Memory by accident: a theory of learning as a byproduct of network stabilization

Basile Confavreux
Gatsby Computational Neuroscience Unit
University College London
basile.confavreux@gmail.com

William Dorrell
Gatsby Computational Neuroscience Unit
University College London

Nishil Patel
Gatsby Computational Neuroscience Unit
University College London

Andrew Saxe
Gatsby Computational Neuroscience Unit
University College London

Abstract

Synaptic plasticity is widely considered to be crucial to the brain's ability to learn throughout life. Decades of theoretical work have therefore been invested in deriving and designing biologically plausible learning rules capable of granting various memory abilities to neural networks. Most of these theoretical approaches optimize directly for a desired memory function; but this procedure can lead to complex, finely-tuned rules, rendering them brittle to perturbations and difficult to implement in practice. Instead, we build on recent work that automatically discovers large numbers of candidate plasticity rules operating in recurrent spiking neural networks. Surprisingly, despite the fact that these rules are selected solely to achieve network stabilization, we observe across a range of network models—feedforward, recurrent; rate and spiking—that almost all these rules endow the network with simple forms of memory such as familiarity detection - seemingly by accident. To understand this phenomenon, we study an analytic toy model. We observe that memory arises from the degeneracy of weight matrices that stabilize a network: where the network lands in this space of stable weights depends on its past inputs—that is, memory. Even simple Hebbian plasticity rules can utilize this degeneracy, creating a zoo of memory abilities with various lifetimes. In practice, the larger the network and the more co-active plasticity rules in the system, the stronger the memory-by-accident phenomenon becomes. Overall, our findings suggest that activity-silent memory is a near-unavoidable consequence of stabilization. Simple forms of memory, such as familiarity or novelty detection, appear to be widely available resources for plastic brain networks, suggesting that they could form the raw materials that were later sculpted into higher-order cognitive abilities.

1 Introduction

Understanding how the brain orchestrates its plastic synapses is a holy grail of neuroscience, a potential stepping stone on the way to generalist models of brain function. Yet, despite concerted effort, empirical plasticity data remains scarce, largely due to the difficulty of recording synapses *in vivo* during learning. Consequently, theory and computation have played an out-sized role in developing and testing plasticity hypotheses [1]. Most theories derive their proposed learning rules by optimizing for some reasonable neural goals; for a memory system these might include the capacity, reliability or decodability of memories, while enforcing biological plausibility constraints such as

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

locality or slowly changing weights [2–5]. Rules so derived are promising candidates, but their fine-tuning to a specific function can make them brittle to perturbations and hard to implement in practice [6–8]. Besides, though frameworks analyzing the learning dynamics of spike-timing dependent plasticity rules at steady state in spiking networks exist—mainly mean-field methods [9–11]—it remains challenging to capture analytically the interactions between co-active plasticity rules operating at different synapse types in parallel, restricting most theories to study rules in isolation.

Instead, recent work has used numerical optimization to propose candidate learning rules, via meta-learning techniques [12–19]. This has led to the discovery of entire families of novel co-active learning rules able to robustly stabilize large recurrent spiking networks [19, 20]. Interestingly, despite being selected *only* to ensure that the spiking network stays stable, the majority of rules display a range of interesting memory behaviors, such as novelty detection, contextual novelty and replay [20]. It seems that basic memory abilities are a natural byproduct of network stabilization.

Here we seek to understand this link between stability and memory. By **stability**, we mean networks with "biologically plausible" activity and weight dynamics over a wide range of inputs - i.e. plausible asynchronous neural activities and slowly changing weights bounded within reasonable ranges - as in previous work [20]. We employ a broader definition of **memory** in this paper than, for example, associative memories in Hopfield networks [21]. Instead, we focus on familiarity/novelty detection: "the ability to discriminate between the relative familiarity or novelty of stimuli" [22], which has a long history in psychology, experimental and computational neuroscience [23, 22, 24]. This form of memory is simpler, and can exist without, for instance, pattern completion or an attractor state for each memory.

We study a range of network models—from large recurrent spiking to shallow linear feedforward rate networks—and show that the link is robust: across models, learning rules built to stabilize neural activity consistently encode memories with various lifetimes and properties. By reverse engineering spiking networks and studying analytic toy models, we show that these memories have a simple origin: for any input there is a degeneracy of stable weight matrices. Which matrix the learning rules select often depends on the network's history, providing the basis for memory storage and recall. Large-scale simulations suggest that the bigger the system and the more co-active learning rules, the longer-lasting and more robust the memories. The remarkable ease with which such memory abilities arise in stable spiking networks may form the basis of higher-order cognitive abilities as compositions of simpler, ubiquitous, memorization skills. As such, we present these ideas as a fresh take on the emergence of memory in the brain: memory by *accident*.

2 Memory is a common byproduct of stabilization by co-active plasticity rules

Our starting point was a manifold of co-active synaptic plasticity rules that enforce stable dynamics in large recurrent spiking networks. There are four types of synapses in these networks —Excitatory (E)-to-E, E-to-Inhibitory (I), I-to-E and I-to-I— each governed by its own plasticity rule. The rules are parameterized by their dependence on the pre-synaptic spike train, post-synpatic spike train and a Hebbian term dependent on both (fig. 1A [19, 20]):

$$\frac{\mathrm{d}w(t)}{\mathrm{d}t} = \eta[\underbrace{S_{\text{pre}}(t)\big(\alpha_{\text{XY}}}_{\text{Pre-synaptic}} + \underbrace{\kappa_{\text{XY}} x_{\text{post}}(t)\big)}_{\text{Hebbian}} + \underbrace{S_{\text{post}}(t)\big(\beta_{\text{XY}}}_{\text{Post-synaptic}} + \underbrace{\gamma_{\text{XY}} x_{\text{pre}}(t)\big)}_{\text{Hebbian}}], \quad \text{X, Y} \in \{\text{E, I}\} \tag{1}$$

where $S_{\text{pre}}(t)$ and $S_{\text{post}}(t)$ are spike trains, $x_{\text{pre}}(t)$ and $x_{\text{post}}(t)$ are their low-pass filtered versions. In sum, for each plasticity rule, there are 6 parameters, four as in eq. (1) ($\alpha, \beta, \gamma, \kappa$), and two timescales ($\tau^{\text{pre}}, \tau^{\text{post}}$), one for each low-pass filtering. Recent work meta-learned thousands of choices of these parameters that led to networks with stable dynamics – meaning the activities and weights in the networks remained in plausible ranges across many inputs for at least 4 hours [20]. This led to a "stability manifold", a stable subset of the 24 dimensional plasticity parameter space.

Despite selecting these rule quadruplets for stability, previous work observed that simple forms of memory were a near ubiquitous byproduct of network stabilization [20]. For example, the networks were tested on a familiarity detection task (fig. 1A): during a training period the networks were presented with a subset of the stimuli, then, after a variable time period, the network weights were frozen and the network was presented with both novel and familiar stimuli. The network was said to remember the familiar stimuli if the average firing rate was significantly different between the familiar and novel stimuli presentations. In this task almost all the rules showed some form of memory, with lifetimes ranging from seconds to hours (fig. 1B). In other words, most rules created memory traces

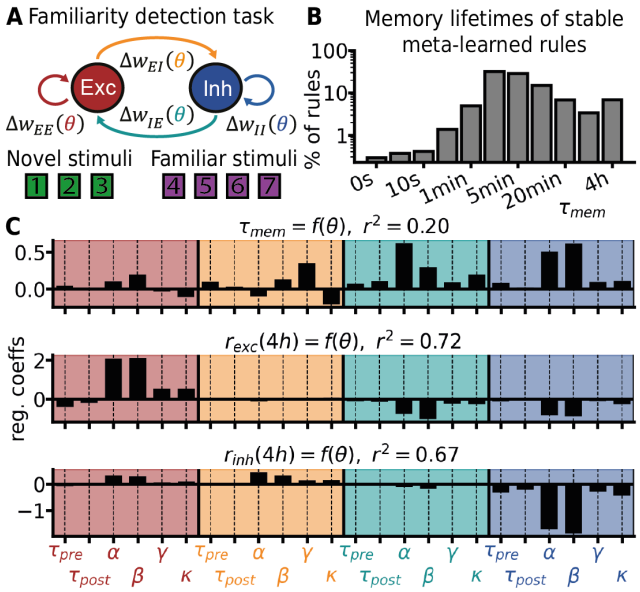


Figure 1: **A**: Recurrent spiking network ($N_{\mathrm{E}} = 4096, N_{\mathrm{I}} = 1024$) with four co-active plasticity rules undergoing a familiarity detection task. **B**: Distribution of elicited memory lifetime in the familiarity task among 2500 stable meta-learned rule quadruplets. Memory lifetime was defined as the last time point at which the network population firing rate was significantly different during novel vs familiar stimulus presentation (Student t-test, $p < 0.05$ for $N_{\mathrm{trials}} = 5$). Original data from [20]. **C**: Linear regressions, predicting from the values of the 24 plasticity parameters: the memory lifetime of each quadruplet on the familiarity task (top); the mean firing rate of the excitatory population after 4h (middle); the mean inhibitory firing rate after 4h (bottom). Dataset of 2500 rule quadruplets taken from [20]. While mean rates can be readily predicted from plasticity, memory lifetimes cannot.

with behaviorally-relevant lifetimes from a single stimulus presentation, despite having Hebbian pairing windows of tens to hundreds of milliseconds.

We wondered which features of the plasticity rules determined their memory abilities. Simply linearly predicting the memory lifetime from the plasticity parameters was impossible (fig. 1C) while, as a control, it was possible to predict other, more basic, network properties such as the excitatory and inhibitory mean firing rates using this strategy (fig. 1C). To understand this memory-by-accident phenomenon, we therefore take a different approach. In the following sections, we reverse engineer these memories across a range of models, including a simple analytic toy model, before returning to test the intuitions we develop on the full co-active spiking networks in the final section.

3 Reverse engineering memory by accident in spiking networks

3.1 Recurrent spiking network with four co-active rules

First, we investigated how the simple co-active rules from the stability manifold were both creating long-lived memories and enforcing stability in recurrent spiking networks. We focused on one rule quadruplet which responded significantly to the familiarity task for over 4h (fig. 2A). This quadruplet stabilized the network by driving it to an activity setpoint at around 2Hz during a pre-training phase of random background inputs (fig. 2A). The network was then perturbed by the input of the (not yet) familiar patterns, though the activity recovered to its pre-training setpoint less than a minute later (fig. 2A). However, probing the network with either familiar or novel stimuli elicited responses of different magnitudes (fig. 2A and fig. S2), and, as advertised, these differences persisted over the remaining hour of simulation. The meta-learned quadruplet thus elicited a form of long-lasting, activity-silent memory [25]. Further, unlike the network activity, neither the mean or variance of the weights recovered to their pre-training values (fig. 2A). This indicated that the weight matrices had experienced long-lasting changes that enabled the different network responses observed for familiar vs novel stimuli.

