Message Passing in the Hierarchical Gaussian Filter

October 22, 2018

1 Computations of Nodes in the HGF

In the HGF, each node has to perform several computations within an experimental trial, which can be ordered in time as shown in the box:

NODE i AT TRIAL k

(`compute` prediction$_i^{(k)}$)
$\leftarrow$ `receive` PE$_{i-1}^{(k)}$ from node$_{i-1}$

UPDATE step
`compute` posterior$_i^{(k)}$
given: PE$_{i-1}^{(k)}$ and prediction$_i^{(k)}$
$\rightarrow$ `send` posterior$_i^{(k)}$ to node$_{i-1}$

PE step
`compute` PE$_i^{(k)}$
given: prediction$_i^{(k)}$ and posterior$_i^{(k)}$
$\rightarrow$ `send` PE$_i^{(k)}$ to node$_{i+1}$
$\leftarrow$ `receive` posterior$_{i+1}^{(k)}$ from node$_{i+1}$

PREDICTION step
`compute` prediction$_i^{(k+1)}$
given: posterior$_i^{(k)}$ and posterior$_{i+1}^{(k)}$

The exact computations in each step depend on the nature of the coupling (via VAPEs vs. VOPEs) with the parent and children nodes and will be

outlined in the following two chapters.

Note that we have placed the PREDICTION step in the end of a trial. This is because usually, we think about the beginning of a trial as starting with receiving a new input, and of a prediction as being present before that input is received. However, in some variants of the HGF the prediction also depends on the time that has passed in between trials, which is something that can only be evaluated once the new input arrives - hence the additional computation of the (current) prediction in the beginning of the trial. Conceptually, it makes most sense to think of the prediction as happening continuously between trials. For implementational purposes, it is however most convenient to only 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.

2 Computations for VAPE coupling

The exact computations of the UPDATE step depend on the nature of the coupling with the child node(s), while both the PE step and the PREDICTION step depend on the coupling with the parent node(s).

2.1 Update Step

If Node i is the value parent of Node $i-1$, then the following update equations apply to Node i:

$$\pi_i^{(k)} = \hat{\pi}_i^{(k)} + \alpha_{i-1,i}^2 \hat{\pi}_{i-1}^{(k)}$$
$$\mu_i^{(k)} = \hat{\mu}_i^{(k)} + \frac{\alpha_{i-1,i}^2 \hat{\pi}_{i-1}^{(k)}}{\alpha_{i-1,i}^2 \hat{\pi}_{i-1}^{(k)} + \hat{\pi}_i^{(k)}} \delta_{i-1}^{(k)}$$

We note here that we can let the update of the precision happen first, and therefore use it for the update of the mean:

$$\pi_i^{(k)} = \hat{\pi}_i^{(k)} + \alpha_{i-1,i}^2 \hat{\pi}_{i-1}^{(k)} \tag{1}$$
$$\mu_i^{(k)} = \hat{\mu}_i^{(k)} + \frac{\alpha_{i-1,i}^2 \hat{\pi}_{i-1}^{(k)}}{\pi_i^{(k)}} \delta_{i-1}^{(k)} \tag{2}$$

In sum, at the time of the update, Node i needs to have access to the following quantities:

Its own predictions: $\hat{\mu}_i^{(k)}$, $\hat{\pi}_i^{(k)}$

Coupling strength: $\alpha_{i-1,i}$

From level below: $\delta_{i-1}^{(k)}$, $\hat{\pi}_{i-1}^{(k)}$

All of these are available at the time of the update. Node i therefore only needs to receive the PE and the predicted precision from the level below to perform its update.

2.2 Prediction Error Step

We will assume in the following, that Node i is the value child of Node $i+1$. Then the following quantities have to be sent up to Node $i+1$ (cf. necessary information from level below in a value parent):

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

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

Node i has already performed the PREDICTION step on the previous trial, so it has already computed the predicted precision of the current trial, $\hat{\pi}_i^{(k)}$. Hence, in the PE step, it needs to perform only the following calculation:

$$\delta_i^{(k)} = \mu_i^{(k)} - \hat{\mu}_i^{(k)} \tag{3}$$

2.3 Prediction Step

Still assuming that Node i is the value child of Node $i+1$, the PREDICTION step consists of the following computations:

$$\hat{\mu}_i^{(k+1)} = \mu_i^{(k)} + \alpha_{i,i+1}\mu_{i+1}^{(k)} \tag{4}$$

$$\hat{\pi}_i^{(k+1)} = \frac{1}{\frac{1}{\pi_i^{(k)}} + \nu_i^{(k+1)}} \tag{5}$$

with

$$\nu_i^{(k+1)} = \exp(\omega_i). \tag{6}$$

Note that if Node i additionally has a VOPE parent node, the estimated volatility $\nu_i^{(k+1)}$ that enters the precision update would also depend on the posterior mean of that volatility parent (cf. PREDICTION step for VOPE coupling).

In general, the prediction of the mean will depend only on whether Node i has a value parent or not, whereas the prediction of the precision only depends on whether Node i has a volatility parent or not.

Thus, the PREDICTION step only depends on knowing the node's own posteriors and receiving the value parent's posterior in time before the new input arrives.

3 Computations for VOPE coupling

As in the case of VAPE coupling, the exact computations of the UPDATE step depend on the nature of the coupling with the child node(s), while both the PE step and the PREDICTION step depend on the coupling with the parent node(s).

To describe the computations entailed by VOPE coupling, we will introduce two changes to the notation. First of all, 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 character δ_i only for VAPEs:

$$\delta_i^{(k)} \equiv \delta_i^{(k,VAPE)} = \mu_i^{(k)} - \hat{\mu}_i^{(k)}, \tag{7}$$

and introduce a new character Δ_i for VOPEs, which we define as

$$\begin{aligned}
\Delta_i^{(k)} \equiv \delta_i^{(k,VOPE)} &= \frac{\frac{1}{\pi_i^{(k)}} + (\mu_i^{(k)} - \hat{\mu}_i^{(k)})^2}{\frac{1}{\pi_i^{(k-1)}} + \nu_i^{(k)}} - 1 \\
&= \hat{\pi}_i^{(k)} \left(\frac{1}{\pi_i^{(k)}} + (\mu_i^{(k)} - \hat{\mu}_i^{(k)})^2 \right) - 1 \\
&= \hat{\pi}_i^{(k)} \left(\frac{1}{\pi_i^{(k)}} + (\delta_i^{(k)})^2 \right) - 1 \\
&= \frac{\hat{\pi}_i^{(k)}}{\pi_i^{(k)}} + \hat{\pi}_i^{(k)} (\delta_i^{(k)})^2 - 1.
\end{aligned} \tag{8}$$

Note that from the first to the second line, we have used the following definition:

$$\hat{\pi}_{i-1}^{(k)} = \frac{1}{\frac{1}{\pi_{i-1}^{(k-1)}} + \nu_{i-1}^{(k)}}.$$

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

That means we are introducing a second prediction error unit Δ_i which is concerned with deviations from predicted uncertainty and is informed by value prediction errors and other estimates of uncertainty. It is this prediction error - a function of the unweighted (squared) value prediction error with a new precision weight - which communicates between a level's nodes and a level's volatility parent's nodes.

Second, we will introduce another quantity, which we term the (auxiliary) expected precision

$$\gamma_i^{(k)} = \nu_i^{(k)} \hat{\pi}_i^{(k)}, \tag{9}$$

which will be computed as part of the PREDICTION step and only serves to simplify the equations and the corresponding message passing.

