CONSTRUCTING BIOLOGICALLY CONSTRAINED RNNS VIA DALE'S BACKPROP AND TOPOLOGICALLY-INFORMED PRUNING

Aishwarya H. Balwani*
School of Electrical & Computer Engineering
Georgia Institute of Technology
abalwani6@gatech.edu

Alex Q. Wang
Computational Science and Engineering Program
Georgia Institute of Technology
alexyang@gatech.edu

Farzaneh Najafi
School of Biological Sciences
Georgia Institute of Technology
fnajafi3@gatech.edu

Hannah Choi*
School of Mathematics
Georgia Institute of Technology
hannahch@gatech.edu

ABSTRACT

Recurrent neural networks (RNNs) have emerged as a prominent tool for modeling cortical function, and yet their conventional architecture is lacking in physiological and anatomical fidelity. In particular, these models often fail to incorporate two crucial biological constraints: i) Dale's law, i.e., sign constraints that preserve the "type" of projections from individual neurons, and ii) Structured connectivity motifs, i.e., highly sparse yet defined connections amongst various neuronal populations. Both constraints are known to impair learning performance in artificial neural networks, especially when trained to perform complicated tasks; but as modern experimental methodologies allow us to record from diverse neuronal populations spanning multiple brain regions, using RNN models to study neuronal interactions without incorporating these fundamental biological properties raises questions regarding the validity of the insights gleaned from them. To address these concerns, our work develops methods that let us train RNNs which respect Dale's law whilst simultaneously maintaining a specific sparse connectivity pattern across the entire network. We provide mathematical grounding and guarantees for our approaches incorporating both types of constraints, and show empirically that our models match the performance of RNNs trained without any constraints. Finally, we demonstrate the utility of our methods for inferring multi-regional interactions by training RNN models of the cortical network to reconstruct 2-photon calcium imaging data during visual behaviour in mice, whilst enforcing data-driven, cell-type specific connectivity constraints between various neuronal populations spread across multiple cortical layers and brain areas. In doing so, we find that the interactions inferred by our model corroborate experimental findings in agreement with the theory of predictive coding, thus validating the applicability of our methods.

1 Introduction

Recent years have seen the increasing adoption of artificial neural networks (ANNs) for modeling brain function both mechanistically and algorithmically [1, 2, 3, 4, 5]. In particular, recurrent neural networks (RNNs) are now an established tool in computational neuroscience research [6, 7], being used to study neuronal computation at varying scales ranging from subsets of neurons sampled across a single brain region, two interacting regions [8, 9, 10], and even numerous populations spread across multiple interacting brain regions [11, 12]. By way of either reproducing desired behaviours [13, 14, 15], task-driven responses [16, 17, 18], or by fitting to recorded neural data [19, 20], RNNs have been shown to successfully capture latent dynamics typical of neural circuits [21, 22] thus making them especially useful for modeling phenomena observed across the cortex. The degree to which ANNs can effectively approximate neural data however depends on two key considerations: i) The literature suggests a direct correlation between the ability of an ANN to learn well on a task and the extent to which its behaviour and learnt representations match real neural data [23, 24, 25], and ii) More biologically realistic architectures aid in the learning of representations that better match

*Corresponding authors

real neuronal data [26, 27, 28, 29]. These factors make it essential that the ability of the ANNs to learn and represent a wide range of function classes be unrestricted, that their training not suffer from hindrances, and their construction respect important anatomical principles, especially when being used as models of the brain to study neuroscientific phenomena [30, 31].

Of the various discrepancies conventional RNN-based neuroscientific models have with their biological counterparts (Fig. 1.A1) [32], two notable ones are their lack of adherence with Dale's principle [33], i.e., the phenomena that restricts a presynaptic neuron to have exclusively either an excitatory or inhibitory effect on all its postsynaptic connections, and structured sparse connectivity amongst neuronal populations, a fundamental feature of brain organization observed across various species [34, 35, 36] and brain regions [37]. Unfortunately, directly incorporating these constraints oftentimes decreases the capacity and flexibility of the network to fit the training data, which leads to a drop in learning performance [38, 39]. While there has been active research towards addressing these issues in both the machine learning (ML) [40, 41, 42, 43] and computational neuroscience communities [38, 44, 45, 46, 47, 48, 49], these efforts have mostly been made to include these constraints into the network structure individually, rather than in conjunction as one would see in biological brains [50]. Subsequently, there remains a need for ways to construct sparse, sign-constrained deep neural networks which can also achieve performance levels comparable to conventional ANNs.

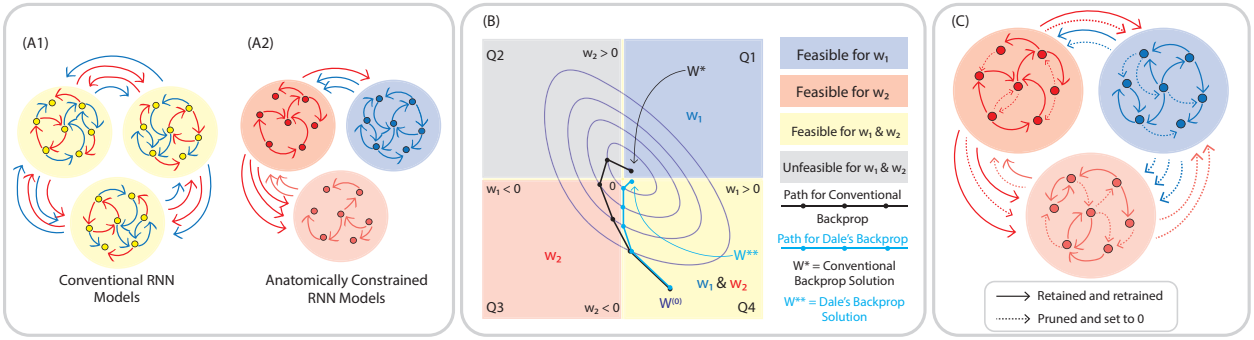

Figure 1: *Schematic for Constructing Biologically Constrained RNN Models.* **(A)** Illustration of conventional vs. biologically constrained RNN models (A1 vs. A2). Conventional RNNs consist of general purpose neurons that project a mix of excitatory and inhibitory signals, with no specific connectivity structure within or across populations. Biologically constrained RNNs restrict populations of neurons to be either strictly excitatory (red) or inhibitory (blue), with anatomically-informed connectivity motifs both within and across populations. **(B)** Optimization in parameter space when training with conventional backpropagation (black) vs. Dale's backprop (blue). Purple contours represent level sets for the positions the algorithms take in parameter space at different time steps. **(C)** Enforcing anatomically-consistent connectivity motifs. Dashed lines represent connections that are set to 0 during the pruning process. Solid lines represent connections that are retained post-pruning.

Our work therefore introduces methods that allow us to easily incorporate both neuronal sign constraints and sparse connectivity motifs into the conventional backpropagation-based RNN training pipeline (Fig. 1.A2). Specifically, we first train a dense network that respects a pre-determined set of sign constraints via a modified version of standard backpropagation [51] which we call *Dale's backpropagation* (Fig. 1.B), after which we prune away weights using a probabilistic pruning rule we call *top-prob pruning* (Fig. 1.C), to achieve a target connectivity pattern. Finally, we retrain the sparse sub-network retained post pruning once again with Dale's backprop. Importantly, both of our methods are mathematically grounded and follow rigorous principles. With Dale's backprop, we provide theoretical guarantees on the linear convergence of the algorithm under specific conditions, ensuring that the training respects anatomical constraints while achieving optimal learning performance. Our pruning rule is motivated by topological principles, particularly the preservation of high-magnitude weights that contribute to the network's zeroth-order connectivity structure, thus enhancing the functional and anatomical plausibility of the model.

Besides being convenient, easily implementable, and scalable using standard ML packages, our approach also aligns with the biological processes of synaptic development and refinement. Given that synaptic connections initially form abundantly, with many connections later being pruned based on activity and functional relevance, by first learning a dense set of weights with Dale's backprop, the network can capture a rich set of connections that adhere to Dale's law, reflecting excitatory and inhibitory roles at a fundamental level. Subsequent application of top-prob pruning mirrors the refinement phase, where weaker, less functionally critical synapses are eliminated, retaining only the most effective pathways. This pruning rule not only emphasizes synaptic efficacy [52] — preserving stronger synapses — but also

adheres to principles of synaptic scaling [53] in the retraining phase[2] by maintaining a balanced level of activity within the network. Overall, this process of initial dense learning followed by selective refinement embodies how biological systems evolve and ensures computational efficiency by optimizing for both anatomical and functional plausibility [54, 55].

We demonstrate the suitability of our methods for studying neuroscientific data by applying them to RNNs trained to fit a two-photon calcium imaging dataset exploring multi-regional interactions that underlie visual behavior in mice when performing a change detection task [56]. We find that our models successfully recapitulate both long- and short-timescale interactions among neuronal populations, capturing transient dynamics as well as sustained signals that are critical for complex perceptual processing. Moreover, our model outputs align with the predictive coding hypothesis [57], as they reflect anticipatory and feedback-driven patterns observed experimentally, suggesting that our approach is well-suited to modeling the layered processing of sensory information in the brain.

Taken together, our results on synthetic and real-world datasets indicate that our methods offer a robust framework for fitting and modeling neural dynamics in a biologically faithful manner. By capturing both anatomically realistic connectivity patterns and functional interactions, our approach provides a set of powerful tools for understanding the complex, hierarchical processing of information across different cell-types, populations, and brain areas. These tools subsequently enable models to better reflect anatomical structures, thereby imparting greater confidence in their findings and enhancing the alignment between RNNs and real neural circuitry.

2 Training Networks with Dale's Backpropagation

In this section, we introduce our sign-constrained learning rule, **Dale's backpropagation** and validate its performance. In particular, we provide intuition for the algorithm, describe it in detail, present the statements of the theoretical analyses performed (i.e., convergence guarantees and error bounds), and finally provide empirical results demonstrating its utility on a set of neuroscience-inspired and ML tasks.

2.1 Dale's Backpropagation: Algorithm

Dale's backpropagation enforces Dale's principle by integrating sign constraints into the conventional backpropagation process. Specifically, it employs a projection step (similar to that of projected gradient descent) on the learnt parameters at every iteration to ensure that the weights remain non-negative for excitatory neurons and non-positive for inhibitory neurons, thus adhering to biological constraints (Fig. 1.B).

Consider the typical Elman RNN [58], whose hidden states h_t at time t are updated as per the rule

$$h_t = \phi(W_{hi}x_t + b_{hi} + W_{hh}h_{t-1} + b_{hh}) \tag{1}$$

where ϕ is the non-linear activation function, W_{hh} is the recurrent weight matrix, and W_{hi} is the projection matrix that acts on inputs x_t. Biases corresponding to the input and hidden states are denoted as b_{hi}, b_{hh} respectively.

When h_t are non-negative, respecting Dale's law simplifies to constraining the recurrent weights W such that if i is the pre-synaptic neuron and j is the post-synaptic neuron,

$$W = \begin{cases} W_{ji} \geq 0 & \text{if neuron } i \text{ is excitatory.} \\ W_{ji} \leq 0 & \text{if neuron } i \text{ is inhibitory.} \end{cases}$$

At initialization, the recurrent matrix W can satisfy the sign constraints by construction. However, given that standard gradient descent-based backpropagation update

$$W^{(i+1)} = W^{(i)} - \eta \nabla \ell \left(W^{(i)} \right) \tag{2}$$

with the step size η and loss function ℓ, there is no guarantee that the updated weights $W^{(i+1)}$ at the next iteration will satisfy the sign constraints set by Dale's law even if they are respected by the matrix $W^{(i)}$ at iteration i.

We note however that our sign constraints always form a convex set [59], enabling us to adapt any gradient-based optimization scheme (e.g., SGD, ADAM, RMSprop, etc.) into its projected version [60]. Hence, after the standard

[2]During the retraining phase, the synaptic strengths within the network are dynamically adjusted, effectively rescaling the remaining connections to maintain overall network activity and prevent neuron underutilization. This mirrors the biological process of synaptic scaling, ensuring that the network retains its capacity to learn and generalize despite the reduced number of connections.

backprop update at every iteration, we project the weights onto their feasible set – the orthant in parameter space where the sign constraints of all individual synaptic weights are met. Mathematically, this new update rule can be expressed as

$$\begin{aligned} W_D^{(i)} &= \mathcal{P}_{\mathcal{C}}\left(W^{(i)}\right) \\ &= \mathcal{P}_{\mathcal{C}}\left(W_D^{(i-1)} - \eta \nabla \ell(W_D)^{(i-1)}\right) \\ &= \max\left(0, W_{[N^+]}^{(i)}\right) \oplus \min\left(0, W_{[N^-]}^{(i)}\right) \end{aligned} \tag{3}$$

where $W_D^{(i)}$ represents the weight matrix that satisfies Dale's law at iteration i and $\mathcal{P}_{\mathcal{C}}$ denotes the projection operator. Weight subsets corresponding to excitatory and inhibitory neurons are denoted by $[N^+]$ and $[N^-]$ respectively. The full derivation of the update is provided in Appendix 6.1. The explicit algorithm for the update under gradient descent as the optimizer is shown in Appendix 6.2.

