AARHUS UNIVERSITY

From Perception to Action: Hierarchical Gaussian Filters (HGFs)

Christoph Mathys

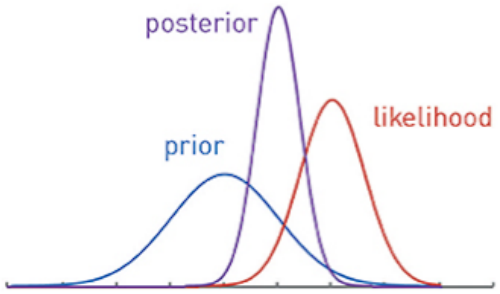
Computational Psychiatry Course

Zurich, 9 September 2020

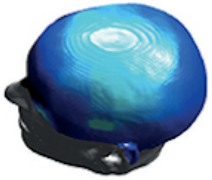
Computational modelling and the inferential brain

Bayes' Theorem

$$p(x|y,m) = \frac{p(y|x,m)p(x|m)}{p(y|m)}$$



Generative models as computational assays



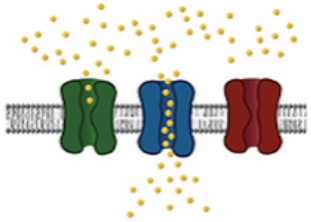
measurement y

forward model

$p(y|x,m)p(x|m)$

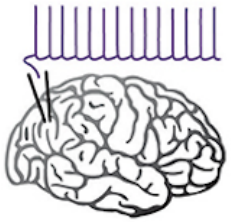
$p(x|y,m)$

inference



hidden system state x

Perception as the inversion of a generative model



neuronal activity y

forward model

$p(y|x,m)p(x|m)$

$p(x|y,m)$

perception

environmental state x

Stephan et al. (2016), Front. Hum. Neurosci., 10:550

Before we return to Bayes: a very simple example of updating in response to new information

Imagine the following situation:

You're on a boat, you're lost in a storm and trying to get back to shore. A lighthouse has just appeared on the horizon, but you can only see it when you're at the peak of a wave. Your GPS etc., has all been washed overboard, but what you can still do to get an idea of your position is to measure the angle between north and the lighthouse. These are your measurements (in degrees):

76, 73, 75, 72, 77

What number are you going to base your calculation on?

Right. The mean: 74.6. How do you calculate that?

Updating the mean of a series of observations

The usual way to calculate the mean $\bar{u}$ of $u_1, u_2, \dots, u_n$ is to take

$$\bar{u} = \frac{1}{n}\sum_{i=1}^{n} u_i$$

This requires you to remember all u_i, which can become inefficient. Since the measurements arrive sequentially, we would like to update $\bar{u}$ sequentially as the u_i come in – without having to remember them.

It turns out that this is possible. After some algebra (see next slide), we get

$$\bar{u}_{n+1} = \bar{u}_n + \frac{1}{n+1}(u_{n+1} - \bar{u}_n)$$

Updating the mean of a series of observations

Proof of sequential update equation:

$$\bar{u}_{n+1} = \frac{1}{n+1}\sum_{i=1}^{n+1} u_i = \frac{1}{n+1}\left(u_{n+1} + n \cdot \frac{1}{n}\sum_{i=1}^{n} u_i\right) =$$

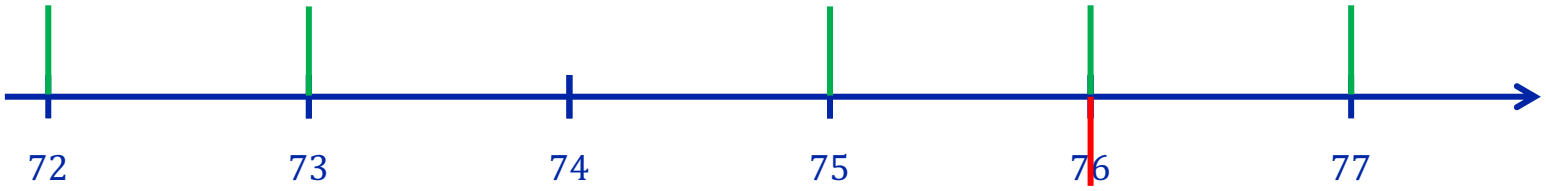
$$= \frac{1}{n+1}(u_{n+1} + n\bar{u}_n) = \frac{1}{n+1}(u_{n+1} - \bar{u}_n + (n+1)\bar{u}_n)$$

$$= \bar{u}_n + \frac{1}{n+1}(u_{n+1} - \bar{u}_n)$$

q.e.d.

Updating the mean of a series of observations

The seqential updates in our example now look like this:



$$\bar{u}_1 = 76$$

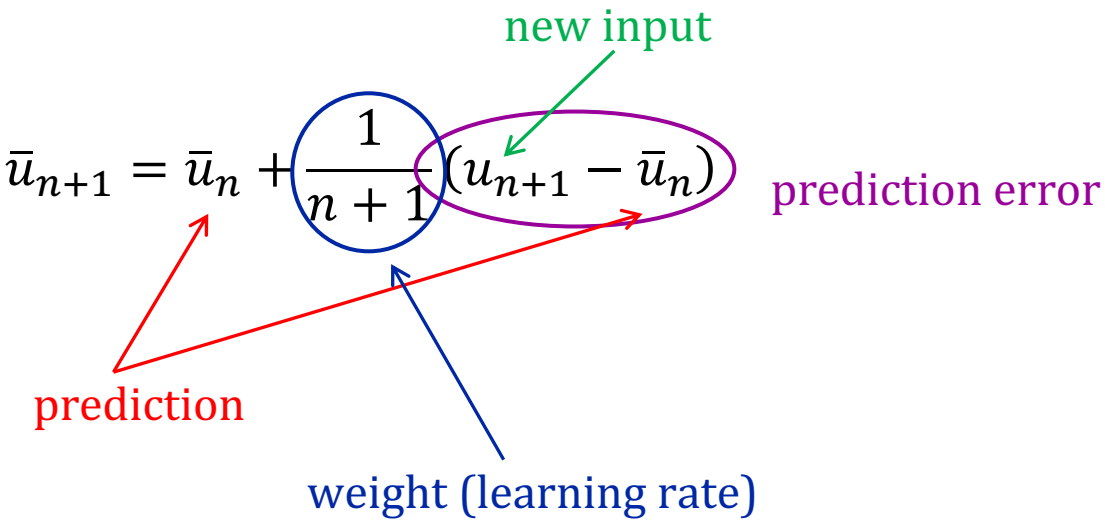
$$\bar{u}_2 = 76 + \frac{1}{2}(73 - 76) = 74.5$$

$$\bar{u}_3 = 74.5 + \frac{1}{3}(75 - 74.5) = 74.\overline{6}$$

$$\bar{u}_4 = 74.\overline{6} + \frac{1}{4}(72 - 74.\overline{6}) = 74$$

$$\bar{u}_5 = 74 + \frac{1}{5}(77 - 74) = 74.6$$

What are the building blocks of the updates we've just seen?



Is this a general pattern?

More specifically, does it generalize to Bayesian inference?

Indeed, it turns out that in many cases, Bayesian inference can be based on parameters that are updated using **precision-weighted prediction errors**.

Updates in a simple Gaussian model

Think boat, lighthouse, etc., again, but now we're doing Bayesian inference.

Before we make the next observation, our belief about the true value of the state x can be described by a Gaussian prior:

$$p(x) \sim \mathcal{N}(\mu_x, \pi_x^{-1})$$

The likelihood of an observation u is also Gaussian, with precision π_ε :

$$p(u|x) \sim \mathcal{N}(x, \pi_\varepsilon^{-1})$$

Bayes' rule now tells us that the posterior is Gaussian again:

$$p(x|u) = \frac{p(u|x)p(x)}{\int p(u|x')p(x')\mathrm{d}x'} \sim \mathcal{N}\left(\mu_{x|u}, \pi_{x|u}^{-1}\right)$$

Updates in a simple Gaussian model

Here's how the updates to the sufficent statistics μ and π describing our belief look like:

$$\pi_{x|u} = \pi_x + \pi_\varepsilon$$

$$\mu_{x|u} = \mu_x + \frac{\pi_\varepsilon}{\pi_{x|u}}(u - \mu_x)$$

prediction error

prediction

weight (learning rate)=$\frac{\text{how much we're learning here}}{\text{how much we already know}}$

The mean is updated by an uncertainty-weighted (more specifically: precision-weighted) prediction error.

The size of the update is proportional to the likelihood precision and inversely proportional to the posterior precision.

This pattern is not specific to the univariate Gaussian case, but generalizes to Bayesian updates for all exponential families of likelihood distributions with conjugate priors (i.e., to all formal descriptions of inference you are ever likely to need).

