

Methods of computational model-based analysis in reinforcement learning and decision making



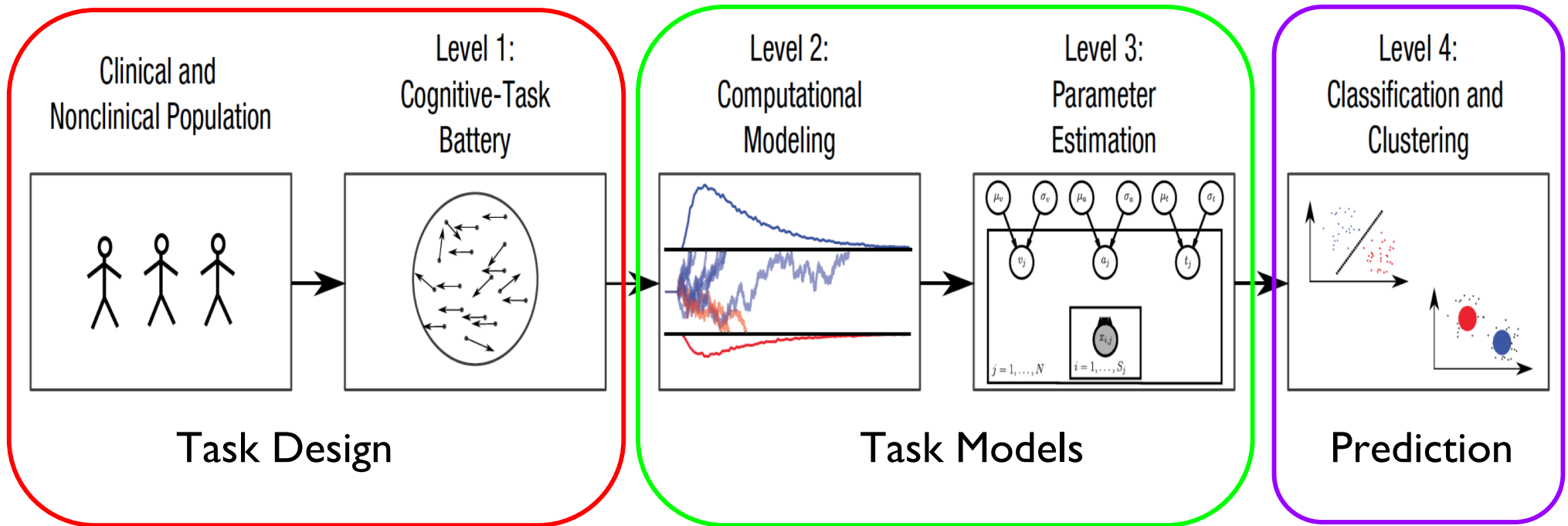
Michael J. Frank
Laboratory for Neural Computation and Cognition
Brown University



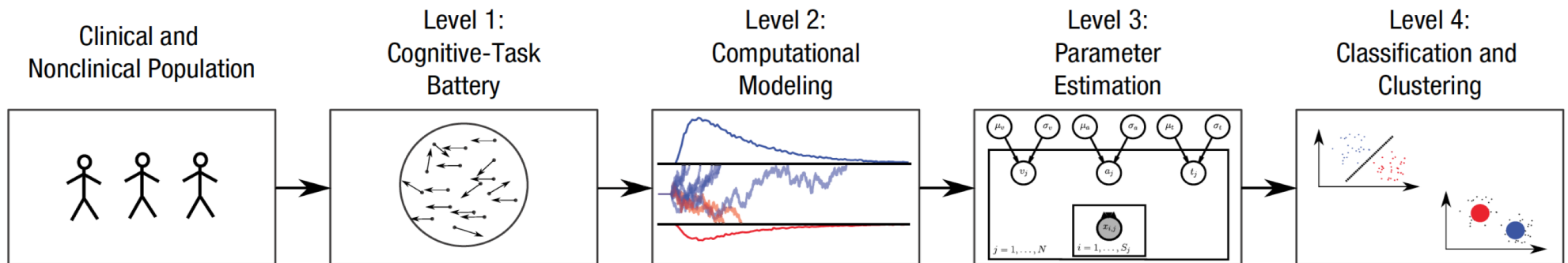
CARNEY INSTITUTE
FOR BRAIN SCIENCE

BROWN UNIVERSITY

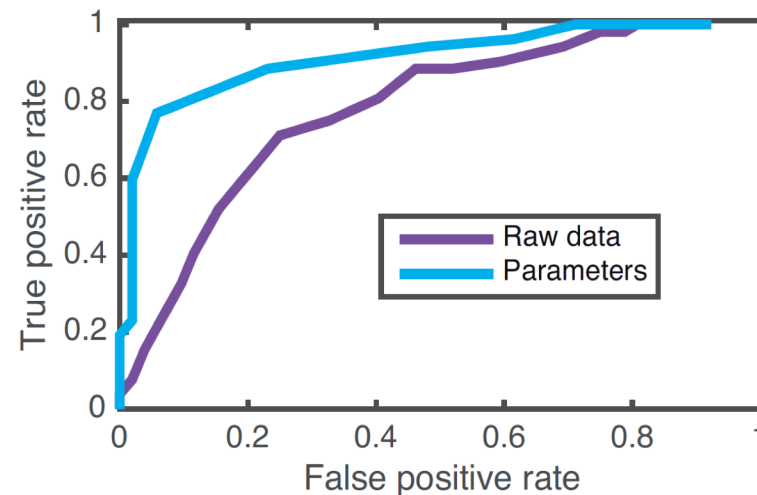
Computation as a link between brain, mind and pathology



Computation as a link between brain, mind and pathology



- Presymptomatic Huntington's
- Suicide attempts
- Impulsivity in Parkinson's
- Negative Sx in schizophrenia
- Depression



Models are (can be) great, but don't use them blindly!

Model building The first step is to build a series of models. Each contains an internal process by which different choice options are valued, and a link function which describes how preferences turn into observed decisions. At least two models should be built: a model M0 of 'no interest' that performs the task, but without involving the process of interest, and a model M1 that does contain the process of interest.

Validation on surrogate data

1. **Data generation:** Run each model on the experiment from which data will be examined. Do the generated data look reasonable?
2. **Surrogate model fitting:** Fit each model to the data generated from it. Are the true parameters readily recovered? Are some parameters not identifiable?
3. **Surrogate model comparison:** Does the model comparison procedure correctly identify the data generated by each model?

Real data analysis

1. **Real model fitting:** Fit each model to the real data.
2. **Real model validation:** Run each model with the fitted parameters on the exact experimental instance presented to that particular subject. Are the key features of the real data captured reasonably?
3. **Real model comparison:** choose the least complex model that best accounts for the data.
4. **Parameter examination:** only at this point should the parameters of the model be examined, and only the parameters of the most parsimonious model should be ascribed meaning.

Overview

- Model fitting
 - Maximum likelihood
 - grid search, gradient / simplex
 - perils and tricks
- Bayesian approach
 - likelihoods, posteriors etc
 - hierarchical models
- Model comparison
- Model Validation
- Linking levels of analysis

Example simple RL model: Q learning

Error in predicted reward:

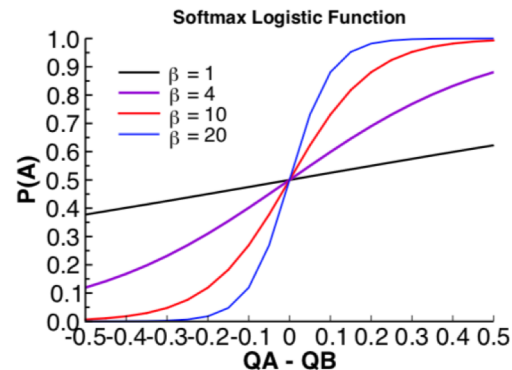
$$\delta_t = \left(r_t + \gamma \max_a Q_t(s_{t+1}, a) \right) - Q_t(s, a)$$

Update value estimate:

$$Q_t(s, a) \leftarrow Q_t(s, a) + \alpha \delta(t)$$

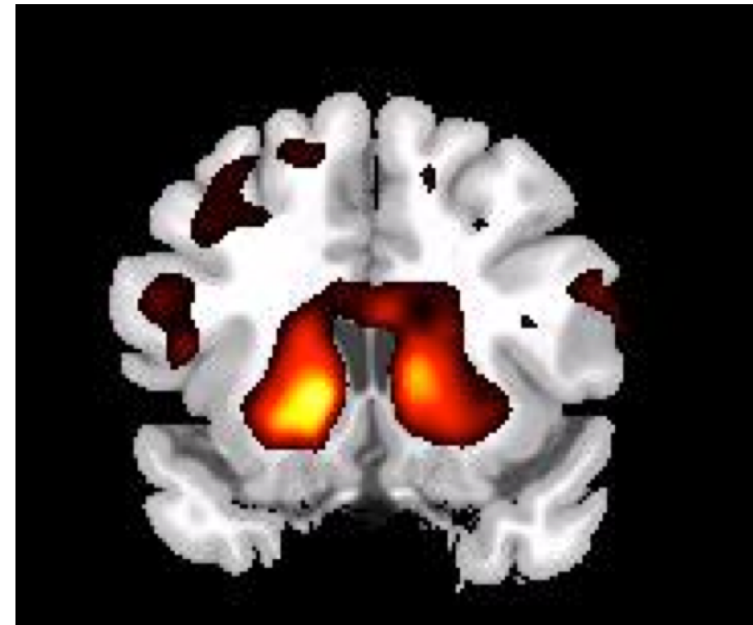
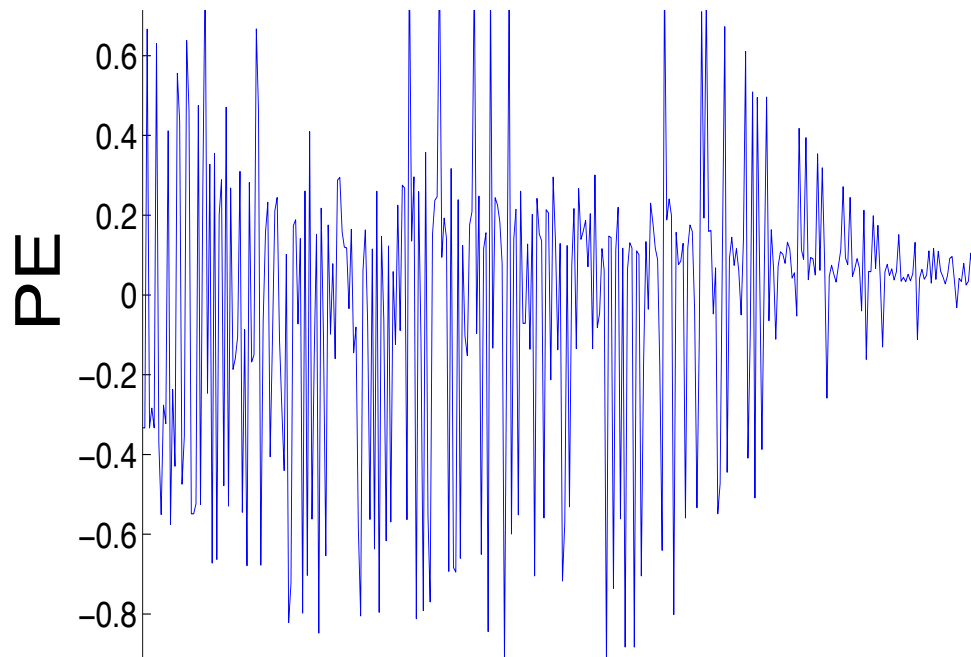
Select among Q values (“softmax”):

$$P_t(a) = \frac{e^{\beta Q_t(s,a)}}{\sum_{i=1}^n e^{\beta Q_t(s,i)}}$$



γ = discount, α = learning rate, β = “temperature” / exploration parameter

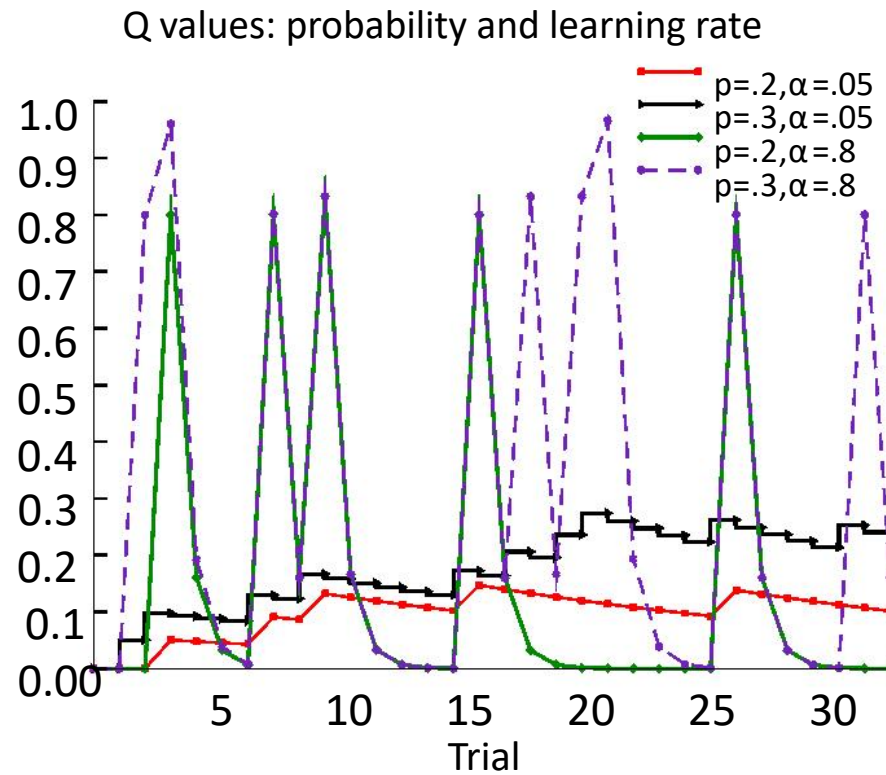
Reward prediction error and human functional imaging



- Parametric contrast (RPE convolved with HRF)
- Modulated by DA drugs and in Parkinsons; predictive of learning ability

O'Doherty et al, 2004; McClure et al, 2003, Caplin et al, 2010;
Badre & Frank, 2012; D'Ardenne et al 08; Niv et al 2012

Parameters matter: learning rate



High learning rates: capture trial-to-trial adaptation

Low learning rates: slow, integrative probabilities

If the brain implements RL:
How can we infer the hidden variables (Q values, parameters, etc)?

- Assume model is correct, but has free parameters θ (for Q learning, $\theta = \{\alpha, \beta, \gamma\}$).
- How to find θ that gives best characterization of behavior?
- Given θ :
 - does this model fit better than other competing models?
 - If so, are there correlates of model variables (Q values) in brain (e.g. striatal activity)?
 - see how manipulation of biology by drugs, genes, lesions.. affects
 - * parameters (learning rates, discount etc)
 - * model (e.g., from actor critic to Q learning).

Model Fitting: Maximum Likelihood

For model M , find θ that maximizes the likelihood of choices y given stimuli, rewards x

e.g., for choices between actions A and B, $y = [A, B, A, A, B, A, A, A, A, B, ..]$

$$\hat{\theta}_{ml} = \arg \max_{\theta} p(y|x, \theta)$$

Across all n trials t :

$$p(y|x, \theta) = \prod_{t=1:n} p(y_t|x_t, \theta)$$

Model Fitting: Maximum Likelihood

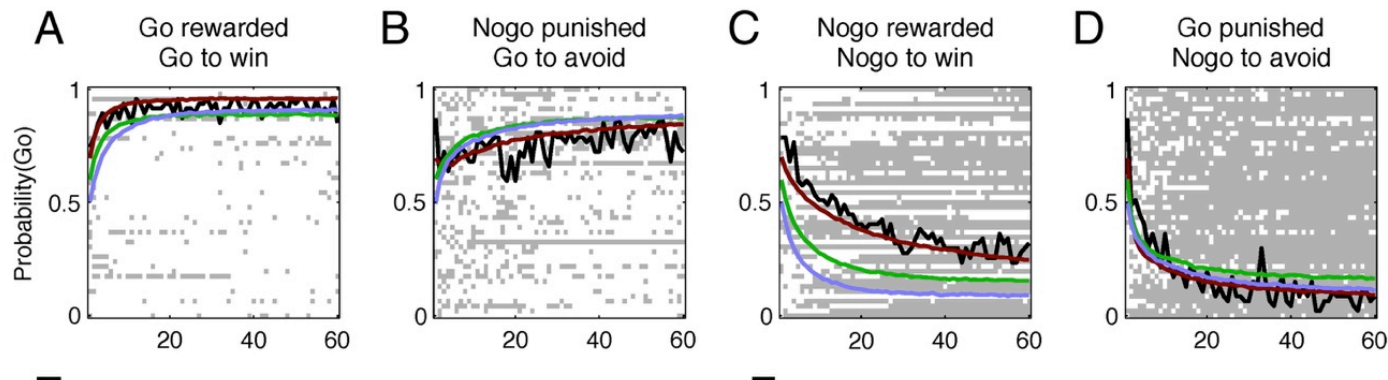
For model M , find θ that maximizes the likelihood of choices y given stimuli, rewards x

e.g., for choices between actions A and B, $y = [A, B, A, A, B, A, A, A, A, B, ..]$

$$\hat{\theta}_{ml} = \arg \max_{\theta} p(y|x, \theta)$$

Across all n trials t :

$$p(y|x, \theta) = \prod_{t=1:n} p(y_t|x_t, \theta)$$



Model Fitting: Maximum Likelihood

For model M , find θ that maximizes the likelihood of choices y given stimuli, rewards x

e.g., for choices between actions A and B, $y = [A, B, A, A, B, A, A, A, A, B, ..]$

$$\hat{\theta}_{ml} = \arg \max_{\theta} p(y|x, \theta)$$

Across all n trials t :

$$p(y|x, \theta) = \prod_{t=1:n} p(y_t|x_t, \theta)$$

$$p(y_t|x_t, \hat{\theta}) = [0.5, 0.4, 0.7, 0.8, 0.4, 0.8, 0.85, 0.9, 0.1, ...]$$

In practice use log likelihood:

$$L = \log(p(y|x, \theta)) = \log(\prod_t p(y_t|x_t, \theta)) = \sum_t \log(p(y_t|x_t, \theta)) = \log(0.5) + \log(0.4) ..$$

Model Fitting: Maximum Likelihood

For model M , find θ that maximizes the likelihood of choices y given stimuli, rewards x

e.g., for choices between actions A and B, $y = [A, B, A, A, B, A, A, A, A, B, ..]$

$$\hat{\theta}_{ml} = \arg \max_{\theta} p(y|x, \theta)$$

Across all n trials t :

$$p(y|x, \theta) = \prod_{t=1:n} p(y_t|x_t, \theta)$$

$$p(y_t|x_t, \hat{\theta}) = [0.5, 0.4, 0.7, 0.8, 0.4, 0.8, 0.85, 0.9, 0.1, ...]$$

In practice use log likelihood:

$$L = \log(p(y|x, \theta)) = \log(\prod_t p(y_t|x_t, \theta)) = \sum_t \log(p(y_t|x_t, \theta)) = \log(0.5) + \log(0.4) ..$$

→ compare to model predicting chance (here $p=0.5$) for all trials: $R = \log(0.5^n) = n \log(0.5)$

→ pseudo- $R^2 = \frac{L-R}{R}$ (Camerer & Ho, 1999)

best fitting model typically in range 0.1-0.7 for RL (depends on performance, difficulty..)

How to find $\hat{\theta}_{ml}$?

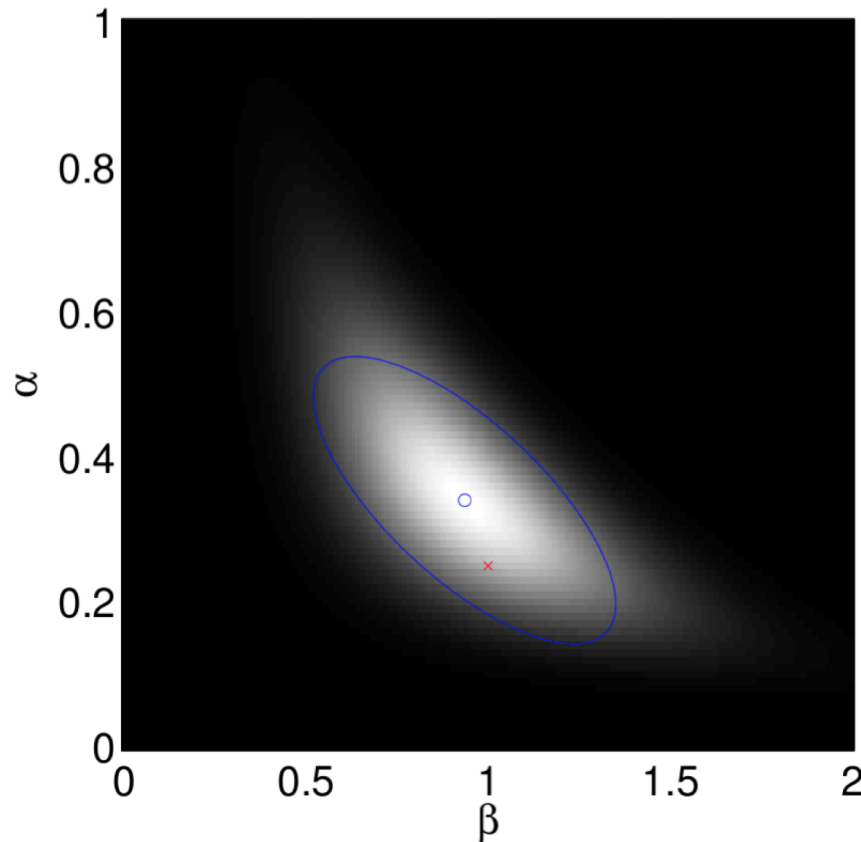
- Grid search
 - vary each parameter θ_i across a wide range with fixed stepsize (e.g. .01). Can plot full likelihood surface for all param combinations
 - time and/or memory intensive
 - for ≤ 3 or 4 params, can use matrix operations to test all param combinations simultaneously, but combinatorial explosion for more params
- nonlinear optimization functions (e.g., gradient descent, “Simplex”)
 - matlab functions *fmincon*, *fminsearch*, *rmsearch* etc
 - find single best combination with arbitrary precision
 - gives estimation of Hessian $\frac{\partial^2}{\partial^2\theta} L$ at $\hat{\theta}_{mle}$: how peaked is the likelihood function? How stable are parameter estimates?

Problems with Maximum Likelihood?

- fast search algorithms, but may get local optima
- → use various starting points for θ (*rmsearch*)
- But what if multiple maxima in likelihood surface that are not that different? That is, how to interpret “MLE” if $\alpha, \beta = 0.2, 1$ gives only slightly better fit than 1,0.2?
- multicollinearity and **identifiability**
- confidence in param estimates depends on model fit, and relatedly, overall performance on the task (for poorer learners, models will fit worse, and params are less identifiable).