Moreover, this projection onto the feasible set has both, a simple interpretation and implementation. At every iteration, weights that violate their assigned sign constraints are set to zero, while those that comply are retained at their updated values. The projection itself can be efficiently implemented by multiplying the weights with a binary mask after each update. This flexibility makes it easy to apply our method across various architectures, seamlessly integrating sign constraints within standard backpropagation frameworks.

Consequently, for a single-layer RNN with N neurons of which N^+ are excitatory and N^- are inhibitory, our entire algorithm for Dale's backpropagation can be summarized as follows

---

Algorithm 1: Dale's Backpropagation

---

1. **Initialize** $W_D^{(0)} = W^{(0)} \in \mathbb{R}^{N \times N}$ such that $W^{(0)} = W_{[N^+]}^{(0)} \oplus W_{[N^-]}^{(0)}$.
   $W_{[N^+]}^{(0)}, W_{[N^-]}^{(0)}$ represent weights from the excitatory and inhibitory neurons respectively.
2. Enforce Dale's law by setting $W_{[N^+]}^{(0)} \in \mathbb{R}_{\geq 0}^{N \times N^+}$ and $W_{[N^-]}^{(0)} \in \mathbb{R}_{\leq 0}^{N \times N^-}$.
3. Sample $W_{[N^+]}^{(0)}$ from $U\left[0, \frac{1}{\sqrt{N}}\right]$ and $W_{[N^-]}^{(0)}$ from $U\left[\frac{-1}{\sqrt{N}}, 0\right]$.
4. Initialize $h_0 \leftarrow \vec{0}$.
5. **For** each time step t, compute and threshold the hidden state h_t to be non-negative as:
$$h_t^+ = (\phi(W_{hi}x_t + b_{hi} + W_D h_{t-1}^+ + b_{hh}))^+$$
6. **For** each iteration i:
   (a) Compute $W^{(i+1)}$ using standard backpropagation.
   (b) Update weights by setting:
$$W_D^{(i+1)} = \max\left(0, W_{[N^+]}^{(i+1)}\right) \oplus \min\left(0, W_{[N^-]}^{(i+1)}\right)$$

We note that our initialization scheme relates closely to popularly used schemes such as Glorot [61] and He [62] initialization. In particular the scheme aligns with (and is equivalent to when the number of excitatory and inhibitory neurons are equal) common weight initialization practices in RNNs where weights are initialized with zero mean and scaled variance – often using the distribution $U\left[\frac{-1}{\sqrt{N}}, \frac{1}{\sqrt{N}}\right]$ – to maintain consistent activation variances, facilitating effective training and convergence [38, 63].

We also observe that thresholding h_t to be non-negative mimics the behavior of biological neurons, in that neuronal firing cannot be negative. Since our goal is to always ensure that h_t is non-negative, we can also increase the threshold to any value greater than 0, depending on the application. Furthermore, it is of interest that previous works were able to enforce sign constraints with FORCE learning [13] in spiking neural networks (SNNs) [47] using an update rule that is the same as ours, but since h_t is non-negative by definition in an SNN, this is not a step that they explicitly needed to incorporate into their training algorithm. Additionally, we can show that despite their different formulations, Dale's backpropagation reinforces an overlapping set of synaptic connections as Hebbian learning [64], thus preserving key aspects of biologically plausible learning dynamics (Appendix 7).

Finally, it is worth mentioning that the Dale's backpropagation update is guaranteed to find the weights W_D that are the closest projection of W under sign constraints **(Theorem 5)**, with respect to the Frobenius norm (Appendix 6.3). Other methods that are similar to ours ideologically choose to enforce Dale's law by always using a ReLU activation function and consequently constraining the post-synaptic weights to be positive (or negative) for excitatory (or inhibitory) neurons [50, 45] by multiplying these non-negative weights with a mask comprised of ± 1. While this method (which we call *rectified backprop*) is equivalent to ours in the case of weights projecting from excitatory neurons, it essentially reverses the sets of weights that are kept vs. zeroed out in the case of inhibitory neurons. In terms of the final update, the new weights learnt by rectified backprop therefore are much farther away than the one originally computed by conventional backpropagation **(Corollary 6)**.

2.2 Dale's backpropagation: Theoretical Results

We now present our key mathematical analyses for Dale's backprop when it utilizes gradient descent as its optimizer. First, we derive the rate of convergence of the algorithm under the assumption of restricted optima, i.e., when we can assume that the optimal set of parameters also have the same sign pattern as those imposed. Second, we quantify the differences between Dale's backprop and standard backpropagation, both in terms of the weights learnt and the final solutions obtained. Together, these results establish a solid theoretical foundation for Dale's backprop, demonstrating its ability to learn effectively and efficiently, thus validating its use in modeling neural data.

2.2.1 Analyzing convergence of Dale's backpropagation under the restricted optimum assumption

We start by examining the behavior of Dale's backpropagation algorithm under the assumption that the optimal set of parameters for a task shares the same sign pattern as the one imposed – a condition we refer to as the *restricted optima assumption* – and show that despite having to learn with constraints, under this assumption, Dale's backprop converges linearly to the optimal solution **(Theorem 2)**. Biologically, this assumption mirrors the idea that the arrangement of excitatory and inhibitory neurons in the network is optimized for such tasks.

Proving this theorem relies on the geometric observation that the restricted optima assumption ensures that the globally optimal set of weights (W^*) lies within the same orthant as our point of initialization $\left(W^{(0)}\right)$. We subsequently prove optimal sign pattern preservation **(Lemma 1)**, which guarantees that every backpropagation iteration never leaves this orthant, implying the signs of the weights remain constant throughout the optimization process. As a result, Dale's backpropagation behaves identically to unconstrained gradient descent within this orthant, making the projection step redundant since the optimization path does not approach the boundaries of the orthant. Consequently, the algorithm can take the most direct path to the optimum without any detours induced by constraint enforcement, allowing it to achieve a linear convergence rate under the Polyak-Łojasiewicz condition.

The statements of our results are presented below, with full proofs deferred to the supplement (Appendix 8.1).

Lemma 1 (Optimal sign pattern preservation)**.** *Let the vector of learnt weights be $W \in \mathbb{R}^n$ with the components w_j, where $j \in \{1, 2, ..., n\}$. Let L be the Lipschitz constant for the gradients $\nabla \ell(W)$, where ℓ is a loss function. Given a gradient descent-based, component-wise sign-preserving learning rule that uses the projection operator $\mathcal{P}_{\mathcal{C}} : \mathbb{R}^n \mapsto \mathbb{R}^n$ defined as*

$$\mathcal{P}_{\mathcal{C}}(w_j) = \begin{cases} w_j & \text{if } sign(z_j) = sign\left(w_j\right) \\ \\ 0 & \text{if } sign(z_j) \neq sign\left(w_j\right) \end{cases}$$

where $z_j = w_j - \frac{1}{L}\nabla \ell\left(w_j\right), sign(z_j) = \frac{z_j}{|z_j|}$ for $z_j \neq 0$, and $sign(0) = 0$. If $sign(W^*) = sign\left(W^{(0)}\right)$ where W^* are the set of weights that can achieve the optimal loss on ℓ, it holds that for any iteration i of regular gradient descent

$$sign\left(W^{(i)}\right) = sign\left(W^{(0)}\right) = sign\left(W^*\right) \ \forall i \in \mathbb{N}, \text{ and } \mathcal{P}_{\mathcal{C}}(w_j) = w_j \text{ for } j \in \{1, 2, ..., n\}.$$

Theorem 2 (Convergence of Dale's Backpropagation)**.** *Let ℓ be a loss function satisfying the $\mu-$Polyak-Łojasiewicz condition, with gradients that are L-Lipschitz such that $L \geq \mu > 0$. Consider the sequence of weights $\{W_D^{(i)}\}$ generated according to the Dale's backpropagation update, with a step size of $\frac{1}{L}$. Given an optimal loss $\ell^* = \ell(W^*) = \operatorname{argmin} \ell(W_D)$ where W^* has the same sign pattern as all $W_D^{(i)}$ and a specific error $\varepsilon > 0$, it holds for iteration i that*

$$\ell(W_D^{(i)}) - \ell^* \leq \varepsilon \text{ when } i \geq \frac{\log\left(\frac{\ell(W_D^{(0)}) - \ell^*}{\varepsilon}\right)}{\log\left(\frac{L}{L-\mu}\right)}$$

Notably, our analysis reveals that under the assumption of the restricted optima condition, Dale's backpropagation achieves a linear convergence rate which matches the performance of unconstrained backpropagation [65]. This is significant, as it demonstrates that under the right conditions, imposing biological constraints through Dale's principle does not necessarily come at the cost of convergence speed. Furthermore, it also suggests that the brain's neural circuitry despite being constrained by Dale's law might also be functionally organized to facilitate efficient learning and task performance. However, it is important to note that these guarantees rely not only on the restricted optima assumption, but also on the gradients satisfying Lipschitzness and the Polyak-Łojasiewicz condition, which from a mathematically rigorous perspective may not always hold in practice.

2.2.2 Analyzing Dale's backpropagation relative to conventional backpropagation

Dale's backprop also lends itself well to analyzing its behaviour with respect to standard backpropagation when we do not make the restricted optima assumption. Specifically in the case of a single-layer recurrent neural network (without biases), we can characterize the distance between the weights found using standard backprop and Dale's backprop **(Lemma 3)**, and therefore subsequently the distances between outputs found using the two weight update schema, allowing us to bound the difference between the final error of the solution found using Dale's backprop in terms of that found using standard backprop **(Theorem 4)**. Formally, we express the above as follows:

Lemma 3 (Distance between learnt weights)**.** *Let $W^{(i)}$ and $W_D^{(i)}$ be the weights at iteration i for standard backpropagation and Dale's backpropagation, respectively. Assume the gradients $\nabla\ell(W)$ and $\nabla\ell(W_D)$ are upper bounded in magnitude by G and Lipschitz continuous with constant L. Then, the distance between the two sets of weights at any iteration i, denoted as $\left\|\delta^{(i)}\right\|_2 = \left\|W^{(i)} - W_D^{(i)}\right\|_2$ is bounded[3] by:*

$$\left\|\delta^{(i)}\right\|_2 \leq \frac{G}{L}\left((1+\eta L)^i - 1\right)$$

where η is the learning rate.

Theorem 4 (Differences in errors between solutions)**.** *Let $f(W)$ be the function represented by a single-layer RNN unrolled over T timesteps, with weights W. Let W_D be the weights learnt using Dale's backpropagation, and W be the weights learnt using standard backpropagation. Assume the non-linearity ϕ is either tanh or ReLU. Then, the error of the solution found using Dale's backpropagation with respect to the ground truth y is bounded by:*

$$\|f(W_D) - y\|_2^2 \leq 2 \cdot \left(\delta^2 \sum_{t=1}^{T}(L_{f_t})^2 + \sum_{t=1}^{T}(\varepsilon_t^*)^2\right)$$

where $f(W_D)$ is the output after K training iterations and $\delta = \frac{G}{L}\left((1+\eta L)^K - 1\right)$, $L_{f_t} = \max(L_{f_t(W)}, L_{f_t(W_D)})$ is the maximum of the Lipschitz constants of the two RNNs at timestep t, and $\varepsilon_t^* = \|f_t(W) - y_t\|$ is the error of the solution found using conventional backpropagation at timestep t.

The lemma on the distance between learnt weights quantifies how Dale's backpropagation diverges from standard backpropagation over time due to the sign constraints, showing that this divergence grows but remains bounded, influenced by factors like the learning rate, the loss landscape's smoothness, and gradient magnitudes. Building on this, the theorem on error between solutions relates the performance of Dale's backpropagation to standard backpropagation, indicating that the error of Dale's method is bounded by both the divergence of the weights (bounded by the lemma) and the sensitivity of the network's output to weight changes, alongside the error of the standard method. Together, these results provide theoretical assurances that, while biological constraints impact learning dynamics, they do not cause uncontrolled error growth, supporting the use of Dale's principle in neural network training.

Full proofs for both the lemma and theorem are provided in the supplement (Appendix 8.2).

2.3 Dale's backpropagation: Empirical results

We evaluate the performance of Dale's backpropagation across three tasks of interest (Fig. 2.A). The first is a 1-bit **flip-flop** task (Fig. 2.A - top row - left), in which the network is required to maintain and toggle between different states in response to a series of binary inputs. Specifically, the network output is meant to start at zero, following which it takes the value ± 1 to match the input signal whenever presented. It is then expected to switch signs if presented with a

[3] $\|\cdot\|_2$ always corresponds to the operator norm $||\cdot||_{\text{op}}$ induced by the 2-norm, which is the Euclidean norm if W and W_D are vectorized, and $\sigma_{\max}(\cdot)$, i.e., the largest singular value of the matrix if W and W_D are considered in their matrix forms.

signal of the opposing sign, or else, maintain the same output as before. Next, is a **wave reconstruction** task (Fig. 2.A - bottom row) where both the excitatory and inhibitory neurons are presented with individual sinusoidal waveforms. The network is tasked with accurately reconstructing both signals simultaneously, reflecting the roles of excitation and inhibition in modulating distinct aspects of signal processing in neural circuits. We finally also test our methods on the **sequential MNIST** task (Fig. 2.A - top row - right), which is a variation of the classical digit classification task in that instead of receiving the entire image as the input, the network instead sequentially receives the rows of the image.

