

The generalized Hierarchical Gaussian Filter

Lilian Aline Weber^{1,2*}, Peter Thestrup Waade³, Nicolas Legrand³, Anna Hedvig Møller³, Klaas Enno Stephan^{2,4}, Christoph Mathys^{2,5,6}

1 Wellcome Centre for Integrative Neuroscience, Department of Psychiatry, University of Oxford, Oxford, United Kingdom

2 Translational Neuromodeling Unit, Institute for Biomedical Engineering, University of Zurich & ETH Zurich, Zurich, Switzerland

3 Cognitive Science Department, Aarhus University, Denmark

4 Max Planck Institute for Metabolism Research, Cologne, Germany

5 Scuola Internazionale Superiore di Studi Avanzati (SISSA), Trieste, Italy

6 Interacting Minds Centre, Aarhus University, Aarhus, Denmark

* lilian.weber@psych.ox.ac.uk

Abstract

Hierarchical Bayesian models of perception and learning feature prominently in contemporary cognitive neuroscience where, for example, they inform computational concepts of mental disorders. This includes predictive coding and hierarchical Gaussian filtering (HGF), which differ in the nature of hierarchical representations. Predictive coding assumes that higher levels in a given hierarchy influence the state (value) of lower levels. In HGF, however, higher levels determine the rate of change at lower levels. Here, we extend the space of generative models underlying HGF to include a form of nonlinear hierarchical coupling between state values akin to predictive coding and artificial neural networks in general. We derive the update equations corresponding to this generalization of HGF and conceptualize them as connecting a network of (belief) nodes where parent nodes either predict the state of child nodes or their rate of change. This enables us to (1) create modular architectures with generic computational steps in each node of the network, and (2) disclose the hierarchical message passing implied by generalized HGF models and to compare this to comparable schemes under predictive coding. We find that the algorithmic architecture instantiated by the generalized HGF is largely compatible with that of predictive coding but extends it with some unique predictions which arise from precision and volatility related computations. Our developments enable highly flexible implementations of hierarchical Bayesian models for empirical data analysis and are available as open source software.

Keywords: hierarchical Gaussian filter, HGF, predictive coding, perceptual inference, neuromodelling, computational psychiatry

1 Introduction

1.1 Overview

In order successfully to navigate complex environments, biological agents need to combine noisy sensory inputs with prior knowledge to infer on the true, but hidden, state of the world. Hierarchical Bayesian models such as predictive coding (Rao & Ballard, 1999; Friston, 2005) have become popular tools to understand brain functions which enable humans to form beliefs based on sparse and ambiguous sensory inputs. These models assume that the brain constructs a hierarchy of beliefs, with higher level beliefs relating to increasingly abstract features of the world. For example, hierarchical Gaussian filtering (HGF, Mathys et al., 2011, 2014a) models a hierarchy in which higher levels encode the rate of change in lower level features, enabling agents adaptively to scale their learning behaviour in response to changes in the stability of their environment. Here, we extend this framework to a different type of hierarchy, where higher-level beliefs influence lower-level beliefs via their expectation. In other words, higher-level beliefs can here also determine the *value* of lower-level beliefs instead of only their rate of change. This extension significantly widens hierarchical Gaussian filtering’s application scope and enables a direct comparison of the implied message-passing scheme with, for example, predictive coding models. Moreover, the new derivations equip HGF with a modular architecture which allows for more versatile and user-friendly implementations of HGF models.

1.2 Hierarchical Bayesian models of perception and learning

Bayesian perspectives on perception have proposed that our brain inverts a hierarchical generative model to infer the causes of its sensory inputs and predict future events (Dayan et al., 1995; Rao & Ballard, 1999; Friston, 2010; Doya et al., 2011; Helmholtz, 1860). Under this “Bayesian brain” view, perception corresponds to integrating expectations (prior beliefs about hidden states of the world) with incoming sensory information to yield a posterior belief, while *learning* refers to the updating of beliefs about the model’s parameters, which takes place more slowly, as experience accumulates. Formally, beliefs are modelled as probability distributions, such that the width of the distribution reflects the uncertainty (inverse precision) associated with that belief. Humans have been shown to take uncertainty into account when combining different sources of information in a manner that conforms to the statistical optimum as prescribed by Bayes’ rule (Ernst & Banks, 2002; Angelaki et al., 2009).

Because sensory signals are generated by interacting causes in the external environment that span multiple spatial and temporal scales, the brain in its Helmholtzian description is assumed to reflect this hierarchy of causes in a correspondingly hierarchical generative model. The Bayesian inversion of this generative model results in a hierarchy of beliefs, where higher levels encode beliefs about increasingly abstract, general, and stable features of the environment. These higher-level beliefs serve as priors for the inference on lower levels. Specifically, at each level of the hierarchy, belief updates serve to reconcile predictions (priors) from higher levels with the actual input (likelihood) from lower levels. Furthermore, under fairly general assumptions (i.e., for all probability distributions from the exponential family, Mathys, 2016; Mathys & Weber, 2020), these belief updates rest on (precision-weighted) prediction errors (PEs), i.e., the (weighted) mismatch between the model’s predictions and the actual input (Friston, 2010).

Popular hierarchical Bayesian models of perception and learning that are built on these ideas are predictive coding (Rao & Ballard, 1999; Friston, 2005) and hierarchical Gaussian filtering (Mathys et al., 2011, 2014a). In these models, estimates of uncertainty are central: the impact of prediction errors on belief updates depends on a precision ratio,

which relates the precision of the prior to that of the observation, thus scaling the relative impact that new information has on belief updates. Put simply, mismatches (PEs) elicit stronger belief updates if the prediction about the input (likelihood) is precise, relative to the belief in the current estimate (prior). This form of adaptive scaling, a key element of healthy inference and learning, has also been proposed to lie at the heart of perceptual disturbances observed in mental disorders. For example, an imbalance between the influence of expectations and sensory inputs has featured prominently in attempts to explain the emergence of positive symptoms in schizophrenia, such as hallucinations and delusions (Stephan et al., 2006; Fletcher & Frith, 2009; Corlett et al., 2009, 2011; Adams et al., 2013; Friston et al., 2016; Sterzer et al., 2018).

The HGF¹ has been particularly useful in this context, as it can be fit to participants' empirically observed behaviour or physiology, and thereby used to infer individual trajectories of precision-weighted PEs and predictions from data. By formulating a response model that links trial-wise perceptual quantities (such as predictions and PEs) to measured quantities (such as choices, reaction times, eye movements, evoked response amplitude in EEG, etc.), the HGF can quantify individual differences in inference and learning in terms of model parameters that encode prior beliefs about higher-order structure in the environment (de Berker et al., 2016; Lawson et al., 2017; Powers et al., 2017; Siegel et al., 2020; Sevgi et al., 2020; Henco et al., 2020; Rossi-Goldthorpe et al., 2021; Suthaharan et al., 2021; Kafadar et al., 2022; Sapey-Triomphe et al., 2022; Fromm et al., 2023; Drusko et al., 2023). Such a mechanistic characterization of inter-subject variability is of particular interest for fields like Computational Psychiatry, because such differences may explain the heterogeneous nature of psychiatric diseases, and form a basis for mapping them out in a continuous conceptual space or dividing them into more homogeneous subgroups (Stephan & Mathys, 2014; Mathys, 2016).

1.3 Volatility-coupling, value-coupling, and noise-coupling

The type of belief hierarchy modelled by any particular approach depends on the nature of the generative model. The HGF assumes a particular form of generative model, where hidden states of the world evolve as coupled random walks in time. In current HGF models, the mean (value) of the higher level determines the variance (step size) of the lower level's random walk. In other words, higher levels encode the volatility (or inverse stability) of lower levels (we will call this **volatility coupling**). This is motivated by the observation that learners must take into account different sources of uncertainty in their belief updates, one of which is the current level of stability in the world: if the world is currently changing (volatile), the agent needs to learn faster. Accordingly, in the HGF, subjective² estimates of increased environmental volatility directly influence the uncertainty associated with lower level beliefs, leading to faster belief updates on the lower level. Previous work has shown that human learners indeed adjust their learning rate according to experimentally manipulated levels of volatility (Behrens et al., 2007).

By contrast, predictive coding models typically focus on hierarchies in which higher levels predict the *value* of lower levels, or, in other words, the mean of the probability distribution that represents the lower-level belief, or in other words, the expectation of the value. We will refer to these hierarchies as implementing **value coupling**. This type of hierarchy is useful for understanding how beliefs about lower-level features depend on higher-level beliefs – for example, how the perceived brightness of a patch in an image depends on the context (objects, shadows) in which that patch is presented (Rao & Ballard, 1999; Adelson, 2005). It is worth noting that the type of hierarchical coupling

¹we use “the HGF” as a shorthand for “hierarchical Gaussian filtering models of various configurations”

²Note that the word subjective does not imply conscious accessibility here - it only differentiates the inferred quantities from the “true” values as produced by the environment or the generative model.

in predictive coding is often used in theoretical treatments of hierarchical Bayesian modelling while the HGF offers a flexibly applicable implementation that is being widely used for empirical data analysis.

In a noteworthy exception to this, [Kanai et al. \(2015\)](#) presented a predictive coding model where higher levels encode the (spatial) precision of lower levels in a static environment. This is different from volatility coupling, where higher levels are concerned with the rate of change on lower levels, but captures a second source of uncertainty in beliefs about hidden states: the level of noise or reliability of the sensory input.

Relatedly, in the learning and decision-making literature, two classes of models separately deal with the estimation of process noise (volatility, [Behrens et al., 2007](#); [Mathys et al., 2011](#); [Piray & Daw, 2020](#)) and observation noise (stochasticity, [Lee et al., 2020](#); [Nassar et al., 2010](#)), with recent attempts to capture both sources of uncertainty in a joint model ([Piray & Daw, 2021](#)).

In this work, we show that the HGF can be extended to encompass both volatility and value coupling both at the level of hidden states, and at the observation stage. Importantly, coupling types that link levels via lower-level belief precision include volatility coupling (between hidden states) as well as a principled way of modelling inference on observation noise: via noise coupling at the level of observations. Together with the different types of value coupling we will introduce, the generalised HGF thus provides a very general modelling framework with a wide range of potential applications.

1.4 Contribution of current work

In this technical note, we extend the generative model of the HGF to consider hierarchical value coupling alongside volatility coupling. Based on the work in [Weber \(2020\)](#), we (1) derive simple, efficient one-step belief update equations for linear and non-linear value coupling, and (2) conceptualize these equations, together with their volatility coupling counterparts, as a network of interacting nodes which can be implemented in a modular architecture.

In brief, we show that:

1. The HGF provides a very general modelling framework that encompasses multiple types of interactions between states in the world - where higher-level hidden states determine the value, the rate of change (volatility), or even the level of noise in lower-level states or observations.
2. The explicit treatment of volatility estimation in the HGF allows for an implementation that comprises both global control of volatility-related precision (implemented by a global high-level volatility belief that affects multiple low-level states), and local or distributed volatility estimation, enabling modality-specific modulation of learning rates.
3. The message passing scheme for value coupling in the HGF is almost equivalent to recently proposed predictive coding architectures, apart from small, but interesting differences. These differences relate on one hand to the discrete nature of the updates in the HGF, and, on the other hand, to the volatility-related updates of belief uncertainty.

2 Introducing value coupling

The generative model for value coupling

The HGF assumes that an agent is trying to infer on (and learn about) a continuous uncertain quantity x in their environment, which moves (changes) over time. Without any information about the specific form of movement, a generic way of describing movement of a continuous quantity is a Gaussian random walk:

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

In the original formulation (Mathys et al., 2011, 2014a), coupling between environmental states at different hierarchical levels was implemented in the form of **volatility coupling**, where the step size ϑ (or rate of change/volatility) of a state x_a varies a function of a higher-level state $x_{\tilde{a}}$:

$$\vartheta = f\left(x_{\tilde{a}}^{(k)}\right) = \exp\left(\kappa_{\tilde{a},a} x_{\tilde{a}}^{(k)} + \omega_a\right), \quad (2)$$

with parameters $\kappa_{\tilde{a},a}$ (scaling the impact of volatility parent $x_{\tilde{a}}$ on x_a) and ω_a (capturing the "tonic" step size or volatility, which does not vary with time). By simultaneously inferring on the state x_a and its rate of change $x_{\tilde{a}}$, the agent can learn faster (slower) in times when x_a is changing more (less). We call $x_{\tilde{a}}$ a **volatility parent** of x_a .

In contrast, here, we consider the case where a higher-level state x_b influences the value (mean) of the lower-level state x_a :

$$x_a^{(k)} \sim \mathcal{N}\left(x_a^{(k-1)} + \rho_a + f\left(x_b^{(k)}\right), \vartheta\right), \quad (3)$$

where

$$f\left(x_b^{(k)}\right) = \alpha_{b,a} g_{b,a}\left(x_b^{(k)}\right) \quad (4)$$

is the coupling function with parameter $\alpha_{b,a}$ (scaling the impact of **value parent** x_b on x_a). The state's mean now evolves as a function of its previous value $x_a^{(k-1)}$, a drift parameter ρ_a , and a function of state $x_b^{(k)}$. Crucially, as we will show below, the HGF can deal with both linear and nonlinear transformation functions g , as long as the function g is twice continuously differentiable almost everywhere. States x_b and x_a interact via **value coupling**.

The effect of a higher-level state on a lower-level state with this kind of coupling can be understood as a phasic drift signal (see Figure 1 for illustration). The drift parameter ρ determines the "tonic" drift, equivalent to the tonic volatility parameter ω . We can use the same configuration also to capture situations where the higher-level state represents a mean value to which the lower-level state reverts back to over time. This is achieved simply by inserting parameter $\lambda_a \in [0, 1]$ that encodes the state's auto-connection strength:

$$x_a^{(k)} \sim \mathcal{N}\left(\lambda_a x_a^{(k-1)} + \rho_a + f\left(x_b^{(k)}\right), \vartheta\right). \quad (5)$$

When $\lambda_a < 1$, the state x_a will change as an autoregressive process, reverting back to the total mean M_a , which is given by:

$$M_a = \frac{\rho_a + f\left(x_b^{(k)}\right)}{1 - \lambda_a}, \quad (6)$$

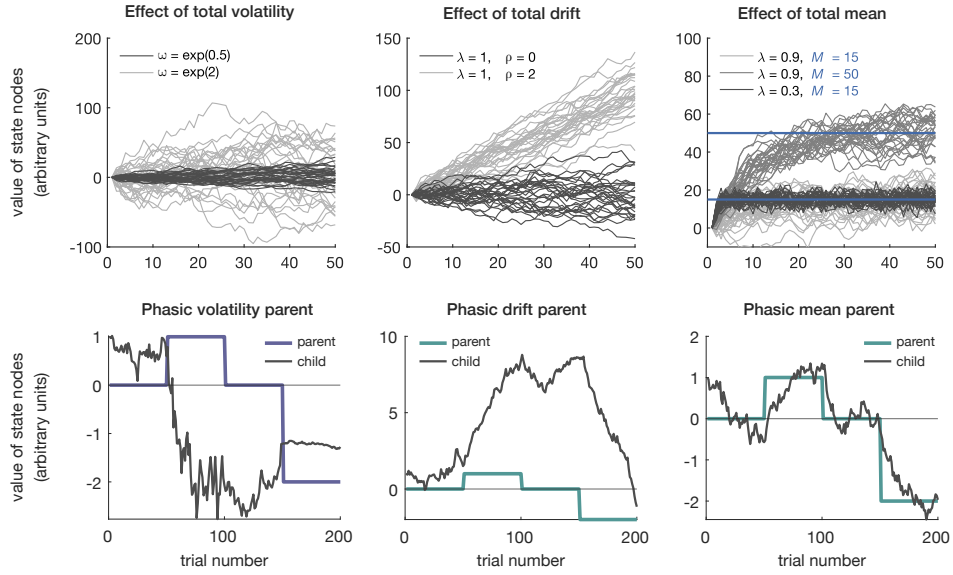


Figure 1. Effects of different coupling types within the generative model of the HGF. *Upper row:* Plots illustrate the effects of a state's total (constant) volatility (left), drift (middle) and mean (right) on the state's evolution over time steps. Each plot shows 30 simulated state trajectories per value of volatility/drift/mean for a continuous state performing a Gaussian random walk over 50 time steps. *Lower row:* Plots illustrate the effects of phasic changes in a state's volatility (left), drift (middle), and mean (right). Each plot shows the trajectory of the parent node and one simulated trajectory of the child node coupled to the parent via volatility/drift/mean coupling.

and the value of λ will determine the speed with which it does so. In other words, the meaning of ρ and x_b for the evolution of x_a depends on λ . The difference between the cases $\lambda = 1$ and $\lambda < 1$ is illustrated in Figure 1, and the derivation based on a first-order autoregressive process is given in Appendix ??). Finally, if $\lambda = 0$, the state does not perform a Gaussian random walk around its own previous mean anymore, instead, its values are normally distributed around a constant or the value of a parent state (reminiscent of an observation of the parent with Gaussian noise).

