CENTER FOR Brains Minds+ Machines

CBMM Memo No. 152 **December 28, 2024**

Self-Assembly of a Biologically Plausible Learning Circuit

Qianli Liao[†,1,5], Liu Ziyin[†,3,4], Yulu Gan[†,1,2], Brian Cheung[1,2], Mark Harnett[5,6], Tomaso Poggio[1,2,5,6]

[†] Equal contribution
[1]Center for Brains, Minds, and Machines, MIT
[2]CSAIL, MIT
[3]Research Laboratory of Electronics, MIT
[4]Physics & Informatics Laboratories, NTT Research
[5]McGovern Institute, MIT
[6]Department of Brain and Cognitive Sciences, MIT

Abstract

Over the last four decades, the amazing success of deep learning has been driven by the use of Stochastic Gradient Descent (SGD) as the main optimization technique. The default implementation for the computation of the gradient for SGD is backpropagation, which, with its variations, is used to this day in almost all computer implementations. From the perspective of neuroscientists, however, the consensus is that backpropagation is unlikely to be used by the brain. Though several alternatives have been discussed, none is so far supported by experimental evidence. Here we propose a circuit for updating the weights in a network that is biologically plausible, works as well as backpropagation, and leads to verifiable predictions about the anatomy and the physiology of a characteristic motif of four plastic synapses between ascending and descending cortical streams. A key prediction of our proposal is a surprising property of self-assembly of the basic circuit, emerging from initial random connectivity and heterosynaptic plasticity rules.

This material is based upon work supported by the Center for Brains, Minds and Machines (CBMM), funded by NSF STC award CCF-1231216.

Self-Assembly of a Biologically Plausible Learning Circuit

Qianli Liao* [1,5], Liu Ziyin* [3,4], Yulu Gan* [1,2], Brian Cheung[1,2], Mark Harnett[5,6], and Tomaso Poggio[1,2,5,6]

[1]Center for Brains, Minds, and Machines, MIT
[2]CSAIL, MIT
[3]Research Laboratory of Electronics, MIT
[4]Physics & Informatics Laboratories, NTT Research
[5]McGovern Institute, MIT
[6]Department of Brain and Cognitive Sciences, MIT

Dec. 27th, 2024

Abstract

Over the last four decades, the amazing success of deep learning has been driven by the use of Stochastic Gradient Descent (SGD) as the main optimization technique. The default implementation for the computation of the gradient for SGD is backpropagation, which, with its variations, is used to this day in almost all computer implementations. From the perspective of neuroscientists, however, the consensus is that backpropagation is unlikely to be used by the brain. Though several alternatives have been discussed, none is so far supported by experimental evidence. Here we propose a circuit for updating the weights in a network that is biologically plausible, works as well as backpropagation, and leads to verifiable predictions about the anatomy and the physiology of a characteristic motif of four plastic synapses between ascending and descending cortical streams. A key prediction of our proposal is a surprising property of self-assembly of the basic circuit, emerging from initial random connectivity and heterosynaptic plasticity rules.

1 Introduction

The rapid development of deep learning architectures has had a significant impact on computational neuroscience, where these networks are increasingly used to model various aspects of information processing in the brain (Cichy et al., 2016; Yamins & DiCarlo, 2016; Güçlü & van Gerven, 2015; Kriegeskorte, 2015). These studies suggest that deep learning models can mimic the way visual information is processed in the brain, providing insights into both simple and complex cognitive functions. Despite this promise, obstacles remain in adopting these models, chiefly due to the large discrepancy between the artificial neurons and synapses used in deep learning and the biological neurons and synapses found in the vertebrate brain. A prime example of this gap is evident in the supervised learning algorithms, such as backpropagation, that are employed to train deep networks. Backpropagation, a specific implementation of Stochastic Gradient Descent (SGD), faces significant biological implausibility. Most neuroscientists agree that directly implementing backpropagation in the brain is highly unlikely (Lillicrap et al., 2020; Whittington & Bogacz, 2019), primarily due to the requirement for identical weights in what is known as the 'weight transport' problem.

The question is then whether equivalent but biologically plausible algorithms can be formulated and experimentally validated. A positive answer would provide a strong support to deep learning models of the brain, connecting in a fundamental way the engineering of machine learning with the science of the

*Equal contribution.

brain. Such a discovery could easily be transformative for both neuroscience and machine learning. Here, we propose a neural circuit for supervised learning that is biologically plausible, works well in our simulations and leads to experimentally verifiable predictions while using well-established properties of real neurons and synapses. In particular, our proposal predicts a characteristic repeating motif of four synapses between ascending and descending cortical streams obeying heterosynaptic plasticity rules. The most surprising aspect of our proposal is an emerging property of self-assembly of the basic circuit starting from random synaptic connections between the ascending and the descending stream. Interestingly, the best performance is obtained when the number of backprojections is around 5 times the connections in the forward stream.

Our work is not the first to propose biologically plausible alternatives to backpropagation (see Liao et al. (2015); Lillicrap et al. (2016); Nøkland (2016); Xiao et al. (2018); Nøkland & Eidnes (2019)). Recent papers include top-down weight alignment methods (see also Max et al. (2024)) and alternative cost minimization schemes. Among several papers Lillicrap et al. (2020) provides a comprehensive review. The most closely related approach to ours is the Kolen-Pollack algorithm, described in Akrout et al. (2019b), which is a very special case of our circuit. Very recently, Abel and Ullman (Abel & Ullman) showed how to integrate learning with other visual tasks, such as visual guidance, by exploiting the combination of ascending and descending cortical pathways. Our circuit could also perform additional tasks in addition to supervised learning.

Our theoretical and experimental results demonstrates that our circuit can not only emulate the computational abilities of conventional deep learning models but do so through mechanisms that align closely with biological evidence. Specifically, our simulations showed that the network trained with our algorithm could achieve competitive performance on tasks such as image classification on CIFAR-10, closely matching or even surpassing traditional backpropagation systems in some cases. Theoretically, we established that our model's learning dynamics could be approximated by a modified form of gradient descent, which is not only compatible with local synaptic plasticity rules but also exhibits convergence behavior that aligns well with observed biological learning processes. This synthesis of engineering efficiency with biological plausibility could pave the way for new models that bridge the gap between artificial and natural systems, enhancing our understanding of both in the process.

2 A Neurally Plausible Circuit for Deep Learning

We propose a neural circuit grounded in simple, biologically plausible synaptic motifs. Our circuit consists of two interacting streams of connections: an ascending stream representing forward pathways and a descending stream representing feedback pathways, potentially mirroring cortical forward and back projection circuits. These streams are defined by a set of synaptic weights, which are subject to localized heterosynaptic plasticity rules. Unlike backpropagation, our model does not require the explicit computation of derivatives, and it does not require weight symmetry between forward and feedback pathways.

Key features that differentiate our families of neural circuits from other approaches include:

- *Local Learning Rules*: All synaptic updates are driven by localized heterosynaptic plasticity;
- *Biological Feasibility*: The architecture aligns with known neurophysiological properties.
- *Reduced Assumptions*: Our circuits do not require any weight symmetry and still achieve competitive performance on challenging datasets like CIFAR-10;
- *Structural Resilience*: Our motif works with any nonlinearity and for an arbitrary width of the forward and backward pathway. Namely, the network is robust to any small changes in the local configurations of the pathways.

Simulations of our circuit demonstrate robust learning capabilities. As the experimental results in Sec. 3.1 indicate, the network's performance is competitive against that of backpropagation-trained models, even though the algorithm does not mimic backpropagation explicitly. These findings highlight the potential of our circuit as a biologically plausible alternative for supervised or semi-supervised learning. The most interesting implications regard the anatomical predictions about the proposed synaptic motifs and their plasticity properties. We will introduce the algorithm and the associated architectural motif in Sec. 2.1, which details the components of the Basic Circuit. Following this, in Sec. 2.2, we demonstrate that our basic circuit is a generalization of SGD, and that its backward pathway can be overparametrized, consistent with discoveries in biological circuits. Sec. 2.3 and Sec. 2.4 will further explore other interesting properties.