How to reveal the precision-weighting of prediction errors when simple exponential-family likelihoods will not do

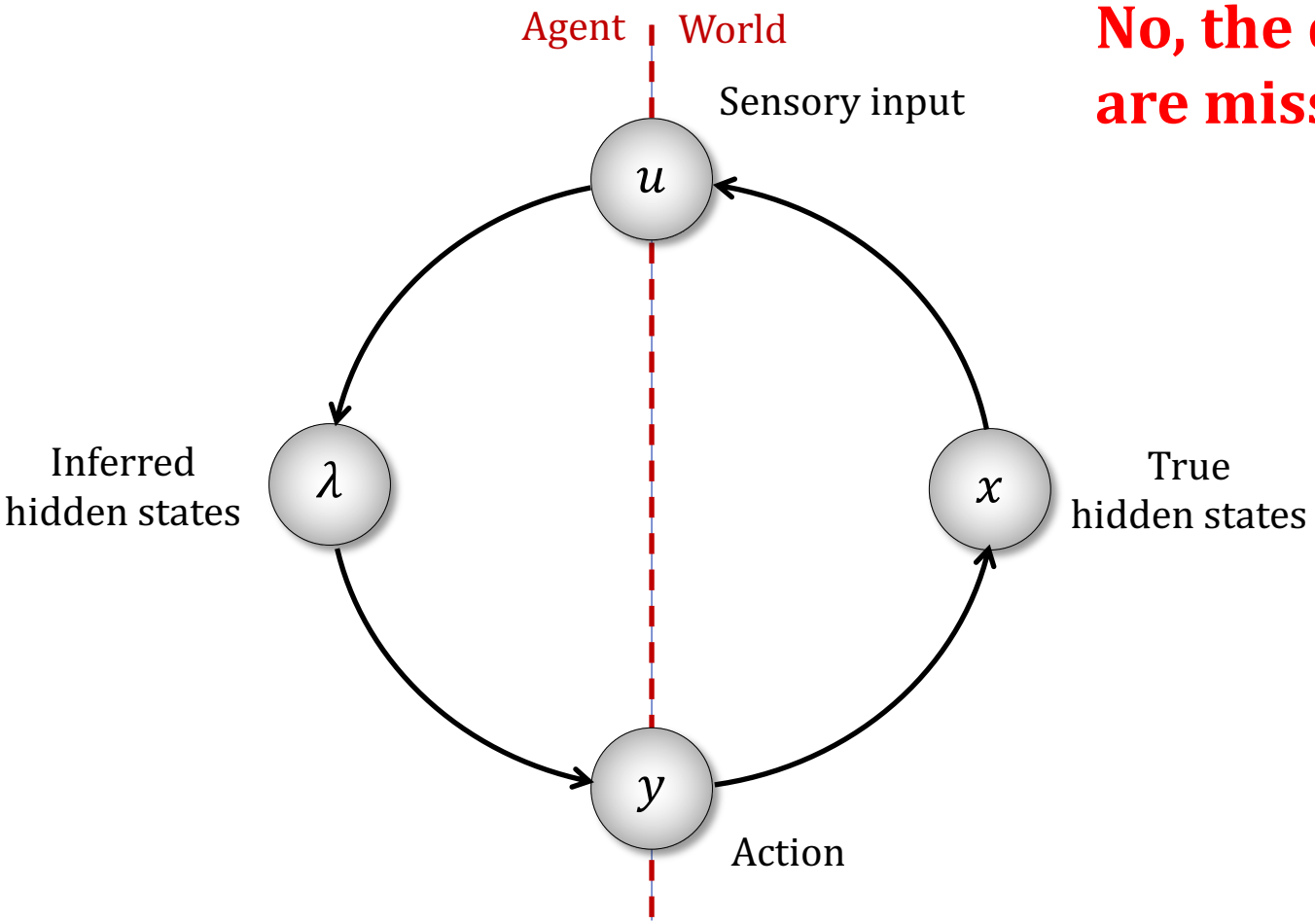
- Formulate the problem hierarchically (i.e., imitate evolution: when it built a brain that supports a mind which is a model of its environment, it came up with a (largely) hierarchical solution)

- Separate levels using a mean-field approximation

- Derive update equations

- Example: HGF

But: does inference as we've described it adequately describe the situation of actual biological agents?



No, the dynamics are missing!

What about dynamics?

Up to now, we've only looked at inference on static quantities, but biological agents live in a continually changing world.

In our example, the boat's position changes and with it the angle to the lighthouse.

How can we take into account that old information becomes obsolete? If we don't, our learning rate becomes smaller and smaller because our eqations were derived under the assumption that we're accumulating information about a stable quantity.

What's the simplest way to keep the learning rate from going too low?

Keep it constant!

So, taking the update equation for the mean of our observations as our point of departure...

$$\bar{u}_n = \bar{u}_{n-1} + \frac{1}{n}(u_n - \bar{u}_{n-1}),$$

... we simply replace $\frac{1}{n}$ with a constant α (and $\bar{u}$ with a generic value q):

$$q_n = q_{n-1} + \alpha(u_n - q_{n-1}).$$

This is called *Rescorla-Wagner learning* [although it wasn't this line of reasoning that led Rescorla & Wagner (1972) to their formulation].

Does a constant learning rate solve our problems?

Partly: it implies a certain rate of forgetting because it amounts to taking only the $n = \frac{1}{\alpha}$ last data points into account.

However, an optimal learning rate

a) Balances the need to learn faster as uncertainty increases with the need to learn more slowly as observation noise increases

b) Takes account of all sources of uncertainty (outcome, informational, environmental)

What we really need is an adaptive learning rate that accurately reflects the changing nature of the environment.

Dealing with nonstationary environments: the Kalman filter

- We return to the Bayesian version of the lighthouse problem
- Relaxing the assumption that the underlying hidden state x is stationary and replacing it with a Gaussian random walk gives us the **Kalman filter:**

$$p\left(x^{(k)}|x^{(k-1)},\vartheta\right)=\mathcal{N}\left(x^{(k)};x^{(k-1)},\vartheta\right)$$

$$p\left(u^{(k)}|x^{(k)},\varepsilon\right)=\mathcal{N}\left(u^{(k)};x^{(k)},\varepsilon\right)$$

- Combining this with the **prior**

$$p\left(x^{(k-1)}\right)=\mathcal{N}\left(x^{(k-1)};\mu_x^{(k-1)},1/\pi_x^{(k-1)}\right),\ldots$$

Dealing with nonstationary environments: the Kalman filter

... and doing some algebra, we get the **posterior**

$$p\left(x^{(k)}\right) = \mathcal{N}\left(x^{(k)}; \mu_x^{(k)}, 1/\pi_x^{(k)}\right)$$

with

$$\pi_x^{(k)} = \frac{1}{\sigma_x^{(k-1)} + \vartheta} + \frac{1}{\varepsilon} = \hat{\pi}_x^{(k-1)} + \hat{\pi}_u$$

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

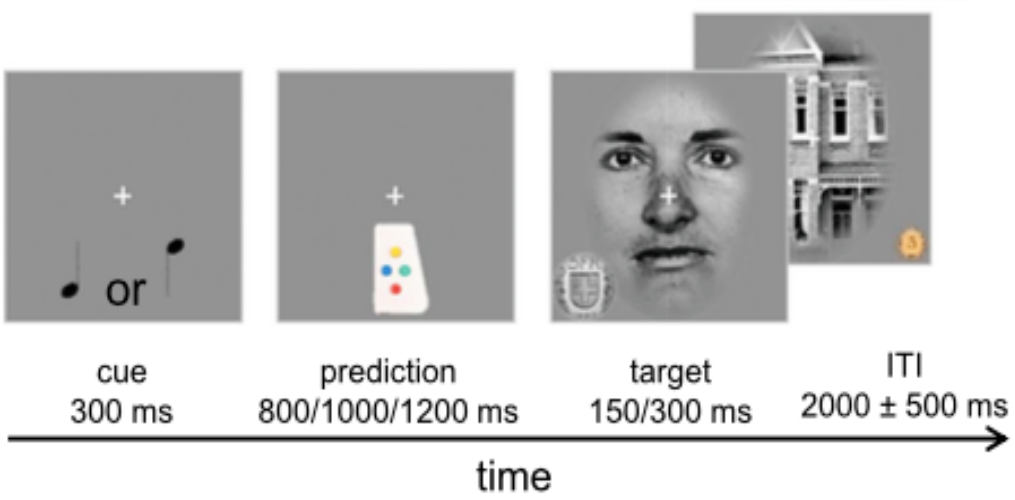
$$= \mu_x^{(k-1)} + \frac{\hat{\pi}_u}{\frac{1}{\sigma_x^{(k-1)} + \vartheta} + \hat{\pi}_u}\left(u^{(k)} - \mu_x^{(k-1)}\right)$$

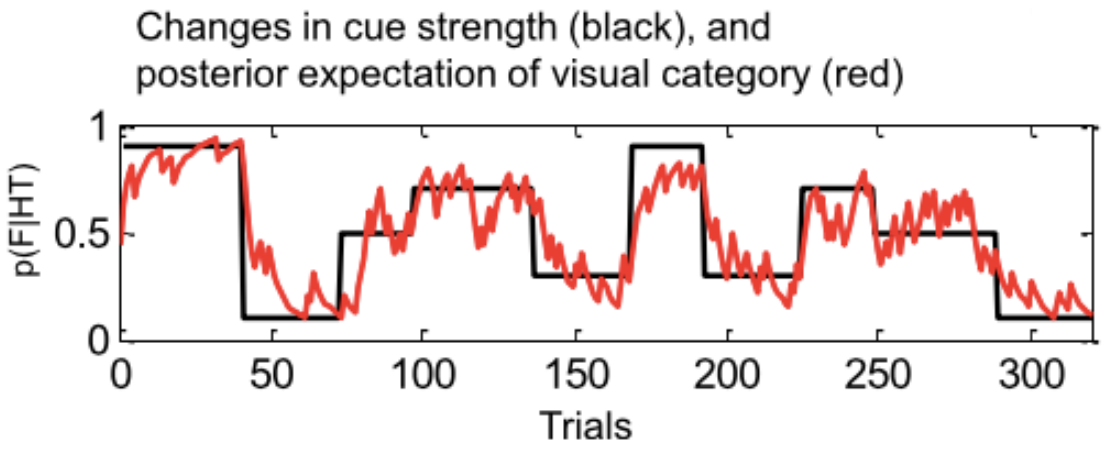
The Kalman filter is optimal for linear dynamic systems.

