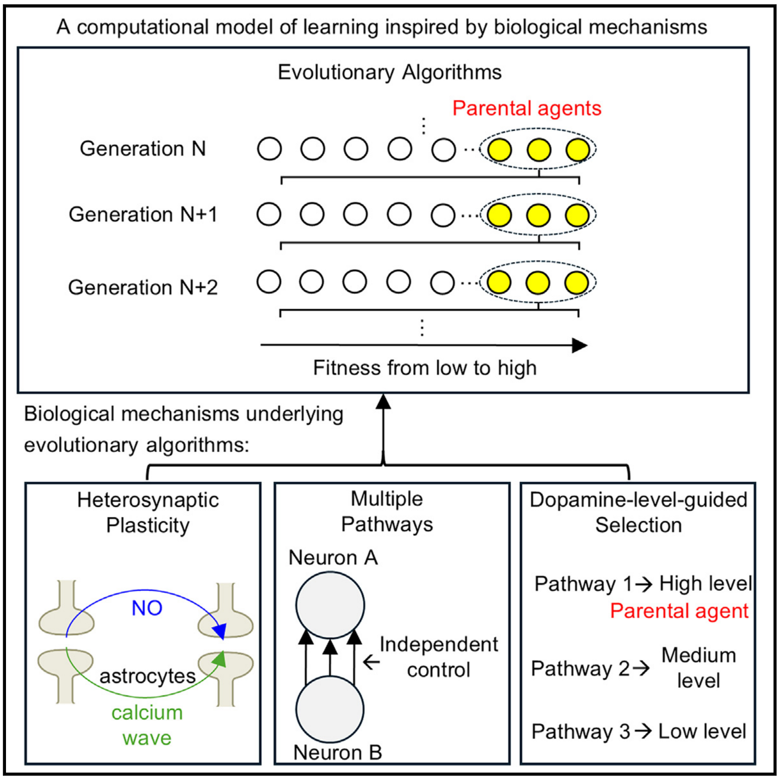
iScience

Article

# Evolutionary learning in neural networks by heterosynaptic plasticity

## Graphical abstract



## Highlights

- Evolutionary algorithms (EAs) offer a gradient-free alternative to backpropagation
- EA model draws inspiration from various biological mechanisms
- Heterosynaptic plasticity aids network training through the EA model
- EA model trains versatile networks, matching brain-like dynamics, and top-tier performance

## Authors

Zedong Bi, Ruiqi Fu, Guozhang Chen, Dongping Yang, Yu Zhou, Liang Tian

## Correspondence

zedong.bi@outlook.com (Z.B.),
liangtian@hkbu.edu.hk (L.T.)

## In brief

Biological sciences; Neuroscience; Biophysics

Bi et al., 2025, iScience 28, 112340
May 16, 2025 © 2025 The Authors. Published by Elsevier Inc.
https://doi.org/10.1016/j.isci.2025.112340

CellPress

iScience

CellPress
OPEN ACCESS

Article

# Evolutionary learning in neural networks by heterosynaptic plasticity

Zedong Bi,[1,8,9,*] Ruiqi Fu,[2,8] Guozhang Chen,[3] Dongping Yang,[4] Yu Zhou,[5] and Liang Tian[2,6,7,*]
[1]Lingang Laboratory, Shanghai 200031, China
[2]Department of Physics, Hong Kong Baptist University, Hong Kong, China
[3]National Key Laboratory for Multimedia Information Processing, School of Computer Science, Peking University, Beijing, China
[4]Research Institute of Artificial Intelligence, Zhejiang Lab, Hangzhou 311121, China
[5]School of Life Sciences and Health, University of Health and Rehabilitation Sciences, Qingdao, Shandong 266011, China
[6]Institute of Computational and Theoretical Studies, Hong Kong Baptist University, Hong Kong, China
[7]Institute of Systems Medicine and Health Sciences, Hong Kong Baptist University, Hong Kong, China
[8]These authors contributed equally
[9]Lead contact
*Correspondence: zedong.bi@outlook.com (Z.B.), liangtian@hkbu.edu.hk (L.T.)
https://doi.org/10.1016/j.isci.2025.112340

## SUMMARY

Training biophysical neuron models provides insights into brain circuits' organization and problem-solving capabilities. Traditional training methods like backpropagation face challenges with complex models due to instability and gradient issues. We explore evolutionary algorithms (EAs) combined with heterosynaptic plasticity as a gradient-free alternative. Our EA models agents with distinct neuron information routes, evaluated via alternating gating, and guided by dopamine-driven plasticity. This model draws inspiration from various biological mechanisms, such as dopamine function, dendritic spine meta-plasticity, memory replay, and cooperative synaptic plasticity within dendritic neighborhoods. Neural networks trained with this model recapitulate brain-like dynamics during cognition. Our method effectively trains spiking and analog neural networks in both feedforward and recurrent architectures, it also achieves performance in tasks like MNIST classification and Atari games comparable to gradient-based methods. Overall, this research extends training approaches for biophysical neuron models, offering a robust alternative to traditional algorithms.

## INTRODUCTION

Training networks with biophysically neuron models provides insights into the cellular-level learning mechanisms underlying brain intelligence.[1,2] An outstanding question in this area is how neuronal networks adapt their components to improve task performance, a challenge known as the "credit assignment problem".[3]

In machine learning, stochastic gradient descent via backpropagation (BP) is a primary method for solving the credit assignment problem.[4] In BP, synaptic weights are updated according to locally available signals propagated through feedback connections.[5,6] However, BP applies gradient information in a manner that is biologically implausible for calculating synaptic weight updates, and it has historically been viewed as problematic in many contexts.[6] Firstly, BP requires stable, differentiable states for effective gradient accumulation, but biophysical models are multi-dimensional nonlinear systems with rapid, discontinuous dynamics due to "spikes" in voltage and state variables. This complexity may lead to numerical stiffness and instability, causing BP to become unstable and divergent when training these models.[2] Secondly, the signed error signals sent backwards through a network in BP often vary widely in magnitude during training, leading to the phenomena of exploding and vanishing gradients.[7] Besides, implementing such feedback of signed errors in complex, deep networks, such as the cortex, is problematic.[8] Moreover, BP requires the adjustment of weights throughout hierarchical structures. However, given certain neuroanatomical insights,[9–13] the dominance of cortico-centric, hierarchical architectures in deep learning, and predictive coding networks is increasingly questioned.[14] These challenges suggest that there may be critical biological factors that BP does not account for highlighting the need to explore alternative methods for training biophysically based neural networks. Our investigation focuses on two such overlooked aspects: an algorithmic framework distinct from gradient-based algorithms and a paradigm for synaptic plasticity that extends beyond local rules, as elaborated in the following text:

Recently, an increasing number of studies have focused on harnessing evolutionary algorithms (EAs) to optimize neural networks.[15] Inspired by natural evolution, EAs involve the evolution of a population of agents (Figure 1A). Each generation of agents are reproduced by the parental agents (i.e., the agents with higher fitness, or in other words, better task performance) in the previous generation through mutation or cross-over; elite agents (i.e., the agents with the highest fitness) become agents in the next generation without mutation, a technique called

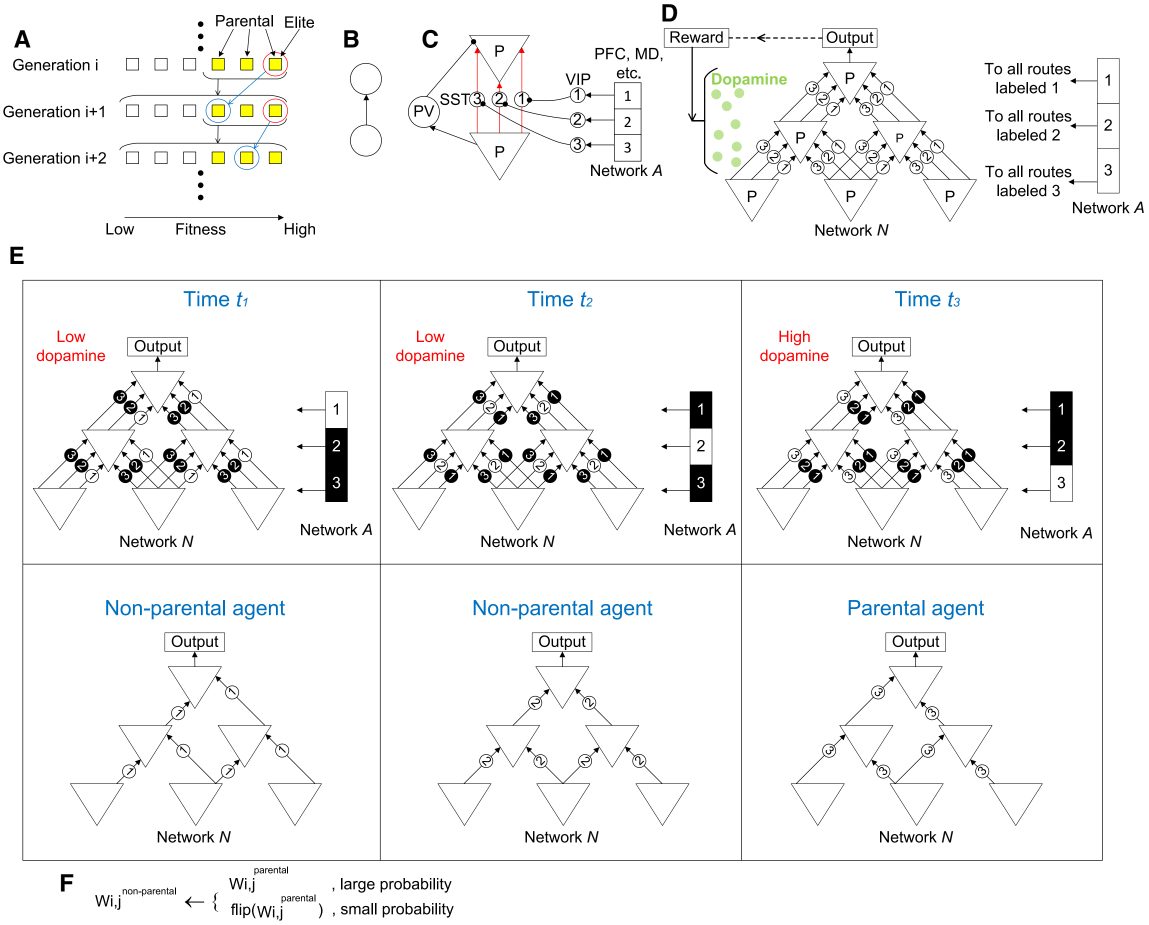


**Figure 1. The network architecture**

(A) Diagram of the evolutionary algorithm: In each generation, agents (depicted as squares) reproduce from multiple high-fitness parental agents (yellow squares) of the previous generation (indicated by black arrows and brackets). The agent with the highest fitness (marked by a red circle) is carried over to the next generation without mutations (illustrated by blue arrows and circles), a method known as elitism.

(B) Traditional models often assume a single connection between two neurons.

(C) Our model supposes that two pyramidal neurons are connected through multiple dendritic routes, which are gated by MD or PFC (network $\mathcal{A}$) through SST and VIP interneurons. Only synapses (red) between pyramidal neurons are plastic in our model, all the other synapses to and from PV, SST and VIP interneurons remain fixed.

(D) Dendritic routes labeled $k$ ($k = 1,2,3$) in network $\mathcal{N}$ are gated by neurons labeled $k$ in network $\mathcal{A}$. Dopamine, which represents the evaluation of the output of network $\mathcal{N}$, are released dispersedly to network $\mathcal{N}$.

(E) Neurons in network $\mathcal{A}$ are activated alternately one at a time. At time $t_k$ ($k = 1,2,3$), only neuron $k$ in network $\mathcal{A}$ is activated (white square) with others silent (black squares), so that the dendritic routes labeled $k$ in network $\mathcal{N}$ are gated on (white circles) with others gated off (black circles). The sub-network connected by all the $k$th dendritic routes in network $\mathcal{N}$ is regarded as the $k$th agent in evolutionary algorithm (lower subplots). The 3rd agent stimulates a high level of dopamine (red text in the upper-right subplot), is therefore the parental agent; the other two agents stimulate a low level of dopamine, are therefore non-parental agents.

(F) The plasticity rule. With a high probability, the synaptic weight $w_{ij}^{\text{non-parental}}$ from the $j$th to $i$th pyramidal neurons on a non-parental agent directly inherits the synaptic weight $w_{ij}^{\text{parental}}$ on a parental agent; with a small probability, synapses on non-parental agents are updated to be descendants of the synapses on parental agent with some mutations (i.e., weight flip).

elitism. EAs provide a more general framework than gradient-descent algorithms by optimizing both synaptic weights and network connections, which are typically static in gradient-descent approaches.[15] Unlike gradient-based models, EAs do not depend on hierarchical architectures and operate solely in the forward pass, making them robust against noisy and rigid losses.[16] Additionally, EAs handle discrete loss formulations and enable broader parameter exploration, positioning them as a preferred alternative to BP.[2]

Synaptic plasticity rules are believed to be a pivotal physiological mechanism for learning.[6] Many learning mechanisms of neural networks are proposed based on the assumption of locality, i.e., the change of a synapse depends on its pre- and post-synaptic activities, stemming from the Hebbian rule.[17] Nevertheless, synaptic plasticity is not strictly localized: a synapse can also be modified by the activities of adjacent, non-connected neurons through intercellular signaling pathways, a process known as heterosynaptic plasticity.[18,19] For example, the diffusive nitric oxide (NO) associated with a synapse potentiated by pre- and post-synaptic spike pairing can potentiate other neighboring synapses[20,21]; the pre-synaptic release of glutamate can stimulate long-range propagating calcium waves in astrocytes, depressing synapses in a broader spatial range.[22,23] The inhibition of heterosynaptic plasticity has been shown to significantly compromise cognitive function.[24] While heterosynaptic plasticity is known for aiding synaptic balance in Hebbian learning[18] as a secondary function, its potential as a powerful learning paradigm in its own right has not been fully explored.

In this paper, we propose an original non-local heterosynaptic learning mechanism that has not been considered in the machine learning community. By incorporating heterosynaptic plasticity into EA, we present a gradient-free method to train biophysical neuron models. In our framework, the network comprises multiple information routes, each representing an EA agent. The performances of these agents are evaluated through alternatively gating on each route with the others gated off. Agent selection and reproduction are realized by dopamine-guided heterosynaptic plasticity. To examine the possible computational function of heterosynaptic plasticity alone, we do not add any local rule in the model.

We show that our EA model achieves three significant merits simultaneously.

(1) Biological inspiration. Our EA model draws inspiration from a comprehensive range of cellular-level experimental findings, including heterosynaptic potentiation mediated by NO,[20,21] heterosynaptic depression mediated by astrocyte calcium wave,[22,23] dendritic gating,[25] binary synapses,[26] dopamine gating of synaptic plasticity,[27] meta-plasticity,[28,29] memory replay,[30] cooperative plasticity between the synapses within a dendritic neighborhood,[31–33] etc. By integrating these biological components, our EA model trains recurrent neural networks to exhibit dynamics analogous to brain dynamics in experimental observations.[34]

(2) Broad competence. Our EA model is capable of effectively training neural networks, whether they are based on feedforward or recurrent architectures and composed of analog or spiking neurons. This versatility maximizes the generalizability across various learning mechanisms.

(3) Powerful capability. Our EA model successfully trains deep neural networks with binary weights to achieve performance levels in MNIST classification and Atari game-playing tasks that are on pair with those of continuous-weight networks refined by gradient-based methods.

Collectively, our research sheds light on an alternative paradigm for training networks with biophysically neuron models. It expands the scope beyond traditional gradient-based approaches, potentially offering insights into the cellular-level learning mechanisms that underpin brain intelligence.

## RESULTS

### Network architecture

In traditional computational neuroscience models, two neurons are connected by a single connection (Figure 1B). However, this simplification does not reflect the actual complexity found within the brain, where two pyramidal neurons may form multiple synaptic connections across various dendrites (Figure 1C).[35] Each of these dendritic pathways can be independently regulated by somatostatin (SST) interneurons,[25] which have the capacity to inhibit dendritic activity. These SST interneurons are, in turn, modulated by "hub" regions such as the prefrontal cortex (PFC) or the mediodorsal thalamus (MD).[36] These hubs exert their influence through extensive corticocortical or thalamocortical networks that activate vasoactive intestinal peptide (VIP) interneurons, which then inhibit the SST interneurons.[37,38] Additionally, parvalbumin (PV) interneurons serve to inhibit pyramidal neurons in proximity to the cell body or soma.[25]

These biological mechanisms and neural substrate provide a framework for constructing EA networks (Figure 1A), illustrated in Figures 1D and 1E, where we hypothesize that the dendritic pathways of pyramidal neurons within a network $\mathcal{N}$, representative of the hippocampus or neocortex, are modulated by a controlling network $\mathcal{A}$, indicative of the PFC or MD regions. If neuron *k* in network $\mathcal{A}$ is active, all the dendrites labeled *k* in network $\mathcal{N}$ are gated on, with the others gated off. Neurons in network $\mathcal{A}$ are alternatively activated one at a time, such that dendrites with different labels are gated-on alternatively (Figure 1E, upper subplots). This gating mechanism allows network $\mathcal{N}$ to generate distinct outputs in response to identical stimuli, which in turn yield varying reward outcomes, sourced either from the external environment or internally estimated by the brain.[39] These reward signals stimulate different levels of dopamine, informed dispersedly to the whole network to guide synaptic plasticity[27,40] (Figure 1D). Within this context, the subset of network $\mathcal{N}$ interconnected by dendrites labeled *k* is conceptualized as the *k*th agent of the EA framework. Dopamine level stimulated by an agent indicates the fitness of that agent, such that high (or low) dopamine level indicates parental (or non-parental) agents (Figure 1E, lower subplots).

