Precision estimation and second-order prediction errors in cortical circuits

Arno Granier*[1], Mihai A. Petrovici[1], Walter Senn†[1], and Katharina A. Wilmes†[1]

[1]Department of Physiology, University of Bern, Switzerland

September 29, 2023

Abstract

Minimization of cortical prediction errors is believed to be a key canonical computation of the cerebral cortex underlying perception, action and learning. However, it is still unclear how the cortex should form and use knowledge about uncertainty in this process of prediction error minimization. Here we derive neural dynamics minimizing prediction errors under the assumption that cortical areas must not only predict the activity in other areas and sensory streams, but also jointly estimate the precision of their predictions. This leads to a dynamic modulatory balancing of cortical streams based on context-dependent precision estimates. Moreover, the theory predicts the existence of second-order prediction errors, i.e. errors on precision estimates, computed and propagated through the cortical hierarchy alongside classical prediction errors. These second-order errors are used to learn weights of synapses responsible for precision estimation through an error-correcting synaptic learning rule. Finally, we propose a mapping of the theory to cortical circuitry.

Introduction

The cerebral cortex has been described as an organ of prediction, where cortical areas attempt to predict the activity in other areas or sensory streams. The computational goal of the cortex would then be to minimize differences between these predictions and actual activity—prediction errors. Neural computations realizing this goal have been proposed as canonical cortical computations [1–5] and as mechanisms supporting the emergence of cognition [6, 7]. Additionally, adopting a probabilistic or Bayesian framework for cortical processing, where uncertainty is taken into account, has proven useful [8, 9]. To harness the power of the probabilistic framework, predictions made by cortical areas should not simply be single potential representations in the target area but rather distributions over the space of potential representations.

In that case, normative theories based on variants of maximum likelihood estimation suggest that cortical prediction errors should be weighted by the precision (reliability, inverse uncertainty) of the predictive distributions. Humans and other animals have indeed been shown to weight prior knowledge and data from multiple modalities by their relative precision during perceptual integration [10, 11], decision-making [12] and sensorimotor control [13, 14], even when the precision changes dynamically [15–18]. This modulatory weighting of prediction errors has gained a central place in the branch of cognitive sciences based on predictive coding [19–21], most notably in models of attention [22–25] and in neuropsychiatry [26–30]. Potential implementations in the cerebral cortex have been discussed, notably in cortico-pulvinar loops [31] or more generally through the action of neuromodulation [32–35]. However, a neurally plausible theoretical formalization of learned and context-dependent prediction error modulation is still missing.

In this work we start with the idea that top-down cortico-cortical gain modulation implements a form of precision weighting of prediction errors [36]. To formalize this idea, we introduce precision estimates computed as a function of current higher-level representations. This in line with rare cases where precision

*corresponding author: arno.granier@unibe.ch

†These authors jointly supervised this work.

was defined as a function of current neuronal activity [22, 31], and in contrast with the majority of literature which formally defines precision as a parameter of the model (e.g. [2, 37, 38]). With this formulation, precision estimates can have a fast, dynamic and context-dependent influence on neural dynamics, while parameters of the precision estimation function, encoded in synaptic weights, slowly integrate statistics of the environment.

We then derive neural dynamics of predictive coding with this additional ingredient. In the resulting neuronal dynamics, the relative importance accorded to bottom-up and top-down cortical streams is dynamically adapted based on estimated precision, in line with Bayes-optimal computation. Additionally, the estimated precision do not have a purely modulatory influence, as the (additive) correction of second-order errors (i.e. errors on the precision estimates) also plays a major role. Moreover, we show that the natural way for a cortical area to learn to estimate the precision of its predictions is through a local synaptic learning rule correcting for postsynaptic second-order errors. Finally, we propose a mapping of our dynamics to cortical circuitry that is consistent with the known laminar target pattern of feedforward and feedback cortico-cortical connections and neural responses of specific cortical cell types.

Results

An energy for cortical function

We propose that one of the goals of the cerebral cortex is to infer a set of latent representations $\boldsymbol{z}$ coherent with an internal model $p(\boldsymbol{x}, \boldsymbol{z}|\boldsymbol{\theta})$ for the current observation $\boldsymbol{x}$ coming from input streams (e.g. sensory information, thalamic activity, etc.). Following the organization of the cortex into specialized areas, we decompose latent representations $\boldsymbol{z}$ into representations $\boldsymbol{u_1}, \dots, \boldsymbol{u_n}$ corresponding to the membrane potentials of neuronal populations in n areas, and denote $\boldsymbol{u_0}$ the observation $\boldsymbol{x}$ (see Fig. 1a). For example, a part of the observation $\boldsymbol{u_0}$ might be encoded in the activity of cells in the retina or the lateral geniculate nucleus (LGN), and latent cortical representations $\boldsymbol{u_1}, \dots, \boldsymbol{u_n}$ might then encode local orientation (V1), shape (IT), color (V4), motion (MT), etc. On longer timescales the cortex should learn parameters $\boldsymbol{\theta}$ of the internal model, corresponding to weights of synaptic connections, so as to better represent its environment.

As a simplifying assumption, we organize areas in a strict generative hierarchy, such that area $l+1$ tries to predict the activity of area l, and nothing else (see Fig. 1a). It does so by sending its output rates $\boldsymbol{r_{l+1}} = \phi(\boldsymbol{u_{l+1}})$ through top-down synapses with plastic weights $\boldsymbol{W_l}$, where ϕ represents the neuronal activation function. Additionally, area $l+1$ similarly estimates and conveys to area l the precision of its prediction through top-down synapses with non-negative plastic weights $\boldsymbol{A_l}$. We further hypothesise that the resulting predictive distribution is the (entropy-maximizing) normal distribution with mean $\boldsymbol{W_l r_{l+1}}$ and precision vector $\boldsymbol{\lambda_l} = \boldsymbol{A_l r_{l+1}}$ (see Fig. 1b). Crucially, the precision is a parameterized function of current higher-level representations, similarly to the mean, and not simply a parameter of the model (see Fig. 1c). This is simply an extension of the notion of prediction, where cortical areas predict the precision (second-order information) in addition of the mean (first-order information).

We can now write our energy or objective for cortical function as the negative log-likelihood

$$E = -\log p(\boldsymbol{x}, \boldsymbol{z}|\boldsymbol{\theta}) = \frac{1}{2}\sum_{l=0}^{n-1} \|\boldsymbol{e_l}\|^2_{\boldsymbol{\lambda_l}} - \frac{1}{2}\sum_{l=0}^{n-1} |\log \boldsymbol{\lambda_l}| + \text{const}\ , \tag{1}$$

where $\boldsymbol{e_l} = \boldsymbol{u_l} - \boldsymbol{W_l r_{l+1}}$ is a prediction error, $\|\cdot\|_{\boldsymbol{\lambda_l}}$ denotes the norm with $\boldsymbol{\lambda_l} = \boldsymbol{A_l r_{l+1}}$ as a metric (i.e. a variance-normalized norm) and $|\cdot|$ denotes (unusually) the sum of components. Note that $\|\boldsymbol{e_l}\|_{\boldsymbol{\lambda_l}}$ is the classical Euclidean norm of standardized errors $\|\boldsymbol{e_l}/\boldsymbol{\sigma_l}\|$, where $\boldsymbol{\sigma_l^2} = \boldsymbol{1}/\boldsymbol{\lambda_l}$ is the estimated variance vector. In other words, here we measure distances as numbers of (estimated) standard deviations away from the (estimated) mean rather than more simply as the Euclidean distance to the (estimated) mean (see Fig. 1d).

From the Bayesian perspective, minimization of E with respect to $\boldsymbol{z}$ leads to a maximum a posteriori estimate of latent variables $\boldsymbol{z^*}$. Then, we can update parameters $\boldsymbol{\theta}$ such that the model assigns a higher probability for the pair of current observation $\boldsymbol{x}$ and optimal latent variables $\boldsymbol{z^*}$, which can be done again by minimizing E, this time with respect to $\boldsymbol{\theta}$. This can be seen as a simplified version of the expectation-maximization

algorithm [39] where we compute a point estimate of latent variables instead of a full posterior distribution.

From the perspective of energy-based models, E as described in the right-hand side of Eqn. 1 seems to be an energy worth minimizing. The first term is a measure of distance between actual representations and predictions. This measure takes into account the precision of predictions: the more a prediction was deemed precise, the more a deviation from it matters. The second term indicates that high precision is preferable. That is, as long as estimating a high precision does not excessively drive up the first term: there must be a balance between the estimated precision and the (average) magnitude of prediction errors. Moreover, this same second term also acts as a regularizer to avoid very small precision estimates, which would be a non-informative solution to minimize the first term.

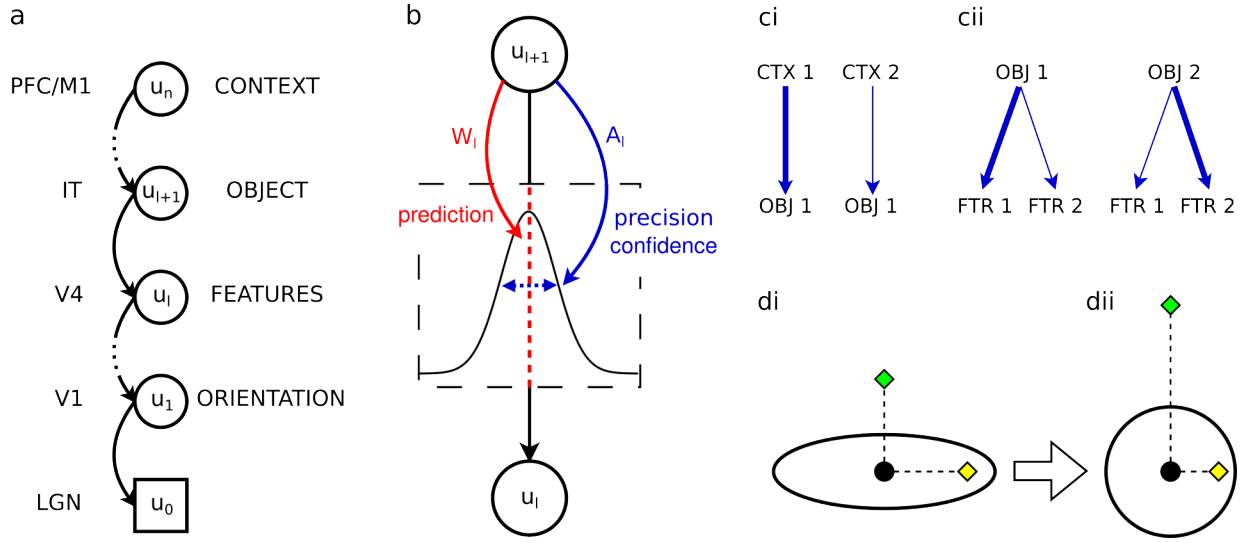

Figure 1: Predictive distributions and generalized distance in the cortical hierarchy. **(a)** Probabilistic model. Latent representations $[\boldsymbol{u_l}]$ are organized in a strict generative hierarchy. **(b)** Predictions are Gaussian distributions. Both the mean $[\boldsymbol{W_l r_{l+1}}]$ (first-order) and the precision $[\boldsymbol{\lambda_l} = \boldsymbol{A_l r_{l+1}}]$ (inverse variance, second-order) are predicted as a function of higher-level activity. **(ci)** The prediction made from the context (CTX) about the presence of a specific object (OBJ) can be more or less precise/confident depending on the context. **(cii)** The prediction made from the presence of a object about the presence of specific features (FTR, e.g. color, shape, etc.) can be more or less precise/confident depending on the object. **(d)** Current representations $[\boldsymbol{u_l}]$ must be compared with predictive distributions. **(di)** Here the green and yellow points are two possible potential current representations $[\boldsymbol{u_l}]$ and the 2d Gaussian is the predictive distribution $[p(\boldsymbol{u_l}|\boldsymbol{u_{l+1}})]$. In that case, the Euclidean distance between points and the mean of the distribution $[\|\boldsymbol{e_l}\| = \|\boldsymbol{u_l} - \boldsymbol{W_l r_{l+1}}\|]$ is insufficient to capture our intuition that the green point lies farther outside the distribution than the yellow one. **(dii)** By taking the precision $[\boldsymbol{\lambda_l}]$ as a metric, or in other words by measuring distances in numbers of standard deviations along axes, we define a more appropriate measure of distance between a point and a Gaussian distribution $[\|\boldsymbol{e_l}\|_{\boldsymbol{\lambda_l}}]$, and our intuition is fulfilled. This is equivalent to measuring the usual Euclidean distance but in a rescaled space where the predictive distribution is the unit circle.

