A solution to temporal credit assignment using cell-type-specific modulatory signals

Yuhan Helena Liu[1,*], Stephen Smith[2], Stefan Mihalas[2], Eric Shea-Brown[1,2,3], and Uygar Sümbül[2,*]

[1]Department of Applied Mathematics, University of Washington, Seattle, WA, USA
[2]Allen Institute, 615 Westlake Ave N, Seattle WA, USA
[3]Computational Neuroscience Center, University of Washington, Seattle, WA, USA
[*]Correspondence: hyliu24@uw.edu, uygars@alleninstitute.org

Abstract

Animals learn and form memories by jointly adjusting the efficacy of their synapses. How they efficiently solve the underlying temporal credit assignment problem remains elusive. Here, we re-analyze the mathematical basis of gradient descent learning in recurrent spiking neural networks (RSNNs) in light of the recent single-cell transcriptomic evidence for cell-type-specific local neuropeptide signaling in the cortex. Our normative theory posits an important role for the notion of neuronal cell types and local diffusive communication by enabling biologically plausible and efficient weight update. While obeying fundamental biological constraints, including separating excitatory vs inhibitory cell types and observing connection sparsity, we trained RSNNs for temporal credit assignment tasks spanning seconds and observed that the inclusion of local modulatory signaling improved learning efficiency. Our learning rule puts forth a novel form of interaction between modulatory signals and synaptic transmission. Moreover, it suggests a computationally efficient on-chip learning method for bio-inspired artificial intelligence.

Introduction

Animals adapt to their environments by learning and executing temporally structured action patterns. A well-established biological substrate for lasting behavioral changes and temporally precise action generation is synaptic plasticity, where the efficacy of synaptic transmission is altered to produce more favorable actions. The challenge of assigning the credit (or blame) to individual synapses for guiding their appropriate strengthening (or weakening) is known as the temporal credit assignment problem. The brain solves this problem seemingly effortlessly in a wide range of tasks. Yet, our understanding of its underpinnings is rudimentary. Despite a rich literature and recent advances [1, 2, 3] on how such intelligence can be constructed artificially, efficient, scalable, real-time solutions to the credit assignment problem remain elusive for artificial intelligence (AI) as well.

While biological plausibility (e.g., locality of update rules) is not a constraint for AI, scalability in memory and computation remains a challenge for theoretical solutions to temporal credit assignment. Computer scientists made remarkable progress in solving the credit assignment problem for artificial neural networks by applying the backpropagation algorithm and its adaptation to problems with a temporal dimension, known as backpropagation through time (BPTT) [4]. However, scalability, which manifests itself as demanding energy requirements for complicated tasks, and the need to store the network history, which precludes real-time updates, remain as significant challenges. The real-time recurrent learning (RTRL) algorithm uses a different factorization of the BPTT objective and solves the online learning problem [5]. However, this is achieved at the expense of a very heavy memory burden.

One conspicuous gap between computational models of neuronal networks and experimental data appears in the concept of cell types. Remarkable diversity and stereotypy have been observed in neuronal phenotypes [6, 7, 8, 9, 10]. Despite recent attempts at bridging this gap by explicitly studying discrete cell types as part of the computational model [11], an overarching role for cell types in synaptic plasticity is not yet known.

Hebbian learning is based on correlation of pre- and post-synaptic firing, and provides a strong foundation for biologically plausible learning. Modern versions also recognize a role for a third factor, top-down instructive or "reward" signals [12, 1, 13, 14, 15, 16], such as secretion of dopamine from diffusely ramifying axons [17, 18, 19], and for persistent “eligibility traces,” to bridge the temporal gap between correlated firing at specific synapses and later,

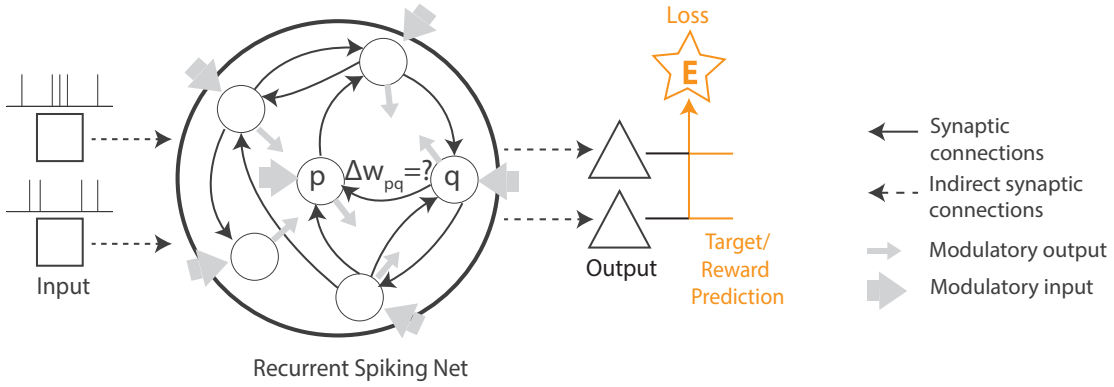

Figure 1: Temporal credit assignment through the interplay of synaptic transmission and modulatory signaling. The network is tasked with producing a desired output signal given a certain input. The challenge involves determining how much each weight (out of potentially thousands or millions of connections) is responsible for overall network performance, so as to guide the strengthening and weakening of the individual weights. In this view of neuronal network as a stack of synaptic and modulatory networks, learning (i.e the update of synaptic weights w) is shaped by both local synaptic activity and modulatory signaling.

more diffuse reward signals [20, 21, 22]. While these approaches succeed in learning more complicated tasks *in-silico*, the performance of biologically plausible learning still lags significantly that of BPTT-based learning in recurrently connected artificial neural networks [23].

In addition to the top-down learning signal as the third factor, local modulatory signals, such as neuropeptide (NP) molecules [24], are also implicated in synaptic learning [25, 26, 27], although their specific roles remain largely unexplored. NPs are secreted by source neurons, broadcast over the local tissue via diffusion, and detected by neurons with the cognate G-protein coupled receptors (GPCRs) [28]. Recent single-cell RNA-seq transcriptomes have revealed that almost all cortical neurons express at least one NP precursor and one NP-selective GPCR gene, and NP-GPCR signaling is cell type-specific [29]. These findings suggest a view of cortical computation and plasticity involving an interplay between fast synaptic and slow, cell-type-based broadcast communication [30] (Figure 1). Thus, biological neural networks are *multidigraphs* – networks having a stack of *multi*ple connection types between cells, each with a *di*rection of action.

A major step forward in biologically plausible learning in recurrent neural networks (RNNs) – widely-adopted high dimensional dynamical models of neural circuits for robust performance in temporal tasks [31] – was brought by two recent theoretical studies that developed local and causal learning rules [32, 23]. These studies derived local approximations to gradient-based learning in RNNs by requiring synaptic weight updates to depend only on local information about pre- and postsynaptic activities in addition to a top-down learning signal pertaining to network output error. While Murray approximated RTRL for rate-based networks [32], Bellec and colleagues approximated BPTT to train recurrent spiking neural networks (RSNNs) [23]; this spike-based communication yields greater biological plausibility and energy efficiency [33, 34, 35]. The same group also explicitly allowed for cellular dynamics to vary according to distinct cell types, including neurons with firing threshold adaptation [36], which were demonstrated to facilitate memory tasks [23].

Building on these recent advances, we test the plausibility of the abstract multidigraph concept by formulating it into an explicit computational model and describe a computational role for cell types in synaptic learning as part of our model. More specifically, we truncate the RTRL algorithm to remove nonlocal dependencies, but include modulatory terms respecting neuronal types to provide nonlocal information in the form of diffusive signaling. (truncations of the RTRL algorithm has received recent attention from the machine learning and neuromorphic hardware communities [37, 38].) Our multidigraph learning rule (MDGL) generalizes multi-factor learning, in which a Hebbian eligibility trace is combined with local cell-type-specific modulatory signals in addition to the top-down instructive signals. We train the multiple-cell-type RSNNs [23] with MDGL to perform tasks involving temporal credit assignment over a timescale of seconds. Although we focus on supervised learning, our theory can be extended to reinforcement learning settings following Bellec *et al.* [23]. Our proof-of-concept implementation of MDGL shows significant improvements over previous literature and advances the field of biologically plausible temporal credit assignment. From a neuroscience perspective, our study proposes a new model of cortical learning shaped by the interplay of local modulatory signaling and synaptic transmission, and potentially brings us closer to understanding biological intelligence. From a computer science perspective, our method offers an energy efficient method for on-chip

neuro-inspired AI.

Results

Mathematical basis of multidigraph learning in RSNNs

To study the basic principles governing plasticity in neuronal circuits, we use a simple and widely adopted recurrent neural network model with the addition that we endow neurons with local modulatory signaling and cell type-specific cognate reception.

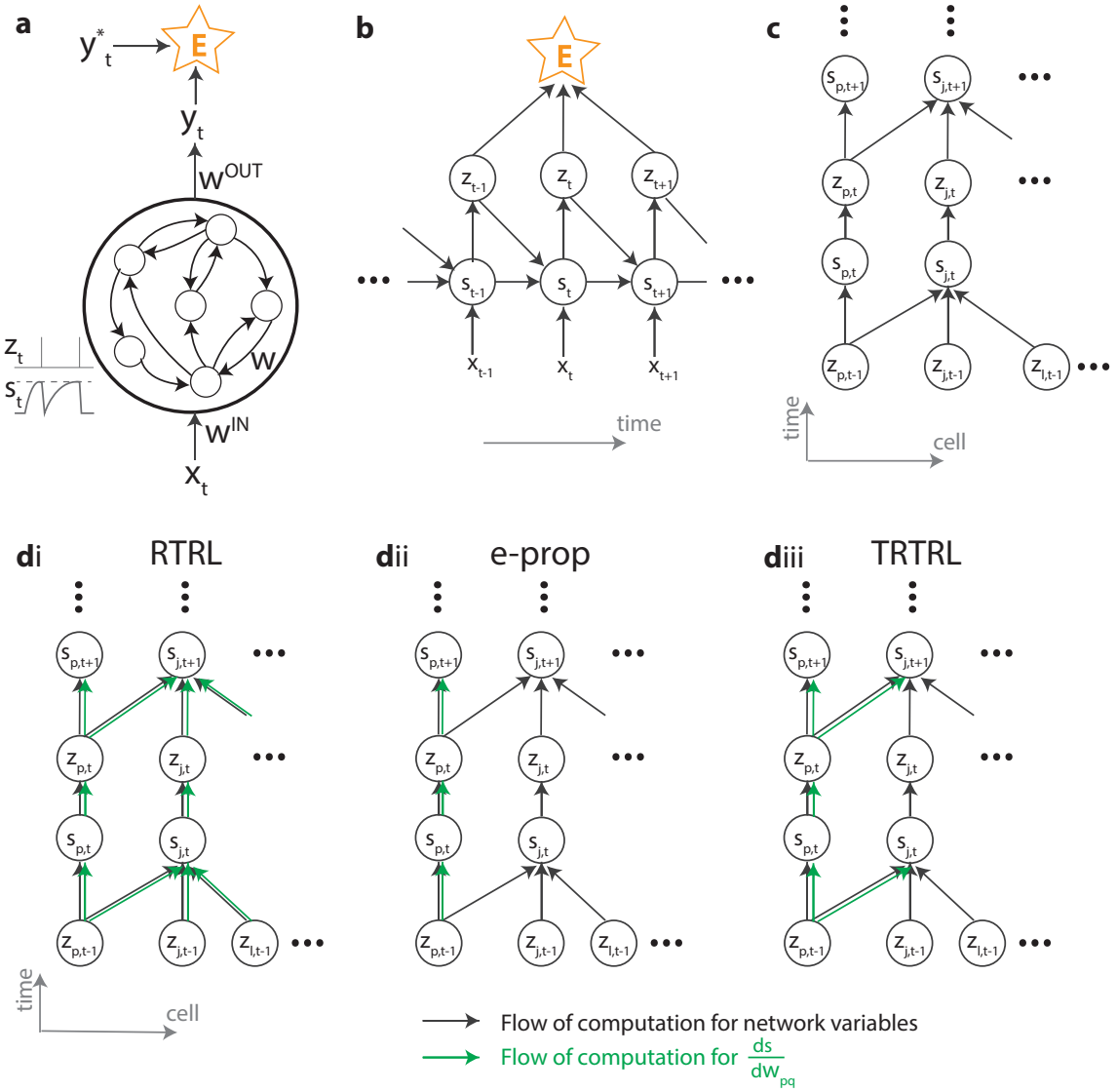

Figure 2: Computational graph and gradient propagation. a) Schematic illustration of the recurrent neural network used in this study. b) The mathematical dependencies of input x, state s, neuron spikes z and loss function E unwrapped across time. c) The dependencies of state s and neuron spikes z unwrapped across time and cells. d) The computational flow of $\mathrm{d}\,s/\,\mathrm{d}\,w_{pq}$ is illustrated for (di) exact gradients computed using RTRL, (dii) e-prop and (diii) our truncation in Eq. (6), where dependency within one connection step has been kept. Black arrows denote the computational flow of network states, output and the loss; for instance, the forward arrows from z_t and s_t going to s_{t+1} are due to the neuronal dynamics equation in Eq. (1). Green arrows denote the computational flow of $\mathrm{d}\,s/\,\mathrm{d}\,w_{pq}$ for various learning rules.

The RSNN used in this study (Figure 1) takes N_{in} spiking inputs $x_{i,t}$ for $i = 1, 2, ...N_{in}$ and $t = 1, 2, ..., T$, where $x_{i,t}$ assumes the value 1 if input unit i fires at time t and 0 otherwise. These inputs are sent to the spiking recurrent units. The output of the jth recurrent neuron at time t, $z_{j,t}$ for $j = 1, 2, ...N$, also takes the value 1 if recurrent

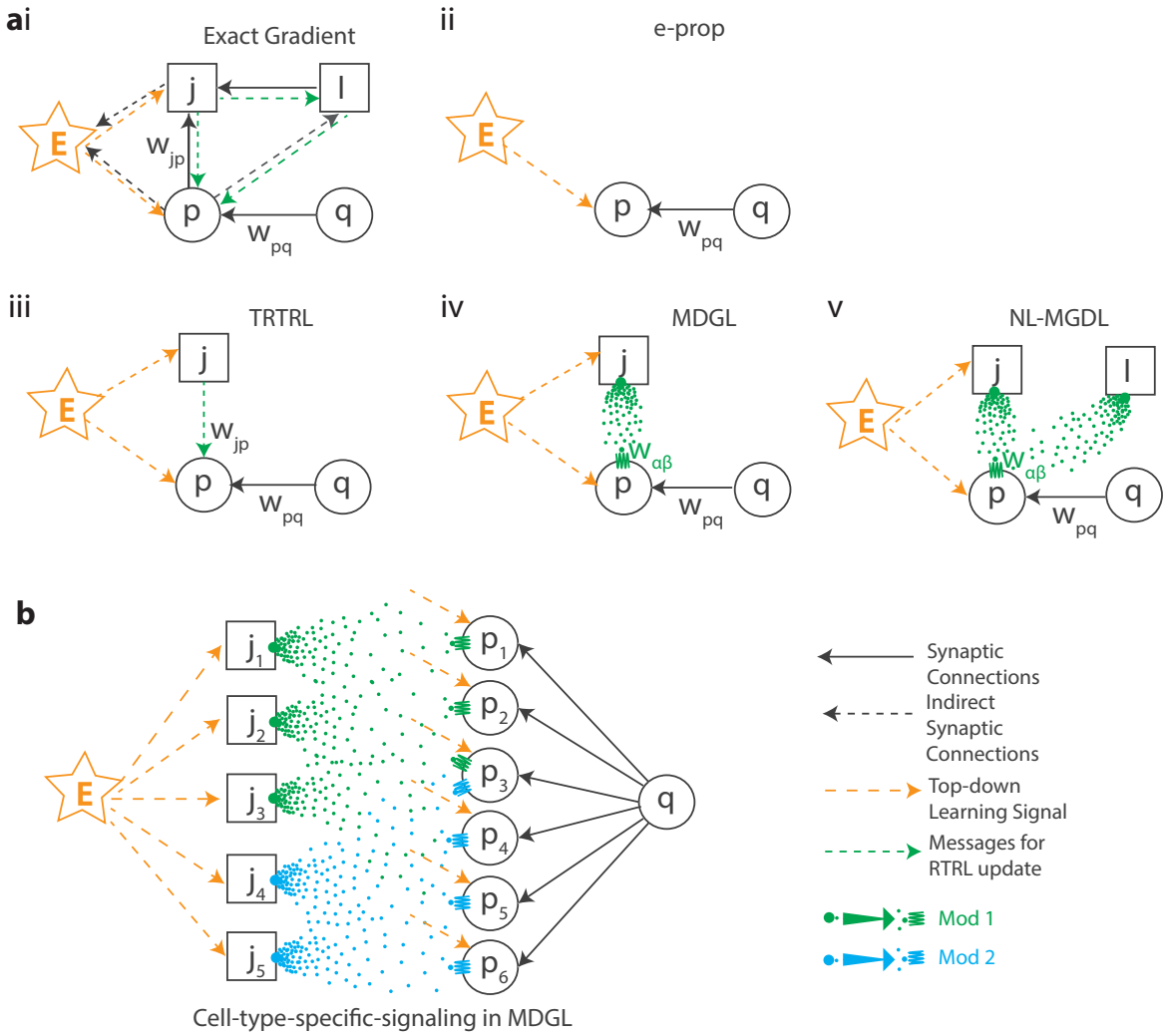

Figure 3: Biologically plausible temporal credit assignment using cell-type-specific modulatory signals. a) Learning rules investigated include (i) the exact gradient: updating weight w_{pq}, the synaptic connection strength from presynaptic neuron q to postsynaptic neuron p, involves nonlocal information inaccessible to neural circuits, i.e. the knowledge of activity for all neurons j and l in the network. This is because w_{pq} affects the activities of many other cells through indirect connections, which will then affect the network output at subsequent time steps. As in Figure 1, E quantifies the network's performance. (ii) E-prop, a state-of-the-art local learning rule, restricts weight update to depend only on pre- and post-synaptic activity as well as a top-down learning signal. (iii) Our truncated weight update (TRTRL) includes dependencies within one connection step, which are omitted in e-prop. (iv) Our multi-digraph learning rule (MDGL) incorporates local cell-type-specific modulatory signaling through which the activity of neuron j can be delivered to neuron p. (v) NL-MDGL: A nonlocal version of MDGL, where modulatory signal diffuses to all cells in the network. b) Illustration of cell-type-specific modulatory signaling in MDGL: source neurons j_1, j_2 and j_3 express the same precursor type, and likewise for j_4 and j_5. Neurons p_1 to p_6 can be grouped into three types based on the different combination of receptors they express. This results in a modulatory network described by a cell-type-specific channel gain on top of the classical synaptic network.

