Temporal stimulus segmentation by reinforcement learning in populations of spiking neurons

Luisa Le Donne[a,b,c], Robert Urbanczik[d,†]
Walter Senn[d] and Giancarlo La Camera[a,b,e,f,*]

[a] Department of Neurobiology and Behavior, Stony Brook University
[b] Program in Neuroscience, Stony Brook University
[c] Neural Computation Laboratory, Center for Neuroscience and Cognitive Systems @UniTn, Istituto Italiano di Tecnologia, Rovereto, Italy
[d] Institute of Physiology, University of Bern, Switzerland
[e] Center for Neural Circuit Dynamics, Stony Brook University
[f] Institute for Advanced Computational Science, Stony Brook University

Abstract. Learning to detect, identify or select stimuli is an essential requirement of many behavioral tasks. In real life situations, relevant and non-relevant stimuli are often embedded in a continuous sensory stream, presumably represented by different segments of neural activity. Here, we introduce a neural circuit model that can learn to identify action-relevant stimuli embedded in a spatio-temporal stream of spike trains, while learning to ignore stimuli that are not behaviorally relevant. The model uses a biologically plausible plasticity rule and learns from the reinforcement of correct decisions taken at the right time. Learning is fully online; it is successful for a wide spectrum of stimulus-encoding strategies; it scales well with population size; and can segment cortical spike patterns recorded from behaving animals. Altogether, these results provide a biologically plausible theory of reinforcement learning in the absence of prior information on the relevance and timing of input stimuli.

[†]Deceased
[*]Address for correspondence: Giancarlo La Camera, Department of Neurobiology and Behavior, Stony Brook University, Stony Brook, NY, USA. Email: giancarlo.lacamera@stonybrook.edu.

Contents

1 Introduction **2**

2 Results **3**
2.1 The multi-choice segmentation task . . . 3
2.2 Robustness to encoding strategies . . . 8
2.3 Segmentation of cortical data . . . 10

3 Discussion **12**
3.1 Scaling performance with population size . . . 13
3.2 Flexibility to coding schemes . . . 13
3.3 Learning cortical spike patterns . . . 14
3.4 Precision vs. recall . . . 15
3.5 Comparison with previous work . . . 15
3.6 Extensions . . . 16

4 Methods **18**
4.1 The network model . . . 18
4.2 The neuron model . . . 19
4.3 The learning rule . . . 20
4.4 Stimuli and computer experiments . . . 22

5 Supplementary Figures **30**
5.1 Supplementary Figure 1: sparse stimulus embedding . . . 30
5.2 Supplementary Figure 2: robustness to spike time jitter . . . 31
5.3 Supplementary Figure 3: robustness to stimulus dimensionality . . . 32
5.4 Supplementary Figure 4: robustness to reduced connectivity . . . 33
5.5 Supplementary Figure 5: performance with correlated stimuli . . . 34
5.6 Supplementary Figure 6: robustness of the homeostatic mechanism . . . 35
5.7 Supplementary Table 1: performance and generalization with cortical datasets . 36

1 Introduction

Human beings have an unmatched ability to learn to abstract information from the environment, use this information to predict the consequences of their actions and develop appropriate behavioral strategies. A major component of this skill is the ability to identify the relevant cues and stimuli from the environment and determine what to do in response to them. Identifying relevant stimuli is a crucial aspect of many cognitive tasks including decision making. Apart from a few exceptions that, however, do not yet admit a biological implementation (Dayan and Daw, 2008), the great majority of learning and decision making models (Wang, 2008; Zhang et al., 2009) typically assume that the relevant stimuli are known to the learning agent.

In these models, learning refers to the process of associating known relevant stimuli with their predicted outcome for the purpose of making decisions (see e.g. (Sutton and Barto, 2018) for a general framework). The underlying assumption is that relevant stimuli can be identified by a form of bottom-up, unsupervised preprocessing; for example, relevant stimuli are so because statistically different from other stimuli. Although this could be the case for perceptually salient stimuli that 'stand out' from the background, in many real-life situations it is not clear which segments in a continuous sensory stream are action relevant, and relevant segments may blend seamlessly into irrelevant ones. Here, we present a spiking network model (the learning agent, or 'agent' for short) able to detect relevant stimuli from the environment by trying to maximize the reward obtained for correct decisions at the right time. The agent is required to solve a detection and a decision task simultaneously: detect which stimuli are behaviorally relevant and decide what to do in response to them. The stimuli are modeled as spatiotemporal patterns of spike trains, and the agent has initially no knowledge of their identity, timing, or associated response. Everything must be discovered by trial and error by exploiting reward feedback received after each decision. This is all the more challenging as relevant and non-relevant stimuli were built so as to have the same overall spike train statistics, so that different segments appear seamlessly embedded in a homogeneous input stream (see Figs. 1-2 for examples).

We emphasize that decisions are never enforced in this paradigm, and no feedback is given in the absence of decisions. This is a particularly demanding requirement because the quiescent state (i.e., the network state where no decisions are ever taken) is a local maximum of the average reward, and can result in the pernicious behavior of forgoing decisions altogether. Here we show that, in a large variety of situations, our agent can learn to identify the relevant stimuli embedded in patterns of spike trains and take the correct decision in response to them. Although more slowly, the agent also learns to ignore non-relevant stimuli. We also show that the agent is equally successful at segmenting patterns of cortical spike trains recorded from behaving rodents. These results were achieved through a fully online synaptic plasticity rule modulated by reward and feedback from the population decision. Overall, our model provides a biologically plausible framework for learning to segment an input stream for the purpose of solving behavioral tasks, and may be one way to form concepts out of seemingly unstructured stimuli (Zeithamova et al., 2019).

2 Results

2.1 The multi-choice segmentation task

We trained an ensemble of populations of spiking neurons (the 'agent') to learn to detect relevant stimuli embedded as segments of a continuous input stream of spatiotemporal patterns of spike trains (Fig. 1**a**). The agent had to simultaneously produce a desired response to each of the detected relevant segments. Each stimulus was a spatio-temporal pattern of spike trains, each lasting 500 ms. All stimuli were presented in a continuous fashion and in random order, so as to

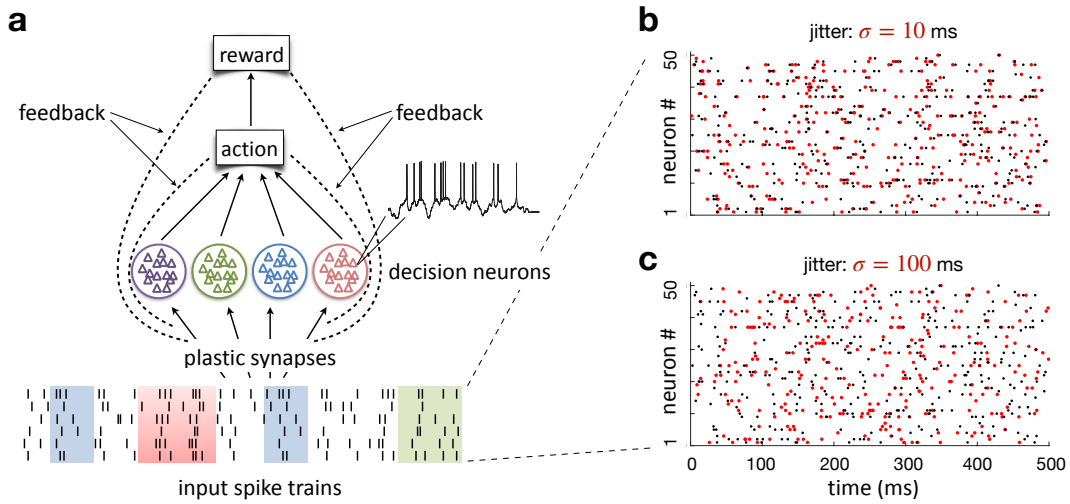

Figure 1: **Stimuli and network architecture**. **a)** Network architecture. Spike trains ('input spike trains') from M input neurons project to N_{dec} populations of spiking neurons (colored circles) through plastic synapses. Each population is responsible for a different decision, which occurs when a measure of the spike count of each population first crosses a predefined threshold. After each decision (and only then), feedback from reward (right vs. wrong) and the decision neurons' activity determines the change of synaptic weights between the input neurons and the population neurons responsible for the current decision, according to Eq. 13. **b)** A segment of the input stream comprising 50 spike trains. Each line is a spike train and each dot is a spike time. The segment shown represents one stimulus prototype (black raster) together with a jittered version of it (red raster; 10 ms random dispersion around the original spike times, see Sec. 4.4 of Methods for details). The prototype could represent a relevant or a non-relevant segment of the stream. **c)** Same prototype shown in panel b), this time jittered by 100 ms (red).

produce a continuous input stream, and were arbitrarily divided into relevant and non-relevant ones. The statistics of all stimuli (relevant and non-relevant) were identical: each input stimulus was generated as a pattern of 50 Poisson spike trains with firing rates uniformly distributed between 2 and 24 spikes/s. Each time a stimulus was presented, its spike times were jittered by a random amount of the order of 10 ms (see Fig. 1**b** and Methods for details), but we will show later that the model is robust to much longer jitter times (see Fig. 1**c** for an example with 100 ms jitter). The agent had to learn to detect the presence of the relevant stimulus by initiating an 'action'. This occurred when the readout of the population activity (a measure of its cumulative spike count) hit a fixed threshold.

A correct decision was met with reward and the removal of the stimulus, as if a target had been hit or captured. Every action incurred a penalty (a small negative reward), to discourage the agent from taking actions with high frequency. No reward was obtained in the absence of a decision. Reward was broadcast to plastic synapses connecting input neurons with decision neurons (Fig. 1**a**). The synaptic plasticity rule reinforced correct decisions taken at the right

time. It did so by using local feedback from the pre- and post-synaptic neurons combined with global feedback signals from reward and population activity. In the simplified case of episodic learning, where the input stream is segregated in trials and only one relevant stimulus is presented in each trial, this plasticity rule performs stochastic gradient ascent in the average reward (Urbanczik and Senn, 2009), a standard approach to reinforcement learning (see Methods for details). However, the problem considered here is much more difficult, because i) the input stream is not subdivided in trials; ii) the identity and timing of relevant stimuli is unknown to the learning agent, and iii) feedback is obtained only in the presence of a decision. For those reasons, this problem is akin to temporal stimulus segmentation combined with a multi-choice decision making task, or 'multi-choice segmentation task' for short, and the plasticity rule is not guaranteed to solve this problem. As we show next, however, a slightly modified version of this plasticity rule, designed to cope with the difficulties of the multi-choice segmentation task, is sufficient to solve this problem in a variety of scenarios.

Detection task

We first illustrate the problem in the simplest possible scenario of one relevant stimulus embedded in an input stream of non-relevant ones. A single population of decision neurons (the learning agent) was trained to respond whenever the relevant stimulus was on. As shown in Fig. 2, the agent learned to successfully identify the relevant stimulus and respond within 50 ms (or less) from its onset, while learning to ignore the other stimuli (Fig. 2**b**). This was achieved despite using equal spike train statistics for relevant and non-relevant segments (notice how the input stream appears uniform on visual inspection, see e.g. the spike trains in black in Fig. 2**b**). The average learning curves across 150 training sets is shown in Fig. 3**a**. High performance on relevant stimuli was achieved at different times across independent training sets, depending on the initial weights, the input prototypes and stochastic decisions. We call the time to achieve high performance the 'learning threshold', and quantify it as the first of 100 consecutive correct responses to relevant stimuli, taken as a criterion that learning has been accomplished. The learning thresholds were reached in every training set and were broadly distributed across independent training sets, as shown in the inset of Fig. 3**a**. Despite this variability, the average learning curve appears to improve gradually over time. Note that a variable learning threshold is an unavoidable consequence of the nature of the task: if the agent is initially set to take random and very sparse decisions, feedback is rare and so are synaptic changes. However, once the agent reaches the learning threshold for relevant stimuli, it quickly reaches the maximal performance for those stimuli. In the detection task, fewer than 250 presentations per stimulus were sufficient to reach criterion for half the training sets (see inset of Fig. 3**a**).