To analyze the weights in this network we defined the "engram" for each stimulus as the neurons with the highest 10% of activities in response to the stimuli. Since, by construction, even the neurons in naive networks were tuned to particular inputs (see fig. S2), this engram was meaningful even during pre-training phases. We observed that training mainly affected the weights to and from the familiar stimuli's engrams (fig. 2B, time point 2, strengthening of W_{IE} engram weights, and weakening of W_{EE} engram weights), and these familiar-specific changes persisted long after the end of training (fig. 2B, time point 3). We hypothesize that it is these familiar-specific weight changes that enabled long-term memory recall. In other words, among the many $(W_{\mathrm{EE}}, W_{\mathrm{EI}}, W_{\mathrm{IE}}, W_{\mathrm{II}})$ matrices able to stabilize the network in background state (the degeneracy of solutions to the stabilization problem), the specific weight matrices reached by plasticity reflected the system's history—here past input stimuli—giving rise to memory.

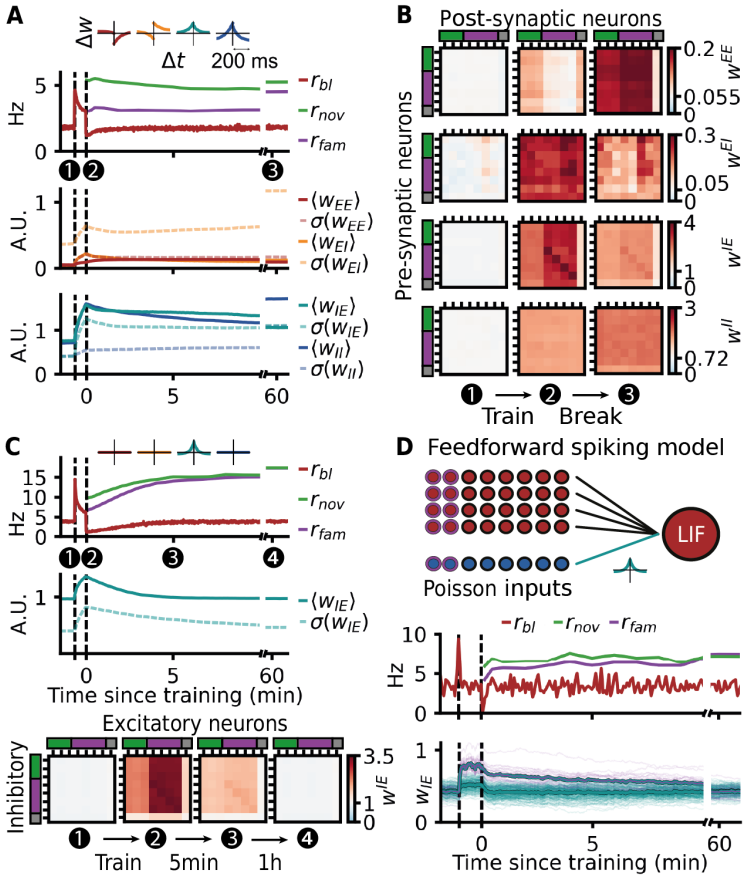


Figure 2: **A**: Example meta-learned rule quadruplet undergoing the familiarity task. Top: visualization of the rule quadruplet - normalised weight changes elicited by a pair of pre-synaptic and post-synaptic spikes with different time-lag between spikes. Second plot: excitatory population firing rate without any active stimuli (red, baseline) during the task, dashed lines denote the start and end of training. After training, average excitatory population firing rate in response to four novel (green) and three familiar stimuli (purple). Third & Fourth plot: Evolution of the mean and standard deviation of the weights during the task. **B**: Weight matrices of the network for each synapse type, at three time points: (1) immediately before the start of training, (2) immediately after training, (3) one hour after the end of training. We sort the weights according to neuron response class: green - neurons responding to novel patterns; purple - neurons responding to familiar patterns (shown during training); grey - all other neurons. **C**: Same as A&B, but for a recurrent spiking network with iSTDP (I-to-E only, [11]). **D**: Feedforward spiking network model. Excitatory and inhibitory Poisson inputs project on a single output neuron. The familiarity task followed a similar structure to the recurrent case. Middle: output neuron firing rate. Bottom: evolution of the 200 inhibitory (plastic) weights. Weights from input neurons belonging to the familiar stimulus are shown in purple. Means of the two groups ("familiar" weights vs rest) are in bold.

3.2 Recurrent spiking network with only I-to-E plasticity

We next sought to test our intuitions in a more tractable setting - recurrent spiking networks with a single operational plasticity rule. Within the stability manifold we recognized iSTDP, a widely-used rule for stabilising spiking networks [11, 26, 27]. This is a symmetric Hebbian rule operating only on the I-to-E connectivity that has been studied analytically (fig. 2C). Networks operating under this rule are known to settle to a stable mean firing rate whose value can be calculated using a mean-field approach [11].

We simulated networks evolving under iSTDP and observed very similar patterns to that of the long-term memory quadruplet: the activity settled both before and after training to a setpoint (3Hz). Further, despite showing no activity-related memory traces, an imprint of the familiar stimuli could be seen in the weights (fig. 2C), though in this case the memories lasted only a few minutes. It seemed that the network exhibited a separation of timescales: fast dynamics that restored the activity to the set point, and slower dynamics that erased the imprint of the memory from the connectivity weights (fig. 2C). This separation led to a period of time in which activity was flat yet the network was able to produce reliable memory-by-accident, matching our findings in the quadruplet rule case.

3.3 Feedforward spiking network with I-to-E plasticity

Finally, we study one further simplified spiking network: a feedforward network undergoing the familiarity detection task (fig. 2D). This model comprised a single leaky-integrate-and-fire output neuron receiving inputs from 800 excitatory and 200 inhibitory Poisson neurons. Only the inhibitory weights were plastic, following the iSTDP rule [11]. Initially, all input neurons fired at the same rate, then we performed stimulus presentation by elevating the activity of 100 excitatory and 25 inhibitory input neurons - the "engram neurons" in this setting (see Supplementary for more details).

Once again, transient memorization of the familiar stimulus could be seen (fig. 2D). Matching our observations in recurrent networks, we observed that weights from inhibitory engram neurons relaxed towards the background weight distribution at two different timescales after stimulus presentation. Fast dynamics with a time constant of a few seconds returned the output neuron's firing rate to the firing rate set point; while a slower relaxation, lasting from minutes to hours, erased the imprint of the memory from the weights. Since the only force that drove weight change in this network was deviation of the output neuron's firing rate from a target, these slower changes seemed to be driven by random fluctuations that slowly overshadowed the memory, leading to observable memory-by-accident phenomena persisting for behavioral timescales. We also verified that this insight was not specific to I-to-E plasticity by running a similar network and task with an E-to-E rule instead (fig. S5).

Overall, our analysis of each plastic spiking network confirmed the idea that the degeneracy of weight configurations capable of stabilizing the network provided the substrate for the formation of "accidental" memories. To understand this precisely we now turn to some analytic toy models.

4 Building a toy model for memory by accident

4.1 Explicit feedforward model

So far we have analyzed a few networks, each following a single set of co-active plasticity rules. However, these simulations do not help us to understand the ubiquity of memory-by-accident across the set of all stabilizing plasticity rules. To that end, we build an analytic toy model from which we can derive the memory properties of networks following various stabilizing plasticity rules. We thus turn to a minimal firing rate model capable of self-stabilization.

We considered a linear feedforward network with two inputs $\boldsymbol{x} = (x_0, x_1)$ and a single output y: $y(t) = w_0(t)x_0(t) + w_1(t)x_1(t)$ (fig. 3). All activities were nonnegative (though weights were unconstrained), and for simplicity we made the inputs unit norm. We considered a four-parameter set of Hebbian/non-Hebbian plasticity rules inspired by the full spiking model:

$$\frac{\partial w_i(t)}{\partial t} = \theta_0 + \theta_1 x_i(t) + \theta_2 y(t) + \theta_3 x_i(t) y(t) \tag{2}$$

Not all these plasticity rules are meaningful, we therefore restricted to rules that produced stable output activity for all possible inputs, a redefinition of *stability* for this simple feedforward model. In

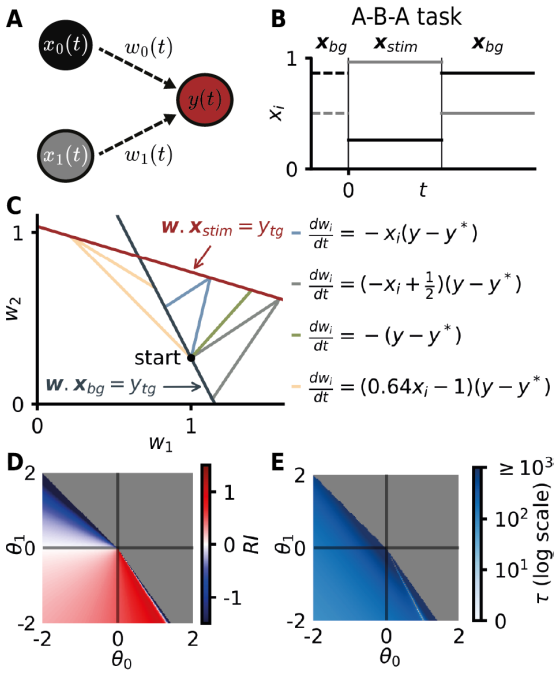


Figure 3: **A**: Feedforward linear network with two input neurons and one output neuron. **B**: Simplified version of the familiarity task. Starting from a steady state solution for $\boldsymbol{x}_{\text{bg}}$, the network was shown $\boldsymbol{x}_{\text{stim}}$ until convergence then $\boldsymbol{x}_{\text{bg}}$ until convergence. **C**: Dynamics of four learning rules in the A-B-A task, shown in weight space. All networks were initialized at the same state $\boldsymbol{w}_0 \in \boldsymbol{W}^*_{\boldsymbol{x}_{\text{bg}}}$ denoted as "start". **D**: Phase portrait of the relative improvement RI as a function of the values of θ_0 and θ_1. This was computed for $\Theta_{\boldsymbol{x}_{\text{bg}}} = \frac{\pi}{6}$ and $\Theta_{\boldsymbol{x}_{\text{stim}}} = \frac{\pi}{6} + \frac{\pi}{4}$. Grey denotes unstable rules. **E**: Same parameter sweep as C, but plotting the time to convergence τ.

particular, we chose a target output firing rate, y^*, and derived constraints on the plasticity parameters $\{\theta_j\}_{0\leq j\leq 3}$ such that the output activity eventually converged to y^* no matter which unit norm input was presented (see Supplementary for derivation). These assumptions reduced the set of feasible learning rules to the following two-parameter model:

$$\frac{\partial w_i(t)}{\partial t} = (y(t) - y^*)(\theta_0 + \theta_1 x_i(t)),\ s.t.\ \theta_0 + \theta_1 < 0 \text{ and } \sqrt{2}\theta_0 + \theta_1 < 0 \tag{3}$$

In this setting we devised a simplified task to assess memory, the A-B-A task (fig. 3B). The network received two stimuli, $\boldsymbol{x}_{\text{bg}}$ and $\boldsymbol{x}_{\text{stim}}$, both unit norm and aligned at a random, smaller than 90°, angle from the x axis. For each input there is a stable subset of weight configurations, and in this case, due to the linearity of the problem, they form a line. We define $\boldsymbol{W}^*_{\boldsymbol{x}} = \{\boldsymbol{w}, \boldsymbol{w}.\boldsymbol{x} = y^*\}$ as the line of weight vectors that produce a stable output firing rate ($y = y^*$) for a given input $\boldsymbol{x}$. We initialized the network at $\boldsymbol{w}_0 \in \boldsymbol{W}^*_{\boldsymbol{x}_{bg}}$, i.e. at a stable point for the background input $\boldsymbol{x}_{\text{bg}}$. We then presented the stimulus, ran the network to convergence, before returning again to the background input and running to convergence (fig. 3C). In this simple setting, we defined the "memory" of the system in two ways. First via the relative improvement RI:

$$\text{RI}(\theta_0, \theta_1, y^*) = \frac{\left(\boldsymbol{w}_{\text{bg}'} - \boldsymbol{w}_{\text{bg}}\right) . \left(\boldsymbol{w}_{\text{bg}\cap\text{stim}} - \boldsymbol{w}_{\text{bg}}\right)}{||\boldsymbol{w}_{\text{bg}\cap\text{stim}} - \boldsymbol{w}_{\text{bg}}||^2} \tag{4}$$

