

Supplementary Information

577	A Framework schematic	16
578	B Gradient derivation for meta-learning plasticity parameters	16
579	B.1 Meta-loss	16
580	B.2 Plasticity gradient with REINFORCE approximation	17
581	B.2.1 Weight updates	17
582	B.2.2 Eligibility trace dynamics	18
583	B.2.3 Exact vs. approximate solution	18
584	Exact solution.	18
585	Approximate solution.	18
586	B.3 Gradient updates	19
587	B.3.1 Meta-gradient weighting	19
588	C Discussion	19
589	D Broader Impact	20
590	E Details on numerical implementation	20
591	E.1 Plastic network model	20
592	E.2 Biologically plausible training (inner loop)	20
593	E.3 Meta-learning loop (outer loop)	21
594	Supplementary References	21

595 A Framework schematic

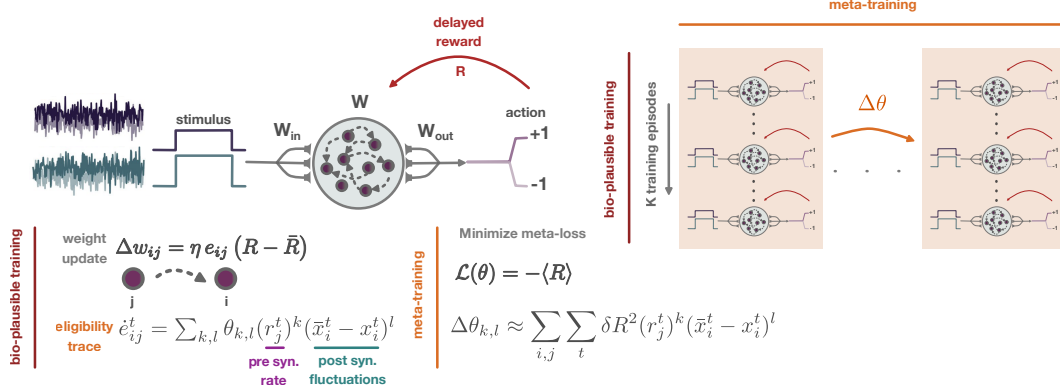


Figure 1

Outline of meta-learning framework.

596 B Gradient derivation for meta-learning plasticity parameters

597 We consider recurrent networks following the dynamics of Eq.(1)-(3). Each network performs a
 598 cognitive task for K **training episodes (trials)**. At the end of each episode, the network receives a
 599 scalar reward signal R that quantifies its task performance based on the output at the readout
 600 units. The network progressively adapts its recurrent connectivity using local, biologically plausible
 601 synaptic updates. The considered rules do not update synaptic weights continuously during each
 602 training episode. Instead, they accumulate correlations between pre- and post-synaptic activity into
 603 eligibility traces over the course of each episode [4, 2]. These traces are subsequently modulated by
 604 the reward signal received at the end of each training episode, implementing thereby structured credit
 605 assignment that reinforces or suppresses specific co-activation patterns based on their contribution
 606 to task performance.

607 Here, inspired by previous successful work on meta-learning plasticity rules [1, 8], instead of pre-
 608 scribing fixed learning rules, we consider a general parametric family of rules with parameters
 609 $\{\theta_{k,l}\}_{k,l=0}^d \in \mathbb{R}$ that indicate the order of correlations effective in the weight update

$$\frac{de_{ij}^t}{dt} = \mathcal{H}_\theta(r_j^t, x_i^t) - \frac{e_{ij}^t}{\tau_e} = \sum_{0 \leq k; l \leq d; } \theta_{k,l} (r_j^t)^k (\bar{x}_i - x_i^t)^l - \frac{e_{ij}^t}{\tau_e} \quad (10)$$

$$\Delta w_{ij} = e_{ij}^T (R - \bar{R}) - \frac{w_{ij}}{\tau_w} \quad (\text{at time } t = T). \quad (11)$$

610 Here τ_e and τ_w are eligibility trace and weight decay time constants, \bar{x}_i is a low-pass filtered version
 611 of the pre-activation of neuron i , and $\theta_{k,l} \in \mathbb{R}$ are learnable coefficients. As mentioned above, the
 612 weight updates happen only upon reception of a reward signal at the end of the training episode (at
 613 time $t = T$).

