University of Zurich[UZH]

ETH Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

UCL

Computational models for imaging analyses

Zurich SPM Course

Christoph Mathys

What the brain is about

- What do our imaging methods measure?
    - Brain activity.
- But when does the brain become active?
    - When predictions (or their precision) have to be adjusted.
- So where do the brain's predictions come from?
    - From a model.

What does this mean for neuroimaging?

If brain activity reflects model updating, we need to understand **what model** is updated **in what way** to make sense of brain activity.

The Bayesian brain and predictive coding

Model-based prediction updating is described by Bayes' theorem.

$\Longrightarrow$ the Bayesian brain

Hermann von Helmholtz

This can be implemented by **predictive coding.**

Advantages of model-based imaging

Model-based imaging permits us

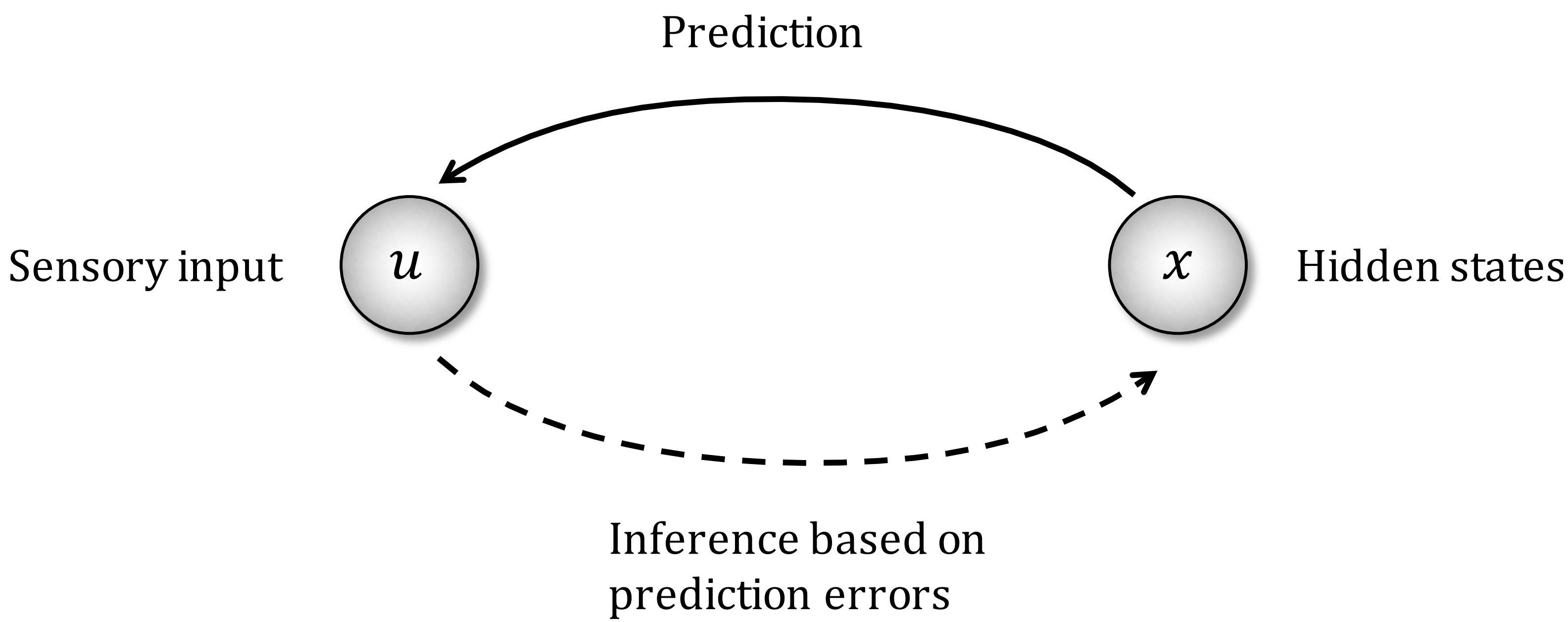
- to **infer** the computational (predictive) mechanisms underlying neuronal activity.

- to **localize** such mechanisms.

- to **compare** different **models.**

How to build a model

Fundamental ingredients:



Example of a simple learning model

Rescorla-Wagner learning:

$$\mu^{(k)} = \mu^{(k-1)} + \alpha\left(u^{(k)} - \mu^{(k-1)}\right)$$

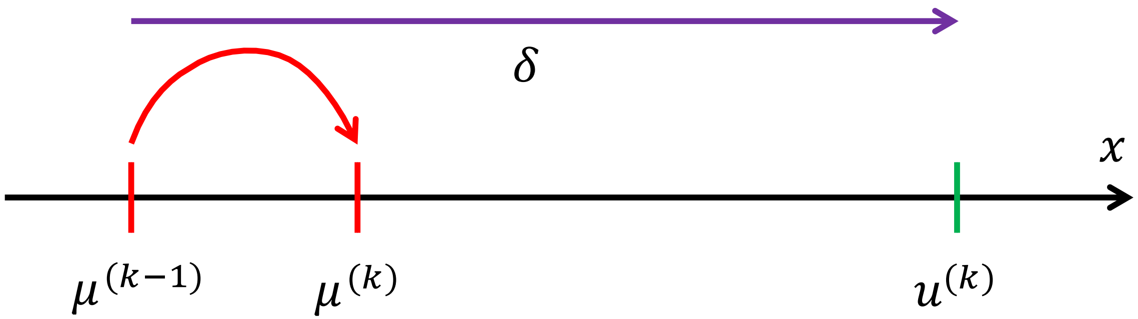
Learning rate: α

Previous value (prediction): $\mu^{(k-1)}$

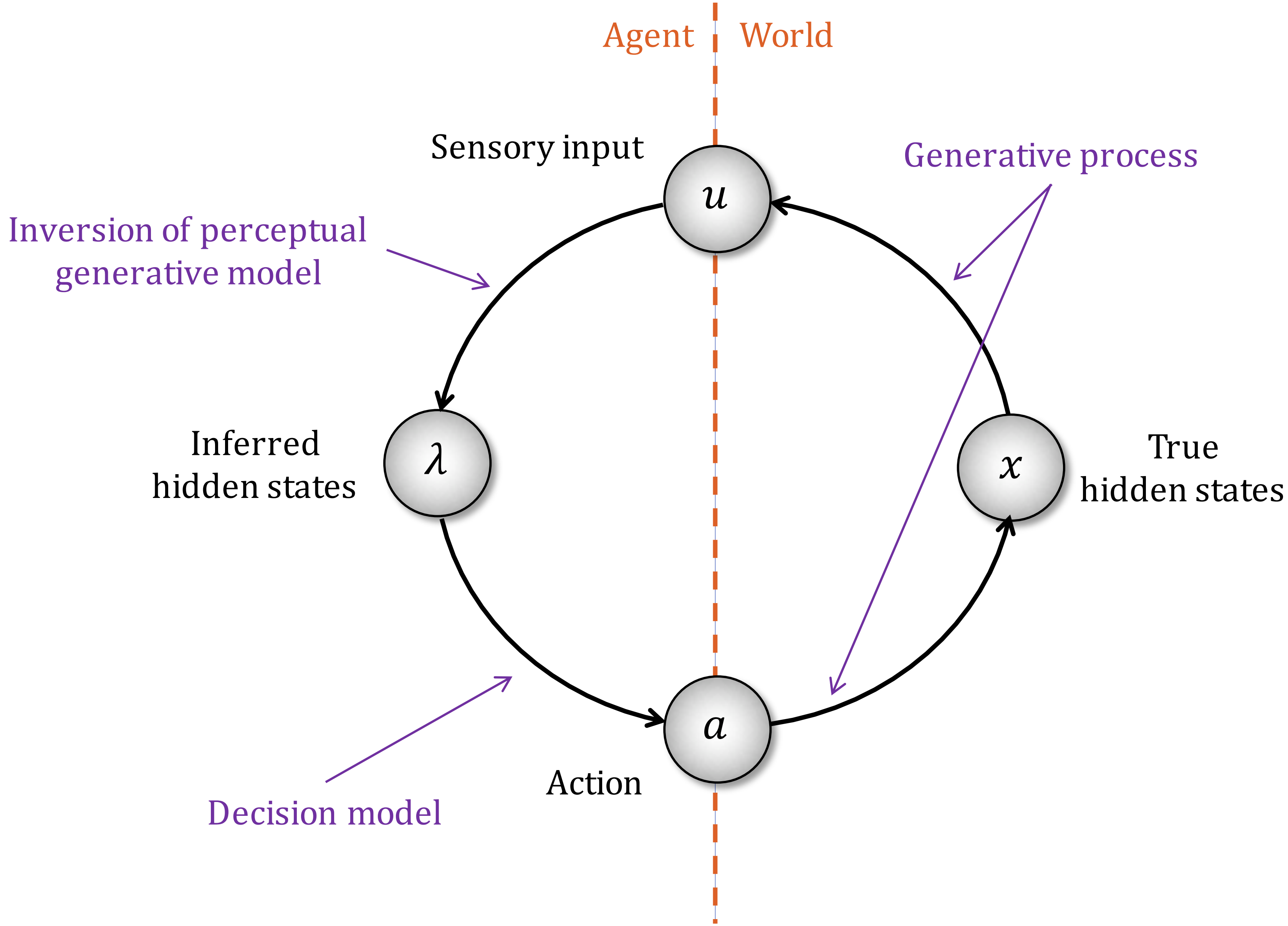
Prediction error (δ): $u^{(k)} - \mu^{(k-1)}$

Inferred value of x: $\mu^{(k)}$, $\mu^{(k-1)}$

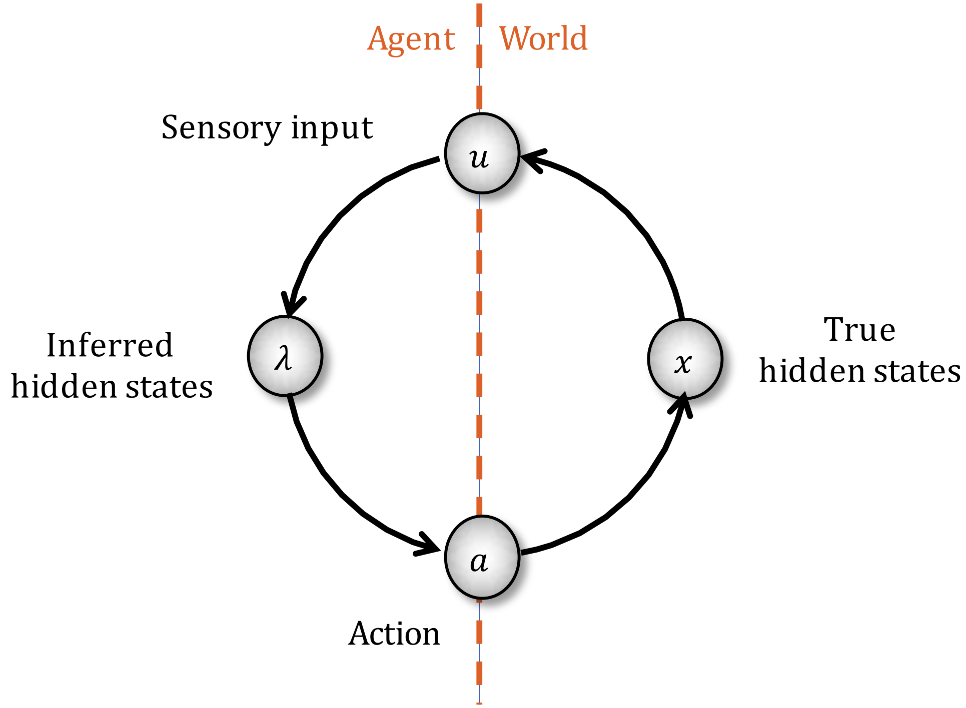
New input: $u^{(k)}$



From perception to action



From perception to action



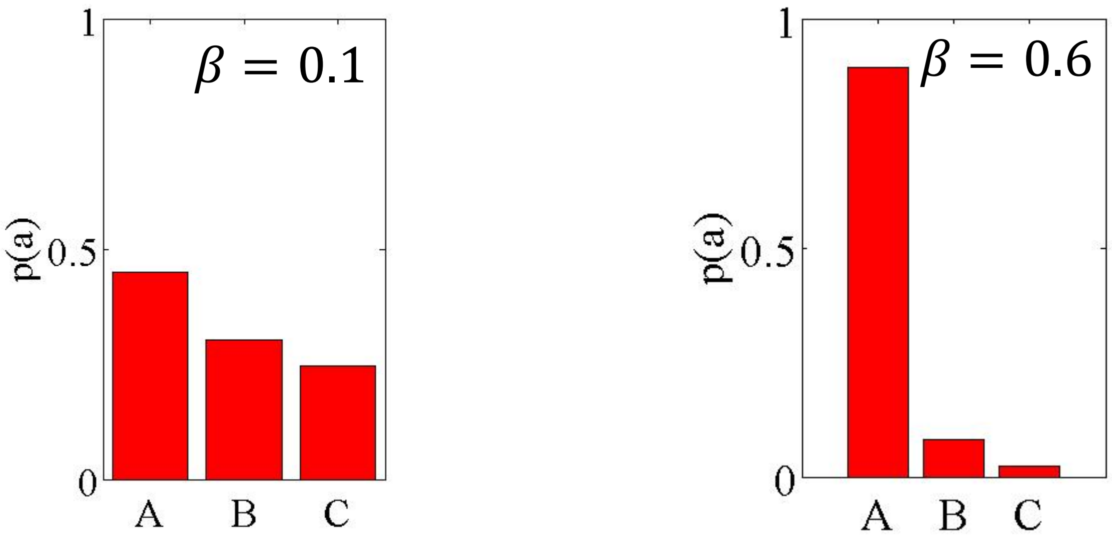
- In behavioral tasks, we observe **actions** (a).
- How do we use them to infer **beliefs** (λ)?
- We invert (i.e., estimate) a **decision model.**

Example of a simple decision model

- Say 3 options A, B, and C have values $v_A = 8$, $v_B = 4$, and $v_C = 2$.
- Then we can translate these values into action probabilities via a «softmax» function:

$$p(a = A) = \frac{e^{\beta v_A}}{e^{\beta v_A} + e^{\beta v_B} + e^{\beta v_C}}$$

