

REINFORCEMENT LEARNING

Francesco Pupillo,
Goethe University Frankfurt

A stylized Greek letter delta (δ) is centered within a black rectangular border.

**CIMCYC Workshop Computational
modelling of behavioral data**

Granada, 2nd June 2022

Part 1 – Basic Concepts

**CIMCYC Workshop Computational
modelling of behavioral data**

Granada, 2nd June 2022

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

$$v_{pasta} = 0.50$$



$$v_{paella} = 0.50$$



Why do we need a computational model?

- Most concepts of cognition rely on verbal theories, on analogies and metaphor and are thus imprecise. 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 formally compare different models based on different assumptions or theories
- They allow to estimate trial-level quantities that are not immediately observable

$$v_{pasta} = 0.50$$



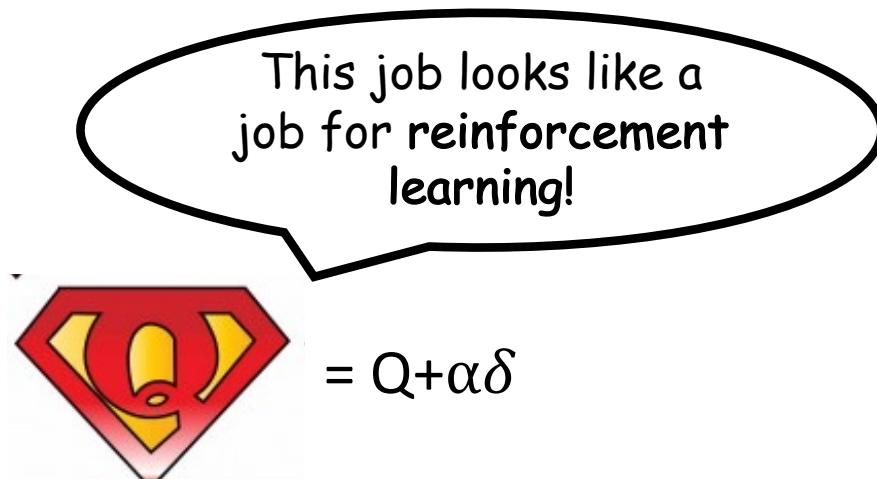
$$v_{paella} = 0.50$$



How do we learn these values?

Trial and error

- We start from some expectations about the options
- We compare the expectations of both options (values) and decide for the better (**Action selection**)
- **Prediction error**: after action selection, we experience the outcome and compare it with our expectations to see whether they have been met
- **Update values/learning**: we use expectation violation (prediction error) to update expectations and make a better choice in the future



$$V_{pasta} = 0.50$$



$$V_{paella} = 0.50$$



What is Reinforcement Learning?

Engineering

A machine learning technique that is different from both supervised and unsupervised learning

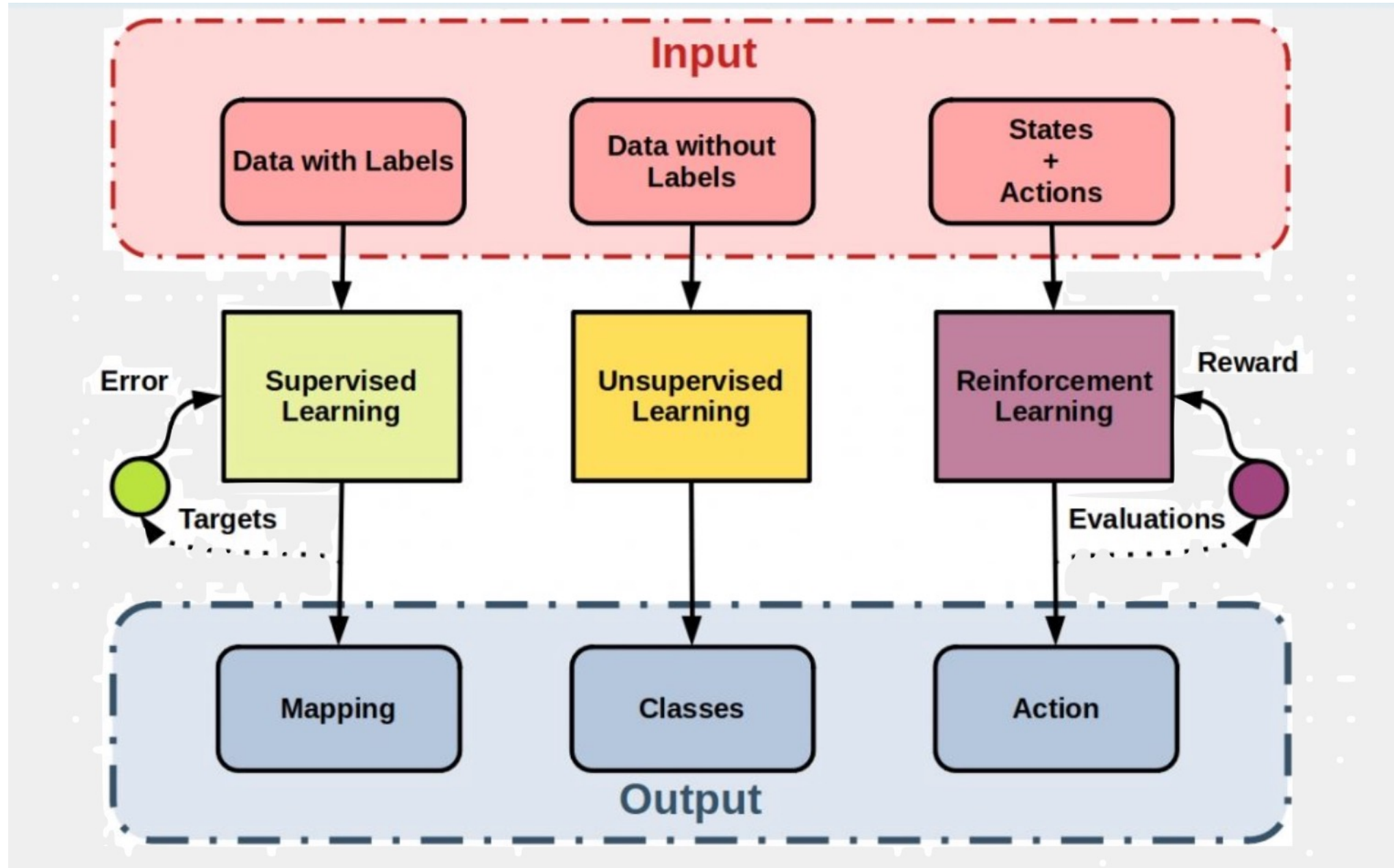
Psychology

Incremental learning, pavlovian-learning, trial-and-error learning; action selection based on evaluative feedback

Neuroscience

A formula to derive prediction errors and link it to the activation of single neurons and brain areas (e.g., VTA, striatum)

Types of Machine Learning



Types of Machine Learning

Supervised Learning



Pasta

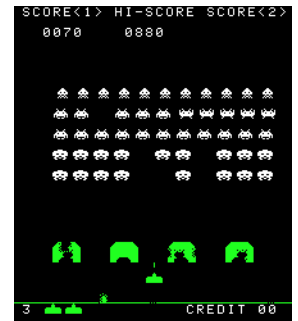
Image classification

- After being trained with a great amount of data, a function learn 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



Types of Reinforcement Learning

Different types of reinforcement learning can be distinguished depending on whether they include **Actions** and **States**

Models that assume no state in the environment

No action: Pavlovian conditioning

Action: Bandit task, instrumental conditioning, reversal learning

Models that assume states in the environment



e.g, Hidden Markov Models



Not covered in this workshop!