neuron j fires at time t and 0 otherwise. The recurrent activity is read out to graded output $y_{k,t}$ for $k = 1, 2, ...N_y$, whose performance for a given task incurs a feedback signal E (e.g., error or negative reward). Throughout, many variables of interest display spatial (e.g., cell index p, synapse index pq) and temporal dependencies. The dynamics of this network is governed by

$$z_{j,t} = H(s_{j,t} - v_{\text{th}})$$
$$s_{j,t+1} = \eta\, s_{j,t} + (1-\eta)\left(\sum_{l\neq j} w_{jl}\, z_{l,t} + \sum_{p} w^{\text{IN}}_{jm}\, x_{m,t+1}\right) - z_{j,t}\, v_{\text{th}} \tag{1}$$

$$y_{k,t} = \kappa\, y_{k,t-1} + (1-\kappa) \sum_j w_{kj}^{\mathrm{OUT}} z_{j,t} + b_k^{\mathrm{OUT}} \tag{2}$$

where $s_{j,t}$ denotes the membrane potential for neuron j at time t, v_{th} denotes the spiking threshold potential, $\eta = e^{-dt/\tau_m}$ denotes the leak factor for simulation time step dt and membrane time constant τ_m, w_{lj} denotes the weight of the synaptic connection from neuron j to l, w_{jm}^{IN} denotes the efficacy of the connection between the input neuron m and neuron j, and H denotes the Heaviside step function. For the output, w_{kj}^{OUT} denotes the efficacy of the connection from neuron j to output neuron k, and b_k^{OUT} denotes the bias of the k-th output neuron.

We study iterative adjustment of all synaptic weights (input weights w^{IN}, recurrent weights w and output weights w^{OUT}) using gradient descent on loss E defined as a function of the difference between the network output y and the desired output $y*$ (see Methods for detailed definitions of E):

$$\begin{aligned} w_{pq,\mathrm{new}} &= w_{pq,\mathrm{old}} - \lambda \Delta w_{pq}, \\ \Delta w_{pq} &= \frac{\mathrm{d}\, E}{\mathrm{d}\, w_{pq,\mathrm{old}}}, \end{aligned} \tag{3}$$

where λ denotes the learning rate, and the gradient of the error with respect to the synaptic weights must be calculated. BPTT and RTRL calculate this by unwrapping the RSNN dynamics over time; this unwrapping is needed because weights influence past network activity, which then influence present and future activity through (1) (Figure 2b). While these two algorithms yield equivalent results, their bookkeeping for the gradient calculations differs [32]. Gradient calculations in BPTT depend on future activity, which poses an obstacle for online learning and biological plausibility. Unlike BPTT, the computational dependency graph of RTRL is causal. Therefore, following RTRL, we factor the error gradient as follows:

$$\begin{aligned} \frac{\mathrm{d}\, E}{\mathrm{d}\, w_{pq}} &= \sum_{j,t} \frac{\partial E}{\partial z_{j,t}} \frac{\mathrm{d}\, z_{j,t}}{\mathrm{d}\, w_{pq}}, \\ \frac{\mathrm{d}\, z_{j,t}}{\mathrm{d}\, w_{pq}} &= \frac{\partial z_{j,t}}{\partial s_{j,t}} \frac{\mathrm{d}\, s_{j,t}}{\mathrm{d}\, w_{pq}}, \end{aligned} \tag{4}$$

where notation ∂ denotes that the derivative accounting only for direct dependency, and notation d denotes that the derivative accounts for all direct and indirectly dependencies (explained in Methods). The factor $\frac{\partial E}{\partial z_{j,t}}$ in (4) is related to the top-down learning signal $L_{j,t} := \sum_k w_{kj}^{OUT}(y_{k,t} - y_{k,t}^*)$ defined in Bellec *et al.* [23]. Section S3.2 of supplementary materials shows that the leak term of the output neurons makes these two terms different, and derives an online implementation that uses the top-down learning signal definition of Bellec *et al* [23]. For readability, however, we simply refer to $\frac{\partial E}{\partial z_{j,t}}$ as top-down learning signal in the main text, leaving the detailed expansion of derivatives to the supplementary materials.

We now discuss the second factor in (4), i.e. $\frac{\mathrm{d}\, z_{j,t}}{\mathrm{d}\, w_{pq}}$. This is expanded in two factors in (4). The first factor, $h_{j,t} := \frac{\partial z_{j,t}}{\partial s_{j,t}}$ is a surrogate gradient (Methods) to overcome the non-differentiability of spiking neurons [39, 23]. The second factor, $\frac{\mathrm{d}\, s_{j,t}}{\mathrm{d}\, w_{pq}}$, accounts for both spatial and temporal dependencies in RSNNs. One can see from (4) that the error gradient is factored across both time and space (recurrent cells). Factoring across time t, as explained above, results from unwrapping the temporal dependencies illustrated in Figure 2b. Factoring across space, however, is due to the indirect dependencies (of all z_t on w and all $z_{t'}$ $(t' < t)$) arising from recurrent connections, which is illustrated in Figure 2c. These recurrent dependencies are all accounted for in the $\frac{\mathrm{d}\, s_{j,t}}{\mathrm{d}\, w_{pq}}$ factor, which can be obtained recursively as follows:

$$\begin{aligned} \frac{\mathrm{d}\, s_{j,t}}{\mathrm{d}\, w_{pq}} &= \frac{\partial s_{j,t}}{\partial w_{pq}} + \sum_l \frac{\partial s_{j,t}}{\partial s_{l,t-1}} \frac{\mathrm{d}\, s_{l,t-1}}{\mathrm{d}\, w_{pq}} \\ &= \frac{\partial s_{j,t}}{\partial w_{pq}} + \frac{\partial s_{j,t}}{\partial s_{j,t-1}} \frac{\mathrm{d}\, s_{j,t-1}}{\mathrm{d}\, w_{pq}} + \underbrace{\textstyle\sum_{l \neq j} w_{jl} \frac{\partial z_{l,t-1}}{\partial s_{l,t-1}} \frac{\mathrm{d}\, s_{l,t-1}}{\mathrm{d}\, w_{pq}}}_{\text{depends on all weights } w_{jl}}. \end{aligned} \tag{5}$$

Thus, factor $\frac{\mathrm{d}\, s_{j,t}}{\mathrm{d}\, w_{pq}}$ is a memory trace of all inter-cellular dependencies (Figures 2di, 3ai), requires $O(N^3)$ memory and $O(N^4)$ computations. This makes RTRL expensive to implement for large networks. Moreover, this last factor poses a serious problem for biological plausibility: it involves nonlocal terms, so that knowledge of all other weights in the network is required in order to update the weight w_{pq}.

To address this, Murray [32] and Bellec *et al.* [23] dropped the nonlocal terms so that the updates to weight w_{pq} would only depend on pre- and post-synaptic activity (Figures 2dii, 3aii, where we use the name *e-prop* given by [23]). Murray [32] applied this truncation to train rate-based neural networks, and Bellec *et al.* [23] used their algorithm to train networks of spiking neurons. While both work succeed in improving over previous biologically plausible learning rules, a significant performance gap with respect to the full BPTT/RTRL algorithms remains. This gap is not surprising given that both algorithms account only for pre- and post-synaptic activities, ignoring by design the many potential contributions to optimal credit assignment from neurons that do not participate directly in the synapse of interest.

A potential role for cell type-specific modulatory signals

To reveal a potential role for cell-type-based modulatory signals in synaptic plasticity, we begin by partially restoring non-local dependencies between cells – those within one connection step. This is the *truncated* RTRL framework (Figures 2diii, 3aiii), and the memory trace term $\frac{\mathrm{d}\, s_{j,t}}{\mathrm{d}\, w_{pq}}$ becomes

$$\frac{\mathrm{d}\, s_{j,t}}{\mathrm{d}\, w_{pq}} \approx \begin{cases} \frac{\partial s_{j,t}}{\partial z_{p,t-1}} \frac{\partial z_{p,t-1}}{\partial s_{p,t-1}} \frac{\mathrm{d}\, s_{p,t-1}}{\mathrm{d}\, w_{pq}} = w_{jp} \frac{\partial z_{p,t-1}}{\partial s_{p,t-1}} \frac{\mathrm{d}\, s_{p,t-1}}{\mathrm{d}\, w_{pq}}, & p \neq j \\ \frac{\partial s_{j,t}}{\partial w_{jq}} + \frac{\partial s_{j,t}}{\partial s_{j,t-1}} \frac{\mathrm{d}\, s_{j,t-1}}{\mathrm{d}\, w_{jq}}, & p = j \end{cases} \tag{6}$$

Thus, when $j = p$, our truncation implements $\frac{\mathrm{d}\, s_{p,t}}{\mathrm{d}\, w_{pq}} \approx \frac{\partial s_{j,t}}{\partial w_{jq}} + \frac{\partial s_{j,t}}{\partial s_{j,t-1}} \frac{\mathrm{d}\, s_{j,t-1}}{\mathrm{d}\, w_{jq}}$, which coincides with e-prop. What we are adding in (6) is the case when $p \neq j$, for which $\frac{\mathrm{d}\, s_{j,t}}{\mathrm{d}\, w_{pq}}$ was simply set to 0 in e-prop. We note that the truncation in (6) resembles the n-step RTRL approximation recently proposed in [37], known as SnAP-n, which stores $\frac{\mathrm{d}\, s_{j,t}}{\mathrm{d}\, w_{pq}}$ only for j such that parameter w_{pq} influences the activity of unit j within n time steps. The computations of SnAp-n converge to those of RTRL as n increases. Our truncation in (6) is similar to SnAp-n with $n = 2$ with two differences: (i) we apply it to spiking neural networks, (ii) we drop the previous time step's Jacobian term $\frac{\mathrm{d}\, s_{j,t-1}}{\mathrm{d}\, w_{pq}}$, which would necessitate the maintenance of a rank-three ("3-d") tensor with costly storage demands ($O(N^3)$) and for which no known biological mechanisms exist.

By substituting equation (6) into (4) and (4), we approximate the overall gradient as

$$\widehat{\frac{\mathrm{d}\, E}{\mathrm{d}\, w_{pq}}} = \sum_t \underbrace{\frac{\partial E}{\partial z_{p,t}} \frac{\partial z_{p,t}}{\partial s_{p,t}} \left(\frac{\partial s_{p,t}}{\partial w_{pq}} + \frac{\partial s_{p,t}}{\partial s_{p,t-1}} \frac{\mathrm{d}\, s_{p,t-1}}{\mathrm{d}\, w_{pq}} \right)}_{\left.\frac{\mathrm{d}\, E}{\mathrm{d}\, w_{pq}}\right|_{\text{e-prop}}} + \underbrace{\sum_j \frac{\partial E}{\partial z_{j,t}} \frac{\partial z_{j,t}}{\partial s_{j,t}} w_{jp} \frac{\partial z_{p,t-1}}{\partial s_{p,t-1}} \frac{\mathrm{d}\, s_{p,t-1}}{\mathrm{d}\, w_{pq}}}_{:=\widehat{\Gamma_{pq,t}}}. \tag{7}$$

Here, the first term alone gives exaclty the e-prop synaptic update rule. The second term, which we define as $\widehat{\Gamma_{pq,t}}$, is a synaptically non-local term that is ignored by e-prop. As seen in (7), our truncation requires maintaining a $\{p, q\}$-dependent double tensor for $\frac{\mathrm{d}\, s_{p,t-1}}{\mathrm{d}\, w_{pq}}$ instead of a triple one, thereby reducing the memory cost reduction from $O(N^3)$ of RTRL to $O(N^2)$.

Importantly, we observe that, for the update to synapse w_{pq} in (7), the terms that depend on cells j *only appear under a sum.* Therefore, the mechanism updating the synapse (pq) does not need to know the individual terms indexed by j. Rather, only their sum suffices. This observation is key in hypothesizing a role for diffuse neuromodulatory signalling as an additional factor in synaptic plasticity. As discussed above, multiple different neuromodulators diffuse in the shared intercellular space [29, 30]; this provides a natural biological substrate for the transmission of weighted, summmed signals .

While it is tempting to consider the first factors in $\widehat{\Gamma_{pq,t}}$, $\frac{\partial E}{\partial z_{j,t}} \frac{\partial z_{j,t}}{\partial s_{j,t}} w_{jp}$, as the modulatory signal emitted by neuron j, the involvement of the synapse from neuron p via w_{jp} and a lack of known biological mechanisms in calculating the composite signal suggest that this is unlikely to be implemented simply. Instead, inspired by the summation over j in Equation (7) and the cell-type-specific nature of peptidergic neuromodulation [29, 30], we propose to approximate the signaling gain w_{jp} in Equation (7) by its cell type-based mean $w_{\alpha\beta}$, where cell j belongs to type α and cell p belongs to type β (i.e., $w_{\alpha\beta} =< w_{jp} >_{j\in\alpha, p\in\beta}$). Specifically, we hypothesize that $w_{\alpha\beta}$ represents the affinity of the G-protein coupled receptors expressed by cells of type β to the peptides secreted by cells of type α (Figure 3aiv, b). Finally, the local diffusion hypothesis discussed in [28] suggests a further approximation, in which this type of signaling is registered only by local synaptic partners and therefore preserves the connectivity structure

of w_{jp}:

$$\frac{\partial E}{\partial z_{j,t}} \frac{\partial z_{j,t}}{\partial s_{j,t}} w_{jp} \approx \begin{cases} a_{j,t} w_{\alpha\beta}, & p \to j \\ 0, & \text{otherwise} \end{cases} \tag{8}$$

where $p \to j$ denotes that there is a synaptic connection from neuron p to j, and

$$a_{j,t} = \frac{\partial E}{\partial z_{j,t}} \frac{\partial z_{j,t}}{\partial s_{j,t}} \tag{9}$$

denotes the activity-dependent modulatory signal emitted by neuron j at time t. In other words, while $a_{j,t}$ is emitted by neuron j, its effect on neuron p through diffusion, as part of a sum, is given by Equation (8). The activity-dependent modulatory signal secreted by neuron j, $a_{j,t}$, has two components: $\frac{\partial E}{\partial z_{j,t}}$, which is referred to as the top-down signal by Bellec *et al.* [23], and $h_{j,t} = \partial z_{j,t}/\partial s_{j,t}$, which is the pseudo-derivative of spiking activity as a function of cell j's membrane potential [23, 40, 41].

As in e-prop [23], define

$$e_{pq,t} := \frac{\partial z_{p,t}}{\partial s_{p,t}} \frac{\mathrm{d}\, s_{p,t}}{\mathrm{d}\, w_{pq}} \tag{10}$$

be the *eligibility trace* maintained by postsynaptic cell p to keep a memory of preceding activity presynaptic cell q and postsynaptic cell p. An explanation for viewing eligibility traces as derivatives can be found in [23].