- The parameter β determines the sensitivity to value differences



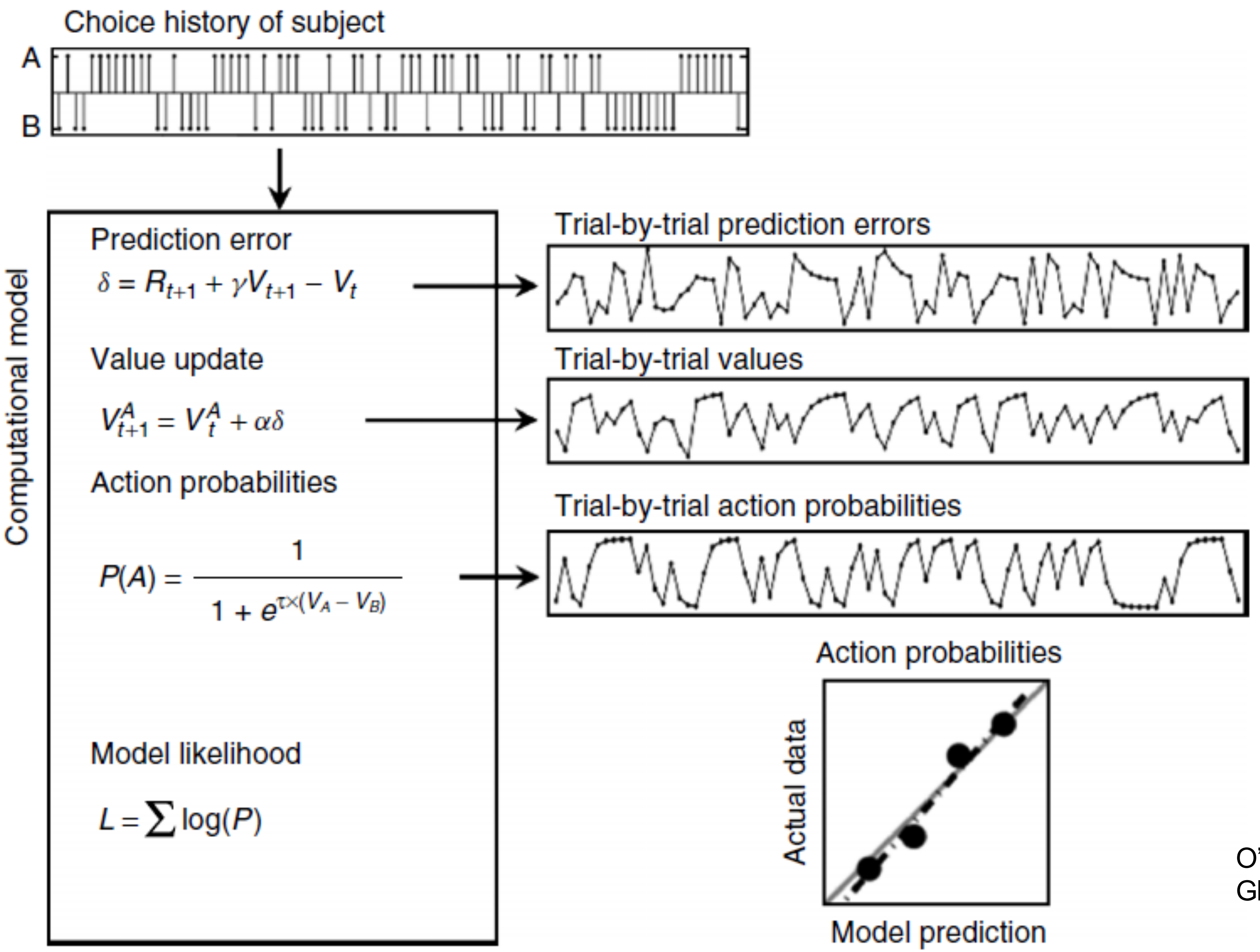
All the necessary ingredients

- Perceptual model (updates based on prediction errors)

- Value function (inferred state -> action value)

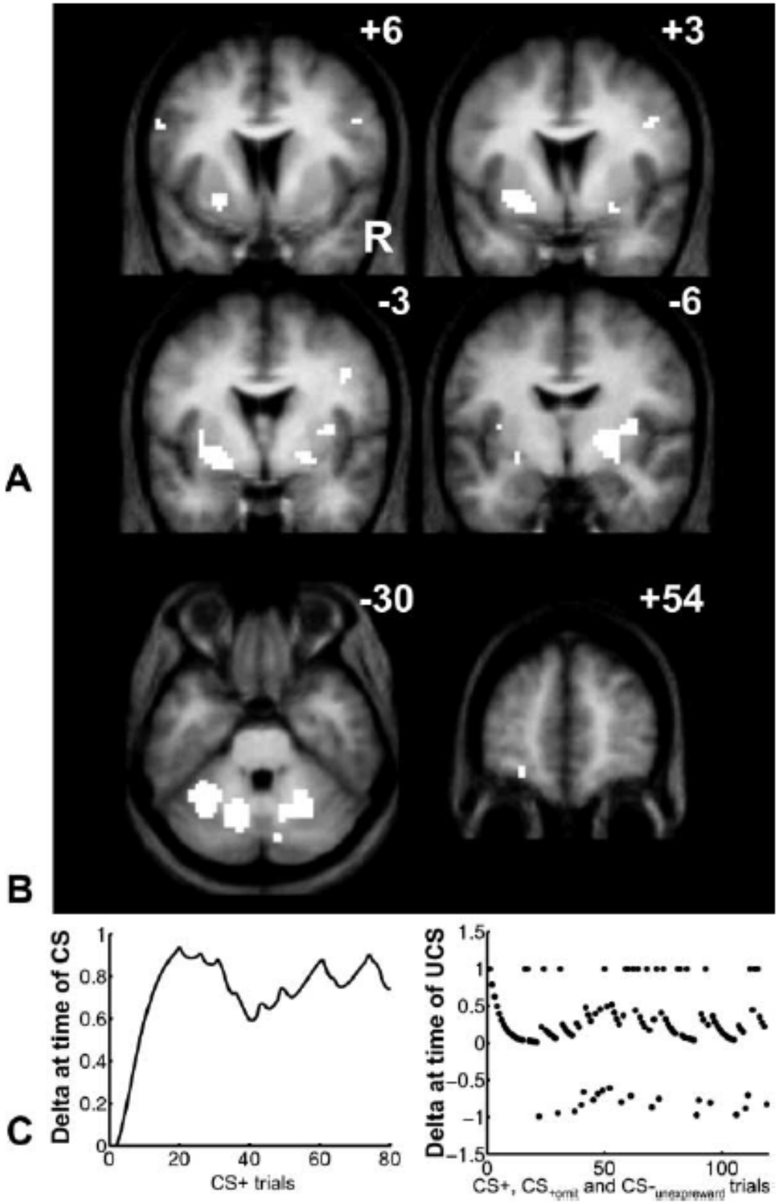
- Decision model (value -> action probability)

Reinforcement learning example (O'Doherty et al., 2003)



O'Doherty et al. (2003), Gläscher et al. (2010)

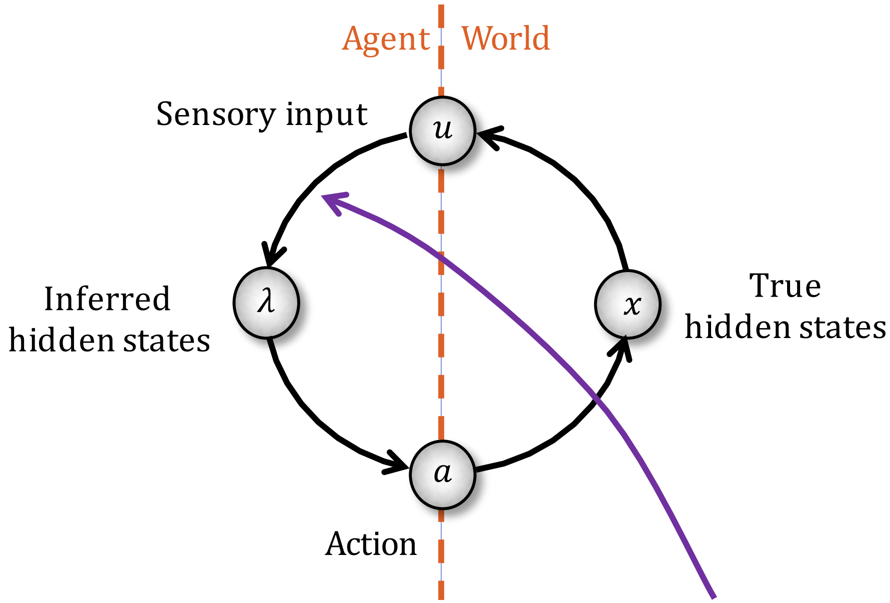
Reinforcement learning example



Significant effects of prediction error with fixed learning rate

O'Doherty et al. (2003)

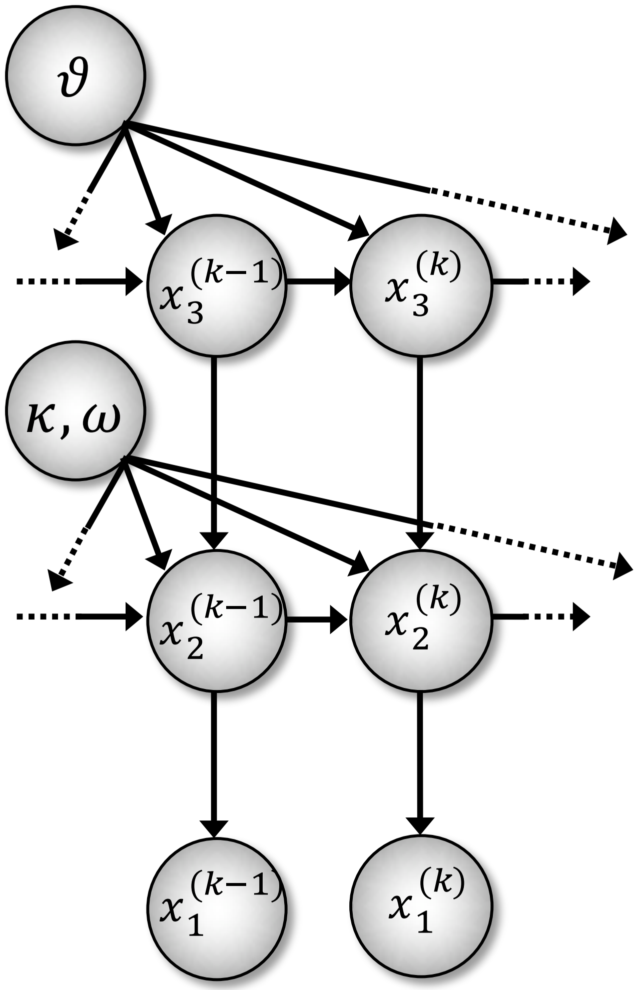
Bayesian models for the Bayesian brain

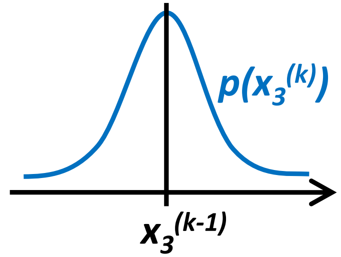
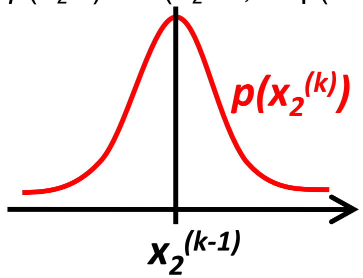
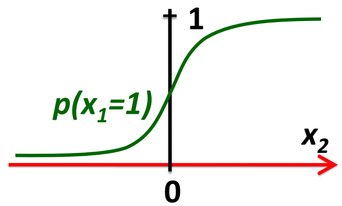


$$\underbrace{p(u|x,\vartheta)}_{\text{likelihood}} \cdot \underbrace{p(x,\vartheta)}_{\text{prior}} \propto \underbrace{p(x,\vartheta|u)}_{\text{posterior}}$$

- Includes **uncertainty** about hidden states.
- I.e., beliefs have **precisions.**
- But how can we make them computationally tractable?

