Café neuro

Probabilistic Models for Characterizing Decision-Making Strategies

State Space models (SSM)

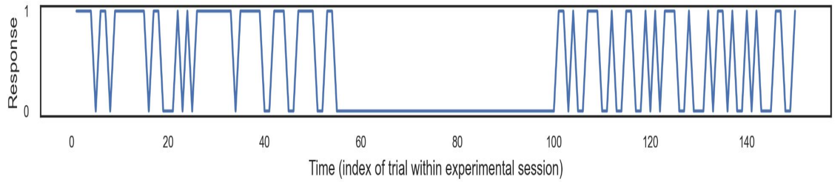
27 septembre 2024

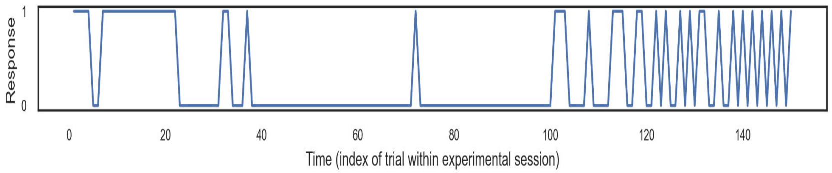
Contents

Linear model to probabilistic models

In our recent study, we employed a reverse correlation experiment using a data-driven approach to find biomarkers of aprosodia in stroke patients by estimating the internal sensory representations and noise. For the representation we computed the difference of the two response class means based on the judgments and to model the relationship between stimulus (input) and response (output). For noise estimation, we used a double-pass procedure, measuring response consistency across repeated trials, and inferred the level of internal noise by simulating how varying degrees of Gaussian noise affected response patterns. These 2 parameters which are static across the experiment, are key factors in differentiating behaviors among stroke patients.

While investigating these parameters, we observed that patient responses exhibited **perseveration**, a repetitive response tendency, which appeared to evolve over time and until now we were assuming a single, unchanging strategy across trials. This unexpected variation led us to investigate on generative model because it describes the process of generating observable data by modeling both the latent states and how they produce the observed responses. We hypothesize the existence of hidden states that influence patient behavior during the experiment. We believe that modeling these hidden states and inferring the parameters from the states could provide deeper insights into the dynamic nature of the patients' responses and their underlying cognitive processes.





GLM

We can model the relationship between the behavioral response (binary outcome) and the stimulus (regressors) using a bernoulli Generalized Linear Model (GLM). The core idea behind Logistic Regression is to transform the output of a linear combination of input features using a sigmoid function, making the predicted values range between 0 and 1.

Input (x): The input x could be the stimulus or some other covariate data that feeds into the model.

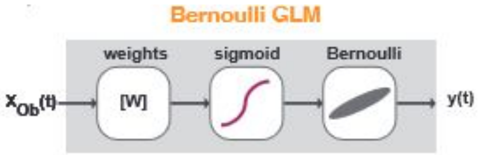
Linear Mapping : The input is linearly transformed using a weight; how the input (stimuli or other covariates) affect the probability of the binary output (decision). The weights are learned during model training are updated in such a way that they maximize the likelihood of the observed data given the model.

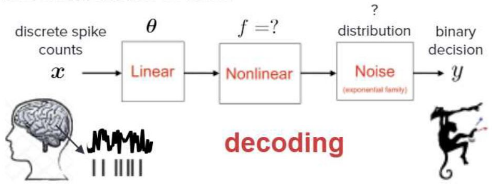
Nonlinearity (σ): After the linear transformation, a nonlinear activation function is applied to map the weighted sum into a probability space. Here, the sigmoid function σ(·) is used, which is typical in logistic regression. This transforms the output into a value between 0 and 1, interpreted as the probability of making one of the two binary decisions- likelihood of the binary outcome .

Noise: Since this is a probabilistic model, noise is incorporated, meaning the final decision isn't deterministic but follows a probability distribution, such as the Bernoulli distribution in this case (binary outcome).

Output (y): The output y is a binary decision (0 or 1) based on the probability computed by the model.

Importantly, in this model, w remains constant over time, suggesting that the response function does not change during stimulus presentation.





$$z(t) = W_1 \cdot x_1(t) + W_2 \cdot x_2(t) + \cdots + W_n \cdot x_n(t) + \epsilon(t)$$

$$p(t) = \sigma(z(t)) = \frac{1}{1+e^{-z(t)}} = \frac{1}{1+e^{-(W \cdot x_{\text{obs}}(t)+\epsilon(t))}}$$

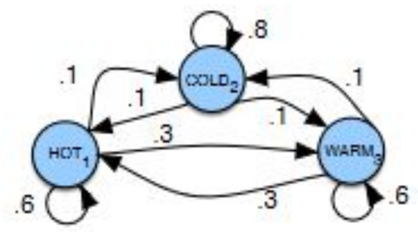
$$P(y(t)=1 \mid x(t)) = \frac{1}{1+e^{-(x(t)\cdot W+\epsilon(t))}}$$

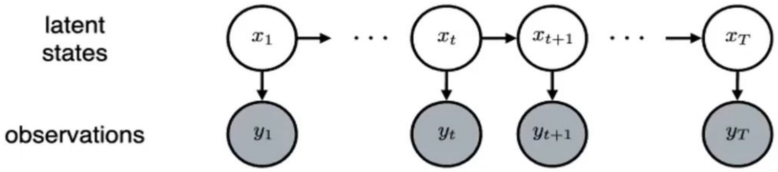
Hidden Markov Models for the Stimulus-Response Relationships of Multistate Neural Systems, Sean 2011

State space models (HMMs)

With simple statistical framework as generalized linear models which does not change in time(static), we can transform the input into a probability of output , to explain many of observed patterns but not different types of behavior that are difficult to predict; particularly when the subjects' internal states or external conditions shift during the experiment. These kind of missing variance can be explained by allowing an internal variable that alters with time like hidden markov models. The hidden states in these models are found through a fitting procedure rather than through manual annotation.

- **Hidden Markov Model (HMM)** is a statistical Markov model in which the system being modeled is assumed to be a Markov process with unobserved (i.e. hidden) states.
- In probability theory, a Markov model is a stochastic model used to model randomly changing systems. It is assumed that future states depend only on the current state, not on the events that occurred before it (the Markov property).





Nuts and Bolts of Modern State Space Models,
Scott Linderman, 2023

Vanilla HMM

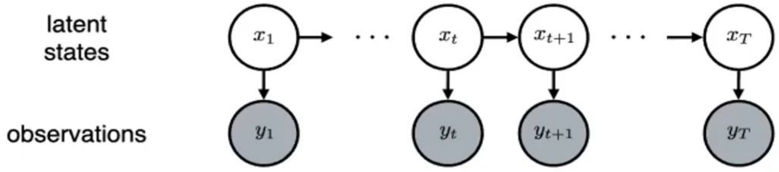
A **Hidden Markov Model (HMM)** is a statistical model in temporal space, which the observed events (observations) are influenced by hidden or **latent states** that cannot be directly observed. The model has two components:

Latent State Process: Follows the **Markovian property**, meaning the current state only depends on the previous state. The transitions between these states are governed by a **transition probability matrix** $A \in R^{K \times K}$, where each element represents the probability of moving from state j to state k: $\mathbf{p(z_{t+1}=k \mid z_t=j)}$

Observation Model (emission probabilities): The observable data at time t, x_t, depends only on the current latent state z_t, and not on previous observations: $\mathbf{p(x_t|x_1,x_2,\ldots,x_{t-1},z_t)=p(x_t|z_t)}$. This can be modeled using **Gaussian Emission Models**, where each hidden state generates observations following a Gaussian (normal) distribution. In this case, the emission probability for a given hidden state z_t is modeled as:

$$p(x_t \mid z_t = k) = \mathcal{N}(x_t; \mu_k, \Sigma_k)$$

Here, μ_k is the mean and Σ_k is the covariance matrix of the Gaussian distribution associated with hidden state k. Alternatively, other distributions like **Gaussian Mixture Models (GMMs)** can be used to capture more complex state-dependent behaviors.



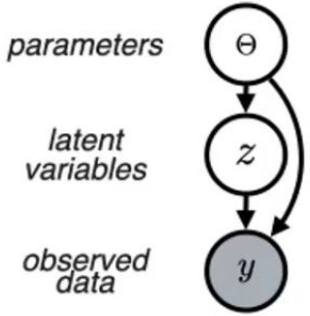
Nuts and Bolts of Modern State Space Models, Scott Linderman, 2023

Fitting HMM

We fit an **HMM (Hidden Markov Model)** to uncover hidden patterns or "states" that drive observable events in time series data. These hidden states, though not directly visible, influence the data we observe. To estimate the parameters of an HMM, we use the **Expectation-Maximization (EM) algorithm**, which alternates between two steps:

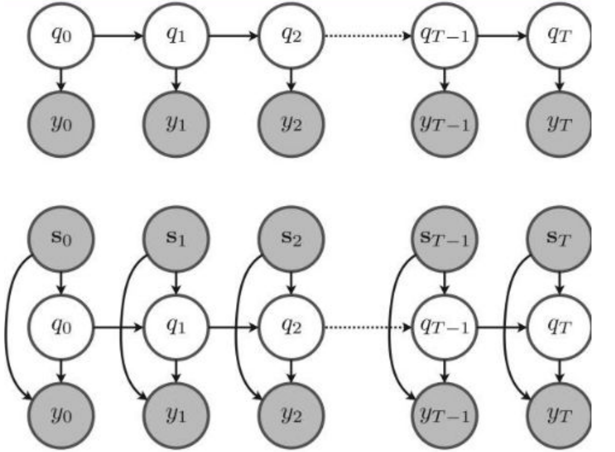
1. **E-step**: We estimate the probability of being in each hidden state at every point in time based on the observations. This is done using the **Forward-Backward Algorithm**, which calculates the posterior probabilities of hidden states (probability distribution over the hidden states at a given time, given the entire sequence of observed data up to that point).
2. **M-step**: Using these probabilities, we update the model's parameters, including the transition matrix (how hidden states change over time) and the emission parameters (how hidden states affect the observations). The goal is to maximize the likelihood of the observed data.

A specific version of the EM algorithm for HMMs is called the **Baum-Welch algorithm**, which is used to iteratively refine the model's parameters.



IO-HMM

An **Input-Output Hidden Markov Model (IO-HMM)** is an extension of the standard HMM that incorporates input variables, allowing it to model the relationship between input covariates and both the hidden states and the observed outputs. Unlike standard HMMs, which only account for hidden states and observations, an IO-HMM conditions both the hidden states and the observable outputs on external inputs.



- **State Dependencies on Inputs**: In an IO-HMM, the hidden states can persist over a large number of trials and are influenced by external inputs such as stimuli or covariates. This allows the model to capture the impact of input variables on the evolution of hidden states over time.
- **Output Dependencies on Inputs**: The observed outputs $Y_{1:T}$ are also conditioned on input variables $X_{1:T}$, allowing the model to account for how the input affects both the hidden states and the outputs.
- **Key Difference from HMM**: In a standard HMM, the hidden states and outputs depend only on previous states and observations. In contrast, the IO-HMM framework adds input dependencies, making it more flexible in modeling systems where external variables influence the *transitions and emissions*.
- **Single Model Training**: Unlike HMMs, where a different model might be trained for each class, an IO-HMM trains a single model to handle multiple classes by incorporating the input variables directly into the state and output distributions.

Hidden Markov Models for the Stimulus-Response Relationships of Multistate Neural Systems, Sean 2011

GLM-HMM

The GLM-HMM is a latent internal state model, combining Hidden Markov Models (HMMs) with multiple sets of per-state Generalized Linear Models (GLMs). Like the IO-HMM, which conditions outputs on input variables, the GLM-HMM incorporates external covariates to influence the state-specific decision process.

- **Latent States**: Represent different cognitive or decision-making strategies, with each state parameterized by a unique set of GLM weights.
- **Discrete States**: A discrete random variable models internal states over time, governed by an HMM.
- **Mapping**: A sigmoidal function (logistic regression) maps the human's binary decision to a weighted representation of covariates (e.g., stimulus, trial history, bias), just like in the IO-HMM, but with state-specific GLM weights.
- **GLM as Emission Model**: Each state-specific GLM describes how covariates are integrated to make a decision. The GLM calculates the output probability in each state based on a linear combination of the covariates.
- **Transition Between States**: Probabilistic transitions between states occur after each trial, governed by a fixed transition probability matrix, allowing states to evolve over time.

By combining the strengths of both GLMs and HMMs, the GLM-HMM captures both the hidden state dynamics and how participants make decisions based on the covariates, unlike the IO-HMM that transition probabilities between hidden states and emission probabilities of the outputs are both conditioned on the external inputs but hidden state transitions may be independent of inputs in GLM-HMM.

Bernoulli GLM component of a **GLM-HMM:**

$$p(y_t = 1 | \mathbf{x}_t, z_t = k) = \frac{1}{1 + e^{-\mathbf{x}_t \cdot \mathbf{w}_k}}$$

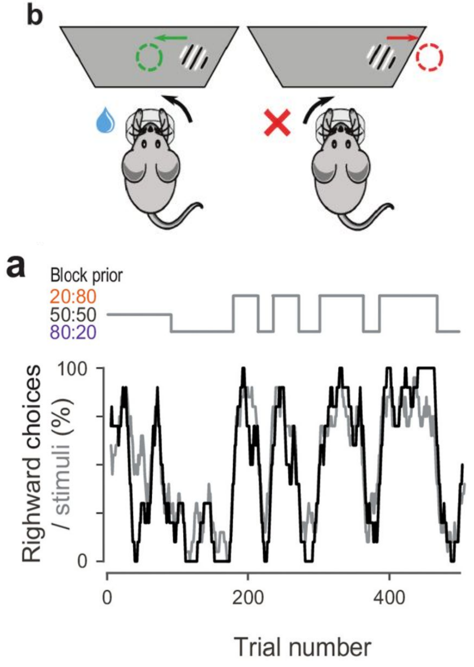
Probabilistic Models for Characterizing Animal Learning and Decision-Making - Ashwood 2022

Example IBL dataset

In this task, mice detect the direction of a Gabor patch on the screen and turn a wheel to the center to indicate whether the stimulus direction is to the right or left. The researchers (Ashwood et al. 2021) applied the **GLM-HMM** model to describe the behavior of mice making right–left decisions based on the contrast of a visual stimulus (input).