Neuronal dynamics with precision estimation

Similarly to previous work [1, 2, 40, 41], we now derive neuronal dynamics minimizing the energy E through gradient descent. Moreover, here we make use of our precision estimates $\boldsymbol{\lambda_l}$ as metrics for our descent [42]. Note that, since the precision is the Hessian of the Gaussian negative log-likelihood, the resulting dynamics can be interpreted as an approximate second-order optimization scheme (see Methods). This leads to the leaky neuronal dynamics

$$\tau \dot{\boldsymbol{u_l}} = -\boldsymbol{\sigma_l^2} \circ \partial E / \partial \boldsymbol{u_l} = -\boldsymbol{u_l} + \boldsymbol{W_l r_{l+1}} + \boldsymbol{\sigma_l^2} \circ \boldsymbol{a_l} \ , \tag{2}$$

integrating top-down predictions $\boldsymbol{W_l r_{l+1}}$ and (uncertainty-weighted) total propagated errors

$$\boldsymbol{a_l} = \boldsymbol{r_l'} \circ (\boldsymbol{W_{l-1}^T}(\boldsymbol{\lambda_{l-1}} \circ \boldsymbol{e_{l-1}}) + \boldsymbol{A_{l-1}^T \delta_{l-1}}) \tag{3}$$

defined as the sum of precision-weighted prediction errors $\boldsymbol{\lambda_l} \circ \boldsymbol{e_l}$ and second-order errors $\boldsymbol{\delta_l} = (\boldsymbol{\sigma_l^2} - \boldsymbol{e_l^2})/2$, both propagated upwards from the lower area (see Fig. 2a). Here $\circ$ is the componentwise (Hadamard) product and $\boldsymbol{e_l^2} = \boldsymbol{e_l} \circ \boldsymbol{e_l}$. The second-order errors $\boldsymbol{\delta_l}$ are not errors on the prediction of the mean $\boldsymbol{W_l r_{l+1}}$

but errors on the precision estimate $\boldsymbol{\lambda_l} = \boldsymbol{A_l r_{l+1}}$, which are expected to be on average 0 if and only if $\boldsymbol{\lambda_l}$ correctly captures the true precision. Following previous work [43], we suppose that total propagated errors $\boldsymbol{a_l}$ are encoded in the apical dendrites of cortical neurons with somatic membrane potential $\boldsymbol{u_l}$.

These neuronal dynamics (Eqs. 2 and 3) entail two major points of interest, one of gain modulation of errors based on precision estimates (see Fig. 2b) and one of second-order error propagation (see Fig. 2c). In the following section we complete our theoretical framework by deriving synaptic learning rules for parameters $\boldsymbol{W_l}$ and $\boldsymbol{A_l}$. We then come back to neuronal dynamics and unpack further these two points of interest.

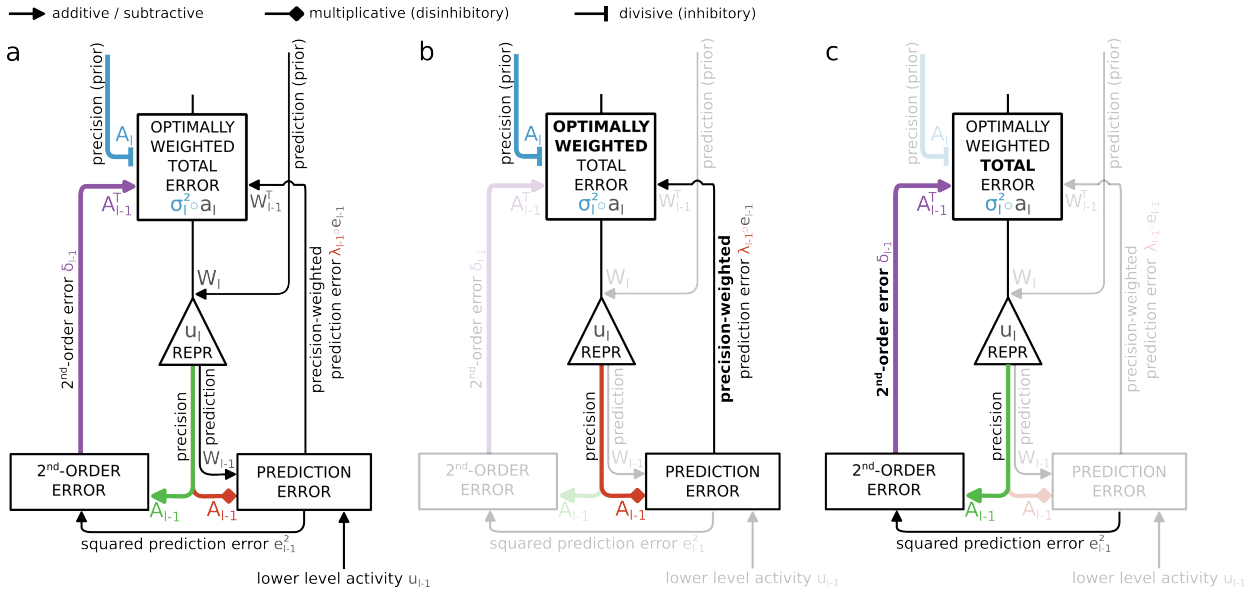

Figure 2: Neuronal dynamics of predictive coding with adaptive precision estimation. **(a)** A schematic depiction of neuronal dynamics (Eqs. 2 and 3). Representations $[\boldsymbol{u_l}]$ are encoded in the somatic membrane potential of pyramidal cells. Prediction errors $[\boldsymbol{e_{l-1}}]$ are first computed by comparing predictions $[\boldsymbol{W_{l-1}r_l}]$ with actual activity or data $[\boldsymbol{u_{l-1}}]$. **(b)** Adaptive balancing of cortical streams based on precision, realized through prediction error modulation. Prediction errors are weighted multiplicatively by the estimated precision of the prediction $[\boldsymbol{\lambda_{l-1}} = \boldsymbol{A_{l-1}r_l}]$. The weighted errors are then propagated upwards $[\boldsymbol{W_{l-1}^T}]$, and weighted divisively by the estimated precision of the higher-level prediction $[\boldsymbol{\lambda_l} = \boldsymbol{A_l r_{l+1}}]$ (multiplication by the prior variance $[\boldsymbol{\sigma_l^2}]$). **(c)** Second-order error propagation. Second-order errors $[\boldsymbol{\delta_{l-1}}]$ are computed by comparing inverse precision estimates $[\boldsymbol{\sigma_{l-1}^2 = 1/\lambda_{l-1}}]$ and squared prediction errors $[\boldsymbol{e_{l-1}^2}]$. They are then uppropagated $[\boldsymbol{A_{l-1}^T}]$ and integrated alongside uppropagated prediction errors into the total error $[\boldsymbol{a_l}]$ which is then used in inference dynamics.

Error-correcting synaptic learning of precision

At equilibrium of neuronal dynamics, weights of synapses carrying predictions can be learned following the gradient

$$\dot{w}_l^{ij} \propto -\partial E/\partial w_l^{ij} = \lambda_l^i e_l^i r_{l+1}^j \,, \tag{4}$$

where w_l^{ij} is the prediction weight from neuron j in area $l{+}1$ to neuron i in area l, $\lambda_l^i e_l^i$ is the postsynaptic precision-weighted prediction error and r_{l+1}^j is the presynaptic rate. This is the classical learning rule for prediction weights in predictive coding.

Weights of synapses carrying precision estimates can also be learned following the gradient of E. The partial derivative $-\partial E/\partial a_l^{ij} = {\delta_l}^i r_{l+1}^j$ indicates that the energy-minimizing update for precision estimation weights a_l^{ij} is one that corrects for postsynaptic second-order errors. To ensure that all components of $\boldsymbol{\lambda_l} = \boldsymbol{A_l r_{l+1}}$ remain positive, we additionally want weights a_l^{ij} to remain non-negative at all time. That is necessary as $\boldsymbol{\lambda_l}$ approximates an inverse variance, which enters both in the energy (Eqn. 1) and the neuronal dynamics (Eqn. 2) as a metric. To enforce this, we postulate that all weights are initialized to positive values and that learning is modulated by the current weight, essentially preventing weights from crossing 0. Since all a_l^{ij}

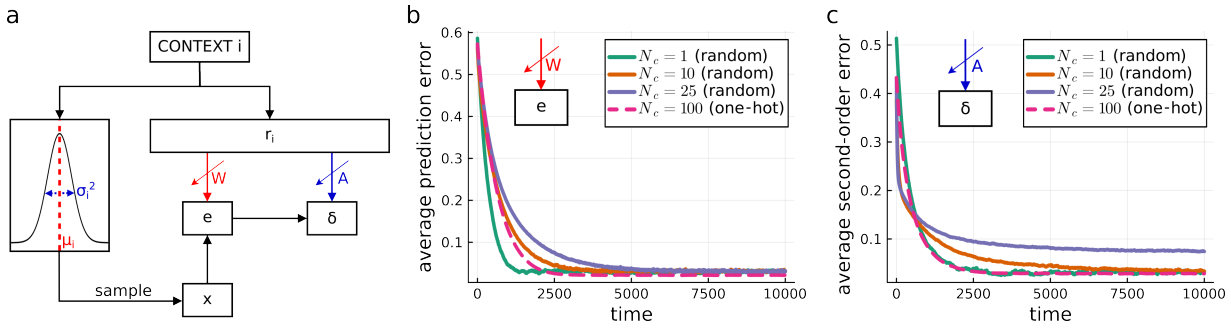

Figure 3: Error-correcting synaptic learning. **(a)** In these simulations, we consider a higher area with N_{l+1} neurons and a lower area with N_l neurons. Specifically, here we take $N_{l+1} = N_l = 100$. The activity vector in the higher area can take N_c different values $[\boldsymbol{r_n},\ n=1,\ldots,N_c]$, to each of which is associated a different mean $[\boldsymbol{\mu_n}]$ and a different variance $[\boldsymbol{\sigma_n^2}]$. The activity in the lower area is then sampled from the Gaussian distribution with this mean and variance. Predictions $[\boldsymbol{W r_i}]$ and precision estimates $[\boldsymbol{A r_i}]$ are formed from the higher-level representation and prediction errors $[\boldsymbol{e} = \boldsymbol{x} - \boldsymbol{W r_i}]$ and second-order errors $[\boldsymbol{\delta} = \boldsymbol{1/A r_i - e^2}]$ are computed and used to learn parameters [$\boldsymbol{W}$ and $\boldsymbol{A}$]. For simulations marked `(random)`, higher-level representations are random binary vectors with an average of 50% of ones. For simulations marked `(one-hot)`, higher-level representations are one-hot encoded. **(b)** Here we show that with the learning rule Eqn. 4 the network correctly learns to estimate the means $[\boldsymbol{\mu_n},\ n=1,\ldots,N_c]$ from higher-level activity $[\boldsymbol{r_n},\ n=1,\ldots,N_c]$. In these simulations we suppose that the precision estimate is 1. **(c)** Here we show that with the learning rule Eqn. 5 the network correctly learns to estimate the precisions $[\boldsymbol{1/\sigma_n^2}]$ from higher-level activity $[\boldsymbol{r_n}]$.

then stay positive, we can interpret this as a simple modulation of the learning rate for precision learning. This leads to the learning rule

$$\dot{a}_l^{ij} \propto -a_l^{ij}\partial E/\partial a_l^{ij} = a_l^{ij}\delta_l^i r_{l+1}^j \ . \tag{5}$$

We proceed to show in simulations that Eqs. 4 and 5 can indeed learn correct mean and precision estimates as a function of higher-level activity. In our simulations, we first randomly select an underlying context. This context determines both the data distribution, from which we sample a data point, and the higher-level representation (see Fig. 3a). The prediction and the precision estimate are functions of the higher-level representation associated with this context. Prediction errors are computed as the distance between the sampled data point and the prediction and are used to learn prediction weights following Eqn. 4, so as to estimate the mean of the data distribution associated with this context (see Fig. 3b). Second-order errors are then computed as the distance between precision estimates and the squared prediction errors and are used to learn the precision estimation weights following Eqn. 5, so as to estimate the precision of the data distribution associated with this context (see Fig. 3c).

These two similar learning rules simply state that synaptic weights evolve towards values that lead to smaller remaining errors. Importantly, all the information needed for learning, namely the presynaptic rate, postsynaptic error and current synaptic weight, is present close to the synapse. With our use of precision estimates as metrics, Eqn. 5 might be seen as a rule for metric learning. Having developed a way to learn top-down precision estimates, we will now further examine how these are used in neuronal dynamics and demonstrate their computational utility.