Multi-choice segmentation tasks

To implement multiple-choice decisions tasks, the decision neurons were partitioned in multiple populations (Fig. 1a). Each population encoded a different decision, i.e., was responsible for

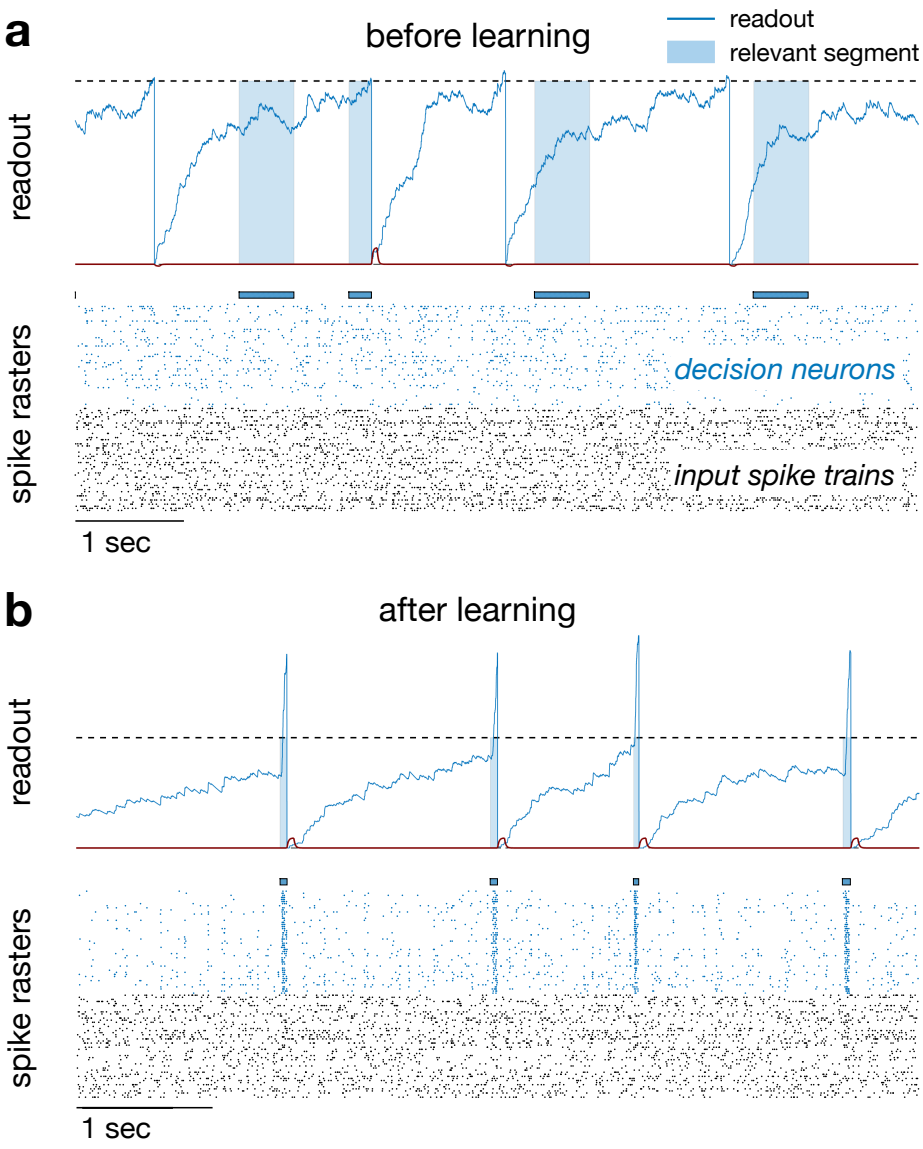

Figure 2: **Learning the 'detection' task**. **a)** Before learning. *Top panel:* readout of the single population of decision neurons, P_s^μ (blue, low pass filter of population spike count; see Eq. 2 of Methods). A decision was taken when P_s^μ crosses a threshold $\theta_D = 1$ (dashed horizontal line). Colored shadings above the rasters mark the presence of a relevant stimulus. Thirty milliseconds after a decision, the population readout was reset to zero. If the decision was a correct decision in the presence of a relevant stimulus, the stimulus was removed and reward was given (red line at the bottom). The initial decisions reflected the random initial weights of the synaptic connections between the input spike trains and the decision neurons. *Bottom panel:* raster plot of neural activity, with input spike trains in black and output spikes of decision neurons in blue. **b)** After training, the population readout correctly (and rapidly) reaches the threshold only in the presence of the relevant stimulus. In the input stream were embedded 4 prototypical segments acting as stimuli, of which 1 relevant (present during the blue shadings) and 3 non-relevant (present outside the blue shadings). Learning was assessed after 2,000 presentations per stimulus; $N = 50$ decision neurons and $N_i = 50$ input spike trains were used.

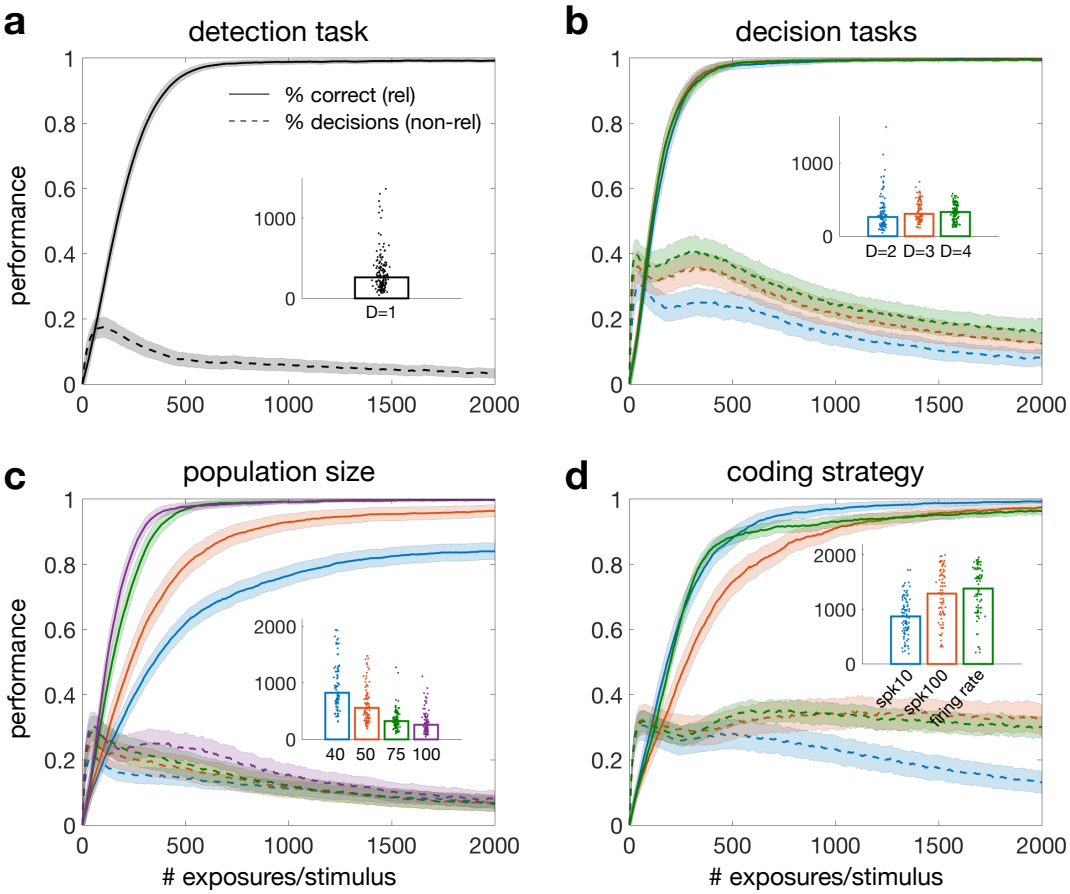

Figure 3: **Average learning curves in temporal segmentation tasks. a)** Detection task with 50 input spike trains and $N = 50$ decision neurons (same model and task as in Figure 2). *Solid curve:* average fraction of correct decisions with relevant stimuli across 150 training sets. *Dashed curve:* average fraction of (any) decisions in the presence of non-relevant stimuli (note: perfect performance with non-relevant stimuli is achieved when the dashed line converges to zero). **b)** Learning curves across 100 training sets for 3 different tasks having $D = 2$ (blue), 3 (red) or 4 (green) decision classes and one relevant stimulus per class out of 8, 12 and 16 total stimuli, respectively. In each case, there were D decision populations (regardless of the number of stimuli), each containing 100 (blue), 150 (red) and 200 (green) neurons. **c)** Learning speeds up population size. Shown are the mean learning curves for the binary decision task with 8 stimuli and $N = 40, 50, 75$ or 100 decision neurons per population (means across 100 training sets except for $N = 40$ where 200 sets were used). **d)** Learning with different encoding strategies: firing rate (green) and spike timing coding with a jitter $\sigma = 10$ ms (blue) and $\sigma = 100$ ms (red). The learning curves were averaged across 200 training sets for firing rate coding (green) and across 100 training sets in the remaining tasks. In all panels, shaded areas define SEM bounds across training sets. *Insets:* dotplots of learning thresholds (see the text) across training sets for each task (color coded as in the main plots); bars show median learning thresholds.

learning to identify stimuli belonging to the same decision class.

Fig. 3**b** shows successful performance in a segmentation task occurring simultaneously with a multiple-choice decision task with 2, 3 or 4 relevant stimuli (in the presence of 6, 9 and 12 non-relevant stimuli, respectively; learning thresholds were reached in every training set). An example of network activity before and after learning the 3-way decision task is shown in Fig. 4.

Note from Fig. 3**b** the nearly perfect overlap of the learning curves for the different tasks, achieved by increasing the number of decision neurons as the task gets more complex. This suggests that learning speeds up with the number of decision neurons (Urbanczik and Senn, 2009), as confirmed in Fig. 3**c**. This panel shows the performance in the binary decision task when different numbers of decision neurons per population are used. As shown in the figure, a larger population size resulted in a faster learning speed (the learning rate was kept constant to allow for proper comparison).

Next, we tested whether the model could handle tasks with multiple relevant stimuli in each decision class, as well as various levels of stimulus sparsity (i.e., ratio of relevant to non-relevant stimuli; see Fig. S1). Although learning was slower for either a larger number of stimuli or for larger ratios, the average performance after training was high in all cases (ranging from 94% to 99% on relevant stimuli; see Fig. S1).

2.2 Robustness to encoding strategies

Stimuli encoding into patterns of spike trains (Fig. 1**b**) was meant to reflect two widespread features of cortical responses: the relative stability of firing rates across trials coupled to unreliability in the precise spike times (Shadlen and Newsome, 1998; Churchland et al., 2010; Dayan and Abbott, 2001). Learning in our model was very robust to jitter in the spike times (a proxy for trial-to-trial variability), tolerating amounts of jitter so large to cover the entire duration of the stimulus (Fig. S2). The latter case was obtained by generating the spike times anew at each stimulus presentation and represents a full firing rate code.

The model was also robust to variations in coding strategies. We tested the model in a 'spike timing' coding mode wherein all input spike trains had the same firing rate so that all information about the stimuli was contained in the spike times – i.e., firing rate could not be used to disambiguate the stimuli (Urbanczik and Senn, 2009). Learning was successful also in this case (Fig. 3d). Notably, despite the crucial role of spike timings in this code, learning was successful in the presence of substantial temporal jitter in the spike times (the case of 10 and 100 ms jitter are shown in Fig. 3d). Learning was slower with larger jitter, but the asymptotic performance was similar in all cases. The performance with small jitter was comparable to the performance with firing rate patterns (shown in green in Fig. 3d).

