

Title

Abstract

blahhh blahhh blah

Introduction

To survive in an ever-changing environment, animals must flexibly adapt their behavior based on previously encoded and novel information. This context-dependency must also be reflected in the information processing of neural networks underlying intelligent behavior. For instance, when you walk down some staircase in your fully lit basement, your brain might entirely rely on the feedforward (bottom-up) input your senses receive (Fig. 1, left). In contrast, when you walk down the same stairs in complete darkness, your brain might rely entirely on feedback (top-down) signals generated from a staircase model it has formed over previous experiences (Fig. 1, middle).

The importance of these feedback inputs has been emphasized by observations showing that top-down projections outnumber feedforward connections (XXX) and that they modulate (XXX) or even entirely drive (XXX) neuron activity. But how do neural networks switch between a feedforward-dominated and a feedback-dominated processing mode? And how do neural networks combine both input streams wisely? For instance, if you hike down an unexplored mountain in very foggy conditions, your brain receives unreliable visual information. In addition, it can only draw on a shaky prediction about what to expect (Fig. 1, right).

A common hypothesis is that the brain weights different inputs according to their reliabilities. A prominent example of this hypothesis is Bayesian multisensory integration (XXX). According to this theory, neural networks represent information from multiple modalities by a linear combination of the uncertainty-weighted single-modality estimates. Multisensory integration is supported by several observations showing that xxx (XXX). It is conceivable that the same concepts can be employed for the weighting of sensory inputs and predictions thereof (XXX). A central point in the weighting of inputs is the estimation of variances as a measure of uncertainty. However, how the variance of both the sensory input and the prediction can be computed on the circuit level is not resolved yet.

We hypothesised that prediction error (PE) neurons provide the basis for the neural computation of variances. PEs are an integral part of the theory of predictive processing which states that the brain constantly compares incoming sensory information with predictions. When those predictions are wrong, the resulting PEs allow the network to revise the model of the world, thereby ensuring that the predictions are more accurate (XXX). Experimental evidence suggests that these PEs may be represented in the activity of distinct groups of neurons, termed PE neurons (XXX). Moreover, these neurons may come in two types when excitatory neurons exhibit near-zero, spontaneous firing rates (XXX). Negative PE (nPE) neurons mainly increase their activity when the prediction is *stronger* than the sensory input, while positive PE (pPE) neurons mainly increase their activity when the prediction is *weaker* than the sensory input. Indeed, it has been shown that excitatory neurons in layer 2/3 of rodent primary sensory areas can encode negative or positive PEs (XXX).

Here, we show that the unique response patterns of nPE and pPE neurons may provide the backbone for computing both the mean and the variance of sensory stimuli. Furthermore, we suggest a network model with a hierarchy of PE circuits to estimate the variance of the prediction, in addition to the variance of the sensory inputs. In line with multisensory integration, predictions are weighted more strongly than the sensory stimuli when the environment is stable (that is, predictable) but the sensory inputs are noisy. Moreover, we show that predictions are integrated more strongly after a change in the environment, even when the new sensory stimulus is reliable. In addition, we unravel the mechanisms underlying a neuromodulator-induced shift in the weighting of sensory inputs and predictions. In our model, these neuromodulators activate groups of inhibitory neurons like parvalbumin-expressing (PV), somatostatin-expressing (SOM), and vasoactive intestinal peptide-expressing (VIP) interneurons (XXX). In a computational model, these interneurons have been shown to establish a multi-pathway balance of excitation and inhibition that is the basis for nPE and pPE neurons (XXX). By breaking this balance, the excitatory neurons change their baseline firing rate and gain, leading to a biased variance estimation. Finally, we show that this weighting can be understood as the neural manifestation of the contraction bias (XXX).

../results/figures/final/Fig_1.pdf

Figure 1. Neural network model to flexibly integrate sensory information and predictions.

(A) Example illustration of context-dependent integration of information. Left: When walking down a staircase that is clearly visible, the brain might rely solely on external sensory information. Middle: When walking down the same stairs in the absence of visual information, the brain might rely on predictions formed by previous experience. Right: When climbing down an unexplored mountain in foggy conditions, the brain might need to integrate sensory information and predictions at the same time. (B) Illustration of a prediction-error (PE) circuit with both negative and positive PE (nPE/pPE) neurons that receive inhibition from three different inhibitory interneuron types: parvalbumin-expressing (PV), somatostatin-expressing (SOM), and vasoactive intestinal peptide-expressing (VIP) interneurons. Local excitatory connections are not shown for clarity. (C) Illustration of network model that estimates the mean and variance of the external sensory stimuli. The core of this network model is the PE circuit shown in (B). The lower-level V neuron encodes the variance, while the lower-level M neuron encodes the mean of the sensory input. (D) Same as in (C) but the feedforward input is the activity of the lower-level M neuron.

Results

nPE and pPE neurons as the basis for estimating mean and variance of sensory stimuli

We hypothesise that the distinct response patterns of negative and positive prediction-error (nPE/pPE) neurons represent the backbone for estimating the mean and the variance of sensory stimuli. nPE neurons only increase their activity relative to a baseline when the sensory input is weaker than predicted, while pPE neurons only increase their activity relative to a baseline when the sensory input is stronger than predicted. Moreover, both nPE and pPE neurons remain at their baseline activity when the sensory input is fully predicted (XXX). Assuming that the prediction equals the mean of the sensory stimulus, the PE neurons, hence, encode the deviation from the mean. Thus, the squared sum of nPE and pPE neuron activity represents the variance of the feedforward input.

To test our hypothesis, we studied a rate-based mean-field network the core of which is a prediction-error (PE) circuit with excitatory nPE and pPE neurons, as well as inhibitory parvalbumin-expressing (PV), somatostatin-expressing (SOM), and vasoactive intestinal peptide-expressing (VIP) interneurons (Fig. 1B). While the excitatory neurons are simulated as two coupled point compartments to emulate the soma and dendrites of elongated pyramidal cells, respectively, all inhibitory cell types were modeled as point neurons. The connectivity of and inputs to the network were chosen such that the excitatory (E) and inhibitory (I) pathways onto the pyramidal cells were balanced because it has been shown that

../results/figures/final/Fig_2.pdf

Figure 2. Prediction-error neurons provide the basis for estimating mean and variance of sensory stimuli. (A) Illustration of the inputs with which the network shown in 1C is stimulated. Network is exposed to a sequence of constant stimuli drawn from a uniform distribution. Stimulus duration is XXX. (B) PE neuron activity hardly changes with stimulus strength (left) but strongly increases with stimulus variability (right). (C) Interneuron activity strongly changes with stimulus strength (left) but hardly changes with stimulus variability (right). (D) M neuron correctly encodes the mean of the sensory stimuli. Left: Illustration of the input synapses onto the M neuron. Middle: Activity of the M neuron over time for a uniform distribution with mean XXX and standard deviation XXX. Right: Normalised mean-squared error (MSE) between the running average and the M neuron activity for different parametrizations of the stimulus distribution. (E) V neuron correctly encodes the variance of the sensory stimuli. Left: Illustration of the input synapses onto the V neuron. Middle: Activity of the V neuron over time for a uniform distribution with mean XXX and standard deviation XXX. Right: Normalised mean-squared error (MSE) between the instantaneous variance and the V neuron activity for different parametrizations of the stimulus distribution.

this E/I balance is necessary for nPE and pPE neurons to emerge (XXX, see Methods).

In addition to this core circuit, we model a memory (M) neuron that perfectly integrates the activity of the PE neurons (Fig. 1C). In accordance with XXX, we assume that the pPE neuron excites the memory neuron, while the nPE neuron inhibits this neuron (for instance, through lateral inhibition, here not modeled explicitly). The M neuron connects to the apical dendrites of the PE neurons and some of the interneurons (here, VIP and PV neurons, see Methods for more details). In this network, the M neuron serves as a prediction that is dynamically updated when new sensory information is available. We furthermore simulate a downstream neuron (termed V neuron), modeled as a leaky integrator with a squared activation function, that receives excitatory output synapses from the PE neurons. Hence, in this setting, the V neuron encodes the variance of the sensory stimuli (Fig. 1C).

To show that this network can indeed represent mean and variance in the respective neurons, we stimulate it with a sequence of step-wise constant inputs drawn from a uniform distribution (Fig. 2A), assuming that the sensory stimulus varies over time. In line with the distinct response patterns for nPE and pPE neurons, these neurons change only slightly with increasing stimulus mean but increase strongly

with input variance (Fig. 2B). This is in contrast to the three interneurons that strongly increase with stimulus mean while they only moderately increase with stimulus variance (Fig. 2C). The activity of the memory neuron M gradually approaches the mean of the sensory inputs (Fig. 2D, middle), while the activity of the V neuron approaches the variance of the inputs (Fig. 2E, middle). This is true for a wide range of input statistics (Fig. 2D-E, right) and input distributions (Fig. S1). Small deviations from the true mean occur mainly for larger input variances, while the estimated variance is fairly independent of the input statistics tested.

XXX coming soon: Paragraph on network beyond mean-field XXX

In summary, nPE and pPE neurons can be the basis to estimate the mean and the variance of sensory stimuli that vary over time.

Estimating variances of sensory inputs and predictions requires a hierarchy of PE circuits

Following the ideas of Bayesian multisensory integration (XXX), the weighting of sensory stimuli and predictions thereof would require knowledge of their variances. As we have shown in the previous section, the variance of the sensory stimulus can be estimated using PE neurons. We hypothesise that the same principles apply to computing the variance of the prediction. Hence, we augment the network with a *higher* PE circuit that receives output synapses from the M neuron of the *lower* PE circuit (Fig. 1D). Both subnetworks are modeled the same, except that the M neuron in the higher PE circuit evolves more slowly than the one in the lower PE circuit.