Mice exhibit at least two distinct internal states. In one state of **high engagement**, they perform the perceptual task with high sensitivity and low bias, accurately responding to the stimulus. However, in a **lower-performance state**, the mice tend to ignore even easy-to-discriminate stimuli and make biased responses, favoring one direction regardless of the stimulus. Mice often switch between these states, alternating several times and sometimes staying in one state for tens of trials or more.

This analysis using **GLM-HMM** effectively captures these hidden states, providing insights into how internal cognitive factors affect decision-making behavior.



Standardized and reproducible measurement of decision making in mice - international brain laboratory 2021

Lapse model

In the classic lapse model (2-state GLM-HMM), the decision-making process is modeled with two states: "engaged" and "lapse."

- **Engaged**: In this state, the decision-making is based on the stimulus with a higher probability of making a correct choice (P(correct) = 0.8).
- **Lapse**: In this state, decisions are assumed to be random, meaning the individual might still make a correct choice purely by chance (P(correct) = 0.2), but without being influenced by the stimulus.

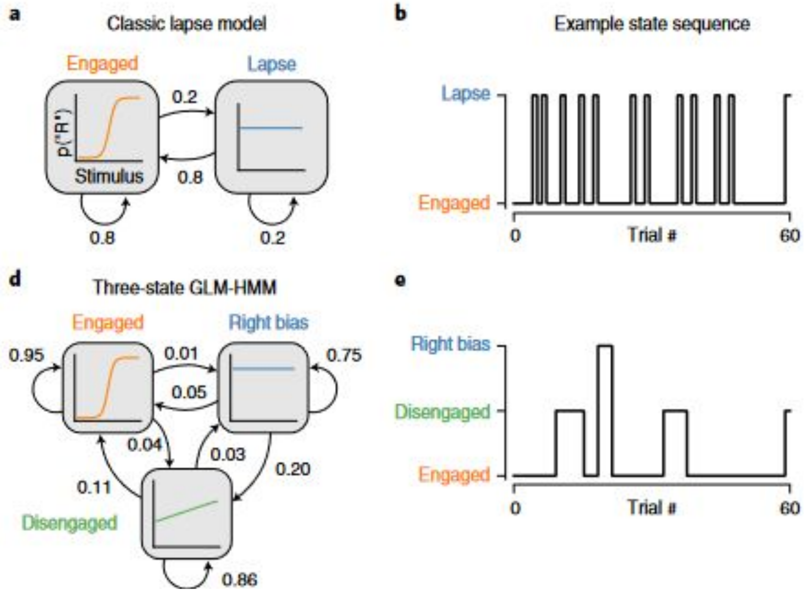
The lapse model assumes that animals alternate between these two strategies. However, this model has limitations, as it treats lapses as independent, random events, not accounting for the possibility that lapses might still be weakly influenced by the stimulus or other external factors and it lapse state last for only one trial not more.

In contrast, the 3-state GLM-HMM model provides a more detailed view, incorporating:

- **Engaged**: A high-performing state where decisions are strongly guided by the stimulus.
- **Disengaged**: A state where decision-making is disconnected from the stimulus (similar to the lapse state but distinct in dynamics).
- **Right Bias**: A third state that represents a bias in decision-making towards a particular choice (e.g., right-sided decisions).

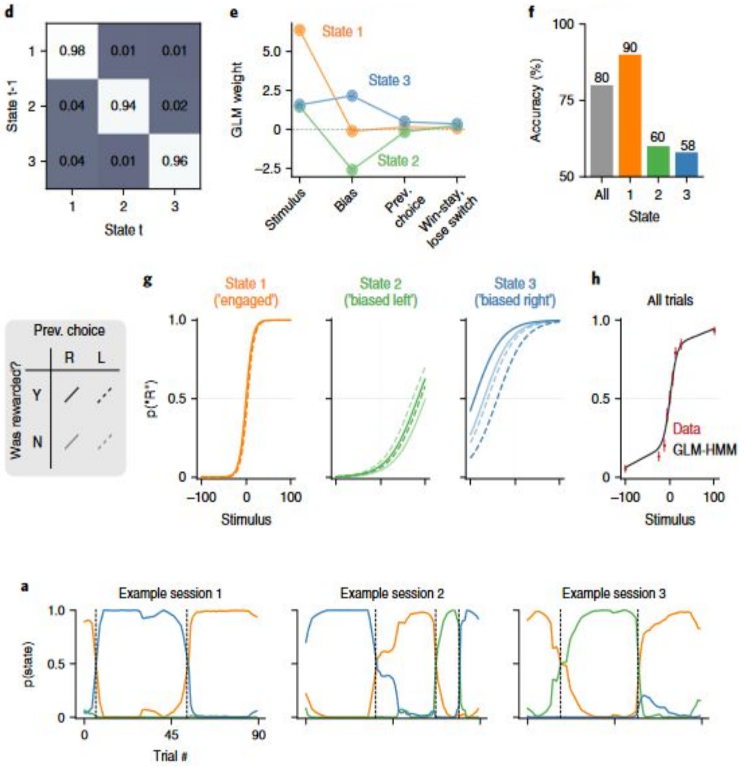
And contrary to lapse model , states intended to persist for many trials in a row.

$$p(y_t = 1 \mid \mathbf{x}_t) = \begin{cases} \frac{1}{1+e^{-\mathbf{w}\cdot\mathbf{x}_t}}, & z_t = \text{“engaged”} \\ \frac{\gamma_r}{\gamma_r+\gamma_l}, & z_t = \text{“lapse”}, \end{cases}$$



Probabilistic Models for Characterizing Animal Learning and Decision-Making - Ashwood 2022

IBL analysis



Probabilistic Models for Characterizing Animal Learning and Decision-Making - Ashwood 2022

Fitting GLM-HMM

As in HMM, also the GLM-HMM uses the **Expectation-Maximization (EM) algorithm** to estimate the parameters and fit the model to behavioral data. As said before this process involves two main steps:

- **E-step**: The model calculates the expected values of the hidden states (posterior probabilities) based on the current parameters using the **forward-backward algorithm**.
- **M-step**: Using these expectations, the model updates its parameters (GLM weights (emission probabilities), transition matrix) to maximize the likelihood of the observed data.

Key elements of the fitting process:

- **Model Parameters**: θ = {w_z (state-specific weights), A (transition matrix), π (initial state distribution)}.
- **Likelihood Estimation**: The EM algorithm iteratively improves the fit by calculating the log-likelihood of the data, based on the joint probability distribution of the states and observed responses.

$$P(z_{1:T}, y_{1:T} | \theta) = P(z_1 | \pi) \prod_{t=2}^{T} P(z_t | z_{t-1}, A) \prod_{t=1}^{T} P(y_t | z_t, x_t, w_z)$$

The model assigns each trial to its most likely hidden state and can show when participants switch between strategies over time.

EM algorithm - E step

1. E-Step (Expectation Step)

In the E-step, we compute the posterior distribution over the hidden states (i.e., the probability of being in each hidden state at each time step) given the observed data and the current model parameters.

Forward Algorithm (Part of the E-step)

filtering, The forward algorithm is used to compute the probability of the system being in a specific state at time t , given all the observations up to time t. It's a recursive algorithm and it takes into account both the previous state's probability and the transition probabilities to arrive at the current state's probability

$$\alpha_t(j) = P(y_1, y_2, ..., y_t, z_t = j|\theta)$$

Backward Algorithm (Part of the E-step)

Smoothing ,The backward algorithm is used to compute the probability of observing the data from time t+1 to the end, conditioned on the hidden state at time t. starts from the final time step and moves backward in time, refining the state probabilities by including information from the future observations.

$$\beta_t(i) = P(y_{t+1}, y_{t+2}, ..., y_T|z_t = i, \theta)$$

Posterior State Probabilities (E-Step Output)

After calculating the forward and backward probabilities, we can compute the **posterior probability** of being in state k at time t, denoted as:

$$P(z_t = k \mid y, \theta) = \gamma_t(k) = \frac{\alpha_t(k)\beta_t(k)}{\sum_{k'} \alpha_t(k')\beta_t(k')}$$

Where $\gamma_t(k)$ is the posterior probability of being in state k at time t, given the data and the current model parameters.The M-step consists of updating the parameters to **maximize the expected log-likelihood** of the data, given the posterior distribution of the hidden states computed in the E-step.

The EM algorithm repeats these steps until the log-likelihood converges to a local optimum.

EM algorithm - M step

2. M-Step (Maximization Step)

In the M-step, we update the model parameters θ = {A, π, W} using the posterior state probabilities computed in the E-step. This step maximizes the expected log-likelihood of the data:

- **Maximizing the Transition Matrix (A)**: The transition matrix A defines the probabilities of switching between states. It is updated based on the expected number of transitions from state *i* to state *j*, divided by the expected number of times the system is in state *i*.
- **Maximizing the GLM Weights (W)**: Each state has its own set of GLM weights *w*, which define the relationship between input covariates and observed outputs (responses). These weights are updated to maximize the likelihood of observations, given the current posterior state probabilities.
- **Maximizing the Initial State Probabilities (π)**: The initial state distribution *π* is updated using the posterior probability of being in each state at time *t = 1*.

Convergence:
The EM algorithm guarantees an increase in log-likelihood with each iteration, alternating between E and M steps until convergence(until the change in the log-likelihood between iterations is smaller than a predefined threshold -the tolerance- level).

Since the algorithm only finds a local optimum, multiple initializations of weights and the transition matrix can be used to find the best solution or the global optimum for the log-likelihood.

Cross-validation is employed to ensure the model captures subtle behavior patterns by comparing the log-likelihoods across different GLM-HMM variants (e.g., 1-state, 2-state lapse model, 3-state GLM-HMM).

Comparison of GLM, HMM and GLM-HMM

Aspect	GLM	HMM	GLM-HMM
Observation Model	models the relationship between input covariates and the outcome using a linear combination of covariates passed through a link function (e.g., sigmoid for logistic regression)	Gaussian or discrete distributions to model the observation probabilities directly from the hidden states.	Uses GLMs to model the observation probabilities. Each hidden state is associated with a unique GLM that links external covariates to observations.
Observation Probabilities	outcome is modeled as a probabilistic function of the covariates, using distributions like Bernoulli	emission probabilities, typically parameterized by mean (μ) and variance (σ^2) or categorical probabilities.	GLMs predict the probability of making a specific observation based on external covariates and a linear combination of state-dependent weights w_j
Latent States	No latent states.	Hidden states represent discrete underlying processes (e.g., weather, mood, etc.). Transitions between states follow a first-order Markov process, and state emissions are independent of input covariates.	Similar hidden states, but the observation probabilities are influenced by external covariates.
Transition Matrix (A)	No transition matrix.	A transition matrix governs the probability of transitioning between hidden states, and these probabilities are fixed	Similar transition matrix, but the GLM-HMM may allow input-driven transitions in more advanced variants
EM	N/A	E-step: Estimate hidden states based on observations, (forward-backward: Used to compute latent state probabilities) M-step: Update transition matrix and emission probabilities to maximize log-likelihood	E-step: Estimate latent states (forward-backward: Used to compute latent state probabilities) and M-step: Update transition matrix and GLM weights to maximize log-likelihood
initialization	N/A	Required for transition matrix, emission probabilities, and initial state probabilities π	Required for transition matrix, state-dependent GLM weights, and initial state probabilities

Application by simulation -generative

We can use SSM: state space models library to set up the parameters of GLM-HMM model.

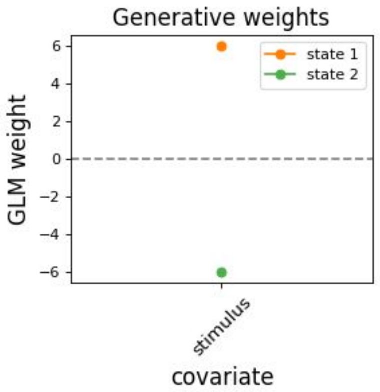
As an example, we define the following parameters:

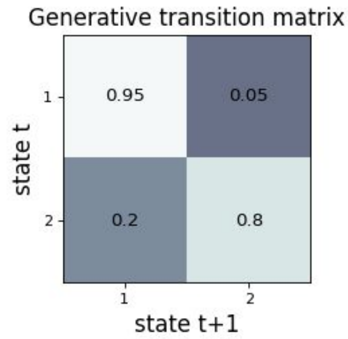
- num_states = 2: Two hidden states.
- obs_dim = 1: One observable dimension (binary responses 0 or 1).
- input_dim = 1: One input variable (the stimulus).

And initialize **transition matrix** and **GLM weights** which define how input covariates (like stimuli) affect the response probabilities (observations) for the generative model.

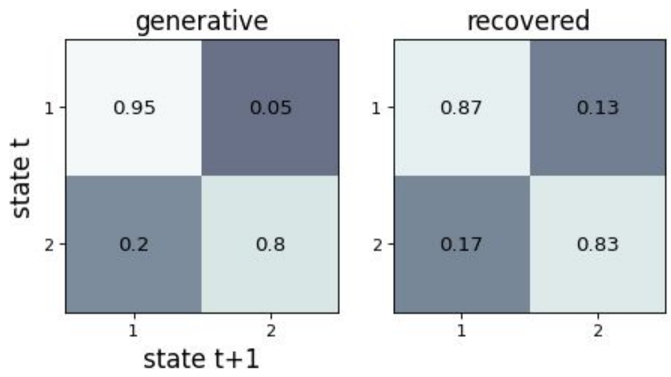
true_glmhmm=ssm.HMM(num_states,obs_dim,input_dim,observations="input_driven_obs", ...)

The true log-likelihood of the generated data based on these parameters is computed, and its value gives us an idea of how well the model represents the data.



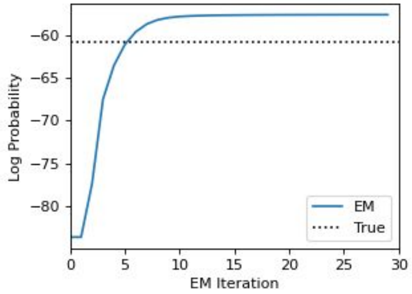


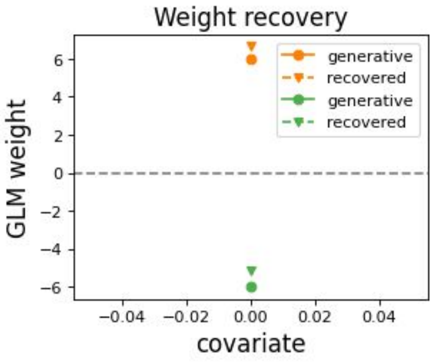
Application by simulation- recovered by fitting



1. Left hand plot shows the transition matrix used to generate the data, while the right hand plot shows the transition matrix recovered during model fitting. The close match between them shows that the EM algorithm recovered the underlying dynamics well.