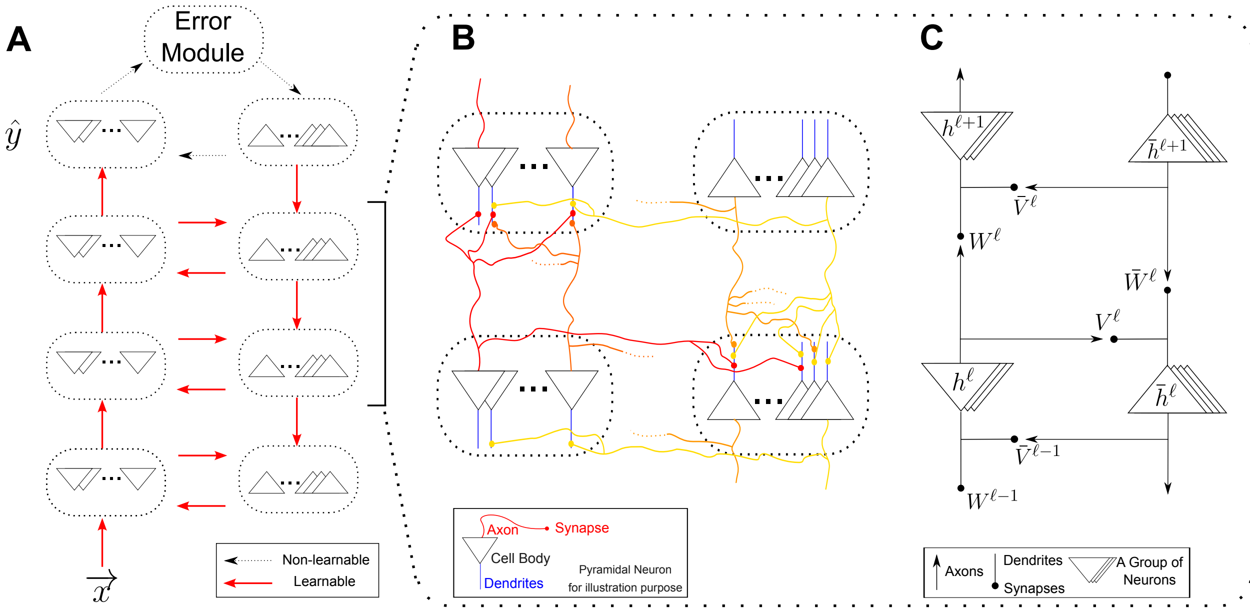

Figure 1: A scheme of the upstream-downstream synaptic motif. **A**: The overall scheme for the upstream-downstream architecture. The upstream consists of a standard fully-connected neural network with multiple layers, possibly corresponding to a multi-region processing pathway in the cortex like V1-V2-V4-IT. The output of the upstream network goes to an error processing module (possibly corresponding to PFC in the brain). The error module computes a local error signal that can be used to immediately train the last layer. This error signal is also sent to the feedback (downstream) pathway, which processes information layer by layer downwards. The black dashed arrows represent non-learnable (identity) connections. Red solid arrows represent learnable connections, each parameterized by a fully-connected weight matrix. **B**: a biological sketch of the smallest unit of the connection motif of A. The neurons in B (as well as the abstract forms in A and C) are illustrated as pyramidal neurons, which are common in the cortex. Pyramidal neurons typically have axons extending from the base of the cell body, while their dendrites can grow from the top (apical dendrites) or bottom (basal dendrites). Only apical dendrites are illustrated here for simplicity. **C**: a mathematical description of this unit. Each arrow (and corresponding axons, dendrites and synapses) represents a set of full connections between two groups of neurons, parameterized by a weight matrix W or V. Every h in C is a vector and refers to the activations of a group of neurons. The connection matrix V allows upstream and downstream networks to have different number of hidden units in corresponding layers.

Notation. Let θ be the parameters of the model. We will use $\Delta f(\theta)$ to denote the difference of the quantity $f(\theta)$ after one step of the learning algorithm: $\Delta f(\theta) := f(\theta_{t+1}) - f(\theta_t)$. Any vector without a transpose superscript is treated as a column vector (e.g. $\nabla F(\theta)$ is a column vector for a scalar function F).

2.1 The Basic Circuit

We assume that each pathway is organized in layers indexed with $\ell \in \{1, ..., L\}$, where L is the depth of the pathway in terms of modules. Let $h^\ell \in \mathbb{R}^{d_\ell}$ denote the neuron activation of the ℓ-th layer *upstream pathway* and $\bar{h}^\ell \in \mathbb{R}^{\bar{d}_\ell}$ the *downstream pathway*. For reasons that will become clear, the upstream is also referred to as the forward pathway, and the downstream is the feedback pathway. Note that it is, in general, the case that the two pathways have different numbers of neurons, and so d_ℓ is not necessarily equal to $\bar{d}_\ell$. The simplest such biological motif is shown in Figure 1, where four synaptic connection matrices connect two consecutive layers of the two pathways. Here, $W^\ell \in \mathbb{R}^{d_{\ell+1} \times d_\ell}$ is the connection from the ℓ-th layer upstream pathway to the $\ell+1$-th layer upstream pathway, $\bar{W}^\ell \in \mathbb{R}^{\bar{d}_\ell \times \bar{d}_{\ell+1}}$ the connection from $\ell+1$-th layer downstream to the ℓ-th layer downstream. $V^\ell \in \mathbb{R}^{\bar{d}_\ell \times d_\ell}$ and $\bar{V}^\ell \in \mathbb{R}^{d_\ell \times \bar{d}_\ell}$ are inter-stream connections: V goes from the downstream to the upstream, and $\bar{V}$ from the upstream to the downstream. The algorithm's Pseudo-code is in Appendix D.

The computation of the two pathways takes place sequentially, where the upstream computation is performed first and eventually used for inference, and the downstream computation takes place after inference and is used for synaptic updates:[1]

$$h^{\ell+1} = D_u(W^\ell h^\ell, \bar{V}^\ell \bar{h}^{\ell+1}) W^\ell h^\ell \tag{1}$$

$$\bar{h}^\ell = D_d(\bar{W}^\ell \bar{h}^{\ell+1}, V^\ell h^\ell) \bar{W}^\ell \bar{h}^{\ell+1}, \tag{2}$$

where h and $\bar{h}$ is the membrane potential of the cells in the upstream and downstream pathways, and D is a diagonal matrix that functions as the neuron nonlinearity and may be different for upstream and upstream pathways. In the most general form, $D(\cdot)$ can depend on the input from both h and $\bar{h}$. The first layer of $h^1 := x$ is the input data, and $h^L := \hat{y}$ is the output of the network. The input to the downstream pathway is an error signal obtained from an objective function F: $\bar{h}^L := \epsilon(\hat{y}) = -\nabla_{\hat{y}} F(\hat{y})$, which is computed from the output of the upstream pathway. The upstream propagates this signal in the reverse direction. One example of F is the MSE loss, where $F(\hat{y}) = \|y - \hat{y}\|^2$, where y is the correct label.

In this work, we always choose D_u to be a diagonal zero-one matrix:

$$D_u = \mathrm{diag}(\mathbb{1}_{(Wh^\ell)_1}, ..., \mathbb{1}_{(Wh^\ell)_d}), \tag{3}$$

where $\mathbb{1}$ is the indicator function, and this corresponds to a ReLU nonlinearity. For D_d, we focus on the simplest case where $D_d = I$ is the identity matrix. Additional experiments that explore different choices of D_d are explored in Section 3.2, and we find that the algorithm is rather robust to different choices of D_d. Another notable choice of D_d is where it is the Jacobian of a ReLU activation:

$$D_d = \mathrm{diag}(\mathbb{1}_{(Vh^\ell)_1}, ..., \mathbb{1}_{(Vh^\ell)_{\bar{d}}}). \tag{4}$$

When $V = \bar{V} = I$ and with this choice of D_d, our algorithm reduces to the KP-algorithm, which converges to the exact gradient descent provably (Kolen & Pollack, 1994).

The dynamics of learning is given by the following heterosynaptic plasticity rules:

$$\Delta W^\ell = \eta_W \bar{V}^\ell \bar{h}^{\ell+1} (h^\ell)^T - \gamma R_1(W^\ell), \tag{5}$$

$$\Delta V^\ell = \eta_V \bar{h}^\ell (h^\ell)^T - \gamma R_2(V^\ell), \tag{6}$$

$$\Delta(\bar{W}^\ell)^T = \eta_{\bar{W}} \bar{h}^{\ell+1} (h^\ell)^T (V^\ell)^T - \gamma R_3(\bar{W}^\ell), \tag{7}$$

$$\Delta(\bar{V}^\ell)^T = \eta_{\bar{V}} \bar{h}^{\ell+1} (h^{\ell+1})^T - \gamma R_4(\bar{V}^\ell), \tag{8}$$

where η is the time constant of learning (i.e., the learning rate), $R_i(\cdot)$ are local regularization terms that, for example, encourage the synapses to have a weight that is of small norm or sparse, and γ is the regularization strength. Both the learning rate and the regularization strength are labeled with a different subscript to emphasize that they could have different time constants. In this work, we always keep the regularization strengths uniform. We empirically search over 64 combinations of the learning rates and use the best one in the experiment.[2] We found it particularly beneficial to fix the ratio between the four learning rates, while tuning the overall factor for different tasks.