614 Our framework comprises two nested training loops: An **inner training loop (bio-plausible train-**
 615 **ing)** where the recurrent network is trained in biologically plausible way with local learning rules and
 616 sparse, delayed reinforcement signals. An **outer training loop (meta-training)** that adjust meta-
 617 parameters that determine the effect of the local learning rules by minimizing a meta-loss computed
 618 over K training episodes. We call each **sequence of K training episodes**, a training **session**.

619 For this outer meta-learning loop, we aim to maximize the expected reward $\langle R \rangle$ accumulated over
 620 K episodes with respect to the plasticity parameters $\{\theta_{k,l}\}$.

621 B.1 Meta-loss

622 We define the meta-loss of our framework as the expected reward under the learning rule parame-
 623 terized by θ

$$\mathcal{L}(\theta) = -\langle R(\theta) \rangle. \quad (12)$$

624 This expectation should be construed as being conditioned on the initial weight configuration (W_0),
 625 and taken over different session realizations. Variability across realizations arises from (i) stochas-
 626 ticity in the stimulus signal, (ii) randomness in the task structure (e.g., trial order), and (iii) small
 627 variability in the initial conditions of the neuronal states (see SI E).

628 Our goal, here, is to identify the set of plasticity parameters that maximize the expected reward, i.e.,
 629 minimize the loss $\mathcal{L}(\theta)$. Thus we need to compute the gradient of the loss with respect to θ , $\nabla_{\theta}\mathcal{L}(\theta)$.

630 B.2 Plasticity gradient with REINFORCE approximation

631 Following the REINFORCE estimator [10], we approximate the gradient of the expected reward by

$$\nabla_{\theta}\langle R \rangle = \langle (R - \bar{R}) \cdot \nabla_{\theta} \log \pi(R | \theta) \rangle. \quad (13)$$

632 This results from applying the log-derivative trick on the expectation of Eq. 13

$$\begin{aligned} \nabla_{\theta}\langle R \rangle &= \nabla_{\theta} \int \pi(R | \theta) R dR \\ &= \int \nabla_{\theta} \pi(R | \theta) R dR \quad (\text{Leibniz integral rule}) \\ &= \int \pi(R | \theta) \frac{\nabla_{\theta} \pi(R | \theta)}{\pi(R | \theta)} R dR \\ &= \int \pi(R | \theta) \nabla_{\theta} \log \pi(R | \theta) R dR \quad (\text{log-derivative trick}) \\ &= \langle R \nabla_{\theta} \log \pi(R | \theta) \rangle_{R \sim \pi(R | \theta)} \\ &\approx \left\langle \underbrace{(R - \bar{R})}_{\text{reward prediction error}} \nabla_{\theta} \log \pi(R | \theta) \right\rangle. \end{aligned}$$

633 In the last expression we have introduced the baseline reward \bar{R} as a control variate [3] commonly
 634 used for variance reduction of the expectation. In our setting, since the policy π is not explicitly
 635 defined, we adopt a deterministic approximation by treating the final synaptic weight matrix $\mathbf{W}(\theta)$
 636 as an **implicit policy**

$$\nabla_{\theta}\langle R \rangle \approx (R - \bar{R}) \frac{d\mathbf{W}}{d\theta}. \quad (14)$$

637 This heuristic uses the **reward prediction error** $\delta R = R - \bar{R}$ as a scaling factor for the direction
 638 of the update. This approximation assumes that R is a smooth functional of \mathbf{W} and that changes
 639 in θ affect R primarily through their effect on the connectivity \mathbf{W} . While $\nabla_{\theta} \log \pi(R | \theta)$ is not
 640 well-defined in our context, the derivative $\frac{d\mathbf{W}}{d\theta}$ serves as a proxy for the sensitivity of the reward to
 641 the meta-parameters.