Bringing these together, the gradient estimate at time t due to our learning rule is given as

$$\left.\frac{\mathrm{d}\, E}{\mathrm{d}\, w_{pq}}\right|_t \approx \left.\frac{\mathrm{d}\, E}{\mathrm{d}\, w_{pq}}\right|_{t,\text{e-prop}} + \Gamma_{pq,t}, \tag{11}$$

$$\Gamma_{pq,t} = \left(\sum_{\alpha \in C} w_{\alpha\beta} \sum_{j \in \alpha, p \to j} a_{j,t} \right) e_{pq,t-1}, \tag{12}$$

where neuron p is of type β, C denotes the set of neuronal cell types, $\Gamma_{pq,t} := \sum_{j \neq p} \frac{\partial E}{\partial z_{j,t}} \frac{\partial z_{j,t}}{\partial s_{j,t}} w_{\alpha\beta} \frac{\partial z_{p,t-1}}{\partial s_{p,t-1}} \frac{\mathrm{d}\, s_{p,t-1}}{\mathrm{d}\, w_{pq}}$ approximates the second term in (7) with cell-type-specific weight averages. Note that (12) factorizes $\Gamma_{pq,t}$ using a double sum; a sum over the cell types and a sum over cells belonging to individual types. Thus, our update rule suggests a new additive term to compute the plasticity update at synapse (pq) at time t, $\Gamma_{pq,t}$, which calculates multiplicative contributions of the modulatory signal $a_{j,t}$ secreted by neuron j, the affinity of receptors of cell type β to ligands of type α, $w_{\alpha\beta}$, and the eligibility trace at the synapse (pq), $e_{pq,t}$.

In summary, we have proposed a new rule for updating a synapse w_{pq}, which we refer to as the multidigraph learning rule, or MDGL, and represent by $\Delta w_{pq}|_{MDGL}$. As illustrated in Fig. 3, this begins with the same basic structure as e-prop, which has the following form:

$$\Delta w_{pq}|_{e-prop} \propto (\text{Top-down learning signal p}) \times (\text{eligibility trace pq}). \tag{13}$$

$\Delta w_{pq}|_{MDGL}$ then adds a term Γ_{pq}, whose form matches that of diffusive communication via a cell-type specific modulatory network:

$$\begin{aligned} \Delta w_{pq}|_{MDGL} &\propto \Delta w_{pq}|_{e-prop} + \Gamma_{pq}, \\ \Gamma_{pq} &\approx \left(\sum_{\alpha \in C} (\text{cell-type-specific gain}) \times \sum_{j \in \alpha} \underbrace{(\text{Mod. signal secreted by j})}_{(\text{Top-down signal j}) \times (\text{activity j})} \right) \times (\text{eligibility trace pq}). \end{aligned} \tag{14}$$

Thus, the Hebbian eligibility trace is not only compounded with top-down learning signals – as in modern biologically plausible learning rules [23] – but also integrated with cell-type-specific, diffuse modulatory signals. This creates a unified framework that integrates the eligibility trace, local and top-down modulatory signals into a new multi-factor learning rule.

Simulation of multidigraph learning in RSNNs

To test the efficiency of the MDGL formulation for synaptic plasticity, we apply it to three well-known supervised learning tasks involving temporal processing: pattern generation, a delayed match to sample task, and evidence accumulation. We use two main cell classes, inhibitory (I) and excitatory (E) cell type, and obey experimentally observed constraints: cells have synapses that are sign constrained with 80% of the population being E type and the rest I. Following the RSNN implementation in [23], we further endow a fraction of the E cells with threshold adaptation. This setup mimics the hierarchical structure of cell types that has been established empirically [6] through its simple example of two main cell types (E and I), one of which has two subtypes (E cells with and without threshold adaptation). Also, refractoriness and synaptic delay are incorporated into all cells' dynamics as in [23]. Moreover, overall connection probability in the RSNN is also constrained, reflecting the sparse connectivity in neuronal circuits [42, 43]. In the main text, all simulated tasks are constrained at 10% sparsity, and the sparsity parameter is varied in Figure S2 in the supplementary materials. This connection sparsity is maintained by fixing inactive synapses with 0 weights. Unlike the stochastic rewiring (Deep R) algorithm [44, 45], our implementation does not allow for rapid, random formation of new synapses after each experience.

To study the impact of our learning rule on network performance and dissect the effects of its different components, we train RSNNs using five different approaches for each task. These, illustrated in 3, are as follows: (i) BPTT, which updates weights using exact gradients shown in Figure 3ai; (ii) E-prop [23], the state-of-the-art method for biologically plausible training of RSNNs, shown Figure 3aii; (iii) TRTRL, the truncated RTRL given in (7) without the cell-type approximation, shown in Figure 3aiii; (iv) MDGL, which incorporates the cell type approximation given in (11) and (12) using only two cell types, shown in Figure 3aiv; (v) NL-MDGL, a nonlocal version of MDGL, where the gain is replaced by $w_{\alpha\beta} =< w_{jp} >_{j\in\alpha, p\in\beta}$ even for $w_{jp} = 0$ so that the modulatory signal diffuses to all cells in the network, shown in Figure 3av. We note that factor $\frac{\partial E}{\partial z_{j,t}}$, which depends on future errors as mentioned earlier, participates in the generation of all training results pertaining to MDGL in the main text (Figures 4–7); in supplementary materials, we derive an online approximation to MDGL and demonstrate (via simulation) that it does not lead to significant performance degradation (Figure S3, Section S3.2).

For each learning rule, we train the following network parameters: input, recurrent and output weights. All approaches update the output weights using backpropagation since the nonlocality problem in (5) applies only to the update of input and recurrent weights. (For updating the weights of a single output layer, random feedback alignment [46] has also been shown to be an effective and biologically plausible solution.)

Multidigraph learning in RSNNs improves efficiency in a pattern generation task

We first trained a RSNN to produce a one-dimensional target output, generated from the sum of five sinusoids, given a fixed Poisson input realization. The task is inspired by that used in [47]. We compare the training for five different learning rules illustrated in Fig. 3 and described above.

While the target output and the random Poisson input realization is fixed across training iterations, we change these along with the initial weight for different training runs, and illustrate the learning curve mean and standard deviation in Figure 4b across five such runs. Here, the learning curve for this task is the reduction of normalized mean square error (NMSE) between the actual and desired output over training iterations. We observe that our methods, TRTRL (blue) and MDGL (green), reduce NMSE faster over training iterations compared to e-prop (magenta). Also, removing the locality of the modulatory signal (NL-MDGL) degraded the efficiency, although learning with spatially non-specific modulation still outperformed that without this modulatory signal (e-prop).

We highlight the fact that approximating the nonlocal learning rule of TRTRL with diffuse, modulatory signaling among two cell types results in only a moderate degradation of performance, as demonstrated by the proximity of the learning curves for MDGL and TRTRL. To better understand why this is the case, we conduct an analysis of the similarity between the $\widehat{\Gamma_{pq}}$ term computed by TRTRL in (7) and its cell-type-based approximation Γ_{pq} computed by MDGL. We quantify this via an alignment angle, which describes similarity in the direction of two vectors (Formulae for alignment angle are provided in Section S3.4 of supplementary materials). We show in Supplementary Table S1 the alignment angle between TRTRL's $\widehat{\Gamma}$ term and MDGL's Γ. The significant alignment between these two signals, despite the underlying vectors lying in very high dimensional spaces of synaptic weights, suggests that the two methods computed similar gradients, thus shedding light on the similarity between their learning curves.

Figure 4c illustrates the input, recurrent network activity, and target versus actual outputs trained using the MDGL method after training for 10, 100 and 500 iterations. Recurrent unit spikes appear to be reasonably irregular over time, broadly consistent with what is typically observed biologically [48, 49], with no obvious patterns of system-wise synchrony by eye throughout training. As shown in the lower panels, the network output approaches

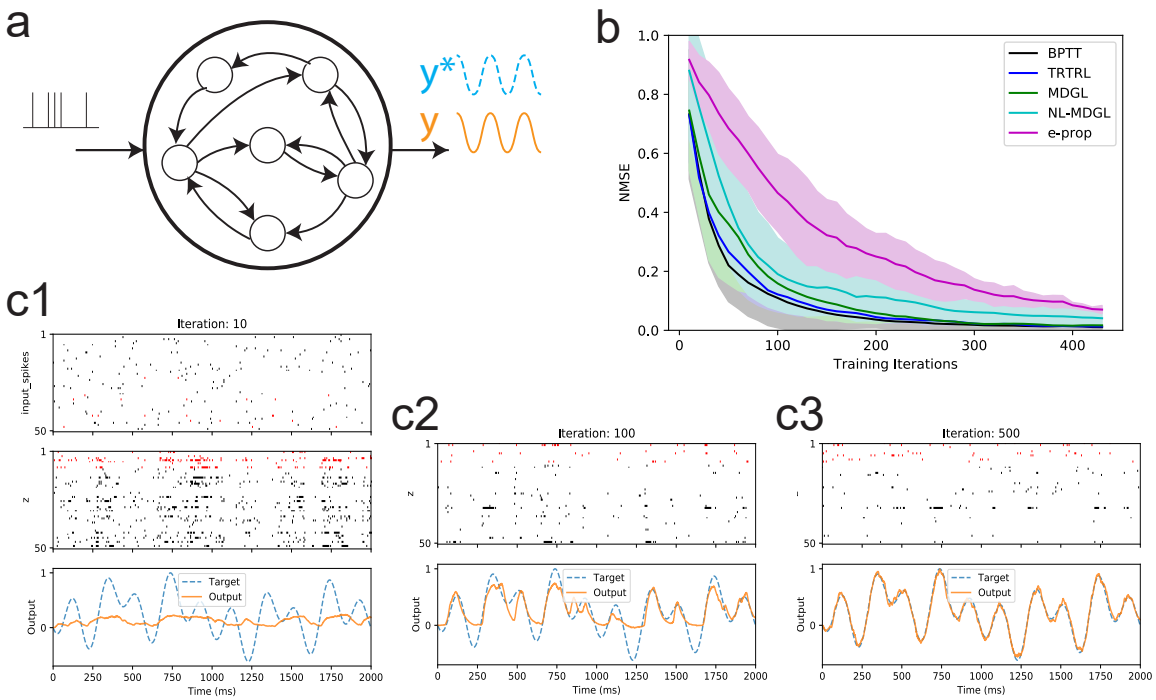

Figure 4: Pattern generation task a) Task setup [47]: a network is trained to produce a target output pattern over time. The target is formed from a sum of five sinusoids. b) Normalized mean squared error (NMSE) over training iterations is illustrated for the five learning rules (see Fig. 3). Solid lines show the mean and shaded regions show the standard deviation across five runs, with different target output, frozen Poisson input and weight initialization between runs (but fixed within each run). Comparing the performance of e-prop with the MDGL method suggests that the addition of cell-type-specific modulatory signals expedites the learning curve. c) Dynamics of the input, output and recurrent units are shown after 1, 100 and 500 iterations of training using the MDGL method. Raster plots are shown for 50 selected sample cells, and E cells and I cells are color coded using black and red, respectively. All recurrent units have fixed thresholds for this task. Recurrent unit spikes are irregular throughout training. Network output approaches the target as training progresses.

the target as training progresses.

Multidigraph learning in RSNNs improves efficiency in a delayed match-to-sample task

To shed light on how neural networks with the learning rules at hand can be trained to integrate a history of past cues to generate responses that impact a reward delivered later, we considered a special case of the delayed match to sample task described in [50]. Here, two cue alternatives are encoded by the presence and absence of input spikes. Our implementation of the task began with a brief fixation period (no cues) followed by two sequential cues, each lasting 0.15s and separated by a 0.75s delay (Figure 5a). A cue of value 1 was represented by 40Hz Poisson spiking input, whereas a cue of value 0 was represented by the absence of input spiking. The network was trained to output 1 (resp. 0) when the two cues have matching (resp. non-matching) values. That is, the RSNN was trained to remember the first cue and learn to compare it with the second cue delivered at a later time.

Figure 5b displays the learning curve for novel (test) inputs for the five plasticity rules described above. We observe that the same general conclusions as for the pattern generation task hold here. Specifically: TRTRL (blue) and MDGL (green) outperformed e-prop (magenta). The performance degraded when we removed the neighborhood specificity of the modulatory signal (NL-MDGL, cyan). Moreover, alignment angles between MDGL, TRTRL, and e-prop are also similar to those of the pattern generation task (Supplementary Tables S1 and S2). We illustrate the network dynamics after training using the MDGL method for 100, 500 and 2000 iterations in Figure 5c1–c3. We observe that as training progresses, the network output decides on the correct prediction with greater confidence, i.e. the output neuron corresponding to the correct target approaches a value of 1.

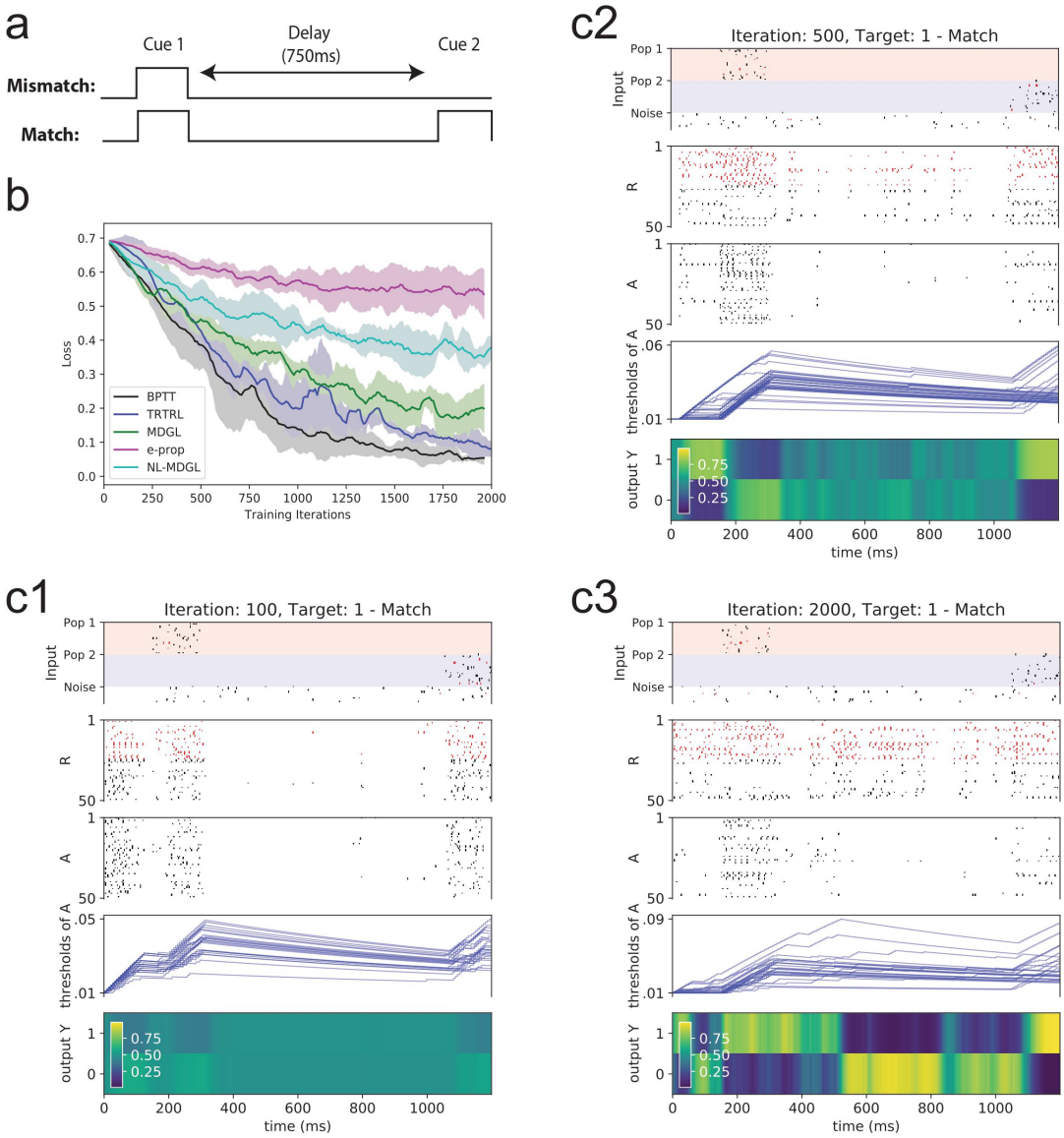

Figure 5: Application of the cell-type-specific modulatory signals to the delayed match-to-sample task. a) Setup of a special case of the delayed match to sample task, where two cue alternatives are represented by the presence and absence of input spikes. b) Learning curves of aforementioned training methods in Figure 3a. Loss is obtained from test data using different realizations of random Poisson input than in training data. Solid lines show the mean and shaded regions show the standard deviation across five runs, with a different weight initialization for each run. Comparing the performance of e-prop with the MDGL method suggests that the addition of cell-type-specific modulatory signals expedites the learning curve. c) Network dynamics of an example trial after 100, 500 and 2000 iterations of training using the MDGL method are illustrated in c1, c2 and c3, respectively, with the iteration count shown in plot titles. To emphasize the change in dynamics over training iterations, we used the same cue pattern for the illustrations. Again, E cells and I cells are color coded using black and red, respectively. For this task, both recurrent units with adaptive threshold (labeled as A) and without (labeled as R) are involved, following the RSNN formulation in [23]. Threshold dynamics of sample neurons are illustrated, and the adaptive thresholds were set to have a time scale similar to the required memory duration in order to facilitate the working memory task [45, 23]. The network makes the correct prediction with greater confidence as training progresses.