For each of the tasks we train RNNs (over 5 runs) with 128 hidden neurons of which $\sim 80\%(102)$ are excitatory and $\sim 20\%(26)$ are inhibitory. In addition to conventional backpropagation, we also include in our experiments "rectified backprop" a similar method from the literature [50, 45] that is also amenable to incorporating sparsity constraints[4] simultaneously. Our experiments justify our proposed weight update and subsequent theoretical analyses, both in terms of how the weights evolve, as well as learning performance. We start by analyzing the distribution of weights before and after training for the three learning rules (Fig. 2.B) averaged across all tasks. While we initialize all three methods identically (Fig. 2.B - middle row - left), we notice that their distributions post-training are visibly different (Fig. 2.B - middle row - right). Specifically, the weights learnt using conventional backprop (black) show the smallest peak around zero and the greatest deviation, followed by those learnt using Dale's backprop (green), and finally rectified backprop (gray). This trend is also visible in the weight matrices themselves (Fig. 2.B - top row), where the negative weights learnt using rectified backprop seem practically to be zero (Fig. 2.B - top row - middle). On first glance the weight matrices for conventional and Dale's backprop seem almost the same, but closer inspection (black boxes – bottom right of the weight matrices) reveals that some of the weights which should have been negative have flipped signs with conventional backprop (Fig. 2.B - top row - left) but there are no such discrepancies with Dale's backprop (Fig. 2.B - top row - right). Finally, we empirically quantify the differences in learnt weights for the two sign-preserving methods by measuring the Kullback-Liebler (KL) divergence amongst their weight distributions post training (Fig. 2.B - bottom row) with respect to conventional backpropagation. We observe that across all three tasks the divergence shown by rectified backprop (gray) from the weights learnt by conventional backprop are much higher than those shown by Dale's backprop (green).

Finally, the learning performance (Fig. 2.C) of Dale's backprop (green) matches that of conventional backprop (black) for all three tasks, at times even learning faster. We conjecture this is a consequence of the regularization introduced by adhering to the sign constraints – by restricting the optimization to the orthant where the signs are preserved, the search space is effectively reduced. This focused parameter space allows for more efficient learning dynamics, as the optimizer concentrates on adjusting weight magnitudes without expending effort on sign changes that violate the constraints. The fixed signs lead to more stable and directed weight updates, resulting in a smoother optimization landscape. Consequently, Dale's backprop can converge more rapidly in the early stages of training, while ultimately achieving similar final performance as standard backpropagation. However, the learning performance of rectified backprop (gray) is both slower and not as competitive as the other two methods, especially on the more complicated sequential MNIST task. To that end, we note that this might be a consequence of the fact that rectified backprop does not allow for activation functions that have any negative outputs (e.g., tanh). This restriction likely leads to exploding gradients and dead neurons – the latter which might also be inferred visually from its weights post training. However Dale's backpropagation does not suffer from such limitations and can use non-linearities that have negative outputs as long as it is centered around 0, i.e., it keeps positive and negative activations, positive and negative respectively.

3 Sparsifying Networks via Topology-Informed Local Pruning

Having established a method that lets us train sign-constrained RNNs leveraging the machinery of autograd-based backpropagation [66, 67], in this section we describe **topologically-informed probabilistic (top-prob) pruning** as a way of sparsifying dense neural networks to reflect a target connectivity pattern amongst neuronal populations (Fig. 1.C). We first describe the pruning rule formally, while also motivating it from both a mathematical and neuroscientific perspective. Our subsequent empirical analyses demonstrate the applicability of our method in conjunction with Dale's backpropagation, wherein it outperforms the random pruning baseline on different tasks.

[4]While the methods of [38, 63] using DANNs would allow us to train with sign constraints, unfortunately adapting it to respect structured sparse motifs in non-trivial. We therefore refrain from incorporating any comparisons with such methods in this work.

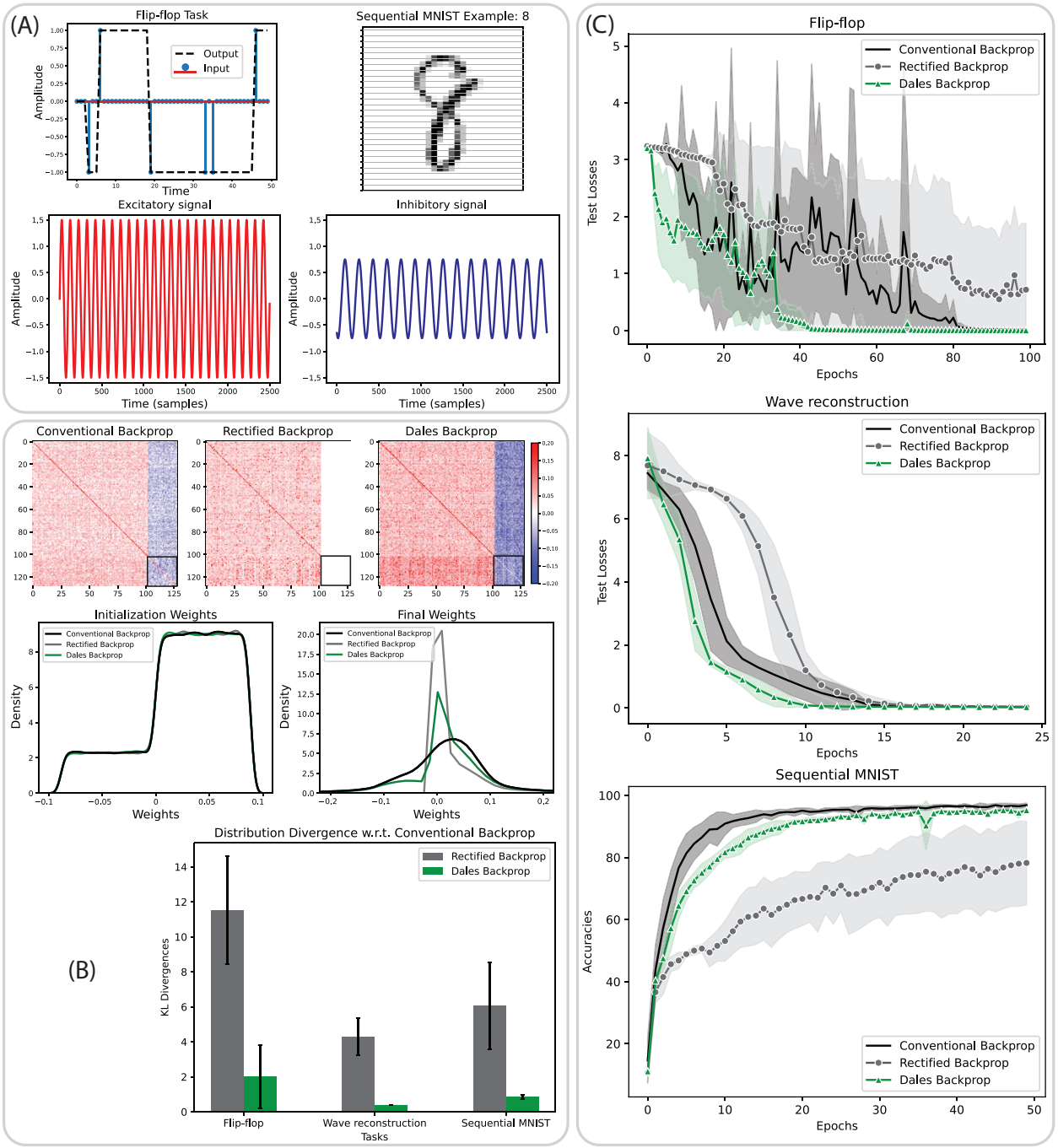

Figure 2: *Training with Dale's Backprop.* **(A)** Task examples: 1-bit flip-flop, Sequential MNIST, Wave reconstruction. **(B)** Distribution of weights: Weight matrices post-training (top row), Weight histograms at initialization and after training (middle row), Relative divergence in weight distributions with rectified and Dale's backpropagation vs. conventional backpropagation (bottom row). **(C)** Test performance of models across different tasks when trained with conventional backpropagation (black), rectified backpropagation (gray), and Dale's backpropagation (green). All statistics computed over 5 independent runs.

3.1 Topologically-informed probabilistic pruning rule

Consider a weight matrix $W \in \mathbb{R}^{N\times N}$ comprised of the synaptic weights $w_{ji}\ \forall i,j \in \{1,2,3,...,N\}$, connecting neuron $i \to j$. The sparsified matrix $W^{sparse} \in \mathbb{R}^{N\times N}$ is obtained using the pruning rule

$$w_{ji}^{sparse} = \begin{cases} w_{ji} & \text{with probability } \ \kappa|w_{ji}|, \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

where $\kappa \in \mathbb{R}^{+}$ is a non-negative scalar that controls the sparsity of the resulting matrix and is defined as

$$\kappa = \frac{(1-s)N^2}{||W||_{L^1}^2} \tag{5}$$

$s \in [0,1]$ is the target sparsity of W^{sparse} and $||W||_{L^1}^2 = \sum_{i=1}^{N}\sum_{j=1}^{N}|w_{ji}|$.

While it is evident that the top-prob pruning rule operates by probabilistically retaining weights of higher magnitude while eliminating weaker ones, we emphasize that this approach mirrors fundamental aspects of synaptic plasticity in biological neural networks. The rule's local nature – where pruning decisions depend solely on individual synaptic weights – aligns with biological constraints, as real neurons modify their connections based only on local synaptic properties rather than global network states. Furthermore, when coupled with Dale's backpropagation, the top-prob pruning mechanism has a propensity for preserving exactly those weights that align with Hebbian learning principles (Appendix 7), thereby ensuring the maintenance of biologically meaningful functional connectivity whilst simultaneously achieving network sparsification.

The top-prob approach is also grounded from an ML and mathematical standpoint. Magnitude-based pruning has a long history [48, 68, 69] and in its iterative form is still a highly competitve empirical baseline for neural network compression via pruning [39, 40]. It also closely relates to methods that look to preserve weights that maintain the dynamics of the network in the spectral sense [49, 70, 71, 72]. Finally, it provides us with an elegant way of maintaining the structural integrity of the network. Recent works have established that the zeroth-order topological information of a graph is fully encapsulated by its maximum spanning tree (MST)[5] [74, 75, 76, 73]. Ergo, probabilistically maintaining higher magnitude weights of the network results in us preserving this topological structure (Appendix 10.1).

Throughout the remainder of this work, we use top-prob pruning in the one-shot sense [41], wherein we prune to a target sparsity level and connectivity pattern in a single step, followed by a single retraining phase to help restore the model's performance. We note however that our approach can just as easily be used at initialization to sub-select a sparse network pre-training, or alternatively used in the iterative manner that is more typical in the ML community, especially for tasks that are more complicated and less amenable to drastic drops in sparsity levels from a fully-trained dense configuration. Additional explanations for how we derive and re-normalize our hyper-parameter κ across different contingencies, as well as adjust the parameter s are provided in Appendix 9.1, 9.2, and 9.3 respectively.

3.2 Topologically-informed probabilistic pruning: Empirical results

We study the behaviour and performance of top-prob pruning by first examining how it impacts the weight distribution and structural integrity of the original dense network, followed by its ability to maintain functional capacity. Our results suggest that top-prob pruning does indeed preserve key network properties, leading to highly sparse yet robust models that do not require extensive retraining to regain performance.

As a preliminary check we observe the distribution of non-zero weights (Fig. 3.A) for dense weight matrix (black), vs. that of a matrix that has been pruned to 90% sparsity using the top-prob pruning rule (green) and random pruning (grey). As expected, we see the dense matrix has weights that are almost uniformly distributed since the weights were sampled from the distribution $U\left(\left[\frac{-1}{\sqrt{N}}, \frac{1}{\sqrt{N}}\right]\right)$ and the randomly pruned network stays faithful to this distribution. The weights retained using top-prob pruning however are heavily skewed towards being higher in magnitude, demonstrating that it successfully prioritizes stronger connections while eliminating weaker ones. We subsequently quantify the notion of "structural integrity preservation" by plotting the fraction overlap of retained weights with the MST of the dense matrix as sparsity increases (Fig. 3.B) for both pruning methods. Again, we observe that empirically top-prob pruning shows a much higher overlap with the MST (green dots) compared to random sparsification in practice (grey crosses) and theoretically (black crosses) (Appendix 10.2). Additionally, it shows lesser variations in the amount of overlap as well.

Moving on to the pruned network's ability to retain information and functional capacity, we note that the errors post-pruning but before fine-tuning (Fig. 3.C) are always higher for models that are pruned randomly (grey) vs. those

[5] For a more thorough treatment of this topic, we refer the interested reader to Sections 2 & 3 of [73].