Unfortunately, except for simple physical systems, **the world is not linear**. Living organisms need to be able to filter inputs whose rate of change changes, in other words: **processes whose volatility is volatile.**

Where would we need a model with an adaptive learning rate?

Task of Iglesias et al., *Neuron*, (2013):



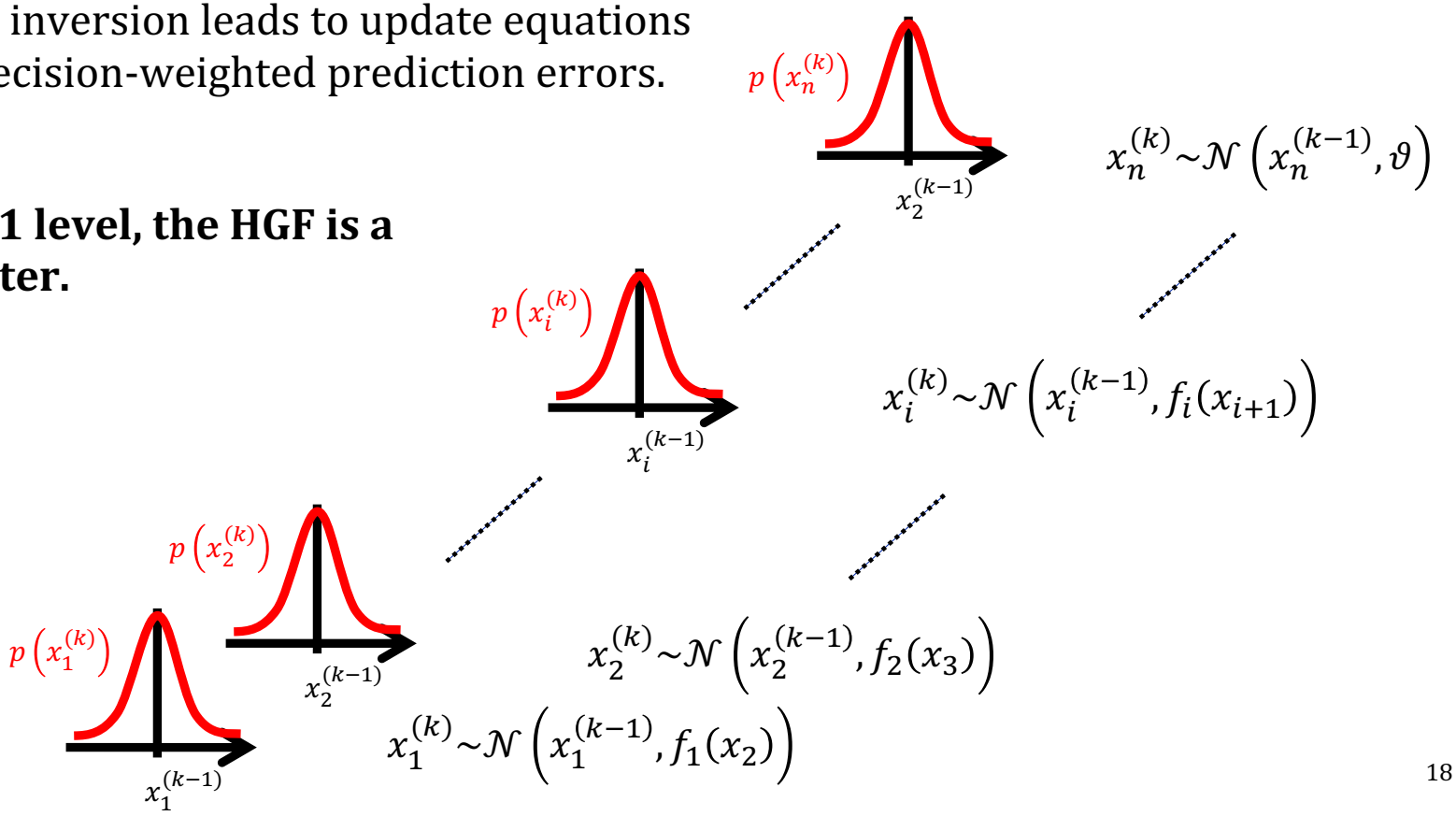


The hierarchical Gaussian filter (HGF, Mathys et al., 2011; 2014)

The HGF provides a generic solution to the problem of adapting one's learning rate in a volatile environment.

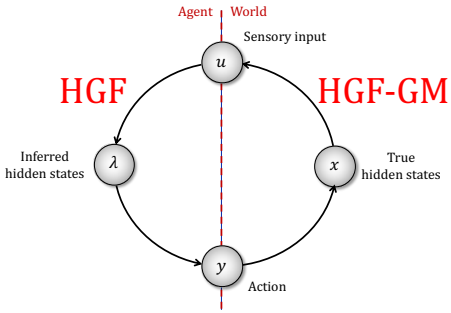
Variational inversion leads to update equations that are precision-weighted prediction errors.

With only 1 level, the HGF is a Kalman filter.



Variational inversion and update equations

- Important distinction: **generative model** (HGF-GM) vs its inversion, **inference model** (HGF proper)



- Inversion of HGF-GM 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** (HGF proper) 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

- The updates are **precision-weighted prediction errors**.

Updates at the first level

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):

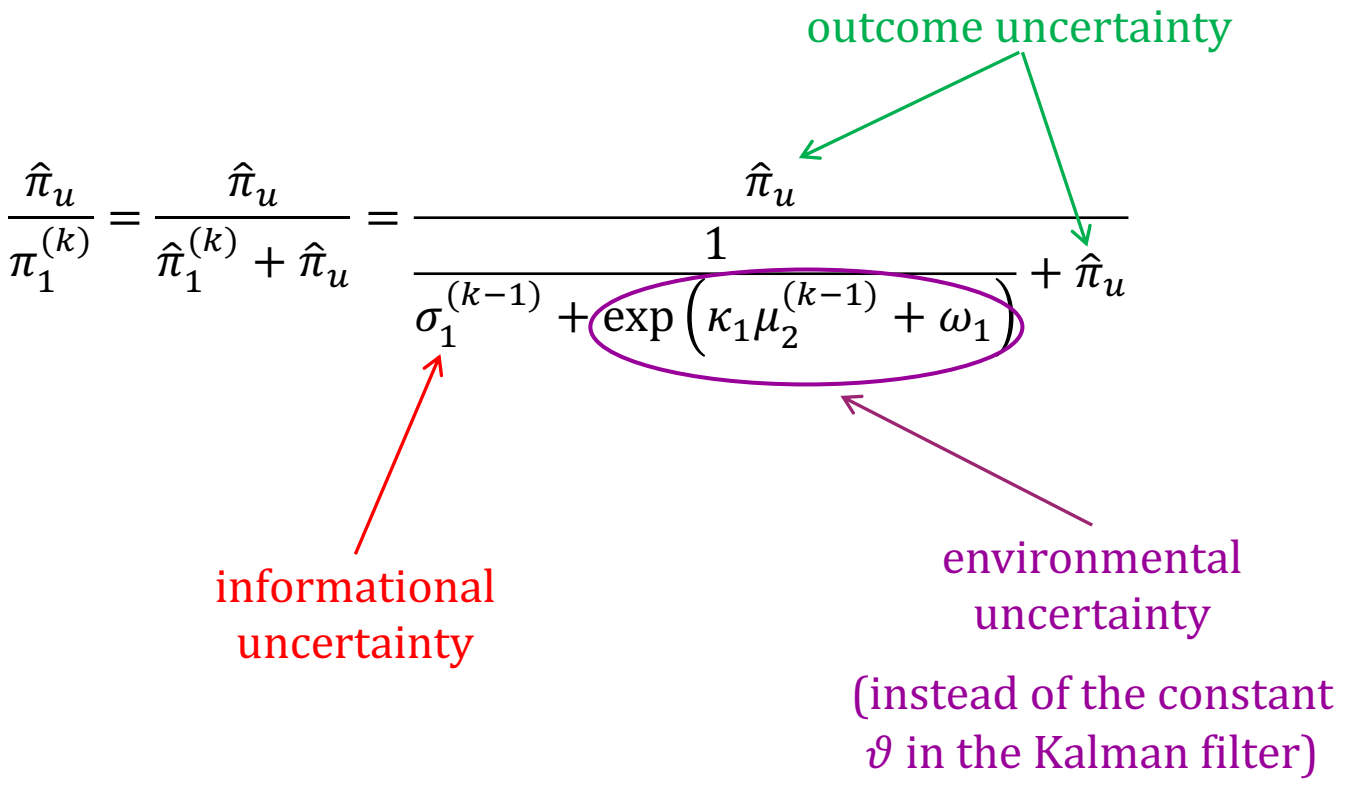
$$\pi_1^{(k)} = \hat{\pi}_1^{(k)} + \hat{\pi}_u$$