Adaptive balancing of cortical streams based on precision

For the neuronal dynamics in our model (Eqs. 2 and 3), the relative importance given to top-down predictions and bottom-up prediction errors is controlled by two mechanisms that both modulate the gain of prediction errors. First, the estimated precision of top-down predictions about what the activity of a neuron should be (the prior) impacts divisively the importance of bottom-up errors in the inference dynamics of this neuron (see Fig. 4a). Second, the estimated precision of predictions that a neuron make about what the activity of other neurons should be impacts multiplicatively the importance of errors entailed by these predictions (see Fig. 4b). This weighting is proportional to the more classical Bayes-optimal weighting of top-down prediction (akin to prior) and bottom-up errors (akin to data) by their respective reliabilities, and leads to a maximum a posteriori estimate of latent variables at equilibrium of neuronal dynamics (Eqn. 2). This is useful when

integrating information from sources with different levels of reliability (or noise), as, for example, necessary during multimodal integration (see Fig. 3c and Methods).

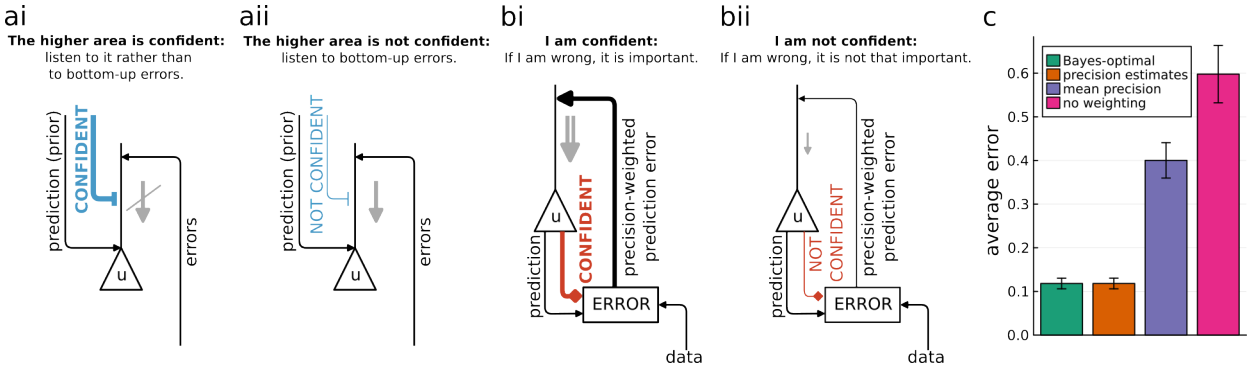

Figure 4: Adaptive balancing of cortical streams based on precision. **(a)** Divisive weighting of errors by the estimated precision of top-down predictions about what the activity of a neuron should be (the prior), corresponding to the multiplicative term $[\boldsymbol{\sigma}_l^2 = \mathbf{1}/\boldsymbol{A}_l \boldsymbol{r}_{l+1}$ in Eqn. 2]. **(b)** Multiplicative weighting of errors by the estimated precision of predictions that a neuron make about what the activity of other neurons should be $[\boldsymbol{\lambda}_{l-1}]$. **(c)** Approximate Bayes-optimal computation in a volatile environment. We consider N_c different classes to which we associated N_c different priors $(\boldsymbol{\mu}_i, \boldsymbol{\sigma}_i^2)$ and data uncertainty $\boldsymbol{\lambda}_i$, $i \in [1, N_c]$. The goal is to infer true latent $\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\sigma}_i^2)$ from noisy data $\boldsymbol{d} \sim \mathcal{N}(\boldsymbol{x}, 1/\boldsymbol{\lambda}_i)$ and prior $\boldsymbol{\mu}_i$. We do that in four different ways that differ in how they take into account uncertainty and precision. `(Bayes-optimal)` a Bayes-optimal estimate, with knowledge of true prior uncertainty and true data precision `(precision estimates)` our dynamics, with knowledge of true prior uncertainty and an estimate of data precision as a function of current representation `(mean precision)` an estimate with knowledge only of the mean prior uncertainty and data precision across classes `(no weighting)` an estimate blind to uncertainty and precision. We plot the average distance between each estimate and the true latent $\boldsymbol{x}$. The error bars indicate the standard deviation.

At the level of a cortical area, the top-down precision estimate controls the balance of bottom-up and top-down information on a neuron-by-neuron basis, providing fine-grained control over what is attended to. We emphasize that, with our formulation of precision estimates as a function of higher-level representations, we can encompass state-, context-, task- or feature- dependent precision signals, depending on what the higher-level representations encode. Moreover, as higher-level representations change, so do the precision signals, providing a mechanism to explain the observed trial-to-trial variability of precision weighting in animals.

Second-order error propagation

In neuronal dynamics (Eqs. 2 and 3), second-order errors $\boldsymbol{\delta}_l$ are propagated through the cortical hierarchy alongside classical precision-weighted prediction errors $\boldsymbol{\lambda}_l \circ \boldsymbol{e}_l$. This forms a second-order stream where cortical areas exchange precision estimates and second-order errors (see Fig. 5a).

To better understand the computational role of this second-order error propagation, we place a network without hidden layers (see Fig. 5b) in supervised learning settings on simple nonlinear binary classification tasks (see Fig 5ci and Methods). Parameters are learned following Eqs. 4 and 5 and, as expected, the precision signal after learning represents the class-specific precision. With our dynamics (see Fig. 5cii) but not with classical predictive coding dynamics (see Fig. 5ciii), the network without hidden layers can solve these nonlinear classification tasks (see Fig. 5d).

At a computational level, this difference can be understood by looking at the way we measure distances. With our model we use the variance-normalized distance between the input and the class distributions, whereas classical predictive coding uses the Euclidean distance between the input and the means of class distributions. At an algorithmic level, the capacity of our network to solve these tasks comes from the computation and propagation of second-order errors. To minimize second-order errors, the network must not only choose the class whose point prediction is closest to the data point (non-informative here), but also the class that best predicts the remaining distance between point prediction and data.

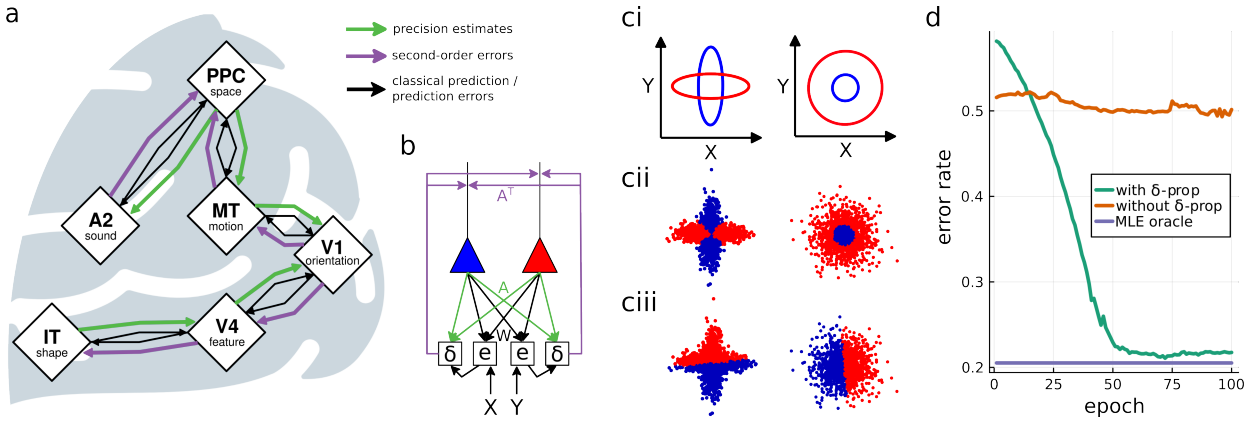

Figure 5: Second-order errors propagation for classification. **(a)** A second-order cortical stream where precision estimates and second-order errors are exchanged between cortical areas. **(b)** A 2x2 network for binary classification. During learning, the X and Y data are sampled from one of the two class distributions and the activity of neurons representing the class is clamped to the one-hot encoded correct class. Parameters $[\boldsymbol{W}, \boldsymbol{A}]$ are then learned following Eqs. 4 and 5. During inference, the activity of neurons representing the class follows neuronal dynamics (without top-down influence) and we read the selected class as the one corresponding to the most active neuron. Prediction error (first-order) propagation is omitted in the depiction. **(c)** The two columns depicts two different tasks. **(ci)** True class distributions. **(cii)** Classification with second-order error propagation. **(ciii)** Classification without second-order error propagation. **(d)** Classification accuracy on the task presented in ci left.

Precision estimation in cortical circuits

We now turn to the task of exploring how our dynamics could be realized in cortical circuits (see Fig. 6). We classically postulate that latent variables $\boldsymbol{u_l}$ are encoded in the somatic activity of a population of pyramidal neurons L6p situated in infragranular cortical layers. Here we choose specifically intracortical pyramidal cells of layer 6 since, as demanded by our theoretical framework, they receive the majority of their input from intracortical long-range projection neurons [44] and send top-down projections to lower cortical areas [45–47]. We propose that these projections notably carry predictions [48–51], but also precision estimates. Following experimental evidence of error encoding in pyramidal cells of cortical layer 2/3 [52–55], we propose that precision-weighted prediction errors $\boldsymbol{\lambda_l} \circ \boldsymbol{e_l}$ and second-order errors $\boldsymbol{\delta_l}$ are computed by two populations of pyramidal neurons situated in supragranular layers, respectively L3e and L3δ. Recent evidence suggests that L3e expresses *Adamts2* and *Rrad* [56], while no functional role has yet been proposed for the third class of supragranular pyramidal cells expressing *Agmat*, which could be L3δ. Additionally, our theory suggests that both type of errors should be integrated into the total propagated errors $\boldsymbol{a_l}$ (as defined in Eqn. 3), which we propose takes place in distal apical dendrites of L6p situated in L4/5a [57], in line with previous work postulating error encoding in segregated dendritic compartments [43, 58].

We now concern ourselves with the precision-balancing of cortical streams entailed by our theory through inhibition and disinhibition of errors. We propose that raw prediction errors $\boldsymbol{e_l}$ are computed in dendrites of L3e by comparing local and top-down inputs from L6p. Precision-weighting of prediction errors might then be realized through top-down gain modulation targeting these dendrites. We propose that this is (at least partially) achieved through a well-known dishinibitory circuit motif involving VIP-expressing interneurons receiving top-down input and inhibiting SST-expressing interneurons which in turn inhibits dendrites of L3e [59–61]. This would entail that VIPs encode a precision signal and supragranular SSTs a variance signal. This is supported by recent 2-photon imaging on rodents placed in an oddball paradigm, where activity ramps up in VIPs and decays in SSTs as a stimulus is repeated (and both show no sign of prediction error computation, contrarily to pyramidal cells) [62]. Moreover, we propose that the uncertainty modulation of total bottom-up errors entailed by our theory (the factor $\boldsymbol{\sigma_l^2}$ in Eqn. 2) is elicited through modulation of L6p apical dendrites by infragranular SST interneurons, which would then encode a precision signal. The laminar specificity of SSTs activity [63] supports this hypothesis.

Finally, we make tentative propositions for circuit-level mechanisms underlying second-order error computation in L3δ. To compute second-order errors, precision estimates must be compared to the magnitude of