In the next subsection, we will show that the heterosynaptic plasticity between pyramidal neurons is equivalent to a learning rule: with a high probability, the non-parental synaptic weight $w_{ij}^{\text{non-parental}}$ directly inherits the parental synaptic weight $w_{ij}^{\text{parental}}$. In this manner, the synapse $w_{ij}^{\text{non-parental}}$, which connects the the *j*th to *i*th pyramidal neurons on a non-parental dendrite, becomes equivalent to the synapse $w_{ij}^{\text{parental}}$ on a parental dendrite; with a small probability, the non-parental synaptic weight $w_{ij}^{\text{non-parental}}$ undergoes a "flip" operation, meaning that synapses on non-parental agents are updated to be descendants of the synapses on parental agent with some mutations (i.e., weight flip) (Figure 1F). In this way, non-parental agents are updated to become descendants of the parental agent, undergoing mutations, such as weight flips due to the stochastic nature of the nervous system.[41,42] All the synapses to and from inhibitory interneurons are kept fixed in our model (Figure 1C). Note that we are presently considering only mutational changes; although the potential for cross-over is acknowledged and can be explored in future research (refer to the discussion section for more details).

### Basic learning process

We explore a delayed reward paradigm commonly observed both in natural settings and experimental conditions: neuronal

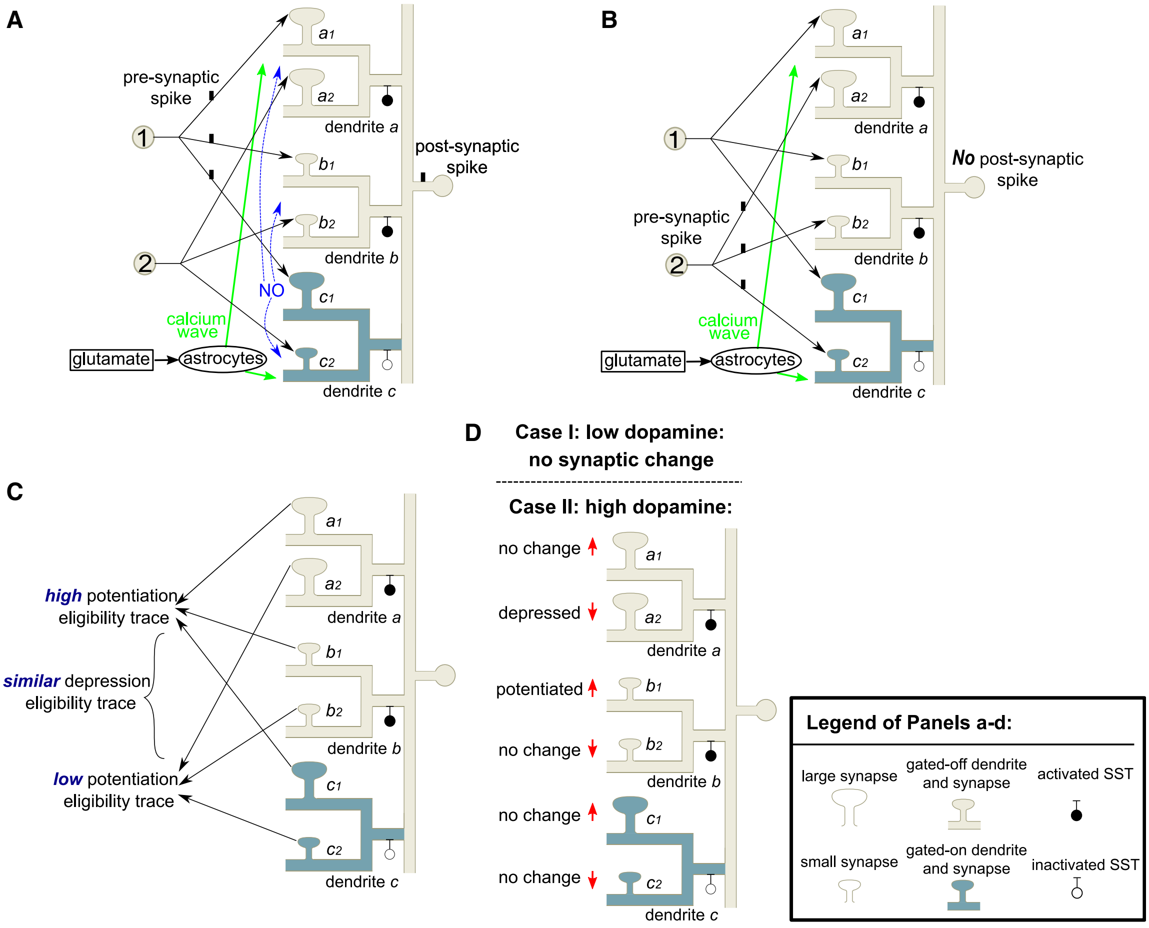


**Figure 2. Heterosynaptic plasticity induces EA**

(A) When neuron 1 emits a spike, the large synapse $c_1$ on the gated-on dendrite $c$ stimulates a post-synaptic spike; this pairing of pre- and post-synaptic spikes produces diffusive NO, which, together with the spikes simultaneously invading synapses $a_1$ and $b_1$ from neuron 1, induces potentiation eligibility traces in these synapses.

(B) When neuron 2 emits a spike, the small synapse $c_2$ on the gated-on dendrite $c$ does not stimulate a post-synaptic spike, therefore no potentiation eligibility traces in synapses $a_2$ and $b_2$. In both (A) and (B), pre-synaptic release of glutamate stimulates astrocytes to produce widely spreading calcium waves, which induce depression eligibility traces in all synapses.

(C) The synapses targeted by neuron 1 have higher potentiation eligibility traces than those targeted by neuron 2, but the depression eligibility traces in these synapses are similar.

(D) Case I: with a low dopamine level, no synaptic change happens. Case II: with a high dopamine level, the synapses $a_1$, $b_1$, $c_1$ (or $a_2$, $b_2$, $c_2$) with high (or low) potentiation eligibility traces tend to be potentiated (or depressed), represented by upper (or lower) red arrows. Synapses $a_1$, $c_1$ (or $b_2$, $c_2$) remain unchanged because they already have large (or small) efficacy.

activities result in eligibility traces in synapses, according to which sparse and delayed rewards then guide synaptic plasticity. Eligibility traces are molecular states in synapses, which can change synaptic efficacies under a high dopamine level but have no effect under a low dopamine level.[43] The aforementioned plasticity rule (Figure 1F) is modeled after heterosynaptic plasticity, drawing on the following cellular-level mechanisms. To illustrate the mechanisms, we consider a scenario involving two pre-synaptic neurons that form connections with a single post-synaptic neuron across three dendritic branches labeled *a*, *b*, and *c*, each featuring binary synaptic efficacy[26] (represented by large or small boutons in Figure 2); at a certain time step, dendrite *c* is gated on (see Figure 2).

(1) If a synapse on the gated-on dendrite *c* has large efficacy (e.g., synapse $c_1$ from neuron 1 in Figure 2A), a pre-synaptic spike from neuron 1 will be very likely to stimulate a post-synaptic spike. This pairing of pre- and post-synaptic spikes produces diffusive messengers such as nitric oxide (NO).[20,44] This NO and simultaneously arriving spikes from neuron 1 generate *high* potentiation eligibility traces in both $c_1$ and neighboring synapses ($a_1$ and $b_1$) from the same pre-synaptic neuron (i.e., neuron 1) on gated-off dendrites *a* and *b*,[20,21,45] see Figure 2C. This heterosynaptic potentiation is subject to a spatial scope comparable with the scale of the dendritic arbor of a pyramidal neuron (~ 150 $\mu$m in hippocampal CA1[20,21]).

Synapses from neuron 2 (i.e., synapses $a_2$, $b_2$ and $c_2$) cannot be potentiated by the diffusive NO without simultaneous pre-synaptic spikes.

(2) If a synapse on a gated-on dendrite *c* has small efficacy (e.g., synapse $c_2$ from neuron 2 in Figure 2B), a pre-synaptic spike from neuron 2 will be unlikely to stimulate a post-synaptic spike, resulting in *low* potentiation eligibility traces in both $c_2$ and neighboring synapses ($a_2$ and $b_2$) from the same pre-synaptic neuron (i.e., neuron 2) on gated-off dendrites *a* and *b*, see Figure 2C.

(3) Pre-synaptic release of glutamate stimulates astrocytes to spread calcium waves[23,46] (Figures 1A and 1B), which induces depression eligibility traces in synapses spatially widespread[22,47,48] (at least 300 $\sim$ 500 in hippocampal CA1[22,49]). The depression eligibility traces on different synapses are supposed to be similar in our model due to the widespread spatial scale, see Figure 2C.

Collectively, if a synapse on a gated-on dendrite has high (or low) efficacy, it will induce high (or low) potentiation eligibility traces in the neighboring synapses from the same pre-synaptic neuron on gated-off dendrites; the depression eligibility traces on different synapses are similar (Figure 2C). These processes are well-documented in the hippocampus[18] (also see the references aforementioned) and have also been observed in the neocortex.[19,45,48,50,51]

If the delayed reward is small so that the dopamine level is low (i.e., the gated-on dendrite *c* is non-parental, see Figure 1E), no synaptic change will happen (case I in Figure 2D). If the delayed reward is large so that the dopamine level is high (i.e., the gated-on dendrite *c* is parental, see Figure 1E), the synapses will be updated[27] according to the following rules (case II in Figure 2D): synapses with large potentiation eligibility traces will be potentiated, while synapses with small potentiation eligibility traces will be depressed by the depression eligibility trace. Given that biological synapses can have binary efficacies,[26] potentiation (or depression) only works in synapses with low (or high) efficacy, whereas those already with high (or low) efficacy remain unchanged (Figure 2D). As a result, synapses on non-parental dendrites are updated toward those on parental dendrites connecting the same pair of pre- and post-synaptic neurons (i.e., synapses $a_1$ and $b_1$ are updated toward $c_1$, whereas $a_2$ and $b_2$ toward $c_2$), while those on parental dendrites (i.e., synapses $c_1$ and $c_2$) remain unchanged (Figure 2D), fulfilling the requirement of the EA (Figure 1F).

## Multiple parents and elitism realized by meta-plasticity

In EA of computer science, each generation usually has multiple parents and at least one elite[52,53] (Figure 1A). Here, we show that both multiple parents and elitism can be implemented in our model by introducing a meta-plasticity mechanism. In this mechanism, if a synapse with large (or small) efficacy on a gated-on dendrite receives potentiation (or depression) signals under high dopamine level, the efficacy of this synapse will get stabler at the large (or small) level, harder to be switched in future learning processes (Figure 3A). This mechanism ensures that high-efficacy synapses (akin to elite agent in EA) are preserved across learning episodes, making them less susceptible to the normal fluctuations of synaptic modification, thereby embodying the principle of elitism within our neural model.

To understand the effect of this meta-plasticity, suppose dendrite *a* is parental at time step 1. Without the meta-plasticity, if another dendrite *b* is parental at a later time step 2, synapses on dendrite *b* will reproduce their efficacies on dendrite *a* (Figure 3B, lower panel), under the mechanism illustrated in case II of Figure 2D. In this case, dendrite *b* will become the parent of dendrite *a*. With the meta-plasticity, however, the synapses on dendrite *a* are stabilized, without changing toward the synapses on dendrite *b*, so that dendrite *b* will not become the parent of dendrite *a* (Figure 3C, lower panel). When extending this consideration to the training of multiple dendrites, it is observed that in the meta-plasticity-free scenario, each dendrite at any given time step has a single parent (Figure 3D). However, the introduction of meta-plasticity allows dendrites to have multiple parents (Figure 3E). Moreover, even with the meta-plasticity, synapses on a stabilized dendrite may also gradually lose their synaptic configuration after a sufficiently long time. The parental dendrite that induces the highest dopamine level (i.e., the elite) has the highest stability, and therefore will be maintained for the longest time, fulfilling the elitism technique in EA (Figure 3E, black boxes).

Physiologically, it is well known that dopamine stabilizes potentiated synapses of pyramidal neurons, after the excitability of pyramidal neurons is increased by the inhibition of SST or PV interneurons[28,54,55] (upper subplot of Figure 3A). However, the stabilization of small synapses (lower subplot of Figure 3A), despite some clues,[56,57] remains to be experimentally tested. A better-established process to fulfill the stabilization of small synapses is dendrite-level synaptic homeostasis,[58,59] which constrains the total synaptic efficacy on a dendrite (Figure 3F). Under this constraint, small synapses are hard to be potentiated if other large synapses on the same dendrite are stabilized, realizing the stabilization of small synapses.

## EA is compatible with offline replay

The no-free-lunch theorem posits that no one learning algorithm can excel across all possible problems.[60] Consequently, it is essential for the brain to utilize a diverse array of learning strategies to optimize the animal's chances of survival. Similarly, it is imperative for researchers to discern the contexts in which EA is most effective to deepen our understanding of their functional roles. In this discussion, we demonstrate that EA is compatible with offline replay.

Consider the following on-line learning scenario which might occur in an awake animal (scenario 1, Figure 4A). In a learning session, the performance of a gated-on dendritic route in response to an input stimulus is scored by the reward feedback from the real world, inducing a corresponding dopamine level; learning sessions may be separated by intervals of variable durations. This scenario is not compatible with EA. To see this, consider a situation where dendritic route 1 brings 10 units of reward in response to stimulus 1 and brings 0 units in response to stimulus 2; dendritic route 2 brings 6 units of reward in response to both stimuli 1 and 2 (Figure 4C). If high dopamine is induced only in the session corresponding to the highest reward (i.e., the session with stimulus 1 and dendritic route 1

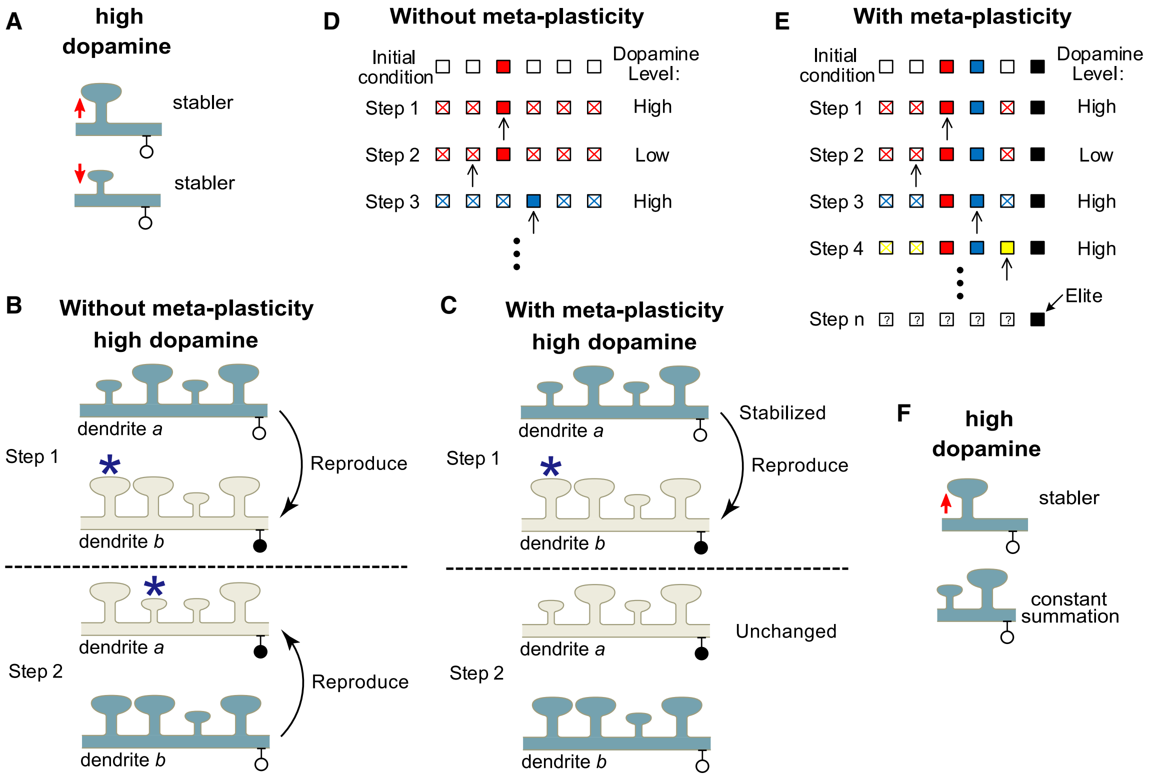


**Figure 3. Multiple parents and elitism realized by meta-plasticity**

(A) Schematic of the meta-plasticity. On a gated-on dendrite (white circles represent inactivated SST interneurons, see the legend of Figure 2), if a large (upper panel) or small (lower panel) synapse receives potentiation or depression signal (upward or downward arrow) at a high dopamine level, the synapse will be stabilized at large or small efficacy.

(B) Schematic of the case without meta-plasticity. At time step 1 (or 2), the gated-on dendrite *a* (or *b*) is parental, so a high dopamine level is induced at both time steps. At step 1 (upper panel), the synaptic configuration of dendrite *a* is reproduced to dendrite *b*. At step 2 (lower panel), the synaptic configuration of dendrite *b* is reproduced to dendrite *a*. The asterisks indicate mutations (i.e., $\epsilon$ in Figure 1F) that happen with a small probability.

(C) With meta-plasticity, the synapses on dendrite *a* are stabilized when dendrite *a* is parental (upper panel), so that when dendrite *b* is parental at a later time step (lower panel), the synapses on dendrite *a* will not change.