To test the network’s ability to estimate the variances correctly, we stimulated the network with a sequence of inputs. In each trial one stimulus is shown to the network. To account for the stimulus variance, each stimulus is composed of n constant values drawn from a normal distribution with mean μ_{stim} and variance σ_{stim}^2 , and presented one after the other. To account for potential changes in the environment, in each trial, we draw μ_{stim} from a uniform distribution (Fig. 3A). Hence, the inputs change on two different time scales, with stimulus variability (faster time scale) and trial variability (slower time scale).

As expected, the neurons’ activity increase for both stimulus and trial variances (Fig. 3B). While the neurons in the lower PE circuit increase more strongly with stimulus variability, the neurons in the higher PE circuit increase more strongly with trial variability, indicating that the different subnetworks process different aspects of the inputs. We first consider two limit cases. In the first limit case, a different but low-variance stimulus is presented in each trial (Fig. 3C, left). In line with the ideas of multisensory integration (XXX), the network should therefore follow the sensory inputs closely and ignore the predictions. When we arithmetically calculate the weighted output (Fig. 3C, middle) based on the feedforward and feedback inputs, and the sensory weight (Fig. 3C, right), the network correctly represents mostly the sensory input (for more details, see Methods). In the second limit case, the same but high-variance stimulus is presented in each trial (Fig. 3D, left). According to the theory, the network should downscale the sensory feedforward input and weight the prediction more strongly. Indeed, the weighted output of the network shows a clear tendency to the mean of the stimuli (Fig. 3D, middle), also reflected in the low sensory weight (Fig. 3D, right).

In a next step, to validate the network responses more broadly, we systematically varied the trial and stimulus variability independently. If both variances are similar, the sensory weight approaches 0.5, reflecting equal contribution of sensory inputs and predictions to the weighted output. Only if both variances are zero, the network represents the sensory input perfectly. In line with the limit case examples above, if the stimulus variance is larger than the trial variance, the network weights the prediction more strongly than the sensory input. This is reversed if the stimulus variance is smaller than the trial variance (Fig. 3E). Because the network dynamically estimates the mean and variances of the sensory input and the prediction, the weighted output and the sensory weight changes accordingly when the input statistics changes (Fig. S2).

The first limit case (Fig. 3C) shows that even in a sensory-driven input regime, the prediction is weighted more at the beginning of a new trial than in the steady state. This is further confirmed in simulations in which the trial duration was shortened. For those simulations, the prediction even outweighs the sensory input, reflected in a very low sensory weight (Fig. 3F). This suggests that predictions influence neural activity more significantly in experiments that rely on very fast stimulus changes.

It has been shown [speculated?], that sensory inputs or predictions are overrated in some psychiatric disorders (XXX). We thus wondered which network properties might bias the estimation of the variances,

../results/figures/final/Fig_3.pdf

Figure 3. Estimating variances of sensory inputs and predictions with hierarchical PE circuits.

(A) Illustration of the stimulation protocol. Network is exposed to a sequence of stimuli (one stimulus per trial). To account for stimulus variability, each stimulus is represented by xxx stimulus values drawn from a normal distribution with mean μ_{stim} and σ_{stim}^2 . To account for the volatility of the environment, in each trial the stimulus mean μ_{stim} is drawn from a uniform distribution (denoted trial variability). Trial duration = xxx. (B) Neuron activity increases with both stimulus and trial variability. Neurons in the lower PE circuit increase more strongly with stimulus variability. Neurons in the higher PE circuit increase more strongly with trial variability. (C) Limit case example in which the stimulus variability is low but the trial variability is high. Left: Illustration of the stimulation protocol. Middle: Weighted output follows closely the sensory stimuli. Right: Sensory weight (function of the variances, see text) close to 1, indicating that the network ignores the prediction. Input statistics: XXX. (D) Limit case example in which the stimulus variability is high but the trial variability is low. Left: Illustration of the stimulation protocol. Middle: Weighted output pushed towards the mean of the sensory stimuli. Right: Sensory weight close to zero, indicating that the network ignores the sensory stimuli. Input statistics: XXX. (E) Predictions are weighted more strongly when the stimulus variability is larger than the trial variability. (F) Predictions are weighted more strongly at the beginning of a new trial and quickly changing stimuli.

and, consequently, the weighting of different input streams. In our network, the M neuron evolves faster in the lower subnetwork than in the higher one. We identified the speed at which the M neurons are updated with new information as a decisive factor in the integration of inputs. To show this, we varied the weights from the PE neurons onto the lower-level M neuron. If the M neuron evolves too slowly, the prediction is overrated. In contrast, if the M neuron incorporates new information too quickly, the sensory input is overrated (Fig. S3A). While the speeds at which the activity of the M neurons evolve may underlie pathological weighting of inputs, the precise activation function of the V neurons is less pivotal. When we replaced the squared activation function with a linear, rectified function, the V neurons do not encode the variance but the averaged absolute deviation of the sensory stimuli. However, the sensory weight is only slightly shifted to larger values for low trial/high stimulus variability (Fig. S3B).

In summary, we show that the variances of both the sensory inputs and predictions thereof can be dynamically computed in networks comprising a lower and higher PE circuit. The model shows that predictions are trusted more strongly at the beginning of a new stimulus, and if sensory inputs are noisy on a short time scale while predictable on longer time scales.

../results/figures/final/Fig_4.png

Figure 4. Neuromodulator-based shifts in the weighting of sensory inputs and predictions.

(A) Neuromodulators acting on the three interneurons may shift the weighting of sensory inputs and predictions. The changes depend on the type/s of interneurons targeted by the additional excitatory input emulating a neuromodulator (additional input = XXX). Considered are two limit cases (upper row: more sensory-driven before modulation, lower row: more prediction-driven before modulation). The combination of interneurons targeted is illustrated below. The results are shown for three different PE circuits, specified in the main text. (B) Illustration showing how the sensory weight depends on changes in both the stimulus and trial variability. (C) The M and V neuron activities depend on the PE neuron activities. Hence, perturbing the nPE and pPE neurons must change the estimation of mean and variance. While stimulating the lower PE neurons affects both the lower and higher mean and variance estimation, stimulating the higher PE neurons only affects the V and M neurons in the same subnetwork. (D) Illustration of the mechanisms underlying the biased estimation of mean and variance when PE neurons are perturbed. Both changes in the baseline (left) and gain (right) of PE neurons can contribute to the changes observed in (C). Illustration based on toy model described in Methods. (E) Modulated interneurons change the weighting by changing the overall baseline and the overall gain of PE neurons (sum of changes in nPE and pPE neurons). Whether and how a neuromodulator changes the sensory weight, hence, depends on the interneuron targeted and the effect this interneuron has on baseline and gain of the PE neurons, which in turn does depend on the network it is embedded in.

Biasing the weighting of sensory inputs and predictions by neuromodulators

The brain’s flexibility and adaptability are not least because a plethora of neuromodulators influence the activity of neurons in a variety of ways (XXX). A prominent target of neuromodulatory inputs is inhibitory neurons (Cardin 2019, XXX). Moreover, distinct interneuron types are differently (in-)activated by those neuromodulators. For instance, it has been shown that XXX (XXX). We, therefore, wondered if and how the weighting of sensory inputs and predictions thereof may be biased when neuromodulators activate distinct interneuron types.

To this end, we modeled the presence of neuromodulators by injecting an additional excitatory input into one or two interneuron types. We reasoned that the network effect of a neuromodulator not only depends on the interneuron type it targets but also on the inputs this neuron receives and the connections it makes with other neurons in the network. We, therefore, tested three different mean-field networks that differ with respect to the distribution of sensory inputs and predictions onto the interneurons, and

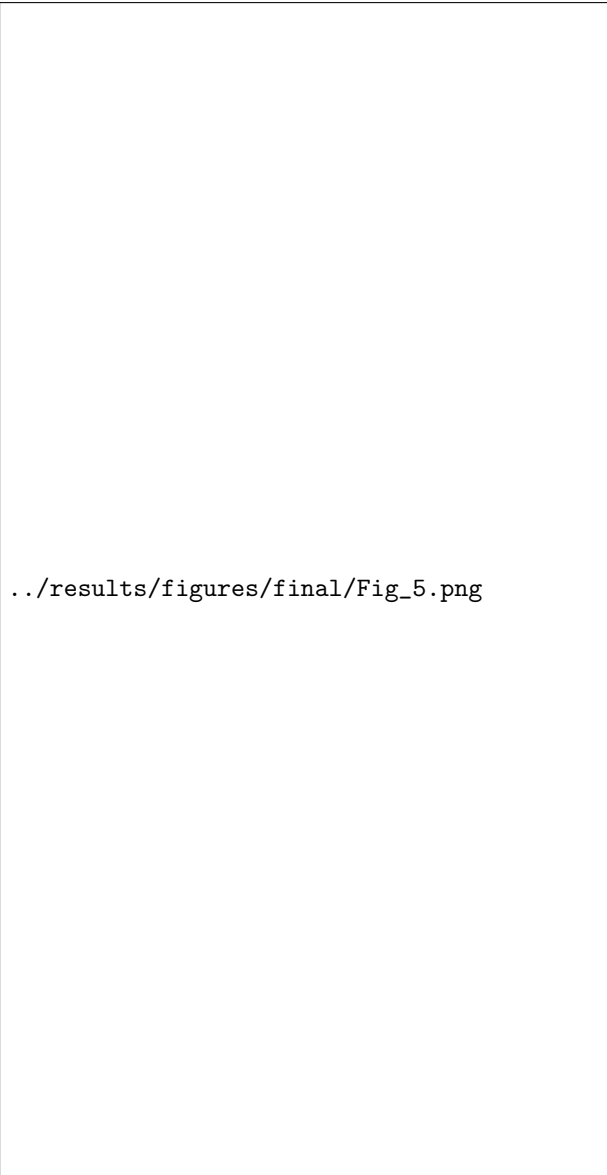
the underlying connectivity. The commonality across those networks is that they exhibit an E/I balance of excitatory and inhibitory pathways onto the PE neurons (XXX). Across the different mean-field networks tested, activating a SOM or VIP neuron individually forces the networks to weigh both inputs more equally. As a consequence, predictions are overrated in a sensory-driven input regime. Similarly, sensory inputs are overrated in a prediction-driven input regime. Interestingly, when both interneuron types are activated to the same degree, this effect disappears (Fig. 4A, left). In contrast, stimulating PV neurons biases the network’s output towards predictions. This effect is even more pronounced when PV and SOM, or PV and VIP neurons are activated simultaneously (Fig. 4A, middle and right).