$\boldsymbol{w}_{\text{bg}}$ denotes the steady state weights for input $\boldsymbol{x}_{\text{bg}}$ at the start of the task , while $\boldsymbol{w}_{\text{bg}'}$ are those for $\boldsymbol{x}_{\text{bg}}$ at the end of the task. $\boldsymbol{w}_{\text{bg}\cap\text{stim}}$ is the intersection between $\boldsymbol{W}^*_{\boldsymbol{x}_{\text{bg}}}$ and $\boldsymbol{W}^*_{\boldsymbol{x}_{\text{stim}}}$. The magnitude of RI encodes the distance between the initial and final state of the network, while its sign indicates whether the final state was closer to the intersection between the two lines of fixed points or not. Zero indicates that the final weights are identical to the initial weights, while positive (negative) values indicate net movement towards (away from) the weight matrix that stabilizes both $\boldsymbol{x}_{\text{bg}}$ and $\boldsymbol{x}_{\text{stim}}$. Second, we evaluated memory with the time to convergence to the fixed point $\tau(\theta_0, \theta_1, y^*) = \min\big(t, |y(t) - y^*| \leq \rho\big)$, with $\rho = 0.01$ a threshold.

Almost all rules exhibited some form of memory in this toy model, i.e. the final and initial network states differed indicating a dependency on the past input $\boldsymbol{x}_{\text{stim}}$ (fig. 3D&E). Both the ubiquitous existence of memory, and the wide diversity of memory timescales were consistent with the observations made in the spiking models (figs. 1 and 2), suggesting that the degeneracy of weights was indeed the key factor in the memory by accident phenomenon. In fact, the only rules that did not yield memories were the purely non-Hebbian ones ($\theta_1 = 0$). We verified that these results held qualitatively when extending this toy model to higher dimensions (see Appendix and fig. S8).

Besides qualitatively reproducing the observation that most stabilizing plasticity rules exhibit memories, this toy model led us to formulate several predictions:

1. **Trade-off between memory and stability**: rules that elicit the strongest memories—longer-lasting and more robust— should be closest to unstable regions (fig. 3D&E).

2. **Learning rates and memory**: *a priori*, the role of the learning rate was unclear to us, as it influenced both the learning and forgetting phases. In the toy model, the learning rate of the rules did not affect RI—the fixed points reached—only the speed of convergence τ. Therefore lower (non-zero) learning rates should lead to longer-lasting memories.

3. **Hebbian vs non-Hebbian terms and memory**: the model predicted that the only rules that do not elicit memory are exclusively non-Hebbian: $\frac{\partial w_i}{\partial t} \propto y - y*$. Further, rules combining the plasticity parameters could be more efficient, i.e. with longer-lasting memories, than exclusively Hebbian or non-Hebbian rules.

4.2 Feedforward implicit model

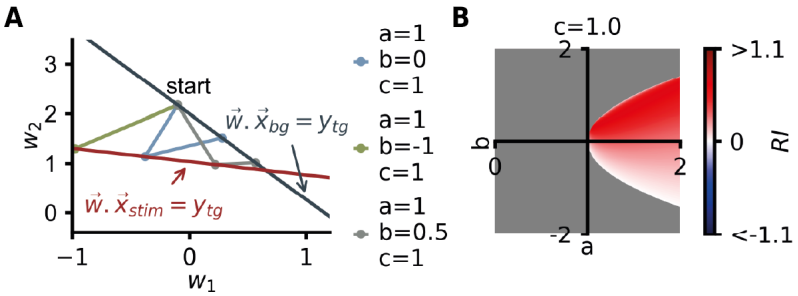


Figure 4: **A**: Fixed points for three rules of the implicit toy model in the A-B-A task, shown in weight space. All networks are initialized at the same state $\boldsymbol{w}_0 \in \boldsymbol{W}^*_{\boldsymbol{x}_{\text{bg}}}$. **B**: Phase portrait of the relative improvement RI as a function of the values of a and b, for fixed c. This was computed for $\Theta_{\boldsymbol{x}_{\text{bg}}} = \frac{\pi}{6}$ and $\Theta_{\boldsymbol{x}_{\text{stim}}} = \frac{\pi}{6} + \frac{\pi}{4}$. Grey denotes unstable rules.

One potential concern with our previous toy model (and indeed, all of our previous approaches) is the dependence on a particular parameterization of the learning rules. In this section we therefore defined a more abstract class of plasticity rules, operating in the same linear feedforward network as in the previous section. In the vein of recent work [28], we forego an explicit parameterization and instead considered rules that minimize different distance metrics in weight space. More specifically, we again considered learning rules that lead to a stable output firing of y^* for *any* input $\boldsymbol{x}$:

$$\forall\,(\boldsymbol{x}, y_0, \boldsymbol{w}_0) \in \mathbb{R}^{2\times1\times2}, \lim_{t\to+\infty} y(t, \boldsymbol{x}, y_0, \boldsymbol{w}_0) = y^* \tag{5}$$

Then we assumed that the stabilizing weight configuration that network would choose $\boldsymbol{W}^*_{\boldsymbol{x}}$ would be the closest according to some distance metric:

$$\forall\,(\boldsymbol{x}, y_0, \boldsymbol{w}_0) \in \mathbb{R}^{2\times1\times2}, \lim_{t\to+\infty} \boldsymbol{w}(t, \boldsymbol{x}, y_0, \boldsymbol{w}_0) = \underset{\boldsymbol{w}\in\boldsymbol{W}^*_{\boldsymbol{x}}}{\text{argmin}} ||\boldsymbol{w} - \boldsymbol{w}_0||^2_\Sigma \tag{6}$$

with $||.||_\Sigma$ the norm induced by the Mahalanobis distance D:

$$\forall(\mathbf{w}_1, \mathbf{w}_2) \in \mathbb{R}^{2\times2},\ D_\Sigma(\mathbf{w}_1, \mathbf{w}_2) = \sqrt{(\mathbf{w}_1 - \mathbf{w}_2)^T \Sigma^{-1} (\mathbf{w}_1 - \mathbf{w}_2)} \tag{7}$$

We defined $\Sigma^{-1} = \begin{pmatrix} a & b \\ b & c \end{pmatrix}$ leaving us with three plasticity parameters: a, b, and, c. To be a well-defined distance, Σ^{-1} needs to be positive semidefinite leading to the further constraint that $a \geq 0$ and $ac - b^2 \geq 0$. From these assumptions, we could calculate the system's fixed point as a function of the initial state (see Supplementary for the full derivation). Despite only describing the fixed points, not the dynamics used to reach them, we found some matches between explicitly and implicitly parameterized rules (fig. S6). For instance, the rule minimizing the L2 norm ($a = 1, b = 0, c = 1$) corresponded to $\theta_0 = 0, \theta_1 = -1$ in the explicit model, which could also be seen as minimizing the L2 mean squared error between the current activity y and the desired activity y^* (see Supplementary).

Again, it appeared that with these stabilizing rules, memory by accident was the rule and not the exception (fig. 4B). This arose for almost all choices of metric minimization, independent of the exact

dynamics chosen to implement these weight updates, suggesting our ideas generalized beyond the particular plasticity rule parameterization we might choose.

5 Testing insights from toy models in spiking networks

Finally, we returned to the full recurrent spiking network with four co-active plasticity rules to test the predictions from our toy models.

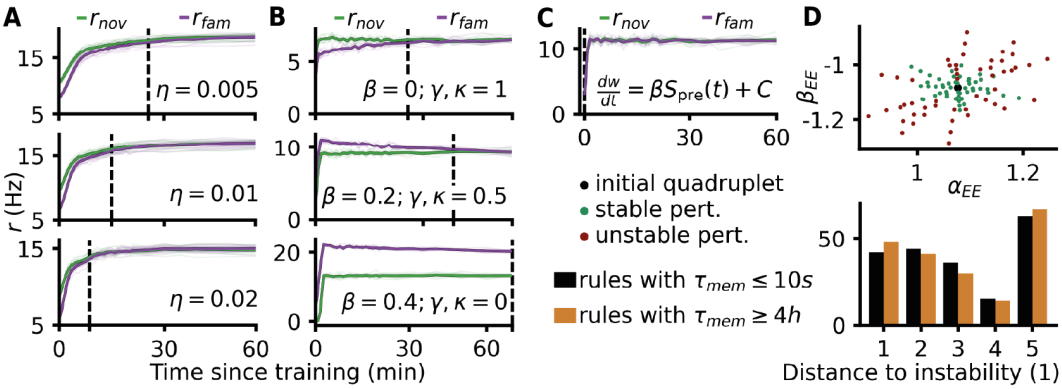


Figure 5: **A**: Recurrent spiking network with variants of iSTDP (I-to-E plasticity only) in the familiarity task, with different learning rates. Simulations were repeated 5 times, averages across seeds are shown in bold, dashed lines denote the last timestep at which the population firing rate in response to familiar stimuli was significantly different to novel responses (Student t-test, $p < 0.05$). **B**: Feedforward spiking network with I-to-E plasticity in the familiarity detection task. Three rules were tested from top to bottom: $\tau^{\text{pre}} = \tau^{\text{post}} = 20\text{ms}, \alpha = -0.12, \beta = 0, \kappa = \gamma = 1$ (original iSTDP) $\tau^{\text{pre}} = \tau^{\text{post}} = 20\text{ms}, \alpha = -0.12, \beta = 0.2, \kappa = \gamma = 0.5$ (half post-only, half Hebbian), $\tau^{\text{pre}} = \tau^{\text{post}} = 20\text{ms}, \alpha = -0.12, \beta = 0.4, \kappa = \gamma = 0$ (see fig. S7 for full simulations). **C**: Same as B, but for a different I-to-E plasticity rule with a spike-independent term, $\frac{dw}{dt} = \beta S_{\text{pre}}(t) + C$, with β and C plasticity parameters, and S_{pre} the pre-synaptic spike train. **D**: Top: example base rule quadruplet in black, and stability of perturbations along 10 random directions in plasticity parameter space. Bottom: distribution of indices of the first unstable perturbation along each direction. Results for 10 quadruplets with long memory lifetimes $\tau_{\text{mem}} \geq 4$h and 10 others with $\tau_{\text{mem}} \leq 10$s.

Learning rate and memory: We varied the learning rate in the recurrent network with only iSTDP (section 3.2); within the range tested, the slower the rule, the longer the memory lifetime (fig. 5A), as predicted.