Generative model to recover parameters

Are parameters separately identifiable? For one combination of “true” α , β :

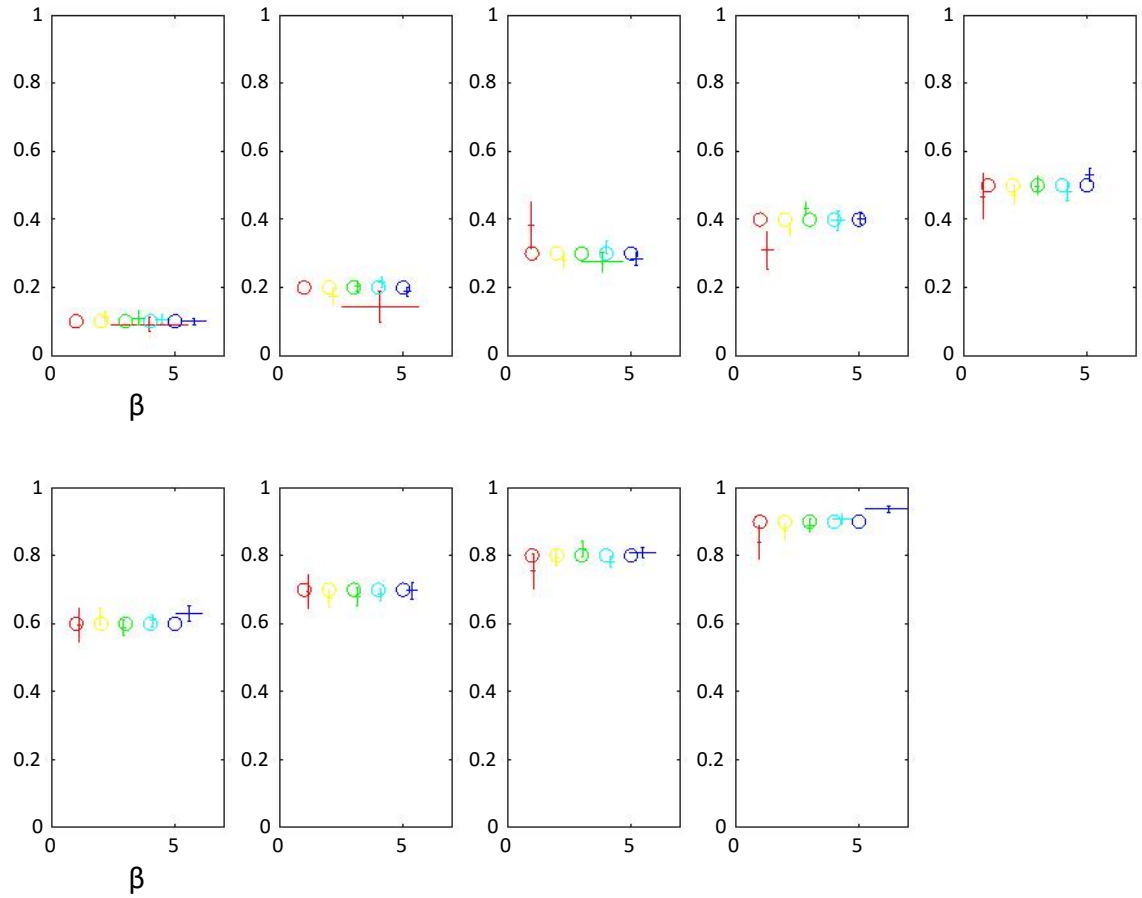


Daw, 2010

- α determines rate of learning, β determines gain/discrimination/exploration.. .
- Co-linear, but separately identifiable - depending on task!
e.g., with increasingly deterministic outcomes, β determines asymptotic accuracy

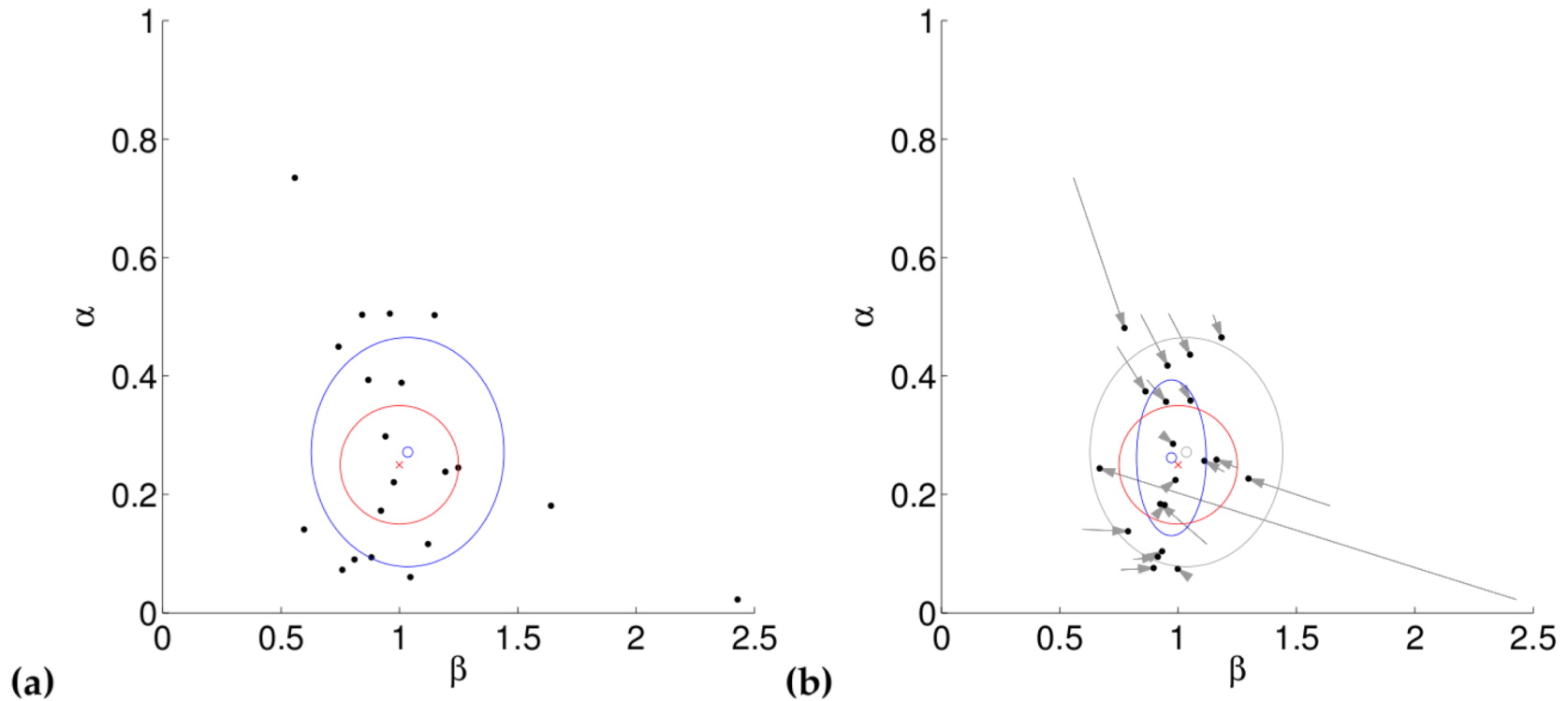
Parameter recovery simulations

For all combinations of $\alpha \in \{0.1:0.1:1\}$, $\beta \in \{1:1:5\}$



with Anne Collins

RL model generated and recovered; hierarchical shrinkage



red: generative distribution; **black:** recovered; **blue:** population stats

- summary statistics across a group gives good estimate of population stats (Holmes & Friston, 1998; Daw 2010).
- variance is over-estimated. Need sufficient N.
- group stats should constrain estimate of individuals. How to do this in principled way?

Example: Expectancy Valence model Iowa Gambling Task

$$v(t) = (1 - w)W(t) + wL(t)$$

w = attentional weight to losses vs wins

$$Ev_k(t + 1) = Ev_k(t) + a(v(t) - Ev_k(t))$$

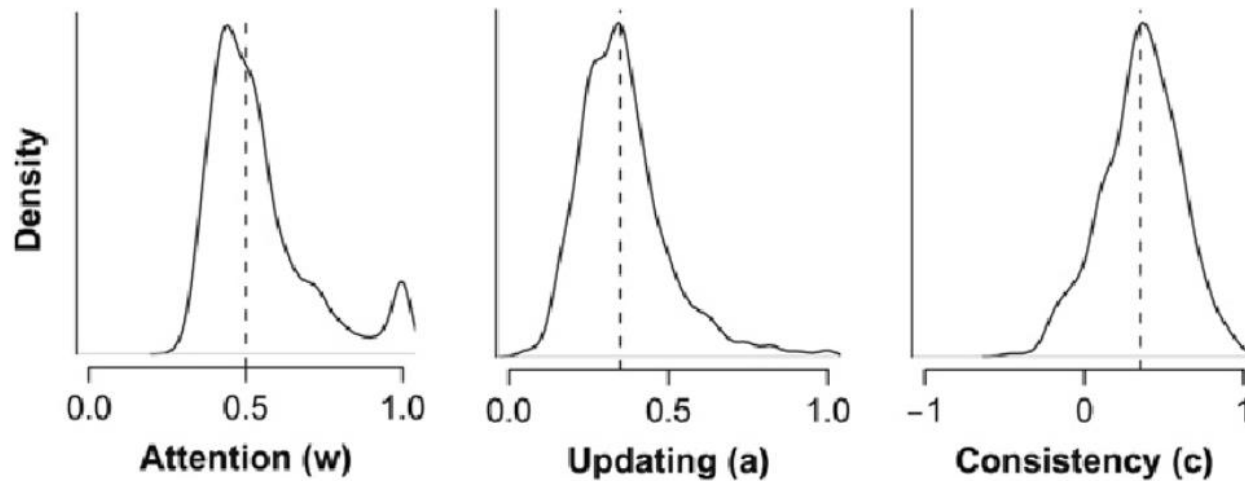
a = recency / updating / learning rate

$$P(S_k(t + 1)) = \frac{\exp(\theta(t)Ev_k)}{\sum_j \exp(\theta(t)Ev_j)}$$

$\theta(t)$ = softmax gain = $(t/10)^c$

c = response consistency

Parameter recovery: Maximum likelihood estimation

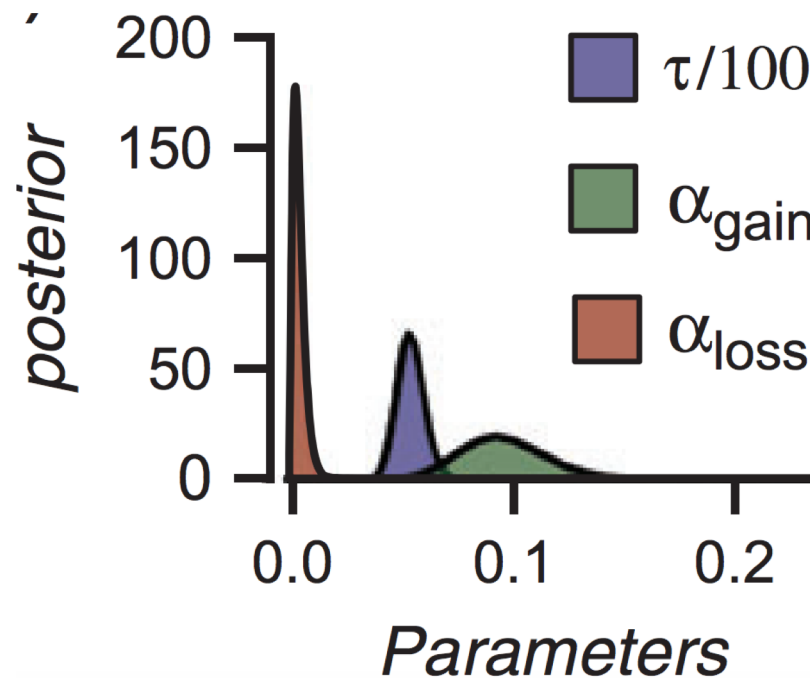


- Generative model:
 - 1000 simulated runs of the same participant (ie same params), 150 trials each
- means of MLE parameters are very close to true values (dashed line)
- some individual runs can be off, and param at bounds
- **in practice, with MLE can't get these distributions for a single subject** (unless we tested them many times and assumed no meta-learning!)

Bayesian approach

Instead of one ML solution, quantify *uncertainty* in probability distributions over parameters:
How much more likely is ML compared to other potential parameter values for the same data?

This is what we want for real data:



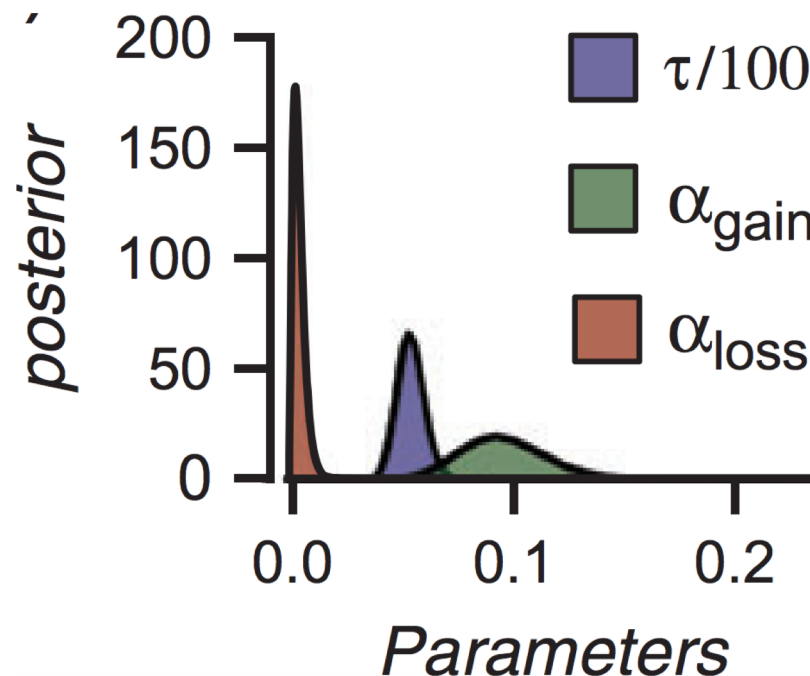
Bayesian approach

Instead of one ML solution, quantify *uncertainty* in probability distributions over parameters.

Given model M parametrized by θ

$$P(\theta|D) = \frac{P(D|\theta)P(\theta)}{\int P(D|\theta)P(\theta)d\theta}$$

all P 's above are full distributions $\forall\theta$, not point estimates



Bayesian approach

Instead of one ML solution, quantify *uncertainty* in probability distributions over parameters.

Given model M parametrized by θ

$$P(\theta|D) = \frac{P(D|\theta)P(\theta)}{\int P(D|\theta)P(\theta)d\theta}$$

all P 's above are full distributions $\forall\theta$, not point estimates

Maximum a posteriori (MAP) estimate:

$$\hat{\theta}_{map} = \arg \max_{\theta} P(y|x, \theta)P(\theta)$$

Takes into account prior $P(\theta)$.

- The prior $P(\theta)$ for each subject can be uniform or constrained by prior knowledge (e.g. from literature; *empirical priors*) and/or from the group (stay tuned).

But exact bayesian inference is hard

- approximate methods
- Markov Chain Monte Carlo (MCMC) sampling
- Variational Inference
- Expectation Maximization

...

Given model M parametrized by θ

$$P(\theta|D) = \frac{P(D|\theta)P(\theta)}{\int P(D|\theta)P(\theta)d\theta}$$

all P's above are full distributions $\forall\theta$, not point estimates

But exact bayesian inference is hard

- approximate methods
- **Markov Chain Monte Carlo (MCMC) sampling**
- Variational Inference
- Expectation Maximization

...

Sampling



What is the average height of everyone in this room?

Method: measure all heights, add them up and divide by N

What is the average height f of people p in Canada?

Surveying works for large and notionally infinite populations.

Sampling



What is the average height of everyone in this room?

Method: measure all heights, add them up and divide by N

What is the average height f of people p in Canada?

$$E_{p \in \mathcal{C}}[f(p)] \equiv \frac{1}{|\mathcal{C}|} \sum_{p \in \mathcal{C}} f(p), \quad \text{“intractable”?}$$

$$\approx \frac{1}{S} \sum_{s=1}^S f(p^{(s)}), \quad \text{for random survey of } S \text{ people } \{p^{(s)}\} \in \mathcal{C}$$

Surveying works for large and notionally infinite populations.

Simple Monte Carlo

Statistical sampling can be applied to any expectation:

In general:

$$\int f(x)P(x) dx \approx \frac{1}{S} \sum_{s=1}^S f(x^{(s)}), \quad x^{(s)} \sim P(x)$$

Example: making predictions

$$\begin{aligned} p(x|\mathcal{D}) &= \int P(x|\theta, \mathcal{D})P(\theta|\mathcal{D}) d\theta \\ &\approx \frac{1}{S} \sum_{s=1}^S P(x|\theta^{(s)}, \mathcal{D}), \quad \theta^{(s)} \sim P(\theta|\mathcal{D}) \end{aligned}$$

Properties of Monte Carlo

Estimator: $\int f(x)P(x) dx \approx \hat{f} \equiv \frac{1}{S} \sum_{s=1}^S f(x^{(s)}), \quad x^{(s)} \sim P(x)$

Estimator is unbiased:

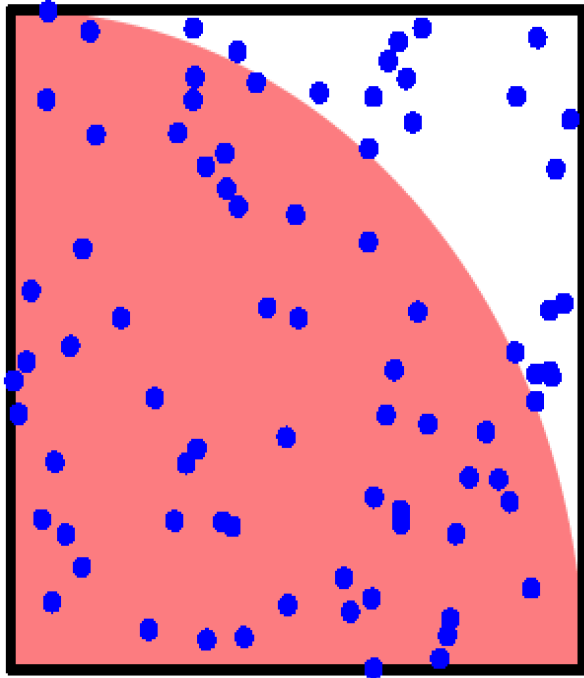
$$\mathbb{E}_{P(\{x^{(s)}\})} [\hat{f}] = \frac{1}{S} \sum_{s=1}^S \mathbb{E}_{P(x)} [f(x)] = \mathbb{E}_{P(x)} [f(x)]$$

Variance shrinks $\propto 1/S$:

$$\text{var}_{P(\{x^{(s)}\})} [\hat{f}] = \frac{1}{S^2} \sum_{s=1}^S \text{var}_{P(x)} [f(x)] = \text{var}_{P(x)} [f(x)] / S$$

“Error bars” shrink like \sqrt{S}

A dumb approximation of π



$$P(x, y) = \begin{cases} 1 & 0 < x < 1 \text{ and } 0 < y < 1 \\ 0 & \text{otherwise} \end{cases}$$

$$\pi = 4 \iint \mathbb{I}((x^2 + y^2) < 1) P(x, y) \, dx \, dy$$

```
octave:1> S=12; a=rand(S,2); 4*mean(sum(a.*a,2)<1)
ans = 3.3333
```

```
octave:2> S=1e7; a=rand(S,2); 4*mean(sum(a.*a,2)<1)
ans = 3.1418
```

Markov Chain Monte Carlo (MCMC)

Sampling isn't random, but is iterative – each point is (stochastically) determined by the previous one: “Markov Chain”

Markov Chain Monte Carlo (MCMC)

Sampling isn't random, but is iterative – each point is (stochastically) determined by the previous one: “Markov Chain”

Choose a **candidate** density $q(\boldsymbol{\theta}^* | \boldsymbol{\theta}^{(t-1)})$ that is a valid density function for every possible value of the conditioning variable $\boldsymbol{\theta}^{(t-1)}$, and satisfies

$$q(\boldsymbol{\theta}^* | \boldsymbol{\theta}^{(t-1)}) = q(\boldsymbol{\theta}^{(t-1)} | \boldsymbol{\theta}^*),$$

i.e., q is **symmetric** in its arguments.

MCMC: Metropolis Algorithm

Given a starting value $\theta^{(0)}$ at iteration $t = 0$, the algorithm proceeds as follows:

• **Metropolis Algorithm:** For $(t \in 1 : T)$, repeat:

1. Draw θ^* from $q(\cdot | \theta^{(t-1)})$

2. Compute the ratio

$$r = p(\theta^*) / p(\theta^{(t-1)})$$

• r •

MCMC: Metropolis Algorithm

Given a starting value $\theta^{(0)}$ at iteration $t = 0$, the algorithm proceeds as follows:

• **Metropolis Algorithm:** For $(t \in 1 : T)$, repeat:

1. Draw θ^* from $q(\cdot | \theta^{(t-1)})$

2. Compute the ratio

$$r = p(\theta^*) / p(\theta^{(t-1)})$$

3. If $r \geq 1$, set $\theta^{(t)} = \theta^*$;

$$\text{If } r < 1, \text{ set } \theta^{(t)} = \begin{cases} \theta^* & \text{with probability } r \\ \theta^{(t-1)} & \text{with probability } 1 - r \end{cases} .$$

MCMC: Metropolis Algorithm

Given a starting value $\theta^{(0)}$ at iteration $t = 0$, the algorithm proceeds as follows:

• **Metropolis Algorithm:** For $(t \in 1 : T)$, repeat:

1. Draw θ^* from $q(\cdot | \theta^{(t-1)})$

2. Compute the ratio

$$r = p(\theta^*) / p(\theta^{(t-1)}) =$$

3. If $r \geq 1$, set $\theta^{(t)} = \theta^*$;

$$\text{If } r < 1, \text{ set } \theta^{(t)} = \begin{cases} \theta^* & \text{with probability } r \\ \theta^{(t-1)} & \text{with probability } 1 - r \end{cases} .$$

Alternate notation:

$$P_{\text{move}} = \min \left(\frac{P(\theta_{\text{proposed}})}{P(\theta_{\text{current}})}, 1 \right)$$

MCMC: Metropolis Algorithm

Given a starting value $\theta^{(0)}$ at iteration $t = 0$, the algorithm proceeds as follows:

• **Metropolis Algorithm:** For $(t \in 1 : T)$, repeat:

1. Draw θ^* from $q(\cdot | \theta^{(t-1)})$

2. Compute the ratio

$$r = p(\theta^*) / p(\theta^{(t-1)}) = \exp[\log p(\theta^*) - \log p(\theta^{(t-1)})]$$

3. If $r \geq 1$, set $\theta^{(t)} = \theta^*$;

$$\text{If } r < 1, \text{ set } \theta^{(t)} = \begin{cases} \theta^* & \text{with probability } r \\ \theta^{(t-1)} & \text{with probability } 1 - r \end{cases} .$$

• Then a draw $\theta^{(t)}$ converges in distribution to a draw from the true posterior density $p(\theta | \mathbf{y})$.

Huh? Why does *that* work?