2. How the **log-likelihood** of the model improved as the EM algorithm progressed through its iterations and how the GLM weights are recovered after model fitting.





Application by simulation- posterior state probability

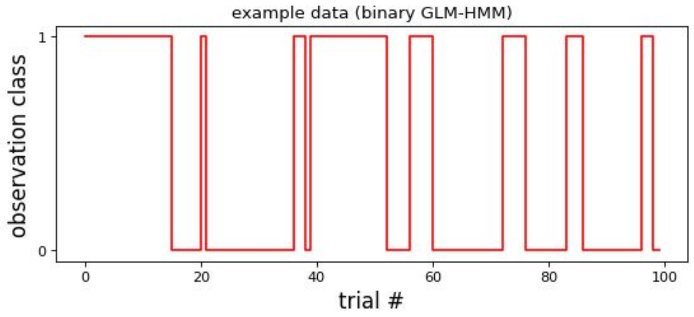
1. Shows the **binary responses** over 100 trials, where the red line represents the observation class (0 or 1). This data is generated from the GLM-HMM based on the transition matrix and GLM weights.

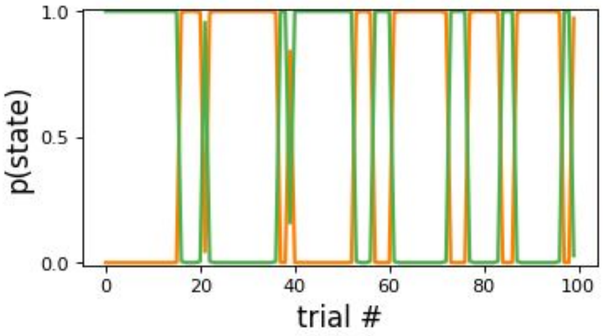
$$P(\text{response} = 1 \mid \text{State } 1) = \frac{1}{1+e^{-6\cdot 1}} = \frac{1}{1+e^{-6}} \approx 0.9975$$

$$P(\text{response} = 1 \mid \text{State } 2) = \frac{1}{1+e^{-(-6)\cdot 1}} = \frac{1}{1+e^{6}} \approx 0.0025$$

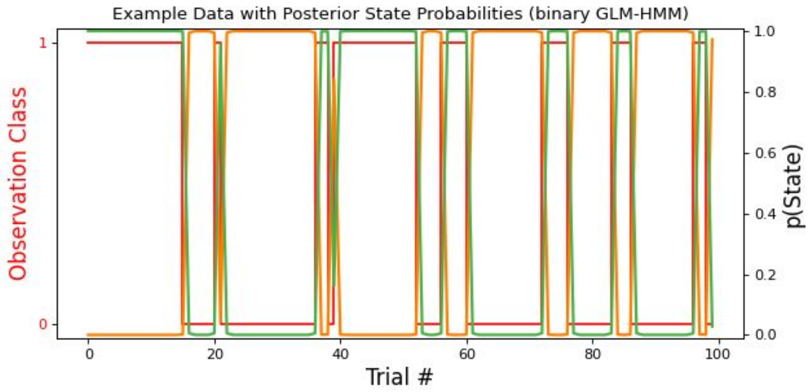
2. Shows the **posterior probabilities** of being in each of the two hidden states at each trial. Indicate which state the system likely occupies at each trial. The sharp transitions between the green and orange lines reflect switches between states.

$$P(z_t = k \mid y, \theta) = \gamma_t(k) = \frac{\alpha_t(k)\beta_t(k)}{\sum_{k'} \alpha_t(k')\beta_t(k')}$$

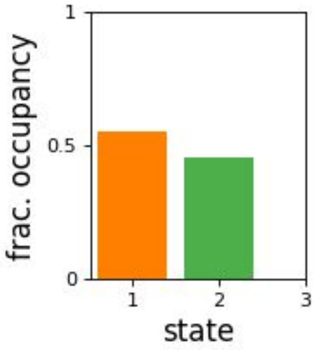




Application by simulation

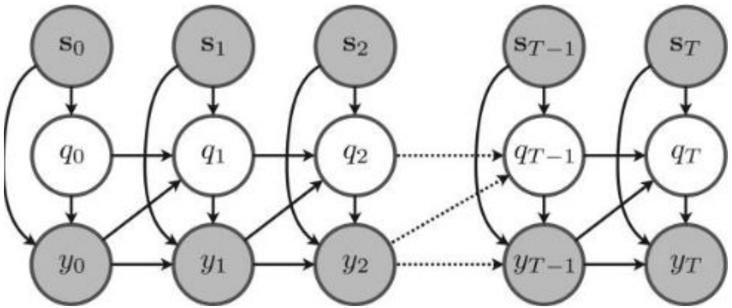


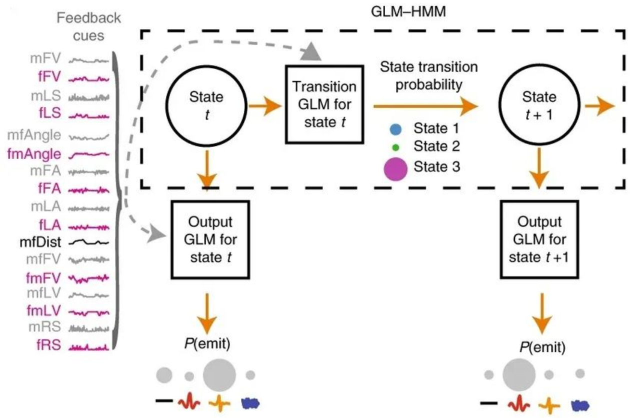
This occupancy reflects how much time the system spends in each state over the course of 100 trials. In this case, the system slightly prefers **State 1**, which produces more responses of 1.

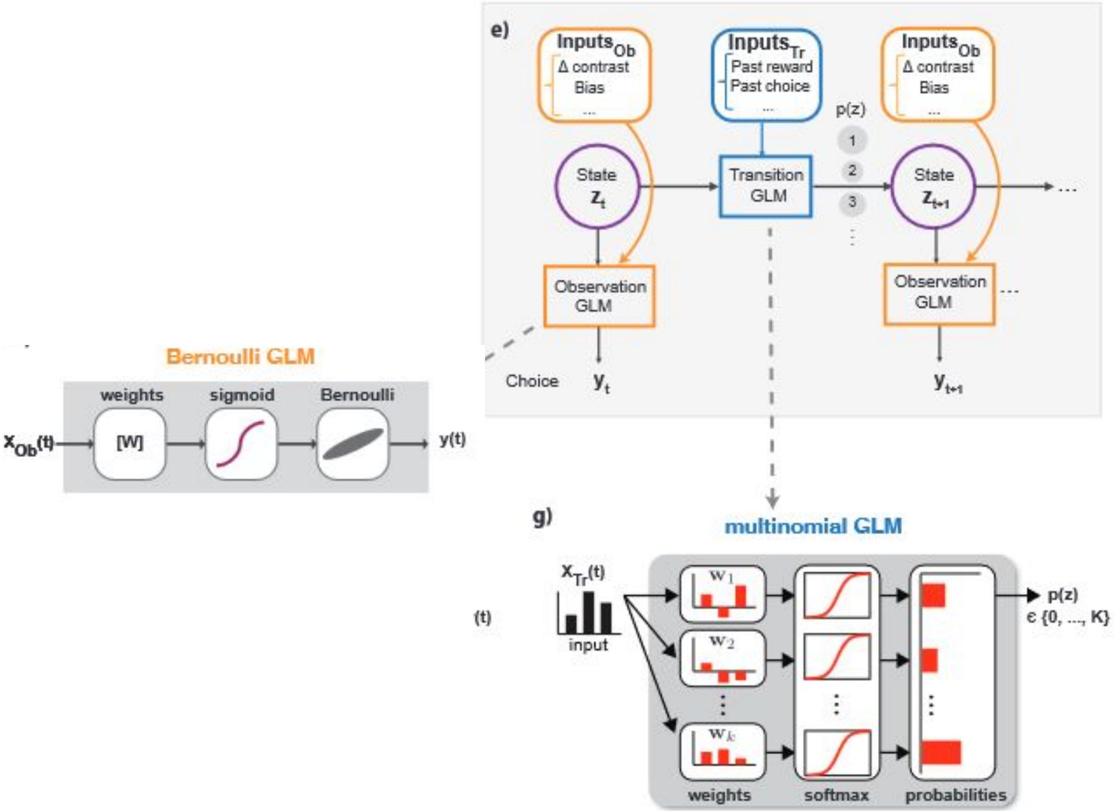


More about state space models

- Possibility of correlation the states with external variables as response time.
- AR-HMM: If inputs are postural data (videos), and observation are more gaussian than binary
- Psytrack: One could imagine, for example, that a state governing the participant's degree of engagement drifts gradually over time, and that the GLM-HMM simply divides these continuous changes into discrete clusters. To address this possibility, we fit the data with PsyTrack, a psychophysical model with continuous latent states, model describes sensory decision-making using an identical Bernoulli GLM, but with dynamic weights that drift according to a Gaussian random walk (defining the kind of human's strategy -discrete or continuous)
- GLM-transitions: We can model the transition by GLM, allowing state transitions to depend on covariates, making the model more adaptive and capable of capturing complex time-varying dependencies in behavioral data (something more alike to IO-HMMs)

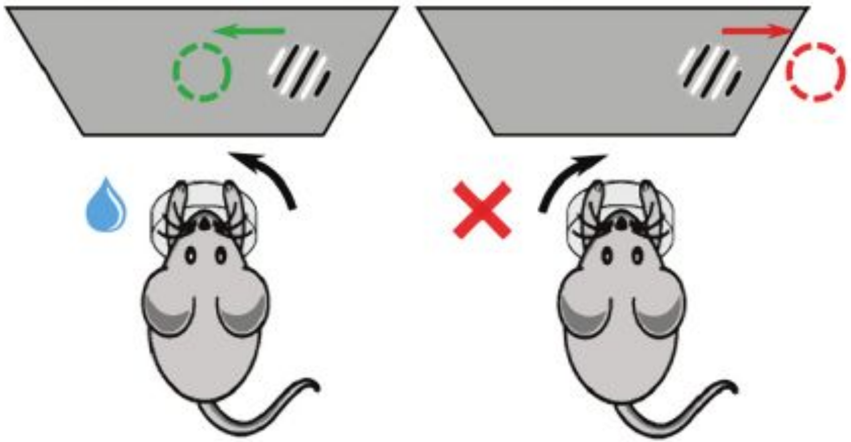


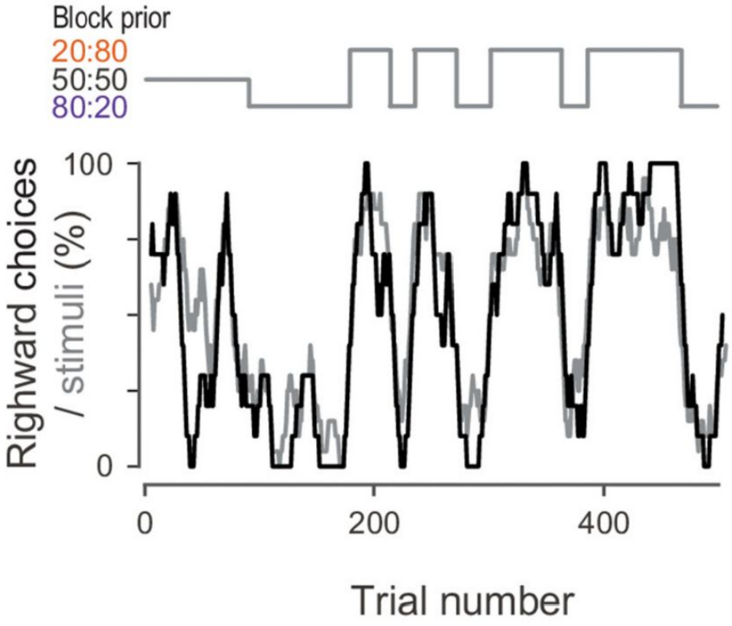




e)
Inputs$_{Ob}$
Δ contrast
Bias
...
Inputs$_{Tr}$
Past reward
Past choice
...
p(z)
1
2
3
State
z_t
Transition
GLM
State
z_{t+1}
Observation
GLM
Choice
y_t
y_{t+1}
Bernoulli GLM
weights
sigmoid
Bernoulli
$x_{Ob}(t)$
[W]
y(t)
g)
multinomial GLM
$x_{Tr}(t)$
input
w_1
w_2
w_k
weights
softmax
probabilities
p(z)
ϵ {0, ..., K}
(t)

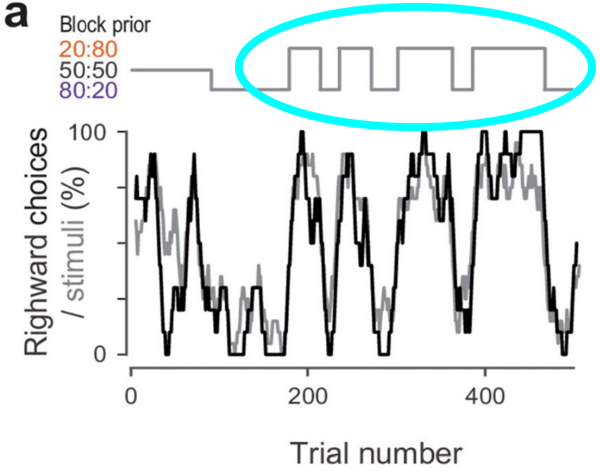
Task Structure





Scientific Question

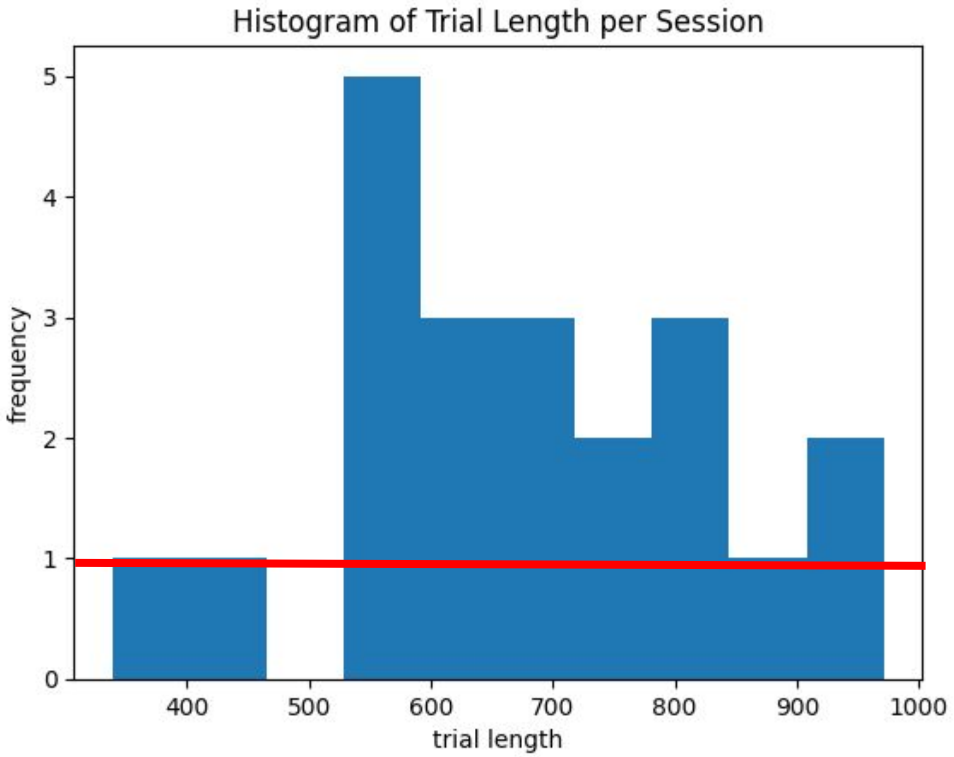
How many different decision-making strategies mice have in a visual discrimination task that varies between stimuli probabilities (20:80 and 80:20)?