No action: Pavlovian conditioning

Stimulus – stimulus association

e.g. associating a tone  (CS) to food  (US)

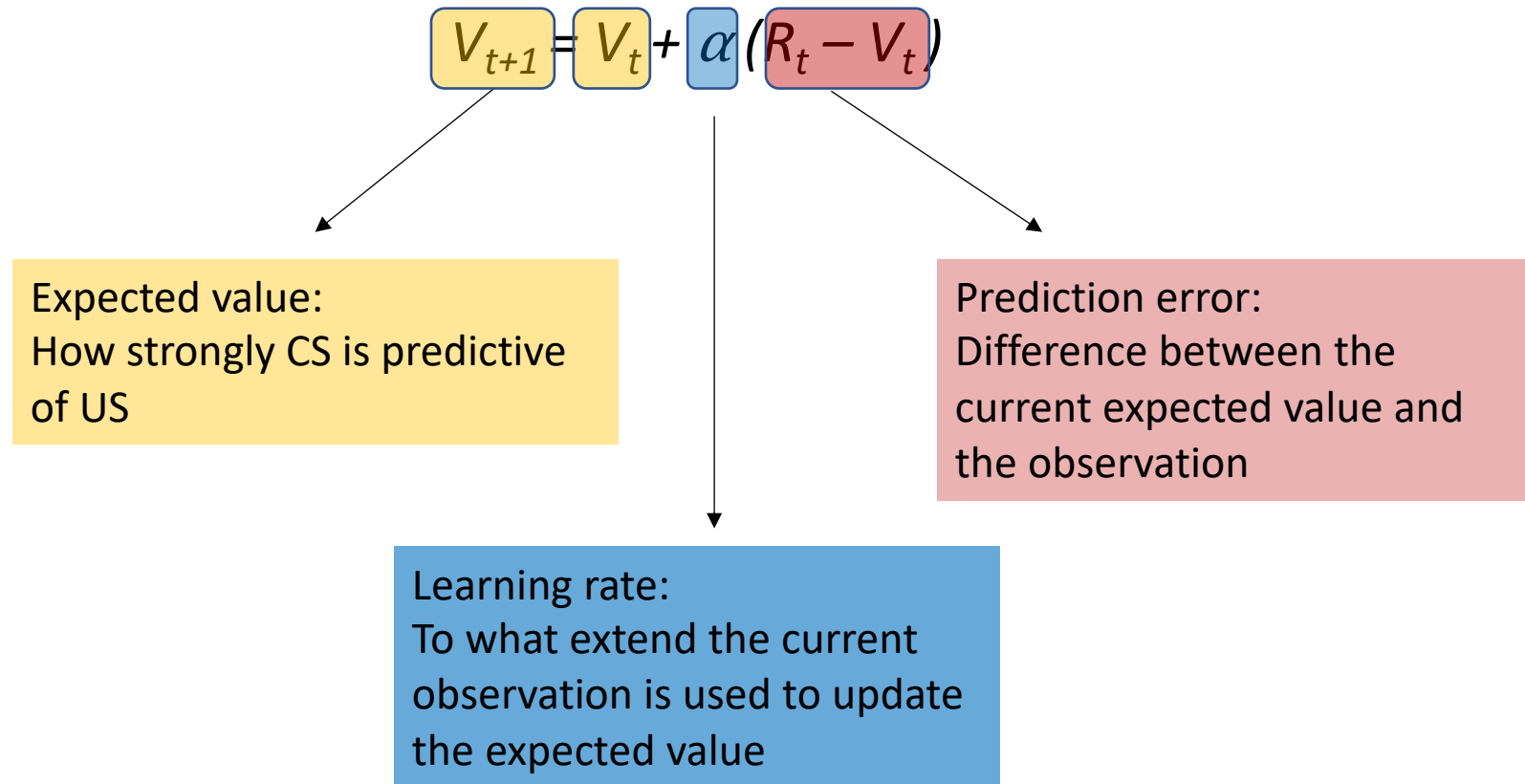
The food is delivered independently of what the agent is doing

The agent 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

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

Question: what is the main implication of this model?
When does learning occurs and when it does not?

Now Simulate it!

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

```
```
```

Rescorla-Wagner model

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

Check what happens if we add probabilistic reward

Simulate it with different learning rates

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

```
```
```

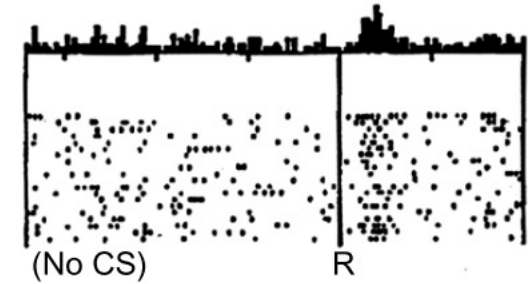
Rescorla-Wagner model

Besides explaining several aspects related to classical conditioning, a slightly modified Rescorla-Wagner model, the Temporal Difference model, was used to derive prediction error which explained

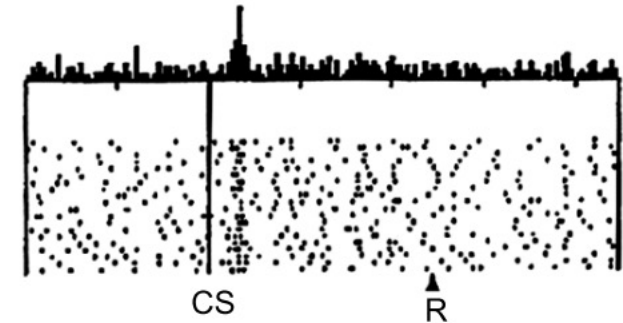
Firing of dopaminergic neurons in the midbrain (Schultz, W. et al. (1997), *Science*)

Do dopamine neurons report an error in the prediction of reward?

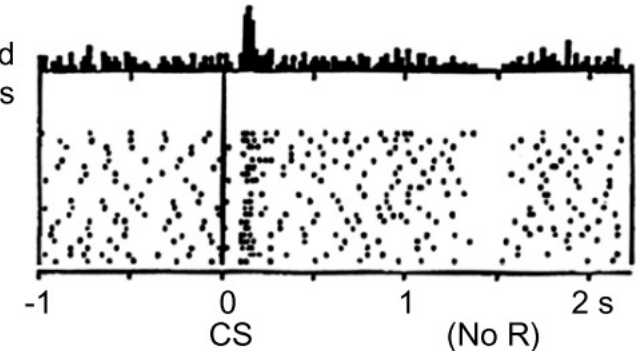
No prediction
Reward occurs



Reward predicted
Reward occurs



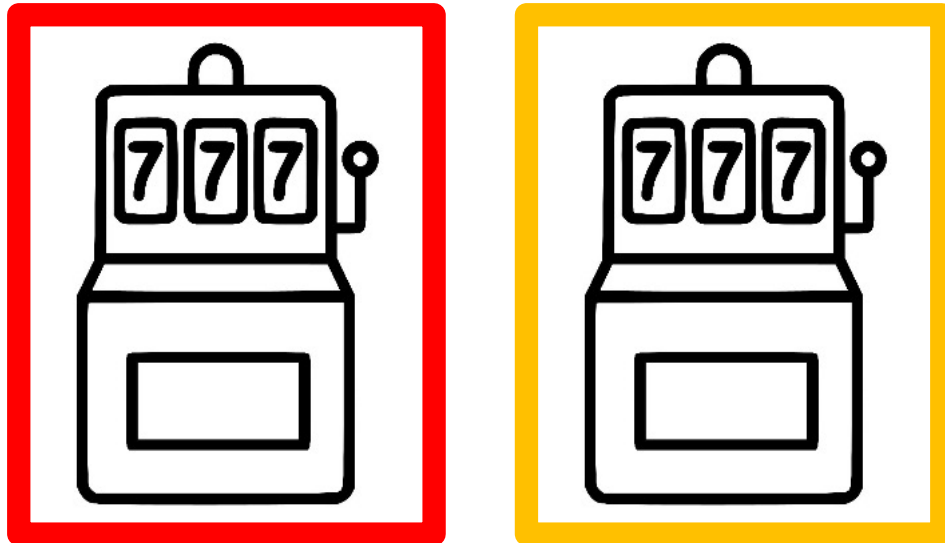
Reward predicted
No reward occurs



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

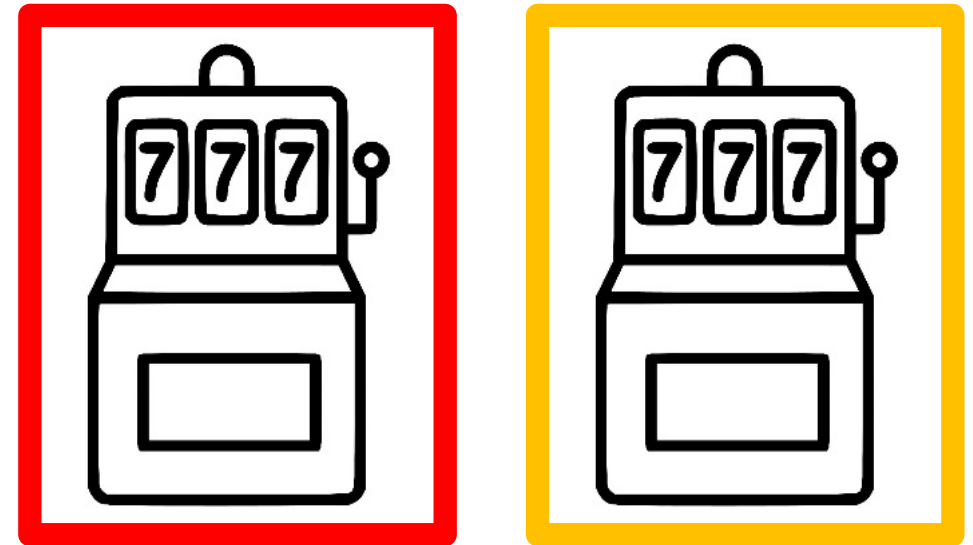
How does the value function look like?