In the previous simulations, we assumed that a neuromodulator acts globally, that is, on the interneurons in both the lower and the higher PE circuit. While this agrees with experimental data showing that XXX (XXX), we note that neuromodulators may also act more locally. The effect of stimulating an interneuron type in the lower PE circuit on the sensory weight is mostly the opposite of activating the same interneuron in the higher PE circuit (Fig. S4). For instance, the sensory inputs are overrated when the higher-level VIP neuron is activated, while the prediction is overrated when the lower-level VIP neuron is activated. When VIP and SOM neurons are stimulated equally, the sensory weight remains unchanged, independently of which PE circuit is targeted by the neuromodulator.

What are the mechanisms that give rise to these effects? And how do the combined local changes give rise to the global one observed in our network simulations? The sensory weight is chosen to be a function of the V neurons of the lower and higher PE circuit. Hence, any changes to the sensory weight result from changes to the neurons encoding the variances (Fig. 4B). In our network, the V neurons only receive excitatory output synapses from PE neurons. Hence, any changes in the sensory weights upon activation of interneurons must be due to changes in the PE neurons. To disentangle the effect of nPE and pPE neurons, we perturbed those neurons individually in both the lower or higher subnetwork by injecting either an inhibitory or excitatory additional input (Fig. 4C). Stimulating either PE neuron in the lower subnetwork increases the activity of the lower-level V neuron strongly. Moreover, the higher-level V neuron is also slightly affected. At first, this is counterintuitive because the V neuron in the higher subnetwork does not receive direct synapses from the PE neurons in the lower subnetwork. However, the activity of the lower-level M neuron encoding the prediction increases with an excitatory input onto the pPE neuron and decreases with an excitatory input onto the nPE neuron (the opposite is true for an inhibitory input). Because neurons in the higher PE circuit receive synapses from the lower-level M neuron, the activity of the higher-level V neuron is also affected. In contrast, stimulating either PE neuron in the higher subnetwork increases the activity of the higher-level V neuron but leaves the lower-level M and V neurons unaffected (Fig. 4C).

Stimulating PE neurons may cause both an increase in the baseline activity and a change in the neuron’s gain. To disentangle both effects, we illustrate each contribution separately using a mathematically tractable toy model (see Methods for more details). The variance estimated in the lower subnetwork increases with both increasing baseline and gain of the lower-level PE neurons. In contrast, when the gain of those PE neurons decreases, so does the variance. Similarly, the variance estimated in the higher subnetwork is equally influenced by changes in baseline and gain of the higher-level PE neurons. Moreover, changes to the baseline and the gain of the lower-level PE neurons increases the higher-level variance as a result of a biased prediction. Furthermore, the mean of the sensory stimuli is overpredicted (underpredicted) when the baseline or the gain of the lower-level pPE (nPE) neuron increases, or when the gain of the lower-level nPE (pPE) neuron decreases. In summary, if an interneuron causes an additional inhibitory input to a PE neuron, the neuron’s gain is reduced. If an interneuron causes an additional disinhibitory input to a PE neuron, the neuron’s baseline and gain are increased (Fig. 4D).

This suggests that to understand the effect of neuromodulators on the sensory weight, we need to unravel the effect of interneuron activation on baseline and gain of PE neurons. To this end, we stimulated either PV, SOM, or VIP neurons independently for all three mean-field networks and measured the changes to baseline and gain of both PE neurons (Fig. 4E). In all three networks tested, activating PV neurons decreases both quantities, leading to a decrease in the estimated variance (Fig. 4E & Fig. S5). Stimulating SOM or VIP neurons decreases the overall gain but increases the baseline activity of the PE neurons. Whether and how much the gain of the nPE or pPE neuron is reduced depends on the inputs onto SOM and VIP neurons, and the connectivity they make with other neurons in the network. Similarly, how much the baseline is elevated depends on the specifics of the mean-field network (Fig. 4E & Fig. S5). Hence, whether stimulating the SOM or VIP neuron decreases or increases the activity of the V neuron depends on the input statistics: for low-mean stimuli, the elevated baseline activity dominates the changes in the variance, while for high-mean stimuli the changes in the gain dominate.



../results/figures/final/Fig_5.png

Figure 5. Mechanisms underlying the contraction bias.

(A) Illustration of the contraction bias. The estimated input is shifted towards the mean of the input distribution. Hence, a linear curve fitted to the data has a slope below 1 (left). The bias is biggest at the end of the stimulus distribution (right). (B) Contraction bias in the model. Left: Example for two different stimulus variabilities. Right: As a consequence of the sensory weight, the slope decreases with stimulus variability (bias increases) and increases with trial variability (bias decreases). Stimulus statistics: XXX. (C) The bias is independent of trial variability when the stimulus variability is zero (left). Equally, the bias is independent of the stimulus variability when the trial variability is zero (right). Stimulus statistics: XXX. (D) The slope depends on the trial duration. If the sensory weight is 1, the slope is independent of the trial duration and the bias vanishes (dashed-dotted line). If the sensory weight is 0, the slope depends on the trial duration and only reaches 1 if the trial duration approaches infinity. (E) To ensure a larger bias for stimuli drawn from the upper end of the stimulus distribution than from the lower end, scalar variability as observed experimentally is needed.

The contraction bias as a result of the weighted integration of sensory inputs and predictions

The weighted integration of sensory inputs and predictions thereof manifests in all-day behavior, in the form of a phenomenon called *contraction bias*. The contraction bias describes the tendency to overestimate sensory stimuli drawn from the lower end of a stimulus distribution and to underestimate stimuli drawn from the upper end of the same distribution (Fig. 5A). This *bias towards the mean* has been reported in different species and modalities (XXX).

The weighted output of our network can be interpreted as a neural manifestation of the contraction bias (see Methods for a thorough analysis). The bias increases with stimulus variance (Fig. 5B), decreasing the slope of the linear fit modeling the relationship between the true and estimated stimuli

(Fig. 5B, right; compare with Fig. 5A). In contrast, the bias decreases with trial variance, so that the slope of the linear fit approaches 1 (Fig. 5B right).

What are the underlying network factors that contribute to this phenomenon? To disentangle the potentially different sources of the bias, we first simulated a network without stimulus variability (variance set to zero) for two different trial variabilities. In this case, a contraction bias emerges but is independent of the volatility of the environment (Fig. 5C, left). We show mathematically that the bias results from the transient neuron activity before reaching a steady state, and vanishes if the trial duration approaches infinity (see Methods for more details, and Fig. 5D). We next resume the limit case in which the same but high-variance stimulus is shown in every trial. In this case, the weighted output exhibits a contraction bias that is largely independent of the stimulus variance (Fig. 5C, right and Fig. 5D). As shown mathematically (see Methods), the bias results from the finite trial duration and the tendency to weight the prediction more strongly than the sensory inputs.

So far, we assumed that the stimulus variance is independent of the trial mean. A consequence of this choice is that the bias on either end of the stimulus distribution is largely the same (but with reversed signs). However, behavioral (neural?) data (XXX) shows that the bias increases for stimuli drawn from the upper end of the distribution, a phenomenon usually attributed to *scalar variability*. To capture this in the model, we assume that the stimulus standard deviation linearly increases with the trial mean. In these simulations, as expected, the bias increases for a stimulus distribution shifted to higher trial means (Fig. 5E).

In summary, the weighted integration of sensory inputs and predictions can be interpreted as a neural manifestation of the contraction bias. Both stimulus and trial variability contribute to the contraction bias but the underlying mechanisms differ.

Discussion

We solved the brain.

Models and methods

Network model

The mean-field network model consists of a *lower* and *higher* PE circuit (Fig. 1). Each PE circuit contains an excitatory nPE neuron and pPE neuron ($N_{\text{nPE}} = N_{\text{pPE}} = 1$), as well as inhibitory neurons. The inhibitory neurons comprise PV, SOM and VIP neurons ($N_{\text{SOM}} = N_{\text{VIP}} = 1$, $N_{\text{PV}} = 2$). In addition to the core PE circuit, each subnetwork also includes one memory neuron M and one variance neuron V .

The excitatory neurons in the PE circuit are simulated as two coupled point compartments, representing the soma and the dendrites of elongated pyramidal cells. All other neurons are modeled as point neurons. The activities of all neurons are represented by a set of differential equations describing the network dynamics.