3.1 Update Step

If Node i is the volatility parent of Node $i-1$, then the following update equations apply to Node i:

$$\begin{aligned}
\pi_i^{(k)} &= \hat{\pi}_i^{(k)} + \frac{1}{2}(\kappa_{i-1}\nu_{i-1}^{(k)}\hat{\pi}_{i-1}^{(k)})^2 * (1 + (1 - \frac{1}{\pi_{i-1}^{(k-1)}\nu_{i-1}^{(k)}})\delta_{i-1}^{(k)}) \\
&= \hat{\pi}_i^{(k)} + \frac{1}{2}(\kappa_{i-1}\nu_{i-1}^{(k)}\hat{\pi}_{i-1}^{(k)})^2 * (1 + (2 - \frac{1}{\hat{\pi}_{i-1}^{(k)}\nu_{i-1}^{(k)}})\delta_{i-1}^{(k)}) \\
\mu_i^{(k)} &= \hat{\mu}_i^{(k)} + \frac{1}{2}\kappa_{i-1}\nu_{i-1}^{(k)}\frac{\hat{\pi}_{i-1}^{(k)}}{\pi_i^{(k)}}\delta_{i-1}^{(k)},
\end{aligned}$$

where we have again used the definition of the predicted precision $\hat{\pi}_{i-1}^{(k)}$ to derive an expression for the posterior precision from the previous trial $\pi_{i-1}^{(k-1)}$:

$$\hat{\pi}_{i-1}^{(k)} = \frac{1}{\frac{1}{\pi_{i-1}^{(k-1)}} + \nu_{i-1}^{(k)}}$$
$$\Leftrightarrow \pi_{i-1}^{(k-1)} = \frac{1}{\frac{1}{\hat{\pi}_{i-1}^{(k)}} - \nu_{i-1}^{(k)}}.$$

With the changes from above, namely the definitions of the VOPE Δ_i and the expected precision $\gamma_i^{(k)}$, the update equations for the precision and the mean in volatility coupling simplify to:

$$\pi_i^{(k)} = \hat{\pi}_i^{(k)} + \frac{1}{2}(\kappa_{i,i-1}\gamma_{i-1}^{(k)})^2 + (\kappa_{i,i-1}\gamma_{i-1}^{(k)})^2\Delta_{i-1}^{(k)} - \frac{1}{2}\kappa_{i,i-1}\gamma_{i-1}^{(k)}\Delta_{i-1}^{(k)} \quad (10)$$

$$\mu_i^{(k)} = \hat{\mu}_i^{(k)} + \frac{1}{2}\frac{\kappa_{i,i-1}\gamma_{i-1}^{(k)}}{\pi_i^{(k)}}\Delta_{i-1}^{(k)} \quad (11)$$

Therefore, at the time of the update, Node i needs to have access to the following quantities:

Its own predictions: $\hat{\mu}_i^{(k)}$, $\hat{\pi}_i^{(k)}$

Coupling strength: $\kappa_{i,i-1}$

From level below: $\Delta_{i-1}^{(k)}$, $\hat{\pi}_{i-1}^{(k)}$, $\nu_{i-1}^{(k)}$, $\gamma_{i-1}^{(k)}$

3.2 Prediction Error Step

The exact computation of the prediction error depends, like the computation of the new prediction, on the nature of the coupling with the parent nodes. We will therefore assume in the following, that Node i is the volatility child of Node $i+1$. Then the following quantities have to be sent up to Node $i+1$ (see also necessary information from level below in a volatility parent):

Volatility estimate: $\nu_i^{(k)}$

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

Expected precision: $\gamma_i^{(k)}$

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

Node i has already performed the PREDICTION step on the previous trial, so it has already computed the predicted precision, $\hat{\pi}_i^{(k)}$, the expected precision, $\gamma_i^{(k)}$, and the volatiliy estimate, $\nu_i^{(k)}$, for the current trial. Hence, in the PE step, it needs to perform only the following calculations:

$$\delta_i^{(k)} = \mu_i^{(k)} - \hat{\mu}_i^{(k)} \tag{12}$$

$$\Delta_i^{(k)} = \frac{\hat{\pi}_i^{(k)}}{\pi_i^{(k)}} + \hat{\pi}_i^{(k)}(\delta_i^{(k)})^2 - 1. \tag{13}$$

3.3 Prediction Step

Still assuming that Node i is the volatility child of Node $i+1$, the PREDICTION step consists of the following simple computations:

$$\hat{\mu}_i^{(k+1)} = \mu_i^{(k)} \tag{14}$$

$$\nu_i^{(k+1)} = \exp(\kappa_i \mu_{i+1}^{(k)} + \omega_i) \tag{15}$$

$$\hat{\pi}_i^{(k+1)} = \frac{1}{\frac{1}{\pi_i^{(k)}} + \nu_i^{(k+1)}} \tag{16}$$

$$\gamma_i^{(k+1)} = \nu_i^{(k+1)} \hat{\pi}_i^{(k+1)} \tag{17}$$

Thus, the prediction for trial $k+1$ depends again only on receiving the posterior mean of Node $i+1$ on trial k, and knowing the Node's own posteriors.

Note that if Node i additionally has a VAPE parent node, the prediction of the new mean, $\hat{\mu}_i^{k+1}$ would also depend on the posterior mean of that value parent (cf. PREDICTION step for VAPE coupling).

4 Implementation HGF coupling

If we assume that a cortical column consists of units (i.e., sub-populations of neurons), one for each computational step, then the computations for VAPE coupling as well as for VOPE coupling could be implemented as sketched in Figure 1.

A few points to note here:

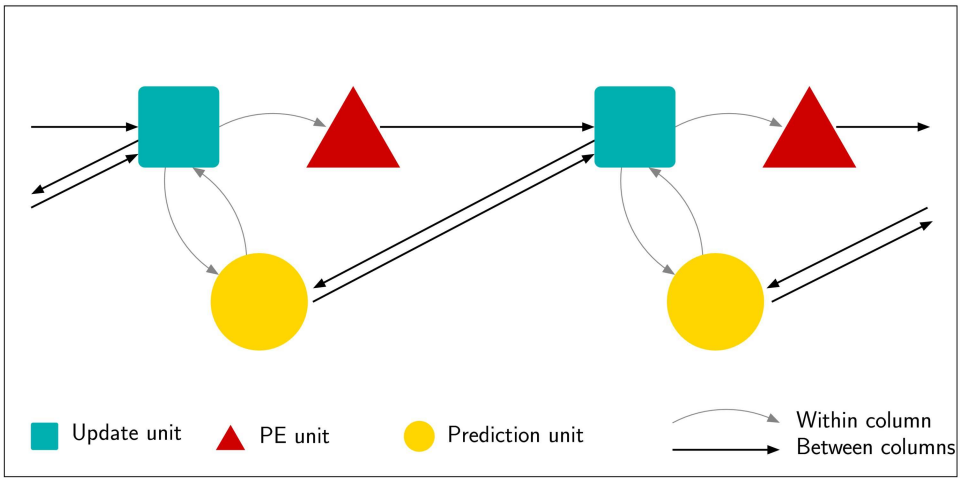

Figure 1: Example implementation of the coupling between cortical columns in the HGF. Figure shows two cortical columns that could be coupled to each other via VAPE or VOPE coupling. In each case, the messages passed along the connections and the computations within the nodes would differ according to the equations presented in the previous chapters.

- In this implementation, we assume that the estimated volatility, ν_i^k, is computed within the Prediction unit. This is motivated by the equations presented above and the fact that this unit receives information about the updated quantities (posterior means and precisions) from the level above.

- Similarly, the only unit that needs to know the value of a node's tonic learning rate, ω_i, is the Prediction unit. Therefore, this parameter could be implemented in terms of self-connections or interneurons within that neuronal population.