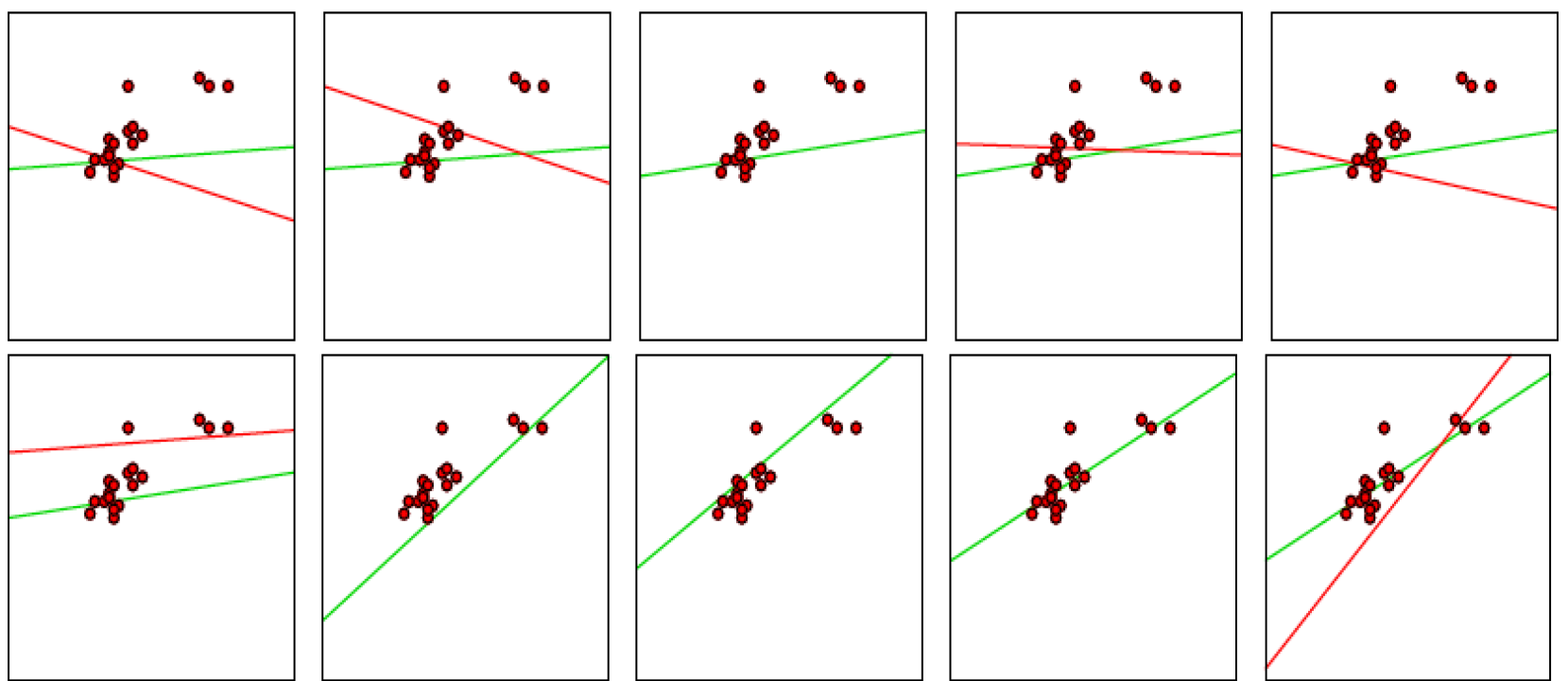
Huh? Why does *that* work?

$$\begin{aligned} \frac{p(\theta \rightarrow \theta+1)}{p(\theta+1 \rightarrow \theta)} &= \frac{.5 \min (P(\theta+1)/P(\theta), 1)}{.5 \min (P(\theta)/P(\theta+1), 1)} \\ &= \begin{cases} \frac{1}{P(\theta)/P(\theta+1)} & \text{if } P(\theta+1) > P(\theta) \\ \frac{P(\theta+1)/P(\theta)}{1} & \text{if } P(\theta+1) < P(\theta) \end{cases} \\ &= \frac{P(\theta+1)}{P(\theta)} \end{aligned}$$

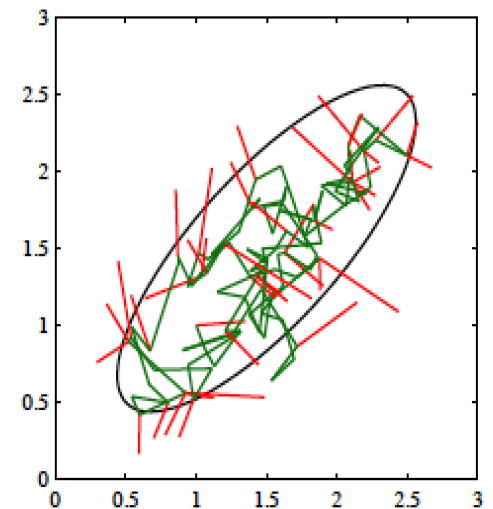
Intuition from last slide

- Probability of transitioning to one point vs the other in parameter space is ratio of their posterior probabilities (with $P = \text{prior} * \text{likelihood}$)
- No need to calculate $P(D)$ integral: cancels out in ratio $P(\theta+1)/P(\theta)$
- So, just plot histogram of how often each point visited; with enough samples can reconstruct posterior distribution!
- While $\text{prior} * \text{likelihood}$ may be high for any one point, need to know how high it is *relative* to other parameters (normalization). We get this only after sampling many points and looking at the final relative density

Metropolis algorithm



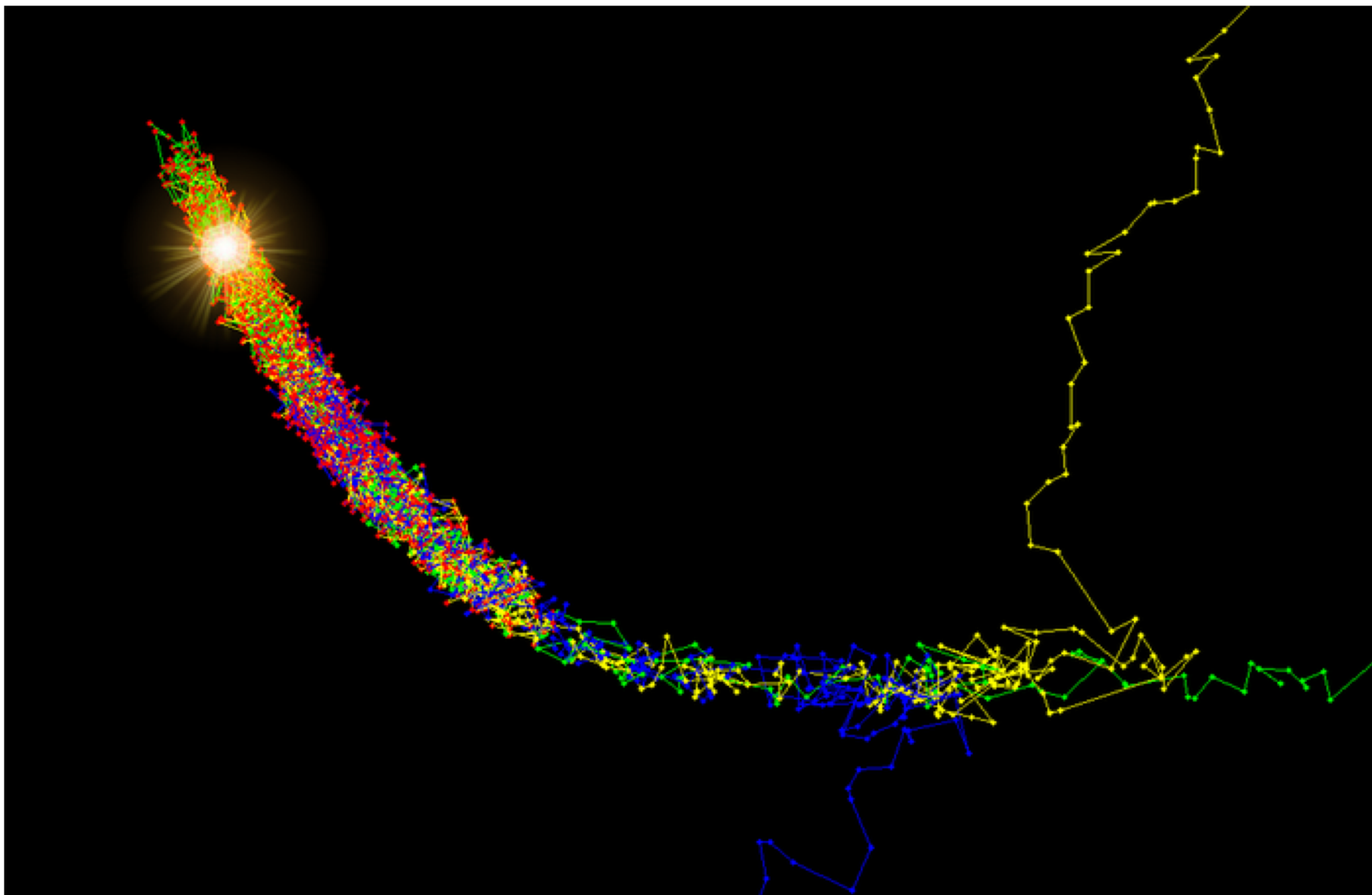
- Perturb parameters: $Q(\theta'; \theta)$, e.g. $\mathcal{N}(\theta, \sigma^2)$
- Accept with probability $\min\left(1, \frac{\tilde{P}(\theta'|\mathcal{D})}{\tilde{P}(\theta|\mathcal{D})}\right)$
- Otherwise **keep old parameters**



Detail: Metropolis, as stated, requires $Q(\theta'; \theta) = Q(\theta; \theta')$

This subfigure from PRML, Bishop (2006)

Metropolis Algorithm: Burn in and convergence



Consistency checks

Do I get the right answer on tiny versions of my problem?

Can I make good inferences about synthetic data drawn from my model?

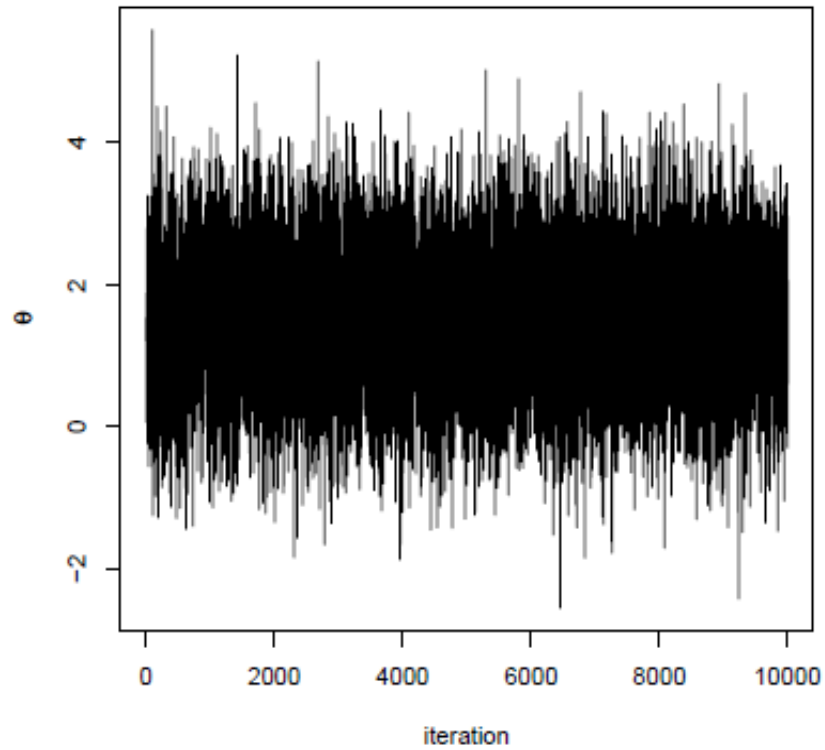
Getting it right: joint distribution tests of posterior simulators, John Geweke, *JASA*, 99(467):799–804, 2004.

Convergence Issues: Trace Plot

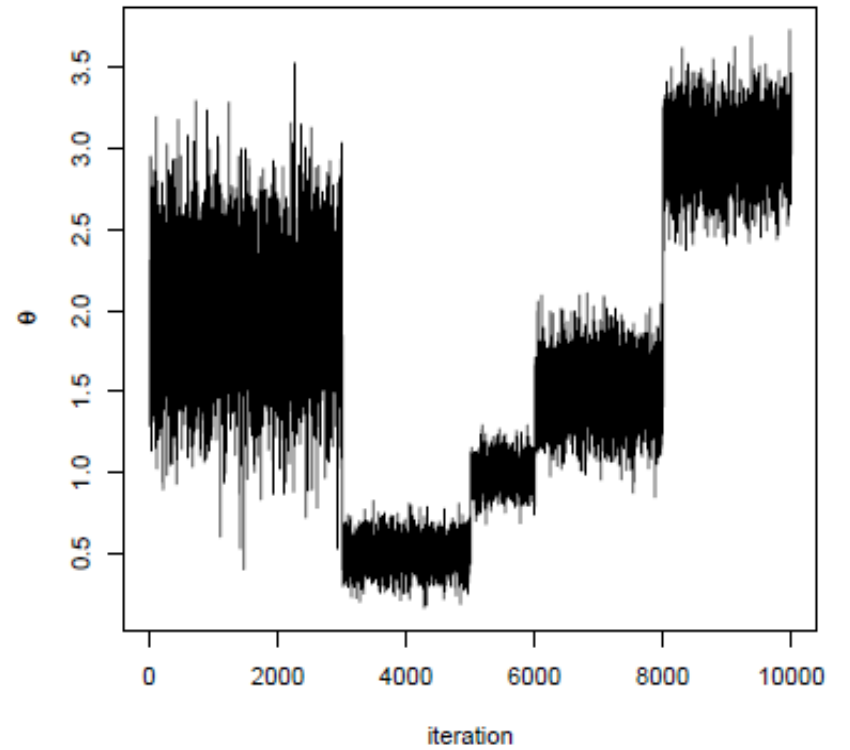
- One way to see if our chain has converged is to see how well our chain is mixing, or moving around the parameter space.
- If our chain is taking a long time to move around the parameter space, then it will take longer to converge.
- A trace plot is a plot of the iteration number against the value of the draw of the parameter at each iteration.

Trace Plot

Good Mixing



Bad Mixing



Autocorrelation

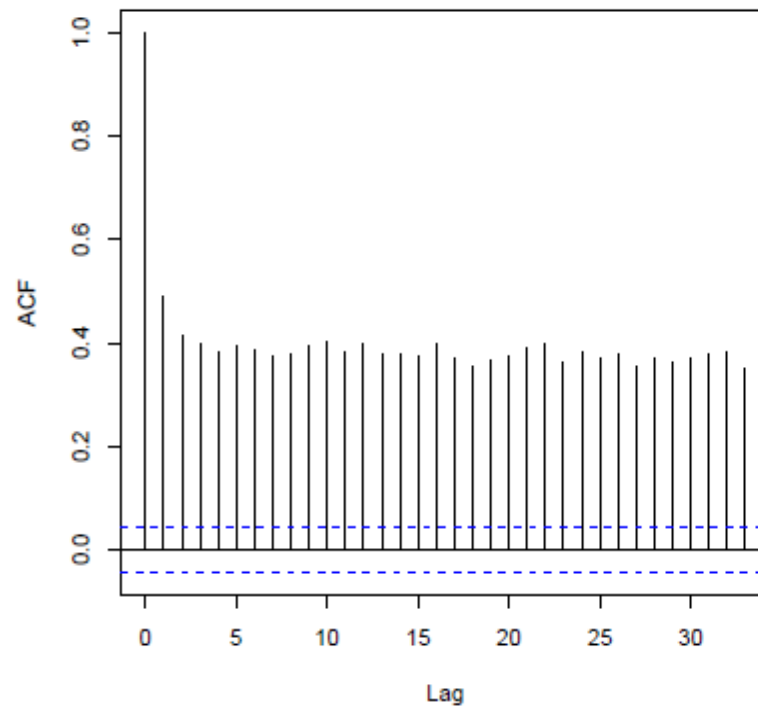
The lag k autocorrelation ρ_k is the correlation between every draw and its k th lag:

We would expect the k th lag autocorrelation to be smaller as k increases

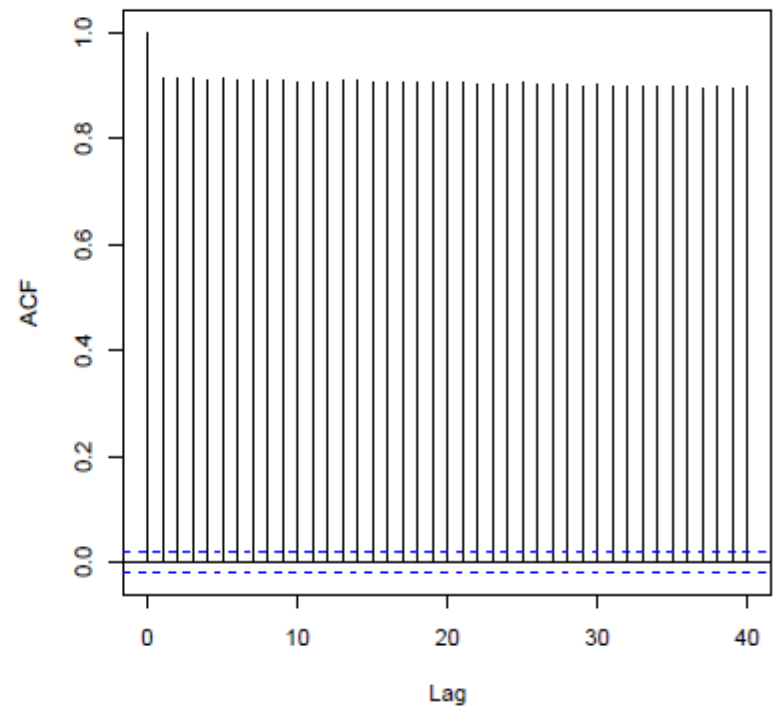
$$\rho_k = \frac{\sum_{i=1}^{n-k} (x_i - \bar{x})(x_{i+k} - \bar{x})}{\sum_{i=1}^n (x_i - \bar{x})}$$

Autocorrelation

Good Mixing



Bad Mixing

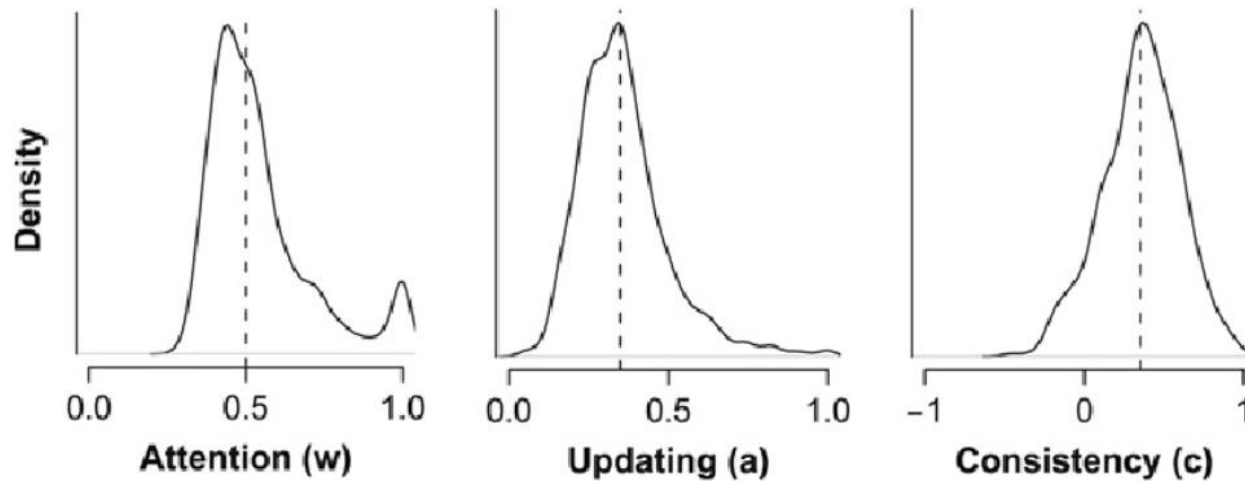


Gelman-Rubin R Statistic:

Run multiple chains and see if they get the same answer

- Run $m \geq 2$ chains of length $2n$ from over-dispersed starting values.
- Discard the first n draws in each chain.
- Calculate the within-chain and between-chain variance.
- Calculate the estimated variance of the parameter as a weighted sum of the within-chain and between-chain variance.
- Calculate the potential scale reduction factor.

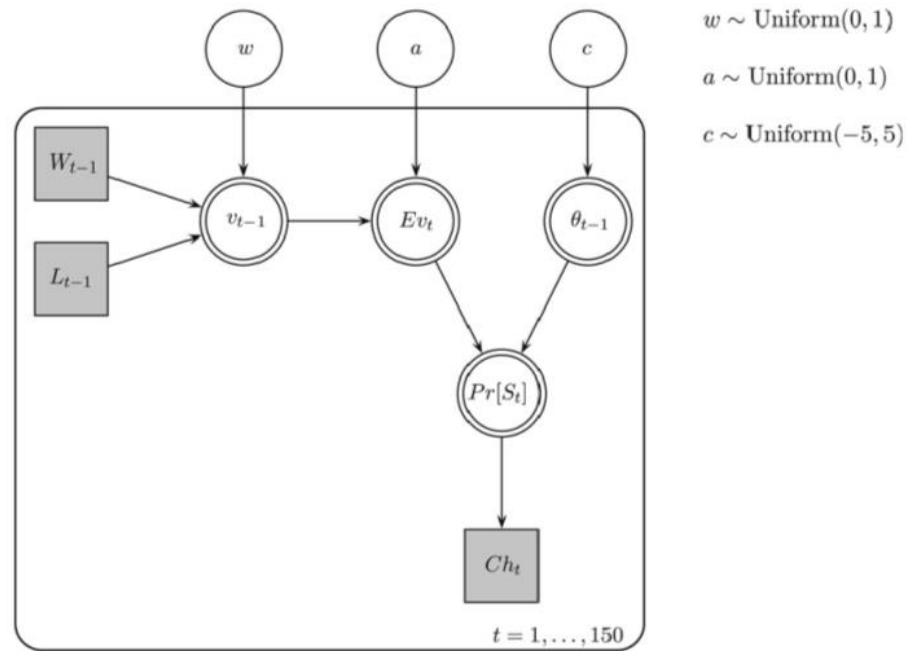
Remember Maximum likelihood estimation?



- Generative model:
1000 **simulated runs** of the same participant (ie same params), 150 trials each
- means of MLE parameters are very close to true values (dashed line)
- some individual runs can be off, and param at bounds
- **in practice, with MLE can't get these distributions for a single subject** (unless we tested them many times and assumed no meta-learning!)