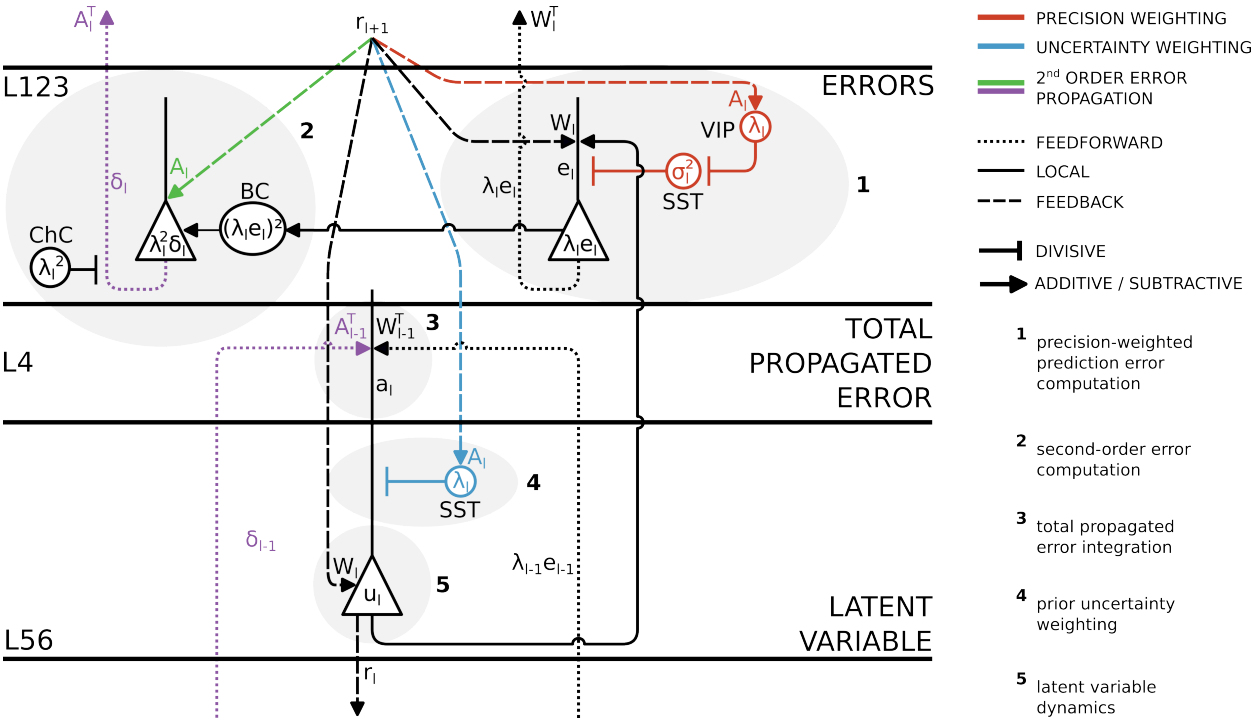

Figure 6: Precision estimation in cortical circuits. Cortical circuit for neuronal dynamics of inference (as described in Eqn. 2 $[\tau \dot{\boldsymbol{u}}_{\boldsymbol{l}} = -\boldsymbol{u}_{\boldsymbol{l}} + \boldsymbol{W}_{\boldsymbol{l}}\boldsymbol{r}_{\boldsymbol{l}+1} + \boldsymbol{\sigma}_{\boldsymbol{l}}^2 \circ \boldsymbol{a}_{\boldsymbol{l}}]$ and Eqn. 3 $[\boldsymbol{a}_{\boldsymbol{l}} = \boldsymbol{r}'_{\boldsymbol{l}} \circ (\boldsymbol{W}_{\boldsymbol{l-1}}^T(\boldsymbol{\lambda}_{\boldsymbol{l-1}} \circ \boldsymbol{e}_{\boldsymbol{l-1}}) + \boldsymbol{A}_{\boldsymbol{l-1}}^T \boldsymbol{\delta}_{\boldsymbol{l-1}})]$). Representations $[\boldsymbol{u}_{\boldsymbol{l}}]$ are held in the somatic membrane potential of L6p. Top-down synapses carrying predictions $[\boldsymbol{W}_{\boldsymbol{l}}\boldsymbol{r}_{\boldsymbol{l+1}}]$ directly excite L6p at proximal dendrites (5). Bottom-up precision-weighted prediction errors $[\boldsymbol{W}_{\boldsymbol{l-1}}^T(\boldsymbol{\lambda}_{\boldsymbol{l-1}} \circ \boldsymbol{e}_{\boldsymbol{l-1}})]$ and second-order errors $[\boldsymbol{A}_{\boldsymbol{l-1}}^T \boldsymbol{\delta}_{\boldsymbol{l-1}}]$ are integrated into total error $[\boldsymbol{a}_{\boldsymbol{l}}]$ in the distal dendrites of L6p as described in Eqn. 3 (3). This total error is then weighted by the prior uncertainty $[\boldsymbol{\sigma}_{\boldsymbol{l}}^2]$ through divisive dendritic inhibition realized by infragranular SST-expressing interneurons (L56-SST) (4). Top-down predictions $[\boldsymbol{W}_{\boldsymbol{l}}\boldsymbol{r}_{\boldsymbol{l+1}}]$ and local representations $[\boldsymbol{u}_{\boldsymbol{l}}]$ are compared in dendrites of L3e. Precision-weighting is then realized through gain modulation of these dendrites by the disinhibitory VIP-expressing (VIP) and SST-expressing (L23-SST) interneurons motif (1). L3δ integrate top-down precision estimates $[\boldsymbol{\lambda}_{\boldsymbol{l}}]$ and local squared precision-weighted prediction errors $[(\boldsymbol{\lambda}_{\boldsymbol{l}} \circ \boldsymbol{e}_{\boldsymbol{l}})^2]$ encoded in basket cells (BC) into re-weighted second-order errors $[\boldsymbol{\lambda}_{\boldsymbol{l}} - (\boldsymbol{\lambda}_{\boldsymbol{l}} \circ \boldsymbol{e}_{\boldsymbol{l}})^2 = \boldsymbol{\lambda}_{\boldsymbol{l}}^2 \circ \boldsymbol{\delta}_{\boldsymbol{l}}]$. Second-order errors $[\boldsymbol{\delta}_{\boldsymbol{l}}]$ are then sent up using the modulatory influence of chandelier cells (ChC) on the axonal initial segment of L3δ.

prediction errors. We propose that the magnitude of (precision-weighted) prediction errors is computed in PV-expressing basket cells [64] from local L3e inputs. At a circuit level, L3e is believed to be separated into two populations L3e^+ and L3e^- encoding the positive and negative part of $\boldsymbol{\lambda}_{\boldsymbol{l}} \circ \boldsymbol{e}_{\boldsymbol{l}}$ respectively [54]. If this is the case, then excitatory projections from L3e^+ and L3e^- to local basket cells, eventually followed by a nonlinear integration by basket cells [65], would be sufficient to perform the needed computation. Basket cells would then project to L3δ realizing a subtractive lateral inhibition [66]. Additionally, we suppose that L3δ receives top-down precision estimates. Now with this setup, the quantity encoded in L3δ would be $\boldsymbol{\lambda}_{\boldsymbol{l}} - (\boldsymbol{\lambda}_{\boldsymbol{l}} \circ \boldsymbol{e}_{\boldsymbol{l}})^2 = \boldsymbol{\lambda}_{\boldsymbol{l}}^2 \circ \boldsymbol{\delta}_{\boldsymbol{l}}$. This is in fact another form of second-order errors, which could be interesting on their own, but to send up $\boldsymbol{\delta}_{\boldsymbol{l}}$ as suggested by our theoretical framework, we postulate that the output of L3δ is modulated by chandelier cells, the other main class of PV-expressing interneurons, which would then encode a squared precision signal. In accordance with this hypothesis, chandelier cells almost exclusively target the axonal initial segment of pyramidal cells and have been shown to be capable of both promoting and inhibiting action potential generation [67]. These propositions, though they are unlikely to prove exactly correct, could provide starting points for experimental investigation of cortical second-order errors.

Discussion

In this work we introduced diagonal estimates of the precision matrix as a function of current higher-level activity and derived neural dynamics of predictive coding with this additional ingredient. In the resulting neuronal dynamics, the relative importance of bottom-up and top-down cortical streams is controlled based on precision estimates, enabling efficient integration of cues with different context-dependent reliabilities. We proposed that in cortical circuits this weighting takes the form of top-down gain modulation realized through a combination of disinhibitory interneuron circuits targeting layer 2/3 pyramidal cells and apical modulation of layer 5/6 pyramidal cells. Moreover, the conditioning of precision estimates on current activity also led to the apparition of second-order prediction errors. Like classical prediction errors, second-order errors are propagated through the cortical hierarchy, leading to nonlinear classification capabilities in a single area. Additionally, these new errors are used for learning weights of synapses responsible for precision estimation.

The brain may use different forms of precision estimates and not only diagonal (vector) estimates as we explored in this work. Obvious alternatives would be scalar and full matrix precision estimates. First, scalar estimates would define the importance granted to all errors in an area, and in that case precision-weighting of errors might be realized through nonspecific release of neuromodulators. Such estimates might be useful for multimodal integration, where one modality as a whole might be reliable or not given context (e.g. vision during the day or during the night). For example, noradrenaline seems to encode environmental volatility [34]. Second, at the other extreme, we might consider full precision matrices. We would then be minimizing an approximate Mahalanobis distance [68] between representations and predictions, taking into account not only stretch but also skew in our metric. Doing so might lead to a theoretically grounded account of lateral connections between prediction error nodes [2], with links to the notion of partial correlations [69]. Moreover, we have conditioned precision estimates on the activity of the same population on which predictions (mean estimates) are conditioned. An alternative would have been to condition precision estimates on a new set of latent variables potentially held by another population of cortical neurons, disentangling the tasks of mean and precision estimation. Furthermore, these estimates might not only be conditioned on cortical but also on subcortical activity. This might help assign computational roles to interactions between the cortex and subcortical structures. Of course all those mechanisms need not be mutually exclusive and could be combined into more complex precision estimates, potentially increasing the explanatory power of predictive coding models of cortical circuits [70]. Note that adaptive precision-weighting and second-order errors might be crucial not only for sensory processes, but also in action selection following the growing tradition of active inference [71].

The dynamics that we presented share some classical weaknesses of predictive coding dynamics concerning biological plausibility that have been tackled elsewhere: weight transport [72, 73], long inference [74], encoding of signed errors [43, 54] and one-to-one connections [75]. Moreover, our learning rules Eqs. 4 and 5 only fulfill weak criteria of locality, as all the information necessary for learning is indeed present in a local patch of cortex around the synapse, but not necessarily precisely at the synapse. Note that to derive our synaptic learning rules we chose implicitly to use the Euclidean metric in our descent scheme. We could consider other metrics, as we did for neuronal dynamics (Eqn. 2), and this might lead to more biologically plausible learning rules. For example, previous work has argued that weights of synapses carrying predictions and targeting proximal dendrites of infragranular pyramidal cells might be learned using the apical activity at equilibrium of neuronal dynamics [43], which in our case would correspond to the learning rule entailed by using, again, precision estimates as a metric for synaptic learning. Additionally, our assumption of a strict hierarchy of latent variables seems at odds with known connectivity between cortical areas. Finally, our account of cortical and notably interneuron circuitry is still incomplete and should definitively be refined and challenged through interactions with experimental work. We believe that our work might provide a theoretical framework to interpret existing experimental results and fruitful directions for following experiments. More than a specific set of predictions, we would like to convey that looking for cortical precision estimates and second-order errors signals might be an interesting venue. More specifically one could look for precision, uncertainty or error magnitude signals in interneuron activity.

In our model, precision estimates computed at each level of the hierarchy as a function of current representations are used as soft multiplicative attention masks on errors, not unlike modern machine learning takes on attention [76]. We hope that our formulation will help link accounts of attention in the predictive

coding framework [22] and in machine learning. Anecdotally, VIP-expressing interneurons were described by experimentalists as "generating a spotlight of attention" [77].

Precision-weighting of prediction errors is often a central element in leading models in the field of neuropsychiatry [26–30]. We hope that this step towards a better theoretical and computational grasp of this mechanism will help in gaining a more holistic understanding of psychopathologies and altered states of mind under the predictive processing framework. The separation of neural mechanisms for prior and data weighting in our model (respectively L6p apical modulation and disinhibition of L3p dendrites) might prove critical to extend models based on pathological over- or under-weighting of either prior or data in a process of Bayesian integration to the whole cortical hierarchy, where activity at one level both represents data for the level above and generates prior for the level below. Moreover, our proposed computational roles for interneuron circuitry might help link accounts of neuropsychiatric disorders in terms of precision-weighting of errors to accounts in terms of cortical excitation-inhibition balance [78, 79]. Future work might include studying the effects of pathological precision estimation in simulations.

Finally, the somatic integration of apical activity in infragranular pyramidal cells has recently been shown to be crucial for perceptual decision-making [80], impaired in anesthesia [81] and placed at the center of theories of conscious processing [82]. Of particular importance is the gain of the coupling compartment between apical and perisomatic regions, controlling the balance between bottom-up and top-down cortical streams. The authors of [82] proposed higher-order thalamus as a major player in the game of controlling this coupling, but also added cortico-cortical and potentially more selective control as an outstanding area of investigation. In our framework, the uncertainty modulation of L6p apical dendrites (or more precisely, of the coupling compartment) would play such a role of locally controlling the relative importance of top-down predictions and bottom-up prediction errors in the inference process (see e.g. Fig. 4a).

Methods

Probabilistic model

Here we precise the form of the probabilistic model. We introduce a notion of strict hierarchy between levels of latent representations by supposing that the joint can be decomposed as

$$p(\boldsymbol{u_0}, \boldsymbol{u_1}, \dots, \boldsymbol{u_n}|\boldsymbol{\theta}) \propto p(\boldsymbol{u_0}|\boldsymbol{u_1}, \boldsymbol{\theta}) p(\boldsymbol{u_1}|\boldsymbol{u_2}, \boldsymbol{\theta}) \dots p(\boldsymbol{u_{n-1}}|\boldsymbol{u_n}, \boldsymbol{\theta}) \tag{6}$$

which can be justified by assuming a Markov property $\forall k,\ p(\boldsymbol{u_k}|\boldsymbol{u_{k+1}}, \dots, \boldsymbol{u_n}, \boldsymbol{\theta}) = p(\boldsymbol{u_k}|\boldsymbol{u_{k+1}}, \boldsymbol{\theta})$ and a uniform top level prior $\boldsymbol{u_n} \sim \mathcal{U}$. Since the distribution of $\boldsymbol{u_l}$ is conditioned on $\boldsymbol{u_{l+1}}$, we call this a generative hierarchy.