$$Q^k_{t+1} = Q^k_t + \alpha (R_t - Q^k_t)$$

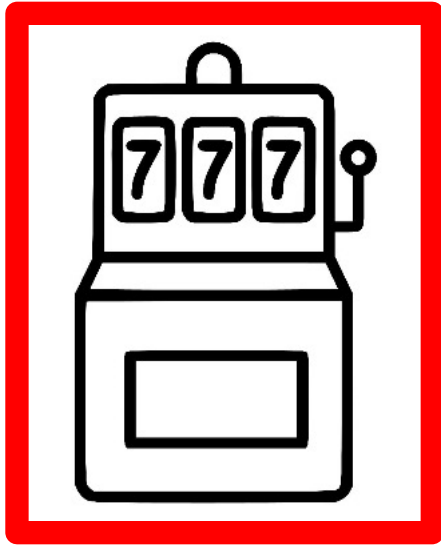
$$Q^{red}_{t+1} = Q^{red}_t + \alpha (R_t - Q^{red}_t)$$

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

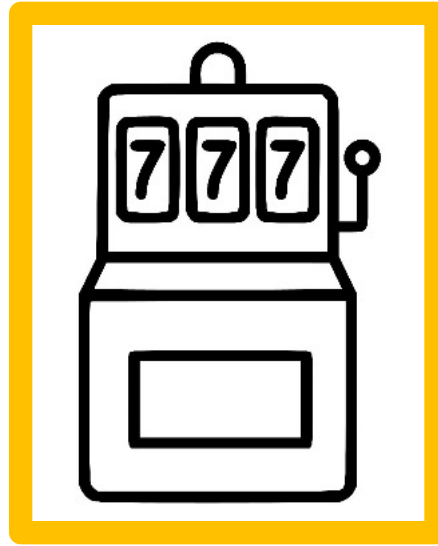
Two-armed bandit task



Instrumental Learning



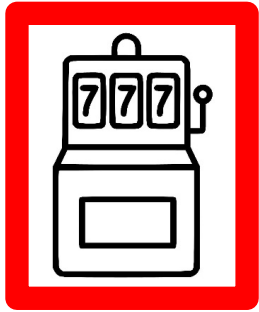
$Q = 0.51$



$Q = 0.49$

How should the agent choose?

Instrumental Learning



$Q = 0.51$



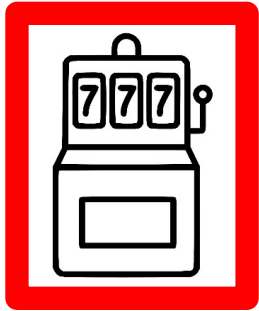
$Q = 0.49$

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

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

```
```
```

Instrumental Learning



$Q = 0.51$



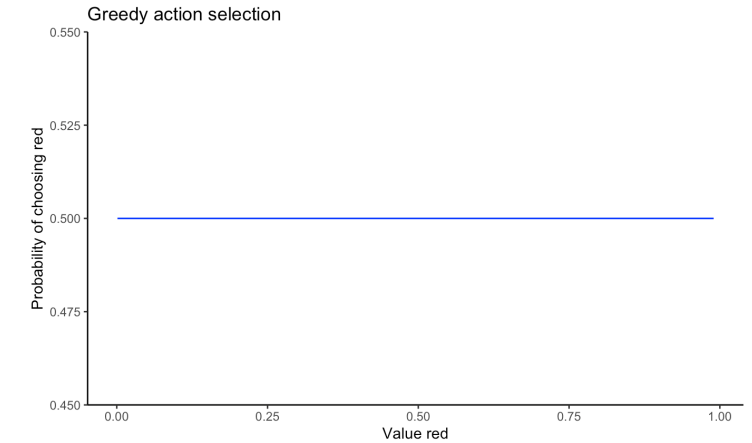
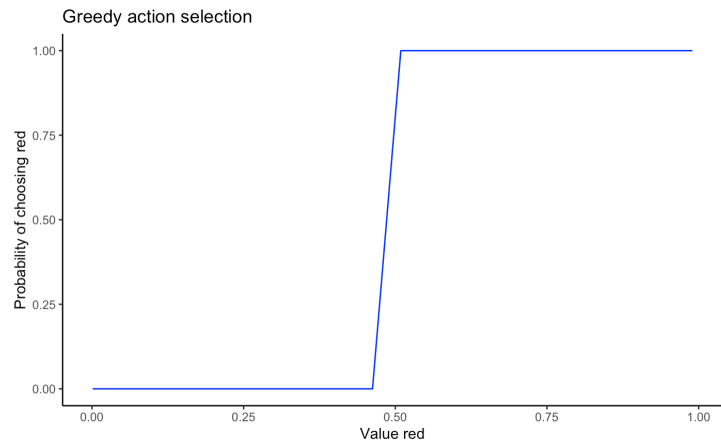
$Q = 0.49$

2. Always explore

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

```
```
```


Instrumental Learning



Exploitation vs Exploration dilemma

We can leave it as a free parameter!

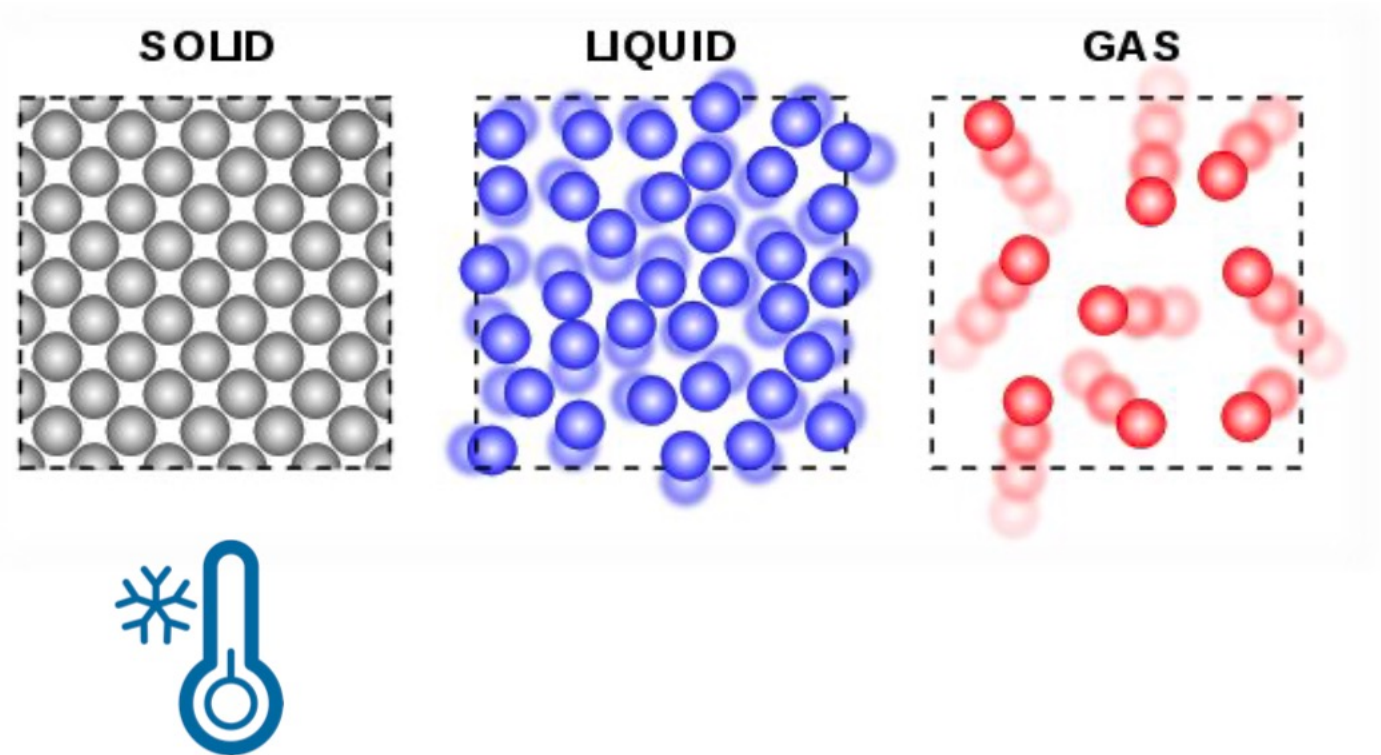
(question: can you think of another parameter we have already talked about that can be considered as "free"?)