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

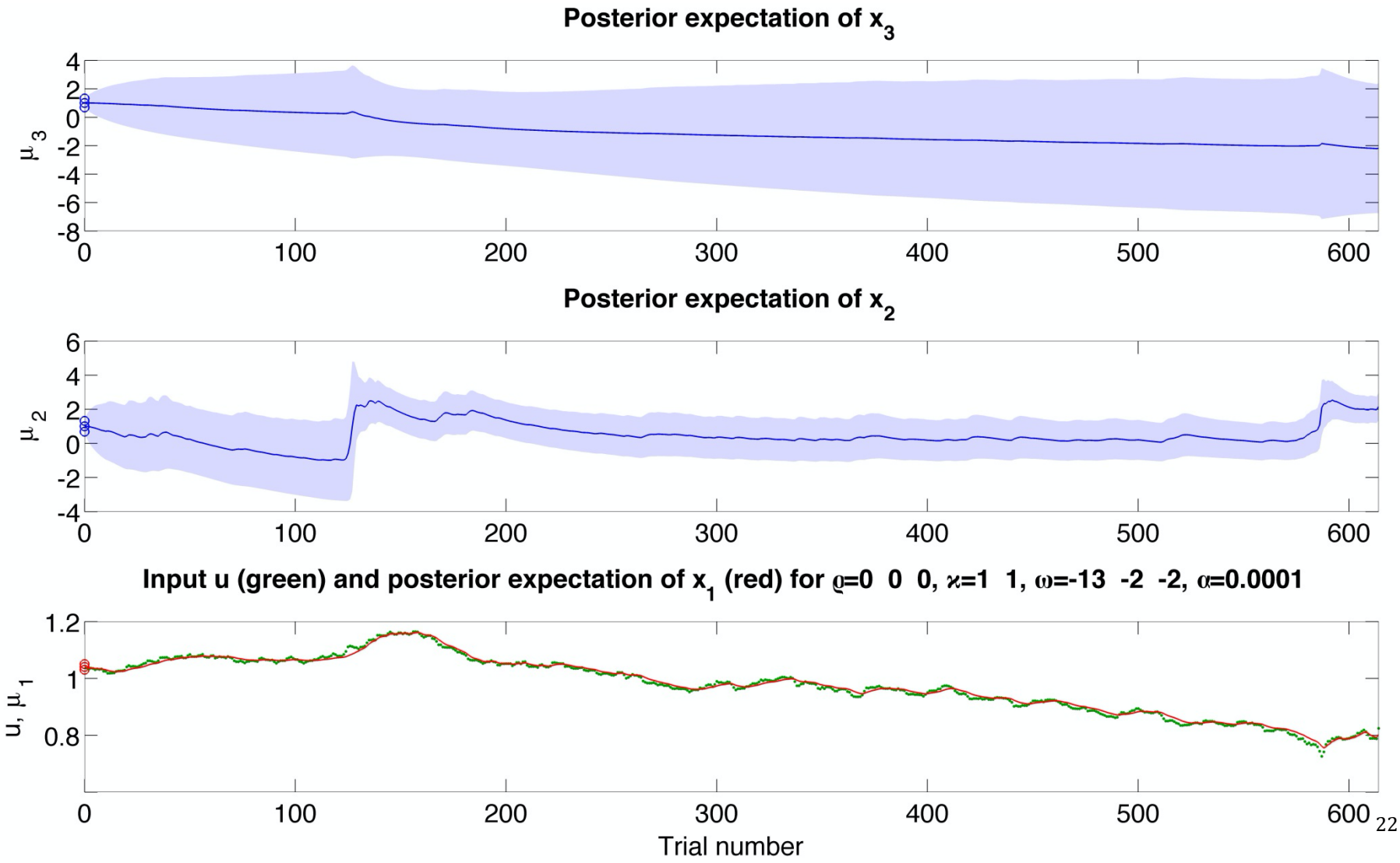
The familiar structure again – but now with a learning rate that is responsive to all kinds of uncertainty, including environmental (unexpected) uncertainty.

The learning rate ('Kalman gain') in the HGF

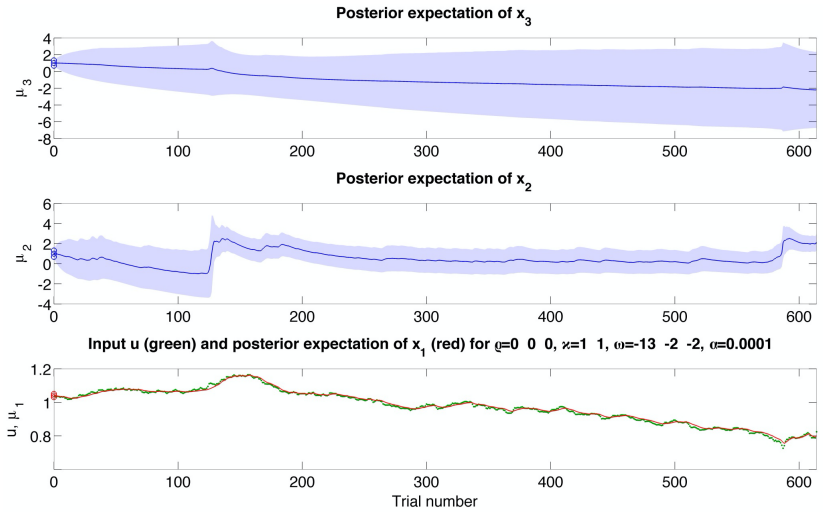
Unpacking the learning rate, we see:

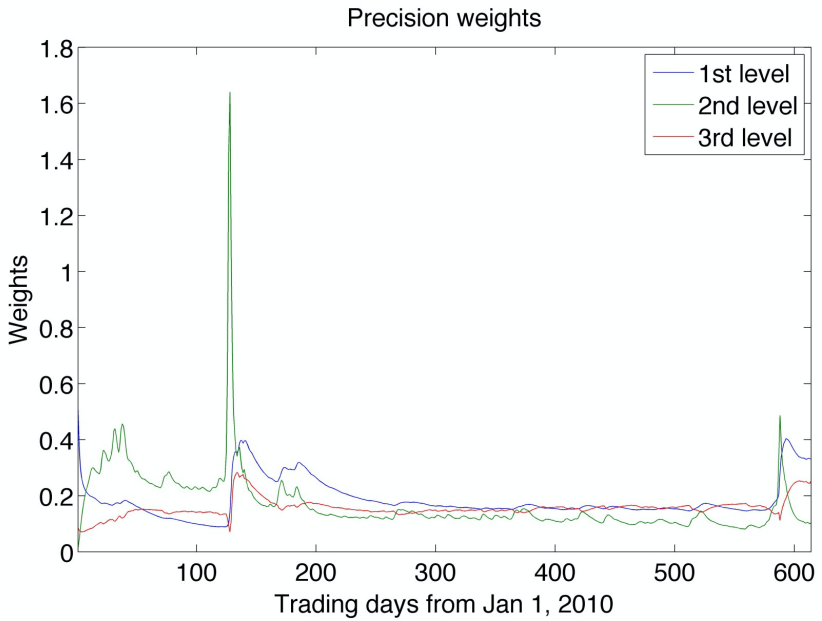


3-level HGF for continuous observations

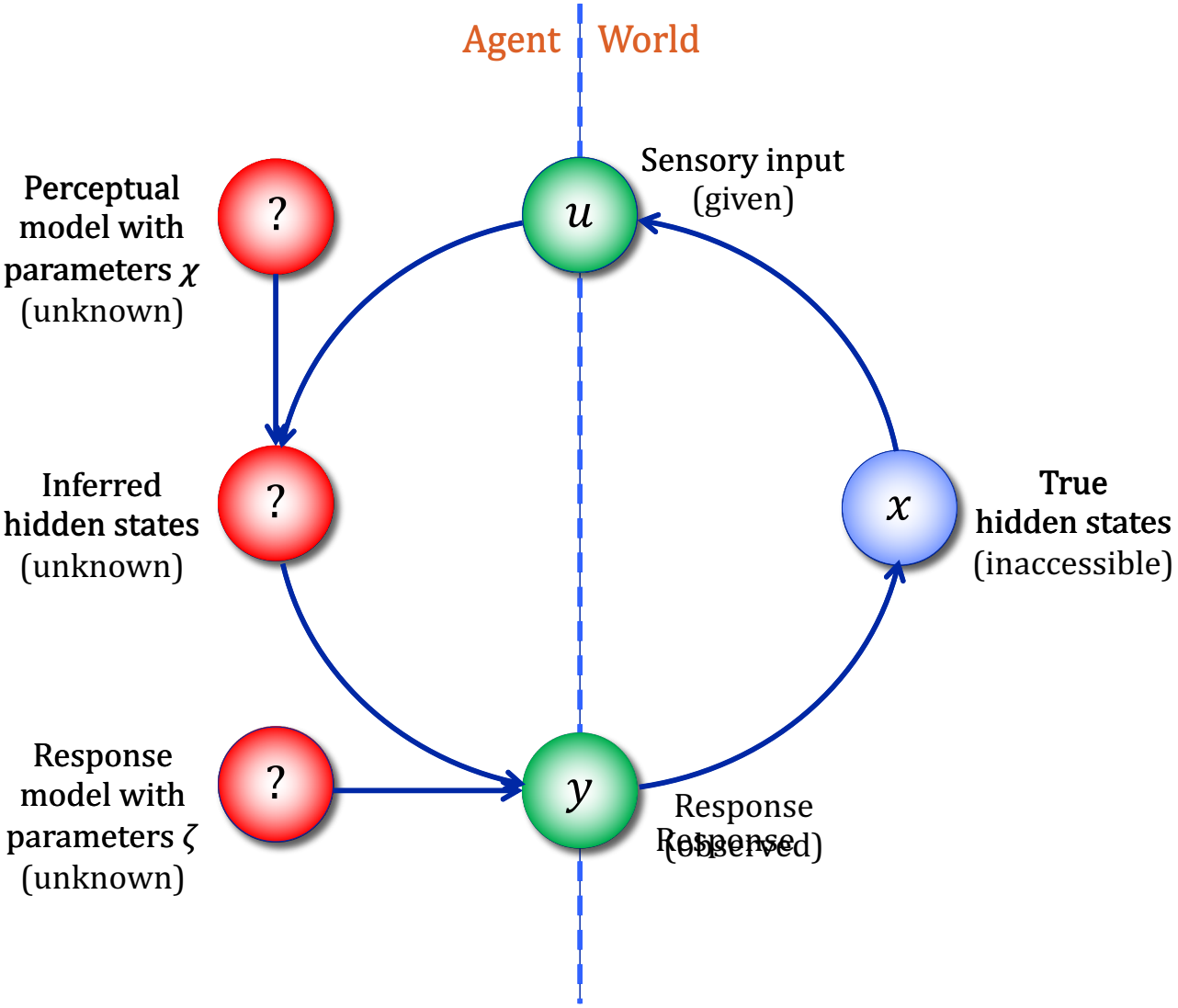


Example of precision weight trajectory

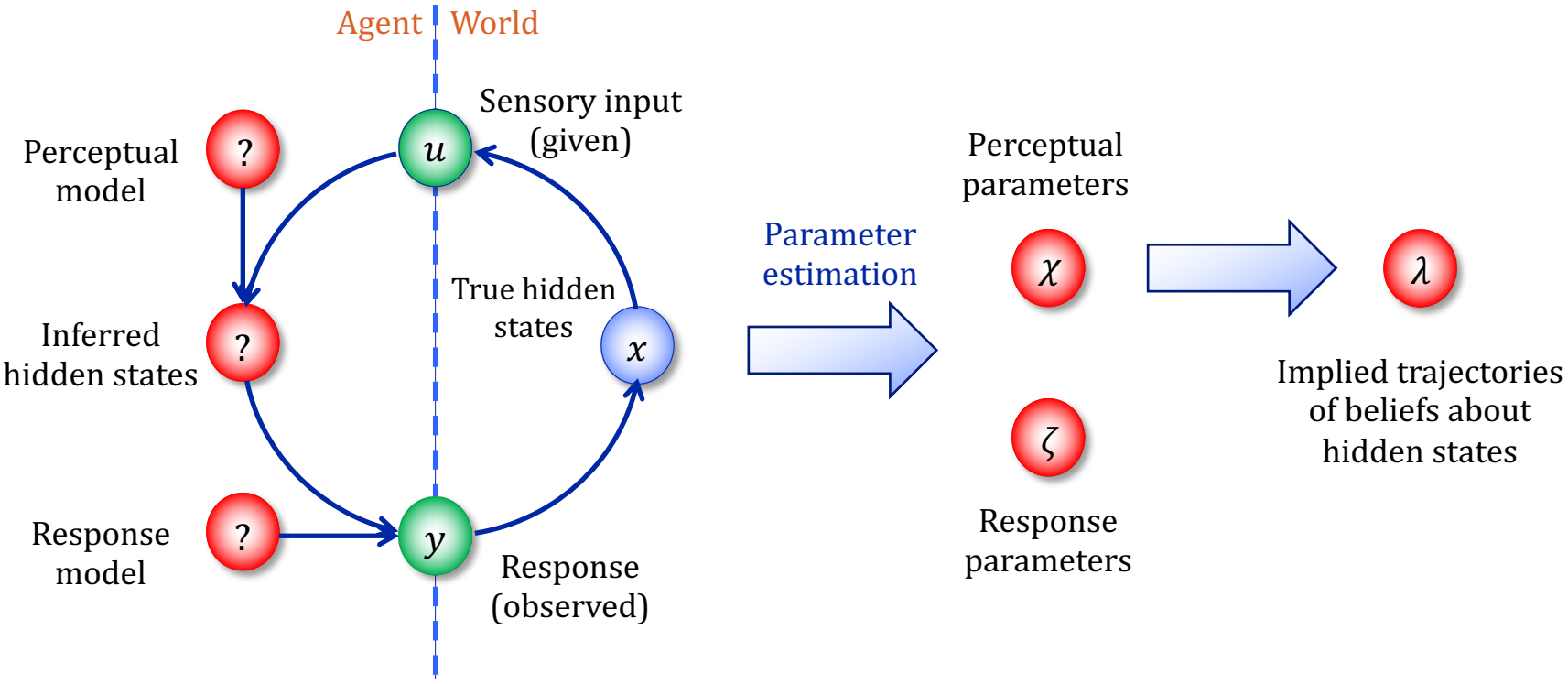




Application to experimental data

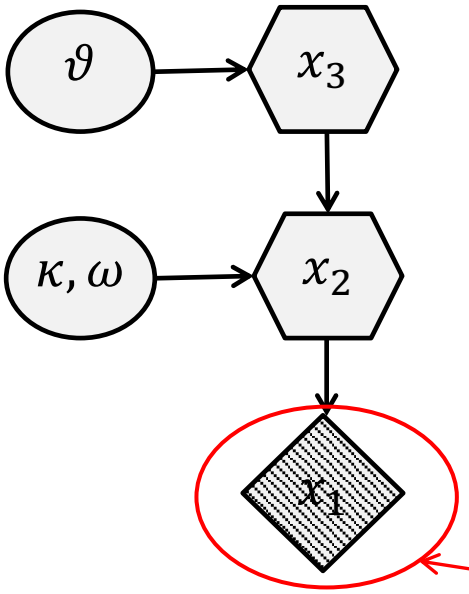


Application to experimental data: parameter estimation



Generative model for trial outcomes in the Iglesias et al. task:

3-level HGF for binary observations



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

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

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

Outcome: 1 if high/low tone
followed by face/house, 0 otherwise

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

Generative model, perceptual model, and decision model

- The *generative mode*l model describes (probabilistically) how the states x_i evolve in time: $x_i^{(k-1)} \longrightarrow x_i^{(k)}$
- Variationally inverting the generative gives us the *perceptual model* (also called *inference model* or *recognition model)*
- The perceptual model describes (deterministically, via update equations) how *beliefs* $\{\mu_i, \pi_i\}$ about states evolve in time: $\left\{\mu_i^{(k-1)}, \pi_i^{(k-1)}\right\} \longrightarrow \left\{\mu_i^{(k)}, \pi_i^{(k)}\right\}$
- The decision model (also called observation model or response model) describes (probabilistically) how beliefs are translated into observed actions: $\left\{\mu_i^{(k)}, \pi_i^{(k)}\right\}_{i=1,\ldots,l} \longrightarrow y^{(k)}$
- In the process of applying the HGF to data, we only need the perceptual and decision models. The generative model has already done its work: it has supplied the update equations of the perceptual model
- Both the perceptual and decision models have parameters that can be estimated individually for each dataset. Doing so requires defining priors for these parameters.

Parameter estimation with the HGF Toolbox

- Available at

  www.translationalneuromodeling.org/tapas

- Start with README, manual, and interactive demo

- Modular, extensible, Matlab-based

```
est2 = tapas_fitModel(sim2.y,...
                      usdchf,...
                      'tapas_hgf_config',...
                      'tapas_gaussian_obs_config',...
                      'tapas_quasinewton_optim_config');
```

```
Parameter estimates for the perceptual model:
    mu_0: [1.0352 1]
    sa_0: [3.7101e-05 0.0996]
     rho: [0 0]
      ka: 1
      om: [-12.8680 -1.8689]
    pi_u: 9.8449e+03
```

```
Parameter estimates for the observation model:
    ze: 2.3970e-05
```