The hierarchical Gaussian filter (HGF): a computationally tractable model for individual learning under uncertainty



State of the world	Model	
Log-volatility x_3 of tendency	Gaussian random walk with constant step size ϑ	$p(x_3^{(k)}) \sim N(x_3^{(k-1)}, \vartheta)$ 
Tendency x_2 towards category "1"	Gaussian random walk with step size $\exp(\kappa x_3+\omega)$	$p(x_2^{(k)}) \sim N(x_2^{(k-1)}, \exp(\kappa x_3+\omega))$ 
Stimulus category x_1 ("0" or "1")	Sigmoid transformation of x_2	$p(x_1=1) = s(x_2)$ $p(x_1=0) = 1-s(x_2)$ 

HGF: variational inversion and update equations

- Inversion proceeds by introducing a mean field approximation and fitting quadratic approximations to the resulting variational energies (Mathys et al., 2011).
- This leads to **simple one-step update equations** for the sufficient statistics (mean and precision) of the approximate Gaussian posteriors of the states x_i.
- The updates of the means have the same structure as value updates in Rescorla-Wagner learning:

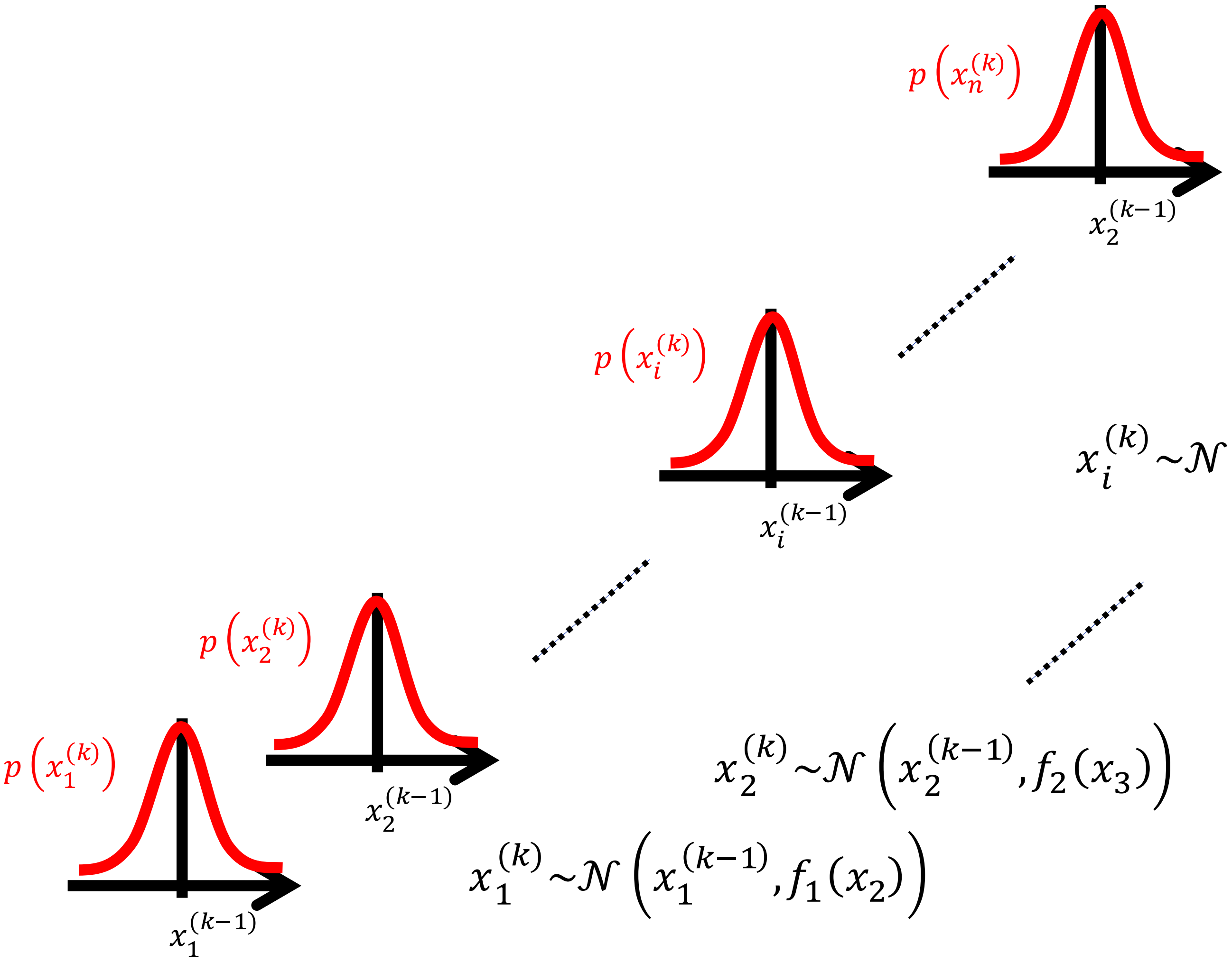
$$\Delta\mu_i \propto \frac{\hat{\pi}_{i-1}}{\pi_i} \delta_{i-1}$$

Prediction error

Precisions determine
learning rate

- Furthermore, the updates are **precision-weighted prediction errors**.

The hierarchical Gaussian filter (HGF)



$$x_n^{(k)} \sim \mathcal{N}\left(x_n^{(k-1)}, \vartheta\right)$$

$$x_i^{(k)} \sim \mathcal{N}\left(x_i^{(k-1)}, f_i(x_{i+1})\right)$$

$$x_2^{(k)} \sim \mathcal{N}\left(x_2^{(k-1)}, f_2(x_3)\right)$$

$$x_1^{(k)} \sim \mathcal{N}\left(x_1^{(k-1)}, f_1(x_2)\right)$$

The hierarchical Gaussian filter (HGF)

At the outcome level (i.e., at the very bottom of the hierarchy), we have

$$u^{(k)} \sim \mathcal{N}\left(x_1^{(k)}, \hat{\pi}_u^{-1}\right)$$

This gives us the following update for our belief on x_1 (our quantity of interest):

$$\pi_1^{(k)} = \hat{\pi}_1^{(k)} + \hat{\pi}_u$$

$$\mu_1^{(k)} = \mu_1^{(k-1)} + \frac{\hat{\pi}_u}{\pi_1^{(k)}}\left(u^{(k)} - \mu_1^{(k-1)}\right)$$

Updates are precision-weighted prediction errors – with a learning rate that is responsive to all kinds of uncertainty, including environmental (unexpected) uncertainty.

The learning rate in the HGF

Unpacking the learning rate, we see:

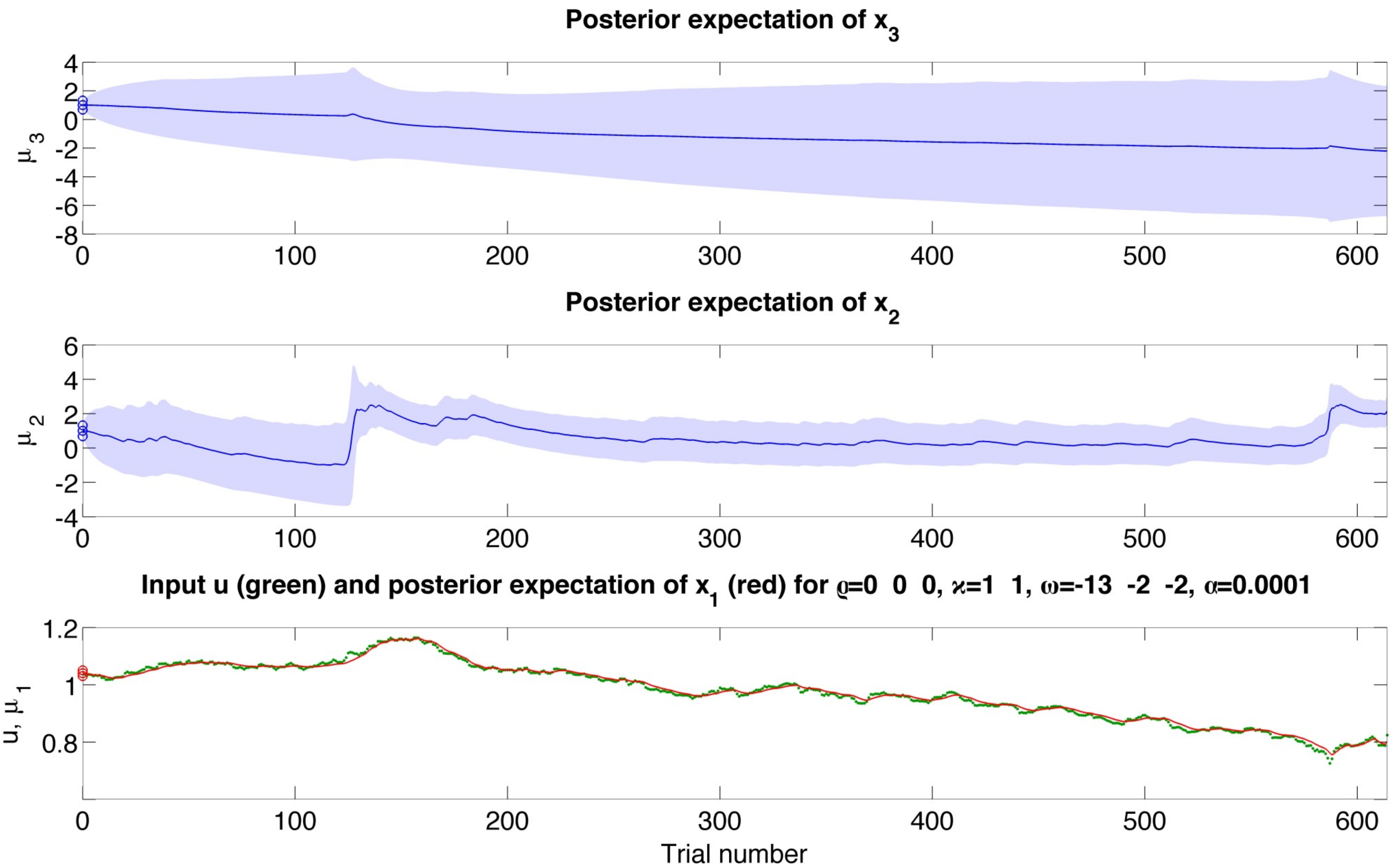
outcome uncertainty

$$\frac{\hat{\pi}_u}{\pi_1^{(k)}} = \frac{\hat{\pi}_u}{\hat{\pi}_1^{(k)} + \hat{\pi}_u} = \frac{\hat{\pi}_u}{\dfrac{1}{\sigma_1^{(k-1)} + \exp\left(\kappa_1 \mu_2^{(k-1)} + \omega_1\right)} + \hat{\pi}_u}$$

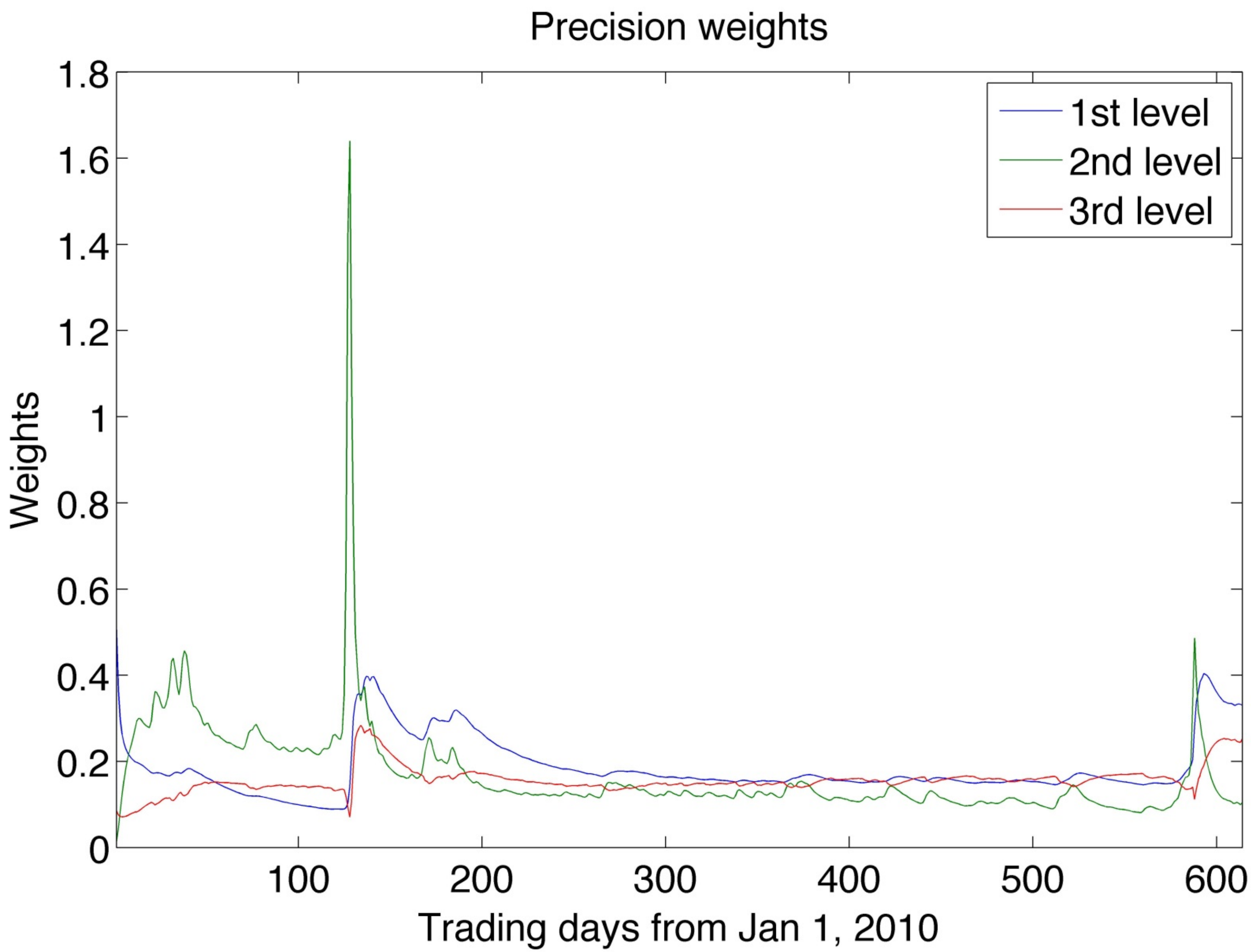
informational uncertainty

environmental uncertainty

3-level HGF for continuous observations



3-level HGF for continuous observations



VAPEs and VOPEs

The updates of the belief on x_1 are driven by **value prediction errors** (VAPEs)

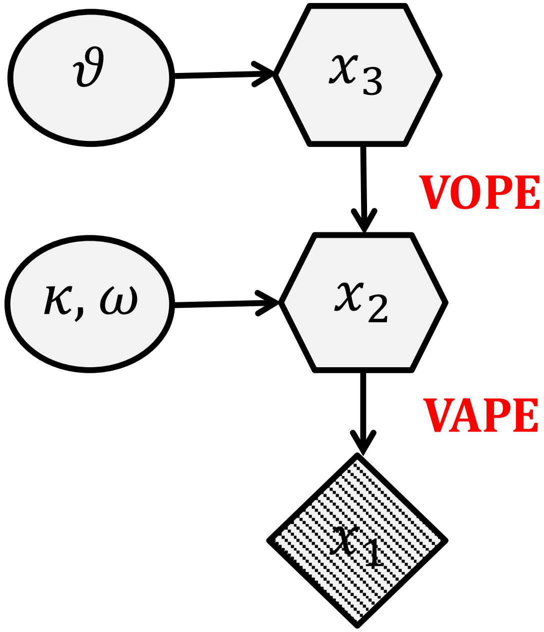
$$\mu_1^{(k)} = \mu_1^{(k-1)} + \frac{\hat{\pi}_u}{\pi_1^{(k)}} \left(u^{(k)} - \mu_1^{(k-1)} \right), \quad \textbf{VAPE}$$

while the x_2-updates are driven by **volatility prediction errors** (VOPEs)

$$\mu_2^{(k)} = \mu_2^{(k-1)} + \frac{1}{2} \kappa_1 \, v_1^{(k)} \frac{\hat{\pi}_1^{(k)}}{\pi_2^{(k)}} \delta_1^{(k)} \quad \textbf{VOPE}$$

$$\delta_1^{(k)} \stackrel{\text{def}}{=} \frac{\sigma_1^{(k)} + \left(\mu_1^{(k)} - \mu_1^{(k-1)} \right)^2}{\sigma_1^{(k-1)} + \exp\left(\kappa_1 \mu_2^{(k-1)} + \omega_1 \right)} - 1$$