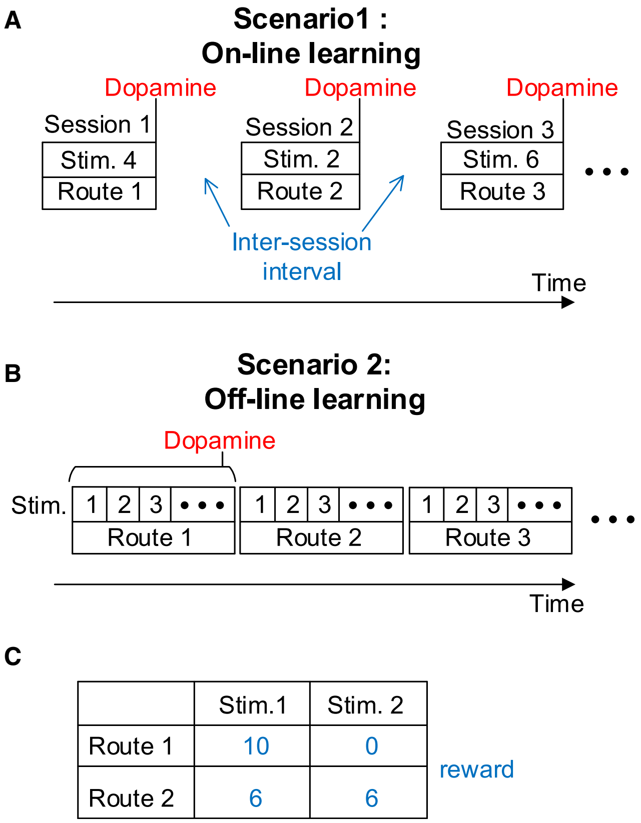
(D) The learning process without meta-plasticity. Each box represents an agent in EA, i.e., a dendritic route in Figure 1E. The initial condition contains one parental agent (red square) that can induce a high dopamine level and five non-parental agents (white squares). In step 1, the red parental agents are activated (black arrow), so that all non-parental agents are updated toward the red agent, becoming its descendants (red crosses). In step 2, the activated agent cannot induce a high dopamine level, so this agent remains non-parental. In step 3, the activated agent induces a high dopamine level, so all the other agents (including the red parental agent) are updated toward the activated agent, becoming its descendants (blue crosses). We see that the agents at every time step have a single parent and no elite without meta-plasticity.

(E) The learning process with meta-plasticity. Notice that the parental agents (red, blue, yellow and black squares) are stabilized, so that in step 1, 3, or 4, non-activated parental agents (i.e., the parental agents not indicated by the black arrow) do not update toward the activated parental agent (i.e., the parental agent indicated by the black arrow) that stimulates high dopamine level. After many steps (last row), parental agents may also lose their synaptic configurations (the question marks mean we do not know the states of those agents); but the elite agent (black square), which can induce the highest dopamine level, is the stablest one, and can last for the longest time.

(F) The stabilization of small synapses can be realized by stabilizing large synapses (upper panel) together with dendrite-level synaptic homeostasis (lower panel).

gated-on), dendritic route 1 will be parental and the synapses on dendritic route 2 will update toward those on dendritic route 1. However, this parental selection is incorrect because it is dendritic route 2 that brings more rewards on average over the two input stimuli, aligning better with the task objective of maximizing the expected average reward (Figure 4C).

This problem can be solved if we suppose that the evaluation of dendritic routes happens during offline replay (scenario 2, Figure 4B): a gated-on dendritic route is quickly evaluated over a number of replayed inputs, and the dopamine level is released according to the average reward associated with a number of input stimuli just replayed.[61] This averaging is crucial because the efficacy of a dendritic route can vary significantly depending on the stimulus it receives. While a particular route might perform poorly for most stimuli, it could perform exceptionally well for specific ones. Hence, to ensure a fair and robust evaluation, we average the performance outcomes across a representative sample of stimuli, as depicted in Figure 4B. In this case, the dopamine level indicates the average performance of the gated-on dendritic route over a batch of input samples just replayed. If we set the batch size to be 2, the dendritic route 2 will be correctly selected as the parental agent during the replay



| | Stim.1 | Stim. 2 |
|---|---|---|
| Route 1 | 10 | 0 |
| Route 2 | 6 | 6 |

**Figure 4. EA is compatible with offline replay**

(A) During an on-line learning scenario, the performance of a gated-on dendritic route in response to an input stimulus is represented by the level of dopamine given at the end of each session. There are intervals of variable durations between sessions (blue text).

(B) During an offline learning scenario, the model can quickly sweep over samples of input stimuli when a dendritic route $k$ ($k = 1, 2, 3, \cdots$) is gated on. Note that the stimuli presented in sequential order (1, 2, 3, …) represent stimuli that have been inputted. The dopamine level represents the average reward associated with the inputted stimuli when the same dendritic route is gated on.

(C) An example situation. Route 1 brings the highest reward (bold blue text) in response to a single stimulus, but route 2 brings the highest average reward (italic blue text) over both stimuli.

in the case of Figure 4C. This method of averaged evaluation does not conflict with the sequential evaluation of parental agents shown in Figure 3E. In both scenarios, each dendritic route is assessed across a range of stimuli to ensure that the evaluation captures a comprehensive view of its performance. Specifically, in Figure 3E, each evaluation step involves sweeping over a sample of stimuli to assess a dendritic route effectively. In conclusion, two levels of replay are involved in the model: first, a gated-on dendritic route is evaluated over a series of stimuli, through the replay of stimuli over a short timescale; second, the dendritic routes are evaluated by alternatively gating on each route while the others are gated off, through the replay of gating-control neurons (as we illustrated in Figure 1D, the network $\mathcal{A}$ is indicative of the neurons in hub areas such as the PFC or MD regions) over a long timescale (see more details in discussion).

The fundamental mismatch between EA and online learning arises from the inherent characteristics of EA. In gradient-descent algorithms, the direction for updating a synaptic weight is computable and corresponds to the average of directions derived from individual input stimuli samples. Consequently, the neural network can be optimized by accumulating small updating steps in response to individual stimuli during online learning, as seen in scenario 1. However, in EA, the synaptic updating direction cannot be directly computed. Instead, it relies on comparing the performance of different dendritic routes. This comparison depends on the performance across multiple stimuli samples (serving as an estimation of performance over the entire set of stimuli) rather than a single stimulus, rendering online learning less appropriate. This offline replay scenario, illustrated in Figure 4B, is a prediction of our model. In the following simulations, EA is performed in an offline manner.

## Cooperative plasticity between synapses from synchronous inputs

From Figure 2A, the simultaneous arrival of the pre-synaptic spikes from neuron 1 at synapses $a_1$, $b_1$, and $c_1$ is a key factor for the potentiation of $a_1$ and $b_1$. The inter-cellular diffusion of NO suggests that such potentiation still happens even if the spikes invading $a_1$ and $b_1$ do not come from neuron 1 but from a different neuron with synchronous firing with neuron 1. This subsection will investigate the EA process when single post-synaptic neurons receive from synchronous inputs from different pre-synaptic neurons.

Pre-synaptic neurons with synchronous activities target close dendritic locations[62,63] through synapses with cooperative plasticity: when a synapse is potentiated, the neighboring synapses can also be potentiated at the same time[33] or become easy to be potentiated by only weak stimulation[31]; when a synapse is depressed, the neighboring synapses can also be depressed at the same time.[32] In our model, the synapses from synchronous-firing pre-synaptic neurons to the same post-synaptic dendrite are mutated downward or upward simultaneously (group mutation, GM) instead of independently (individual mutation, IM), modeling the cooperative plasticity between neighboring synapses due to cellular mechanisms in the post-synaptic dendrite.[31,32] We found that GM is essential for the success of EA learning.

To illustrate the mechanisms, we consider three pre-synaptic neurons (1, 2, 3) with synchronous activities targeting a post-synaptic neuron through two dendrites ($a$, $b$), see Figures 5A and 5B. At a certain time, dendrite $a$ is gated on while dendrite $b$ is gated off, and the synapses from the three pre-synaptic neurons on dendrite $a$ have large efficacies at the beginning (Figure 5A, left column). Under IM, each synapse is mutated independently with low probability. In this case, suppose synapse $a_3$ on dendrite $a$ is mutated downward (Figure 5A, upper row). High NO is emitted from the large-efficacy synapses $a_1$ and $a_2$ on the gated-on dendrite $a$ (Figure 5A, upper row, middle column) when the spikes from neurons 1 and 2 invading $a_1$ and $a_2$. When this happens, the spikes from neurons 1 and 2 arrive at synapses $b_1$ and $b_2$ simultaneously, and the spikes from neuron 3 also arrive at synapses $a_3$ and $b_3$ simultaneously due to the synchronous activity of neuron 3 with neurons 1 and 2. Therefore, there are high potentiation eligibility traces in all the

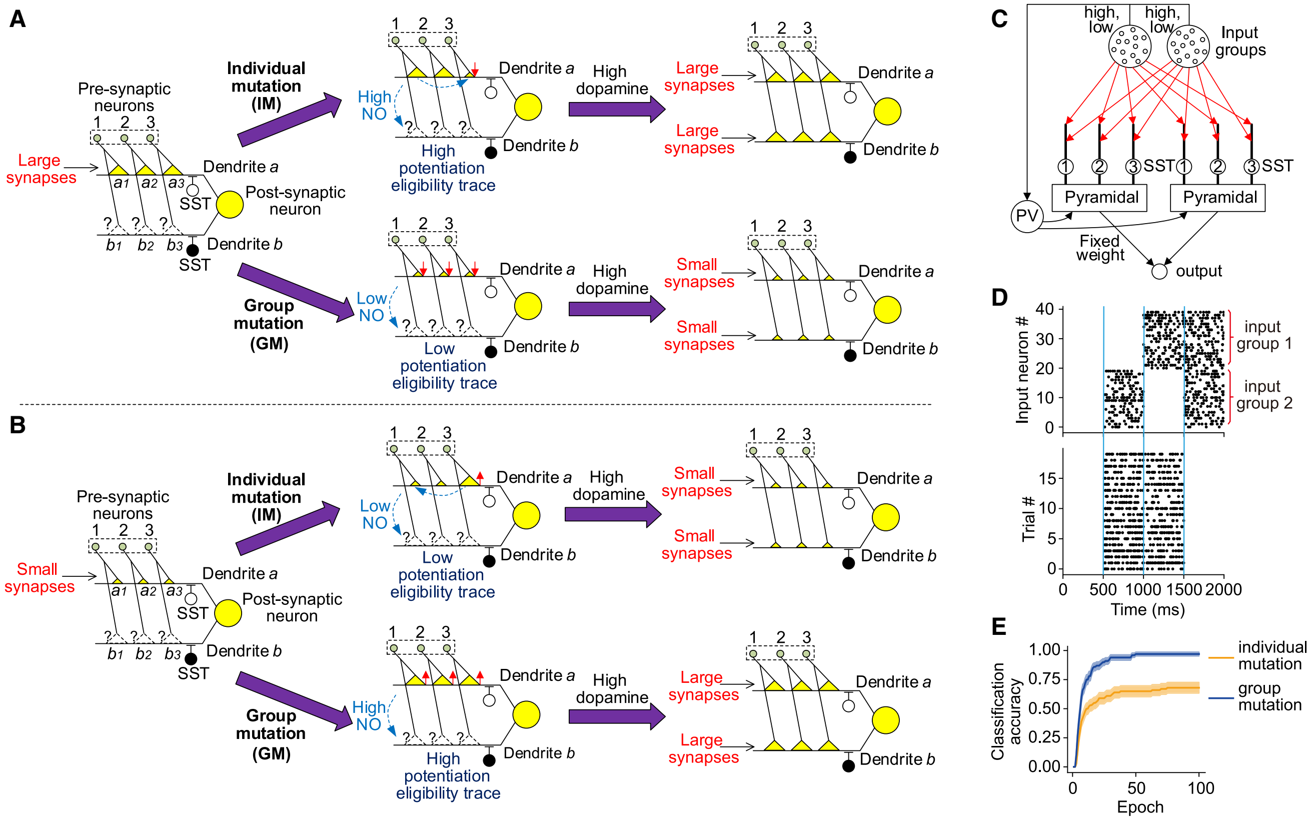


**Figure 5. The computational function of the cooperative plasticity**

(A and B) Schematic to illustrate the advantage of group mutation. The green circles in dashed boxes represent pre-synaptic neurons with synchronous activity. Large (or small) yellow triangles represent large (or small) synapses. Dashed triangles with question marks represent synapses with no matter large or small efficacy. Dendrite *a* or *b* is gated on or off by inactivated or activated SST interneurons (white or black circle). Upward (or downward) arrows indicate that the synapses are mutated upward (or downward). If the gated-on dendrite *a* is parental (i.e., induces high dopamine) after IM (or GM), its mutated configuration will not (or will) be reproduced to dendrite *b*.

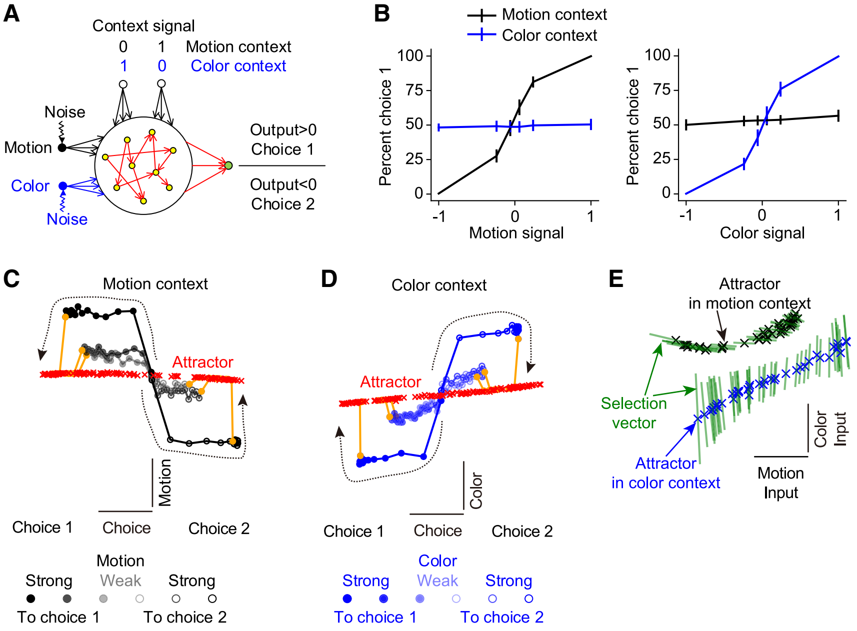
(C) The structure of the network to perform the XOR task. Pyramidal neurons receive from two input groups (whose activity levels can be high or low) through synapses (red arrows) on dendrites gated by SST interneurons. Red synapses are to be changed during training, and the other synapses are fixed.

(D) Raster plot of the input neurons in one simulation trial (upper panel) and the single output neuron in multiple trials (lower panel) after training under group mutation. Vertical blue lines delimit four intervals in which the two input groups have different high or low activities.

(E) Classification accuracy during the training process for individual (orange) or group (blue) mutation. Error belts indicate the standard error of the mean (s.e.m.) over 100 training trials.

synapses ($a_1$, $a_2$, $a_3$, $b_1$, $b_2$, $b_3$) on both dendrites (Figure 5A, upper row, middle column). If the gated-on dendrite *a* is parental so that high dopamine is induced, all these synapses will become large (Figure 5A, upper row, right column): this means that the parental synaptic configuration on the parental dendrite *a* after IM (i.e., synapses $a_1$, $a_2$, and $a_3$ have large, large, and small efficacies, respectively) cannot be reproduced to dendrite *b*, which contradicts with the EA requirement (Figure 1F). Under GM, however, the synapses from the three pre-synaptic neurons on dendrite *a* are mutated downward simultaneously (Figure 5A, lower row, middle column). In this case, the NO received by the synapses on both dendrites is low, inducing low potentiation eligibility traces. If dendrite *a* is parental so that high dopamine is induced, all the synapses in both dendrites will become small (Figure 5A, lower row, right column): this means that the parental synaptic configuration on the parental dendrite *a* after GM (i.e., synapses $a_1$, $a_2$, and $a_3$ all have small efficacies) gets reproduced to dendrite *b*. A similar situation happens when the three synapses on dendrite *a* have small efficacies at the beginning (Figure 5B, left column): if dendrite *a* is parental (i.e., induces high dopamine) after IM (or GM), its mutated configuration will not (or will) be reproduced to dendrite *b* (Figure 5B), inconsistent (or consistent) with the EA requirement (Figure 1F).

We demonstrate the aforementioned mechanism by training a spiking neural network on the Exclusive OR (XOR) task. In this network (Figure 5C), a population of pyramidal neurons receives Poisson spike trains emitted from two neuronal groups, and then give output to a single neuron. The synapses from the two input groups to the pyramidal neurons (i.e., red arrows in Figure 5C) should be trained so that the output neuron has a high firing rate when one input group has a high firing rate while the other is silent, and the output neuron has a low firing rate when both input groups have high or low firing rates (Figure 5D). In this model, neurons belonging to the same input group have



### Figure 6. EA-trained neural networks mimic brain dynamics

(A) The architecture of the neural network model utilizes a context signal, either (0,1) or (1,0), to specify whether it is operating within a motion or color context. Within the given context, the network must select either 1 or 2 based on whether the relevant signal—motion or color, which is compromised by noise—exceeds zero. Importantly, it disregards the signal not specified by the context, either color or motion, respectively.

(B) Psychometric curve, which shows the percentage of choice 1 as a function of the motion (left panel) or color (right panel) signal in the motion (black curve) or color (blue curve) context for the neural network after trained by EA. Note that in the motion (or color) context, this percentage increases with the motion (or color) signal, but is irrelevant to the color (or motion) signal. Error bars represent SEM over 8 training trials.

