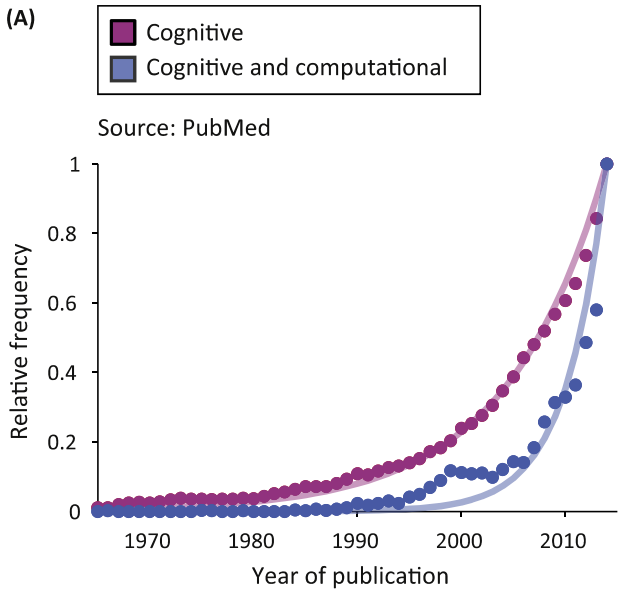
Introduction to Computational Cognitive Modeling

Ti-Fen Pan
UC Berkeley
SBI summer school, July 27-31, 2026

Slide credits to Anne Collins

Computational modeling is useful

- Precise, testable quantitative predictions
- Descriptive, but also generative/mechanistic explanations
  → Cognitive scientists and cognitive neuroscientists
- Relate neural function to mechanisms giving rise to behavior
  →Cognitive and behavioral neuroscientists
- Summarize individual differences into interpretable variables
  → Development, computational psychiatry