Instrumental Learning

τ = temperature parameter

Low temperature

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

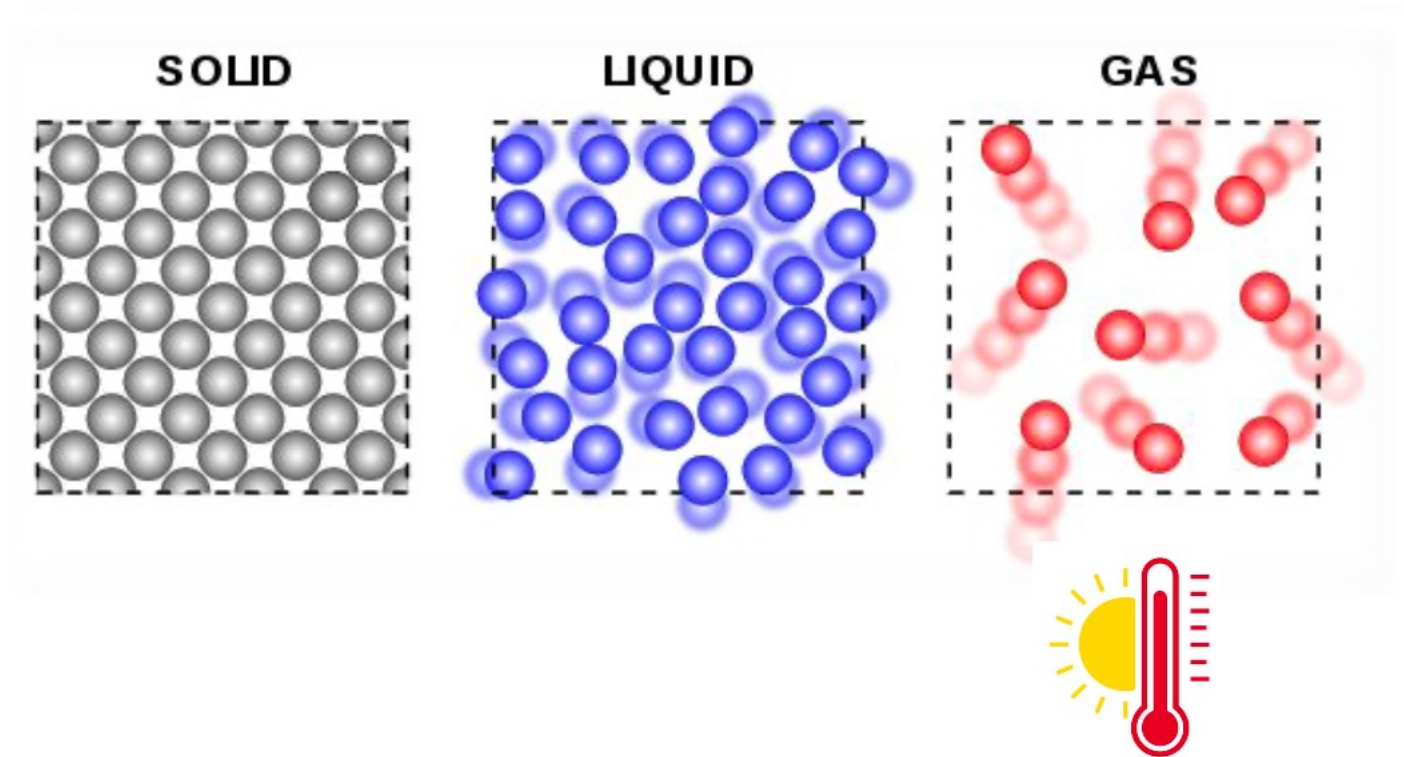


Instrumental Learning

τ = 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, β determines the extent to which value estimates influence choice behaviour.

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

The higher, the more deterministic the choice will be.