(C) The dynamics of the neural population within the motion context are projected into a subspace defined by the motion and choice axes. These axes are specifically determined so that the projections of the neural population's activity onto these axes best correlate with the motion signal and the output choice, respectively. Each fixed point, marked by a red cross, has an eigenvalue close to zero, these fixed points collectively approximate a line attractor. The states of the neural network start near the center of the panel and flow along the black dots and curves (the dots and curves with different grayscales represent trajectories at motion signals of different strength) toward upper left or lower right, indicated by the dashed arrows; then the motion signal is turned off, so that the states of the neural network relax back toward the line attractor along the orange curves. Analogous to Figures 2 and 5 of the study by Hiratani N. and Fukai T.[34]

(D) Similar to (C), but in the color context and in the subspace spanned by the axes of choice and color.

(E) The line attractor (black or blue crosses) and selection vector (green) at each fixed point, in the subspace spanned by the motion and color input weights. Inputs are selected by the selection vector and integrated along the line attractor. Note that the selection vector of the attractor in the motion (or color) context is orthogonal to the color (or motion) input: this means that the dynamics of the network near the line attractor is almost irrelevant to the color (or motion) input in the motion (or color) context. Analogous to Figure 6C of the study by Hiratani N. and Fukai T.[34]

correlated activities (Figure 5D, upper panel), and therefore have higher firing synchrony than the neurons belonging to different input groups.

We investigated the performance of two synaptic mutation strategies: IM, where each synapse mutates independently, and GM, where synapses from neurons belonging to the same input group and projecting onto the same post-synaptic dendrite mutate in a coordinated fashion. Our findings indicate that GM is a superior strategy compared to IM. Specifically, GM allows the network to rapidly converge to a high level of classification accuracy. Conversely, with IM, the network's performance tends to plateau at a lower level of accuracy (Figure 5E). Therefore, cooperative plasticity between synapses from synchronous input neurons is necessary for EA to train the neural network successfully. See more details in Figure S1. We also demonstrate the proportion of synaptic weight flips during training process for IM and GM in the XOR task. Our findings show that the proportion of synaptic weight flips remains generally below 5% in each epoch. See more details in Figure S4.

## EA-trained neural networks mimic brain dynamics

To further explore our EA model, we trained recurrent neural networks to perform a context-dependent decision-making task[34] using EA. We then compared the resulting dynamics of these EA-trained neural networks with those observed in biological brains and with neural networks trained using BP.[34] The task required the networks to make a binary choice based on one of two input channels. These channels correspond to the motion or color coherence of random dots, similar to the experiment described.[34] Depending on whether the context signal pertains to motion or color (Figure 6A), the network must choose option 1 or 2 based on whether the motion (or color) coherence is positive or negative, while ignoring the color (or motion) signal. The networks consist of firing-rate units, with each unit representing a group of spiking neurons that exhibit correlated activity, analogous to input groups shown in Figure 5C) and reflective of neuronal groupings in the brain. We trained the neural networks using the following simple algorithm (*EA-Simple*) to model the group mutation (Figures 5A and 5B) of the synapses from the same neuronal group.

(1) Each synaptic weight takes binary values, one positive $\Delta w$, one negative $-\Delta w$, modeling the effective synaptic weight when the excitatory synapse has large or small efficacy under the global inhibition of PV interneurons (Figure 5C).
(2) Descendants are mutated from parental agents with a small probability of synaptic flip (Figure 1F).

EA-trained networks mimic the brain and BP-trained networks in the following four aspects.[34] First, EA-trained networks have successfully learned the task, as evidenced by their

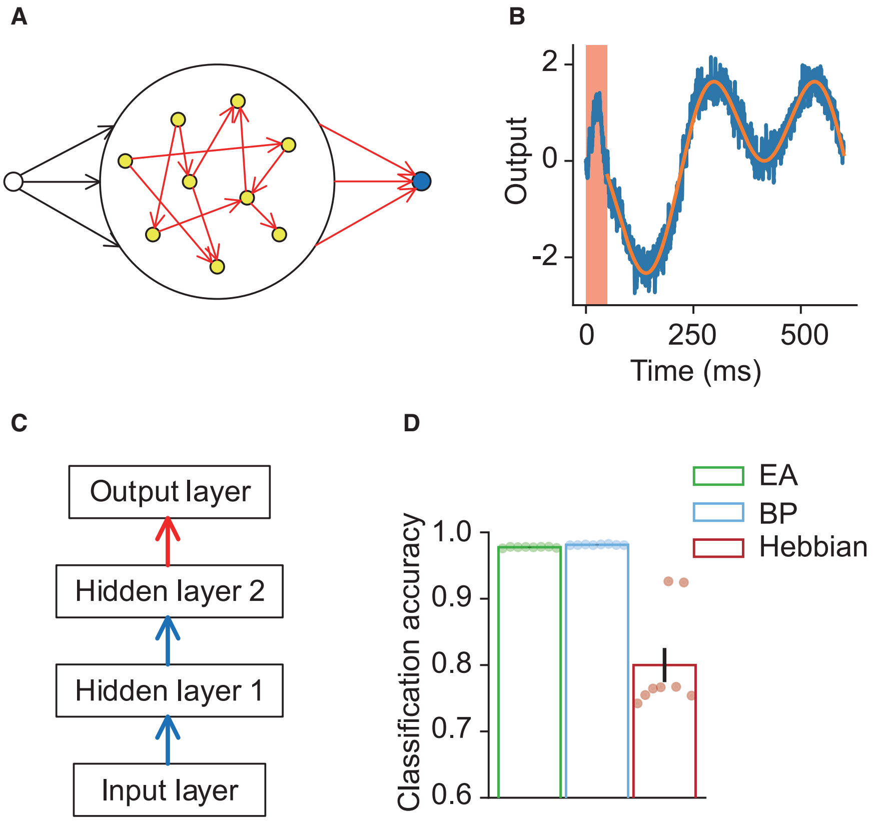


**Figure 7. Training spiking and analog neural networks with recurrent and feedforward architectures using EA**

(A) Schematic of the SNN to generate trajectories. Recurrent units (yellow circles) are sparsely connected. The output unit (blue circle) is trained to generate an arbitrary trajectory. Both the recurrent and output connections (red arrows) are binary and plastic.

(B) An example of the actual output (blue) and target (orange) trajectory. The input signal is on during the initial 50 ms (red shading) and off afterward.

(C) The architecture of the ANN to classify MNIST images. When studying the Hebbian algorithm in (D), the lower layers were trained using the Hebbian algorithm (blue arrows), and the top layer was trained by gradient-descent algorithm supervisedly (red arrows).

(D) Classification accuracy on the test dataset when training the deep network using BP, EA, and Hebbian rule. Error bars represent SEM over 8 training trials.

behavioral psychometric functions. These functions demonstrate that the networks can accurately perform the required binary choice based on the motion or color coherence cues (Figure 6B). Second, when the networks make different choices, the trajectories of their state spaces align with a line attractor model. These trajectories diverge in directions parallel to the line attractor, with their separation distance being proportional to the coherence strength of the inputs. This behavior suggests that the network state changes systematically with the strength of the evidence (Figures 6C and 6D). Third, after defining three axes that respectively capture the most across-trial variance in the state space due to the choice, the motion, and color coherence, we found that context usually has no substantial effect on the directions of the axes of choice, motion, and color (Figure S2C). This invariance to context indicates that the decision-making process is robust and consistent, regardless of the specific contextual cues, reflecting a neural mechanism that maintains stable decision criteria across different conditions. Fourth, the selection vectors (i.e., the left eigenvector of the largest eigenvalue) of a line of attractors are aligned with the motion (or color) input and orthogonal to the color (or motion) input in the motion (or color) context (Figure 6E), which indicates that the relevant input pushes the network state along the direction of the line attractor, whereas the irrelevant input has no effect. This orthogonality suggests that the networks can effectively isolate and process relevant sensory information while ignoring irrelevant inputs, a characteristic feature of neural processing in the brain.

Together, these results suggest that EA-trained networks can emulate the complex dynamics of the brain during cognitive tasks. It hints at the possibility that similar computational principles may be at work in the learning processes of biological neural systems. This resemblance to biological processes is not only theoretically intriguing but could also have practical implications for the design of artificial neural networks and the understanding of cognitive functions in neuroscience. See more details in Figure S2.

### EA is a broadly competent and powerful learning paradigm

To demonstrate that EA is competent for various network architectures and capable of training neural networks for complicated tasks, we trained neural networks with binary weights using the EA-Simple algorithm on various tasks.

We first trained a sparse recurrent spiking neural network (SNN) to produce an arbitrary time-dependent output trajectory[64,65] (Figure 7A, see STAR Methods). Despite the binary nature of the weights in both recurrent and output synapses, the network was able to closely replicate the desired trajectory (Figures 7B and S5A).

We then trained a feedforward analog neural network (ANN) with two hidden layers (Figure 7C) to classify the modified national institute of standards and technology (MNIST) images.[66] The classification performance of this binary-weight network was on par with that of a continuous-weight network with an identical architecture trained using BP (Figures 7B and S5B). We also trained continuous-weight networks layer-by-layer using a competitive Hebbian algorithm, except for the last layer, which was trained using a supervised gradient-descent algorithm (Figure 7C), and found worse final performance (Figure 7D, third bar). Interestingly, if we supervisedly trained the last layer

**Table 1. Binary-weight networks trained by EA are competitive with continuous-weight networks trained by DQN and A3C in Atari games**

| Game name | DQN, cw [16] | A3C, cw [16] | EA, cw [16] | EA, bw, ours | Hebbian-Q, cw, ours |
|---|---|---|---|---|---|
| amidar | 978 | 264 | 263 | 251 | 10 |
| assault | 4,280 | 5,475 | 714 | 873 | 130 |
| asterix | 4,359 | 22,140 | 1,850 | 1,923 | 92 |
| asteroids | 1,365 | 4,475 | 1,661 | 1,596[a] | 146 |
| atlantis | 279,987 | 911,091 | 76,273 | 74,563 | 1,400 |
| enduro | 729 | −82 | 60 | 32[b] | 5 |
| frostbite | 797 | 191 | 4,536 | 4,520[c] | 30 |
| gravitar | 473 | 304 | 476 | 589[c] | 26 |
| kangaroo | 7,259 | 94 | 3,790 | 3,997[b] | 151 |
| seaquest | 5,861 | 2,355 | 798 | 800 | 85 |
| skiing | −13,062 | −10,911 | −6,502 | −6,503[c] | −28,383 |
| venture | 163 | 23 | 969 | 1,070[c] | 22 |
| zaxxon | 5,363 | 24,622 | 6,180 | 6,195[a] | 24 |

[a] The scores of EA that are higher than those of DQN, but lower than A3C.

[b] The scores of EA that are higher than those of A3C, but lower than DQN.

[c] The scores of EA-trained binary-weight (bw) networks that are higher than both the scores of DQN- and A3C-trained continuous-weight (cw) networks.

on top of the first hidden layer, the performance got better than the two-hidden-layer case (Figures S5C and S5D), similar to the finding in the study by Amato G. et al.[67] This phenomenon implies that Hebbian algorithms cannot coordinate weights in different layers to fulfill better performance.

Our experiments further showcase the effectiveness of EA by training deep neural networks with binary weights to play Atari games. The performance of these EA-trained networks is on par with that of continuous-weight networks that have been trained using gradient-based methods, such as deep Q-networks (DQN) and asynchronous advantage actor-critic (A3C),[68,69] as well as other EA-based methods[16] (Table 1). In a comparative experiment, we employed a hybrid Hebbian-Q learning algorithm, wherein the lower layers of the network were trained using a competitive Hebbian learning algorithm,[70] while the top layer was trained using Q-learning.[68] The performance of networks trained with this Hebbian-Q learning algorithm was markedly inferior to that achieved by networks trained using DQN, A3C, and EA.

These results demonstrate that our EA is a broadly competent algorithm capable of training deep neural networks on complicated tasks.

## DISCUSSION

Overall, by unifying a broad spectrum of experimental evidence into a coherent picture (Figures 1, 2, 3, 4, and 5), we demonstrate that incorporating heterosynaptic plasticity, in conjunction with dendrite gating, offers a foundation for constructing EA networks. Our neural network's dynamics closely align with experimental observations,[34] suggesting that while not proving the brain uses EA for learning, EA-based models can replicate brain dynamics without contradicting experimental evidence (Figure 6). This supports the potential utility of EA in modeling aspects of neural function. Our EA model manifests its broad competence and powerful capability in training both spiking and analog neural networks (Figure 7; Table 1). The successful application of our EA model across different neural network architectures and complex tasks underscores its significance as a tool for understanding and replicating the sophisticated learning functions of the brain. A key contribution of this work is demonstrating how these biological mechanisms and neural substrates serve as a foundation for constructing EA networks, with corresponding implementations within the EA framework that effectively simulate these biological processes (Table 2), though we acknowledge limitations in the model's biological plausibility due to necessary computational simplifications (see more details in limitations of the study). This will equip future neuroscientists with a fresh lens through which they can interpret data and formulate predictions, complementing traditional frameworks centered around local learning rules and backpropagation.

### Computational advantages of EA

Here, we propose that our EA model is realized by heterosynaptic plasticity mechanisms. To examine the possible computational function of heterosynaptic plasticity alone, we do not add any local rule in the model. Compared to homosynaptic mechanisms (i.e., local plasticity rules) reviewed in Lillicrap T.P. et al.,[8] our mechanism has the following computational advantages.

(1) Model-free approach. Our model does not necessitate detailed knowledge of the network architecture or the specific objectives of the task at hand. In contrast to backpropagation-based algorithms, which rely on intricate feedback systems to direct synaptic modifications,

Table 2. Biological mechanisms have corresponding implementations within EA framework

| Biological mechanisms | Implementation in EA |
|---|---|
| Multiple dendrites between neurons (Figure 1E) | Agents |
| Dopamine concentration (Figure 1D) | Evaluation of agents |
| Heterosynaptic plasticity (Figure 2) | Reproduction |
| Neural noise (Figures 3B and 3C), Co-plasticity of nearby synapses (Figure 5) | Mutation |
| Meta-plasticity (Figure 3) | Elitism |
| Memory replay (Figure 4) | Offline learning |

A main contribution of this work is demonstrating how these biological mechanisms and neural substrates serve as a foundation for constructing EA networks, with corresponding implementations within the EA framework that effectively simulate these biological processes.

synaptic updates in our EA model arise from spontaneous mutations. This key difference underpins broad applicability of our EA model and training algorithm across various network architectures.

(2) Scalability via agent-limited evaluation. Our EA algorithm aims to select the optimal pattern from an exponentially growing number of patterns (e.g., for a subcircuit of $N$ neurons, under the assumption of binary synapses, the possible connectivity patterns is $2^{N(N-1)}$). Although the total number of patterns grows exponentially, the number we need to evaluate per iteration to find the optimal one is limited to the number of agents, keeping the computational load manageable.

(3) Multi-objective optimization. EA excels at addressing multi-objective optimization problems, which are ubiquitous in real-world scenarios.[71] These problems typically involve navigating trade-offs between competing objectives. For example, the process of purchasing a house. Buyers often face a conflict between cost and location; a budget-friendly house is rarely found in a downtown area. Consequently, a potential homeowner might weigh options between an expensive residence in the city center and a more affordable one in the suburbs, each offering distinct advantages in terms of convenience or cost. Gradient-based optimization methods tend to approach such dilemmas by constructing a single objective function. This function is a weighted linear combination of the various goals, with the weights reflecting predetermined preferences. A homebuyer might assign greater weight to either the price or the location based on their priorities. In contrast, EA can explore a spectrum of potential solutions through a population of agents, each evolving toward different trade-offs. This approach enables the discovery of the Pareto front: a set of optimal solutions where no objective can be improved without worsening another.[71] Therefore, EA provides a more creative and comprehensive strategy for tackling problems with conflicting objectives, revealing a range of viable solutions that might not be immediately apparent through gradient-based methods.

(4) Inspired by biological mechanisms. Natural selection creates our brain, endowing us with amazing learning capabilities, such as meta-learning,[72] zero-shot (or few-shot) learning,[73] transfer learning,[74] continuous learning,[75] etc. Therefore, EA resonates with a universal principle for the evolutionary development of high intelligence, especially in the context of complex environments. By incorporating biological learning mechanism, our EA is capable of replicating brain dynamics without contradicting experimental evidence, while also optimizing its adaptability to solve different tasks. In essence, EA expands the scope beyond traditional gradient-based approaches and local rules, potentially offering insights into the cellular-level learning mechanisms that underpin brain intelligence.

(5) Alternative to BP. BP is a highly successful algorithm in deep learning, which requires precise error signals to be transmitted through feedback connections, a process that seems problematic in biological brains.[8] Furthermore, BP relies on the use of the same weight matrices for both forward and backward paths, which is not feasible in biological neural networks due to the physical independence of synapses in feedforward and feedback pathways.[76] To address these limitations, alternative methods such as feedback alignment,[76,77] and weight mirror[77] have been proposed to circumvent the weight transport issue. Feedback alignment and weight mirror address the weight transport problem by using random fixed matrices and adjusting feedback weights to approximate transpose of forward weights, respectively. Also, the forward-forward algorithm replaces the forward and backward passes of BP with two forward passes, thus avoiding backpropagation of the error signal.[77,78] The PEPITA algorithm achieves local updates by modulating the input during the second forward pass using information about the error from the first forward pass[77] and shares similar learning principles with forward-forward. Although these alternative methods improve BP to varying degrees by employing a forward-only approach, their weight updates depend on local information. In our work, we introduce an EA that also operates solely in the forward pass. Unlike the aforementioned methods, our EA does not rely on local information updates. Inspired by the heterosynaptic plasticity mechanism observed in the brain, our EA represents an original non-local heterosynaptic learning mechanism not previously considered in the machine learning community.