The dynamics of the neurons in the lower and higher PE circuits ($r_{\text{PE}}^{\text{low}}$ and $r_{\text{PE}}^{\text{high}}$) are given by

$$\begin{aligned} \dot{r}_{\text{PE}}^{\text{low}} &= \left[\dot{h}_{\text{PE}}^{\text{low}} \right]_+ \\ \dot{r}_{\text{PE}}^{\text{high}} &= \left[\dot{h}_{\text{PE}}^{\text{high}} \right]_+ \end{aligned} \quad (1)$$

with

$$\begin{aligned} T \cdot \dot{h}_{\text{PE}}^{\text{low}} &= -\dot{h}_{\text{PE}}^{\text{low}} + W_{\text{PE} \leftarrow \text{PE}} \cdot r_{\text{PE}}^{\text{low}} + \underline{w}_{\text{PE} \leftarrow \text{M}} \cdot r_{\text{M}}^{\text{low}} + \underline{w}_{\text{PE} \leftarrow \text{FF}} \cdot s + \underline{I}_{\text{PE}} \\ T \cdot \dot{h}_{\text{PE}}^{\text{high}} &= -\dot{h}_{\text{PE}}^{\text{high}} + W_{\text{PE} \leftarrow \text{PE}} \cdot r_{\text{PE}}^{\text{high}} + \underline{w}_{\text{PE} \leftarrow \text{M}} \cdot r_{\text{M}}^{\text{high}} + \underline{w}_{\text{PE} \leftarrow \text{FF}} \cdot r_{\text{M}}^{\text{low}} + \underline{I}_{\text{PE}}. \end{aligned} \quad (2)$$

We follow the notation that column and row vectors are indicated by small letters with an underscore \bullet , matrices are denoted by capital letters, and scalars are given by small letters without an underscore. Furthermore, a time derivative (e.g., $\frac{dx}{dt}$) is denoted by a dot above the letter (e.g., \dot{x}). The rate vector $r_{\text{PE}}^{\text{loc}} = [r_{\text{nE}}^{\text{loc}}, r_{\text{pE}}^{\text{loc}}, r_{\text{nD}}^{\text{loc}}, r_{\text{pD}}^{\text{loc}}, r_{\text{PV}_1}^{\text{loc}}, r_{\text{PV}_2}^{\text{loc}}, r_{\text{SOM}}^{\text{loc}}, r_{\text{VIP}}^{\text{loc}}]$ with $\text{loc} \in [\text{low}, \text{high}]$ contains the activities of all neurons or compartments in the PE circuit (soma of nPE/pPE neurons: nE/pE, dendrites of nPE/pPE neurons: nD/pD). The network receives time-dependent stimuli s and neuron/compartment-specific

external background input \underline{I}_{PE} . The connection strengths between the *pre*-synaptic neuron (population) and the neurons of the PE circuit are denoted by $W_{PE \leftarrow pre}$ and $\underline{w}_{PE \leftarrow pre}$, respectively. The activities of the neurons evolve with time constants summarized in T .

The activities of the lower and higher M neuron evolve according to a perfect integrator. The memory neurons receive synapses from both nPE and pPE neurons of the same subnetwork,

$$\begin{aligned}\dot{r}_M^{\text{low}} &= \underline{w}_{M \leftarrow PE}^{\text{low}} \cdot r_{PE}^{\text{low}} = w_{M \leftarrow pPE}^{\text{low}} \cdot r_{pPE}^{\text{low}} - w_{M \leftarrow nPE}^{\text{low}} \cdot r_{nPE}^{\text{low}} \\ \dot{r}_M^{\text{high}} &= \underline{w}_{M \leftarrow PE}^{\text{high}} \cdot r_{PE}^{\text{high}} = w_{M \leftarrow pPE}^{\text{high}} \cdot r_{pPE}^{\text{high}} - w_{M \leftarrow nPE}^{\text{high}} \cdot r_{nPE}^{\text{high}}.\end{aligned}\quad (3)$$

The activities of the lower and higher V neuron evolve according to a leaky integrator with quadratic activation function. The variance neurons receive synapses from both nPE and pPE neurons of the same subnetwork,

$$\begin{aligned}\tau_V^{\text{low}} \cdot \dot{r}_V^{\text{low}} &= -r_V^{\text{low}} + \left(\underline{w}_{V \leftarrow PE} \cdot r_{PE}^{\text{low}}\right)^2 = -r_V^{\text{low}} + \left(w_{V \leftarrow pPE} \cdot r_{pPE}^{\text{low}} + w_{V \leftarrow nPE} \cdot r_{nPE}^{\text{low}}\right)^2 \\ \tau_V^{\text{high}} \cdot \dot{r}_V^{\text{high}} &= -r_V^{\text{high}} + \left(\underline{w}_{V \leftarrow PE} \cdot r_{PE}^{\text{high}}\right)^2 = -r_V^{\text{high}} + \left(w_{V \leftarrow pPE} \cdot r_{pPE}^{\text{high}} + w_{V \leftarrow nPE} \cdot r_{nPE}^{\text{high}}\right)^2.\end{aligned}\quad (4)$$

All values for neuron and network parameters, details on the model equations, as well as supporting analyses can be found in the supplementary material.

Weighting of sensory inputs and predictions

We arithmetically calculated the weighted output of sensory inputs and predictions, r_{out} , based on Bayesian multisensory integration (XXX),

$$r_{\text{out}} = \alpha \cdot s + (1 - \alpha) \cdot r_M^{\text{low}}, \quad (5)$$

where α denotes the sensory weight (that is, the reliability of the sensory input) and is given by

$$\alpha = \left(1 + \frac{r_V^{\text{low}}}{r_V^{\text{high}}}\right)^{-1}. \quad (6)$$

Inputs

The network receives feedforward stimuli s that may vary between trials. To account for noise, each stimulus is composed of N_{in} piece-wise constant values drawn from a normal distribution with mean μ_{in} and standard deviation σ_{in} . To account for changes in the environment, μ_{in} is drawn from a uniform distribution $U(a, b)$ with mean $\mu_{\text{trial}} = \frac{a+b}{2}$ and standard deviation $\sigma_{\text{trial}} = \frac{b-a}{\sqrt{12}}$. The parameterisation of both distributions varies across the experiments. All stimulus/input parameters can be found in the supplementary material.

Simulations

All simulations were performed in customized Python code written by LH. Source code to reproduce the simulations, analyses and figures will be available after publication at XXXX. Differential equations were numerically integrated using a 2nd-order Runge-Kutta method. Neurons were initialized with $r = 0/s$. Further details and values for simulation parameters can be found in the supplementary material.

Acknowledgments

References

Supporting Information

A Detailed Methods	11
A.1 Network model	11
A.1.1 Prediction-error network model	11
A.1.2 Memory and variance neuron	12
A.1.3 Weighted output	13
A.2 Connectivity	13
A.2.1 Connections between neurons of the PE circuit	13
A.2.2 Connections between the PE circuit and the M neuron	13
A.2.3 Connections between the PE circuit and the V neuron	14
A.3 Inputs	14
A.4 Simulations	14
B Supporting analyses	16
B.1 Activity of M and V neuron in a simplified model	16
B.2 Impact of PE neurons' gain on estimating mean and variance	16
B.3 Impact of PE neurons' baseline on estimating mean and variance	17
B.4 Modelling the impact of neuromodulators on the sensory weight	18
B.5 Sensory weight and contraction bias	18
B.6 Comparison to Kalman filter and Bayes Factor surprise	19
C Supplementary Figures	19

A Detailed Methods

In the following, we describe in more detail the equations describing the dynamics of the neurons in the prediction-error circuit, as well as the memory and variance neurons. We then provide the connectivity of the network and the inputs to the neurons for both the mean-field and population model. Finally, to ensure reproducibility, we give all simulation parameters used for the results shown in the figures.

XXX Please note that xxx model parameters chosen do not change overall results are not particular to get the results xxx. Important factors are: 1) long enough to get to steady state (as PE circuits used were learned for steady state, 2) lower M updates faster than higher M, 3) update speed for M and V (see lambda and theta etc.) must be chosen in accordance with trial duration stimulus duration. XXX

A.1 Network model

The network model consists of a *lower* and *higher* mean-field PE circuit (Fig. 1). Each PE circuit contains an excitatory nPE neuron and pPE neuron ($N_{\text{nPE}} = N_{\text{pPE}} = 1$), as well as inhibitory neurons. The inhibitory neurons comprise PV, SOM and VIP neurons ($N_{\text{SOM}} = N_{\text{VIP}} = 1$, $N_{\text{PV}} = 2$). In addition to the core PE circuit, each subnetwork also includes one memory neuron M and one variance neuron V .

In Figure 2 and the corresponding supporting figures, only the lower subnetwork is simulated. In Fig. ??, we replaced this lower mean-field PE circuit with a heterogeneous population model containing 200 neurons ($N_{\text{SOM}} = N_{\text{VIP}} = N_{\text{PV}} = 20$, 140 excitatory neurons). In Fig. ??, the lower PE circuit comprises 1000 copies of the mean-field network to account for selectivity.

In the following, we describe the dynamics of the neurons/compartments in the mean-field network. The equations for the population PE circuit (Fig. ??) are directly deduced from the mean-field equations and can also be found in XXXref your own paperXXX.

A.1.1 Prediction-error network model

Each excitatory pyramidal cell (that is, nPE or pPE neuron) is divided into two coupled compartments, representing the soma and the dendrites, respectively. The dynamics of the firing rates of the somatic

compartments r_{nE} (nPE neuron) and r_{pE} (pPE neuron) obey (?)

$$\begin{aligned} r_{nE} &= [h_{nE}]_+ \quad \text{with} \quad \tau_E \frac{dh_{nE}}{dt} = -h_{nE} + w_{nE \leftarrow nD} \cdot r_{nD} - w_{nE \leftarrow PV_1} \cdot r_{PV_1} - w_{nE \leftarrow PV_2} \cdot r_{PV_2} + I_{nE}, \\ r_{pE} &= [h_{pE}]_+ \quad \text{with} \quad \tau_E \frac{dh_{pE}}{dt} = -h_{pE} + w_{pE \leftarrow pD} \cdot r_{pD} - w_{pE \leftarrow PV_1} \cdot r_{PV_1} - w_{pE \leftarrow PV_2} \cdot r_{PV_2} + I_{pE} \end{aligned} \quad (7)$$

where τ_E denotes the excitatory rate time constant ($\tau_E=60$ ms), the weights $w_{nE \leftarrow nD}$ and $w_{pE \leftarrow pD}$ describe the connection strength between the dendritic compartment and the soma of the same neuron, and $w_{nE \leftarrow PV_1}$, $w_{nE \leftarrow PV_2}$, $w_{pE \leftarrow PV_1}$ and $w_{pE \leftarrow PV_2}$ denote the strength of somatic inhibition from PV neurons. The overall input I_{nE} and I_{pE} comprise the external background and feedforward inputs (see “Inputs” below). Firing rates are rectified to ensure positivity ($[\bullet]_+$).