Trade-off between memory and stability: we sought to verify that rule quadruplets with the longest memory lifetimes were those closest to the edge of the stability manifold. Calculating such distances involves simulating networks perturbed in all directions in the 24 dimensional parameter space, so is very compute-intensive. Here, therefore, we only report a trend when perturbing 10 stable rule quadruplets with memory lifetimes of 10s or less vs perturbing 10 stable rule quadruplets with memory lifetimes of 4h or more (fig. 5D). Overall, the distance to the instability border depended both on the rule quadruplet and on the direction chosen, and marginally more directions were immediately unstable for the long-memory quadruplets. Whether this is a reflection of a limited-compute budget or a real phenomenon remains to be seen.

Hebbian vs non-Hebbian terms and memory: we focused on I-to-E plasticity, to avoid potential confounds induced by co-active rules. Using mean-field analysis as in previous work [11], we derived the activity setpoint for spiking rules from eq. (1):

$$r_{\text{exc}} = \frac{-\alpha_{\text{IE}} r_{\text{inh}}}{\beta_{\text{IE}} + r_{\text{inh}}(\kappa_{\text{IE}} \tau_{\text{IE}}^{\text{post}} + \gamma_{\text{IE}} \tau_{\text{IE}}^{\text{pre}})} \tag{8}$$

The iSTDP rule [11] chooses $\beta_{\text{IE}} = 0$, thus simplifying r_{inh} out of the equation, which effectively ensures network stabilization for all inhibitory activity levels. In recurrent networks, rules with $\beta_{\text{IE}} \neq 0$ display unstable network dynamics (see fig. S4D). However, many unstable I-to-E rules were in fact stable as part of a quadruplet [20]. To be able to test these rules in a controlled setting,

we reverted to the feedforward spiking network. We tested I-to-E rules that had the same learning rates and target firing rates, but which traded off Hebbian terms κ and γ for the post-only term β. Surprisingly, the more post-only potentiation, the longer-lasting the memory (fig. 5B).

These experiments verified two predictions at once, though in an unexpected way. First, adding terms besides the Hebbian term could indeed be useful for memory-by-accident. Second, we found an unexpected verification of the stability-memory trade-off: the higher the β which destabilizes the rule, the longer the memory. Finally, we verified that the only rule predicted not to elicit memory by accident, $\frac{\partial w_i}{\partial t} \propto y - y^*$, behaved as predicted in the spiking case. To do so, we defined an equivalent rule, $\frac{dw}{dt} = \beta S_{\text{pre}}(t) + C$. Note that such a rule was not part of the search space defined in eq. (1). We verified that it did not elicit memory by accident in the feedforward spiking model (fig. 5C).

6 Discussion

This study sought to understand a puzzling observation: almost all plasticity rules that produce stable network activity also lead to memory abilities [20]. By studying three different spiking network models we linked this phenomena to the degeneracy of weight matrices capable of stabilizing a network for a given input. In essence, from amongst the degenerate space of stable weights, the configuration that the plasticity rules chose depends on past inputs, creating a form of memory. We demonstrated the existence of this memory-by-accident phenomena all the way down to a 3-neuron linear rate model. We then used these tractable toy models to understand the phenomenon's ubiquity, finding it in nearly all rules, whether parameterized explicitly or implicitly. Instead of a seemingly fortunate accident, our analysis leads us to pose basic memory abilities as a near-unavoidable consequence of network stabilization.

In our toy models the only non-memorizing rules were purely non-Hebbian; do our findings boil down to "Hebbian learning creates memories"? We think not. While the link between Hebbian rules and memory is natural and longstanding, we find a wide diversity of combinations of Hebbian and non-Hebbian terms that produce long-lasting memories. Further, we found in our feedforward spiking models that the rules with the longest-lived "memories-from-accidents" were exclusively non-Hebbian (fig. 5B). Rather, we argue this is a generic property of stabilizing plasticity rules.

A promising aspect of memory by accident is that it seems to benefit from the system's complexity: the more neurons and weights, the more degenerate solutions; the more co-active learning rules, the more opportunities to exploit this degeneracy. This near-trivial phenomenon in toy models transferred to large recurrent network models and elicited complex and long-lasting memory abilities. Our recurrent spiking models, though relatively complex by today's standards, are but a simplistic reflection of brain regions, which are composed of orders of magnitude more neurons and synapses, and use hundreds of different synapse types [29], each of which could have their own plasticity rule. For now we can only speculate on the potential computations that this phenomenon could unlock in systems of comparable scale to the brain.

The memories we have been discussing exhibit a diversity of timescales, from seconds to hours (and potentially beyond). While our longer-lasting memories look like classic episodic memories that have often been modeled with spike-timing dependent plasticity rules like ours, the shorter of these timescales instead match those discussed by the activity-silent working memory literature [25]. Interestingly, classic models in this area rely on qualitatively different plasticity mechanisms such as short-term plasticity [30, 31]. It is exciting that our theory proposes a unifying explanation for both phenomena operating on different timescales.

The systematic emergence of basic forms of memory from relatively simple unsupervised and local plasticity rules could also provide hints on the evolution of the complex learning abilities observed. For plastic neural networks to be useful they must be stable. We find that as soon as they are stable they permit memory, potentially presenting an easy stepping stone to higher-order cognitive abilities from compositions of readily available memory motifs.

Limitations: Given the compute-load of simulating large plastic recurrent spiking networks with a high dimensional plasticity space, we could only partially verify the predictions made by the toy models. Moreover, many other phenomena influence the memory of the system besides those captured in the toy model. For instance, we noticed that some rule quadruplets don't respond to any

memory task, but have very high firing rates (> 30Hz), effectively making them recurrent driven and oblivious to their inputs.

We have also not modeled the effects of co-active rules, which in practice appeared to be a key element to make memories generated by the pairwise rules robust and noticeable on behavioural timescales [20]. Indeed, most of the original set of co-active plasticity rules used as a starting point for this study [20] were unstable when considered in isolation, or in a mean-field model with co-activity (fig. S1). This effectively prevented us from drawing conclusions on their memory capacity in this work. However, understanding how rules unstable in isolation can be stable together and produce longer-lasting memories than their individual counterparts is an important avenue for future work, perhaps using more refined versions of mean-field analysis than was performed here [32, 9, 10, 33, 7].

We might also wonder how these mechanisms could be extended to embed lifelong memories. In our networks it appears that memories are ultimately erased by random fluctuations (fig. 2C), something not captured by our toy models. However, as a first approximation, the larger the RI (the further away the final state is from the initial), the longer the memory will last before being erased by noise. Further, some learning rules seem to be able to dramatically postpone this eventual forgetting, especially through co-activity (fig. 2A). Together these mechanisms seem sufficient to encode a memory long enough for it to be consolidated by other systems, such as the wake-sleep cycle. On the contrary, a memory system with graceful forgetting and robust stability enforced as default may be desirable.

To sum up, our distillation of automatically discovered plasticity rules in large recurrent networks resulted in a remarkably general and simple phenomenon, memory by accident, that highlights the unreasonable effectiveness of simple unsupervised rules at making memories.

Acknowledgments and Disclosure of Funding

We thank Tim Vogels for his help and support, as well as Lukas Braun, Everton Agnes, Peter Latham and Antonio Sclocchi for useful discussions. This work was supported by a Schmidt Science Polymath Award, the Sainsbury Wellcome Centre Core Grant from Wellcome (219627/Z/19/Z) and the Gatsby Charitable Foundation (GAT3850).

References

[1] Larry F Abbott and Sacha B Nelson. Synaptic plasticity: taming the beast. *Nature neuroscience*, 3(11):1178–1183, 2000.

[2] Erkki Oja. Simplified neuron model as a principal component analyzer. *Journal of mathematical biology*, 15(3):267–273, 1982.

[3] Elie L Bienenstock, Leon N Cooper, and Paul W Munro. Theory for the development of neuron selectivity: orientation specificity and binocular interaction in visual cortex. *Journal of Neuroscience*, 2(1):32–48, 1982.

[4] Cengiz Pehlevan, Tao Hu, and Dmitri B Chklovskii. A hebbian/anti-hebbian neural network for linear subspace learning: A derivation from multidimensional scaling of streaming data. *Neural computation*, 27(7):1461–1495, 2015.

[5] Manu Srinath Halvagal and Friedemann Zenke. The combination of hebbian and predictive plasticity learns invariant object representations in deep sensory networks. *Nature Neuroscience*, 26(11):1906–1915, 2023.

[6] Abigail Morrison, Markus Diesmann, and Wulfram Gerstner. Phenomenological models of synaptic plasticity based on spike timing. *Biological cybernetics*, 98:459–478, 2008.

[7] Friedemann Zenke, Guillaume Hennequin, and Wulfram Gerstner. Synaptic plasticity in neural networks needs homeostasis with a fast rate detector. *PLoS computational biology*, 9(11): e1003330, 2013.

[8] Friedemann Zenke, Everton J Agnes, and Wulfram Gerstner. Diverse synaptic plasticity mechanisms orchestrated to form and retrieve memories in spiking neural networks. *Nature communications*, 6(1):1–13, 2015.

[9] Richard Kempter, Wulfram Gerstner, and J Leo Van Hemmen. Hebbian learning and spiking neurons. *Physical Review E*, 59(4):4498, 1999.

[10] Richard Kempter, Wulfram Gerstner, and J Leo Van Hemmen. Intrinsic stabilization of output rates by spike-based hebbian learning. *Neural computation*, 13(12):2709–2741, 2001.

[11] Tim P Vogels, Henning Sprekeler, Friedemann Zenke, Claudia Clopath, and Wulfram Gerstner. Inhibitory plasticity balances excitation and inhibition in sensory pathways and memory networks. *Science*, 334:1569–1573, 2011.

[12] Yoshua Bengio, Samy Bengio, and Jocelyn Cloutier. Learning a synaptic learning rule. *IJCNN-91-Seattle International Joint Conference on Neural Networks*, 2:969, 1991. doi: 10.1109/IJCNN.1991.155621.

[13] Luke Metz, Niru Maheswaranathan, Brian Cheung, and Jascha Sohl-Dickstein. Learning unsupervised learning rules. *arXiv preprint*, 1804.00222, 2018.

[14] Jack Lindsey and Ashok Litwin-Kumar. Learning to learn with feedback and local plasticity. *Advances in Neural Information Processing Systems 34 (NeurIPS)*, 2020.

[15] Basile Confavreux, Friedemann Zenke, Everton J Agnes, Timothy Lillicrap, and Tim P Vogels. A meta-learning approach to (re) discover plasticity rules that carve a desired function into a neural network. *Advances in Neural Information Processing Systems 34 (NeurIPS)*, 2020.

[16] Jakob Jordan, Maximilian Schmidt, Walter Senn, and Mihai A Petrovici. Evolving interpretable plasticity for spiking networks. *eLife*, 10:e66273, 2021.

[17] Danil Tyulmankov, Guangyu Robert Yang, and LF Abbott. Meta-learning synaptic plasticity and memory addressing for continual familiarity detection. *Neuron*, 110(3):544–557, 2022.

[18] Thomas Miconi. Learning to acquire novel cognitive tasks with evolution, plasticity and meta-meta-learning. *International Conference on Machine Learning*, pages 24756–24774, 2023.

[19] Basile Confavreux*, Poornima Ramesh*, Pedro J. Goncalves, Jakob H. Macke, and Tim P. Vogels. Meta-learning families of plasticity rules in recurrent spiking networks using simulation-based inference. *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[20] Basile Confavreux, Zoe Harrington, Maciej Kania, Poornima Ramesh, Anastasia N. Krouglova, Panos Bozelos, Jakob H. Macke, Andrew Saxe, Pedro J. Goncalves, and Tim P. Vogels. Memory by a thousand rules: Automated discovery of multi-type plasticity rules reveals variety degeneracy at the heart of learning. *bioArXiv*, 2025.