## Potential strategies to speed up learning

Compared to gradient-based methods, EA exhibits a slower learning speed. For instance, classifying MNIST images requires upwards of 10,000 learning epochs when using EA (Figure S4B), while gradient-based methods might only need dozens to hundreds of epochs. However, the rate of learning may not be a pivotal concern under certain circumstances because EA is believed to primarily facilitate offline learning (Figure 4). Here, we list potential strategies that may speed up learning.

(1) Combination with homosynaptic mechanisms. One such strategy involves the integration of homosynaptic mechanisms that align with gradient-based methods.[8] While our model is designed to showcase the distinct effectiveness of heterosynaptic plasticity, hence omitting any local homosynaptic rules, the inclusion of these rules could direct mutations along the gradient, providing a hint toward the optimal solution and, consequently, speeding up the learning process. This integrated approach could marry the broad search capabilities of EA with the faster convergence of gradient-based learning, potentially leading to a more efficient learning paradigm.

(2) Curriculum learning. It involves learning the easier aspects of the task first and then gradually increasing the task difficulty, which can significantly improve the speed and success rate of learning complicated tasks.[79,80] To achieve optimal learning effects, the brain may adapt the contents of replay (so that the replayed contents are neither too simple nor too complex) to its current capability in solving problems: a process called autocurriculum.[80] It has been found that the brain prioritizes high-rewarded memories for replay,[81] which may be the mechanism of autocurriculum: the brain automatically adjusts the complexity of replayed contents and prioritizes the highest-rewarded situations within the current problem-solving ability.

(3) Incorporation of cross-over operations. A third strategy is to add cross-over operation into EA. A recent theoretical study[82] suggests that compared to EA with only mutation, EA that utilizes cross-over operation exhibits a polynomial order of improvement in speed relative to the number of agents. Our model currently incorporates only the mutation operation; however, it can be easily expanded to cross-over. See more details in Figure S6.

## Simplification of EA in comparison to biological systems

To facilitate a clear and comprehensible presentation of the core concepts to our readers, we have intentionally streamlined our model by omitting certain biological complexities. Here we provide a thorough discussion of the reasons behind these simplifications and explore the potential impact of these simplifications in the following texts.

(1) Exclusion of local rules. We acknowledge the simultaneous existence of both local and non-local rules in the brain.[83] However, in our simulation, we did not incorporate any local rules into the model. This simplification facilitates the isolation and focused examination of the effects of non-local rules of heterosynaptic plasticity, thereby streamlining our analysis of these less-understood mechanisms. For instance, local rules such as 3-factor rules and the role of inhibitory synapse plasticity are not included in the EA model. The concept of 3-factor rules posits that synapse plasticity depends on the activities of the pre- and post-synaptic neurons and is additionally modulated by dopamine levels, categorizing it within the domain of local rules.[84] Furthermore, in our current model, all synaptic connections to and from inhibitory interneurons are fixed. We maintain fixed synapses to and from inhibitory interneurons for two reasons: first, existing experimental literature provides a clearer understanding of the heterosynaptic plasticity mechanisms in excitatory neurons.[35] In contrast, the mechanisms for inhibitory synapses are primarily aimed at achieving network homeostasis. Second, our study isolates the computation of heterosynaptic plasticity (a non-local rule) by excluding local rules, including inhibitory synapse plasticity. While this could potentially be addressed by introducing an indirect connection via an inhibitory synapse, such a modification would allow the net output from one excitatory neuron to another to be inhibitory, which aligns with established neurobiological principle. We recognize the crucial role that plasticity in inhibitory synapses plays in maintaining neuronal homeostasis and enhancing the network's feature selectivity.[85] To clearly delineate the distinct contributions of non-local plasticity rules without the confounding effects of local synaptic changes, our model intentionally excludes changes in inhibitory efficacies. Even though local rules are currently not incorporated in our model, we believe that non-local and local rules are not mutually exclusive, they may work together to enhance both the learning capabilities and the stability of the model.[83]

(2) Non-consideration of Dale's law. Dale's law mandates that neurons are exclusively either excitatory or inhibitory,[86] a principle not strictly enforced in our initial model setup, except for the XOR task. For example, in our simulation, the presence of inhibitory synapses directly between excitatory neurons not strictly follows the setting of Dale's law. We acknowledge the significance of Dale's law in neural architecture; however, our decision to not strictly enforce Dale's law in our initial model setup is informed by prior computational studies.[34,87] These studies indicate that networks can replicate dynamics observed in biological experiments even without strict adherence to Dale's law. This suggests that while Dale's law is a fundamental biological principle, its rigorous implementation may not be essential for achieving dynamics that qualitatively resemble those seen in the brain. Despite the limitations of our simplified framework, we believe it provides a valuable foundation for further exploration into how EA can be constructed based on various biological factors. Future research could investigate more complex activation patterns and interactions between agents to enhance and validate the biological plausibility of our model.

(3) Omission of VIP neurons and axonal arbors in the simulation. VIP neurons, known for their role in inhibiting SST neurons and receiving long-range connections from other brain areas,[37,38] were depicted in the schematic diagram (Figure 1C) to illustrate the biological inspiration behind our computational model. Their inclusion was intended to provide a comprehensive view of the interneuronal interactions that can influence cortical processing and plasticity, which are pivotal to the theoretical framework of our study. However, to simplify the computational models for

the simulations, we decided to omit VIP neurons from the subsequent numerical analysis. This decision was aimed at streamlining the simulation process and concentrating on the fundamental aspects of our model, thereby avoiding the additional complexity that the inclusion of VIP neurons would introduce at this stage of our research. Also, our work focuses on dendritic rather than axonal arbors due to the more robust inhibitory control systems present in dendrites. Research shows that dendrites are modulated by inhibitory neurons, such as SST interneurons,[25] which effectively regulate dendritic activity. While axonal control systems may exist, they are less well characterized. To clearly articulate our model's intentions and leverage well-understood biological mechanisms, we have chosen not to include axonal arbors in our current study.

## Neurophysiological underpinnings of replay involved in EA

This dynamic reconfiguration of information routes (Figure 1E) is a fundamental hypothesis of our EA model, essential for encapsulating the multi-agent dynamics of EA within the neural framework. We speculate that this dynamic reconfiguration is similar to how slow timescale replay events in hub regions like the PFC and MD might control the sequential gating-on of dendritic routes in other brain areas,[30,88] particularly during offline replay (Figure 4). We discuss the neurophysiological foundations of multiple timescales and the dynamic reconfiguration of our EA as given in the following texts.

(1) Multiple timescales of replay. Here, two levels of replay are involved in our model. First, a gated-on dendritic route is evaluated over a series of stimuli, through the replay of stimuli over a short timescale (as in Figure 4B). Second, the dendritic routes are evaluated by alternatively gating on each route while the others are gated off, through the replay of gating-control neurons (as we illustrated in Figure 1D, the network $\mathcal{A}$ is indicative of the neurons in hub areas such as the PFC or MD regions) over a long timescale. This resembles the activities across multiple timescales during sleep or in a resting state, where the hippocampus handles fast timescale replay for immediate evaluation, and the PFC manages slower timescale information integration.[89] The rapid replay in the hippocampus, often occurring during slow-wave sleep, facilitates the transfer of information to the neocortex for long-term storage.[90] Conversely, PFC is involved in slow replay events, which are typically associated with the integration and long-term consolidation of memories. During sleep, specific patterns of brain waves, such as sleep spindles and slow oscillations in the PFC, support these processes, allowing for the gradual integration of information over extended periods.[91] This multi-timescale dynamic interaction of hippocampus and PFC during sleep facilitates the consolidation and integration of memories.[89] In our EA model, the processes of short timescale stimulus replay and long timescale gating-control neuron replay closely resemble the multi-time-scale replay of hippocampus and PFC during sleep or resting state. Such collaboration is likely crucial for the sequential, one-at-a-time activation process in our model. Furthermore, this sequential activation may be driven by these multi-timescale replays, which also suggests that our model is well-suited for offline replay.

(2) Offline replay learning shares similar neurophysiological conditions as sleep. The compatibility of EA with offline replay suggests that EA might play a crucial role in the transformation of episodic memory into unconscious, dexterous skills during periods of rest or sleep. It is well established in cognitive neuroscience that certain cognitive processes, such as memory replay and consolidation, are predominantly facilitated during sleep.[92] We propose that our EA might operate under similar neurophysiological conditions, leveraging the unique environment that sleep provides for such cognitive processes. This includes: (1) neurotransmitter levels: sleep is characterized by significant reductions in neurotransmitters like acetylcholine, noradrenaline, and cortisol, creating an optimal environment for memory consolidation;[93] (2) hormonal levels: elevated melatonin levels during the evening, which regulate the sleep-wake cycle, are favorable for memory consolidation;[92] (3) brain wave activity: sleep-specific brain wave patterns, such as spindles and sharp wave-ripples, are crucial for memory consolidation[94] and may provide the neural dynamics that facilitates the selective modulation of synaptic connections required by our EA. In essence, our EA might mirror the memory consolidation process, occurring under specific neurophysiological conditions related to neurotransmitter and hormonal levels, and brain wave activity, thereby operating offline during sleep.

These perspectives shed light on innovative experimental research. We encourage investigators to delve into the mechanisms and effects of dynamic route reconfiguration in neural microcircuits. Examining the brain's modulation of informational pathways across various states may provide profound insights into learning and memory consolidation processes. Pursuing such research could substantially advance our comprehension of neural plasticity and the brain's capacity for adaptation and evolution through experience.

## Limitations of the study

Our model, while simplified and in need of further empirical validation, effectively introduces potential insights into the cellular-level learning mechanisms that underpin brain intelligence. However, our limitations stem from a lack of strong evidence supporting the biological plausibility of the one-at-a-time activation mechanism, the number of agents required to perform different tasks, and its extension to a population of neurons. We offer our speculations on these points and address the limitations of the EA framework as follows.

## Speculations about neuroscience involved in EA

(1) One-at-a-time activation. The primary focus of our study is to delve into how biological mechanisms and

neural substrates found in the brain can help to construct EA networks. We have established a set of implementations within the EA framework that effectively simulate these biological processes (Table 2). Our EA model represents a simplified framework where the model comprises multiple information routes, each represented by an EA agent. The one-at-a-time activation mechanism involves evaluating agents by alternately activating each route while deactivating the others. Although there is no direct evidence supporting this mechanism, we provide indirect connections. In the brain, multiple timescale replay levels exist during sleep or rest: the hippocampus handles fast timescale replay for immediate evaluation, while the PFC manages slow timescale information integration. Our model simulates this with two replay levels: first, by evaluating a gated-on dendritic route over a series of stimuli on a short timescale; second, by alternately gating and evaluating dendritic routes over a long timescale using gating-control neurons. This one-at-a-time activation mechanism resembles the slow timescale replay in hub regions like the PFC and MD, which may sequentially gate dendritic routes in other brain areas, facilitating sequential activation and reinforcement of neuronal pathways.[89] However, given the brain's vastly more complex and concurrent information processing capabilities, the one-at-a-time activation mechanism is insufficient to fully capture the complexity of biological scenarios, many situations that are biologically plausible are not accommodated within our model, as illustrated in Figure 8A. In Figure 8A, the ideal scenario represents the simplified mode supported by our model, where only one agent is activated at each time. In contrast, non-ideal scenario A demonstrates multiple agents being activated simultaneously at each time, while non-ideal scenario B shows multiple agents undergoing partial activation concurrently. Both of these non-ideal scenarios are biologically plausible, but our model does not account for them. This discrepancy highlights the model's reduced biological plausibility compared to actual neural system functionalities. We are developing another algorithm that overcomes the constraints of the one-at-a-time activation mechanism. In essence, the one-at-a-time activation approach is a fundamental hypothesis of our model, essential for encapsulating the multi-agent dynamics of EA within the neural framework. While the lack of direct evidence to support one-at-a-time activation is a limitation of our model, our EA model with heterosynaptic plasticity offers a valuable perspective for training biophysical neuron models and represents an original non-local heterosynaptic learning mechanism not previously considered in the machine learning community.

(2) Task-specific agent requirements and population coding. An additional consideration that may challenge the study is the potential requirement for a substantial number of agents to successfully navigate complex tasks. For example, in the context-dependent decision-making task, we observed that if the number of agents is smaller than 50, the performance of the EA model starts to deteriorate (Figure S3C); this contrasts with the observation that performance hardly decreases if the number of agents is as small as 2 in the simpler XOR task (Figures S3A and S3B). This implies that for complex tasks, a considerable quantity of agents may be necessary to sustain peak performance. However, as the number of agents increases, the number of dendritic compartments requiring independent control also grows, which may be challenging to reconcile with biological reality. To address this concern, we propose a supplementary interpretation wherein each neuron in our model (see Figure 2) could be representative of a cluster of physically proximate pyramidal neurons rather than an individual neuron (Figure 8B, upper panel). It aligns with the concept of population coding, a well-accepted framework in computational neuroscience that emphasizes the collective behavior of neurons.[95,96] Specifically, populations are defined by the synchronization of neural firing patterns, where neurons sharing response properties or engaged in the same cognitive task exhibit coordinated activity.[97–100] Within the framework of population coding, heterosynaptic mechanisms (Figure 2) mediated by intercellular pathways remain effective even when synapses are located on different pre- or post-synaptic neurons, provided they are in close spatial proximity.[45] Since neuronal populations are composed of numerous neurons (Figure 8B, middle panel), it is plausible that many independently controlled connections exist between two populations during complex tasks such as context-dependent decision-making task. This population coding approach reflects the cooperative nature of brain regions, where neurons often act in concert rather than in isolation.[99] However, this approach introduces ambiguity in defining population boundaries due to uncertainties in neuronal synchronization, leading to unclear delineation between different neuronal populations (Figure 8B, lower panel). Additionally, due to concerns about simplicity and computational resources, we opted not to develop detailed numerical experiments on neuronal populations.

Despite the limitations inherent in our simplified framework, we believe it serves as a valuable starting point for how EA can be constructed based on various biological factors. Future research could delve into more complex activation patterns and interactions between agents to enhance and validate the biological plausibility of our model.

## RESOURCE AVAILABILITY

### Lead contact

Requests for further information and resources should be directed to and will be fulfilled by the lead contact, Zedong Bi (zedong.bi@outlook.com).

### Materials availability

This study did not generate new materials.

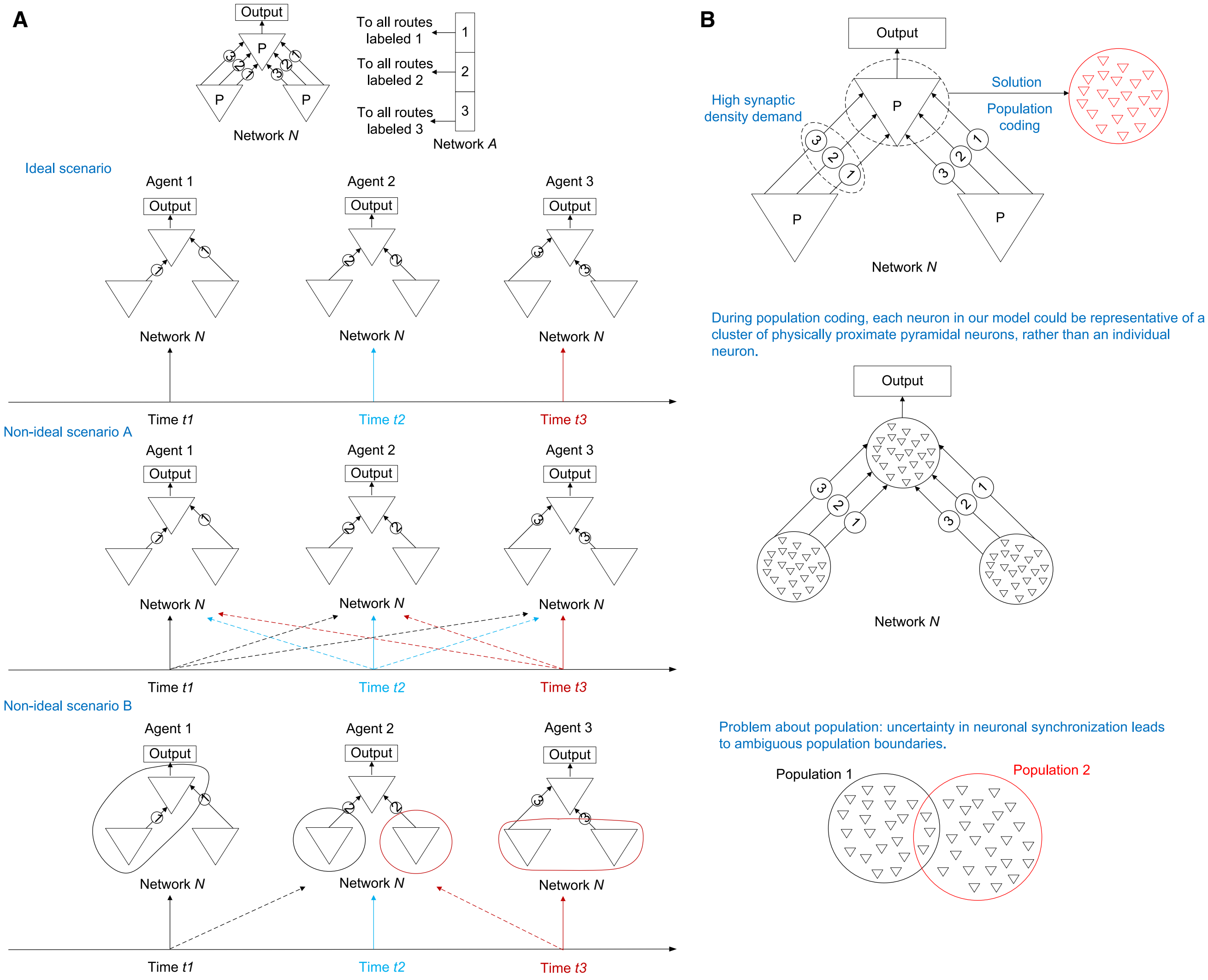

A
Output
P
P
P
Network N
To all routes labeled 1
To all routes labeled 2
To all routes labeled 3
1
2
3
Network A
Ideal scenario
Agent 1
Agent 2
Agent 3
Output
Network N
Time t1
Time t2
Time t3
Non-ideal scenario A
Non-ideal scenario B
B
Output
P
High synaptic density demand
Solution
Population coding
Network N
During population coding, each neuron in our model could be representative of a cluster of physically proximate pyramidal neurons, rather than an individual neuron.
Output
Network N
Problem about population: uncertainty in neuronal synchronization leads to ambiguous population boundaries.
Population 1
Population 2

**Figure 8. Biological plausibility limitations of EA**

(A) Our EA model employs one-at-a-time activation pattern, where each agent is activated sequentially at distinct time $t_k$ ($k = 1,2,3$) (i.e., the ideal scenario where black, blue, and red solid arrows point to agents 1, 2, and 3 at times $t_1$, $t_2$ and $t_3$, respectively). However, this approach cannot fully capture the complexity of biological scenarios. For example, non-ideal scenario A depicts multiple agents are activated simultaneously at each time point. Specifically, at $t_1$, agent 1 is primarily activated (black solid arrow) alongside agent 2 and agent 3 (black dashed arrows). Similarly, at $t_2$ and $t_3$, agents 2 and 3 are primarily activated (marked by blue and red solid arrows), respectively, with concurrent activations of the other agents shown by dashed arrows of corresponding colors. In non-ideal scenario B, partial activations of multiple agents at each time point. Specifically, at $t_1$, agent 1 is partially activated (marked by a black solid arrow, with activated neurons highlighted by a black circle), agent 2 is also partially activated (marked by a black dashed arrow, with activated neurons highlighted by a black circle). At $t_3$, agent 3 is partially activated (marked by a red solid arrow, with activated neurons highlighted by a red circle), agent 2 is also partially activated (marked by a red dashed arrow, with activated neurons highlighted by a red circle). These concurrent activation patterns are biologically plausible but are not supported by our model.