We further assume that predictions $\boldsymbol{u_l}|\boldsymbol{u_{l+1}}$ follow a multivariate Gaussian distribution

$$\boldsymbol{u_l}|\boldsymbol{u_{l+1}} \sim \mathcal{N}\left(\boldsymbol{W_l r_{l+1}}, \mathrm{diag}(\boldsymbol{A_l r_{l+1}})^{-1}\right) \tag{7}$$

with mean at point predictions $\boldsymbol{W_l r_{l+1}}$ and diagonal covariance matrix with positive diagonal $\boldsymbol{\sigma_l^2} = \boldsymbol{1}/\boldsymbol{A_l r_{l+1}}$.

Under these two assumptions described in Eqs. 6 and 7, we have the right-hand side equality in Eqn. 1, which we derive in more details in Supplementary Note 1.

Precision estimates as metrics

Modern machine learning has made extensive use of Euclidean gradient descent, such that we now often confound the gradient and the partial derivative [42]. But more generally, for a metric characterized by the positive definite metric tensor $\boldsymbol{D}$, the gradient of the energy is given by

$$(\nabla E)(\boldsymbol{x}) = \boldsymbol{D}^{-1} \frac{\partial E}{\partial \boldsymbol{x}} \tag{8}$$

In this work we chose precision estimates as a metric for neuronal dynamics Eqn. 2, i.e. $\boldsymbol{D} = \mathrm{diag}(\boldsymbol{\lambda_l})$. There are two justifications for this. First and foremost, the resulting neuronal dynamics Eqn. 2 appears to us more neurally plausible, with the explicit leak $-\boldsymbol{u_l}$ and the apical modulation factor $\boldsymbol{\sigma_l^2}$. Second, remark that the precision is the Hessian of the Gaussian negative log-likelihood

$$\frac{\partial^2 - \log f(\boldsymbol{u}; \boldsymbol{m}, \boldsymbol{v})}{\partial \boldsymbol{u}^2} = \mathrm{diag}(\boldsymbol{1}/\boldsymbol{v}) \tag{9}$$

with f the density of a multivariate Gaussian and, importantly, $\boldsymbol{m}, \boldsymbol{v}$ not functions of $\boldsymbol{u}$. Second derivatives of the objective are known to have desirable properties as metrics, from Newton's method to natural gradient descent [83]. Of course, the precision is only a crude approximation of the actual Hessian $\partial^2 E/\partial \boldsymbol{u_l}^2$, since both means and variances in E are in fact functions of current activity $\boldsymbol{u}$. In other words, if we make the approximation of ignoring dependencies of distribution parameters on current activity, the precision is the Hessian of the energy. This is equivalent to considering at each level l that the activity in level

$l+1$ is fixed. In short, our approximation abandons mathematical exactness but retains the idea of a second derivative metric. An intuition of the effect on neuronal dynamics Eqn. 2 is as normalizing the balance of importance between local and lower prediction errors such that the importance of local errors is 1.

Now, to take precision estimates $\boldsymbol{\lambda_l} = \boldsymbol{A_l r_{l+1}}$ as metrics, we do need to be cautious that elements of $\boldsymbol{\lambda_l}$ are strictly positive. Indeed this a condition for $\boldsymbol{\lambda_l}$ to even define a proper (Riemannian) metric. Let us look at how the precision estimation weights evolve through time as defined by Eqn. 5. For each weight a_l^{ij} in $\boldsymbol{A_l}$, we have

$$\lim_{a_l^{ij} \to 0^+} \dot{a}_l^{ij} = 0 \tag{10}$$

Hence, if we initialize elements of $\boldsymbol{A_l}$ to positive values, then at all time $a_l^{ij} > 0$. If we additionally assume that at all time at least one element of $\boldsymbol{r_{l+1}}$ is nonzero, then at all time elements of $\boldsymbol{\lambda_l}$ are strictly positive. With this, skeptics about this change of metric can at least be reassured that we are following a descent direction on E.

Simulation details

Pseudocode for simulations is available in the Supplementary Information, and an implementation in Julia is available at `github.com/arnogranier/precision-estimation`.

For all simulations, we take ϕ the ReLU activation function.

Precision learning

For simulations presented in Fig. 3c, we follow the simulation setup presented in Fig. 3a and described in more details bellow and in Supplementary Algorithm 1.

We consider a higher area with N_{l+1} neurons and a lower area with N_l neurons. We consider N_c different classes of inputs, each with its own distribution $\mathcal{N}(\boldsymbol{\mu_i}, \boldsymbol{\sigma_i^2}), i \in [1, N_c]$, where $\boldsymbol{\mu_i}$ and $\boldsymbol{\sigma_i^2}$ are vectors of size N_l. We initialize all $\boldsymbol{\mu_i}$ following a $\mathcal{U}(-1, 1)$ and all $\boldsymbol{\sigma}_i^2$ following a $\mathcal{U}(1/4, 1)$. Then we choose the representational mode of the higher area, either random binary vectors or one-hot encoded and initialize higher-level representations $\boldsymbol{r_i}, i \in [1, N_c]$ as random binary vectors of size N_{l+1} with on average p ones or one-hot encoded i in N_{l+1}, respectively. The precision estimation matrix $\boldsymbol{A}$ is then initialized as a matrix filled with α, with $\alpha = 1/pN_{l+1}$ for the random binary vector case and $\alpha = 1$ for the one-hot encoded case. We then repeat the following procedure for multiple epochs (1) For each class, sample a data $\boldsymbol{x_i}$ from $\mathcal{N}(\boldsymbol{\mu_i}, \boldsymbol{\sigma_i^2})$ (2) Set the higher-level representation to $\boldsymbol{r_i}$ (3) Compute the precision estimate $\boldsymbol{\lambda_i} = \boldsymbol{A r_i}$ (4) Compute the second-order error $\boldsymbol{\delta_i} = (1/\boldsymbol{\lambda_i} - (\boldsymbol{x_i} - \boldsymbol{\mu_i})^2)/2$ (5) Update $\boldsymbol{A}$ following Eqn. 5. In Fig. 3c, we plot the evolution of $(\sqrt{N_l} N_c)^{-1} \sum_i \|\boldsymbol{\sigma_i^2} - 1/\boldsymbol{A r_i}\|$ through epochs. For Fig. 3c, parameters are $T = 10000, N_{l+1} = N_l = 100, \eta = 0.001$ with N_c varying depending on the simulation.

A similar procedure is used for Fig. 3b, but following Eqn. 4.

Approximate Bayes-optimal integration

For simulations presented in Fig. 4c, we follow the simulation procedure described bellow. Pseudocode for these simulations is presented in Supplementary Algorithm 2 and a mathematical intuition is given in Supplementary Note 3.

We consider a higher area with N_{l+1} neurons and a lower area with N_l neurons. We consider N_c different classes of inputs, each with its own distribution $\mathcal{N}(\boldsymbol{\mu_i}, \boldsymbol{\sigma_i^2}), i \in [1, N_c]$, where $\boldsymbol{\mu_i}$ and $\boldsymbol{\sigma_i^2}$ are vectors of size N_l. We initialize all $\boldsymbol{\mu_i}$ following a $\mathcal{U}(0, 2/N_l)$ and all $\boldsymbol{\sigma_i^2}$ by randomly choosing each component in $\{0.1, 2\}$ with a 50% chance. We initialize the precision estimation matrix $\boldsymbol{A}$ following a $\mathcal{U}(0, 2)$. We additionally collect the mean prior variance vector across classes $\bar{\boldsymbol{\sigma}}^2 = \frac{1}{N_c} \sum_i \boldsymbol{\sigma_i^2}$ and the mean data precision vector across classes $\bar{\boldsymbol{\lambda}} = \frac{1}{N_c} \sum_i \boldsymbol{A}\phi(\boldsymbol{\mu_i})$. We then repeat across epochs the following procedure. For each class i (1) we sample a true target latent $\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{\mu_i}, \boldsymbol{\sigma_i^2})$. We consider that the precision estimation weights are correct such that the precision of the data is $\boldsymbol{\lambda} = \boldsymbol{A}\phi(\boldsymbol{x})$. (2) Then we sample noisy data. Here we want to focus on precision estimation and not mean prediction, so we suppose that the prediction function is the identity, and we then sample data $\boldsymbol{d} \sim \mathcal{N}(\boldsymbol{x}, 1/\boldsymbol{\lambda})$. The goal is then to infer $\boldsymbol{x}$ from data $\boldsymbol{d}$ and prior $\boldsymbol{\mu_i}$. We do that in four different ways that differ in how they take into account uncertainty and precision:
(3i) a Bayes-optimal estimate, with knowledge of true prior variance and true data precision

$$\boldsymbol{u} = (\boldsymbol{\lambda} \circ \boldsymbol{d} + \boldsymbol{\sigma_i^{-2}} \circ \boldsymbol{\mu_i})/(\boldsymbol{\lambda} + \boldsymbol{\sigma_i^{-2}}) \tag{11}$$

(3ii) our dynamics, with knowledge of true prior variance and data precision estimation

$$\tau \dot{\boldsymbol{u}} = -\boldsymbol{u} + \boldsymbol{\mu_i} + \boldsymbol{\sigma_i^2} \circ \boldsymbol{A}\phi(\boldsymbol{u}) \circ (\boldsymbol{d} - \boldsymbol{u}) \tag{12}$$

(3iii) an estimate with knowledge only of the mean prior variance and data precision across classes

$$\tau \dot{\boldsymbol{u}} = -\boldsymbol{u} + \boldsymbol{\mu_i} + \bar{\boldsymbol{\sigma}} \circ \bar{\boldsymbol{\lambda}} \circ (\boldsymbol{d} - \boldsymbol{u}) \tag{13}$$

(3iv) an estimate blind to variance and precision

$$\tau \dot{\boldsymbol{u}} = -\boldsymbol{u} + \boldsymbol{\mu_i} + (\boldsymbol{d} - \boldsymbol{u}) \tag{14}$$

In Fig. 4c, we plot the average distance between each estimate and the true latent $(N_c N_e \sqrt{N_l})^{-1} \|x - u\|$ and its standard deviation.

Nonlinear binary classification

For simulations presented in Fig. 5cd , we built the datasets by sampling $N = 1000$ points $(x_1, y_1), \dots, (x_N, y_N)$ from each of the gaussian distributions represented in Fig. 5ci (first column: $\mathcal{N}([0,0], \text{diag}([1, 1/4]))$ and $\mathcal{N}([0,0], \text{diag}([1/4, 1]))$, second column: $\mathcal{N}([0,0], \text{diag}([9,9]))$ and $\mathcal{N}([0,0], \text{diag}([1/3, 1/3]))$, represented by their 99.7% confidence ellipses) and attaching the corresponding class label (either red or blue).

We then build a 2x2 network where the top level activity is a one-hot representation of the class and the bottom level activity is the coordinate in space (x, y). We train this network in supervised learning settings on the dataset by clamping [84] both top and bottom area to the corresponding elements of the dataset and perform one step of parameters learning as described in Eqs. 4 and 5.

We then test the capacity of our network to classify data by only clamping the bottom level to the data and letting the top level activity follow Eqn. 2. We then select as the output class index the index of the maximum top level activity, and plot the corresponding classification in Fig. 5cii.

Pseudocode for the training and testing procedures is provided in Supplementary Algorithms 3 and 4.

For comparison, we also plot in Fig. 5ciii the classification results obtained with the same 2x2 architecture but using classical predictive coding dynamics

$$\tau \dot{\boldsymbol{u}}_{\boldsymbol{l}} = -\boldsymbol{u}_{\boldsymbol{l}} + \boldsymbol{W}_{\boldsymbol{l}} \boldsymbol{r}_{\boldsymbol{l+1}} + \boldsymbol{r}'_{\boldsymbol{l}} \circ \boldsymbol{W}^{\boldsymbol{T}}_{\boldsymbol{l-1}} \boldsymbol{e}_{\boldsymbol{l-1}} \tag{15}$$

$$\dot{\boldsymbol{W}}_{\boldsymbol{l}} \propto \boldsymbol{e}_{\boldsymbol{l}} \boldsymbol{r}^{T}_{\boldsymbol{l+1}} \tag{16}$$

and following the same training and testing procedures.

In Fig. 5d we plot the associated performance, with the addition of the maximum likelihood estimate with perfect knowledge of the means and variances.

Data availability

All data is generated by the simulation code (see Code availability statement below).

Code availability

Simulation code for this paper can be accessed at `github.com/arnogranier/attention-pc`.