Multidigraph learning in RSNNs improves efficiency in an evidence accumulation task

Finally, we study an evidence accumulation task [52, 51], which involves integration of several cues in order to produce the desired output at a later time. Here, an agent moves along a straight path while encountering a series of sensory cues presented either on the right or left side of the track (Fig. 6a). Each cue is represented by 40 Hz Poisson spiking input for 100ms and cues were separated by 50ms. After a delay of 850ms when the agent reaches a

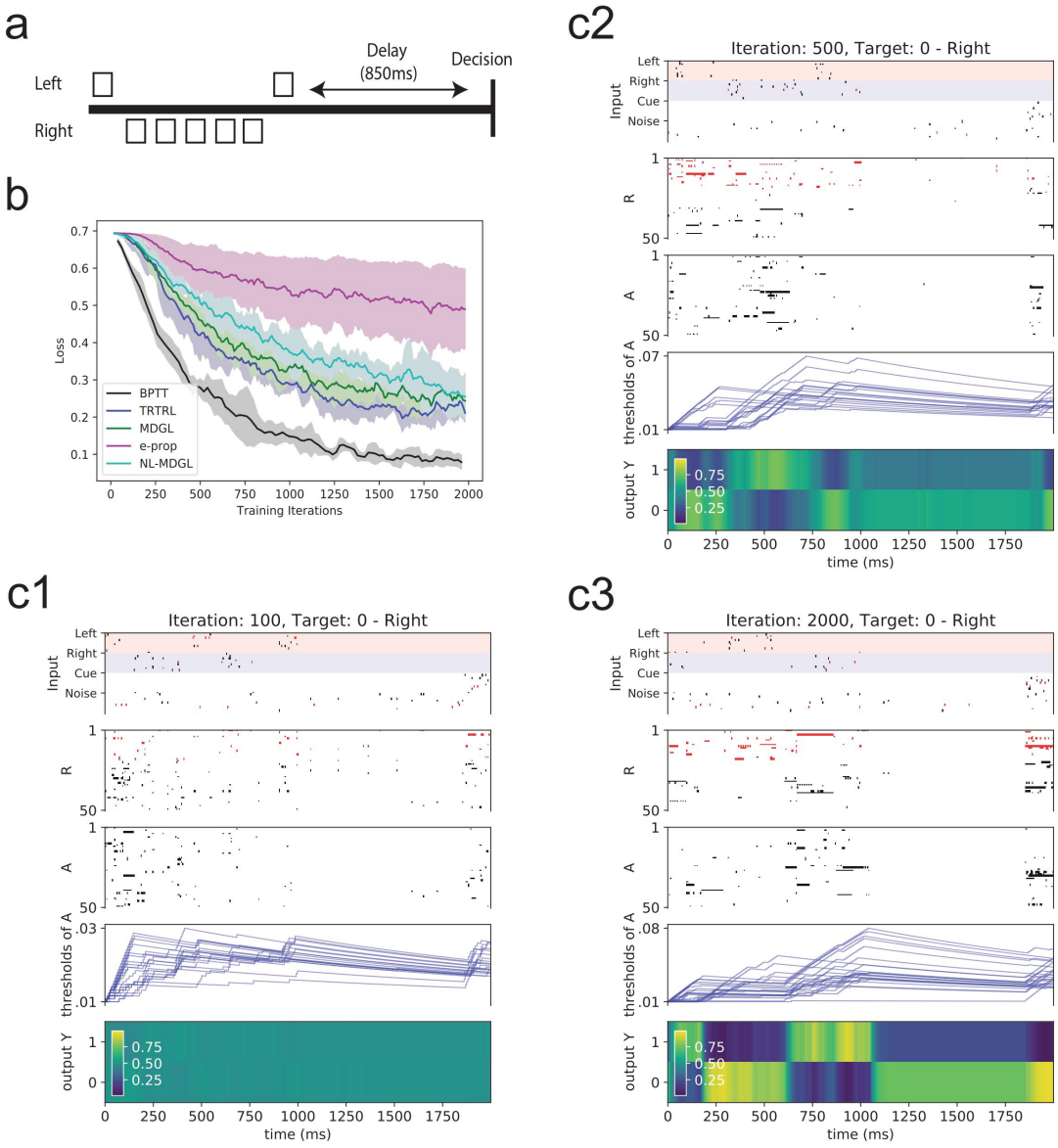

Figure 6: Application to the evidence accumulation task. a) Setup of the task inspired by [51, 52]. b) Learning curves of aforementioned training methods in Figure 3a. Loss is obtained from test data using different realizations of random Poisson input than in training data. Solid lines show the mean and shaded regions show the standard deviation across five runs, with a different weight initialization for each run. Comparing the performance of e-prop with the MDGL method suggests that the addition of cell-type-specific modulatory signals expedites the learning curve. c) Network dynamics (Input spikes, recurrent unit spikes and readout) of an example trial after 100, 500 and 2000 iterations of training using the MDGL method are illustrated in c1, c2 and c3, respectively, with iteration count shown in the plot titles. To emphasize the change in dynamics over training iterations, we used the same target direction for the illustrations. Similar to the previous task, both recurrent units with adaptive threshold (labeled as A) and without (labeled as R) are involved following [23], and threshold dynamics of sample neurons are illustrated. The network makes the correct prediction with greater confidence as training progresses. For all methods, results were obtained without using stochastic rewiring, which allows for random formation of new synapses in each experience (Deep R) [44, 45], as previously mentioned. This suggests that learning with local modulatory signals can relax the need for rapid and stochastic creation/pruning of synapses synapses after each iteration.

T-junction, it has to decide if more cues were received on the left or right. Thus, this task requires not only recalling past cues, but also being able to count the cues separately for each side and then process these cues for a reward delivered at a much later time. Our implementation is inspired by Bellec et al. [23], where the authors showed that e-prop can be used to train a RSNN to solve this task, although it required (i) significantly more training iterations than BPTT, and (ii) rapid and stochastic creation/pruning of synapses after each iteration [44]. Therefore, we test our learning rule to see if the addition of diffuse modulatory signals can indeed bring the learning curve closer to

BPTT, without relying on stochastic rewiring.

Figure 6b, and Supplementary Tables S1 and S2 demonstrate that all of our conclusions in the previous two experiments continue to hold in this task: MDGL is closest to BPTT, and NL-MDGL gives relatively degraded performance yet still outperforms e-prop. We also observe that the timing of recurrent unit spiking patterns tightly follows that of input cue presentation, suggesting that the network is taking immediate action ("counting") for each cue. We illustrate the network dynamics after training using the MDGL method for 100, 500 and 2000 iterations in Figure 6c1–c3. We observe that as training progresses, the network output decides on the correct prediction with greater confidence.

Multidigraph learning in RSNNs produce fast synaptic signaling and slow modulatory signaling

Modulatory signaling channels, owing to their diffusive nature and the fact that GPCRs act on significantly longer time scales than receptors for many synaptic neurotransmitters [30], is limited to be slower and smoother (i.e., have lower bandwidth) than direct synaptic channels. Since our model does not explicitly limit the communication bandwidths of either of these channels, comparing the frequency content of these two channels offers an important check for biological plausibility. It also provides a test of our assumption that the summation over j in Eq. (7) acts as a smoothing operation, enabling subsequent approximation with diffuse modulatory signaling. Figure 7 shows the results. The modulatory input indeed has significantly lower frequency content than the synaptic input, for all three tasks studied above.

In sum, our results on the slow timescales of cell-type averaged signaling – together with the gradient alignment results presented above – help to illustrate why modulatory signaling that is non-specific across both space and time can nonetheless lend significant improvement to network learning curves. This form of signaling removes the need for specifically tuned, reciprocal physical contacts and signaling among cell pairs and hence biologically implausible features of anatomical organization that plague solutions to the credit assignment problem via pairwise synaptic communication alone.

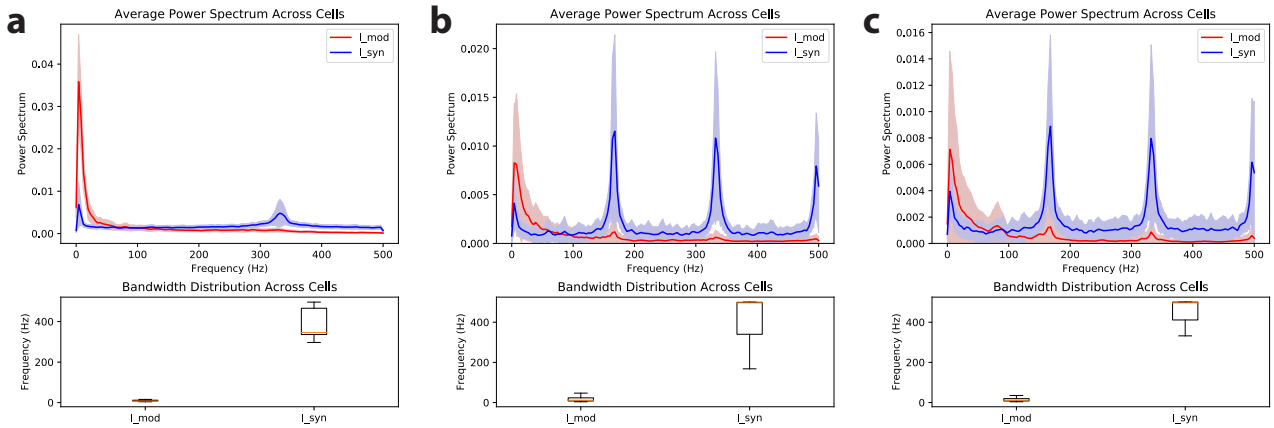

Figure 7: Frequency distribution of modulatory versus synaptic input. Comparing the power spectrum and bandwidth distributions between modulatory and synaptic input for a) pattern generation, b) delayed match to sample and c) evidence accumulation tasks. In the top panel, the solid lines denote the average and shaded regions show the standard deviation of power spectrum across recurrent cells. In the bottom panel, box plots show the minimum, lower quartile, median, upper quartile and maximum bandwidth across the cells. Here, the bandwidth is quantified by 3dB frequency, where the power halves compare to the center frequency power. Nyquist's theorem dictates that the simulation interval of 1ms limits the maximum frequency to 500Hz. Modulatory input is the total cell-type-specific modulatory signals detected by each cell p, defined as $I_{mod,p} := \sum_{\alpha \in C} w_{\alpha\beta} \sum_{j \in \alpha, p \to j} a_{j,t}$ (see (12)). Synaptic input is the total input received through synaptic connections by each cell j, defined as $I_{syn,j} := \sum_{l \neq j} w_{jl}\, z_{l,t} + \sum_p w^{\text{IN}}_{jm}\, x_{m,t+1}$ (see (1)). We remind the reader that secretion by cell j, $a_{j,t}$, consists of factor $\frac{\partial E}{\partial z_{j,t}}$ that depends on future errors and cannot be implemented online (Section S3.2 in supplementary materials). Thus, we repeated the spectral analysis for the online implementation of modulatory signaling in (S4) and observed similar conclusions in; these are shown in Figure S4 in supplementary materials. Here, the plots are obtained at the end of training (after 500 iterations for pattern generation and 2000 iterations for delayed match to sample and evidence accumulation tasks), but similar trends are observed at other training snapshots. This comparison suggests that modulatory signaling is significantly slower than synaptic transmission.

Discussion

In this paper, we presented a normative theory of temporal credit assignment and an associated biologically plausible learning rule where neurons are allowed to communicate via not only the synaptic connections but also a secondary, non-specific "modulatory" channel. In particular, we explain how the recent observation of wide-spread and cell-type-specific modulatory signaling [30], when integrated with synaptic networks to interconnect cortical neurons, can promote efficient learning. We demonstrate that the associated biologically plausible learning rule achieves performance close to that of ideal, but biologically unrealistic, rules on multiple *in silico* learning tasks – pattern generation, delayed match-to-sample, and evidence accumulation – that require temporal credit assignment over timescales spanning seconds. These experiments used sparsely and recurrently connected spiking neural networks of sign-constrained spiking neurons, capturing some of the well-studied aspects of brain architecture and computation [42, 43]. While our demonstrations use the supervised learning paradigm, the same machinery can be easily applied to reinforcement learning tasks following the approach in Bellec *et al* [23].

The existence of multiple directed connections between neurons suggests a *multidigraph* view of brain connectivity, in which each neuron is connected by multiple different connection types (seen in Figures 1 and 3b) [30]. Therefore, we call our learning paradigm for RSNNs, multidigraph learning (MDGL). MDGL begins with the same basic structure as the e-prop [23] and RFLO [32] learning rules, but then adds the local modulatory network to achieve a better approximation of the ideal synaptic update weights. Importantly, while we did not constrain the temporal dynamics of this secondary signaling mechanism, we found that this cell-type-specific broadcast channel communicated at significantly lower frequencies in all our experiments. This is in agreement with well-known properties of diffusive modulatory signaling.

More generally, our work fits under the wide umbrella of neoHebbian multi-factor learning theory (see comprehensive reviews in [13] [22]). This theory posits additional modulatory factors, which combine with factors based on pre- and postsynaptic activity to determine synaptic plasticity. These additional factors could arise from a diverse set of modulatory signals found in the brain. Most existing theoretical models conceive these factors as coming from distant top-down learning signals such as dopamine, noradrenaline and many others. Here, we also include top-down learning signals – directly based on the errors E in network outputs – but add further modulatory factors, released form within the recurrent network performing the computation at hand. These signals represent locally secreted modulatory signals, such as those carried by neuropeptides, released from nearby cells of a given type. Such signalling is abundantly present in the brain but had yet to be interpreted in terms of the credit assignment problem.

Many learning rules seek for some form of improvement that can be quantified by a certain loss function, and the direction of the steepest descent in the loss function can be identified from the exact gradient. Learning rules that follow this exact gradient, RTRL and BPTT, are well established, but are not biologically plausible and have unmanageably vast memory storage demands. However, a growing body of studies have demonstrated that learning rules that only partially follow the gradient, while alleviating some of these problems of the exact rules, can still lead to desirable outcomes [53] [54]. An example is the seminal paper [46] which introduces the concept of feedback alignment, which rivals backpropagation on a variety of tasks even using random feedback weights for credit assignment. In addition, approximations to RTRL have been proposed [55] [56] [57] for efficient online learning in RNNs. Roth *et al.* [58] trains rate-based neurons by applying a 2D sensitivity matrix to track the dependency of cell j with $\frac{ds_{p,t}}{dw_{pq}}$ so as to only maintain the memory of a rank-2 tensor for every $\{p, q\}$ rather than a rank-3 ("triple") tensor of $\{l, p, q\}$. The sensitivity matrix is similar to the inter-cellular communication gain $w_{\alpha\beta}$ in (12) but it is obtained through node perturbation training rather than simple averaging; it is also not cell-type and sparsity constrained. Reference [32] also derived a $O(N^2)$ rule that maintained only a rank-2 tensor for training rate-based RNNs; as mentioned, the method fully truncates the inter-cellular dependencies involving cells other than pre- and postsynaptic neurons. Instead of the full truncation, the algorithm SnAp-n [37, 38] stores Jacobian $\frac{\mathrm{d}\, s_{j,t}}{\mathrm{d}\, w_{pq}}$ for all j influenced by w_{pq} within n time steps. SnAp-1 is effectively the truncation in [32] and the performance of SnAp-n increases substantially with n, resonating with our improved performance when more terms are included than e-prop. However, SnAp-n needs to maintain a triple tensor for $n \geq 2$. Our approximation is similar to SnAp-2, but only needs to store double tensor $\frac{\mathrm{d}\, s_{p,t}}{\mathrm{d}\, w_{pq}}$ for every $\{p, q\}$. As discussed above, our fundamental contribution is to further constrain the inter-cellular signaling via diffuse pathways organized by cell types, a step toward greater biological plausibility.