(B) When extensive synapses are required between neurons, each neuron in our EA model could be representative of a cluster of physically proximate pyramidal neurons; this aligns with the concept of population coding. However, this population coding approach introduces ambiguity in defining population boundaries due to uncertain neuronal synchronization, leading to unclear delineation between different neuronal populations.

### Data and code availability

All original code is publicly available as of the date of publication. DOIs are listed in the key resources table.

## ACKNOWLEDGMENTS

This work was supported by the Lingang Laboratory Startup Fund (to Z.B.), the National Natural Science Foundation of China (grant no. 32000694 to Z.B., grant no. 12275229 to L.T., grant no. 12175242 to D.Y., and grant nos. 32371211, 32071141 to Y.Z.), the Hong Kong Research Grants Council (grant nos. C2005-22Y, 12301723, and 12301624 to L.T.), the Natural Science Foundation of Zhejiang Province (grant no. LZ24A050007 to D.Y.), the Natural Science Foundation of Shandong Provice (grant no. ZR2019ZD34 to Y.Z.), the Research Initiation Project of Zhejiang Lab (grant no. K2022KI0PI01 to D.Y.), the Hong Kong Chinese Medicine Development Fund (grant no. 22B2/049A to L.T.), and the Hong Kong Baptist University

Research Committee (grant nos. RC-FNRA-IG/23-24/SCI/05 and CRMS/23-24/03 to L.T.).

## AUTHOR CONTRIBUTIONS

Conceptualization, Z.B. and L.T.; methodology, Z.B., R.F., and L.T.; investigation, Z.B., R.F., and L.T.; writing–original draft, Z.B., G.C., D.Y., Y.Z., and L.T.; writing–review and editing, Z.B., R.F., and L.T.; funding acquisition, Z.B., D.Y., and L.T.; resources, Z.B. and L.T.; supervision, Z.B. and L.T.

## DECLARATION OF INTERESTS

The authors declare no competing interests.

## DECLARATION OF GENERATIVE AI AND AI-ASSISTED TECHNOLOGIES IN THE WRITING PROCESS

During the preparation of this work the author(s) used ChatGPT4.0 in order to improve language and readability, with caution. After using this tool/service, the author(s) reviewed and edited the content as needed and take(s) full responsibility for the content of the publication.

## STAR★METHODS

Detailed methods are provided in the online version of this paper and include the following:

- KEY RESOURCES TABLE
- METHOD DETAILS
  - XOR task
  - Context-dependent decision-making task
  - Trajectory generation task
  - MNIST classification task
  - Atari game task
- QUANTIFICATION AND STATISTICAL ANALYSIS

## SUPPLEMENTAL INFORMATION

Supplemental information can be found online at https://doi.org/10.1016/j.isci.2025.112340.

Received: May 9, 2024
Revised: June 29, 2024
Accepted: March 31, 2025
Published: April 3, 2025

## REFERENCES

1. Izhikevich, E.M. (2004). Which model to use for cortical spiking neurons? IEEE Trans. Neural Network. *15*, 1063–1070.
2. Hazelden, J., Liu, Y.H., Shlizerman, E., and Shea-Brown, E. (2023). Evolutionary algorithms as an alternative to backpropagation for supervised training of biophysical neural networks and neural odes. Preprint at arXiv. https://doi.org/10.48550/arXiv.2311.10869.
3. Richards, B.A., Lillicrap, T.P., Beaudoin, P., Bengio, Y., Bogacz, R., Christensen, A., Clopath, C., Costa, R.P., de Berker, A., Ganguli, S., et al. (2019). A deep learning framework for neuroscience. Nat. Neurosci. *22*, 1761–1770.
4. Rumelhart, D.E., Hinton, G.E., and Williams, R.J. (1986). Learning representations by back-propagating errors. Nature *323*, 533–536.
5. Lillicrap, T.P., Cownden, D., Tweed, D.B., and Akerman, C.J. (2016). Random synaptic feedback weights support error backpropagation for deep learning. Nat. Commun. *7*, 13276.
6. Payeur, A., Guerguiev, J., Zenke, F., Richards, B.A., and Naud, R. (2021). Burst-dependent synaptic plasticity can coordinate learning in hierarchical circuits. Nat. Neurosci. *24*, 1010–1019.
7. Pascanu, R. (2013). On the difficulty of training recurrent neural networks. Preprint at arXiv. https://doi.org/10.48550/arXiv.1211.5063.
8. Lillicrap, T.P., Santoro, A., Marris, L., Akerman, C.J., and Hinton, G. (2020). Backpropagation and the brain. Nat. Rev. Neurosci. *21*, 335–346.
9. Silvanto, J. (2015). Why is “blindsight” blind? a new perspective on primary visual cortex, recurrent activity and visual awareness. Conscious. Cognit. *32*, 15–32.
10. Zeki, S. (2018). The rough seas of cortical cartography. Trends Neurosci. *41*, 242–244.
11. Petroni, F., Panzeri, S., Hilgetag, C.-C., Kötter, R., and Young, M.P. (2001). Simultaneity of responses in a hierarchical visual network. Neuroreport *12*, 2753–2759.
12. D’Souza, R.D., Wang, Q., Ji, W., Meier, A.M., Kennedy, H., Knoblauch, K., and Burkhalter, A. (2022). Hierarchical and nonhierarchical features of the mouse visual cortical network. Nat. Commun. *13*, 503.
13. Siegle, J.H., Jia, X., Durand, S., Gale, S., Bennett, C., Graddis, N., Heller, G., Ramirez, T.K., Choi, H., Luviano, J.A., et al. (2021). Survey of spiking in the mouse visual system reveals functional hierarchy. Nature *592*, 86–92.
14. Suzuki, M., Pennartz, C.M.A., and Aru, J. (2023). How deep is the brain? the shallow brain hypothesis. Nat. Rev. Neurosci. *24*, 778–791.
15. Stanley, K.O., Clune, J., Lehman, J., and Miikkulainen, R. (2019). Designing neural networks through neuroevolution. Nat. Mach. Intell. *1*, 24–35.
16. Such, F.P., Madhavan, V., Conti, E., Lehman, J., Stanley, K.O., and Clune, J. (2017). Deep neuroevolution: Genetic algorithms are a competitive alternative for training deep neural networks for reinforcement learning. Preprint at arXiv. https://doi.org/10.48550/arXiv.1712.06567.
17. Dayan, P., and Abbott, L.F. (2001). Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems (The MIT Press).
18. Chistiakova, M., Bannon, N.M., Bazhenov, M., and Volgushev, M. (2014). Heterosynaptic plasticity: Multiple mechanisms and multiple roles. Neuroscientist *263*, 532–536.
19. Chistiakova, M., and Volgushev, M. (2009). Heterosynaptic plasticity in the neocortex. Exp. Brain Res. *199*, 377–390.
20. Schuman, E.M., and Madison, D.V. (1994). Locally distributed synaptic potentiation in the hippocampus. Science *263*, 532–536.
21. Bonhoeffer, T., Staiger, V., and Aertsen, A. (1989). Synaptic plasticity in rat hippocampal slice cultures: local "Hebbian" conjunction of pre- and postsynaptic stimulation leads to distributed synaptic enhancement. Proc. Natl. Acad. Sci. USA *86*, 8113–8117.
22. Chen, J., Tan, Z., Zeng, L., Zhang, X., He, Y., Gao, W., Wu, X., Li, Y., Bu, B., Wang, W., and Duan, S. (2013). Heterosynaptic long-term depression mediated by ATP released from astrocytes. Glia *61*, 178–191.
23. Deitmer, J.W., Verkhratsky, A.J., and Lohr, C. (1998). Calcium signalling in glial cells. Cell Calcium *24*, 405–416.
24. Chasse, R., Malyshev, A., Fitch, R.H., and Volgushev, M. (2021). Altered heterosynaptic plasticity impairs visual discrimination learning in adenosine a1 receptor knock-out mice. J. Neurosci. *41*, 4631–4640.
25. Kepecs, A., and Fishell, G. (2014). Interneuron cell types are fit to function. Nature *505*, 318–326.
26. O’Connor, D.H., Wittenberg, G.M., and Wang, S.S.-H. (2005). Graded bidirectional synaptic plasticity is composed of switch-like unitary events. Proc. Natl. Acad. Sci. USA *102*, 9679–9684.
27. Roelfsema, P.R., and Holtmaat, A. (2018). Control of synaptic plasticity in deep cortical networks. Nat. Rev. Neurosci. *19*, 166–180.
28. Duszkiewicz, A.J., McNamara, C.G., Takeuchi, T., and Genzel, L. (2019). Novelty and dopaminergic modulation of memory persistence: A tale of two systems. Trends Neurosci. *42*, 102–114.
29. Hulme, S.R., Jones, O.D., Raymond, C.R., Sah, P., and Abraham, W.C. (2014). Mechanisms of heterosynaptic metaplasticity. Phil. Trans. R. Soc. B *369*, 20130148.

30. Ji, D., and Wilson, M.A. (2007). Coordinated memory replay in the visual cortex and hippocampus during sleep. Nat. Neurosci. *10*, 100–107.

31. Harvey, C.D., and Svoboda, K. (2007). Locally dynamic synaptic learning rules in pyramidal neuron dendrites. Nature *450*, 1195–1200.

32. Hayama, T., Noguchi, J., Watanabe, S., Takahashi, N., Hayashi-Takagi, A., Ellis-Davies, G.C.R., Matsuzaki, M., and Kasai, H. (2013). GABA promotes the competitive selection of dendritic spines by controlling local Ca2+ signaling. Nat. Neurosci. *16*, 1409–1416.

33. Engert, F., and Bonhoeffer, T. (1997). Synapse specificity of long-term potentiation breaks down at short distances. Nature *388*, 279–284.

34. Mante, V., Sussillo, D., Shenoy, K.V., and Newsome, W.T. (2013). Context-dependent computation by recurrent dynamics in prefrontal cortex. Nature *503*, 78–84.

35. Hiratani, N., and Fukai, T. (2018). Redundancy in synaptic connections enables neurons to learn optimally. Proc. Natl. Acad. Sci. USA *115*, E6871–E6879.

36. Oldham, S., and Fornito, A. (2019). The development of brain network hubs. Dev. Cogn. Neurosci. *36*, 100607.

37. Wall, N.R., De La Parra, M., Sorokin, J.M., Taniguchi, H., Huang, Z.J., and Callaway, E.M. (2016). Brain-wide maps of synaptic input to cortical interneurons. J. Neurosci. *36*, 4000–4009.

38. Pfeffer, C.K., Xue, M., He, M., Huang, Z.J., and Scanziani, M. (2013). Inhibition of inhibition in visual cortex: the logic of connections between molecularly distinct interneurons. Nat. Neurosci. *16*, 1068–1076.

39. Stott, J.J., and Redish, A.D. (2015). Representations of value in the brain: An embarrassment of riches? PLoS Biol. *13*, e1002174.

40. Fischer, A.G., and Ullsperger, M. (2017). An update on the role of serotonin and its interplay with dopamine for reward. Front. Hum. Neurosci. *11*, 484.

41. Mongillo, G., Rumpel, S., and Loewenstein, Y. (2017). Intrinsic volatility of synaptic connections - a challenge to the synaptic trace theory of memory. Curr. Opin. Neurobiol. *46*, 7–13.

42. Faisal, A.A., Selen, L.P.J., and Wolpert, D.M. (2008). Noise in the nervous system. Nat. Rev. Neurosci. *9*, 292–303.

43. Gerstner, W., Lehmann, M., Liakoni, V., Corneil, D., and Brea, J. (2018). Eligibility traces and plasticity on behavioral time scales: Experimental support of neohebbian three-factor learning rules. Front. Neural Circ. *12*, 53.

44. Schuman, E.M., and Madison, D.V. (1991). A requirement for the intercellular messenger nitric oxide in long-term potentiation. Science *254*, 1503–1506.

45. Kossel, A., Bonhoeffer, T., and Bolz, J. (1990). Non-hebbian synapses in rat visual cortex. Neuroreport *1*, 115–118.

46. Cornell-Bell, A.H., Finkbeiner, S.M., Cooper, M.S., and Smith, S.J. (1990). Glutamate induces calcium waves in cultured astrocytes: Long-range glial signaling. Science *247*, 470–473.

47. Scanziani, M., Malenka, R.C., and Nicoll, R.A. (1996). Role of intercellular interactions in heterosynaptic long-term depression. Nature *380*, 446–450.

48. Min, R., and Nevian, T. (2012). Astrocyte signaling controls spike timing-dependent depression at neocortical synapses. Nat. Neurosci. *15*, 746–753.

49. Andersson, M., Blomstrand, F., and Hanse, E. (2007). Astrocytes play a critical role in transient heterosynaptic depression in the rat hippocampal CA1 region. J. Physiol. *585*, 843–852.

50. Lee, C.M., Stoelzel, C., Chistiakova, M., and Volgushev, M. (2012). Heterosynaptic plasticity induced by intracellular tetanization in layer 2/3 pyramidal neurons in rat auditory cortex. J. Physiol. *590*, 2253–2271.

51. Tsumoto, T., and Suda, K. (1979). Cross-depression: an electrophysiological manifestation of binocular competition in the developing visual cortex. Brain Res. *168*, 190–194.

52. Mirjalili, S. (2019). Evolutionary Algorithms and Neural Networks: Theory and Applications (Springer).

53. Gao, S., Zhou, M., Wang, Y., Cheng, J., Yachi, H., and Wang, J. (2019). Dendritic neuron model with effective learning algorithms for classification, approximation, and prediction. IEEE Transact. Neural Networks Learn. Syst. *30*, 601–614.

54. Holtmaat, A., and Caroni, P. (2016). Functional and structural underpinnings of neuronal assembly formation in learning. Nat. Neurosci. *19*, 1553–1562.

55. Josselyn, S.A., and Tonegawa, S. (2020). Memory engrams: Recalling the past and imagining the future. Science *367*, eaaw4325.

56. Abraham, W.C., Mason-Parker, S.E., Bear, M.F., Webb, S., and Tate, W.P. (2001). Heterosynaptic metaplasticity in the hippocampus in vivo: a BCM-like modifiable threshold for LTP. Proc. Natl. Acad. Sci. USA *98*, 10924–10929.

57. Hulme, S.R., Jones, O.D., Ireland, D.R., and Abraham, W.C. (2012). Calcium-dependent but action potential independent BCM-like metaplasticity in the hippocampus. J. Neurosci. *32*, 6785–6794.

58. Sutton, M.A., Ito, H.T., Cressy, P., Kempf, C., Woo, J.C., and Schuman, E.M. (2006). Miniature neurotransmission stabilizes synaptic function via tonic suppression of local dendritic protein synthesis. Cell *125*, 785–799.

59. Barnes, S.J., Franzoni, E., Jacobsen, R.I., Erdelyi, F., Szabo, G., Clopath, C., Keller, G.B., and Keck, T. (2017). Deprivation-induced homeostatic spine scaling in vivo is localized to dendritic branches that have undergone recent spine loss. Neuron *96*, 871–882.e5.

60. Wolpert, D.h., and Macready, W.g. (1997). No free lunch theorems for optimization. IEEE Trans. Evol. Comput. *1*, 67–82.

61. Schultz, W., Dayan, P., and Montague, P.R. (1997). A neural substrate of prediction and reward. Science *275*, 1593–1599.

62. Larkum, M.E., and Nevian, T. (2008). Synaptic clustering by dendritic signalling mechanisms. Curr. Opin. Neurobiol. *18*, 321–331.