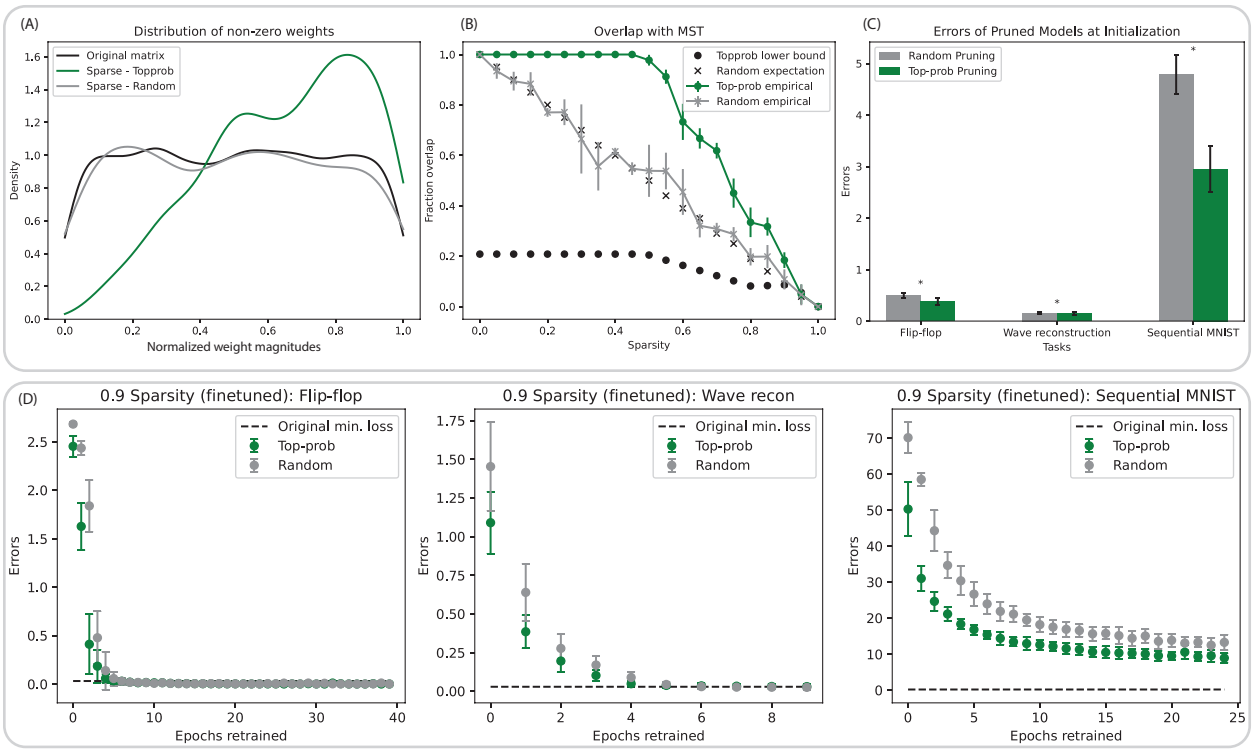

Figure 3: *Topologically-informed probabilistic pruning.* **(A)** Distribution of non-zero weights for dense (black), and sparse matrices pruned randomly (grey) and with top-prob pruning (green). **(B)** Fraction overlap that retained weights have with the MST of the dense matrix. **(C)** Errors of pruned models, without any retraining using random (grey) and top-prob pruning (green). **(D)** Performance of sparsified and fine-tuned models across different tasks, when pruned randomly (grey) vs. top-prob pruning (green). All statistics computed over 5 independent runs. * indicates $p \leq 0.05$.

pruned with the top-prob pruning rule (when pruned to 90% sparsity). Moreover, the difference in errors becomes more significant as the task complexity increases (p = 0.022 for the flip-flop and wave reconstruction tasks, while p = 0.012 for sequential MNIST). Fine-tuning with Dale's backpropagation for $\sim 50\%$ the number of epochs as the original training, while retaining the sparse structure identified via pruning follows a similar trend (Fig. 3.D), with networks that that were pruned randomly (grey) showing higher errors at the epoch of re-training than those pruned with top-prob pruning (green). While both methods lead to models that seem to eventually approach the original model's performance (dashed line) after fine-tuning, top-prob pruning consistently starts from a better initial error, converges faster to optimal performance, and shows more stable learning. We therefore conclude that our pruning rule effectively identifies and retains functionally important weights. The preserved connectivity aligns well with the network's core topology (MST) which leads to efficient and robust performance of the sparsely structured RNN, in conjunction with Dale's backpropagation.

4 Application to Visual Behaviour in Mice: Functional Connectivity and Predictive Coding

Having established the efficacy of our methods in successfully constructing and training highly sparse RNNs which respect Dale's law, we apply them to study visual behaviour in mice under the predictive coding hypothesis [57, 77]. Specifically, we model data from the Allen Institute Visual Behavior dataset [78, 79], which comprises two-photon (2p) calcium imaging recordings from mice performing a change detection task, when presented with expected and unexpected stimuli. This experimental paradigm allows us to investigate how different neuronal populations in the cortical circuit interact and process information under varying predictive contexts, shedding light on how prediction errors may be communicated across hierarchically-related cortical regions. Moreover, since our modeling framework captures these interactions while respecting anatomical connectivity and signaling constraints, our analyses reveal how functional connectivity between populations adapts to the inherent biological scaffolding to support such predictive processing, providing insights into the circuit-level implementations of predictive coding in the visual cortex.

In the following subsections we provide details of the specific experimental setup and curated dataset, followed by our model architecture and training methodology. Our results align well with previous observations made in the experimental literature studying the data, and strongly support the predictive coding hypothesis. Furthermore, by "learning" the functional connectivity under various conditions, our approach not only corroborates previous experimental results, but also gives us a way to generate new hypotheses about how different prediction violations engage distinct patterns of feedforward and feedback connectivity across cortical layers and cell types, offering novel insights into the principles governing cortical circuit organization in predictive processing.

4.1 Dataset and experimental setup

The Visual Behavior Dataset [78, 79] entails a visually-guided, go/no-go task where mice are shown a continuous series of briefly presented natural images and they earn water rewards by correctly reporting when the identity of the image changes [80]. Responses from the mice are collected as they are presented with two different sets of images; A **familiar set** (Fig. 4.B - top row) comprising images that they were trained on, and a **novel set** (Fig. 4.B - bottom row) that are only presented at test time, during the recordings. While the trials themselves are longitudinal spanning multiple image changes, we restrict ourselves to modeling two full image presentations (Fig. 4.C - Top). If the identity of the second image is the same as that of the first one, we refer to the condition as **no change** (Fig. 4.C - second row). If the identity of the second image is different from that of the first one, we refer to it as the **change** condition (Fig. 4.C - third row). Both images are always from the same set, i.e., they are both either familiar or novel. In a small subset of the trials ($\sim 5\%$), the second image is omitted, and instead replaced by a blank screen (Fig. 4.C - first row) allowing for analysis of expectation signals. We call this the **omission** condition. For more details on the experimental setup, see [79].

For each of our conditions we consider two temporal windows. In the **full-set presentation** (Fig. 4.C - indicated at the bottom), we model neural activity across the entire two-image sequence (first image (250ms), inter-stimulus interval (500ms), second presentation/omission (250ms), and post-stimulus interval (500ms)), which allows us to capture the sustained dynamics underlying predictive computation across time. In contrast, the **half-set presentation** (Fig. 4.C - indicated at the top) models neural activity following the second presentation/omission, enabling us to isolate the transient neural responses that implement the mechanistic components of prediction and error signalling. This complementary approach provides insights into both, the overarching dynamics and the immediate neural interactions that support predictive coding, and gives us the flexibility to infer both long-term and short-term functional interactions.

The complete dataset includes multi-regional 2-photon data from *two hierarchically adjacent areas*, **VISp** (i.e., primary visual cortex or **V1**) and **VISl** (i.e., the lateromedial area or **LM**). For both areas we collect recordings at *depths* roughly corresponding to **layers 2/3, 4, and 5** in the cortical column for *excitatory*, i.e., pyramidal **(Pyr)** neurons and *two types of inhibitory neurons*, viz. somatostatin **(Sst)** and vasoactive intestinal peptide **(Vip)** expressing interneurons (sampling depths distributions provided in Appendix 11). In total, we therefore model the activities of 18 different interacting populations (Fig 4.A).

To curate the training data for our RNNs, we compute the neuron-averaged response for every experiment corresponding to each of our individual neuronal populations (e.g., LM L5 Vip) from the Allen Institute Visual Behavior-2P dataset [78]. We then randomly sample (with replacement) 100 averaged responses from the total set of averaged responses, take their mean, and pass the same through a 1-D Gaussian filter ($\sigma = 1$) to produce a single training sample (Fig. 4.C - dashed black curve). We subsequently produce 2000 such samples for each of our individual neuronal populations.

4.2 CelltypeRNN: Architecture and training

We model the data as described previously with the anatomically constrained **CelltypeRNN** that replicates the inter-areal structure of the canonical cortical microcircuit with two hierarchically related cortical areas (Fig. 4.A) [81, 82, 83, 84] whilst simultaneously enforcing intra-areal lateral connectivity among different cell types within the cortical column as established by [85]. Moreover, given that the CelltypeRNN is constructed to be able to replicate experimentally obtained response patterns in different cell populations as specified by their cell type, cortical layer and area, by learning the connection weights, we in turn represent the inferred functional interactions between the populations across the cortical circuit [20, 11] under different stimulus conditions.

Subsequently, we first train a dense, unbiased Elman RNN using Dale's backpropagation, following which we prune the network's recurrent connections block-wise with top-prob pruning (Fig. 4.D, left) to achieve their individual target connectivity sparsities. We subsequently fine-tune the post-pruning non-zero RNN weights to achieve an overall performance that is at least as good as that of the RNN pre-pruning (Fig. 4.C, bottom). In our specific instantiation, the ratio of Pyr:Sst:Vip neurons in every layer is 12:2:1 (making the excitatory:inhibitory neuronal ratio 4:1), which with a scaling factor of 16 gives us a total of 240 neurons per layer and 1440 overall in the model. Our lateral connectivity probabilities across populations follow experimental data [85] and are explicitly stated in Appendix 12. Longer range

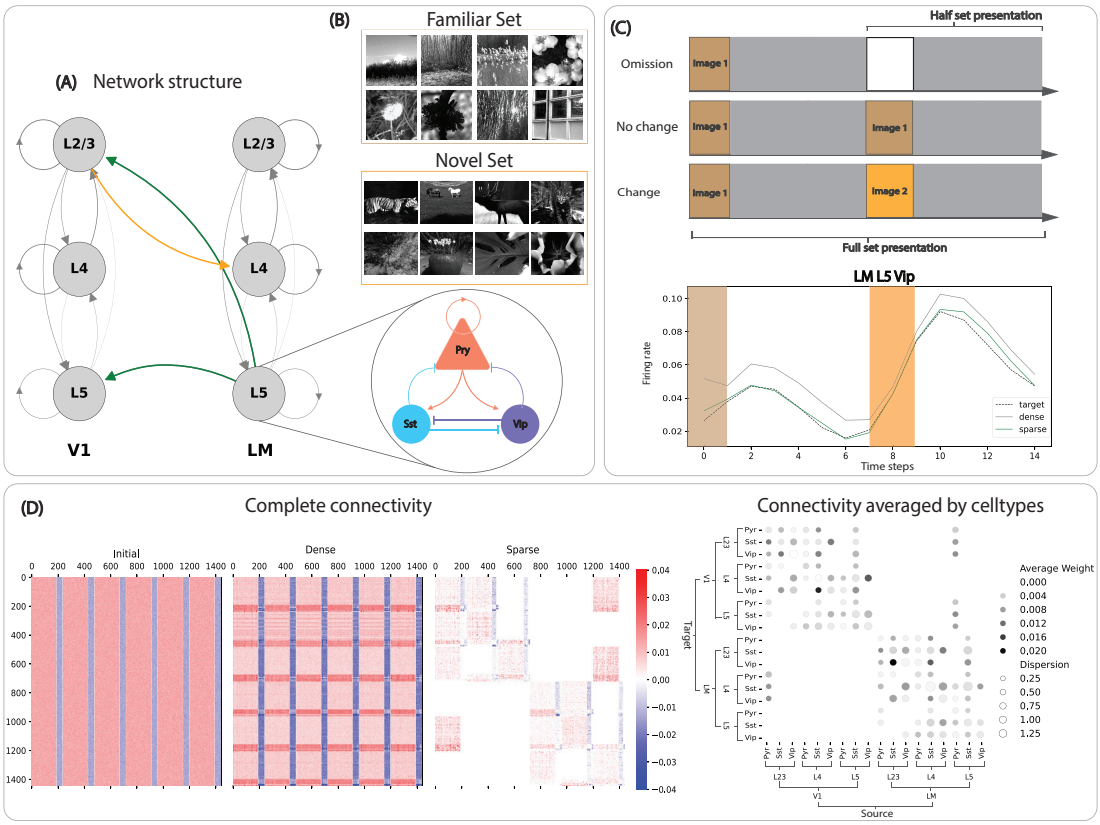

Figure 4: *Dataset, network structure, and task schematics.* **(A)** General architecture of the CelltypeRNN. **(B)** Familiar and Novel image sets used for training mice on the visual change detection task. Reproduced from the Allen Institute Visual Behavior-2p dataset (open source) [78]. **(C)** Top - Examples of different stimuli conditions in the visual change detection task, depiction of full and half set presentation timescales. Bottom - An example of target activity (dashed black curve), dense RNN output (solid grey curve), and sparse RNN output (solid green curve) from LM L5 Vip population. **(D)** Examples of inferred functional connectivity: Left - Complete neuron-to-neuron connectivity at initialization, after training with Dale's backprop, and after sparsification (and retraining) with top-prob pruning. Neurons are ordered by area (V1 followed by LM) within which they are ordered by layer (L4, L2/3, L5), and type (Pyr, Sst, Vip). Right - Example of the sparse connectivity matrix where activity is averaged by cell type in every layer (bigger circles imply higher dispersion and darker colours imply stronger connections; dispersion is computed as the fraction of standard deviations to the mean activity in the population).

inter-areal projections are sparsified to have a connection probability of 0.3, and are strictly excitatory, i.e., Feedforward connections: V1 L2/3 Pyr → LM L4 Pyr, Sst, Vip. Feedback connections: V1 L2/3 Pyr, Sst, Vip ← LM L5 Pyr and V1 L5 Pyr, Sst, Vip ← LM L5 Pyr.

