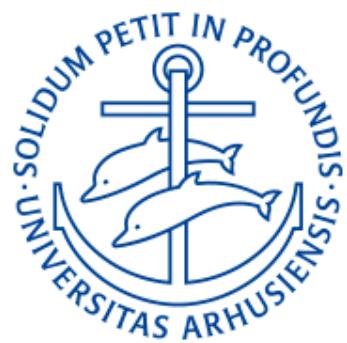


Data Science 12

Chris Mathys



Master's Degree Programme in Cognitive Science
Spring 2022

A 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

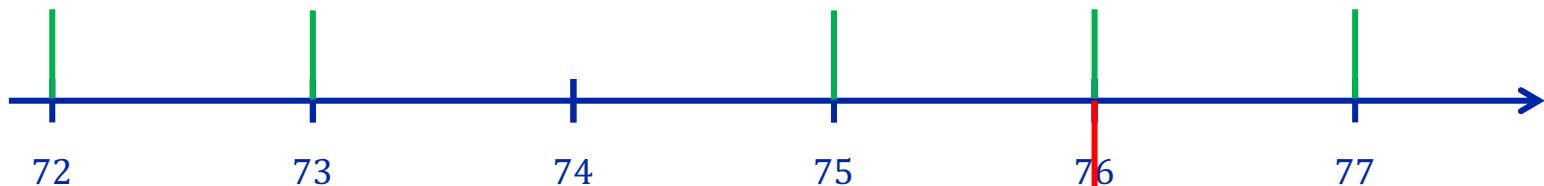
Proof of sequential update equation:

$$\begin{aligned}\bar{u}_{n+1} &= \frac{1}{n+1} \sum_{i=1}^{n+1} u_i = \frac{1}{n+1} \left(u_{n+1} + n \cdot \frac{1}{n} \sum_{i=1}^n u_i \right) = \\ &= \frac{1}{n+1} (u_{n+1} + n\bar{u}_n) = \frac{1}{n+1} (u_{n+1} - \bar{u}_n + (n+1)\bar{u}_n) \\ &= \bar{u}_n + \frac{1}{n+1} (u_{n+1} - \bar{u}_n)\end{aligned}$$

q.e.d.

Updating the mean of a series of observations

The sequential updates in our example now look like this:



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

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

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

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

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

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

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

new input

prediction error

prediction

weight (learning rate)

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:

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

Updates in a simple Gaussian model

Here's how the updates to the sufficient 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).

Reduction to mean updating

Reminder (Gaussian update):

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

Reducing by π_ε the fraction of precisions that make the learning rate, we get

$$\mu_{x|u} = \mu_x + \frac{1}{\frac{\pi_x}{\pi_\varepsilon} + 1}(u - \mu_x)$$

As we shall see, this is the equation for updating an arithmetic mean, but with the number of observations n replaced by $\frac{\pi_x}{\pi_\varepsilon}$.

This shows that Bayesian inference on the mean of a Gaussian distribution entails nothing more than updating the arithmetic mean of observations with $\frac{\pi_x}{\pi_\varepsilon} =: \nu$ as a proxy for the number of prior observations, i.e. for the **weight of the prior relative to the observation**.

Generalization to all exponential families of distributions

Many of the most widely used probability distributions are families of exponential distributions.

For example, the Gaussian distribution is an exponential family of distributions (and so are the beta, gamma, binomial, Bernoulli, multinomial, categorical, Dirichlet, Wishart, Gaussian-gamma, log-Gaussian, multivariate Gaussian, Poisson, and exponential distributions, among others). This means it can be written the following way:

$$p(\mathbf{x}|\boldsymbol{\vartheta}) = h(\mathbf{x}) \exp(\boldsymbol{\eta}(\boldsymbol{\vartheta}) \cdot \mathbf{T}(\mathbf{x}) - A(\boldsymbol{\vartheta})) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x - \mu)^2}{2\sigma^2}\right)$$

with

$$\mathbf{x} = x, \quad \boldsymbol{\vartheta} = (\mu, \sigma)^T, \quad h(\mathbf{x}) = \frac{1}{\sqrt{2\pi}}, \quad \boldsymbol{\eta}(\boldsymbol{\vartheta}) = \left(\frac{\mu}{\sigma}, -\frac{1}{2\sigma}\right)^T, \quad \mathbf{T}(\mathbf{x}) = (x, x^2)^T, \quad A(\boldsymbol{\vartheta}) = \frac{\mu^2}{\sigma^2} + \frac{\ln \sigma}{2}$$

This allows us to look at Bayesian belief updating in a very general way for all exponential families of distributions.

Generalization to all exponential families of distributions (Mathys, 2016; 2020)

Our likelihood is an exponential family in its general form:

$$p(\mathbf{x}|\boldsymbol{\vartheta}) = h(\mathbf{x}) \exp(\boldsymbol{\eta}(\boldsymbol{\vartheta}) \cdot \mathbf{T}(\mathbf{x}) - A(\boldsymbol{\vartheta}))$$

The vector $\mathbf{T}(\mathbf{x})$ (a function of the observation \mathbf{x}) is called the sufficient statistic.

For the prior, we may assume that we have made v observations with sufficient statistic $\boldsymbol{\xi}$:

$$p(\boldsymbol{\vartheta}|\boldsymbol{\xi}, v) = z(\boldsymbol{\xi}, v) \exp(v(\boldsymbol{\eta}(\boldsymbol{\vartheta}) \cdot \boldsymbol{\xi} - A(\boldsymbol{\vartheta}))) \quad (\text{where } z(\boldsymbol{\xi}, v) \text{ is a normalization constant})$$

It then turns out that the posterior has the same form, but with an updated $\boldsymbol{\xi}$ and v replaced with $v + 1$:

$$p(\boldsymbol{\vartheta}|\mathbf{x}, \boldsymbol{\xi}, v) = z(\boldsymbol{\xi}', v + 1) \exp((v + 1)(\boldsymbol{\eta}(\boldsymbol{\vartheta}) \cdot \boldsymbol{\xi}' - A(\boldsymbol{\vartheta})))$$

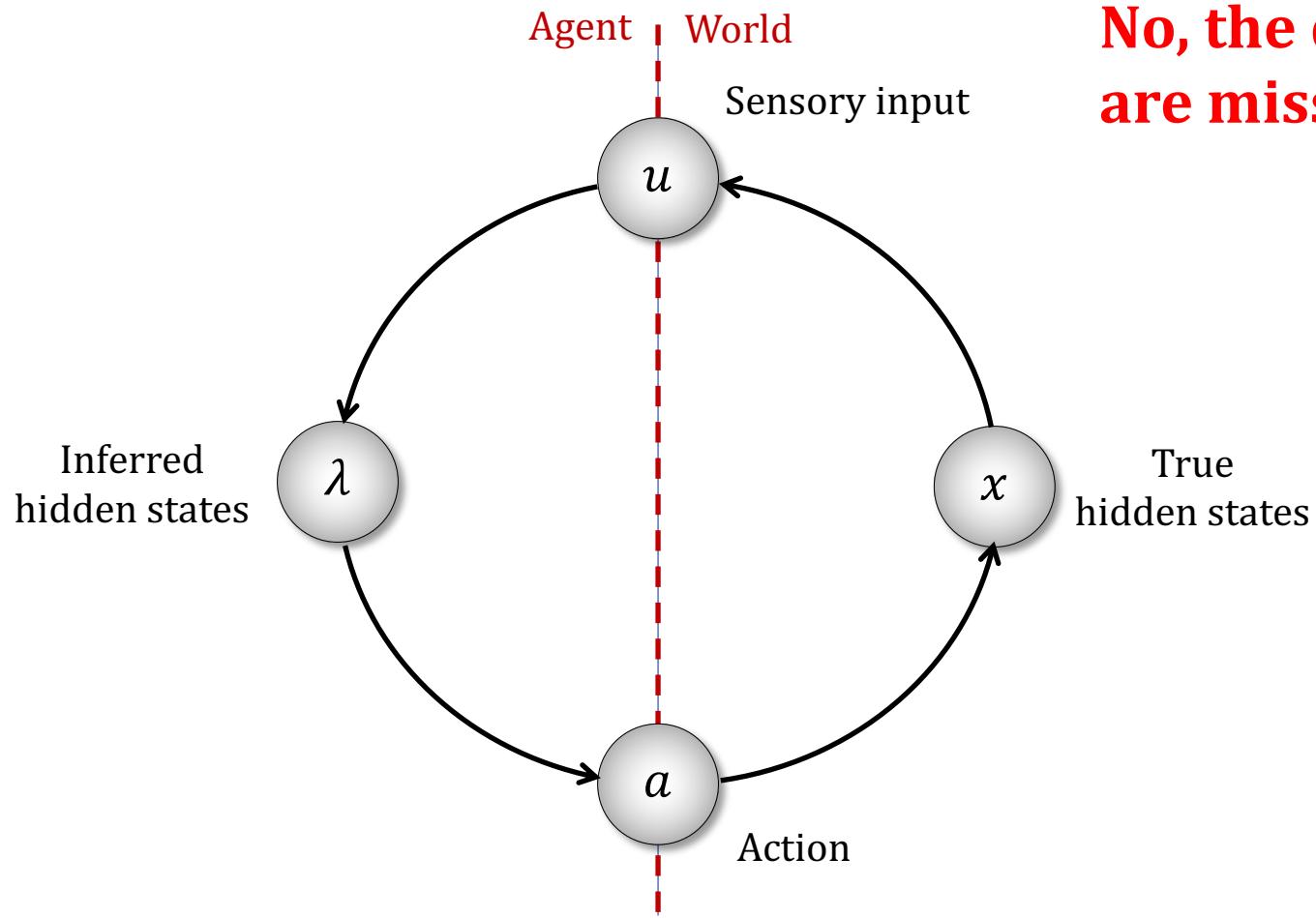
$$\boldsymbol{\xi}' = \boldsymbol{\xi} + \frac{1}{v + 1}(\mathbf{T}(\mathbf{x}) - \boldsymbol{\xi})$$