Importantly, different forms of coupling can be present at the same time: A state x_a can have both a **volatility parent** $x_{\tilde{a}}$ (generating changes in its rate of change) and a **value parent** x_b (generating changes in its mean value). It can also have a drift parameter ρ_a which is a constant influencing its mean – equivalent to its tonic volatility parameter ω_a which determines its step size in the absence of a phasic volatility influence. Finally, we allow for inputs to arrive at irregular intervals; therefore, we multiply the total variance of the random walk and the total mean drift by the time $t^{(k)}$ that has passed since the arrival of the previous sensory input at index $k - 1$ (Mathys et al., 2014a). Together with suitably chosen priors on parameters and initial states (see Mathys et al., 2014a), the following equation forms the generative model for a state x_a with both volatility and value parent:

$$x_a^{(k)} \sim \mathcal{N} \left(\lambda_a x_a^{(k-1)} + t^{(k)} \left(\rho_a + \alpha_{b,a} g_{b,a} \left(x_b^{(k)} \right) \right), t^{(k)} \exp \left(\omega_a + \kappa_{\tilde{a},a} x_{\tilde{a}}^{(k)} \right) \right). \quad (7)$$

In the even more general case, a state could have multiple value parents and multiple volatility parents, each affecting the mean value and rate of change of state x_a in proportion to their respective coupling strengths³:

$$x_a^{(k)} \sim \mathcal{N} \left(\lambda_a x_a^{(k-1)} + \underbrace{t^{(k)} \left(\rho_a + \sum_i \alpha_{b_i,a} g_{b_i,a} \left(x_{b_i}^{(k)} \right) \right)}_{\text{total drift: } P_a^{(k)}}, \underbrace{t^{(k)} \exp \left(\omega_a + \sum_j \kappa_{\tilde{a}_j,a} x_{\tilde{a}_j}^{(k)} \right)}_{\text{total volatility: } \Omega_a^{(k)}} \right). \quad (8)$$

Because a given state can also be parent to multiple child states at the same time, these extensions allow us to model fairly complex networks of interacting states of the world. In Figure 2 we have drawn an example setup with 11 different environmental states and two outcomes. For this example, and together with priors on parameters and initial states (see Mathys et al., 2014a), the following equations describe the generative model (for simplicity, the example uses linear **value coupling**, no tonic drifts ($\rho = 0$),

³We are here only considering the additive effect of multiple parents on a given state, but more sophisticated interactions are conceivable.

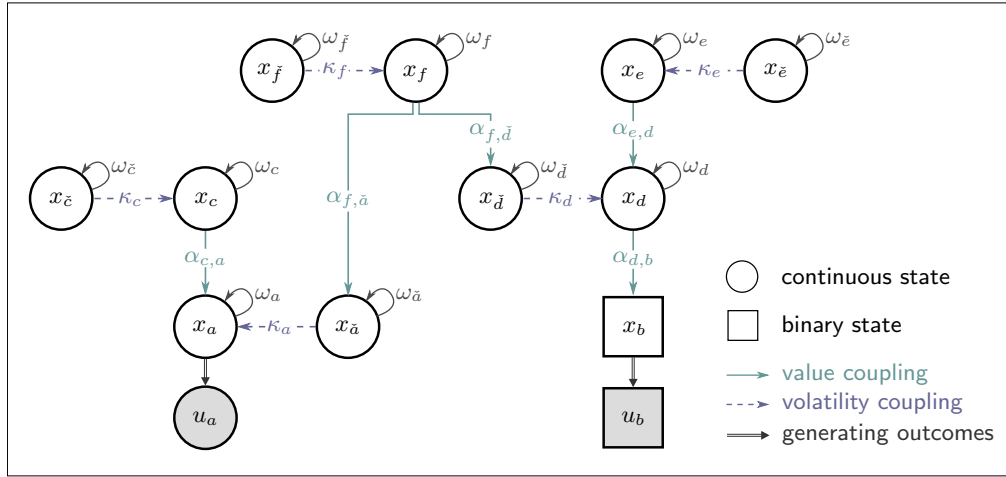


Figure 2. An example of a generative model of sensory inputs with 11 hidden states and two observable outcomes. In this example, the volatility parents $x_{\tilde{a}}$ and $x_{\tilde{d}}$ share a value parent x_f , which represents a "global" or shared volatility state. Circles – continuous states, squares – binary states, observable outcomes – shaded. Volatility coupling – dashed lines, value coupling – straight lines, links of outcomes to their hidden states – double arrows.

$\lambda = 1$), and inputs at regular intervals, i.e., $t^{(k)} \equiv 1 \forall k$):

$$u_a^{(k)} \sim \mathcal{N}(x_a^{(k)}, \varepsilon_u) \quad (9)$$

$$x_a^{(k)} \sim \mathcal{N}(x_a^{(k-1)} + \alpha_{c,a}x_c^{(k)}, \exp(\kappa_a x_{\tilde{a}}^{(k)} + \omega_a)) \quad (10)$$

$$x_{\tilde{a}}^{(k)} \sim \mathcal{N}(x_{\tilde{a}}^{(k-1)} + \alpha_{f,\tilde{a}}x_f^{(k)}, \exp(\omega_{\tilde{a}})) \quad (11)$$

$$x_c^{(k)} \sim \mathcal{N}(x_c^{(k-1)}, \exp(\kappa_c x_{\tilde{c}}^{(k)} + \omega_c)) \quad (12)$$

$$x_{\tilde{c}}^{(k)} \sim \mathcal{N}(x_{\tilde{c}}^{(k-1)}, \exp(\omega_{\tilde{c}})) \quad (13)$$

$$u_b^{(k)} \sim \text{Bern}(x_b^{(k)}) \quad (14)$$

$$x_b^{(k)} \sim \text{Bern}(S(x_d^{(k)})) \quad (15)$$

$$(16)$$

$$x_d^{(k)} \sim \mathcal{N}(x_d^{(k-1)} + \alpha_{e,d}x_e^{(k)}, \exp(\kappa_d x_{\tilde{d}}^{(k)} + \omega_d)) \quad (17)$$

$$x_{\tilde{d}}^{(k)} \sim \mathcal{N}(x_{\tilde{d}}^{(k-1)} + \alpha_{f,\tilde{d}}x_f^{(k)}, \exp(\omega_{\tilde{d}})) \quad (18)$$

$$x_e^{(k)} \sim \mathcal{N}(x_e^{(k-1)}, \exp(\kappa_e x_{\tilde{e}}^{(k)} + \omega_e)) \quad (19)$$

$$x_{\tilde{e}}^{(k)} \sim \mathcal{N}(x_{\tilde{e}}^{(k-1)}, \exp(\omega_{\tilde{e}})) \quad (20)$$

$$x_f^{(k)} \sim \mathcal{N}(x_f^{(k-1)}, \exp(\kappa_f x_{\tilde{f}}^{(k)} + \omega_f)) \quad (21)$$

$$x_{\tilde{f}}^{(k)} \sim \mathcal{N}(x_{\tilde{f}}^{(k-1)}, \exp(\omega_{\tilde{f}})). \quad (22)$$

Note that this example also includes binary states (x_b) and observable outcomes (u_a and u_b). Our main discussion will focus on continuous states performing Gaussian random walks, but we will briefly touch on other types of states (binary, categorical, input) in section 6.3.

Using this example network, we introduce two general motifs. First, all states that are value parents of other states (or outcomes) by default have their own volatility parent (and volatility parents therefore share the index with their child node, for example, states x_a and $x_{\bar{a}}$). Even if in practice, many environmental states might have constant volatility, from the perspective of the agent, it makes sense to a-priori allow for phasic changes in volatility. From a modelling perspective, these volatility parents could be removed in scenarios with constant volatility. On the other hand, more than one volatility levels may sometimes be required.

Second, states that are volatility parents to other states can either have a value parent (as states $x_{\bar{a}}$ and $x_{\bar{d}}$), or no parents (as states $x_{\bar{c}}$, $x_{\bar{e}}$ and $x_{\bar{f}}$). This is because in practice, volatility parents of volatility parents are rarely required. Instead, we suggest that value parents of volatility states are more useful. In particular, these can be used to model “global” or shared volatility states that affect multiple lower-level volatility beliefs (such as state x_f in this example, which influences volatility beliefs about states x_a and x_d), separately from “local” volatility states that only affect speed of change in a single lower-level state (such as the volatility states $x_{\bar{c}}$, $x_{\bar{e}}$ and $x_{\bar{f}}$).

In this section, we have introduced value coupling to the generative model of the HGF, for the first time considering the case where the mean of x_a is a function not only of its own previous value but also (some transformation of) the current value of some higher-level state x_b , scaled by a coupling parameter $\alpha_{b,a}$. We will now show how an agent can infer on the values of such hidden states.

The belief update equations for value coupling

An agent employing a generative model of the kind described above to do perceptual inference holds a belief about the current value of each of the states (i.e., every x) of this model at every time point k . We describe this belief about state x at time k as a Gaussian distribution, fully characterized by its mean $\mu^{(k)}$ and its inverse variance, or precision, $\pi^{(k)}$ at time k .

In the approximate inversion of the generative model for volatility coupling, [Mathys et al. \(2011\)](#) derived a set of single-step update equations that represent the approximately Bayes-optimal changes in these beliefs in response to incoming stimuli. Repeating this derivation for the case of value coupling similarly leads us to simple one-step equations for updating beliefs about states that serve as value parents (for the full derivation of these equations, cf. [Appendix 6.1](#)). Assuming state x_b is a value parent to state x_a with a coupling strength $\alpha_{b,a}$, then the new posterior belief about state x_b after observing a new input at time step k is given by:

$$\begin{aligned}\pi_b^{(k)} &= \hat{\pi}_b^{(k)} + \hat{\pi}_a^{(k)} \left(\alpha_{b,a}^2 g'_{b,a} \left(\mu_b^{(k-1)} \right)^2 - g''_{b,a} \left(\mu_b^{(k-1)} \right) \delta_a^{(k)} \right) \\ \mu_b^{(k)} &= \hat{\mu}_b^{(k)} + \frac{\hat{\pi}_a^{(k)} \alpha_{b,a} g'_{b,a} \left(\mu_b^{(k-1)} \right)}{\pi_b^{(k)}} \delta_a^{(k)},\end{aligned}\tag{23}$$

where $\hat{\pi}_b^{(k)}$ and $\hat{\mu}_b^{(k)}$ refer to the prediction about state x_b before seeing the new input, and $\delta_a^{(k)}$ is the prediction error about the child state x_a .

Note that these equations also hold when introducing the parameter λ from above. This parameter will only feature in the computation of the prediction $\hat{\mu}_a^{(k)}$ of state x_a and thus affect the prediction error $\delta_a^{(k)}$.

In the case of linear value coupling ($g_{b,a}(x) = Ax + B$), the update further simplifies, as $g'_{b,a}(x) = A$ (a factor that we can absorb into $\alpha_{b,a}$) and $g''_{b,a}(x) = 0$:

$$\begin{aligned}\pi_b^{(k)} &= \hat{\pi}_b^{(k)} + \alpha_{b,a}^2 \hat{\pi}_a^{(k)} \\ \mu_b^{(k)} &= \hat{\mu}_b^{(k)} + \frac{\alpha_{b,a} \hat{\pi}_a^{(k)}}{\pi_b^{(k)}} \delta_a^{(k)}.\end{aligned}\tag{24}$$

As in the case of volatility coupling, the belief updates are thus driven by precision-weighted prediction errors about the lower belief state in the hierarchy.

To get an intuition for these update equations in the case of nonlinear value coupling, let us consider an example where state x_a acts as a rectified linear unit (ReLU), that is, we choose the function g as the rectifier function, a popular activation function for deep neural networks (Hahnloser et al., 2000; Lecun et al., 2015):

$$g_{b,a}(x_b) := \max(0, x_b).\tag{25}$$

The first and second derivative of g are then

$$g'_{b,a}(x_b) = [x_b > 0] := \begin{cases} 1, & \text{if } x_b > 0 \\ 0, & \text{otherwise} \end{cases}\tag{26}$$

and

$$g''_{b,a}(x_b) = \delta(x_b = 0),\tag{27}$$

respectively. For our purposes, we can treat the second derivative as

$$g''_{b,a}(x_b) \equiv 0.\tag{28}$$

Plugging this into equation 23, we get

$$\pi_b^{(k)} = \hat{\pi}_b^{(k)} + \alpha_{b,a}^2 \hat{\pi}_a^{(k)} [\mu_b^{(k-1)} > 0]\tag{29}$$

and

$$\mu_b^{(k)} = \hat{\mu}_b^{(k)} + \frac{\alpha_{b,a} \hat{\pi}_a^{(k)} [\mu_b^{(k-1)} > 0]}{\pi_b^{(k)}} \delta_a^{(k)}.\tag{30}$$

In other words, the impact of lower-level prediction errors $\delta_a^{(k)}$ on the posterior belief at the higher level $\mu_b^{(k)}$ and $\pi_b^{(k)}$ depends on the previous state of the higher-level node, such that beliefs only change in response to inputs if they were above zero (or “active”) in the first place.

Similarly, we can construct the reverse coupling function to model saturation:

$$g_{b,a}(x_b) := \min(0, x_b - \nu)\tag{31}$$

Here, state x_b only exerts an influence on the lower-level state x_a , if its own value is below a threshold ν . In the inference on state x_b , this means that the belief about the parent node, μ_b , will stop being updated after crossing this threshold ν (i.e., after becoming saturated):

$$\pi_b^{(k)} = \hat{\pi}_b^{(k)} + \alpha_{b,a}^2 \hat{\pi}_a^{(k)} [\mu_b^{(k-1)} < \nu]\tag{32}$$

and

$$\mu_b^{(k)} = \hat{\mu}_b^{(k)} + \frac{\alpha_{b,a} \hat{\pi}_a^{(k)} [\mu_b^{(k-1)} < \nu]}{\pi_b^{(k)}} \delta_a^{(k)}.\tag{33}$$

This cessation of updates after crossing a threshold does not mean that a node will remain stuck in place for all future time points. The thresholding only affects the HGF's update step while the prediction step keeps affecting $\hat{\pi}_b^{(k)}$ and $\hat{\mu}_b^{(k)}$, eventually possibly moving back to a region where updates take place again.

These examples demonstrate that our extension of the HGF allows us to build deep networks with non-linear coupling functions useful for both artificial and physiological neural networks.

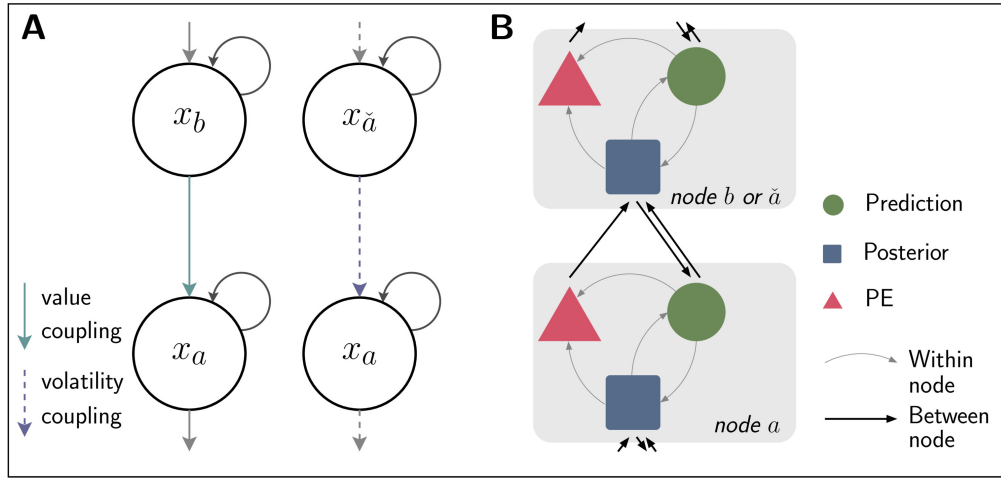


Figure 3. Comparing the flow of information in the generative model of the HGF with the implied belief network. **A** In the generative model, higher-level states influence the evolution of lower-level states (top-down information flow), either by affecting their mean (value coupling, left) or by changing their evolution rate (volatility coupling, right). **B** Representation of the message-passing within and between belief nodes as implied by the HGF’s belief update equations. New observations cause a cascade of message-passing between nodes that includes bottom-up and top-down information flow. Higher-level beliefs send down their posteriors to inform lower-level predictions. Lower-level belief nodes send prediction errors and the precision of their own prediction bottom-up to drive higher-level belief updating. Within a node, we have placed separate units for the three computational steps that each node has to perform at a given time: the prediction step (green), the update step which results in a new posterior belief (blue), and the prediction error step (red). This message passing scheme generalizes across value and volatility coupling, although the specific messages passed along the connections as well as the computations within the nodes will depend on coupling type (see main text and Figures 4 and 5 for details).