The dynamics of the activity of the dendritic compartments r_{nD} (nPE neuron) and r_{pD} (pPE neuron) obey (?)

$$\begin{aligned} r_{nD} &= [h_{nD}]_+ \quad \text{with} \quad \tau_E \frac{dh_{nD}}{dt} = -h_{nD} + w_{nD \leftarrow nE} \cdot r_{nE} + w_{nD \leftarrow pE} \cdot r_{pE} + w_{nD \leftarrow M} \cdot r_M \\ &\quad - w_{nD \leftarrow SOM} \cdot r_{SOM} + I_{nD}, \\ r_{pD} &= [h_{pD}]_+ \quad \text{with} \quad \tau_E \frac{dh_{pD}}{dt} = -h_{pD} + w_{pD \leftarrow nE} \cdot r_{nE} + w_{pD \leftarrow pE} \cdot r_{pE} + w_{pD \leftarrow M} \cdot r_M \\ &\quad - w_{pD \leftarrow SOM} \cdot r_{SOM} + I_{pD}, \end{aligned} \quad (8)$$

where the weights $w_{nD \leftarrow nE}$, $w_{nD \leftarrow pE}$, $w_{pD \leftarrow nE}$ and $w_{pD \leftarrow pE}$ denote the recurrent excitatory connections between PCs, including backpropagating activity from the soma to the dendrites. $w_{nD \leftarrow SOM}$ and $w_{pD \leftarrow SOM}$ represent the strength of dendritic inhibition from the SOM neuron. $w_{nD \leftarrow M}$ and $w_{pD \leftarrow M}$ denote the strength of connection between the memory neuron and the dendrites. The overall inputs I_{nD} and I_{pD} comprise fixed, external background inputs (see “Inputs” below). We assume that any excess of inhibition in a dendrite does not affect the soma, that is, the dendritic compartment is rectified at zero.

Similarly, the firing rate dynamics of each interneuron is modeled by a rectified, linear differential equation,

$$\begin{aligned} r_X &= [h_X]_+ \quad \text{with} \quad \tau_I \frac{dh_X}{dt} = -h_X + I_X + w_{X \leftarrow nE} \cdot r_{nE} + w_{X \leftarrow pE} \cdot r_{pE} + w_{X \leftarrow M} \cdot r_M - w_{X \leftarrow PV_1} \cdot r_{PV_1} \\ &\quad - w_{X \leftarrow PV_2} \cdot r_{PV_2} - w_{X \leftarrow SOM} \cdot r_{SOM} - w_{X \leftarrow VIP} \cdot r_{VIP}, \end{aligned} \quad (9)$$

where r_X denotes the firing rate of interneuron type X , and the weight $w_{X \leftarrow Y}$ denotes the strength of connection between the presynaptic neuron Y and the postsynaptic neuron X ($X \in \{PV_1, PV_2, SOM, VIP\}$, $Y \in \{nPE, pPE, PV_1, PV_2, SOM, VIP, M\}$). The rate time constant τ_I was chosen to resemble a fast GABA_A time constant, and set to 2 ms for all interneuron types included. The overall input I_X comprises fixed, external background inputs and feedforward sensory inputs (see “Inputs” below).

A.1.2 Memory and variance neuron

In addition to the core PE circuit, we simulate a memory neuron M and a variance neuron V . The memory neuron is modeled as a perfect integrator, receiving synapses from both the nPE and pPE neuron,

$$\tau_E \cdot \frac{dr_M}{dt} = w_{M \leftarrow pE} \cdot r_{pE} - w_{M \leftarrow nE} \cdot r_{nE}. \quad (10)$$

$w_{M \leftarrow pE}$ denotes the connection strength between the pPE neuron and the memory neuron, and $w_{M \leftarrow nE}$ denotes the connection strength between the nPE neuron and the memory neuron. The time constant $\tau_E = 60$ ms.

The dynamics of the variance neuron obeys a non-linear differential equation with leak term,

$$\tau_V \cdot \frac{dr_V}{dt} = -r_V + (w_{V \leftarrow pE} \cdot r_{pE} + w_{V \leftarrow nE} \cdot r_{nE})^2. \quad (11)$$

The weight $w_{V \leftarrow pE}$ represents the connection strength between the pPE neuron and the variance neuron, while $w_{V \leftarrow nE}$ denotes the connection strength between the nPE neuron and the variance neuron. To ensure that the V neuron encodes the variance, we chose a quadratic activation function. In Fig. S3, we used a linear activation function to investigate the impact of the input-output transfer function on the weighting of sensory inputs and predictions. The time constant $\tau_V = 5$ s.

A.1.3 Weighted output

The weighted output $r_{\text{out}}(t)$ is a linear combination of the current sensory input $s(t)$ and the activity of the memory neuron, $r_{\text{M}}(t)$, inspired by Bayesian multisensory integration (XXX)

$$r_{\text{out}}(t) = \alpha \cdot s(t) + (1 - \alpha) \cdot r_{\text{M}}(t). \quad (12)$$

How strongly either the sensory input or the prediction thereof contributes to the output is denoted by the sensory weight α ,

$$\begin{aligned} \alpha &= \frac{r_{\text{V}_{\text{lower}}}^{-1}}{r_{\text{V}_{\text{lower}}}^{-1} + r_{\text{V}_{\text{higher}}}^{-1}} \\ &= \left(1 + \frac{r_{\text{V}_{\text{lower}}}}{r_{\text{V}_{\text{higher}}}} \right)^{-1}. \end{aligned} \quad (13)$$

A.2 Connectivity

A.2.1 Connections between neurons of the PE circuit

The connectivity between neurons of the PE circuit, both for the mean-field and the population network, was taken from XXX. XXX ADD heatmaps for the connectivity? or simply refer to the 2022 paper (maybe rather that)

A.2.2 Connections between the PE circuit and the M neuron

While the nPE neurons inhibit the M neuron, the pPE neurons excite it. To ensure that the activities of the memory neurons represent the mean of the sensory stimuli in the lower PE circuit and the mean of the prediction in the higher subnetwork, respectively, the net effect of nPE and pPE neurons must cancel in the steady state (see Analysis in B.2). Hence, the weights need to account for the neurons' potentially different gain factors (g_{nPE} and g_{pPE} , see Table XXX) and the neuron numbers (N_{nPE} and N_{pPE} , see Table XXX):

$$\begin{aligned} w_{\text{M} \leftarrow \text{nPE}} &= \frac{-\lambda}{g_{\text{nPE}} \cdot N_{\text{nPE}}} \\ w_{\text{M} \leftarrow \text{pPE}} &= \frac{\lambda}{g_{\text{pPE}} \cdot N_{\text{pPE}}} \end{aligned}$$

where λ denotes the speed at which the perfect integrator evolves (lower PE circuit: $\lambda = 3 \cdot 10^{-3}$ in Fig. 2 and $\lambda = 4.5 \cdot 10^{-2}$ otherwise, higher PE circuit: $\lambda = 7 \cdot 10^{-4}$).

XXX Here you must say that this is true for the MFN and the one in which you copied the MFN but in case of the population, each neuron has a different gain, lambda is xxx for allXXX not sure how to show all the different gain factors? Also, you must say that the weights were systematically changed in Supp Fig. XXX to investigate correlated and uncorrelated changes XXX

The memory neuron M connects to the post-synaptic neurons X in the PE circuit with the connection strength $w_{X \leftarrow M}$. If a connection exists, we assume $w_{X \leftarrow M} = 1$. In all mean-field networks and the population network, the dendrites of nPE and pPE neurons and one of the two (populations of) PV neurons receive connections from the memory neuron. We considered three mean-field networks that differed in terms of the inputs onto the SOM and VIP neurons:

Network name	MFN 1	MFN 2	MFN 3
SOM	0	1	0
VIP	1	0	0

Table 1. XXX Table shows $w_{X \leftarrow M}$

In the population model, 30% of the SOM neurons and 70% of the VIP neurons receive input from the memory neuron.

A.2.3 Connections between the PE circuit and the V neuron

Both nPE and pPE neurons excite the V neuron. To ensure that the activity of the V neuron represents the variance of the input (see Analysis in B.2), the weights must account for differences in the gains (g_{nPE} and g_{pPE} , see Table XXX) and numbers (N_{nPE} and N_{pPE}) of the PE neurons,

$$w_{V \leftarrow \text{nPE}} = \frac{\theta}{g_{\text{nPE}} \cdot N_{\text{nPE}}}$$

$$w_{V \leftarrow \text{pPE}} = \frac{\theta}{g_{\text{pPE}} \cdot N_{\text{pPE}}}.$$

θ XXX represents a factor that ensures that for population network V still represents variance, no cross -terms, in mean-field network it is 1, XXX check if for population network, it is ok to ignore the $\text{nPE}_1 * \text{nPE}_2$ terms, I mean you have cross-terms that do something with the variance, do they XXX

We assume a quadratic activation function for the V neuron. In case of a mean-field network, the choice of the activation function ensures that the activity of the V neuron represents the variance of the input. This is only true because nPE and pPE neurons are not active at the same time and have a baseline activity of zero. \Rightarrow cross-terms are zero. However, if we increase the number of nPE and pPE neurons, cross-terms of the nature $\text{nPE}_1 * \text{nPE}_2$ will introduce deviations from the the variance. These effects can be mitigated or fully corrected by choosing θ accordingly. θ depends on the distribution at hand. In our simulations, we only considered a uniform distribution.

$$\begin{aligned} \left(\frac{\theta}{N} \sum_i |s_i - \mu| \right)^2 &= \left(\frac{\theta}{N} \sum_{s_i \geq \mu}^{N/2} (s_i - \mu) + \frac{\theta}{N} \sum_{s_i \leq \mu}^{N/2} (\mu - s_i) \right)^2 \\ &= \frac{\theta^2}{4} \left(\frac{2}{N} \sum_{s_i \geq \mu}^{N/2} s_i - \frac{2}{N} \sum_{s_i \leq \mu}^{N/2} s_i \right)^2 \\ &= \frac{\theta^2}{4} \left(\frac{b + \mu}{2} - \frac{\mu + a}{2} \right)^2 = \frac{(b - a)^2}{16} \cdot \theta^2 \end{aligned} \quad (14)$$

Hence, to ensure the variance of a uniform distribution given by $\frac{(b-a)^2}{12}$, $\theta = \frac{2}{\sqrt{3}}$.