The simplest type of such regularization is $R_1 = W$, $R_2 = V$, $R_3 = \bar{W}^T$, and $R_4 = \bar{V}$, which corresponds to having a weight decay term in the learning dynamics. Our theory and experiment will focus on this type of regularization, although it is fully possible to have different and more complex regularization effects for different pathways, which we discuss briefly in the conclusion. A key feature of this algorithm is that it allows all possible connections of the two-pathway motif to become self-assembled. We thus tentatively name this algorithm "Self-Assembling Learning" (**SAL**). Strictly speaking, this type of udpate rule is not Hebbian as the classical Hebbian rule is homosynaptic, whereas this update rule is a form of heterosynaptic update rule. We discuss the biological evidences for this type of rules in Appendix B.

[1]This sequential ordering is consistent with, for example, the observation that inactivating V1 also leads to the inactivation of V2, while deactivating V2 has no effect on the activations of V1 (Anderson & Martin, 2009).

[2]See Section C.

2.2 Gradient Learning in Overparametrized Downstream Pathway

The circuits described can be viewed as a generalized version of SGD, with a matrix form and learnable learning rate – in this sense, the algorithm is learning the learning algorithm itself. The following theorem is an informal statement of this result. The full formal detail and proof are provided in Appendix A.

Theorem 1. *Consider a ReLU neural network with an arbitrary width and depth and with an overparametrized downstream pathway. If both V^ℓ and $\bar{V}^\ell$ are full-rank for all ℓ, then, for any x such that for all $\ell \in [L]$, $\Delta V^\ell = O(\epsilon)$ and $\Delta \bar{V}^\ell = O(\epsilon)$,*[3]

$$\Delta_{\text{SAL}} W^\ell = H \Delta_{\text{sgd}} W^\ell - \gamma W^\ell + O(\epsilon), \tag{9}$$

for a positive definite matrix $H = \bar{V}^\ell (\bar{V}^\ell)^T$, and $\Delta_{\text{sgd}} W^\ell = -\nabla_W F$ is the SGD update.

Remark. *This theorem essentially shows that when V and $\bar{V}$ reach stationarity, the algorithm will run like a SGD algorithm with a matrix learning rate H and weight decay. The fact that matrix H is PD, is crucial from a mathematical perspective, as it guarantees that the loss objective of training will decrease in expectation and that the stationary points of the circuit will be identical to that of SGD. For example, besides SGD itself, Adam or Natural Gradient Descent can also be seen as having a learning rate corresponding to a PD matrix.*

It is worth noticing that the algorithm is different from SGD. H is not only a layer-wise matrix learning rate: it is explicitly dependent on time and is also being learned by the algorithm! Therefore, our algorithm can be seen as a generalization of SGD to the case where the learning algorithm itself is being learned, which can be conceptually regarded as a form of meta-learning (Metz et al., 2018).

Another interesting aspect of the theorem is that it allows (but does not require) the backward pathway to be overparametrized. That is, the number of downstream neurons can be larger than the number of forward neurons. This is congruent with observations in biological circuits, but it is inconsistent with previous biologically inspired algorithms of learning, which require the same number of connections for ascending and descending pathways. Also, it turns out that the backward pathway is highly flexible in the choice of the activation; with any choice of D_d, the theorem holds. This prediction of the robustness of the backward pathway is numerically supported by the experiment in Section 3.2.

When is it possible to satisfy the condition that V and $\bar{V}$ are stationary? There are many possible ways. Two notable cases are when V and $\bar{V}$ are permutation matrices, and D_d is the Jacobian matrix of D_u. In this case, VV^T is identity, and so our algorithm is completely identical to SGD. In a different case, the feedback pathway is very expressive (e.g., by having a large number of feedback neurons), and so it is capable, from an approximation perspective, of approximating the forward net.

2.3 Short-Term Modulation and Long-Term Learning

Above, we have explained how the circuit could be used to update the synaptic weight, which is a form of long-term learning. Now, we show that the proposed circuit can also serve the dual purpose of performing meaningful short-term modulation, which has been hypothesized to be the main function of feedback connections in the brains of mammals (Briggs, 2020). Suppose that the circuit receives the input signal x and has completed the forward and backward computation cycle once, which can happen in the brain in a relatively fast time scale, often of the order of tens to a few hundred milliseconds.

The circuit now allows the transmission of signal $\bar{V}\bar{h}$ to affect h, and the forward neuron can accumulate this signal to compute $h \to h + \bar{V}\bar{h}$ at every layer. This has the effect of modulating the signal in the forward pass and leads to a reduced loss function value for this particular data point. To see this, note that $\bar{V}\bar{h} = -\nabla_h F(h)$. Applying this argument to every layer shows that the effect of this modulation is exactly accumulating the gradient of h onto the corresponding neuron, which reduces the objective value for a sufficiently small time constant. In this sense, the feedback pathway functions like a recurrent computation

[3] Also, note that this algorithm can be regarded as running an alternative form of L_2 regularization. If we regard H as the learning rate, then the weight decay term is equivalent to running SGD on a matrix-form weight decay strength: $\gamma \text{Tr}[H^{-1}WW^T]$. It decays the weights stronger in directions of H with a smaller eigenvalue.

module that modulates the forward pathway by learning "in context." That this circuit can serve the forward computation also makes the algorithm appealing from an evolutionary point of view, as this circuit can be immediately leveraged to enhance functionality.

Table 1: Performance of the proposed method compared with SGD and biologically plausible algorithm baselines. **Bold** denotes the best performing algorithm, and *italics* denote the best runner-up algorithm. We see SGD and the SAL are comparable, while significantly outperforming existing biologically plausible learning algorithms. For each experiment, we conduct five runs and calculate the error bars.

	CIFAR10	MNIST	Fashion MNIST	ChestMNIST	PathMNIST	SVHN	STL10
				Acc. ↑			
SGD	*50.36 ± 0.16*	*98.44 ± 0.09*	*96.80 ± 0.05*	**73.26 ± 0.14**	**70.11 ± 0.10**	*78.20 ± 0.13*	*41.00 ± 0.07*
FA	47.85 ± 0.20	97.80 ± 0.19	95.75 ± 0.20	70.44 ± 0.27	69.90 ± 0.10	78.03 ± 0.15	39.29 ± 0.23
WM	49.35 ± 0.13	98.40 ± 0.23	96.00 ± 0.25	70.14 ± 0.08	70.00 ± 0.15	78.08 ± 0.15	39.48 ± 0.19
KP Algorithm	48.16 ± 0.09	98.45 ± 0.09	96.13 ± 0.07	71.03 ± 0.10	69.51 ± 0.05	78.20 ± 0.05	40.32 ± 0.11
SAL (ours)	**51.91 ± 0.21**	**98.85 ± 0.10**	**97.23 ± 0.08**	**73.21 ± 0.11**	**70.23 ± 0.15**	**78.50 ± 0.07**	**41.50 ± 0.10**

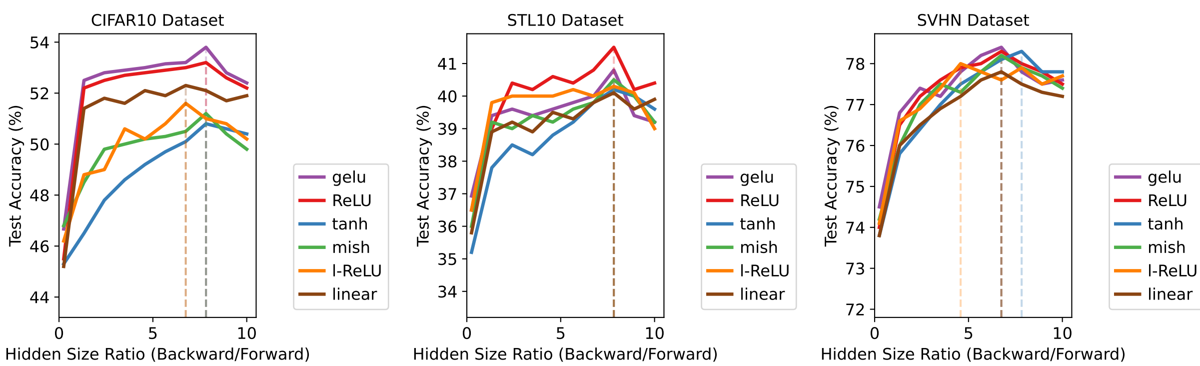

Figure 2: The impact of the width of the downstream pathway on performance. We use different activation functions in the downstream pathway. The width of the upstream pathway was kept constant at 200, while varying the width of the downstream pathway. We find that having a wider downstream network improves the performance of the feedforward network up to an overparametrization ratio of 7.5. In biology, the ratios can also depend on biological constraints such as energy consumption or wiring volume and may be region-dependent.