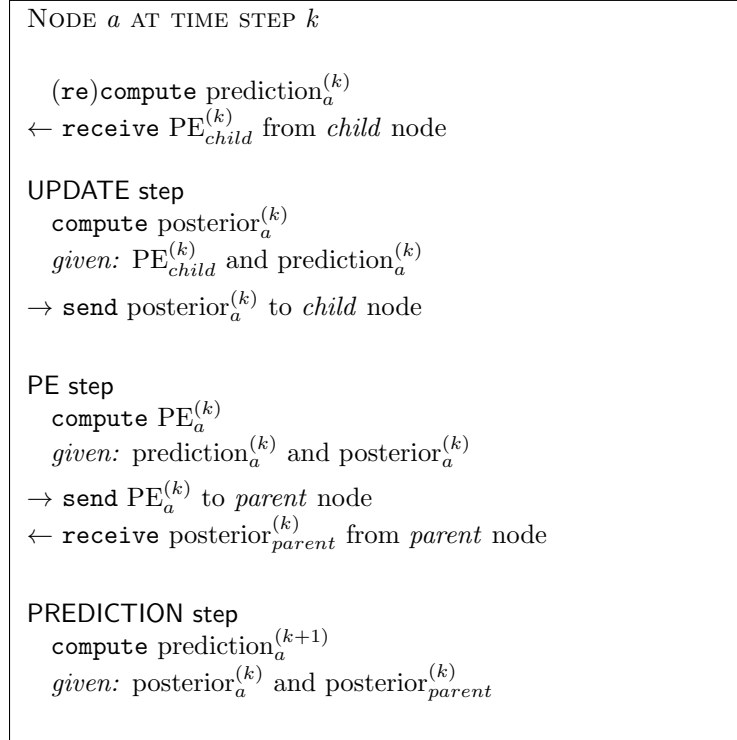
3 Belief updates in the HGF: A network of nodes

Just like we can model complex networks of interacting states in the environment using the generative model of the HGF (Figure 2), we can also think of the inference process of an agent in that environment as a network of interdependent beliefs. The agent entertains a belief about each of the relevant environmental states, and updates these beliefs based on new sensory inputs. Because the agent models its world as a set of hierarchically interacting states, its beliefs about these states will form a hierarchy as well (Figure 3). Before new inputs arrive, higher-level beliefs inform predictions about lower-level beliefs (top-down information flow), whereas after the arrival of a new piece of information, changes in lower-level beliefs trigger updates of higher-level beliefs (bottom-up information flow). Previous work using the HGF has tended to depict the generative model (Figure 3A), which then defines the inference model or set of belief update equations. However, the part of the HGF that represents a model of cognition - the evolution of hierarchically interacting beliefs and the relevant flow of information through this hierarchy - is contained in the inference model (or belief update equations, Figure 3B).

We conceptualize each belief modelled by the HGF as a node in a network, where belief updates involve computations within nodes as well as message passing between nodes. The specific within-node computations and messages passed between nodes will depend on the nature of the coupling. Putting the equations for value and volatility

coupling side by side discloses a modular network architecture. This has important consequences for the implementation of the inference model (both in a computer and in a brain), which we summarize in Figure 3B. We will mainly focus on continuous HGF nodes here - nodes which represent beliefs about continuous quantities that evolve in time as Gaussian random walks. Subsequently, we will consider other types of nodes in HGF models, including categorical nodes and input nodes.

We start by noting that the computations of any node within a time step can be subdivided into three steps, an **update step**, where a new posterior belief is computed based on a prediction and an incoming input or prediction error (PE), a **PE step**, where the difference between expectation (prediction) and new posterior is computed for further message passing upwards, and a **prediction step**, where the new posterior is used to predict the value at the next time step. These can be ordered in time as shown in the box:



Two things are worth noting here. First of all, the **PE step** is a computation that the node performs in service of its parents. From the perspective of the parent node b , the $\text{prediction}_a^{(k)}$ represents an expectation of the child node's state, and the $\text{posterior}_a^{(k)}$ corresponds to the actual state of this child at time step k . The difference between the two amounts to the prediction error which will serve to update the parent node - in other words, the parent's PE.

Second, we have placed the **prediction step** at the end of a time step. This is because usually, we think about the beginning of a time step as starting with receiving a new input, and of a prediction as being present before that input is received. However, in some cases the prediction also depends on the time that has passed in between time steps (e.g., when considering drifts), which is something that can only be evaluated once the new input arrives - hence the additional computation of the (current) prediction at the beginning of the time step. Conceptually, it makes the most sense to think of

the prediction as happening continuously between time steps. It is however implementationally more convenient only to compute the prediction once the new input (and with it its arrival time) enters. This ensures both that the posterior means of parent nodes have had enough time to be sent back to their children for preparation for the new input, and that the arrival time of the new input can be taken into account appropriately.

A node of the above kind is the first computational sub-unit in our perceptual model, and it can be connected to other nodes via **volatility** or **value coupling** depending on the underlying generative model. For node a , another node b can function as a parent node if the two are connected and node b represents a belief about a higher-level quantity which affects the belief about node a according to the generative model. On the other hand, if node b refers to a lower-level quantity and is connected to node a , it serves as a child node for node a .

Computations of nodes in the HGF

In the following, we examine the exact computations at time step k within a node for each of the three steps introduced above. We will compare the relevant computations between **volatility** and **value coupling** and identify the messages that have to be sent and received in each step. Since the update step relies on quantities computed in the (previous) prediction step, we start with the computation of the predictions for the current time step (which, as explained above, we think of as being computed prior to the arrival of a new input). We will first only consider the case of linear value coupling (alongside volatility coupling), and then separately examine any differences in these computations for the case of nonlinear value coupling. The results of this analysis are summarized in Figures 3 – 5.

The prediction step

In the **prediction step**, node a prepares for receiving the new input. This entails computing a new prediction, based on the previously updated posterior beliefs. In general, the prediction of a new mean (or the new mean of the predictive distribution) will depend on whether node a has any value parents, whereas the precision of the new prediction will be influenced by the presence and posterior mean of the node’s volatility parents.

In the general case, the mean and precision of the new prediction are computed as follows:

$$\hat{\mu}_a^{(k)} = \lambda_a \mu_a^{(k-1)} + P_a^{(k)} \quad (34)$$

$$\hat{\pi}_a^{(k)} = \frac{1}{\frac{1}{\pi_a^{(k-1)}} + \Omega_a^{(k)}}, \quad (35)$$

where $P_a^{(k)}$ is the **total predicted drift of the mean** and $\Omega_a^{(k)}$ is the **total predicted volatility** (or step size). In the case of value coupling, the relative contribution of the node’s own previous value and the predicted drift is determined by the parameter λ (see Section 2). Both the total predicted drift as well as the volatility are computed as a sum of a constant (or tonic) term, given by a model parameter, and a time-varying (or phasic) term which is driven by their parents:

The total predicted mean drift equals the sum of constant term ρ_a (the tonic drift

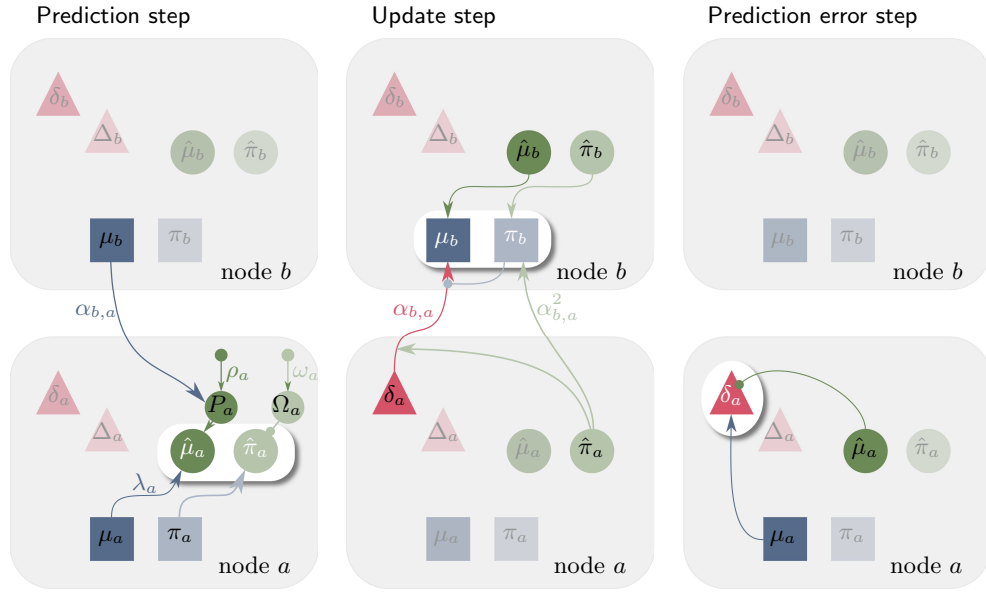


Figure 4. Message-passing for value coupling. Interactions of two nodes, node a and its value parent node b , are shown during the three steps of a trial (Prediction step, left; Update step, middle; Prediction error step, right). The quantities that are being computed in each step are highlighted in white. Note that each step, we only show the computations for either the parent or the child node. Connections with arrowheads indicate positive (excitatory) influences, connections with circular heads indicate negative (inhibitory) influences. Arrows ending on units indicate additive influences, those ending on other arrows indicate multiplicative influences. Each HGF quantity that changes across trials is assigned its own unit. Parameters (α , κ , λ , ω and ρ) determine connection strengths. For clarity, the volatility and drift nodes Ω and P are only shown during the prediction step.

parameter of node a) and the sum of posterior means of all value parents $\mu_{b_i}^{(k-1)}$ (where parents are indexed by i) at $k - 1$, weighted by their connection strengths $\alpha_{b_i,a}$:

$$P_a^{(k)} = t^{(k)} \left(\rho_a + \sum_{i=1}^{N_{vapa}} \alpha_{b_i,a} \mu_{b_i}^{(k-1)} \right), \quad (36)$$

where $t^{(k)}$ denotes the time that has passed between $k - 1$ and k , and N_{vapa} is the number of value parents. Similarly, the total predicted volatility $\Omega_a^{(k)}$ is a function of a constant term ω_a (the tonic volatility parameter) and the posterior means of all volatility parents $\mu_{\tilde{a}_j}^{(k-1)}$ at the previous time point $k - 1$, weighted by their connection strengths $\kappa_{\tilde{a}_j,a}$:

$$\Omega_a^{(k)} = t^{(k)} \exp \left(\omega_a + \sum_{j=1}^{N_{vopa}} \kappa_{\tilde{a}_j,a} \mu_{\tilde{a}_j}^{(k-1)} \right) \quad (37)$$

If node a does not have any parents, both the predicted drift P_a and the predicted volatility Ω_a are fully determined by constant parameters (ρ_a and ω_a) and the time between subsequent observations. In HGFS without drift ($\rho_a = 0$), the predicted mean for the next time step is equal to the posterior mean of the current time step. Equations 34 to 37 nicely reflect the roles that value parents and volatility parents play in the generative model, where value parents model a phasic influence on a child node's mean, and volatility parents model a phasic influence on a child node's step size or volatility.

In sum, the **prediction step** for node a only depends on knowing its own posterior belief from the previous time step and having received its parents' posteriors in time before the new input arrives. The implied message passing for this computational step is visualized in the left panel of Figure 4 for value coupling, and Figure 5 for volatility coupling.

The update step

The **update step** consists of computing a new posterior belief, i.e., a new mean $\mu^{(k)}$ and a new precision $\pi^{(k)}$, given a new input from the level (node) below (usually, a prediction error δ), and the node's own prediction ($\hat{\mu}^{(k)}$ and $\hat{\pi}^{(k)}$). In this case, the exact computations within a node depend on the nature of its children: If node b is the **value parent** of node a , then the following update equations apply to node b :

$$\pi_b^{(k)} = \hat{\pi}_b^{(k)} + \alpha_{b,a}^2 \hat{\pi}_a^{(k)} \quad (38)$$

$$\mu_b^{(k)} = \hat{\mu}_b^{(k)} + \frac{\alpha_{b,a} \hat{\pi}_a^{(k)}}{\pi_b^{(k)}} \delta_a^{(k)} \quad (39)$$

Thus, at the time of the update, node i needs to have access to the following quantities:

Its own prediction: $\hat{\mu}_b^{(k)}, \hat{\pi}_b^{(k)}$

Coupling strength: $\alpha_{b,a}$

From level below: $\delta_a^{(k)}, \hat{\pi}_a^{(k)}$

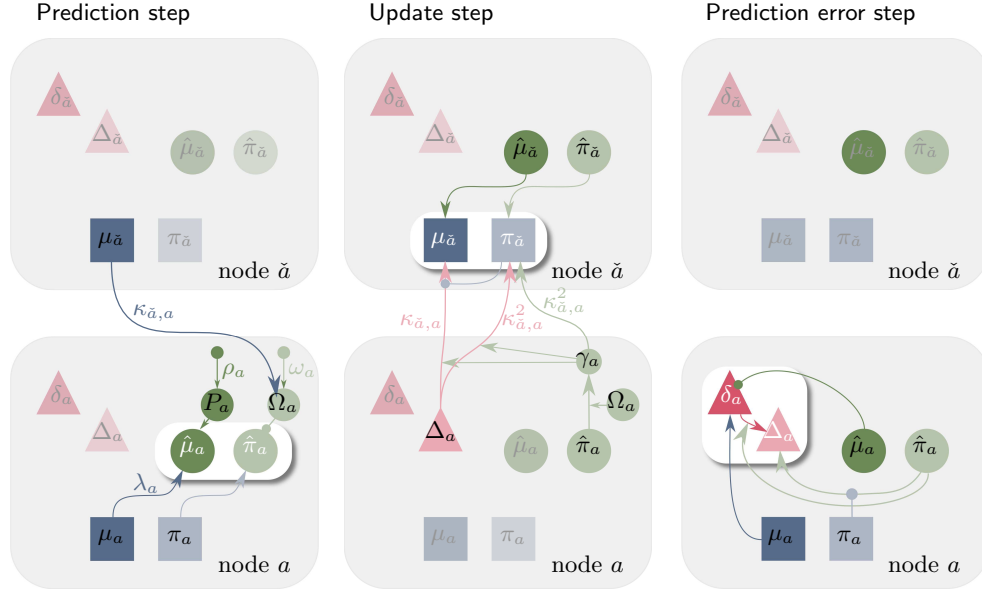


Figure 5. Message-passing for volatility coupling. Interactions of two nodes, node a and its volatility parent node \tilde{a} , are shown during the three steps of a trial (Prediction step, left; Update step, middle; Prediction error step, right). The quantities that are being computed in each step are highlighted in white. Logic of display as in figure 4.

All of these are available at the time of the update. Node b therefore only needs to receive the PE and the precision of the prediction from the child nodes to perform its update. The middle panel of Figure 4 illustrates these computations.

Note that from the equations above, we can define another quantity, the precision-weighted prediction error (pwPE):

$$\psi_{a,b}^{(k)} = \frac{\hat{\pi}_a^{(k)}}{\pi_b^{(k)}} \delta_a^{(k)}. \quad (40)$$

This quantity summarizes the size of the belief update in node b due to changes in node a (before accounting for connection strength) and is often of interest in experimental investigations of neural correlates of prediction errors in belief updating (Iglesias et al., 2013a; Diaconescu et al., 2020; Weber et al., 2020, 2022).