And it also normalizes the Qs

Let's try to understand it better!

````{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**

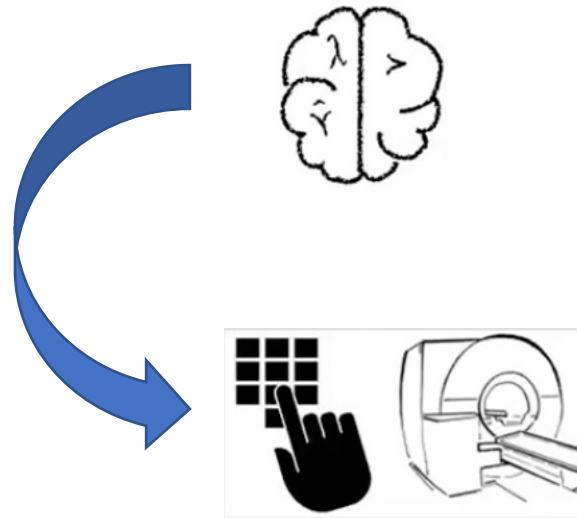
Granada, 2nd June 2022

Model Fitting

What we created is a **Generative Model**

$$Q_{t+1}^k = Q_t^k + \alpha (R_t - Q_t^k)$$
$$p_t^k = \frac{\exp(\beta Q_t^k)}{\sum_{i=1}^K \exp(\beta Q_t^i)} \quad \Rightarrow \quad \theta = \{ \alpha, \beta \}$$

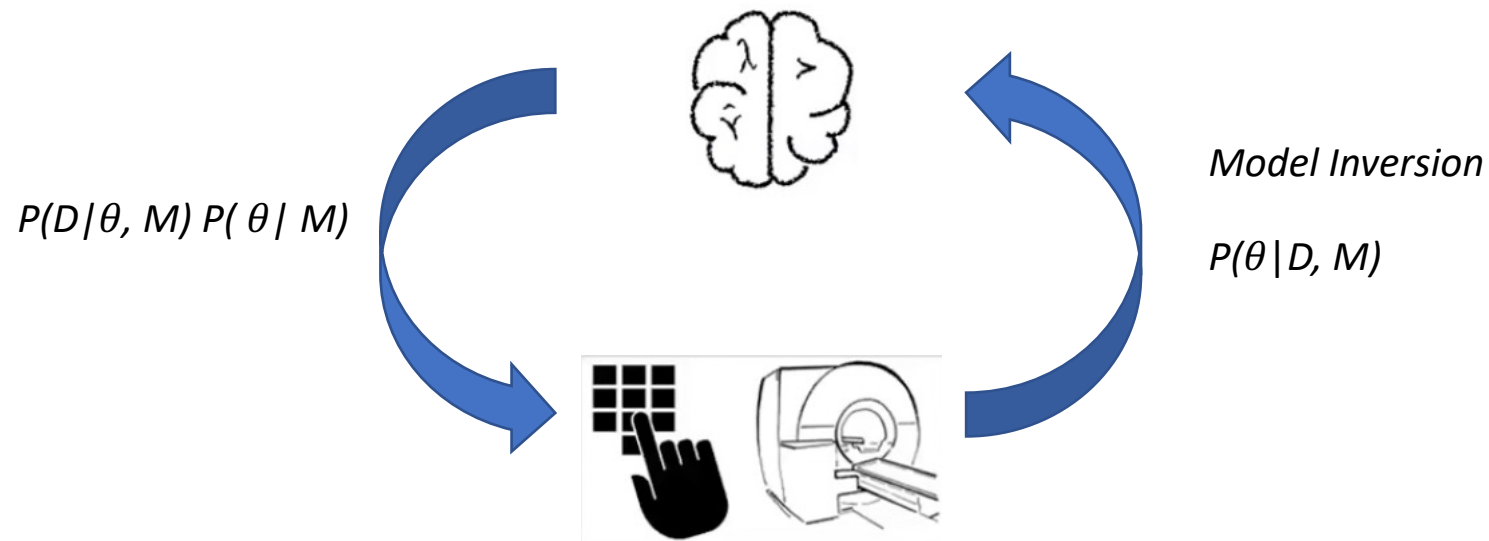
$$P(D|\theta, M) P(\theta | M)$$



Model Fitting

What we created is a **Generative Model**

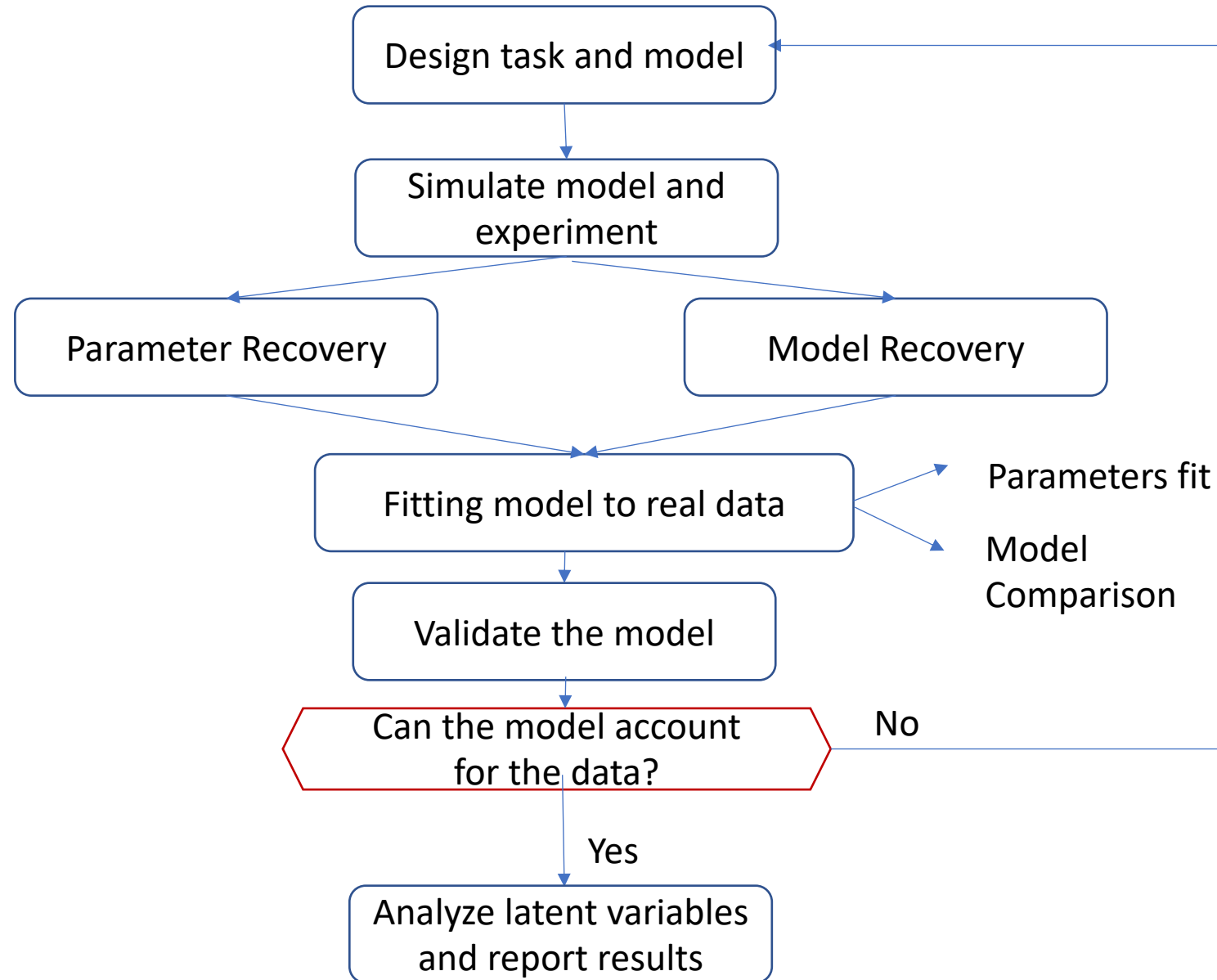
$$Q_{t+1}^k = Q_t^k + \alpha (R_t - Q_t^k)$$
$$p_t^k = \frac{\exp(\beta Q_t^k)}{\sum_{i=1}^K \exp(\beta Q_t^i)} \quad \Rightarrow \quad \theta = \{ \alpha, \beta \}$$



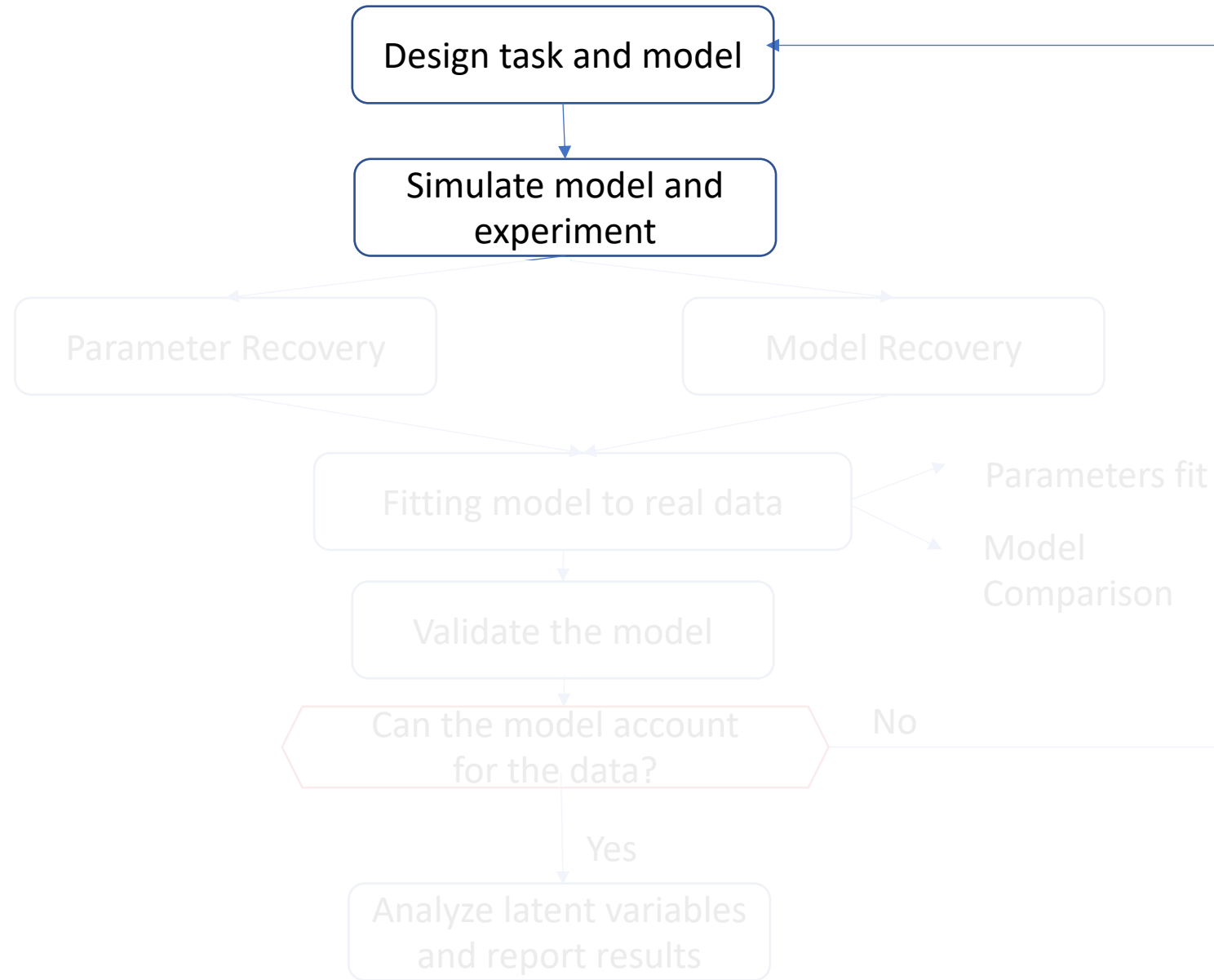
Model Fitting

Fitting model to real data

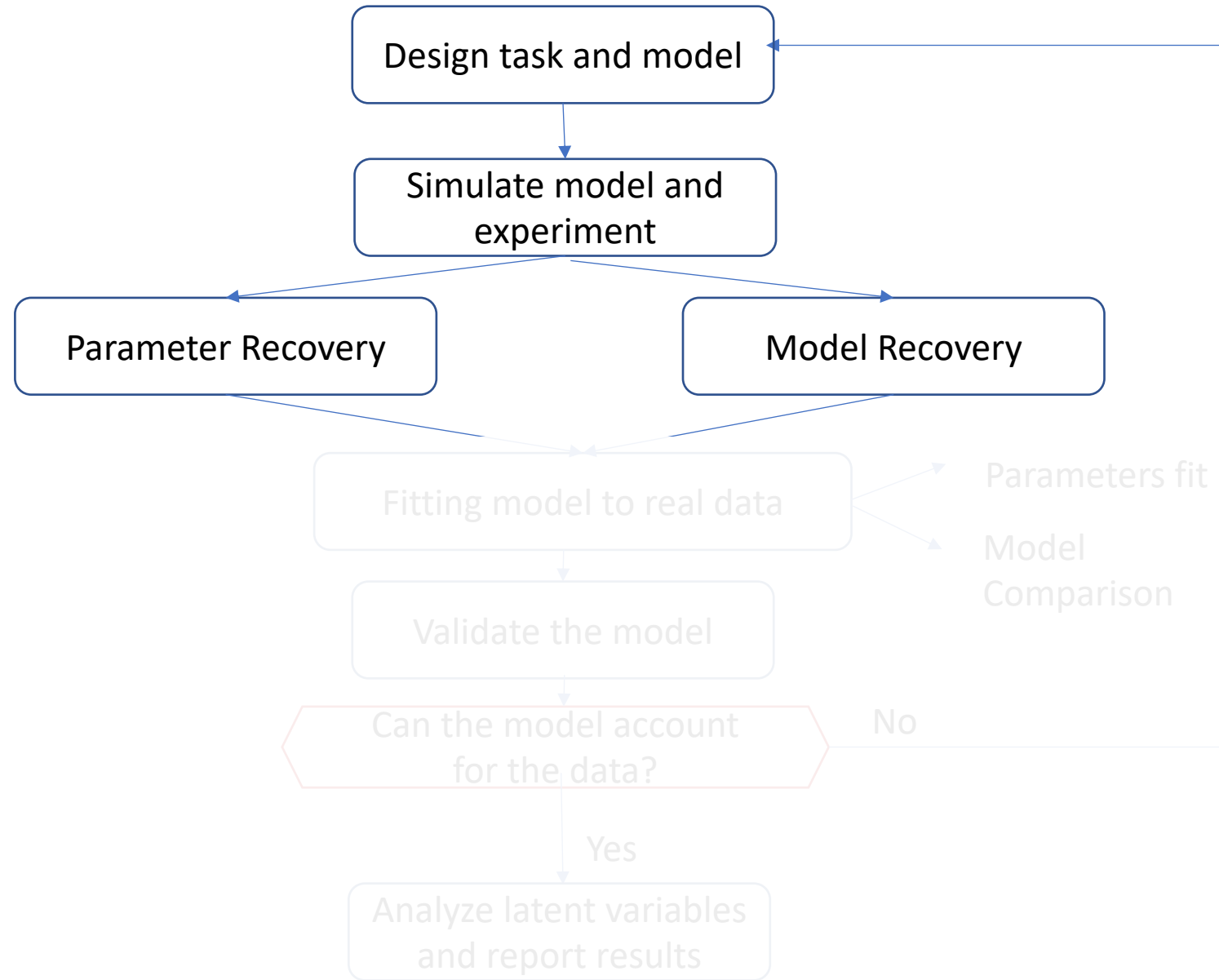
Model Fitting



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 model to data we are estimating the parameters that are maximizing the likelihood of observing those data.

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

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

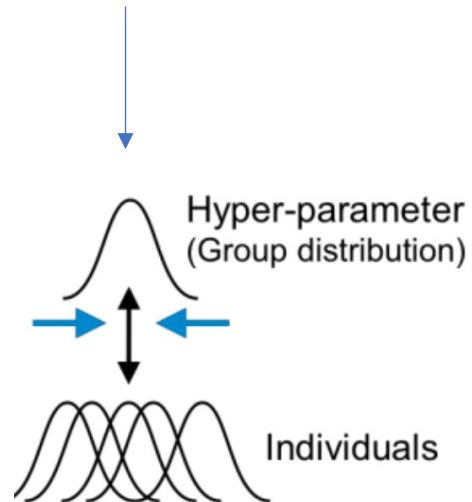
The *optim* function from R uses optimization algorithms to find the values that minimize each functions:
As it looks for a minimum, we need to feed it the Negative Log Likelihood

Parameter Recovery

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

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

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



Parameter Recovery

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