Data Selection

- 80:20 and 20:80 blocks only

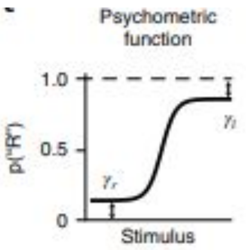
- Same number of trials per session



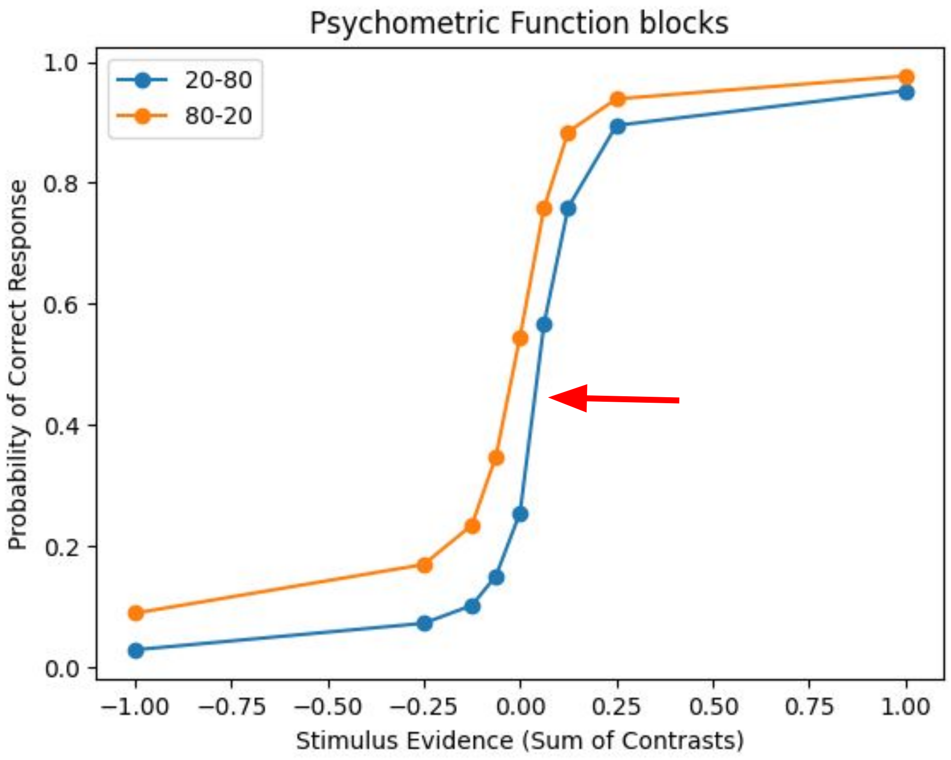
Psychometric Function

- There is a shift of the function towards the left choice (bias left)

  > bias as a regressor of the performance



Ashwood ZC et al.,Nature neuroscience (2020)



Hidden Markov Model

Inputs selection (regressors):

- contrast
- bias
- previous choice
- win-stay/lose-switch

```
# Make a GLM-HMM
true_glmhmm = ssm.HMM(num_states, obs_dim, input_dim, observations="input_driven_obs",
                      observation_kwargs=dict(C=num_categories), transitions="standard")
```

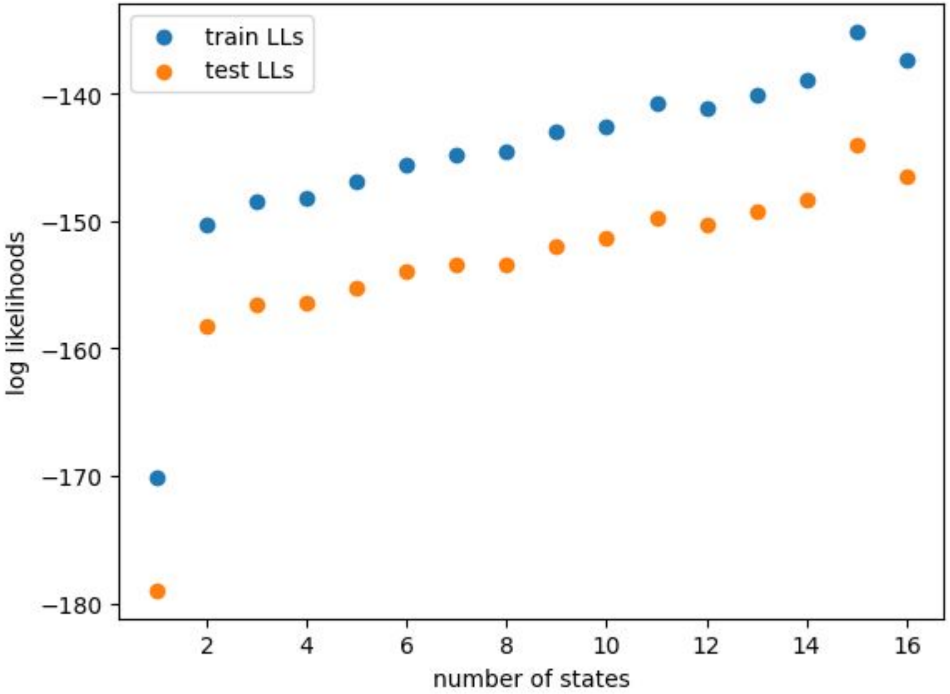
number of states: [2, 4, 6] ?

observed dimension: 1

number of categories: 2 [-1; 1]

input dimension: 2

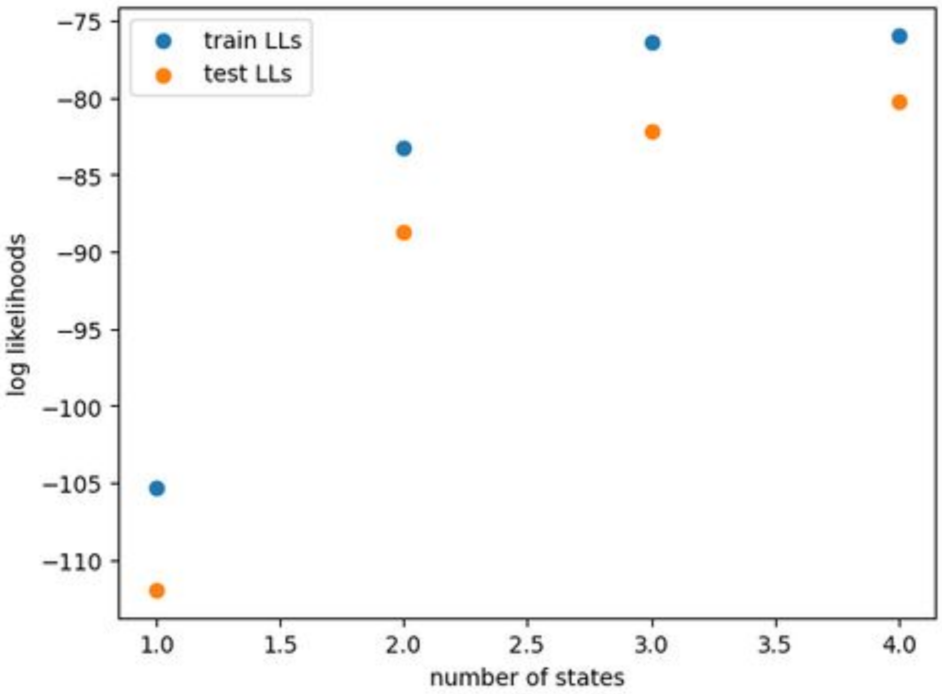
Likelihood



Leave-one-out cross validation.

Bigger step from state_1 and state_2, which might better explain the mouse’s strategy.

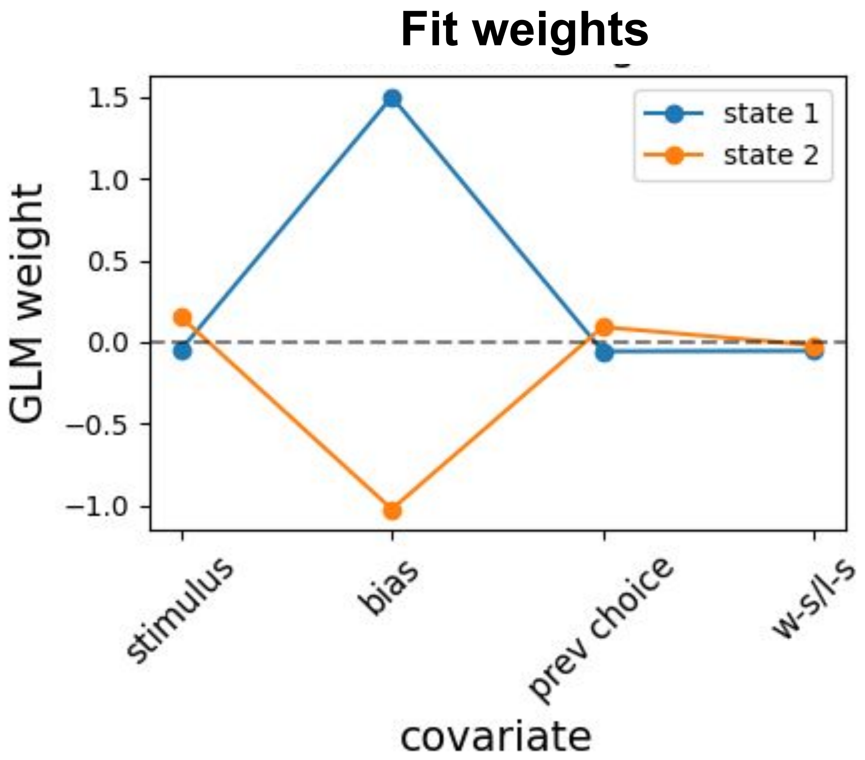
Likelihood



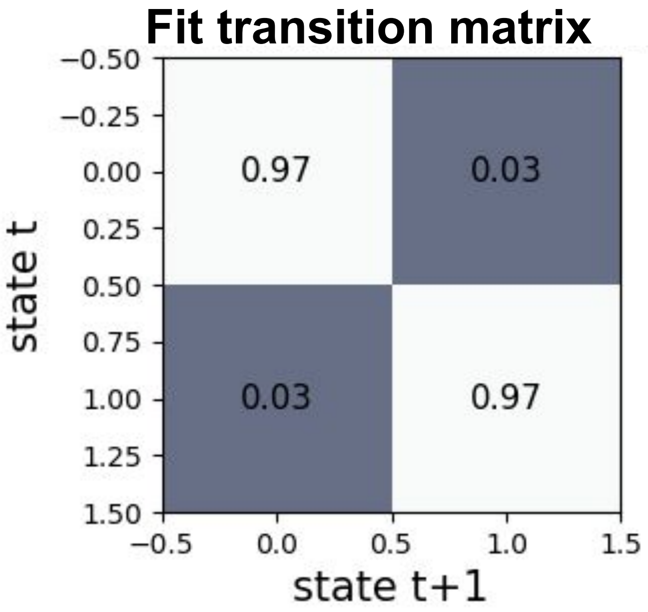
Leave-one-out cross validation.

Bigger step from state_1 and state_2, which might better explain the mouse’s strategy.