642 We emphasize that this is a heuristic substitution. A fully rigorous treatment would require either
 643 introducing explicit stochasticity (e.g. input and task structure noise), or differentiating through the
 644 full network–task–environment loop, both of which are beyond the scope of the present work.

645 B.2.1 Weight updates

646 We update synaptic weights at the end of each training episode according to

$$\Delta w_{ij} = e_{ij}^T \delta R - \frac{w_{ij}}{\tau_w}, \quad (15)$$

647 where $w_{ij} \doteq [\mathbf{W}]_{ij}$ indicates the synaptic connection from the j -th pre-synaptic neuron to the i -th
 648 post-synaptic one.

649 We focus on the term that depends on the eligibility trace and thus on θ

$$\frac{d\Delta w_{ij}}{d\theta_{k,l}} = \delta R \frac{de_{ij}}{d\theta_{k,l}} \quad (16)$$

650 B.2.2 Eligibility trace dynamics

651 Eligibility traces evolve during each training episode according to the equation

$$\frac{de_{ij}}{dt} = \sum_{k=0}^d \sum_{l=0}^d \theta_{k,l} (r_j(t))^k (\bar{x}_i(t) - x_i(t))^l - \frac{e_{ij}}{\tau_e}. \quad (17)$$

652 Assuming discretised time steps and an online accumulation of $\frac{de_{ij}}{d\theta_{k,l}}$ during the trial, we obtain an
653 approximate derivative with respect to the plasticity parameters

$$\frac{de_{ij}}{d\theta_{k,l}} \approx \sum_{t=1}^T (r_j^t)^k (\bar{x}_i^t - x_i^t)^l. \quad (18)$$

654 Thus, the meta-gradient becomes

$$\frac{d\langle R \rangle}{d\theta_{k,l}} \approx \sum_{i,j} \delta R^2 \sum_{t=1}^T (r_j^t)^k (\bar{x}_i^t - x_i^t)^l. \quad (19)$$

655 B.2.3 Exact vs. approximate solution

656 The eligibility trace between pre-synaptic neuron j and post-synaptic neuron i evolves according to
657 the parameterized differential equation

$$\frac{de_{ij}^t}{dt} = \sum_{k=0}^d \sum_{l=0}^d \theta_{k,l} (r_j(t))^k (\bar{x}_i^t - x_i^t)^l - \frac{e_{ij}^t}{\tau_e}, \quad (20)$$

658 where \bar{x}_i^t is a low-pass filtered version (running average) of x_i^t , and τ_e is a decay time constant for
659 the trace.

660 **Exact solution.** Assuming $e_{ij}^0 = 0$ at the start of each training episode, the exact solution of this
661 equations is

$$e_{ij}^T = \int_0^T \exp\left(-\frac{T-s}{\tau_e}\right) \cdot \sum_{k=0}^d \sum_{l=0}^d \theta_{k,l} (r_j^t(s))^k (\bar{x}_i^t(s) - x_i^t(s))^l ds \quad (21)$$

662 This integral shows that the eligibility trace is a convolution over time of the nonlinear basis func-
663 tions with an exponential kernel. Consequently, its derivative with respect to $\theta_{k,l}$ is

$$\frac{de_{ij}(T)}{d\theta_{k,l}} = \int_0^T \exp\left(-\frac{T-s}{\tau_e}\right) (r_j^t(s))^k (\bar{x}_i^t(s) - x_i^t(s))^l ds. \quad (22)$$

664 This exact gradient includes a history-dependent term that integrates contributions from the entire
665 episode, weighted by an exponential decay.

666 **Approximate solution.** Here, we approximate the integral using a discretised online update of the
667 eligibility trace during task execution by assuming that **i.**) the gradient of e_{ij} with respect to $\theta_{k,l}$ can
668 be computed by tracking the eligibility basis function terms over time (the activity fluctuations and
669 rate product), **ii.**) the decay effect is absorbed into the eligibility trace update, and thus the gradient
670 can be approximated as a simple sum.