Parameter estimation with the HGF Toolbox

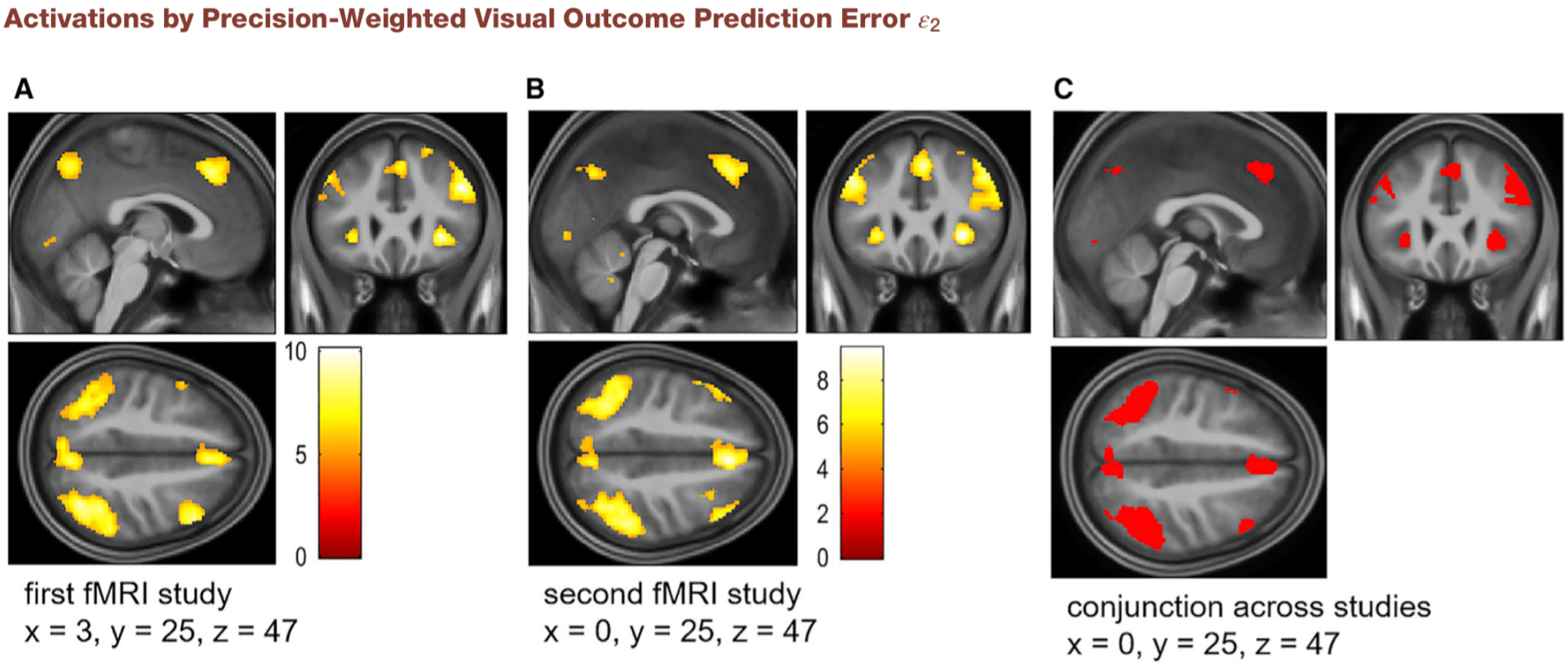


Figure 2. Whole-Brain Activations by ε_2

Activations by precision-weighted prediction errors 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 peak-level 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.

Iglesias et al, 2019 (Correction to Iglesias et al., 2013)

Parameter estimation with the HGF Toolbox

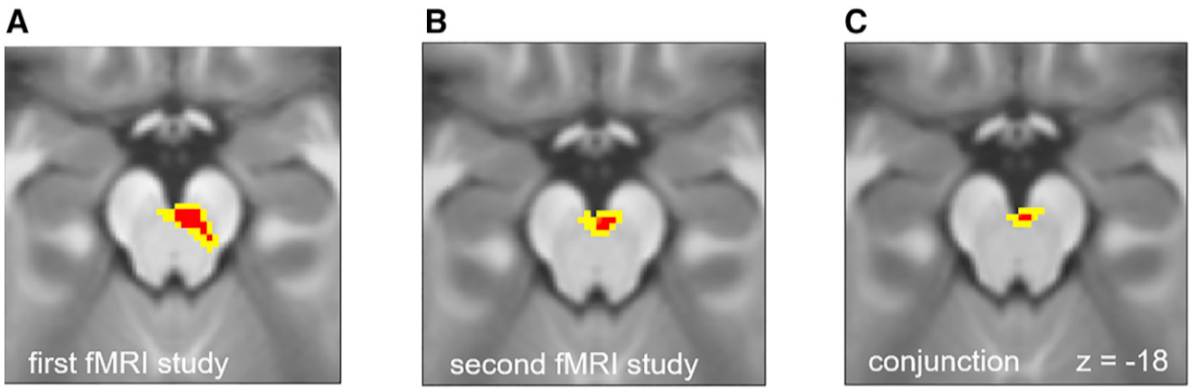


Figure 3. Midbrain Activation by ε_2

Activation of the dopaminergic VTA/SN by precision-weighted prediction errors about visual outcome, ε_2. The activation at $p < 0.05$ FWE peak-level corrected for the volume of our anatomical mask (comprising both dopaminergic and cholinergic brain structures: VTA/SN, PPT/LDT, and basal forebrain) is shown in red. The activation thresholded at $p < 0.001$ uncorrected is shown in yellow.
(A) Results from the first fMRI study. (B) Second fMRI study. (C) Conjunction (logical AND) across both studies.

Iglesias et al, 2019 (Correction to Iglesias et al., 2013)

Parameter estimation with the HGF Toolbox

Activations by Precision-Weighted Prediction Error about Stimulus Probabilities ε_3

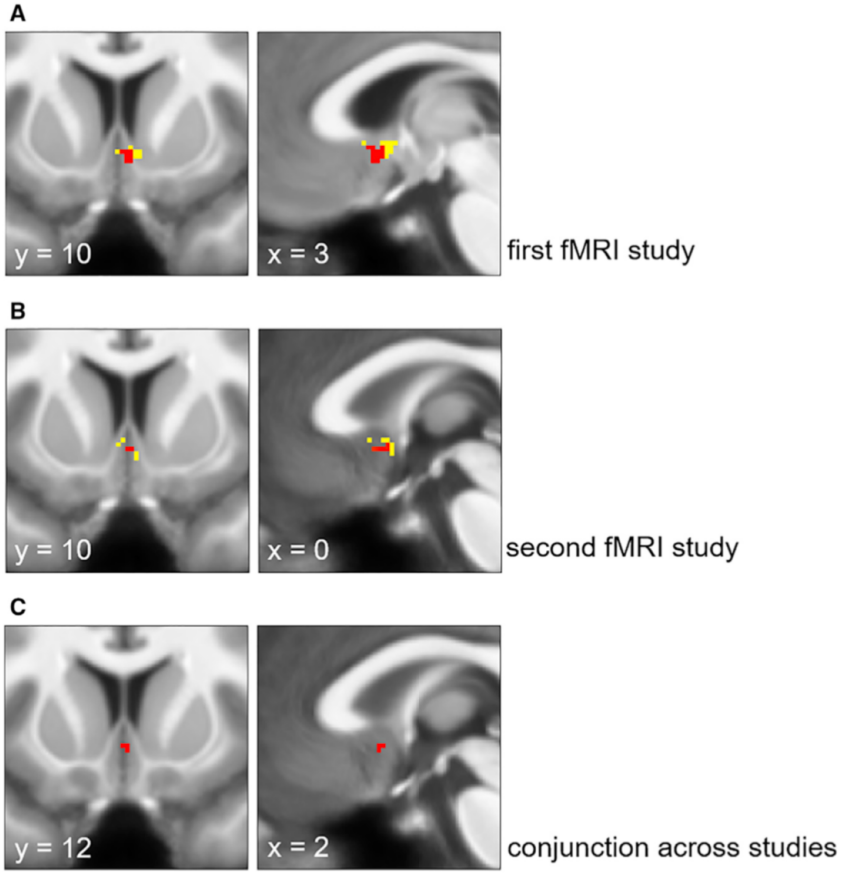
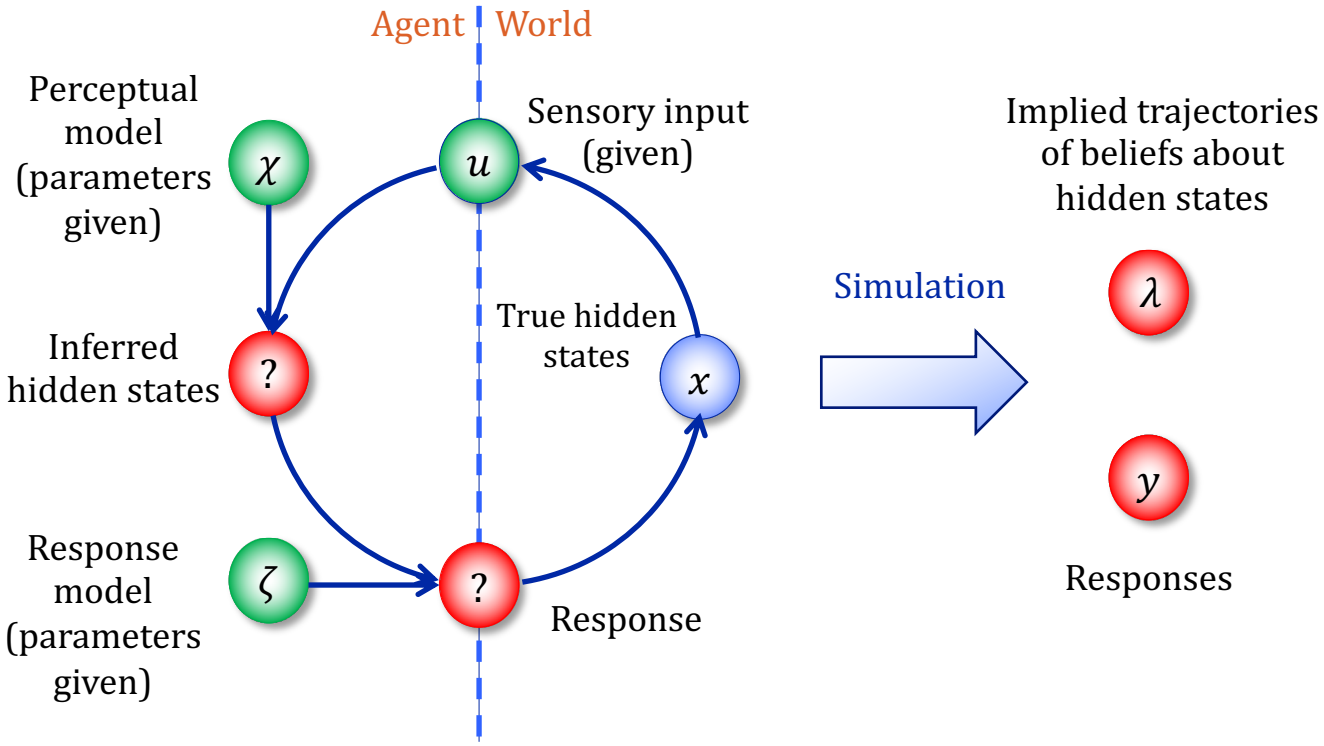