- A noteworthy arrow in this picture is the one from the Prediction unit of one node to the Update unit of its parent node. This seems to violate the classical assumptions that while top-down connections signal predictions, bottom-up connections would only signal prediction errors. Due to the update equations in the HGF, however, any given parent node needs access to the predicted precision from its child node to perform an update. To avoid this arrow, we could also assume that the Prediction unit sends the predicted precision to the PE unit of the same node, which then sends this quantity up to the Update unit of the parent node. However, this seems more complicated for two reasons: First, the PE unit doesn't actually need this quantity to perform its computation. Secondly, the PE is computed only somewhat after the prediction of the current trial (which is already computed after the

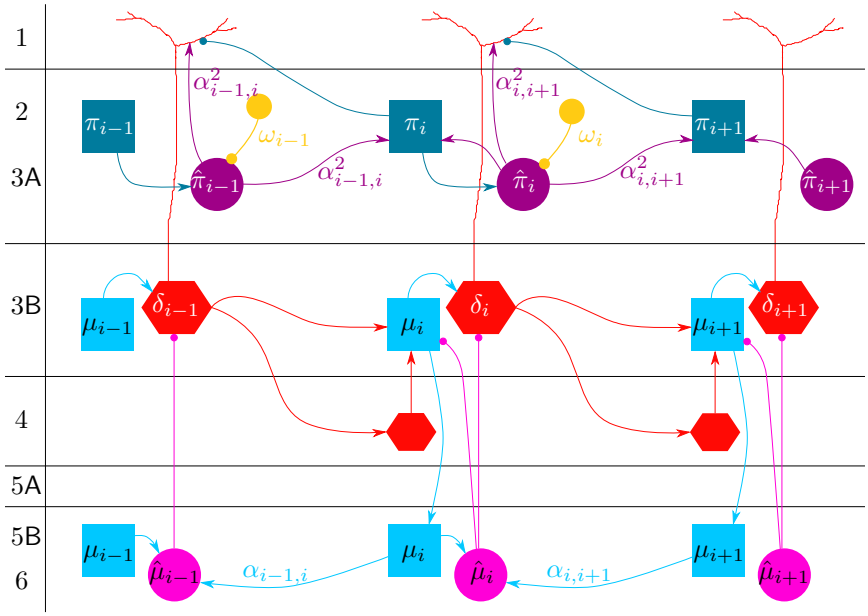

Figure 2: Overview of possible layer-specific message passing in VAPE coupling. Assignment of nodes to layers is loosely based on Figure 3 in [2]

.

previous trial). Thus, the two signals would be sent up at different times.

- Note that we did not distinguish between the computational steps Prediction and Update for the mean versus for the precision here. It is likely however, that these would be encoded in separate neuronal populations. Also, the implementation of VAPE and VOPE parents might differ. We will address these specifications below in comparison to classical predictive coding schemes.

5 Relation to Predictive Coding

In this section, we consider an implementation of the message passing implicated by the HGF which is as close as possible to current proposals of neural architectures for predictive coding [2].

We will separately consider VAPE and VOPE coupling and realize that the message passing for VAPE coupling is almost equivalent to the proposed PC architecture, while the VOPE coupling comes with more challenges, simply because it hasn't been addressed in detail in existing PC proposals on neural architecture.

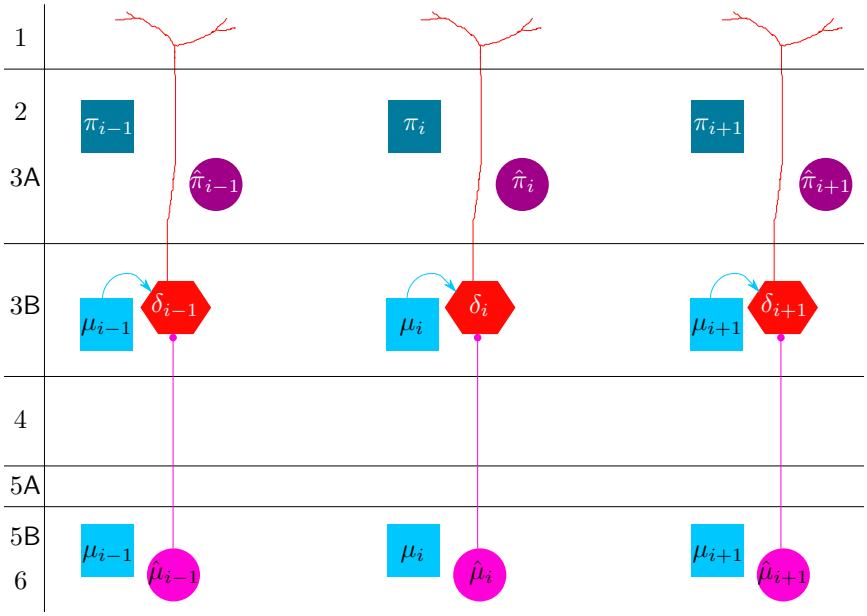

Figure 3: Implementation of VAPE PE computation on three cortical levels: $\delta_i = \mu_i - \hat{\mu}_i$.

5.1 VAPE coupling

Figure 2 sketches a possible architecture implementing all computations involved in VAPE coupling for three example cortical regions (levels). The assignment of neural elements to cortical layers follows the proposal in in figure 3 of [2]. For example, we've placed all precision-related nodes into the upper layers, while expectations and predictions of the mean live in intermediate and deep layers, respectively. In the following, we go through the different computational steps and note differences to the proposal for predictive coding.

For the PE step, we have the following computation:

$$\delta_i^{(k)} = \mu_i^{(k)} - \hat{\mu}_i^{(k)}. \tag{18}$$

The message passing implicated is depicted in figure 3 and is in line with PE computations as suggested by [2].

In the Update step, the posterior estimates of mean and variance are given by:

$$\mu_i^{(k)} = \hat{\mu}_i^{(k)} + \frac{\alpha_{i-1,i}^2 \hat{\pi}_{i-1}^{(k)}}{\pi_i^{(k)}} \delta_{i-1}^{(k)} \tag{19}$$

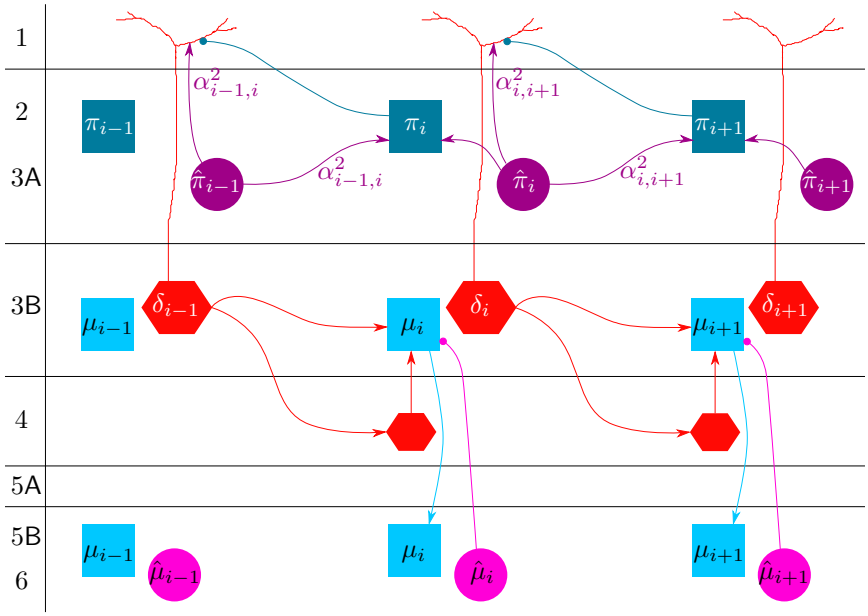

Figure 4: Message passing entailed by the Update step in VAPE coupling.

and

$$\pi_i^{(k)} = \hat{\pi}_i^{(k)} + \alpha^2_{i-1,i}\hat{\pi}_{i-1}^{(k)}. \tag{20}$$

A possible message passing implementation for these updates is shown in figure 4. There are two important differences to [2] to note here.

First, in PC proposal, the update of the mean μ_i is driven both by the lower-level PE δ_{i-1} and the PE on the level itself, δ_i, which mediates the influence of the empirical prior, or the prediction, on the mean.

In the HGF equations, we do not have this negative feedback loop from δ_i to μ_i. Instead, the prediction $\hat{\mu}_i$ has a direct influence on μ_i (see equations above), hence the arrow from the $\hat{\mu}_i$ node to the μ_i node in figure 4.