For a node \tilde{a} which is the volatility parent of node a , the update equations for computing a new posterior mean $\mu_{\tilde{a}}^{(k)}$ and a new posterior precision $\pi_{\tilde{a}}^{(k)}$ have been described by Mathys et al. (2011). Here, we will introduce two changes to the notation to simplify the equations themselves and their implementation:

First, we will express the volatility PE, or VOPE, as a function of the previously defined value PE, or VAPE. That means from now on, we will use the symbol δ only for VAPEs:

$$\delta_a^{(k)} \equiv \delta_a^{(k,VAPE)} = \mu_a^{(k)} - \hat{\mu}_a^{(k)}. \quad (41)$$

We use the symbol Δ for VOPEs, which we define as

$$\begin{aligned}\Delta_a^{(k)} &\equiv \delta_a^{(k, VOPE)} := \hat{\pi}_a^{(k)} \left(\frac{1}{\pi_a^{(k)}} + \left(\delta_a^{(k)} \right)^2 \right) - 1 \\ &= \frac{\hat{\pi}_a^{(k)}}{\pi_a^{(k)}} + \hat{\pi}_a^{(k)} \left(\delta_a^{(k)} \right)^2 - 1.\end{aligned}\tag{42}$$

For a derivation of this definition based on the equations given in [Mathys et al. \(2011\)](#), cf. [Appendix 6.2](#).

Second, we will introduce another quantity, which reflects the volatility-weighted precision of the prediction:

$$\gamma_a^{(k)} := \Omega_a^{(k)} \hat{\pi}_a^{(k)},\tag{43}$$

which will be computed as part of the **prediction step** and will be termed **effective precision of the prediction** owing to its role in the update equations. This definition serves to simplify the equations and the corresponding message passing.

With these two definitions, namely those of the VOPE $\Delta^{(k)}$ and of the effective precision of the prediction $\gamma^{(k)}$, the update equations for the precision and the mean of volatility parent \tilde{a} simplify to:

$$\pi_{\tilde{a}}^{(k)} = \hat{\pi}_{\tilde{a}}^{(k)} + \frac{1}{2} \left(\kappa_{\tilde{a},a} \gamma_a^{(k)} \right)^2 + \left(\kappa_{\tilde{a},a} \gamma_a^{(k)} \right)^2 \Delta_a^{(k)} - \frac{1}{2} \kappa_{\tilde{a},a}^2 \gamma_a^{(k)} \Delta_a^{(k)}\tag{44}$$

$$\mu_{\tilde{a}}^{(k)} = \hat{\mu}_{\tilde{a}}^{(k)} + \frac{1}{2} \frac{\kappa_{\tilde{a},a} \gamma_a^{(k)}}{\pi_{\tilde{a}}^{(k)}} \Delta_a^{(k)}\tag{45}$$

This means that at the time of the update, volatility parent node \tilde{a} needs to have access to the following quantities:

Its own prediction: $\hat{\mu}_{\tilde{a}}^{(k)}, \hat{\pi}_{\tilde{a}}^{(k)}$

Coupling strength: $\kappa_{\tilde{a},a}$

From level below: $\Delta_a^{(k)}, \gamma_a^{(k)}$

These equations are illustrated in the middle panel of Figure 5. We note the structural similarities between nodes that serve as value parents and nodes that serve as volatility parents: updates of the mean are always driven by precision-weighted prediction errors, and updates of the precision require some estimate of the prediction of the precision of the child node ($\hat{\pi}_a$ or γ_a). These similarities allow us to make statements about the message passing architecture within and across nodes that generalize across coupling types (see Figure 3B).

An interesting difference to the implied message passing in predictive coding proposals ([Bastos et al., 2012](#); [Shipp, 2016](#)) arises from the **update step**: The HGF architecture requires that not only (precision-weighted) prediction errors are being sent bottom-up between nodes, but also estimates of prediction precision ($\hat{\pi}_a$ or γ_a) which serve to update belief precision in the higher-level node.

The prediction error step

Finally, in the **PE step**, a node computes the deviation of its recently updated posterior from its time step-specific prediction. This can result in two different types of PEs: **VAPes** and **VOPEs**. These will, in turn, be used to communicate with the node's parent nodes, if it has any. Therefore, this step again depends on the nature of a node's parent nodes and can also be considered as the process of gathering all the information required by any existing parents. In addition to the PE, parent nodes will require some estimate of the precision of the prediction (see the previous section on the **update step**).

If node a is the value child of node b , the following quantities have to be sent up to node b :

Precision of the prediction: $\hat{\pi}_a^{(k)}$

Prediction error: $\delta_a^{(k)}$

Node a has already performed the **prediction step** (see above), so it has already computed the precision of the prediction for the current time step, $\hat{\pi}_a^{(k)}$. Hence, in the **PE step**, it needs to perform only the following calculation (illustrated in the right panel of Figure 4):

$$\delta_a^{(k)} = \mu_a^{(k)} - \hat{\mu}_a^{(k)} \quad (46)$$

Note that $\delta_a^{(k)}$ represents a prediction error from the perspective of the parent node - the difference between the expected state of the child and the actual state at time step k . From the perspective of the child node a , the difference between its prior and its posterior instead represents a belief update (Bayesian surprise).

If node a is the volatility child of node \tilde{a} , the following quantities have to be sent up to node \tilde{a} (see also necessary information from level below in a volatility parent's **update step**):

Effective precision of the prediction: $\gamma_a^{(k)}$

Prediction error: $\Delta_a^{(k)}$

Node a has already performed the **prediction step** at the previous time step, so it has already computed the precision of the prediction, $\hat{\pi}_a^{(k)}$, and the total predicted volatility, $\Omega_a^{(k)}$, and out of these the effective precision of the prediction, $\gamma_a^{(k)}$, for the current time step. Hence, in the **PE step**, it needs to perform only the following calculations (illustrated in the right panel of Figure 5):

$$\delta_a^{(k)} = \mu_a^{(k)} - \hat{\mu}_a^{(k)} \quad (47)$$

$$\Delta_a^{(k)} = \frac{\hat{\pi}_a^{(k)}}{\pi_a^{(k)}} + \hat{\pi}_a^{(k)} \left(\delta_a^{(k)} \right)^2 - 1. \quad (48)$$

In other words, if node a has any parents, the **VAPE** will always be computed (as it features in both scenarios), whereas the computation of a **VOPE** is only necessary if node a has a volatility parent.

Our framework allows for multiple parent nodes of either type (e.g., more than one volatility parent). However, it is important to note that all parent nodes of the same type will be sent the same bottom-up prediction errors by their child node. The only difference between the parent nodes is in their coupling strength with the child node. Thus, their relative contribution to the belief about the child node is determined by their starting value and coupling strength. This reflects the fact that the learning agent does not have direct access to these latent states and only learns about them through the information from the sensory inputs (or lower-level beliefs).

As soon as the agent has access to some other source of information about these states, they can be decoupled. For example, if two parent nodes share a child node, but one of them is additionally linked to another child node, the agent can form independent beliefs about the two parent nodes given the different bottom-up signals derived from the child nodes.

Differences for nonlinear value coupling

So far, we have assumed linear value coupling in presenting the computations of value parent and children nodes. In the case of nonlinear **value coupling**, the update equations only change slightly. Specifically, in the **prediction step**, we now have the function g during the computation of the new predicted mean. Assuming that node a is the (nonlinear) value child of nodes $b_{1:N_{vapa}}$, the total predicted mean drift for time step k (previously equation 36) will be

$$P_a^{(k)} = t^{(k)} \left(\rho_a + \sum_{i=1}^{N_{vapa}} \alpha_{b_i,a} g_{b_i,a} \left(\mu_{b_i}^{(k-1)} \right) \right). \quad (49)$$

In other words, the influence of the higher-level belief $\mu_{b_i}^{(k-1)}$ on the prediction of the lower-level belief $\hat{\mu}_a^{(k)}$ is mediated by the function $g_{b_i,a}$, just as we would expect it to be based on the generative model (equation 52).

In the **update step** for the value parent (previously equation 38), we now have:

$$\pi_b^{(k)} = \hat{\pi}_b^{(k)} + \hat{\pi}_a^{(k)} \left(\alpha_{b,a}^2 g' \left(\mu_b^{(k-1)} \right) - \alpha_{b,a} g'' \left(\mu_b^{(k-1)} \right) \delta_a^{(k)} \right) \quad (50)$$

$$\mu_b^{(k)} = \hat{\mu}_b^{(k)} + \frac{\alpha_{b,a} g' \left(\mu_b^{(k-1)} \right) \hat{\pi}_a^{(k)}}{\pi_b^{(k)}} \delta_a^{(k)} \quad (51)$$

Consequently, for the **update step**, the node b now also needs access to its own previous posterior mean $\mu_b^{(k-1)}$. Apart from these changes, all update equations from the previous section apply. This extension allows us to account for the fact that most states in the world interact non-linearly.

Other types of nodes

We have thus far focused our discussion on nodes that represent beliefs about continuous states which evolve in time as random walks (whether Gaussian or auto-regressive). However, the generalized HGF can also accommodate beliefs about any type of state governed by an exponential family distribution by filtering this distribution's sufficient

statistics Mathys & Weber (2020). This makes it possible to track binary and categorical states, where value parents track the probabilities with which one of several possible states is occupied, or with which transitions happen. This can be applied in a range of contexts, from experimental tasks where categorical probabilities must be learned Marshall et al. (2016) to inversions of other classes of discrete state space generative models like the Partially Observable Markov Decision Process models often used in Active Inference (Parr et al., 2022).

Furthermore, at the lowest level of the state hierarchy, we find states that are observable. These typically do not perform random walks, but are instead generated by their parent states independently on every time step. For continuous states in the case of value coupling, this corresponds to setting λ to zero so that the state only depends on its value parent and not on its own past anymore. Similarly, a volatility parent of an outcome state becomes a noise parent - because the variance in the Gaussian distribution no longer corresponds to a step size with respect to a previous mean but instead to the deviation from the current mean which is fully determined by the state’s value parent(s). In other words, an agent using this type of coupling in the generative model forms an explicit and dynamic belief about the level of observation noise (stochasticity) in a particular outcome (noise coupling, see also Example Simulation 2, Figure 7).

Input nodes are directly fed with observations (sensory inputs) (which can be continuous or binary) instead of receiving prediction errors from other nodes. In many applications, we would assume that the lowest level modelled (e.g., primary sensory cortices) is already somewhat distant to the actual sensors (e.g., the retina), which means we can cast its inputs as prediction errors generated during downstream processing input (e.g., in subcortical structures). However, in the case of noise coupling (see Simulation Example 2, Figure 7), and for the algorithmic implementation of our perceptual model, the equations governing these nodes matter. Their treatment is presented in Appendix 6.3.

Summary: A network of nodes

In this section, we have used the update equations of the HGF to propose a conceptualization of the inference machinery as a network of nodes which compute beliefs (i.e., probability distributions) and exchange messages with other nodes. Every node in this network represents an agent’s current belief about a hidden state in its environment, on which it infers given its sensory inputs. Within every node, belief updating in response to a new input proceeds in three steps (an update step, a PE step, and a prediction step).

We have presented the computations for these steps for the two different kinds of coupling that the HGF comprises: value coupling and volatility coupling. While the update equations for volatility coupling have been derived and discussed previously (Mathys et al., 2011, 2014a), approximately Bayes-optimal inference equations for (linear and nonlinear) value coupling under the HGF have not been considered prior to our treatment here. Furthermore, our analysis identifies not only the computations entailed by each computational step, but also the message passing between nodes that is required by each step. This is interesting from a theoretical point of view, where we can compare our architecture to other proposals of belief propagation.

From a practical point of view, the division of the belief updating machinery into subunits (nodes) allows for a modular implementation, where networks can easily be extended and modified by adding or removing nodes, or by changing the type of coupling between nodes, without having to derive the relevant equations for the whole network anew. In two open-source projects, we provide such an implementation (in Python: <https://github.com/ilabcode/pyhgf>, Legrand et al. (in prep); in Julia: <https://github.com/ilabcode/HierarchicalGaussianFiltering.jl>, Thestrup Waade et al.

(in prep)), which allows users to flexibly design their own HGF structures that can be used for simulation and empirical parameter estimation. These tools will also be available as part of the TAPAS software collection (Frässle et al., 2021).

Which conclusions can be drawn with respect to the message passing implied by the HGF? First, while the exact computations performed during the three computational steps depend on the position of the node within the network (e.g., number of children and parent nodes) and the nature of the coupling to other nodes (value vs. volatility coupling), we have identified generic structures in these equations (see Figure 3B), which are of interest from a theoretical point of view, but also facilitate implementation.

For example, belief updates in a node always require messages from lower-level nodes that contain prediction errors (δ_a for value coupling, Δ_a for volatility coupling) and estimates of precision ($\hat{\pi}_a^{(k)}$ in value coupling and $\gamma_a^{(k)}$ in volatility coupling). Similarly, forming a new prediction always entails modifying the mean of the belief by an expectation of drift, and modifying the precision by an expectation of volatility during the next time step. Expectations of drift will be driven by value parents, expectations of volatility by volatility parents, but the structure of the equations is the same for both types of coupling (see equations 34 to 37).

In Figures 4 and 5, we additionally provide a more detailed overview of what happens within and between nodes for specific coupling types. For this purpose, we additionally consider separate subunits for calculations concerning means versus precisions versus prediction errors. Exploring in how far these architectures might map onto structures and networks in biological brains will be an interesting future task.

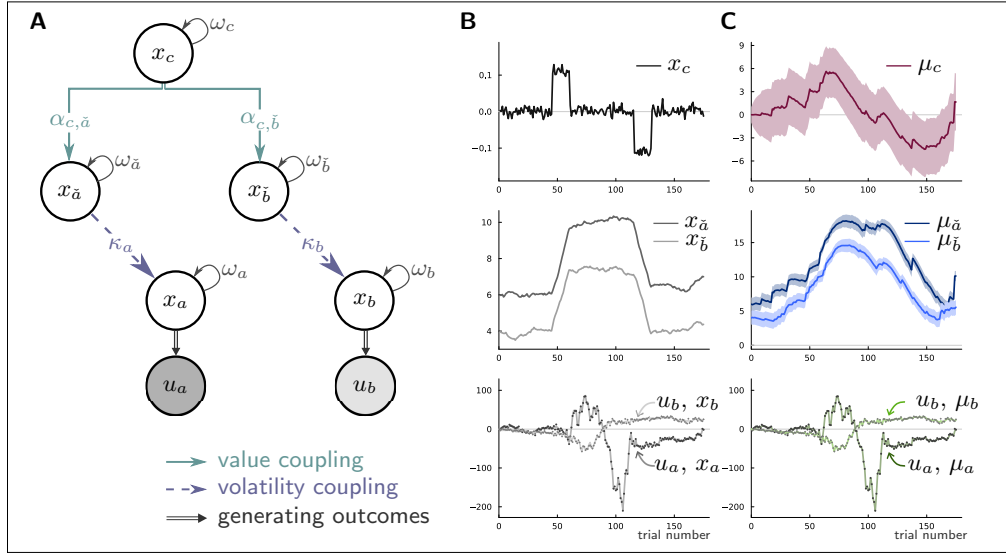


Figure 6. Example simulation 1: Local versus global volatility. **A** Generative model. Global volatility state x_c is a drift value parent to two local volatility states $x_{\tilde{a}}$ and $x_{\tilde{b}}$. **B** Simulated state trajectories (generative model) and observable outcomes. The two bursts in global volatility x_c around trials 30 and 100 (top panel) result in an upward and downward drift in local volatilities, respectively, as seen in the middle panel. States x_a and x_b start off with different levels of local volatility and this difference remains throughout the simulation, demonstrating how a drift parent provides increases and decreases in the value of its children that ride on top of the child state’s mean. **C** Simulated inference. Belief trajectories results from running the belief update equations on the sequence of observations u_a and u_b . The simulated agent correctly infers on the different levels of local volatility in the two hidden states, and also detects the changes in the global volatility state (top panel). Parameters used for this simulations are given in table 1.

4 Examples

In this section, we provide a few example simulations which show the range of generative models our generalization of the HGF encompasses. In the first example, two hidden states separately generate two streams of observations (Figure 6). Each state also has its own phasic volatility parent. However, these volatility states are both influenced by a shared value parent that encodes a higher-level state of global environmental volatility. The simulation shows that the inference network is able to infer the individual as well as the shared (“global”) volatility states. The level of noise in the outcomes (u_a and u_b) was chosen relatively low. This model architecture could be used to explore how beliefs about environmental volatility generalize across different environmental states.

Second, we model a situation where an agent observes two cues (e.g., an auditory and a visual one) that each provide information about a hidden state of interest x_c (such as the location of an object, Figure 7). Both observation sequences are subject to dynamically changing noise (hidden states $x_{\tilde{a}}$ and $x_{\tilde{b}}$). The simulations show that the agent can infer these changing noise levels and adjust its inference on the hidden state x_c according to its current estimate of relative cue reliability.