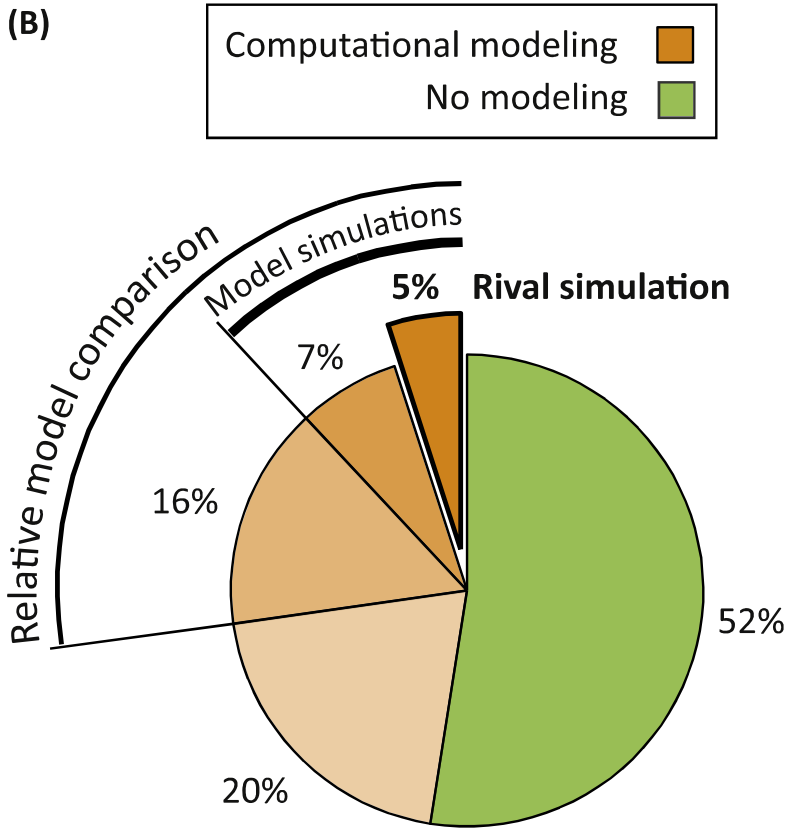


Palminteri et al, 2017

Computational modeling can be dangerous

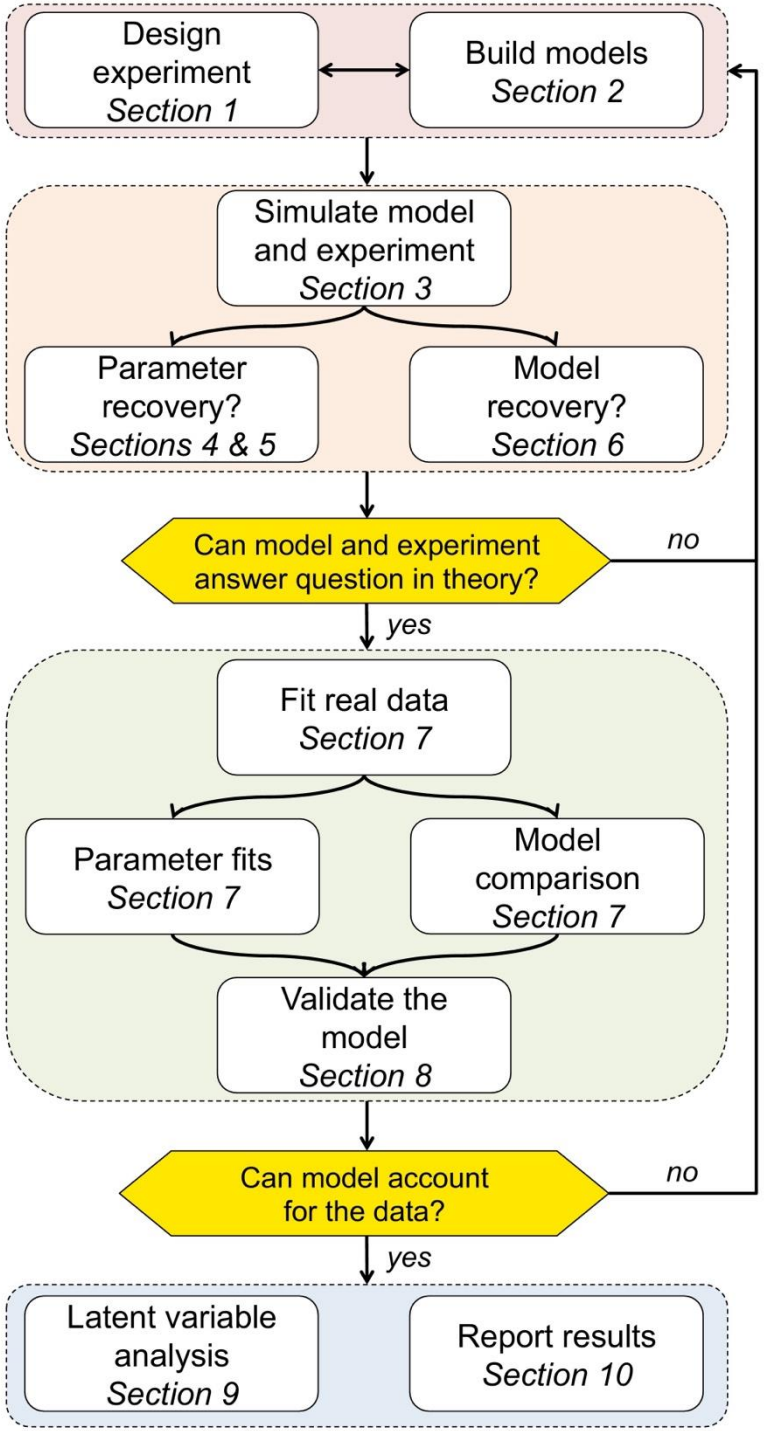
- It's important to understand the basics very well and all the potential pitfalls (the goal of this talk!).

- More bad than good examples in the literature. Be careful!



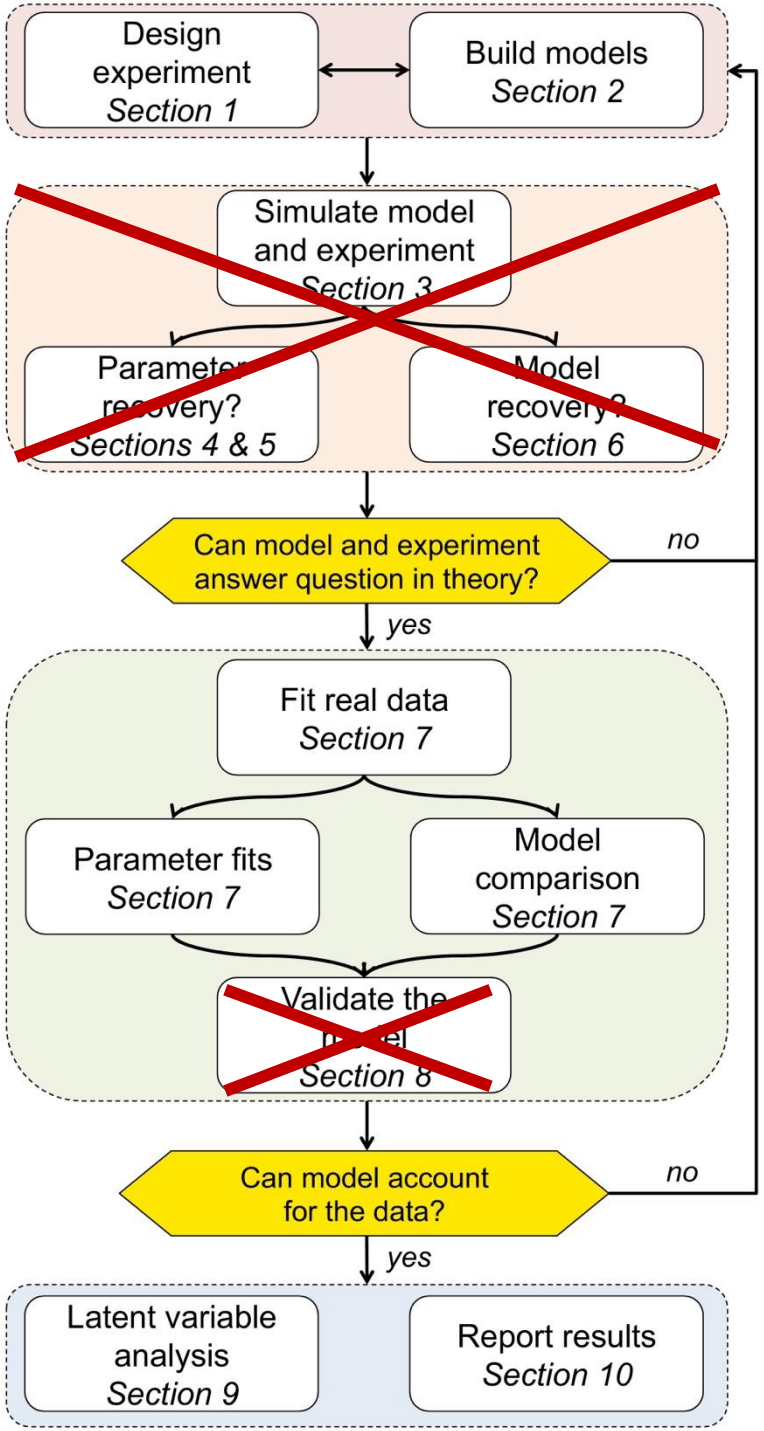
Palminteri et al, 2017

Life cycle of a modeling project



Wilson et al, 2019

Life cycle of a modeling project



Wilson et al, 2019

Steps

1. Data analysis
2. Model simulation
   - Role of model parameters
   - Setting up models representing competing hypotheses
3. Parameter estimation (model fitting)
4. Model comparison
5. Model validation
6. Improving models