Next, we tested the model's robustness to reduced stimulus dimensionality (i.e., smaller number of input spike trains). Learning was faster for higher-dimensional input patterns, but reached perfect asymptotic performance for as few as 25 input spike trains (Fig. S3). Similar results were found with sparse connectivity, i.e., after reducing the number of synapses project-

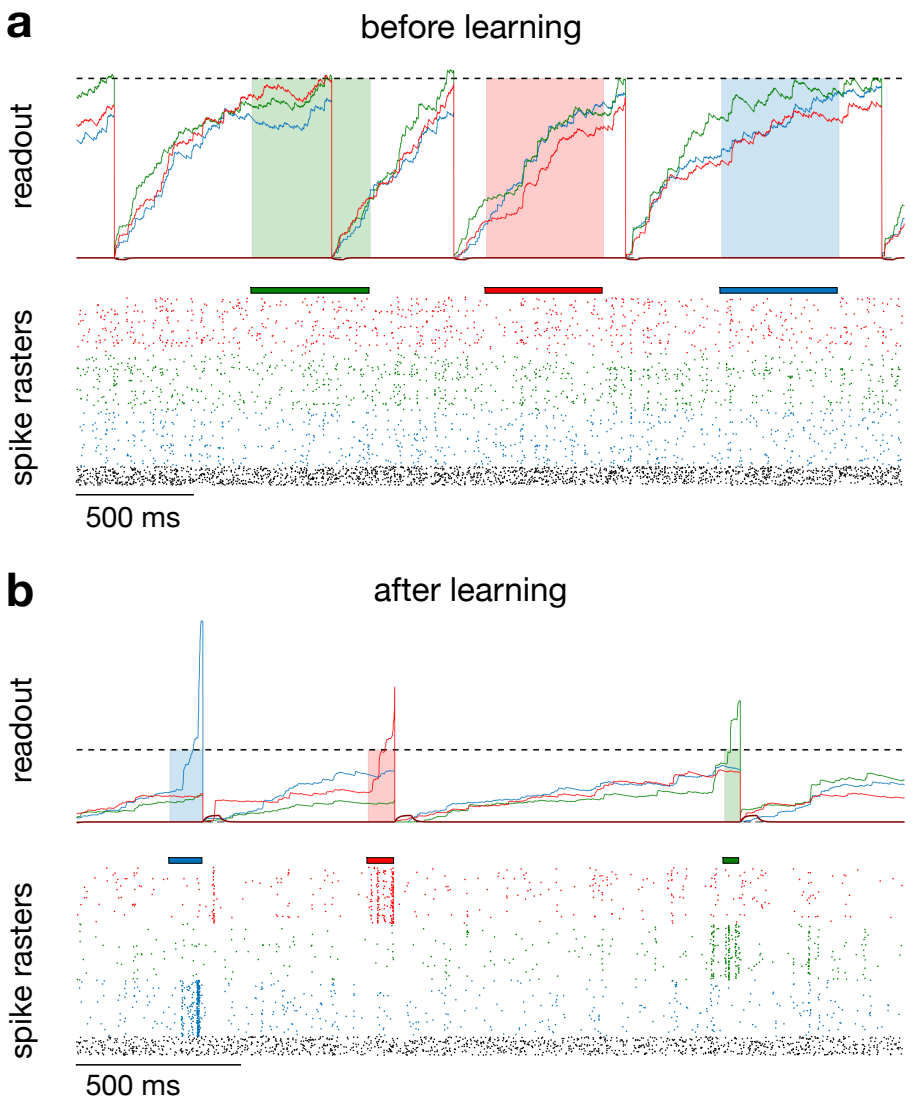

Figure 4: **Learning a multiple choice segmentation task. a)** Before learning. *Top panel:* readout of the population of decision neurons for three decision populations (color-coded; same keys as in Fig. 2). A decision was taken when the corresponding population readout crossed a threshold $\theta_D = 1$ (dashed horizontal line). Colored shadings above the rasters mark the presence of a relevant stimulus (different colors correspond to different stimulus classes). Reward was given after a correct decision in the presence of a relevant stimulus (see caption of Fig. 2 for details). *Bottom panel:* raster plot of neural activity, with input spike trains in black and output spikes of decision neurons in color (same color code as in top panel). **b)** After training. Same keys as in panel *a*. Learning was assessed after 2,000 presentations per stimulus; $N = 150$ decision neurons and $N_i = 50$ input spike trains were used.

ing to the decision neurons (Fig. S4). Each decision neuron received input from a randomly chosen subset of input spike trains. There was no difference in learning speed between the full connectivity and the case of 80% connectivity, and no appreciable degradation of final learning performance for levels of connectivity down to 30% (Fig. S4).

Finally, we tested the model with correlated input stimuli. In all cases shown so far, the spike trains were independently generated. Cortical spike trains however have some finite degree of pairwise or higher order correlations (Zohary et al., 1994; Schneidman et al., 2006; Montani et al., 2009; Ohiorhenuan et al., 2010; Yu et al., 2011; Ganmor et al., 2011; Doiron et al., 2016; Mazzucato et al., 2016). We first tested the model on spike trains with known pair-wise correlations. These spike trains were generated with the method of the dichotomized Gaussian model (Macke et al., 2009), which allows to generate Poisson spike trains with desired firing rates and pair-wise correlations ρ (examples are shown in Fig. S5). After 2,000 presentations per stimulus, the model had reached perfect performance on relevant stimuli for $\rho = 0.1$ and $\rho = 0.3$, and nearly perfect performance for $\rho = 0.5$ (Fig. S5d); mean performance with non-relevant stimuli was 94% or higher for any ρ.

2.3 Segmentation of cortical data

The correlated stimuli of the previous section allow to quantify the learning performance as a function of a controlled amount of pair-wise correlations. The ultimate test, however, is the ability to learn real cortical data. To this purpose, we built prototypes of relevant and non-relevant segments from input spike trains recorded simultaneously from the gustatory cortex of awake behaving rats (Mazzucato et al., 2015; Samuelsen et al., 2012) (see Methods for details). The rats had to press a lever to obtain one of 4 tastants (sucrose, citric acid, sodium chloride and quinine). In between trials, rats also received passive delivery of tastes at random times. Data collected prior and after passive taste delivery were used to build the prototypes (see Methods for details). Ensembles of 5 to 9 single-units were recorded in each session, and units from all sessions were combined to obtain spike trains from 101 neurons (such patterns for two sucrose trials are shown in Fig. 5a). Of these, 50 neurons were randomly chosen to create prototype stimuli (Fig. 5b; different subsets of neurons were sampled for different training sets). This dataset was particularly appropriate for our purposes due to its long inter-trial intervals and because the neuronal activity had been recorded simultaneously from multiple single units. The learning problems were the same as with the surrogate Poisson spike trains described earlier. In the binary decision task, relevant stimuli were chosen to be the neural responses to two tastants (e.g., sucrose and citric acid); the responses to three different tastants were used in the 3-way decision task, and the responses to all four tastants were used in the 4-way decision task. For each tastant, the portion of the recordings immediately following stimulus delivery was taken as the relevant segment, whereas recordings ending 1s before stimulus delivery were taken as non-relevant segments (see Fig. 5a-b and Methods for details).

The results are shown in Fig. 5c and are similar to the results obtained with surrogate spike trains (Fig. 3b). We also tested the ability of the agent to generalize to unseen stimuli of the

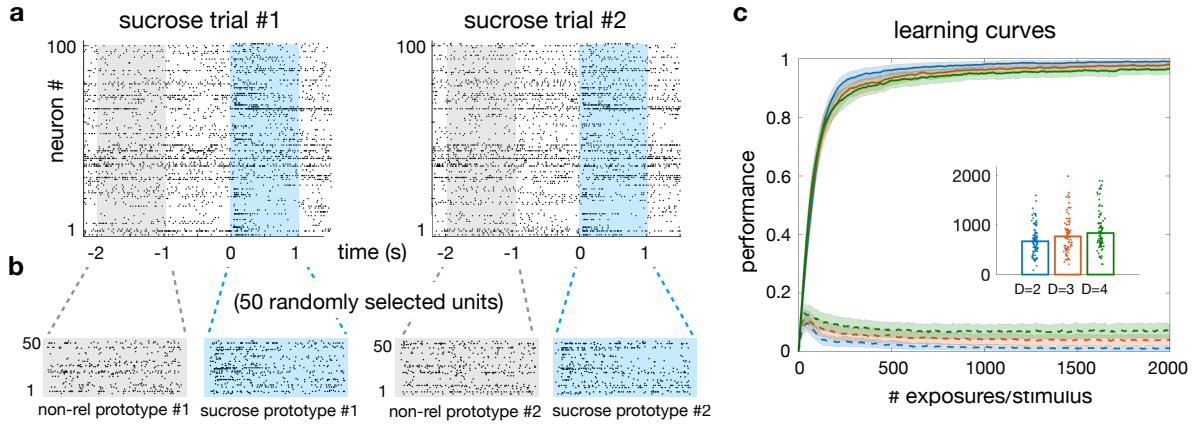

Figure 5: **Segmentation of cortical data**. **a)** Procedure used to build stimulus prototypes from cortical spike trains. Shown are spike trains from 101 neurons recorded over multiple sessions during the first two sucrose trials of each session (i.e., trials in which the tastant 'sucrose' was delivered). Taste delivery occurred at time zero. **b)** Generation of a single training set from cortical spike trains in (a). Data in a temporal window of $[-2, -1]$ s were used to generate the prototypes for non-relevant stimuli (grey rasters); data in a temporal window of $[0, 1]$ s were used to generate the prototypes for relevant stimuli (blue rasters). The same random subset of 50 neurons were used for all relevant and non-relevant stimuli in each training set. Prototypes for different tastants were built in the same way. Different random sets of neurons were used to generate different training sets (see Methods for details). **c)** Average learning curves with the stimuli obtained as shown in panels (a-b) for 3 different tasks having 2 (blue), 3 (red) or 4 (green) decision classes and 1 relevant stimulus per class out of 8, 12 and 16 total stimuli, respectively. Each curve is the average across 100 training sets generated from cortical data as described above (same keys as in Fig. 3b). *Inset:* dotplots of learning thresholds for each task. Learning thresholds were reached in 87% ($D = 4$), 92% ($D = 3$) and 100% ($D = 2$) of training sets.

same class (i.e., elicited by the same tastant, but in trials not used for training). The results are reported in Table S1 (column 'generalization'). To assess the quality of generalization, its performance was compared to the performance on unseen stimuli of a different class (i.e., elicited by a different tastant than used for training; see Methods for details). As shown in Table S1 (compare columns 'generalization' with 'control'), the generalization performance was significantly higher than control in all tasks ($p = 0.0073$ or lower, one-tailed Wilcoxon rank sum test; see the caption of Table S1 for details). Note that the low absolute value of generalization performance ($< 50\%$) was the consequence of training on one stimulus per class while testing on 6 unseen stimuli of each class. Generalization performance can be boosted by the common practice of training on multiple stimuli. For example, training in the binary decision task with 6 stimuli per class, and then testing on one unseen stimulus of each class, gave a mean generalization performance of 74% (Table S1).

3 Discussion

Learning to abstract relevant information from the environment is a crucial component of decision making; yet, standard approaches typically assume that the relevant stimuli, and especially the times at which they occur, are known to the decision maker (Corrado and Doya, 2007; Gold and Shadlen, 2007; Lee and Wang, 2009). Even assuming partial ignorance about the current state or about the rules of a decision task (Dayan and Daw, 2008), there is usually no ambiguity about the presence of a cue that, possibly incompletely, signals the current state. This is especially true for the type of decision tasks typically modeled by signal detection theory (Green and Swets, 1966). For example, in psychophysical discrimination tasks with random dot displays (Britten et al., 1992; Churchland et al., 2008; Park et al., 2014), the aggregate direction of motion may not be perceptually clear, but the presence of the random dot display occurs at known times and there is no doubt on its relevance to the task.

In the problems studied here, relevant stimuli are continuously embedded in spatiotemporal patterns of neural activity, and occur at times that are unknown to the agent. Such a task requires the temporal segmentation of the input stream, which is possible only if the agent takes the correct action at the right time. Thus, both the presence of relevant segments and the correct action associated to each segment must be discovered by the learning agent. Since this learning process ends up assigning meaning to the relevant segments, it is tempting to speculate that this reward-based mechanism of parsing input streams may be one way to form concepts out of seemingly unstructured stimuli (Hampton et al., 2006; Zeithamova et al., 2008).

Learning in this scenario was accomplished in a biologically plausible model with populations of spiking neurons. The model is plausible in several ways: i) it uses a local learning rule based on pre- and postsynaptic spikes as well as post-synaptic potentials, all quantities directly accessible to each synapse; ii) the learning rule receives global reward from decisions and rewards via neuromodulation (Schultz, 1998; Yu and Dayan, 2002); (iii) learning can be achieved with erratic cortical spike trains and variable spike counts, with different coding strategies (spike

timings or firing rates), in the presence of variable degrees of correlations, and with no need to customize the learning rule to each case separately. We start by discussing the salient features of our approach and later compare it with existing approaches for reinforcement learning and decision making.

3.1 Scaling performance with population size

Reinforcement learning with spiking neurons is notoriously slow (e.g. Seung (2003); Izhikevich (2007); Frémaux et al. (2010)) and does not scale well with population size (Xie and Seung, 2004; Werfel et al., 2005; Fiete et al., 2007). This is because larger populations of neurons make the spatial credit assignment problem more severe, resulting in slower learning (Urbanczik and Senn, 2009; Friedrich et al., 2010). A slowdown of learning speed with population size is also contrary to the principle of population coding, according to which performance in any task should increase with population size (Averbeck et al., 2006). Thanks to a learning rule that averages the gradient of the reward over the outcomes of each neuron, in our model learning speed increases with the size of decision populations (Fig. 3**c**). This means that increased task difficulty can be compensated for by increasing the number of decision neurons, so that similar learning speeds can be reached in more difficult tasks, for example, when increasing the number of decision classes (Fig. 3**b**), the number of input stimuli, or the number of non-relevant stimuli embedded in the input stream (Fig. S1).