Figure 6. Basal Forebrain Activations by ε_3

Activation of the basal forebrain by 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 peak-level corrected for the entire anatomical mask (red) on the results thresholded at $p < 0.001$ (yellow; the yellow cluster also survives $p < 0.05$ FWE cluster-level correction for the entire anatomical mask). The anatomical mask comprised both dopaminergic and cholinergic brain structures: VTA/SN, PPT/LDT, and basal forebrain. (A) and (B) show results from the first (A: local maximum at x = 4, y = 12, z = –11, $t = 4.71$) and the second fMRI study (B: local maximum at x = 0, y = 10, z = –8, $t = 5.09$). (C) shows the conjunction analysis ("logical AND") across both studies. To ease visual comparison with Iglesias et al. (2013), the figure sections (x and y coordinates are indicated on each panel) are not located at the local maxima but correspond closely to those in Iglesias et al. (2013).

Iglesias et al, 2019
(Correction to Iglesias et al., 2013)

Generation of new data: simulation



Remarks on parameter recovery

- *Parameter recovery* means the ability to estimate the 'ground truth' parameter values put into a simulation
- This is never possible to do exactly because simulation is stochastic
- The stochastic nature of the simulation means that the parameter value best able to explain the simulated data is not the same as that used in the simulation. By nature, the simulation only noisily represents the 'ground truth' parameter values.
- It is therefore misleading to call the value used in the simulation 'true'. One could just as well argue that the estimated value is the 'true' one since it best explains the data.
- In order to get an idea of how exactly a parameter can be recovered, run many (on the order of tens to hundreds) simulations and look at the distribution of parameter estimates.
- **Estimating a parameter that isn't well recovered from simulations can still make sense because adjusting its value can improve the model by leading to a better explanation of the data**

Remarks on model comparison / model selection

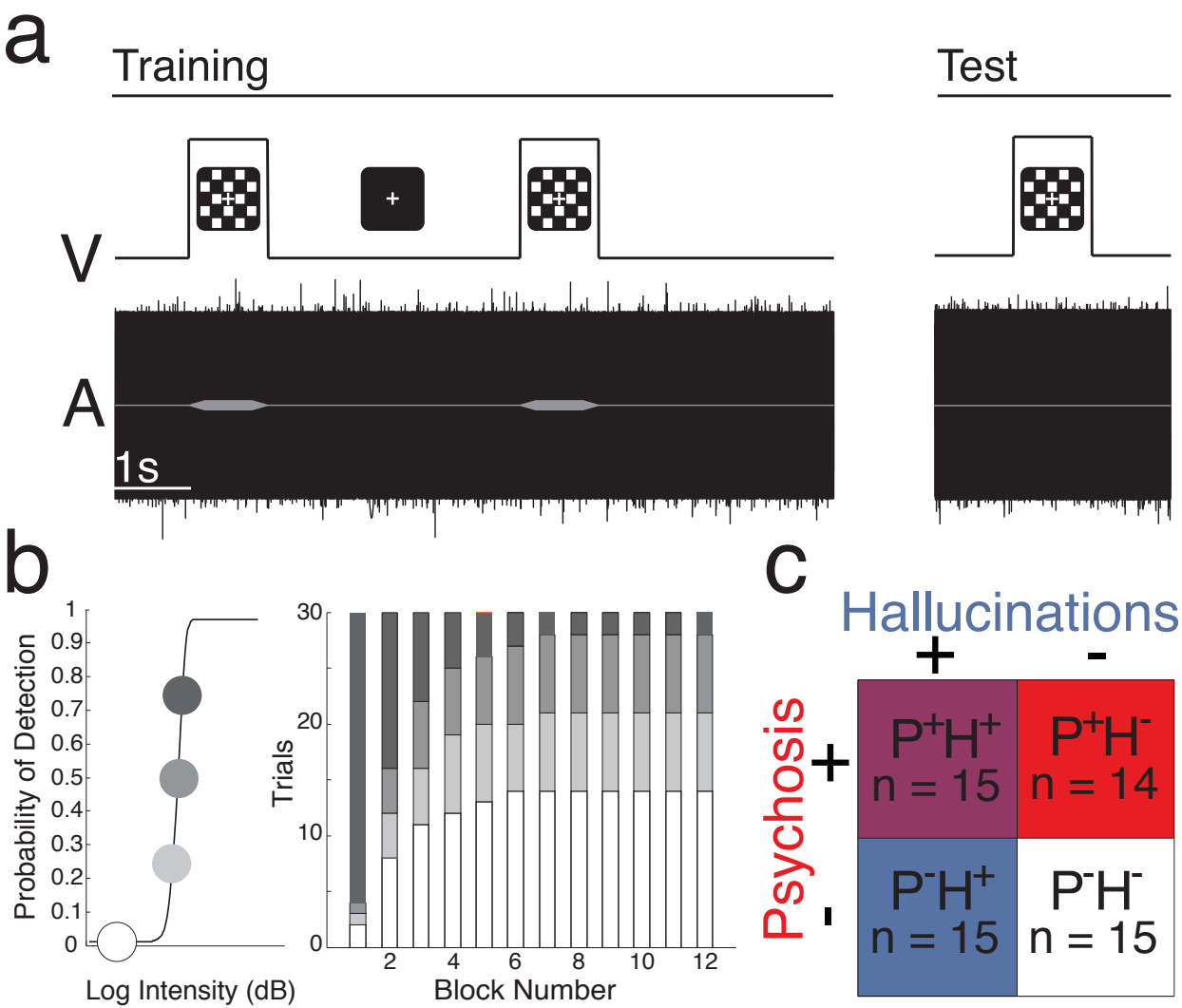
- There is a range of scores that help in choosing a well-performing model: AIC (Akaike information criterion), BIC (Bayesian information criterion), Bayes factors, LME (log-model evidence), free energy, etc.
- Each model gets a particular score (which is on its own uninterpretable!)
- The difference in score between models is what counts
- However, model selection is not straightforward. AIC and BIC penalize complexity based on simple heuristics, which may not reflect complexity accurately. LME is better on that count, but is very sensitive to the modeller's choice of priors.
- **Three important considerations:**

  1. **Does the model allow me to answer my question of interest?**
  2. **Does the *prior predictive* distribution of observations make sense?**
  3. **Does the *posterior predictive* distribution of observations make sense?**

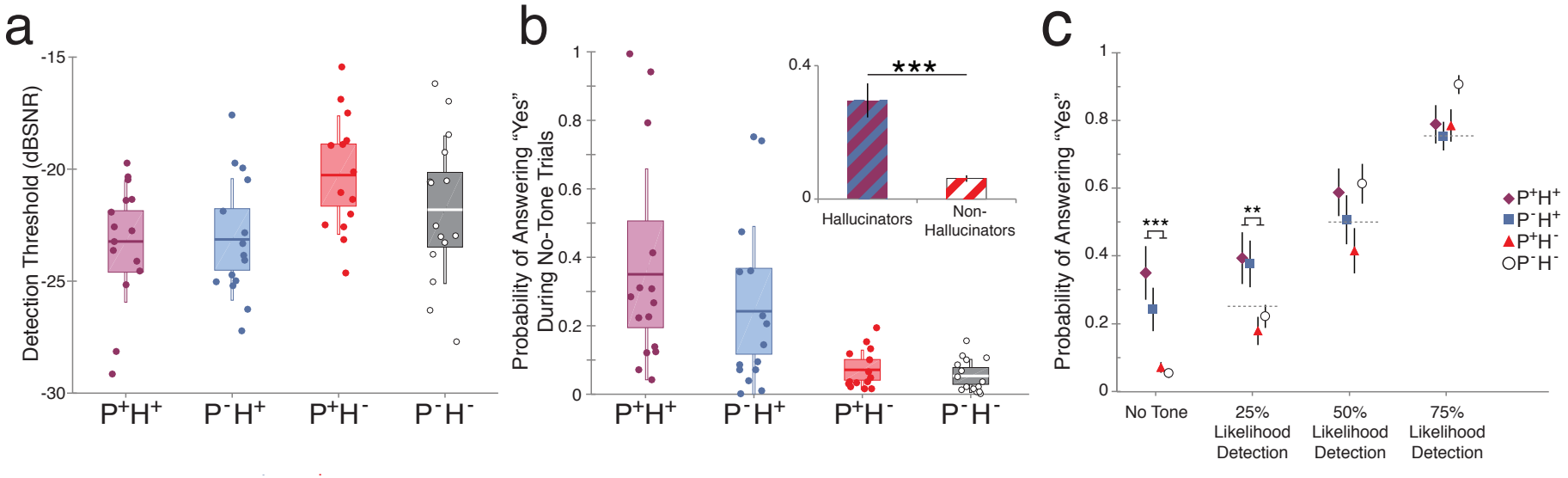
  When the answer to all three is yes, the model is fine.