In addition to the weights of the RNN – i.e., input weights W_{hi} and recurrent weights W_{hh} – we also have readout weights that project the recurrent RNN activity of individual neuronal populations onto their respective output space, using randomly initialized, fully connected linear layers. The readout weights are frozen at the time of initialization of the dense RNN itself, and remain so throughout the training procedure. By doing so we ensure that any changes in the model's behavior come from changes in the recurrent dynamics, and not the model "cheating" by simply adjusting its output mapping. It subsequently also makes it easier to interpret and compare how the internal representations and computations change across conditions. To that end, we also mask the input weight matrix W_{hi} so that recurrent neurons corresponding to a specific population do not receive inputs from any other populations.

Our training objective requires each individual population to be able to reconstruct its activity predictively one timestep into the future (Fig. 4.C, bottom), giving us the loss function

$$\mathcal{L}_{total} = \frac{1}{n_{pop}} \sum_{n=1}^{n_{pop}} \sum_{t=1}^{T-1} \|x_{n,t+1} - \hat{x}_{n,t+1}\|_2^2 \tag{6}$$

where n_{pop} is the number of interacting neuronal populations and T is the total number of timesteps in the sequence. $x_{n,t+1}$ is the input that will be received for population n at timestep $t+1$ while $\hat{x}_{n,t+1}$ is that predicted by the RNN. The loss function is kept the same during both, the dense training (Fig. 4.C, bottom - grey curve) and fine-tuning post-pruning stages (Fig. 4.C, bottom - green curve). However, we fine-tune for only half the number of epochs (50) as we train for with the dense network (100).

We train separate models for each of our twelve different conditions (Familiar/Novel $\times$ Change/No Change/Omission $\times$ Full Set/Half Set presentation) and compare their connection weights across various spatial scales, the results of which are discussed in the following subsection.

Our codebase to download and pre-process the data, as well as construct and train the celltypeRNN models across various conditions and timescales is made publicly available at https://hchoilab.github.io/biologicalRNNs.

4.3 Insights and results

Using the anatomically-constrained CelltypeRNN architecture, we examine how distinct cell types across the layers and hierarchy in the visual cortex communicate both expected and unexpected information by comparing inferred connectivity patterns across different experimental conditions and timescales. By fitting neuronal responses of interacting populations through one-step-ahead predictive modeling, we capture the dynamic temporal dependencies inherent in neural activity and the RNN's resulting connectivity matrix serves as a functional proxy for interactions amongst populations, reflecting how signals propagate within the cortical network. Analyzing how the RNN adjusts its connectivity across varying predictive contexts and timescales provides insights into circuit-level implementations of predictive coding, particularly in prediction error communication and modulation of feedforward and feedback pathways over both, entire stimulus sequences and immediate neural responses to prediction confirmations or violations. Our results can be broken down into three key comparisons:

Familiar No Change vs. Familiar Change (Full-set Presentation): In the full-set presentation of familiar images, we observe significant differences in the inter-areal feedforward and feedback connections (Fig. 5.A). When the activities are averaged across layers, there is a stark increase in the projection V1 L2/3 $\rightarrow$ LM L4 when there is a change in the image compared to when there is not, suggesting that the expectation violation causes enhanced forward communication from V1 to LM (Fig. 5.A - left, middle). Likewise, feedback projections V1 L2/3 $\leftarrow$ LM L5 and V1 L5 $\leftarrow$ LM L5 are strengthened as well in the change case (Fig. 5.A - left, middle). Even at the scale of cell-types, we observe that the change condition leads to an increase in functional connectivity for both inter-areal feedforward and feedback projections (Fig. 5.A - right, magenta boxes). Additionally, we note that the changes are predominantly red, i.e., the inferred weights in the change condition are generally higher than that in the case of expected stimuli and conditions being perceived (5.A - middle). This trend also holds when we compare familiar and novel stimuli (Appendix 13, Fig. 7) in both the change and no change cases, i.e., the introduction of novelty leads to increased inter/intra area connectivity. In agreement with previous literature [86], this suggests that novelty and unexpectedness increase the brain's excitability, which in turn could facilitate plasticity and aid learning.

Familiar No Change vs. Familiar Change (Half-set Presentation): When focusing on the half-set presentation for familiar images however, we found a contrasting pattern of connectivity with the feedback signaling (Fig. 5.B - left, middle). While we see almost no difference in the weights V1 L5 $\leftarrow$ LM L5 across the change and no change cases, the weights V1 L2/3 $\leftarrow$ LM L5 are distinctly higher in the *no change* case than the change case. The feedforward projection V1 L2/3 $\rightarrow$ LM L4 however still maintains the same trend as the full presentation case, wherein it is higher when an image change occurs than when it does not. This suggests that while the feedforward communication is relatively immediate using shorter timescales, propagating feedback information occurs over a longer timescale [83, 87, 88], making a case for further investigation of role of inter-areal, cortico-cortical time-delays [89] when studying predictive coding [84]. The cell-type-specific analysis further reveals that Vip neurons in L2/3 are less inhibited by Sst neurons of the same layer in the change case in both V1 and LM (Fig. 5. B - right, green boxes), compared to the full presentation case (Fig. 5. A - right, green boxes), once again speaking to the transience of the change and also supporting the idea that Vip neurons could encode unexpectedness as hypothesized by the predictive coding theory.

Familiar Change vs. Familiar Omission: Our setup also allows us to compare how the network processes different types of expectation violations, by contrasting the learnt weights in the case of an image change vs. image omission. In the setting of familiar images over the length of a full set presentation (Fig. 5.C), we notice that while most of the weights are quite similar for both types of expectation violations, there is an appreciable increase in the feedback projection V1 L5 $\leftarrow$ LM L5 in the case of image omission (Fig. 5.C - left, middle). Additionally, this increase seems

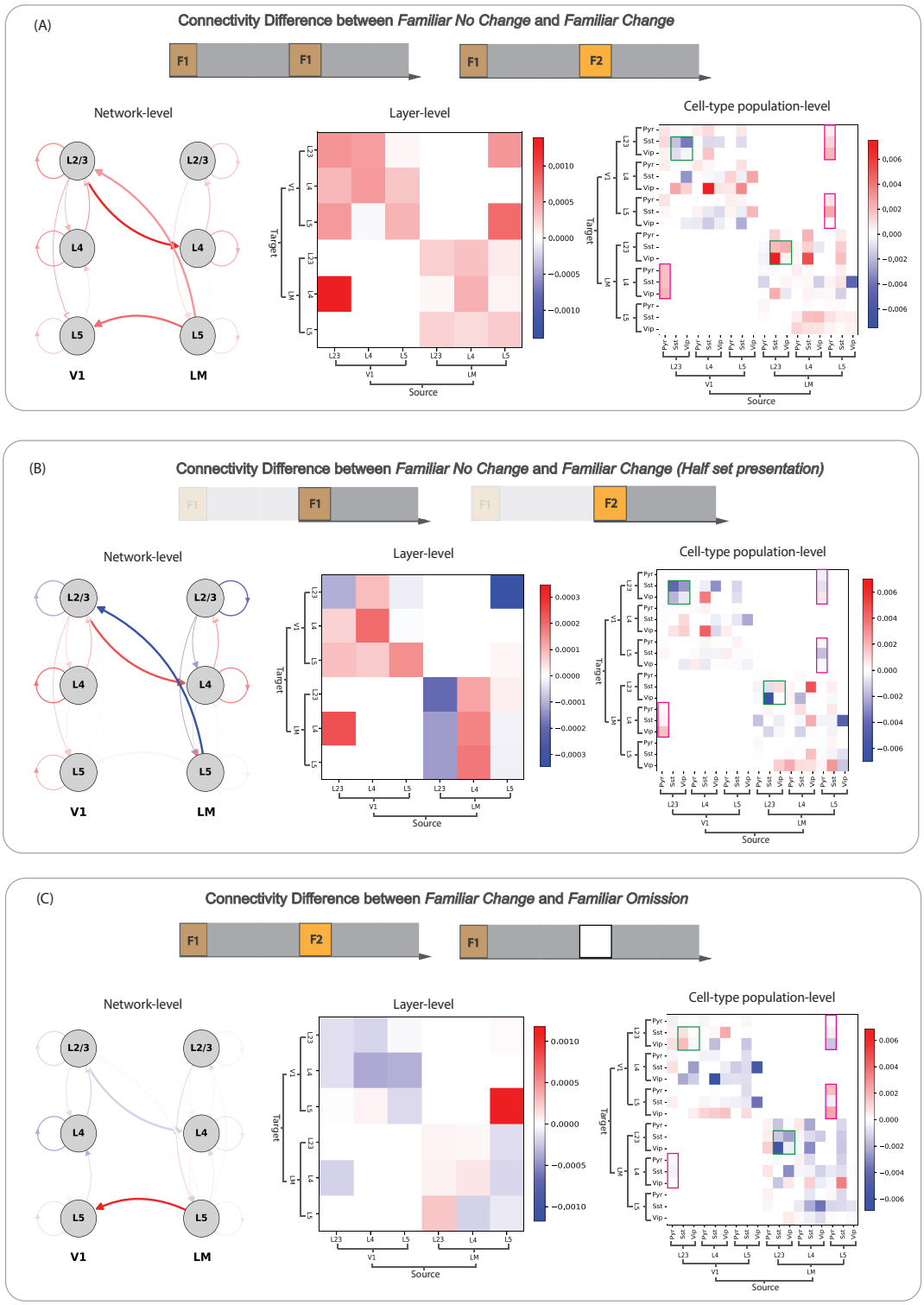

Figure 5: *Connectivity differences across timescales and test conditions.* **(A)** Familiar No Change vs. Familiar Change (Full set presentation). **(B)** Familiar No Change vs. Familiar Change (Half set presentation). **(C)** Familiar Change vs. Familiar Omission (Full set presentation). All differences are computed as `Second condition - First condition`; Blue implies higher weights in the first condition, while red indicates higher weights in the second. In all three cases, the left and middle plots are a graphical representation of the weights averaged across layers, while the rightmost plot averages weights by cell-type within each layer. Magenta boxes highlight the feedforward and feedback connections, i.e., those originating at V1 L2/3 and LM L5 respectively. Green boxes highlight all Sst-ViP interactions in L2/3 of both V1 and LM.

to be driven by an increase in the feedback connection weight to the Vip cells in V1 L5[6] (Fig. 5.C - right). These observations are in agreement with experimental findings that omissions trigger signaling in Vip neurons in V1 [90]. We also note that across the RNN, weights to the Vip neurons from locally adjacent Sst neurons are reduced in the omission case, suggesting that these neurons are not as inhibited during the processing of omissions, thus potentially emphasizing their role in processing prediction violations. In the supplement, we also provide results studying the No Change vs. Omission case with both familiar and novel images (Appendix 13, Fig. 8) for fair comparison.

Collectively, our analysis demonstrates a hierarchical organization of predictive processing in the visual cortex operating over different timescales. We find that feedforward projections are consistently enhanced during all prediction violations across both long and short timescales, emphasizing their crucial role in transmitting prediction error signals (and fundamentally driving synaptic plasticity in the brain [91, 92, 93], facilitating learning and adaptation). In contrast, feedback projections are modulated by both the type of prediction error and the temporal window over which neuronal responses are modeled. Notably, the behavior of feedback projections differs when targeting different cortical layers: feedback projections to L2/3 are more prominently modulated during unviolated predictions over shorter timescales (Fig. 5.B), while feedback projections to L5 are more responsive during negative prediction errors such as omissions of expected visual input [94] (Fig. 5.C). This differential modulation suggests that while feedforward pathways rapidly convey unexpected sensory information, feedback pathways adjust more selectively based on the context, timing, and targeted cortical layer of the prediction error. These patterns are further corroborated by our observations comparing change, no change, and omission across familiar and novel conditions during the full-set presentation (Appendix 13, Fig. 7). In particular, we note that the presentation of a novel image always increases the feedforward connectivity (and ergo the projection) from V1 L2/3 $\rightarrow$ LM L4. On the finer-scale level of cell-types instead of entire layers, there is consistent prominent involvement of Vip interneurons during prediction violations[7] which highlights their critical role in modulating cortical circuits in response to unexpected stimuli. Overall, our findings therefore provide circuit-level evidence supporting the predictive coding framework, illustrating how the brain dynamically adjusts its functional neural connectivity in response to varying predictive contexts, timescales, and cortical layers. The dynamic interplay of feedforward and feedback mechanisms facilitates efficient processing of sensory information, enabling the brain to anticipate and adapt to constantly changing environments.