3.2 Flexibility to coding schemes

Our learning rule is also robust to a large class of coding schemes and can handle real cortical datasets. We have shown successful learning in 3 different coding scenarios: i) when stimuli are coded by prototypical spike patterns, despite large amounts of jitter in the spike times (Fig. 3**b** and Fig. S2), ii) when stimuli vary only in their spike timings across neurons, with equal firing rates across neurons (spike timing coding, Fig. 3**d**)), and iii) when stimuli vary both in their firing rate patterns and in their single neurons' spike times (a full firing rate code; green curve in Fig. 3**d**). We emphasize that in the case of precise spike patterns (scenario ii), all stimuli had exactly the same statistics, and thus they could not be encoded as firing rate vectors, nor could be learned by an unsupervised algorithm that looks for idiosyncratic regularities in the stimuli. Successful learning occurred also in the case of stimuli encoded as firing rate patterns, where the spike trains had different firing rates but were generated anew at each stimulus presentation. In these more challenging tasks learning was correspondingly slower, albeit still comparable to the main coding scenario, as shown in Fig. 3**d**. Moreover, the model performs well on stimuli with correlated spike trains (Fig. S5) as well as cortical spike patterns recorded from behaving animals (Fig. 5; more on this later).

The fact that learning is successful in the case of lax spike timing reproducibility is particularly appealing, giving that cortical spike trains are notoriously erratic. Our results suggest that our model could work in the face of the large trial-to-trial variability observed in cortex

(Shadlen and Newsome, 1994; Holt et al., 1996). This is notable given that the learning rule was initially designed to handle spike patterns that are precisely timed. We note that, within a framework similar to ours, learning rules that depend directly on firing rates can be built, an approach that would allow for faster learning (Friedrich et al., 2014), at least in more traditional tasks. We decided not to use this approach for the segmentation task of this paper, in part for reasons of biological plausibility, as it is not clear how plasticity could depend on firing rates directly; and in part to obtain a more general learning rule that could handle a wider range of codes. The problem of neural coding is an open question and this approach allowed us not to commit to a specific coding scenario. Our results demonstrate the flexibility of our model in dealing with coding schemes interpolating between precise spike timing and firing rate coding with various degree of pairwise correlations, while also showing robustness to low levels of connectivity (Fig. S4) and stimulus dimensionality (Fig. S3).

Regarding the decision code, we have used the simple and widely used approach of having separate populations of neurons competing for decisions (Bogacz et al., 2006; Wang, 2008). Although it is possible, in principle, to adapt the model to handle decisions based on other coding scenarios, such as phase coding (Hartley et al., 2014; O'Keefe and Recce, 1993; Skaggs et al., 1996) or latency coding (Friedrich et al., 2010; Ponce-Alvarez et al., 2012), we have not tested these possibilities. The emphasis of this work was on learning to segment unknown stimuli, with less relevance given to the decision mechanism. It would be of interest to develop a biologically plausible model capable of learning its own 'valid' actions in addition to its own relevant stimuli. This is left for future work.

3.3 Learning cortical spike patterns

The model can segment cortical spike train patterns recorded in the gustatory cortex of behaving rats, with good generalization performance to unseen segments recorded under the same conditions. Testing the model on cortical data has two main advantages over surrogate stimuli. First, it allows a fairer evaluation of the generalization performance. We expect generalization to surrogate stimuli of the same class to be easy, given that unseen stimuli would be generated by the same prototypes used for training. Secondly, it allows to quantify the ability of the model to extract essential features of the data related to the external stimuli. This is particularly relevant during generalization. Evoked cortical patterns in the gustatory cortex can be successfully decoded (Jones et al., 2007; Jezzini et al., 2013), suggesting that some common features of the data are preserved across stimulus presentations. However, it is not clear how to extract these common features. Previous efforts have analyzed the higher order correlations of cortical spike trains in search for identifiable features (Schneidman et al., 2006; Montani et al., 2009; Ohiorhenuan et al., 2010; Yu et al., 2011; Ganmor et al., 2011; Shimazaki et al., 2015), but this remains a difficult task. It is reassuring and, in fact, quite remarkable that we can learn to extract the essential features of spike patterns by reinforcement of correct decisions.

3.4 Precision vs. recall

The performance of the model on relevant stimuli was generally higher than on non-relevant stimuli, especially with surrogate data. The tasks themselves were not symmetrical, since non-relevant stimuli must be simply ignored and produce no reward. It is possible to improve performance with non-relevant stimuli at the expense of performance with relevant stimuli by choosing a different cost/reward ratio. Our main goal, however, was to demonstrate the ability to segment unknown relevant stimuli. In order to learn, the agent must explore by taking random actions, and acting during non-relevant segments is a reasonable price to pay to discover the relevant ones. In the language of pattern recognition, this issue can be stated in terms of 'precision', the fraction of relevant stimuli among those detected as relevant, vs. 'recall', the fraction of relevant stimuli identified as such (Fawcett, 2006; Saito and Rehmsmeier, 2015). Lower performance with non-relevant stimuli lowers the precision while keeping the recall intact. We have basically prioritized the recall, i.e., the performance on relevant stimuli, which is close to perfect in all our tasks. As learning problems become more difficult, it is reassuring to know that we may still attain a high recall at the cost of losing precision, if necessary.

3.5 Comparison with previous work

Segmentation tasks are often solved by unsupervised methods such as Hidden Markov Models (HMMs) (Rabiner, 1989), which have been successfully applied to the segmentation of cortical patters of spike trains similar to those modeled here (Abeles et al., 1995; Jones et al., 2007; Ponce-Alvarez et al., 2012; Mazzucato et al., 2015; Maboudi et al., 2018; Mazzucato et al., 2019). However, HMMs require a-priori knowledge of the structure of the relevant stimuli (e.g., Poisson-HMM vs. Gaussian-HMM) and they also require knowledge of the total number of relevant stimuli (or hidden states, in HMM parlance). Although non-parametric variations of HMM exist (Gershman and Blei, 2012; Chen, 2013), they are based on recursive offline algorithms. Online procedures to approximate the expectation-maximization steps required in HMM can learn to perform tasks based on the temporal statistics of stimuli (Celeux and Diebolt, 1985; Neal and Hinton, 1998; Mongillo and Deneve, 2008; Kappel et al., 2014), but would not be able to learn to segment a stream where the input spike trains all have the same firing rate, as is the case of the spike timing code of Fig. 3**d**.

Our model also differs from the class of recurrent neural circuit models of decision making (Wang, 2008). As most traditional models of decision making, these models require a neural population encoding each stimulus, *a priori* knowledge of the relevant stimuli, and knowledge of their onset and offset times. Following an approach more similar to ours, the 'tempotron' (Gütig and Sompolinsky, 2006) can learn to separate spike patterns into two classes, which could be interpreted as 'relevant' vs. 'non-relevant'. However, the tempotron needs to know when stimuli start and end, and is given feedback for non-responses to relevant stimuli, which helps their identification. A modification of the tempotron able to perform "aggregate-label" learning is closest to our problem and approach (Gütig, 2016). In aggregate-label learning, a neuron is

trained to respond to different stimuli with a different number of spikes to each, with no *a priori* knowledge of which stimuli are relevant and when they are being presented. However, also in this case, there are crucial differences with our work. First, in aggregate-label learning, the task is divided in trials and feedback is always given at the end of each trial. Moreover, the algorithm is supervised, in that it needs knowledge of the total number of spikes required in the current trial. This number is externally calculated and fed back to the neuron for comparison with the actual number of emitted spikes during that trial. Finally, each segment is reliably coded by the same number of spikes in each trial in which it appears. Given the well known variability of cortical spike trains (Shadlen and Newsome, 1994; Holt et al., 1996), it is also implausible to assume that each stimulus is represented by the same exact number of spikes in every trial.

In contrast to the approaches reviewed above, our spiking network model learns online, does not require *a priori* knowledge of the relevant stimuli or their occurrence times, it is based on a biologically plausible learning rule, it handles cortical-like patterns of spike trains with arbitrary variability and correlation, and it allows for direct comparison with neurobiological data.

3.6 Extensions

Our work can be extended in a number of directions. One possible direction is to include non-stationary stimuli. In one example, spike patterns slowly appearing and disappearing on top of a noisy background would have to be detected and segmented according to action-relevance. Different subsets of sensory neurons would encode different stimuli, while the remaining neurons would represent streaming noise (different subsets for different stimuli). Preliminary computer experiments show encouraging results, at least in the presence of 2-8 stimuli with negligible coding overlap (not shown). Another possible direction is the extension to sequential decision tasks (Friedrich et al., 2011), wherein actions and/or rewards are required only after a sequence of relevant segments. In such a scenario, decisions depend not just on current state but also on previous history, and proper segmentation could lead to context-dependent representations of relevant stimuli (i.e., according to the sequence in which they are embedded (Rigotti et al., 2010a)). Neural representations of planning sequential behavior are essential to many complex behaviors and are related to concept learning and the representation of abstract rules (Genovesio et al., 2005; Hampton et al., 2006; Tanji et al., 2007; Rigotti et al., 2010b; La Camera et al., 2018; Zeithamova et al., 2019; Mansouri et al., 2020). An extension of our framework to deal with more complex embeddings of stimuli could potentially lead to context-dependent representations of relevant events or even abstract concepts. A comprehensive investigation of these directions is left for future studies.

Acknowledgments

We acknowledge support from the National Science Foundation Grant IIS-1161852 (GLC) and the Swiss National Science Foundation through the SystemsX.ch initiative "Neurochoice" (WS).

We thank Dr. Alfredo Fontanini for sharing the cortical datasets and Dr. Luca Mazzucato for useful discussions. This work is dedicated to the memory of Robert Urbanczik.

Author contributions

GLC, RU and WS designed research. GLC supervised research. GLC and LLD performed research. GLC wrote the manuscript. All authors approved the final version of the manuscript.

4 Methods

4.1 The network model

Task and network architecture. The model network comprised N_{dec} populations of spiking neurons receiving input spike trains via plastic synapses (Fig. 1A), with N_{dec} varying from 1 (detection task) to 4. Unless stated otherwise, each decision population comprised $N = 50, 100, 150$ or 200 neurons in tasks with $N_{dec} = 1, 2, 3$ or 4 decision classes, respectively. Except in the study of reduced connectivity (Fig. S4), all input spike trains projected to all decisions neurons. Each population μ coded for a specific decision and initiated the corresponding action whenever a measure of population activity P_s^μ crossed a decision threshold, i.e., a pre-defined level of activity $\Theta_D = 1$ (the same for all populations). As long as all population scores were below threshold, no decisions were taken and no feedback was given to the agent. A precise definition of P_s^μ is given below.

Stimuli were continuously presented to the agent as streaming spatio-temporal patterns of spike trains, and were divided into relevant and non-relevant stimuli (details in Section "Stimuli and computer experiments" below). Each relevant stimulus was arbitrarily associated with a correct decision, one among the N_{dec} possible decisions. When a decision occurred, a rewarding feedback was given after a minor delay (30ms), the stimulus was removed, and the population readout was reset to zero. Every decision (whether correct or incorrect) incurred a small cost ($R = -0.1$) to prevent the agent from continuously taking decisions; positive reward ($R = 1$) was given only for a correct decision in the presence of a relevant stimulus (netting a total reward of $R = 0.9$). In the (rare) case of multiple populations reaching threshold simultaneously, only one was selected pseudorandomly as the population responsible for the decision.

Population scores and decision dynamics. The ongoing activity of population μ, defined as

$$A^\mu(t) = \int_0^t ds \sum_k \delta(s - t_k^\mu), \tag{1}$$

is a measure of the collective spike count of its neurons between times zero and t. t_k^μ are the times of the spikes emitted up to time t by neurons of population μ, and $\delta(t)$ is Dirac's delta function. Since essential to our learning problem is the fact that the learning agent does not know when a stimulus starts or ends, $A^\mu(t)$ was approximated by a low-pass filter of its temporal derivative, $\dot{A}^\mu(t)$, to produce the online 'population score' P_s^μ responsible for initiating a decision:

$$\tau_P \frac{dP_s^\mu}{dt} = -P_s^\mu + \frac{dA^\mu}{dt}. \tag{2}$$

The time constant τ_P was 500 ms, the same as the mean stimulus durations. A decision μ was said to have occurred at the first time t_d such that $P_s^\mu(t_d) > \Theta_D > P_s^{\neq\mu}(t_d)$ since the last decision, and at time $t_d^* \equiv t_d + 30$ms all population scores P_s^μ were reset to zero. The 30 ms delay

between the decision and the reset adds biological realism while allowing a confidence signal to build up, as follows. At time t_d^* following a decision by population μ, a feedback on the decision is given to all neurons in the population through the quantity $\tilde{P}_s^\mu$ obeying

$$\tau_p \frac{d\tilde{P}_s^\mu}{dt} = -\tilde{P}_s^\mu + \mathbb{1}_{50,D^\mu} \exp\left(-\left(\frac{P_s^\mu(t_d^*) - \Theta_D}{\Theta_D}\right)^2\right), \tag{3}$$

with $\tau_p = 50$ ms and $\mathbb{1}_{50,D^\mu} = 0$ except for 50 ms after t_d^*, when it is set to either $+1$ or to -1, depending on whether or not the population μ was responsible for the current decision. The driving term $\tilde{P}_s^\mu$ has a double role: its magnitude can represent the population's own confidence in the decision at time t_d^* and as such it was used to attenuate learning (details in Sec. 4.3); its sign was used to build an individualized reward signal as proposed by Urbanczik and Senn (2009) (see Eq. 14).