3-level HGF for binary observations



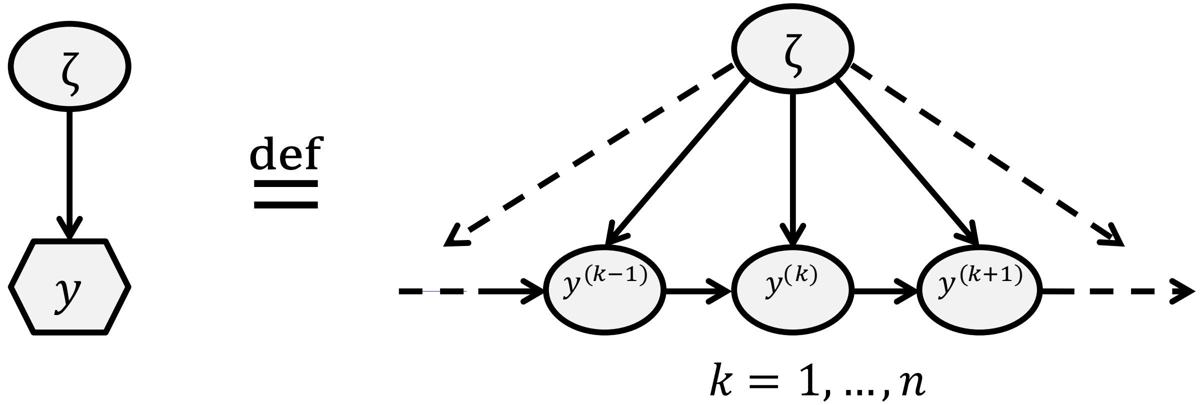
$$x_3^{(k)} \sim \mathcal{N}\left(x_3^{(k-1)}, \vartheta\right)$$

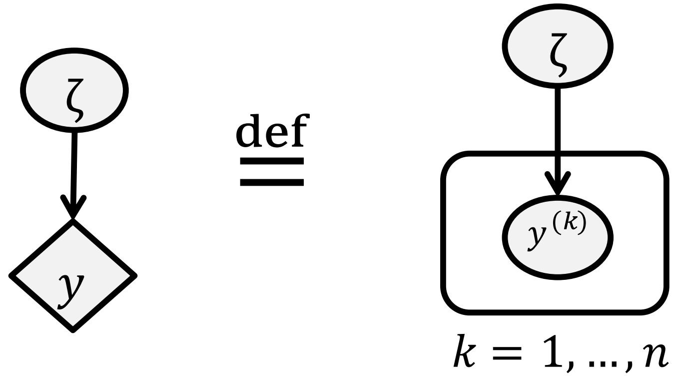
$$x_2^{(k)} \sim \mathcal{N}\left(x_2^{(k-1)}, \exp\left(\kappa x_3^{(k)} + \omega\right)\right)$$

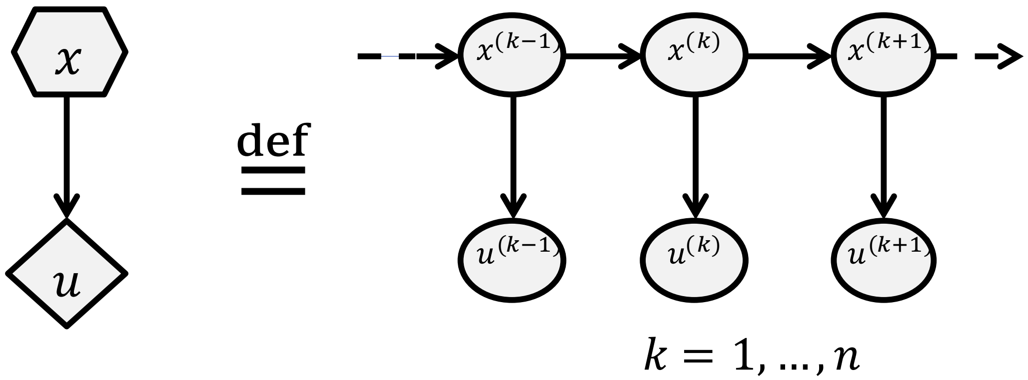
$$x_1^{(k)} \sim \mathrm{Bern}\left(x_2^{(k)}\right)$$

Mathys et al., 2011; Iglesias et al., 2013; Vossel et al., 2014a; Hauser et al., 2014; Diaconescu et al., 2014; Vossel et al., 2014b; Vossel et al., 2015...

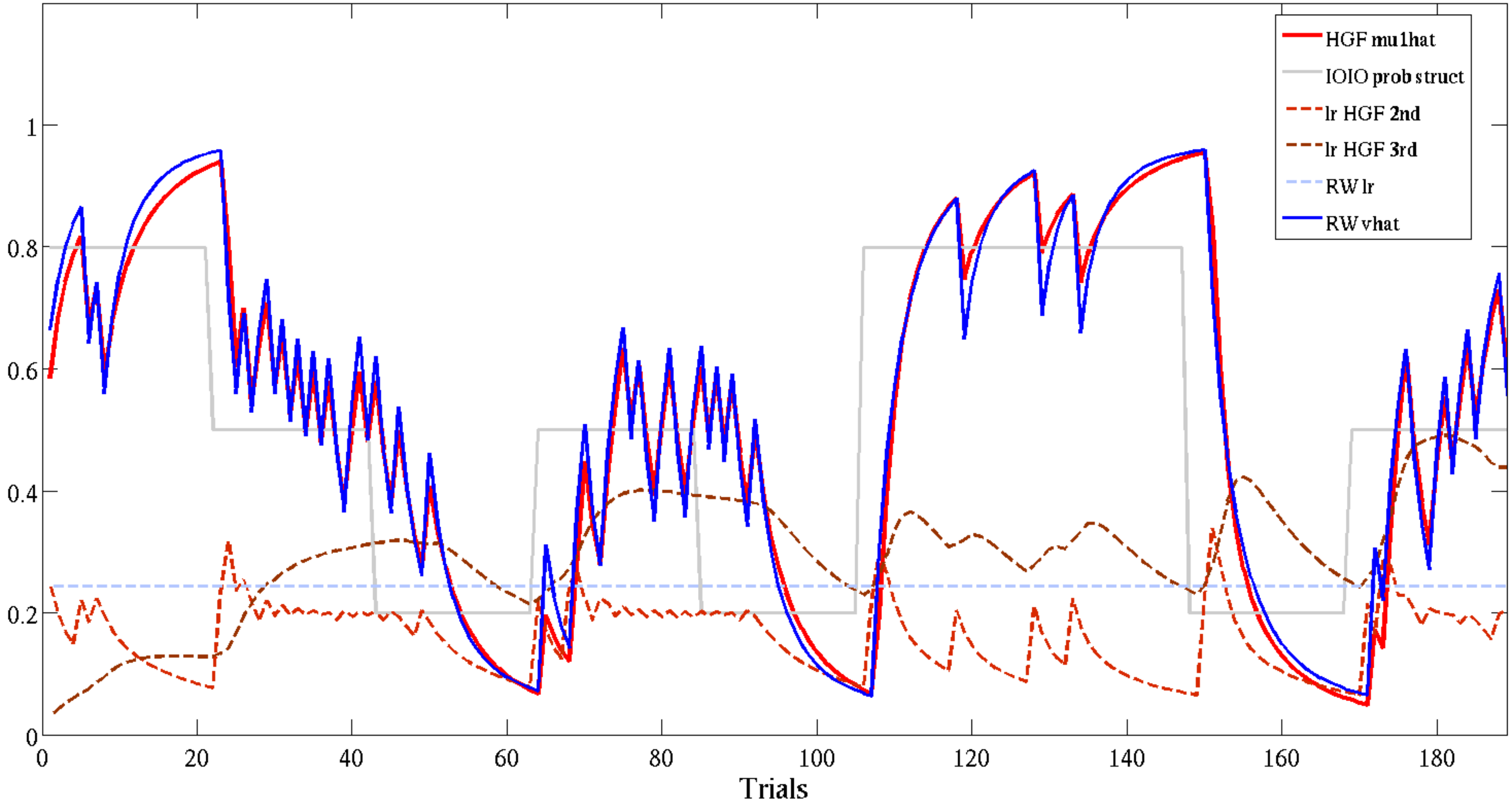
Notation


ζ
y
def
ζ
y(k−1)
y(k)
y(k+1)
k = 1, …, n


ζ
y
def
ζ
y(k)
k = 1, …, n

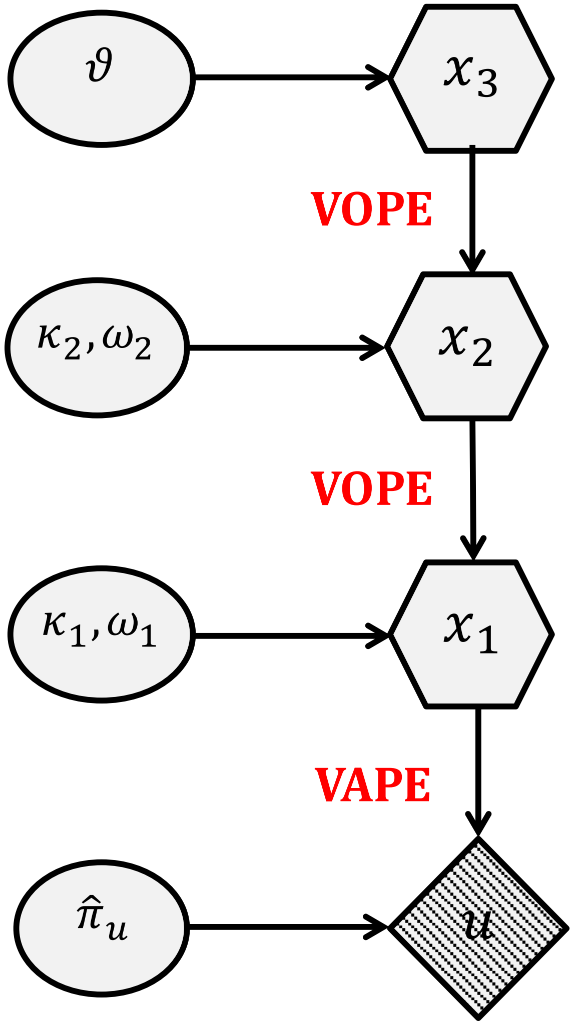

x
u
def
x(k−1)
x(k)
x(k+1)
u(k−1)
u(k)
u(k+1)
k = 1, …, n

The learning rate in the HGF



Andreea Diaconescu

3-level HGF for continuous observations



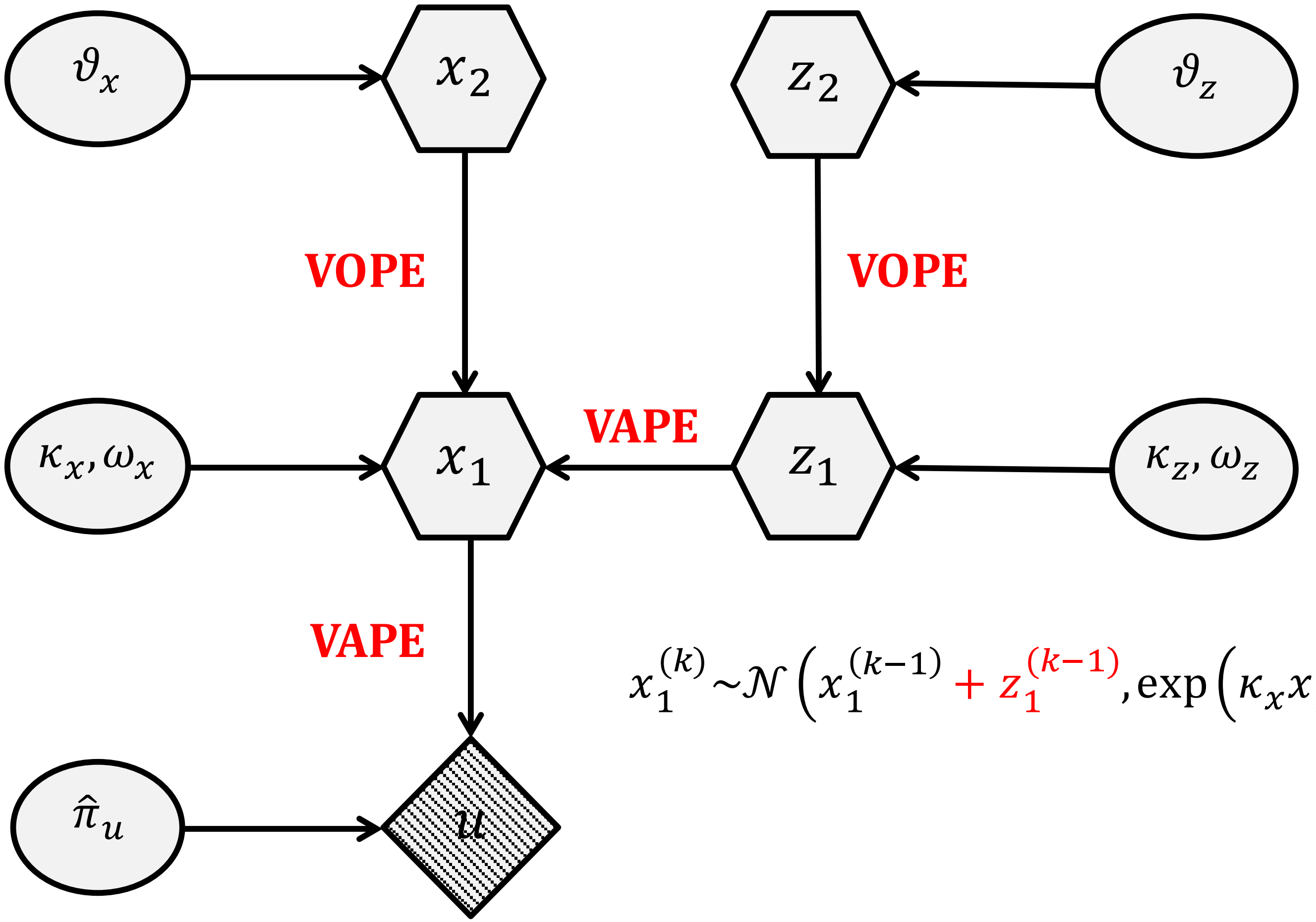
$$x_3^{(k)} \sim \mathcal{N}\left(x_3^{(k-1)}, \vartheta\right)$$

$$x_2^{(k)} \sim \mathcal{N}\left(x_2^{(k-1)}, \exp\left(\kappa_2 x_3^{(k)} + \omega_2\right)\right)$$

$$x_1^{(k)} \sim \mathcal{N}\left(x_1^{(k-1)}, \exp\left(\kappa_1 x_2^{(k)} + \omega_1\right)\right)$$