A.3 Inputs

Each neuron (type) receives an overall input I_i ,

$$I_i = I_i^{BL} + w_i \cdot I_i^{FF}$$

where I_i^{FF} denotes a feedforward input and I_i^{BL} represents an external background input that ensures reasonable baseline firing rates in the absence of sensory inputs and predictions thereof. In the case of the mean-field network, these inputs were set such that the baseline firing rates are $r_{\text{pE}} = r_{\text{pD}} = r_{\text{nE}} = r_{\text{nD}} = 0 \text{ s}^{-1}$ and $r_{\text{P}} = r_{\text{S}} = r_{\text{V}} = 4 \text{ s}^{-1}$. In the case of the population network, we set the external inputs of all neuron types to 5 s^{-1} , while the background inputs to the dendrites were computed such that the dendrites are inactive during baseline.

The feedforward input is either the direct sensory input s for the lower PE circuit, or the activity of the M neuron r_{M} for the higher PE circuit. In general, for the three mean-field networks tested, we chose $w_i = 1 - w_{\text{X} \leftarrow \text{M}}$ (see Table 1). In the population network, 70% of the SOM neurons and 30% of the VIP neurons receive the feedforward input.

A.4 Simulations

All simulations were performed in customized Python code written by LH. Source code to reproduce the simulations, analyses and figures will be available after publication at XXXX. Differential equations were numerically integrated using a 2nd-order Runge-Kutta method. Neurons were initialized with $r = 0/s$.

In the following, we give all figure-specific parameters not directly visible or mentioned in the Figures and captions. Please note that the qualitative results are robust to the choice of the simulation parameters and are here stated merely to ensure the reproducibility of all figures. Furthermore, to

increase readability, we do not include units for the parameters. All units can be deduced from the equations above.

Figure 1: Simulation time steps = 10^5 , constant prediction (fixed) = 5, input mean $\in [0, 10]$, input standard deviation 0.

Figure 2: Simulation time steps = 10^5 , 200 constant values (each 500 time steps long) drawn from a uniform distribution, B and C: input standard deviation fixed at 5 when mean is varied, input mean fixed at 4.5 when input variance is varied.

Figure 2 Supplementary Fig. 1: Simulation time steps = 10^5 , 200 constant values (each 500 time steps long) drawn from a uniform distribution with mean of 5 and variance of 4. Number of repetitions with different seeds: 20.

Figure 2 Supplementary Fig. 2: Number of time steps were 10^5 (C-E) or $2 \cdot 10^5$ (B) with 400 (C-E) or 200 (B) constant values drawn from a uniform distribution with mean of 5 and variance of 4. Time step was 0.1. The connections from the PE neurons to the M or V neuron were altered by a factor γ drawn from a normal distribution. If not stated otherwise (in the Figure), the mean of this normal distribution was 1 and the variance 0, while the connection probability was 1. Number of repetitions with different seeds: 10.

Figure 2 Supplementary Fig. 3: Number of identical mean-field networks: 1000. Simulation time steps = 10^5 .

Figure 3: 100 trials, each simulated for 5000 time steps. In a trail, 10 constant values were drawn from a normal distribution (each 500 time steps long). The variance of this distribution was 0 (C, F) and 5.3 XXX (D). In each trial, the stimulus mean was drawn from an uniform distribution $U(a, b)$ (C: $U(1, 9)$, D: $U(5, 5)$). In F, U was parameterized such that the trail variability was 3.

Figure 3 Supplementary Fig. 1: 120 trials, each simulated for 5000 time steps. In a trail, 10 constant values were drawn from a normal distribution $N(\mu_{in}, \sigma_{in})$, each 500 time steps long. In each trial, the stimulus mean was drawn from an uniform distribution $U(a, b)$. Switch of input statistics occurs after 60 trails. State 1: stimulus $\in N(\mu_{in}, 0)$ with $\mu_{in} \in U(5, 5)$, State 2: stimulus $\in N(\mu_{in}, 3)$ with $\mu_{in} \in U(5, 5)$, State 3: stimulus $\in N(\mu_{in}, 3)$ with $\mu_{in} \in U(0, 10)$, State 4: stimulus $\in N(\mu_{in}, 0)$ with $\mu_{in} \in U(0, 10)$.

Figure 3 Supplementary Fig. 2: 200 trials, each simulated for 5000 time steps. In a trail, 10 constant values were drawn from a normal distribution with variance σ_{in}^2 , each 500 time steps long. In each trial, the stimulus mean was drawn from an uniform distribution with mean of 5 and a variance of σ_{trial}^2 . $\sigma_{trial}^2 \in \{0, 0.75, 1.5, 2.25, 3\}$, $\sigma_{in}^2 \in \{3, 2.25, 1.5, 0.75, 0\}$. A: scaling factors of $w_{M \leftarrow PE}$ were 0.3 and 6.7 XXX.

Figure 4: 100 – 200 trials, each simulated for 5000 time steps. In a trail, 10 constant values were drawn from a normal distribution with variance σ_{in}^2 , each 500 time steps long. In each trial, the stimulus mean was drawn from an uniform distribution with mean of 5 and a variance of σ_{trial}^2 . A, top: $\sigma_{in}^2 = 0$ and $\sigma_{trial}^2 = 1$. A, bottom: $\sigma_{in}^2 = 1$ and $\sigma_{trial}^2 = 0$. C: $\sigma_{in}^2 = 1$ and $\sigma_{trial}^2 = 1$. D: $\sigma_{in}^2 = 5.3XXX$ and $\sigma_{trial}^2 = 0$. Additional input (perturbation) was either fixed at 1 (A, D) or systematically varied between -1 and 1 (C). To estimate changes in baseline and gain of nPE and pPE neurons, we fitted a linear function to the PE neuron activity for the input range $[0, 2.5]$.

Figure 4 Supplementary Fig. 1: All parameters are from Figure 4 A.

Figure 4 Supplementary Fig. 2: In the default setting baseline was 0 and gain of PE neurons was 1 (see Eqs. XXX). The results (left) were computed for baselines $\in [0, 3]$, while the results (right) were computed for gains $\in [0.5, 1.5]$. XXX Here refer to equations XXX.

Figure 4 Supplementary Fig. 3: All parameters are from Figure 4 D.

Figure 5: 200 trials, each simulated for 5000 time steps (except in panel E). In a trail, 10 constant values were drawn from a normal distribution (each 500 time steps long) with standard deviation σ_{in} . In each trial, the stimulus mean μ_{in} was drawn from an uniform distribution $U(a, b)$. A: $\sigma_{in} \in \{1, 7\}$, $\mu_{in} \in U(15, 25)$. B: $\sigma_{in} \in [0, 8]$, $\mu_{in} \in U(15, 25)$, and $\sigma_{in} = 5$, $\mu_{in} \in U(15, b)$ with $b \in [20, 48]$. C: $\sigma_{in} \in \{2, 5\}$, $\mu_{in} \in U(15, 15)$. D: $\sigma_{in} = 0$, $\mu_{in} \in U(15, 25)$ or $U(10, 30)$. E: $\sigma_{in} = 0$, $\mu_{in} \in U(15, 25)$, or $\sigma_{in} = 5$, $\mu_{in} \in U(15, 15)$, Time steps per trail increased from 5000 to 10^4 .

Figure 5 Supplementary Fig. 1: 200 trials, each simulated for 5000 time steps. In a trail, 10 constant values were drawn from a normal distribution (each 500 time steps long) with standard deviation σ_{in} . In each trial, the stimulus mean μ_{in} was drawn from an uniform distribution $U(a, b)$. We used two different uniform distributions $U(15, 25)$ and $U(25, 35)$, and introduced scalar variability so that σ_{in} is a linear function of μ_{in} . Specifically, we chose $\sigma_{in} = \mu_{in} - 14$.