4.2 The neuron model

The decision neurons producing scores P_s^μ were modeled as spiking neuron models driven by post-synaptic potentials and having a probabilistic threshold for spike emission (Pfister et al., 2006). Specifically, neuron ν is defined by membrane potential u^ν obeying

$$u^\nu(t|\mathbf{X}) = u_{rest} + \sum_i w_i^\nu PSP_i(t, \mathbf{X}) - \sum_{t^\nu \in Y} e^{-\frac{t-t^\nu}{\tau_m}} \Theta(t - t^\nu), \tag{4}$$

where: $u_{rest} = -1$ is the resting potential (in arbitrary units); w_i^ν is the synaptic weight connecting neuron ν to its presynaptic neuron i; $\mathbf{X} \equiv \{X_1, X_2, \ldots, X_M\}$ is the collection of the M presynaptic input spike trains, where $X_i = \{t_1^{(i)}, t_2^{(i)}, \ldots, t_n^{(i)}\}$ are the spike times of the ith spike train; $PSP_i(t, \mathbf{X})$ is the postsynaptic potential due to the input spike train X_i,

$$PSP_i(t, \mathbf{X}) = \sum_{s \in X_i} \frac{1}{\tau_m} e^{-\frac{t-s}{\tau_m}} \Theta(t - s), \tag{5}$$

with $\tau_m = 10$ ms being the membrane time constant of the neuron; Y is the output spike train (comprising spike times t^ν); and $\Theta()$ is the Heaviside function ($\Theta(t) = 1$ for $t > 0$, and $\Theta(t) = 0$ otherwise). A neuron with membrane potential u in a short time bin dt emitted a spike with probability

$$\phi(u)dt = \lambda e^{\beta u} dt. \tag{6}$$

Here, $\beta = 5$ controls the stochasticity of spike emission whereas λ has an additive or subtractive effect on spike probability, i.e., changes in λ shift $\phi(u)$ rightwards or leftwards. λ was slowly modified during learning to help keep the decision populations' activities near the decision threshold (see Eq. 16).

4.3 The learning rule

Given input spike pattern $\mathbf{X}$ lasting from 0 to t, the log-likelihood that a decision neuron produces the output spike train Y^ν is (Pfister et al., 2006)

$$\mathcal{L}_{\mathbf{w}^\nu}(Y^\nu|\mathbf{X}) \equiv \ln \mathcal{P}_{\mathbf{w}^\nu}(Y^\nu|\mathbf{X}) = \sum_{t^\nu \in Y^\nu} \ln \phi(u(t^\nu)) - \int_0^t dt' \phi(u(t')). \tag{7}$$

The learning rule was designed to perform a variation of gradient ascent in the average reward. Following Williams (1992), stochastic gradient ascent in the average reward R requires the gradient of the log-likelihood $\mathcal{L}_{\mathbf{w}^\nu}$:

$$\frac{dw_i^\nu}{dt} \propto R \frac{\partial}{\partial w_i^\nu} \mathcal{L}_{\mathbf{w}^\nu}(Y^\nu|\mathbf{X}). \tag{8}$$

From Eqs. 4, 6, 7, the gradient of the log-likelihood is given by

$$\frac{\partial}{\partial w_i^\nu} \mathcal{L}_{\mathbf{w}^\nu}(Y^\nu|\mathbf{X}) = \sum_{t^\nu \in Y^\nu} \beta PSP_i(t^\nu|\mathbf{X}) - \int_0^t \beta \phi(u(t')) PSP_i(t'|\mathbf{X}) dt'. \tag{9}$$

This is the so-called 'eligibility trace'; since in our problem the onset and offset times of the stimulus are not known, once again we resort to the low-pass filter of its time-derivative,

$$\tau_M \dot{E}_i^\nu = -E_i^\nu + \frac{d}{dt} \frac{\partial}{\partial w_i^\nu} \mathcal{L}_{\mathbf{w}}(Y^\nu|\mathbf{X}), \tag{10}$$

where, from Eq. 9,

$$\frac{d}{dt} \frac{\partial}{\partial w_i^\nu} \mathcal{L}_{\mathbf{w}}(Y^\nu|\mathbf{X}) = \beta \left(\sum_{t^\nu \in Y^\nu} \delta(t - t^\nu) - \phi(u^\nu(t)) \right) PSP_i(t, \mathbf{X}). \tag{11}$$

With these quantities in hand, an online learning rule performing gradient ascent in the average reward would read

$$\frac{dw_i^\nu}{dt} = \eta R_t E_i^\nu, \tag{12}$$

where η is the learning rate and R_t is the reward at time t. However, to alleviate the spatial credit assignment problem, we follow Urbanczik and Senn (2009) and replace the global reward with an individualized reward signal r^ν that equals 1 if neuron ν took the right decision, and equals -1 otherwise (see subsection 'Individualized reward' below for its biological implementation). Specifically, we used the learning rule

$$\frac{dw_i^\nu}{dt} = \eta |R_t| a(\tilde{P}_s^\mu)(r^\nu - 1) E_i^\nu, \tag{13}$$

for each postsynaptic neuron ν belonging to population μ responsible for the current decision. The factor $a(\tilde{P}_s^\mu)$ is an attenuation factor equal to 1 (no attenuation) for an incorrect decision ($R_t < 0$) and equal to $|\tilde{P}_s^\mu|$ for a correct decision ($R_t > 0$), with $\tilde{P}_s^\mu$ given by Eq. 3. Since R_t is different from zero only around a decision time (when reward is given; see subsection 'Individualized reward' below), the factor $|R_t|$ confines the synaptic update to a temporal window around reward delivery. In all numerical experiments reported here, $\eta = 10$.

Before describing each term of the learning rule in detail, we make two observations:

(i) In the case of episodic learning, the learning rule Eq. 13 performs stochastic gradient ascent in a monotonic function of reward and population activity, as proved by Urbanczik and Senn (2009). This learning rule can be understood as an improvement over Williams's general gradient learning rule, Eq. 12. The need to introduce the individualized reward signal r^ν arises because otherwise learning worsens as the population size increases (Urbanczik and Senn, 2009). Instead, learning through Eq. 13 speeds up with population size, as demonstrated in (Urbanczik and Senn, 2009) for more standard tasks and in this work (Fig. 3c) for the combined segmentation/decision-making task.

(ii) The right hand side of Eq. 11 contains the product of a postsynaptic term (the difference between the actual firing rate $\sum_{t^\nu} \delta(t - t^\nu)$ and the average firing rate $\phi(u^\nu)$ of neuron ν), and a presynaptic term (the postsynaptic potential PSP_i). Together with the reward-modulated term in Eq. 13, the latter is a three-factor synaptic plasticity rule as found e.g. in cortico-striatal synapses, a major locus for reinforcement learning (Reynolds and Wickens, 2002).

Learning attenuation. According to Eq. 13, only the synapses targeting neurons voting for the wrong decision ($r^\nu = -1$) are updated after a decision. Moreover, the update is full in case of an incorrect decision and attenuated by a factor $|\tilde{P}_s^\mu| \approx e^{-(P_s^\mu(t_d^*) - \Theta_D)^2/\Theta_D^2}$ in case of a correct decision. The rationale for this choice is the following: if P_s^μ is large or small compared to Θ_D soon after a *correct* decision, the population is highly confident of its vote and no further learning is required (and indeed in this case $a(\tilde{P}_s^\mu) = |\tilde{P}_s^\mu| \approx 0$ in Eq. 13); otherwise, if $P_s^\mu(t_d^*) \approx \Theta_D$, the population is maximally undecided about the current decision, and full learning is required ($a(\tilde{P}_s^\mu) \approx 1$).

Individualized reward. The individualized reward signal could be made available locally at each synapse by broadcasting feedback from the population activity via three neuromodulatory signals: 1) $\tilde{P}_s^\mu$ (representing the deviation from equilibrium of the concentration of a neurotransmitter such as acetylcholine or serotonin), 2) each neuron's own activity S_t^ν (e.g., through intracellular calcium transients), and 3) the global reward feedback R_t (e.g., through extracellular dopamine). Using these three signals, an online estimate of the individualized reward signal for the νth neuron in population μ, to be used in Eq. 13, was computed as

$$r^\nu = \text{sign}(R_t \tilde{P}_s^\mu (S_t^\nu - \theta_S)), \tag{14}$$

where $\theta_S = e^{-1.1}$ is a threshold for S_t^ν. The reward signal R_t was transiently driven by the global reward R for 50 ms after each decision time t_d^*, and then decayed to zero exponentially with time constant of 10 ms. S_t^ν was given by

$$\dot{S}_t^\nu = -\frac{S_t^\nu}{\tau_S} + (1 - S_t^\nu) \sum_{t^\nu} \delta(t - t^\nu), \tag{15}$$

where the $\{t^\nu\}$ are the neuron's output spike times, and $\tau_S = 500$ms. If, for example, the decision was correct ($R_t > 0$) and population μ was above threshold ($\tilde{P}_s^\mu > 0$), the neurons of the population which had been sufficiently active ($S^\nu > \theta_S$) had voted for the correct decision, hence for them $r^\nu = 1$, whereas those neurons that had not been active ($S^\nu < \theta_S$) had voted against the correct decision, and for them $r^\nu = -1$. One can similarly work out all other possible scenarios and confirm that the Eq. 14 indeed provides the correct feedback in each case.

Homeostatic mechanism. Neurons' activities were kept close to the decision threshold via a homeostatic mechanism. This was implemented via the update of the parameter λ in Eq. 6 controlling the neurons' spike probabilities. Specifically, λ^μ appearing in Eq. 6 (the same for all neurons in population μ) was changed according to $\lambda^\mu \to \lambda^\mu + \Delta\lambda^\mu$, with

$$\Delta\lambda^\mu = \tilde{\eta}(\Theta_D - f_t^\mu), \tag{16}$$

where f_t^μ is a running average of the spike count similar to P_s^μ (Eq. 2) but without post-decision reset and a much longer time constant (4 seconds). The update rate $\tilde{\eta}$ was 0.01 in all computer experiments. The update Eq. 16 occurred at random intervals distributed according to a uniform distribution between zero and 2 seconds. In between updates, λ^μ was kept fixed not to interfere with learning the weights. Note that λ^μ is constrained to be positive. Other update schedules gave similar results and in general the mechanism was very robust to the mean length of the interval between updates or the interval length distribution, see Fig. S6. Mean inter-update times as short as 50 ms also led to successful learning (Fig. S6). Deterministic schedules wherein λ^μ was updated at regular intervals (or every fixed number of presentations) worked just as well, however they require knowledge of the time (or the number of stimulus presentations) elapsed since the last update, and were dismissed as less biologically plausible. This homeostatic mechanism is beneficial because, as learning modifies the weights, it also induces changes in population activity. These may occasionally produce very low activity in some populations, resulting in lack of decisions for those populations. When this occurs, the agent is trapped in a local maximum of the average reward, and learning practically stops for that population. In such a case, the homeostatic mechanism slowly increases the firing probability in single neurons until the desired population activity level is restored and decisions are again possible.

4.4 Stimuli and computer experiments

Main stimuli. The stimuli were jittered versions of prototypical spatiotemporal patterns of 50 Poisson spike trains with constant firing rates sampled from a uniform distribution between 2 and 24 spikes/s. In Fig. S3, the number of spike trains was varied from 25 to 200. Note that the

choice of a Poisson process is convenient but not strictly necessary, i.e., other distributions could have been used instead, including the empirical distribution of cortical data (Oram et al., 1999; Wiener and Richmond, 2002). With the exception of Fig. S4, all-to-all connectivity between input spike trains and decision neurons was used. Each stimulus presentation lasted for an average 500 ms (with small variations due to spike times jittering, see below), except when removed early as a consequence of a decision (learning was robust to variations in stimulus durations). Stimuli prototypes were initially generated and then, prior to each presentation, temporal jitter was added to the spike times to produce a noisy version of the prototypes. For each spike time t_{sp}, jitter was added by sampling from a Gaussian distribution centered on t_{sp} and having a standard deviation of $\sigma = 10$ ms (Fig. 1**b**), unless specified otherwise to test for robustness to larger values of jitter (Fig. S2). Stimuli were presented as a continuous input stream while learning occurred online as described in previous sections. The majority of the stimuli (a 3:1 ratio) were always defined as 'non-relevant' and the agent had to learn to ignore them not to incur the small cost attached to any decision (whether correct or incorrect; see Sec. 4.1). Different ratios of relevant to non-relevant stimuli were used to test robustness as shown in Fig. S1. Contrary to standard reinforcement learning methods, decisions were not enforced at stimulus offset. Thus, in the absence of decisions, no learning would be possible.