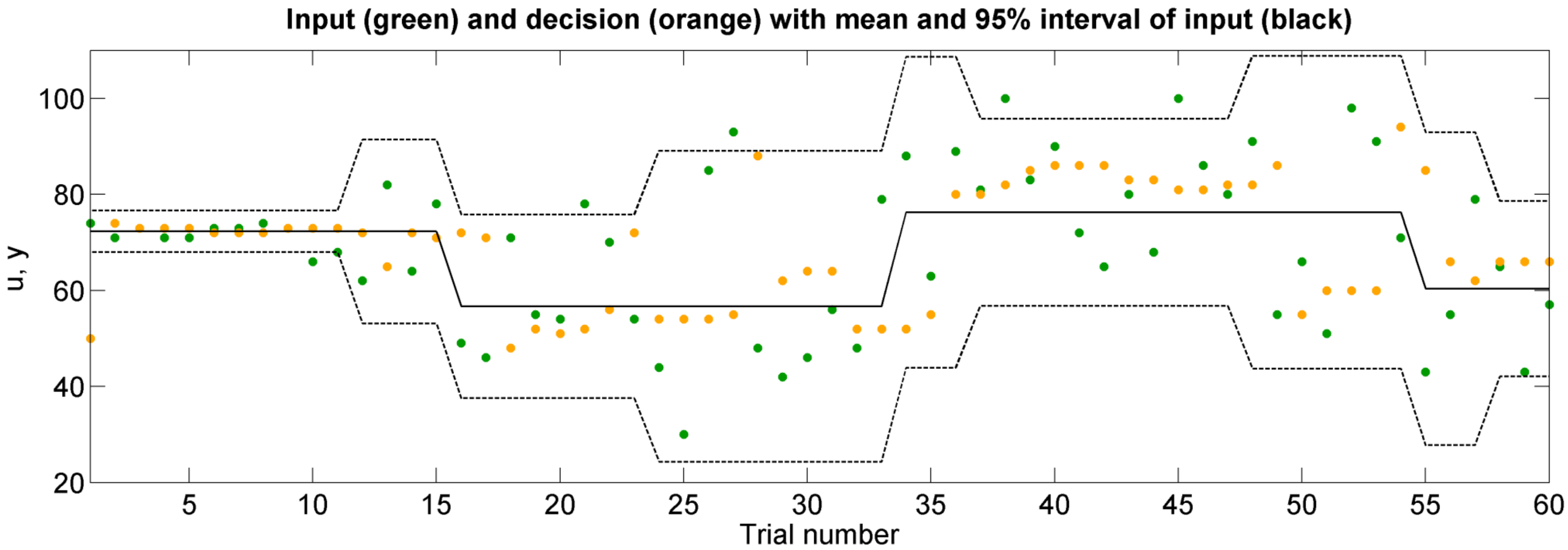
$$u^{(k)} \sim \mathcal{N}\left(x_1^{(k)}, \hat{\pi}_u^{-1}\right)$$

Variable drift



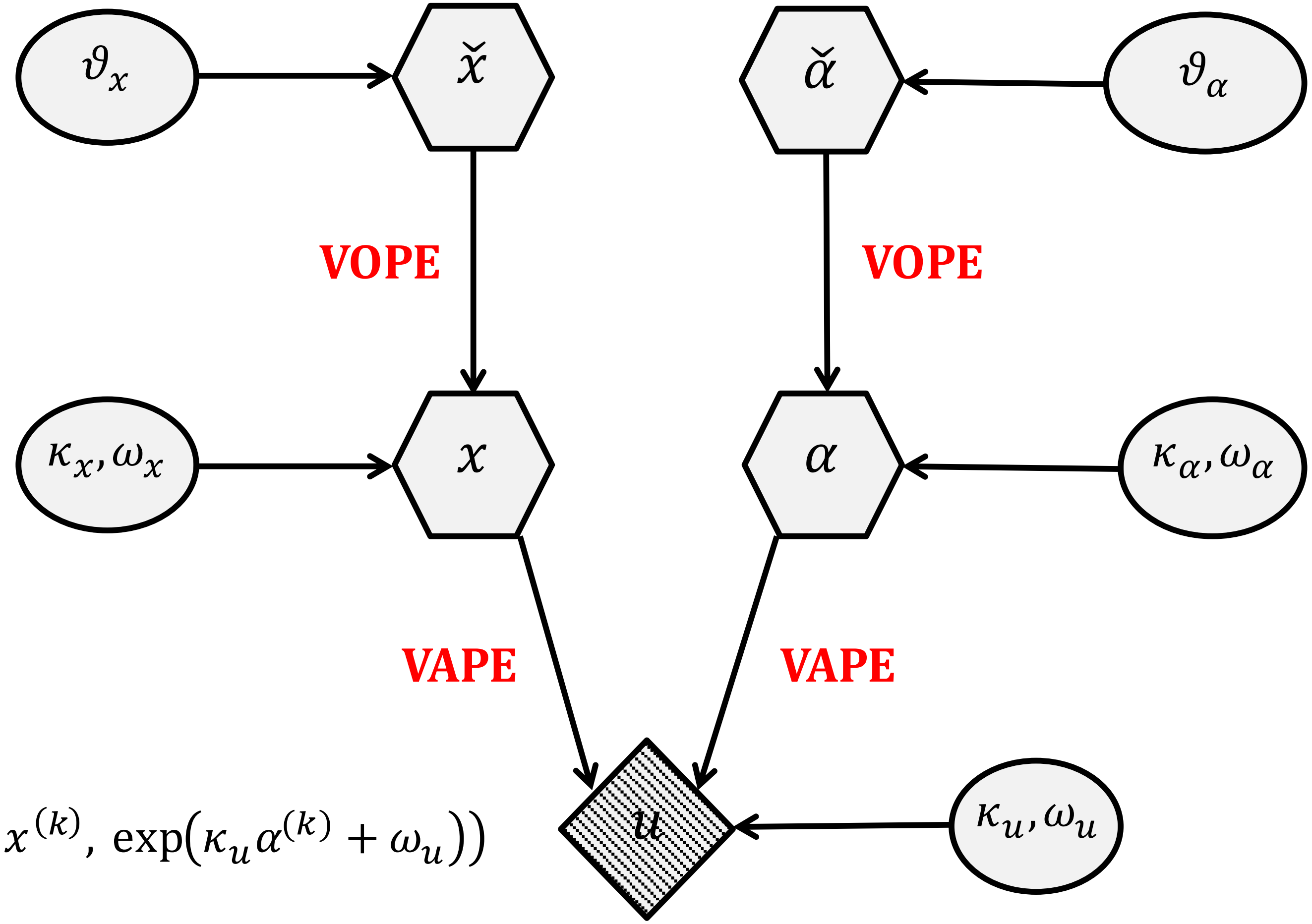
$$x_1^{(k)} \sim \mathcal{N}\left(x_1^{(k-1)} + z_1^{(k-1)}, \exp\left(\kappa_x x_2^{(k)} + \omega_x\right)\right)$$

Jumping Gaussian estimation task



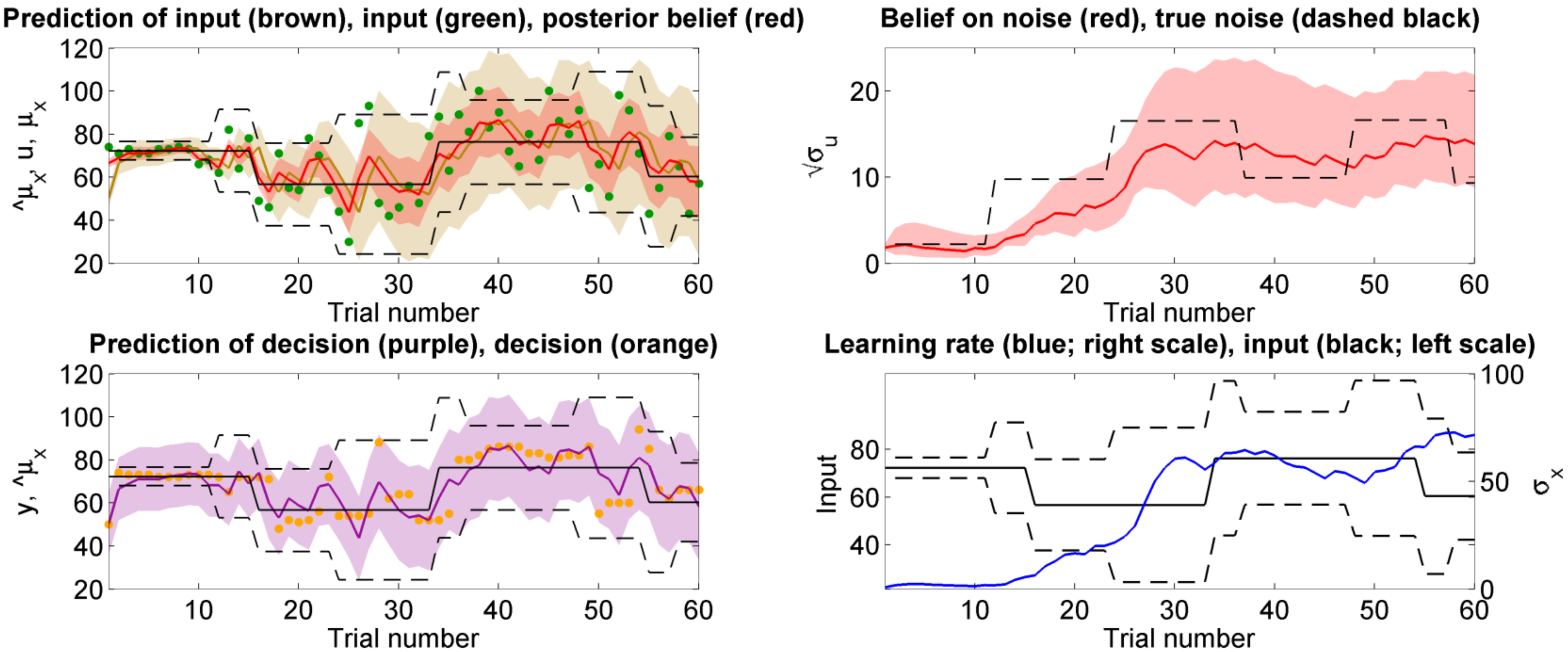
Chaohui Guo

Independent mean and variance model



$$u^{(k)} \sim \mathcal{N}\left(x^{(k)}, \exp\left(\kappa_u \alpha^{(k)} + \omega_u\right)\right)$$

Jumping Gaussian estimation task



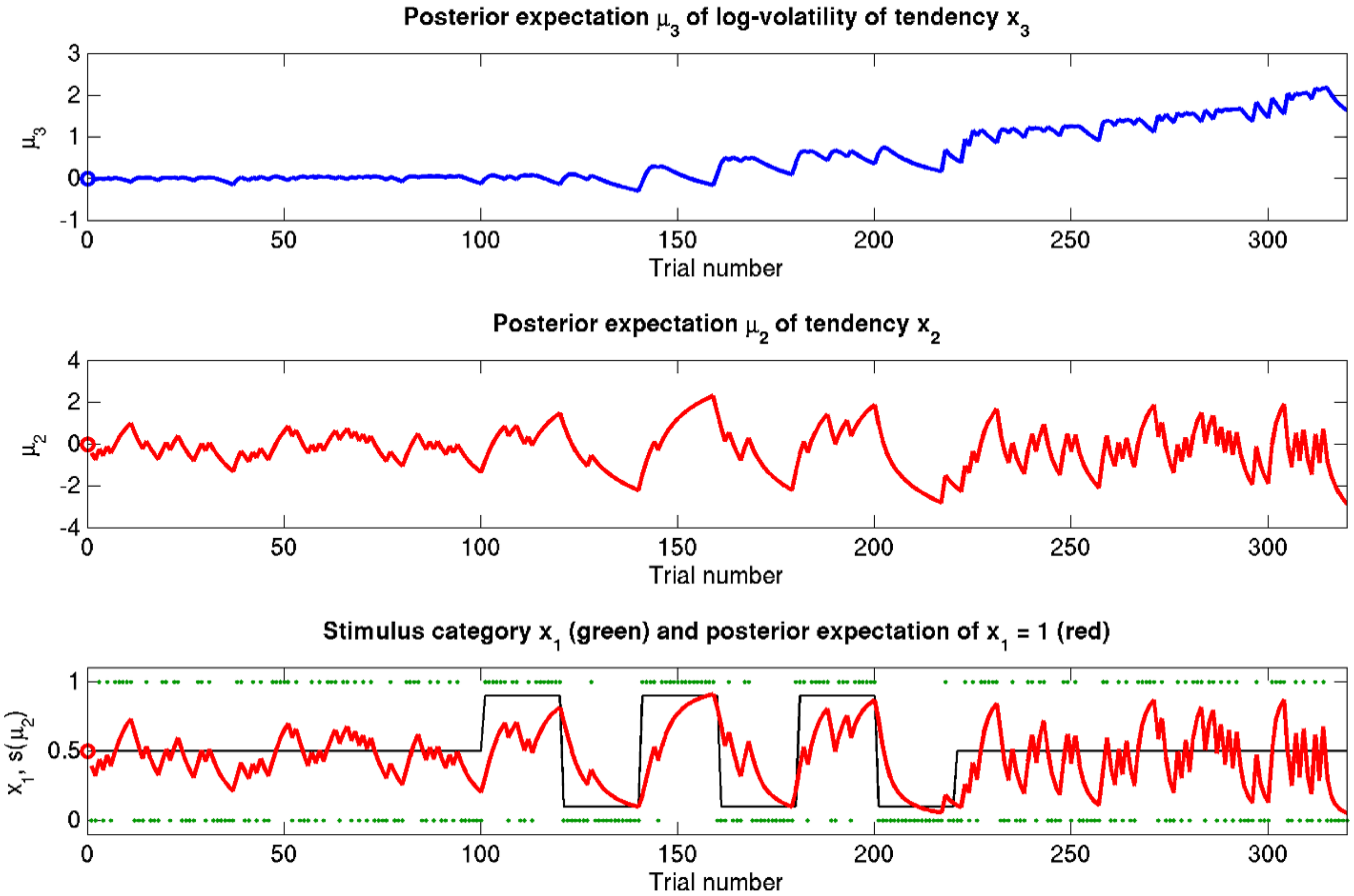
Example: Iglesias et al. (2013)