671 Hence, we obtain the simplified expression

$$\frac{de_{ij}}{d\theta_{k,l}} \approx \sum_{t=1}^T (r_j^t)^k (\bar{x}_i^t - x_i^t)^l. \quad (23)$$

672 This is equivalent to approximating the convolution in Eq. 22 with a rectangular window over dis-
673 crete time bins and ignoring the exponential decay envelope in the derivative path.

674 The approximation introduces a bias in the meta-gradient since it does not correctly weight the con-
675 tributions across time according to the decay constant τ_e . Nevertheless, the empirical performance
676 is acceptable when τ_e is large and the eligibility trace accumulates slowly over time. The approx-
677 imation significantly reduces computational cost and avoids the need to store memory demanding
678 activity trajectories.

679 B.3 Gradient updates

680 We update the plasticity parameters with

$$\theta_{k,l}^{m+1} \leftarrow \theta_{k,l}^m + \gamma \sum_{i,j} \delta R^2 \sum_{t=1}^T (r_j^t)^k (\bar{x}_i^t - x_i^t)^l, \quad (24)$$

681 where γ stands for the gradient learning rate, and m is the index of the meta-optimization step.

682 This derivation shows how the plasticity rule parameters are pushed in directions that correlate
683 with observed reward prediction errors and specific nonlinear interactions between pre- and post-
684 synaptic activity. Note, that this is fully differentiable and analytically tractable, avoiding the need
685 for backpropagation through time.

686 B.3.1 Meta-gradient weighting

687 The update outlined above considers only one training episode. However, it is highly unlikely that
688 any biological or artificial network will learn to execute a task in one shot, i.e. within a single training
689 episode, unless the initial weight configuration is already near an "optimal" (for the task) weight
690 configuration. Thus, here, we consider that training happens over K training episodes (trials) taking
691 part sequentially. We compute the gradient updates as a (weighted) average over the K training
692 episodes.

693 In our framework, we aim to optimize the expected cumulative reward over a sequence of K episodes
694 (or trials) during learning. The meta-objective is then expressed in terms of an expectation over the
695 episodes

$$\theta^* = \arg \max_{\theta} \langle R(\theta) \rangle \quad . \quad (25)$$

696 To perform this optimization, we compute the weighted empirical average of the gradient with re-
697 spect to plasticity parameters of the expected reward over the K episodes

$$\nabla_{\theta} \langle R \rangle \approx \sum_{p=1}^K w_p \nabla_{\theta} R_p(\theta). \quad (26)$$

698 This weighting allows us to give more importance to later episodes, where the network has had
699 time to adapt and the influence of the plasticity rule parametrised by θ on reward becomes more
700 pronounced. In the early stages of learning, the network's behavior is often dominated by random
701 initializations, making early rewards less informative for meta-optimization. Therefore, weighting
702 the gradient contributions from each episode (e.g., using exponentially increasing weights w_t) leads
703 to a more reliable estimate of the meta-gradient. This reflects the fact that early episodes provide
704 weaker or noisier gradient signals, while later ones carry more reliable credit assignment informa-
705 tion.

706 C Discussion

707 In this work, we introduced a meta-learning method for identifying biologically plausible synaptic
708 plasticity rules supporting structured credit assignment from sparse, delayed rewards. Our approach
709 parameterizes local learning rules as nonlinear combinations of pre- and post-synaptic activity, mod-
710 ulated by a reward prediction error, and optimizes their parameters through an outer-loop gradient-
711 based meta-optimization.

712 For the meta-optimization we avoided backpropagation through time by relying on analytically de-
713 rived gradient approximations. The key technical contribution of our work is this derivation of
714 fully analytic meta-gradients for the plasticity parameters, avoiding backpropagation through time.