Results for all comparisons across both the full set and half set presentations are publicly available at the project website.

5 Discussion

Our work develops methods for constructing RNNs that simultaneously incorporate two fundamental biological constraints: *Dale's law* and *structured sparse connectivity motifs*. We provide mathematical grounding for these methods, including convergence guarantees and error bounds, demonstrating that they can match the performance of unconstrained RNNs. Empirical results on standard synthetic tasks support the efficacy of our approach, demonstrating that our biologically constrained RNNs can achieve performance comparable to conventional, unconstrained networks. Furthermore, by aligning computational models more closely with biological reality, we enhance their utility for neuroscientific research, providing tools for more accurate modeling of neural dynamics and brain function.

Our approach also differs significantly from CURBD [20], an existing method in the literature for inferring multi-regional interactions, in two key aspects. First, while CURBD successfully models neural dynamics and iteractions, it does not incorporate sign constraints during training, limiting its ability to differentiate between excitatory and inhibitory cellular mechanisms. Second, and more critically, CURBD's reliance on FORCE training makes it poorly suited for implementing experimentally-informed sparse connectivity patterns among neuronal populations. Every iteration with FORCE is a least-squares update that is dense and doesn't respect the sparsity constraints of the matrix at the previous iteration - it is non-trivial to subsequently enforce the sparsity pattern, or alternatively solve a recursive least-squares update for every sub-matrix defined by the sparsity pattern at each update, which quickly becomes computationally infeasible. These limitations consequently motivated our development of a our backpropagation-based weight update method that efficiently handles both Dale's law constraints and structured sparsity.

Applying our methods to the Allen Institute Visual Behavior dataset, we inferred multi-regional neuronal interactions underlying visual behavior in mice performing a change detection task. Our anatomically and physiologically constrained *celltypeRNNs* not only replicated the experimental data but also provided insights consistent with the theory of predictive coding. Specifically, the models revealed dynamic interplay between feedforward and feedback mechanisms

[6] As well as an overall increase in the connectivity weights targeted to V1 L5 Vip neurons.

[7] We note however that these interactions do not directly confirm the experimental results of [56, 79], in that they show a coding change in the relevant populations but not in what would seem to be the same direction, i.e., Sst $\rightarrow$ Vip reduces in the case of novelty or omission.

across cortical layers and cell types, capturing how the brain adjusts functional neural connectivity in response to varying predictive contexts and timescales.

We note that much of our methodological work can easily be extended to other deep architectures, and is not in fact restricted to simply RNNs. That said, a key area for incorporation of additional biological realism would be in the way we inherently solve the credit assignment problem. Backpropagation suffers from needing a global error signal and weight symmetry [95], prompting the need for more biologically plausible learning rules that can still learn as effectively. One hypothesis is that using local learning rules may contribute to the emergence of more modular network representations by promoting the formation of localized activity clusters, thus leading to deeper insights into how functional specialization arises in neural systems and its role in facilitating learning.

Furthermore, our findings highlight differential neural responses to different types of prediction errors, emphasizing the importance of the nature of the violations in shaping neural dynamics. In our study, the change in the familiar image case represents a "global oddball" – an unexpected stimulus that violates established patterns while maintaining the local context. Conversely, the omission of an expected stimulus constitutes a "local oddball", introducing a novel scenario for the network. This distinction is significant, as recent work [96] has found that global oddballs elicit responses in non-granular layers, differing from local oddballs that evoke early responses in superficial layers 2/3, consistent with conventional predictive coding theory. Our findings align with this pattern for global oddballs but present discrepancies in the case of local oddballs (omissions). This underscores the need for further exploration into stimulus dependency in error encoding [97], and suggests that normative predictive coding computations may need to account for the type of prediction error to fully capture neural processing dynamics.

Finally, we note that an important consideration in our study is the limited scope of recorded celltypes and brain regions, which poses challenges in interpreting our results. Specifically, we do not have recordings from all interacting celltypes and areas that may be involved in the visual processing tasks we modeled. This limitation means that our models might capture neural responses that are more correlational rather than causal, as they are based solely on the observed data from recorded populations. The absence of data could lead to incomplete or biased representations of neural interactions, especially at the finer grained level of celltypes as opposed to the coarser level of layers, where the absence of a particular subpopulation's influence is more easily subsumed within aggregate dynamics. To address this gap, future work could involve developing methods that account for unobserved interactions, perhaps through incorporating prior knowledge of anatomical and functional connectivity or using computational techniques to infer missing information. Additionally, expanding experimental recordings to include more brain areas and cell-types would provide a more comprehensive dataset, enabling our models to capture the full complexity of neural dynamics and leading to more causally robust conclusions.

Acknowledgments

We thank Anqi Wu for insightful comments and feedback. This work was supported by the Alfred P. Sloan Foundation Fellowships in Neuroscience (to H.C.) and the National Eye Institute of the National Institutes of Health under Award Number R00 EY030840 (to H.C.). The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health.

References

[1] Yarden Cohen, Tatiana A Engel, Christopher Langdon, Grace W Lindsay, Torben Ott, Megan AK Peters, James M Shine, Vincent Breton-Provencher, and Srikanth Ramaswamy. Recent advances at the interface of neuroscience and artificial neural networks. *Journal of Neuroscience*, 42(45):8514–8523, 2022.

[2] Andrew Saxe, Stephanie Nelli, and Christopher Summerfield. If deep learning is the answer, what is the question? *Nature Reviews Neuroscience*, 22(1):55–67, 2021.

[3] 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. *Nature neuroscience*, 22(11):1761–1770, 2019.

[4] Tim C Kietzmann, Patrick McClure, and Nikolaus Kriegeskorte. Deep neural networks in computational neuroscience. *BioRxiv*, page 133504, 2017.

[5] Daniel LK Yamins and James J DiCarlo. Using goal-driven deep learning models to understand sensory cortex. *Nature neuroscience*, 19(3):356–365, 2016.

[6] Omri Barak. Recurrent neural networks as versatile tools of neuroscience research. *Current opinion in neurobiology*, 46:1–6, 2017.

[7] Guangyu Robert Yang and Xiao-Jing Wang. Artificial neural networks for neuroscientists: a primer. *Neuron*, 107(6):1048–1070, 2020.

[8] Matthew T Kaufman, Mark M Churchland, Stephen I Ryu, and Krishna V Shenoy. Cortical activity in the null space: permitting preparation without movement. *Nature neuroscience*, 17(3):440–448, 2014.

[9] Matthew G Perich, Juan A Gallego, and Lee E Miller. A neural population mechanism for rapid learning. *Neuron*, 100(4):964–976, 2018.

[10] João D Semedo, Amin Zandvakili, Christian K Machens, M Yu Byron, and Adam Kohn. Cortical areas interact through a communication subspace. *Neuron*, 102(1):249–259, 2019.

[11] Matthew G Perich and Kanaka Rajan. Rethinking brain-wide interactions through multi-region 'network of networks' models. *Current opinion in neurobiology*, 65:146–151, 2020.

[12] Leo Kozachkov, Michaela Ennis, and Jean-Jacques Slotine. Rnns of rnns: Recursive construction of stable assemblies of recurrent neural networks. *Advances in neural information processing systems*, 35:30512–30527, 2022.

[13] David Sussillo and Larry F Abbott. Generating coherent patterns of activity from chaotic neural networks. *Neuron*, 63(4):544–557, 2009.

[14] Brian DePasquale, Christopher J Cueva, Kanaka Rajan, G Sean Escola, and LF Abbott. full-FORCE: A target-based method for training recurrent networks. *PloS one*, 13(2):e0191527, 2018.

[15] David Sussillo, Mark M Churchland, Matthew T Kaufman, and Krishna V Shenoy. A neural network that finds a naturalistic solution for the production of muscle activity. *Nature neuroscience*, 18(7):1025–1033, 2015.

[16] Valerio Mante, David Sussillo, Krishna V Shenoy, and William T Newsome. Context-dependent computation by recurrent dynamics in prefrontal cortex. *Nature*, 503(7474):78–84, 2013.

[17] Chethan Pandarinath, Daniel J O'Shea, Jasmine Collins, Rafal Jozefowicz, Sergey D Stavisky, Jonathan C Kao, Eric M Trautmann, Matthew T Kaufman, Stephen I Ryu, Leigh R Hochberg, et al. Inferring single-trial neural population dynamics using sequential auto-encoders. *Nature methods*, 15(10):805–815, 2018.

[18] Abigail A Russo, Ramin Khajeh, Sean R Bittner, Sean M Perkins, John P Cunningham, Laurence F Abbott, and Mark M Churchland. Neural trajectories in the supplementary motor area and motor cortex exhibit distinct geometries, compatible with different classes of computation. *Neuron*, 107(4):745–758, 2020.

[19] Kanaka Rajan, Christopher D Harvey, and David W Tank. Recurrent network models of sequence generation and memory. *Neuron*, 90(1):128–142, 2016.

[20] Matthew G Perich, Charlotte Arlt, Sofia Soares, Megan E Young, Clayton P Mosher, Juri Minxha, Eugene Carter, Ueli Rutishauser, Peter H Rudebeck, Christopher D Harvey, et al. Inferring brain-wide interactions using data-constrained recurrent neural network models. *BioRxiv*, pages 2020–12, 2020.

[21] Niru Maheswaranathan, Alex Williams, Matthew Golub, Surya Ganguli, and David Sussillo. Reverse engineering recurrent networks for sentiment classification reveals line attractor dynamics. *Advances in neural information processing systems*, 32, 2019.

[22] David Sussillo and Omri Barak. Opening the black box: low-dimensional dynamics in high-dimensional recurrent neural networks. *Neural computation*, 25(3):626–649, 2013.

[23] Daniel LK Yamins, Ha Hong, Charles F Cadieu, Ethan A Solomon, Darren Seibert, and James J DiCarlo. Performance-optimized hierarchical models predict neural responses in higher visual cortex. *Proceedings of the national academy of sciences*, 111(23):8619–8624, 2014.

[24] Alexander JE Kell, Daniel LK Yamins, Erica N Shook, Sam V Norman-Haignere, and Josh H McDermott. A task-optimized neural network replicates human auditory behavior, predicts brain responses, and reveals a cortical processing hierarchy. *Neuron*, 98(3):630–644, 2018.

[25] Jonas Kubilius, Martin Schrimpf, Kohitij Kar, Rishi Rajalingham, Ha Hong, Najib Majaj, Elias Issa, Pouya Bashivan, Jonathan Prescott-Roy, Kailyn Schmidt, et al. Brain-like object recognition with high-performing shallow recurrent anns. *Advances in neural information processing systems*, 32, 2019.

[26] Martin Schrimpf, Jonas Kubilius, Ha Hong, Najib J Majaj, Rishi Rajalingham, Elias B Issa, Kohitij Kar, Pouya Bashivan, Jonathan Prescott-Roy, Franziska Geiger, et al. Brain-score: Which artificial neural network for object recognition is most brain-like? *BioRxiv*, page 407007, 2018.

[27] Jonathan A Michaels, Stefan Schaffelhofer, Andres Agudelo-Toro, and Hansjörg Scherberger. A neural network model of flexible grasp movement generation. *biorxiv*, page 742189, 2019.

[28] Aran Nayebi, Daniel Bear, Jonas Kubilius, Kohitij Kar, Surya Ganguli, David Sussillo, James J DiCarlo, and Daniel L Yamins. Task-driven convolutional recurrent models of the visual system. *Advances in neural information processing systems*, 31, 2018.

[29] Grace W Lindsay. Convolutional neural networks as a model of the visual system: Past, present, and future. *Journal of cognitive neuroscience*, 33(10):2017–2031, 2021.

[30] Demis Hassabis, Dharshan Kumaran, Christopher Summerfield, and Matthew Botvinick. Neuroscience-inspired artificial intelligence. *Neuron*, 95(2):245–258, 2017.

[31] Rylan Schaeffer, Mikail Khona, and Ila Fiete. No free lunch from deep learning in neuroscience: A case study through models of the entorhinal-hippocampal circuit. *Advances in neural information processing systems*, 35:16052–16067, 2022.

[32] Adam H Marblestone, Greg Wayne, and Konrad P Kording. Toward an integration of deep learning and neuroscience. *Frontiers in computational neuroscience*, 10:215943, 2016.

[33] John Carew Eccles. From electrical to chemical transmission in the central nervous system: the closing address of the sir henry dale centennial symposium cambridge, 19 september 1975. *Notes and records of the Royal Society of London*, 30(2):219–230, 1976.

[34] Harini Eavani, Theodore D Satterthwaite, Roman Filipovych, Raquel E Gur, Ruben C Gur, and Christos Davatzikos. Identifying sparse connectivity patterns in the brain using resting-state fmri. *Neuroimage*, 105:286–299, 2015.

[35] Marcus Kaiser. Connectomes: from a sparsity of networks to large-scale databases. *Frontiers in Neuroinformatics*, 17:1170337, 2023.

[36] Janne K Lappalainen, Fabian D Tschopp, Sridhama Prakhya, Mason McGill, Aljoscha Nern, Kazunori Shinomiya, Shin-ya Takemura, Eyal Gruntman, Jakob H Macke, and Srinivas C Turaga. Connectome-constrained networks predict neural activity across the fly visual system. *Nature*, pages 1–9, 2024.