2.4 Equivalent Local Losses

Another interesting perspective of the algorithm is that it can be viewed as having local losses for the update rules of V, $\bar{V}$ and $\bar{W}$. Note that their update rules are identical to running a gradient descent step on the following three quadratic loss functions:

$$F_V^\ell = \|\bar{V}^\ell \bar{h}^{\ell+1}(h^\ell)^T - \gamma V^\ell\|_F^2, \tag{10}$$

$$F_{\bar{V}}^\ell = \|\bar{h}^{\ell+1}(h^\ell)^T(W^\ell)^T - \gamma \bar{V}^\ell\|_F^2, \tag{11}$$

$$F_{\bar{W}}^\ell = \|\bar{h}^{\ell+1}(h^\ell)^T(V^\ell)^T - \gamma \bar{W}^\ell\|_F^2, \tag{12}$$

The first term within each loss is independent of the parameter being updated. This also makes our suggested learning circuit relevant to the study of local losses in deep learning (Nøkland & Eidnes, 2019). This perspective suggests that one can generalize the update rules to alternative convex loss functions, which can lead to more efficient, biologically plausible learning dynamics.

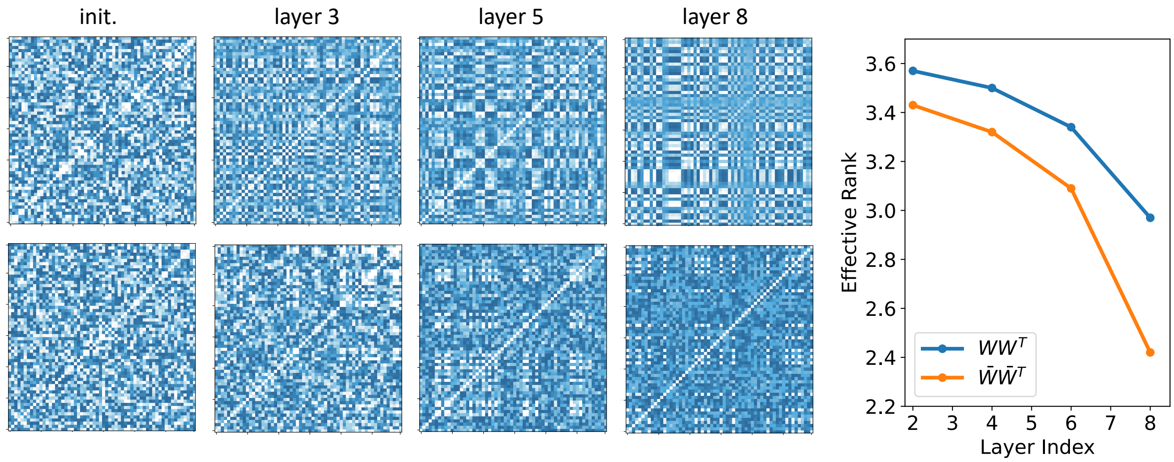

Figure 3: The matrices $\bar{V}\bar{V}^T$ (**upper**) and WW^T (**lower**) before and after training. Also, recall that $\bar{V}\bar{V}^T = H$ is the matrix learning rate for W (Theorem 1). The neurons within the same layer become correlated after training. The spectra of the weight matrices of the two pathway are interestingly found to be effectively low-rank (**right**). This visualization was done using the SVHN dataset with a network having 8 hidden layers.

3 Simulations

To demonstrate the effectiveness of our algorithm in image classification, we present its performance on seven widely-used image classification datasets in Section 3.1. We also evaluate the effect of different backward pathway widths on performance in Section 3.2. The representations learned by our model are visually and quantitatively analyzed in Section 3.3. In Section 3.4, we investigate the impact of selectively ablating weights in both forward and backward pathways (W, $\bar{W}$, V, and $\bar{V}$), revealing insights into the critical components of our architecture. Lastly, we present additional numerical results in Figure 5, exploring the effects of varying learning rates on our model's training dynamics and final performance. Unless stated otherwise, we deploy our algorithm on a 5-layer Multilayer Perceptron (MLP), including input and output layers, with each hidden layer having 200 neurons in both the upstream and downstream pathways. The activation function for the upstream pathway is ReLU.

3.1 Benchmark Comparisons

We evaluate our algorithm on seven widely used image classification datasets: CIFAR10 (Krizhevsky et al., 2009), MNIST (Yann, 1998), Fashion MNIST (Xiao et al., 2017), ChestMNIST (Netzer et al., 2011), PathMNIST (Yang et al., 2021, 2023), SVHN (Netzer et al., 2011), and STL10 (Coates et al., 2011). For details about these datasets, see Appendix C. We compare our algorithm's performance against the standard backpropagation and other biologically plausible algorithms. These include Feedback Alignment (FA) (Lillicrap et al., 2016), which uses random fixed weights for error propagation instead of transposed weights; Weight Mirroring (WM) (Akrout et al., 2019a), a method enforcing symmetry between the forward and backward passes by mirroring weights; and the KP algorithm (Akrout et al., 2019a), which adjusts synaptic weights without requiring weight transport, simplifying network architecture and enhancing learning in large networks. While the primary goal of this paper is not to pursue state-of-the-art (SOTA) performance but to design a neural network algorithm that can be deployed in the brain, our approach achieves SOTA among biologically plausible algorithms on all datasets (see in Table 1). Notably, our algorithm outperforms models trained with backpropagation on six out of the seven datasets. The only exception is ChestMNIST, where it performs *0.02%* lower than backpropagation.

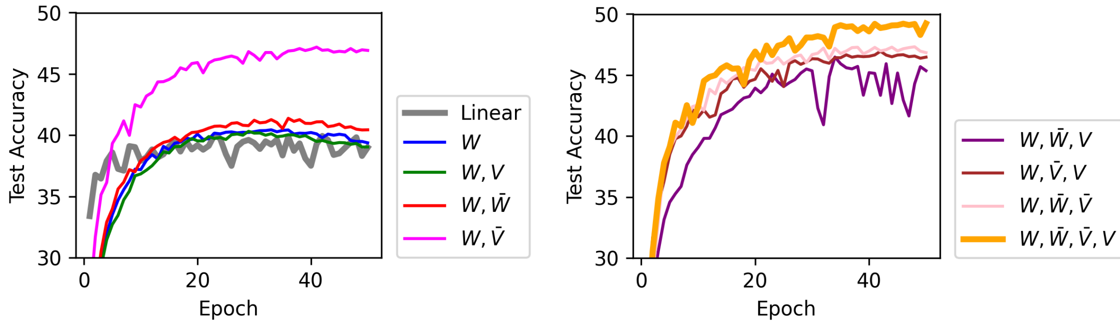

Figure 4: Ablation study on roles of $\bar{V}$, $\bar{W}$ and V on CIFAR-10. Here, we make a subset of all interconnections that are not trainable. The legend labels indicate trainable components. "Linear" refers to a linear network trained with SGD. We see that (1) making $\bar{V}$ plastic is more important in making other connections plastic, and (2) making everything plastic significantly improves the performance further.

3.2 Overparametrization in the Downstream Pathway

In anatomy, there is evidence that the number of fibers in the backprojection can be much larger than that of the forward stream. For example, the feedback connections V1 to LGN is can be roughly 10 times the RGC synapses from the retina (Briggs, 2020). Interestingly, we find an explanation of this phenomenon in our experiments. We discover that the model performs better as the feedback pathway becomes overparametrized and achieves the best performance across multiple datasets when the parameter count of the downstream pathway is approximately 5 to 8 times that of the upstream pathway. Specifically, as shown in Figure 2, for CIFAR10, the performance peaks when the parameter count of the downstream pathway is 8 times that of the upstream pathway. We perform experiments on the three most challenging datasets from the seven datasets used in Table 1: CIFAR10 (Krizhevsky et al., 2009), SVHN (Netzer et al., 2011), and STL10 (Coates et al., 2011). Another important prediction of the theory supported by this experiment is that the backward pathway is rather robust to different choices of activation in the backward pathway.

3.3 Examples of Learned Representations

Now, we show some examples of the learned weight matrices. We train a width-60 network on SVHN dataset for 50 epochs. See Figure 3. As is common in deep learning, we plot the matrix WW^T and also $\bar{V}\bar{V}^T$. We see that before training, these matrices are primarily diagonal matrices due to i.i.d. Gaussian initialization. After training, correlations between neurons and weights appear. We characterize the rank of a matrix through a continuous metric referred to as the effective rank (Roy & Vetterli, 2007). We find that these weight matrices also become effectively low-rank.

Figure 3 can be compared with similar matrices trained with SGD in Ziyin et al. (2024) and Huh et al. (2021), for example, and they seem to have similar high-level structures, suggesting that our algorithm can lead to a meaningful representation learning. Detailed analysis of the learned representation using our algorithm may be an interesting future problem.

3.4 Pathway Ablation Study

A remaining interesting question is whether learning some of the interconnections is more important, and if so, which one is more important. On top of that, is there any interconnection that is not important at all? We perform this ablation study on CIFAR-10, where we initialize all the parameters randomly but only update a subset of all the connections. See Figure 4. Note that W is always plastic. To gauge how much the network has learned about the nonlinear relations in the data, we also compare with the learning trajectory of a linear model trained with SGD.