A prominent component of our learning rule is the local intercellular signaling that could find its analog with the recent findings on the existence of slow peptidergic modulatory cortical networks [30]. Key features of our learning rule inspired by the transcriptomic data include the abundance and ubiquity of neuropeptide secretion and reception across mammalian cortical neurons, as well as the precursor and receptor-type specificity of neuropeptide signaling.

As in Eq. (12), every cell p receives molecules secreted by nearby cells through a multiplicative gain $w_{\alpha\beta}$ that could be mapped to the affinity of type β receptor to type α precursor. It is important to note that although our model is inspired by neuropeptide signaling, its biological underpinnings could involve additional cell-type-based local diffusive modulatory signals that have yet to be uncovered. This is because our model is based on abstract units that do not have the resolution to capture precise molecular-level dynamics.

This new local modulatory signaling is integrated with two classical types of biological signals: eligibility traces and top-down signals. Eligibility trace, which is local to a synapse and keeps a memory of past pre- and post-synaptic activity, tags a given synapse as eligible for modification. While its biological underpinning remains elusive, molecules that could function as eligibility traces include calcium ions and activated CaMKII enzymes [59]. In addition, there exists an abundance of top-down learning signals in the brain, such as dopamine, noradrenaline and neural firing [60], that have been shown to modulate learning rules (e.g. STDP function) [13]. Following [23], we took top-down learning signals to be cell-specific (j) rather than global. This is justified in part by recent reports that dopamine signals [51] and error-related neural firing [60] can be specific for a population of neurons. On the other hand, as explained in Section S3.3 in supplementary materials, this learning signal cell-specificity could be loosened by approximating the feedback weights with random weights [46] or via a similar approach as the cell-type-specific weight approximation in Eq. (8). The concept of plasticity induced by combining top-down signals and eligibility traces [22] underpins many existing multi-factor learning rules, notably e-prop. Similar to dopaminergic regulation of plasticity [61], neuropeptides have also been shown to modulate plateau potential, which could alter the amount of plasticity [62]. Therefore, in our learning rule, eligibility traces are compounded with local cell-type-specific signals in addition to the classical top-down signals, thereby generalizing multi-factor learning rules. Our model also posits that top-down learning signals can influence the secretion of cell-type-based modulatory signals. Indeed, dopamine can alter neuronal firing [61], which can then impact neuropeptide release in an activity-dependent manner [28].

As a proof of concept, we tested a minimal implementation of MDGL: there are just two local modulatory types mapping to the two main cortical cell classes (E and I), and the fine-grained E cell subtypes (those with and without firing threshold adaptation) were grouped into one modulatory type. Even the minimal implementation led to a marked improvement in learning efficiency compare to existing biological learning rules. However, brain cells are extremely diverse [63, 64, 65, 66, 67, 68] with a matching diversity in the expression of peptidergic genes; at least 18 NP precursor and 29 NP-GPCR genes were reported to have widespread and cell-type-specific expression in the mouse cortex [29]. Therefore, further studies of the interplay of task complexity, learning, and the diversity of cell types may uncover fundamental insights about neural circuit organization and computing. Also, we considered weight averages within a type as a simple implementation of cell-type-specific (but not neuron specific) receptor efficacy. This inevitably assumes that modulatory receptor affinities co-evolve with synaptic weights during learning. Figure S5 in the supplementary materials shows that these signaling gain values indeed drift over training, but this drift is slow relative to the change in individual synaptic weights. This opens the door of investigating mechanisms by which a link between synaptic transmission and modulatory signaling could, plausibly, be maintained across learning in biological circuits, such as the involvement of calcium dynamics in both GPCR signaling and synaptic transmission. Future work could investigate the existence and extent of this co-adaptation, and the possibility that it can be relaxed, the latter again drawing inspiration from ideas of feedback alignment [46]. A related point is that, here, we distinguished local modulatory types mainly based on precursor-receptor affinities and omitted the potential diversity of time scales across different types of NP signals [28]. Future work may involve investigating the role of distinct timescales in local modulatory signaling. Similarly, here, we followed the setup in [23] for implementing threshold adaptivity, but have not investigated the diversity of adaptivity. Future studies of how the distribution of threshold adaptivity affects learning and computation, possibly together with cell-type specific organization, are another intriguing frontier.

Our MDGL learning rule assumes that cell-type-based modulatory signals diffuse locally such that they are registered only by local synaptic partners of the source neuron. On the other hand, NL-MDGL assumes that these signals diffuse to all cells of a given type in the network. The biological reality may lie somewhere in between. Studying the extent of the local diffusion hypothesis, where neuropeptide signals act on both synaptic partners and nonsynaptic partners in the vicinity [28] – and how it relates to the two learning implementations here or new, intermediate implementations – will suggest new learning rules in the future and deepen our understanding of their underlying mechanisms.

Our results on how diffuse modulatory signaling can accelerate learning advances our understanding of biological networks, but also may play a role in bio-inspired, engineered neural networks. From a machine learning perspective, our model further advances the efficacy of approximated gradient-based learning methods and continues the line of research in energy-efficient on-chip learning through spike-based communications [40, 39]. Such efficient

approximations of the gradient computation can be especially important as artificial networks become ever larger and are used to tackle ever more complex tasks under both time and energy efficiency constraints.

Methods

An overview of our network model and the mathematical basis of our learning rule is given in the beginning of Results. In this section, we focus on implementation details.

Network Model: We consider the discrete-time implementation of RSNNs previously described in [23], which offers a general formalism applicable to many recurrent neural network models. The network, as shown in Figure 2a, denotes the observable states, i.e. spikes, as z_t at time t, and the corresponding hidden states as s_t. For LIF cells, the state s_t corresponds to membrane potential and the dynamics of those states are provided in (1).

Following references [45, 23], which implemented adaptive threshold LIF (ALIF) units [36] and observed that this neuron model improves computing capabilities of RSNNs relative to networks with LIF neurons, we also include ALIF cells in our model. In addition to the membrane potential, ALIF cells have a second hidden variable, a_t, governing the adaptive threshold. The spiking dynamics of both LIF and ALIF cells can be characterized by the following set of equations:

$$s_{j,t+1} = \eta\, s_{j,t} + (1-\eta)(\sum_{l\neq j} w_{jl}\, z_{l,t} + \sum_p w^{\mathrm{IN}}_{jm}\, x_{m,t+1}) - z_{j,t}\, v_{\mathrm{th}} \tag{15}$$

$$z_{j,t} = H(s_{j,t} - A_{j,t}) \tag{16}$$

$$A_{j,t} = v_{\mathrm{th}} + \beta a_{j,t} \tag{17}$$

$$a_{j,t} = \rho a_{j,t-1} + (1-\rho) z_{j,t-1}, \tag{18}$$

where the voltage dynamics in (15) is the same as (1). A spike is generated when the voltage $s_{j,t}$ exceeds the dynamic threshold $A_{j,t}$. Parameter β controls how much adaptation affects the threshold and state $a_{j,t}$ denotes the variable component of the dynamic threshold. The decay factor ρ is given by e^{-dt/τ_a} for simulation time step dt and adaptation time constant τ_a, which is typically chosen on the behavioral task time scale. For regular LIF neurons without adaptive threshold, one can simply set $\beta = 0$.

Differentiation in RSNNs: Gradient descent is problematic for spiking neurons due to the discontinuous step function H in (15), whose derivative is not defined at 0 and is 0 everywhere else. We overcome this issue by approximating the decay of the derivative using a piece-wise linear function [45, 40, 41]. Here, the pseudoderivative $h_{j,t}$ is defined as follows:

$$h_{j,t} = \frac{\mathrm{d}\, z_{j,t}}{\mathrm{d}\, s_{j,t}} \tag{19}$$

$$\approx \gamma \max\left(0, 1 - \left|\frac{s_{j,t} - A_{j,t}}{v_{\mathrm{th}}}\right|\right), \tag{20}$$

The dampening factor γ (typically set to 0.3) dampens the increase of backpropagated errors in order to improve the stability of training very deep (unrolled) RSNNs [45]. Throughout this study, refractoriness is implemented as in [23], where $h_{j,t}$ and $z_{j,t}$ are fixed at 0 after each spike of neuron j for 2 to 5ms.

Network output and loss function: Dynamics of leaky, graded readout neurons was implemented as $y_{k,t} = \kappa\, y_{k,t-1} + (1-\kappa) \sum_j w^{\mathrm{OUT}}_{kj} z_{j,t} + b^{\mathrm{OUT}}_k$ [23]. The $(1-\kappa)$ factor in the second term was dropped in writing for readability but kept during the actual implementation. Here, $\kappa \in (0,1)$ defines the leak and $\kappa = e^{-dt/\tau_{OUT}}$ for output membrane time constant τ_{OUT}. We provide the online implementation for this readout convention in Supplementary Section S3.2.

We quantify how well the network output matches the desired target using error function E. For regression tasks such as pattern generation, we use $E = \sum_{k,t}(y^*_{k,t} - y_{k,t})^2$ given time-dependent target $y^*_{k,t}$. For classification tasks such as delayed match to sample and evidence accumulation, $E = -\sum_{k,t} \pi^*_{k,t} log \pi_{k,t}$ with one-hot encoded target $\pi^*_{k,t}$ and predicted category probability $\pi_{k,t} = \mathrm{softmax}_k(y_{1,t}, \ldots, y_{N_{OUT},t}) = \exp(y_{k,t}) / \sum_{k'} \exp(y_{k',t})$. We provide all simulation and training parameters in Section S3.4.

Notation for Derivatives: There are two types of computational dependencies in RSNNs: direct and indirect dependencies. For example, variable w_{pq} can impact state $s_{p,t}$ directly through (1) as well as indirectly via its influence through other cells in the network. Following the convention in [23], we distinguish direct dependencies versus all dependencies (including indirect ones) using partial derivatives (∂) versus total derivatives (d).

Eligibility Trace Implementation: As introduced in Eq. (12), eligibility trace is defined as [23]:

$$e_{pq,t} := \frac{\partial z_{p,t}}{\partial s_{p,t}} \frac{\mathrm{d}\, s_{p,t-1}}{\mathrm{d}\, w_{pq}}, \tag{21}$$

$$\frac{\mathrm{d}\, s_{p,t}}{\mathrm{d}\, w_{pq}} = \frac{\partial s_{p,t}}{\partial w_{pq}} + \frac{\partial s_{p,t}}{\partial s_{p,t-1}} \frac{\mathrm{d}\, s_{p,t-1}}{\mathrm{d}\, w_{pq}}, \tag{22}$$

where Eq. (22) follows directly from Eq.(6). $\frac{\mathrm{d}\, s_{p,t}}{\mathrm{d}\, w_{pq}}$ can be obtained recursively and is referred to as the eligibility vector [23]. $e_{pq,t}$ keeps a fading memory of activity pertaining to presynaptic cell q and postsynaptic cell p. A comprehensive discussion of interpreting eligibility traces as derivatives can be found in [23]. Here, we briefly explain its implementation by expanding the factors in Eqs. (21) and (22) for both LIF and ALIF cells.

For LIF cells, there is no adaptive threshold so the hidden state consists only of the membrane potential. Thus, we have factors $\frac{\partial z_{p,t}}{\partial s_{p,t}} = h_{j,t}$ with pseudo-derivative $h_{j,t}$ defined in (19), $\frac{\partial s_{p,t}}{\partial w_{pq}} = z_{q,t-1}$ and $\frac{\partial s_{p,t+1}}{\partial s_{p,t}} = \eta - v_{\mathrm{th}} h_{j,t}$ following (1).

For ALIF cells, there are two hidden variables so the eligibility vector is now a two dimensional vector $\frac{\mathrm{d}\, s_{p,t}}{\mathrm{d}\, w_{pq}} = [\frac{\mathrm{d}\, s^v_{p,t}}{\mathrm{d}\, w_{pq}}, \frac{\mathrm{d}\, s^a_{p,t}}{\mathrm{d}\, w_{pq}}] \in \mathbb{R}^{2x1}$ pertaining to membrane potential $v_{p,t}$ and adaptive threshold state $a_{p,t}$. Following (15), one can obtain factors $\frac{\partial z_{p,t}}{\partial s_{p,t}} = [\frac{\partial z_{p,t}}{\partial v_{p,t}}, \frac{\partial z_{p,t}}{\partial a_{p,t}}] = [h_{j,t}, -\beta h_{j,t}] \in \mathbb{R}^{1x2}$, $\frac{\partial s_{p,t}}{\partial w_{pq}} = [z_{q,t-1}, 0] \in \mathbb{R}^{2x1}$ and $\frac{\partial s_{p,t}}{\partial s_{p,t-1}}$ is now a 2-by-2 matrix:

$$\frac{\partial s_{p,t}}{\partial s_{p,t-1}} = \begin{bmatrix} \frac{\partial v_{p,t}}{\partial v_{p,t-1}} & \frac{\partial v_{p,t}}{\partial a_{p,t-1}} \\ \frac{\partial a_{p,t}}{\partial v_{p,t-1}} & \frac{\partial a_{p,t}}{\partial a_{p,t-1}} \end{bmatrix} = \begin{bmatrix} \eta - v_{\mathrm{th}} h_{j,t} & v_{\mathrm{th}} \beta h_{j,t} \\ (1-\rho) h_{j,t} & \rho - (1-\rho)\beta h_{j,t} \end{bmatrix} \in \mathbb{R}^{2x2}. \tag{23}$$

Thus, the eligibility trace $e_{pq,t}$ would be scalar valued regardless of the dimension of the eligibility vector.

Acknowledgements

We wish to thank the Allen Institute for Brain Science founder, Paul G Allen, for his vision, encouragement and support. Helena Liu is supported by Natural the Science and Engineering Research Council (NSERC) Postgraduate Scholarships - Doctoral (NSERC PGS-D) program. This work was facilitated through the use of advanced computational, storage, and networking infrastructure provided by the Hyak supercomputer system at the University of Washington.

References

[1] Pieter R. Roelfsema and Anthony Holtmaat. "Control of synaptic plasticity in deep cortical networks". In: *Nature Reviews Neuroscience* 19.3 (Feb. 2018), pp. 166–180. ISSN: 14710048. DOI: 10.1038/nrn.2018.6.

[2] Nan Rosemary Ke, Anirudh Goyal ALIAS PARTH GOYAL, Olexa Bilaniuk, Jonathan Binas, Michael C Mozer, Chris Pal, and Yoshua Bengio. "Sparse attentive backtracking: Temporal credit assignment through reminding". In: *Advances in neural information processing systems*. 2018, pp. 7640–7651.

[3] Blake A. Richards and Timothy P. Lillicrap. "Dendritic solutions to the credit assignment problem". In: *Current Opinion in Neurobiology* 54 (Feb. 2019), pp. 28–36. ISSN: 18736882. DOI: 10.1016/j.conb.2018.08.003.

[4] Yann Lecun, Yoshua Bengio, and Geoffrey Hinton. "Deep learning". In: *Nature* 521.7553 (May 2015), pp. 436–444. ISSN: 14764687. DOI: 10.1038/nature14539.

[5] Ronald J. Williams and David Zipser. "A Learning Algorithm for Continually Running Fully Recurrent Neural Networks". In: *Neural Computation* 1.2 (June 1989), pp. 270–280. ISSN: 0899-7667. DOI: 10.1162/neco.1989.1.2.270.

[6] Bosiljka Tasic, Zizhen Yao, Lucas T. Graybuck, Kimberly A. Smith, Thuc Nghi Nguyen, Darren Bertagnolli, Jeff Goldy, Emma Garren, Michael N. Economo, Sarada Viswanathan, Osnat Penn, Trygve Bakken, Vilas Menon, Jeremy Miller, Olivia Fong, Karla E. Hirokawa, Kanan Lathia, Christine Rimorin, Michael Tieu, Rachael Larsen, Tamara Casper, Eliza Barkan, Matthew Kroll, Sheana Parry, Nadiya V. Shapovalova, Daniel Hirschstein, Julie Pendergraft, Heather A. Sullivan, Tae Kyung Kim, Aaron Szafer, Nick Dee, Peter Groblewski, Ian Wickersham, Ali Cetin, Julie A. Harris, Boaz P. Levi, Susan M. Sunkin, Linda Madisen, Tanya L. Daigle, Loren Looger, Amy Bernard, John Phillips, Ed Lein, Michael Hawrylycz, Karel Svoboda, Allan R. Jones, Christof Koch, and Hongkui Zeng. "Shared and distinct transcriptomic cell types across neocortical areas". In: *Nature* 563.7729 (Nov. 2018), pp. 72–78. ISSN: 14764687. DOI: 10.1038/s41586-018-0654-5.

[7] Nathan W Gouwens, Staci A Sorensen, Jim Berg, Changkyu Lee, Tim Jarsky, Jonathan Ting, Susan M Sunkin, David Feng, Costas A Anastassiou, Eliza Barkan, et al. "Classification of electrophysiological and morphological neuron types in the mouse visual cortex". In: *Nature neuroscience* 22.7 (2019), pp. 1182–1195.