Expectancy Valence model Iowa Gambling Task, Bayes (MCMC)

$$P(\theta|D) \propto P(D|\theta)P(\theta)$$

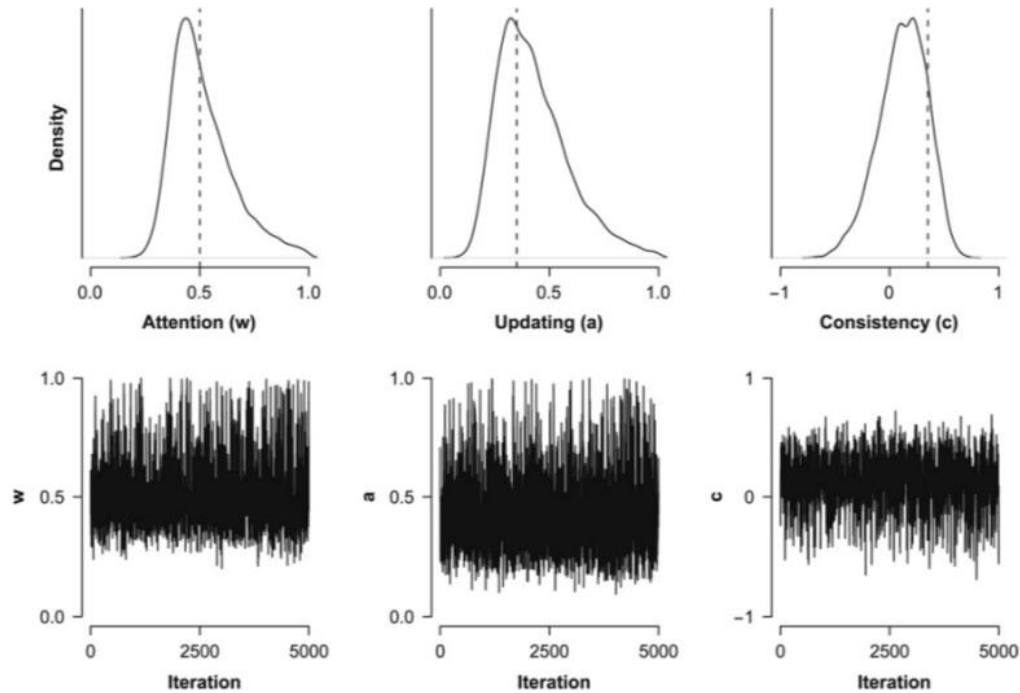


use sampling (MCMC) to do bayes inference (integrals hard to compute analytically)

Tools: STAN, JAGS, WinBUGS and matlab or R; Python (pyMC)

Posterior distributions

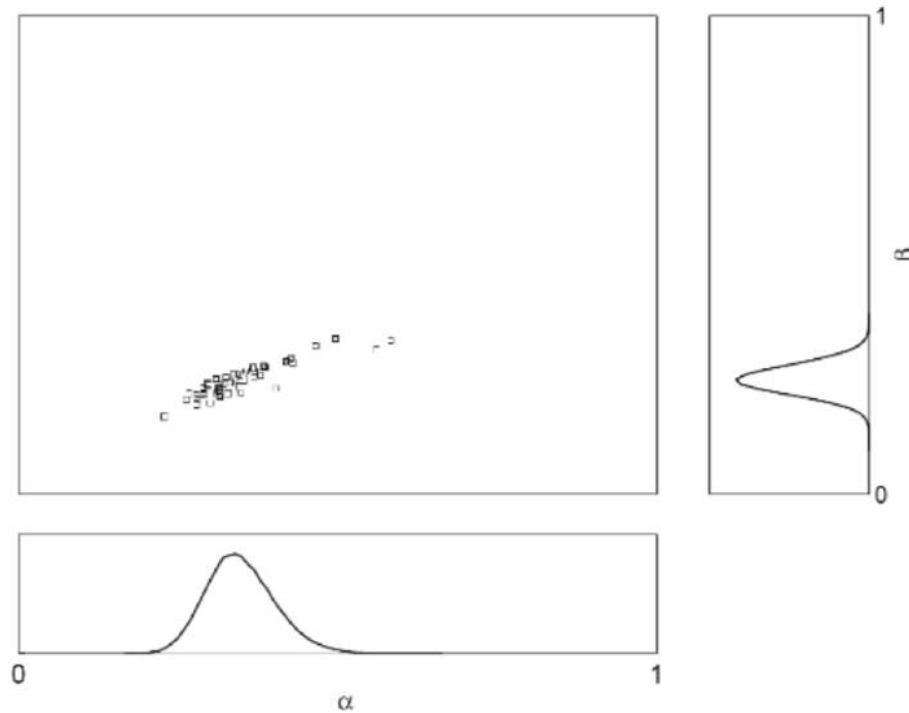
one simulated subject, one run, 150 trials



- Posterior distribution shows uncertainty: single subject fit!

Joint and Marginal Posterior Distributions

- Note that the joint posterior distribution has more information than the marginal distributions



Fitted RL-DDM, joint distributions ADHD on/off meds

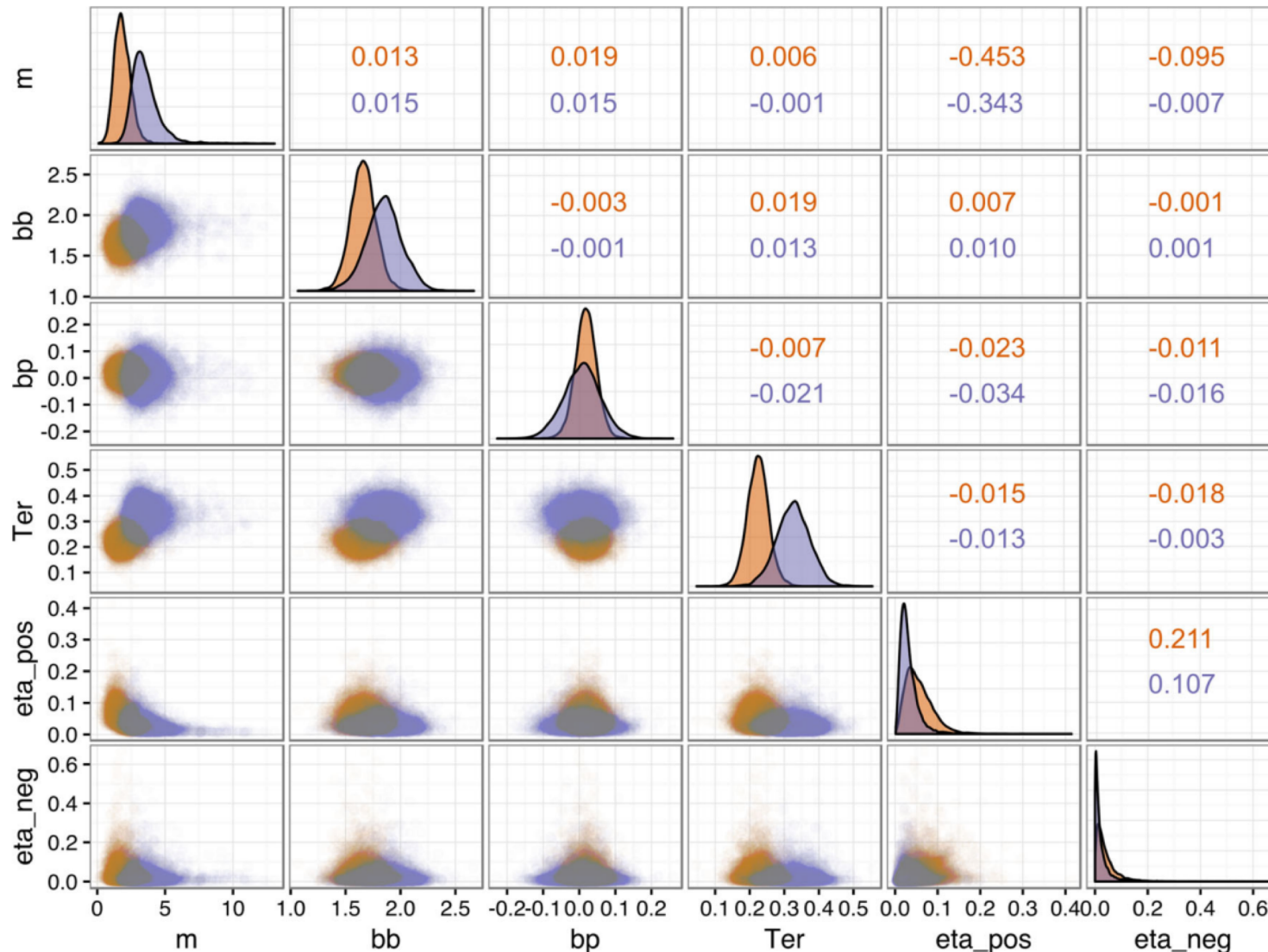


Fig. 2 Scatterplot and density of group parameter estimates from posterior distributions off (red) and on (purple) medication. bb = boundary baseline, bp = boundary power, eta_pos = learning

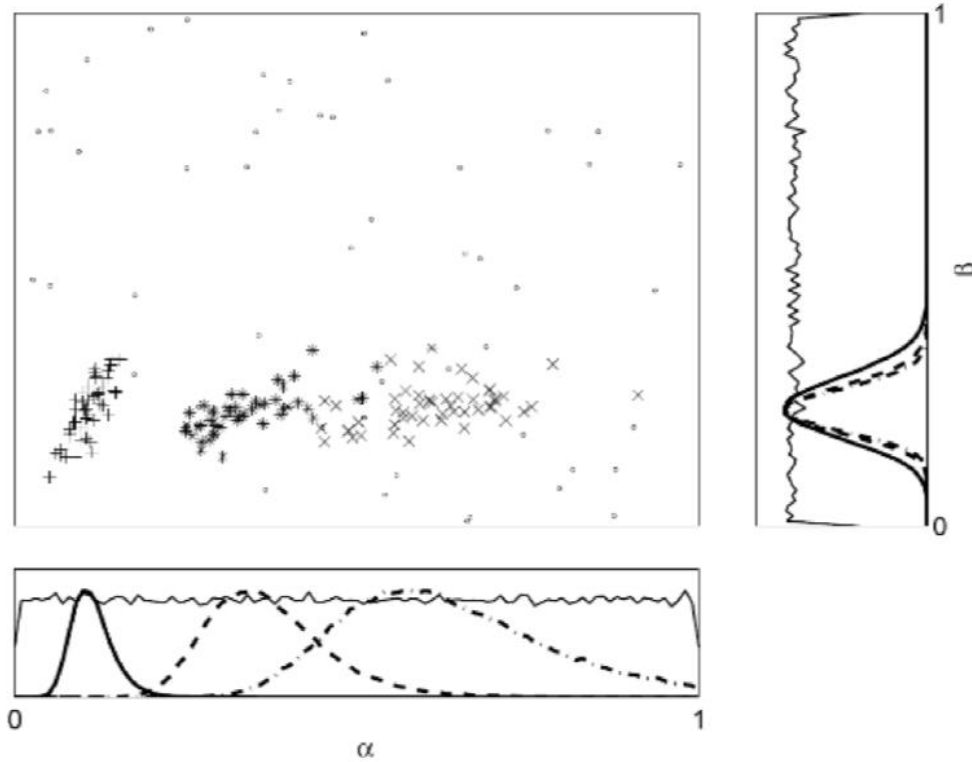
rate for positive prediction errors (PEs), eta_neg = learning rate for negative PEs, m = drift rate scaling, T_{er} = nondecision time

Groups and individuals

- Two extremes:
 - Fit every participant separately, as if they are completely unrelated;
 - Pool the data and assume that participants are identical copies
- Both assumptions are unreasonable

Joint and Marginal Posterior Distributions: Separate individual model

- The individual differences show in the posterior, but the final (data-less) subject is still modeled by the prior



Groups and individuals

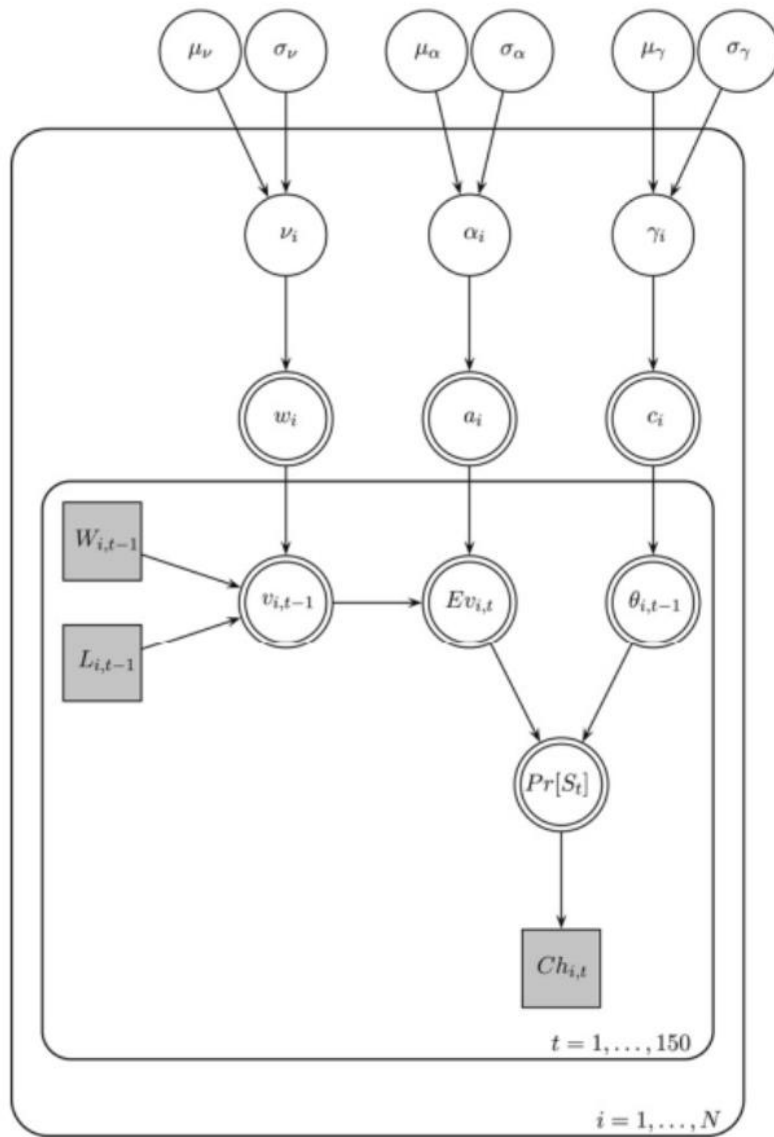
- Two extremes:
 - Fit every participant separately, as if they are completely unrelated;
 - Pool the data and assume that participants are identical copies
- Both assumptions are unreasonable
- Compromise: participants are similar, yet different (random effects)
- Individual subject parameters are drawn from distributions, and we also infer the parameters of those group distributions...

Groups and individuals

- Two extremes:
 - Fit every participant separately, as if they are completely unrelated;
 - Pool the data and assume that participants are identical copies
- Both assumptions are unreasonable
- Compromise: participants are similar, yet different (random effects)
- Individual subject parameters are drawn from distributions, and we also infer the parameters of those group distributions...

$$P(\theta, \lambda|D) = \frac{P(D|\theta)P(\theta|\lambda)P(\lambda)}{P(D)}$$

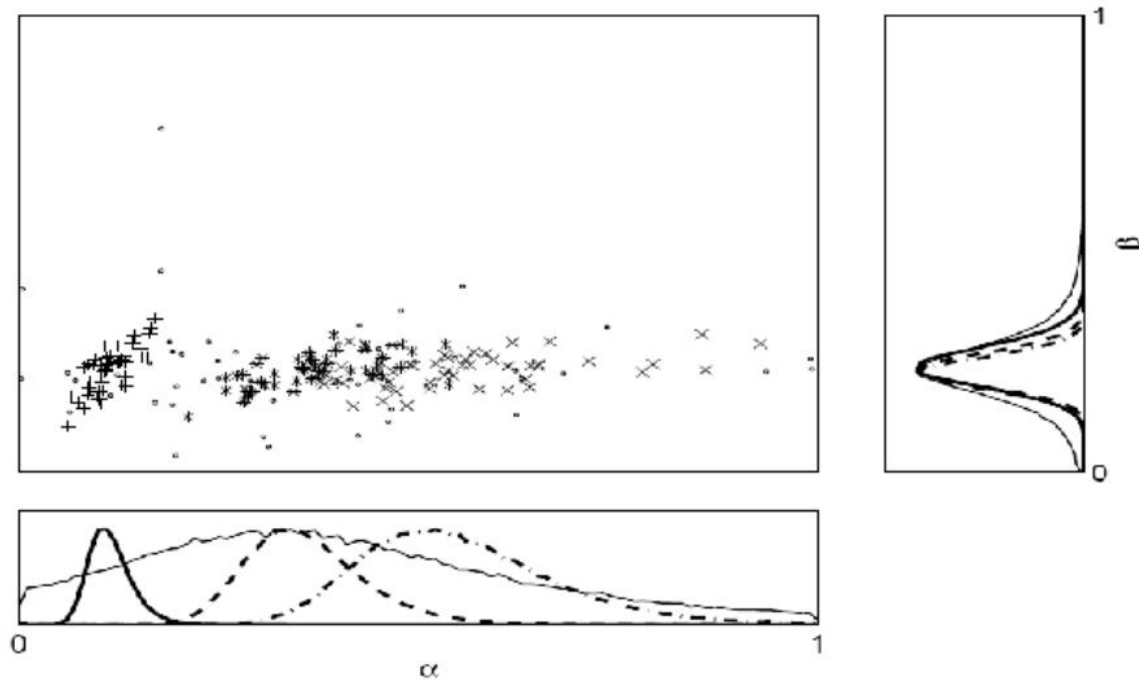
$$P(\theta, \lambda|D) = \frac{P(D|\theta)P(\theta|\lambda)P(\lambda)}{\int P(D|\theta)P(\theta|\lambda)P(\lambda)d\theta d\lambda}$$



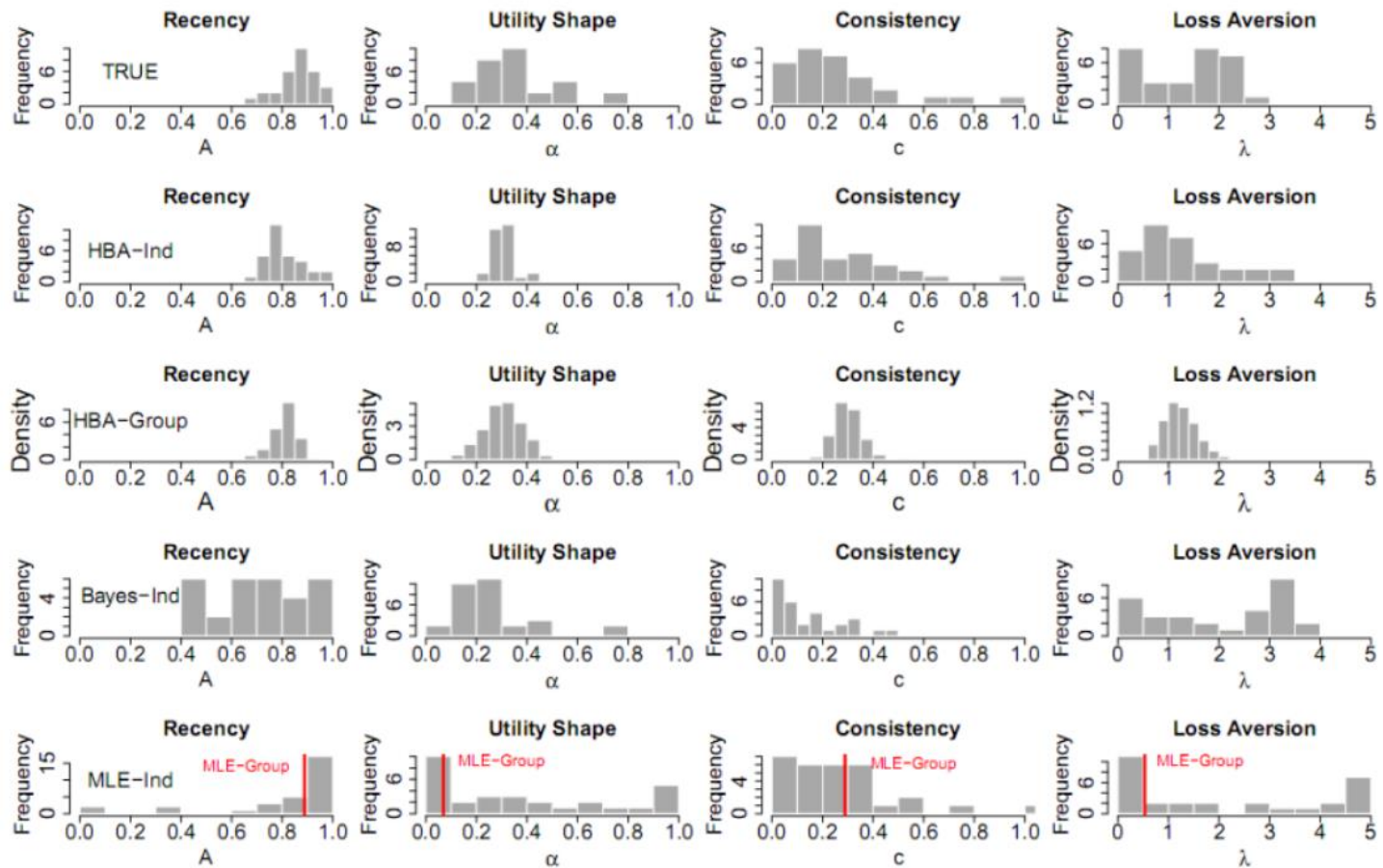
$\mu \sim \text{Normal}(0, 1)$
 $\sigma \sim \text{Uniform}(0, 1.5)$
 $\nu_i \sim \text{Normal}(\mu_\nu, \lambda_\nu)$
 $\alpha_i \sim \text{Normal}(\mu_\alpha, \lambda_\alpha)$
 $\gamma_i \sim \text{Normal}(\mu_\gamma, \lambda_\gamma)$
 $\nu_i = \text{Probit}(w_i)$
 $\alpha_i = \text{Probit}(a_i)$
 $\gamma_i = \text{Probit}(c_i)$

Joint and Marginal Posterior Distributions: Structured individual differences model

- First three subjects give almost the same parameter inference, but now the fourth subject borrows from what is learned about them (“sharing statistical strength”)



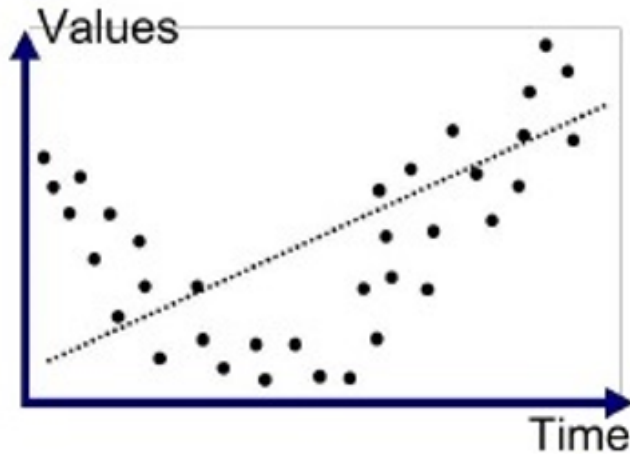
Hierarchical Bayes Improves Parameter Recovery in RL and DM models



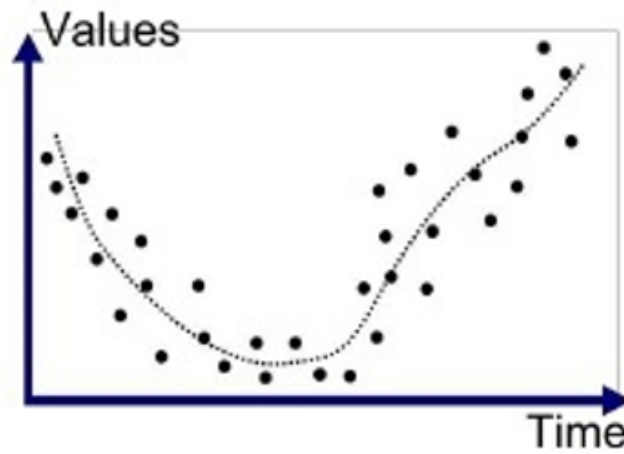
Ahn et al in 2011
Wiecki, Sofer & Frank 2013
Pedersen, Frank & Biele 2017

Model Selection / Comparison

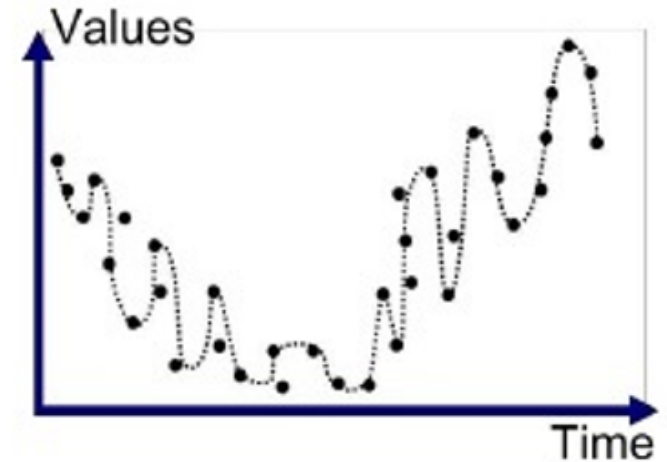
- *“A scientific theory should be as simple as possible, but no simpler”*



Underfitted



Good Fit/Robust



Overfitted

- Various metrics of model selection (Bayes Factor, BIC, AIC, DIC, WAIC, LOO, FE..),
- but all are, roughly: $-P(D|M) + \text{complexity}(M)$
- None is perfect. Do model recovery simulations for your task and models!