We discover the following interesting phenomena:

Table 2: The upstream-downstream paradigm of biologically plausible learning, emphasizing the necessity for four types of connections between the two pathways: downstream to downstream ($d \to d$), downstream to upstream ($d \to u$), upstream to downstream ($u \to d$), and upstream to upstream ($u \to u$). Each connection must exhibit plasticity. For a learning rule to be biologically plausible, it must demonstrate both **connectivity** and **plasticity**.

	Connectivity				Plasticity			
	$u \to u$	$d \to d$	$u \to d$	$d \to u$	$u \to u$	$d \to d$	$u \to d$	$d \to u$
SGD	✓	✗	✗	✗	✓	✗	✗	✗
Feedback Alignment	✓	✓	✗	✗	✓	✗	✗	✗
Weight Mirroring	✓	✓	✗	✗	✓	✓	✗	✗
SAL	✓	✓	✓	✓	✓	✓	✓	✓

- Only updating W works but only works as well as the linear model; making other parts plastic is thus important for learning nonlinear functions;
- If only one additional connection is made plastic, $\bar{V}$ is the most important one, improving performance from 40% to roughly 46%;
- Making $\bar{V}$ plastic is similar to making W, V plastic together, suggesting that some loss of learning capabilities due to connection lesions can be compensated by other connections (but not all);
- Making all four connections plastic improves the performance further, to the SGD level (cf. Table 1); this means that all connections need to be plastic to achieve the best performance.

Therefore, none of the pathways we introduced are redundant, and this partially explains the outperformance of our proposed algorithm over the existing two-pathway biologically plausible learning algorithms, as they can be seen as ablated special cases of our algorithm. We elaborate on this point in the next section.

4 A Synaptic Motif for Supervised Learning in Cortex

A recurring theme in recent proposals of biologically plausible learning algorithms is the existence of both upstream and downstream pathways. For example, this assumption is key for the feedback alignment proposal and also in the earlier KP algorithm (Akrout et al., 2019b). An early description of stream and counterstream pathways for optimization tasks is due to Ullman (1995).

For learning to happen, there must be synaptic connections between the ascending and the descending pathways. From the point of view of the development of the correct connectivity before experience-based learning begins, it is natural to assume that all these weight matrices should be initialized from zero or randomly and should be assumed to be plastic. This is a minimal set of assumptions that do not require any sophisticated genetic and developmental program to wire precisely each ascending chain of neurons with one and only one descending chain.

This raises two important biological constraints that previous works often fall short of: the cross-connection between the two streams need to be (1) initialized randomly and (2) plastic. SGD and existing two-pathway learning algorithms can all be viewed from this perspective of having four plastic or nonplastic connections between two pathways. For example, SGD can be regarded as having a bi-pathway structure with a special nonplastic inter-pathway connection: $V = \bar{V} = I$, and $\bar{W} = W^T$, and only W is plastic during training. This also explains why SGD is the least biologically plausible among these algorithms, as it assumes a lot of fixed and perfect wiring between the downstream and upstream. The feedback alignment has $\bar{V} = V = I$, and $\bar{W}$ is random and nonplastic. The weight mirroring (and KP) algorithm has fixed V and $\bar{V}$ (equal to identity), and a plastic $\bar{W}$. In this sense, our algorithm can be seen as a general version of these learning schemes based on ascending and descending streams. See Table 2 for previous proposals.

5 Conclusion

As we mentioned in the introduction, optimization algorithms that can adjust the inner weights of a deep network are critical for state-of-the-art machine learning. This does not imply, however, that the same statement holds true for the brain.

In this paper we have introduced a biologically plausible learning circuit that self-assembles from random initial connectivity and achieves SGD-like performance through local plasticity rules. The self-assembly requires an ascending and a descending stream – known to exist in all cortices – and initial random connections between them with synapses obeying reasonable plasticity rules. The circuit predicts specific anatomical and physiological features of cortical networks, including heterosynaptic plasticity and a specific synaptic motif.

The circuit we described is only one specific instance within a broad family of circuits with somewhat different synaptic rules and information flows. This paper suggests that biologically plausible learning algorithms can rival established machine learning algorithms while respecting the brain's biophysical constraints. The experimental discovery of such circuits may transform our understanding of learning in both biological and artificial systems.

It is possible to make the regularization term we introduced more biologically plausible, introducing a Winner-Takes-All form of regularization. Such a mechanism can lead to one-to-one synapses between ascending and descending streams. It remains an open problem whether the cortex uses this mechanism. A brief review of the biological plausibility of the associated plasticity rules is presented in Appendix B.

The proposed circuit makes several predictions that are, at least in principle, experimentally verifiable: 1. *Synaptic Motifs*: Reciprocal synaptic connections between ascending and descending pathways, involving four plastic synapses (W, $\bar{W}$, V, $\bar{V}$), are expected to be ubiquitous in cortical networks. 2. *Heterosynaptic Plasticity*: Changes in synaptic strength depend not only on the activity of individual synapses but also on the activity of neighboring synapses, consistent with recent findings in neuroscience. 3. *Self-Assembly*: The circuit assembles itself from random initial connections between the two streams, guided by local plasticity rules. 4. *Separation of Pathways*: The back projections should exhibit different electrophysiological properties, such as nonlinearities depending on synaptic inputs, compared to feedforward pathways, also possibly with different time constants. 5. *Relative size of the streams*: The computation suggests that learning results in better performance when the feedback neurons and connections are more numerous than the feedforward.

Acknowledgments This work was supported by the Center for Brains, Minds and Machines (CBMM), funded by NSF STC award CCF - 1231216, the DARPA Knowledge Management at Scale and Speed (KMASS) program, and the DARPA Mathematics for the DIscovery of ALgorithms and Architectures (DIAL) program.

References

Abel, R. and Ullman, S. Biologically-inspired learning model for instructed vision. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Akrout, M., Wilson, C., Humphreys, P., Lillicrap, T., and Tweed, D. B. Deep learning without weight transport. *Advances in neural information processing systems*, 32, 2019a.

Akrout, M., Wilson, C., Humphreys, P. C., Lillicrap, T. P., and Tweed, D. B. Deep learning without weight transport. *CoRR*, abs/1904.05391, 2019b. URL http://arxiv.org/abs/1904.05391.

Anderson, J. C. and Martin, K. A. The synaptic connections between cortical areas v1 and v2 in macaque monkey. *Journal of Neuroscience*, 29(36):11283–11293, 2009.

Briggs, F. Role of feedback connections in central visual processing. *Annual review of vision science*, 6(1): 313–334, 2020.

Cichy, R. M., Khosla, A., Pantazis, D., Torralba, A., and Oliva, A. Comparison of deep neural networks to spatio-temporal cortical dynamics of human visual object recognition reveals hierarchical correspondence. *Scientific Reports*, 6:27755, 2016.

Coates, A., Ng, A., and Lee, H. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 215–223. JMLR Workshop and Conference Proceedings, 2011.

Güçlü, U. and van Gerven, M. A. Deep neural networks reveal a gradient in the complexity of neural representations across the ventral stream. *Journal of Neuroscience*, 35(27):10005–10014, 2015.

Harvey, C. D. and Svoboda, K. Locally dynamic synaptic learning rules in pyramidal neuron dendrites. *Nature*, 450(7173):1195–1200, 2007.

Huh, M., Mobahi, H., Zhang, R., Cheung, B., Agrawal, P., and Isola, P. The low-rank simplicity bias in deep networks. *arXiv preprint arXiv:2103.10427*, 2021.

Kandel, E. R. and Tauc, L. Heterosynaptic facilitation in neurones of the abdominal ganglion of aplysia depilans. *The Journal of Physiology*, 181(1):1–27, 1965. doi: https://doi.org/10.1113/jphysiol.1965.sp007742. URL https://physoc.onlinelibrary.wiley.com/doi/abs/10.1113/jphysiol.1965.sp007742.

Kolen, J. and Pollack, J. Back-propagation without weight transport. *Proceedings of 1994 IEEE International Conference on Neural Networks (ICNN'94)*, 3:1375–1380, 1994.

Kriegeskorte, N. Deep neural networks: A new framework for modeling biological vision and brain information processing. *Annual Review of Vision Science*, 1:417–446, 2015.

Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images. 2009.

Liao, Q., Leibo, J., and Poggio, T. How important is weight symmetry in backpropagation? In *Proceedings of the AAAI Conference on Artificial Intelligence 2016, arXiv:1510.05067*, 2015.

Lillicrap, T. P., Cownden, D., Tweed, D. B., and Akerman, C. J. Random synaptic feedback weights support error backpropagation for deep learning. *Nature communications*, 7(1):13276, 2016.