However, by considering the within-trial temporal evolution of the μ_i node, we can get the same feedback loop. Noting that μ_i as computed in the above equation will be the endpoint of the update, i.e., the equilibrium value to which the node should stabilize at the end of trial k, we can, in the style of [1], propose the following temporal evolution:

$$\dot{\mu}_i^{(k)} = \hat{\mu}_i^{(k)} + \frac{\alpha^2_{i-1,i}\hat{\pi}_{i-1}^{(k)}}{\pi_i^{(k)}}\delta_{i-1}^{(k)} - \mu_i^{(k)} \tag{21}$$

Clearly, a node with this dynamic converges to the required posterior

value, which can be seen by setting $\dot{\mu}_i$ to zero. Now we note that

$$\dot{\mu}_i^{(k)} = \frac{\alpha_{i-1,i}^2 \hat{\pi}_{i-1}^{(k)}}{\pi_i^{(k)}} \delta_{i-1}^{(k)} - \mu_i^{(k)} + \hat{\mu}_i^{(k)} \tag{22}$$

can be summarized as

$$\dot{\mu}_i^{(k)} = \frac{\alpha_{i-1,i}^2 \hat{\pi}_{i-1}^{(k)}}{\pi_i^{(k)}} \delta_{i-1}^{(k)} - \delta_i^{(k)}. \tag{23}$$

Following this logic, we could also place the arrow between δ_i and μ_i. The exercise of writing down the within-trial temporal dynamics of the nodes will be repeated below for all nodes, which results in equations for within-trial network dynamics. These can be used to simulate within-trial responses of all nodes, examining network stability, and predicting neural dynamics which could for example be compared with electrophysiological data.

The second difference to the PC proposal is rather irreducible. As already mentioned in the previous section, the messages sent up to parent nodes do not only consist of prediction errors or their precision weights. Instead, to update the precision node π_i, node i also needs access to the prediction precision $\hat{\pi}_{i-1}$ from the level below. This is reflected in the violet arrow travelling from superficial layers of level i to level $i+1$.

Finally, in the Prediction step, we have:

$$\hat{\mu}_i^{(k+1)} = \mu_i^{(k)} + \alpha_{i,i+1} \mu_{i+1}^{(k)} \tag{24}$$

and

$$\hat{\pi}_i^{(k+1)} = \frac{1}{\frac{1}{\pi_i^{(k)}} + \exp(\omega_i)}. \tag{25}$$

This again seems unproblematic, although the arrow from μ_i to $\hat{\mu}_i$ would not appear in the classical PC architecture. Also, we've placed the predicted precision $\hat{\pi}$ in the superficial layers here, for convenience, as it interacts only with the posterior precision π and acts as a weight on the prediction error, and also inaccordance with [2], where precision signals arise from layers 2 and 3A.

5.2 VOPE coupling

For volatility parents, we propose the following: Each level of a cortical hierarchy implements its own volatility parent in superficial layers. Coupling across layers is exclusively value-related. This means that having a volatility parent is nothing more but a more sophisticated way of predicting and updating a level's precision estimates.

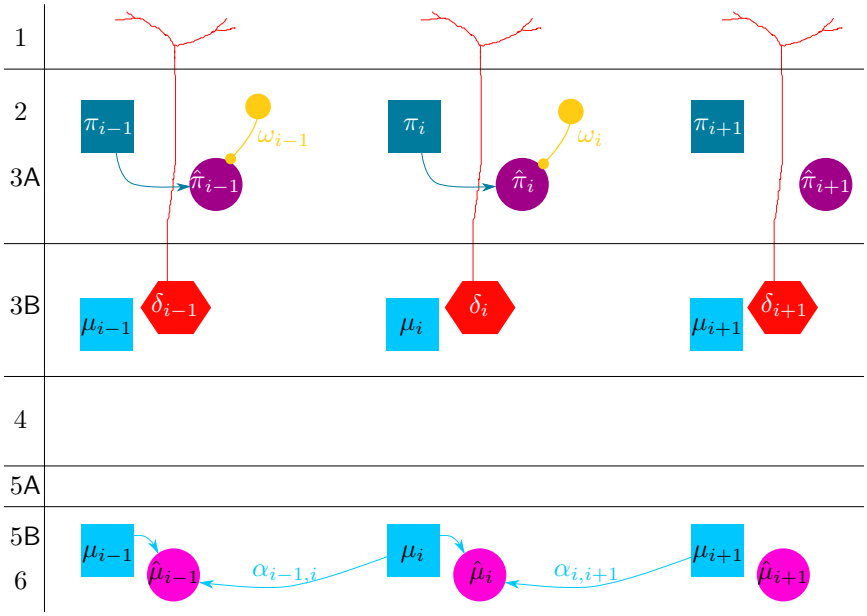

Figure 5: Message passing entailed by the Prediction step in VAPE coupling.

This setup excludes volatility parents of volatility parents. However, it allows for a cortical hierarchy where higher levels predict either the mean, or the volatility, of lower levels, both via value coupling. In other words, any given level receives top-down predictions not only about its mean, but also about its volatility, which is implemented as the mean of its volatility parent.

Note that this setup also allows for a centralized or global volatility estimation. For example, a cortical or subcortical region which monitors the overall environmental volatility could be coupled to all local volatility units via value coupling. Our proposal is more flexible than previous ones [**?**] though, in that it also enables differential volatility estimation, e.g. within different sensory channels.

Moreover, in principle, both the mean μ_i as well as the volatility estimate $\check{\mu}_i$ of a given level could have (one or more) value parents in higher cortical areas. This allows for both separate and combined higher-level prediction of both the precision and the mean of level i.

Due to our proposal, we slightly change the notation for volatility coupling from here on. Level i from now on refers to hierarchical level within a cortical hierarchy. Nodes on this level comprise the 'original' nodes μ_i, π_i, $\hat{\mu}_i$, $\hat{\pi}_i$, and δ_i, which are concerned with value estimation, but also all the nodes belonging to a volatility parent of these nodes, which we now denote

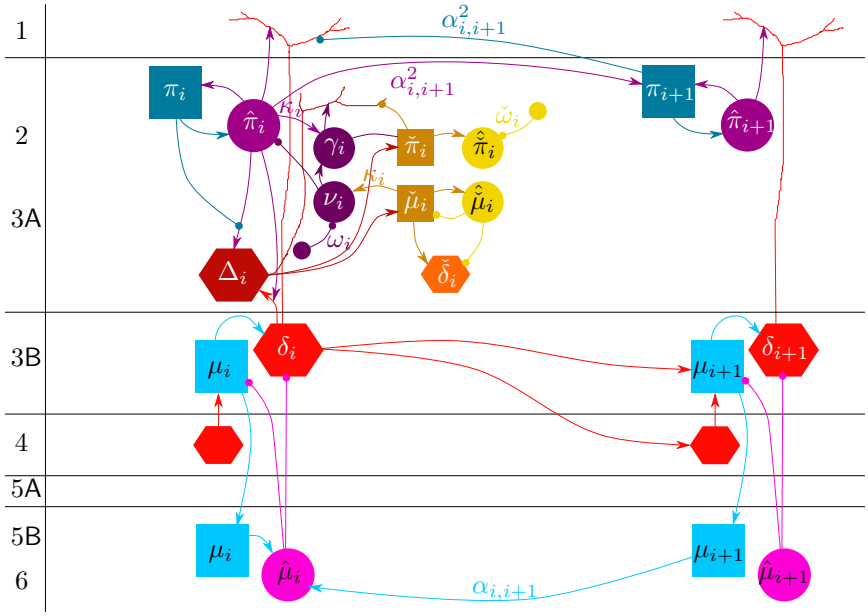

Figure 6: Overview of possible layer-specific message passing in VOPE coupling. Assignment of nodes to layers is loosely based on Figure 3 in [2]

.

as $\check{\mu}_i$, $\check{\pi}_i$, $\hat{\check{\mu}}_i$, $\hat{\check{\pi}}_i$, and $\check{\delta}_i$. Here, $\check{\delta}_i$ refers to the value prediction error about the mean of the volatility, $\check{\mu}_i$.