Posterior probabilities and Bayes factors (BF)

Posterior probability for each model

$$\pi(\mathcal{M}_k|\mathbf{y}) = \frac{\pi(\mathbf{y}|\mathcal{M}_k)\pi(\mathcal{M}_k)}{\sum_i \pi(\mathbf{y}|\mathcal{M}_i)\pi(\mathcal{M}_i)}$$

Marginal likelihood: $\pi(\mathbf{y}|\mathcal{M}_k) = \int \pi(\mathbf{y}|\boldsymbol{\theta}_k, \mathcal{M}_k)\pi(\boldsymbol{\theta}_k|\mathcal{M}_k)d\boldsymbol{\theta}_k$

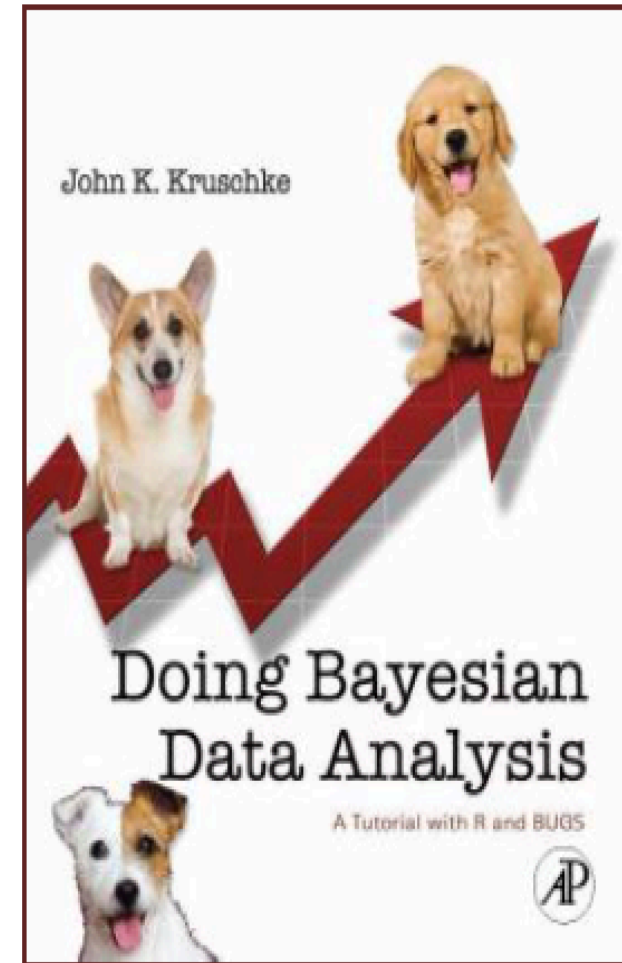
Posterior odds:

$$\frac{\pi(\mathcal{M}_k|\mathbf{y})}{\pi(\mathcal{M}_j|\mathbf{y})} = \underbrace{\frac{\pi(\mathbf{y}|\mathcal{M}_k)}{\pi(\mathbf{y}|\mathcal{M}_j)}}_{\text{Bayes factor}} \times \underbrace{\frac{\pi(\mathcal{M}_k)}{\pi(\mathcal{M}_j)}}_{\text{prior odds}}$$

- ▶ Choose the model with the largest probability $\pi(\mathcal{M}_k|\mathbf{y})$
- ▶ BF gives a measure of evidence for model \mathcal{M}_k versus \mathcal{M}_j

Model Comparison

*The magazine model comparison game
Leaves all of us wishing that we looked like them.
But they have mere fantasy's bogus appeal,
'Cause none obeys **fact or** respects what is real.*



Warning! (and opportunities...)

Just because a model accounts for most variance doesn't mean it is for the reason that we think. Parameters soak up variance as best they can given model constraints.

e.g., α is NOT corticostriatal synaptic plasticity. α = update of organism as a whole..

Warning! (and opportunities...)

How well does the “best” (least worst) model fit key features of the data?

.

Warning! (and opportunities...)

How well does the “best” (least worst) model fit key features of the data?

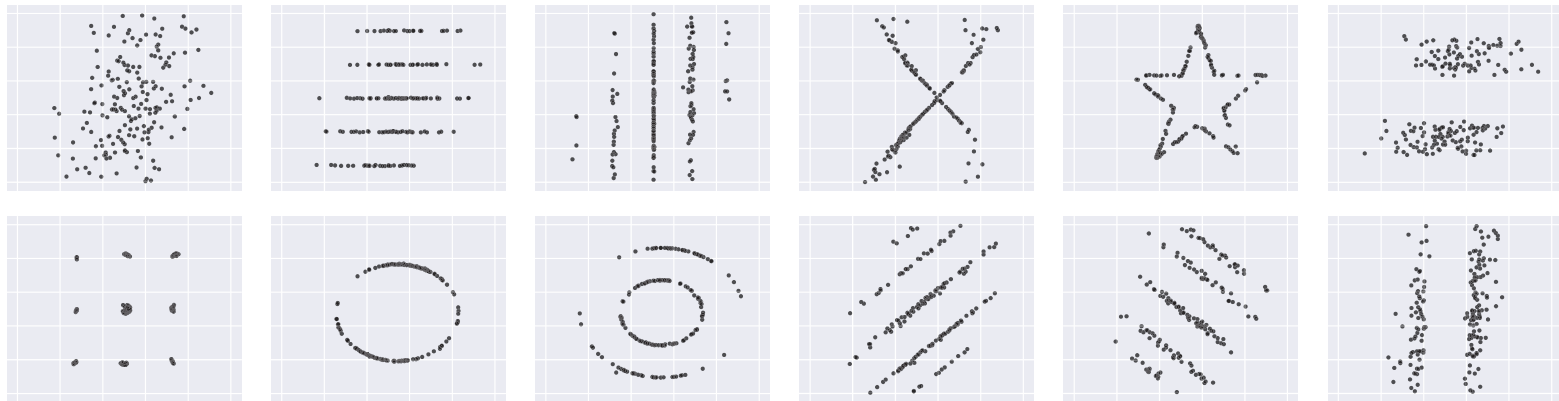


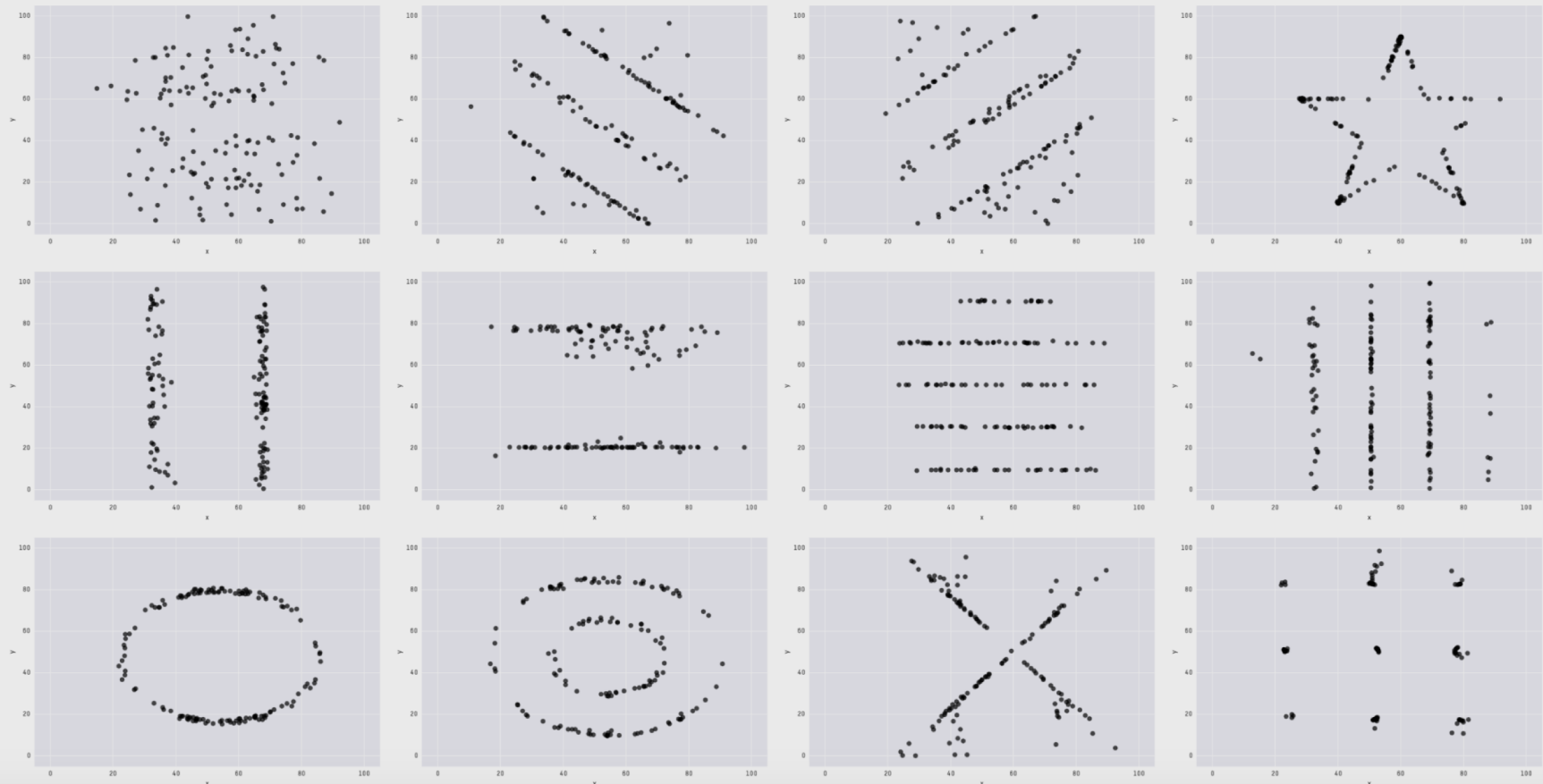
Figure 1. A collection of data sets produced by our technique. While different in appearance, each has the same summary statistics (mean, std. deviation, and Pearson's corr.) to 2 decimal places. ($\bar{x}=54.02$, $\bar{y}=48.09$, $sd_x=14.52$, $sd_y=24.79$, Pearson's $r=+0.32$)

Warning! (and opportunities...)

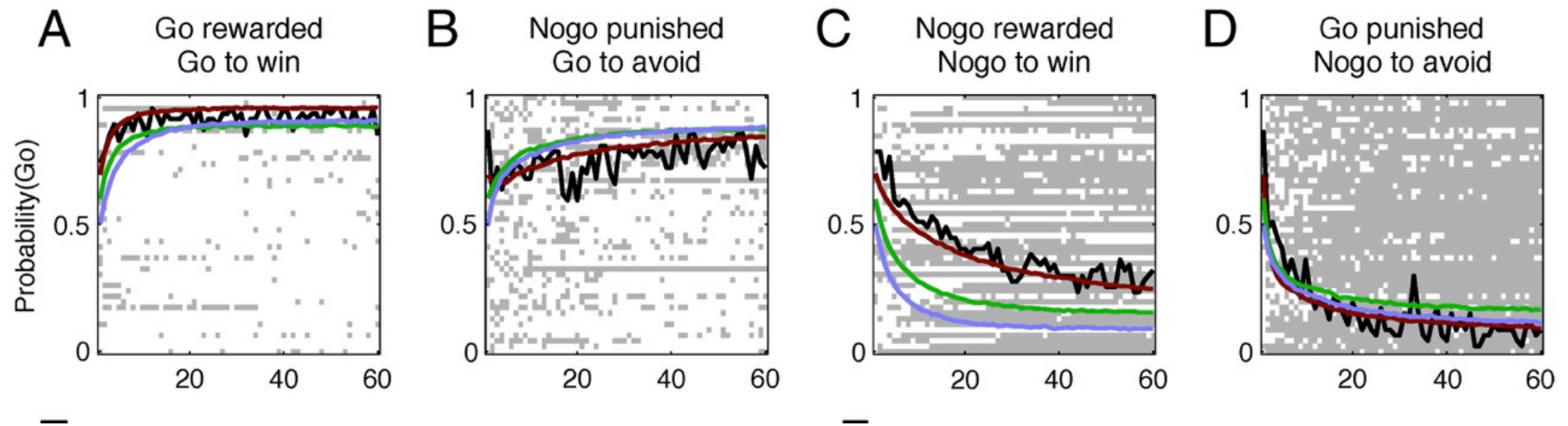
How well does the “best” (least worst) model fit key features of the data?



X Mean: 54.26
Y Mean: 47.83
X SD : 16.76
Y SD : 26.93
Corr. : -0.06



Model validation is informative about where your model misses



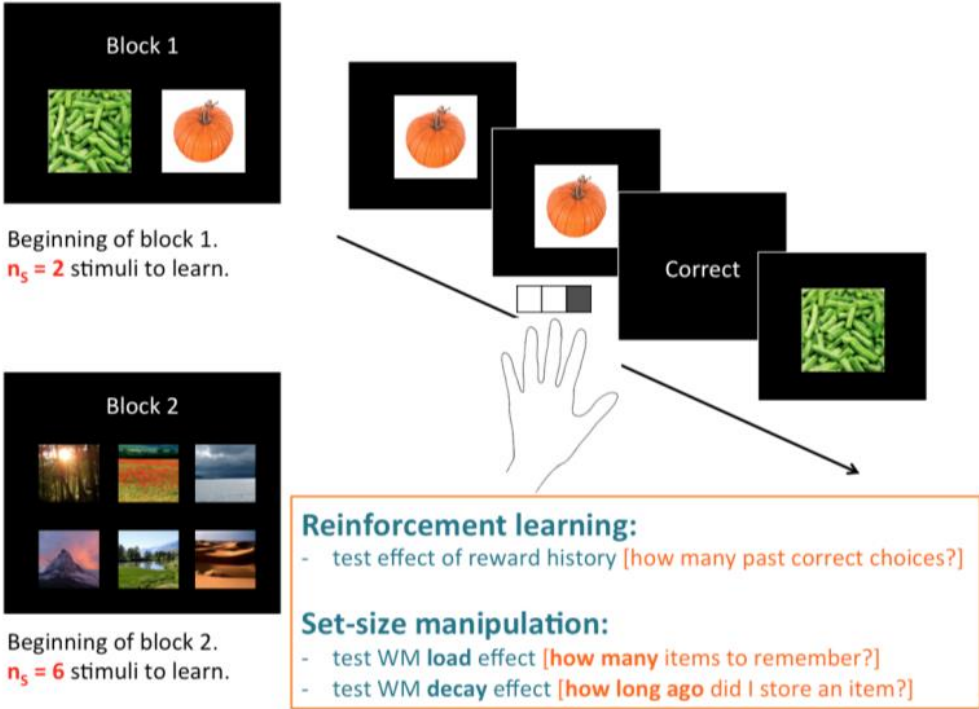
Model validation: simple RL model

$$Q_{t+1} \leftarrow Q_t + \alpha \text{RPE}$$

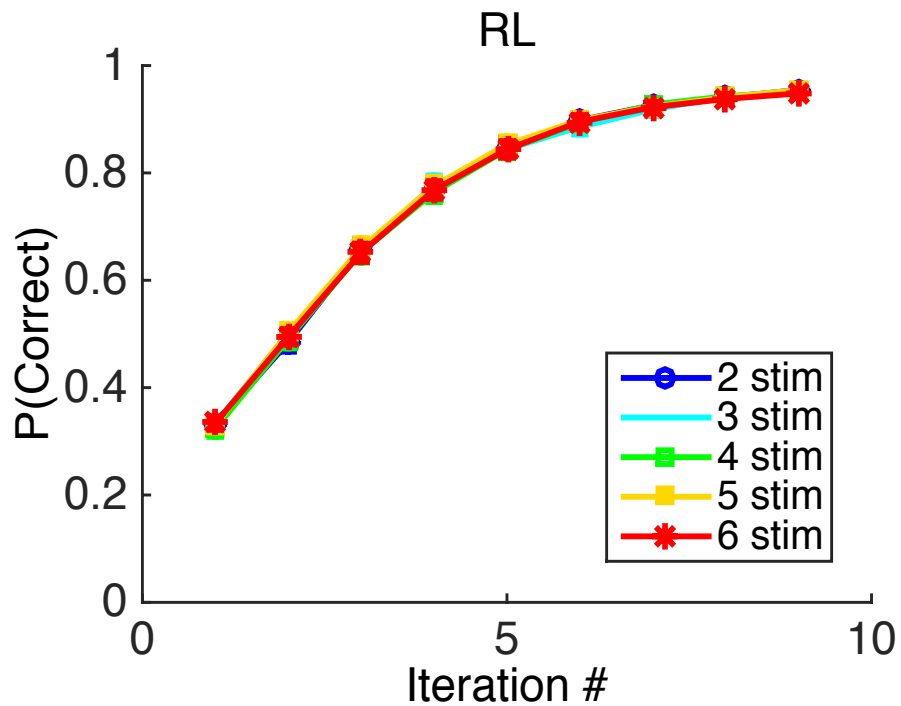
Each option value is updated with RPE and stored until needed again

RLWM Task

/



RL Model simulations



of stimuli to learn
in the block

$$n_s = 2$$

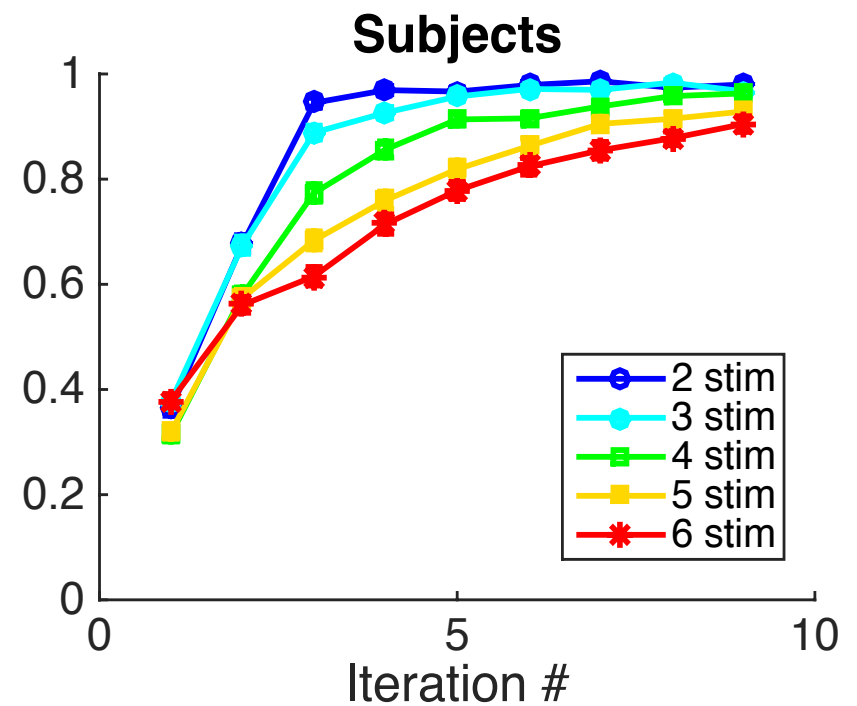
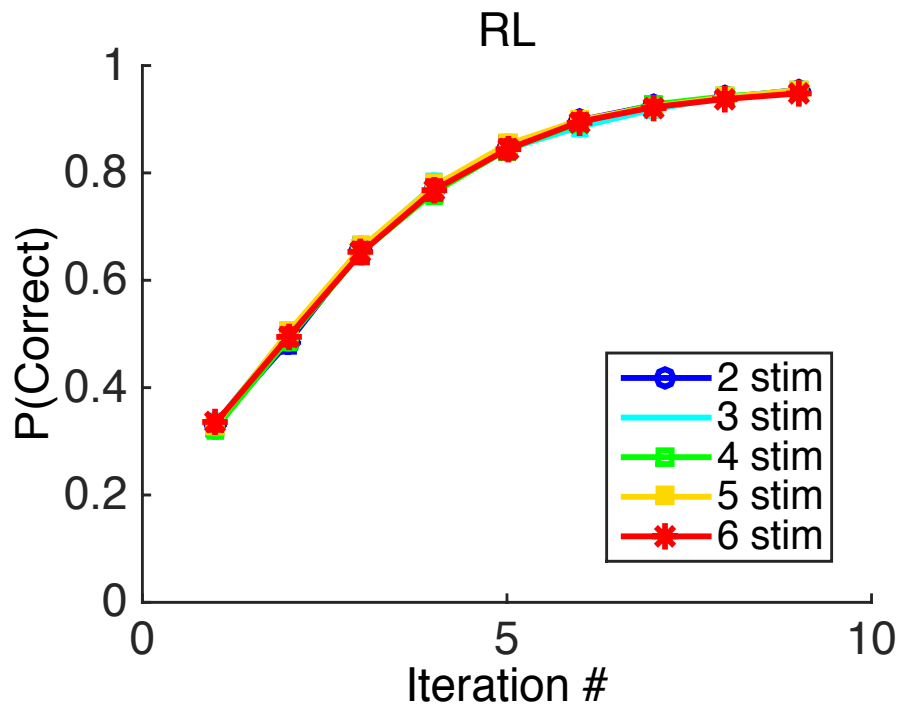
$$n_s = 3$$

$$n_s = 4$$

$$n_s = 5$$

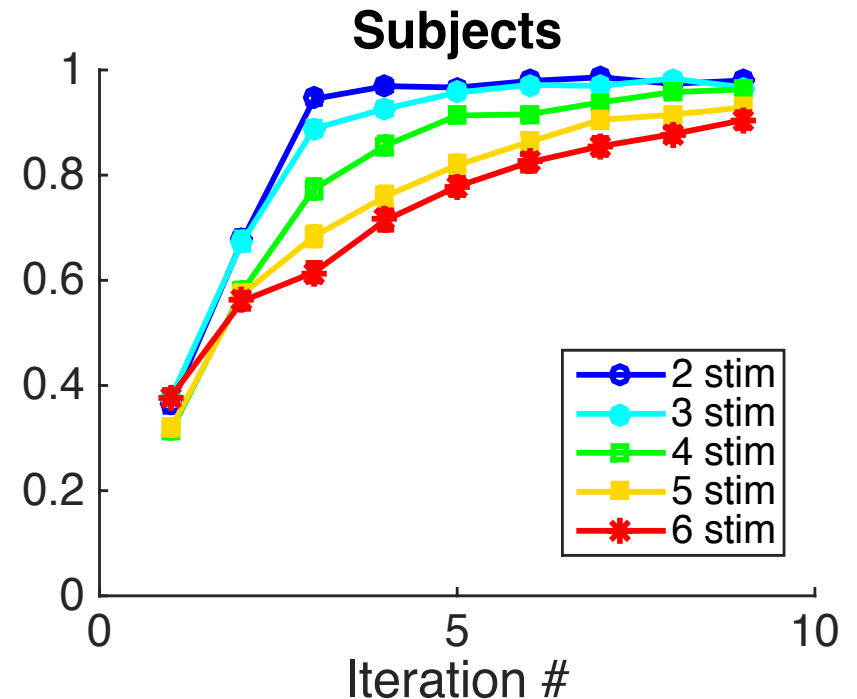
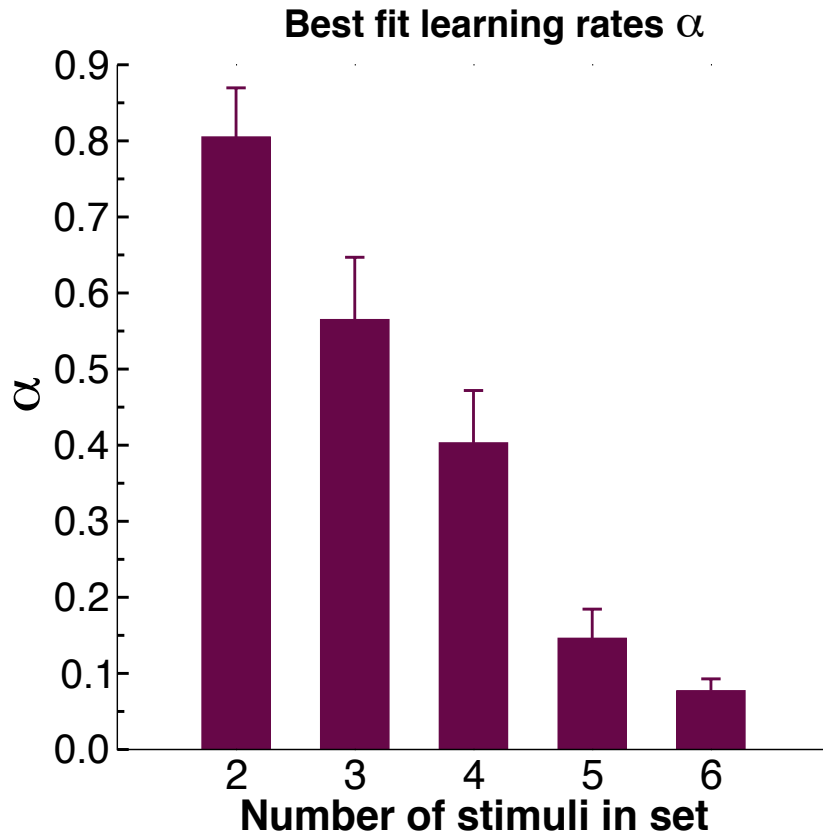
$$n_s = 6$$

Model validation failure: simulate your data!