Weight fits



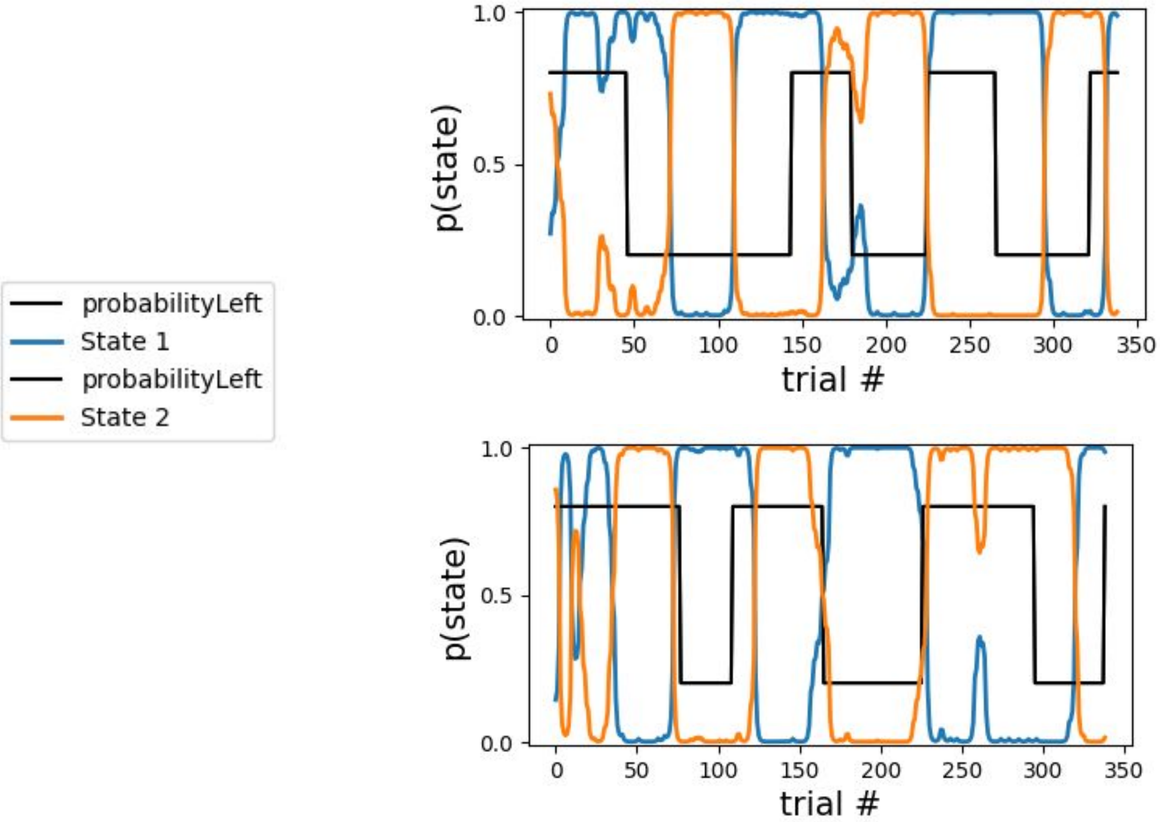
Fit weights
state 1
state 2
1.5
1.0
0.5
0.0
−0.5
−1.0
GLM weight
stimulus
bias
prev choice
w-s/l-s
covariate



Fit transition matrix
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00
1.25
1.50
state t
0.97
0.03
0.03
0.97
−0.5
0.0
0.5
1.0
1.5
state t+1

Posterior State Probabilities

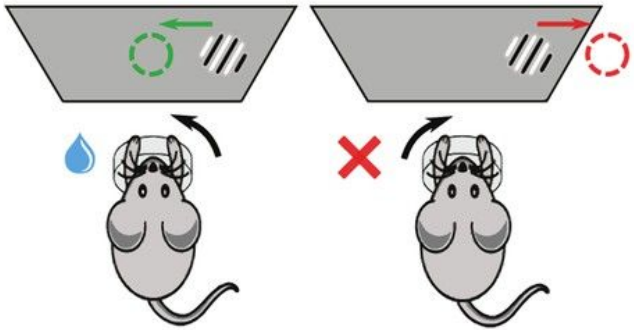
The probability of the animal being in state k at trial t

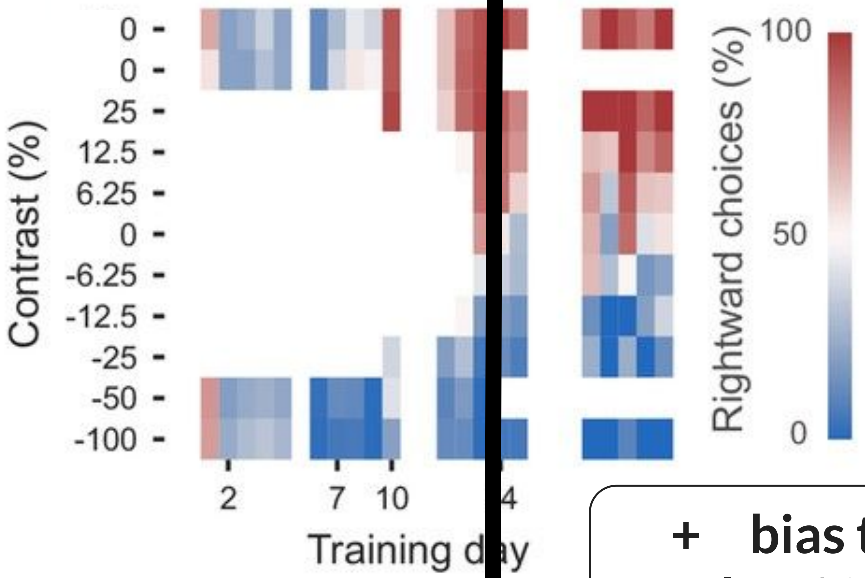


vision

vision

recent experience





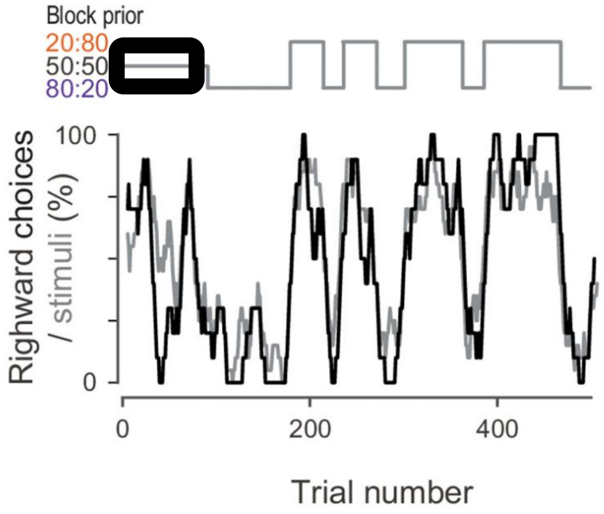
+ bias to left/right

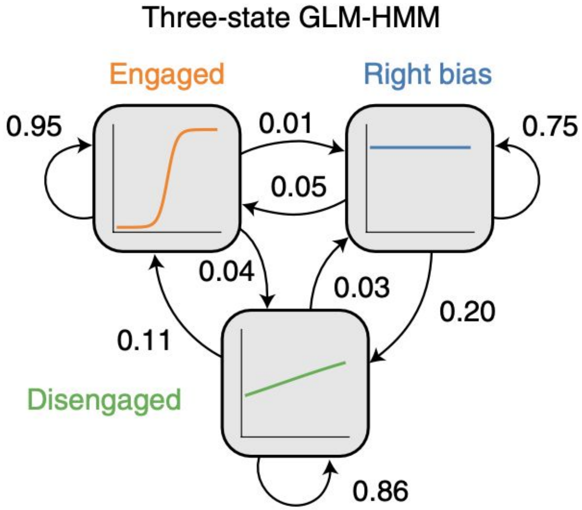
training task

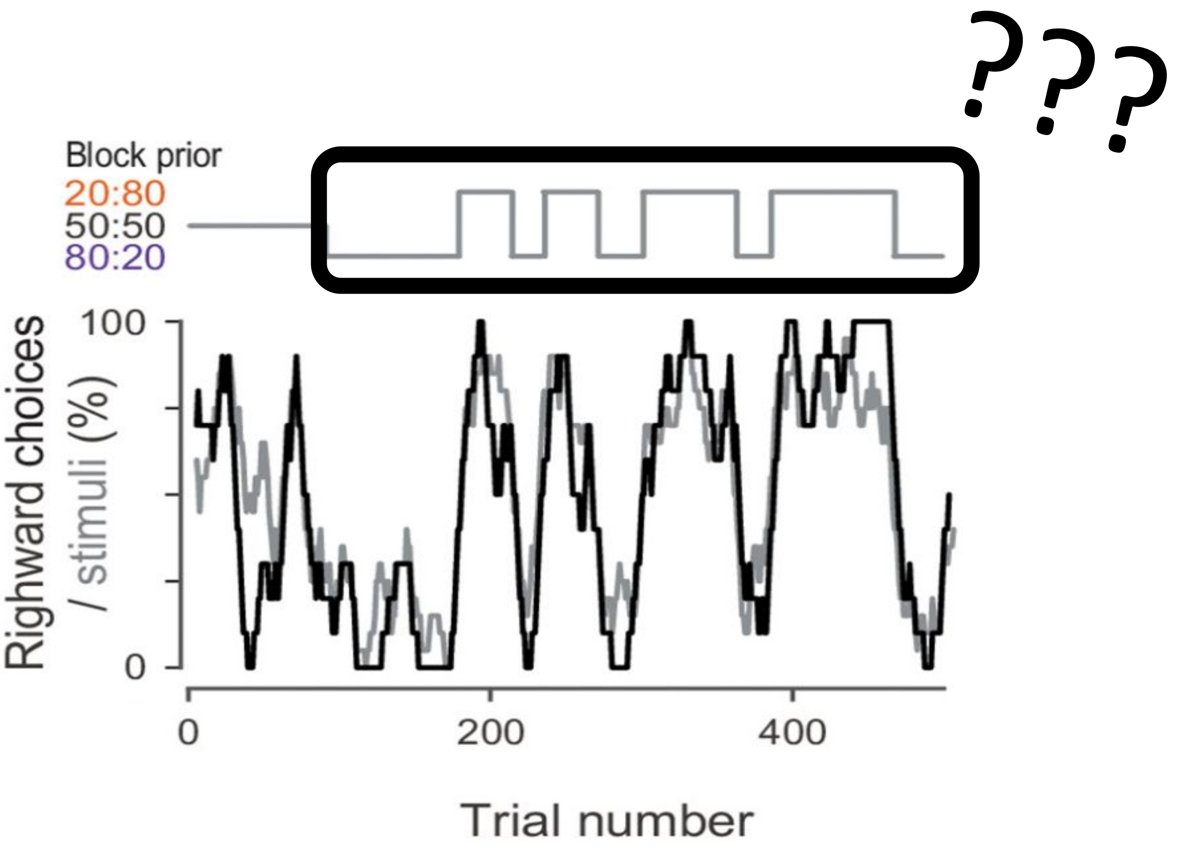
full task

Mice alternate between discrete strategies during perceptual decision-making

Zoe C. Ashwood[1,2] ✉, Nicholas A. Roy[2], Iris R. Stone[2], The International Brain Laboratory*, Anne E. Urai[3], Anne K. Churchland[4], Alexandre Pouget[5] and Jonathan W. Pillow[2,6] ✉

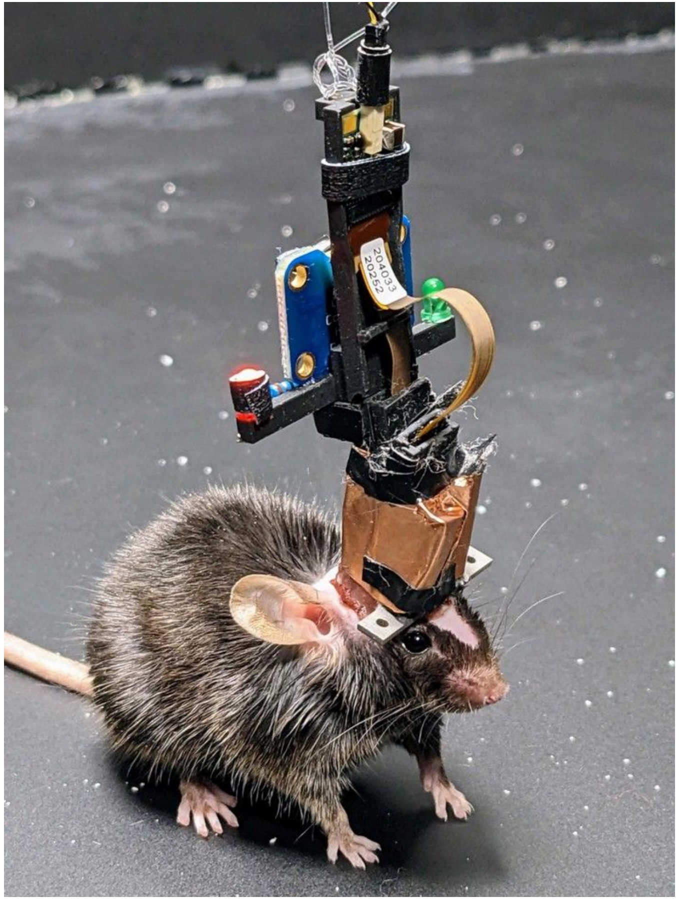




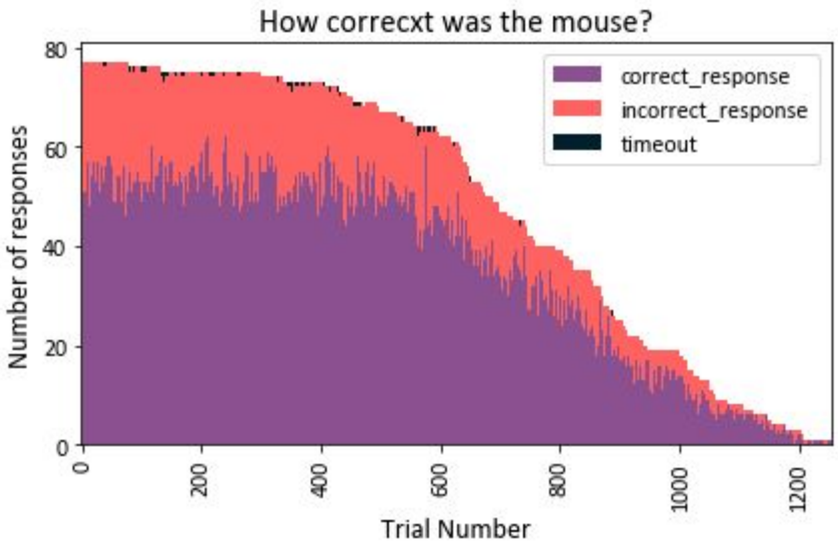

???
Block prior
20:80
50:50
80:20
Righward choices
/ stimuli (%)
100
0
0
200
400
Trial number