63. Takahashi, N., Kitamura, K., Matsuo, N., Mayford, M., Kano, M., Matsuki, N., and Ikegaya, Y. (2012). Locally synchronized synaptic inputs. Science *335*, 353–356.

64. Nicola, W., and Clopath, C. (2017). Supervised learning in spiking neural networks with FORCE training. Nat. Commun. *8*, 2208.

65. DePasquale, B., Churchland, M., and Abbott, L.F. (2016). Using firing-rate dynamics to train recurrent networks of spiking model neurons. Preprint at arXiv. https://doi.org/10.48550/arXiv.1601.07620.

66. Yann, L. (2010). Mnist Handwritten Digit Database (ATT Labs).

67. Amato, G., Carrara, F., Falchi, F., Gennaro, C., and Lagani, G. (2019). Hebbian learning meets deep convolutional neural networks. In International Conference on Image Analysis and Processing, pp. 324–334.

68. Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A.A., Veness, J., Bellemare, M.G., Graves, A., Riedmiller, M., Fidjeland, A.K., Ostrovski, G., et al. (2015). Human-level control through deep reinforcement learning. Nature *518*, 529–533.

69. Mnih, V., Badia, A.P., Mirza, M., Graves, A., Harley, T., Lillicrap, T.P., Silver, D., and Kavukcuoglu, K. (2016). Asynchronous methods for deep reinforcement learning. Preprint at arXiv. https://doi.org/10.48550/arXiv.1602.01783.

70. Krotov, D., and Hopfield, J.J. (2019). Unsupervised learning by competing hidden units. Proc. Natl. Acad. Sci. USA *116*, 7723–7731.

71. Zhou, Z.-H., Yu, Y., and Qian, C. (2019). Evolutionary Learning: Advances in Theories and Algorithms (Springer).

72. Hospedales, T., Antoniou, A., Micaelli, P., and Storkey, A. (2020). Meta-learning in neural networks: a survey. Preprint at arXiv. https://doi.org/10.48550/arXiv.2004.05439.

73. Soysal, O.A., and Guzel, M.S. (2020). An introduction to zero-shot learning: An essential review. In 2020 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA).

74. Zhuang, F., Qi, Z., Duan, K., Xi, D., Zhu, Y., Zhu, H., Xiong, H., and He, Q. (2020). A comprehensive survey on transfer learning. Preprint at arXiv. https://arxiv.org/pdf/1911.02685.

75. Parisi, G.I., Kemker, R., Part, J.L., Kanan, C., and Wermter, S. (2019). Continual lifelong learning with neural networks: A review. Preprint at arXiv. https://doi.org/10.48550/arXiv.1802.07569.

76. Akrout, M., Wilson, C., Humphreys, P., Lillicrap, T., and Tweed, D.B. (2019). Deep learning without weight transport. Adv. Neural Inf. Process. Syst. *32*.

77. Srinivasan, R., Mignacco, F., Sorbaro, M., Refinetti, M., Cooper, A., Kreiman, G., and Dellaferrera, G. (2023). Forward learning with top-down feedback: Empirical and analytical characterization. Preprint at arXiv. https://doi.org/10.48550/arXiv.2302.05440.

78. Hinton, G. (2022). The forward-forward algorithm: Some preliminary investigations. Preprint at arXiv. https://doi.org/10.48550/arXiv.2212.13345.

79. Elman, J.L. (1993). Learning and development in neural networks: The importance of starting small. Cognition *48*, 71–99.

80. Sukhbaatar, S., Lin, Z., Kostrikov, I., Synnaeve, G., Szlam, A., and Fergus, R. (2017). Intrinsic motivation and automatic curricula via asymmetric self-play. Preprint at arXiv. https://doi.org/10.48550/arXiv.1703.05407.

81. Gruber, M.J., Ritchey, M., Wang, S.-F., Doss, M.K., and Ranganath, C. (2016). Post-learning hippocampal dynamics promote preferential retention of rewarding events. Neuron *89*, 1110–1120.

82. Qian, C., Bian, C., and Feng, C. (2020). Subset selection by pareto optimization with recombination. In Proceedings of the AAAI Conference on Artificial Intelligence.

83. Isaacson, J.S., and Scanziani, M. (2011). How inhibition shapes cortical activity. Neuron *72*, 231–243.

84. Reynolds, J.N.J., and Wickens, J.R. (2002). Dopamine-dependent plasticity of corticostriatal synapses. Neural Netw. *15*, 507–521.

85. Turrigiano, G. (2012). Homeostatic synaptic plasticity: local and global mechanisms for stabilizing neuronal function. Cold Spring Harbor Perspect. Biol. *4*, a005736.

86. Strata, P., and Harvey, R. (1999). Dale's principle. Brain Res. Bull. *50*, 349–350.

87. Miconi, T. (2017). Biologically plausible learning in recurrent neural networks reproduces neural dynamics observed during cognitive tasks. Elife *6*, e20899.

88. Zielinski, M.C., Tang, W., and Jadhav, S.P. (2020). The role of replay and theta sequences in mediating hippocampal-prefrontal interactions for memory and cognition. Hippocampus *30*, 60–72.

89. Diekelmann, S., and Born, J. (2010). The memory function of sleep. Nat. Rev. Neurosci. *11*, 114–126.

90. Ólafsdóttir, H.F., Bush, D., and Barry, C. (2018). The role of hippocampal replay in memory and planning. Curr. Biol. *28*, R37–R50.

91. Brodt, S., Inostroza, M., Niethard, N., and Born, J. (2023). Sleep-a brain-state serving systems memory consolidation. Neuron *111*, 1050–1075.

92. Rasch, B., and Born, J. (2013). About sleep's role in memory. Physiol. Rev. *93*, 681–766.

93. Hobson, J.A., and Pace-Schott, E.F. (2002). The cognitive neuroscience of sleep: neuronal systems, consciousness and learning. Nat. Rev. Neurosci. *3*, 679–693.

94. Siapas, A.G., and Wilson, M.A. (1998). Coordinated interactions between hippocampal ripples and cortical spindles during slow-wave sleep. Neuron *21*, 1123–1128.

95. Pouget, A., Dayan, P., and Zemel, R. (2000). Information processing with population codes. Nat. Rev. Neurosci. *1*, 125–132.

96. Averbeck, B.B., Latham, P.E., and Pouget, A. (2006). Neural correlations, population coding and computation. Nat. Rev. Neurosci. *7*, 358–366.

97. Singer, W. (1999). Neuronal synchrony: a versatile code for the definition of relations? Neuron *24*, 49–125.

98. Gray, C.M., König, P., Engel, A.K., and Singer, W. (1989). Oscillatory responses in cat visual cortex exhibit inter-columnar synchronization which reflects global stimulus properties. Nature *338*, 334–337.

99. Buzsaki, G., and Draguhn, A. (2004). Neuronal oscillations in cortical networks. Science *304*, 1926–1929.

100. Headley, D.B., Latimer, B., Aberbach, A., and Nair, S.S. (2024). Spatially targeted inhibitory rhythms differentially affect neuronal integration. Elife *13*, RP95562.

101. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al. (2011). Scikit-learn: Machine learning in python. J. Mach. Learn. Res. *12*, 2825–2830.

102. Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., et al. (2016). TensorFlow : a system for Large-Scale machine learning. In 12th USENIX symposium on operating systems design and implementation (OSDI 16), pp. 265–283.

103. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. (2019). Pytorch: An imperative style, high-performance deep learning library. Adv. Neural Inf. Process. Syst. *32*.

104. Harris, C.R., Millman, K.J., Van Der Walt, S.J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N.J., et al. (2020). Array programming with numpy. Nature *585*, 357–362.

105. Stimberg, M., Brette, R., and Goodman, D.F. (2019). Brian 2, an intuitive and efficient neural simulator. Elife *8*, e47314.

106. Seung, H.S. (2003). Learning in spiking neural networks by reinforcement of stochastic synaptic transmission. Neuron *40*, 1063–1073.

107. Kim, C.M., and Chow, C.C. (2018). Learning recurrent dynamics in spiking networks. Elife *7*, e37124.

108. Anand, A., Racah, E., Ozair, S., Bengio, Y., Côté, M.-A., and Hjelm, R.D. (2019). Unsupervised state representation learning in Atari. In Conference on Neural Information Processing Systems.

# STAR★METHODS

## KEY RESOURCES TABLE

| REAGENT or RESOURCE | SOURCE | IDENTIFIER |
|---|---|---|
| Deposited data | | |
| Database: MNIST | Yann et al.[66] | http://yann.lecun.com/exdb/mnist/ |
| Software and algorithms | | |
| Python version >3.7 | Python Software Foundation | https://www.python.org |
| Sklearn | Pedregosa et al.[101] | https://scikit-learn.org |
| Tensorflow-gpu >1.2 | Abadi et al.[102] | https://github.com/tensorflow |
| Pytorch | Paszke et al.[103] | https://github.com/pytorch/pytorch |
| Numpy | Harris et al.[104] | https://github.com/numpy/numpy |
| Brian 2 | Stimberg et al.[105] | https://github.com/brian-team/brian2 |
| Scripts for analysis | This paper | https://drive.google.com/drive/folders/1rpoqedy9YW0tffkUC6CAIMdQCo0aoXaQ?dmr=1&ec=wgc-drive-hero-goto |

## METHOD DETAILS

### XOR task

In the network of Figure 5C, 20 pyramidal neurons receive from two groups of excitatory neurons (each group has 20 neurons) through 20 dendrites gated by different SST interneurons. These pyramidal neurons are inhibited by PV interneurons, which receive input from both groups of excitatory input neurons. Pyramidal neurons also give output to a single neuron. Pyramidal neurons, PV interneurons and the output neuron are modeled with leaky integrate-and-fire equations:

$$C\frac{dV_i}{dt} = -g_L(V_i - V_L) - \sum_j g_{ij}(V_i - V_{ij}) + I_{background}, \qquad \text{(Equation 1)}$$

where $V_L = -74\text{mV}$, $g_L = 25\text{nS}$, $C = 500\text{pF}$, and $I_{background} = 425\text{pA}$ (parameters are chosen according to[106]). When the membrane potential $V_i$ reaches the firing threshold $V_\theta = 54\text{mV}$, a spike is recorded and $V_i$ is reset to $V_{reset} = -60\text{mV}$ . The reversal potential $V_{ij}$ is 0mV if the synapse from $j$ to $i$ is excitatory, or $-70$ mV if this synapse is inhibitory. The dynamics of the synaptic conductance $g_{ij}$ is

$$\tau_{ij}\frac{dg_{ij}}{dt} = -g_{ij} + w_{ij}\sum_l \delta(t - t_{l;j}), \qquad \text{(Equation 2)}$$

with $t_{l;j}$ being the time of the $l$th spike of the $j$th neuron. The time constant $\tau_{ij}$ is 10ms for the synapses from the PV interneurons to the pyramidal neurons and 5ms for the other synapses. The synaptic efficacy $w_{ij}$ is 4.8nS from the excitatory input neurons to the PV interneurons, 2.4nS from the inhibitory input neurons to the PV interneurons, 2.4nS from the PV interneurons to the pyramidal neurons, and 4.8nS from the pyramidal neurons to the single output neuron. All the above synapses are connected with probability 0.4, and fixed during training. The synapses from the excitatory input neurons to the pyramidal neurons are all-to-all connected in every dendrite of the pyramidal neurons, and have either large efficacy of 2.4nS or small efficacy of 0nS. The excitatory input neurons produce Poisson spike trains. The firing rates of the two groups of excitatory input neurons are respectively (0Hz, 0Hz), (20Hz, 0Hz), (0Hz, 20Hz), and (20Hz, 20Hz) in the four successive intervals with duration 500ms in each training epoch (Figure 5D, upper panel). The pyramidal neurons also receive from a group of inhibitory neurons (not depicted in Figure 5C), which output Poisson spike trains of 40Hz. Our simulation was performed in the Brian simulator[105] using the exponential Euler method with time step of 0.1ms.

The potentiation eligibility trace of the synapse from a pre-synaptic excitatory input neuron to a post-synaptic pyramidal neuron through the $d$th dendrite is

$$e^{pot}_{pre,post,d} = A(d)\sum_{l,m}\Theta(t_{m;post} - t_{l;pre})e^{-\left(t_{m;post} - t_{l;pre}\right)/\tau_{pot}}, \qquad \text{(Equation 3)}$$

where $\Theta(\cdot)$ is the step function, $t_{m;post}$ (or $t_{l;pre}$) is the time of the $m$th (or $l$th) spike of the post- (or pre-) synaptic neuron, $\tau_{pos} = 20\text{ms}$ is the characteristic time window indicating the simultaneity of the pre- and post-synaptic spikes that induces potentiation eligibility trace, $A(d) = 1$ for a synapse on the gated-on dendrite, and $A(d) = 0.5$ for a synapse on a gated-off dendrite modeling the effect of diffusive NO emitted from the gated-on dendrite. Here we suppose that the potentiation effect of the NO emitted from a gated-on dendrite of a pyramidal neuron can only influence the dendrites of the same pyramidal neuron, and has no influence on the dendrites of other pyramidal neurons, based on the experimental observation that the spatial scope of this potentiation effect is comparable with the dendritic arbor of a pyramidal neuron.[20,21]

Synaptic efficacy was updated according to the eligibility trace and the dopamine level, satisfying the following rules or constraints, explained below:

$$\begin{cases} w_{ij,d}(T) = 0 \text{ or } w_{max} \\ w_{ij,d}(T) = w_{ij,d}(T-1) + \Theta\left(e^{pot}_{ij,d}(T-1) - e^{pot}_{\theta}\right) B\left(r(T), \{r(t)\}_{t<T}\right) \\ \sum_j w_{ij,d}(T) = \sum_j w_{ij,d}(T-1) \\ w_{ij,d}(T) = w_{ij,d}(T-1), \text{if } h_{ij,d}(T-1) > 0 \text{ and } w_{ij,d}(T-1) = w_{max} \\ h_{ij,d}(T) = h_{ij,d}(T-1) + \Theta\left(e^{pot}_{ij,d}(T-1) - e^{pot}_{\theta}\right) C\left(r(T), \{r(t)\}_{t<T}\right), \text{if } w_{ij,d}(T) = w_{max} \\ h_{ij,d}(T) = h_{ij,d}(T-1) - 1 \\ 0 \le h_{ij,d}(T) \le h_{max} \end{cases} \quad \text{(Equation 4)}$$

The first equation indicates that the synapse $w_{ij,d}(T)$ from the $j$th excitatory input neuron to the $i$th pyramidal neuron through the $d$th dendrite at the $T$th epoch has binary efficacies 0 or $w_{max} = 2.4\text{nS}$. The second equation indicates that $w_{ij,d}$ is potentiated if $e^{pot}_{ij,d}$ is larger than a threshold value $e^{pot}_{\theta} = 0.45$ and the factor $B$ controlled by the reward $r(T)$ at the $T$th epoch is larger than zero. In our simulation, $B = 1$ if $r(T)$ is not smaller than the second largest reward obtained in the last 20 training epochs, and $B = 0$ otherwise. The reward at an epoch is defined as

$$r = -[o(\text{high}, \text{high})]_+ - [-o(\text{high}, \text{low}) + 10]_+ - [-o(\text{low}, \text{high}) + 10]_+, \quad \text{(Equation 5)}$$

where $o(\text{high}, \text{low})$ means the number of spikes of the output neuron during the interval of 500ms when Group 1 of the excitatory input neurons have high firing rate and Group 2 have low firing rate; $o(\text{high}, \text{high})$ and $o(\text{low}, \text{high})$ have similar definitions; $o(\text{low}, \text{low}) = 0$ all the time. The third equation in Equation 4 indicates dendrite-level synaptic homeostasis (Figure 3C, lower panel), which keeps constant the total synaptic efficacy on a dendrite (specifically, half of the synapses on a dendrite are large). The fourth equation indicates that if the hidden state $h_{ij,d}$ in a synapse with large efficacy is larger than zero, this synapse will remain unchanged in the current epoch. In other words, a large synapse can only be depressed when its hidden state is zero. Note that the hidden state remains at zero for synapses with low efficacy. The fifth equation indicates that if the eligibility trace $e^{pot}_{ij,d}$ is larger than $e^{pot}_{\theta}$ and the factor $C$ controlled by the reward is larger than zero, the hidden state in the large synapse will be increased. In our simulation, $C$ is 20 if $r(T)$ is not smaller than the largest reward obtained in the last 20 training epochs, $C$ is 10 if $r(T)$ is smaller than the largest but not smaller than the second largest reward obtained in the last 20 training epochs, and $C = 0$ otherwise. The sixth equation indicates that the hidden state in every large synapse is decreased by 1 in each training epoch, so that the synapse gradually becomes less stable. The seventh equation indicates that the hidden state is non-negative and no larger than a maximum value $h_{max} = 20$. Here in the sixth and seventh equations, for each synapse, they have hidden state $h$, without high dopamine level, the hidden state in every large synapse is decreased by 1 in each training epoch. The hidden state has maximum value $h_{max}$. When the hidden state of synapse reaches $h_{max}$, the synapse will stay at a large efficacy $w_{max}$ for a long time, which indicates elitism (see Figure 3E).