Data analysis and model simulation

- First step: data analysis. Know your data set!
  - What questions are you trying to ask?
  - What questions can you answer in a model-independent way?
  - What questions is your protocol not designed to answer?
  - How do participants behave?

RL example

Two uncertainty levels

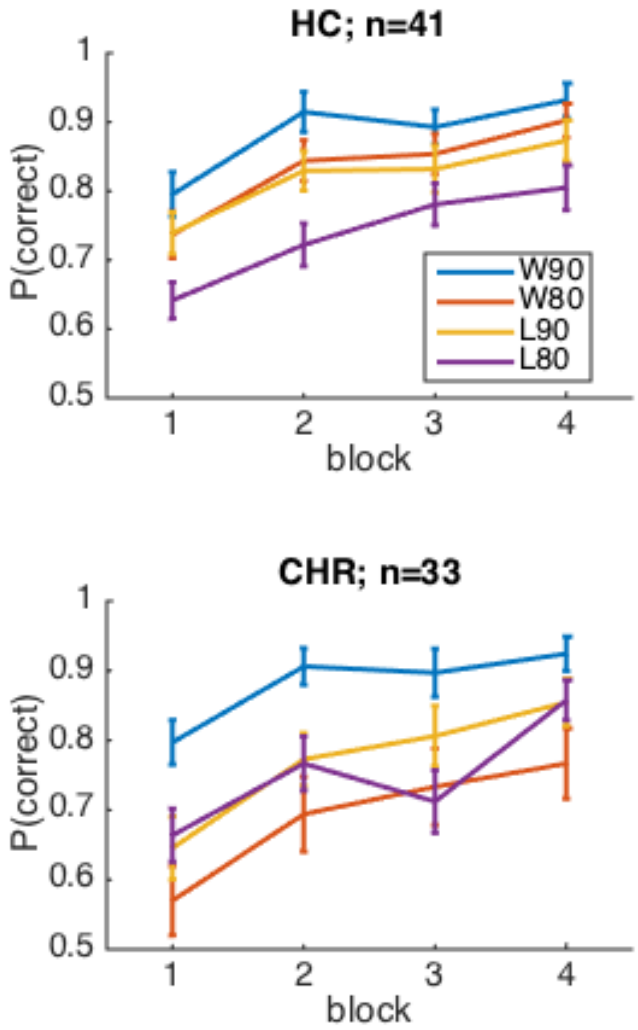
	90%	80%
Win: $0.05 vs. 0	AB	CD
Lose 0 vs. -$0.05	EF	GH

- All conditions interleaved
- 4 blocks of 4*10 = 40 trials

- 41 youths at risk for schizophrenia
- 33 matched controls

Waltz et al 2012

Preliminary behavioral results, modeling targets



- ***Question:*** Do we learn differently from seeking gains vs. avoiding losses? Do youths at risk for schizophrenia learn differently from matched controls?
- Behavior:
  - Learning from ***losses*** is slower than learning from ***gains***
  - Effect of ***uncertainty*** on learning is different in the gain and loss domains for High Risk individuals
  - → two modeling targets

Collins & Strauss

2. model simulation

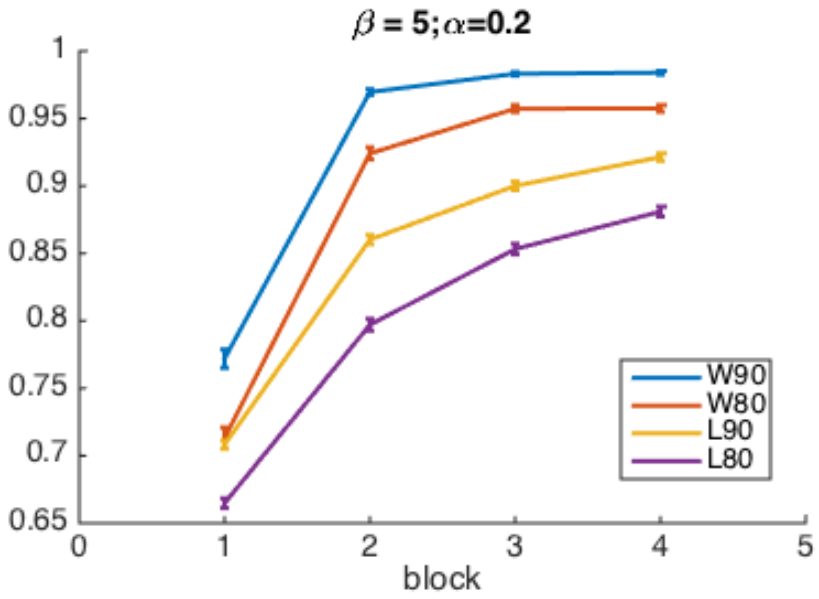
- A: model building
  - Develop a model that represents your main hypothesis
  - Develop competing models