Finally, in Figure 8, we consider an extension of a model that has found widespread application in decision neuroscience: Participants need to track the probability of a binary outcome over a sequence of observations (e.g., whether a certain choice or stimulus is rewarded or not). The sequence includes reversals and a stable period of $p = 0.5$. In

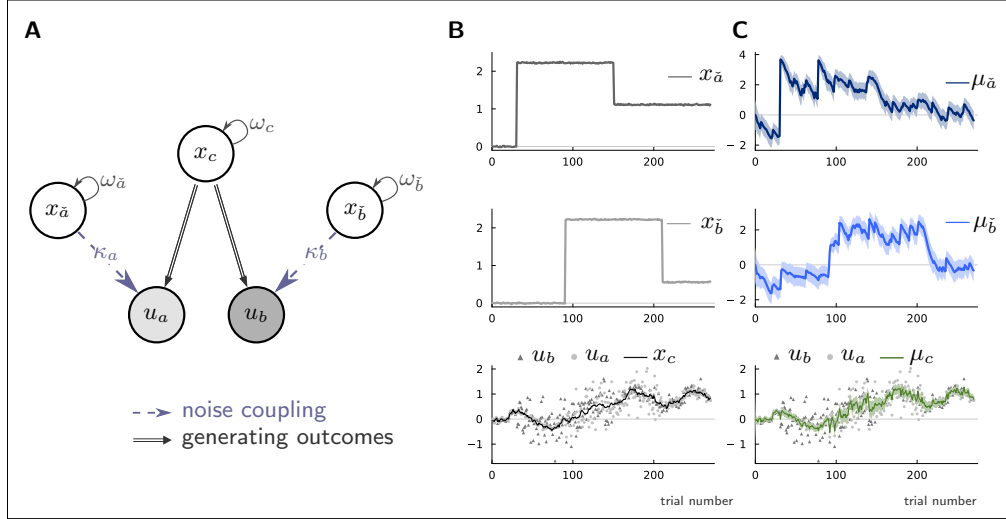


Figure 7. Example simulation 2: Multisensory cue combination with dynamic noise. **A** Generative model. State x_c generates two observations on each trial, u_a and u_b . These could correspond to cues in different modalities, for example a visual and an auditory cue. Both observations are corrupted by noise, the level of which can change from trial to trial according to the hidden noise states x_a and x_b . **B** Simulated state trajectories (generative model) and observable outcomes. Both cues start off with low noise values but go on to experience periods of high and medium noise levels at different times. **C** Simulated inference based on the sequence of observations u_a and u_b . The jumps in the noise are correctly detected (upper and middle panels). When one cue becomes unreliable (e.g., between trials 25 and 90), the inference is driven relatively more by the precise cue. When both cues become noisy, the overall increase in uncertainty is reflected in the simulated agent’s belief precision (lower panel, trials 90 to 150). Parameters used for this simulations are given in table 1.

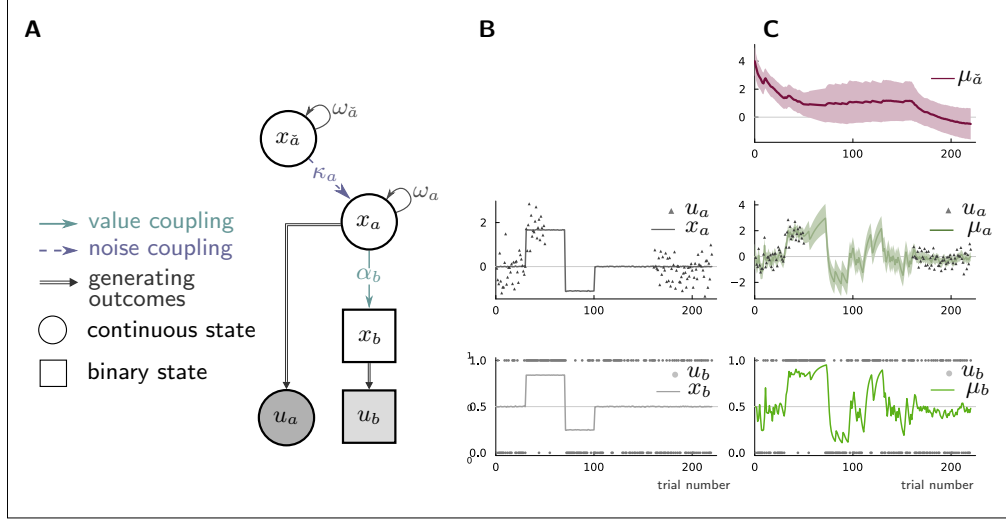


Figure 8. Example simulation 3: Multimodel observations. **A** Generative model. State x_a produces two observations on every trial: continuous observations u_a and, through binary hidden state x_b , binary observations u_b . This could reflect an experiment where the agent has access to binary observations, but also a continuous readout of the probability with which these observations are generated - at least on some trials. The timecourse of this probability can additionally be influenced by a volatility parent $x_{\bar{a}}$. **B** Simulated state trajectories (generative model) and observable outcomes. The trajectory for state x_a was hand-crafted to reflect a typical experimental protocol in decision-making studies ('bandit' tasks, where state u_b corresponds to a reward outcome, and the probability of being rewarded reverses at some points during the task). Between trials 50 and 150 the participant is only presented the binary outcomes. **C** Simulated inference. Belief trajectories result from running the belief update equations on the sequence of observations u_a and u_b . The simulated agent can infer on the hidden state x_a , even in the absence of continuous observations, as long as the jumps/reversals are large. Picking up on the more subtle jump around trial 100 is much harder only based on binary observations. Moreover, when the true probability is around 0.5 (from trial 100 onwards), the agent tends to infer a fluctuating probability as opposed to the stable $p = 0.5$, which improves once they also receive continuous observations (from trial 175), leading to a drop in estimated volatility (top panel). Parameters used for this simulations are given in table 1.

addition to the binary outcomes on every trial, the agent sometimes also has access to a (noisy) sample of the continuous hidden probability state. The simulation shows that inference on small jumps in probability as well as stable periods of 50/50 is difficult based on binary observations alone, but that adding a continuous observation of the probability itself, even if it is only a very noisy readout of the actual probability, stabilizes inference.

| Example 1 | | | Example 2 | | | Example 3 | | |
|------------------------|---------|---------|-----------------------|---------|--------|-----------------------|---------|--------|
| Θ | Process | Model | Θ | Process | HGF | Θ | Process | Model |
| ω_c | n/a | 0 | ω_c | -3 | -3 | $\omega_{\tilde{a}}$ | n/a | -3 |
| $\omega_{\tilde{a}}$ | -3 | -3 | $\omega_{\tilde{a}}$ | n/a | -3 | ω_a | n/a | -3 |
| $\omega_{\tilde{b}}$ | -3 | -3 | $\omega_{\tilde{b}}$ | n/a | -3 | κ_a | 1 | 1 |
| ω_a | -2 | -2 | κ_a | 1 | 1 | α_b | 1 | 1 |
| ω_b | -2 | -2 | κ_b | 1 | 1 | ϵ_a | n/a | -.5 |
| $\alpha_{c,\tilde{a}}$ | 2 | .05 | ϵ_a | -3 | -3 | $x_a^{(0)}$ | n/a | (0, 1) |
| $\alpha_{c,\tilde{b}}$ | 2 | .05 | ϵ_b | -3 | -3 | $x_{\tilde{a}}^{(0)}$ | n/a | (4, 1) |
| κ_a | .5 | .5 | $x_c^{(0)}$ | 0 | (0, 1) | | | |
| κ_b | .5 | .5 | $x_{\tilde{a}}^{(0)}$ | n/a | (0, 1) | | | |
| ϵ_a | 1 | 1 | $x_{\tilde{b}}^{(0)}$ | n/a | (0, 1) | | | |
| ϵ_b | 1 | 1 | | | | | | |
| $x_c^{(0)}$ | n/a | (0, 1) | | | | | | |
| $x_{\tilde{a}}^{(0)}$ | 6 | (6, 1) | | | | | | |
| $x_{\tilde{b}}^{(0)}$ | 4 | (4, 1) | | | | | | |
| $x_a^{(0)}$ | 0 | (0, .5) | | | | | | |
| $x_b^{(0)}$ | 0 | (0, .5) | | | | | | |

Table 1. Parameter values Θ used for the example simulations, for the generative process in the environment as well as the HGF's generative model. Starting states $x^{(0)}$ in the HGF's generative model are Gaussian beliefs with mean and precision (μ_0, π_0) . In all simulations, all drifts ρ were set to zero, and all autoconnection strengths λ were set to 1 (no autoregression). Values are indicated as n/a when state trajectories have been pre-specified instead of simulated, making the parameter value irrelevant. This includes the global volatility drift x_c in example 1, the two noise trajectories $x_{\tilde{a}}$ and $x_{\tilde{b}}$ in example 2, and the probability x_a and its volatility $x_{\tilde{a}}$ in example 3.

5 Conclusions

The work presented here makes several contributions. First, our extension to **value coupling** includes principles of predictive coding in the HGF framework. This offers a general and versatile modelling framework, offering an approximation to optimal Bayesian inference for different types of interactions between states in the world, allowing for inter-individual differences in the dynamics of belief updating, and providing a principled treatment of the multiple forms of uncertainties agents are confronted with.

Second, we present a modular architecture for HGF networks, where beliefs represent nodes that perform three basic computational steps: an **Update** step, a **PE** step, and a **prediction** step, in response to new input. While the equations for these steps differ depending on the coupling of a node to other nodes, we identify a generic structure that allows for a modular implementation, in which nodes can easily be added to or removed from a network, without having to derive the corresponding update equations for the model anew. This feature takes significant load off researchers wanting to apply this modelling framework and to create custom models that suit their experiments. We provide such implementations in two open-source projects (in Python: <https://github.com/ilabcode/pyhgf>, Legrand et al. (in prep); in Julia: <https://github.com/ilabcode/HierarchicalGaussianFiltering.jl>, Thestrup Waade et al. (in prep); inclusion in the TAPAS software collection (Frässle et al., 2021) is pending).

Finally, by considering the case of nonlinear **value coupling** and deriving the message passing scheme implied by this, we enable a formal comparison to other proposed architectures for hierarchical Bayesian inference, most prominently Bayesian (or generalized) predictive coding (Bastos et al., 2012; Shipp, 2016) and belief propagation in active inference (e.g., Fig. 5.1 in Parr et al., 2022).

5.1 Modelling different sources of uncertainty

Whenever agents are faced with observations that violate their expectations, they need to arbitrate between different explanations – has the world changed, requiring an update of beliefs about hidden states, or was the deviation merely due to noise in their observations? As has been shown previously (Mathys et al., 2011, 2014a), the HGF models belief updating in an agent who takes into account several forms of uncertainty for determining the optimal learning rate in the face of new observations: sensory uncertainty (how noisy are the sensory inputs I receive), informational uncertainty (how much do I already know about the hidden state that generates the inputs), and environmental uncertainty (what is the rate of change I expect in the hidden state). All of these together will determine whether (and how much) the agent updates its beliefs about a hidden state in response to unexpected observations.

Here, we show that under the HGF, the agent cannot only learn about environmental volatility – where higher estimates of volatility lead to faster learning, but in an undirected manner –, but also about higher-level hidden states that cause changes in lower-level hidden states in a directed fashion. For example, the weather might be more volatile in some seasons compared to others, making the agent less certain in its predictions (and faster to learn) about the likelihood of rainfall (**volatility coupling**). On the other hand, it might expect more or less rainfall in certain seasons (**value coupling**).

This flexibility in building models of hierarchically interacting states in the world allows for some particularly interesting use cases. In Figure 6, we have provided an example where two hidden states evolve with their own respective evolution rates (both determined by a tonic component and a phasic component), but share a higher-level value parent (“global volatility”) that drives changes in the mean of their respective phasic volatility states. This setup allows for separate estimation of “local” volatility

(specific to each hidden state) and more global influences on volatility. For example, the rate of change in the availability of two different foods in the environment might vary over time and seasons in a way that is specific to each type of food, but when switching to a different environment (or after a global change to the overall climate), the availability of both foods might become more or less stable. Investigating how the brain represents each form of volatility, and thus adjusts learning rates in a modality-specific as opposed to a general manner is an important part of understanding how the brain achieves and maintains the delicate balance of precision across different hierarchical levels (Kanai et al., 2015; Clark, 2013) which appears crucial for mental health (Petzschner et al., 2017; Sterzer et al., 2018).

Finally, agents are confronted not only with phasic variations of volatility (driving changes in the agent’s environmental uncertainty), but also with dynamically changing sensory (or observation) noise. While existing modelling approaches have focused on either accounting for changes in volatility (Behrens et al., 2007; Mathys et al., 2011; Piray & Daw, 2020) or changes in stochasticity or noise (Lee et al., 2020; Nassar et al., 2010), it has recently been pointed out that real-world agents need to be able to detect (and distinguish) changes in both at the same time (Piray & Daw, 2021). In the HGF, dynamic changes in observation noise can be accommodated by hidden states that serve as noise parents to observable outcome states (see Appendix 6.3 for the equations, Figure 7 for an example simulation, and Mikus et al. (2023) for an application to a dataset). Jointly modelling an agent’s inference on volatility and stochasticity will be crucial to understanding the computational origin of maladaptive inference and learning in different psychiatric conditions (Piray & Daw, 2021; Pulcu & Browning, 2019; Mikus et al., 2023).

5.2 Implementing the HGF’s message passing scheme

Hierarchical filtering and predictive coding are two prominent classes of hierarchical Bayesian models that cast perception as inference and model belief updates in proportion to precision-weighted prediction errors. Models from both classes are widely used, both in basic (computational) neuroscience, and for understanding mental disorders in computational psychiatry (Petzschner et al., 2017).

While the message passing architecture implied by different predictive coding models has been examined in detail and partly matched with neuroanatomy and -physiology (for overviews, see Spratling, 2019; Keller & Mrsic-Flogel, 2018; Bastos et al., 2012; Shipp, 2016), and hierarchical filtering models share many similarities with predictive coding models, it is currently not clear whether the respective inference networks would place distinct requirements on implementation (in computers or brains), or make distinct predictions about neural readouts of perceptual inference and learning. This is partly due to their non-overlapping applications: predictive coding models consider hierarchies in which higher levels affect the mean of lower levels, and they are typically used to model inference about static sensory inputs in continuous time.

We reduce this gap by introducing the HGF scheme for **value coupling** alongside **volatility coupling**. Our results show (1) that HGF inference networks for value coupling are largely compatible with recently proposed predictive coding architectures in that messages passed between nodes of the network entail a bottom-up signalling of precision-weighted prediction errors, and a top-down influence on predictions; and (2) that there are slight but interesting differences in the updating of belief uncertainty.

One noteworthy difference between the architectures is that the update equations in the HGF require a bottom-up transmission of lower-level precision estimates (Figures 3 and 4)⁴. This is interesting, given that recent neuroanatomical studies point to additional

⁴It is not surprising that the differences between the models concern the signalling of precision:

pathways besides the classical forward (from lower-level supragranular to higher-level granular layers) and backward (from higher-level infragranular to lower-level extragranular layers) connections (Markov et al., 2013, 2014). Our architecture is compatible with an ascending connection within supragranular layers (for bottom-up communication of lower-level precision) that runs in parallel to a descending connection within these layers (for top-down modulation of PEs by higher-level precision), reminiscent of the “cortical counter streams” identified by these studies. We hope to capitalize on methodological advancements in high-resolution laminar fMRI (Stephan et al., 2019; Haarsma et al., 2022) in future studies to test these predictions.

5.3 Dynamics within versus across time steps

Although potential neurobiological implementations of approximate Bayesian inference in the brain are still hotly debated (Knill & Pouget, 2004; Aitchison & Lengyel, 2017), a growing body of literature suggests that predictive coding-like architectures can account for a large range of neurophysiological findings in perception research (for a recent overview, see Walsh et al., 2020), making these architectures particularly relevant for understanding human perception and inference. It would thus be interesting to examine in detail the commonalities and potential differences between message-passing schemes implied by different forms of coupling in HGF models and classical predictive coding schemes, as we have started to do here.

Importantly, however, direct comparisons of the two models are further complicated by differences in each model’s concept of time: while the HGF captures belief updates across time steps in discrete time (i.e., across sequentially arriving sensory inputs), predictive coding describes the evolution of beliefs in continuous time and, typically, in response to static sensory input (Rao & Ballard, 1999; Friston, 2005)⁵. Differential equations capture the evolution of beliefs and predictions errors and are used to simulate perceptual inference, starting with new input to the lowest level of the belief hierarchy, and ending when the ensuing PEs have been reconciled, i.e., a stable new posterior belief has emerged (Bogacz, 2017). This can be used to make predictions about neural activity that can be compared against measurements. While, to our knowledge, existing predictive coding models have not been fit to data, the simulated neural dynamics display many features that are observed in real data, such as oscillatory tendencies, even in very small networks (Bogacz, 2017). We refer to these simulations as **within-step dynamics** of belief updating.