Proof of the update equation

$$\begin{aligned}
& \overbrace{p(\boldsymbol{\vartheta}|\mathbf{x}, \xi, \nu)}^{\text{posterior}} \propto \overbrace{p(\mathbf{x}|\boldsymbol{\vartheta})}^{\text{likelihood}} \overbrace{p(\boldsymbol{\vartheta}|\xi, \nu)}^{\text{prior}} \\
& = h(\mathbf{x}) \exp(\boldsymbol{\eta}(\boldsymbol{\vartheta}) \cdot \mathbf{T}(\mathbf{x}) - A(\boldsymbol{\vartheta})) z(\xi, \nu) \exp(\nu(\boldsymbol{\eta}(\boldsymbol{\vartheta}) \cdot \xi - A(\boldsymbol{\vartheta}))) \\
& \propto \exp(\boldsymbol{\eta}(\boldsymbol{\vartheta}) \cdot (\mathbf{T}(\mathbf{x}) + \nu \xi) - (\nu + 1)A(\boldsymbol{\vartheta})) \\
& = \exp\left((\nu + 1)\left(\boldsymbol{\eta}(\boldsymbol{\vartheta}) \cdot \frac{1}{\nu + 1}(\mathbf{T}(\mathbf{x}) + \nu \xi) - A(\boldsymbol{\vartheta})\right)\right) \\
& = \exp\left((\nu + 1)\left(\boldsymbol{\eta}(\boldsymbol{\vartheta}) \cdot \left(\xi + \frac{1}{\nu + 1}(\mathbf{T}(\mathbf{x}) + \nu \xi - (\nu + 1)\xi)\right) - A(\boldsymbol{\vartheta})\right)\right) \\
& = \exp\left((\nu + 1)\left(\boldsymbol{\eta}(\boldsymbol{\vartheta}) \cdot \underbrace{\left(\xi + \frac{1}{\nu + 1}(\mathbf{T}(\mathbf{x}) - \xi)\right)}_{=\xi'} - A(\boldsymbol{\vartheta})\right)\right) \\
& \Rightarrow p(\boldsymbol{\vartheta}|\mathbf{x}, \xi, \nu) = z(\xi', \nu') \exp(\nu'(\boldsymbol{\eta}(\boldsymbol{\vartheta}) \cdot \xi' - A(\boldsymbol{\vartheta}))) \\
& \quad \text{with } \nu' := \nu + 1, \quad \xi' := \xi + \frac{1}{\nu + 1}(\mathbf{T}(\mathbf{x}) - \xi)
\end{aligned}$$

q.e.d.

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 equations 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].

How are we treating observations in Rescorla-Wagner learning?

Rewriting the RW learning rule reveals that q_n is an average weighted by α of the observation x_n and the previous value q_{n-1}

$$\begin{aligned} q_n &= q_{n-1} + \alpha(u_n - q_{n-1}) \\ &= (1 - \alpha)q_{n-1} + \alpha u_n \\ &= (1 - \alpha)((1 - \alpha)q_{n-2} + \alpha u_{n-1}) + \alpha u_n \\ &= (1 - \alpha)^2 q_{n-2} + (1 - \alpha)\alpha u_{n-1} + \alpha u_n \\ &= (1 - \alpha)^3 q_{n-3} + (1 - \alpha)^2 \alpha u_{n-2} + (1 - \alpha)\alpha u_{n-1} + \alpha u_n \\ &= (1 - \alpha)^n q_0 + \alpha \sum_{i=1}^n (1 - \alpha)^{n-i} u_i \end{aligned}$$

Recursively unpacking the content of q_n reveals that **observations u_i are exponentially discounted into the past.**

How are we treating observations in Rescorla-Wagner learning?

Taking $q_0 = 0$ and $\gamma := 1 - \alpha$, we get

$$\begin{aligned} q_n &= q_{n-1} + \alpha(u_n - q_{n-1}) \\ &= (1 - \alpha)q_{n-1} + \alpha u_n \\ &= \alpha \sum_{i=1}^n (1 - \alpha)^{n-i} u_i \\ &= (1 - \gamma) \sum_{i=1}^n \gamma^{n-i} u_i \\ &= (1 - \gamma) \sum_{i=0}^{n-1} \gamma^i u_{n-i} \end{aligned}$$

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. But...

... if the learning rate is supposed to reflect uncertainty in Bayesian inference, then how do we

- (a) know that α reflects the right level of uncertainty at any one time, and
- (b) account for changes in uncertainty if α is constant?

What we really need is an adaptive learning that accurately reflects uncertainty.

Needed: an adaptive learning rate that accurately reflects uncertainty

This requires us to think a bit more about what kinds of uncertainty we are dealing with.

A possible taxonomy of uncertainty is (cf. Yu & Dayan, 2003; Payzan-LeNestour & Bossaerts, 2011):

- (a) **outcome uncertainty** that remains unaccounted for by the model, called *risk* by economists (π_ε in our Bayesian example); this uncertainty remains even when we know all parameters exactly,
- (b) **informational** or *expected* uncertainty about the value of model parameters ($\pi_{\vartheta|x}$ in the Bayesian example),
- (c) **environmental** or *unexpected* uncertainty owing to changes in model parameters (not accounted for in our Bayesian example, hence unexpected).

An adaptive learning rate that accurately reflects uncertainty

Various efforts have been made to come up with an adaptive learning rate:

- Kalman (1960)
- Sutton (1992)
- Nassar et al. (2010)
- Payzan-LeNestour & Bossaerts (2011)
- Mathys et al. (2011)
- Wilson et al. (2013)

We will look at two of these:

- **The Kalman filter** is optimal for linear dynamical systems, but realistic data usually require non-linear models.
- Mathys et al. use **a generic non-linear hierarchical Bayesian model** that allows us to derive update equations that are optimal in the sense that they minimize surprise.