B Supporting analyses

We first describe a simplified model and show that the M neuron represents the mean, while the V neuron represents the variance of the feedforward input. We then investigate the impact of the gain and baseline of PE neurons on estimating the mean and variance. Furthermore, we use the simplified model to discuss the effect of neuromodulators in our network. Finally, we reveal the connection between the sensory weight and the contraction bias, and compare our network with the Kalman filter as well as Bayes factor surprise. XXX

B.1 Activity of M and V neuron in a simplified model

To show that the M neuron encodes the mean, while the V neuron encodes the variance of the feedforward input, we resume a toy model in which the activity of the nPE and pPE neuron is replaced by its ideal output

$$\begin{aligned} r_{\text{nE}} &= [r_{\text{M}} - s_{\text{FF}}]_+ \\ r_{\text{pE}} &= [s_{\text{FF}} - r_{\text{M}}]_+ \end{aligned} \quad (15)$$

with s_{FF} denoting the time-dependent feedforward input. The activity of the M neuron can then be described as

$$\tau_{\text{M}} \cdot \frac{dr_{\text{M}}}{dt} = r_{\text{pE}} - r_{\text{nE}} \quad (16)$$

If $r_{\text{M}} \geq s_{\text{FF}}$, we get

$$\tau_{\text{M}} \cdot \frac{dr_{\text{M}}}{dt} = -r_{\text{nE}} = -r_{\text{M}} + s_{\text{FF}}. \quad (17)$$

If $r_{\text{M}} \leq s_{\text{FF}}$, we also get

$$\tau_{\text{M}} \cdot \frac{dr_{\text{M}}}{dt} = r_{\text{pE}} = -r_{\text{M}} + s_{\text{FF}}.$$

Hence, the activity of r_{M} is given by

$$r_{\text{M}} = \frac{1}{\tau_{\text{M}}} \int_0^t e^{-(t-x)/\tau_{\text{M}}} \cdot s_{\text{FF}}(x) dx \quad (18)$$

for zero activity at time $t = 0$. In the limit of $t \rightarrow \infty$ (steady state), this is exactly the exponential average of the feedforward input, $E(s_{\text{FF}})$.

With the simplified activity of the nPE and pPE neuron, the activity of the V neuron can then be described as

$$\tau_{\text{V}} \cdot \frac{dr_{\text{V}}}{dt} = -r_{\text{V}} + (r_{\text{pE}} + r_{\text{nE}})^2 = -r_{\text{V}} + (r_{\text{M}} - s_{\text{FF}})^2, \quad (19)$$

leading to the time-dependent solution

$$r_{\text{V}} = \frac{1}{\tau_{\text{V}}} \int_0^t e^{-(t-x)/\tau_{\text{V}}} \cdot [r_{\text{M}}(x) - s_{\text{FF}}(x)]^2 dx. \quad (20)$$

In the limit of $t \rightarrow \infty$, r_{V} approaches $E(s_{\text{FF}} - E(s_{\text{FF}}))^2$.

B.2 Impact of PE neurons' gain on estimating mean and variance

The gains of the PE neurons, if not equal between nPE and pPE neuron on average, can bias the activity of both the M and V neuron. To show this, we resume the toy model from section B.1.

$$\begin{aligned} g_{\text{pPE}} \langle r_{\text{nPE}} \rangle &= g_{\text{nPE}} \langle r_{\text{pPE}} \rangle \\ g_{\text{pPE}} \langle [s_{\text{FF}} - P]_+ \rangle &= g_{\text{nPE}} \langle [P - s_{\text{FF}}]_+ \rangle \\ g_{\text{pPE}} \int_P^\infty (x - P) f(x) dx &= g_{\text{nPE}} \int_{-\infty}^P (P - x) f(x) dx. \end{aligned} \quad (21)$$

Here, P denotes the prediction encoded in the M neuron, and $f(x)$ is the distribution of feedforward inputs. In case of a uniform distribution, $f(x) = 1/(b-a)$ for $x \in [a, b]$ and 0 otherwise, we get

$$P = \begin{cases} \frac{a+b}{2} & \text{if } g_{\text{nPE}} = g_{\text{pPE}} = g \\ \frac{g_{\text{pPE}} \cdot b - g_{\text{nPE}} \cdot a \pm \sqrt{g_{\text{nPE}} g_{\text{pPE}} (a-b)}}{g_{\text{pPE}} - g_{\text{nPE}}} & \text{otherwise.} \end{cases} \quad (22)$$

XXX Maybe mention the SuppFig. in which you show the function XXX and explain what you see for gnPE \neq gpPE or vice versa ...

Similarly, the variance is affected by the gain of the nPE and pPE neuron,

$$V = g_{\text{pPE}} \langle [s_{\text{FF}} - P]_+^2 \rangle + g_{\text{nPE}} \langle [P - s_{\text{FF}}]_+^2 \rangle. \quad (23)$$

In case of a uniform distribution, we get

$$\begin{aligned} V &= \frac{g_{\text{pPE}}}{b-a} \int_P^b (x-P)^2 dx + \frac{g_{\text{nPE}}}{b-a} \int_a^P (P-x)^2 dx \\ &= \frac{g_{\text{pPE}}}{3} \cdot \frac{(b-P)^3}{b-a} + \frac{g_{\text{nPE}}}{3} \cdot \frac{(P-a)^3}{b-a}. \end{aligned} \quad (24)$$

Inserting eqs. (22) yields

$$V = \begin{cases} \frac{(b-a)^2}{12} & \text{if } g_{\text{nPE}} = g_{\text{pPE}} = 1 \\ \frac{(b-a)^2}{3 (g_{\text{pPE}} - g_{\text{nPE}})^3} \cdot [g_{\text{nPE}} \cdot (g_{\text{pPE}} \mp \gamma)^3 - g_{\text{pPE}} \cdot (g_{\text{nPE}} \mp \gamma)^3] & \text{otherwise.} \end{cases} \quad (25)$$

with $\gamma = \sqrt{g_{\text{nPE}} g_{\text{pPE}}}$. XXX As above, discuss cases from the Supp Figs. XXX Important: for P and V please check with sign is correct!

B.3 Impact of PE neurons' baseline on estimating mean and variance

The baselines of the PE neurons, if not equal between nPE and pPE neuron on average, can bias the activity of both the M and V neuron. By means of the toy model from section B.1, we can write

$$\begin{aligned} \langle r_{\text{pPE}} \rangle &= \langle r_{\text{nPE}} \rangle \\ \langle [s_{\text{FF}} - P]_+ + p_0 \rangle &= \langle [P - s_{\text{FF}}]_+ + n_0 \rangle \\ \int_P^\infty (x-P) f(x) dx + p_0 \underbrace{\int_a^b f(x) dx}_{=1} &= \int_{-\infty}^P (P-x) f(x) dx + n_0 \underbrace{\int_a^b f(x) dx}_{=1}. \end{aligned} \quad (26)$$

n_0 and p_0 denote the baseline activity of the nPE and pPE neuron, respectively. In case of a uniform distribution (c.f. B.2), we get

$$P = \frac{b+a}{2} + \frac{p_0 - n_0}{b-a}. \quad (27)$$

Thus, the M neuron encodes the true mean of the feedforward input only if $p_0 = n_0$. Otherwise, XXX refer to Suppfig and state briefly the cases n \neq p or n

XXX.

With non-zero baseline activities, the steady state activity of the V neuron is given by

$$\begin{aligned} V &= \langle (r_{\text{pPE}} + r_{\text{nPE}})^2 \rangle \\ &= \langle [s_{\text{FF}} - P]_+^2 \rangle + \langle [P - s_{\text{FF}}]_+^2 \rangle + (p_0 + n_0)^2 + 2 (p_0 + n_0) (\langle [s_{\text{FF}} - P]_+ \rangle + \langle [P - s_{\text{FF}}]_+ \rangle) \end{aligned} \quad (28)$$

In case of a uniform distribution, this expression yields

$$V = \frac{1}{3 (b-a)} [(b-P)^3 + (P-a)^3] + (p_0 + n_0)^2 + \frac{(p_0 + n_0)}{b-a} [(b-P)^2 + (a-P)^2]. \quad (29)$$

Inserting the expression for P (Eg. 27) which is itself a function of the baseline activities, gives

$$V = \frac{(b-a)^2}{12} + \frac{(p_0 - n_0)^2}{(b-a)^2} \left(1 + 2 \frac{p_0 + n_0}{b-a} \right) + (p_0 + n_0) \left(p_0 + n_0 + \frac{b-a}{2} \right). \quad (30)$$

Thus, for the V neuron to encode the variance unbiased, $n_0 = p_0 = 0$. If XXX discuss what you see in the supplementary figures

B.4 Modelling the impact of neuromodulators on the sensory weight

- also note here that while prediction and variance might be biased, no necessarily lead to changes in sensory weight ... explain why

Assume that neuromodulator act through activating INs. INs on the other hand modulate both gain and BL of nPE and pPE neurons Changes in gain and BL will affect the overall activity and the balance between nPE and pPE neurons =, this will affect both the prediction and the variance.

Modulate PE in lower: variance in lower is combination of direct changes in PE and indirect through prediction (that is changed through PE mod) variance in higher is just because of differences in prediction

Modulation in higher: variance in lower should be unaffected variance in higher is combination of direct changes in PE and indirect through prediction (that is changed through PE mod)

Variance as function of bias in mean:

$$\begin{aligned}
 V &= \frac{1}{n} \sum_i (x_i - (\mu \pm \delta\mu))^2 \\
 &= \frac{1}{n} \sum_i \{(x_i - \mu)^2 + \delta\mu^2 \mp 2\delta\mu(x_i - \mu)\} \\
 &= V_{\text{unmod}} + \delta\mu^2 \mp 2\delta\mu \left(\frac{1}{n} \sum_i x_i - \mu \right) \\
 &= V_{\text{unmod}} + \delta\mu^2
 \end{aligned}$$

B.5 Sensory weight and contraction bias

The contraction bias can be formalized through the slope in a linear equation describing the relationship between the estimated and true sensory stimulus. If the slope is close to 1, the bias is small. In contrast, if the slope is close to zero, the bias is large.

