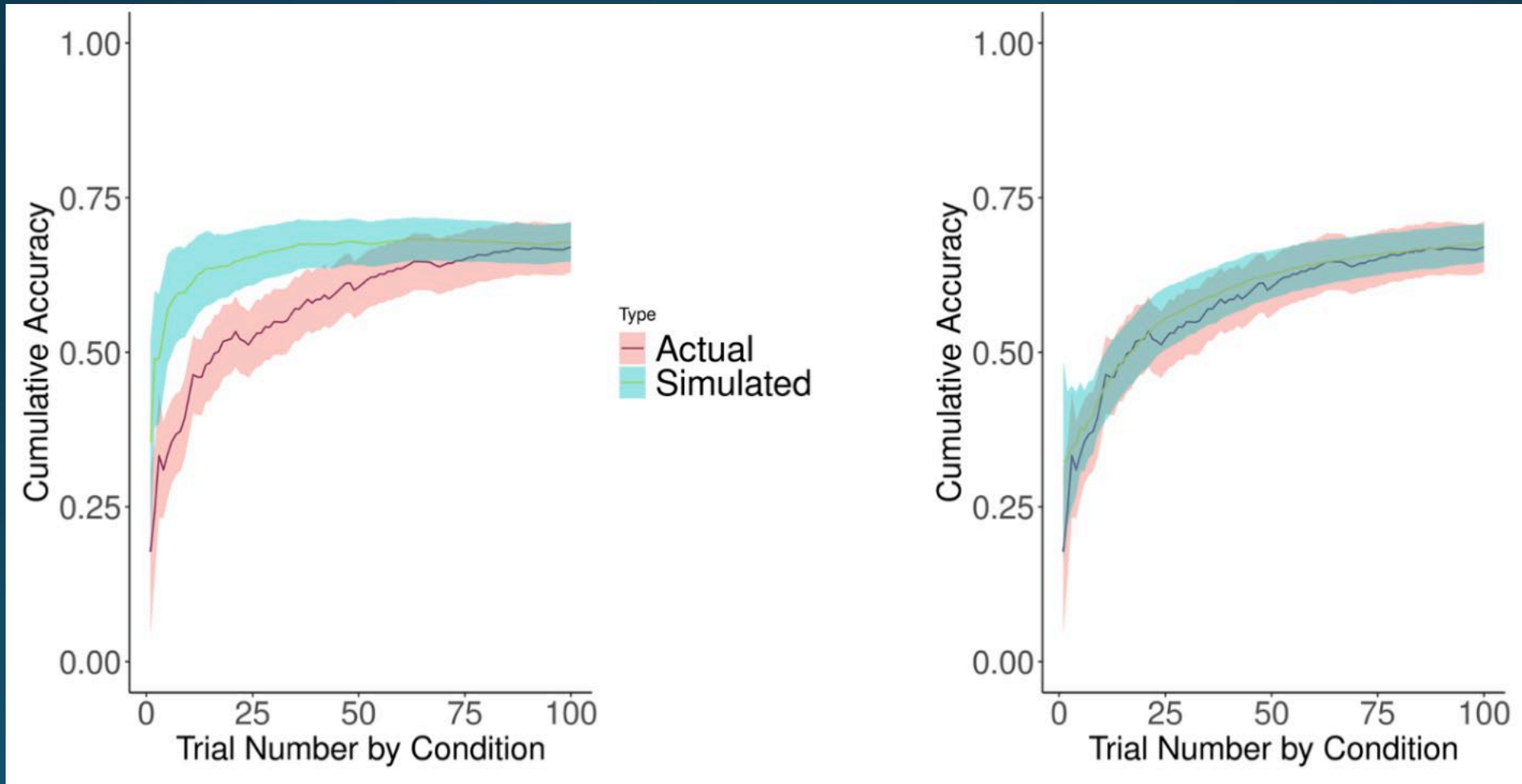
Gluth, S., Hotaling, J. M., and Rieskamp, J. (2017). The attraction effect modulates reward prediction errors and intertemporal choices. *Journal of Neuroscience*, 37(2):371–382.

# Model Fitting



# Validate the model

Validating the model means to check whether data simulated by the model, with the parameters of best fit, replicate pattern observed in the empirical data (posterior predictive check)





# Variants of RL

- Pos vs negative learning rate

$$Q_t = \begin{cases} Q_t + \alpha^+ \delta_t, & \text{if } \delta_t > 0 \\ Q_t + \alpha^- \delta_t, & \text{if } \delta_t < 0 \end{cases}$$

- Factual vs counterfactual updating

$$Q_t^i = \begin{cases} Q_t^i + \alpha^c \delta_t, & \text{if } i = c_t \\ Q_t^i + \alpha^u \delta_t, & \text{otherwise} \end{cases}$$

- Updating vs reward sensitivity

$$Q_{t+1} = Q_t + \alpha(\rho R_t - Q_t)$$

- Dynamic learning rate – Pearce hall model

$$\alpha_{t+1} = \gamma |\delta_t| + (1 - \gamma) \alpha_t$$

# Suggested Readings

- Sutton, R. S., & Barto, A. G. (2018). Reinforcement Learning: An Introduction, Second Edition. In *The Lancet* (Vol. 258, Issue 6685). [https://doi.org/10.1016/S0140-6736\(51\)92942-X](https://doi.org/10.1016/S0140-6736(51)92942-X)
- Daw, N. D. (2011). Trial-by-trial data analysis using computational models. *Decision Making, Affect, and Learning: Attention and Performance XXIII*, 1–26. <https://doi.org/10.1093/acprof:oso/9780199600434.003.0001>
- Daw, N. D., & Tobler, P. N. (2013). Value Learning through Reinforcement: The Basics of Dopamine and Reinforcement Learning. *Neuroeconomics: Decision Making and the Brain: Second Edition*, 283–298. <https://doi.org/10.1016/B978-0-12-416008-8.00015-2>
- Wilson, R. C., & Collins, A. G. E. (2019). Ten simple rules for the computational modeling of behavioral data. *ELife*, 8, 1–33. <https://doi.org/10.7554/eLife.49547>

# Complete the exercises

## 1. Basic functions

Simulate the probabilistic pavlovian model at different learning rates – plot the values

## 2. Model Fitting

- Improve parameter recovery by setting different boundaries

## 3. Exercise: Double update model

- Change a pre-existing script to simulate a double update model and plot its behavior

Thank you!