```
``
```

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?

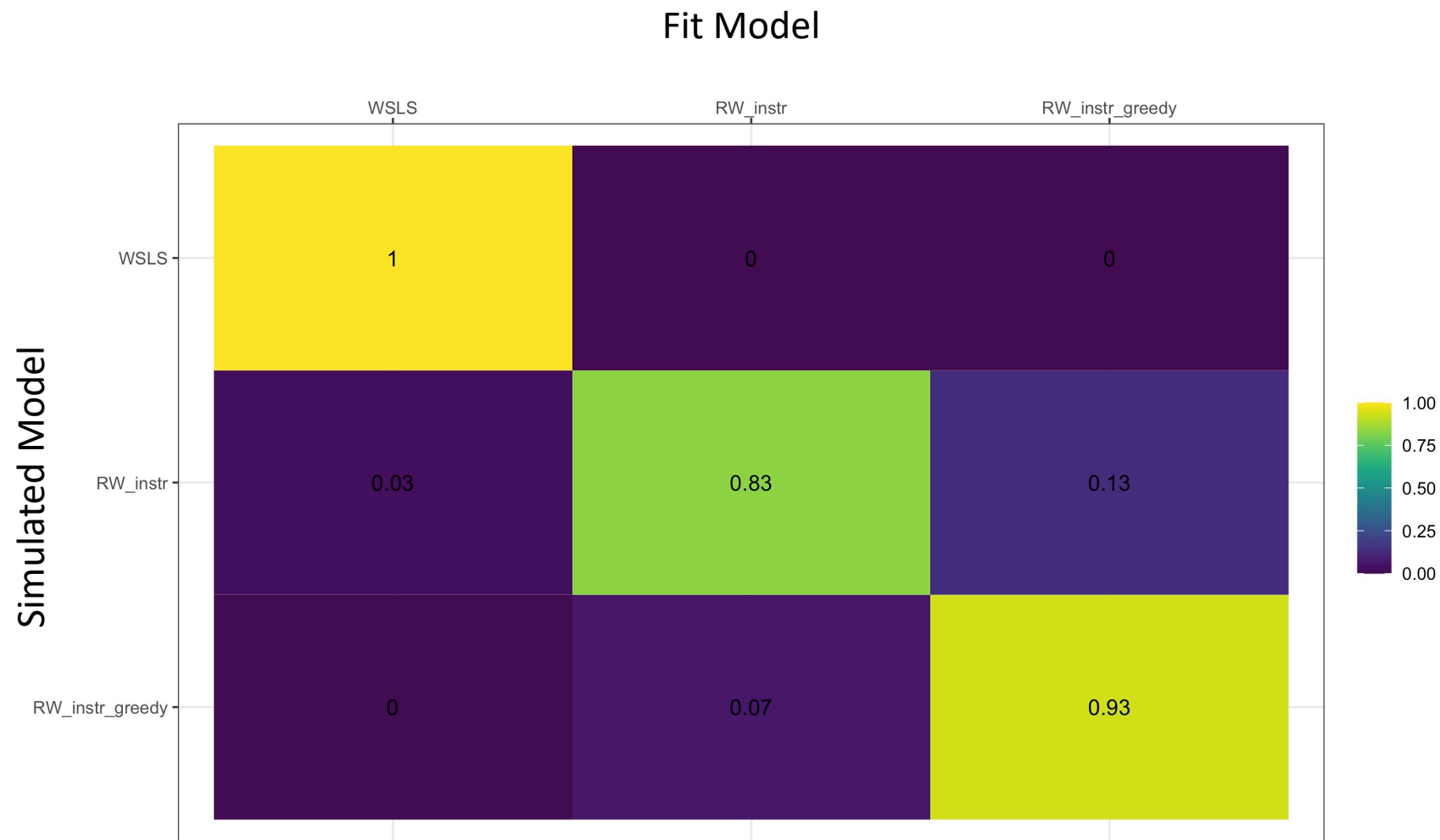
In model comparison, our goal is to figure out which model of a set of possible models is most likely to have generated the data.

Therefore, for each model, we need to compute the probability that the data were generated by a model: $p(m/d_{1:t})$

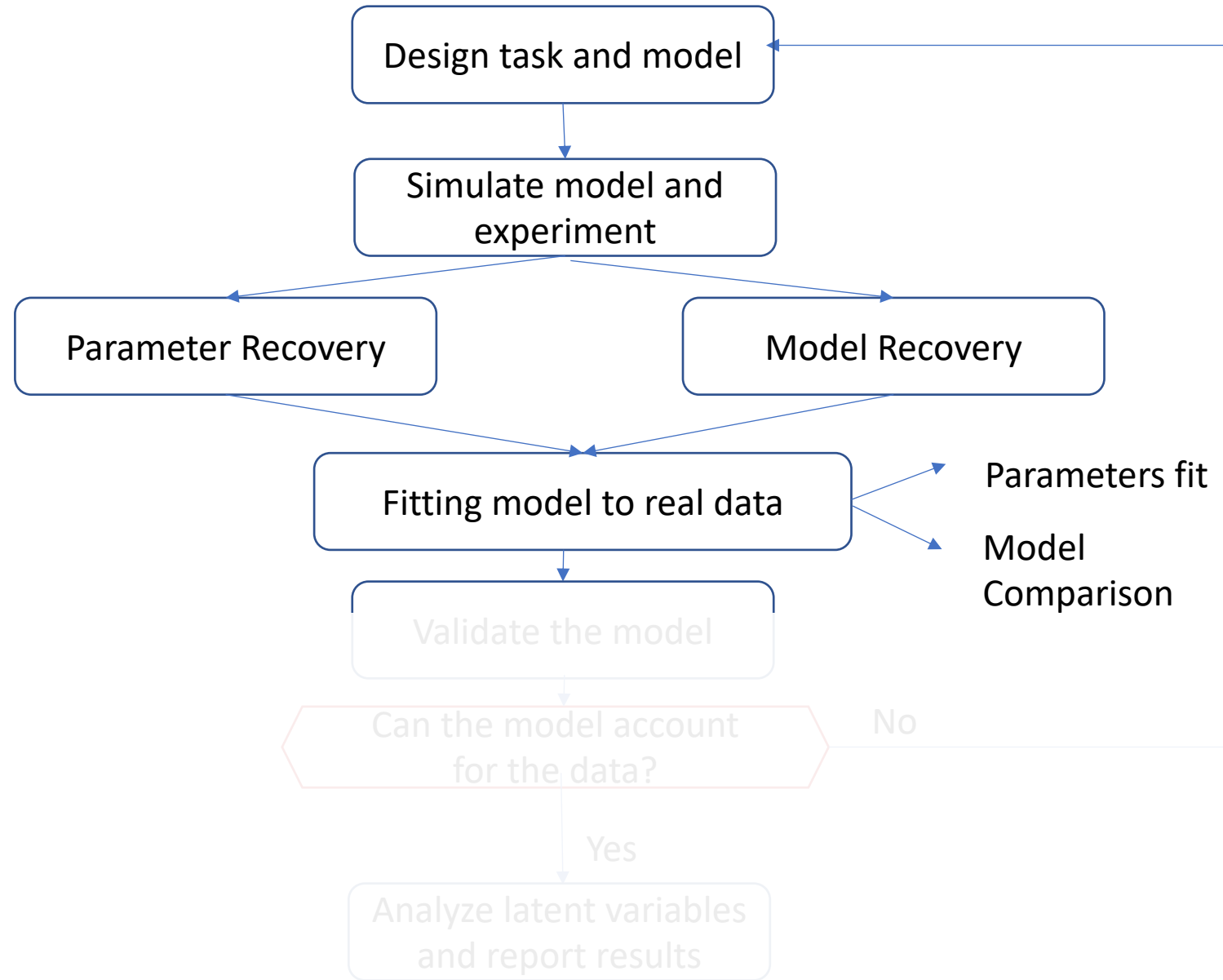
The method commonly used is the Bayesian Information Criterion :

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

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}