715 Taking advantage of the structure of the weight update rule, where the weight changes depend on
716 accumulated eligibility traces, we compute gradients with respect to the plasticity rule parameters
717 through a REINFORCE-inspired approximation. This gradient requires access only to pre- and post-
718 synaptic activity traces and observed rewards, and can be computed without the memory-intensive
719 backpropagation through time that would otherwise be required to differentiate through the entire
720 learning session spanning K training episodes.

721 Here, we parameterized plasticity rules as accumulated nonlinear combinations of pre- and post-
722 synaptic activity, together with a decay term controlling the effective time scale of weight changes.
723 These rules were multiplicatively modulated by a reward prediction error, following the work of
724 Miconi [4]. While in our formulation we used a multiplicative interaction between the reward signal
725 and the accumulated co-activation (eligibility traces), supported by experimental evidence [7], alter-
726 native ways of introducing the reward into the synaptic update (e.g., nonlinear interactions) could
727 offer interesting directions for future work.

728 **D Broader Impact**

729 In this work we introduced a framework for discovering biologically plausible learning rules that
730 support structured credit assignment using only local synaptic information and delayed feedback.
731 We do not foresee any direct negative societal impact related to this work. The methods presented
732 here are theoretical and computational in nature, and are not intended for direct deployment in real-
733 world decision-making systems.

734 We foresee potential broader impact in systems neuroscience. This framework could be extended
735 to allow incorporating experimentally derived connectivity or activity constraints, or used in con-
736 junction with neural recordings to relate synaptic plasticity relates with computations resulting in
737 observed behavior.

738 **E Details on numerical implementation**

739 We performed all simulations in Python (PyTorch [6]) on a single CPU core. For generating the
740 experiment structure we used the Neurogym package (v2.0.0) [5] together with Gymnasium [9].
741 Unless otherwise stated we used a time-step of $\Delta t = 1$ (time unit), and decay time-constant $\tau = 10$
742 (time units).

743 **E.1 Plastic network model**

744 We used a rate-based RNN with $N = 200$ recurrent units, N_{in} inputs and N_{out} read-outs. We
745 simulated the continuous-time dynamics with forward Euler integration of the dynamical equations
746 Eq.(1)-(3).

747 For each synapse we assign one variable indicating its weight and one indicating the current state
748 of the associated eligibility trace. For computation of the activity fluctuation terms entering the
749 eligibility trace $(\bar{x}_i - x_i)$ we used a leaky integration with time constant $\tau_{\text{avg}} = 50$.

750 **E.2 Biologically plausible training (inner loop)**

751 During an episode weights do not change. Instead eligibility traces accumulate locally. At the
752 episode’s end, upon receiving reward R , synapses are updated by the three-factor rule of Eq.8. We
753 accumulate reward baselines for each trial type independently.

Algorithm 1 Inner loop – reward-modulated local plasticity

Require: Environment \mathcal{E} providing episodes of length T (in steps)

```
1: Plastic network model  $(W, W_{\text{in}}, W_{\text{out}}, \Theta)$ 
2: Number of episodes  $K$ , learning rate  $\eta$ , decay constants  $(\tau, \tau_e, \tau_w)$ ,

3: for  $p \leftarrow 1$  to  $K$  do ▷ loop over episodes
4:    $\mathcal{E}.\text{RESET}()$ ; obtain episode type  $c$ 
5:   Initialize neural states:  $x \leftarrow \mathcal{U}[-0.1, 0.1]$ ,  $\bar{x} \leftarrow x$ ,  $r \leftarrow 0$ 
6:   Zero eligibility traces:  $e_{ij} \leftarrow 0$ ,  $\forall i, j$ 
7:   for  $t \leftarrow 1$  to  $T$  do ▷ loop over timesteps
8:     Advance network state by one step
9:     for all synapses  $(i, j)$  in network do
10:      Update parametric eligibility
11:    end for
12:  end for
13:   $R \leftarrow \mathcal{E}.\text{REWARD}()$  ▷ scalar feedback
14:   $\bar{R} \leftarrow (1 - \beta) \bar{R} + \beta R$  ▷ per trial type
15:   $\delta R \leftarrow R - \bar{R}$ 
16:  for all synapses  $(i, j)$  do
17:    Update weight  $w_{ij}$ 
18:     $e_{ij} \leftarrow 0$  ▷ reset trace for next episode
19:  end for
20: end for
```