[8] Ken Sugino, Erin Clark, Anton Schulmann, Yasuyuki Shima, Lihua Wang, David L Hunt, Bryan M Hooks, Dimitri Tränkner, Jayaram Chandrashekar, Serge Picard, et al. "Mapping the transcriptional diversity of genetically and anatomically defined cell populations in the mouse brain". In: *Elife* 8 (2019), e38619.

[9] Maria Antonietta Tosches and Gilles Laurent. "Evolution of neuronal identity in the cerebral cortex". In: *Current opinion in neurobiology* 56 (2019), pp. 199–208.

[10] Amit Zeisel, Hannah Hochgerner, Peter Lönnerberg, Anna Johnsson, Fatima Memic, Job Van Der Zwan, Martin Häring, Emelie Braun, Lars E Borm, Gioele La Manno, et al. "Molecular architecture of the mouse nervous system". In: *Cell* 174.4 (2018), pp. 999–1014.

[11] Hannah Bos, Anne-Marie Oswald, and Brent Doiron. "Untangling stability and gain modulation in cortical circuits with multiple interneuron classes". In: *bioRxiv* (2020).

[12] Verena Pawlak, Jeffery R Wickens, Alfredo Kirkwood, and Jason ND Kerr. "Timing is not everything: neuromodulation opens the STDP gate". In: *Frontiers in synaptic neuroscience* 2 (2010), p. 146.

[13] Jeffrey C. Magee and Christine Grienberger. "Synaptic Plasticity Forms and Functions". In: *Annual Review of Neuroscience* 43.1 (July 2020), pp. 95–117. ISSN: 0147-006X. DOI: 10.1146/annurev-neuro-090919-022842.

[14] Timothy P Lillicrap, Adam Santoro, Luke Marris, Colin J Akerman, and Geoffrey Hinton. "Backpropagation and the brain". In: *Nature Reviews Neuroscience* (2020), pp. 1–12.

[15] Zuzanna Brzosko, Susanna B Mierau, and Ole Paulsen. "Neuromodulation of Spike-Timing-Dependent plasticity: past, present, and future". In: *Neuron* 103.4 (2019), pp. 563–581.

[16] Marco P Lehmann, He A Xu, Vasiliki Liakoni, Michael H Herzog, Wulfram Gerstner, and Kerstin Preuschoff. "One-shot learning and behavioral eligibility traces in sequential decision making". In: *Elife* 8 (2019), e47463.

[17] Aparna Suvrathan. "Beyond STDP — towards diverse and functionally relevant plasticity rules". In: *Current Opinion in Neurobiology* 54 (Feb. 2019), pp. 12–19. ISSN: 18736882. DOI: 10.1016/j.conb.2018.06.011.

[18] Nicolas Frémaux and Wulfram Gerstner. "Neuromodulated spike-timing-dependent plasticity, and theory of three-factor learning rules". In: *Frontiers in Neural Circuits* 9.JAN2016 (Jan. 2015), p. 85. ISSN: 16625110. DOI: 10.3389/fncir.2015.00085.

[19] Michael A. Farries and Adrienne L. Fairhall. "Reinforcement Learning With Modulated Spike Timing–Dependent Synaptic Plasticity". In: *Journal of Neurophysiology* 98.6 (Dec. 2007), pp. 3648–3665. ISSN: 0022-3077. DOI: 10.1152/jn.00364.2007.

[20] Stijn Cassenaer and Gilles Laurent. "Conditional modulation of spike-timing-dependent plasticity for olfactory learning". In: *Nature* 482.7383 (Feb. 2012), pp. 47–51. ISSN: 14764687. DOI: 10.1038/nature10776.

[21] Sho Yagishita, Akiko Hayashi-Takagi, Graham C.R. Ellis-Davies, Hidetoshi Urakubo, Shin Ishii, and Haruo Kasai. "A critical time window for dopamine actions on the structural plasticity of dendritic spines". In: *Science* 345.6204 (Sept. 2014), pp. 1616–1620. ISSN: 10959203. DOI: 10.1126/science.1255514.

[22] Wulfram Gerstner, Marco Lehmann, Vasiliki Liakoni, Dane Corneil, and Johanni Brea. "Eligibility Traces and Plasticity on Behavioral Time Scales: Experimental Support of NeoHebbian Three-Factor Learning Rules". In: *Frontiers in Neural Circuits* 12 (July 2018), p. 53. ISSN: 16625110. DOI: 10.3389/fncir.2018.00053. arXiv: 1801.05219.

[23] Guillaume Bellec, Franz Scherr, Anand Subramoney, Elias Hajek, Darjan Salaj, Robert Legenstein, and Wolfgang Maass. "A solution to the learning dilemma for recurrent networks of spiking neurons". In: *Nature Communications* 11.1 (Dec. 2020), pp. 1–15. ISSN: 20411723. DOI: 10.1038/s41467-020-17236-y.

[24] Stephen D Meriney and Erika Fanselow. "Synaptic Transmission". In: Academic Press, 2019. Chap. Neuropeptide Transmitters, pp. 421–434.

[25] Éva Borbély, Bálint Scheich, and Zsuzsanna Helyes. "Neuropeptides in learning and memory". In: *Neuropeptides* 47.6 (Dec. 2013), pp. 439–450. ISSN: 01434179. DOI: 10.1016/j.npep.2013.10.012.

[26] C. R. Gøtzsche and D. P.D. Woldbye. "The role of NPY in learning and memory". In: *Neuropeptides* 55 (Feb. 2016), pp. 79–89. ISSN: 15322785. DOI: 10.1016/j.npep.2015.09.010.

[27] Sarah Melzer, Elena Newmark, Grace Or Mizuno, Minsuk Hyun, Adrienne C Philson, Eleonora Quiroli, Beatrice Righetti, Malika R Gregory, Kee Wui Huang, James Levasseur, et al. "Bombesin-like peptide recruits disinhibitory cortical circuits and enhances fear memories". In: *Available at SSRN 3724673* (2020).

[28] Anthony N. van den Pol. "Neuropeptide Transmission in Brain Circuits". In: *Neuron* 76.1 (Oct. 2012), pp. 98–115. ISSN: 08966273. DOI: 10.1016/j.neuron.2012.09.014.

[29] Stephen J. Smith, Uygar Sümbül, Lucas T. Graybuck, Forrest Collman, Sharmishtaa Seshamani, Rohan Gala, Olga Gliko, Leila Elabbady, Jeremy A. Miller, Trygve E. Bakken, Jean Rossier, Zizhen Yao, Ed Lein, Hongkui Zeng, Bosiljka Tasic, and Michael Hawrylycz. "Single-cell transcriptomic evidence for dense intracortical neuropeptide networks". In: *eLife* 8 (Nov. 2019). ISSN: 2050084X. DOI: 10.7554/eLife.47889.

[30] Stephen J. Smith, Michael Hawrylycz, Jean Rossier, and Uygar Sümbül. "New light on cortical neuropeptides and synaptic network plasticity". In: *Current Opinion in Neurobiology* 63 (Aug. 2020), pp. 176–188. ISSN: 18736882. DOI: 10.1016/j.conb.2020.04.002. arXiv: 2004.07975.

[31] Valerio Mante, David Sussillo, Krishna V. Shenoy, and William T. Newsome. "Context-dependent computation by recurrent dynamics in prefrontal cortex". In: *Nature* 503.7474 (Nov. 2013), pp. 78–84. ISSN: 00280836. DOI: 10.1038/nature12742.

[32] James M. Murray. "Local online learning in recurrent networks with random feedback". In: *eLife* 8 (May 2019). ISSN: 2050084X. DOI: 10.7554/eLife.43299.

[33] Martin Boerlin, Christian K. Machens, and Sophie Denève. "Predictive Coding of Dynamical Variables in Balanced Spiking Networks". In: *PLoS Computational Biology* 9.11 (Nov. 2013), p. 1003258. ISSN: 1553734X. DOI: 10.1371/journal.pcbi.1003258.

[34] Eric Hunsberger and Chris Eliasmith. "Spiking deep networks with LIF neurons". In: *arXiv preprint arXiv:1510.08829* (2015).

[35] Robert Kim, Yinghao Li, and Terrence J. Sejnowski. "Simple framework for constructing functional spiking recurrent neural networks". In: *Proceedings of the National Academy of Sciences of the United States of America* 116.45 (Nov. 2019), pp. 22811–22820. ISSN: 10916490. DOI: 10.1073/pnas.1905926116.

[36] Corinne Teeter, Ramakrishnan Iyer, Vilas Menon, Nathan Gouwens, David Feng, Jim Berg, Aaron Szafer, Nicholas Cain, Hongkui Zeng, Michael Hawrylycz, et al. "Generalized leaky integrate-and-fire models classify multiple neuron types". In: *Nature communications* 9.1 (2018), pp. 1–15.

[37] Jacob Menick, Erich Elsen, Utku Evci, Simon Osindero, Karen Simonyan, and Alex Graves. "A Practical Sparse Approximation for Real Time Recurrent Learning". In: *arXiv preprint arXiv:2006.07232* (2020).

[38] Friedemann Zenke and Emre O Neftci. "Brain-Inspired Learning on Neuromorphic Substrates". In: *arXiv preprint arXiv:2010.11931* (2020).

[39] Emre O. Neftci, Hesham Mostafa, and Friedemann Zenke. "Surrogate Gradient Learning in Spiking Neural Networks: Bringing the Power of Gradient-based optimization to spiking neural networks". In: *IEEE Signal Processing Magazine* 36.6 (Nov. 2019), pp. 51–63. ISSN: 15580792. DOI: 10.1109/MSP.2019.2931595.

[40] Dongsung Huh and Terrence J Sejnowski. "Gradient Descent for Spiking Neural Networks". In: *32nd Conference on Neural Information Processing Systems*. 2018, pp. 1433–1443.

[41] Steven K. Esser, Paul A. Merolla, John V. Arthur, Andrew S. Cassidy, Rathinakumar Appuswamy, Alexander Andreopoulos, David J. Berg, Jeffrey L. McKinstry, Timothy Melano, Davis R. Barch, Carmelo Di Nolfo, Pallab Datta, Arnon Amir, Brian Taba, Myron D. Flickner, and Dharmendra S. Modha. "Convolutional networks for fast, energy-efficient neuromorphic computing". In: *Proceedings of the National Academy of Sciences of the United States of America* 113.41 (Oct. 2016), pp. 11441–11446. ISSN: 10916490. DOI: 10.1073/pnas.1604850113. arXiv: 1603.08270.

[42] Valentino Braitenberg and Almut Schüz. *Cortex: statistics and geometry of neuronal connectivity*. Springer Science & Business Media, 2013.

[43] Stephanie C. Seeman, Luke Campagnola, Pasha A. Davoudian, Alex Hoggarth, Travis A. Hage, Alice Bosma-Moody, Christopher A. Baker, Jung Hoon Lee, Stefan Mihalas, Corinne Teeter, Andrew L. Ko, Jeffrey G. Ojemann, Ryder P. Gwinn, Daniel L. Silbergeld, Charles Cobbs, John Phillips, Ed Lein, Gabe Murphy, Christof Koch, Hongkui Zeng, and Tim Jarsky. "Sparse recurrent excitatory connectivity in the microcircuit of the adult mouse and human cortex". In: *eLife* 7 (Sept. 2018). ISSN: 2050084X. DOI: 10.7554/eLife.37349.

[44] Guillaume Bellec, David Kappel, Wolfgang Maass, and Robert Legenstein. "Deep rewiring: Training very sparse deep networks". In: *arXiv preprint arXiv:1711.05136* (2017).

[45] Guillaume Bellec, Darjan Salaj, Anand Subramoney, Robert Legenstein, and Wolfgang Maass. "Long short-term memory and learning-to-learn in networks of spiking neurons". In: *32nd Conference on Neural Information Processing Systems*. 2018, pp. 787–797.

[46] Timothy P Lillicrap, Daniel Cownden, Douglas B Tweed, and Colin J Akerman. "Random synaptic feedback weights support error backpropagation for deep learning". In: *Nature communications* 7.1 (2016), pp. 1–10.

[47] Wilten Nicola and Claudia Clopath. "Supervised learning in spiking neural networks with FORCE training". In: *Nature Communications* 8.1 (Dec. 2017), pp. 1–15. ISSN: 20411723. DOI: 10.1038/s41467-017-01827-3. arXiv: 1609.02545.

[48] Peter Dayan and Laurence F Abbott. *Theoretical neuroscience: computational and mathematical modeling of neural systems*. Computational Neuroscience Series, 2001.

[49] William R Softky and Christof Koch. "The highly irregular firing of cortical cells is inconsistent with temporal integration of random EPSPs". In: *Journal of Neuroscience* 13.1 (1993), pp. 334–350.

[50] Travis Meyer, Xue Lian Qi, Terrence R. Stanford, and Christos Constantinidis. "Stimulus selectivity in dorsal and ventral prefrontal cortex after training in working memory tasks". In: *Journal of Neuroscience* 31.17 (Apr. 2011), pp. 6266–6276. ISSN: 02706474. DOI: 10.1523/JNEUROSCI.6798-10.2011.

[51] Ben Engelhard, Joel Finkelstein, Julia Cox, Weston Fleming, Hee Jae Jang, Sharon Ornelas, Sue Ann Koay, Stephan Y. Thiberge, Nathaniel D. Daw, David W. Tank, and Ilana B. Witten. "Specialized coding of sensory, motor and cognitive variables in VTA dopamine neurons". In: *Nature* 570.7762 (June 2019), pp. 509–513. ISSN: 14764687. DOI: 10.1038/s41586-019-1261-9.

[52] Ari S. Morcos and Christopher D. Harvey. "History-dependent variability in population dynamics during evidence accumulation in cortex". In: *Nature Neuroscience* 19.12 (Dec. 2016), pp. 1672–1681. ISSN: 15461726. DOI: 10.1038/nn.4403.

[53] Blake A Richards, Timothy P Lillicrap, Philippe Beaudoin, Yoshua Bengio, Rafal Bogacz, Amelia Christensen, Claudia Clopath, Rui Ponte Costa, Archy de Berker, Surya Ganguli, et al. "A deep learning framework for neuroscience". In: *Nature neuroscience* 22.11 (2019), pp. 1761–1770.

[54] Drew Linsley, Alekh Karkada Ashok, Lakshmi Narasimhan Govindarajan, Rex Liu, and Thomas Serre. "Stable and expressive recurrent vision models". In: *arXiv preprint arXiv:2005.11362* (2020).

[55] Owen Marschall, Kyunghyun Cho, and Cristina Savin. "A unified framework of online learning algorithms for training recurrent neural networks". In: *Journal of Machine Learning Research* 21.135 (2020), pp. 1–34.

[56] Asier Mujika, Florian Meier, and Angelika Steger. "Approximating Real-Time Recurrent Learning with Random Kronecker Factors". In: *32nd Conference on Neural Information Processing Systems*. 2018, pp. 6594–6603.

[57] Corentin Tallec and Yann Ollivier. "Unbiased Online Recurrent Optimization". In: *ICLR*. Feb. 2018.

[58] Christopher Roth, Ingmar Kanitscheider, and Ila Fiete. "Kernel RNN Learning (KERNL)". In: *ICLR*. Sept. 2019.

[59] Magdalena Sanhueza and John Lisman. "The CaMKII/NMDAR complex as a molecular memory". In: *Molecular Brain* 6.1 (Feb. 2013), p. 10. ISSN: 17566606. DOI: 10.1186/1756-6606-6-10.

[60] Amirsaman Sajad, David C. Godlove, and Jeffrey D. Schall. "Cortical microcircuitry of performance monitoring". In: *Nature Neuroscience* 22.2 (Feb. 2019), pp. 265–274. ISSN: 15461726. DOI: 10.1038/s41593-018-0309-8.

[61] Nicolas X. Tritsch and Bernardo L. Sabatini. "Dopaminergic Modulation of Synaptic Transmission in Cortex and Striatum". In: *Neuron* 76.1 (Oct. 2012), pp. 33–50. ISSN: 08966273. DOI: 10.1016/j.neuron.2012.09.023.

[62] Trevor J. Hamilton, Sara Xapelli, Sheldon D. Michaelson, Matthew E. Larkum, and William F. Colmers. "Modulation of distal calcium electrogenesis by neuropeptide Y1 receptors inhibits neocortical long-term depression". In: *Journal of Neuroscience* 33.27 (July 2013), pp. 11184–11193. ISSN: 02706474. DOI: 10.1523/JNEUROSCI.5595-12.2013.

[63] Gord Fishell and Adam Kepecs. "Interneuron types as attractors and controllers". In: *Annual review of neuroscience* 43 (2019).