[21] John J Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the national academy of sciences*, 79(8):2554–2558, 1982.

[22] Rafal Bogacz and Malcolm W Brown. Comparison of computational models of familiarity discrimination in the perirhinal cortex. *Hippocampus*, 13(4):494–524, 2003.

[23] Lionel Standing. Learning 10000 pictures. *The Quarterly journal of experimental psychology*, 25(2):207–222, 1973.

[24] Timothy F Brady, Talia Konkle, George A Alvarez, and Aude Oliva. Visual long-term memory has a massive storage capacity for object details. *Proceedings of the National Academy of Sciences*, 105(38):14325–14329, 2008.

[25] Mark G Stokes. 'activity-silent'working memory in prefrontal cortex: a dynamic coding framework. *Trends in cognitive sciences*, 19(7):394–405, 2015.

[26] Ashok Litwin-Kumar and Brent Doiron. Formation and maintenance of neuronal assemblies through synaptic plasticity. *Nature Communications*, 5, 2014.

[27] Auguste Schulz, Christoph Miehl, Michael J Berry II, and Julijana Gjorgjieva. The generation of cortical novelty responses through inhibitory plasticity. *Elife*, 10:e65309, 2021.

[28] Roman Pogodin, Jonathan Cornford, Arna Ghosh, Gauthier Gidel, Guillaume Lajoie, and Blake Aaron Richards. Synaptic weight distributions depend on the geometry of plasticity. *The Twelfth International Conference on Learning Representations*, 2024.

[29] Zizhen Yao, Cindy TJ van Velthoven, Michael Kunst, Meng Zhang, Delissa McMillen, Changkyu Lee, Won Jung, Jeff Goldy, Aliya Abdelhak, Matthew Aitken, et al. A high-resolution transcriptomic and spatial atlas of cell types in the whole mouse brain. *Nature*, 624(7991): 317–332, 2023.

[30] Gianluigi Mongillo, Omri Barak, and Misha Tsodyks. Synaptic theory of working memory. *Science*, 319(5869):1543–1546, 2008.

[31] Matthijs Pals, Terrence C Stewart, Elkan G Akyürek, and Jelmer P Borst. A functional spiking-neuron model of activity-silent working memory in humans based on calcium-mediated short-term synaptic plasticity. *PLoS computational biology*, 16(6):e1007936, 2020.

[32] Ofer Hendin, David Horn, and Misha V Tsodyks. The role of inhibition in an associative memory model of the olfactory bulb. *Journal of computational neuroscience*, 4:173–182, 1997.

[33] Mark CW Van Rossum, Guo Qiang Bi, and Gina G Turrigiano. Stable hebbian learning from spike timing-dependent plasticity. *Journal of neuroscience*, 20(23):8812–8821, 2000.

[34] Friedemann Zenke and Wulfram Gerstner. Limits to high-speed simulations of spiking neural networks using general-purpose computers. *Frontiers in neuroinformatics*, 8:76, 2014.

[35] Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, September 2020.

[36] James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. JAX: composable transformations of Python+NumPy programs. 2018. URL `http://github.com/jax-ml/jax`.

[37] J. D. Hunter. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9 (3):90–95, 2007.

[38] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020.

[39] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

A Technical Appendices and Supplementary Material

The code and data to reproduce the results in this paper can be found on Github. Spiking network simulations were performed on a single CPU using the Auryn simulator [34]. The perturbations of rule quadruplets in fig. 5C (≈3000 simulations) were performed on 500 CPUs on the ISTA HPC cluster. Rate models and analysis of all simulations was performed using numpy [35], JAX [36], matplotlib [37], SciPy [38] and scikit-learn [39].

A.1 Recurrent spiking network model

A.1.1 Network model

We considered a recurrent spiking network with $N_{\mathrm{E}} = 4096$ excitatory neurons and $N_{\mathrm{I}} = 1024$ inhibitory neurons (leaky-integrate and fire point neurons with variable threshold, AMPA and NMDA currents, and conductance-based synapses). This network was based on previous work [8, 20]. The membrane potential dynamics of neuron j (excitatory or inhibitory) followed:

$$\tau_m \frac{\mathrm{d}}{\mathrm{d}t} V_j(t) = -\left(V_j(t) - V_{\text{rest}}\right) - g_j^{\mathrm{E}}(t)\left(V_j(t) - E_{\mathrm{E}}\right) - g_j^{\mathrm{I}}(t)\left(V_j(t) - E_{\mathrm{I}}\right), \tag{9}$$

with $\tau_m = 20$ ms, $V_{\text{rest}} = -70$ mV, $E_{\mathrm{E}} = 0$ mV and $E_{\mathrm{I}} = -80$ mV.

A postsynaptic spike occurred whenever the membrane potential $V_j(t)$ crossed a threshold $V_j^{\text{th}}(t)$, with an instantaneous reset to $V_{\text{reset}} = -70$ mV. This threshold $V_j^{\text{th}}(t)$ was incremented by $V_{\text{spike}}^{\text{th}} = 100$ mV every time neuron j spiked and otherwise decayed following:

$$\tau_{\text{th}} \frac{\mathrm{d}}{\mathrm{d}t} V_j^{\text{th}}(t) = V_{\text{base}}^{\text{th}} - V_j^{\text{th}}(t), \tag{10}$$

with $V_{\text{base}}^{\text{th}} = -50$ mV. The excitatory and inhibitory conductances, g^{E} and g^{I} evolved such that

$$\begin{aligned}
g_j^{\mathrm{E}}(t) &= a g_j^{\text{AMPA}}(t) + (1-a) g_j^{\text{NMDA}}(t) \quad \text{and} \\
\frac{\mathrm{d}}{\mathrm{d}t} g_j^{\mathrm{I}}(t) &= -\frac{g_j^{\mathrm{I}}(t)}{\tau_{\text{GABA}}} + \sum_{i \in \text{Inh}} w_{ij}(t) S_i(t) \\
\text{with} \quad \frac{\mathrm{d}}{\mathrm{d}t} g_j^{\text{AMPA}}(t) &= -\frac{g_j^{\text{AMPA}}(t)}{\tau_{\text{AMPA}}} + \sum_{i \in \text{Exc}} w_{ij}(t) S_i(t) \quad \text{and} \\
\frac{\mathrm{d}}{\mathrm{d}t} g_j^{\text{NMDA}}(t) &= \frac{g_j^{\text{AMPA}}(t) - g_j^{\text{NMDA}}(t)}{\tau_{\text{NMDA}}},
\end{aligned} \tag{11}$$

with $w_{ij}(t)$ the connection strength between neurons i and j (unitless), $a = 0.23$ (unitless), $\tau_{\text{GABA}} = 10$ ms, $\tau_{\text{AMPA}} = 5$ ms, $\tau_{\text{NMDA}} = 100$ ms, $S_i(t) = \sum \delta(t - t_i^*)$ the spike train of presynaptic neuron i, where t_i^* denotes the spike times of neuron k, and δ the Dirac delta.

The network was initialized with random sparse connectivity (10%), with $w_{\text{EE}}^{\text{init}} = w_{\text{EI}}^{\text{init}} = 0.1$ and $w_{\text{IE}}^{\text{init}} = w_{\text{II}}^{\text{init}} = 1$.

The excitatory neurons in the network received $N_{\text{inp}\to\text{E}} = 11025$ inputs from Poisson neurons firing at $r_{\text{bg}}^{\text{inp}} = 10Hz$. When a stimulus was active, a subset of the input neurons increased their firing rate to $r_{\text{active}}^{\text{seq}} = 100Hz$. The connectivity from input neurons to excitatory and inhibitory neurons was receptive-field-like: for each recurrent neuron, we selected a random input neuron as the center of the circular receptive field of radius 8. The connections from neurons of this circular patch of input neurons to the considered recurrent neuron was $w_{\text{inp}} = 0.075$, and 0 to all other input neurons. The inhibitory neurons received inputs from $N_{\text{inp}\to\text{I}} = 4096$ Poisson neurons with w_{inp} and similar receptive field connectivity than for the excitatory population. However, inhibitory neurons only received background inputs ($r_{\text{bg}}^{\text{input}}$) and no specific stimulus patterns.

A.1.2 Plasticity parameterization

This parameterization of plasticity rules included variations of spike-timing-dependent plasticity, and was taken from previous work [19, 20]. The weight from neuron i to neuron j of type X and Y (excitatory or inhibitory) evolved such that:

$$\frac{\mathrm{d}w_{ij}(t)}{\mathrm{d}t} = \eta[S_i(t)\left(\alpha_{\mathrm{XY}} + \kappa_{\mathrm{XY}} x_j(t)\right) + S_j(t)\left(\beta_{\mathrm{XY}} + \gamma_{\mathrm{XY}} x_i(t)\right)] \tag{12}$$

with $\eta = 0.01$ a fixed learning rate, $S_i(t) = \sum_k \delta(t - t_k^i)$ the spike train of neuron i, δ the Dirac delta function to denote the presence of a pre (post)-synaptic spike at time t. The synaptic traces x_i and x_j are low-pass filters of the activity of presynaptic neuron i and postsynaptic neuron j, with time constants τ_{pre} and τ_{post}, such that:

$$\frac{\mathrm{d}}{\mathrm{d}t}x_i(t) = -\frac{x_i(t)}{\tau_{\text{pre}}^{\mathrm{XY}}} + S_i(t) \quad \text{and} \quad \frac{\mathrm{d}}{\mathrm{d}t}x_j(t) = -\frac{x_j(t)}{\tau_{\text{post}}^{\mathrm{XY}}} + S_j(t), \tag{13}$$

Overall, this search space comprised 6 tunable plasticity parameters per synapse type XY (X, Y $\in$ (E, I)): $\theta_{\mathrm{XY}} = [\alpha_{\mathrm{XY}}, \beta_{\mathrm{XY}}, \gamma_{\mathrm{XY}}, \kappa_{\mathrm{XY}}, \tau_{\text{pre}}^{\mathrm{XY}}, \tau_{\text{post}}^{\mathrm{XY}}]$, for a total of 24 plasticity parameters across all four synapse types.

Note that all weights in the network were capped at all times, in the $[0, w_{\max}]$ range, with $w_{\max} = 20$, though rule quadruplets in [20] were considered unstable if more than 10% of the weights at any synapse type reached these extreme values.

A.1.3 Familiarity detection task

We considered a familiarity detection task that was similar to previous work [20]. The network first received nonspecific background inputs for 1h—pre-training phase—, followed by a 40s training phase during which four non-overlapping input patterns—the familiar stimuli—were active in alternation. When a given stimulus was active, ≈10% of the input neurons to the excitatory population had elevated firing rates, while the others remained at background. After training, we reverted to background inputs and regularly probed the network with familiar and novel stimuli for an hour—post-training phase.

Given the input structure and connectivity described above, each stimulus, novel or familiar excited a different subset of recurrent neurons. We defined "engrams" for each stimulus pattern (novel or familiar) by probing the network before training started, and labeling the top 10% of excitatory and inhibitory neurons as part of the engram for the presented stimulus. In practice, since the stimuli were non overlapping, the engrams defined this way also had little overlap.

Note that each stimulus elicited a different network response, in the sense that each stimulus preferentially excited a different subset of recurrent neurons (this can be seen in fig. S2) and thus the stimulus identity could be decoded from the population vector of neuron activities. However, whether this stimulus has been encountered by the network in the past —its novelty or familiarity— was unknown.