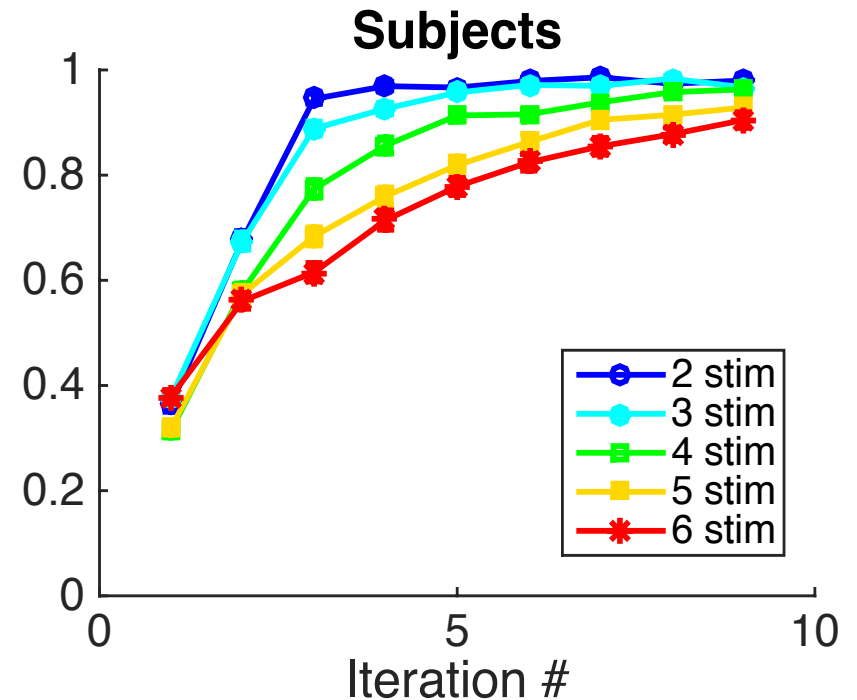
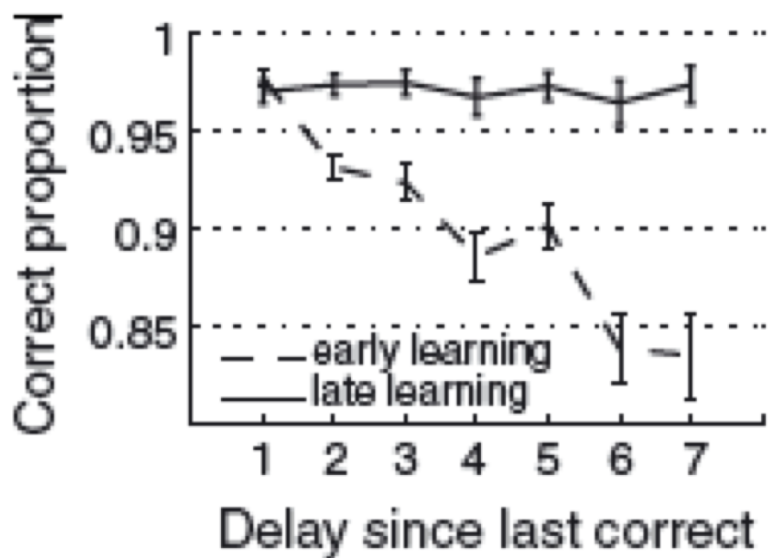
Effect of set-size on learning is not accounted for by vanilla RL

Model validation failure



Effect of set-size on learning is not accounted for by vanilla RL (even with multiple learning rates)

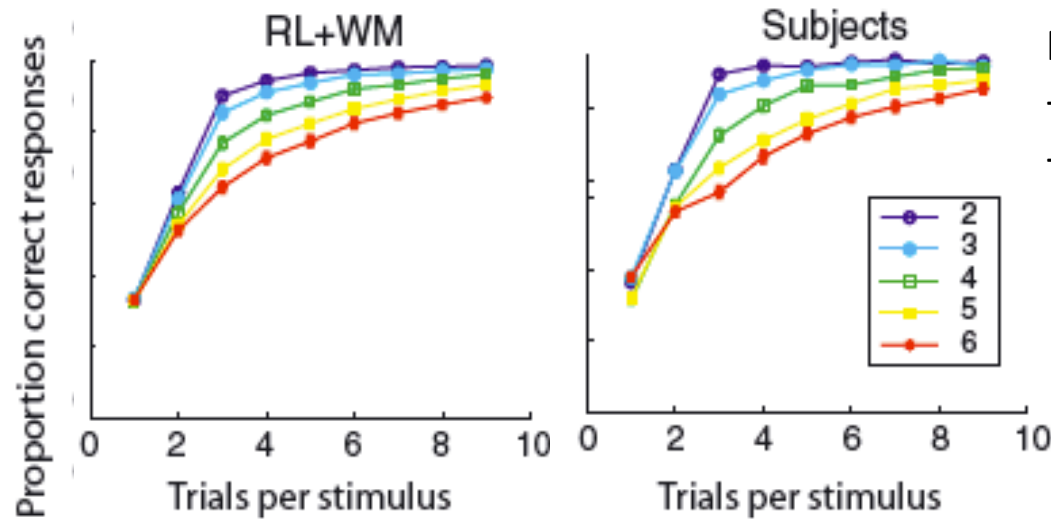
Model validation failure



*Effect of **delay** on learning is not accounted for by vanilla RL*

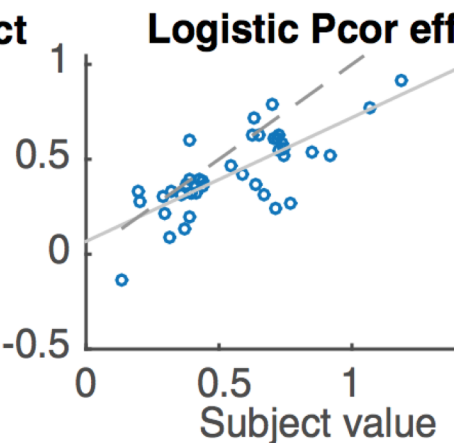
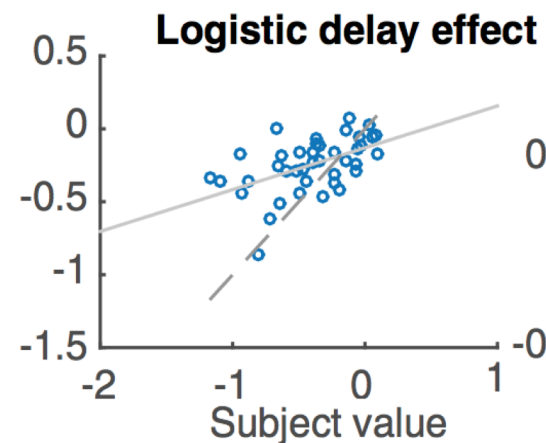
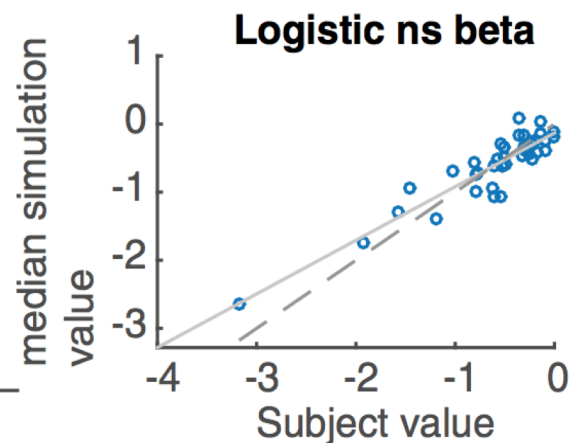
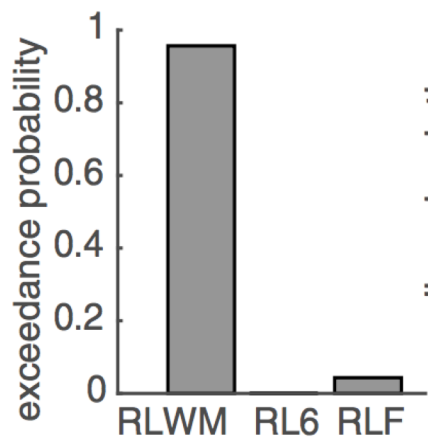
Model validation success!

RL+WM explains behavior

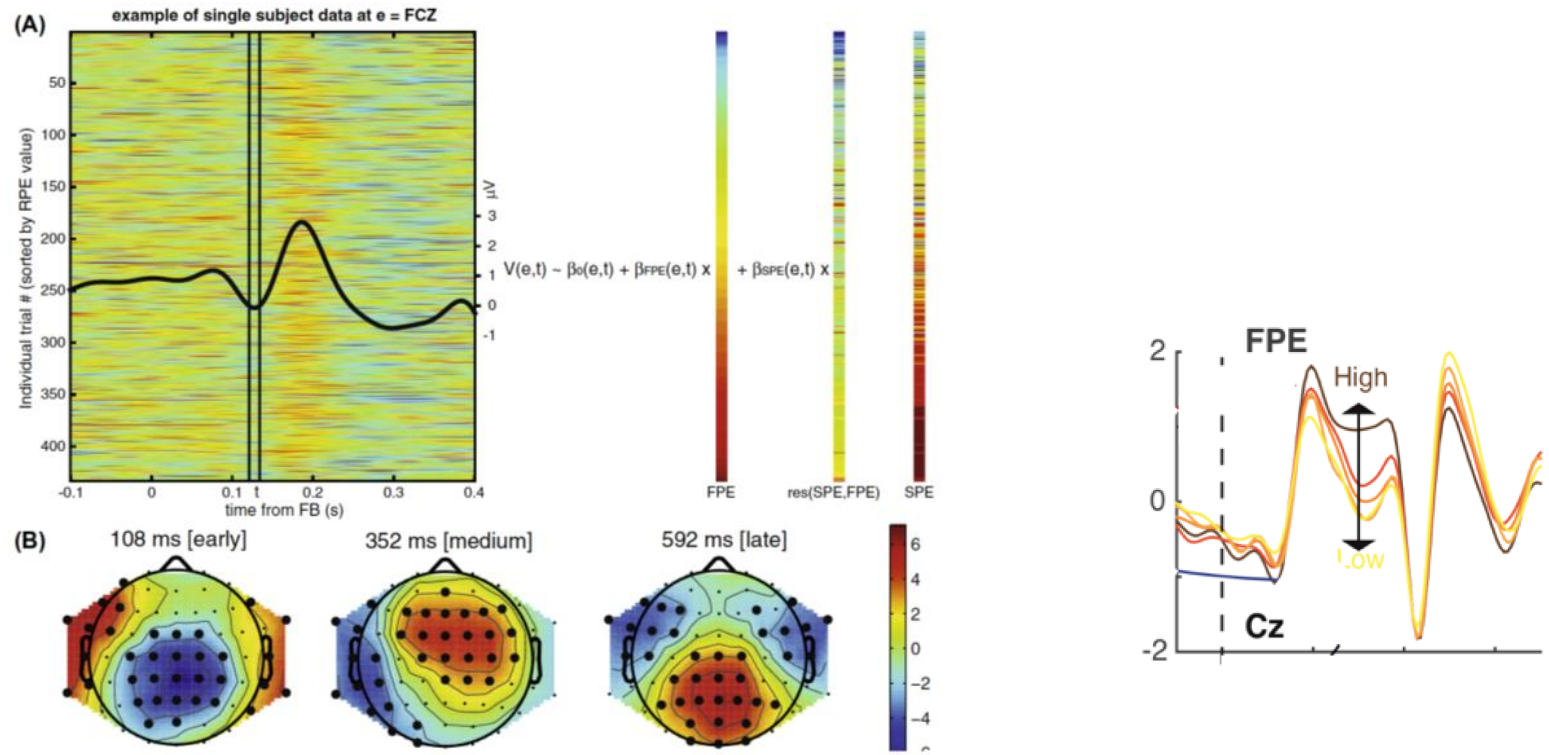


RLWM accounts for both load and delay effects

- Decrease in **load** effect as RL learns
- Decrease in **delay** effect as RL becomes more reliable than WM

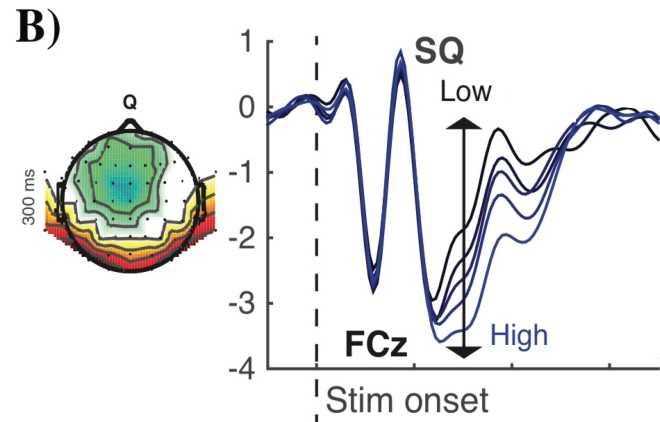


Model-based EEG



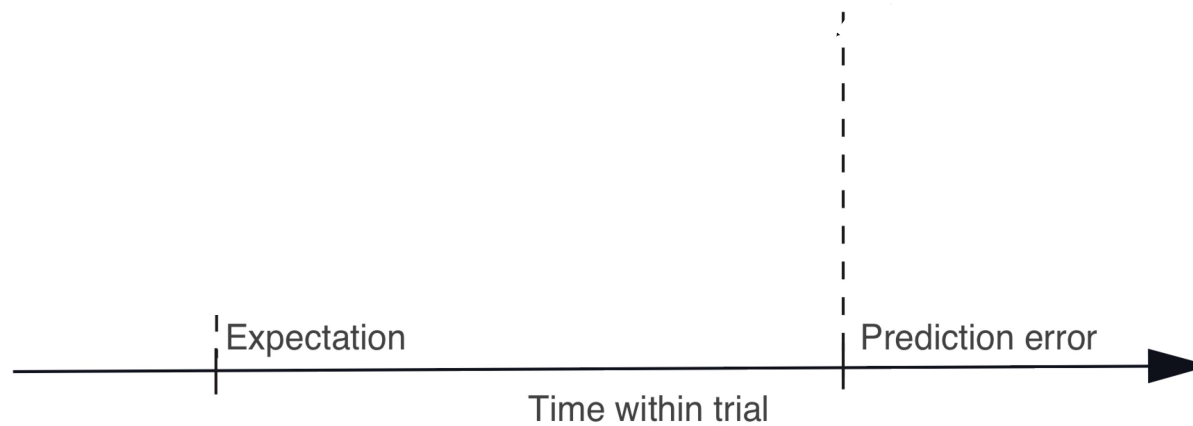
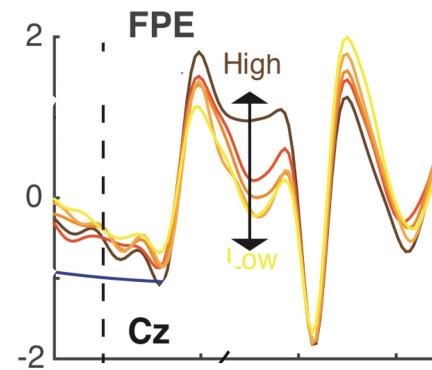
Collins & Frank 2016 *Cognition*
 Collins & Frank 2018 *PNAS*

Within-trial dynamics: RL

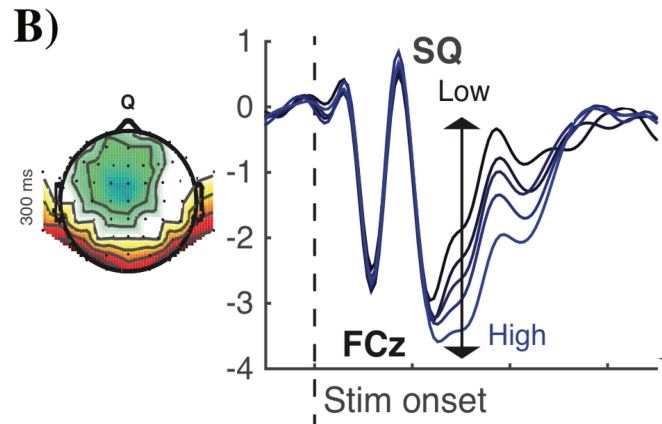


$$RPE = R(t) - Q(t)$$

Note: reward = 1
In all cases

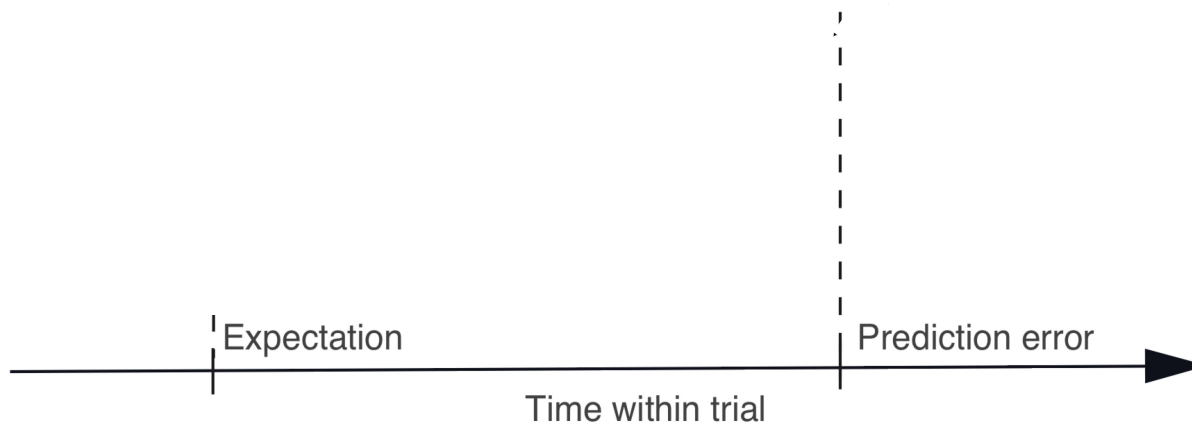
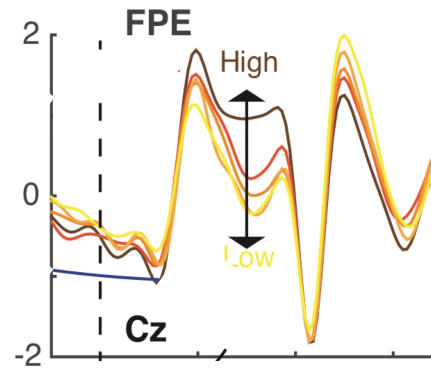


Within-trial dynamics: RL

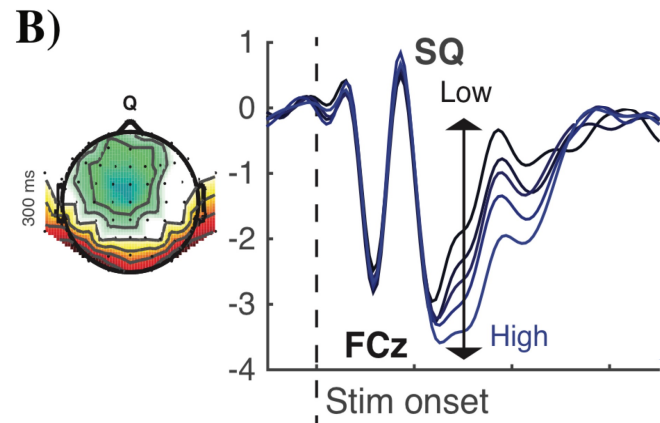


$$RPE = R(t) - \hat{Q}(t)$$

Note: reward = 1
In all cases

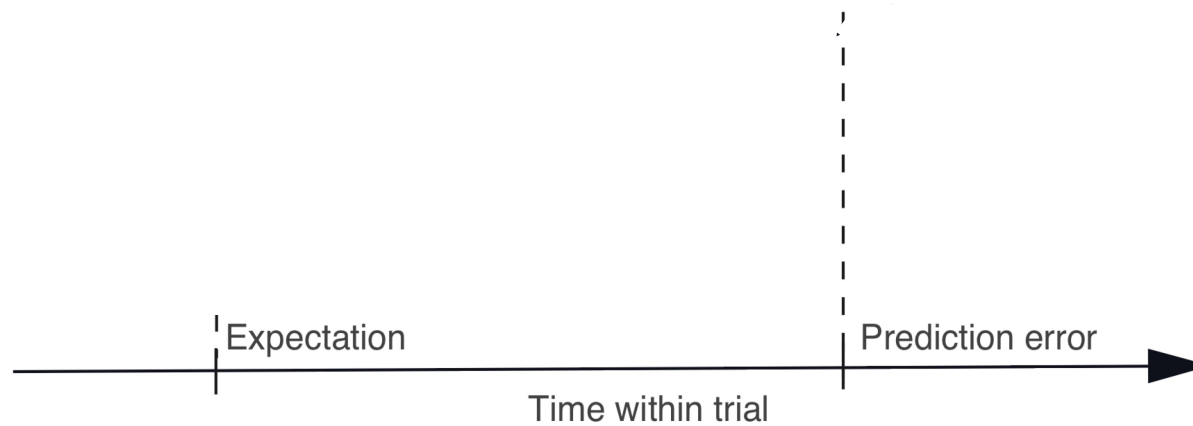
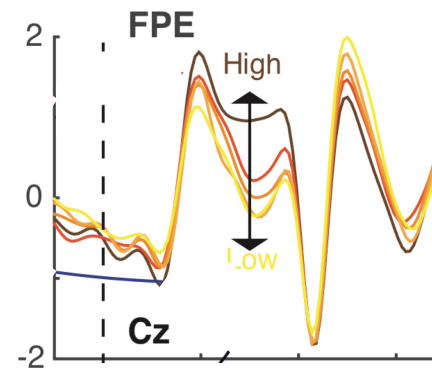