Model comparison:

	Behavioral study		fMRI study 1		fMRI study 2	
BMS results	PP	XP	PP	XP	PP	XP
HGF1	0.8435	1	0.7422	1	0.7166	1
HGF2	0.0259	0	0.0200	0	-	-
HGF3	0.0361	0	0.1404	0	0.1304	0
Sutton	0.0685	0	0.0710	0	0.0761	0
RW	0.0260	0	0.0264	0	0.0769	0

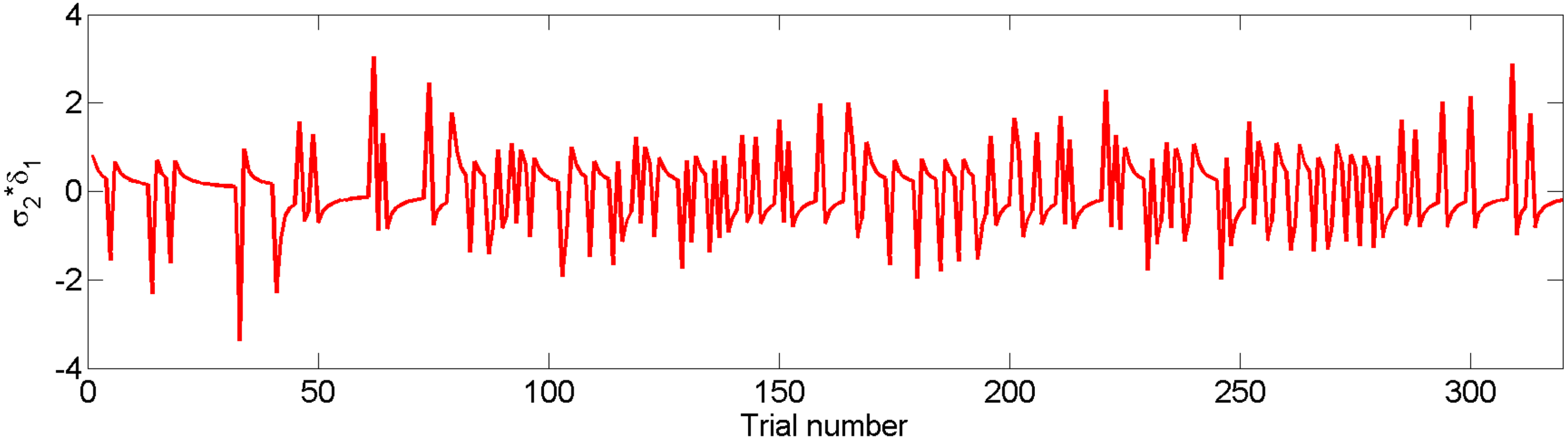
HGF: adaptive learning rate

Simulation: $\vartheta = 0.5,\ \omega = -2.2,\ \kappa = 1.4$

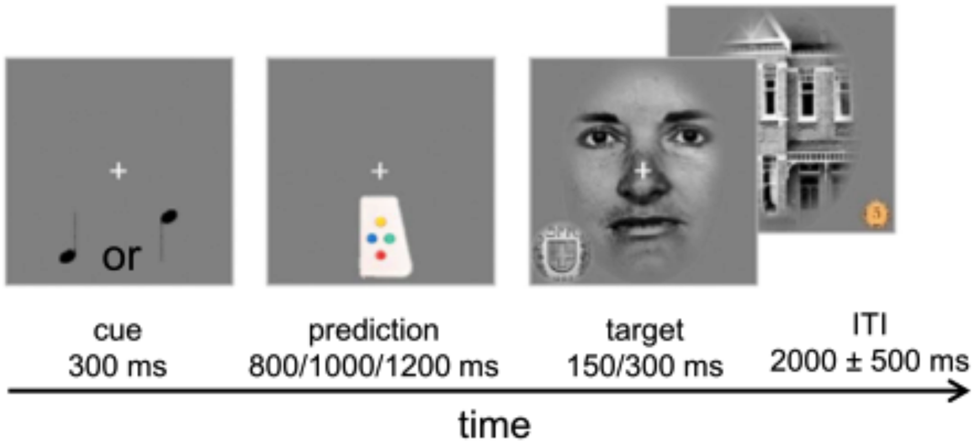


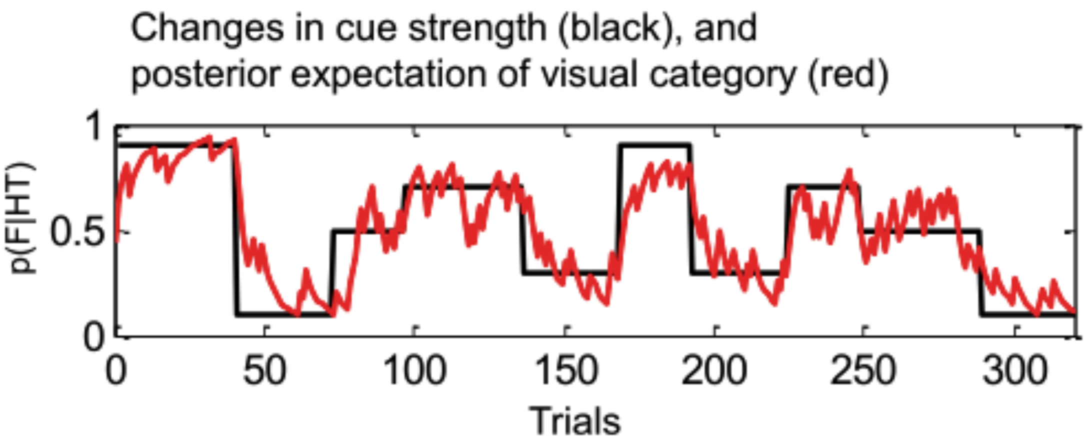
Individual model-based regressors

Uncertainty-weighted prediction error $\sigma_2 \cdot \delta_1$



Example: Iglesias et al. (2013)





Example: Iglesias et al. (2013)

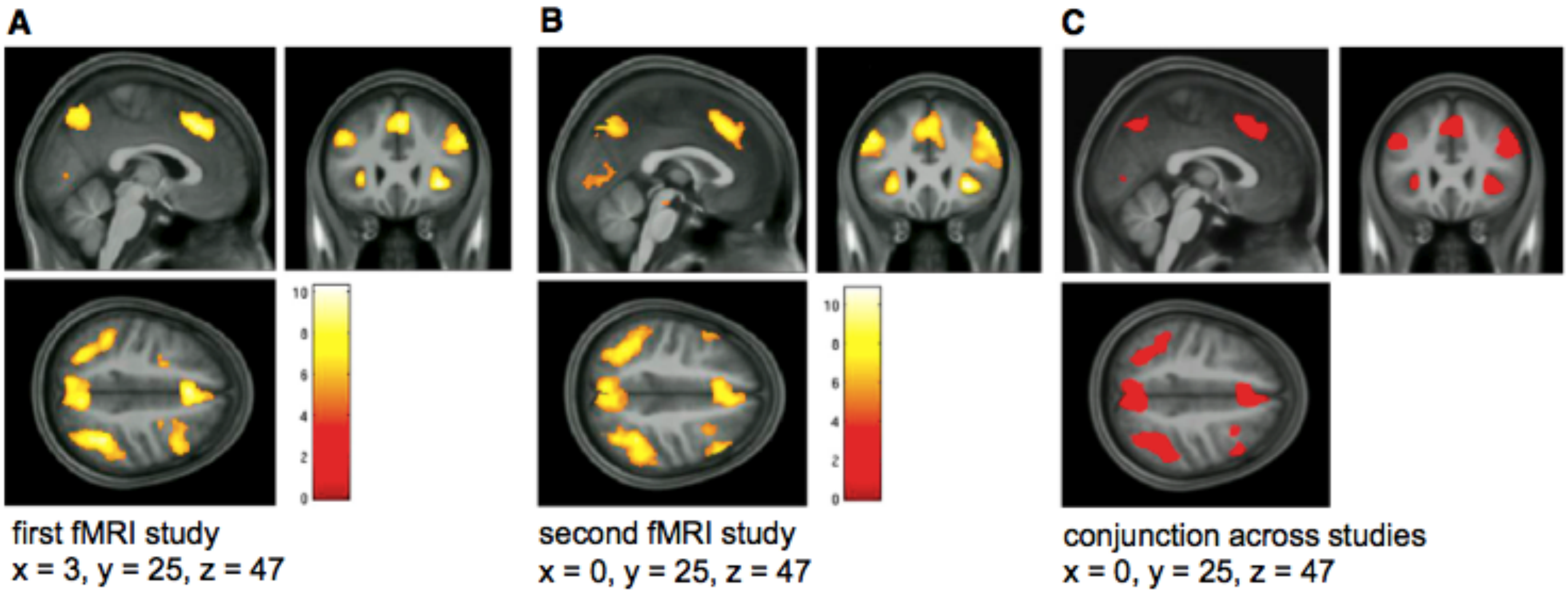


Figure 2. Whole-Brain Activations by ε_2
Activations by precision-weighted prediction error about visual stimulus outcome, ε_2, in the first fMRI study (A) and the second fMRI study (B). Both activation maps are shown at a threshold of $p < 0.05$, FWE corrected for multiple comparisons across the whole brain. To highlight replication across studies, (C) shows the results of a "logical AND" conjunction, illustrating voxels that were significantly activated in both studies.

Example: Iglesias et al. (2013)

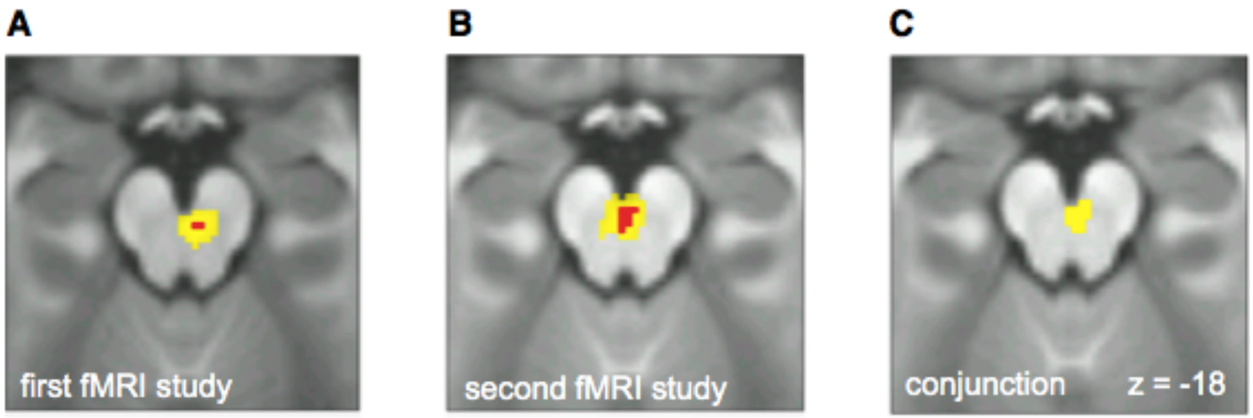


Figure 3. Midbrain Activation by ε_2
Activation of the dopaminergic VTA/SN associated with precision-weighted prediction error about stimulus category, ε_2. This activation is shown both at $p < 0.05$ FWE whole-brain corrected (red) and $p < 0.05$ FWE corrected for the volume of our anatomical mask comprising both dopaminergic and cholinergic nuclei (yellow).
(A) Results from the first fMRI study.
(B) Second fMRI study.
(C) Conjunction (logical AND) across both studies.

Example: Iglesias et al. (2013)

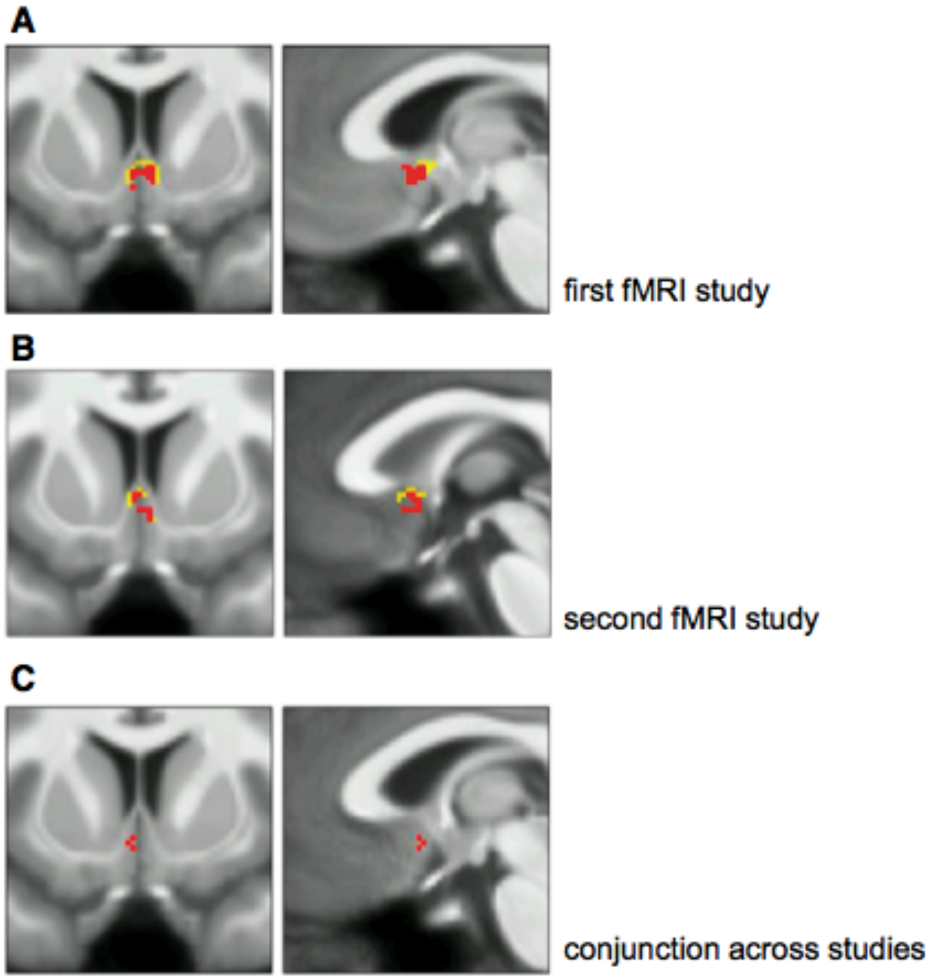


Figure 6. Basal Forebrain Activations by ε_3
Activation of the cholinergic basal forebrain associated with precision-weighted prediction error about stimulus probabilities ε_3 within the anatomically defined mask. For visualization of the activation area we overlay the results thresholded at $p < 0.05$ FWE corrected for the entire anatomical mask (red) on the results thresholded at $p < 0.001$ uncorrected (yellow) in the first (A: $x = 3, y = 9, z = -8$) and the second fMRI study (B: $x = 0, y = 10, z = -8$). (C) The conjunction analysis ("logical AND") across both studies ($x = 2, y = 11, z = -8$).