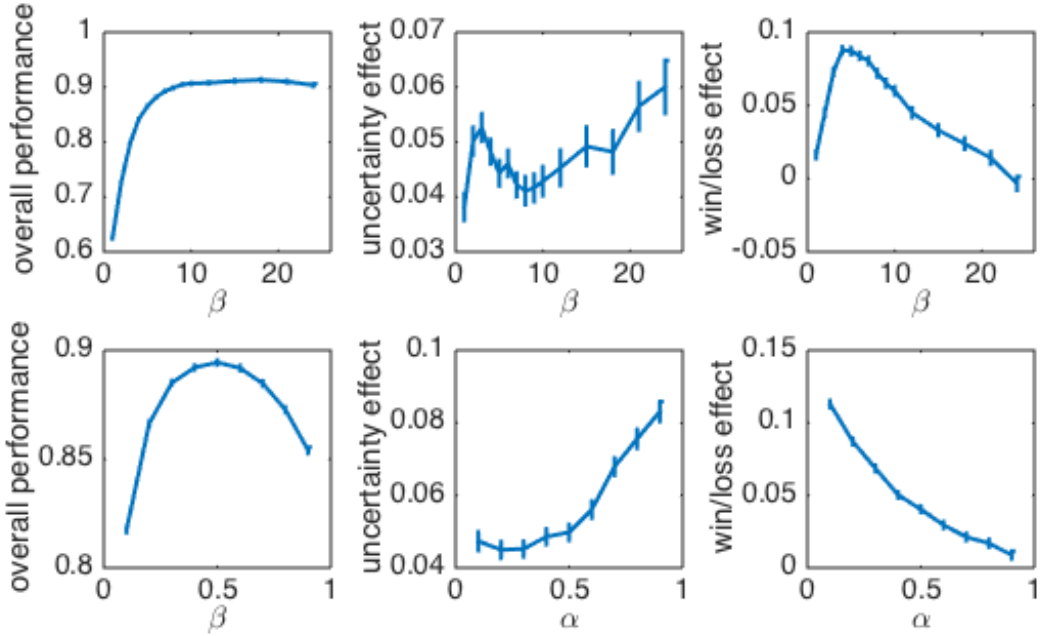
$$Q_{t+1} \leftarrow Q_t + \alpha \, RPE$$
$$Choice \sim \exp(\beta \, Q)$$

- B: model simulation
  - Simulate your models **in the exact experimental environment**
  - Compare qualitative predictions of different models
  - Know the influence of all model parameters on model behavior

2. model simulation

- B: model simulation
  - Simulate your models **in the exact experimental environment**
  - Compare qualitative predictions of different models
  - Know the influence of all model parameters on model behavior





Steps

1. Data analysis
2. Model simulation
   - Role of model parameters
   - Setting up models representing competing hypotheses
3. Parameter estimation (model fitting)
4. Model comparison
5. Model validation
6. Improving models

3. Relate model to data – model fitting

- Bayes Rule!

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

Posterior probability of the parameters

Likelihood of the data

Prior on model parameters

- What we want to know:
  - How likely is it that the data was generated with parameters (α^*, β^*)?
- What we can compute:
  - How likely is the data given model assumptions and parameters (α^*, β^*)?

The theory –a simple trick

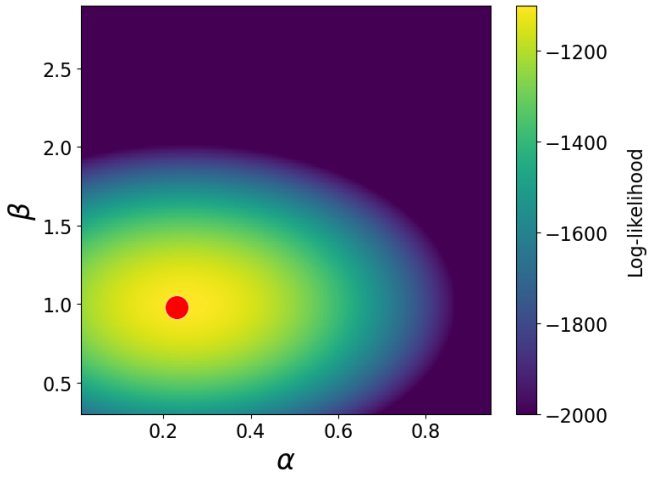
- Likelihood is a tiny number: product of many numbers<1

$$LLH = \log(LH) = log\left(\prod P(a_t|H_{t-1})\right) = \sum log\left(P(a_t|H_{t-1})\right)$$

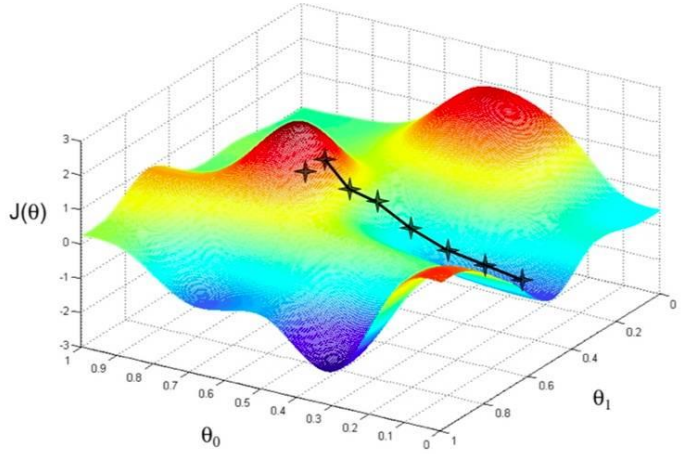
Example!

```
>> probabilities = rand(1,1000);
>> prod(probabilities)

ans =

     0

>> sum(log(probabilities))

ans =

 -965.7124
```