On the other hand, the generative model of the HGF represents a Markovian process in discrete time; in the inference model, one-step update equations, derived based on a mean field approximation to the full Bayesian solution, quantify the change from prior to posterior on all levels of the belief hierarchy. The model is thus examined in sequential input settings to capture step-by-step learning - in other words, **across-step dynamics** of belief updating. The model provides an approximately Bayes-optimal solution to stepwise belief updating, useful for ideal observer analyses (Stefanics et al., 2018; Weber et al., 2020, 2022; Hauke et al., 2022), but the parameters of the HGF can also be fit to behavioural responses of individual participants. Model-derived agent-specific trajectories of predictions and PEs have proven particularly useful for identifying potential neural and physiological correlates of computational quantities in empirical data (Iglesias et al.,

The HGF derivation explicitly includes update equations for the precision associated with beliefs - as do other hierarchical Bayesian architectures based on Markovian processes (Friston et al., 2013). In contrast, most predictive coding schemes only focus on the optimization of the first moment (mode or expectation) of the posterior distribution for perceptual inference (Friston, 2005; Bogacz, 2017), although the variational approach does allow approximation of the full posterior distribution including its variance.

⁵In fact, the standard form of the generative model in predictive coding does not contain temporal dynamics of the hidden states, but see Friston (2008, 2010) for extensions.

2013b; Vossel et al., 2015; de Berker et al., 2016; Diaconescu et al., 2017; Weinhhammer et al., 2018; Katthagen et al., 2018; Palmer et al., 2019; Deserno et al., 2020; Henco et al., 2020; Cole et al., 2020; Lawson et al., 2021; Hein et al., 2021; Hein & Herrojo Ruiz, 2022; Harris et al., 2022; Fromm et al., 2023). In the future, we propose explicitly to consider potential within-step dynamics of belief updates compatible with the update equations of the HGF (see Weber, 2020, for a first attempt). Such equations could for example be derived by treating the HGF posterior values of all nodes (quantities) as the equilibrium point towards which all dynamics must converge (inspired by Bogacz, 2017). Establishing equations for within-step belief updating dynamics under the HGF might allow for empirical tests of the proposed architecture in a two-step procedure: first, individual trajectories of predictions and prediction errors are inferred from observed behaviour by fitting the HGF to participants’ responses, second, these stepwise point estimates are subsequently used to simulate expected continuous-time neuronal responses according to the differential equations.

A modular implementation of the generalized HGF makes it easy to build large networks with considerable hierarchical depth, opening up exciting possibilities of applying this model architecture in machine learning applications and comparing its performance to alternative neural network architectures (Whittington & Bogacz, 2017; Millidge et al., 2022; Song et al., 2024). When applied as a model of human cognition, however, an increase in model complexity must be matched by sufficiently rich data to successfully fit the model. In the generalised HGF, a branching out into the higher levels of the belief hierarchy (multiple parent nodes) will typically require some amount of branching out in the other direction (multiple child nodes) to disambiguate beliefs about hidden states further up. Moreover, in this paper, we use the original update equations for volatility parents from Mathys et al. (2011). These are derived from a particular quadratic approximation to the variational energy of the parent node which can sometimes lead to a (logically impossible) negative value for the posterior precision of that node. In most cases, this can be circumvented by choosing appropriate priors. However, it can also be avoided entirely by adapting the details of the quadratic approximation. Such adaptations will be included in the generalized HGF as soon as they are published separately.

In summary, we have presented a generalization of the HGF that extends its scope of hierarchical inference mechanisms to include cross-level couplings as proposed by predictive coding. Furthermore, we have demonstrated how this extension can be cast as a modular architecture that allows for flexible changes to a model without having to re-derive update equations. We hope that the availability of these developments as open source software will expand the toolkit of computational psychiatry and facilitate future investigations of perceptual inference in health and disease.

Acknowledgments

This work was supported by the René and Susanne Braginsky Foundation (KES), the Aarhus Universitets Forskningsfond (grant AUFF-E-2019-7-10) (CM), and the Carlsberg Foundation (grant CF21-0439) (CM).

References

Adams, R. A., Napier, G., Roiser, J. P., Mathys, C., & Gilleen, J. (2018). Attractor-like dynamics in belief updating in schizophrenia. *Journal of neuroscience*, 38(44), 9471–9485.

- Adams, R. A., Stephan, K. E., Brown, H. R., Frith, C. D., & Friston, K. J. (2013). The computational anatomy of psychosis. *Frontiers in Psychiatry*, 4(47), 1–26.
- Adelson, E. H. (2005). Checkershadow illusion. 1995. URL <http://web.mit.edu/persci/people/adelson/checkershadow-illusion.html>.
- Aitchison, L., & Lengyel, M. (2017). With or without you: predictive coding and bayesian inference in the brain. *Current opinion in neurobiology*, 46, 219–227.
- Angelaki, D. E., Gu, Y., & DeAngelis, G. C. (2009). Multisensory integration: psychophysics, neurophysiology, and computation. *Current Opinion in Neurobiology*, 19(4), 452–458.
- Bastos, A. M., Usrey, W. M., Adams, R. A., Mangun, G. R., Fries, P., & Friston, K. J. (2012). Canonical Microcircuits for Predictive Coding. *Neuron*, 76(4), 695–711.
- Behrens, T. E. J., Woolrich, M. W., Walton, M. E., & Rushworth, M. F. S. (2007). Learning the value of information in an uncertain world. *Nat Neurosci*, 10, 1214–1221.
- Bogacz, R. (2017). A tutorial on the free-energy framework for modelling perception and learning. *Journal of Mathematical Psychology*, 76, 198–211.
- Clark, A. (2013). Whatever next? Predictive brains, situated agents, and the future of cognitive science. *Behavioral and Brain Sciences*, 36(3), 181–204.
- Cole, D. M., Diaconescu, A. O., Pfeiffer, U. J., Brodersen, K. H., Mathys, C. D., Julkowski, D., Ruhrmann, S., Schilbach, L., Tittgemeyer, M., Vogeley, K., & Stephan, K. E. (2020). Atypical processing of uncertainty in individuals at risk for psychosis. *NeuroImage: Clinical*, 26, 102239.
- Corlett, P. R., Frith, C. D., & Fletcher, P. C. (2009). From drugs to deprivation: A Bayesian framework for understanding models of psychosis. *Psychopharmacology*, 206(4), 515–530.
- Corlett, P. R., Honey, G., Krystal, J., & Fletcher, P. (2011). Glutamatergic model psychoses: Prediction error, learning, and inference. *Neuropsychopharmacology*, 36, 294–315.
- Dayan, P., Hinton, G. E., Neal, R. M., & Zemel, R. S. (1995). The Helmholtz Machine. *Neural Computation*, 7(5), 889–904.
- de Berker, A. O., Rutledge, R. B., Mathys, C., Marshall, L., Cross, G. F., Dolan, R. J., & Bestmann, S. (2016). Computations of uncertainty mediate acute stress responses in humans. *Nature Communications*, 7(1), 10996. Number: 1 Publisher: Nature Publishing Group.
- Deserno, L., Boehme, R., Mathys, C., Katthagen, T., Kaminski, J., Stephan, K. E., Heinz, A., & Schlagenhauf, F. (2020). Volatility Estimates Increase Choice Switching and Relate to Prefrontal Activity in Schizophrenia. *Biological Psychiatry: Cognitive Neuroscience and Neuroimaging*, 5(2), 173–183.
- Diaconescu, A. O., Mathys, C. D., Weber, L. A., Kasper, L., Mauer, J., & Stephan, K. E. (2017). Hierarchical prediction errors in midbrain and septum during social learning. *Social Cognitive and Affective Neuroscience*, 12(4), 618–634.
- Diaconescu, A. O., Wellstein, K. V., Kasper, L., Mathys, C., & Stephan, K. E. (2020). Hierarchical Bayesian models of social inference for probing persecutory delusional ideation. *Journal of Abnormal Psychology*, 129, 556–569. Place: US Publisher: American Psychological Association.
- Doya, K., Ishii, S., Pouget, A., & Rao, R. P. N. (2011). *Bayesian brain: probabilistic approaches to neural coding*. Cambridge: MIT Press.
- Drusko, A., Baumeister, D., McPhee Christensen, M., Kold, S., Fisher, V. L., Treede, R.-D., Powers, A., Graven-Nielsen, T., & Tesarz, J. (2023). A novel computational approach to pain perception modelling within a Bayesian framework using quantitative sensory testing. *Scientific Reports*, 13(1), 3196. Number: 1 Publisher: Nature Publishing Group.

- Ernst, M. O., & Banks, M. S. (2002). Ernst 2002 Humans integrate visual and haptic information in a. *Nature*, *415*(January), 429–433.
- Fletcher, P. C., & Frith, C. D. (2009). Perceiving is believing: a Bayesian approach to explaining the positive symptoms of schizophrenia. *Nature Reviews Neuroscience*, *10*(1), 48–58.
- Friston, K. (2008). Hierarchical Models in the Brain. *PLoS Computational Biology*, *4*(11), e1000211.
- Friston, K., Schwartenbeck, P., Fitzgerald, T., Moutoussis, M., Behrens, T., & Dolan, R. (2013). The anatomy of choice: active inference and agency. *Frontiers in Human Neuroscience*, *7*.
- Friston, K. J. (2005). A theory of cortical responses A theory of cortical responses. *Philosophical transactions of the Royal Society of London. Series B, Biological sciences*, *360*, 815–836.
- Friston, K. J. (2010). The free-energy principle: a unified brain theory? *Nature reviews. Neuroscience*, *11*(2), 127–38.
- Friston, K. J., Brown, H. R., Siemerikus, J., & Stephan, K. E. (2016). The dysconnection hypothesis (2016). *Schizophrenia Research*, *176*(2-3), 83–94.
- Fromm, S., Katthagen, T., Deserno, L., Heinz, A., Kaminski, J., & Schlagenhauf, F. (2023). Belief Updating in Subclinical and Clinical Delusions. *Schizophrenia Bulletin Open*, *4*(1), sgac074.
- Frässle, S., Aponte, E. A., Bollmann, S., Brodersen, K. H., Do, C. T., Harrison, O. K., Harrison, S. J., Heinzle, J., Iglesias, S., Kasper, L., Lomakina, E. I., Mathys, C., Müller-Schrader, M., Pereira, I., Petzschner, F. H., Raman, S., Schöbi, D., Toussaint, B., Weber, L. A., Yao, Y., & Stephan, K. E. (2021). TAPAS: An Open-Source Software Package for Translational Neuromodeling and Computational Psychiatry. *Frontiers in Psychiatry*, *12*, 680811.
- Haarsma, J., Kok, P., & Browning, M. (2022). The promise of layer-specific neuroimaging for testing predictive coding theories of psychosis. *Schizophrenia Research*, *245*, 68–76.
- Hahnloser, R. H., Sarpeshkar, R., Mahowald, M. A., Douglas, R. J., & Seung, H. S. (2000). Digital selection and analogue amplification coexist in a cortex- inspired silicon circuit. *Nature*, *405*(6789), 947–951.
- Harris, D. J., Arthur, T., Vine, S. J., Liu, J., Abd Rahman, H. R., Han, F., & Wilson, M. R. (2022). Task-evoked pupillary responses track precision-weighted prediction errors and learning rate during interceptive visuomotor actions. *Scientific Reports*, *12*(1), 22098. Number: 1 Publisher: Nature Publishing Group.
- Hauke, D. J., Charlton, C. E., Schmidt, A., Griffiths, J., Woods, S. W., Ford, J. M., Srihari, V. H., Roth, V., Diaconescu, A. O., & Mathalon, D. H. (2022). Aberrant hierarchical prediction errors are associated with transition to psychosis: A computational single-trial analysis of the mismatch negativity. Pages: 2022.12.20.22283712. URL <https://www.medrxiv.org/content/10.1101/2022.12.20.22283712v1>
- Hein, T. P., de Fockert, J., & Ruiz, M. H. (2021). State anxiety biases estimates of uncertainty and impairs reward learning in volatile environments. *NeuroImage*, *224*, 117424.
- Hein, T. P., & Herrojo Ruiz, M. (2022). State anxiety alters the neural oscillatory correlates of predictions and prediction errors during reward-based learning. *NeuroImage*, *249*, 118895.
- Helmholtz, H. v. (1860). *Handbuch der physiologischen Optik (Vol. 3)*. English translation (1962): Southall JPC.
- Henco, L., Brandi, M.-L., Lahnakoski, J. M., Diaconescu, A. O., Mathys, C., & Schilbach, L. (2020). Bayesian modelling captures inter-individual differences in social belief computations in the putamen and insula. *Cortex*, *131*, 221–236.

- Iglesias, S., Mathys, C., Brodersen, K., Kasper, L., Piccirelli, M., den Ouden, H., & Stephan, K. (2013a). Hierarchical Prediction Errors in Midbrain and Basal Forebrain during Sensory Learning. *Neuron*, 80(2), 519–530.
- Iglesias, S., Mathys, C. D., Brodersen, K. H., Kasper, L., Piccirelli, M., DenOuden, H. E., & Stephan, K. E. (2013b). Hierarchical Prediction Errors in Midbrain and Basal Forebrain during Sensory Learning. *Neuron*, 80(2), 519–530.
- Kafadar, E., Fisher, V. L., Quagan, B., Hammer, A., Jaeger, H., Mourgues, C., Thomas, R., Chen, L., Imtiaz, A., Sibarium, E., Negreira, A. M., Sarisik, E., Polisetty, V., Benrimoh, D., Sheldon, A. D., Lim, C., Mathys, C., & Powers, A. R. (2022). Conditioned Hallucinations and Prior Overweighting Are State-Sensitive Markers of Hallucination Susceptibility. *Biological Psychiatry*, 92(10), 772–780.
- Kanai, R., Komura, Y., Shipp, S., & Friston, K. J. (2015). Cerebral hierarchies: predictive processing, precision and the pulvinar. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 370(1668), 20140169–20140169.
- Katthagen, T., Mathys, C., Deserno, L., Walter, H., Kathmann, N., Heinz, A., & Schlagenhauf, F. (2018). Modeling subjective relevance in schizophrenia and its relation to aberrant salience. *PLOS Computational Biology*, 14(8), e1006319. Publisher: Public Library of Science.
- Keller, G. B., & Mrsic-Flogel, T. D. (2018). Predictive processing: a canonical cortical computation. *Neuron*, 100(2), 424–435.
- Knill, D. C., & Pouget, A. (2004). The bayesian brain: the role of uncertainty in neural coding and computation. *TRENDS in Neurosciences*, 27(12), 712–719.
- Lawson, R. P., Bisby, J., Nord, C. L., Burgess, N., & Rees, G. (2021). The Computational, Pharmacological, and Physiological Determinants of Sensory Learning under Uncertainty. *Current Biology*, 31(1), 163–172.e4.
- Lawson, R. P., Mathys, C., & Rees, G. (2017). Adults with autism overestimate the volatility of the sensory environment. *Nature Neuroscience*, 20(9), 1293–1299. Number: 9 Publisher: Nature Publishing Group.
- Lecun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436–444.
- Lee, S., Gold, J. I., & Kable, J. W. (2020). The human as delta-rule learner. *Decision*, 7(1), 55.
- Legrand, N., Weber, L., Thestrup Waade, P., Møller, A. H., Allen, M., & Mathys, C. (in prep). pyhgf: The generalized, nodalized and multilevel Hierarchical Gaussian Filter for predictive coding.
- Markov, N. T., Ercsey-Ravasz, M., Van Essen, D. C., Knoblauch, K., Toroczkai, Z., & Kennedy, H. (2013). Cortical high-density counterstream architectures. *Science*, 342(6158).
- Markov, N. T., Vezoli, J., Chameau, P., Falchier, A., Quilodran, R., Huissoud, C., Lamy, C., Misery, P., Giroud, P., Ullman, S., Barone, P., Dehay, C., Knoblauch, K., & Kennedy, H. (2014). Anatomy of hierarchy: Feedforward and feedback pathways in macaque visual cortex. *Journal of Comparative Neurology*, 522(1), 225–259.
- Marshall, L., Mathys, C., Ruge, D., Berker, A. O. d., Dayan, P., Stephan, K. E., & Bestmann, S. (2016). Pharmacological Fingerprints of Contextual Uncertainty. *PLOS Biology*, 14(11), e1002575. Publisher: Public Library of Science.
- Mathys, C., & Weber, L. (2020). Hierarchical Gaussian Filtering of Sufficient Statistic Time Series for Active Inference. In T. Verbelen, P. Lanillos, C. L. Buckley, & C. De Boom (Eds.) *Active Inference*, vol. 1326, (pp. 52–58). Cham: Springer International Publishing. Series Title: Communications in Computer and Information Science.
URL https://link.springer.com/10.1007/978-3-030-64919-7_7