References

1. Rao, R. P. & Ballard, D. H. Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects. *Nature neuroscience* **2,** 79–87 (1999).
2. Friston, K. A theory of cortical responses. *Philosophical transactions of the Royal Society B: Biological sciences* **360,** 815–836 (2005).
3. Bastos, A. M. *et al.* Canonical microcircuits for predictive coding. *Neuron* **76,** 695–711 (2012).
4. Keller, G. B. & Mrsic-Flogel, T. D. Predictive processing: a canonical cortical computation. *Neuron* **100,** 424–435 (2018).
5. De Lange, F. P., Heilbron, M. & Kok, P. How do expectations shape perception? *Trends in cognitive sciences* **22,** 764–779 (2018).
6. Clark, A. Whatever next? Predictive brains, situated agents, and the future of cognitive science. *Behavioral and brain sciences* **36,** 181–204 (2013).
7. Hohwy, J. *The predictive mind* (OUP Oxford, 2013).
8. Fiser, J., Berkes, P., Orbán, G. & Lengyel, M. Statistically optimal perception and learning: from behavior to neural representations. *Trends in cognitive sciences* **14,** 119–130 (2010).
9. Pouget, A., Beck, J. M., Ma, W. J. & Latham, P. E. Probabilistic brains: knowns and unknowns. *Nature neuroscience* **16,** 1170–1178 (2013).
10. Ernst, M. O. & Banks, M. S. Humans integrate visual and haptic information in a statistically optimal fashion. *Nature* **415,** 429–433 (2002).
11. Stein, B. E. & Stanford, T. R. Multisensory integration: current issues from the perspective of the single neuron. *Nature reviews neuroscience* **9,** 255–266 (2008).
12. Kiani, R. & Shadlen, M. N. Representation of confidence associated with a decision by neurons in the parietal cortex. *Science* **324,** 759–764 (2009).

13. Körding, K. P. & Wolpert, D. M. Bayesian integration in sensorimotor learning. *Nature* **427,** 244–247 (2004).
14. Darlington, T. R., Beck, J. M. & Lisberger, S. G. Neural implementation of Bayesian inference in a sensorimotor behavior. *Nature neuroscience* **21,** 1442–1451 (2018).
15. Morgan, M. L., DeAngelis, G. C. & Angelaki, D. E. Multisensory integration in macaque visual cortex depends on cue reliability. *Neuron* **59,** 662–673 (2008).
16. Fetsch, C. R., Turner, A. H., DeAngelis, G. C. & Angelaki, D. E. Dynamic reweighting of visual and vestibular cues during self-motion perception. *Journal of Neuroscience* **29,** 15601–15612 (2009).
17. Qamar, A. T. *et al.* Trial-to-trial, uncertainty-based adjustment of decision boundaries in visual categorization. *Proceedings of the National Academy of Sciences* **110,** 20332–20337 (2013).
18. Noppeney, U. Perceptual inference, learning, and attention in a multisensory world. *Annual review of neuroscience* **44,** 449–473 (2021).
19. Clark, A. The many faces of precision. *Frontiers in psychology* **4,** 270 (2013).
20. Friston, K. Does predictive coding have a future? *Nature neuroscience* **21,** 1019–1021 (2018).
21. Yon, D. & Frith, C. D. Precision and the Bayesian brain. *Current Biology* **31,** R1026–R1032 (2021).
22. Feldman, H. & Friston, K. Attention, uncertainty, and free-energy. *Frontiers in human neuroscience* **4,** 215 (2010).
23. Kok, P., Rahnev, D., Jehee, J. F., Lau, H. C. & De Lange, F. P. Attention reverses the effect of prediction in silencing sensory signals. *Cerebral cortex* **22,** 2197–2206 (2012).
24. Hohwy, J. Attention and conscious perception in the hypothesis testing brain. *Frontiers in psychology* **3,** 96 (2012).
25. Jiang, J., Summerfield, C. & Egner, T. Attention sharpens the distinction between expected and unexpected percepts in the visual brain. *Journal of Neuroscience* **33,** 18438–18447 (2013).
26. Van de Cruys, S. *et al.* Precise minds in uncertain worlds: predictive coding in autism. *Psychological review* **121,** 649 (2014).
27. Barrett, L. F., Quigley, K. S. & Hamilton, P. An active inference theory of allostasis and interoception in depression. *Philosophical Transactions of the Royal Society B: Biological Sciences* **371,** 20160011 (2016).
28. Sterzer, P. *et al.* The predictive coding account of psychosis. *Biological psychiatry* **84,** 634–643 (2018).
29. Carhart-Harris, R. L. & Friston, K. REBUS and the anarchic brain: toward a unified model of the brain action of psychedelics. *Pharmacological reviews* **71,** 316–344 (2019).
30. Friston, K. Computational psychiatry: from synapses to sentience. *Molecular Psychiatry,* 1–13 (2022).
31. Kanai, R., Komura, Y., Shipp, S. & Friston, K. Cerebral hierarchies: predictive processing, precision and the pulvinar. *Philosophical Transactions of the Royal Society B: Biological Sciences* **370,** 20140169 (2015).
32. Moran, R. J. *et al.* Free energy, precision and learning: the role of cholinergic neuromodulation. *Journal of Neuroscience* **33,** 8227–8236 (2013).
33. Schwartenbeck, P., FitzGerald, T. H., Mathys, C., Dolan, R. & Friston, K. The dopaminergic midbrain encodes the expected certainty about desired outcomes. *Cerebral cortex* **25,** 3434–3445 (2015).
34. Lawson, R. P., Bisby, J., Nord, C. L., Burgess, N. & Rees, G. The computational, pharmacological, and physiological determinants of sensory learning under uncertainty. *Current Biology* **31,** 163–172 (2021).
35. Haarsma, J. *et al.* Precision weighting of cortical unsigned prediction error signals benefits learning, is mediated by dopamine, and is impaired in psychosis. *Molecular psychiatry* **26,** 5320–5333 (2021).
36. Shipp, S. Neural elements for predictive coding. *Frontiers in psychology* **7,** 1792 (2016).
37. Bogacz, R. A tutorial on the free-energy framework for modelling perception and learning. *Journal of mathematical psychology* **76,** 198–211 (2017).
38. Millidge, B., Seth, A. & Buckley, C. L. Predictive coding: a theoretical and experimental review. *arXiv preprint arXiv:2107.12979* (2021).
39. Neal, R. M. & Hinton, G. E. in *Learning in graphical models* 355–368 (Springer, 1998).
40. Whittington, J. C. & Bogacz, R. An approximation of the error backpropagation algorithm in a predictive coding network with local hebbian synaptic plasticity. *Neural computation* **29,** 1229–1262 (2017).
41. Scellier, B. & Bengio, Y. Equilibrium propagation: Bridging the gap between energy-based models and backpropagation. *Frontiers in computational neuroscience* **11,** 24 (2017).

42. Surace, S. C., Pfister, J.-P., Gerstner, W. & Brea, J. On the choice of metric in gradient-based theories of brain function. *PLoS computational biology* **16,** e1007640 (2020).
43. Sacramento, J., Ponte Costa, R., Bengio, Y. & Senn, W. Dendritic cortical microcircuits approximate the backpropagation algorithm. *Advances in neural information processing systems* **31** (2018).
44. Zolnik, T. A. *et al.* Layer 6b is driven by intracortical long-range projection neurons. *Cell reports* **30,** 3492–3505 (2020).
45. Markov, N. T. *et al.* Anatomy of hierarchy: feedforward and feedback pathways in macaque visual cortex. *Journal of Comparative Neurology* **522,** 225–259 (2014).
46. Vezoli, J. *et al.* Cortical hierarchy, dual counterstream architecture and the importance of top-down generative networks. *Neuroimage* **225,** 117479 (2021).
47. Rockland, K. S. What do we know about laminar connectivity? *Neuroimage* **197,** 772–784 (2019).
48. Schwiedrzik, C. M. & Freiwald, W. A. High-level prediction signals in a low-level area of the macaque face-processing hierarchy. *Neuron* **96,** 89–97 (2017).
49. Leinweber, M., Ward, D. R., Sobczak, J. M., Attinger, A. & Keller, G. B. A sensorimotor circuit in mouse cortex for visual flow predictions. *Neuron* **95,** 1420–1432 (2017).
50. Schneider, D. M., Sundararajan, J. & Mooney, R. A cortical filter that learns to suppress the acoustic consequences of movement. *Nature* **561,** 391–395 (2018).
51. Garner, A. R. & Keller, G. B. A cortical circuit for audio-visual predictions. *Nature neuroscience* **25,** 98–105 (2022).
52. Keller, G. B., Bonhoeffer, T. & Hübener, M. Sensorimotor mismatch signals in primary visual cortex of the behaving mouse. *Neuron* **74,** 809–815 (2012).
53. Fiser, A. *et al.* Experience-dependent spatial expectations in mouse visual cortex. *Nature neuroscience* **19,** 1658–1664 (2016).
54. Jordan, R. & Keller, G. B. Opposing influence of top-down and bottom-up input on excitatory layer 2/3 neurons in mouse primary visual cortex. *Neuron* **108,** 1194–1206 (2020).
55. Thomas, E. R. *et al.* Predictions and errors are distinctly represented across V1 layers. *bioRxiv,* 2023–07 (2023).
56. O'Toole, S. M., Oyibo, H. K. & Keller, G. B. Prediction error neurons in mouse cortex are molecularly targetable cell types. *BioRxiv,* 2022–07 (2022).
57. Ledergerber, D. & Larkum, M. E. Properties of layer 6 pyramidal neuron apical dendrites. *Journal of Neuroscience* **30,** 13031–13044 (2010).
58. Mikulasch, F. A., Rudelt, L., Wibral, M. & Priesemann, V. Where is the error? Hierarchical predictive coding through dendritic error computation. *Trends in Neurosciences* (2022).
59. Pi, H.-J. *et al.* Cortical interneurons that specialize in disinhibitory control. *Nature* **503,** 521–524 (2013).
60. Lee, S., Kruglikov, I., Huang, Z. J., Fishell, G. & Rudy, B. A disinhibitory circuit mediates motor integration in the somatosensory cortex. *Nature neuroscience* **16,** 1662–1670 (2013).
61. Zhang, S. *et al.* Long-range and local circuits for top-down modulation of visual cortex processing. *Science* **345,** 660–665 (2014).
62. Bastos, G. *et al.* A frontosensory circuit for visual context processing is synchronous in the theta/alpha band. *bioRxiv,* 2023–02 (2023).
63. Muñoz, W., Tremblay, R., Levenstein, D. & Rudy, B. Layer-specific modulation of neocortical dendritic inhibition during active wakefulness. *Science* **355,** 954–959 (2017).
64. Wilmes, K. A., Petrovici, M. A., Sachidhanandam, S. & Senn, W. Uncertainty-modulated prediction errors in cortical microcircuits. *bioRxiv,* 2023–05 (2023).
65. Cornford, J. H. *et al.* Dendritic NMDA receptors in parvalbumin neurons enable strong and stable neuronal assemblies. *Elife* **8** (2019).
66. Tremblay, R., Lee, S. & Rudy, B. GABAergic interneurons in the neocortex: from cellular properties to circuits. *Neuron* **91,** 260–292 (2016).
67. Szabadics, J. *et al.* Excitatory effect of GABAergic axo-axonic cells in cortical microcircuits. *Science* **311,** 233–235 (2006).
68. Mahalanobis, P. C. On the generalized distance in statistics. *National Institute of Science of India* (1936).
69. Lauritzen, S. L. *Graphical models* (Clarendon Press, 1996).