Grid Search



Gradient Descent

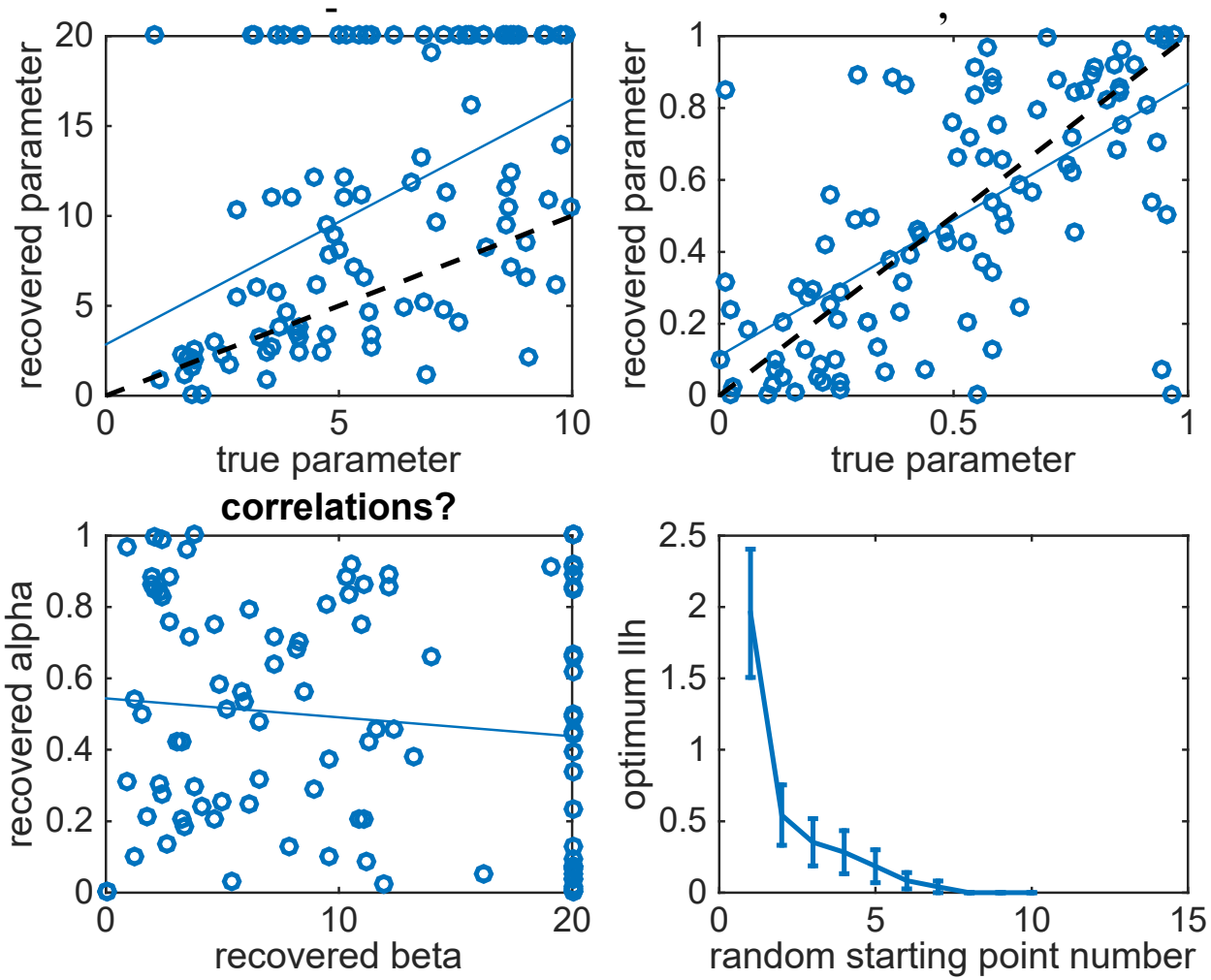


Parameter recovery: Generate and recover!

Validate this procedure!

1. Simulate the model with a reasonable range of parameters to create ***"fake" (synthetic) data***
2. Fit the "fake data"
3. Compare the recovered parameters with the true parameters

n=1 block
10 starting points



Tips and tricks: Parameter fitting

- More data always leads to better parameter recovery
- The generate and recover procedure, on ***your*** experimental protocol, shows you whether you can recover parameters satisfactorily
- Carefully choose constraints on parameters.
- Beware of local minima (always run fitting multiple times with random initial conditions)
- **Once you are satisfied that you can recover parameters for a model, you can fit your real data!**

Steps

1. Data analysis
2. Model simulation
   - Role of model parameters
   - Setting up models representing competing hypotheses
3. Parameter estimation (model fitting)
4. Model comparison
5. Model validation
6. Improving models

4. Model comparison: The question

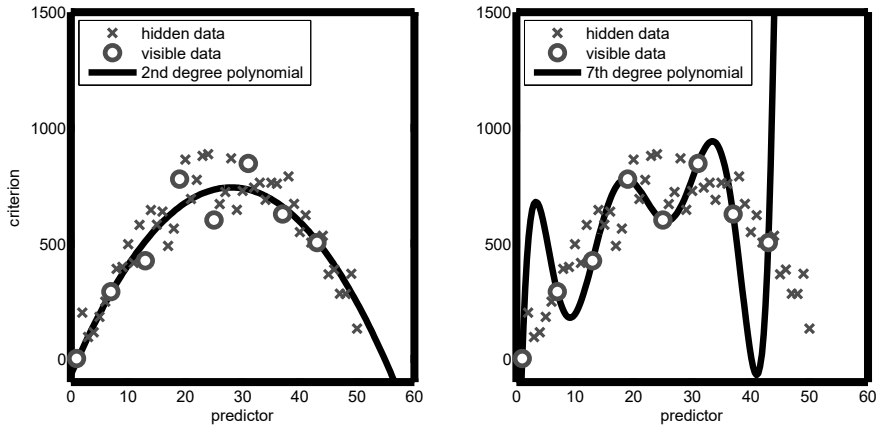
- Which of two (or n) models fits the data best?