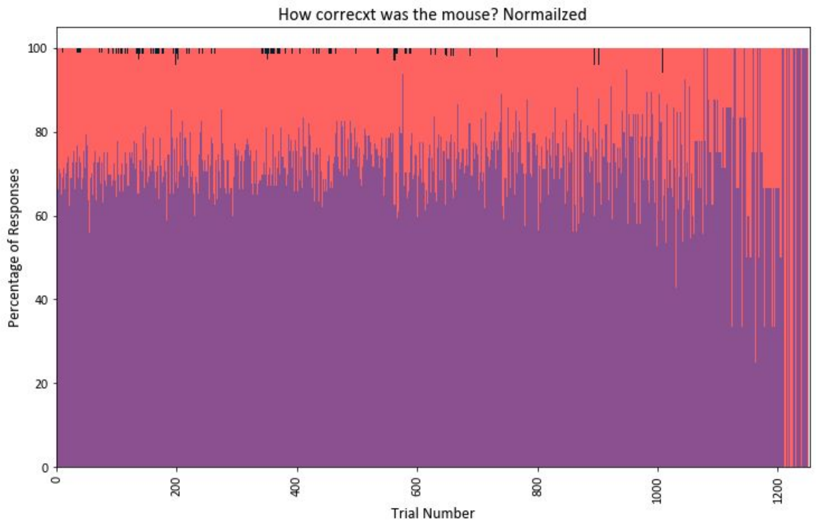
Step 2: Inspect the Data

training task	52
full task	21
training task + ephys	4

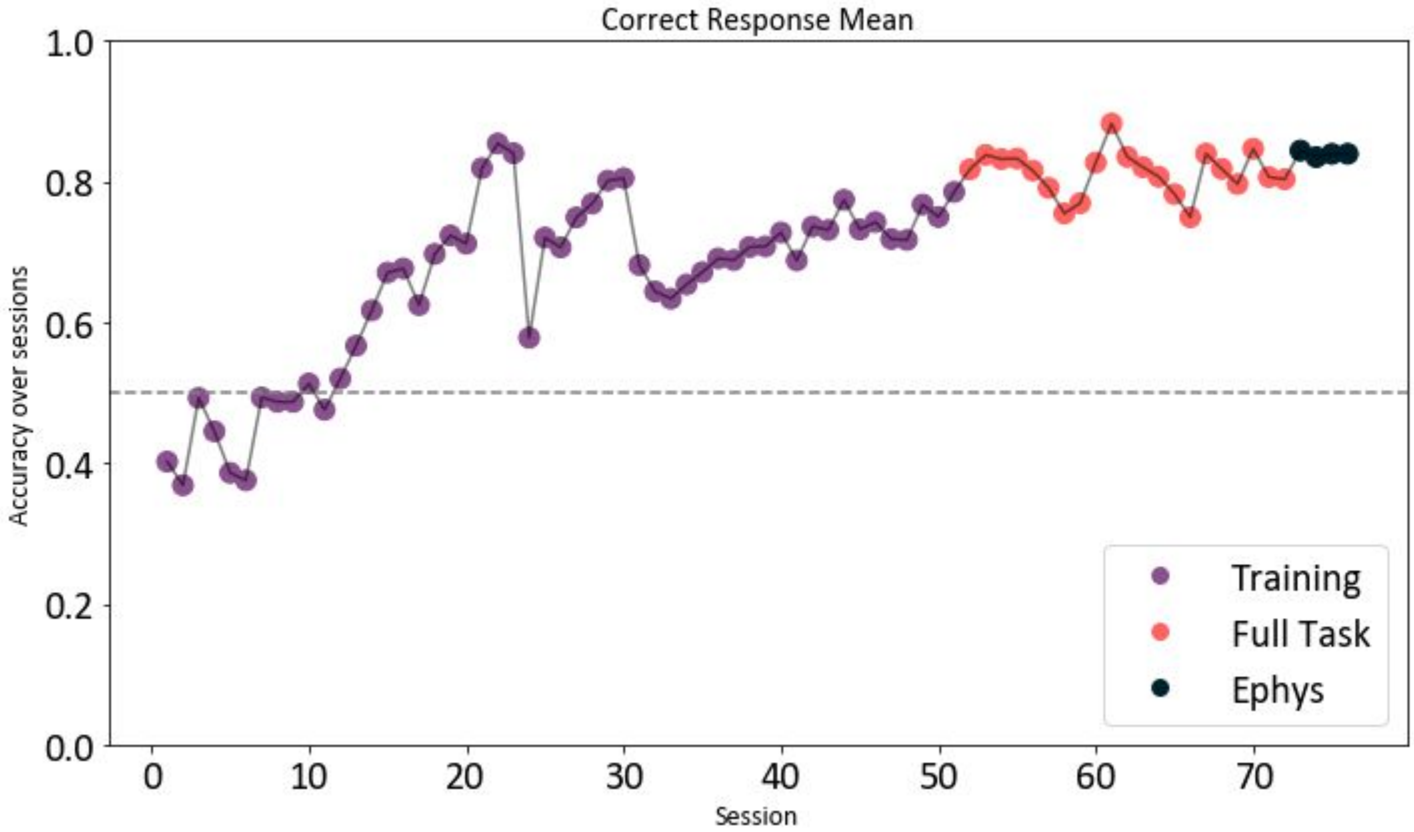


Step 2: Inspect the data

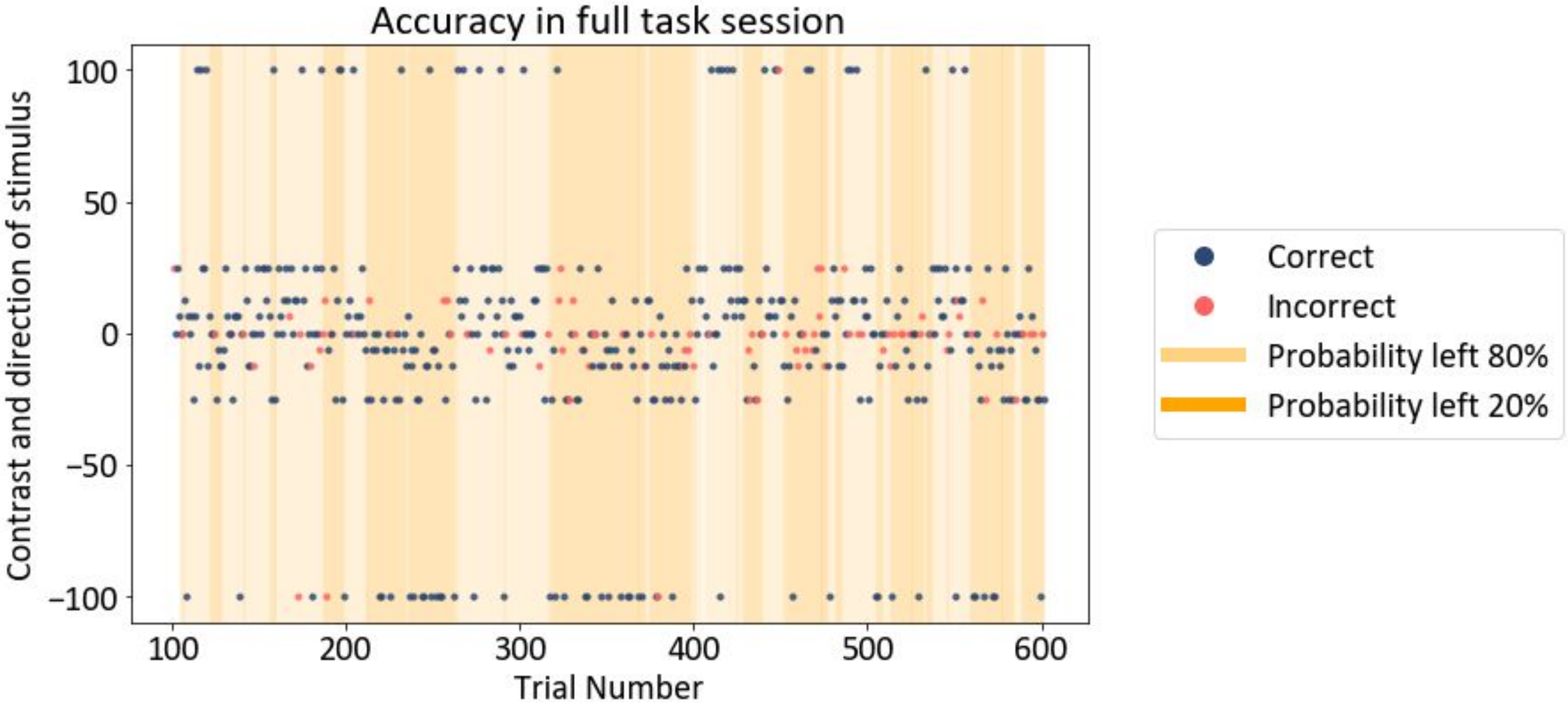


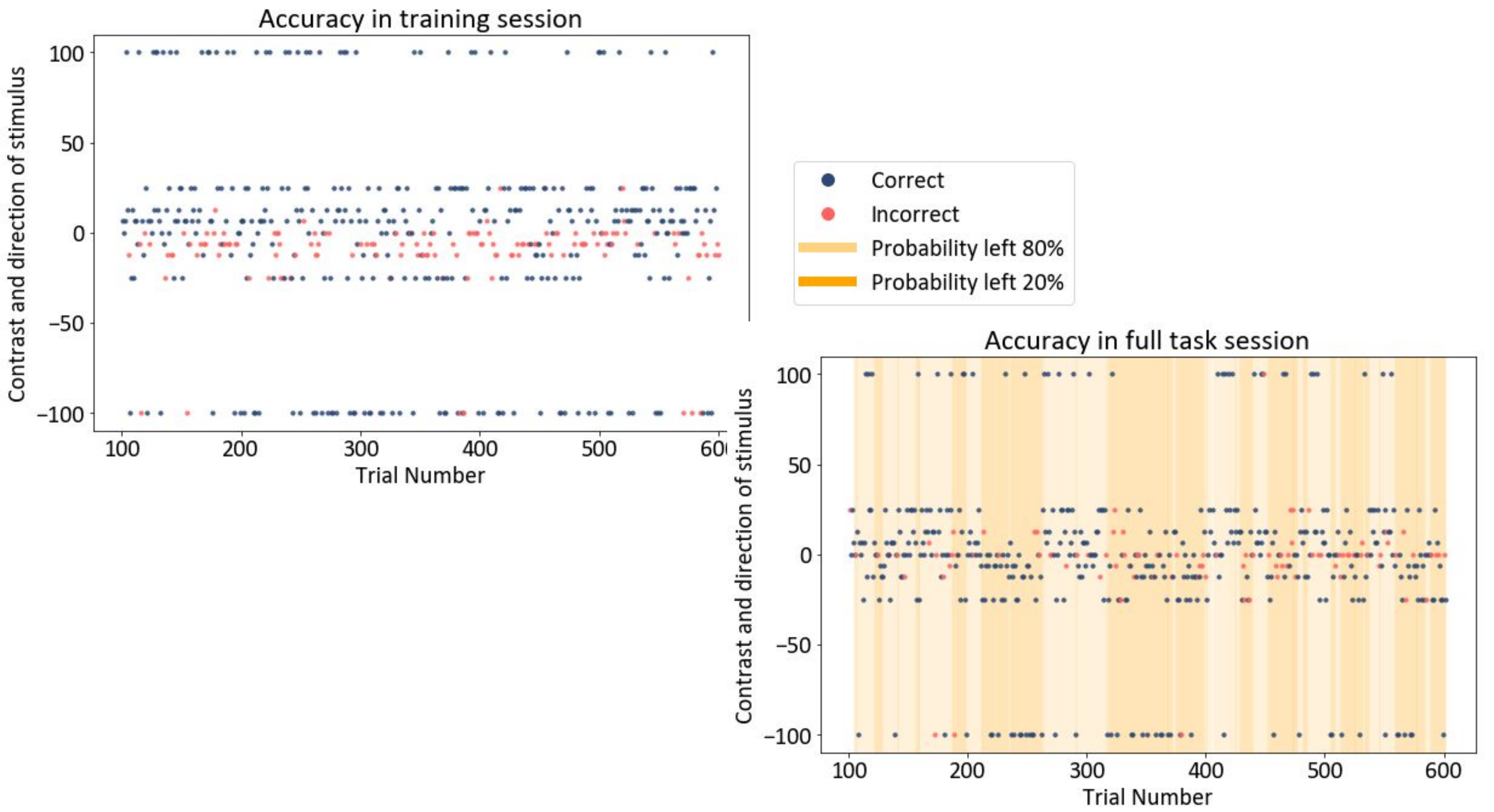


Is the mouse learning?



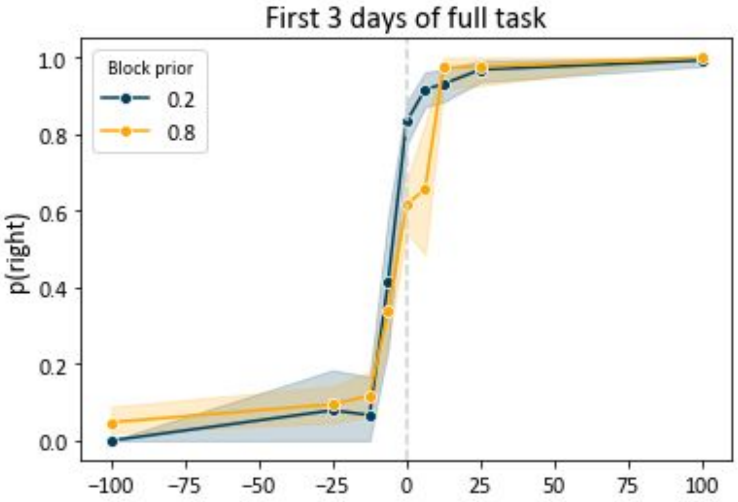
Can the mouse do stats?