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**:

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

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

- Combining this with the **prior**

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

Dealing with nonstationary environments: the Kalman filter

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

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

with

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

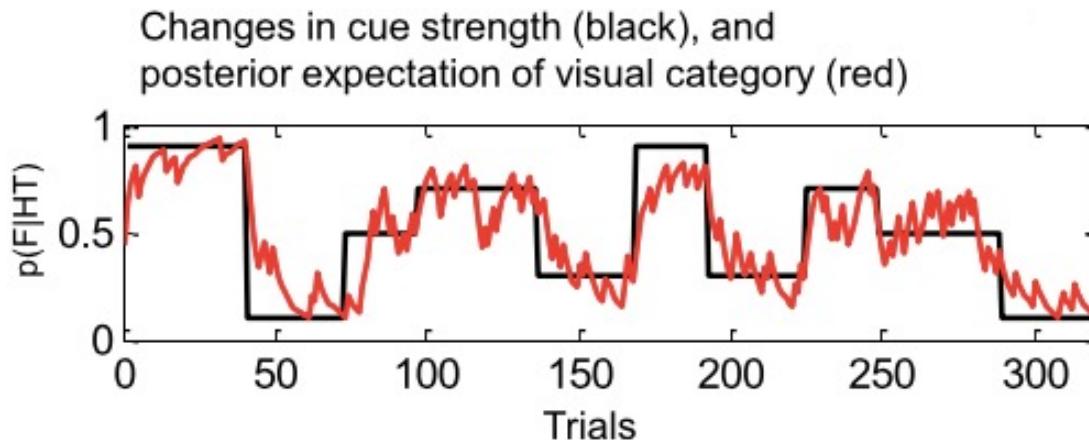
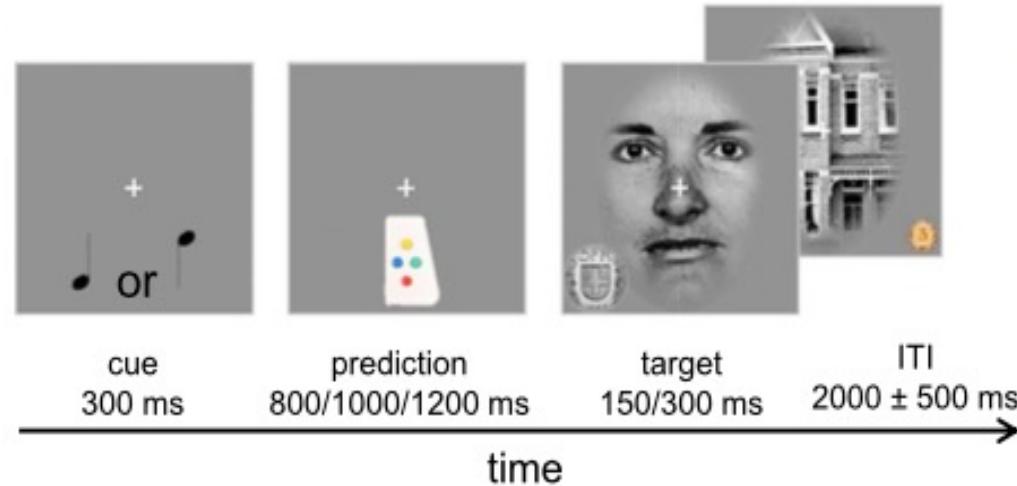
$$\begin{aligned}\mu_x^{(k)} &= \mu_x^{(k-1)} + \frac{\hat{\pi}_u}{\pi_x^{(k)}} (u^{(k)} - \mu_x^{(k-1)}) \\ &= \mu_x^{(k-1)} + \frac{\hat{\pi}_u}{\frac{1}{\sigma_x^{(k-1)} + \vartheta} + \hat{\pi}_u} (u^{(k)} - \mu_x^{(k-1)})\end{aligned}$$

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?

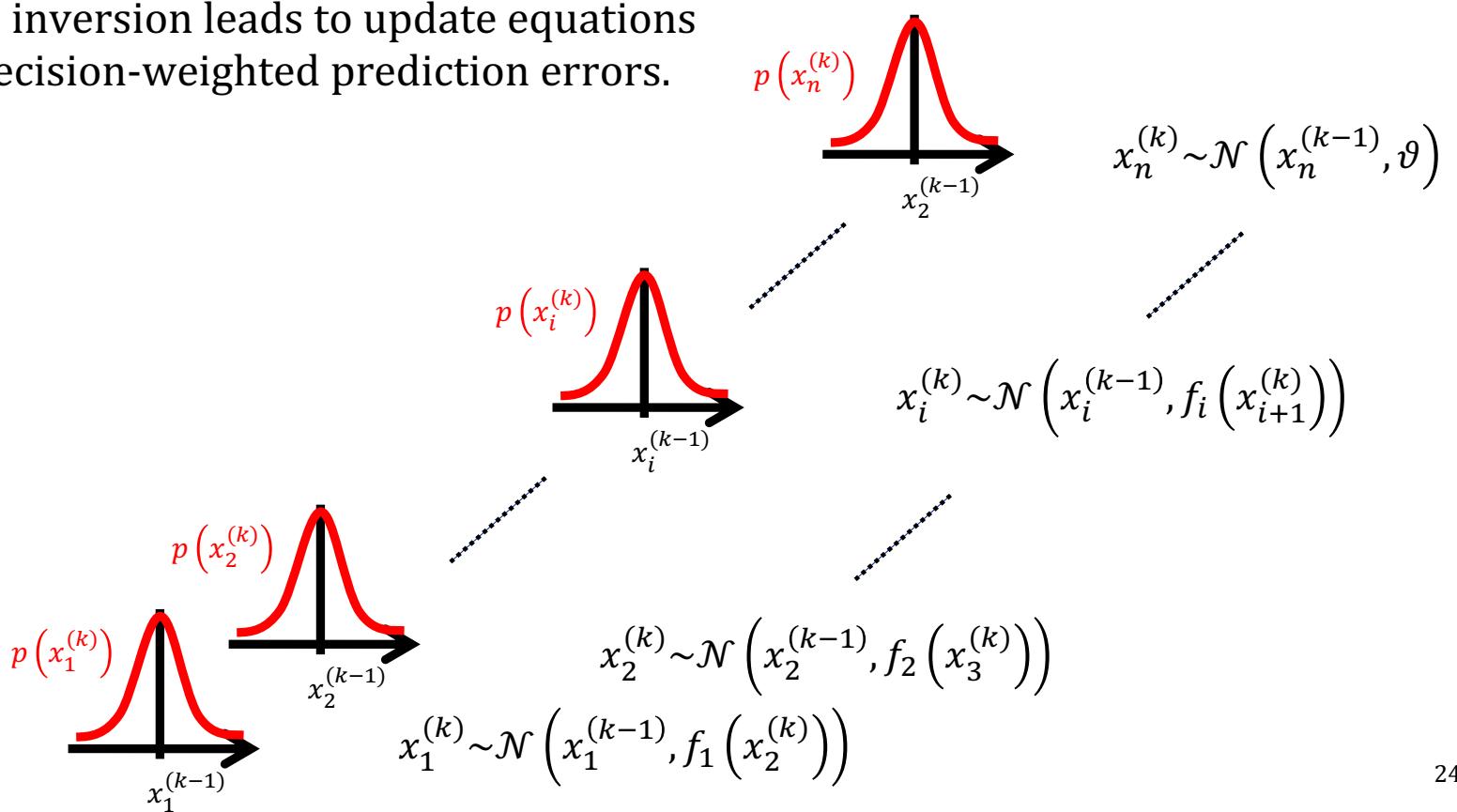
Task of Igloos



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.



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:

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

Precisions determine learning rate

Prediction error

The diagram illustrates the update equation for the mean $\Delta\mu_i$. It features two overlapping circles: an orange circle labeled "Precisions determine learning rate" containing the term $\frac{\hat{\pi}_{i-1}}{\pi_i}$, and a purple circle labeled "Prediction error" containing the term δ_{i-1} . The two circles overlap, representing their product.

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

Precision-weighting of volatility updates

Comparison to the simple non-hierarchical Bayesian update:

HGF: $\mu_i^{(k)} = \mu_i^{(k-1)} + \frac{1}{2} \kappa_{i-1} v_{i-1}^{(k)} \cdot \frac{\hat{\pi}_{i-1}^{(k)}}{\pi_i^{(k)}} \cdot \delta_{i-1}^{(k)}$

Precision-weighted
prediction error

Simple Gaussian: $\mu_{\vartheta|x} = \mu_{\vartheta} + \frac{\pi_{\varepsilon}}{\pi_{\vartheta|x}} (x - \mu_{\vartheta})$

Updates at the outcome 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 in the HGF

Unpacking the learning rate, we see:

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

outcome uncertainty

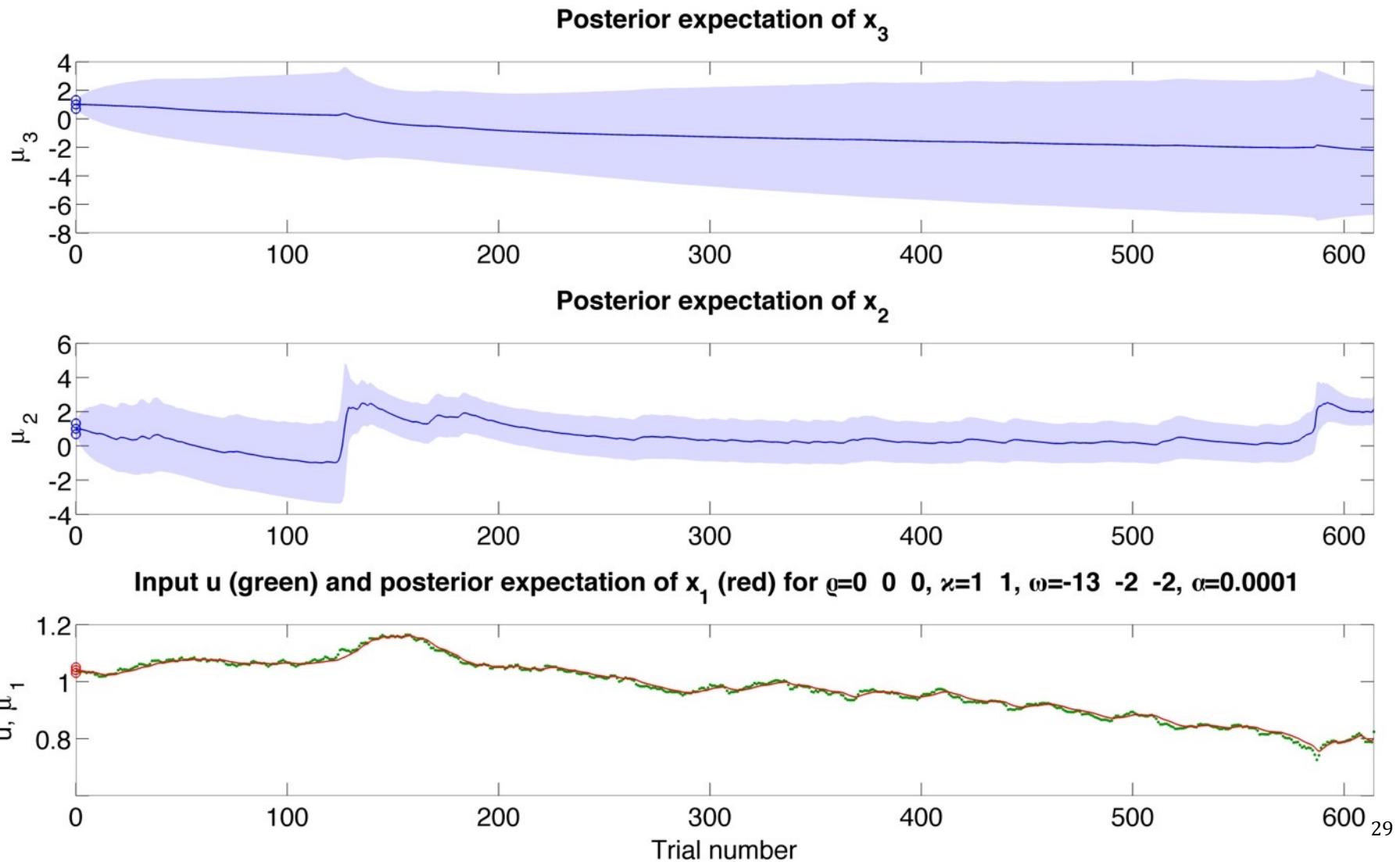
informational uncertainty

environmental uncertainty

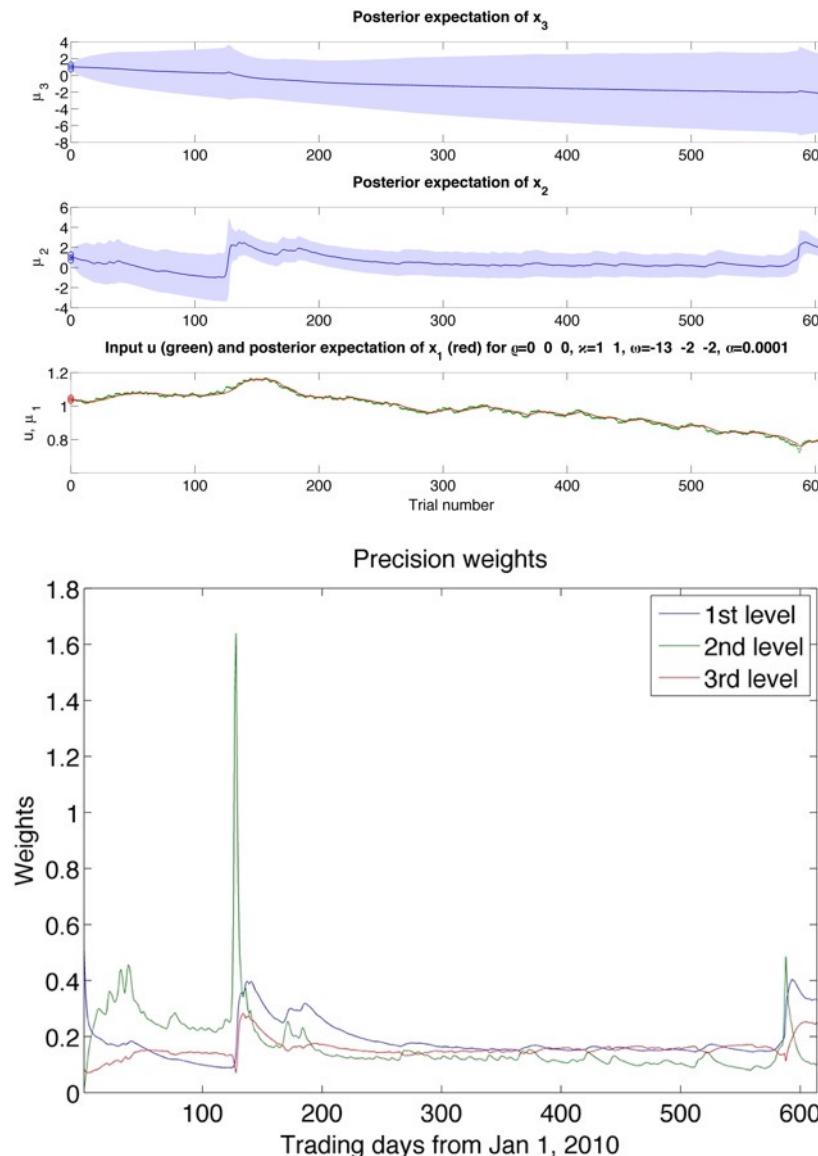
(instead of the constant ϑ in the Kalman filter)

The diagram illustrates the components of the HGF learning rate formula. A horizontal line represents the total probability $\hat{\pi}_u$. Above the line, a green arrow labeled "outcome uncertainty" points downwards to the right. Below the line, a purple oval encloses the term $\frac{1}{\sigma_1^{(k-1)} + \exp(\kappa_1 \mu_2^{(k-1)} + \omega_1)}$. Two arrows point towards this term from below: a red arrow labeled "informational uncertainty" from the left, and a purple arrow labeled "environmental uncertainty" from the right.