Lillicrap, T. P., Santoro, A., Marris, L., Akerman, C. J., and Hinton, G. Backpropagation and the brain. *Nature Reviews Neuroscience*, 21(6):335–346, 2020.

Max, K., Kriener, L., García, G. P., Nowotny, T., Jaras, I., Senn, W., and Petrovici, M. A. Learning efficient backprojections across cortical hierarchies in real time. *Nat. Mac. Intell.*, 6(6):619–630, 2024. URL http://dblp.uni-trier.de/db/journals/natmi/natmi6.html#MaxKGNJSP24.

Metz, L., Maheswaranathan, N., Cheung, B., and Sohl-Dickstein, J. Meta-learning update rules for unsupervised representation learning. *arXiv preprint arXiv:1804.00222*, 2018.

Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., Ng, A. Y., et al. Reading digits in natural images with unsupervised feature learning. In *NIPS workshop on deep learning and unsupervised feature learning*, volume 2011, pp. 4. Granada, 2011.

Nøkland, A. Direct feedback alignment provides learning in deep neural networks. *Advances in neural information processing systems*, 29, 2016.

Nøkland, A. and Eidnes, L. H. Training neural networks with local error signals. In *International conference on machine learning*, pp. 4839–4850. PMLR, 2019.

Roy, O. and Vetterli, M. The effective rank: A measure of effective dimensionality. In *2007 15th European signal processing conference*, pp. 606–610. IEEE, 2007.

Ullman, S. Sequence Seeking and Counter Streams: A Computational Model for Bidirectional Information Flow in the Visual Cortex. *Cerebral Cortex*, 5(1):1–11, 01 1995. ISSN 1047-3211. doi: 10.1093/cercor/5.1.1. URL https://doi.org/10.1093/cercor/5.1.1.

Whittington, J. C. and Bogacz, R. Theories of error back-propagation in the brain. *Trends in Cognitive Sciences*, 23(3):235–250, 2019.

Xiao, H., Rasul, K., and Vollgraf, R. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

Xiao, W., Chen, H., Liao, Q., and Poggio, T. Biologically-plausible learning algorithms can scale to large datasets. *International Conference on Learning Representations (ICLR) 2019, arXiv:1811.03567*, 2018.

Yamins, D. L. and DiCarlo, J. J. Using goal-driven deep learning models to understand sensory cortex. *Nature Neuroscience*, 19(3):356–365, 2016.

Yang, J., Shi, R., and Ni, B. Medmnist classification decathlon: A lightweight automl benchmark for medical image analysis. In *IEEE 18th International Symposium on Biomedical Imaging (ISBI)*, pp. 191–195, 2021.

Yang, J., Shi, R., Wei, D., Liu, Z., Zhao, L., Ke, B., Pfister, H., and Ni, B. Medmnist v2-a large-scale lightweight benchmark for 2d and 3d biomedical image classification. *Scientific Data*, 10(1):41, 2023.

Yann, L. The mnist database of handwritten digits. *R*, 1998.

Ziyin, L., Chuang, I., Galanti, T., and Poggio, T. Formation of representations in neural networks. *arXiv preprint arXiv:2410.03006*, 2024.

A Theory

The following Lemma shows that the interstream connections evolve towards becoming aligned with each other after training.

Lemma 1. *If $\Delta V^l = 0$ and $\Delta \bar{V}^l = 0$ for all l, then*

$$\bar{V}^{l-1} = (V^l)^T. \tag{13}$$

Proof. This is a consequence of the dynamics of $\bar{V}^{l-1}$ and $\bar{V}^l$ being essentially identical. By definition, we have that

$$\Delta V^l = \bar{h}^l (h^l)^T - \gamma V^l, \tag{14}$$

$$\Delta (\bar{V}^{l-1})^T = \bar{h}^l (h^l)^T - \gamma (\bar{V}^{l-1})^T. \tag{15}$$

This implies that

$$\Delta (V^l - (V^{l-1})^T) = -\gamma (V^l - (V^{l-1})^T), \tag{16}$$

which is an exponential decay to zero. This finishes the proof. □

The following definition states what it means for the downstream pathway to be "overparametrized."

Definition 1. *The downstream pathway is said to be overparametrized if for all ℓ and $V^\ell \in \mathbb{R}^{\bar{d}_\ell \times d_\ell}$,*

$$d_\ell \le \bar{d}_\ell. \tag{17}$$

We also say that the matrix V^ℓ is overparametrized if this condition holds.

The following Lemma is a technical step in the theorem proof.

Lemma 2. *If $V^\ell (W^\ell)^T D_u^\ell = D_d^\ell \bar{W}^\ell (\bar{V}^\ell)^T$, then,*

$$P D_d^\ell \bar{W}^\ell (\bar{V}^\ell)^T = D_d^\ell \bar{W}^\ell (\bar{V}^\ell)^T, \tag{18}$$

where $P = ((V^\ell)^T)^+ (V^\ell)^T$ is a projection matrix and A^+ denotes the pseudoinverse of the matrix A.

Proof. We have

$$P D_d^\ell \bar{W}^\ell (\bar{V}^\ell)^T = P V^\ell (W^\ell)^T D_u^\ell = V^\ell (W^\ell)^T D_u^\ell = D_d^\ell \bar{W}^\ell (\bar{V}^\ell)^T. \tag{19}$$

This finishes the proof. □

Now, we explain briefly that the sample-wise gradient of a ReLU network can be written in a specific form. As implied in the main text, the output of a ReLU network can be written in the following form

$$f(x) = W^L D_u^{L-1} \ldots. D^1 W^1 x \tag{20}$$

where D_u^ℓ is the Jacobian of the ReLU activation at the i-th layer.

Let $F(f(x), y)$ be the loss function for the data point x and label y. The above discussion implies that the gradient of F with respect to W can be written as

$$\nabla_{W^\ell} F = \underbrace{(\nabla_f^T F W^L D_u^{L-1} \ldots D^\ell)^T}_{\text{error signal}} \underbrace{((D_u^{\ell-1}) W^{\ell-1} \ldots W_1 x)^T}_{\text{forward signal}}. \tag{21}$$

We will show that the downstream pathway will become self-assembled in a way that it computes the error signal.

Now, we are ready to the prove the main theorem. For the ease of reference, we state the theorem again here.

Theorem 2. *Consider a ReLU neural network with an arbitrary width and depth and with an over-parametrized downstream pathway. If both V^ℓ and $\bar{V}^\ell$ are full-rank for all ℓ, then, for any x such that for all $\ell \in [L]$, $\Delta V^\ell = O(\epsilon)$ and $\Delta \bar{V}^\ell = O(\epsilon)$,*

$$\Delta_{\rm SAL} W^\ell = H \Delta_{\rm sgd} W^\ell - \gamma W^\ell + O(\epsilon), \tag{22}$$

for a positive definite matrix $H = \bar{V}^\ell (\bar{V}^\ell)^T$, and $\Delta_{\rm sgd} W^\ell = -\nabla_W F$ is the SGD update.

Proof. By definition of the algorithm,

$$\Delta V^\ell = \bar{h}^\ell (h^\ell)^T - \gamma V^\ell = D_d^\ell \bar{W}^\ell \bar{h}^{\ell+1} (h^\ell)^T - \gamma V^\ell, \tag{23}$$

$$\Delta (\bar{V}^\ell)^T = \bar{h}^{\ell+1} (h^{\ell+1})^T - \gamma (\bar{V}^\ell)^T = \bar{h}^{\ell+1} (h^\ell)^T (W^\ell)^T D_u^\ell - \gamma (\bar{V}^\ell)^T. \tag{24}$$

If $\Delta V^\ell = O(\epsilon)$ and $\Delta \bar{V}^\ell = O(\epsilon)$, we have

$$(\Delta V^\ell)(W^\ell)^T D_u^\ell = O(\epsilon), \tag{25}$$

$$D_d^\ell \bar{W}^\ell \Delta (\bar{V}^\ell)^T = O(\epsilon). \tag{26}$$

Thus, we have

$$V^\ell (W^\ell)^T D_u^\ell = D_d^\ell \bar{W}^\ell (\bar{V}^\ell)^T + O(\epsilon). \tag{27}$$

Because V^ℓ is full-rank and overparametrized, $G_\ell := (V^\ell)^T V^\ell$ is an invertible matrix. We can thus multiply $G^{-1} (V^\ell)^T$ on the left to obtain that

$$(W^\ell)^T D_u^\ell = G_\ell^{-1} (V^\ell)^T D_d^\ell \bar{W}^\ell (\bar{V}^\ell)^T + O(\epsilon). \tag{28}$$