- Mathys, C. D. (2016). How could we get nosology from computation? In *Computational Psychiatry: New Perspectives on Mental Illness*, (pp. 121–135). MIT Press.
URL <https://esforum.de/publications/sfr20/ComputationalPsychiatry.html>
- Mathys, C. D., Daunizeau, J., Friston, K. J., & Stephan, K. E. (2011). A Bayesian foundation for individual learning under uncertainty. *Frontiers in Human Neuroscience*, 5(May), 1–20.
- Mathys, C. D., Lomakina, E. I., Daunizeau, J., Iglesias, S., Brodersen, K. H., Friston, K. J., & Stephan, K. E. (2014a). Uncertainty in perception and the Hierarchical Gaussian Filter. *Frontiers in Human Neuroscience*, 8(November), 1–24.
- Mathys, C. D., Lomakina, E. I., Daunizeau, J., Iglesias, S., Brodersen, K. H., Friston, K. J., & Stephan, K. E. (2014b). Uncertainty in perception and the Hierarchical Gaussian Filter. *Frontiers in Human Neuroscience*, 8.
- Mikus, N., Eisenegger, C., Mathys, C., Clark, L., Müller-Sedgwick, U., Robbins, T., Lamm, C., & Naef, M. (2023). Blocking D2/D3 dopamine receptors in male participants increases volatility of beliefs when learning to trust others. *Nature Communications*, 14.
- Millidge, B., Salvatori, T., Song, Y., Bogacz, R., & Lukasiewicz, T. (2022). Predictive coding: Towards a future of deep learning beyond backpropagation?
URL <https://arxiv.org/abs/2202.09467>
- Nassar, M. R., Wilson, R. C., Heasly, B., & Gold, J. I. (2010). An approximately bayesian delta-rule model explains the dynamics of belief updating in a changing environment. *Journal of Neuroscience*, 30(37), 12366–12378.
- Palmer, C. E., Aukstulewicz, R., Ondobaka, S., & Kilner, J. M. (2019). Sensorimotor beta power reflects the precision-weighting afforded to sensory prediction errors. *NeuroImage*, 200, 59–71.
- Parr, T., Pezzulo, G., & Friston, K. J. (2022). *Active Inference: The Free Energy Principle in Mind, Brain, and Behavior*. The MIT Press.
URL <https://doi.org/10.7551/mitpress/12441.001.0001>
- Petzschner, F. H., Weber, L. A., Gard, T., & Stephan, K. E. (2017). Computational Psychosomatics and Computational Psychiatry: Toward a Joint Framework for Differential Diagnosis. *Biological Psychiatry*, 82(6), 421–430.
- Piray, P., & Daw, N. D. (2020). A simple model for learning in volatile environments. *PLoS computational biology*, 16(7), e1007963.
- Piray, P., & Daw, N. D. (2021). A model for learning based on the joint estimation of stochasticity and volatility. *Nature communications*, 12(1), 6587.
- Powers, A. R., Mathys, C., & Corlett, P. R. (2017). Pavlovian conditioning-induced hallucinations result from overweighting of perceptual priors. *Science (New York, N.Y.)*, 357(6351), 596–600.
- Pulcu, E., & Browning, M. (2019). The misestimation of uncertainty in affective disorders. *Trends in Cognitive Sciences*, 23(10), 865–875.
- Rao, R. P. N., & Ballard, D. H. (1999). Predictive Coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects. *Nature Neuroscience*, 2(1), 79–87.
- Rossi-Goldthorpe, R. A., Leong, Y. C., Leptourgos, P., & Corlett, P. R. (2021). Paranoia, self-deception and overconfidence. *PLOS Computational Biology*, 17(10), e1009453. Publisher: Public Library of Science.
- Sapey-Triomphe, L.-A., Weilhhammer, V. A., & Wagemans, J. (2022). Associative learning under uncertainty in adults with autism: Intact learning of the cue-outcome contingency, but slower updating of priors. *Autism*, 26(5), 1216–1228.

- Sevgi, M., Diaconescu, A. O., Henco, L., Tittgemeyer, M., & Schilbach, L. (2020). Social Bayes: Using Bayesian Modeling to Study Autistic Trait-Related Differences in Social Cognition. *Biological Psychiatry*, 87(2), 185–193.
- Shipp, S. (2016). Neural elements for predictive coding. *Frontiers in Psychology*, 7, 1–21.
- Siegel, J. Z., Curwell-Parry, O., Pearce, S., Saunders, K. E. A., & Crockett, M. J. (2020). A Computational Phenotype of Disrupted Moral Inference in Borderline Personality Disorder. *Biological Psychiatry: Cognitive Neuroscience and Neuroimaging*, 5(12), 1134–1141.
- Song, Y., Millidge, B., Salvatori, T., Lukasiewicz, T., Xu, Z., & Bogacz, R. (2024). Inferring neural activity before plasticity as a foundation for learning beyond backpropagation. *Nature neuroscience*, 27(2), 348–358.
- Spratling, M. (2019). Fitting predictive coding to the neurophysiological data. *Brain research*, 1720, 146313.
- Stefanics, G., Heinzle, J., Horváth, A. A., & Stephan, K. E. (2018). Visual Mismatch and Predictive Coding: A Computational Single-Trial ERP Study. *Journal of Neuroscience*, 38(16), 4020–4030. Publisher: Society for Neuroscience Section: Research Articles.
- Stephan, K. E., Baldeweg, T., & Friston, K. J. (2006). Synaptic Plasticity and Dysconnection in Schizophrenia. *Biological Psychiatry*, 59(10), 929–939.
- Stephan, K. E., & Mathys, C. (2014). Computational approaches to psychiatry. *Current Opinion in Neurobiology*, 25, 85–92. Theoretical and computational neuroscience.
- Stephan, K. E., Petzschner, F. H., Kasper, L., Bayer, J., Wellstein, K. V., Stefanics, G., Pruessmann, K. P., & Heinzle, J. (2019). Laminar fMRI and computational theories of brain function. *NeuroImage*, 197(August 2017), 699–706.
- Sterzer, P., Adams, R. A., Fletcher, P., Frith, C., Lawrie, S. M., Muckli, L., Petrovic, P., Uhlhaas, P., Voss, M., & Corlett, P. R. (2018). The predictive coding account of psychosis. *Biological psychiatry*, 84(9), 634–643.
- Suthaharan, P., Reed, E. J., Leptourgos, P., Kenney, J. G., Uddenberg, S., Mathys, C. D., Litman, L., Robinson, J., Moss, A. J., Taylor, J. R., Groman, S. M., & Corlett, P. R. (2021). Paranoia and belief updating during the COVID-19 crisis. *Nature Human Behaviour*, 5(9), 1190–1202. Number: 9 Publisher: Nature Publishing Group.
- Thestrup Waade, P., Møller, A. H., Comoglio, J., Mikus, N., Legrand, N., Stephan, K. E., Weber, L., & Mathys, C. (in prep). The Generalized Hierarchical Gaussian Filter in Julia.
- Vossel, S., Mathys, C., Stephan, K. E., & Friston, K. J. (2015). Cortical Coupling Reflects Bayesian Belief Updating in the Deployment of Spatial Attention. *Journal of Neuroscience*, 35(33), 11532–11542. Publisher: Society for Neuroscience Section: Articles.
- Walsh, K. S., McGovern, D. P., Clark, A., & O’Connell, R. G. (2020). Evaluating the neurophysiological evidence for predictive processing as a model of perception. *Annals of the New York Academy of Sciences*, 1464(1), 242–268.
- Weber, L. A. (2020). *Perception as Hierarchical Bayesian Inference-Toward non-invasive readouts of exteroceptive and interoceptive processing*. Ph.D. thesis, ETH Zurich. URL: <https://doi.org/10.3929/ethz-b-000476505>.
- Weber, L. A., Diaconescu, A. O., Mathys, C., Schmidt, A., Komater, M., Vollenweider, F., & Stephan, K. E. (2020). Ketamine affects prediction errors about statistical regularities: A computational single-trial analysis of the mismatch negativity. *Journal of Neuroscience*, 40(29), 5658–5668.
- Weber, L. A., Tomiello, S., Schöbi, D., Wellstein, K. V., Mueller, D., Iglesias, S., & Stephan, K. E. (2022). Auditory mismatch responses are differentially sensitive to changes in muscarinic acetylcholine versus dopamine receptor function. *eLife*, 11, e74835.

- Weilnhhammer, V. A., Stuke, H., Sterzer, P., & Schmack, K. (2018). The Neural Correlates of Hierarchical Predictions for Perceptual Decisions. *Journal of Neuroscience*, 38(21), 5008–5021. Publisher: Society for Neuroscience Section: Research Articles.
- Whittington, J. C. R., & Bogacz, R. (2017). An Approximation of the Error Backpropagation Algorithm in a Predictive Coding Network with Local Hebbian Synaptic Plasticity. *Neural Computation*, 29(5), 1229–1262.

6 Appendix

6.1 Approximate inversion for value coupling

In [Mathys et al. \(2011\)](#), we presented a variational approximation to the exact Bayesian inversion of our generative model which employed a mean-field approximation, and derived analytic one-step update equations using a new quadratic approximation to the variational energies. Following this procedure for the case of **value coupling**, we specify the variational energy for a value parent to derive the update equations in the main text.

The generative model for a state x_a with a (non)linear value parent x_b (and a volatility parent $x_{\tilde{a}}$) is given by⁶

$$x_a^{(k)} \sim \mathcal{N}\left(x_a^{(k-1)} + \alpha_{b,a}g\left(x_b^{(k)}\right), \exp\left(\kappa_{\tilde{a},a}x_{\tilde{a}}^{(k)} + \omega_a\right)\right), \quad (52)$$

where the **value coupling** between x_a and x_b is mediated by function g , which can be nonlinear.

Using the mean-field approximation as in [Mathys et al. \(2011\)](#), the variational energy and its first two derivatives for the value parent x_b are given by

$$\begin{aligned} I\left(x_b^{(k)}\right) = & -\frac{1}{2}\hat{\pi}_a^{(k)}\left(\frac{1}{\pi_a^{(k)}} + \left(\mu_a^{(k)} - \left(\mu_a^{(k-1)} + g\left(x_b^{(k)}\right)\right)\right)^2\right) \\ & -\frac{1}{2}\hat{\pi}_b^{(k)}\left(x_b^{(k)} - \mu_b^{(k-1)}\right)^2 + \text{const.} \end{aligned} \quad (53)$$

$$\begin{aligned} I'\left(x_b^{(k)}\right) = & \hat{\pi}_a^{(k)}g'\left(x_b^{(k)}\right)\left(\mu_a^{(k)} - \left(\mu_a^{(k-1)} + g\left(x_b^{(k)}\right)\right)\right) \\ & -\hat{\pi}_b^{(k)}\left(x_b^{(k)} - \mu_b^{(k-1)}\right) \end{aligned} \quad (54)$$

$$\begin{aligned} I''\left(x_b^{(k)}\right) = & \hat{\pi}_a^{(k)}\left(g''\left(x_b^{(k)}\right)\left(\mu_a^{(k)} - \left(\mu_a^{(k-1)} + g\left(x_b^{(k)}\right)\right)\right)\right) \\ & -g'\left(x_b^{(k)}\right)^2 - \hat{\pi}_b^{(k)} \end{aligned} \quad (55)$$

We calculate the mean and precision of the Gaussian posterior for the value parent $x_b^{(k)}$ using the rules as stated in [Mathys et al. \(2011\)](#) (equations 38 and 40 there), which follow a quadratic approximation to the variational energy with expansion point at the posterior belief $\mu_b^{(k-1)}$ from the previous time step. For this, we need the derivatives of the variational energies at this point:

$$I'\left(\mu_b^{(k-1)}\right) = \hat{\pi}_a^{(k)}g'\left(\mu_b^{(k-1)}\right)\left(\mu_a^{(k)} - \left(\mu_a^{(k-1)} + g\left(\mu_b^{(k-1)}\right)\right)\right) \quad (56)$$

Here, we identify the prediction of the mean $\hat{\mu}_a^{(k)}$ about the value child state x_a as

$$\hat{\mu}_a^{(k)} = \mu_a^{(k-1)} + g\left(\mu_b^{(k-1)}\right) \quad (57)$$

and thus the prediction error about x_a as

$$\delta_a^{(k)} = \mu_a^{(k)} - \left(\mu_a^{(k-1)} + g\left(\mu_b^{(k-1)}\right)\right). \quad (58)$$

Therefore, the first derivative of the variational energy becomes

$$I'\left(\mu_b^{(k-1)}\right) = \hat{\pi}_a^{(k)}g'\left(\mu_b^{(k-1)}\right)\delta_a^{(k)}. \quad (59)$$

⁶Note that for brevity, we are omitting all priors here - strictly speaking, these equations only form a generative model if combined with appropriate priors on the model parameters and the initial states.

Similarly, the second derivative then reads:

$$I'' \left(\mu_b^{(k-1)} \right) = \hat{\pi}_a^{(k)} \left(g'' \left(\mu_b^{(k-1)} \right) \delta_a^{(k)} - g' \left(\mu_b^{(k-1)} \right)^2 \right) - \hat{\pi}_b^{(k)} \quad (60)$$

With these, we can specify the update equations for the precision π_b and the mean μ_b of the value parent (see Mathys et al. (2011) and Appendix B of Mathys et al. (2014a)):

$$\begin{aligned} \pi_b^{(k)} &= -I'' \left(\mu_b^{(k-1)} \right) \\ &= \hat{\pi}_b^{(k)} + \hat{\pi}_a^{(k)} \left(g' \left(\mu_b^{(k-1)} \right)^2 - g'' \left(\mu_b^{(k-1)} \right) \delta_a^{(k)} \right) \end{aligned} \quad (61)$$

$$\begin{aligned} \mu_b^{(k)} &= \hat{\mu}_b^{(k)} + \frac{I' \left(\mu_b^{(k-1)} \right)}{\pi_b^{(k)}} \\ &= \hat{\mu}_b^{(k)} + \frac{\hat{\pi}_a^{(k)} g' \left(\mu_b^{(k-1)} \right)}{\pi_b^{(k)}} \delta_a^{(k)} \end{aligned} \quad (62)$$

If $g(x) = x$ (linear value coupling), then $g'(x) = 1$ and $g''(x) = 0$, and we obtain the update equations specified in section 3.

6.2 Definition of a VOPE

In the main text, we introduced a new definition Δ of the volatility prediction error or VOPE, which we express as a function of the previously defined value prediction error δ , or VAPE. Here, we show how our new definition derives from the definition contained in earlier work (Mathys et al., 2011):

$$\begin{aligned} \Delta_a^{(k)} \equiv \delta_a^{(k, VOPE)} &:= \frac{\frac{1}{\pi_a^{(k)}} + \left(\mu_a^{(k)} - \hat{\mu}_a^{(k)} \right)^2}{\frac{1}{\pi_a^{(k-1)}} + \Omega_a^{(k)}} - 1 \\ &= \hat{\pi}_a^{(k)} \left(\frac{1}{\pi_a^{(k)}} + \left(\mu_a^{(k)} - \hat{\mu}_a^{(k)} \right)^2 \right) - 1 \\ &= \hat{\pi}_a^{(k)} \left(\frac{1}{\pi_a^{(k)}} + \left(\delta_a^{(k)} \right)^2 \right) - 1 \\ &= \frac{\hat{\pi}_a^{(k)}}{\pi_a^{(k)}} + \hat{\pi}_a^{(k)} \left(\delta_a^{(k)} \right)^2 - 1. \end{aligned} \quad (63)$$

From the first to the second line, we have used the following definition:

$$\hat{\pi}_a^{(k)} := \frac{1}{\frac{1}{\pi_a^{(k-1)}} + \Omega_a^{(k)}}.$$

This ensures that a given node does not need to have access to the posterior precision from the level below: $\pi_a^{(k-1)}$, which facilitates implementation.

In sum, we are introducing a second prediction error unit Δ which is concerned with deviations from predicted uncertainty and is informed by value prediction errors and estimates of uncertainty. It is this prediction error - a function of the squared value prediction error - which communicates between a node and its volatility parent.

6.3 Computations of other types of nodes

In the main text, we focus on continuous state nodes (i.e., states performing a Gaussian or auto-regressive random walk). We now also specify the nodalized implementation of inference for binary (and, by extension, categorical) belief nodes as well as input nodes (corresponding to observable/outcome states).