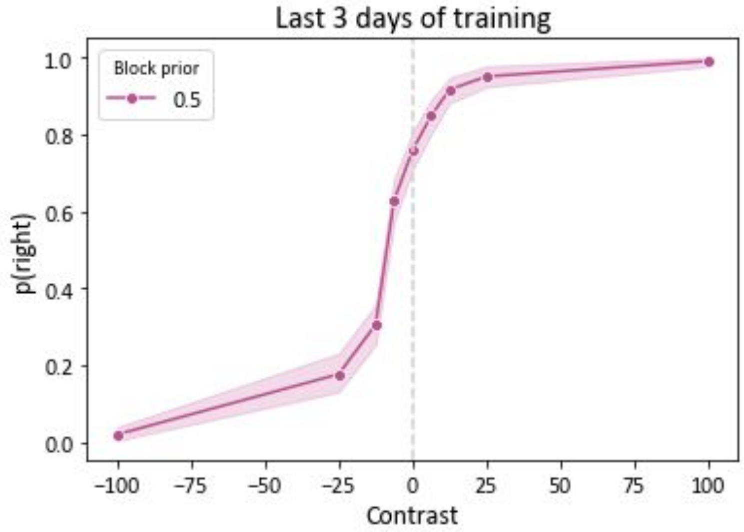


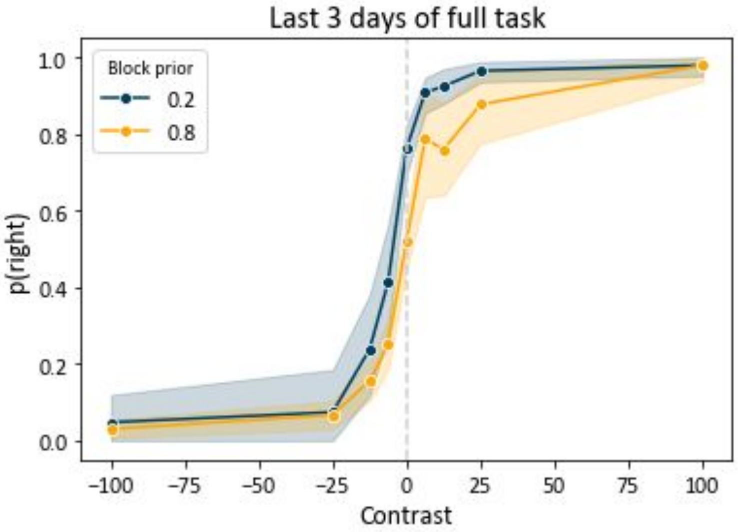


Accuracy in training session
Contrast and direction of stimulus
100
50
0
−50
−100
100
200
300
400
500
600
Trial Number
Correct
Incorrect
Probability left 80%
Probability left 20%
Accuracy in full task session
Contrast and direction of stimulus
100
50
0
−50
−100
100
200
300
400
500
600
Trial Number

Conclusion:
The mouse can do stats

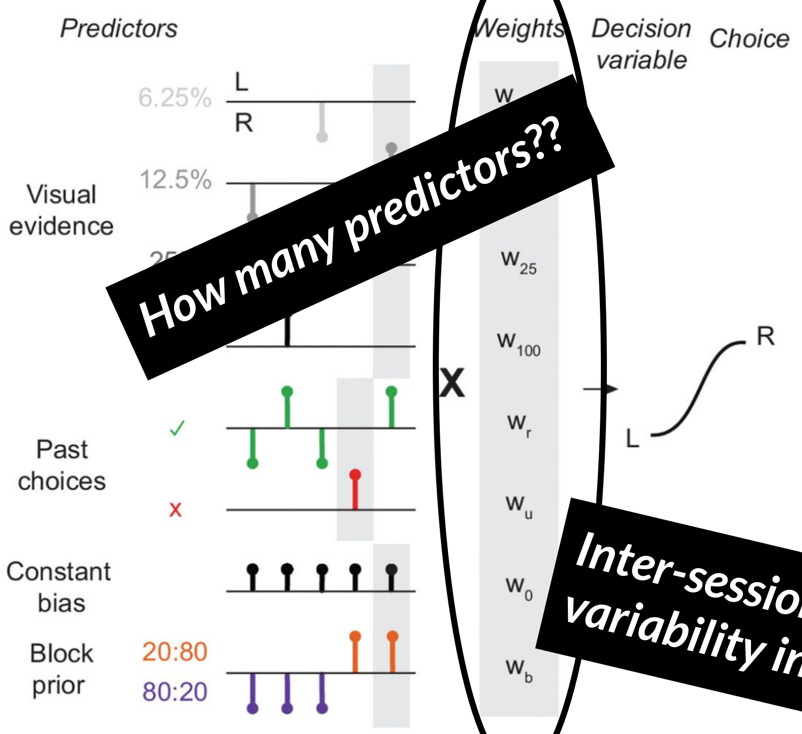






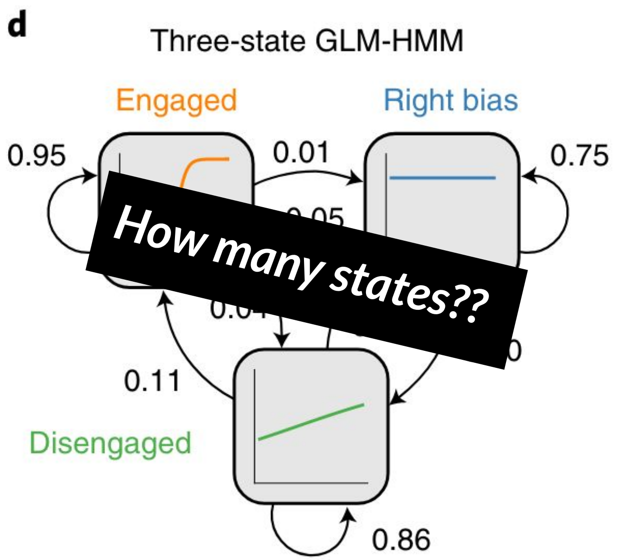
Step 3: Implementing the GLM-HMM

GLM:



How many predictors??

HMM:

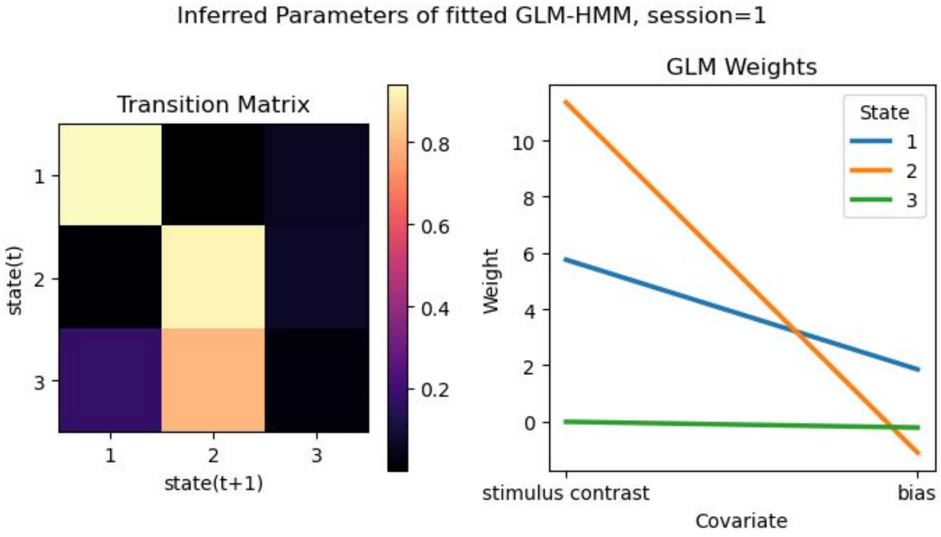


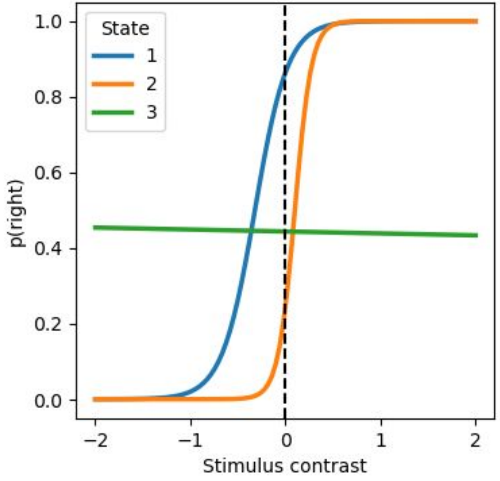
How many states??

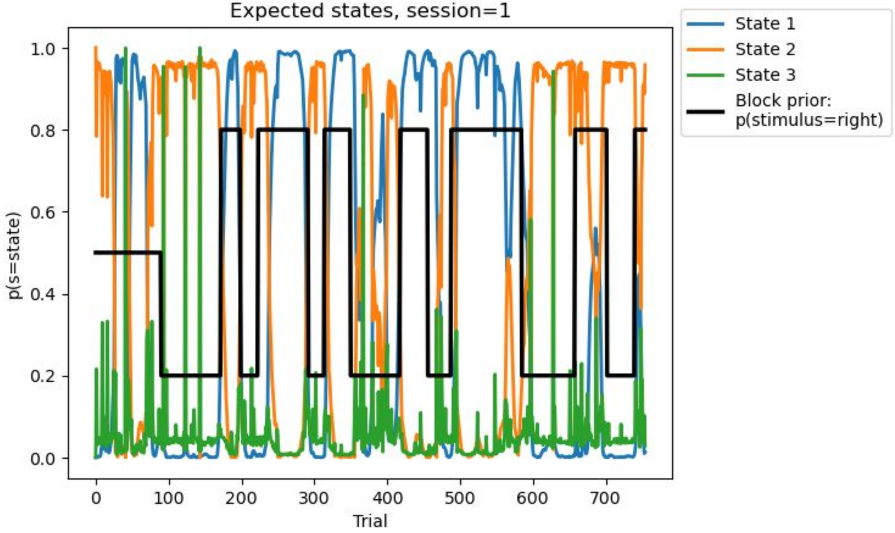
Inter-session or inter-block variability in strategies??

$$1|\mathbf{x}_t, z_t = k) = \frac{1}{1 + e^{-\mathbf{x}_t \cdot \mathbf{w}_k}}$$

Step 4: Fitting the GLM-HMM (all data)

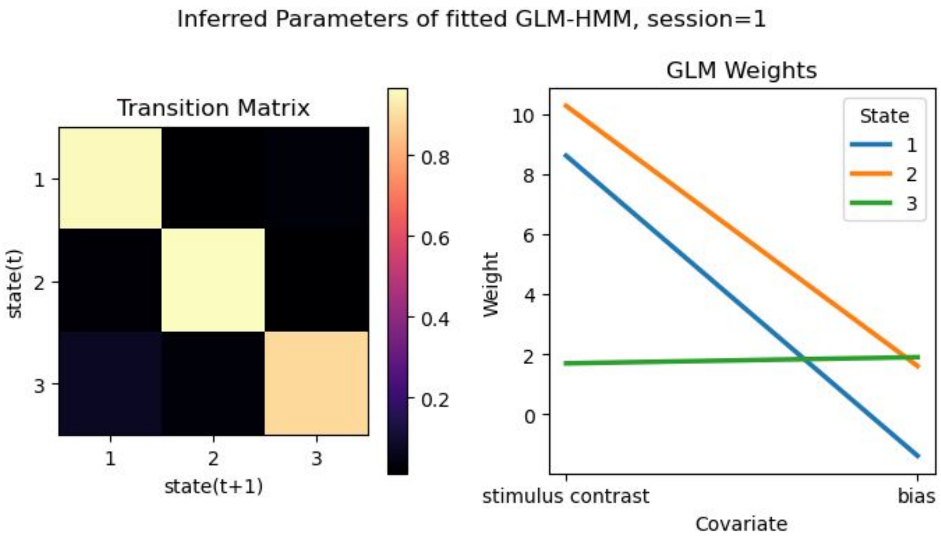


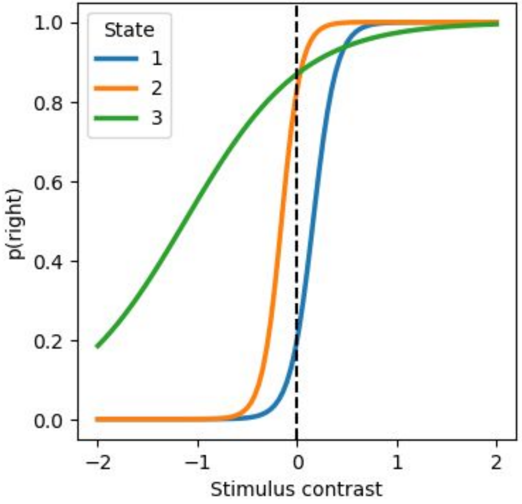


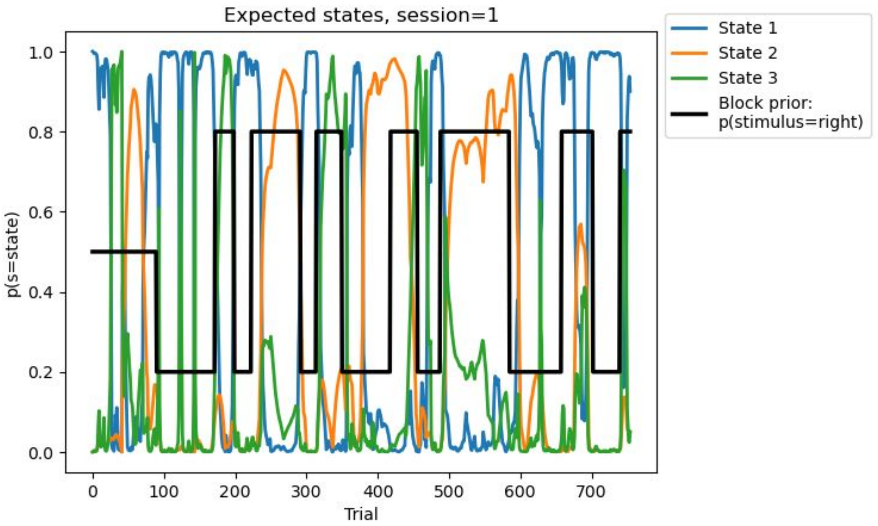


State 1 → engaged w/ right-bias?
State 2 → engaged w/ left-bias?
State 3 → disengaged?