- A better question:
  - Which of the models generalizes better?
  - Which of the models will let me predict the participant's behavior best?

Bias-variance tradeoff / Occam's razor

- More is not always better!
  - Capture signal in the data, not noise
- Best solution: **out-of-sample fit measure**

  → how well your fit model predicts new data (generalization)

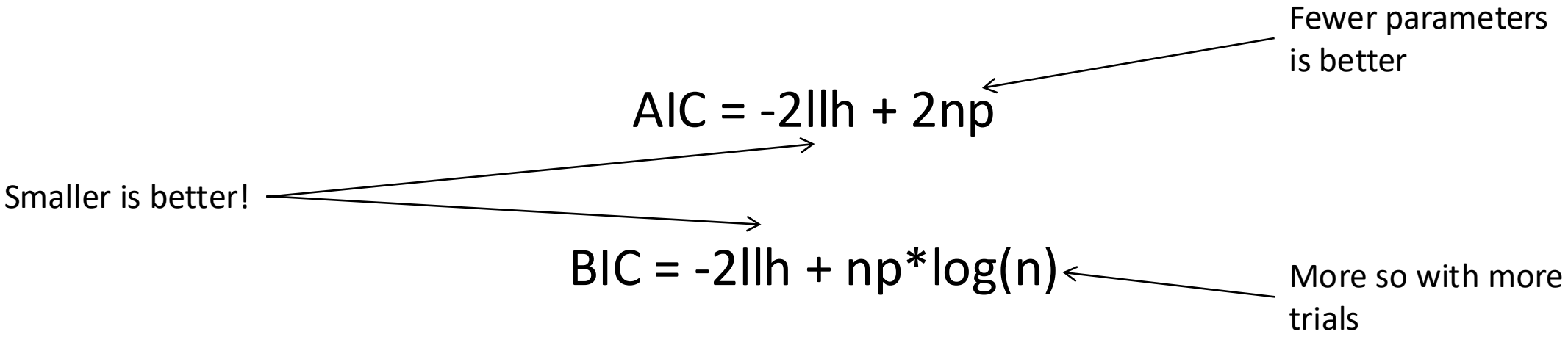


Measure fit out of sample. Why we rarely do it.

- Not enough data
- Not possible to separate data into independent data sets
  - for example, if trying to model a single learning block.

Approximate criteria for model comparison

- Principle:
  - measure model fit by likelihood, but penalize for model complexity.
- Various Information Criteria come from different mathematical theories, with different interpretations and guarantees
  - Bayesian Information Criterion (BIC)
  - Akaike Information Criterion (AIC)



Model comparison – validating the procedure

- Which model comparison method should I use?
  - I.e. are my models ***identifiable***? With my comparison method and the experimental protocol, can I distinguish between the models?
- Test it in the best-case scenario (where the true model is known: synthetic dataset). Create confusion matrices:
  - Generate data with models M1 and M2
  - Fit the generated data with model M1 and M2
  - Verify that data generated with Mi is best fit by model Mi

	M1's AIC better	M2's AIC is better
Data generated by M1	95%	5%
Data generated by M2	0	100%

What other factors could influence model identifiability?

- Model domain: range of parameters over which the model is simulated.
  - ***Case study:*** if range of beta is very low, choice will be mostly noisy, so other parameters will have little influence on the likelihood. This will influence model identifiability.
  - Many other such situations…

- In practice: what range of parameters should I pick to do my model recovery analyses?
  - **Before you have fit to real data: pick a range of parameters where the model behavior mimick expected experimental range.**
  - After you have fit to real data: use real fit parameters. This allows you to draw conclusions for your actual sample, but this will be less generalizable.

Tips and tricks: Model comparison

- Confusion matrices are computationally intensive.
  - Start small!
    - Small number of simulations/participants per model
    - Small number of starting points for fitting
    - Small number of trials (if possible)
- Model + parameter recovery
  - When doing model recovery, you're also doing parameter recovery (you're fitting M1 data with M1, M2 data with M2, etc…)
  - But in practice, to minimize computation cost, you can combine the two steps.
  - Downside: doing parameter recovery is faster (n instead of n^2), and you might not use all models in model recovery analysis – so it sometimes still makes sense to do the two steps separately.

What **can** you conclude so far?
What can you **not** conclude so far?

Yes

- My procedure identifies Mx as most likely from {M1, …, Mn} to generate the data
- Best fit parameters for Mx are the most likely ones.

No

- Mx is a good model of my cognitive process of interest
- Mx is the best model out of {M1, …, Mn} to model my cognitive process.