Input nodes differ from regular nodes in that the inputs do not perform a random walk ($\lambda = 0$), but are noisily generated by peripheral states at each time step (see section 2 of the main text). We refer to the observations or sensory stimuli that enter the network as “inputs” and therefore call the receiving nodes input nodes. However, from the perspective of the generative model, these are the observable outputs of the network.

The input nodes are important elements in the HGF belief network. Processes which need to take place in an input node at a given time step are:

- Receive a new input and store it
- Either receive as a second input the exact time interval since the previous input, or infer the time as ‘previous plus 1’ (e.g., next time step)
- Compute all quantities which need to be signalled to the parent node (e.g., prediction error)
- Send these quantities to the parent node
- Receive top-down messages from the parent node (e.g., $\hat{\mu}$)
- Compute surprise (i.e., the negative logarithm of the input’s probability given the prediction)

The quantities being signalled bottom-up, and the computation of surprise, depend on the nature of the input node (continuous or binary) and on the nature of the coupling with the parent.

Because input nodes are different from HGF state nodes, but rather serve as a relay station for the input and for computing surprise, and because they capture any observation noise that might be inherent in the input, the message passing and the within-node computations differ from the generic scheme presented in the main text.

Continuous input nodes

A continuous input node receives inputs u which can be any real number. In terms of the generative model, we think of these inputs as being sampled from a Gaussian distribution with a mean determined by the state node it is coupled to and a variance which is either constant or determined by another HGF node. This variance is the observation noise.

As with the coupling types introduced in the main text, we call the state node that determines the input’s mean its **value parent**. However, the state node which represents the phasic component of the input’s variance (or observation noise), is not a **volatility parent**, since the input has no volatility because it is not a state. Instead, we call such a parent node a **noise parent**. Every continuous input node has one **value parent**, but having a **noise parent** is optional. When noise is not determined by a noise parent, it is a constant parameter of the input node.

Value parents of continuous input nodes

The predicted mean of an input node is simply the prediction from the value parent p :

$$\hat{\mu}_u^{(k)} = \hat{\mu}_p^{(k)} \quad (64)$$

Since the mean parent might have a drift parameter, the current prediction can only be computed once the new input has arrived. Then it needs to be signalled top-down immediately.

In the absence of a noise parent, the precision of the prediction for the input node is fully determined the input node's noise parameter ε :

$$\hat{\pi}_u^{(k)} = \frac{1}{\exp(\varepsilon_u)} \quad (65)$$

However, in the presence of a noise parent q , this will additionally depend on the posterior $\mu_q^{(k-1)}$ of that parent at the previous time step, and the coupling parameter $\kappa_{q,u}$ of the input node with its noise parent q :

$$\hat{\pi}_u^{(k)} = \frac{1}{\exp\left(\kappa_{q,u}\mu_q^{(k-1)} + \varepsilon_u\right)}. \quad (66)$$

In the **update step**, the posterior mean of the input node is the input itself (the posterior precision is not required, but would be infinite, as the input is known):

$$\mu_u^{(k)} = u^{(k)} \quad (67)$$

Finally, in the **PE step**, the value PE (or VAPE) will be computed as the difference the prediction and the posterior:

$$\delta_u^{(k)} = \mu_u^{(k)} - \hat{\mu}_u^{(k)} = u^{(k)} - \hat{\mu}_u^{(k)} \quad (68)$$

This means that prior to the update of the input node, it needs to receive the current prediction $\hat{\mu}_p^{(k)}$ of its parent node.

The update of the value parent node will look like the regular value coupling updates from previous chapters:

$$\pi_p^{(k)} = \hat{\pi}_p^{(k)} + \hat{\pi}_u^{(k)} \quad (69)$$

$$\mu_p^{(k)} = \hat{\mu}_p^{(k)} + \frac{\hat{\pi}_u^{(k)}}{\pi_p^{(k)}} \delta_u^{(k)} \quad (70)$$

This means that the input node needs to signal bottom-up to its mean parent:

Precision of the prediction: $\hat{\pi}_u^{(k)}$

Prediction error: $\delta_u^{(k)}$.

The implicit assumption here is that the connection between a continuous input node and its value parent has a connection weight of $\alpha = 1$. Should a use case arise where this is inconvenient, it can easily be changed to be a variable parameter.

It may seem slightly artificial to construct the computational steps for the input node in this way since the actual belief about the input is represented in the parent node. However, this allows for a more modular implementation where the value parent can remain agnostic as to whether its child is another state node or a continuous input node.

Finally, to compute the surprise associated with the current input, the node needs to compute the negative logarithm of the probability of input $u^{(k)}$ under a Gaussian prediction with $\hat{\mu}_u^{(k)}$ as mean and $\hat{\pi}_u^{(k)}$ as the precision:

$$-\log\left(p\left(u^{(k)}\right)\right) = \frac{1}{2} \left(\log(2\pi) - \log\left(\hat{\pi}_u^{(k)}\right) + \hat{\pi}_u^{(k)} \left(u^{(k)} - \hat{\mu}_p^{(k)}\right)^2 \right). \quad (71)$$

Noise parents of continuous input nodes

Having a noise parent for a continuous input node means that a noise PE (or NOPE) will be computed and signalled bottom-up during the PE step. We denote this PE with the symbol ϵ_u . Importantly, the NOPE (as opposed to the VOPE) is not a direct function of the VAPE. Instead, both the posterior precision as well as the posterior mean are taken from the value parent p :

$$\epsilon_u^{(k)} = \frac{\hat{\pi}_u^{(k)}}{\pi_p^{(k)}} + \hat{\pi}_u^{(k)} \left(u^{(k)} - \mu_p^{(k)} \right)^2 - 1. \quad (72)$$

This in turn requires that the update of the value parent happens before the computation of the NOPE, and the posterior of the value parent is already available to the input node.

The **update step** for the noise parent is similar to the update in volatility parents (equations 45) with a modified prediction error and an effective precision term γ_u fixed to 1:

$$\mu_q^{(k)} = \hat{\mu}_q^{(k)} + \frac{1}{2} \frac{\kappa_{q,u} \gamma_u^{(k)}}{\pi_q^{(k)}} \epsilon_u^{(k)} \quad (73)$$

$$= \hat{\mu}_q^{(k)} + \frac{1}{2} \frac{\kappa_{q,u}}{\pi_q^{(k)}} \epsilon_u^{(k)}. \quad (74)$$

This similarity again means that the parent node can remain agnostic as to whether it serves as a volatility or a noise parent - as long as the input node also signals a value of 1 as the effective precision term γ at every time step. Importantly, this also works for the update of the precision of the noise parent. Setting γ_u to 1 (and replacing the VOPE Δ with the NOPE ϵ and *vopa* with q) in the previously established precision update for volatility parents (equation 45) leads to:

$$\pi_q^{(k)} = \hat{\pi}_q^{(k)} + \frac{1}{2} \left(\kappa_{q,u} \gamma_u^{(k)} \right)^2 + \left(\kappa_{q,u} \gamma_u^{(k)} \right)^2 \epsilon_u^{(k)} - \frac{1}{2} \kappa_{q,u}^2 \gamma_u^{(k)} \epsilon_u^{(k)} \quad (75)$$

$$= \hat{\pi}_q^{(k)} + \frac{1}{2} (\kappa_{q,u})^2 + (\kappa_{q,u})^2 \epsilon_u^{(k)} - \frac{1}{2} \kappa_{q,u}^2 \epsilon_u^{(k)} \quad (76)$$

$$= \hat{\pi}_q^{(k)} + \frac{1}{2} (\kappa_{q,u})^2 + \frac{1}{2} (\kappa_{q,u})^2 \epsilon_u^{(k)} \quad (77)$$

$$= \hat{\pi}_q^{(k)} + \frac{1}{2} (\kappa_{q,u})^2 \left(1 + \epsilon_u^{(k)} \right). \quad (78)$$

Peculiarities of continuous input nodes and consequences for their parents

The possibility for a given state node to be the value parent or the noise parent of a continuous input node has a number of consequences for the implementation of state nodes:

First, owing to the dependence of the NOPE on the posterior beliefs of the mean parent, the continuous input node needs to communicate with its value parent first and wait for the posteriors to be computed there and sent top-down in order to trigger an update in its noise parent.

Second, the value parent needs to send top-down not only the posterior mean, but also the posterior precision, for the same reason.

Third, the connection weight for value connections will always be $\alpha = 1$.

Fourth, for issuing a new prediction $\hat{\mu}_u$, the node needs to receive the predicted mean of its value parent at the beginning of a new time step. This means it must be possible to elicit a new prediction in regular **hgf** nodes without actually sending a prediction error, instead by only sending a new time point. The **hgf** node needs to react to this by

sending top-down the new predicted mean, such that the input node can compute the PE and signal it back bottom-up for an update.

Thus, the steps for a continuous input node are:

- receive input u
- determine time of input
- send bottom-up to value parent: time of input (to elicit a prediction)
- receive top-down: predicted mean $\hat{\mu}_p$
- compute prediction $\hat{\mu}_u$ and retrieve $\hat{\pi}_u$
- compute surprise using u , $\hat{\mu}_u$ and $\hat{\pi}_u$
- compute VAPE using u and $\hat{\mu}_u$
- send bottom-up to value parent: VAPE, $\hat{\pi}_u$, and time
- receive top-down: posteriors μ_p and π_p
- if relevant, compute NOPE using u , $\hat{\pi}_u$, μ_p and π_p
- send bottom-up to noise parent: NOPE, $\gamma_u = 1$, and time
- receive top-down: posterior μ_q
- compute new precision of its prediction $\hat{\pi}_u^{(k+1)}$ using its tonic observation noise ε and, if present, the posterior mean of its noise parent $\mu_q^{(k)}$.

The value parent of a continuous input node needs to

1. be able to elicit new predictions based on time input
2. send new predictions top-down immediately in the case of a continuous input node child
3. send down not only its posterior mean, but also the precision after each update.

Binary input nodes

Binary input nodes serve to receive inputs that can only take on one of two values. These input nodes can only have one value parent because their stochastic properties are fully described by a Bernoulli distribution which only has one parameter. The value parents binary input nodes are binary state nodes, which are special cases of state nodes which themselves only have value parents. This implementation results in the value parents of binary HGF nodes being regular state nodes which can be agnostic as to whether their child node is a regular state node, a binary HGF node, or a continuous input node.

For binary input nodes, the observation noise is given by their noise parameter ε . Therefore, the precision of the input prediction $\hat{\pi}_u$ is constant (i.e., we can treat it as a parameter). We here only present the case without observation noise (i.e., $\varepsilon_u = 0$ or $\hat{\pi}_u = \text{inf}$), and leave the case of finite precision for a future treatment.

In general, the steps for a binary input node are:

- receive input u

- determine time of input
- compute prediction errors, if necessary
- send bottom-up: u , input precision, and time
- receive top-down: prediction of parent $\hat{\mu}_{pa}$
- compute surprise based on message from parent.

If $\hat{\pi}_u$ is infinite, then the bottom-up messages are simply this precision $\hat{\pi}_u$ itself, and u . The surprise computation is also very simple:

$$surprise^{(k)} = \begin{cases} -\log \left(1 - \hat{\mu}_{pa}^{(k)} \right), & \text{for } u^{(k)} = 1 \\ -\log \left(\hat{\mu}_{pa}^{(k)} \right), & \text{for } u^{(k)} = 0. \end{cases} \quad (79)$$

The special cases that follow for the update of the parent node are restricted to binary HGF nodes, which therefore represent their own special case of HGF nodes.

Binary state nodes

Binary nodes are parents of binary input nodes. Their cycle starts with receiving a bottom-up message from their child node, which, first of all, needs to trigger the **prediction step**. Similarly to continuous input nodes, the predictions of a binary HGF node depend on its parent's predictions (Adams et al., 2018):

$$\hat{\mu}_{bin}^{(k)} = \frac{1}{1 + \exp \left(-\kappa_{bin} \hat{\mu}_{pa}^{(k)} \right)} \quad (80)$$

$$\hat{\pi}_{bin}^{(k)} = \frac{1}{\hat{\mu}_{bin}^{(k)} \cdot (1 - \hat{\mu}_{bin}^{(k)})}. \quad (81)$$

The precision of the prediction is a direct function of the mean owing to the binary nature of the state.

Again, we need to introduce an additional top-down signalling step at the beginning of each time step, where the parent node sends down its current prediction of the mean, given the time interval since the last input.

The bottom-up message that binary state nodes receive consists of three quantities ($\hat{\pi}_u$, $u^{(k)}$, and the time of the input), and the time since the last input). The updates read:

$$\mu_{bin}^{(k)} = u^{(k)} \quad (82)$$

$$\pi_{bin}^{(k)} = \hat{\pi}_u \quad (83)$$

Finally, in the **PE step**, the binary node computes a VAPE for its value parent:

$$\delta_{bin}^{(k)} = \mu_{bin}^{(k)} - \hat{\mu}_{bin}^{(k)}. \quad (84)$$

The parent node will also perform its **update step** according to the following equations (Mathys et al., 2014b; Adams et al., 2018):

$$\pi_{pa}^{(k)} = \hat{\pi}_{pa}^{(k)} + \frac{\kappa_{bin}^2}{\hat{\pi}_{bin}^{(k)}}. \quad (85)$$

$$\mu_{pa}^{(k)} = \hat{\mu}_{pa}^{(k)} + \frac{\kappa_{bin}}{\pi_{pa}^{(k)}} \delta_{bin}^{(k)}. \quad (86)$$

For the implementation, this means that we can either give the HGF node knowledge about who its child is and let the exact update depend on that, or we can let this be solved by the value connection, in which case this connection would need to signal the precision weight that is used for the mean update separately from the term that is used for the precision update.

In any case, the information which needs to be sent bottom-up from a binary HGF node to its value parent is:

Prediction error: $\delta_{bin}^{(k)}$

Predicted precision: $\hat{\pi}_{bin}^{(k)}$

In case the parent does not have knowledge about its child, the information would have to be sent in the following form:

Prediction error: $\delta_{bin}^{(k)}$

Precision weight for mean update: 1

Precision term for precision update: $\frac{1}{\hat{\pi}_{bin}^{(k)}}$

Summary: Implementational consequences

Due to the special cases of continuous input nodes and binary state nodes, which both can be potential children of regular state nodes, we need to introduce a few changes to the update and connection logic of the regular state nodes:

- State nodes need to emit new predictions if prompted by receiving information about the time of the new input, and send this prediction top-down. This is needed both for the computation of surprise in the continuous input nodes, but also for the computation of prediction error in continuous input nodes, and for the computation of predictions in binary state nodes.
- In the case of value parents of continuous input nodes, state nodes need to signal top-down not only their posterior mean, but also their posterior precision.
- Value-coupling connections need separately to signal bottom-up the precision weight of the upcoming prediction error, and the precision term needed to update the parent's precision.
- Implementing noise and volatility connections in the same way allows for an implementation where regular state nodes are completely unaware about which kind of node their child is. The computation necessary for the precision update, which is more elaborate in volatility and noise coupling is then part of the connection logic.

Everything else that is unusual about the computations within binary input nodes, continuous input nodes, and binary state nodes can then be implemented within these nodes without affecting the regular continuous node implementation.

6.4 Notation overview

Table 2 provides an overview of the notation used throughout this paper.

| Variable | Notation | Node type |
|---|----------------|-------------------------|
| Free parameters Θ | | |
| coupling strength | κ | all nodes |
| tonic volatility | ω | continuous state nodes |
| tonic drift | ρ | |
| autoconnection strength | λ | |
| initial mean | μ_0 | |
| initial precision | π_0 | |
| tonic input noise | ε | continuous input nodes |
| bias | b | |
| Belief states θ | | |
| <i>Prediction step</i> | | |
| prediction mean | $\hat{\mu}$ | all nodes |
| prediction precision | $\hat{\pi}$ | |
| effective precision | γ | continuous state nodes |
| total predicted volatility | Ω | |
| total predicted drift | P | |
| time since last input | τ | |
| implied learning rate | ν | categorical state nodes |
| predicted category probabilities | ξ | |
| <i>Belief update step</i> | | |
| posterior mean | μ | state nodes |
| posterior precision | π | |
| input value | u | input nodes |
| <i>Prediction error step</i> | | |
| value prediction error | δ | all nodes |
| precision prediction error | Δ | |
| precision-weighted value prediction error | ψ | |
| precision-weighted precision prediction error | Ψ | |
| surprise | \mathfrak{S} | |

Table 2. List of variables in the HGF and their notation. This includes the free parameters χ , as well as various changing belief states.