Within-trial dynamics: RL

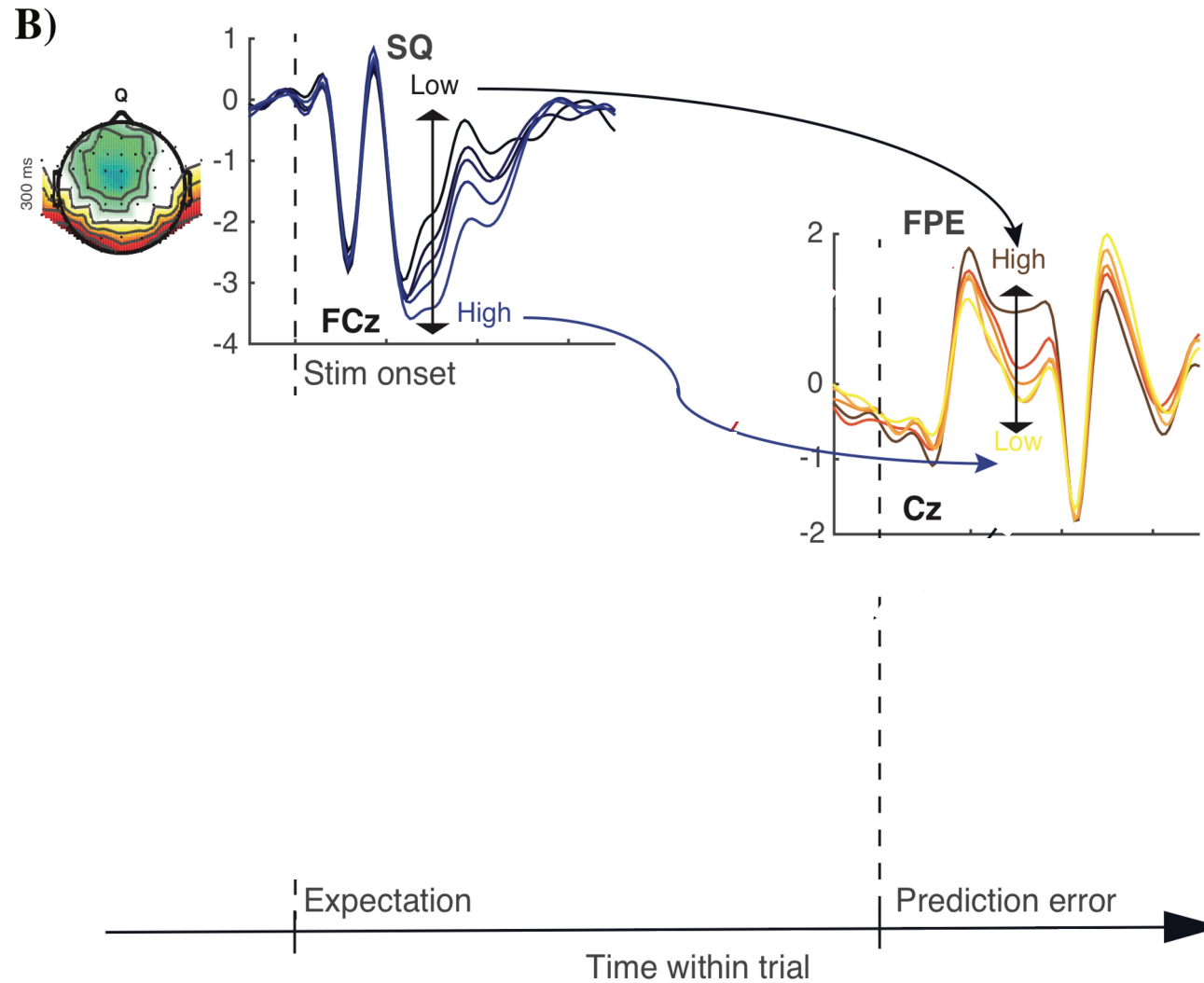


$$FPE = R(t) - SQ(t)$$

Note: reward = 1
In all cases



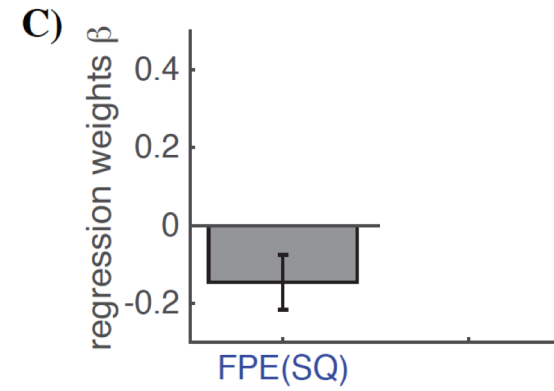
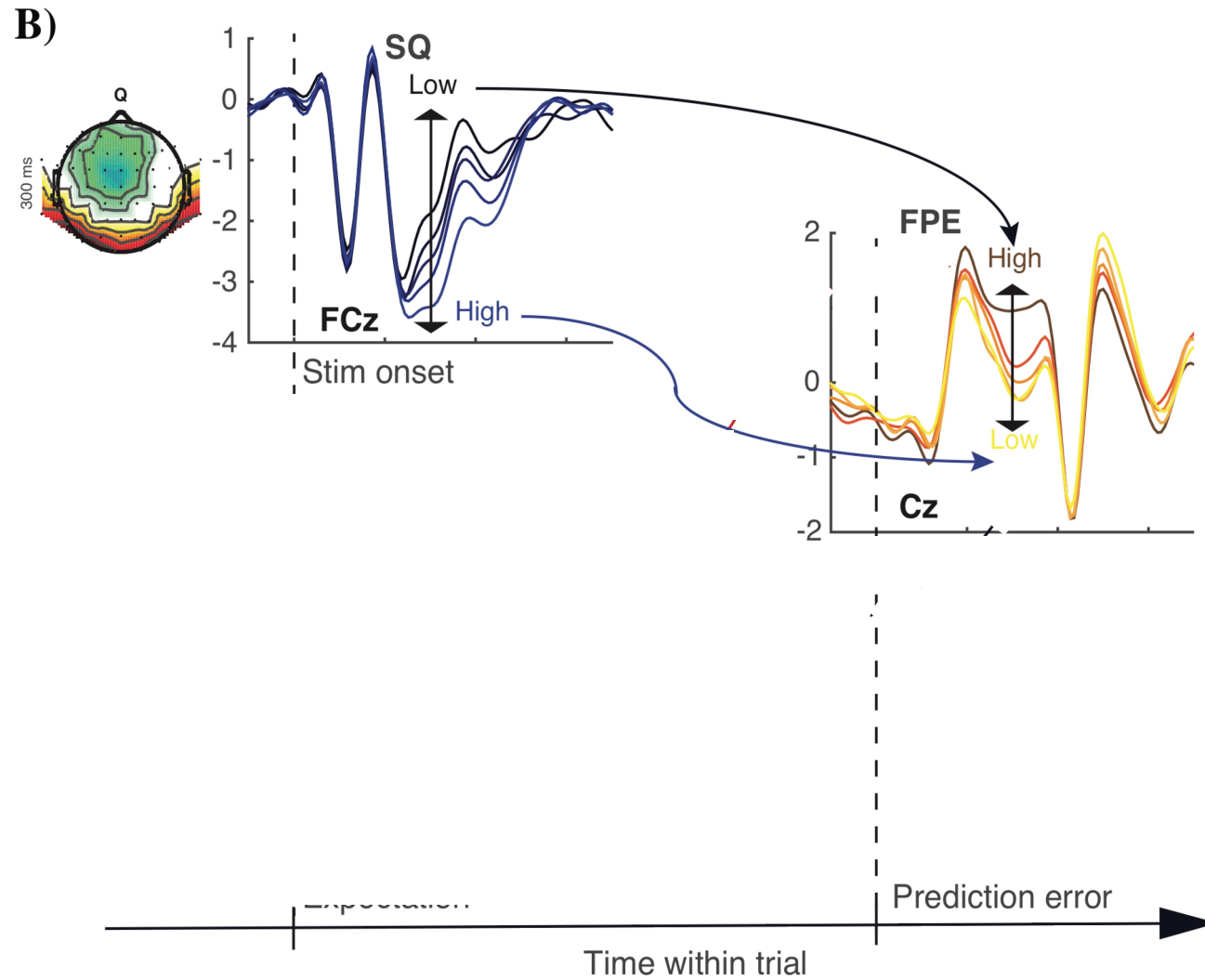
Within-trial dynamics: RL



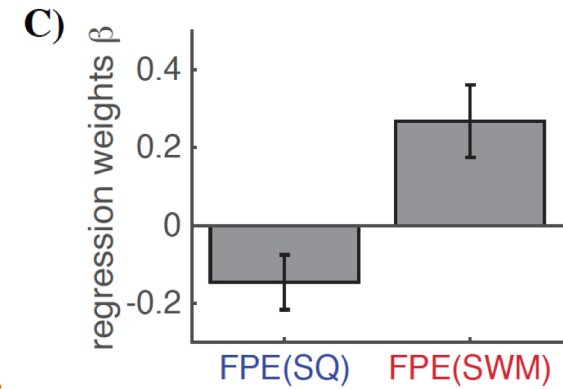
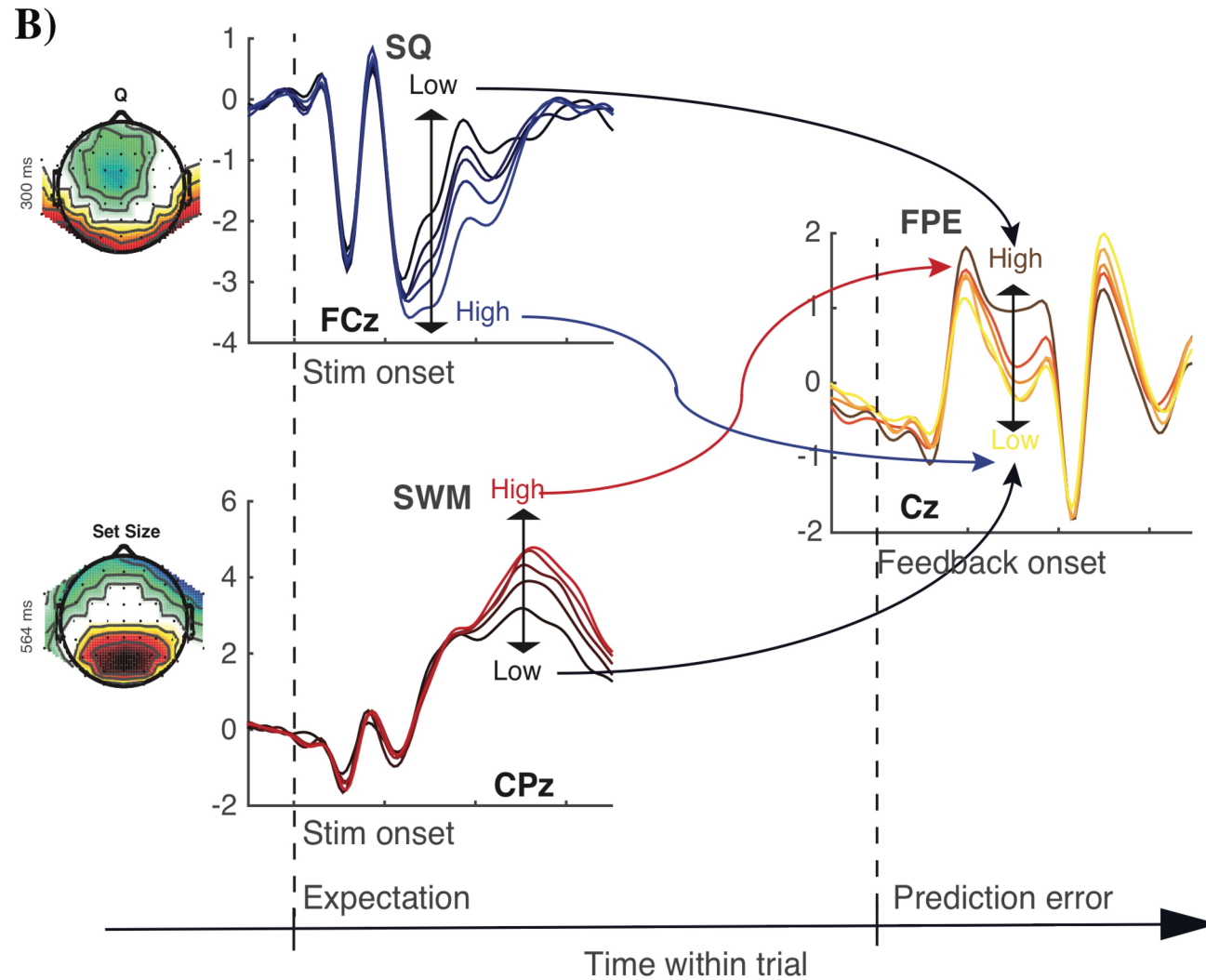
$$FPE = R(t) - SQ(t)$$

Note: reward = 1
In all cases

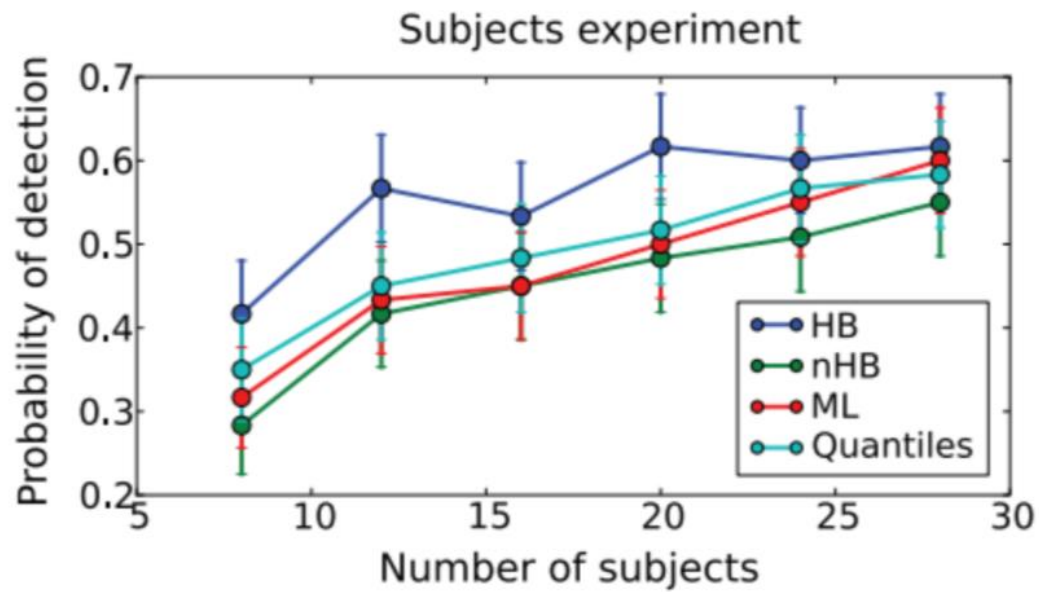
Within-trial dynamics: RL



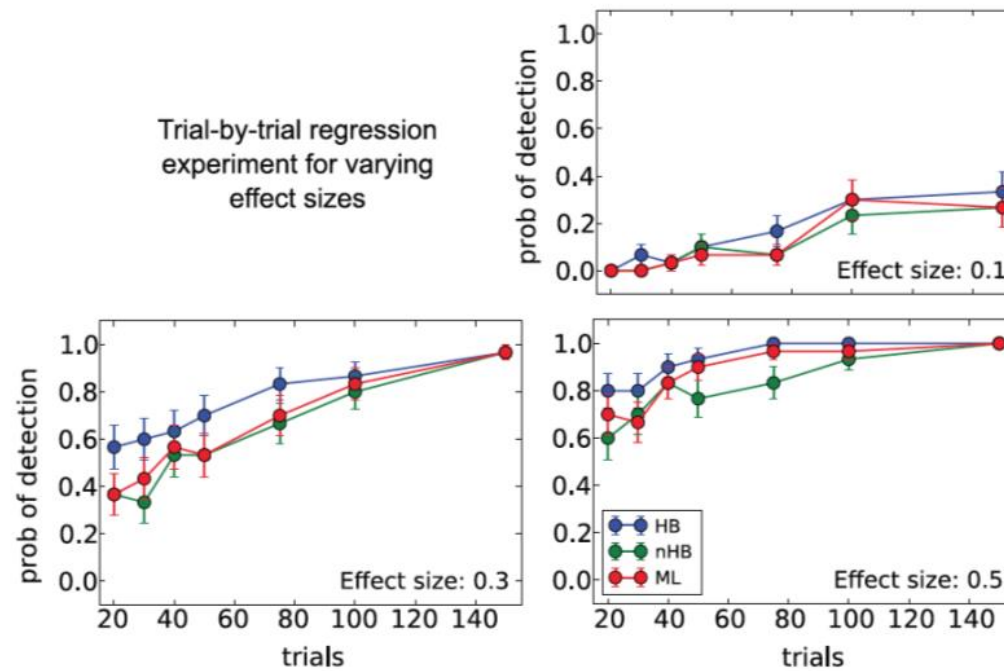
Within-trial dynamics: WM/RL interactions



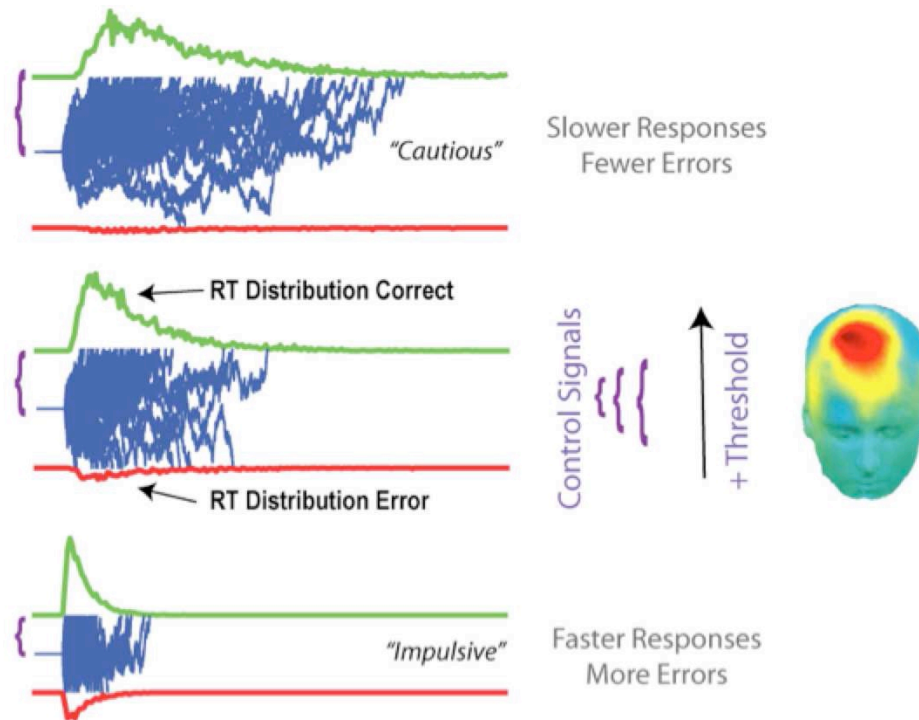
Why use hierarchical bayes: Detection of effect as a function of # Subjects



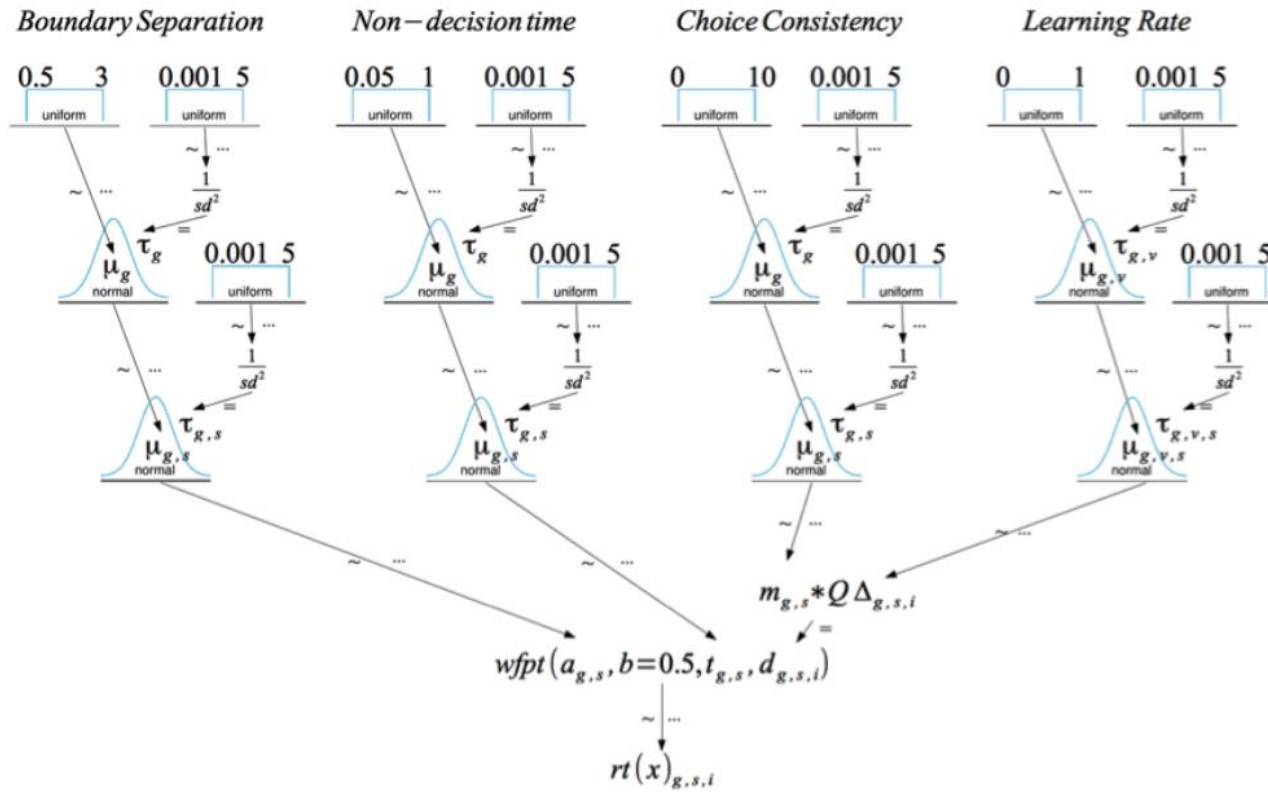
Regression of physio variable onto DDM param: Probability of detecting an effect