If the prediction encoded in the activity of the M neuron changes very slowly, the bias is well-captured by the sensory weight,

$$r_{\text{out}} = \alpha_S \cdot s_{\text{FF}} + \text{constant} \quad (31)$$

However, the prediction is normally a function of the history of sensory stimuli the network received, and cannot be approximated by a constant. To investigate how the slope (describing the contraction bias) depends on the sensory weight and other factors, let us resume an approximation in which the prediction decays exponentially to a presented constant stimulus value, s ,

$$P = P_0 \cdot e^{-t/\tau_M} + s \cdot (1 - e^{-t/\tau}) \quad (32)$$

with P_0 describing the initial value at time $t = 0$. Let us assume that within a trial with trial duration T , the stimulus value changes n times ($T = n \cdot \Delta t$). The prediction during the presentation of the n th stimulus value can be expressed as

$$P_n = P_0 \cdot e^{-\Delta t/\tau_M} + \left(1 - e^{-\Delta t/\tau_M}\right) \sum_{i=1}^n s_i \cdot e^{-(n-i)\Delta t/\tau_M} \quad (33)$$

To obtain an estimate for the prediction at the end of a trial, P_n must be averaged over the stimulus distribution, $\langle P_n \rangle_s$. For the sake of simplicity, let us assume the stimulus values are drawn from a uniform distribution $U(s - \frac{\sigma_S}{12}, s + \frac{\sigma_S}{12})$. Moreover, we assume that the initial state, P_0 , at the beginning of a new trial is drawn from a uniform distribution $U(\mu - \frac{\sigma_P}{12}, \mu + \frac{\sigma_P}{12})$. With these assumptions, $\langle P_n \rangle_s$ is given by

$$\langle P_n \rangle_s = e^{-\Delta t/\tau_M} \int_{\mu - \frac{\sigma_P}{12}}^{\mu + \frac{\sigma_P}{12}} P_0 f(P_0) dP_0 + \left(1 - e^{-\Delta t/\tau_M}\right) \sum_{i=1}^n e^{-(n-i)\Delta t/\tau_M} \int_{s - \frac{\sigma_S}{12}}^{s + \frac{\sigma_S}{12}} x f(x) dx \quad (34)$$

Solving the integrals yield

$$\langle P_n \rangle_s = \mu \cdot e^{-T/\tau_M} + \left(1 - e^{-\Delta t/\tau_M}\right) \sum_{i=1}^n e^{-(n-i)\Delta t/\tau_M} \cdot s \quad (35)$$

Making use of the geometric series, the expression simplifies to

$$\langle P_n \rangle_s = \mu \cdot e^{-T/\tau_M} + \left(1 - e^{-T/\tau_M}\right) \cdot s$$

Inserting the expression in the equation for the weighted output yields

$$r_{\text{out}} = \left[\alpha_S e^{-T/\tau_M} + \left(1 - e^{-T/\tau_M}\right) \right] \cdot s + (1 - \alpha_S) e^{-T/\tau_M} \mu$$

Hence, the slope is a function of both the sensory weight and the trial duration.

XXX check the derivation above and then polish the below XXX Also, it might be important to state that alpha itself is a function of the trial duration! xx In a prediction-driven input regime ($\alpha_S \sim 0$), the slope is independent of the sensory weight and only determined by the trial duration, $m \sim (1 - e^{-T/\tau})$. In a sensory-driven input regime ($\alpha_S \sim 1$), the contraction bias vanishes ($m \sim 1$).

If the trail duration is short ($T \rightarrow 0$), the slope is given by the sensory weight. If the trail duration approaches infinity, the slope would be 1 again (however, this seems rather unrealistic, this would only be true in an ideal system without memory decay or reproduction and accumulation noise ...).

B.6 Comparison to Kalman filter and Bayes Factor surprise

Kalman filter. Initialisation

$$\begin{aligned} x_{0|init} &= 0 \\ P_{0|init} &= \sigma^2 I \end{aligned}$$

with x being the system state (in my terms the prediction), P is the covariance matrix of the errors of x (in my terms the var of the predictions) and I is the identity matrix. Then the "correction" is given by

$$\begin{aligned} K_k &= P_{k|k-1} H_k^T (H_k P_{k|k-1} H_k^T + R_k)^{-1} \\ x_k &= x_{k|k-1} + K_k (z_k - H_k x_{k|k-1}) \\ P_k &= (I - K_k H_k) P_{k|k-1} \end{aligned}$$

with K the kalman gain matrix, H the observation matrix ($z_k = H_k x_k + \text{noise}$), R the covaraince of the measurement noise and z a new observation. The last part of the Kalman filter is the "prediction":

$$\begin{aligned} x_{k|k-1} &= F_{k-1} x_{k-1} + B_{k-1} u_{k-1} \\ P_{k|k-1} &= F_{k-1} P_{k-1} F_{k-1}^T + Q_{k-1} \end{aligned}$$


with F the transition matrix ($x_{k|k-1} = F_{k-1} x_{k-1}$), u a deterministic perturbation, B the dynamics of the deterministic perturbation. In our terms

$$\alpha = K_k = \frac{P_{k|k-1}}{R_k + P_{k|k-1}}$$

$P_{k|k-1}$, is however σ_P^2 in my implementation and R_k is fixed variance of inputs σ_S^2 . Hence, my implementation represents (?) the Kalman filter. Important to note is, that in my implementation we estimate the variance of inputs dynamically, so it is not set! Another nice advantage here is that I don't need a good estimate for P . I can basically initiate it as I want. Another difference is that I consider the optimal weighting in my "output neuron" and not the prediction itself

XXX Comparison to Bayes Factor surprise

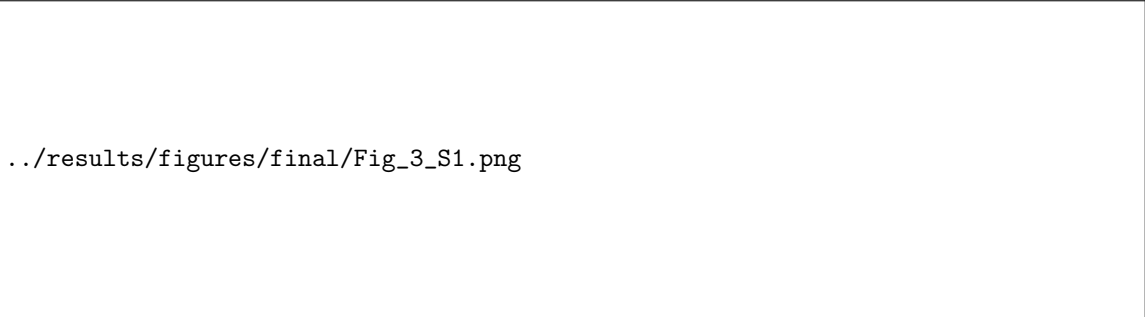
C Supplementary Figures



../results/figures/final/Fig_2_S1.png

Figure S1. Estimation of mean and variance for different stimulus distributions.

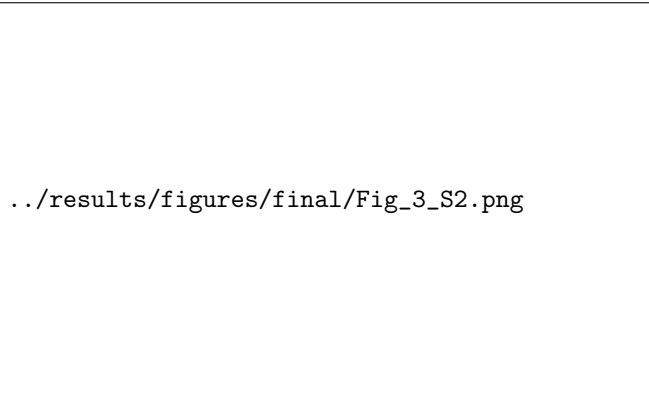
Top: The mean-squared error (MSE) between the running average and the activity of the M neuron decreases to a near-zero level for all stimulus distributions tested. Bottom: The MSE between the instantaneous variance and the V neuron decreases to a low level with minor differences between the distributions tested. Zoom-in shows the last half of the trial. Mean of the stimulus distribution = XXX, Variance of the stimulus distribution = XXX.



../results/figures/final/Fig_3_S1.png

Figure S2. Dynamic variance estimation allows flexible adaptation to changes in the stimulus statistics and environment.

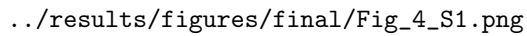
(A) Sensory weight for different input statistics (same as in Fig. 3E). Numbers denote specific example states. Arrows denote the transitions between those states. (B) The sensory weight over time is shown for all transitions in (A). The switch to a new input statistics occurs at trial 60. Parameters are taken from Fig. 3.



../results/figures/final/Fig_3_S2.png

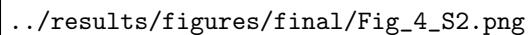
Figure S3. Perturbing the weighting of sensory inputs and predictions by altering network properties.

(A) The weights from the lower-level PE neurons to the M neuron are scaled by a factor below 1 (here, xxx) or above 1 (here, xxx), leading to a distorted weighting. If the update of the M neuron in the lower subnetwork is too slow (◄), the prediction is overrated. If the update of the M neuron in the lower subnetwork is too fast (►), the sensory input is overrated. (B) The precise activation function for the V neurons does not have a major impact on the sensory weight. Only for inputs with high stimulus variability, the sensory stimulus is slightly overrated when the squared activation function is replaced by a linear, rectified activation function.



../results/figures/final/Fig_4_S1.png

Figure S4. Neuromodulators acting locally either on interneurons in the lower or higher PE circuit. (A) Sensory weight changes with neuromodulators acting on interneurons in the lower PE circuit. (B) Sensory weight changes with neuromodulators acting on interneurons in the lower PE circuit. Simulation parameters, labels and colors as in Fig. 4.



../results/figures/final/Fig_4_S2.png

Figure S5. Perturbing the interneurons changes the baseline and gain of nPE and pPE neurons. (A) Changes in the baseline activity of nPE and pPE neurons for different interneurons targeted. 3 different mean-field networks are tested. (B) Same as in (A) but for the gain of nPE and pPE neurons. Simulation parameters, labels, and colors as in Fig. 4.