The VOPE Δ_i, which we have written as a function of the VAPE δ_i, of level i, is what communicates between a level's nodes and a level's volatility parent's nodes. Value estimation and volatility estimation both happen within on cortical level or column. For the ensuing equations (simple reformulations of the computations in VOPE coupling as presented before) and visualizations of this proposal, please refer to the appendix.

Discuss similiraties with and differences to [**?**] here!

From here on until end of chapter: Appendix

Figure 6 displays one proposal for this setup, where we've zoomed in to a level i of the cortical hierarchy (and a value parent $i+1$) and added its volatility parent to the superficial layers. An alternative implementation of this idea is displayed in figure 7. Here, only the precision-related computations of the volatility parent are placed in the superficial layers, while the corresponding prediction errors, Δ_i and $\check{\delta}_i$ as well as the actual volatility estimate $\check{\mu}_i$ live in middle layers, and the prediction of the volatility estimate, $\hat{\check{\mu}}_i$, lives in layer 6. This setup stresses the structural similarities to the message passing entailed by mean coupling. Moreover, using this setup,

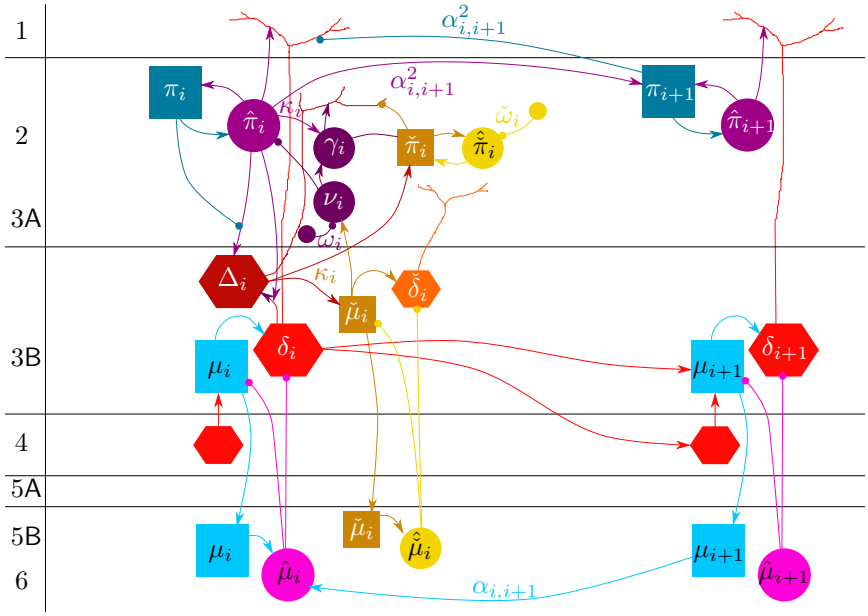

Figure 7: Zoomed-in version of the coupling between a hierarchical level i with its parent $i+1$ in the case of VAPE coupling, with an additional VOPE parent *within* node i.

.

we can now depict how a higher cortical level $i+1$ would serve as a value parent to the volatility estimate in level i - instead of predicting the mean - by simply exchanging the arrows between the levels as shown in figure 8.

The Update steps in volatility coupling are:

$$\check{\mu}_i^{(k)} = \hat{\check{\mu}}_i^{(k)} + \frac{1}{2}\frac{\gamma_i^{(k)}}{\check{\pi}_i^{(k)}}\Delta_i^{(k)} \tag{26}$$

$$\check{\pi}_i^{(k)} = \hat{\check{\pi}}_i^{(k)} + \frac{1}{2}(\gamma_i^{(k)})^2 + (\gamma_i^{(k)})^2\Delta_i^{(k)} - \frac{1}{2}\gamma_i^{(k)}\Delta_i^{(k)} \tag{27}$$

In the PE step, the node computes:

$$\Delta_i^{(k)} = \frac{\hat{\pi}_i^{(k)}}{\pi_i^{(k)}} + \hat{\pi}_i^{(k)}(\delta_i^{(k)})^2 - 1. \tag{28}$$

Finally, In the Prediction step, we now need to compute four nodes: the predicted (volatility) mean

$$\hat{\check{\mu}}_i^{(k+1)} = \check{\mu}_i^{(k)}, \tag{29}$$

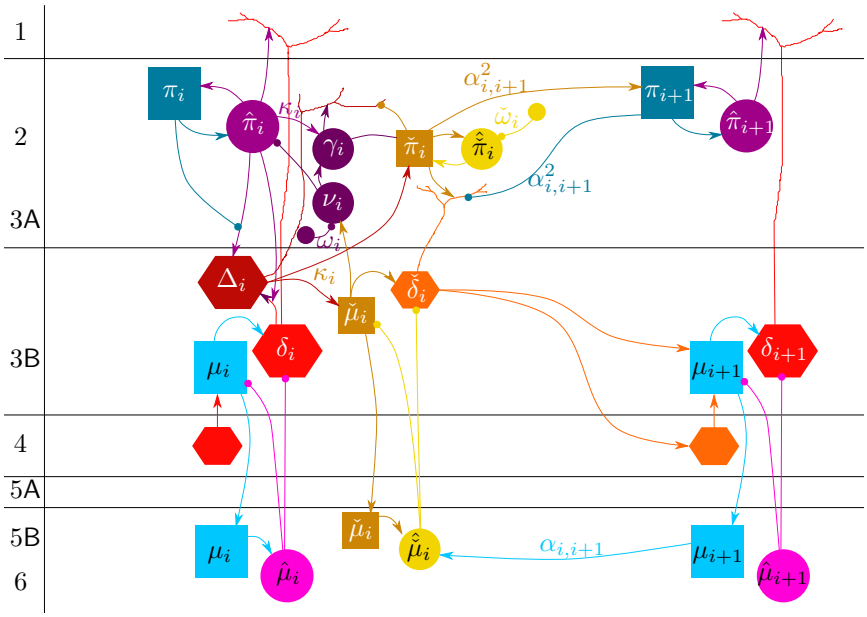

Figure 8: Zoomed-in version of the coupling between a hierarchical level i with its parent $i+1$ in the case of VOPE coupling. Note that the coupling is of VOPE type within level i, to update $\check{\mu}_i$, which is turn predicted by level $i+1$, where the higher level now serves as a VAPE parent to the volatility estimate $\check{\mu}_i$.

the precision of that prediction

$$\hat{\hat{\pi}}_i^{(k+1)} = \frac{1}{\frac{1}{\hat{\pi}_i^{(k)}} + \nu_i^{(k+1)}}, \tag{30}$$

the predicted environmental uncertainty (as a function of the next higher level in the hierarchy, μ_{i+1})

$$\nu_i^{(k+1)} = \exp(\kappa_{i,i+1}\mu_{i+1}^{(k)} + \omega_i), \tag{31}$$

and the new (auxiliary) expected precision

$$\gamma_i^{(k+1)} = \kappa_{i+1,i}\nu_i^{(k+1)}\hat{\pi}_i^{(k+1)}. \tag{32}$$

The last node is only defined for convenience in terms of simplifying the equations and the corresponding message passing.

6 Within-trial time dynamics

In this section, we derive differential equations which lay out the within-trial update dynamics entailed by the HGF. To this end, we consider the posterior values of all our nodes (quantities) as given by the HGF update equations as the equilibrium point towards which all dynamics must converge.

We will note that these equations imply a slightly different coupling between nodes compared to the previous section. On first glance, this complicates the picture, as we're introducing more and more nodes. On the other hand, we increase biological plausibility by making the computations of a single node or connection simpler and simpler.

We divide the equations into value estimation and volatility estimation, noting that VAPE and VOPE coupling are not useful divisions anymore (the mean of the value estimation of a level i, μ_i, will be VAPE-coupled to its parent μ_{i+1}, while the precision π_i of the value estimation will be VOPE-coupled to the volatility parent $\check{\mu}_i$ on level i itself; on the other hand, the mean of the volatility estimation, $\check{\mu}_i$, can again have a VAPE coupling to a higher level, e.g. μ_{i+2}, and the precision of volatility, $\check{\pi}_i$, will, in the current framework, not be coupled to any parent, i.e., its prediction will depend only on the parameter $\check{\omega}_i$).