Firing rate and spike-timing coding. For the results presented in Fig. 3d, stimuli were encoded by either firing rates only, or by spike timing only. In the former case, the prototypes were generated as for the main stimuli, but at each new presentation the spike trains were generated anew. This is akin to having a jitter in the spike times covering the entire duration of the stimuli. In spike-timing coding, the spikes trains of the prototypes had all the same firing rate of 12 spikes/s; the spike times were then jittered in time by $\sigma = 10$ or 100 ms before each new presentation.

Stimuli with specified pair-wise correlations. Spike trains with desired firing rates and pair-wise correlations were generated by using the dichotomized Gaussian model (Macke et al., 2009). The same distribution of firing rates as for the main stimuli was used, with a desired pair-wise correlation of 0.1, 0.3 or 0.5 (see Fig. S5 for examples). One hundred training sets were generated for each correlation level, and training proceeded as for the main stimuli. In particular, each presented stimulus was a jittered version of its prototype, with temporal jitter of $\sigma = 10$ ms.

Learning stimuli from cortical datasets. Cortical data were obtained from multi-electrode recordings of single units as described in (Samuelsen et al., 2012). Briefly, Long-Evans rats were trained to press a lever within two seconds of hearing a pure tone to obtain one of 4 tastes delivered directly into their mouth through an intra-oral cannula (this procedure allows high temporal resolution on stimulus delivery). Tastes and concentrations were: 100 mM sodium chloride, 100 mM sucrose, 100 mM citric acid, and 1 mM quinine. Water (50 μl) was delivered to rinse the mouth 5 seconds after the delivery of each tastant. Unexpectedly to the rats, the same tastants would also be passively delivered at random times during the experiment. In each session, rats performed 7 trials with passively delivered stimuli, and 7 trials with expected stimuli, for a total of 28 passively delivered tastants

and 28 expected tastants. Multiple single-unit recordings from the primary gustatory cortex were collected while the rats performed the task. Neurons with firing rates lower than 2 spikes/s and neurons exhibiting a large peak around the 6-10 Hz in the power spectrum of their spike trains, potentially reflecting the motor response (Katz et al., 2001; Jezzini et al., 2013; Horst and Laubach, 2013), where excluded. The remaining 101 neurons would form the patterns shown in Fig. 5a and were used to generate the stimulus prototypes as follows.

For each training set, we randomly sampled 50 input neurons out of the 101 to obtain each prototype (sampling with replacement for each prototype; see Fig. 5b). To build relevant stimuli, we took the data falling in a temporal window between zero and 1 s after the passive delivery of a tastant; for non relevant stimuli, we took data falling in a 1 s window ending 1 second before the passive delivery of a tastant (during the so-called 'ongoing activity', Mazzucato et al., 2015). Due to long and random inter-trial intervals following water rinse (averaging about 40s), it is reasonable to assume that no residual trace of the previous stimulus was present at the time of a new delivery. Hence, the data prior to stimulus delivery can be considered non-stimulus related and therefore suitable to represent non-relevant stimuli. As with the main stimuli, we presented the stimuli for an average duration of 500 ms, with the spike times being jittered by 10 ms before each presentation. For each stimulus, we had 7 prototypes, corresponding to the 7 experimental trials with a given tastant; the 7 prototypes were considered as relevant stimuli of the same class (i.e., demanding the same decision; two prototypes for sucrose are shown in Fig. 5b). In each simulated decision task of Fig. 5c, we used 1 relevant segment for each class (say, sucrose and citric acid from trial 1 for the binary decision task), and 3 times as many non-relevant segments, built as described above and randomly selected from the pool of all non-relevant segments. We trained the model on 100 sets; for each new training set, we resampled the 50 neurons and repeated the procedure just described.

After training, we tested the ability of the model to generalize to unseen stimuli of the same class. Specifically, for each training set, we had the agent perform the same task, with plasticity switched off, using unseen data from trials 2 to 7 of the same class as relevant stimuli. For example, if the training stimulus was in the class of sucrose, we tested the model on sucrose prototypes not used during training. These relevant segments were pitted against unseen instances of non-relevant stimuli. Each generalization test included 10 presentations per stimulus. In the binary task with 6 relevant stimuli per class (see the last row in Table S1), generalization was tested on the remaining unseen stimulus of each class.

To establish the significance of the generalization performance, we compared the latter against the performance in a 'control' task which used instances of the remaining tastants as the relevant segments. For example, if sucrose and citric acid had been used for training in the binary decision task, sodium chloride and quinine were used for testing. For the 4-way decision task, in which all classes had been used for training, unseen instances of class 2, 3 and 4 were used as instances of class 1; unseen stimuli of class 1, 3 and 4 were used for class 2, and so on. Note that not only these stimuli had not been used during training, but they had been obtained in response to different tastants during the experiments. Therefore, performance in the control task was taken as a proxy for chance performance on unseen stimuli. The generalization perfor-

mance across 100 validation sets (one for each training set) was compared with the performance in the corresponding control task using a one-tailed Wilcoxon rank sum test (Table S1).

Computer experiments. Learning experiments were run in Matlab using custom computer code. A first-order Euler scheme was used for the numerical integration of differential equations with an elementary time step $dt = 0.2$ ms (smaller values of dt had similar effects to increasing the learning rate and did not qualitatively modify our results; not shown). The membrane potentials were initially set to their resting values. Initial synaptic values were sampled from a Gaussian distribution with zero mean and standard deviation of 2. The time course of learning performance was quantified as the mean fraction f_r of correct decisions for relevant stimuli, and as the mean fraction $f_{d,nr}$ of decisions in the presence of non-relevant stimuli. Examples of f_r and $f_{d,nr}$ are shown in Fig. 3; note that perfect performance is achieved when $f_r = 1$ and $f_{d,nr} = 0$. Prior to plotting in Fig. 3, both measures of performance were averaged across multiple training sets. Each training set comprised newly randomly generated prototype stimuli and synaptic weights. After averaging, learning curves were low-pass filtered according to $\bar{f}_n = (1 - 0.01)\bar{f}_{n-1} + 0.01 f_n$, where f_n is the performance during stimulus exposure # n and $\bar{f}_n$ its running average up to exposure # n.

To ensure a fair comparison between learning curves displayed in the same figure, the initial value of the spike probability λ_0 was selected so as to produce the same initial rate of decisions in all learning problems. This was particularly relevant when modifying the number of decision neurons (Fig. 3b-c), the number of input spike trains in each stimulus (Fig. S3), the level of input connectivity (Fig. S4), or the amount of pair-wise correlations (Fig. S5). The learning rate was kept the same across tasks.

References

Abeles M, Bergman H, Gat I, Meilijson I, Seidemann E, Tishby N, Vaadia E (1995) Cortical activity flips among quasi-stationary states. *Proc. Natl. Acad. Sci. U. S. A.* 92: 8616–8620.

Averbeck BB, Latham PE, Pouget A (2006) Neural correlations, population coding and computation. *Nat Rev Neurosci* 7: 358–366.

Bogacz R, Brown E, Moehlis J, Holmes P, Cohen JD (2006) The physics of optimal decision making: a formal analysis of models of performance in two-alternative forced-choice tasks. *Psychol Rev* 113: 700–65.

Britten KH, Shadlen MN, Newsome WT, Movshon JA (1992) The analysis of visual motion: a comparison of neuronal and psychophysical performance. *J Neurosci* 12: 4745–65.

Celeux G, Diebolt J (1985) The sem algorithm: a probabilistic teacher algorithm derived from the em algorithm for the mixture problem. *Computational Statistics Quarterly* 2: 2: 73–82.

Chen Z (2013) An overview of bayesian methods for neural spike train analysis. *Comput Intell Neurosci* 2013: 251905.

Churchland AK, Kiani R, Shadlen MN (2008) Decision-making with multiple alternatives. *Nat Neurosci* 11: 693–702.

Churchland MM, Yu BM, Cunningham JP, Sugrue LP, Cohen MR, Corrado GS, Newsome WT, Clark AM, Hosseini P, Scott BB, Bradley DC, Smith MA, Kohn A, Movshon JA, Armstrong KM, Moore T, Chang SW, Snyder LH, Lisberger SG, Priebe NJ, Finn IM, Ferster D, Ryu SI, Santhanam G, Sahani M, Shenoy KV (2010) Stimulus onset quenches neural variability: a widespread cortical phenomenon. *Nat Neurosci* 13: 369–78.

Corrado G, Doya K (2007) Understanding neural coding through the model-based analysis of decision making. *J. Neurosci.* 27: 8178–8180.

Dayan P, Abbott LF (2001) *Theoretical neuroscience: computational and mathematical modeling of neural systems.* Massachusetts Institute of Technology Press, Cambridge, Mass.

Dayan P, Daw ND (2008) Decision theory, reinforcement learning, and the brain. *Cogn Affect Behav Neurosci* 8: 429–53.

Doiron B, Litwin-Kumar A, Rosenbaum R, Ocker GK, Josić K (2016) The mechanics of state-dependent neural correlations. *Nat Neurosci* 19: 383–93.

Fawcett T (2006) An introduction to ROC analysis. *Pattern Recognition Letters* 27: 861–874.

Fiete IR, Fee MS, Seung HS (2007) Model of birdsong learning based on gradient estimation by dynamic perturbation of neural conductances. *J Neurophysiol* 98: 2038–57.

Frémaux N, Sprekeler H, Gerstner W (2010) Functional requirements for reward-modulated spike-timing-dependent plasticity. *J Neurosci* 30: 13326–37.

Friedrich J, Urbanczik R, Senn W (2010) Learning spike-based population codes by reward and population feedback. *Neural Comput* 22: 1698–717.

Friedrich J, Urbanczik R, Senn W (2011) Spatio-temporal credit assignment in neuronal population learning. *PLoS Comput Biol* 7: e1002092.

Friedrich J, Urbanczik R, Senn W (2014) Code-specific learning rules improve action selection by populations of spiking neurons. *Int J Neural Syst* 24: 1450002.

Ganmor E, Segev R, Schneidman E (2011) Sparse low-order interaction network underlies a highly correlated and learnable neural population code. *Proc Natl Acad Sci U S A* 108: 9679–84.

Genovesio A, Brasted PJ, Mitz AR, Wise SP (2005) Prefrontal cortex activity related to abstract response strategies. *Neuron* 47: 307–20.

Gershman SJ, Blei DM (2012) A tutorial on bayesian nonparametric models. *Journal of Mathematical Psychology* 56: 1 – 12.

Gold JI, Shadlen MN (2007) The neural basis of decision making. *Annu Rev Neurosci* 30: 535–74.

Green D, Swets J (1966) *Signal detection theory and psychophysics.* Wiley: New York.

Gütig R (2016) Spiking neurons can discover predictive features by aggregate-label learning. *Science* 351: aab4113.

Gütig R, Sompolinsky H (2006) The tempotron: a neuron that learns spike timing-based decisions. *Nat Neurosci* 9: 420–8.

Hampton AN, Bossaerts P, O'Doherty JP (2006) The role of the ventromedial prefrontal cortex in abstract state-based inference during decision making in humans. *J Neurosci* 26: 8360–7.

Hartley T, Lever C, Burgess N, O'Keefe J (2014) Space in the brain: how the hippocampal formation supports spatial cognition. *Philos Trans R Soc Lond B Biol Sci* 369: 20120510.

Holt G, Softky W, Koch C, Douglas R (1996) Comparison of discharge variability in vitro and in vivo in cat cortex neurons. *J. Neurophysiol.* 75: 1806–1814.

Horst NK, Laubach M (2013) Reward-related activity in the medial prefrontal cortex is driven by consumption. *Front Neurosci* 7: 56.

Izhikevich EM (2007) Solving the distal reward problem through linkage of STDP and dopamine signaling. *Cereb Cortex* 17: 2443–52.

Jezzini A, Mazzucato L, La Camera G, Fontanini A (2013) Processing of hedonic and chemosensory features of taste in medial prefrontal and insular networks. *J Neurosci* 33: 18966–78.

Jones LM, Fontanini A, Sadacca BF, Miller P, Katz DB (2007) Natural stimuli evoke dynamic sequences of states in sensory cortical ensembles. *Proc Natl Acad Sci U S A* 104: 18772–7.

Kappel D, Nessler B, Maass W (2014) STDP installs in winner-take-all circuits an online approximation to hidden markov model learning. *PLoS Comput Biol* 10: e1003511.