Steps

1. Data analysis
2. Model simulation
   - Role of model parameters
   - Setting up models representing competing hypotheses
3. Parameter estimation (model fitting)
4. Model comparison
5. Model validation (after fitting your behavior data)
6. Improving models

Quantitative measures of model fit is not enough

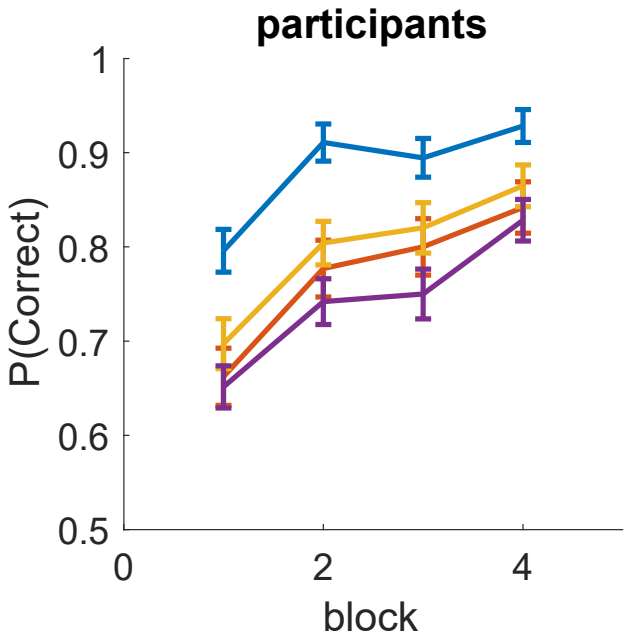
Qualitative questions:

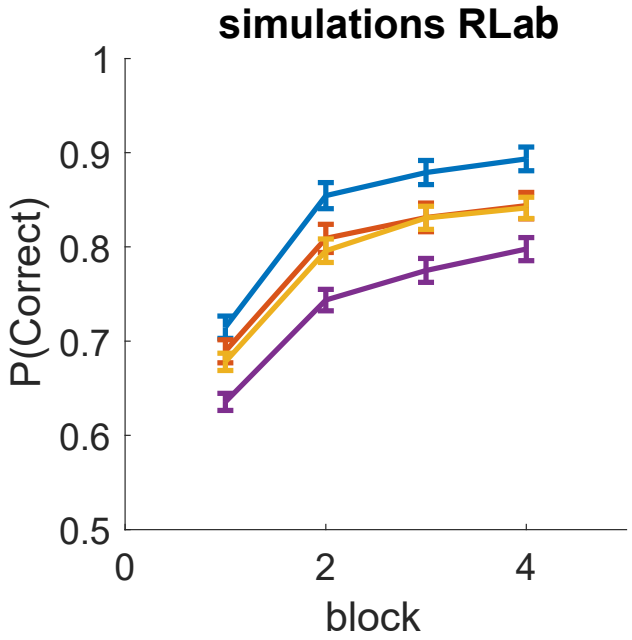
- Can this model produce the behavior patterns I'm expecting from this cognitive process, this experiment, this participant sample?
- Does the behavior of the model produce match participants' behavior?

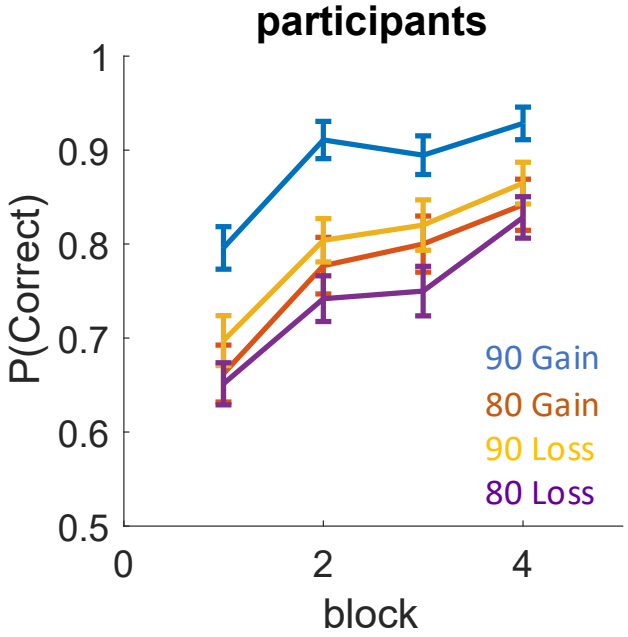
Validate your model!

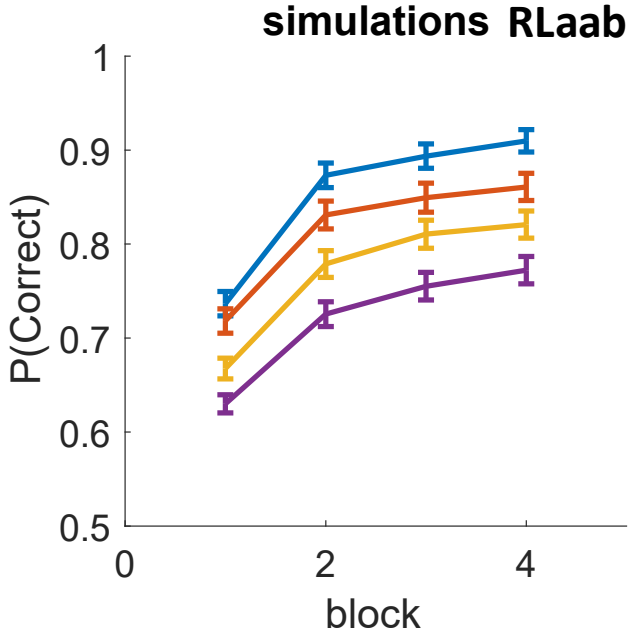
How do I validate my model? (also called *posterior predictive check*)