Now, consider the matrix product between different layers

$$(D_u^\ell W^l D_u^{\ell-1} W^{l-1})^T = (V^{l-1})^{-1} D_d^{\ell-1} \bar{W}^{l-1} (\bar{V}^{l-1})^T G_\ell^{-1} (V^\ell)^T D_d^\ell \bar{W}^l (\bar{V}^l)^T + O(\epsilon). \tag{29}$$

However, by Lemma 1, at stationarity, we must have

$$\bar{V}^{l-1} = (V^l)^T, \tag{30}$$

and so

$$(\bar{V}^{l-1})^T G_\ell^{-1} (V^\ell)^T = (V^l)^T G_\ell^{-1} (V^\ell)^T = P^2 = P, \tag{31}$$

where $P = ((V^\ell)^T)^+ (V^\ell)^T$ is a projection matrix, and the + superscript denotes the pseudoinverse. Using Lemma 2, we have that

$$P D_d^\ell \bar{W}^l (\bar{V}^l)^T = D_d^\ell \bar{W}^l (\bar{V}^l)^T, \tag{32}$$

which implies that

$$(D_u^\ell W^l D_u^{\ell-1} W^{l-1})^T = (V^{l-1})^{-1} D_d^{\ell-1} \bar{W}^{l-1} D_d^\ell \bar{W}^l (\bar{V}^l)^T + O(2\epsilon). \tag{33}$$

Namely, the intermediate projections P have no effect. This argument can be applied repetitively to show that

$$(D_u^\ell W^l ... D_u^{\ell-n} W^{l-n})^T = (V^{l-n})^{-1} D_d^{\ell-n} \bar{W}^{l-n} ... D_d^\ell \bar{W}^l (\bar{V}^l)^T + O((\ell-n)\epsilon) \tag{34}$$

for any l and $n < l$. This means that the backward net has essentially become the transpose of the forward net.

Now, by the definition of our algorithm, we have

$$\bar{V}^l \bar{h}^{l+1} (h^l)^T = \bar{V}^l D_d^{\ell+1} \bar{W}^{l+1} ... D_d^{L-1} \bar{W}^{L-1} \bar{h}^L (h^l)^T \tag{35}$$

$$= \bar{V}^l V^{l+1} (D_u^\ell W^l ... D_u^{L-1} W^{L-1})^T ((\bar{V}^l)^T)^{-1} \bar{h}^L (h^l)^T \tag{36}$$

$$= \bar{V}^l (\bar{V}^l)^T (D_u^\ell W^l ... D_u^{L-1} W^{L-1})^T ((\bar{V}^l)^T)^{-1} \bar{h}^L (h^l)^T \tag{37}$$

$$= H (D_u^\ell W^l ... D_u^{L-1} W^{L-1})^T \bar{h}^L (h^l)^T + O(\ell\epsilon), \tag{38}$$

where $H = \bar{V}^l (\bar{V}^l)^T$ is positive definite, we have used the definition $\bar{V}^L = I$. By definition,$\bar{h}^L$ is the label gradient; this update rule is thus a linear transformation of the GD. The proof is complete.

□

Proposition 1. *If H is positive semi-definite, and if we update the parameters by*

$$\dot{\theta} = -H\nabla_\theta L, \tag{39}$$

then, $L(\theta(t))$ is a monotonic decreasing function of t for any initialization $\theta(0)$. If H is full-rank, then this dynamics has the same stationary points as GD.

Proof. For the first part, consider the time evolution of

$$\dot{L} = (\nabla L)^T \dot{\theta} = (\nabla L)^T H \nabla_\theta L \leq 0. \tag{40}$$

For the second part, if H is full-rank, $\dot{\theta}$ is zero only when $\nabla L = 0$. Therefore, the algorithm has the same stationary points as GD.

□

B Biological Evidence for Synaptic Plasticity Rules

The learning rules corresponding to Equations (5)-(8) are not strictly Hebb. In fact, the classical Hebb rule is an example of homosynaptic plasticity, whereas the proposed algorithm requires the plasticity of neighboring synapses through heterosynaptic interactions. One of the first examples of heterosynaptic facilitation is in Kandel & Tauc (1965). There is empirical evidence that reinforcement of a synapse based on its own input and nearby synaptic input can occur independently of large postsynaptic depolarization, often through localized biochemical signaling rather than direct electrical depolarization of the postsynaptic cell. In this case, synaptic plasticity is influenced by the activity of neighboring synapses without necessitating a strong postsynaptic action potential. A possible mechanism is Spike-Timing-Dependent Plasticity (STDP) with Local Modulation: in some forms of STDP, nearby synapses on the same dendrite can influence plasticity based on the timing of inputs. For example, the timing of presynaptic spikes at two nearby excitatory synapses can trigger local signaling cascades within the dendritic segment that reinforce or weaken synaptic strength without requiring a large postsynaptic potential. Another is calcium-dependent modulation: local increases in calcium concentration can drive synaptic reinforcement or weakening in a spatially restricted manner. When one synapse is active, it can cause a modest calcium influx that can extend to nearby synapses without significantly depolarizing the postsynaptic neuron. This local calcium increase can activate signaling pathways, such as CaMKII or other kinases, that modify the strength of nearby synapses proportionally to their own activity levels and those of neighboring inputs. A third possible mechanism is retrograde signaling, which involves retrograde messengers (e.g., endocannabinoids or nitric oxide) that are released by the postsynaptic neuron in response to local synaptic activity and can act on presynaptic terminals. These messengers can diffuse locally and affect only the nearby synapses that are also active, reinforcing or weakening them proportionally to their individual input activities. An interesting case study is by Harvey & Svoboda (2007) on synaptic tagging in hippocampal neurons. It demonstrated that synapses can interact biochemically on a local level. When a strong input induces LTP at one synapse, it can create a "tag" that allows nearby synapses to capture plasticity-related proteins and reinforce their strength based on their activity, even if they don't independently trigger large postsynaptic depolarization. These mechanisms allow synaptic connections to encode complex input patterns across local networks, effectively performing computations that go beyond simple, strict Hebbian principles[4].

In the main text, we have mainly focused on the simplest type of regularization. A more biologically plausible regularization is perhaps a Winner-Take-All regularization, and this could be a direction of future research.

Biological evidence for competition among incoming synapses on a single neuron (or dendrite) of a single neuron. Research does suggest a competitive process among synapses on the same dendritic branch, particularly when multiple inputs converge on a single location. This competition is thought to be governed

[4]There are a few different possibilities of how to implement them biologically. One special detail one needs to pay attention to is the time ordering of firing.

by mechanisms of synaptic plasticity, where synapses that are frequently active and contribute to the cell's firing are strengthened, while less active ones may weaken or even be pruned. One well-documented example of this competition is seen in activity-dependent plasticity processes, such as Hebbian plasticity and synaptic tagging and capture. When neighboring synapses are active at the same time, they may compete for limited resources, like proteins that help stabilize and strengthen synaptic connections. This often results in the selective strengthening of certain synapses and the weakening of others, a process referred to as heterosynaptic plasticity. In this way, some synapses are favored to persist, while others may diminish over time if they fail to "compete" effectively. This competitive mechanism is believed to play a crucial role in refining neural circuits, helping to maintain efficient and effective synaptic networks by eliminating redundant or less useful connections, especially during development and learning.

Biological evidence for competition among synapses from a single neuron (or axon) onto different neurons. There is evidence supporting the idea of presynaptic competition between synapses originating from the same neuron when they connect to different target neurons. One of the most studied examples of presynaptic competition is in motor neurons and their connections to muscle fibers, especially during development. Motor neurons initially form multiple synapses on various muscle fibers, but over time, a pruning process occurs, often resulting in a single, stable connection with one muscle fiber. This phenomenon, known as *synaptic pruning*, is a process where less active synapses are eliminated while only the more effective and frequently active connections are maintained and strengthened. This presynaptic competition is driven by neuronal activity and molecular signals. Research on long-term synaptic plasticity (LTP and LTD) suggests that competing synapses can affect each other's likelihood of survival, even when they connect to different cells. Synchronized activity at certain synapses tends to stabilize those connections, while less synchronized or less active synapses are gradually weakened and may be removed. Presynaptic competition is essential not only for refining neural networks but also for developing functional connections between neurons. It ensures that only the most efficient and relevant connections are maintained, optimizing neural circuitry for effective function.

C Experimental Setup

All experiments are conducted with PyTorch on one NVIDIA A100 80GB GPU. The batch size is set to 256.

Datasets. We use six datasets to evaluate our method. *CIFAR-10* (Krizhevsky et al., 2009) consists of 60,000 color images in 10 different classes, with each image having a resolution of 32x32 pixels. The dataset is divided into 50,000 training images and 10,000 testing images. The classes are mutually exclusive and include common objects such as airplanes, cars, cats, and dogs.

The *MNIST* (Yann, 1998) contains 70,000 grayscale images of handwritten digits (0-9), each with a resolution of 28x28 pixels. The dataset is divided into 60,000 training images and 10,000 testing images. MNIST is known for its simplicity and has been a standard dataset for testing machine learning algorithms.