```

Model Comparison

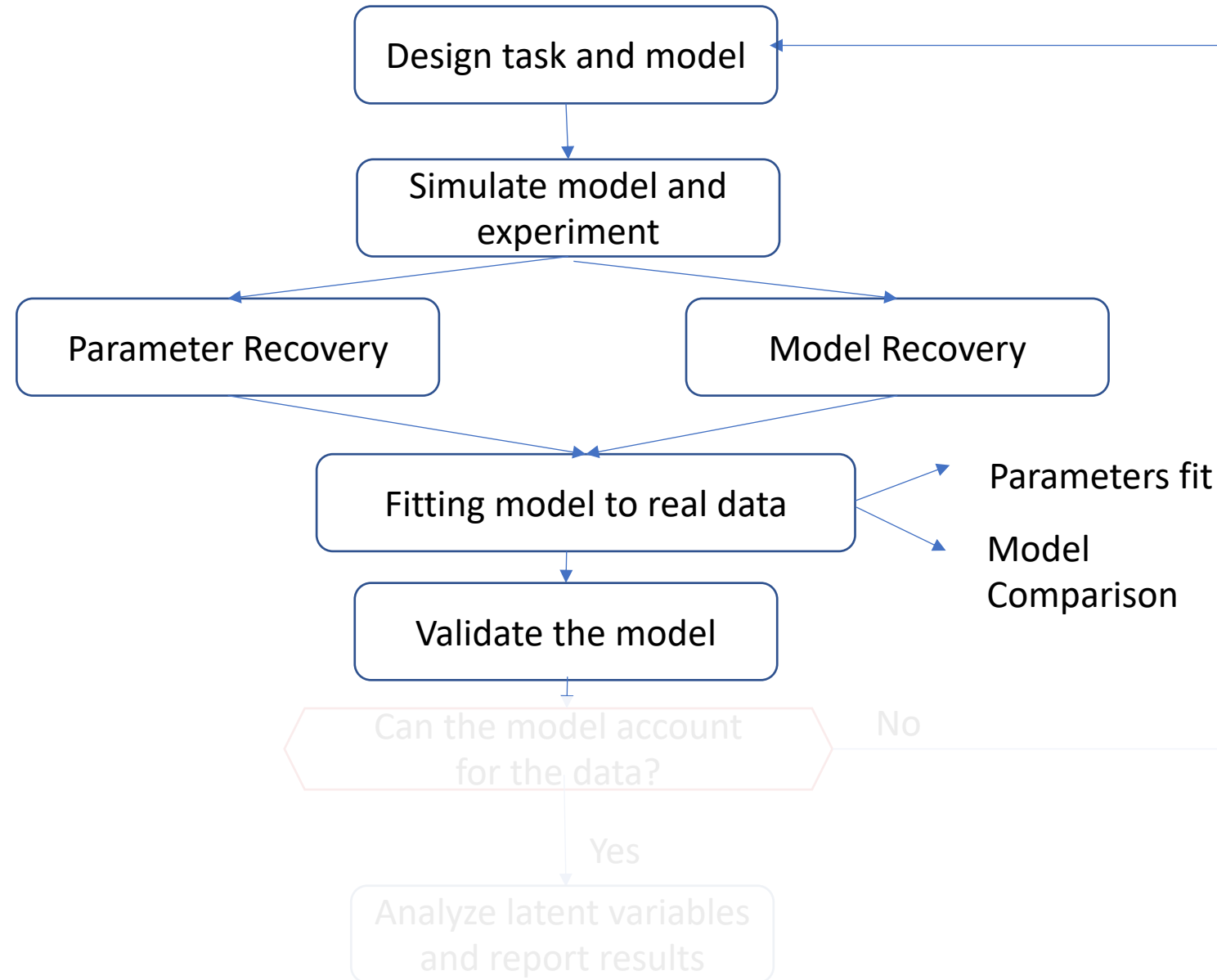
After calculating the BIC, we can count the number of participants for which each model was the best fit.