6.1 Value estimation

For the PE about the mean, we want to reach the following posterior:

$$\delta_i^{(k)} = \mu_i^{(k)} - \hat{\mu}_i^{(k)}. \tag{33}$$

This can be easily achieved by a node with these dynamics:

$$\dot{\delta}_i^{(k)} = \mu_i^{(k)} - \hat{\mu}_i^{(k)} - \delta_i^{(k)}. \tag{34}$$

From this it follows that we require an additional inhibitory self-connection for the PE node.

We now consider an alternative implementation for the PE computation, which will turn out to be very useful.

Instead of considering the unweighted δ_i, we can also directly model a node which corresponds to the precision-weighted prediction error ε_i:

$$\varepsilon_i = \frac{\alpha^2 \hat{\pi}_i}{\pi_{i+1}}(\mu_i - \hat{\mu}_i) = \frac{\alpha^2 \hat{\pi}_i}{\pi_{i+1}}\delta_i. \tag{35}$$

This node could evolve according to

$$\dot{\varepsilon}_i = \mu_i - \hat{\mu}_i - \frac{\pi_{i+1}}{\alpha^2 \hat{\pi}_i}\varepsilon_i, \tag{36}$$

which allows us to re-introduce the unweighted PE δ_i:

$$\begin{aligned} \delta_i &= \frac{\pi_{i+1}}{\alpha^2 \hat{\pi}_i}\varepsilon_i \\ &= \frac{\pi_{i+1}(\mu_i - \hat{\mu}_i)\alpha^2 \hat{\pi}_i}{\alpha^2 \hat{\pi}_i \pi_{i+1}} \\ &= \mu_i - \hat{\mu}_i, \end{aligned} \tag{37}$$

such that

$$\dot{\varepsilon}_i = \mu_i - \hat{\mu}_i - \delta_i \tag{38}$$

and

$$\dot{\delta}_i = \varepsilon_i - \alpha^2 \frac{\hat{\pi}_i}{\pi_{i+1}}\delta_i. \tag{39}$$

This has the advantage that the weighted PE ε_i can travel up the hierarchy to update higher levels, while the unweighted PE δ_i can be used to feedback to the mean on the same level, an element that will be introduced next.

In the Update step, the posterior estimate of the mean is given by:

$$\mu_i^{(k)} = \hat{\mu}_i^{(k)} + \frac{\alpha_{i-1,i}^2 \hat{\pi}_{i-1}^{(k)}}{\pi_i^{(k)}}\delta_{i-1}^{(k)}. \tag{40}$$

Here, we propose the following temporal evolution:

$$\dot{\mu}_i^{(k)} = \hat{\mu}_i^{(k)} + \frac{\alpha_{i-1,i}^2 \hat{\pi}_{i-1}^{(k)}}{\pi_i^{(k)}} \delta_{i-1}^{(k)} - \mu_i^{(k)} \tag{41}$$

Clearly, a node with this dynamic converges to the required posterior value, which can be seen by setting $\dot{\mu}_i$ to zero. Now we note that

$$\dot{\mu}_i^{(k)} = \frac{\alpha_{i-1,i}^2 \hat{\pi}_{i-1}^{(k)}}{\pi_i^{(k)}} \delta_{i-1}^{(k)} - \mu_i^{(k)} + \hat{\mu}_i^{(k)} \tag{42}$$

can be summarized as

$$\dot{\mu}_i^{(k)} = \varepsilon_{i-1}^{(k)} - \delta_i^{(k)}. \tag{43}$$

We will therefore consider a connection between δ_i and μ_i, instead of connecting the prediction node $\hat{\mu}_i$ with the mean μ_i. As already outlined in the previous section, this also resolves one of the apparent differences between our message passing scheme and classical PC proposals.

For the posterior precision, we have

$$\pi_i^{(k)} = \hat{\pi}_i^{(k)} + \alpha_{i-1,i}^2 \hat{\pi}_{i-1}^{(k)}. \tag{44}$$

Here, we stick with the same approach as for the PE node, and simply add a self-inhibitory connection to implement the dynamics:

$$\dot{\pi}_i^{(k)} = \hat{\pi}_i^{(k)} + \alpha_{i-1,i}^2 \hat{\pi}_{i-1}^{(k)} - \pi_i^{(k)}. \tag{45}$$

Finally, in the Prediction step, we want to reach the following prediction of the mean:

$$\hat{\mu}_i^{(k+1)} = \mu_i^{(k)} + \alpha_{i,i+1} \mu_{i+1}^{(k)}. \tag{46}$$

Rewriting this as a differential equation

$$\dot{\hat{\mu}}_i^{(k+1)} = \alpha_{i,i+1} \mu_{i+1}^{(k)} + \mu_i^{(k)} - \hat{\mu}_i^{(k+1)}, \tag{47}$$

we again see the PE node appear:

$$\dot{\hat{\mu}}_i^{(k+1)} = \alpha_{i,i+1} \mu_{i+1}^{(k)} + \delta_i^{(k)}. \tag{48}$$

Note that $\delta_i^{(k)}$ is actually defined in terms of $\hat{\mu}_i^{(k)}$, and not $\hat{\mu}_i^{(k+1)}$, but given that we are in continuous time now, we suspect that this is probably equivalent here or can be resolved by introducing a delay between the computation of the prediction and the prediction error. For simplicity, we will

skip the trial indices from now on.

The precision of this prediction should converge to:

$$\hat{\pi}_i = \frac{1}{\frac{1}{\pi_i} + \exp(\omega_i)}. \tag{49}$$

We can write the dynamics as:

$$\begin{aligned} \dot{\hat{\pi}}_i &= 1 - (\frac{1}{\pi_i} + \exp(\omega_i))\hat{\pi}_i \\ &= 1 - \frac{\hat{\pi}_i}{\pi_i} - \exp(\omega_i)\hat{\pi}_i. \end{aligned} \tag{50}$$

To simplify the implementation, we introduce an additional node p_{i1}:

$$p_{i1} = \frac{\hat{\pi}_i}{\pi_i}. \tag{51}$$

This new node can have the following dynamcis:

$$\dot{p}_{i1} = \hat{\pi}_i - \pi_i p_{i1}, \tag{52}$$

and the dynamics for the precision of the prediction $\hat{\pi}_i$ become

$$\dot{\hat{\pi}}_i = 1 - p_{i1} - \exp(\omega_i)\hat{\pi}_i, \tag{53}$$

where the last term can be implemented as a self-inhibition with a connection weight of $\exp(\omega_i)$. However, to facilitate the implementation of the influence of a volatility parent on the precision estimate, we introduce another additional node, p_{i2}, here and also already consider a node that estimates volatility, ν_i. In the case of value estimation without a volatility node, this estimate will only depend on the parameter value ω_i, i.e., the tonic volatility.