[64] Ozgun Gokce, Geoffrey M Stanley, Barbara Treutlein, Norma F Neff, J Gray Camp, Robert C Malenka, Patrick E Rothwell, Marc V Fuccillo, Thomas C Südhof, and Stephen R Quake. "Cellular taxonomy of the mouse striatum as revealed by single-cell RNA-seq". In: *Cell reports* 16.4 (2016), pp. 1126–1137.

[65] Rebecca D Hodge, Trygve E Bakken, Jeremy A Miller, Kimberly A Smith, Eliza R Barkan, Lucas T Graybuck, Jennie L Close, Brian Long, Nelson Johansen, Osnat Penn, et al. "Conserved cell types with divergent features in human versus mouse cortex". In: *Nature* 573.7772 (2019), pp. 61–68.

[66] Sinisa Hrvatin, Daniel R Hochbaum, M Aurel Nagy, Marcelo Cicconet, Keiramarie Robertson, Lucas Cheadle, Rapolas Zilionis, Alex Ratner, Rebeca Borges-Monroy, Allon M Klein, et al. "Single-cell analysis of experience-dependent transcriptomic states in the mouse visual cortex". In: *Nature neuroscience* 21.1 (2018), pp. 120–129.

[67] Z Josh Huang and Anirban Paul. "The diversity of GABAergic neurons and neural communication elements". In: *Nature Reviews Neuroscience* 20.9 (2019), pp. 563–572.

[68] Daniel J Miller, Aparna Bhaduri, Nenad Sestan, and Arnold Kriegstein. "Shared and derived features of cellular diversity in the human cerebral cortex". In: *Current opinion in neurobiology* 56 (2019), pp. 117–124.

[69] Nal Kalchbrenner, Erich Elsen, Karen Simonyan, Seb Noury, Norman Casagrande, Edward Lockhart, Florian Stimberg, Aaron van den Oord, Sander Dieleman, and Koray Kavukcuoglu. "Efficient neural audio synthesis". In: *arXiv preprint arXiv:1802.08435* (2018).

[70] Sharan Narang, Erich Elsen, Gregory Diamos, and Shubho Sengupta. "Exploring sparsity in recurrent neural networks". In: *arXiv preprint arXiv:1704.05119* (2017).

[71] Peter D. Welch. "The Use of Fast Fourier Transform for the Estimation of Power Spectra: A Method Based on Time Averaging Over Short, Modified Periodograms". In: *IEEE Transactions on Audio and Electroacoustics* 15.2 (1967), pp. 70–73. ISSN: 00189278. DOI: 10.1109/TAU.1967.1161901.

[72] Diederik P. Kingma and Jimmy Lei Ba. "Adam: A method for stochastic optimization". In: *ICLR*. International Conference on Learning Representations, ICLR, Dec. 2015. arXiv: 1412.6980.

Supplementary Materials

S1 Supplementary Figure

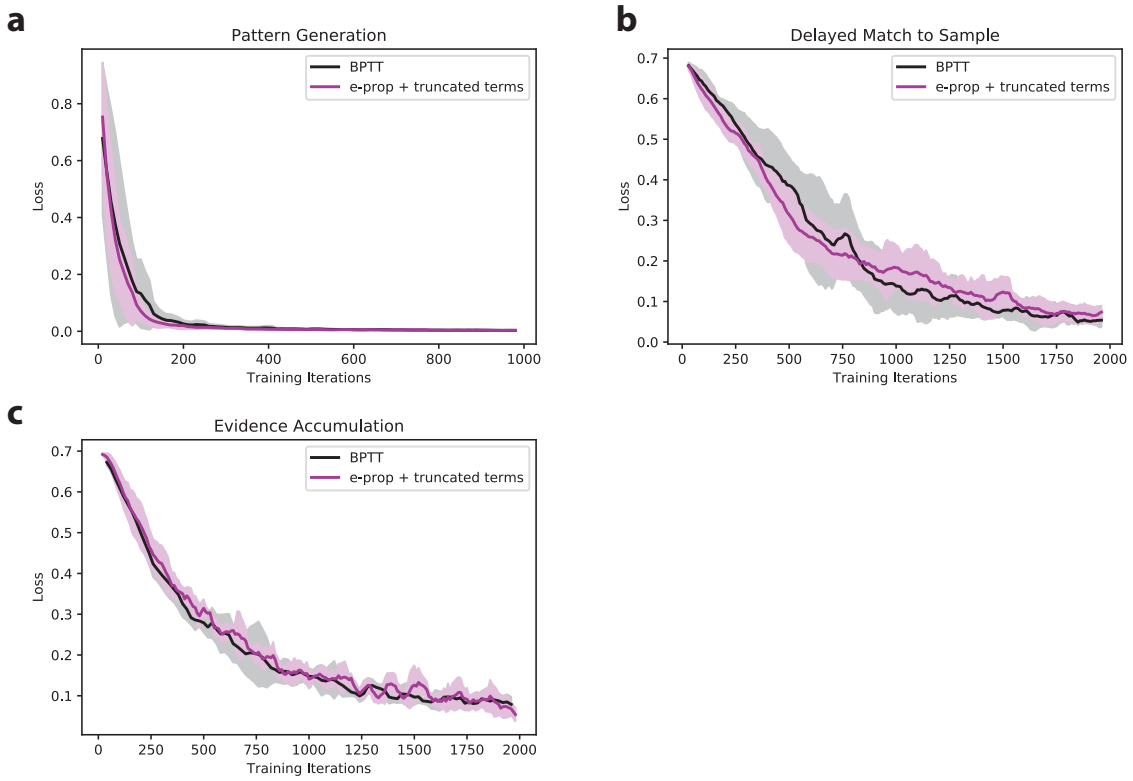

Figure S1: Checking e-prop implementation – recovering ignored terms recovers the performance of BPTT. As a sanity check, learning curves are plotted for e-prop plus all the truncated terms (see Eq. (5)) to verify that the resulting learning rule recovers the performance of BPTT. The check is applied to a) pattern generation, b) delayed match to sample and c) evidence accumulation tasks. Solid lines show the mean averaged across five runs and shaded regions show the standard deviation. For all tasks, the learning curves do not differ significantly, suggesting the e-prop implementation is accurate.

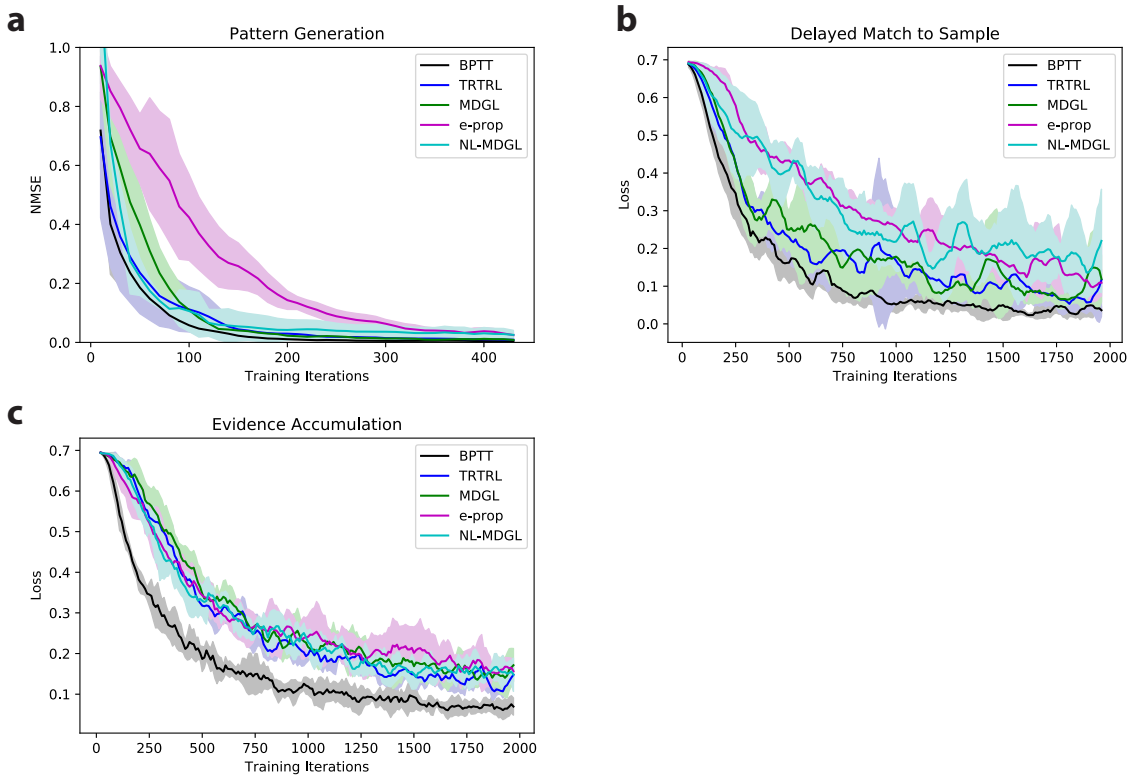

Figure S2: Learning performance of RSNNs with 30% **connection probability**. Learning curves for training methods in Figure 3a are illustrated for the following tasks: a) pattern generation, b) delayed match to sample and c) evidence accumulation tasks. The learning curve gap here between e-prop and MDGL is narrower than that of the network with 10% connection probability shown in the main text. This suggests that MDGL is more helpful under sparse scenarios. As noted in the main text, connection sparsity is widely observed in the brain [42] and has various computational advantages [69, 70].

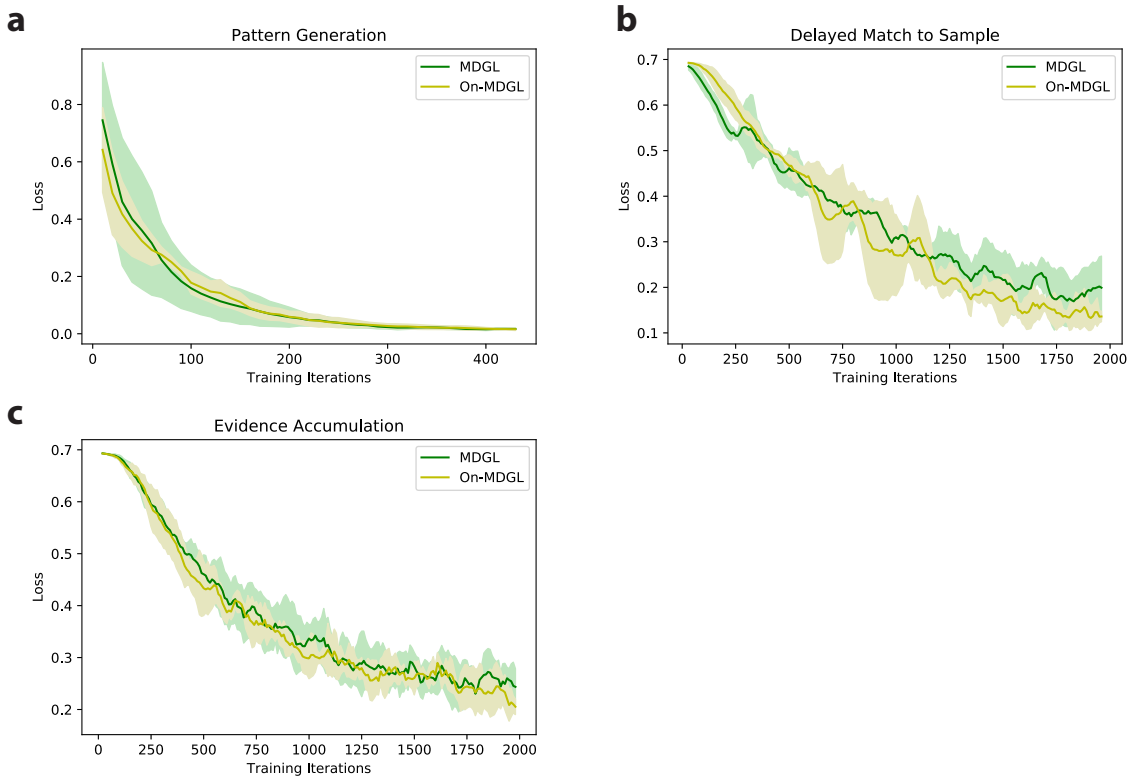

Figure S3: No significant degradation in performance observed for the online approximation of MDGL in Eq. (S4). For outputs with a leak term defined in Eq. (2), $\frac{\partial E}{\partial z_{j,t}}$ depends on future errors and an approximation is introduced in Eq. (S4) for online implementation of MDGL. To check if this approximation leads to significant degradation in performance, learning curves are plotted for a) pattern generation, b) delayed match to sample and c) evidence accumulation tasks. For all tasks, there is no significant deviation in learning curves between MDGL and the online approximation (On-MDGL).

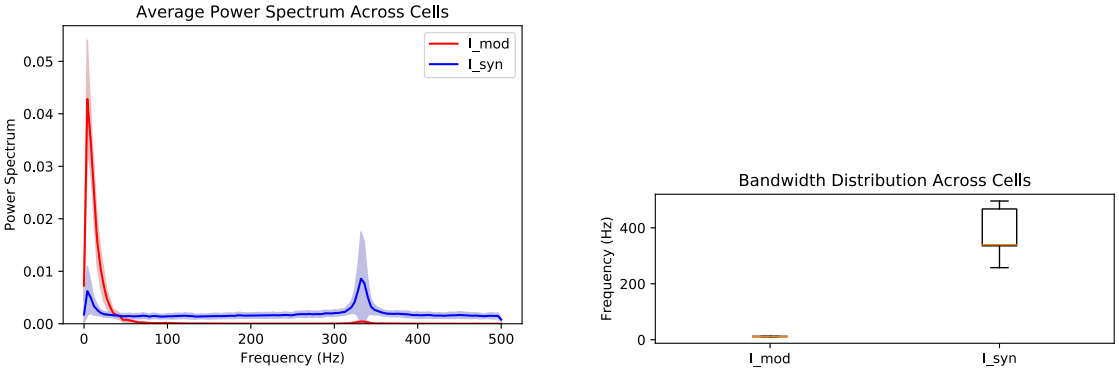

Figure S4: Slowness in modulatory signaling for the online approximation of MDGL. We repeat the spectral analysis in Figure 7, but for the online implementation of local modulatory input in Eq. (S4), i.e. $I_{mod,p} := \sum_{\alpha \in C} w_{\alpha\beta} \sum_{j \in \alpha, p \to j} \bar{a}_{j,t}$ with $\bar{a}_{j,t}$ defined in Eq. (S4). The observations here match those of Figure 7, where modulatory input is significantly slower than synaptic input. We note that the analysis here is done on the pattern generation task only, because for the other two tasks, the error signal is not available until the end of the trial, making the modulatory input too short (see Eq. (S4)) for any meaningful spectral analysis. We expect this "slowness" of modulatory signaling to generalize as the modulatory input is a weighted summation of slow changing leaky outputs and low-pass filtered activity $h_{j,t}$.

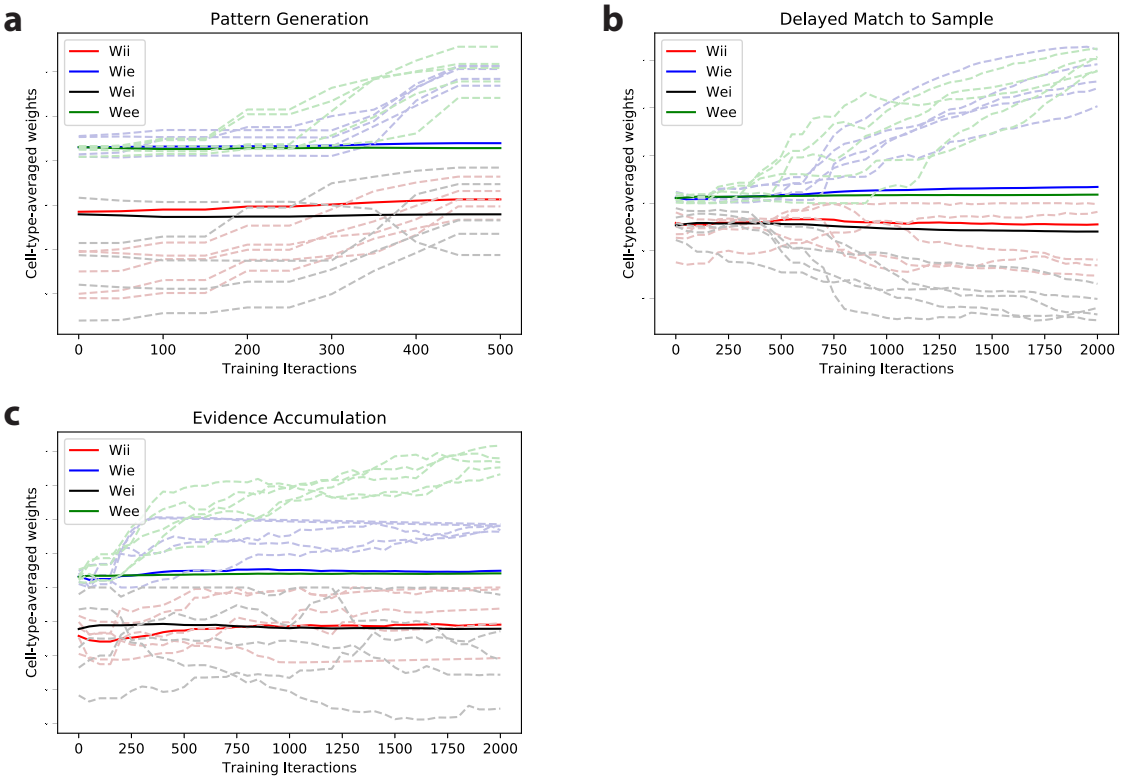

Figure S5: Cell-type-specific signaling gain drifts slowly over training. The four cell-type-specific signaling gains for MDGL with two cell-types, i.e. $w_{\alpha\beta}$ in Eq. (8) and Eq. (12) with $\alpha, \beta \in \{E, I\}$, are illustrated in solid lines for the three tasks investigated. Several sample individual weights with large changes are illustrated in faint dashed lines. Because we implemented cell-type-specific signaling gain using weight averages, it is not surprising that they drift over training as weights adapt. This drift, however, is slow compare to how fast some individual weights can change.