We chose a simple decoding strategy for stimulus familiarity: the mean firing rate of the excitatory population (each excitatory neuron contributes equally to the decoding). In the paper that inspired this work [20], a Student t-test was performed over several seeds/simulations/stimuli over mean firing rates in response to familiar or novel stimuli to determine whether novel stimuli elicited statistically different mean firing rates than familiar stimuli. Note that by design, all stimuli elicited statistically indistinguishable mean firing rates in static or naive plastic networks. Thus this task flagged a stimulus-specific change in network activity due to synaptic plasticity.

A.2 Feedforward spiking network model

A.2.1 Network model

1000 Poisson neurons (800 excitatory and 200 inhibitory) projected onto a single output neuron, with the same neuron model and parameters as in the previous section. The excitatory weights were fixed at $w_{ee} = 0.1$, the inhibitory weights were plastic with the parameterization defined for the recurrent spiking case, and initialized at $w_{ie} = 1$.

Note that in fig. 5C, we used a different learning rule, not part of the plasticity parameterization described above. For this rule, $\frac{dw}{dt} = \beta S_{\text{pre}}(t) + C$ with $\beta = 0.4$ and $C = 0.00012$ to obtain a target firing rate of 3Hz given the integration time-step of the simulation (0.1ms).

A.2.2 Familiarity detection task

The task closely resembled the task in the recurrent case. During a pre-training phase of 1h, all input neurons fired at 10Hz. During the training phase that lasted 60s, 100 excitatory and 25 inhibitory increased their firing rates to 100Hz and 50Hz respectively (the "familiar" stimulus). After the training phase, the network was regularly probed on its network response to the familiar stimulus and another, novel stimulus of the same structure than the familiar stimulus but with different neurons (no overlap).

A.3 Toy model 1: Explicit parameterization

A.3.1 2D version

The model is a linear feedforward network with two inputs $\boldsymbol{x} = (x_0, x_1)$ projecting on a single output neuron y:

$$y(t) = w_0(t)x_0(t) + w_1(t)x_1(t) \tag{14}$$

Besides, $\forall t \geq 0,\ y(t) \geq 0,\ x_0(t) \geq 0,\ x_1(t) \geq 0$; $||\boldsymbol{x}|| = 1$ with $||.||$ the L2 norm. Weights were plastic and unconstrained.

Initially, we considered a four-parameter set of Hebbian/non-Hebbian plasticity rules inspired by the full spiking model (see mean-field section below for the relationship between spike-based and rate-based plasticity):

$$\frac{\partial w_i(t)}{\partial t} = \eta\big(\theta_0 + \theta_1 x_i(t) + \theta_2 y(t) + \theta_3 x_i(t) y(t)\big) \tag{15}$$

with θ_0, θ_1, θ_2 and θ_3 four plasticity parameters, and $\eta = 0.01$ a fixed learning rate (omitted below). From this full search space, we only considered rules that admitted a target output firing rate $y^* \in \mathbb{R}^+$ as a stable fixed point for all inputs $\boldsymbol{x}$ considered here. The existence of y^* as a fixed point implied that for all inputs x_i

$$\theta_0 + \theta_1 x_i + \theta_2 y^* + \theta_3 x_i y^* = 0 \implies \theta_0 = -\theta_2 y^* \text{ and } \theta_1 = -\theta_3 y^* \tag{16}$$

Thus the search space became two-dimensional:

$$\begin{cases} \dfrac{\partial w_0}{\partial t} = (y - y^*)(\theta_0 + \theta_1 x_0) \\ \dfrac{\partial w_1}{\partial t} = (y - y^*)(\theta_0 + \theta_1 x_1) \end{cases} \implies \dot{\mathbf{w}} = A\mathbf{w} + B \tag{17}$$

with $A = \begin{pmatrix} x_0(\theta_0 + \theta_1 x_0) & x_1(\theta_0 + \theta_1 x_0) \\ x_0(\theta_0 + \theta_1 x_1) & x_1(\theta_0 + \theta_1 x_1) \end{pmatrix}$ and $B = -y^* \begin{pmatrix} \theta_0 + \theta_1 x_0 \\ \theta_0 + \theta_1 x_1 \end{pmatrix}$.

This system has only one non-zero eigenvalue: $\lambda_1 = \theta_0(x_0 + x_1) + \theta_1(x_0^2 + x_1^2)$. The system thus has a neutral mode ($\lambda_0 = 0$), for the system to converge to a point on the line attractor, we need $\lambda_1 < 0$. Because $||\mathbf{x}|| = 1$ and $x_0, x_1 \geq 0$, $x_0 + x_1 \in [1, \sqrt{2}]$ and $x_0^2 + x_1^2 = 1$. As a result, we need θ_0 and θ_1 to be below the lines $\theta_0 + \theta_1 = 0$ and $\theta_0\sqrt{2} + \theta_1 = 0$.

For the numerical results reported in the paper, we simulated the system in the A-B-A task with: $y^* = 1$, $\boldsymbol{w}_0 = (1, 0.27) \in \boldsymbol{W}^*_{\boldsymbol{x}_{\text{bg}}}$, $\boldsymbol{x}_{\text{bg}} \angle \boldsymbol{x}_{\text{stim}} = \frac{\pi}{4}$. We ran the system until convergence at each phase of the A-B-A task, in practice we found that $T = 20000$ epochs was sufficient.

We verified that the results of the parameter sweeps on θ_0 and θ_1 had similar trends for different stimuli angles and initializations.

A.3.2 Extension to higher input dimensions

In the main text, we chose the smallest model possible (2D) for ease of visualization. However, the findings readily extend to higher dimensions ($N_{\text{in}} > 2$ input neurons).

We consider N_{in} input neurons with activity $\boldsymbol{x}$ projecting on a single output neuron y with weights $\boldsymbol{w}$: $y = \boldsymbol{W}^T\boldsymbol{x}$. We choose $\boldsymbol{x}$ to be of unit norm with nonnegative entries. The plasticity rules are the same as in the 2D case:

$$\frac{\partial w_i}{\partial t}(t) = (y(t) - y^*)\big(\theta_0 + \theta_1 x_i(t)\big),\ i = 1, \cdots, N_{\text{in}} \tag{18}$$

This is an affine system of ODEs, which can be written in vector form as $\dot{\boldsymbol{w}} = A\boldsymbol{w} + \boldsymbol{b}$, with $A = \boldsymbol{x}(\theta_0\mathbf{1} + \theta_1\boldsymbol{x})$, and $\mathbf{1}$ a N_{in}-dimensional vector of ones.

For a constant input $\mathbf{x}$, this system is rank one, and the non-zero eigenvalue is $\lambda = \theta_0 \sum_{i=1}^{N_{\text{in}}} x_i + \theta_1 \sum_{i=1}^{N_{\text{in}}} x_i^2 = \theta_0 \sum_{i=1}^{N_{\text{in}}} x_i + \theta_1$ for unitary norm inputs. Note that this is a generalization of the derivation presented above.

Edge of stability: For the system to be stable, we need $\lambda < 0$, which translate for unit-norm, nonnegative inputs to $\theta\sqrt{N_{\text{in}}} + \theta_1 < 0$ and $\theta_0 + \theta_1 < 0$.

Defining a metric to evaluate memory: As in the 2D case, we define $\boldsymbol{w}_{\text{bg}}$, the steady state weights for input $\boldsymbol{x}_{\text{bg}}$ at the start of the A-B-A task, and $\boldsymbol{w}_{\text{bg}'}$ are those for $\boldsymbol{x}_{\text{bg}}$ at the end of the task. However, the 2D definition of RI (eq. (4)) does not generalize readily to N-dimensions, as $\boldsymbol{w}_{\text{bg}\cap\text{stim}}$, the intersection between the hyperplanes $\boldsymbol{W}^*_{\boldsymbol{x}_{\text{bg}}}:\ \boldsymbol{w}^T\boldsymbol{x}_{\text{bg}} = y^*$ and $\boldsymbol{W}^*_{\boldsymbol{x}_{\text{stim}}}:\ \boldsymbol{w}^T\boldsymbol{x}_{\text{stim}} = y^*$ is not unique (assuming these hyperplanes are not parallel). As a proxy, we define $RI = ||\boldsymbol{w}_{\text{bg}'} - \boldsymbol{w}_{\text{bg}}||$, which only evaluates how far the final state is from the initial one, and not whether the change is pushing towards the intersection or not.

Overall, as can be seen in fig. S8 increasing the dimensionality of the toy models did not change qualitatively the findings reported in the main paper.

A.4 Toy model 2: Implicit parameterization

A.4.1 2D version

This toy model only described the fixed point reached by an implicitly-defined class of learning rules operating in the same linear feedforward network as in previous section.

Specifically, we made two assumptions on the learning rules:

$$\text{Stabilization: } \forall\,(\boldsymbol{x}, y_0, \boldsymbol{w}_0) \in \mathbb{R}^{2\times1\times2},\ \lim_{t\to+\infty} y(t, \boldsymbol{x}, y_0, \boldsymbol{w}_0) = y^* \tag{19}$$

$$\text{Distance minimization: } \forall\,(\boldsymbol{x}, y_0, \boldsymbol{w}_0) \in \mathbb{R}^{2\times1\times2},\ \lim_{t\to+\infty} \boldsymbol{w}(t, \boldsymbol{x}, y_0, \boldsymbol{w}_0) = \underset{\boldsymbol{w}\in\boldsymbol{W}^*_{\boldsymbol{x}}}{\operatorname{argmin}} ||\boldsymbol{w} - \boldsymbol{w}_0||^2_\Sigma \tag{20}$$

with $||.||_\Sigma$ the norm induced by the Mahalanobis distance D:

$$\forall(\mathbf{w}_1, \mathbf{w}_2) \in \mathbb{R}^{2\times2},\ D_\Sigma(\mathbf{w}_1, \mathbf{w}_2) = \sqrt{(\mathbf{w}_1 - \mathbf{w}_2)^T\Sigma^{-1}(\mathbf{w}_1 - \mathbf{w}_2)} \tag{21}$$

with $\Sigma^{-1} = \begin{pmatrix} a & b \\ b & c \end{pmatrix}$, where a, b, and,c are the three plasticity parameters in this search space. To be a well-defined distance, Σ^{-1} needs to be positive semi-definite leading to the further constraint that $a \geq 0$ and $ac - b^2 \geq 0$.

From these assumptions, we derived the final network state $\boldsymbol{w}^f = (w_0^f, w_1^f)$ as a function of the initialization $\boldsymbol{w}^i = (w_0^i, w_1^i)$, $\boldsymbol{x}^i$ (input for which the network is initialized), and (fixed) input $\boldsymbol{x}^f$. Since the final state belongs to $\boldsymbol{W}^*_{\boldsymbol{x}^f}$, we have $\mathbf{w}^{f\,T}\mathbf{x}^f = y^* \implies w_1^f = \frac{y^* - w_0^f x_0^f}{x_1^f}$. We minimize the distance D:

$$D(\mathbf{w}, \mathbf{w}^i)^2 = \frac{||\mathbf{x}^f||^2_{\Sigma^{-1}}}{{x_1^f}^2} {w_0}^2 + 2\frac{-a w_0^i {x_1^f}^2 + b x_1^f \left[y^* + w_0^i x_0^f - w_1^i x_1^f\right] + c x_0^f \left[w_1^i x_1^f - y^*\right]}{{x_1^f}^2} w_0 \tag{22}$$

$$+ ||\mathbf{w}^i||^2_\Sigma + \frac{y^* \left[-2b w_0^i x_1^f + c(-2 w_1^i x_1^f + y^*)\right]}{{x_1^f}^2} \tag{23}$$

Since $\frac{||\mathbf{x}^f||^2_{\Sigma^{-1}}}{{x_1^f}^2} > 0$, the expression above has a single minimum:

$$w_0^f = \frac{a w_0^i {x_1^f}^2 + b x_1^f (w_1^i x_1^f - w_0^i x_0^f - y^*) + c x_0^f (y^* - w_1^i x_1^i)}{||\mathbf{x}^f||^2_{\Sigma^{-1}}} \tag{24}$$

We applied the result above twice, for each phase of the A-B-A task.