The second and third equations in Equation 4 may conflict with each other. In our simulation, when too many synapses had large potentiation eligibility traces and were to be potentiated, we kept synaptic homeostasis (i.e., the third equation) non-violated, and set the synapses with largest potentiation eligibility traces at large efficacy. Specifically, at the $T$th epoch when $B > 0$, we collected the synapses either with $w_{ij,d}(T-1) > 0$ and $h_{ij,d}(T-1) = 0$ or with $w_{ij,d}(T-1) = 0$ and $e^{pot}_{ij,d}(T-1) > e^{pot}_{\theta}$ into a set $\mathcal{S}$, sorted the potentiation eligibility traces in the synapses in $\mathcal{S}$ in descending order, and let the first $N_{\text{input}}/2 - N_{\text{stable large}}$ synapses have large efficacy, and let the rest synapses have small efficacy. Here, $N_{\text{input}} = 40$ is the number of synapses on a single dendrite (which is the total number of excitatory neurons in the two input groups), and we constrained the number of synapses with large efficacy on a dendrite fixed at $N_{\text{input}}/2$, modeling dendrite-level synaptic homeostasis (the third equation of Equation 4). $N_{\text{stable large}}$ is the number of synapses with $w_{ij,d}(T-1) > 0$ and $h_{ij,d}(T-1) > 0$ on the $d$th dendrite.

In our model, each pyramidal neuron has 20 dendrites, labeled from 1 to 20. Every neuron in the two input populations gives output to every dendrite (Figure 5C). At each training epoch, the dendrites of all the pyramidal neurons with the same randomly selected label were gated-on, with the other dendrites gated off (Figure 1E). Synaptic mutation was performed at the gated-on dendrites at the beginning of each epoch. In individual mutation, synapses with large efficacy and zero hidden state were randomly set to small efficacy with probability 0.25 at the gated-on dendrites; at the same time, the same number of randomly selected synapses with small efficacy were also set to large efficacy, to maintain the total number of large synapses on every gated-on dendrite. In group mutation, all the synapses (except for those synapses with non-zero hidden states) from a randomly selected input group to the gated-on dendrite of a randomly selected pyramidal neuron were set to small efficacy; at the same time, the same number of synapses on the gated-on dendrite from another input group to the selected pyramidal neuron were set to large efficacy to maintain the total number of large synapses on that gated-on dendrite. In both mutation strategies, a single large synapse with zero hidden state has almost equal probability 0.025 to be mutated (Figure S1C), so that the performance difference under these two strategies (Figures 5E, S1B, and S1C) is caused by these strategies themselves, instead of the difference of mutation rate.

## Context-dependent decision-making task

Our model to study the context-dependent decision-making task in Figure 6 closely follows that of.[34] The recurrent neural network contains $N = 100$ neurons, and is all-to-all connected with the following dynamics:

$$\tau\dot{\boldsymbol{x}} = -\boldsymbol{x} + \boldsymbol{J}\boldsymbol{r} + \boldsymbol{b}^c u_c + \boldsymbol{b}^m u_m + \boldsymbol{b}^{cc} u_{cc} + \boldsymbol{b}^{cm} u_{cm} + \boldsymbol{c}^x + \rho_x \qquad \text{(Equation 6)}$$

$$\boldsymbol{r} = \tanh(\boldsymbol{x})$$

$$z = \boldsymbol{w}_z^T \boldsymbol{r} + c^z$$

The variable $\boldsymbol{x}(t)$ is an $N$-dimensional vector containing the total synaptic current of each neuron in the network, and $\boldsymbol{r}(t)$ are the corresponding firing rates. Each neuron has a time constant $\tau = 10$ ms. The matrix $\boldsymbol{J}$ defines the recurrent connections in the network. The network receives 4-dimensional input, $\boldsymbol{u}(t) = [u_c(t), u_m(t), u_{cc}(t), u_{cm}(t)]^T$, through synaptic weights, $\boldsymbol{B} = [\boldsymbol{b}^c, \boldsymbol{b}^m, \boldsymbol{b}^{cc}, \boldsymbol{b}^{cm}]$. These four inputs represent, respectively, the sensory evidence for color and motion, and the contextual cues instructing the network to integrate either the color or the motion input. Finally, $\boldsymbol{c}^x$ is a vector of offset currents and $\rho_x$ is a white noise drawn at each time step with standard deviation 0.1. The output $z$ of the network is a weighted sum of the firing rates, with weights $\boldsymbol{w}_z^T$ and bias $c^z$. All the synaptic connections in $\boldsymbol{J}$, $\boldsymbol{B}$ and $\boldsymbol{w}_z$ are binary, taking values of $\pm 1/\sqrt{N}$, $\pm 1$ and $\pm 1/\sqrt{N}$, respectively. During training, the network dynamics were integrated for the duration $T = 750$ ms using Euler updates with $\Delta t = 1$ ms. After training, model dynamics were integrated for an additional 200 ms with the sensory inputs turned off, according to which we plotted the orange dots and lines in Figures 6C and 6D.

The contextual inputs $u_{cm}$ and $u_{cc}$ were constant for the duration of the trial. In the motion context $u_{cm}(t) = 1$ and $u_{cc}(t) = 0$, while in the color context $u_{cm}(t) = 0$ and $u_{cc}(t) = 1$. The motion and color inputs $u_m$ and $u_c$ are one-dimensional white-noise signals:

$$u_m(t) = d_m + \rho_m(t)$$

$$u_c(t) = d_c + \rho_c(t)$$

The white noise terms $\rho_m$ and $\rho_c$ have zero mean and standard deviation 1. During training, the offsets $d_m$ and $d_c$ were randomly chosen on each trial from the range $[-0.1875, 0.1875]$. During simulations after training, $d_m$ and $d_c$ took 6 values $(\pm 0.009, \pm 0.036, \pm 0.15)$, corresponding to weak, intermediate, and strong evidence toward either choice. In the psychometric curves (Figure 6B), the coherence value is normalized so that $\pm 0.15$ of $d_m$ or $d_c$ corresponds to $\pm 1$ of the horizontal coordinate in the figure.

The target $p$ of network training is that the output $z$ approaches 1 (or $-1$) at time $T$ when $d_m > 0$ (or $\langle 0$) in the motion context, and approaches 1 (or $-1$) when $d_c > 0$ (or $\langle 0$) in the color context. In the EA, the number of agents was 100, the number of elite agents in each generation was 5. These agents were evaluated by averaging $(p - z(T))^2$ over 128 simulation trials. At each training epoch, every non-parental agent copied the synaptic weights $\boldsymbol{J}$, $\boldsymbol{B}$ and $\boldsymbol{w}_z$ as well as the bias $\boldsymbol{c}^x$ and $c^z$ from a randomly chosen elite agent, with the following mutation: every synaptic weight was flipped with probability 0.0005; to every bias was added a random Gaussian number with zero mean and standard deviation 0.002. The mutation of bias models the mutation of synaptic weights inputted from other brain areas.

The axes of choice, motion and color in Figures 6C and 6D were obtained by first linear regressing the trajectory $\boldsymbol{x}(t)$ of all simulation trials and then orthogonalizing the regression coefficient using QR-decomposition.[34] Similar to,[34] we plotted $\boldsymbol{x}(t) - \langle \boldsymbol{x}(t) \rangle_{\text{trial}}$ in Figures 6C and 6D, with the urgency signal $\langle \boldsymbol{x}(t) \rangle_{\text{trial}}$ being the average trajectory over all simulation trials in the motion (Figure 6C) or color (Figure 6D) context.

The attractors in Figures 6C–6E were obtained by minimizing the function

$$q(\boldsymbol{x}) = \frac{1}{2} |\boldsymbol{F}(\boldsymbol{x})|^2, \qquad \text{(Equation 7)}$$

where $\boldsymbol{F}(\boldsymbol{x})$ is the right-hand side of Equation 6 after setting $u_c = u_m = \rho_x = 0$, representing the speed of changing of neural state. To get the attractors in the motion context, $u_{cm}(t) = 1$ and $u_{cc}(t) = 0$; to get the attractors in the color context, $u_{cm}(t) = 0$ and $u_{cc}(t) = 1$. We used the L-BFGS-B algorithm of the 'minimize' routine of the Scipy package to minimize $q(\boldsymbol{x})$, initializing $\boldsymbol{x}$ to be a random point in the trajectory of $\boldsymbol{x}(t)$. To find slow-dynamic points on the line attractor instead of the two stable fixed points, we set the tolerance parameter 'ftol = 0.002' to early stop the minimization algorithm. The found slow-dynamic points have a close-to-zero negative eigenvalue and many large negative eigenvalues. The small bars in Figure 6E are the left eigenvectors of the close-to-zero eigenvalues of these points (i.e., selection vectors), projected in the subspace spanned by the input weights $\boldsymbol{b}^c$ and $\boldsymbol{b}^m$.

### Trajectory generation task

In the trajectory generation task (Figures 7A and 7B), we considered a network of $N = 1000$ randomly and sparsely connected (connection probability $p = 0.1$) quadratic integrate-and-fire neurons with dynamics[107]

$$\tau \frac{dv_i}{dt} = I_{in}(t) + \sum_k \mathbb{I}(k) \sum_j s_{ij,k}(t) + v_i^2, \tag{Equation 8}$$

where $\tau = 10\text{ms}$; $v_i$ is a dimensionless variable representing the membrane potential; $s_{ij,k}$ is the synaptic current from neuron $j$ to neuron $i$ through dendrite $k$; and $\mathbb{I}(k) = 1$ or 0, indicating whether or not dendrite $k$ is gated on. The dynamics of $s_{ij,k}$ is

$$\tau_s \frac{ds_{ij,k}}{dt} = -s_{ij,k} + w_{ij,k} \sum_l \delta(t - t_{l;j}), \tag{Equation 9}$$

where $\tau_s = 20\text{ms}$, $w_{ij,k}$ is the synaptic weight from neuron $j$ to neuron $i$ through dendrite $k$, and $t_{l;j}$ is the time of the $l$th spike of neuron $j$. To simulate the dynamics of quadratic integrate-and-fire neurons, we used theta neuron model derived by a simple change of variables $v_i = \tan(\theta_i / 2)$, getting

$$\tau \frac{d\theta_i}{dt} = 1 - \cos\theta_i + \left( I_{in}(t) + \sum_j \mathbb{I}(k) s_{ij,k}(t) \right) (1 + \cos\theta_i). \tag{Equation 10}$$

The output of the network is

$$o(t) = \sum_k \mathbb{I}(k) \sum_j w_{oj,k} s_{oj,k}(t) \tag{Equation 11}$$

where $w_{oj,k}$ is the output weight of neuron $j$ through dendrite $k$, $s_{oj,k}$ is the synaptic current with the dynamics

$$\tau_s \frac{ds_{oj,k}}{dt} = -s_{oj,k} + w_{oj,k} \sum_l \delta(t - t_{l;j}). \tag{Equation 12}$$

The recurrent synapse $w_{ij,k}$ takes binary values: either $20/\sqrt{pN}$ or $-20/\sqrt{pN}$. The output synapse $w_{oj,k}$ also takes binary values: either either $1/\sqrt{N}$ or $-1/\sqrt{N}$.

The target trajectory (orange curve in Figure 7B) was defined as $f(t) = A\sin(2\pi(t - T_0)/T_1)\sin(2\pi(t - T_0)/T_2)$, where $A$, $T_0$, $T_1$ and $T_2$ were randomly sampled from intervals $[0.5, 1.5]$, $[0, 1000\text{ ms}]$, $[500\text{ ms}, 1000\text{ ms}]$ and $[100\text{ ms}, 500\text{ ms}]$ respectively. At the beginning of each simulation session, every neuron was stimulated for 50 ms with constant external stimulus that had random amplitude sampled from $[-1, 1]$ (red shading in Figure 7B). The pattern of the stimulation was the same for the same target function. Simulations were performed in the Brian simulator[105] using the Euler method with time step of 0.1ms.

Our EA was performed with the aim to reduce the loss function

$$\text{loss} = \sum_{t > 50\text{ms}} (f(t) - o(t))^2. \tag{Equation 13}$$

In the EA, the number of agent was 100, the number of elite agents in each generation was 5. At each training step, every non-parental agent copied the synaptic weights of a randomly chosen elite agent, then every synaptic weight was flipped with probability 0.0005.

### MNIST classification task

In Figures 7C and 7D, we train a multi-layer perceptron with 2 hidden layers with 256 and 128 neurons respectively to classify the MNIST images (size 28×28 = 784 pixels) into 10 classes. Adjacent layers in the feedforward network are all-to-all connected. The non-linear activation function is ReLU.

The EA was performed on networks whose synaptic weights took binary values $\{-\sqrt{1/N_{\text{in}}}, \sqrt{1/N_{\text{in}}}\}$, where $N_{\text{in}}$ represents input size: 784 for weights between the input layer and the first hidden layer; 256 for weights between the first and second hidden layers; and 128 between the second hidden layer and the output layer. We initialized 100 network configurations. At every training epoch, negative log likelihood loss averaged over all the 60000 images in the training dataset was evaluated for every configuration. The elite configuration (the one with the smallest loss) remained unchanged, whereas each weight of the other configurations was updated to be the corresponding weight of the elite configuration with probability 0.99995, and to be different with probability 0.00005 (mutation). Therefore, there is only one parent in each generation of agents in this algorithm. We also studied the case of dendrite-level synaptic homeostasis (Figure 3C, lower panel), in which the number of synapses with large efficacy received by a neuron in a configuration was kept fixed after the random initialization of synaptic weights. Adding this homeostasis mechanism hardly influences the classification performance (Figure S5B).

Back-propagation was performed on networks whose synaptic weights took continuous values. We used Adam optimizer of Pytorch for the training. Batch size was 100, learning rate was 0.0001, the number of training epochs was 400.

The Hebbian learning was also performed on continuous-weight networks. All the low-level layers (except the top layer) were trained by competitive Hebbian learning.[70] In every step, the weights were updated by

$$\Delta w_{ij} \propto g_i \left[ 2u_j - \left( \sum_k w_{ik} u_k \right) w_{ij} \right], \qquad \text{(Equation 14)}$$

where $w_{ij}$ is the synaptic weights from the $j$th pre-synaptic neuron to the $i$th post-synaptic neuron, $u_j$ is the input from the $j$th pre-synaptic neuron, $g_i$ is 1 for the highest activated post-synaptic neuron, $-0.4$ for the second highest activated post-synaptic neuron, and 0 for others. Batch normalization was performed on the pre-synaptic input $u_j$. Layers were trained one after another, such that when a higher-level layer was trained, all the lower-level layers were fixed. Adam optimizer was used to supervisedly train the top layer. Following,[70] the learning rate of Hebbian learning linearly decreased from the maximal value 0.04 at the first epoch to 0 at the last epoch, and the learning rate of supervised learning was kept at 0.0001. Both Hebbian and supervised training was performed with minibatch size 100 for 1000 epochs.

### Atari game task

The structure of the deep neural network to play Atari games closely followed those of[68] and.[16] Specifically, the neural network mapped 4 recent frames of size 84×84 to actions through 3 convolutional layers and 2 fully connected layers. Of the three convolutional layers, the kernel sizes were 8 × 8, 4×4 and 3×3 respectively, the strides were 4, 2 and 1 respectively, and the numbers of features were 32, 64 and 64 respectively. The first fully connected layers had 512 neurons. ReLU activation function was used.

EA was used to train neural networks whose synaptic weights took binary values $\{-\sqrt{2/N_{\text{in}}}, \sqrt{2/N_{\text{in}}}\}$, with $N_{\text{in}}$ being the input size. The training protocol closely followed that of.[16] Specifically, each generation had 1000 agents, each was evaluated by one episode (i.e., from the start to the end of a game, when the player succeeded or was killed in the game). The top 20 agents were selected to be parents and reproduced the next generation by flipping their binary weights with probability 0.002 (mutations). Then the top 10 agents were further evaluated by 30 episodes, based on which the elite agent was selected to be the agent that achieved the highest score. The elite agent became a member of the next generation without any mutation. The performance of EA at any training epoch was the average score of the elite agent in further 200 episodes. We trained the network in $2.5\times 10^8$ epochs. Each number of EA in Table 1 represents the median value of the scores in the final epochs of 5 training trials.

To perform Hebbian-Q learning, batch normalization was inserted between each layer of the network. The low-level layers were trained using Equation 14, and the last layer was trained supervisedly using Adam optimizer. When training a higher-level layer, all lower-level layers were kept fixed.

When unsupervisedly training a low-level layer, at each time step, a random action was taken of a game,[108] then minibatches of size 32 were randomly extracted from a buffer of $10^6$ recent frames to train the network using Equation 14. To train convolutional layers, we averaged over the weight updating at different spatial locations to update the sharing weights. Similar to,[70] learning rate linearly decreased from the maximal value 0.02 at the first epoch to 0 at the last epoch. The three convolutional layers were trained in $1.6\times 10^5$ epochs, and the first fully connected layer was trained in $4.8\times 10^5$ epochs.

When supervisedly training the top layer, minibatches of size 32 were randomly extracted from a buffer of $10^6$ recent frames at each epoch in a similar protocol to.[68] The action policy during training was $\epsilon$-greedy, with $\epsilon$ linearly annealed from 1 to 0.1 in the first $10^6$ epochs. The target network was cloned from the Q-network every 10000 epochs. Every $2\times 10^5$ epochs, the performance of the agent was evaluated by the average score over 200 episodes of games. We trained the top layer for $2\times 10^6$ epochs, and found no sign of continuing increase of performance. Each number of Hebbian-Q in Table 1 represents the maximal value of the evaluation scores in 2 training trials.

## QUANTIFICATION AND STATISTICAL ANALYSIS

All statistical analyses are presented as mean ± standard error of the mean (s.e.m.), calculated as $\sigma/\sqrt{n}$, where $\sigma$ is the standard deviation and $n$ is the number of trials. The s.e.m. is shown with error belts in Figure 5E and error bars in Figures 6B and 7D. Specifically, the classification accuracy in Figure 5E was averaged across 100 independent training trials ($n = 100$) and visualized with error belts. The psychometric curves in Figure 6B show the averaged choice percentages across 8 training trials ($n = 8$) with error bars. Similarly, the classification accuracy in Figure 7D was averaged across 8 training trials ($n = 8$) and displayed with error bars. All the results were analyzed using Python version 3.8 (Python Software Foundation, Wilmington, DE, USA) with NumPy package.