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}

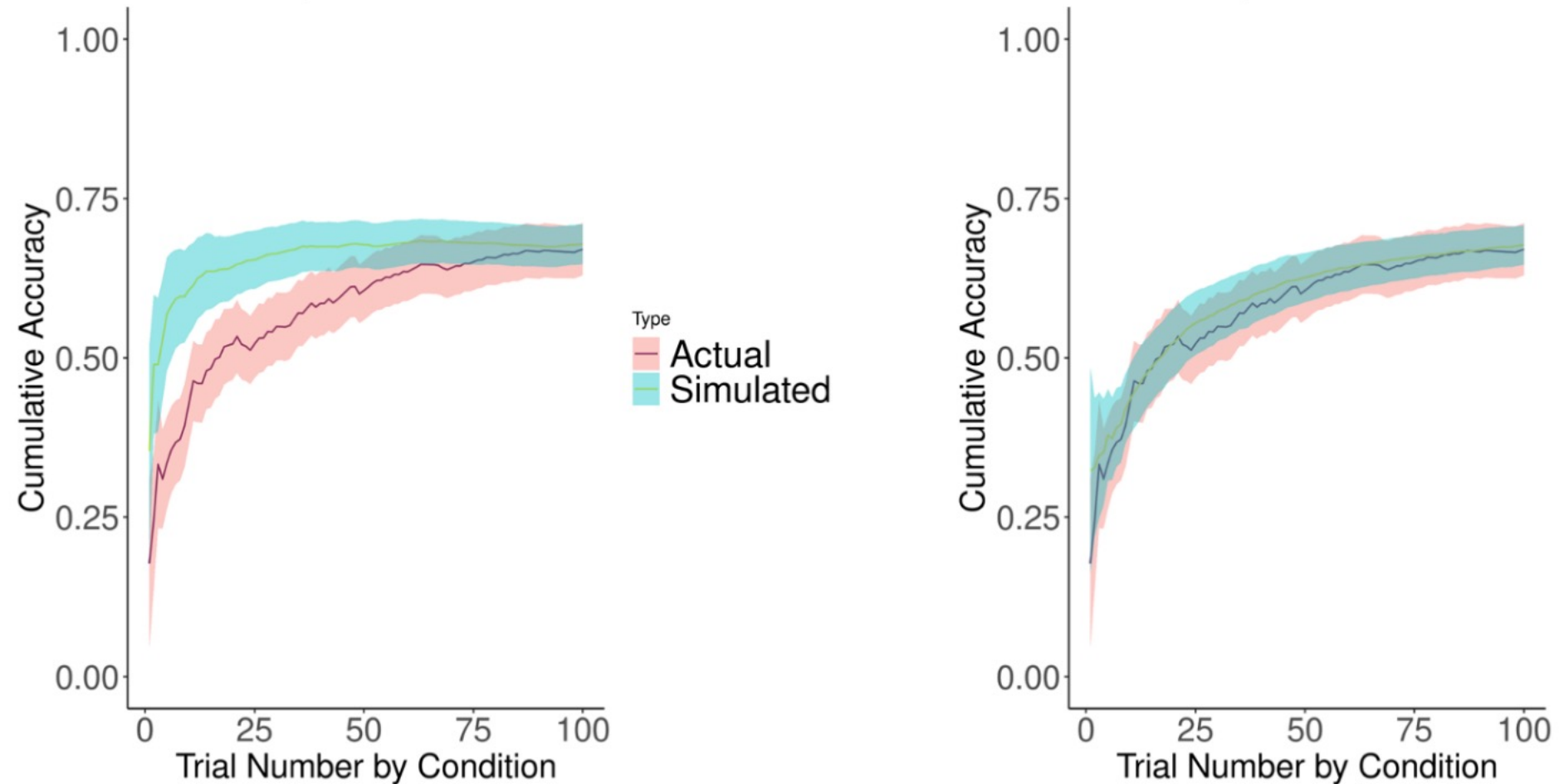
```

Model Fitting

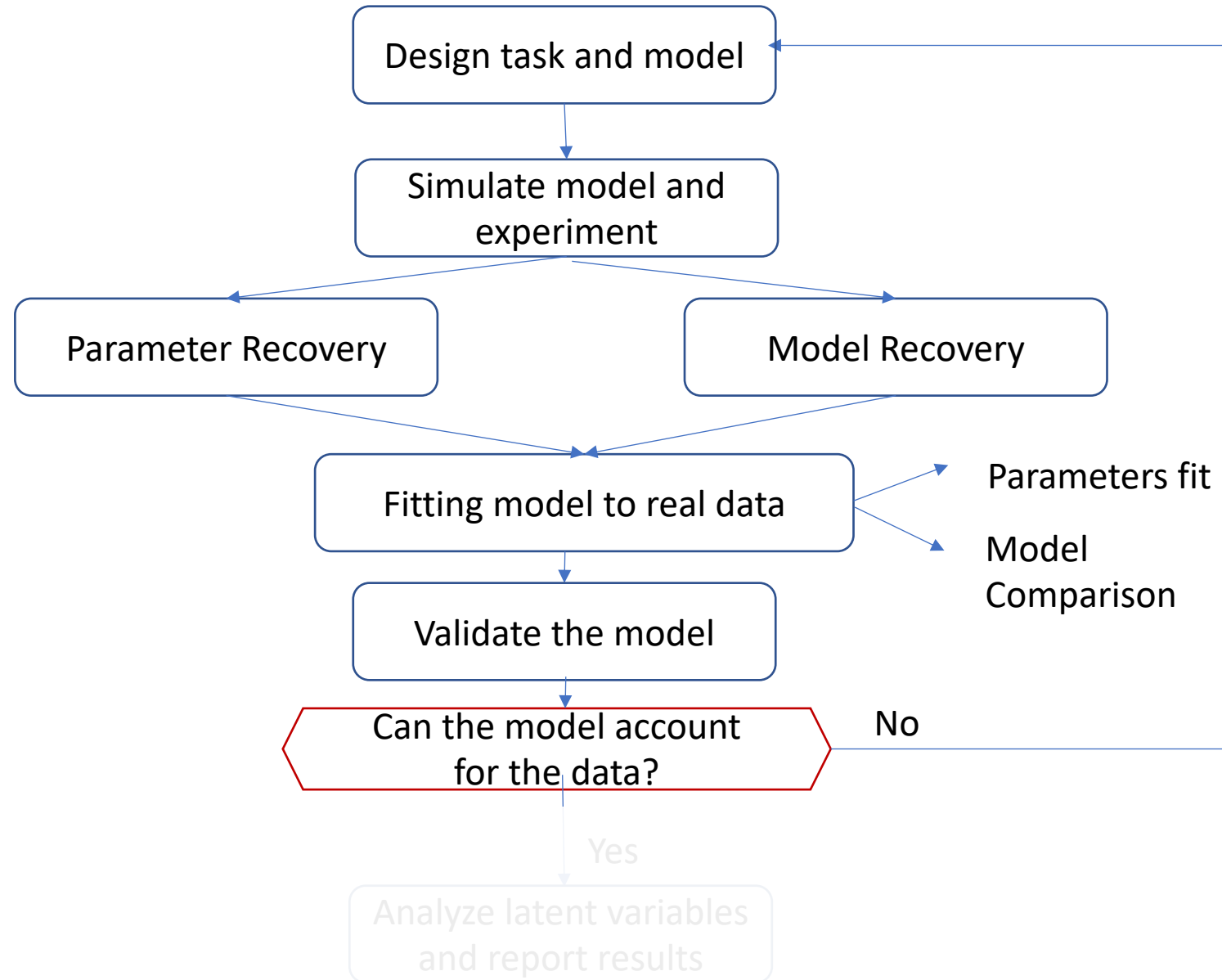


Model Fitting

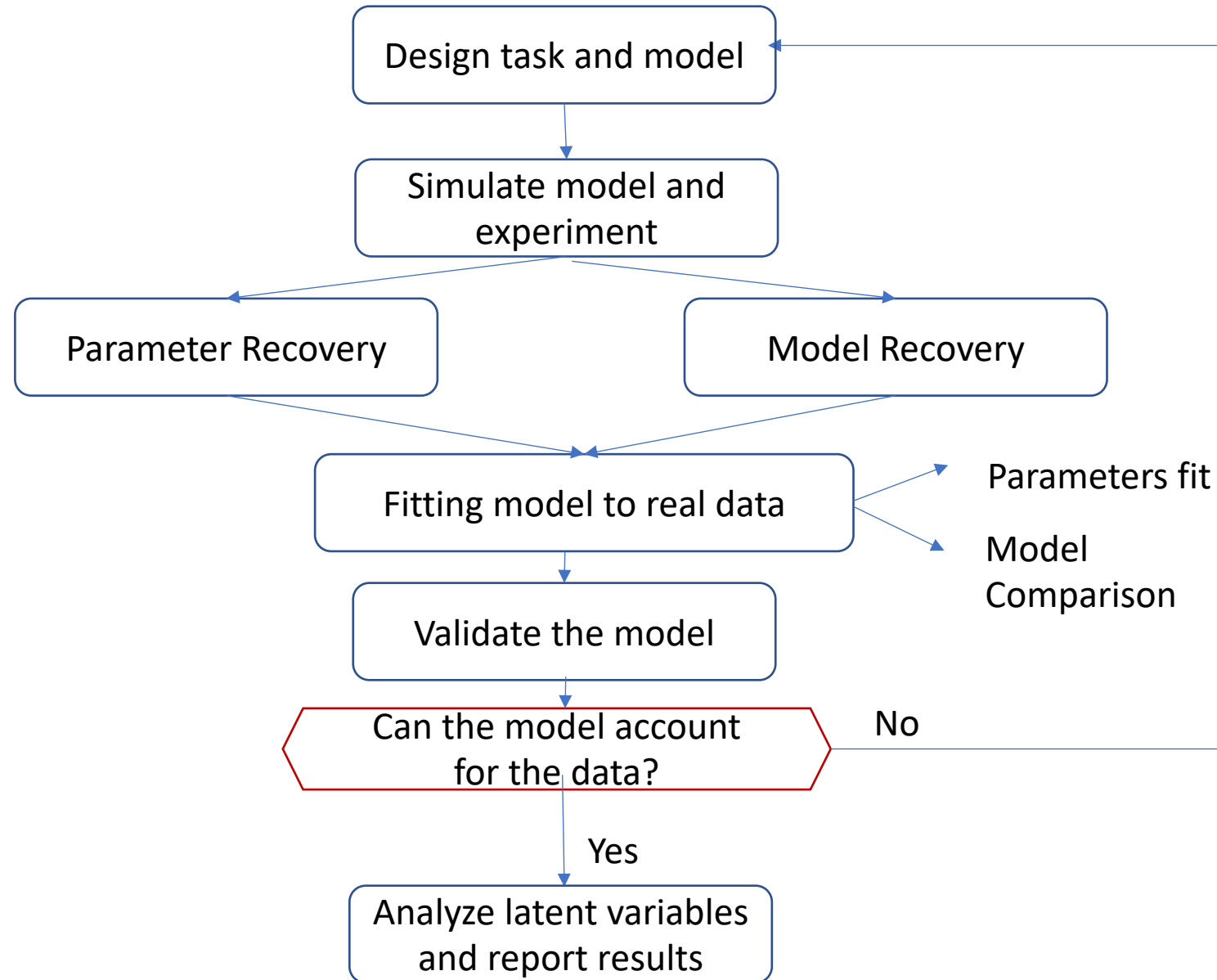
Validating the model means to check whether data simulated by the model, with the parameters of best fit, replicate pattern observed in the empirical data (posterior predictive check)



Model Fitting



Model Fitting



Model Fitting

Thank you!

Questions??

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>