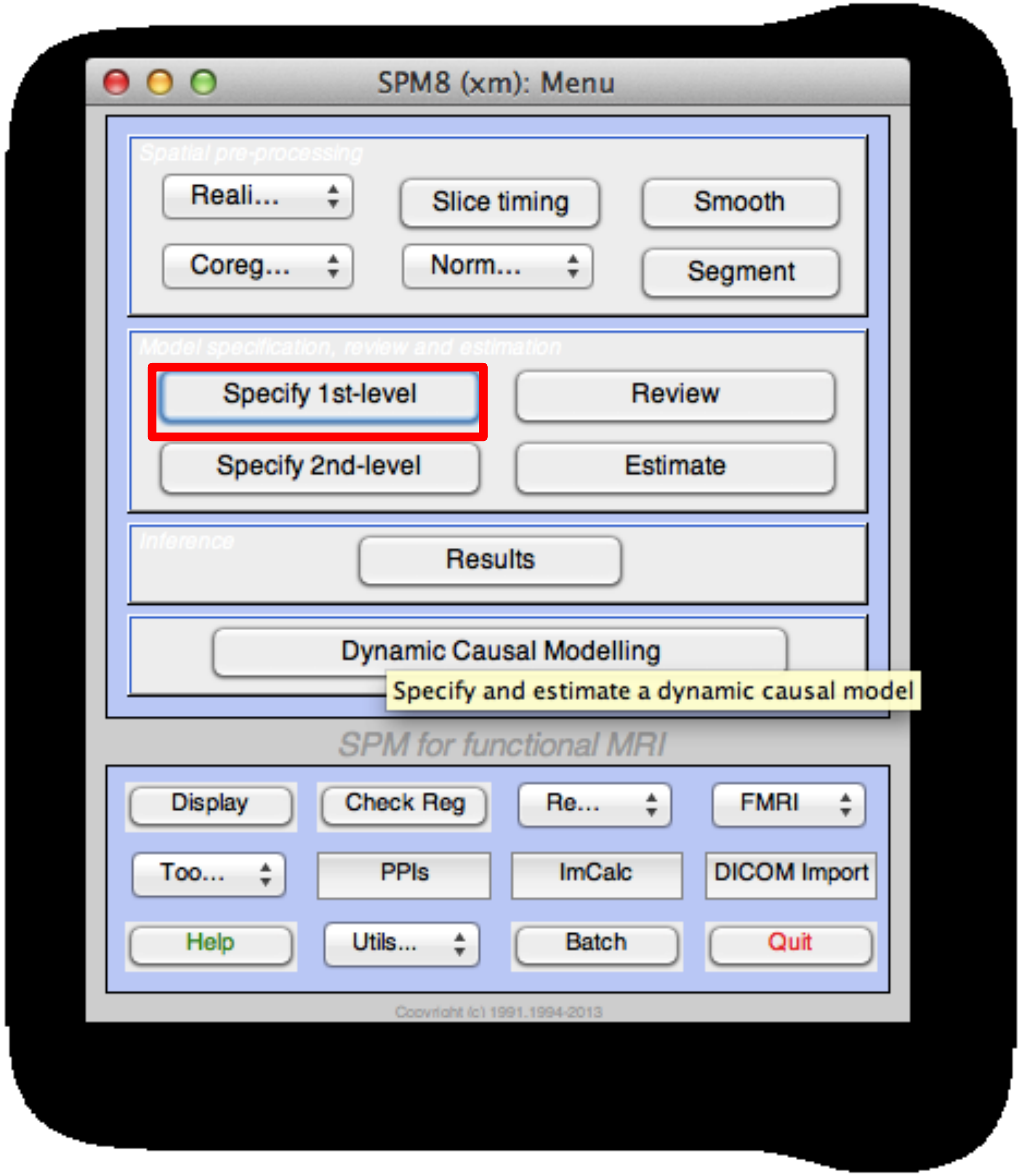
How to estimate and compare models: the HGF Toolbox

- Available at

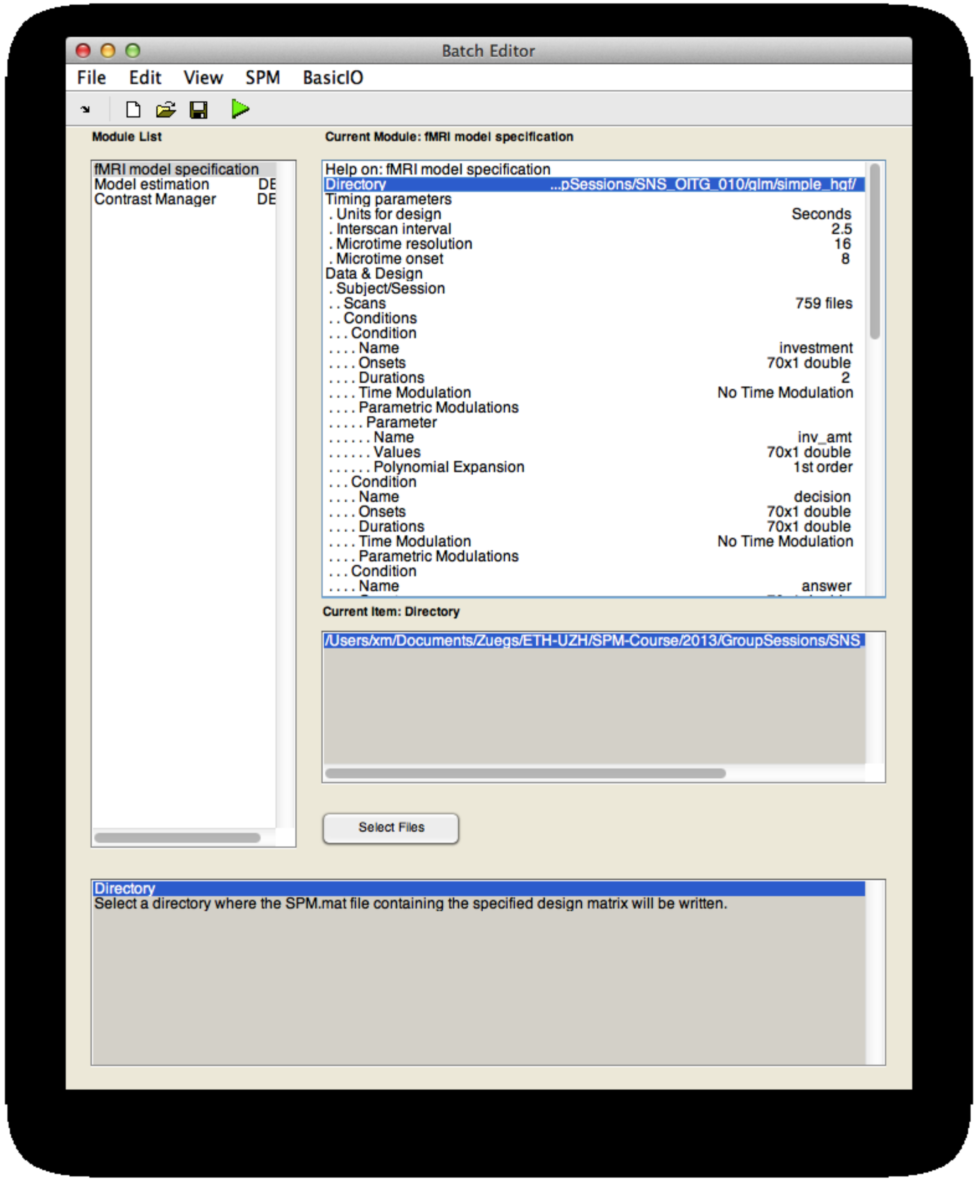
  http://www.tranlsationalneuromodeling.org/tapas