E.3 Meta-learning loop (outer loop)

The outer loop maximizes the expected cumulative reward across K episodes by updating $\Theta = \{\theta_{k,l}, \tau_e, \tau_w\}$. For each meta-iterations we ran either one or three independent sessions, i.e. independent realisations of the K episodes to account for stochasticity arising from task structure. Throughout we set $d = 3$, yielding polynomials of pre-synaptic rate and post-synaptic fluctuations up to order 3.

Algorithm 2 Outer loop – meta-optimization of plasticity parameters

Require: initial Θ_0 , learning rate γ , number of meta-iterations M

```
1: for  $m = 0, \dots, M$  do
2:   Initialize network weights  $W^{(0)}$  (Gaussian,  $\sigma = 1/\sqrt{N}$ )
3:   for  $s = 1, \dots, S$  do ▷ independent iterations of inner loop - multiple sessions
4:     Run Algorithm 1, obtain reward  $R_p$  and activity accumulator
5:   end for
6:   Compute meta-gradient  $\nabla_{\Theta} \langle R \rangle$  as an average over the gradients obtained from the  $S$  sessions
7:   Update plasticity parameters:  $\Theta \leftarrow \Theta + \gamma \nabla_{\Theta} \langle R \rangle$ 
8: end for
```

Supplementary References

- [1] Basile Confavreux, Poornima Ramesh, Pedro J Goncalves, Jakob H Macke, and Tim Vogels. Meta-learning families of plasticity rules in recurrent spiking networks using simulation-based inference. *Advances in Neural Information Processing Systems*, 36:13545–13558, 2023.
- [2] Eugene M Izhikevich. Solving the distal reward problem through linkage of stdp and dopamine signaling. *Cerebral cortex*, 17(10):2443–2452, 2007.
- [3] Christiane Lemieux. Control variates. *Wiley StatsRef: Statistics Reference Online*, pages 1–8, 2014.
- [4] Thomas Miconi. Biologically plausible learning in recurrent neural networks reproduces neural dynamics observed during cognitive tasks. *Elife*, 6:e20899, 2017.

- 770 [5] Manuel Molano-Mazon, Joao Barbosa, Jordi Pastor-Ciurana, Marta Fradera, Ru-Yuan Zhang,
771 Jeremy Forest, Jorge del Pozo Lerida, Li Ji-An, Christopher J Cueva, Jaime de la Rocha, et al.
772 Neurogym: An open resource for developing and sharing neuroscience tasks. 2022.
- 773 [6] A Paszke. Pytorch: An imperative style, high-performance deep learning library. *arXiv*
774 *preprint arXiv:1912.01703*, 2019.
- 775 [7] Adithya E Rajagopalan, Ran Darshan, Karen L Hibbard, James E Fitzgerald, and Glenn C
776 Turner. Reward expectations direct learning and drive operant matching in drosophila. *Pro-*
777 *ceedings of the National Academy of Sciences*, 120(39):e2221415120, 2023.
- 778 [8] Navid Shervani-Tabar and Robert Rosenbaum. Meta-learning biologically plausible plasticity
779 rules with random feedback pathways. *Nature Communications*, 14(1):1805, 2023.
- 780 [9] Mark Towers, Ariel Kwiatkowski, Jordan Terry, John U Balis, Gianluca De Cola, Tristan
781 Deleu, Manuel Goulao, Andreas Kallinteris, Markus Krimmel, Arjun KG, et al. Gym-
782 nasium: A standard interface for reinforcement learning environments. *arXiv preprint*
783 *arXiv:2407.17032*, 2024.
- 784 [10] Ronald J Williams. Simple statistical gradient-following algorithms for connectionist rein-
785 forcement learning. *Machine learning*, 8(3-4):229–256, 1992.