The dynamics of $\hat{\pi}_i$ then become:

$$\dot{\hat{\pi}}_i = 1 - p_{i1} - p_{i2}, \tag{54}$$

with the node p_{i2} defined as

$$p_{i2} = \hat{\pi}_i \exp(\omega_i), \tag{55}$$

with dynamics described by

$$\dot{p}_{i2} = \exp(\omega_i)\hat{\pi}_i - p_{i2} \tag{56}$$

in the absence of a volatility parent (see below for the extension to volatility estimation). We introduce the volatility estimate ν_i as

$$\nu_i = \exp(\omega_i), \tag{57}$$

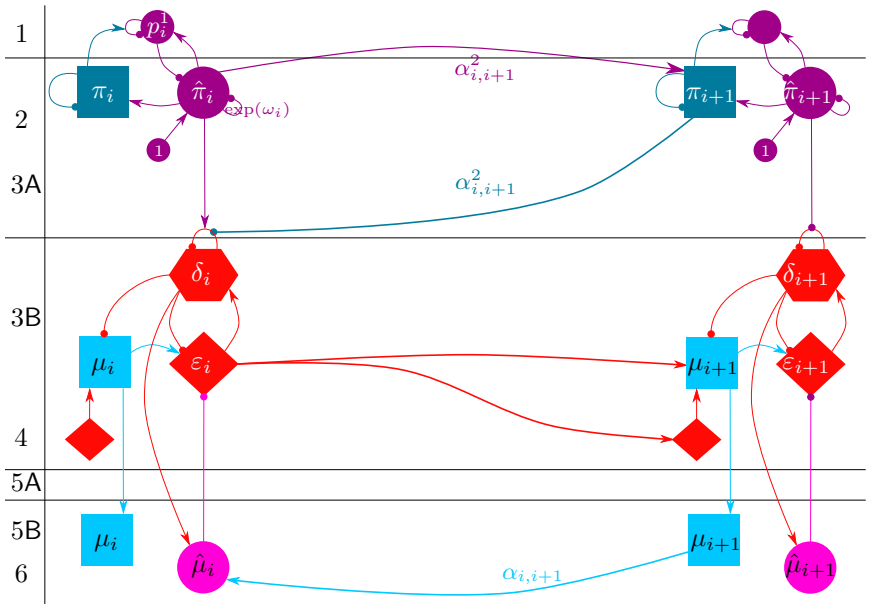

Figure 9: The coupling between a hierarchical level i with its parent $i+1$ in the case of VAPE coupling: implementing the differential equations that describe within-trial dynamics of the nodes.

evolving as

$$\dot{\nu}_i = \exp(\omega_i) - \nu_i, \tag{58}$$

leading to

$$\dot{p}_{i2} = \nu_i \hat{\pi}_i - p_{i2}. \tag{59}$$

A visualization of the network entailed by the differential equations so far - in the absence of volatility estimation - is depicted in figure 9.

Note that the dynamics of ν_i can be implemented by a self-inhibition and a positive influence from a tonicly active neuron (with activity 1) with a connection weight of $\exp(\omega_i)$. The advantage of this description is that if we add a volatility parent, we simply have to let this tonicly active neuron be driven by the volatility estimate $\check{\mu}_i$, weighted by the coupling parameter κ_i, to arrive at

$$\nu_i = \exp(\kappa_i \check{\mu}_i) \exp(\omega_i). \tag{60}$$

Here we note that according to our proposal above, any cortical level i will implement its own volatility estimation, along with the value estimation. Therefore, the precision of the prediction, $\hat{\pi}_i$, will always depend on the value

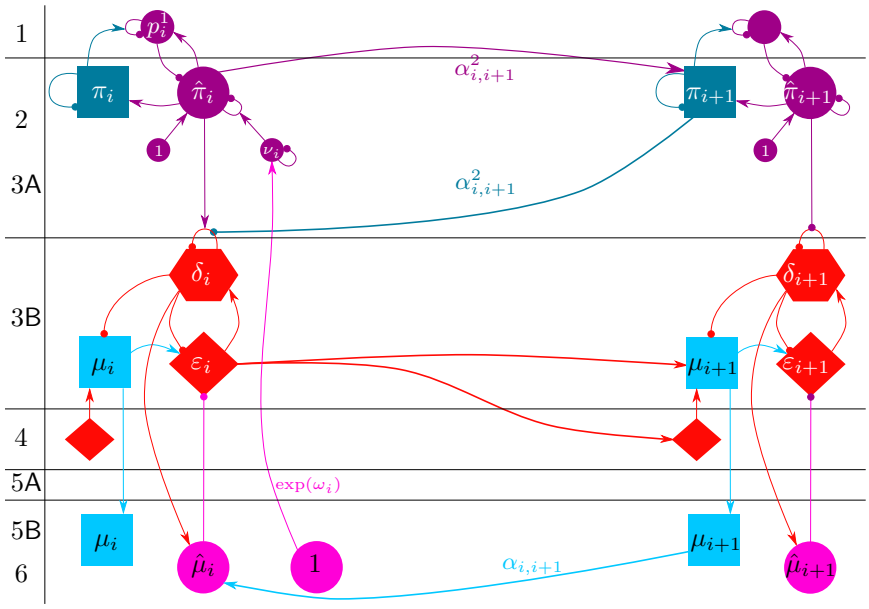

Figure 10: The coupling between a hierarchical level i with its parent $i+1$ in the case of VAPE coupling, using within-trial dynamics: introducing ν as the (tonic) volatility estimation.

of the volatility parent $\check{\mu}_i$:

$$\hat{\pi}_i = \frac{1}{\frac{1}{\pi_i} + \exp(\kappa_i \check{\mu}_i + \omega_i)}. \tag{61}$$

This variant of the VAPE implementation, which is representing volatility estimation explicitely even in the absence of a VOPE parent, is depicted in figure 10.

We now turn to the computations entailed by phasic volatility estimation, including the modified dynamics for $\hat{\pi}_i$ and ν_i.

6.2 Volatility estimation

Here we start by considering the changes in the dynamics of the precision prediction in node i, i.e. the dynamics of $\hat{\pi}_i$ and ν_i, which are the consequence of adding an estimate of phasic volatility $\check{\mu}_i$.

We simply have to add a (multiplicative) term in the dynamics of the predicted volatility ν_i, which will, through the dependence of p_{i2} on ν_i and $\hat{\pi}_i$ on p_{i2}, lead to the desired influence:

$$\dot{\nu}_i = \exp(\kappa_i \check{\mu}_i) \exp(\omega_i) - \nu_i. \tag{62}$$

Additionally, to simplify the precision weighting of the volatility prediction error, we use the auxiliary expected precision node γ_i as introduced in

the previous section:

$$\gamma_i = \kappa_i \nu_i \hat{\pi}_i \tag{63}$$

with, for example,

$$\dot{\gamma}_i = \kappa_i \nu_i \hat{\pi}_i - \gamma_i. \tag{64}$$

For the PE about the level of volatility Δ_i, we have previously seen that we want to reach the following posterior:

$$\Delta_i^{(k)} = \frac{\hat{\pi}_i^{(k)}}{\pi_i^{(k)}} + \hat{\pi}_i^{(k)} (\delta_i^{(k)})^2 - 1. \tag{65}$$

This corresponds to the unweighted volatility prediction error. Again, we start by considering a node which in turn reflects the precision-weighted (VO)PE, which we will call E_i:

$$E_i^{(k)} = \frac{1}{2} \frac{\gamma_i^{(k)}}{\check{\pi}_i^{(k)}} \Delta_i^{(k)}. \tag{66}$$

This could evolve according to

$$\dot{E}_i^{(k)} = \Delta_i^{(k)} - \frac{2\check{\pi}_i^{(k)}}{\gamma_i^{(k)}} E_i^{(k)}, \tag{67}$$

which can also be written as

$$\dot{E}_i^{(k)} = \frac{\hat{\pi}_i^{(k)}}{\pi_i^{(k)}} + \hat{\pi}_i^{(k)} (\delta_i^{(k)})^2 - 1 - \Delta_i^{(k)}, \tag{68}$$

with the unweighted PE node Δ_i

$$\Delta_i^{(k)} = \frac{2\check{\pi}_i^{(k)}}{\gamma_i^{(k)}} E_i^{(k)} \tag{69}$$

evolving as

$$\dot{\Delta}_i^{(k)} = E_i^{(k)} - \frac{\gamma_i^{(k)}}{2\check{\pi}_i^{(k)}} \Delta_i^{(k)}. \tag{70}$$

This leads us to the same setup as for the value prediction error nodes, where we considered both an ε_i node and an unweighted PE node δ_i, as depicted in figure 11. In the case of volatility estimation, this again turns