[37] Giuseppe Giacopelli, Domenico Tegolo, Emiliano Spera, and Michele Migliore. On the structural connectivity of large-scale models of brain networks at cellular level. *Scientific Reports*, 11(1):4345, 2021.

[38] Jonathan Cornford, Damjan Kalajdzievski, Marco Leite, Amélie Lamarquette, Dimitri M Kullmann, and Blake Richards. Learning to live with dale's principle: Anns with separate excitatory and inhibitory units. *bioRxiv*, pages 2020–11, 2020.

[39] Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. *arXiv preprint arXiv:1803.03635*, 2018.

[40] Hidenori Tanaka, Daniel Kunin, Daniel L Yamins, and Surya Ganguli. Pruning neural networks without any data by iteratively conserving synaptic flow. *Advances in neural information processing systems*, 33:6377–6389, 2020.

[41] Namhoon Lee, Thalaiyasingam Ajanthan, and Philip HS Torr. Snip: Single-shot network pruning based on connection sensitivity. *arXiv preprint arXiv:1810.02340*, 2018.

[42] Chaoqi Wang, Guodong Zhang, and Roger Grosse. Picking winning tickets before training by preserving gradient flow. *arXiv preprint arXiv:2002.07376*, 2020.

[43] Song Han, Jeff Pool, John Tran, and William Dally. Learning both weights and connections for efficient neural network. *Advances in neural information processing systems*, 28, 2015.

[44] Thomas Miconi. Biologically plausible learning in recurrent neural networks reproduces neural dynamics observed during cognitive tasks. *Elife*, 6:e20899, 2017.

[45] Sun Minni, Li Ji-An, Theodore Moskovitz, Grace Lindsay, Kenneth Miller, Mario Dipoppa, and Guangyu Robert Yang. Understanding the functional and structural differences across excitatory and inhibitory neurons. *bioRxiv*, page 680439, 2019.

[46] Alessandro Ingrosso and LF Abbott. Training dynamically balanced excitatory-inhibitory networks. *PloS one*, 14(8):e0220547, 2019.

[47] Wilten Nicola and Claudia Clopath. Supervised learning in spiking neural networks with force training. *Nature communications*, 8(1):2208, 2017.

[48] Yann LeCun, John Denker, and Sara Solla. Optimal brain damage. *Advances in neural information processing systems*, 2, 1989.

[49] Eli Moore and Rishidev Chaudhuri. Using noise to probe recurrent neural network structure and prune synapses. *Advances in neural information processing systems*, 33:14046–14057, 2020.

[50] H Francis Song, Guangyu R Yang, and Xiao-Jing Wang. Training excitatory-inhibitory recurrent neural networks for cognitive tasks: a simple and flexible framework. *PLoS computational biology*, 12(2):e1004792, 2016.

[51] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986.

[52] Lixiao Zhang, Xianwei Wang, Ramón Cueto, Comfort Effi, Yuling Zhang, Hongmei Tan, Xuebin Qin, Yong Ji, Xiaofeng Yang, and Hong Wang. Biochemical basis and metabolic interplay of redox regulation. *Redox biology*, 26:101284, 2019.

[53] Christian Tetzlaff, Christoph Kolodziejski, Marc Timme, and Florentin Wörgötter. Synaptic scaling in combination with many generic plasticity mechanisms stabilizes circuit connectivity. *Frontiers in computational neuroscience*, 5:47, 2011.

[54] Peter R Huttenlocher et al. Synaptic density in human frontal cortex-developmental changes and effects of aging. *Brain Res*, 163(2):195–205, 1979.

[55] Ed Bullmore and Olaf Sporns. The economy of brain network organization. *Nature reviews neuroscience*, 13(5):336–349, 2012.

[56] Marina Garrett, Sahar Manavi, Kate Roll, Douglas R Ollerenshaw, Peter A Groblewski, Nicholas D Ponvert, Justin T Kiggins, Linzy Casal, Kyla Mace, Ali Williford, et al. Experience shapes activity dynamics and stimulus coding of vip inhibitory cells. *elife*, 9:e50340, 2020.

[57] Rajesh PN Rao and Dana H Ballard. Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects. *Nature neuroscience*, 2(1):79–87, 1999.

[58] Jeffrey L Elman. Finding structure in time. *Cognitive science*, 14(2):179–211, 1990.

[59] Stephen P Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[60] Dimitri P. Bertsekas. *Nonlinear Programming*. Athena Scientific, third edition, 2016.

[61] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256. JMLR Workshop and Conference Proceedings, 2010.

[62] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.

[63] Pingsheng Li, Jonathan Cornford, Arna Ghosh, and Blake Richards. Learning better with dale's law: A spectral perspective. *Advances in Neural Information Processing Systems*, 36, 2024.

[64] Donald Olding Hebb. *The organization of behavior: A neuropsychological theory*. Psychology press, 2005.

[65] Jong Chul Ye. *Geometry of Deep Learning*. Springer, 2022.

[66] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

[67] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

[68] Michael C Mozer and Paul Smolensky. Skeletonization: A technique for trimming the fat from a network via relevance assessment. *Advances in neural information processing systems*, 1, 1988.

[69] Stephen Hanson and Lorien Pratt. Comparing biases for minimal network construction with back-propagation. *Advances in neural information processing systems*, 1, 1988.

[70] Daniel A Spielman and Nikhil Srivastava. Graph sparsification by effective resistances. In *Proceedings of the fortieth annual ACM symposium on Theory of computing*, pages 563–568, 2008.

[71] Daniel A Spielman and Shang-Hua Teng. Spectral sparsification of graphs. *SIAM Journal on Computing*, 40(4):981–1025, 2011.

[72] Joshua Batson, Daniel A Spielman, Nikhil Srivastava, and Shang-Hua Teng. Spectral sparsification of graphs: theory and algorithms. *Communications of the ACM*, 56(8):87–94, 2013.

[73] Aishwarya Balwani and Jakob Krzyston. Zeroth-order topological insights into iterative magnitude pruning. In *Topological, Algebraic and Geometric Learning Workshops 2022*, pages 6–16. PMLR, 2022.

[74] Bastian Rieck, Matteo Togninalli, Christian Bock, Michael Moor, Max Horn, Thomas Gumbsch, and Karsten Borgwardt. Neural persistence: A complexity measure for deep neural networks using algebraic topology. *arXiv preprint arXiv:1812.09764*, 2018.

[75] Harish Doraiswamy, Julien Tierny, Paulo JS Silva, Luis Gustavo Nonato, and Claudio Silva. Topomap: A 0-dimensional homology preserving projection of high-dimensional data. *IEEE Transactions on Visualization and Computer Graphics*, 27(2):561–571, 2020.

[76] Théo Lacombe, Yuichi Ike, Mathieu Carriere, Frédéric Chazal, Marc Glisse, and Yuhei Umeda. Topological uncertainty: Monitoring trained neural networks through persistence of activation graphs. *arXiv preprint arXiv:2105.04404*, 2021.

[77] Karl Friston. A theory of cortical responses. *Philosophical transactions of the Royal Society B: Biological sciences*, 360(1456):815–836, 2005.

[78] Visual behavior - 2p - brain-map.org. `https://portal.brain-map.org/circuits-behavior/visual-behavior-2p`. (Accessed on 11/20/2024).

[79] Marina Garrett, Peter Groblewski, Alex Piet, Doug Ollerenshaw, Farzaneh Najafi, Iryna Yavorska, Adam Amster, Corbett Bennett, Michael Buice, Shiella Caldejon, et al. Stimulus novelty uncovers coding diversity in visual cortical circuits. *bioRxiv*, pages 2023–02, 2023.

[80] Peter A Groblewski, Douglas R Ollerenshaw, Justin T Kiggins, Marina E Garrett, Chris Mochizuki, Linzy Casal, Sissy Cross, Kyla Mace, Jackie Swapp, Sahar Manavi, et al. Characterization of learning, motivation, and visual perception in five transgenic mouse lines expressing gcamp in distinct cell populations. *Frontiers in Behavioral Neuroscience*, 14:104, 2020.

[81] Rodney J Douglas, Kevan AC Martin, and David Whitteridge. A canonical microcircuit for neocortex. *Neural computation*, 1(4):480–488, 1989.

[82] Vernon B Mountcastle. The columnar organization of the neocortex. *Brain: a journal of neurology*, 120(4):701–722, 1997.

[83] Andre M Bastos, W Martin Usrey, Rick A Adams, George R Mangun, Pascal Fries, and Karl J Friston. Canonical microcircuits for predictive coding. *Neuron*, 76(4):695–711, 2012.

[84] Aishwarya Balwani, Suhee Cho, and Hannah Choi. Exploring the architectural biases of the canonical cortical microcircuit. *bioRxiv*, 2024.

[85] Luke Campagnola, Stephanie C Seeman, Thomas Chartrand, Lisa Kim, Alex Hoggarth, Clare Gamlin, Shinya Ito, Jessica Trinh, Pasha Davoudian, Cristina Radaelli, et al. Local connectivity and synaptic dynamics in mouse and human neocortex. *Science*, 375(6585):eabj5861, 2022.

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

[87] Conrado A Bosman, Jan-Mathijs Schoffelen, Nicolas Brunet, Robert Oostenveld, Andre M Bastos, Thilo Womelsdorf, Birthe Rubehn, Thomas Stieglitz, Peter De Weerd, and Pascal Fries. Attentional stimulus selection through selective synchronization between monkey visual areas. *Neuron*, 75(5):875–888, 2012.

[88] João D Semedo, Anna I Jasper, Amin Zandvakili, Aravind Krishna, Amir Aschner, Christian K Machens, Adam Kohn, and Byron M Yu. Feedforward and feedback interactions between visual cortical areas use different population activity patterns. *Nature communications*, 13(1):1099, 2022.

[89] Joon-Young Moon, Kathrin Müsch, Charles E Schroeder, Taufik A Valiante, and Christopher J Honey. Inter-regional delays fluctuate in the human cerebral cortex. *bioRxiv*, pages 2022–06, 2022.

[90] Farzaneh Najafi, Simone Russo, and Jerome Lecoq. Unexpected events modulate context signaling in vip and excitatory cells of the visual cortex. *bioRxiv*, pages 2024–05, 2024.

[91] Loreen Hertäg and Henning Sprekeler. Learning prediction error neurons in a canonical interneuron circuit. *Elife*, 9:e57541, 2020.

[92] Joost Haarsma, PC Fletcher, JD Griffin, HJ Taverne, Hisham Ziauddeen, TJ Spencer, Chantal Miller, Teresa Katthagen, I Goodyer, KMJ Diederen, et al. Precision weighting of cortical unsigned prediction error signals benefits learning, is mediated by dopamine, and is impaired in psychosis. *Molecular psychiatry*, 26(9):5320–5333, 2021.

[93] Clara Kwon Starkweather and Naoshige Uchida. Dopamine signals as temporal difference errors: recent advances. *Current Opinion in Neurobiology*, 67:95–105, 2021.

[94] Loreen Hertäg and Claudia Clopath. Prediction-error neurons in circuits with multiple neuron types: Formation, refinement, and functional implications. *Proceedings of the National Academy of Sciences*, 119(13):e2115699119, 2022.

[95] Blake A Richards and Timothy P Lillicrap. Dendritic solutions to the credit assignment problem. *Current opinion in neurobiology*, 54:28–36, 2019.

[96] Jacob A. Westerberg, Yihan S. Xiong, Hamed Nejat, Eli Sennesh, Séverine Durand, Hannah Cabasco, Hannah Belski, Ryan Gillis, Henry Loeffler, Ahad Bawany, Carter R. Peene, Warren Han, Katrina Nguyen, Vivian Ha, Tye Johnson, Conor Grasso, Ben Hardcastle, Ahrial Young, Jackie Swapp, Ben Ouellete, Shiella Caldejon, Ali Williford, Peter A. Groblewski, Shawn R. Olsen, Carly Kiselycznyk, Jerome A. Lecoq, Alexander Maier, and André M. Bastos. Stimulus history, not expectation, drives sensory prediction errors in mammalian cortex. *bioRxiv*, 2024.

[97] Shohei Furutachi, Alexis D. Franklin, Andreea M. Aldea, Thomas Mrsic-Flogel, and Sonja B. Hofer. Cooperative thalamocortical circuit mechanism for sensory prediction errors. *Advances in neural information processing systems*, 633:398—-406, 2024.

[98] Rolf Schneider. *Convex bodies: the Brunn–Minkowski theory*, volume 151. Cambridge university press, 2013.

[99] Hyunjik Kim, George Papamakarios, and Andriy Mnih. The lipschitz constant of self-attention. In *International Conference on Machine Learning*, pages 5562–5571. PMLR, 2021.

[100] Herbert Federer. *Geometric measure theory*. Springer, 2014.

Appendix

6 Dale's backpropagation update

This section studies in detail the Dale's backpropagation update rule, beginning with the explicit derivation of the same. The following subsections detail the algorithm for implementing this update in a gradient descent framework, and provide proofs regarding the optimality of the resulting weight matrix projection under the Frobenius norm.