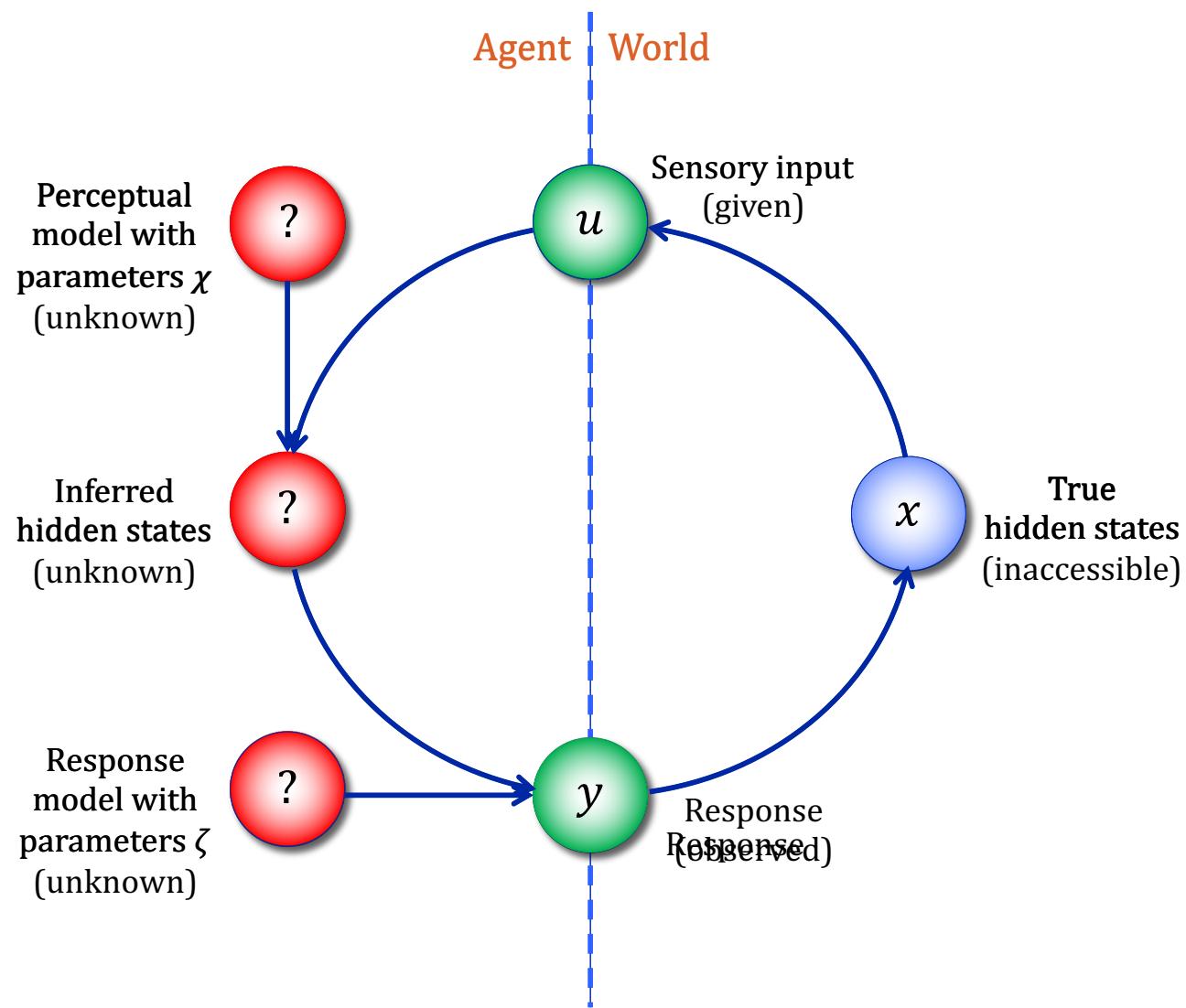
3-level HGF for continuous observations



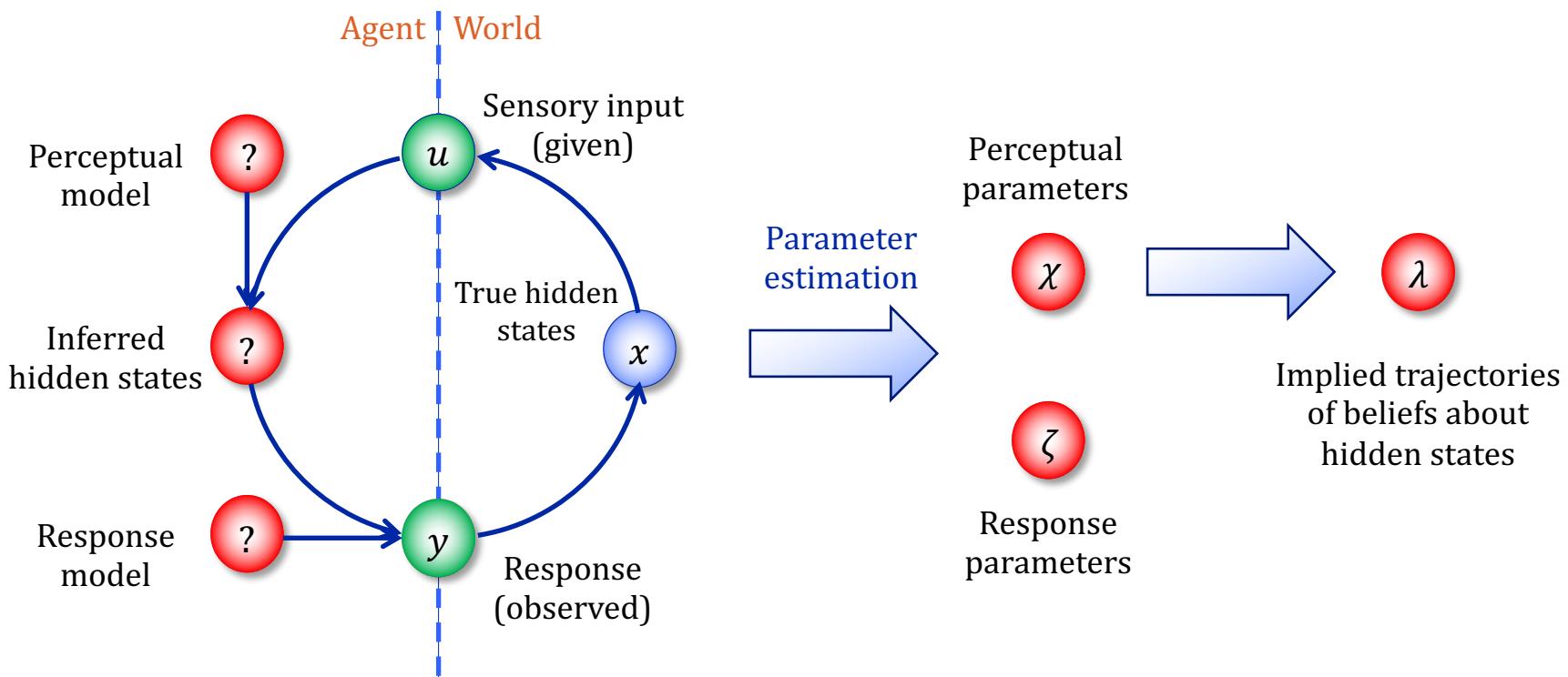
Example of precision weight trajectory



Application to experimental data

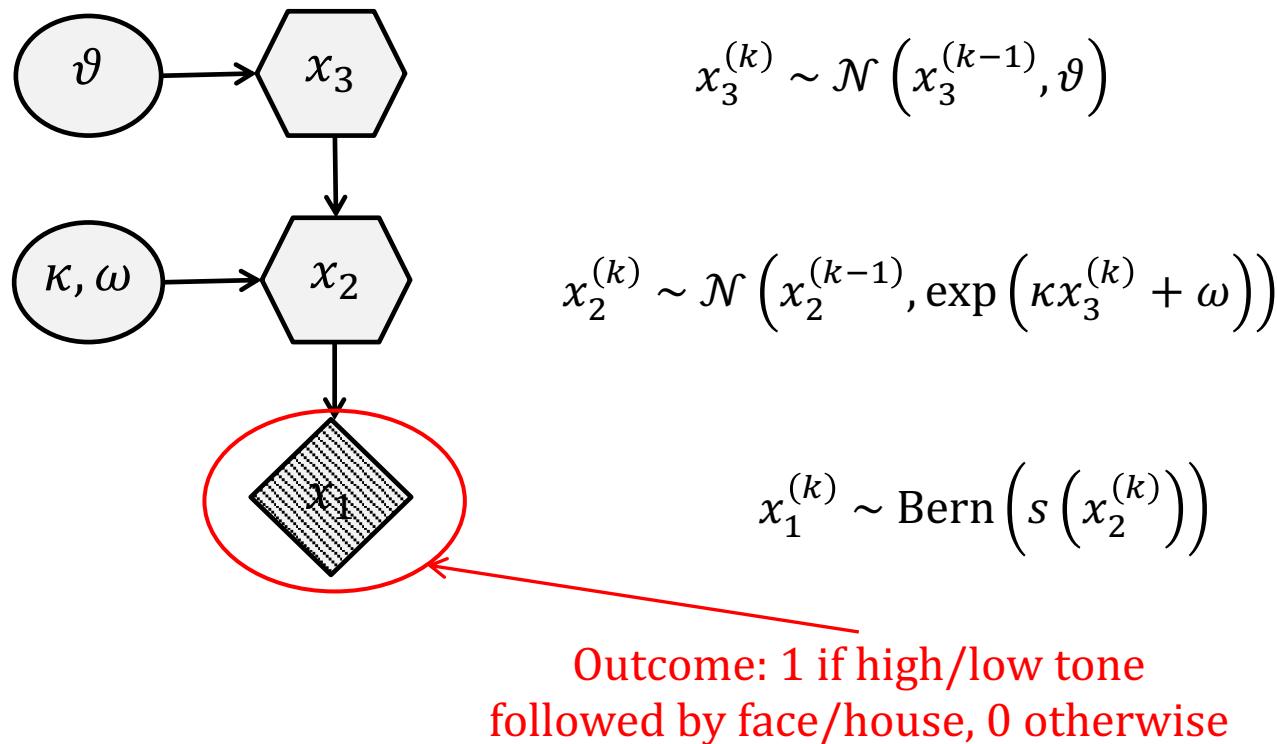


Application to experimental data: parameter estimation



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

3-level HGF for binary observations



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 model* model describes (probabilistically) how the states x_i evolve in time: $x_i^{(k-1)} \rightarrow x_i^{(k)}$
- 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: $\{\mu_i^{(k-1)}, \pi_i^{(k-1)}\} \rightarrow \{\mu_i^{(k)}, \pi_i^{(k)}\}$