S2 Supplementary Table

Table S1: Local modulatory terms of MDGL and TRTRL are significantly aligned

Task	Beginning Alignment (°)	Beginning Z-score	Middle Alignment (°)	Middle Z-score	End of Training Alignment (°)	End of Training Z-score
Pattern Generation	36	-387	36	-385	44	-317
Delayed Match to Sample	22	-114	48	-74	32	-99
Evidence Accumulation	50	-67	44	-81	55	-59

The alignment angles are computed between the local modulatory terms due to neuron-specific rather than type-specific gain (the vectorization of 2D matrices $\widehat{\Gamma_{pq}} = \sum_t \widehat{\Gamma_{pq,t}}$ of TRTRL in Eq. (7) and those due to the cell-type-specific approximation $\Gamma_{pq} = \sum_t \Gamma_{pq,t}$ of MDGL in Eq. (11) using E and I cell types. The angles listed here are for recurrent weight update, and similar values are observed for input weight updates. Here, beginning, middle and end of training refer to after 10, 100 and 500 training iterations, respectively, for the pattern generation task, and after 100, 500 and 2000 training iterations, respectively, for the delayed match to sample and evidence accumulation tasks. Z-scores were computed to show the significance of the difference of alignment from 90°; these scores suggest that TRTRL and MDGL are significantly aligned. Details for computing Z-scores and alignment angles are provided in Section S3.4.

Table S2: Local modulatory term of MDGL is not significantly aligned with the approximated gradient by e-prop

Task	Beginning Alignment (°)	Beginning Z-score	Middle Alignment (°)	Middle Z-score	End of Training Alignment (°)	End of Training Z-score
Pattern Generation	89	-8.4	80	-69	92	16
Delayed Match to Sample	94	5.3	88	-2.3	92	2.2
Evidence Accumulation	91	1.2	91	1.4	89	-1.2

Similar to Table S1, but the alignment angles are computed between the vectorization of $\Gamma_{pq} = \sum_t \Gamma_{pq,t}$ of MDGL in (11) (the local modulatory term using E and I cell types) and the approximated error gradient $\frac{\mathrm{d}\,E}{\mathrm{d}\,w_{pq}}\Big|_{e-prop}$ of e-prop. Z-scores are computed to determine the significance of the difference of alignment from 90° (which indicates gradients are not aligned). These scores show that the alignment angles are not significantly below 90°, indicating that cell-type-specific local modulatory signaling is offering additional information over and above e-prop. Formulae for Z-scores and alignment angle are provided in Section S3.4.

S3 Training Details

S3.1 Firing Rate Regularization

In addition to accuracy optimization described in Methods, we added a firing rate regularization term E_{reg} to the loss function to ensure sparse firing [23]:

$$E_{\text{reg}} = \frac{1}{2} c_{\text{reg}} \sum_j (f_j^{\text{av}} - f_j^{\text{target}})^2, \tag{S1}$$

where f_j^{target} and $f_j^{\text{av}} = \frac{1}{T}\sum_t z_{j,t}$ are the desired and actual average firing rate for cell j, respectively, and c_{reg} is a positive coefficient that controls the strength of the regularization.

S3.2 Online Learning for Leaky Output

Consider a supervised learning task with loss $E = \sum_{k,t}(y^*_{k,t} - y_{k,t})^2$ and leaky output $y_{k,t} = \kappa\, y_{k,t-1} + \sum_j w_{kj}^{\text{OUT}} z_{j,t} + b_k^{\text{OUT}}$, we have the following partial derivative

$$\frac{\partial E}{\partial z_{j,t}} = \sum_k w_{kj}^{OUT} \sum_{t' \geq t} (y^*_{t',k} - y_{t',k}) \kappa^{t'-t}. \tag{S2}$$

This seemingly provides an obstacle for online learning, because $\frac{\partial E}{\partial z_{j,t}}$ depends on future errors. However, a solution to this problem has been proposed in [23] by changing the summation order and can be generalized to the

classification task with a simple replacement of $(y^*_{k,t} - y_{k,t})$ by $(\pi^*_{k,t} - \pi_{k,t})$:

$$\begin{aligned}
\left.\frac{\mathrm{d}\,E}{\mathrm{d}\,w_{pq}}\right|_{e-prop} &= \sum_{t'} \frac{\partial E}{\partial z_{p,t'}} e^{t'}_{pq} \\
&= \sum_{k,t'} w^{OUT}_{kj} \sum_{t \geq t'} (y^*_{k,t} - y_{k,t}) \kappa^{t-t'} e^{t'}_{pq} \\
&= \sum_{k,t} w^{OUT}_{kj} (y^*_{k,t} - y_{k,t}) \underbrace{\textstyle\sum_{t' \leq t} \kappa^{t-t'} e^{t'}_{pq}}_{\mathcal{F}_\kappa(e^t_{pq})},
\end{aligned} \tag{S3}$$

where the order of summations was changed in the last line, and operator $\mathcal{F}_\kappa$ denotes low-pass filtering with $\mathcal{F}_\kappa(x_t) = \kappa \mathcal{F}_\kappa(x_{t-1}) + x_t$. In our actual implementation, we used an exponential smoothing with $\mathcal{F}_\kappa(x_t) = \kappa \mathcal{F}_\kappa(x_{t-1}) + (1-\kappa) * x_t$, but dropped the factor $(1-\kappa)$ in writing for readability following [23].

We also apply the change of summation order trick to the cell-type-specific modulatory signal Γ^t_{pq} and assume that activity of neuron j is not correlated with the eligibility trace of synapse pq:

$$\begin{aligned}
\sum_{t'} \Gamma^t_{pq} &= \sum_{t',j \neq p} \frac{\partial E}{\partial z_{j,t'}} h_{j,t'} w_{\alpha\beta} e^{t'-1}_{pq} \\
&= \sum_{k,t',j \neq p} w^{OUT}_{kj} \sum_{t \geq t'} (y^*_{k,t} - y_{k,t}) \kappa^{t-t'} h_{j,t'} w_{\alpha\beta} e^{t'-1}_{pq} \\
&\overset{(a)}{=} \sum_t \sum_{j \neq p} \sum_k (y^*_{k,t} - y_{k,t}) w^{OUT}_{kj} w_{\alpha\beta} \underbrace{\textstyle\sum_{t' \leq t} \kappa^{t-t'} h_{j,t'} e^{t'-1}_{pq}}_{\approx (t-t'+1)\mathbb{E}_{t' \leq t}\left[\kappa^{t-t'} h_{j,t} e^{t-1}_{pq}\right]} \\
&\overset{(b)}{\approx} \sum_t \sum_{j \neq p} \sum_k (y^*_{k,t} - y_{k,t}) w^{OUT}_{kj} w_{\alpha\beta} \mathbb{E}_{t' \leq t}\left[h_{j,t}\right] \underbrace{(t-t'+1)\mathbb{E}_{t' \leq t}\left[\kappa^{t-t'} e^{t-1}_{pq}\right]}_{=\mathcal{F}_\kappa(e^{t-1}_{pq})} \\
&= \sum_t \sum_{j \neq p} \sum_k (y^*_{k,t} - y_{k,t}) w^{OUT}_{kj} w_{\alpha\beta} \underbrace{\mathbb{E}_{t' \leq t}\left[h_{j,t}\right]}_{\approx \mathcal{F}_\kappa(h_{j,t})} \mathcal{F}_\kappa(e^{t-1}_{pq}) \\
&\overset{(c)}{\approx} \sum_t \sum_{j \neq p} \sum_k (y^*_{k,t} - y_{k,t}) w^{OUT}_{kj} w_{\alpha\beta} \mathcal{F}_\kappa(h_{j,t}) \mathcal{F}_\kappa(e^{t-1}_{pq}) \\
&= \sum_t \sum_{j \neq p} \underbrace{\left[\textstyle\sum_k (y^*_{k,t} - y_{k,t}) w^{OUT}_{kj} \mathcal{F}_\kappa(h_{j,t})\right]}_{:= \bar{a}_{j,t}} w_{\alpha\beta} \mathcal{F}_\kappa(e^{t-1}_{pq}) \\
&\overset{(d)}{=} \sum_t \mathcal{F}_\kappa(e^{t-1}_{pq}) \sum_{\alpha \in C} w_{\alpha\beta} \sum_{j \in \alpha} \bar{a}_{j,t},
\end{aligned} \tag{S4}$$

where (a) changes the summation order; (b) assumes uncorrelatedness between activity $h_{j,t}$ and $\kappa^{t-t'} e^{t-1}_{pq}$ such that $\mathbb{E}_{t' \leq t}\left[\kappa^{t-t'} h_{j,t} e^{t-1}_{pq}\right] \approx \mathbb{E}_{t' \leq t}\left[h_{j,t}\right] \mathbb{E}_{t' \leq t}\left[\kappa^{t-t'} e^{t-1}_{pq}\right]$; (c) approximates the temporal average of $h_{j,t}$ using an exponential filter $\mathbb{E}_{t' \leq t}\left[h_{j,t}\right] \approx \mathcal{F}_\kappa(h_{j,t})$; (d) is a simple change of summation order. We test the validity of above approximation in Figure S3 and observe no significant performance degradation due to this approaximation.

S3.3 Detailed Breakdown of MDGL's Components

In the main text, we stated that our MDGL learning rule combines the eligibility trace with both top-down learning signals and cell-type-specific weighted summation of secreted, diffuse modulators. We so far only expressed these components as derivatives. With the derivation of the online implementation for MDGL in (S4), we are now ready to provide the detailed expressions for each of these components. Combining (S4) with (12) and rearranging the

summation order gives the following component breakdown for our online approximation to MDGL:

$$\widehat{\frac{\mathrm{d}\,E}{\mathrm{d}\,w_{pq}}} \approx \left[\sum_k (y^*_{t-1,k} - y_{t-1,k}) \left(w^{OUT}_{kp} + \underbrace{\sum_{\alpha \in C} w_{\alpha\beta} \sum_{j \in \alpha} w^{OUT}_{kj} \mathcal{F}_\kappa(h_{j,t-1})}_{\text{Our addition}} \right) \right] \mathcal{F}_\kappa(e_{pq,t-1}). \tag{S5}$$

Similar to [23], non-neuron-specific error signal $(y^*_{t,k} - y_{t,k})$ is passed to cells through neuron-specific feedback weights w^{OUT}_{kj}, thereby forming neuron-specific learning signal at the receiving end $L_{j,t} = \sum_k w^{OUT}_{kj}(y^*_{t,k} - y_{t,k})$. Thus, loosening this neuron-specificity of learning signal can be achieved through approximations to the feedback weights, such as replacing them with random weights [46] or cell-type-specific gains as in Eq. (8). Upon receipt, neuron j multiplies $L_{j,t}$ with $\mathcal{F}_\kappa(h_{j,t})$, its low-pass filtered activity, and sends the packaged signal $a_{j,t} = L_{j,t}\mathcal{F}_\kappa(h_{j,t})$. In updating w_{pq}, our addition allows postsynaptic cell p to collect information regarding the activities and learning signals of other cells through cell-type-specific gain $w_{\alpha\beta}$, and combine the received modulatory input with its low-pass filtered eligibility trace.

S3.4 Analysis and Simulation Details

Throughout this study, the alignment angle θ between two vectors, a and b, was computed by $\theta = acos(\left\|a^T b\right\| / \|a\| \, \|b\|)$. The alignment between two 2D matrices was computed by flattening the matrices into vectors. To obtain the significance of alignment in Tables S1 and S2, we randomly shuffled the matrices, calculated the resulting alignment angle and repeated for 1000 times to obtain an empirical distribution of alignment angles. The mean μ and standard deviation σ were computed from the distribution to report the Z-score $= \frac{\theta - \mu}{\sigma}$.

For spectral analysis, we first performed root mean square normalization on the signal and then computed the power spectral density using Welch's method [71]. We then found the 3dB frequency by identifying the maximum frequency at which the power is halved from the peak power.

For the pattern generation task, our network consisted of 400 LIF neurons. All neurons had a membrane time constant of $\tau_m = 30$ms, a baseline threshold of $v_{\text{th}} = 0.01$ and a refractory period of 2ms. Input to this network was provided by 100 Poisson spiking neurons with a rate of 10Hz. The fixed target signal had a duration of 2000ms and given by the sum of five sinusoids, with fixed frequencies of 0.5Hz, 1Hz, 2Hz, 3Hz and 4Hz. For learning, we used mean squared loss function and for visualization, we used normalized mean squared error NMSE $= \frac{\sum_{k,t}(y^*_{k,t} - y_{k,t})^2}{\sum_{k,t}(y^*_{k,t})^2}$ for zero-mean target output $y^*_{k,t}$. All weight updates were implemented using Adam with default parameters [72] and a learning rate of 1×10^{-3}. In addition, we applied firing rate regularization with $c_{\text{reg}} = 10$ and $f^{\text{target}} = 10Hz$.

For the delayed match to sample task, our network consisted of 50 LIF neurons and 50 ALIF neurons. All neurons had a membrane time constant of $\tau_m = 20$ms, a baseline threshold of $v_{\text{th}} = 0.01$ and a refractory period of 5ms. The time constant of threshold adaptation was set to $\tau_a = 1400$ms, and its impact on the threshold was set to $\beta = 1.8$. Input to this network was provided by three populations, as illustrated in Figure 5B. The first (resp. second) population consisted of 20 units and produced Poisson spike trains with a rate of 40Hz when the first (resp. second) cue takes a value of 1, otherwise it stays quiescent. The last input population of 10 units produced Poisson spike trans of 10Hz throughout the trial in order to prevent the network from being quiescent during the delay. For learning, we used cross-entropy loss function and the target corresponding to the correct output was given at the end of the trial. As done in the evidence accumulation task, a weight update was applied once every 64 trials and the gradients were accumulated during those trials additively. All weight updates were implemented using Adam with default parameters [72] and a learning rate of 2.5×10^{-3}. In addition, we applied firing rate regularization with $c_{\text{reg}} = 0.1$ and $f^{\text{target}} = 10$Hz.

For the evidence accumulation task, our network consisted of 50 LIF neurons and 50 ALIF neurons. All neurons had a membrane time constant of $\tau_m = 20$ms, a baseline threshold of $v_{\text{th}} = 0.01$ and a refractory period of 5ms. The time constant of threshold adaptation was set to $\tau_a = 2000$ms, and its impact on the threshold was set to $\beta = 1.8$. Input to this network was provided by four populations of 10 neurons each, as illustrated in Figure 6B. The first (resp. the second) population produced Poisson spike trains with a rate of 40Hz when a cue was presented on the left (resp. right) side of the track. The third input population spiked randomly through the decision period with a firing rate of 40Hz and was silent otherwise. The last input population produced Poisson spike trains with a rate of 10Hz throughout the trial in order to prevent the network from being quiescent during the delay. For learning, we used the cross-entropy loss function and the target corresponding to the correct output was given at the end of the trial.

As done in [23], a weight update was applied once every 64 trials and the gradients were accumulated during those trials additively. All weight updates were implemented using Adam with default parameters [72] and a learning rate of 2.5×10^{-3}. In addition, we applied firing rate regularization with $c_{\text{reg}} = 0.1$ and $f^{\text{target}} = 10Hz$. For all simulations, we used a time step of 1ms, as done in [23]. We also assumed a synaptic delay of 1ms for all synapses.