Step 4: Fitting the GLM-HMM (session 1)

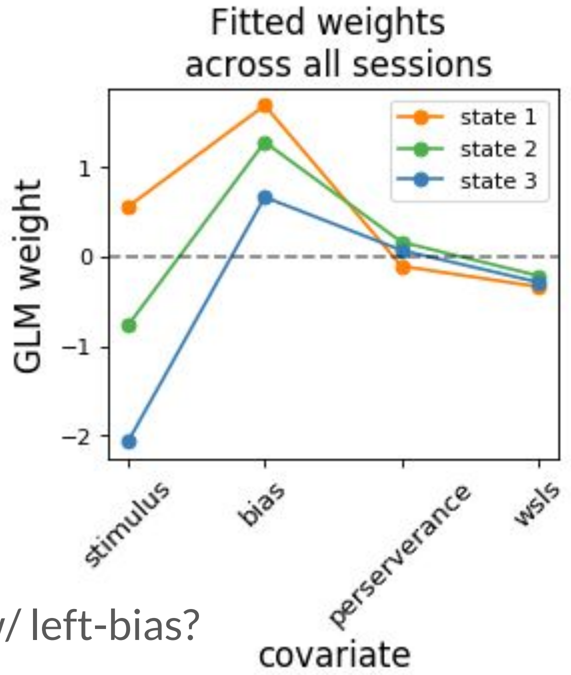


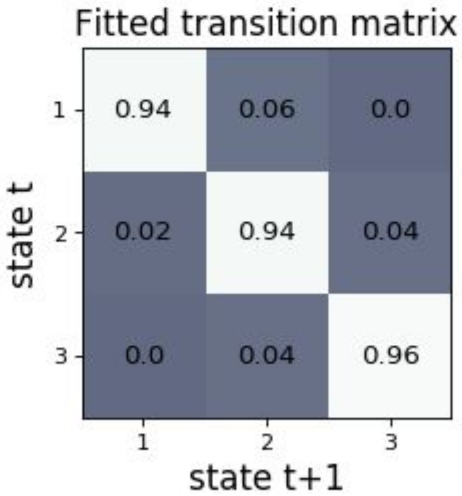




State 1 → engaged w/ left-bias?
State 2 → engaged w/ right-bias?
State 3 → disengaged ***w/ right-bias?***

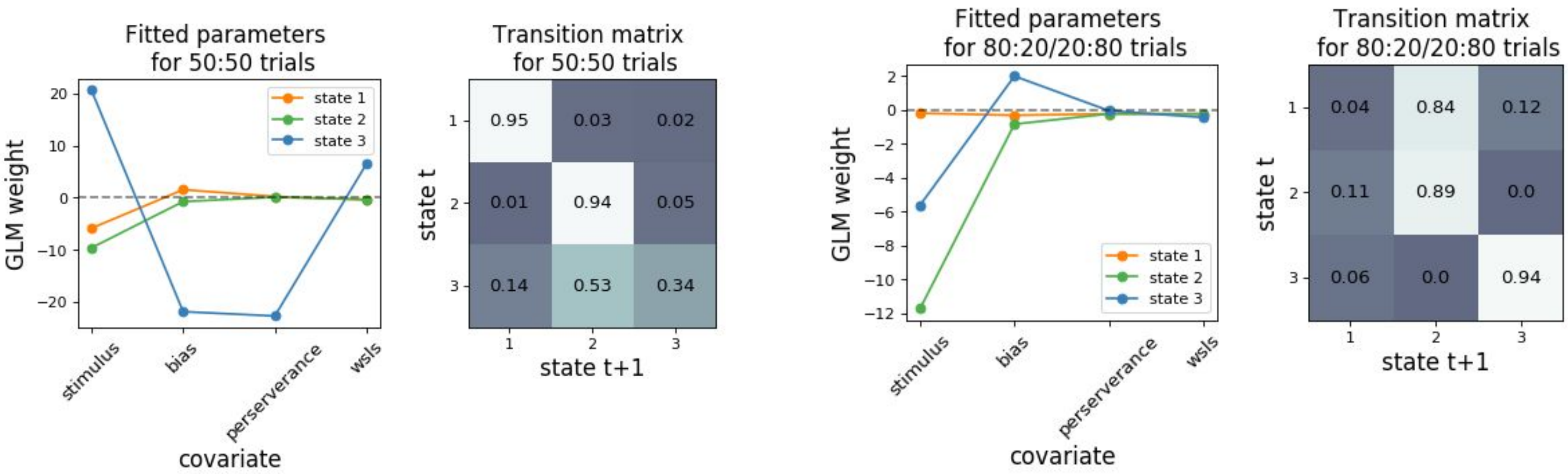
Replicating design matrix in Ashwood et al. (2020)



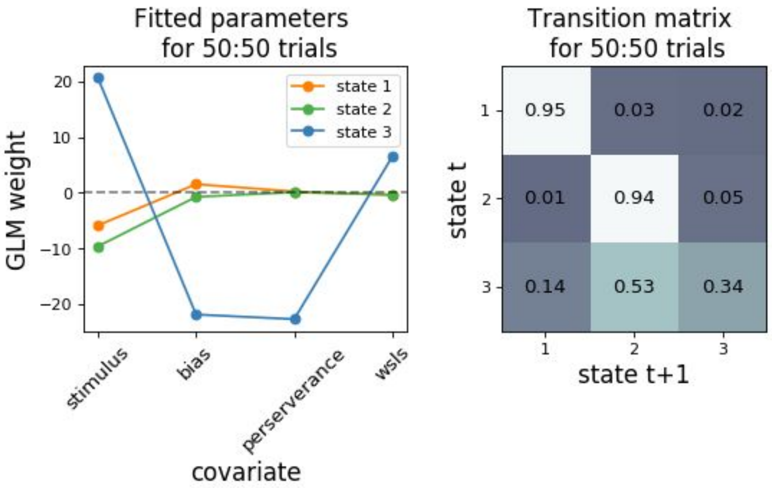


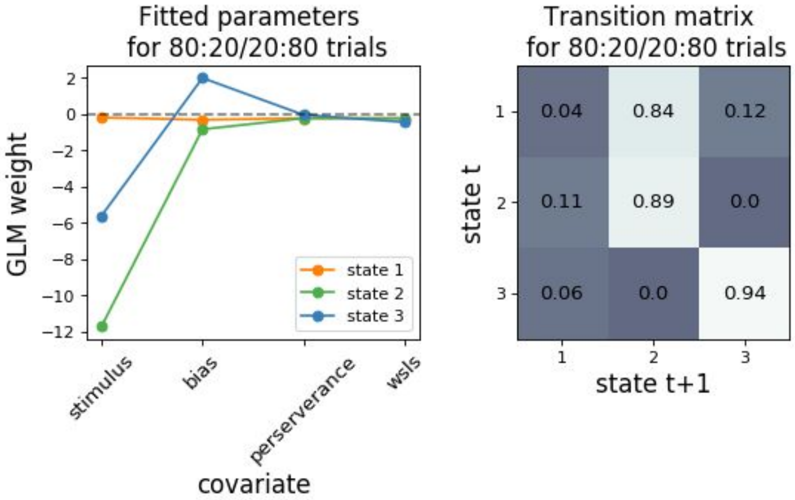
State 1 → engaged w/ left-bias?
State 2 → ?
State 3 → ?

Split data by block prior

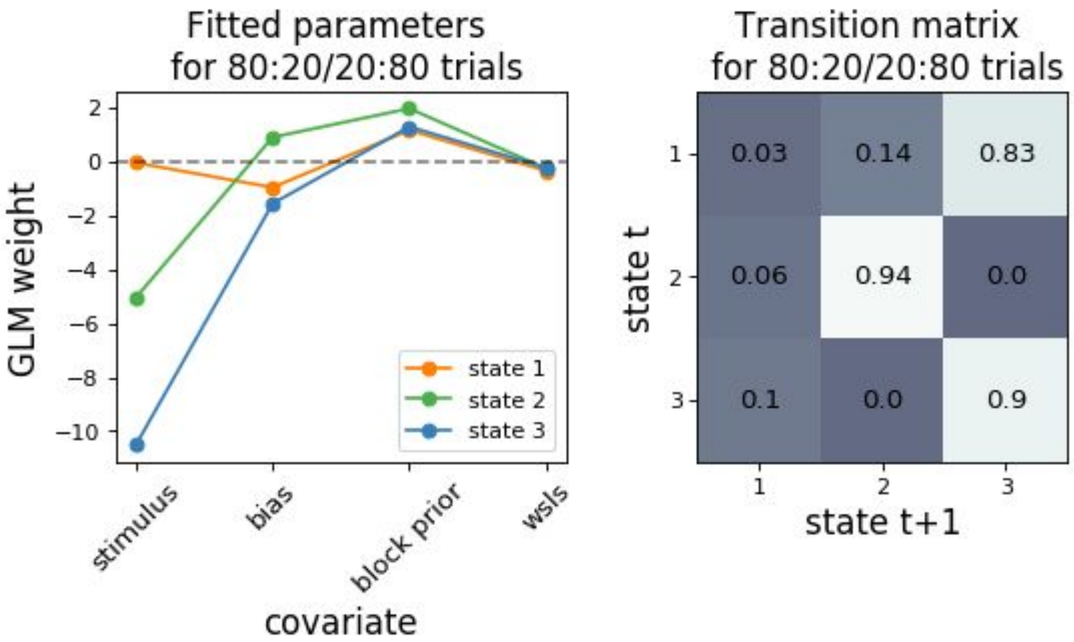


Split data by block prior





Adding block prior as a predictor for the GLM



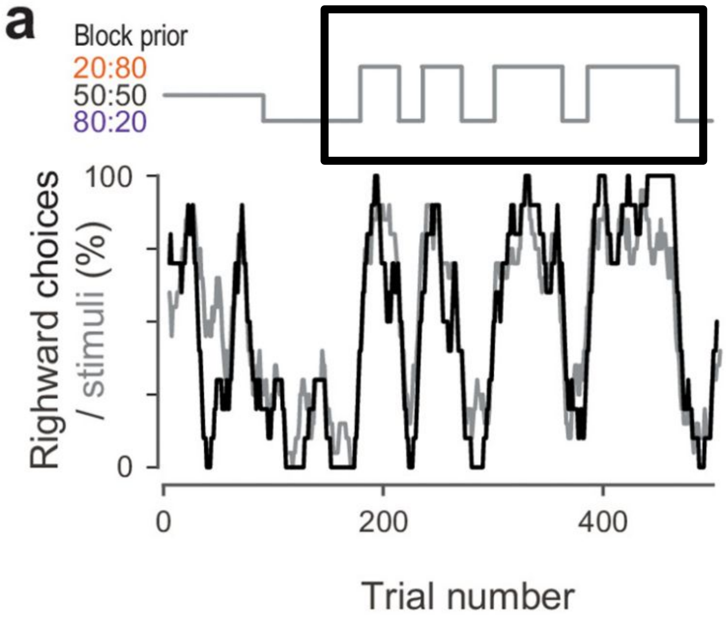
Remaining questions

How many states should we initialize in the first place?

How to interpret what each state represent?

How to design a meaningful design matrix for GLM?

Flexible Switching of Strategies

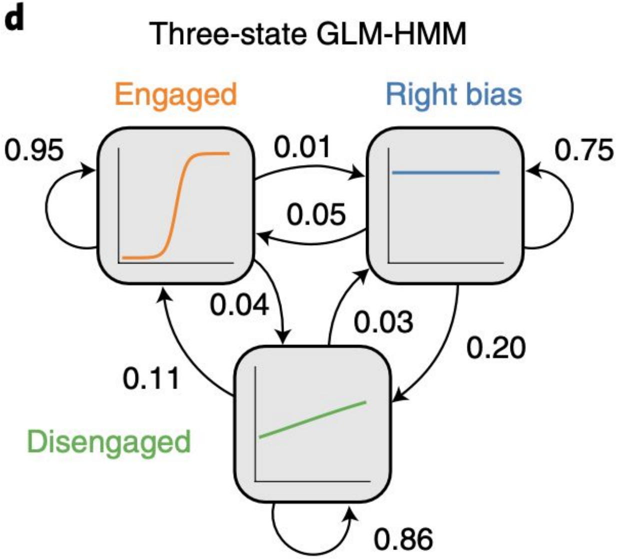


How do mice detect contingency reversals?

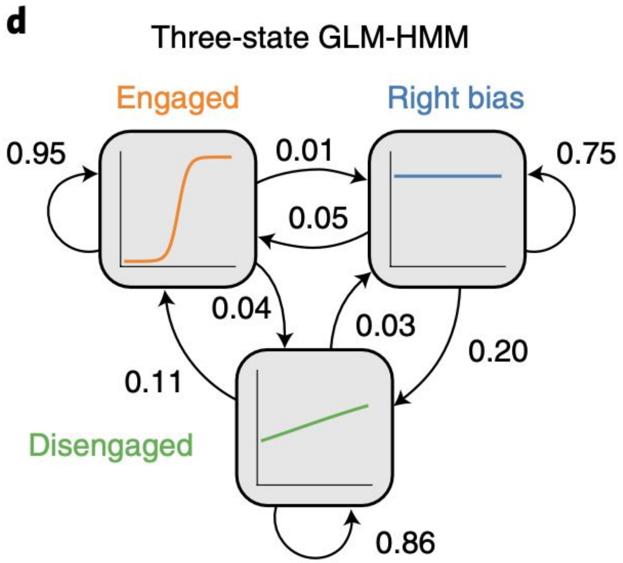
- Stable vs volatile switching?
- Gradually evolving beliefs vs discrete strategies?
- Disengagement vs exploration-exploitation?

GLM-HMM

- HMM governs the distribribution over latent states
- GLMS specifying the decision making strategy employed in each state



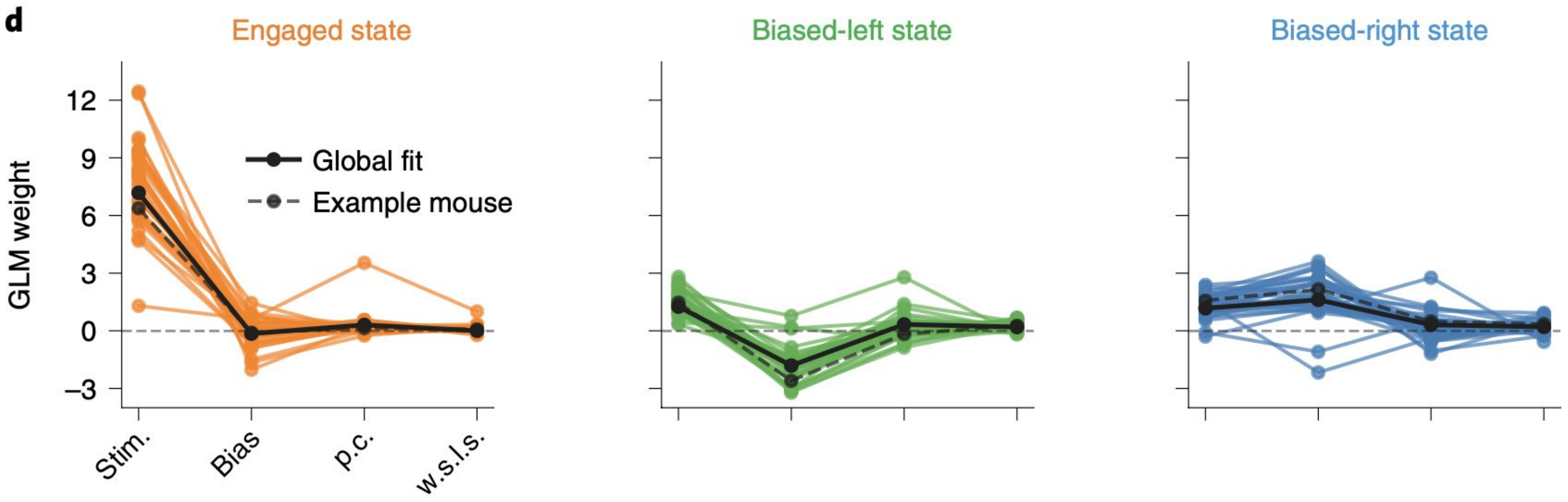
Question 1 & 2 – Stable vs volatile switching; Gradually evolving beliefs vs discrete strategies?



$$V_{t+1} = V_t + \Delta V_t$$

$$\Delta V_t = \alpha\beta(\lambda - V_t)$$

Question 3 –
Disengagement vs exploration-exploitation?



To understand how mice detect reversal…

Analyze data from 80:20 stable and 80:20 volatile trials

	Stable	Volatile
High contrast	Both sensory and statistical information	Sensory information
Low contrast	statistical information	statistical information