- The decision model (also called observation model or response model) describes (probabilistically) how beliefs are translated into observed actions:
$$\{\mu_i^{(k)}, \pi_i^{(k)}\}_{i=1,\dots,l} \rightarrow y^{(k)}$$
- 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
<https://translationalneuromodeling.github.io/tapas>
- Start with README, manual, and interactive demo
- Modular, extensible, Matlab-based
- Julia version coming in September 2022

```
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

Activations by Precision-Weighted Visual Outcome Prediction Error ε_2

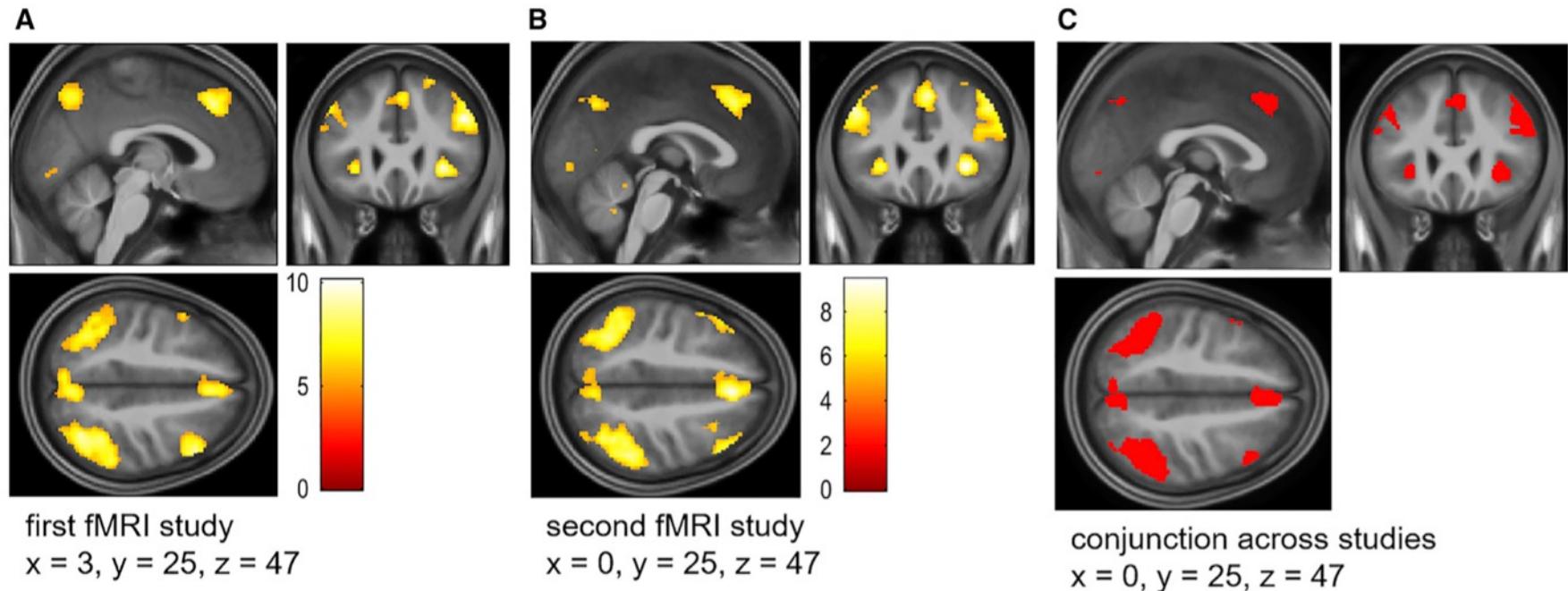


Figure 2. Whole-Brain Activations by ε_2