Thanks

Conditioned hallucinations (Powers, Mathys, & Corlett, Science, 2017)



Conditioned hallucinations



Conditioned hallucinations

- Belief/percept formation model specifically created for this task
- Probability that the subject will respond "yes" to detection on a given trial:

$$P(\text{"yes"}|belief) = sigmoid(belief)$$

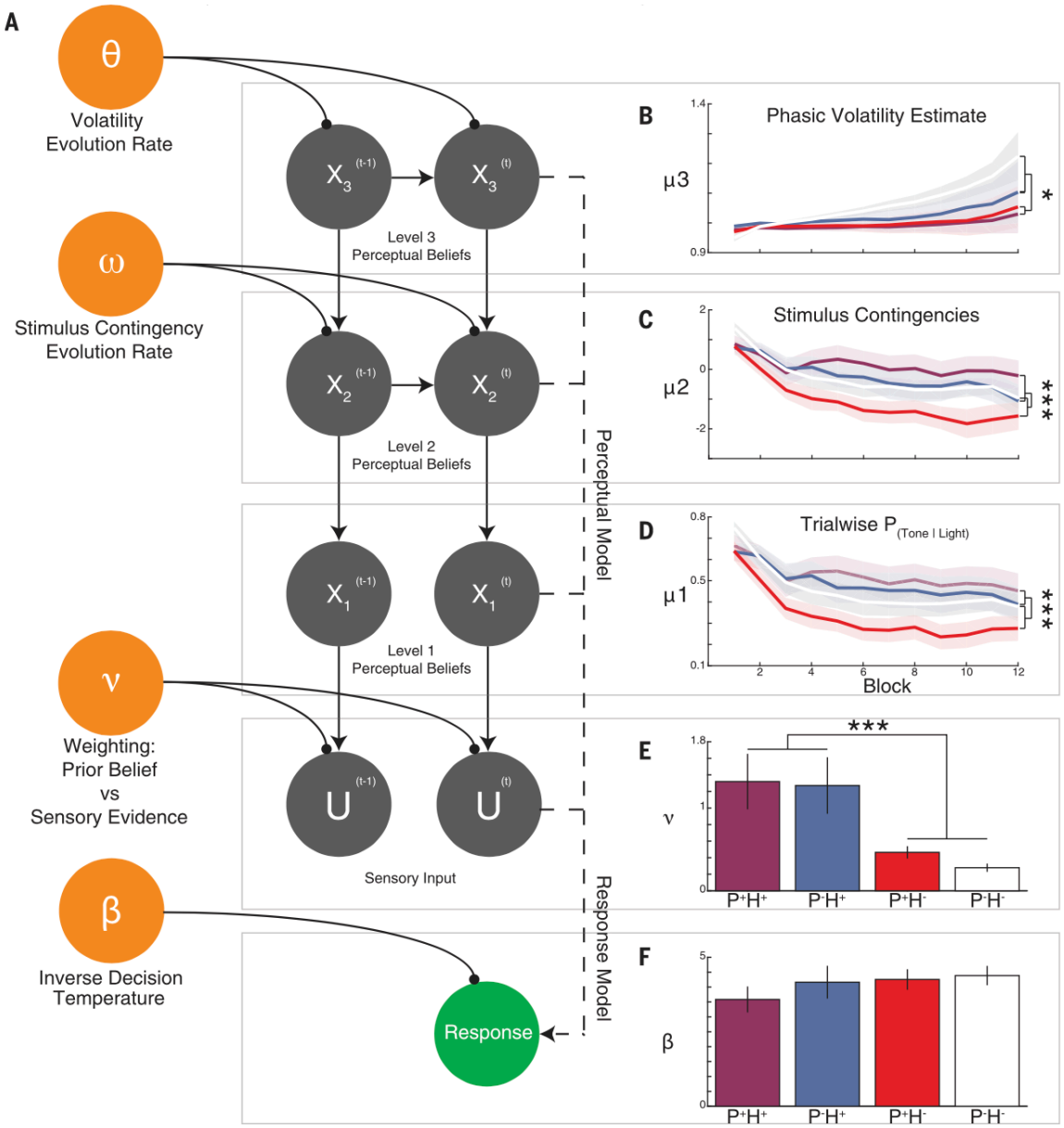
- $belief$ is formalized as the Bayesian posterior mean of a beta distribution

$$belief = prior + \frac{1}{1+\nu}(input - prior)$$

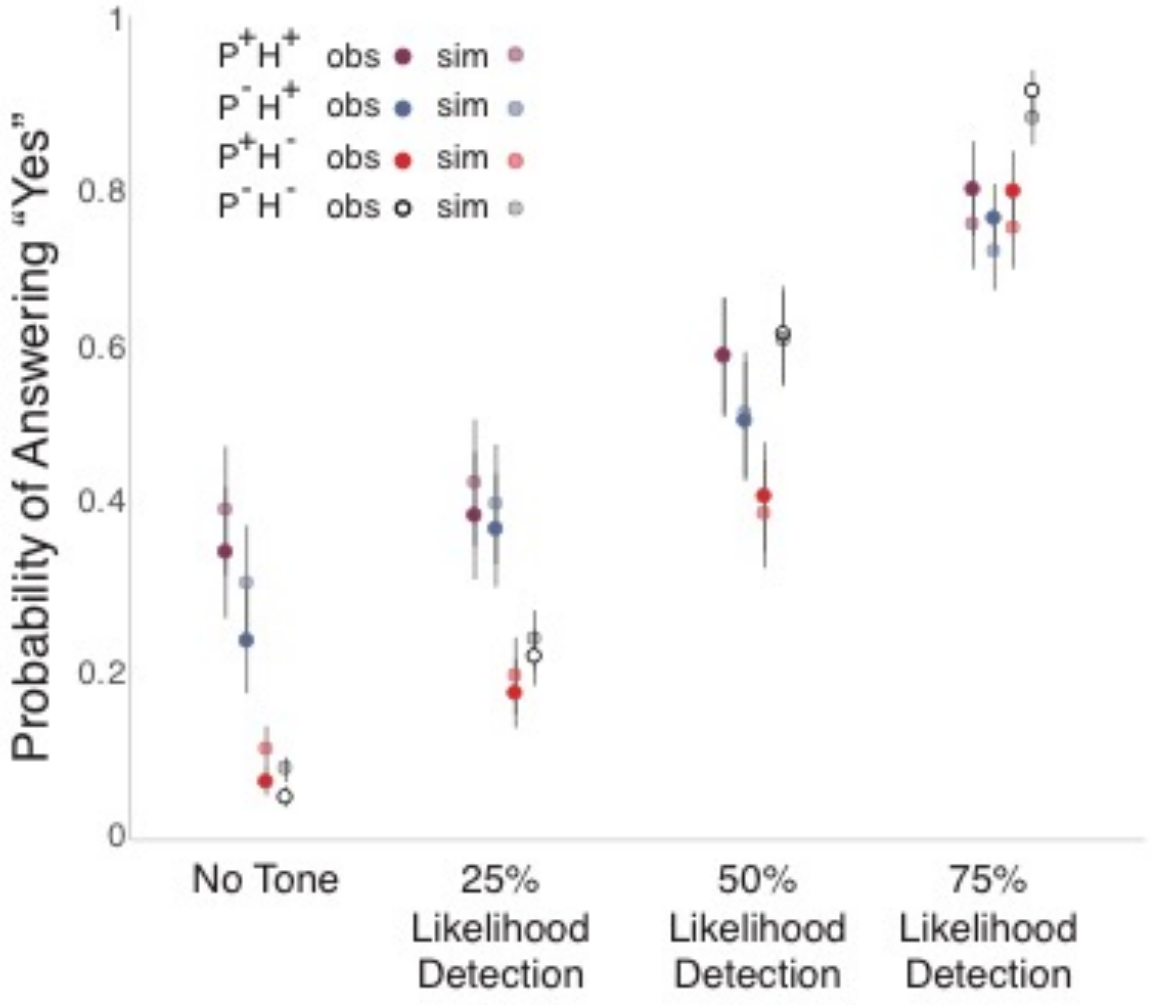
where

- $input$ is given by experimental design: the true positive rate of the tone presented without light at each trial: 25%, 50%, or 75%
- $prior$ is the prior from learning using the HGF: $\hat{\mu}_1$
- ν is a subject-specific parameter indicating the relative weight of the prior compared to the input.
- For $\nu = 1$, prior and input have equal weight; for $\nu > 1$ the prior has more weight than the input; and for $\nu < 1$ the input has more weight than the prior.

Conditioned hallucinations



Conditioned hallucinations



Conditioned hallucinations