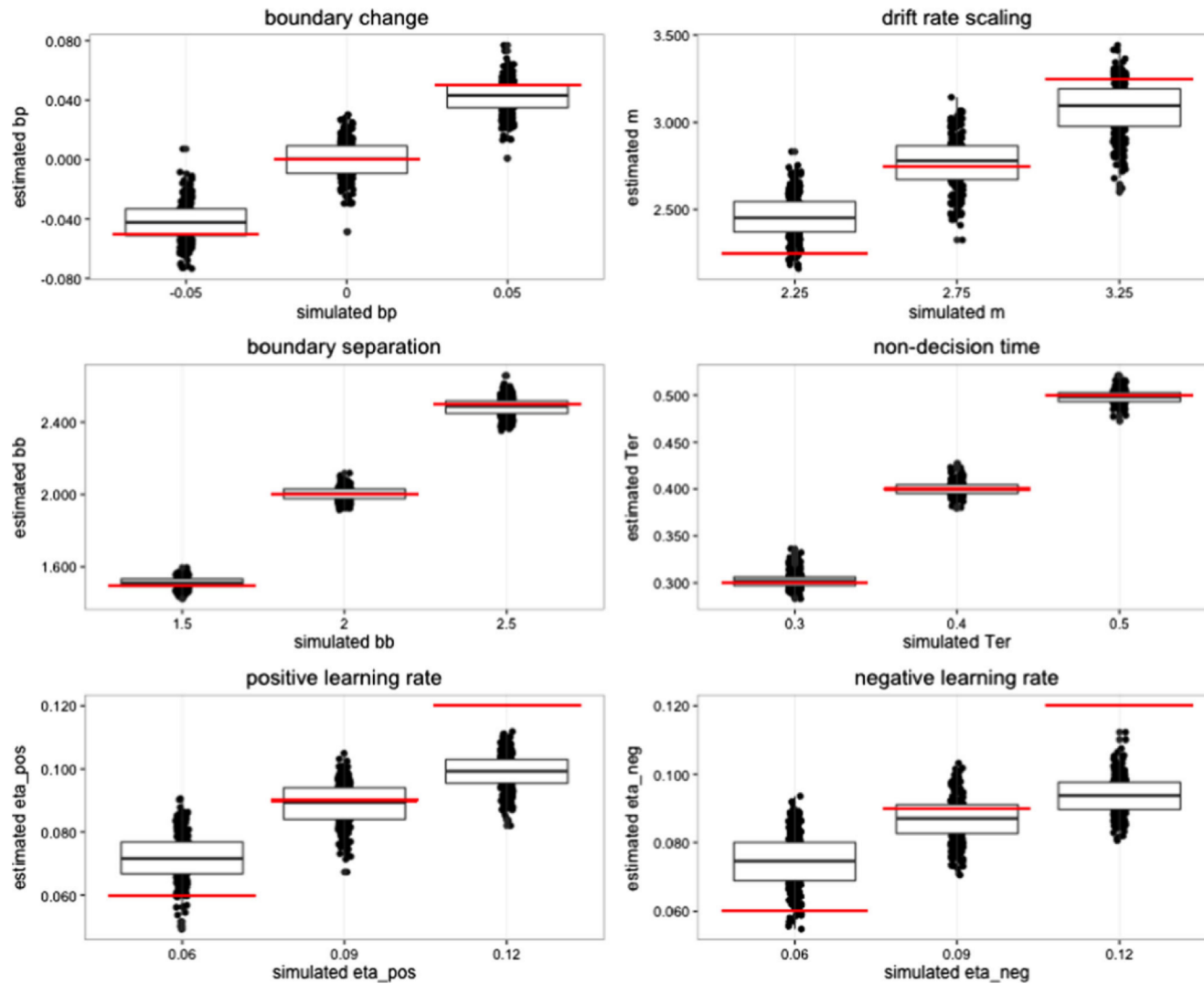
Drift diffusion model as the choice rule in reinforcement learning



Simultaneous Estimation of RL and DDM



Parameter recovery



Fitted parameters, joint distributions ADHD on/off meds

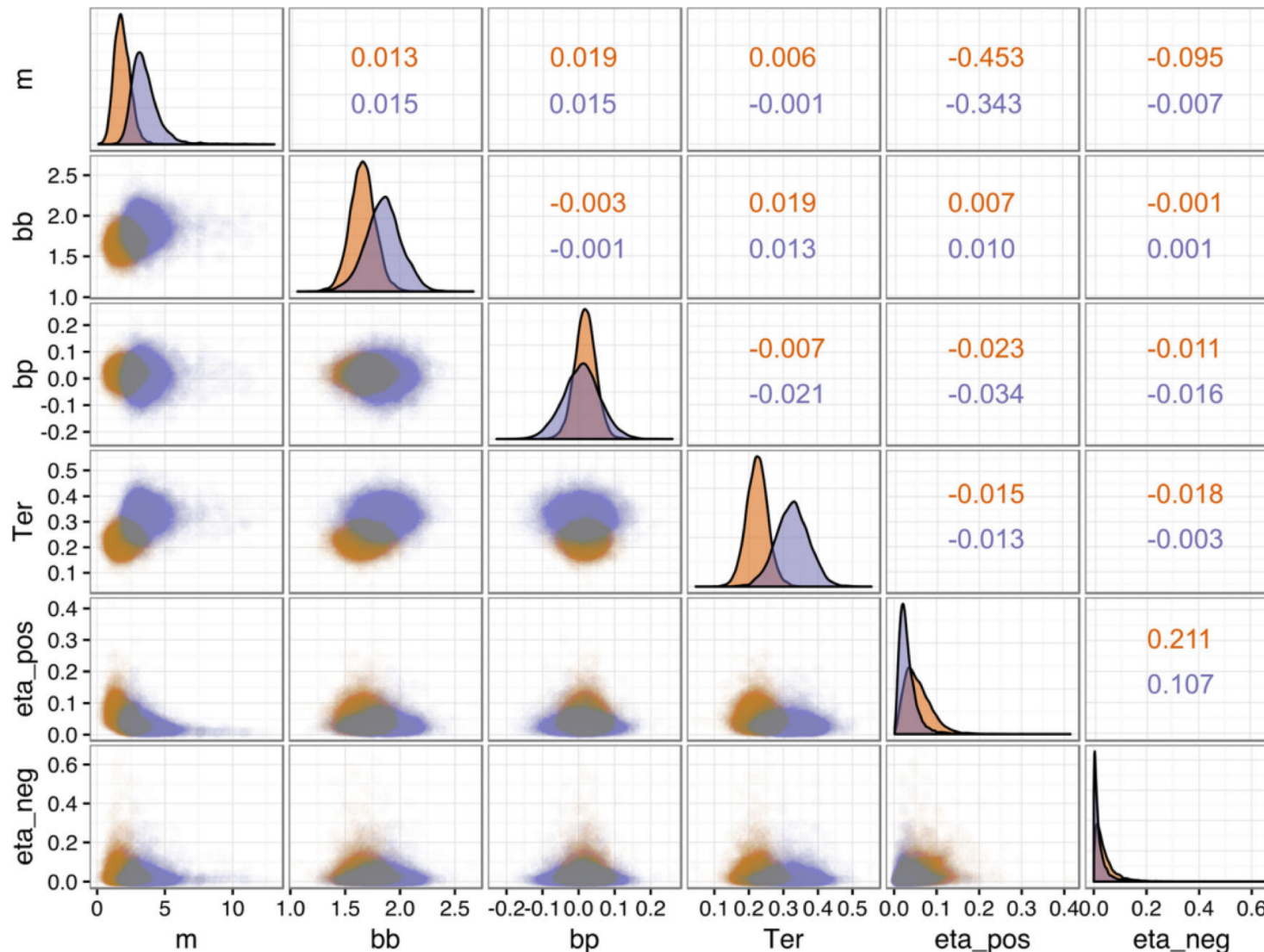
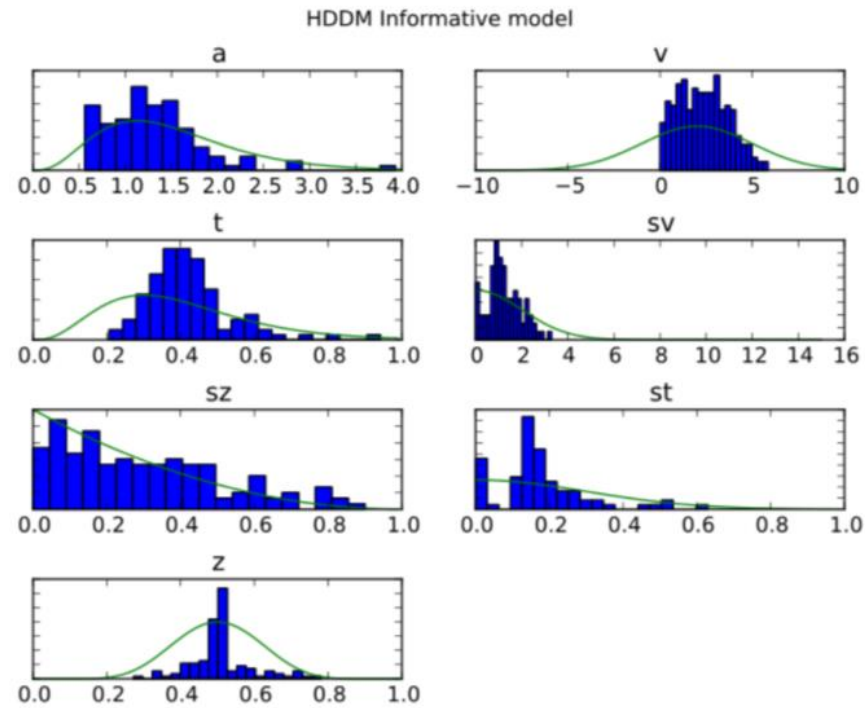


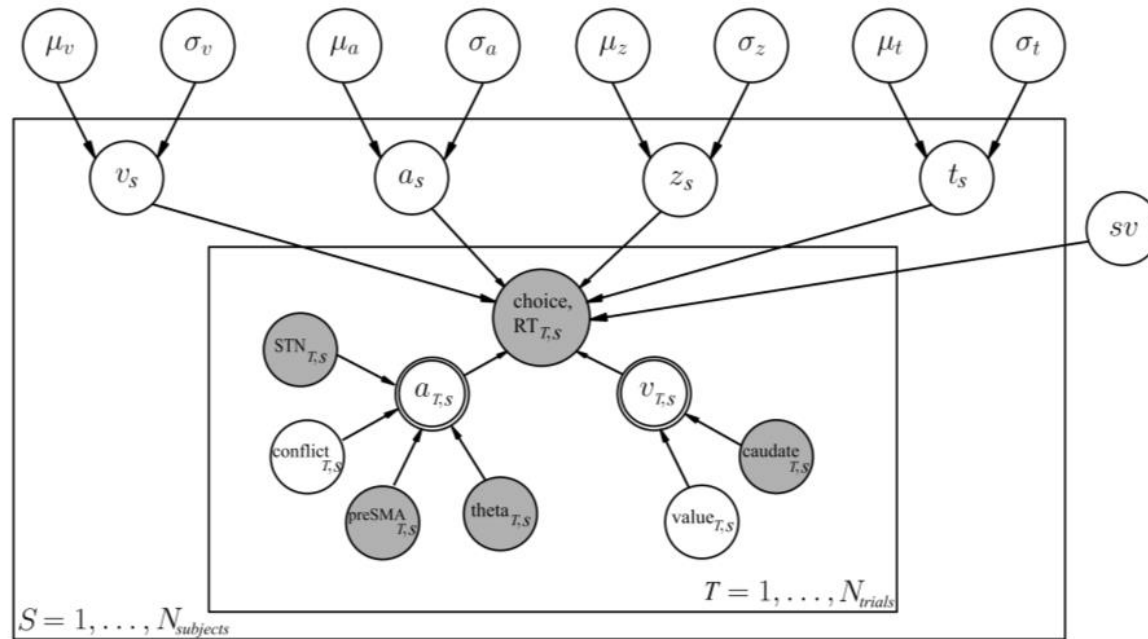
Fig. 2 Scatterplot and density of group parameter estimates from posterior distributions off (red) and on (purple) medication. bb = boundary baseline, bp = boundary power, eta_pos = learning

rate for positive prediction errors (PEs), eta_neg = learning rate for negative PEs, m = drift rate scaling, T_{er} = nondecision time

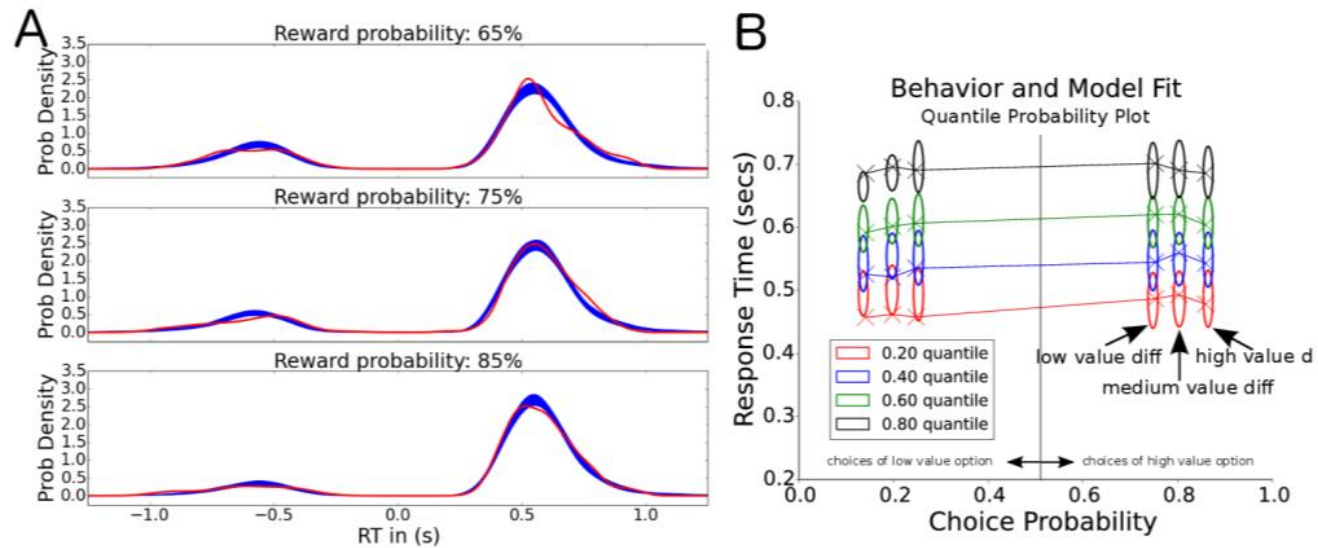
HDDM empirical priors



fMRI and EEG experiment

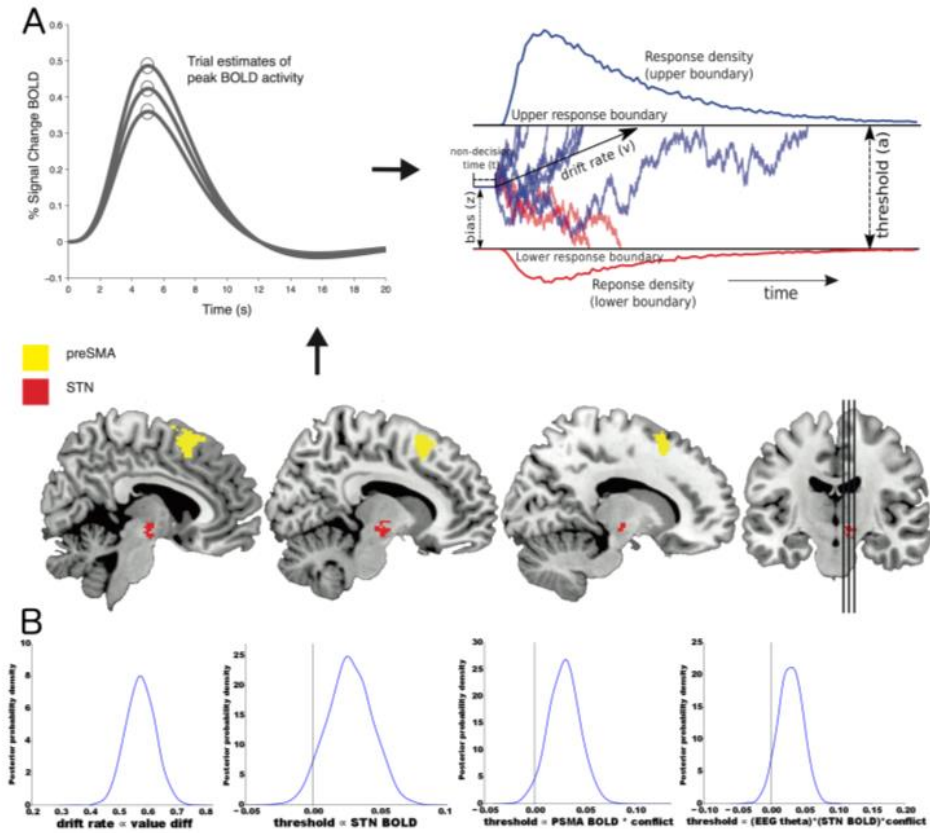


fMRI and EEG experiment: behavior



posterior predictive checks are important!

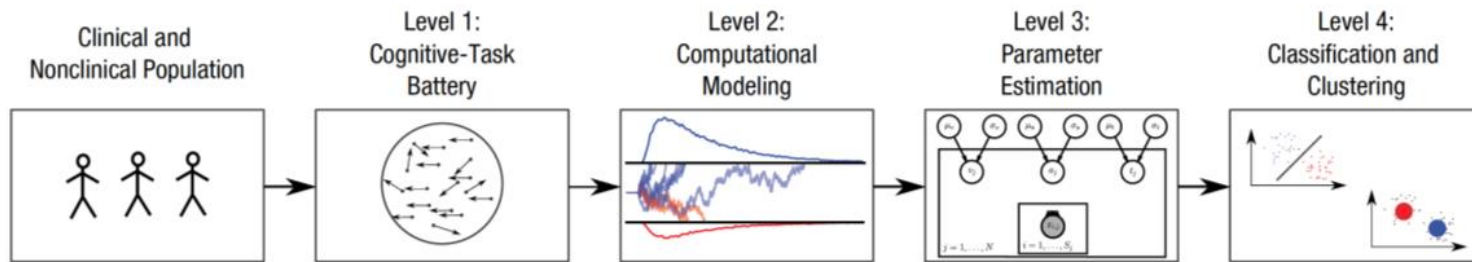
fMRI and EEG experiment



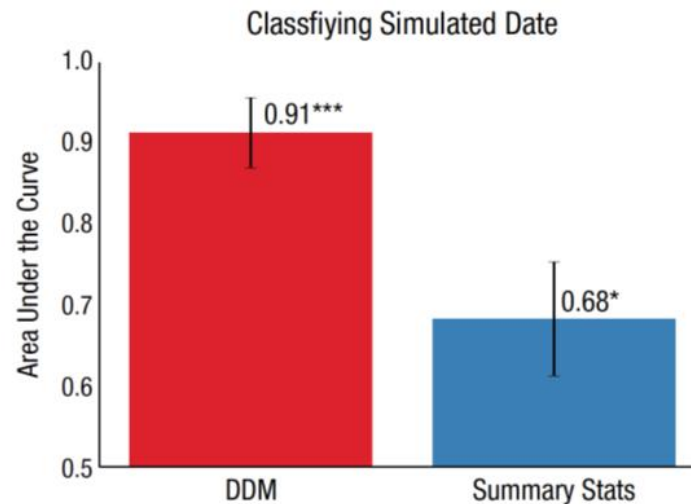
Frank et al 2015

also: STN spikes directly linked to threshold during conflict task, Herz et al 2016

Application to Computational Psychiatry and Neurology

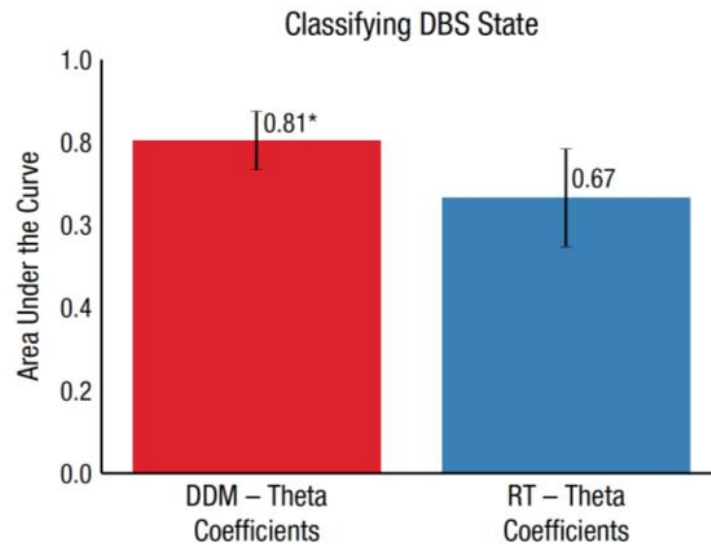


Why use DDM: Simulation experiment and classification of groups



- generated data from DDM with two groups with different parameters
- classification of observed data based on fitted model params or raw behavioral summary statistics

Real data: classification of DBS state



Wiecki, Poland & Frank 2013

Also classifies Huntington's disease before symptom onset! Wiecki, et al., 2016

Linking levels

Linking Across Levels of Computation in Model-Based Cognitive Neuroscience

Michael J. Frank

Reinforcement-Based Decision Making in Corticostriatal
Circuits: Mutual Constraints by Neurocomputational
and Diffusion Models

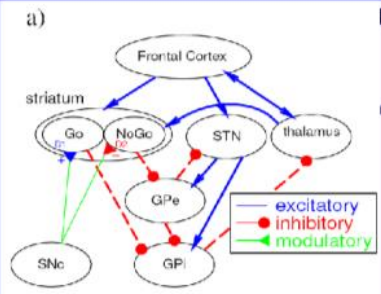
- Strategy to interpret and link across levels of description
- Mutually informative: algorithm informs biological interpretation; biology informs abstraction

Frank, 2015; Collins & Frank, 2013; Ratcliff & Frank, 2012; Franklin & Frank, 2016)

Multiple levels of analysis

Basal Ganglia

- Role of the basal ganglia in action selection and initiation: both motor actions & "cognitive actions"
- Convoluted architecture is there to ensure that actions are not triggered accidentally

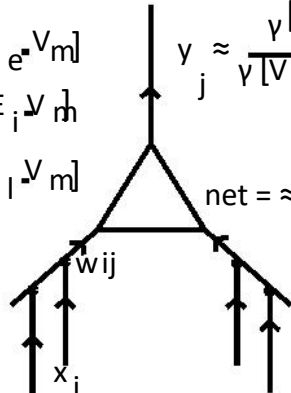


a)

$$cV_m = g_e g_e [E_e V_m] + g_i g_i [E_i V_m] + g_l g_l [E_l V_m] + \dots$$

$$y_j \approx \frac{\gamma [V_m^- \Theta]}{\gamma [V_m^- \Theta] + \beta}$$

net = $\sum_i g_i x_i w_{ij} > \beta$



$$\Delta w_{ij} \approx (x_i^p y_j^p) - (x_i^t y_j^t)$$

Modeling the BG model