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

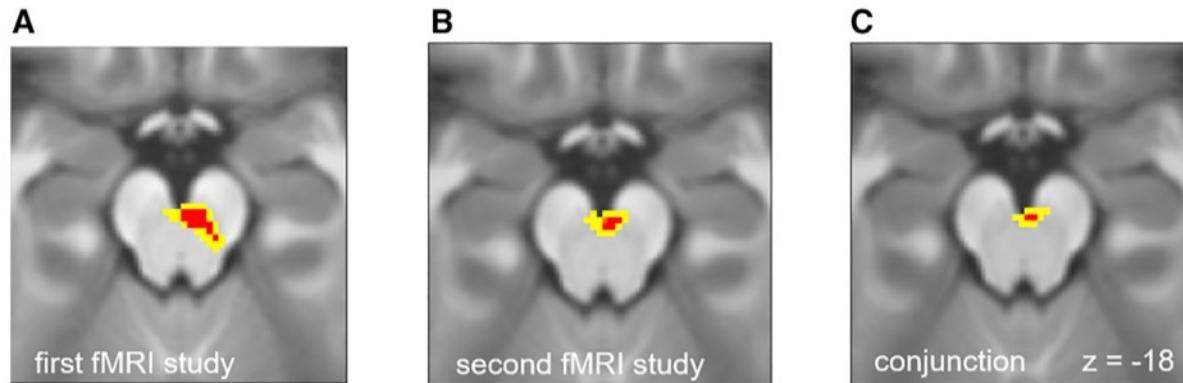


Figure 3. Midbrain Activation by ε_2

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

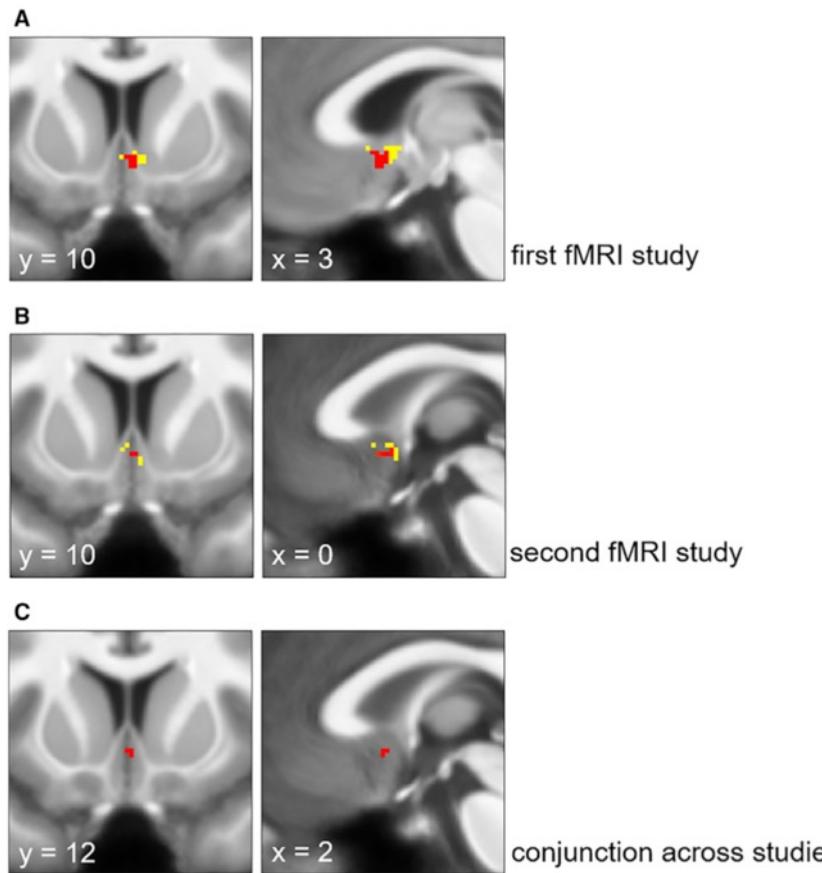
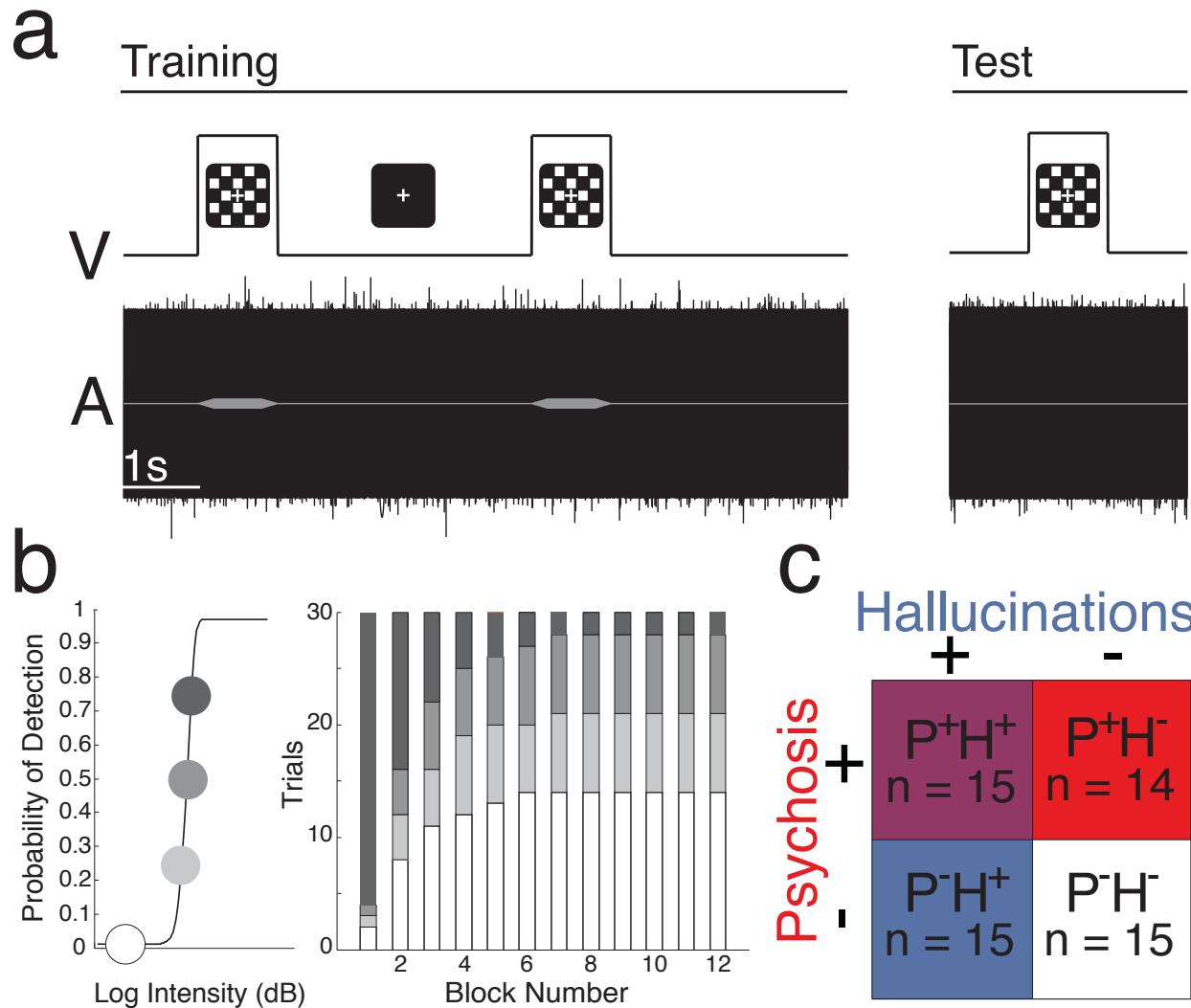


Figure 6. Basal Forebrain Activations by ε_3

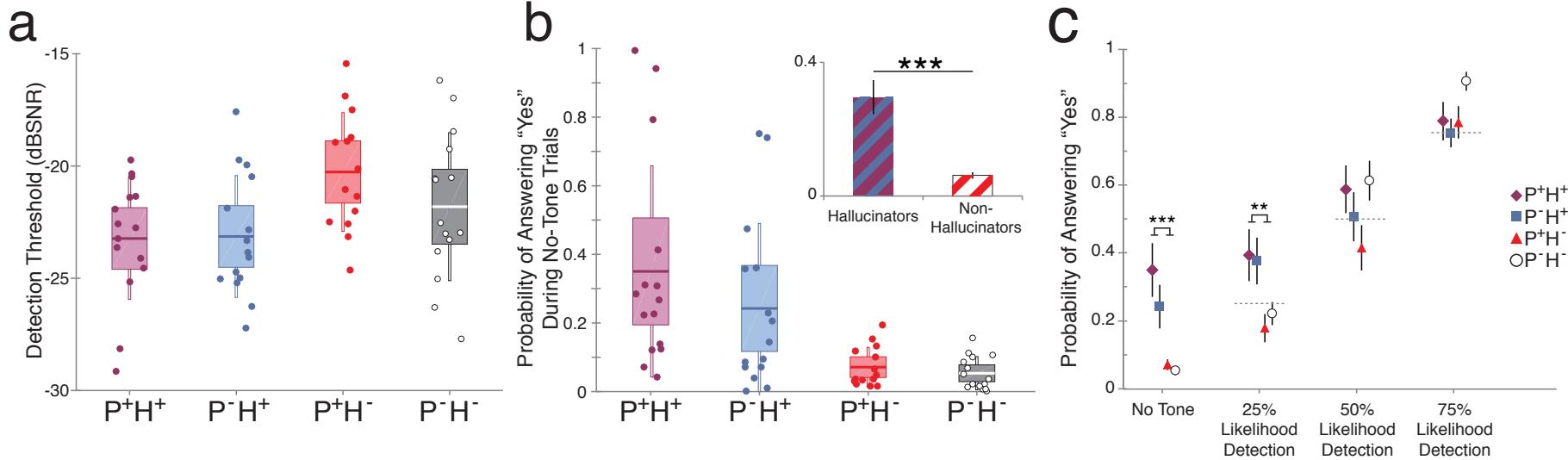
Activation of the basal forebrain by precision-weighted prediction error about stimulus probabilities ε_3 within the anatomically defined mask. For visualization of the activation area, we overlay the results thresholded at $p < 0.05$ FWE peak-level corrected for the entire anatomical mask (red) on the 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)