Fashion MNIST (Xiao et al., 2017) contains 70,000 grayscale images of clothing items like T-shirts, trousers, shoes, and bags, with each image having a resolution of 28x28 pixels. The dataset is also split into 60,000 training and 10,000 testing images. Fashion MNIST is considered more challenging than MNIST due to the variability in the visual features of clothing items.

The *SVHN* (Netzer et al., 2011) contains over 600,000 color images of digits (0-9), each with a resolution of 32x32 pixels. The dataset is divided into a training set of 73,257 images, a testing set of 26,032 images, and an additional 531,131 images for extra training. SVHN is challenging due to the varying digit sizes, orientations, and complex backgrounds.

The ChestMNIST (Yang et al., 2021, 2023) comprises chest X-ray images for multi-label classification tasks. Derived from the NIH ChestX-ray14 dataset, it includes images annotated with 14 different pathological labels, such as pneumonia and emphysema. It is widely used in medical imaging studies to develop and evaluate machine learning models capable of diagnosing multiple conditions from X-ray images.

PathMNIST (Yang et al., 2021, 2023) is a pathology image dataset designed for multi-class classification tasks. It originates from the NCT-CRC-HE-100K dataset and includes histological images of nine different

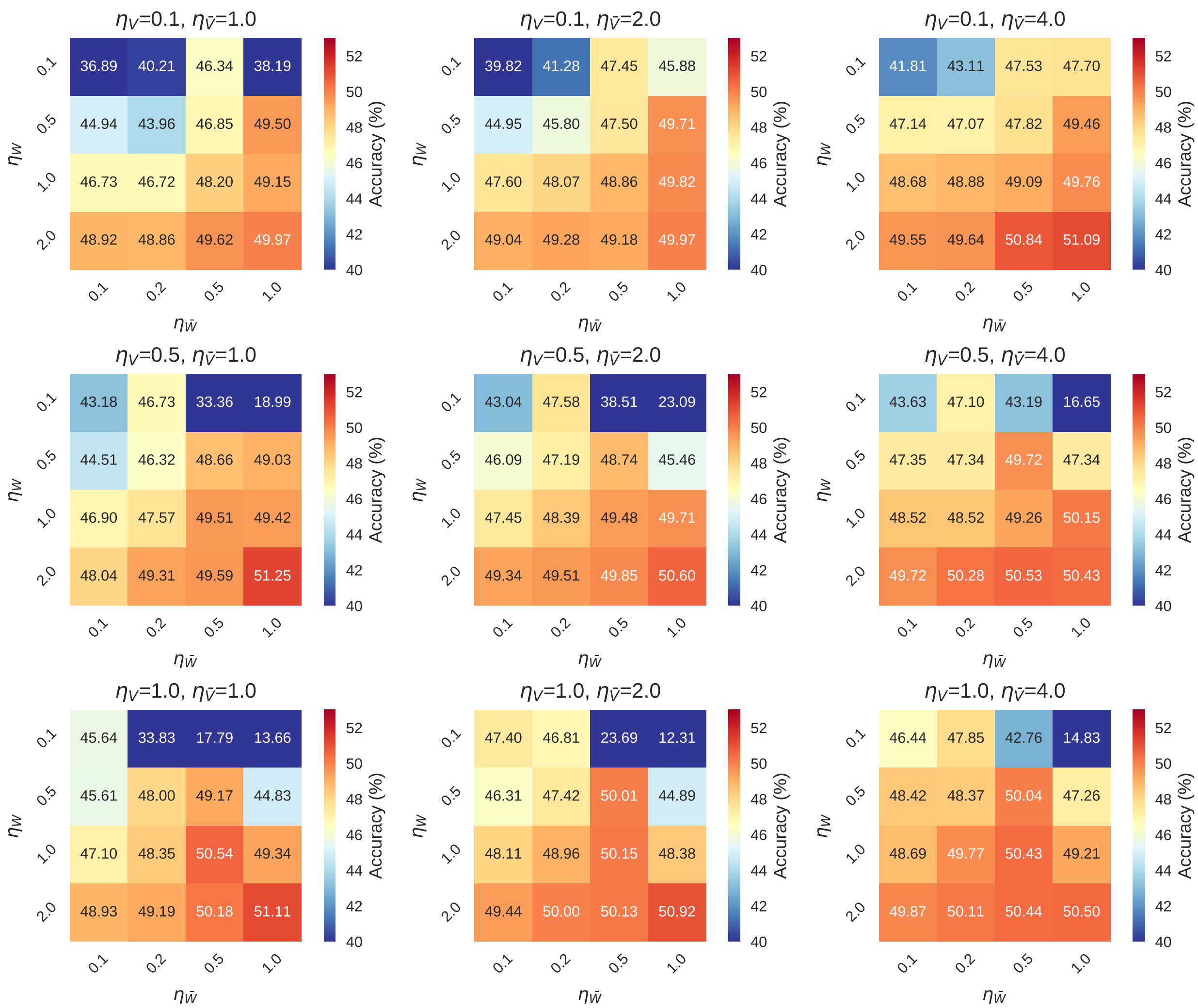

Figure 5: Performance of a four-layer MLP for different choices of learning rates. Interestingly, the best performance is achieved when η_V is the smallest.

types of tissue from colorectal cancer samples, such as adenocarcinoma and lymph node tissue. This dataset is used primarily for training models to differentiate between various cancerous tissue types based on their histopathological features.

STL-10 (Coates et al., 2011) consists of 13,000 labeled images across 10 different classes and 100,000 unlabeled images. Each image has a resolution of 96x96 pixels, which is significantly larger than CIFAR-10 images, making the dataset more challenging. The dataset has a training set of 5,000 labeled images and a test set of 8,000 labeled images.

Baselines. We compare our method with four baselines, including SGD and three biologically-motivated algorithms.

Stochastic Gradient Descent (SGD) is the standard optimization method that updates parameters iteratively using a subset of data to minimize the objective function.

Weight Mirroring (Akrout et al., 2019a) is a technique generally used in neural networks to enforce symmetry between forward and backward passes. This can be implemented by mirroring the weights used in the forward pass for use in the backward pass, which often helps with learning efficiency and stability.

Feedback Alignment (Lillicrap et al., 2016) offers an alternative to traditional backpropagation by using random, fixed weights instead of transposed forward-pass weights to propagate error signals.

Kolen-Pollack (KP) algorithm (Kolen & Pollack, 1994) enables synaptic weight adjustments in neural networks without weight transport, simplifying the architecture and enhancing learning in large networks.

D Pseudo Code

See Algorithm 1.

Algorithm 1 Self-Assembling Learning Algorithm

Initialize: For each layer $\ell = 1$ to L, initialize weight matrices W^ℓ, V^ℓ, $\bar{W}^\ell$, and $\bar{V}^\ell$
for each training example **do**
  Forward Pass (Upstream Pathway):
  $h^1 \leftarrow$ input data x
  for $\ell = 1$ to $L-1$ **do**
    $h^{\ell+1} \leftarrow D(W^\ell h^\ell)$
  end for
  Compute Loss:
  Compute loss F based on output h^L and target y
  Initialize Error Signal:
  $\bar{h}^L \leftarrow \epsilon(h^L, y)$ (e.g., $\bar{h}^{L+1} = -\frac{\partial F}{\partial h^{L+1}}$)
  Backward Pass (Downstream Pathway):
  for $\ell = L-1$ down to 1 **do**
    $\bar{h}^\ell \leftarrow D(\bar{W}^\ell \bar{h}^{\ell+1})$
  end for
  Update Synaptic Weights:
  for $\ell = 1$ to L **do**
    $\Delta W^\ell \leftarrow \bar{V}^\ell \bar{h}^{\ell+1} (h^\ell)^\top - \gamma W^\ell$
    $\Delta V^\ell \leftarrow \bar{W}^\ell \bar{h}^{\ell+1} (h^\ell)^\top - \gamma V^\ell$
    $\Delta \bar{W}^\ell \leftarrow \left(\bar{h}^{\ell+1} (h^\ell)^\top (V^\ell)^\top - \gamma (\bar{W}^\ell)^\top\right)^\top$
    $\Delta \bar{V}^\ell \leftarrow \left(\bar{h}^{\ell+1} (h^\ell)^\top (W^\ell)^\top - \gamma (\bar{V}^\ell)^\top\right)^\top$
    $W^\ell \leftarrow W^\ell + \eta \Delta W^\ell$
    $V^\ell \leftarrow V^\ell + \eta \Delta V^\ell$
    $\bar{W}^\ell \leftarrow \bar{W}^\ell + \eta \Delta \bar{W}^\ell$
    $\bar{V}^\ell \leftarrow \bar{V}^\ell + \eta \Delta \bar{V}^\ell$
  end for
end for