70. Pakan, J. M., Francioni, V. & Rochefort, N. L. Action and learning shape the activity of neuronal circuits in the visual cortex. *Current opinion in neurobiology* **52,** 88–97 (2018).
71. Parr, T., Pezzulo, G. & Friston, K. J. *Active inference: the free energy principle in mind, brain, and behavior* (MIT Press, 2022).
72. Akrout, M., Wilson, C., Humphreys, P., Lillicrap, T. & Tweed, D. B. Deep learning without weight transport. *Advances in neural information processing systems* **32** (2019).
73. Max, K. *et al.* Learning efficient backprojections across cortical hierarchies in real time. *arXiv preprint arXiv:2212.10249* (2022).
74. Haider, P. *et al.* Latent Equilibrium: Arbitrarily fast computation with arbitrarily slow neurons. *Advances in Neural Information Processing Systems* **34** (2021).
75. Millidge, B., Tschantz, A., Seth, A. & Buckley, C. L. Relaxing the constraints on predictive coding models. *arXiv preprint arXiv:2010.01047* (2020).
76. Vaswani, A. *et al.* Attention is all you need. *Advances in neural information processing systems* **30** (2017).
77. Karnani, M. M. *et al.* Opening holes in the blanket of inhibition: localized lateral disinhibition by VIP interneurons. *Journal of neuroscience* **36,** 3471–3480 (2016).
78. Marín, O. Interneuron dysfunction in psychiatric disorders. *Nature Reviews Neuroscience* **13,** 107–120 (2012).
79. Sohal, V. S. & Rubenstein, J. L. Excitation-inhibition balance as a framework for investigating mechanisms in neuropsychiatric disorders. *Molecular psychiatry* **24,** 1248–1257 (2019).
80. Takahashi, N., Oertner, T. G., Hegemann, P. & Larkum, M. E. Active cortical dendrites modulate perception. *Science* **354,** 1587–1590 (2016).
81. Suzuki, M. & Larkum, M. E. General anesthesia decouples cortical pyramidal neurons. *Cell* **180,** 666–676 (2020).
82. Aru, J., Suzuki, M. & Larkum, M. E. Cellular mechanisms of conscious processing. *Trends in Cognitive Sciences* **24,** 814–825 (2020).
83. Amari, S.-I. Natural gradient works efficiently in learning. *Neural computation* **10,** 251–276 (1998).
84. Meulemans, A., Zucchet, N., Kobayashi, S., von Oswald, J. & Sacramento, J. The least-control principle for learning at equilibrium. *arXiv preprint arXiv:2207.01332* (2022).

Acknowledgements

This work has received funding from the European Union 7th Framework Programme under grant agreement 604102 (HBP), the Horizon 2020 Framework Programme under grant agreements 720270, 785907 and 945539 (HBP) and the Manfred Stärk Foundation.

We thank Jakob Jordan and Jean-Pascal Pfister for helpful discussions.

Author contributions

A.G. conceptualized the outlines of the paper, compiled and organized literature materials, performed the simulations and wrote the original draft. A.G. and W.S. participated in the derivation and presentation of the mathematical formalism. All authors participated in conceptualization, interpretation of results, reviewing and editing of the manuscript and approved the final manuscript.

Material and correspondence

Correspondence should be addressed to Arno Granier (arno.granier@unibe.ch).

Competing interests

The authors declare no competing interests.

Supplementary Information

Supplementary Note 1 Energy

The density of a multivariate Gaussian with diagonal covariance $\mathbf{\Sigma} = \mathrm{diag}(\boldsymbol{\sigma}^2)$ with $\boldsymbol{\sigma}^2 > 0$ is

$$f(\boldsymbol{u}; \boldsymbol{\mu}, \boldsymbol{\sigma}^2) = (2\pi)^{-k/2}\det(\mathbf{\Sigma})^{-1/2}\exp\left(-\frac{1}{2}(\boldsymbol{u}-\boldsymbol{\mu})^T\mathbf{\Sigma}^{-1}(\boldsymbol{u}-\boldsymbol{\mu})\right) \tag{17}$$

$$= (2\pi)^{-k/2}\left(\prod_i \lambda_i\right)^{1/2}\exp\left(-\frac{1}{2}\|\boldsymbol{e}\|^2_{\boldsymbol{\lambda}}\right) \tag{18}$$

noting $\boldsymbol{\lambda} = \mathbf{1}/\boldsymbol{\sigma}^2$ where the division is taken elementwise and $\|\boldsymbol{e}\|^2_{\boldsymbol{\lambda}} = \|\boldsymbol{u}-\boldsymbol{\mu}\|^2_{\mathbf{\Sigma}^{-1}} = (\boldsymbol{u}-\boldsymbol{\mu})^T\mathbf{\Sigma}^{-1}(\boldsymbol{u}-\boldsymbol{\mu})$. For the determinant, remark that the determinant of diagonal matrix is the product of its diagonal elements.

We now derive the right-hand side equality in Eqn. 1

$$-\log p(\boldsymbol{x}, \boldsymbol{z}|\boldsymbol{\theta}) = -\log\left(K\prod_{l=0}^{n-1} p(\boldsymbol{u_l}|\boldsymbol{u_{l+1}}, \boldsymbol{\theta})\right) \tag{19}$$

$$= -\sum_{l=0}^{n-1}\log p(\boldsymbol{u_l}|\boldsymbol{u_{l+1}}, \boldsymbol{\theta}) + K \tag{20}$$

$$= -\sum_{l=0}^{n-1}\log\left((2\pi)^{-k_l/2}\left(\prod_i(\boldsymbol{\lambda_l})_i\right)^{1/2}\exp\left(-\frac{1}{2}\|\boldsymbol{e_l}\|^2_{\boldsymbol{\lambda_l}}\right)\right) + K \tag{21}$$

$$= -\sum_{l=0}^{n-1}\log\left((2\pi)^{-k_l/2}\right) - \frac{1}{2}\sum_{l=0}^{n-1}\log\left(\prod_i(\boldsymbol{\lambda_l})_i\right) + \frac{1}{2}\sum_{l=0}^{n-1}\|\boldsymbol{e_l}\|^2_{\boldsymbol{\lambda_l}} + K \tag{22}$$

$$= \frac{1}{2}\sum_{l=0}^{n-1}\|\boldsymbol{e_l}\|^2_{\boldsymbol{\lambda_l}} - \frac{1}{2}\sum_{l=0}^{n-1}|\log\boldsymbol{\lambda_l}| + K \tag{23}$$

where to get Eqn. 19 we used Eqn. 6 and to get Eqn. 21 we used Eqs. 7 and 18.

Supplementary Note 2 Partial derivatives of the energy

We now give a high-level view of the derivation of partial derivatives of the energy E used in neuronal and synaptic dynamics Eqs. 2, 4 and 5. We omit calculation details for the sake of brevity. As a reminder, we set $\boldsymbol{e_l} = \boldsymbol{u_l} - \boldsymbol{W_l}\phi(\boldsymbol{u_{l+1}})$, $\boldsymbol{\lambda_l} = \boldsymbol{A_l}\phi(\boldsymbol{u_{l+1}})$, $\boldsymbol{\sigma_l^2} = \mathbf{1}/\boldsymbol{\lambda_l}$, $\boldsymbol{\delta_l} = (\boldsymbol{\sigma_l^2} - \boldsymbol{e_l^2})/2$ and $\circ$ is the componentwise (Hadamard) product.

For this, we will make use of the following matrix calculus formulas:

$$\forall \boldsymbol{M} \text{ symmetric }, \frac{\partial \boldsymbol{x}^T \boldsymbol{M} \boldsymbol{x}}{\partial \boldsymbol{x}} = 2\boldsymbol{M}\boldsymbol{x} \tag{i}$$

$$\frac{\partial \boldsymbol{g}(\boldsymbol{x})}{\partial \boldsymbol{x}} = \frac{\partial g(f(\boldsymbol{x}))}{\partial f(\boldsymbol{x})} \frac{\partial f(\boldsymbol{x})}{\partial \boldsymbol{x}} \quad \text{(chain rule)} \tag{ii}$$

$$\frac{\partial \mathbf{1}^T \log(\boldsymbol{M}\boldsymbol{x})}{\partial \boldsymbol{x}} = \mathbf{1}^T(\text{diag}(\mathbf{1}/(\boldsymbol{M}\boldsymbol{x}))\boldsymbol{M}) = \boldsymbol{M}^T(\mathbf{1}/(\boldsymbol{M}\boldsymbol{x})) \text{ (with the division being componentwise)} \tag{iii}$$

$$\frac{\partial f(\boldsymbol{x})^T g(\boldsymbol{x})}{\partial \boldsymbol{x}} = \frac{\partial f(\boldsymbol{x})}{\partial \boldsymbol{x}} g(\boldsymbol{x}) + \frac{\partial g(\boldsymbol{x})}{\partial \boldsymbol{x}} f(\boldsymbol{x}) \tag{iv}$$

$$\frac{\partial f(\boldsymbol{x}) \circ g(\boldsymbol{x})}{\partial \boldsymbol{x}} = \frac{\partial f(\boldsymbol{x})}{\partial \boldsymbol{x}} \text{diag}(g(\boldsymbol{x})) + \frac{\partial g(\boldsymbol{x})}{\partial \boldsymbol{x}} \text{diag}(f(\boldsymbol{x})) \tag{v}$$

Latent variables

The derivative with respect to $\boldsymbol{u_l}$ can be decomposed in three terms

$$2\frac{\partial E}{\partial \boldsymbol{u_l}} = \frac{\partial \|\boldsymbol{e_l}\|^2_{\boldsymbol{\lambda_l}}}{\partial \boldsymbol{u_l}} + \frac{\partial \|\boldsymbol{e_{l-1}}\|^2_{\boldsymbol{\lambda_{l-1}}}}{\partial \boldsymbol{u_l}} - \frac{\partial |\log \boldsymbol{\lambda_{l-1}}|}{\partial \boldsymbol{u_l}} \tag{24}$$

We compute those three terms independently.

For the first term, the derivation is straightforward and follow directly from (i) and (ii)

$$\frac{\partial \|\boldsymbol{e_l}\|^2_{\boldsymbol{\lambda_l}}}{\partial \boldsymbol{u_l}} = 2\boldsymbol{\lambda_l} \circ \boldsymbol{e_l} \tag{25}$$

For the second term we first remark that it can be written as $\frac{\partial \boldsymbol{e_{l-1}^T}(\boldsymbol{\lambda_{l-1}} \circ \boldsymbol{e_{l-1}})}{\partial \boldsymbol{u_l}}$, apply (iv) and then develop $\frac{\partial \boldsymbol{\lambda_{l-1}} \circ \boldsymbol{e_{l-1}}}{\partial \boldsymbol{u_l}}$ following (v)

$$\frac{\partial \|\boldsymbol{e_{l-1}}\|^2_{\boldsymbol{\lambda_{l-1}}}}{\partial \boldsymbol{u_l}} = -2\phi'(\boldsymbol{u_l}) \circ \left(\boldsymbol{W_{l-1}^T}(\boldsymbol{\lambda_{l-1}} \circ \boldsymbol{e_{l-1}}) - \frac{1}{2}\boldsymbol{A_{l-1}^T}\boldsymbol{e_{l-1}^2} \right) \tag{26}$$

For the third term remark that $|\log \boldsymbol{x}| = \mathbf{1}^T \log \boldsymbol{x}$, then a straightforward application of (iii) is sufficient

$$\frac{\partial |\log \boldsymbol{\lambda_{l-1}}|}{\partial \boldsymbol{u_l}} = \phi'(\boldsymbol{u_l}) \circ \boldsymbol{A_{l-1}^T}\boldsymbol{\sigma_{l-1}^2} \tag{27}$$

Finally putting it all together we have

$$\frac{\partial E}{\partial \boldsymbol{u_l}} = \boldsymbol{\lambda_l} \circ \boldsymbol{e_l} - \phi'(\boldsymbol{u_l}) \circ \left(\boldsymbol{W_{l-1}^T}(\boldsymbol{\lambda_{l-1}} \circ \boldsymbol{e_{l-1}}) + \boldsymbol{A_{l-1}^T}\boldsymbol{\delta_{l-1}} \right) \tag{28}$$

Prediction weights

The derivative with respect to $\boldsymbol{W_l}$ is simply

$$2\frac{\partial E}{\partial \boldsymbol{W_l}} = \frac{\partial \|\boldsymbol{e_l}\|^2_{\boldsymbol{\lambda_l}}}{\partial \boldsymbol{W_l}} \tag{29}$$

The derivation is straightforward and follow directly from (i) and (ii)

$$\frac{\partial \|\boldsymbol{e_l}\|^2_{\boldsymbol{\lambda_l}}}{\partial \boldsymbol{W_l}} = \frac{\partial \boldsymbol{e_l}^T \mathrm{diag}(\boldsymbol{\lambda_l})\boldsymbol{e_l}}{\partial \boldsymbol{e_l}} \frac{\partial \boldsymbol{e_l}}{\partial \boldsymbol{W_l}} = -2(\boldsymbol{\lambda_l} \circ \boldsymbol{e_l})\phi(\boldsymbol{u_{l+1}})^T \tag{30}$$

and

$$\frac{\partial E}{\partial \boldsymbol{W_l}} = -(\boldsymbol{\lambda_l} \circ \boldsymbol{e_l})\phi(\boldsymbol{u_{l+1}})^T \tag{31}$$

Precision estimation weights

The derivative with respect to $\boldsymbol{W_l}$ can be decomposed in two terms

$$2\frac{\partial E}{\partial \boldsymbol{A_l}} = \frac{\partial \|\boldsymbol{e_l}\|^2_{\boldsymbol{\lambda_l}}}{\partial \boldsymbol{A_l}} - \frac{\partial |\log \boldsymbol{\lambda_l}|}{\partial \boldsymbol{A_l}} \tag{32}$$

We compute those two terms independently. For these we find it easier to compute derivatives element by element.