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:

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

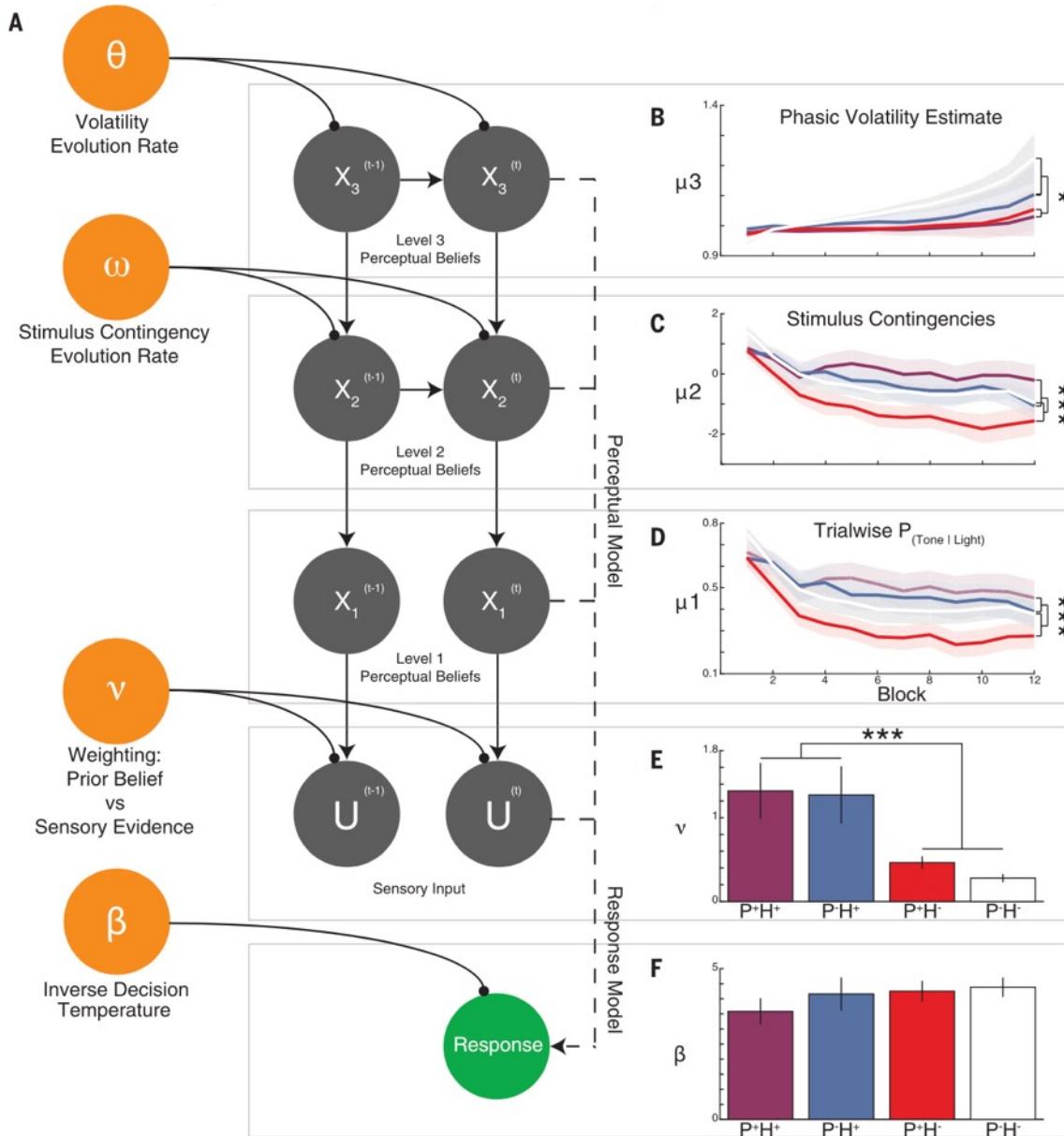
- *belief* is formalized as the Bayesian posterior mean of a beta distribution

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

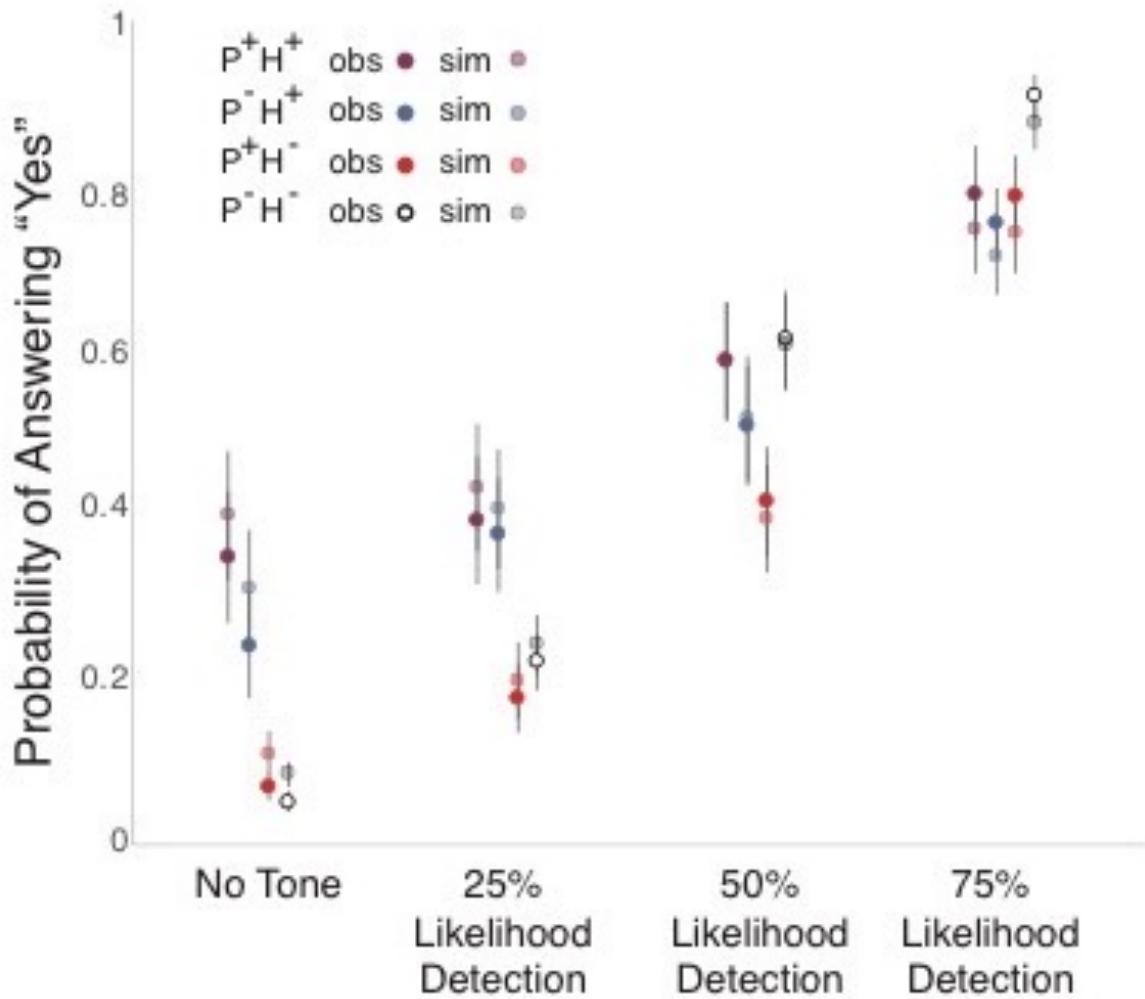
where

- *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



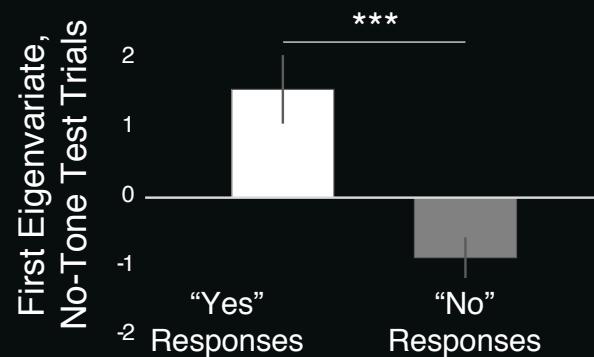
Conditioned hallucinations

a

Thresholding, Tone-Responsive Regions

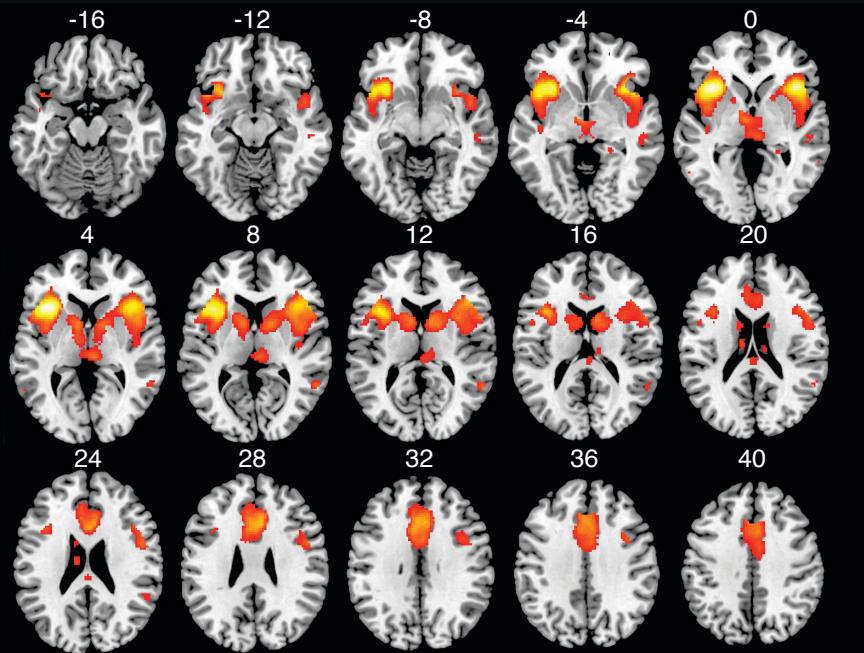


b



c

Test, "Yes" > "No" Responses, No-Tone Trials



d

Areas Active During AVH Symptom Capture