Relationship between explicit and implicit toy models: although we don't have a general mapping from the implicit to the explicit rules in both toy models, some rules have the same steady states in both cases.

Notably, the Hebbian rule $\frac{\partial w_i}{\partial t} = -x_i(y - y^*)$ ($\theta_0 = 0, \theta_1 = -1$) had identical fixed point to the rule minimizing the Euclidean distance in the implicit model ($a = 1, b = 0, c = 1$), see fig. S6D. The explicit form of this rule could also be seen as the gradient descent update wrt the loss function $\mathcal{L} = \frac{(y - y^*)^2}{2}$, i.e. $\frac{\partial \mathcal{L}}{\partial w_i} = x_i(y - y^*) = -\frac{\partial w_i}{\partial t}$.

A.4.2 Extension to higher input dimensions

This toy model has the same network architecture as above, but the number of input neurons does change the number of plasticity parameters, as the rules are this time parameterized by a distance metric of the Mahalanobis family (with covariance $\Sigma \in \mathbb{R}^{N_{\text{in}} \times N_{\text{in}}}$, leaving us with $\frac{N_{\text{in}}(N_{\text{in}}+1)}{2}$ plasticity parameters.

Edge of stability: This corresponds to Σ losing its positive semi-definite property (i.e. at least one eigenvalue becomes 0).

A.5 Toy-model 3: Mean-field-inspired model

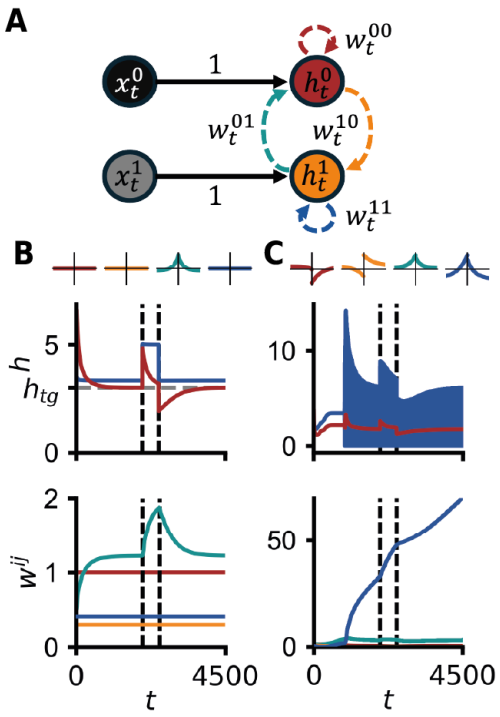
A common method to study spike-timing-dependent plasticity is to perform mean-field analysis [33, 6, 11], which assumes a large network of uncorrelated neurons. Under these assumptions, the weight updates in the spiking network eq. (1) become:

$$\langle \frac{\mathrm{d}w(t)}{\mathrm{d}t} \rangle = \eta[r_{pre}\alpha_{XY} + r_{post}\beta_{XY} + r_{pre}r_{post}(\kappa_{XY}\tau_{XY}^{post} + \gamma_{XY}\tau_{XY}^{pre})] \tag{25}$$

with r_{pre} and r_{post} the firing rates of the pre- and post-synaptic neurons. This method allowed us to get a rate "equivalent" of each spike-timing dependent rule defined ineq. (1). We embedded the rate-equivalent rule quadruplets in a 2-neuron linear recurrent network (2RNN) undergoing the familiarity detection task (fig. S1). The activities r_E, r_I of the 2RNN, representing the excitatory and inhibitory spiking populations, followed:

$$\begin{cases} r_E(t+1) = w_{ee}(t) r_E(t) - w_{ie}(t) r_I(t) + x_E(t) \\ r_I(t+1) = w_{ei}(t) r_E(t) - w_{ii}(t) r_I(t) + x_I(t) \end{cases}. \tag{26}$$

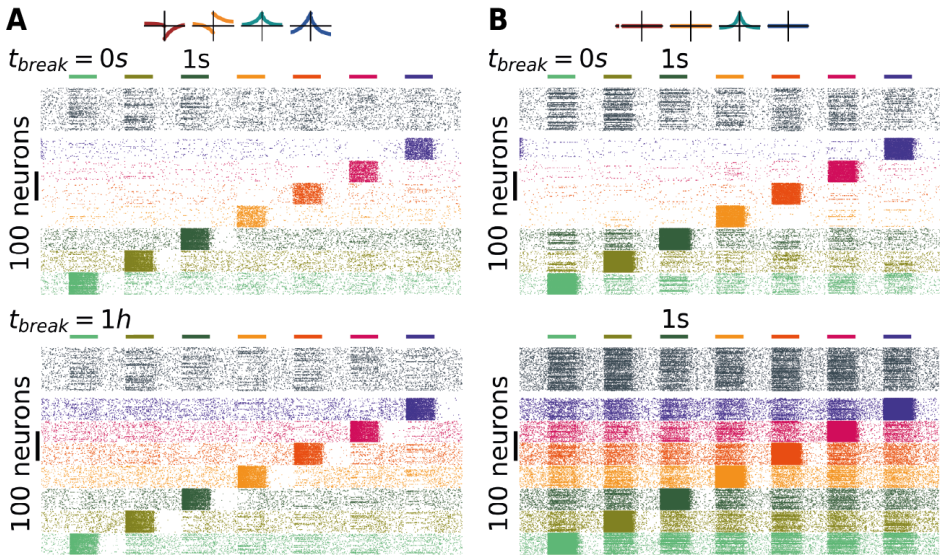
The activities r and weights w were constrained to be positive at all times. The familiarity task was ported to this setting by providing background input to the network $\boldsymbol{x}_{bg} = (1, 1)$, followed by a training period with stimulus $\boldsymbol{x}_{stim} = (1.5, 1)$ before reverting to $\boldsymbol{x}_{bg}$. Over 95% of the rule quadruplets that were stable in the recurrent spiking model elicited diverging weight or activity dynamics in the rate model, such as the rule quadruplet shown in fig. 1 (fig. S1B). Nevertheless, this 2RNN satisfyingly approximated some rules, particularly those evolving in isolation, such as the rate-equivalent of iSTDP (fig. S1C).



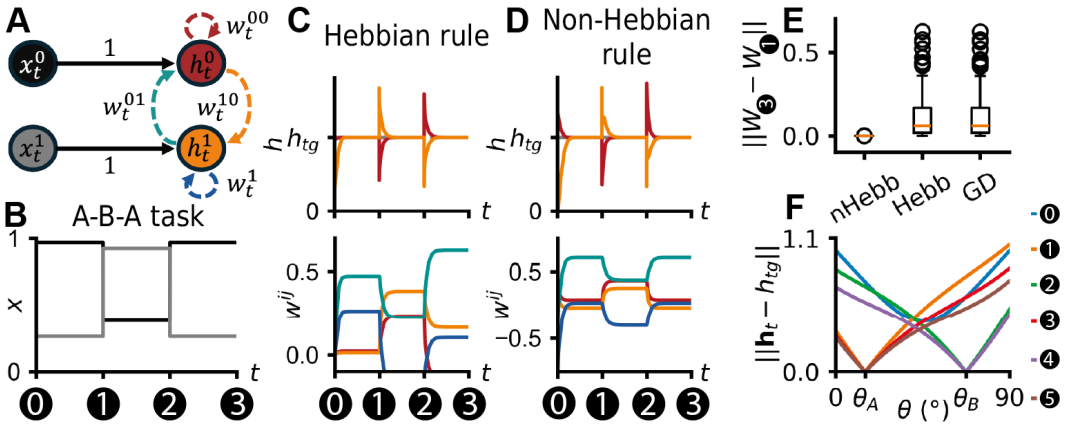
Supplementary Figure S1: **Mean-field-inspired model.** **A**: Linear 2-neuron recurrent network (2RNN) and notations. **B**: 2RNN evolving with the rate-equivalent of the iSTDP rule during the familiarity task (see fig. 2A for spiking equivalent). Dashed lines denote the onset and offset of training. Top: network activities during the task. Bottom: evolution of the four recurrent weights. Colors match the cartoon in A. **C**: Same as B, but for the meta-learned rule quadruplet shown in fig. 1.

Overall this suggested that the assumptions made to obtain the rate equivalent rules were not valid in the case of co-active rules. Indeed, the one rule for which the 2RNN model performed qualitatively similarly like the spiking model is iSTDP, which was shown to decorrelate neuronal activities [11], thus ensuring that the assumption of uncorrelated neuron activities holds. We thus moved to a more abstract setting to understand the memory by accident phenomenon.

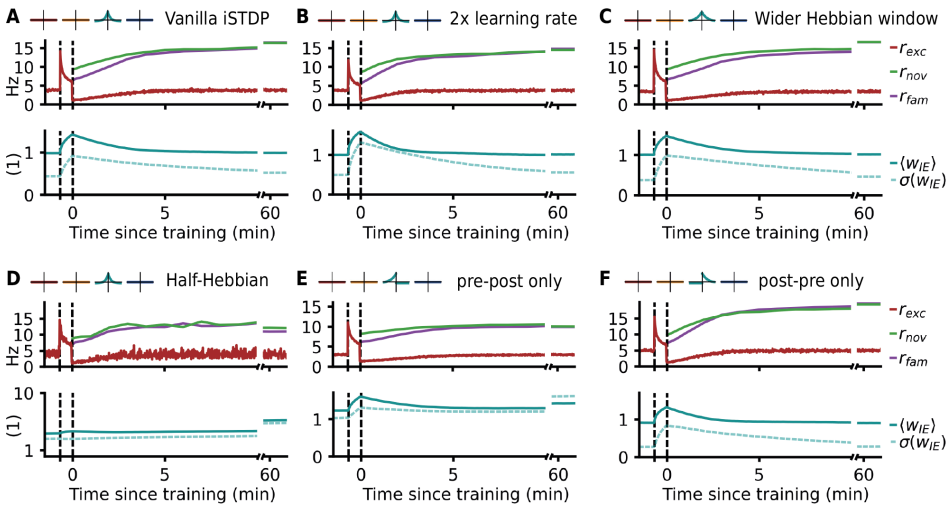
A.6 Supplementary figures



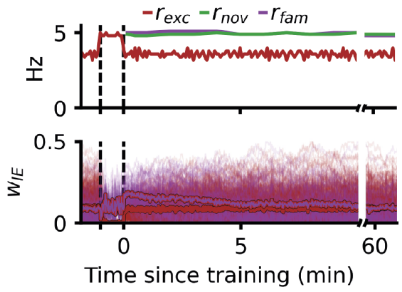
Supplementary Figure S2: **Visualization of network activities for fig. 1. A,B**: Raster plots during two test sessions of the familiarity detection task. Neurons are colored by which engram they belong to (see methods for definition of engrams, gray shows neurons not part of any engram).