For the first term remark that $\|\boldsymbol{e_l}\|^2_{\boldsymbol{\lambda_l}} = \sum_i \left(\boldsymbol{e_l}^2\right)_i \sum_j \left(\boldsymbol{A_l}\right)_{i,j} \left(\phi(\boldsymbol{u_{l+1}})\right)_j$ and then it is simple to see that

$$\frac{\partial \|\boldsymbol{e_l}\|^2_{\boldsymbol{\lambda_l}}}{\partial (\boldsymbol{A_l})_{i,j}} = \left(\boldsymbol{e_l}^2\right)_i \left(\phi(\boldsymbol{u_{l+1}})\right)_j \tag{33}$$

For the second term remark that $|\log \boldsymbol{\lambda_l}| = \sum_i \log(\boldsymbol{\lambda_l})_i = \sum_i \log\left(\sum_j (\boldsymbol{A_l})_{i,j}(\phi(\boldsymbol{u_{l+1}}))_j\right)$

$$\frac{\partial |\log \boldsymbol{\lambda_l}|}{\partial (\boldsymbol{A_l})_{i,j}} = \frac{\left(\phi(\boldsymbol{u_{l+1}})\right)_j}{\sum_j (\boldsymbol{A_l})_{i,j} \left(\phi(\boldsymbol{u_{l+1}})\right)_j} = \left(\boldsymbol{\sigma_l^2}\right)_i \left(\phi(\boldsymbol{u_{l+1}})\right)_j \tag{34}$$

Putting it together and writing it in matrix form

$$\frac{\partial E}{\partial \boldsymbol{A_l}} = -\boldsymbol{\delta_l}\phi(\boldsymbol{u_{l+1}})^T \tag{35}$$

Supplementary Note 3 Intuition at equilibrium in the linear case

At equilibrium of Eqn. 2, noting $\boldsymbol{\Lambda_k} = \text{diag}(\boldsymbol{\lambda_k})$, ignoring second-order errors ($\boldsymbol{\delta_{l-1}} = 0$) and working in the linear case $\phi(\boldsymbol{x}) = \boldsymbol{x}$ we have the value at equilibrium of Eqn. 2

$$\boldsymbol{u_l^*} = (\boldsymbol{\Lambda_l} + \boldsymbol{W_{l-1}^T \Lambda_{l-1} W_{l-1}})^{-1}(\boldsymbol{\Lambda_l W_l u_{l+1}} + \boldsymbol{W_{l-1}^T \Lambda_{l-1} u_{l-1}}) \tag{36}$$

where the first term can be interpreted as a normalization factor and the second term as a weighted sum of higher and lower representations "translated in the language" of the local level l through prediction weight matrices. Remark that, if the precision $\boldsymbol{\lambda_{l-1}}$ of the prediction that level l makes about level $l-1$ is negligible compared to the precision $\boldsymbol{\lambda_l}$ of the prediction that level $l+1$ makes about level l, which we will note $\boldsymbol{\lambda_{l-1}/\lambda_l} \to 0$, then the activity of level l goes to the prediction made by level $l+1$ (the prior)

$$\boldsymbol{\lambda_{l-1}/\lambda_l} \to 0 \implies \boldsymbol{u_l^*} \to \boldsymbol{W_l u_{l+1}} \tag{37}$$

Inversely, when the prediction that level $l+1$ makes about what the activity in level l should be (the prior) is deemed unreliable compared to the prediction that level l makes about what the activity of level $l-1$ should be, then the activity of level l goes to a value such that its prediction is the activity in level $l-1$

$$\boldsymbol{\lambda_l/\lambda_{l-1}} \to 0 \implies \boldsymbol{W_{l-1} u_l^*} \to \boldsymbol{u_{l-1}} \tag{38}$$

Supplementary Algorithm 1 Precision learning

Require: T, N_{l+1} N_l, N_c, η, overlap, p
  $\boldsymbol{\sigma}^2 = [1.5 \text{ rand}(N_l) + 0.5 \text{ for } \_ \text{ in } 1{:}N_c]$ ▷ $\boldsymbol{\sigma}^2$ initialization, random uniform between 1/2 and 2
  $\boldsymbol{\mu} = [2 \text{ rand}(N_l) - 1 \text{ for } \_ \text{ in } 1{:}N_c]$ ▷ $\boldsymbol{\mu}$ initialization, random uniform between −1 and 1
  if overlap **then**
    $\boldsymbol{r} = [\text{rand}(N_{l+1}) < p \text{ for } \_ \text{ in } 1{:}N_c]$ ▷ $\boldsymbol{r}$ initialization, random binary vector with p% ones on average
    $\boldsymbol{A} = \text{ones}(N_l, N_{l+1})/(pN_{l+1})$ ▷ $\boldsymbol{A}$ initialization (such that the mean starting $\boldsymbol{\lambda}$ is one)
  else
    $\boldsymbol{r} = [[(\text{j==i}) \ ? \ 1 : 0 \text{ for j in } 1{:}N_{l+1}] \text{ for i in } 1{:}N_c]$ ▷ $\boldsymbol{r}$ initialization, onehot encoded
    $\boldsymbol{A} = \text{ones}(N_l, N_{l+1})$ ▷ $\boldsymbol{A}$ initialization (such that the mean starting $\boldsymbol{\lambda}$ is one)
  end if
  store = []
  for t in 1:T **do**
    for i in $1{:}N_c$ **do**
      $\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{\mu}[\text{i}], \boldsymbol{\sigma}^2[\text{i}])$ ▷ sample lower level data
      $\boldsymbol{\lambda} = \boldsymbol{A}\boldsymbol{r}[\text{i}]$ ▷ Compute precision estimate
      $\boldsymbol{\delta} = 0.5(\mathbf{1}/\boldsymbol{\lambda} - (\boldsymbol{x} - \boldsymbol{\mu}[\text{i}])^2)$ ▷ compute second-order errors
      $\boldsymbol{A} \leftarrow \boldsymbol{A} + \eta \boldsymbol{A} \circ \boldsymbol{\delta}\boldsymbol{r}[i]^T$ ▷ update $\boldsymbol{A}$ following Eqn. 5
    end for
    store[t] = sum(norm([$\boldsymbol{\sigma}^2$[i] - $\mathbf{1}/\boldsymbol{A}\boldsymbol{r}$[i] for i in $1{:}N_c$]))/($\sqrt{N_l}N_c$) ▷ distance between (real) $\boldsymbol{\sigma}^2$ and $\mathbf{1}/\boldsymbol{\lambda}$
  end for

Supplementary Algorithm 2 Approximate Bayes-optimal integration

Require: N_{l+1}, N_l, N_c ϕ, τ, T, N_e

$\boldsymbol{\sigma}^2 = [\text{choice}([0.1, 2]) \text{ for } \_ \text{ in } 1 : N_l] \text{ for } \_ \text{ in } 1{:}N_c]$ ▷ Initialize prior variance

$\boldsymbol{\mu} = [2\text{rand}(N_l)/N_l \text{ for } \_ \text{ in } 1{:}N_c]$ ▷ Initialize prior mean

$\boldsymbol{A} = 2\text{rand}(N_l, N_{l+1})$ ▷ Initialize precision estimation weights

$\bar{\boldsymbol{\sigma}} = \text{sum}(\boldsymbol{\sigma^2})/N_c$ ▷ mean prior variance

$\bar{\boldsymbol{\lambda}} = \text{sum}([\boldsymbol{A\mu}[i] \text{ for i in } 1 : N_c])/N_c$ ▷ mean data precision

err1s, err2s, err3s, err4s = [ ], [ ], [ ], [ ]

for t in 1:N_e **do**

  for i in 1 N_c **do**

    $\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{\mu}[i], \boldsymbol{\sigma}^2[i])$ ▷ sample true data

    $\boldsymbol{\lambda} = \boldsymbol{A}\phi(\boldsymbol{x})$ ▷ compute precision estimate at true data

    $\boldsymbol{d} \sim \mathcal{N}(\boldsymbol{x}, 1/\boldsymbol{\lambda})$ ▷ sample noisy data

    $\hat{\boldsymbol{x}} = (\boldsymbol{\lambda} \circ \boldsymbol{d} + \boldsymbol{\sigma}^{-2}[i] \circ \boldsymbol{\mu}[i])/(\boldsymbol{\lambda} + \boldsymbol{\sigma}^{-2}[i])$ ▷ Bayes-optimal estimate

    err1s.append(norm$(\boldsymbol{x} - \hat{\boldsymbol{x}})/\sqrt{N_l}$)

    $\boldsymbol{u} = \boldsymbol{1}$

    for t in 1:T **do**

      $\boldsymbol{u}$ += $(1/\tau) * (-\boldsymbol{u} + \boldsymbol{\mu}[i] + \boldsymbol{\sigma}^2[i] \circ \boldsymbol{A}\phi(\boldsymbol{u}) \circ (\boldsymbol{d} - \boldsymbol{u}))$ ▷ dynamics with precision estimation

    end for

    err2s.append(norm$(\boldsymbol{x} - \boldsymbol{u})/\sqrt{N_l}$)

    $\boldsymbol{u} = \boldsymbol{1}$

    for t in 1:T **do**

      $\boldsymbol{u}$ += $(1/\tau) * (-\boldsymbol{u} + \boldsymbol{\mu}[i] + \bar{\boldsymbol{\sigma}} \circ \bar{\boldsymbol{\lambda}} \circ (\boldsymbol{d} - \boldsymbol{u}))$ ▷ dynamics with average precision and prior variance

    end for

    err3s.append(norm$(\boldsymbol{x} - \boldsymbol{u})/\sqrt{N_l}$)

    $\boldsymbol{u} = \boldsymbol{1}$

    for t in 1:T **do**

      $\boldsymbol{u}$ += $(1/\tau) * (-\boldsymbol{u} + \boldsymbol{\mu}[i] + (\boldsymbol{d} - \boldsymbol{u}))$ ▷ no weighting

    end for

    err4s.append(norm$(\boldsymbol{x} - \boldsymbol{u})/\sqrt{N_l}$)

  end for

end for

Supplementary Algorithm 3 Training

Require: dataset, $\boldsymbol{W}$, $\boldsymbol{A}$, ϕ, η_w, η_a
Ensure: terms of A are strictly positive, range of ϕ is positive
  for $(\boldsymbol{d}, \boldsymbol{t})$ in dataset **do** ▷ $(\boldsymbol{d}, \boldsymbol{t})$ is ([x,y] data, one-hot target)
    $\boldsymbol{\lambda} = \boldsymbol{A}\phi(\boldsymbol{t})$ ▷ precision estimate
    $\boldsymbol{e} = \boldsymbol{d} - \boldsymbol{W}\phi(\boldsymbol{t})$ ▷ raw error
    $\boldsymbol{\delta} = 0.5(\mathbf{1}/\boldsymbol{\lambda} - \boldsymbol{e}^2)$ ▷ second-order error
    $\boldsymbol{W} \leftarrow \boldsymbol{W} + \eta_w(\boldsymbol{\lambda} \circ \boldsymbol{e})\boldsymbol{t}^T$ ▷ prediction weight learning, Eqn. 4
    $\boldsymbol{A} \leftarrow \boldsymbol{A} + \eta_a \boldsymbol{A} \circ \boldsymbol{\delta}\boldsymbol{t}^T$ ▷ precision estimation weight learning, Eqn. 5
  end for

Supplementary Algorithm 4 Testing

Require: data, $\boldsymbol{W}$, $\boldsymbol{A}$, ϕ, ϕ' τ, T
  inferred_labels = dict()
  $\boldsymbol{t}$ = [0.5, 0.5] ▷ Uniform initialization of top level
  for $\boldsymbol{d}$ in data **do**
    for $i = 1..\mathrm{T}$ **do**
      $\boldsymbol{\lambda} = \boldsymbol{A}\phi(\boldsymbol{t})$ ▷ precision estimate
      $\boldsymbol{e} = \boldsymbol{d} - \boldsymbol{W}\phi(\boldsymbol{t})$ ▷ raw error
      $\boldsymbol{\delta} = 0.5(\mathbf{1}/\boldsymbol{\lambda} - \boldsymbol{e}^2)$ ▷ second-order error
      $\boldsymbol{a} = \phi'(\boldsymbol{t}) \circ (\boldsymbol{W}^T(\boldsymbol{\lambda} \circ \boldsymbol{e}) + \boldsymbol{A}^T\boldsymbol{\delta})$ ▷ Total propagated error Eqn. 3
      $\boldsymbol{t} \leftarrow \boldsymbol{t} + \tau^{-1}(-\boldsymbol{t} + \boldsymbol{a})$ ▷ Neuronal dynamics Eqn. 2 without top down influence
    end for
    inferred_labels[$\boldsymbol{d}$] = argmax($\boldsymbol{t}$) ▷ Most probable class index
  end for