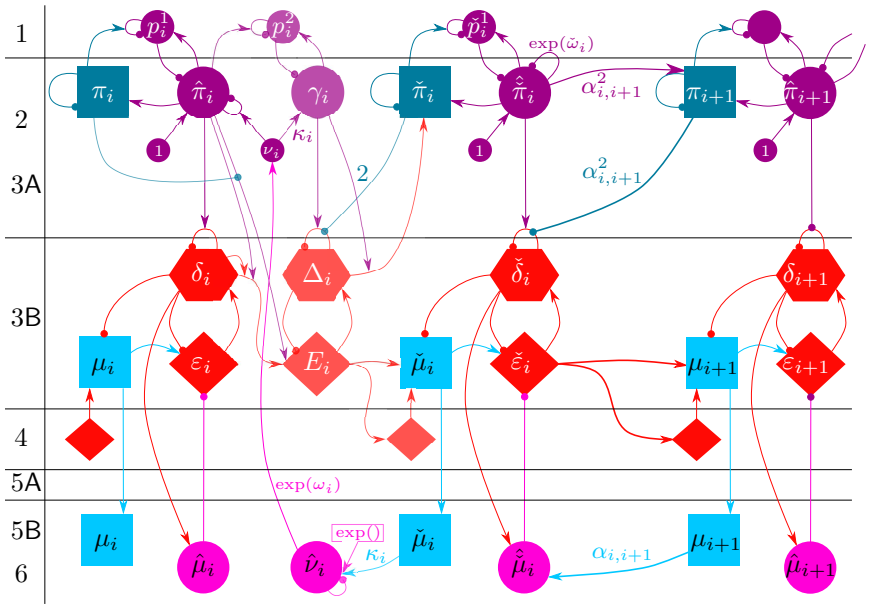

Figure 11: The coupling between a hierarchical level i with its parent $i+1$ in the case of VOPE coupling: implementing the differential equations that describe within-trial dynamics of the nodes.

out to be useful: As before, the precision-weighted PE node E_i can be passed on to Update the volatility parent node $\check{\mu}_i$:

$$\check{\mu}_i^{(k)} = \hat{\check{\mu}}_i^{(k)} + \frac{1}{2}\frac{\gamma_i^{(k)}}{\check{\pi}_i^{(k)}}\Delta_i^{(k)} = \hat{\check{\mu}}_i^{(k)} + E_i^{(k)} \tag{71}$$

with

$$\dot{\check{\mu}}_i^{(k)} = E_i^{(k)} - (\check{\mu}_i^{(k)} - \hat{\check{\mu}}_i^{(k)}) = E_i^{(k)} - \check{\delta}_i. \tag{72}$$

There are three things worth mentioning here. First, in contrast to the previous section, we now use γ_i and Δ_i in the update equations (instead of γ_{i-1} and Δ_{i-1}). This is simply due to the fact that we have put the volatility parent $\check{\mu}_i$ into the same cortical level i as the node μ_i itself. Instead of moving from level i to level $i+1$ for the volatility estimation, we now denote value-related nodes as μ and volatility-related nodes as $\check{\mu}$.

Second, the (unweighted) PE node $\check{\delta}_i$, which is used to feedback to the update of $\check{\mu}_i$, is again a value PE, but now about the mean of the volatility estimate. This VAPE can again be used (in its precision-weighted form $\check{\varepsilon}_i$) to communicate with higher cortical levels μ_{i+1}, completely parallel to the precision-weighted value prediction about the mean, ε_i.

Third, we note that in volatility estimation, we have to distinguish between the (weighted and unweighted) prediciton errors that are used to

update the volatility estimate, E and Δ, and the (weighted and unweighted) prediciton errors *about* the volatility estimate, $\check{\varepsilon}_i$ and $\check{\delta}_i$, which will be used to update any higher cortical level $i+1$ predicting this volatility estimate.

Finally, although the unweighted volatility PE Δ_i is not needed for a feedback to μ_i or $\check{\mu}$, it is still very useful as it is used in the update of the precision of the volatility estimation $\check{\pi}_i$:

$$\check{\pi}_i^{(k)} = \hat{\check{\pi}}_i^{(k)} + \frac{1}{2}(\gamma_i^{(k)})^2 + (\gamma_i^{(k)})^2\Delta_i^{(k)} - \frac{1}{2}\gamma_i^{(k)}\Delta_i^{(k)}, \tag{73}$$

which can be summarized as

$$\check{\pi}_i^{(k)} = \hat{\check{\pi}}_i^{(k)} + \frac{1}{2}(\gamma_i^{(k)})^2 + ((\gamma_i^{(k)})^2 - \frac{1}{2}\gamma_i^{(k)})\Delta_i^{(k)}. \tag{74}$$

This could have the temporal evolution

$$\dot{\check{\pi}}_i^{(k)} = \hat{\check{\pi}}_i^{(k)} + \frac{1}{2}(\gamma_i^{(k)})^2 + ((\gamma_i^{(k)})^2 - \frac{1}{2}\gamma_i^{(k)})\Delta_i^{(k)} - \check{\pi}_i^{(k)}. \tag{75}$$

Finally, turning towards the Prediction step in the volatility estimation, this is again very simple, as we only consider value parents for volatility nodes. For the prediction of the mean, we have

$$\hat{\check{\mu}}_i^{(k+1)} = \check{\mu}_i^{(k)} + \alpha_{i,i+1}\mu_{i+1}^{(k)}, \tag{76}$$

which could evolve as

$$\dot{\hat{\check{\mu}}}_i^{(k+1)} = \check{\mu}_i^{(k)} - \hat{\check{\mu}}_i^{(k+1)} + \alpha_{i,i+1}\mu_{i+1}^{(k)} = \check{\delta}_i + \alpha_{i,i+1}\mu_{i+1}^{(k)}, \tag{77}$$

and for the precision of that prediction, we get

$$\hat{\check{\pi}}_i^{(k+1)} = \frac{1}{\frac{1}{\check{\pi}_i^{(k)}} + \exp(\check{\omega}_i)}, \tag{78}$$

which could evolve as shown above in the value estimation (using the auxiliary nodes $\check{p}_{i1}$):

$$\dot{\hat{\check{\pi}}}_i = 1 - \check{p}_{i1} - \hat{\check{\pi}}_i \exp(\check{\omega}_i) \tag{79}$$

with

$$\dot{\check{p}}_{i1} = \hat{\check{\pi}}_i - \check{\pi}_i\check{p}_{i1}. \tag{80}$$

Note that we do not bother to introduce $\check{p}_{i2}$ or $\check{\nu}_i$ here, as we exclude the possibility of a volatility parent $\check{\mu}_i$ having a volatility parent.

7 Open Questions and Issues

- To Do:
  - Decide on the most useful visualizations for the paper.
  - Check which improvements (in terms of expressing the coupling in the equations) can be translated from the differential equations to the one-step update equations (e.g., introduction of the precision-weighted PE node).
  - Implement the differential equations, examine the stability of the network and present exemplary simulations.
  - Examine the computations for input nodes and their coupling (this is probably not needed for the theoretical treatment of VAPEs and VOPEs, but relevant for the python implementation).
  - Discuss overlap and differences with Karls precision paper (Kanai et al. 2015).
- Equations: The coupling parameters α and κ must be available to both the connection between a node's PE unit and the parent's UPDATE unit, as well as the connection between a node's PREDICTION unit and the parent's UPDATE unit. How can it be ensured that the coupling strength is the same for both connections? For this question we would actually have to consider the learning or update equations for the parameters of the model (see for example the approach in [1]). However, even without writing them down, we could assume that under 'healthy' conditions, the weights of these different connections automatically converge to the same (or at least very similar) values. This opens up new possibilities of modeling abnormal inference: Using the simulations presented above, we could find out how an agent's inference would be altered if these values were not the same, and what range of difference would actually lead to observable changes in the inference.
- Implementation: To whom does the knowledge about parameters, in particular coupling strengths, belong? Ideally, this would belong to the connections themselves and be accessible to both child and parent node. However, in terms of synaptic signalling, how can a node have direct access to a coupling strength other than via the signalled value (which is the product of the coupling strength and the signal)?
- Not covered here: Computations of input nodes and binary input nodes.

References

[1] Bogacz, R. (2017). A tutorial on the free-energy framework for modelling perception and learning. *Journal of Mathematical Psychology*, 76:198–211.

[2] Shipp, S. (2016). Neural elements for predictive coding. *Frontiers in Psychology*, 7(NOV):1–21.