- Simulate the model on the **exact** experiment, with fit parameters
  - For each participant, with each participant's best fit parameters
  - Multiple times per participant, to average out randomness in choices
- Analyze the behavior of each simulation as if it was a participant
  - Create all the summary measurements you identified as important when you explored participants' behavior
  - Average measures across simulations for each participant, treat this as the validated participant's measure
- Analyze the behavior across participants, and compare to real data







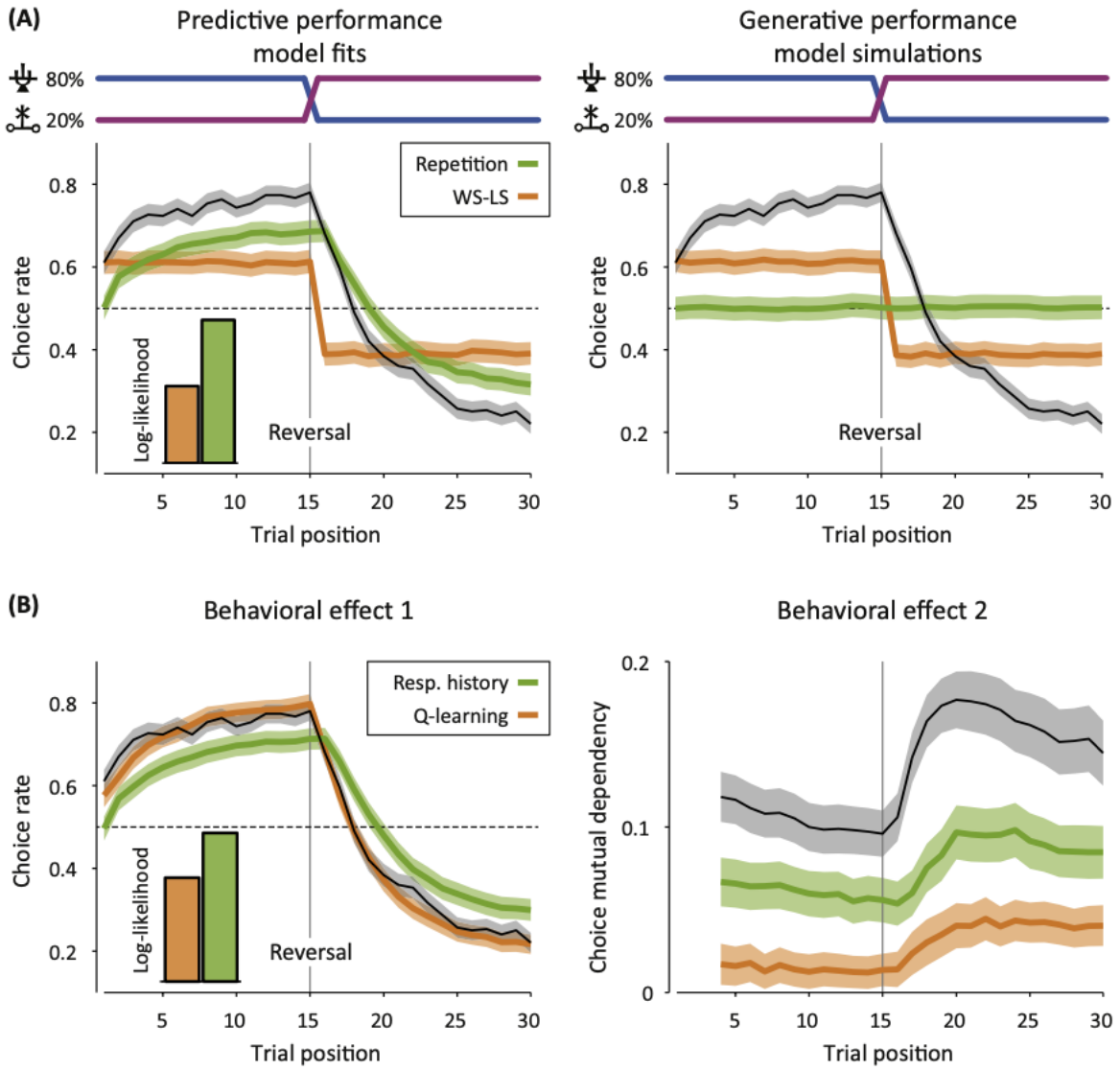


Model validation

- 100 simulations per participant, not looking at group effects yet.
- Good:
  - both capture Learning, Gain/Loss effects
- Not good:
  - underperformance
  - not capturing uncertainty *valence effects
  - Rlaab predicts 80%gain>90% loss learning. Wrong!

Model Predictions and Simulations Are Not Necessarily Related.

- Generating data is different from predicting data (A)
- Quantitative measures don't tell the whole story (A,B)



Palminteri et al, 2017

Simulation for validation, falsification, and model improvement

- Model fitting procedures don't ensure that the model can generate the observed data (*"predictive performance"*).

- Simulations **validate** this and can **falsify** other models.

- When they falsify a model, they indicate where the model can be **improved**.

Questions?

eLIFE elifesciences.org

REVIEW ARTICLE

Ten simple rules for the computational modeling of behavioral data

Robert C Wilson[1,2†*], Anne GE Collins[3,4†*]

Trends in Cognitive Sciences

CellPress

Opinion

The Importance of Falsification in Computational Cognitive Modeling

Stefano Palminteri,[1,2,*,‡] Valentin Wyart,[1,2,*,‡] and Etienne Koechlin[1,2,*]