Katz DB, Simon SA, Nicolelis MA (2001) Dynamic and multimodal responses of gustatory cortical neurons in awake rats. *J Neurosci* 21: 4478–89.

La Camera G, Bouret S, Richmond BJ (2018) Contributions of lateral and orbital frontal regions to abstract rule acquisition and reversal in monkeys. *Front Neurosci* 12: 165.

Lee D, Wang XJ (2009) *Mechanisms for Stochastic Decision Making in the Primate Frontal Cortex: Single-neuron Recording and Circuit Modeling*, chapter 31, pp. 481–502. Academic Press, London, UK.

Maboudi K, Ackermann E, de Jong LW, Pfeiffer BE, Foster D, Diba K, Kemere C (2018) Uncovering temporal structure in hippocampal output patterns. *Elife* 7.

Macke JH, Berens P, Ecker AS, Tolias AS, Bethge M (2009) Generating spike trains with specified correlation coefficients. *Neural Comput* 21: 397–423.

Mansouri FA, Freedman DJ, Buckley MJ (2020) Emergence of abstract rules in the primate brain. *Nat Rev Neurosci* 21: 595–610.

Mazzucato L, Fontanini A, La Camera G (2015) Dynamics of multistable states during ongoing and evoked cortical activity. *J Neurosci* 35: 8214–31.

Mazzucato L, La Camera G, Fontanini A (2019) Expectation-induced modulation of metastable activity underlies faster coding of sensory stimuli. *Nat Neurosci* 22: 787–796.

Mazzucato L, Fontanini A, La Camera G (2016) Stimuli reduce the dimensionality of cortical activity. *Front. Syst. Neurosci.* 10: 11.

Mongillo G, Deneve S (2008) Online learning with hidden markov models. *Neural Comput* 20: 1706–16.

Montani F, Ince RAA, Senatore R, Arabzadeh E, Diamond ME, Panzeri S (2009) The impact of high-order interactions on the rate of synchronous discharge and information transmission in somatosensory cortex. *Philos Trans A Math Phys Eng Sci* 367: 3297–310.

Neal RM, Hinton GE (1998) *A view of the EM algorithm that justifies incremental sparse, and other variants. In: Learning in Graphical Models*, pp. 355–368. Kluwer Academic Press.

Ohiorhenuan IE, Mechler F, Purpura KP, Schmid AM, Hu Q, Victor JD (2010) Sparse coding and high-order correlations in fine-scale cortical networks. *Nature* 466: 617–21.

O'Keefe J, Recce ML (1993) Phase relationship between hippocampal place units and the eeg theta rhythm. *Hippocampus* 3: 317–30.

Oram M, Wiener M, Lestienne R, Richmond B (1999) Stochastic nature of precisely timed spike patterns in visual system neural responses. *J. Neurophysiol.* 81: 3021–3033.

Park IM, Meister MLR, Huk AC, Pillow JW (2014) Encoding and decoding in parietal cortex during sensorimotor decision-making. *Nat Neurosci* 17: 1395–403.

Pfister JP, Toyoizumi T, Barber D, Gerstner W (2006) Optimal spike-timing-dependent plasticity for precise action potential firing in supervised learning. *Neural Comput* 18: 1318–48.

Ponce-Alvarez A, Nácher V, Luna R, Riehle A, Romo R (2012) Dynamics of cortical neuronal ensembles transit from decision making to storage for later report. *J. Neurosci.* 32: 11956–11969.

Rabiner L (1989) A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE* 77: 257–286.

Reynolds JN, Wickens JR (2002) Dopamine-dependent plasticity of corticostriatal synapses. *Neural Netw* 15: 507–521.

Rigotti M, Ben Dayan Rubin D, Morrison SE, Salzman CD, Fusi S (2010a) Attractor concretion as a mechanism for the formation of context representations. *Neuroimage* 52: 833–47.

Rigotti M, Ben Dayan Rubin D, Wang XJ, Fusi S (2010b) Internal representation of task rules by recurrent dynamics: the importance of the diversity of neural responses. *Front Comput Neurosci* 4: 24.

Saito T, Rehmsmeier M (2015) The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. *PLoS One* 10: e0118432.

Samuelsen C, Gardner M, Fontanini A (2012) Effects of cue-triggered expectation on cortical processing of taste. *Neuron* 74: 410–22.

Schneidman E, Berry n MJ, Segev R, Bialek W (2006) Weak pairwise correlations imply strongly correlated network states in a neural population. *Nature* 440: 1007–12.

Schultz W (1998) Predictive Reward Signal of Dopamine Neurons. *J. Neurophysiol.* 80: 1–27.

Seung HS (2003) Learning in spiking neural networks by reinforcement of stochastic synaptic transmission. *Neuron* 40: 1063–1073.

Shadlen M, Newsome W (1994) Noise, neural codes and cortical organization. *Current Opinion in Neurobiology* 4: 569–579.

Shadlen M, Newsome W (1998) The variable discharge of cortical neurons: implications for connectivity, computation and information coding. *J. Neurosci.* 18: 3870–3896.

Shimazaki H, Sadeghi K, Ishikawa T, Ikegaya Y, Toyoizumi T (2015) Simultaneous silence organizes structured higher-order interactions in neural populations. *Sci Rep* 5: 9821.

Skaggs WE, McNaughton BL, Wilson MA, Barnes CA (1996) Theta phase precession in hippocampal neuronal populations and the compression of temporal sequences. *Hippocampus* 6: 149–72.

Sutton RS, Barto AG (2018) *Reinforcement learning: an introduction* Adaptive computation and machine learning series. MIT Press, Cambridge, MA, second edition edition.

Tanji J, Shima K, Mushiake H (2007) Concept-based behavioral planning and the lateral prefrontal cortex. *Trends Cogn Sci* 11: 528–34.

Urbanczik R, Senn W (2009) Reinforcement learning in populations of spiking neurons. *Nat Neurosci* 12: 250–2.

Wang XJ (2008) Decision making in recurrent neuronal circuits. *Neuron* 60: 215–34.

Werfel J, Xie X, Seung HS (2005) Learning curves for stochastic gradient descent in linear feedforward networks. *Neural Comput* 17: 2699–718.

Wiener MC, Richmond BJ (2002) Model based decoding of spike trains. *Biosystems* 67: 295–300.

Williams RJ (1992) Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning* 8.

Xie X, Seung HS (2004) Learning in neural networks by reinforcement of irregular spiking. *Phys Rev E Stat Nonlin Soft Matter Phys* 69: 041909.

Yu AJ, Dayan P (2002) Acetylcholine in cortical inference. *Neural Netw* 15: 719–30.

Yu S, Yang H, Nakahara H, Santos GS, Nikolić D, Plenz D (2011) Higher-order interactions characterized in cortical activity. *J Neurosci* 31: 17514–26.

Zeithamova D, Mack ML, Braunlich K, Davis T, Seger CA, van Kesteren MTR, Wutz A (2019) Brain mechanisms of concept learning. *J Neurosci* 39: 8259–8266.

Zeithamova D, Maddox WT, Schnyer DM (2008) Dissociable prototype learning systems: evidence from brain imaging and behavior. *J Neurosci* 28: 13194–201.

Zhang J, Bogacz R, Holmes P (2009) A comparison of bounded diffusion models for choice in time controlled tasks. *J Math Psychol* 53: 231–241.

Zohary E, Shadlen MN, Newsome WT (1994) Correlated neuronal discharge rate and its implications for psychophysical performance. *Nature* 370: 140–3.

5 Supplementary Figures

5.1 Supplementary Figure 1: sparse stimulus embedding

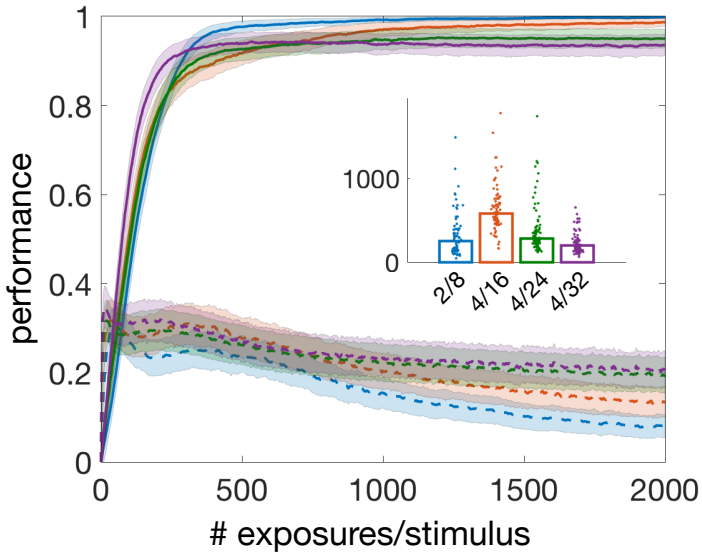

Figure S1: Average learning curves for varying numbers of stimuli and varying ratios of non-relevant to relevant stimuli in the binary decision task (same keys as Fig. 3**b**). We trained the model with 16 (red), 24 (green) and 32 stimuli (purple). In each case, 4 stimuli were relevant (2 stimuli in each decision class), giving rise to 4:1, 6:1 and 8:1 ratios of non-relevant to relevant stimuli. The case with 8 stimuli of which 2 relevant is the same as shown in blue in Fig. 3**b** and is shown here for comparison (blue curves). Performance slightly decreased as the number of stimuli increased, but was high in all cases. We used $N = 100$ decision neurons except in the case of 32 stimuli (purple), where $N = 200$ decision neurons were used to boost performance. Mean performance with relevant stimuli raised from 85% when $N = 100$ to 94% with $N = 200$; no difference was found for non-relevant stimuli. Performance was computed in the last 1% trials of each training session and then averaged across 100 training sets for each task. Notice the faster initial learning speed with $N = 200$ due to a larger population size despite a greater number of stimuli (see Fig. 3**c**). *Inset:* Dotplots of learning thresholds for each task, with: '2/8' = 2 relevant stimuli out of 8; '4/16' = 4 relevant stimuli out of 16; and so on. Learning thresholds were reached in $100, 96, 88$ and 83 training sets out of 100 for the task with $8, 16, 24$ and 32 stimuli, respectively.

5.2 Supplementary Figure 2: robustness to spike time jitter

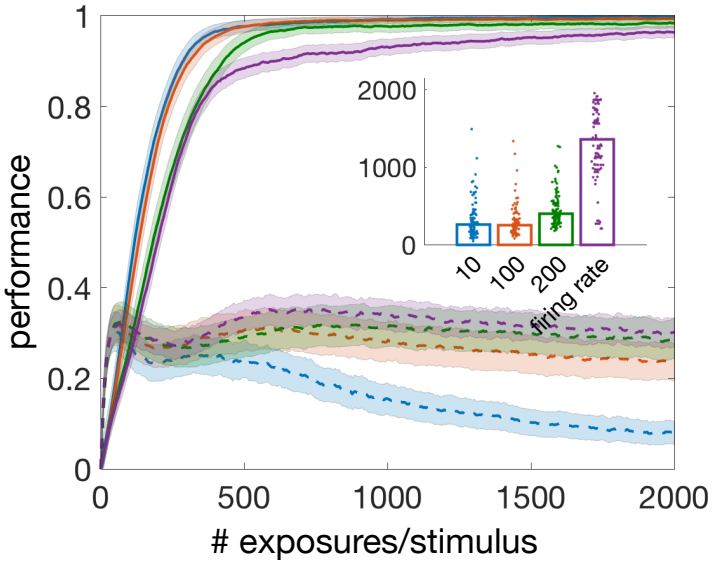

Figure S2: Learning curves for varying jitter in the spike times of the input spike trains for the main binary decision task of Fig. 3**b** (same keys as Fig. 3**b**). Shown are the learning curves for a level of jitter $\sigma = 10$ ms (main stimuli; here in blue), 100 (red), 200 (green) and firing rate coding (magenta) equivalent to a jitter of 500 ms (see the main text). In firing rate coding, the spike times were generated anew at each stimulus presentation. In each case we averaged across 100 learning curves except for firing rate coding where we averaged across 200 learning curves (magenta). *Inset:* Dotplots of learning thresholds for each task, with: 10, 100 and 200 being the value of σ in ms. Learning thresholds were always reached for $\sigma \leq 200$ ms, and were reached in 65% of the training sets in the firing rate coding case. These results show that the model is tolerant to a high level of jitter in the spike times, including jitter extending over the entire duration of the stimuli. Note that 10 ms jitter and firing rate coding are, respectively, the easier and the most difficult tasks, and therefore their learning curves (blue and magenta) bound the learning curves obtained at intermediate levels of jitter.