- Start with README and tutorial there
- Modular, extensible
- Matlab-based

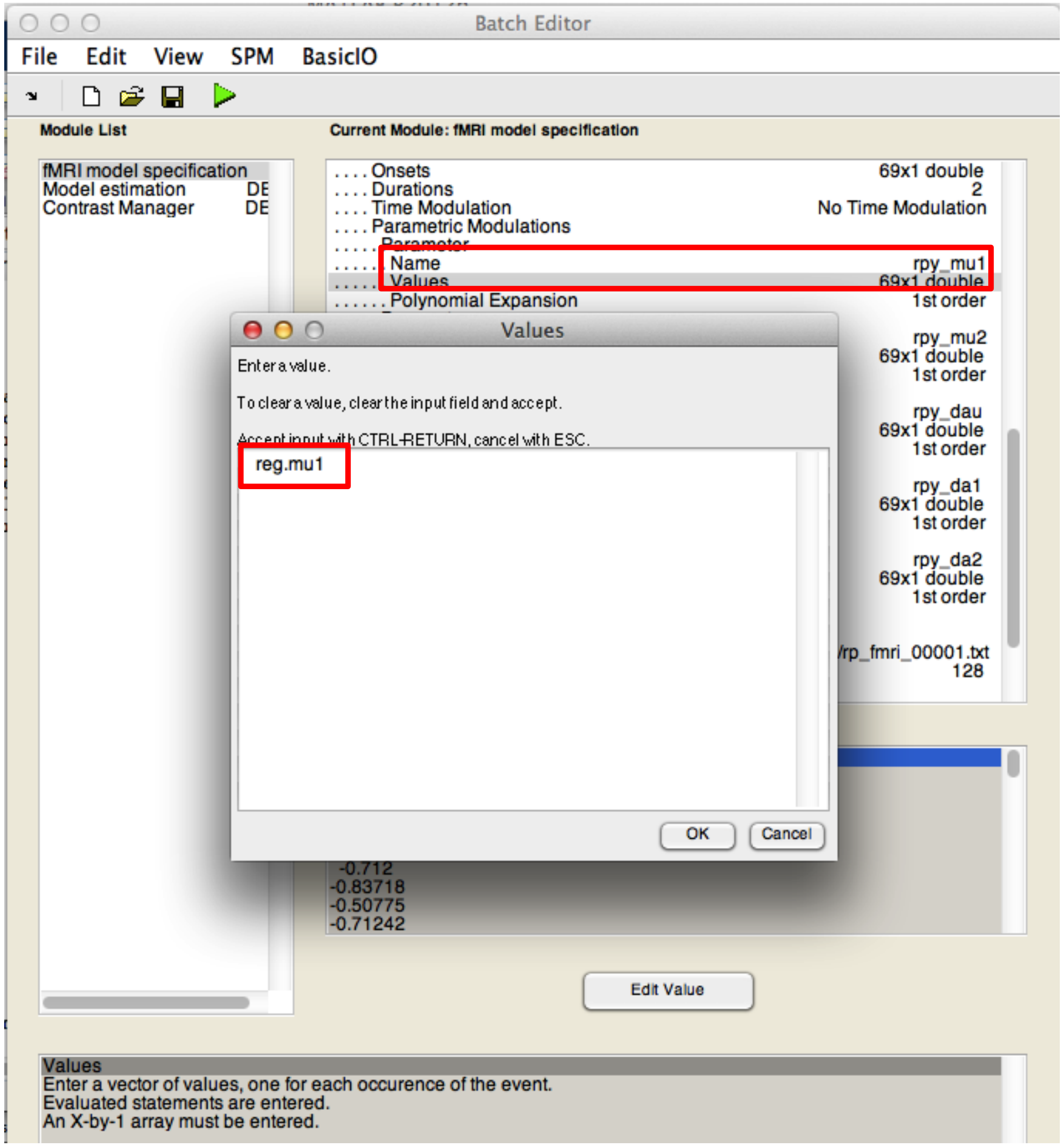
How it’s done in SPM



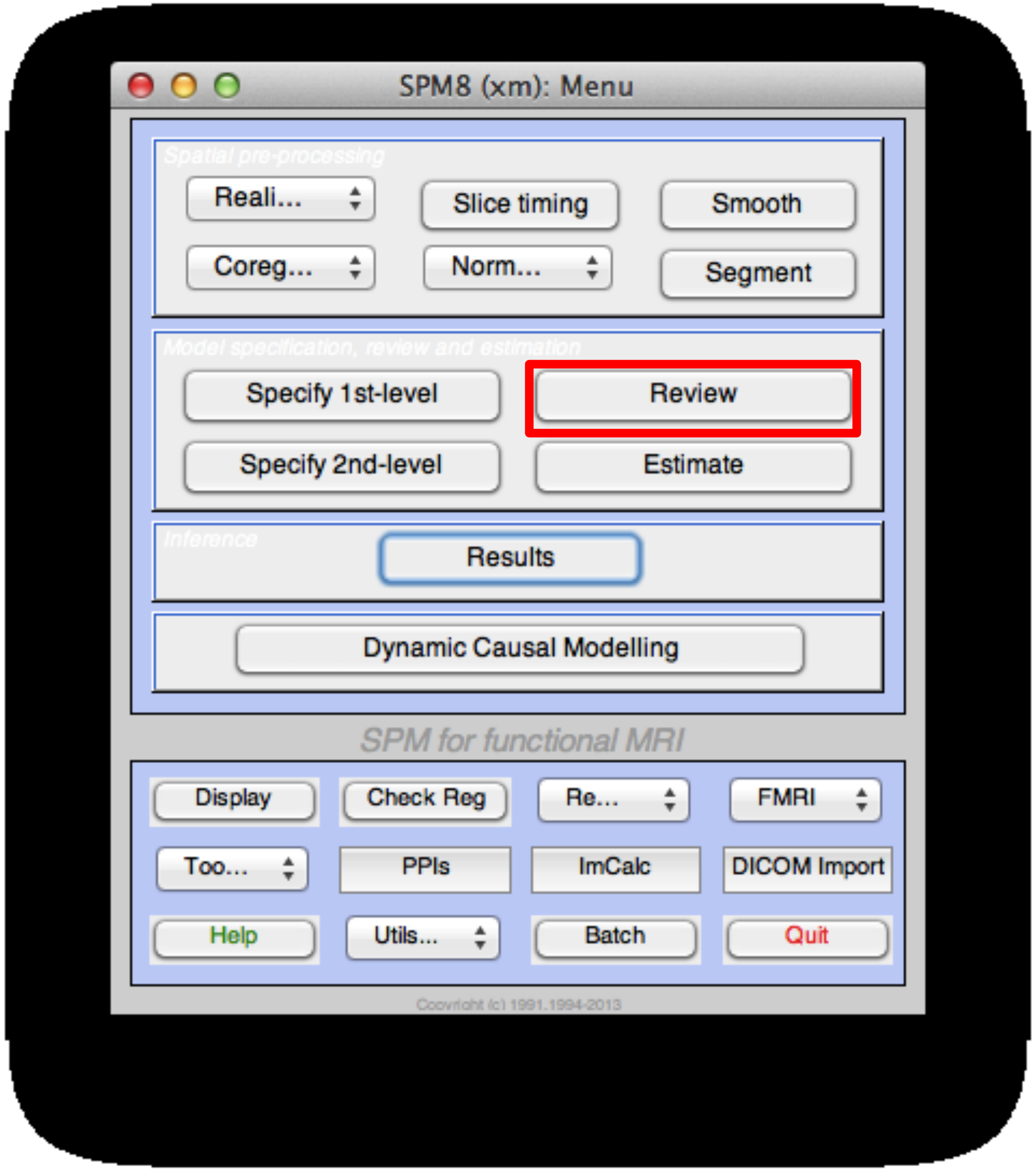
How it's done in SPM



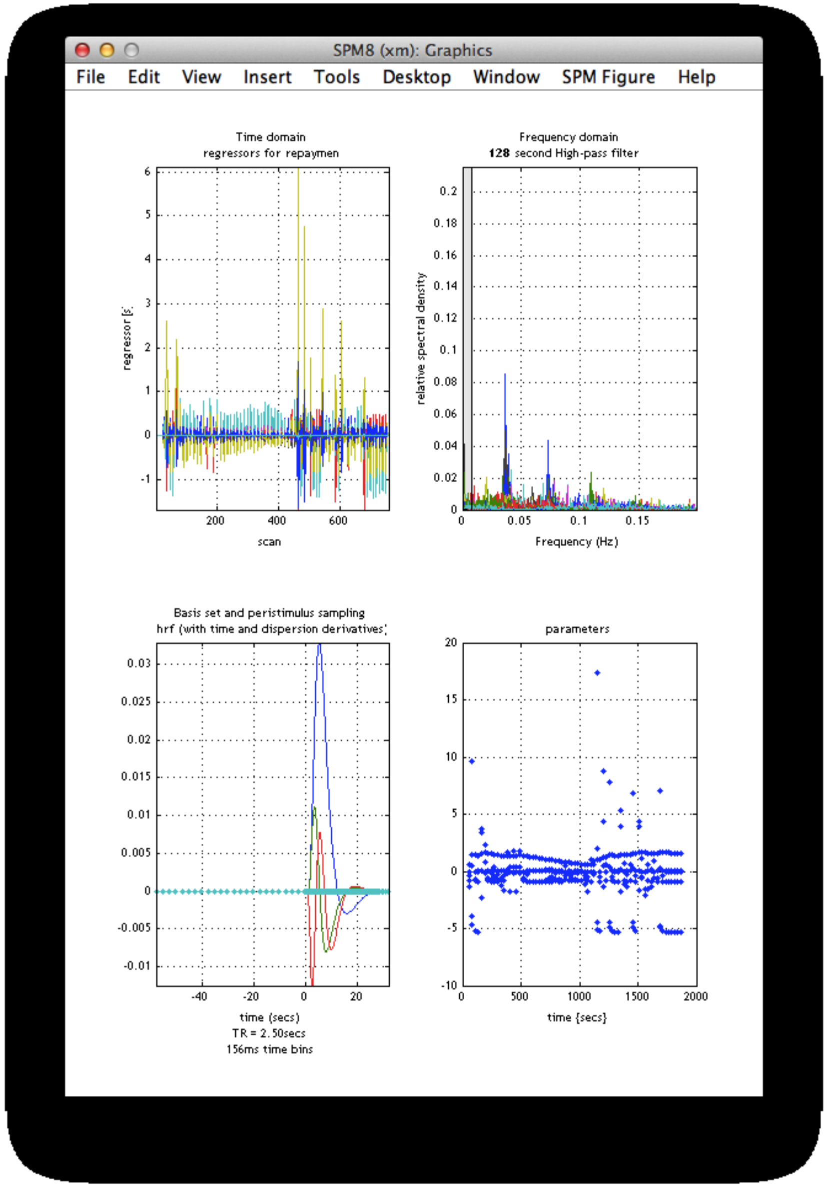
How it's done in SPM



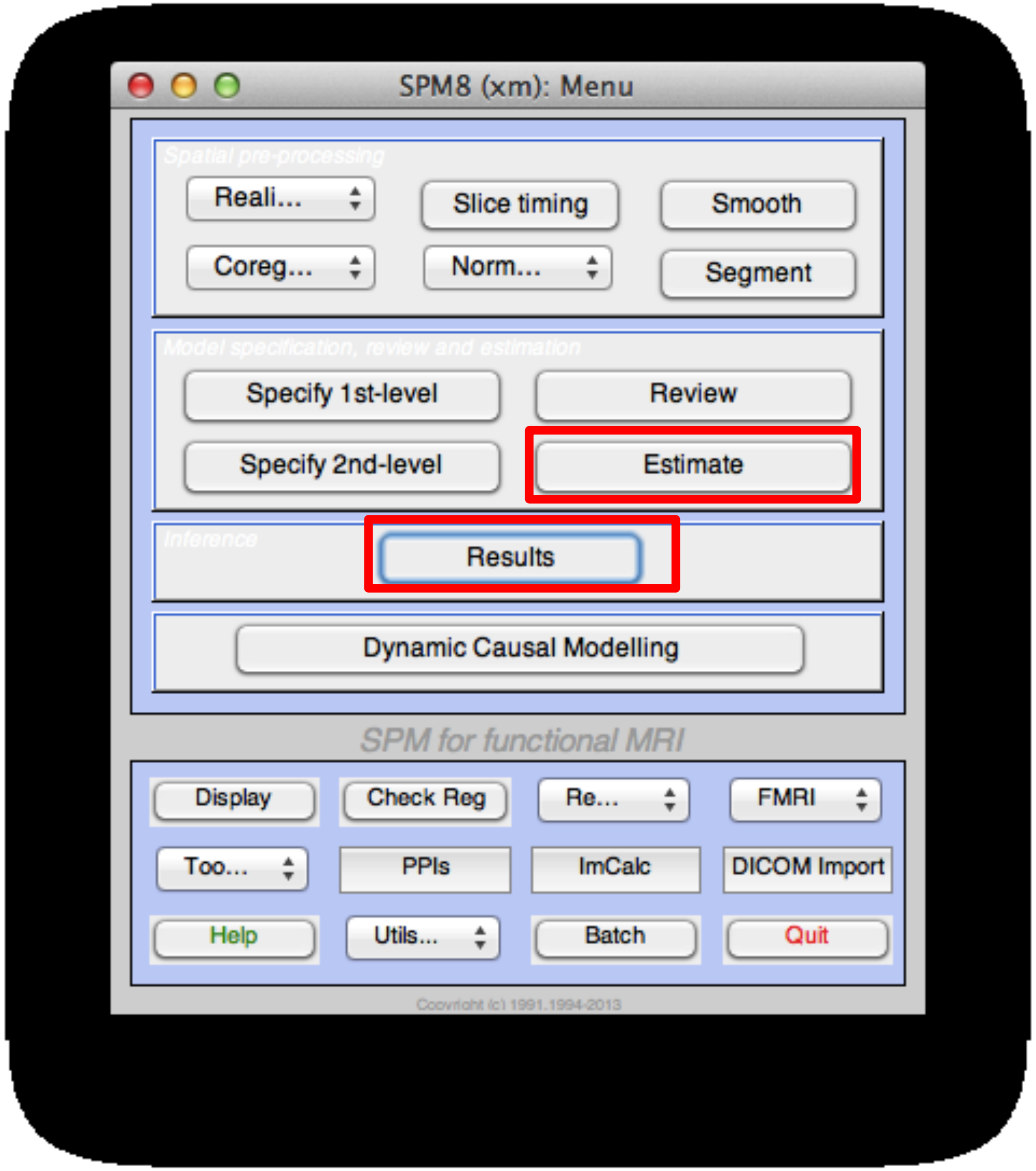
How it's done in SPM



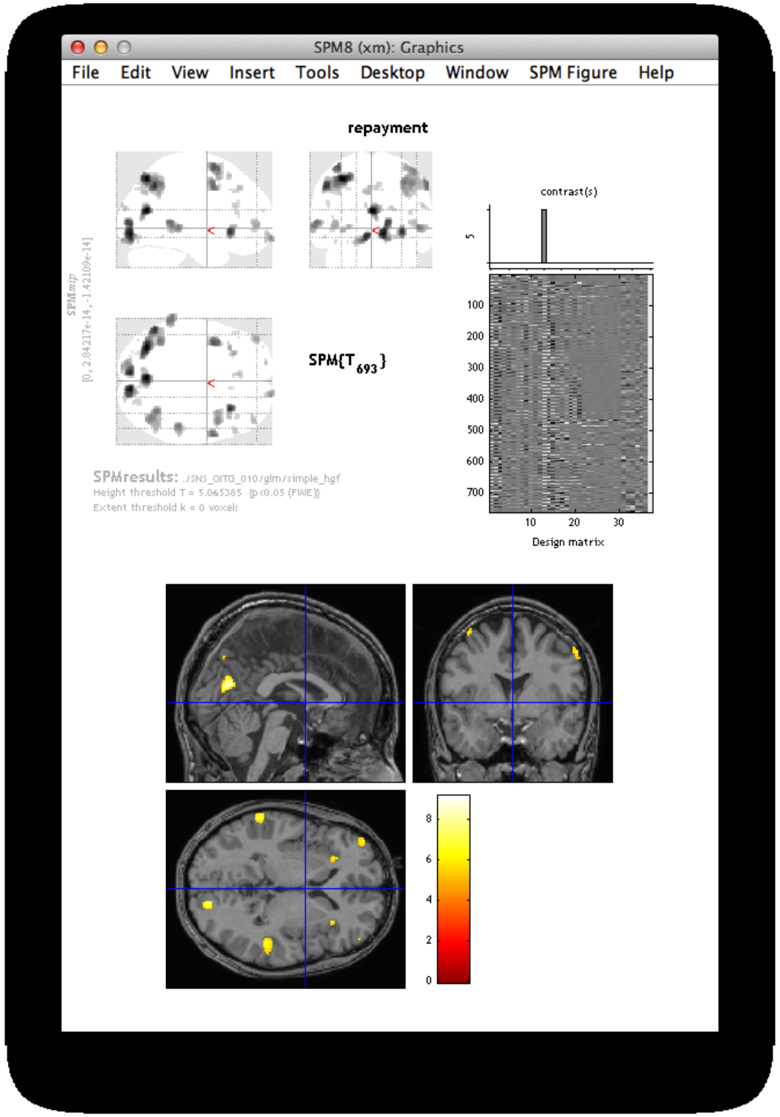
How it's done in SPM



How it’s done in SPM

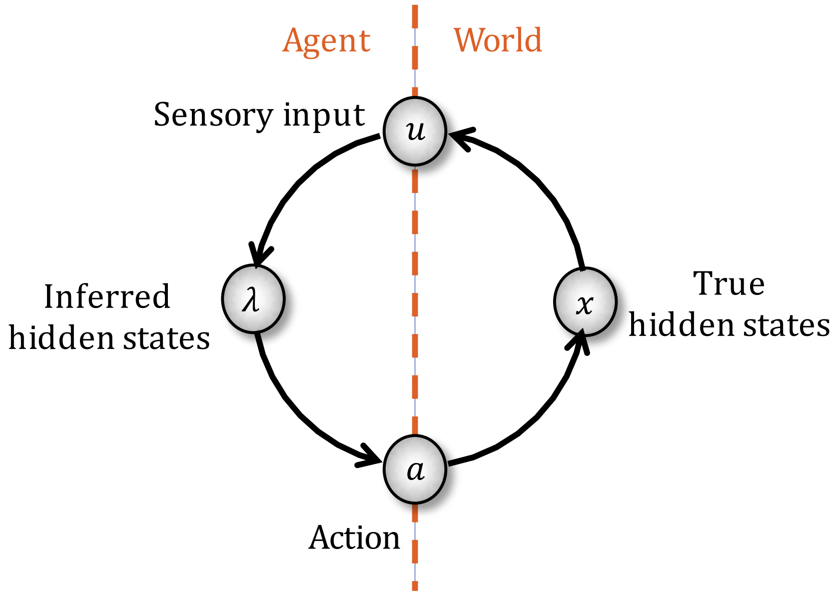


How it’s done in SPM



Take home

- The brain is an organ whose job is prediction.
- To make its predictions, it needs a model.
- Model-based imaging infers the model at work in the brain.
- It enables **inference** on **mechanisms, localization** of mechanisms, and **model comparison**.



Thank you