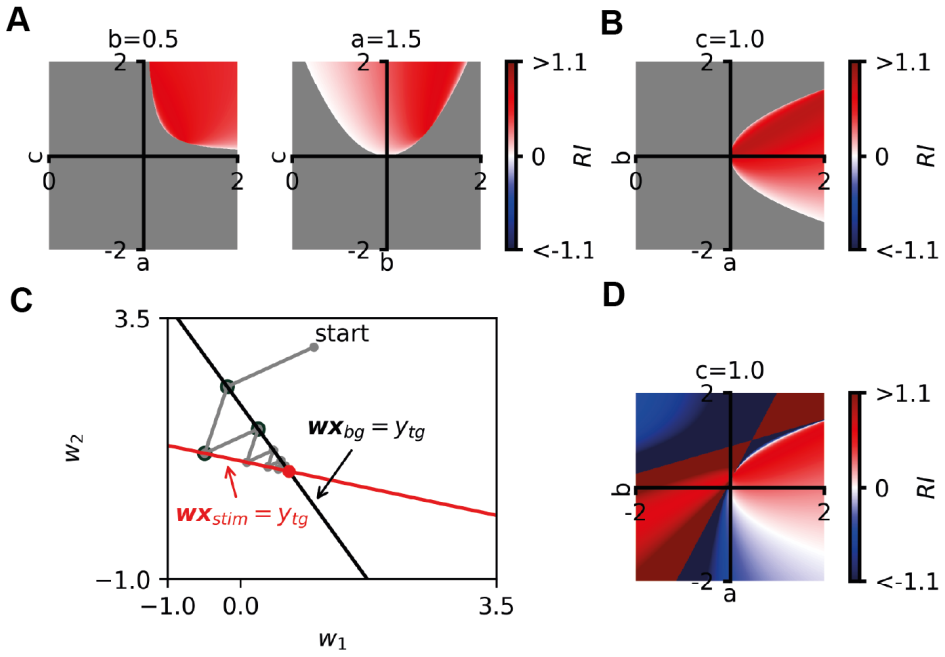
Supplementary Figure S3: **A linear RNN reproduces aspects of memory by accident A**: Network and notations, activities are restricted to be positive. **B**: Example input activities in the A-B-A task. Each stimulus presentation is chosen to be long enough for any potential fixed point to be reached. **C**: Hebbian rule in the A-B-A task: $\Delta w_t^{pre\,post} \propto (h_{tg} - h_t^{post}) h_t^{pre}$ with h_{tg} a fixed target (1). Top: Recurrent neurons activities, Bottom: 4 network weights. **D**: Same as C for a non-Hebbian rule: $\Delta w_t^{pre\,post} \propto (h_{tg} - h_t^{post})$. **E**: Distance between the network weights at the end of the first or the second presentation of stimulus A: averaged over many simulations for the three learning rules tested. "GD" is online gradient descent on the mean squared error loss of the network activity compared to the target activity h_{tg}. **F**: Network response profile for stimuli of various angles at different timepoints of the A-B-A task.



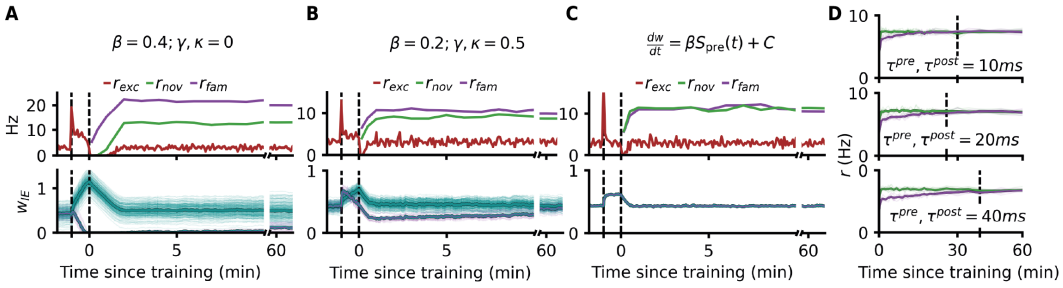
Supplementary Figure S4: **Variations of iSTDP and memory by accident**: Recurrent spiking network undergoing the familiarity detection task, for 6 variants of the iSTDP rule. All variants have the same target excitatory rate of 3Hz. **A**: $\tau_{IE}^{pre} = \tau_{IE}^{post} = 20ms, \alpha_{IE} = -0.12, \beta_{IE} = 0, \kappa_{IE} = \gamma_{IE} = 1$. **B**: $\tau_{IE}^{pre} = \tau_{IE}^{post} = 20ms, \alpha_{IE} = -0.24, \beta_{IE} = 0, \kappa_{IE} = \gamma_{IE} = 2$. **C**: $\tau_{IE}^{pre} = \tau_{IE}^{post} = 40ms, \alpha_{IE} = -0.12, \beta_{IE} = 0, \kappa_{IE} = \gamma_{IE} = 0.5$. **D**: $\tau_{IE}^{pre} = \tau_{IE}^{post} = 20ms, \alpha_{IE} = -0.12, \beta_{IE} = 0.06, \kappa_{IE} = \gamma_{IE} = 0.5$. **E**: $\tau_{IE}^{pre} = \tau_{IE}^{post} = 20ms, \alpha_{IE} = -0.12, \beta_{IE} = 0, \kappa_{IE} = 0, \gamma_{IE} = 1$. **F**: $\tau_{IE}^{pre} = \tau_{IE}^{post} = 20ms, \alpha_{IE} = -0.12, \beta_{IE} = 0, \kappa_{IE} = 1, \gamma_{IE} = 0$.



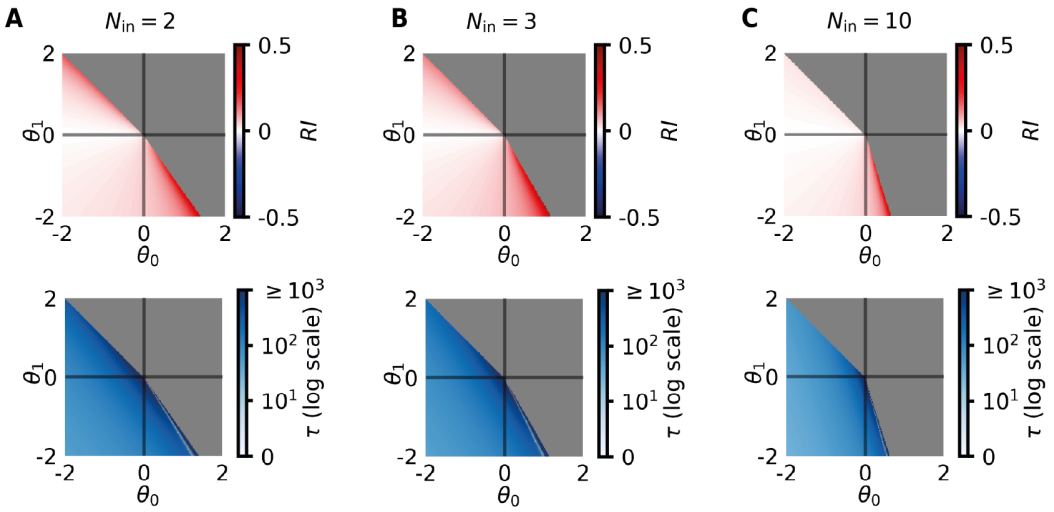
Supplementary Figure S5: **Feedforward spiking mode with E-to-E plasticity.** Top: output neuron firing rate. Bottom: evolution of the 800 excitatory (plastic) weights. Weights from all excitatory input neurons are in red, the subset of weights belonging to the familiar stimulus are overlayed in purple. The means of the two groups ("familiar" weights vs rest) are in bold. The plasticity rule used is $\tau_{EE}^{pre} = \tau_{IE}^{post} = 100ms, \alpha_{EE} = 1, \beta_{EE} = -0.8, \kappa_{EE} = \gamma_{EE} = -1$.



Supplementary Figure S6: **Additional analysis of the implicit feedforward toy model**: **A**: Similar parameter sweeps as in fig. 4B, but varying other parameter combinations. **B**: Same parameter sweep as in fig. 4B, but for an angle of $\frac{\pi}{3}$ between the two inputs. **C**: Grey: dynamics of the $\theta_0 = 0, \theta_1 = -1$ rule from the explicit parameterization in the A-B-A task. Black dots represent the fixed points of the rule associated to $a = 1, b = 0, c = 0$ in the implicit parameterization. **D**: Parameter sweep on the plasticity rules (varying a and b, c fixed. But relaxing the assumption that Σ^{-1} needs to be positive semi-definite.



Supplementary Figure S7: **Additional analysis on testing predictions from toy models. A, B, C**: Top: firing rate of the post-synaptic neuron during simulation associated to fig. 5B&C (red), as well as the firing rate in response to the novel and familiar stimuli. Bottom: inhibitory weights, weights from inhibitory input neurons aprt of the familiar stimulus are in purple, the rest is in teal. Averages of the two groups are in bold. Dashed lines indicate training onset and offset. **D**: Recurrent spiking network with variants of iSTDP (I-to-E plasticity only) in the familiarity task, with different Hebbian windows. Simulations were repeated 5 times, averages across seeds are shown in bold, dashed lines denote the last timestep at which the population firing rate in response to familiar stimuli was significantly different to novel responses (Student t-test, $p < 0.05$).



Supplementary Figure S8: **Extending the explicit toy model to higher dimensions. A**: Top: Phase portrait of the relative improvement RI as a function of the values of θ_0 and θ_1. Note that here, RI is defined as in appendix A.3.2, to extend to $N_{\text{in}} > 2$ input dimensions. Grey denotes unstable rules. Bottom: Same parameter sweep as C, but plotting the time to convergence τ. These two plots are generated for $N_{\text{in}} = 2$ (same as main). **B,C**: Same as A, but for $N_{\text{in}} = 3$ and $N_{\text{in}} = 10$.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We show simulations and derivations in four models (recurrent and feedforward spiking networks, two toy feedforward linear models).

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We included a dedicated paragraph in the discussion. We include and discuss negative results from some large scale numerical experiments in the last figure.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings,

model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.

- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: For the three parts of this paper that rely on analytical results (mean-field analysis for I-to-E plasticity to compute target firing rates), and the two toy models, we included a dedicated section in appendix for the full derivation.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We included a more detailed method section in Supplementary with the exact parameter values to run our models. The code to reproduce our simulations and analysis is also provided.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.

- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The dataset for meta-learned rule quadruplets is publicly accessible from previous public work. The code to reproduce all simulations and analysis is available on github.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: Supplementary methods go into more details for the exact experiments presented in the paper.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: All statistical tests performed are described, and the code provided includes the analysis.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Paragraph in Supplementary material about the compute-requirements of this work.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.

- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification:

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: this paper is a very fundamental analysis of the emergence of memory in the brain, and has no immediate societal impact.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: the models used in this study have no foreseeable potential for misuse.

    Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All libraries upon which the simulations and analysis is built on are cited.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: a link to the code is provided.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

Justification: No human subjects are used in this study.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: No human subjects were used in this study.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.