5.3 Supplementary Figure 3: robustness to stimulus dimensionality

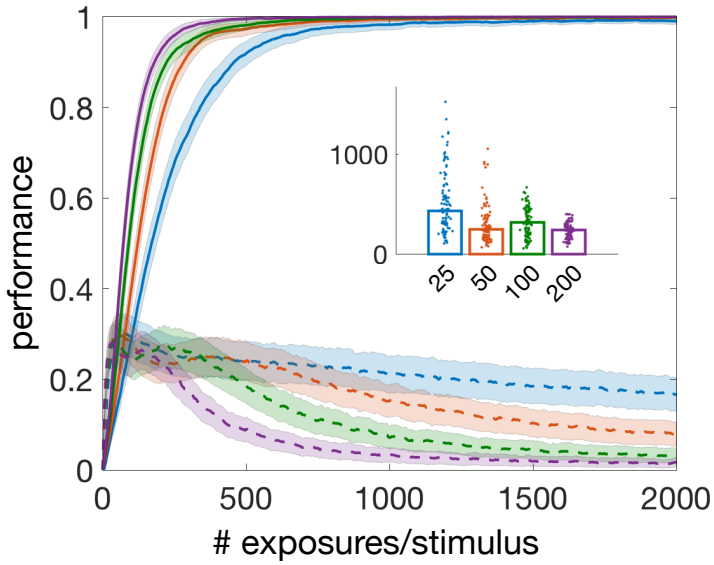

Figure S3: Learning curves for varying numbers of presynaptic spike trains for the main binary decision task of Fig. 3**b** (same keys as Fig. 3**b**). We used 25 (blue), 50 (red), 100 (green) and 200 (magenta) input spike trains and mean learning curves averaged across 100 training sets in each case. The initial value of the spike probability λ_0 (see Methods) was selected so as to produce an equal initial rate of decisions as the number of input spike trains varied. Final performance with relevant stimuli was high even with only 25 input spike trains, and overall learning performance improved with stimulus dimensionality. *Inset:* Dotplots of learning thresholds for each learning problem, with labels representing the number of input spike trains. Learning thresholds were reached for all training sets.

5.4 Supplementary Figure 4: robustness to reduced connectivity

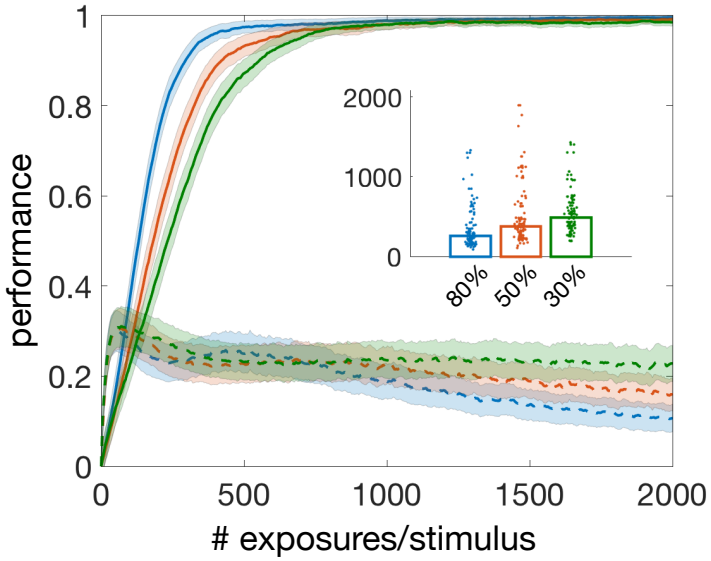

Figure S4: Learning curves for varying levels of input connectivity (average fraction of synaptic connections between input spike trains and decision neurons) for the main binary decision task of Fig. 3**b** (same keys as Fig. 3**b**). Connectivity values were 80% (blue), 50% (red) and 30% (green). The curve with full connectivity (blue curve in Fig. 3**b**) almost completely overlapped with the curve with 80% connectivity (not shown). To reduce the level of connectivity from 100% to $c\% < 100\%$, we removed a uniformly random fraction of $(100-c)/100$ synaptic projections to the decision neurons from the 50 input spike trains. This was done independently for each prototype before training started and was then kept fixed during training in each training set. Different randomly chosen synapses were removed in different training sets. The initial value of the spike probability λ_0 (see Methods) was selected so as to produce an equal initial rate of decisions as connectivity varied. Decision performance after training was high also when input connectivity was sparse. *Inset:* Dotplots of learning thresholds (labels represent levels of input connectivity). Learning thresholds were reached for all training sets.

5.5 Supplementary Figure 5: performance with correlated stimuli

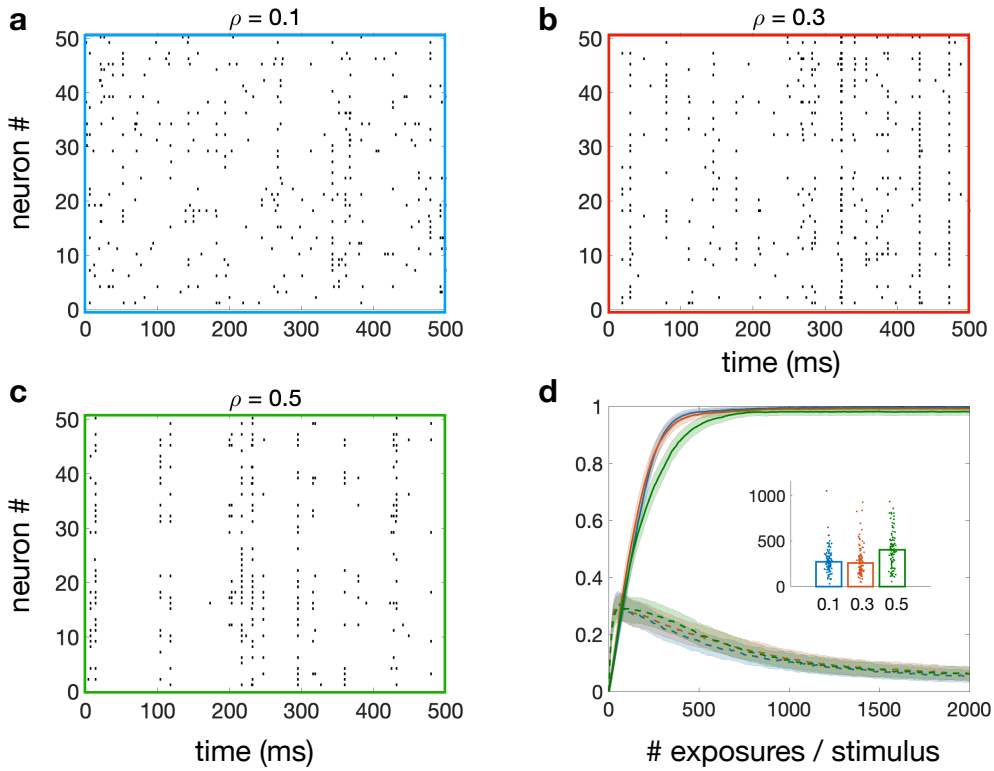

Figure S5: Stimuli and learning curves for the main binary decision task of Fig. 3**b** with correlated input stimuli. **a, b, c)** Examples of correlated stimuli with pair-wise correlation $\rho = 0.1, 0.3, 0.5$, respectively, and firing rates uniformly distributed between 2 and 24 spikes/s (same as for the default stimuli). The spike trains were generated with the method of the dichotomized Gaussian model (Macke et al., 2009) (see Methods). **d)** Average learning curves with correlated stimuli with $\rho = 0.1$ (blue), $\rho = 0.3$ (red) and $\rho = 0.5$ (green) for the task with 2 decisions and 2 relevant stimuli out of 8 (same keys as Fig. 3b). For each value of ρ, 24 prototypes were generated and on each of 100 training sets, 8 out of 24 prototypes were randomly selected to be the input prototypes, jittered by 10ms on each presentation as done with the main stimuli. *Inset:* Dotplots of learning thresholds (labels represent pair-wise correlation). Learning thresholds were reached for all training sets.

5.6 Supplementary Figure 6: robustness of the homeostatic mechanism

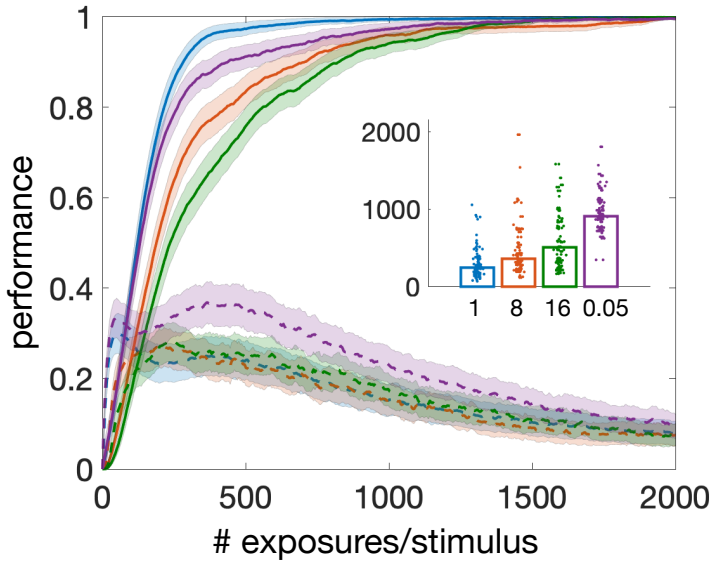

Figure S6: Learning curves for different *mean* update times T of the homeostatic mechanism in the 2-way decision task of Fig. 3**b**. In each case, the update times for the spike probability (Eq. 16 of the main text) were uniformly distributed between zero and $2T$. T was changed from $T = 50$ ms (magenta) to $T = 16$ s (green), with $T = 1$ s (blue) being the default value used in all other computer experiments presented in this paper. Shorter mean intervals resulted in faster learning times for T down to our default value of 1 s, however the performance at the end of training converged to the same level for all values of T. Learning with mean update times of 50 ms was slower than with $T = 1$ s, however, results could be improved by simultaneously decreasing the magnitude of change by decreasing the value of $\tilde{\eta}$ (Eq. 16 of the main text). For example, if $\tilde{\eta}$ was decreased to 0.001 from 0.01, both the learning speed and the statistics of the learning thresholds with $T = 50$ ms matched that obtained with $T = 1$ s and $\tilde{\eta} = 0.01$ (not shown). Alternative update schedules were also tested, based on: i) intervals drawn from a Gaussian distribution, ii) constant intervals, or iii) updates occurring exactly every 32 stimulus presentations. These alternative schedules also gave similar results (not shown), confirming the robustness of the homeostatic mechanism with regard to its specific implementation and timing. *Inset:* Dotplots of learning thresholds for each experiment, with T (in seconds) reported under each bar. Learning thresholds were reached for all training sets.

5.7 Supplementary Table 1: performance and generalization with cortical datasets

PERFORMANCE WITH CORTICAL DATASETS

D	**P** (P_{rel})	**N**	**training** rel. − non-rel. mean(SD) − mean(SD)	**generalization** rel. − non-rel. mean(SD) − mean(SD)	**control** rel. − non-rel. mean(SD) − mean(SD)
2	8 (2)	100	99(2) − 99(1)	48(21) − 98(3)	37(20) − 98(6)
3	12 (3)	150	98(4) − 96(2)	40(18) − 94(7)	34(11) − 94(7)
4	16 (4)	200	97(5) − 93(4)	38(13) − 84(15)	21(10) − 83(14)
2	18 (12)	300	93(3) − 92(7)	74(17) − 94(10)	54(12) − 93(9)

Table S1: Average performance for learning and generalization of the tasks with cortical data, where 'D' was the number of decisions, P was the number of stimuli, P_{rel} was the number of relevant stimuli, and 'N' was the number of decision neurons in each population. The learning curves of tasks with $(D, P) = (2, 8)$, $D = 3$ and $D = 4$ are plotted in Fig. 5 of the main text. The task with $D = 2$ and $P = 18$ was used to train on 6 stimuli per class and test on one unseen stimulus per class (see the main text for details). 'Training': performance after training (over the last 1% portion of all stimulus presentations) across 100 training sets, reported as mean ± SD in percentage units. 'Generalization': generalization performance to unseen stimuli of the same class across 100 validation sets, one for each training set used for 'training'. Each generalization run covered 10 presentations per stimulus, with learning and homeostasis disabled. 'Control': average generalization performance across 100 sets on unseen stimuli of different class, used as a measure of chance generalization performance (see Methods). Generalization performance on relevant stimuli was significantly higher on stimuli of the same class ('generalization') compared to stimuli of different class ('control'): p-values were $p = 1.67 \times 10^{-4}$ ($P = 8$), $p = 0.0073$ ($P = 12$), $p = 8.34 \times 10^{-18}$ ($P = 16$), and $p = 6.44 \times 10^{-15}$ ($P = 18$; Wilcoxon rank sum test, one-tailed).