6.1 Derivation

$$\begin{aligned}
W_D^{(i+1)} &= \mathcal{P}_{\mathcal{C}}\left(W^{(i+1)}\right) \\
&= \mathcal{P}_{\mathcal{C}}\left(W_D^{(i)} - \eta\nabla\ell\left(W_D^{(i)}\right)\right) \\
&= \mathcal{P}_{\mathcal{C}}\left(W_{D[N^+]}^{(i)} - \eta\nabla\ell\left(W_{D[N^+]}^{(i)}\right)\right) \oplus \mathcal{P}_{\mathcal{C}}\left(W_{D[N^-]}^{(i)} - \eta\nabla\ell\left(W_{D[N^-]}^{(i)}\right)\right) \\
&= \max\left(0, W_{D[N^+]}^{(i)} - \eta\nabla\ell\left(W_{D[N^+]}^{(i)}\right)\right) \oplus \min\left(0, W_{D[N^-]}^{(i)} - \eta\nabla\ell\left(W_{D[N^-]}^{(i)}\right)\right) \\
&= \max\left(0, W_{[N^+]}^{(i+1)}\right) \oplus \min\left(0, W_{[N^-]}^{(i+1)}\right)
\end{aligned}$$

6.2 Algorithm

Algorithm 2: Dale's Backpropagation Update Rule (under the gradient descent optimization scheme)

Input: Initial weights $W_D^{(0)}$, step size η, maximum iterations K
Output: Final weights $W_D^{(K)}$
for $k = 0$ **to** $K - 1$ **do**
  Compute gradient $\nabla\ell\left(W_D^{(k)}\right)$;
  $W^{(k+1)} \leftarrow W_D^{(k)} - \eta\nabla\ell\left(W_D^{(k)}\right)$; `// Compute weight updates with backpropagation`
  for *each component* j **do**
    if *sign* $\left(W_j^{(k+1)}\right) =$ *sign* $\left(W_{D_j}^{(k)}\right)$ **then**
      $W_{D_j}^{(k+1)} \leftarrow W_j^{(k+1)}$; `// Keep weight update if sign constraint is respected`
    else
      $W_{D_j}^{(k+1)} \leftarrow 0$; `// Set weight to 0 if sign constraint is violated`
    end
  end
end
return $W_D^{(K)}$

6.3 Closest sign-constrained projection under the Frobenius norm

Theorem 5 (Dale's backpropagation provides the closest sign-constrained projection of W under the Frobenius norm)**.** *Let $W \in \mathbb{R}^{N\times N}$ be a real square matrix. Define the set $S \subset \mathbb{R}^{N\times N}$ as, (i) Columns 1 to k: All entries are non-negative (≥ 0), (ii) Columns $k+1$ to N: All entries are non-positive (≤ 0). Then, the matrix W_D obtained by applying the Dale's backprop update to W is the closest projection of W onto S under the Frobenius norm. That is,*

$$W_D = \arg\min_{X\in S} \|W - X\|_F.$$

Proof. To find the projection of W onto S under the Frobenius norm, we need to solve:

$$\min_{X\in S} \|W - X\|_F^2 = \min_{X\in S} \sum_{i=1}^{N}\sum_{j=1}^{N}(W_{ji} - X_{ji})^2.$$

Since the Frobenius norm is separable over the entries of W and X, we can minimize each $(W_{ji} - X_{ij})^2$ independently, subject to the sign constraints on X_{ji}:

- **For columns** 1 **to** k**:** The constraint is $X_{ji} \geq 0$.
- **For columns** $k+1$ **to** N**:** The constraint is $X_{ji} \leq 0$.

Minimization for each X_{ji}**:**

- **If** $j \leq k$**:**

$$X_{ij}^* = \arg\min_{x \geq 0} (W_{ji} - x)^2.$$

The solution is:

$$X_{ji}^* = \begin{cases} W_{ji}, & \text{if } W_{ji} \geq 0, \\ 0, & \text{if } W_{ji} < 0. \end{cases}$$

- **If** $j > k$**:**

$$X_{ji}^* = \arg\min_{x \leq 0} (W_{ji} - x)^2.$$

The solution is:

$$X_{ji}^* = \begin{cases} W_{ji}, & \text{if } W_{ji} \leq 0, \\ 0, & \text{if } W_{ij} > 0. \end{cases}$$

Therefore, the optimal X_{ij}^* corresponds exactly to the entries of W_D obtained by Dale's backprop and W_D is the projection of W onto S under the Frobenius norm. □

Corollary 6. *Let W_R be the matrix obtained from W using rectified backprop. Then, the Dale's backprop matrix W_D is closer to W than W_R is to W under the Frobenius norm:*

$$\|W - W_D\|_F \leq \|W - W_R\|_F.$$

Proof. By Theorem 5, W_D is the closest matrix in S to W under the Frobenius norm. Since $W_R \in S$, it follows that:

$$\|W - W_D\|_F \leq \|W - W_R\|_F$$

□

7 Alignment of Dale's backpropagation and Hebbian learning in reinforcing high-magnitude weights

We show that despite the differences in their explicit formulations, both Hebbian learning and Dale's backpropagation tend to strengthen (i.e., increase the magnitude of) similar weights. In particular, the weights strengthened by Hebbian learning form a subset of those strengthened via the gradient-based Dale's backprop. Given that weights of higher magnitude inevitably influence the functional connectivity amongst the neurons, preserving weights of higher magnitudes implies preserving those weights which would've been important from a statistical learning perspective as well being biologically relevant.

Learning requires a weight w_{ji} joining pre-synaptic neuron i to post-synaptic neuron j be changed according to the rule

$$w_{ji}^{(k+1)} = w_{ji}^{(k)} + \Delta w_{ji}^{(k)}$$

Hebbian learning postulates the update Δw_{ji} is given as

$$\Delta w_{ji_{Hebb}} = \eta \cdot a_i \cdot a_j$$

where $a_i,\ a_j$ are the activations of neurons i, j respectively while η is the learning rate.

On the other hand, the backpropagation update (without loss of generality, in the absence of bias terms) is given as

$$\Delta w_{ji_{BP}} = -\eta \cdot \frac{\partial \ell}{\partial w_{ji}} = -\eta \cdot \underbrace{\frac{\partial \ell}{\partial a_j} \cdot \phi'\left(\sum_i w_{ji} a_i\right)}_{\varepsilon_j} \cdot a_i$$

where ℓ is the loss function, ϕ is the activation function, ε_j is the error corresponding to neuron j computed using the chain rule, and a_i has the same meaning as before.

In Dale's backpropagation, we constrain all activations to be non-negative through a thresholding operation, and weights are restricted to maintain their assigned signs. Under these constraints, the following statements hold true:

Analysis for $w_{ji} \geq 0$: In the case of Hebbian learning, given our construction, if w_{ji} is non-negative, we would need both a_i, a_j to be positive to increase $|w_{ji}|$.

For Dale's backprop, for a weight $w_{ji} \geq 0$ to increase in magnitude, we require that

$$\frac{\partial \ell}{\partial w_{ji}} < 0 \implies \frac{\partial \ell}{\partial a_j} < 0.$$

since $\phi'(\cdot)$ is always non-negative for monotonically-increasing ϕ such as ReLU and $\tanh$. This means that as ℓ decreases, the neuron a_j contributes positively to reducing the loss. In turn, as learning progresses and reduces the loss ℓ, this would lead to an increase in a_j when $a_j > 0$, matching Hebbian learning.

Analysis for $w_{ji} \leq 0$: In the case of Hebbian learning, if w_{ji} is non-positive, we would require that the *action* of $a_i \leq 0$ and $a_j \geq 0$ to increase $|w_{ji}|$.

In the case of Dale's backprop, for a weight $w_{ji} \leq 0$ to increase in magnitude, we now require

$$\frac{\partial \ell}{\partial w_{ji}} > 0 \implies \frac{\partial \ell}{\partial a_j} > 0.$$

Since $\frac{\partial \ell}{\partial a_j} > 0$ indicates that increasing a_j would increase the loss, the learning process will instead push to decrease a_j. Strengthening a negative weight (making it more negative) lowers $z_j = \sum_i w_{ji} a_i$ when $a_i \geq 0$, thereby reducing $a_j = \phi(z_j)$ and in turn helping to reduce the loss ℓ.

This correspondence between Dale's backprop and Hebbian learning, facilitated by the non-negative activation constraint, suggests that weights strengthened during learning align with those of biological significance. Consequently, when these weights are preferentially retained by our pruning rule, we preserve functionally important connectivity patterns that emerge through biologically plausible learning dynamics.

8 Theoretical guarantees for Dale's backpropagation

8.1 Analyzing convergence of Dale's backpropagation under the restricted optimum assumption

Lemma (Optimal sign pattern preservation)**.** *Let the vector of learnt weights be $W \in \mathbb{R}^n$ with the components w_j, where $j \in \{1, 2, ..., n\}$. Let L be the Lipschitz constant for the gradients $\nabla\ell(W)$, where ℓ is a loss function. Given a gradient descent-based, component-wise sign-preserving learning rule that uses the projection operator $\mathcal{P}_{\mathcal{C}} : \mathbb{R}^n \mapsto \mathbb{R}^n$ defined as*

$$\mathcal{P}_{\mathcal{C}}(w_j) = \begin{cases} w_j & \text{if } sign(z_j) = sign\left(w_j\right) \\ \\ 0 & \text{if } sign(z_j) \neq sign\left(w_j\right) \end{cases}$$

where $z_j = w_j - \frac{1}{L}\nabla\ell\left(w_j\right), sign(z_j) = \frac{z_j}{|z_j|}$ for $z_j \neq 0$, and $sign(0) = 0$. If $sign(W^*) = sign\left(W^{(0)}\right)$ where W^* are the set of weights that can achieve the optimal loss on ℓ, it holds that for any iteration i of regular gradient descent

$$sign\left(W^{(i)}\right) = sign\left(W^{(0)}\right) = sign\left(W^*\right)\ \forall i \in \mathbb{N},\ \text{and } \mathcal{P}_{\mathcal{C}}(w_j) = w_j \text{ for } j \in \{1, 2, ..., n\}.$$

Proof. We show by induction that $sign\left(W^{(i)}\right) = sign\left(W^{(0)}\right) = sign\left(W^*\right)\ \forall i \in \mathbb{N}$.

Base case ($i = 0$): The statement trivially holds true since $sign\left(W^{(0)}\right) = sign\left(W^*\right)$, by assumption.

Inductive hypothesis: For some iteration $i > 0, sign\left(W^{(i)}\right) = sign\left(W^{(0)}\right) = sign\left(W^*\right)$.

To show that for the iteration $i + 1$ it also holds that $sign\left(W^{(i+1)}\right) = sign\left(W^{(0)}\right) = sign\left(W^*\right)$, we consider z_j as defined, which is the j^{th} component of

$$z = W - \frac{1}{L}\nabla\ell(W).$$

By the Lipschitz continuity of the gradient, we have that

$$\left\|\nabla\ell\left(w_j^{(i)}\right) - \nabla\ell\left(w_j^*\right)\right\|_2 \leq L\left\|w_j^{(i)} - w_j^*\right\|_2$$

Since W^* is the optimal set of weights, we know that $\nabla\ell(w_j^*) = 0,\ \forall\ j \in \{1, 2, ..., n\}$. Therefore,

$$\left\|\nabla\ell\left(w_j^{(i)}\right)\right\|_2 \leq L\left\|w_j^{(i)} - w_j^*\right\|_2 \implies \frac{1}{L}\left|\nabla\ell\left(w_j^{(i)}\right)\right| \leq \left|w_j^{(i)} - w_j^*\right|$$

Consider the case where $w_j^{(i)} < w_j^*$.

Here, $w_j^{(i)} - w_j^* < 0 \implies \left|w_j^{(i)} - w_j^*\right| = -w_j^{(i)} + w_j^*$. Furthermore, the gradient descent update moves $w_j^{(i)}$ towards w_j^* by increasing its value, implying that $\nabla\ell\left(w_j^{(i)}\right)$ itself is negative. Consequently,

$$\begin{aligned} z_j^{(i)} &= w_j^{(i)} - \frac{1}{L}\nabla\ell\left(w_j^{(i)}\right) \\ &\leq w_j^{(i)} + (-w_j^{(i)} + w_j^*) \\ &= w_j^* \end{aligned}$$

This leads us to the conclusion that $w_j^{(i)} < z_j^{(i)} < w_j^*$.

Since $sign\left(w_j^{(i)}\right) = sign\left(w_j^*\right)$ by the induction hypothesis, $sign\left(w_j^{(i)}\right) = sign\left(z_j^{(i)}\right) = sign\left(w_j^*\right)$ also holds. As a result, $\mathcal{P}_{\mathcal{C}}\left(z_j^{(i)}\right) = z_j^{(i)}$ and $sign\left(w^{(i+1)}\right) = sign\left(w_j^{(i)}\right) = sign\left(w_j^{(0)}\right) = sign\left(w_j^*\right)$.

The case where $w_j^{(i)} > w_j^*$ follows similarly, with the difference that since the gradient $\nabla\ell\left(w_j^{(i)}\right)$ is positive and $\left|w_j^{(i)} - w_j^*\right| = w_j^{(i)} - w_j^*$, we instead have $w_j^{(i)} > z_j^{(i)} > w_j^*$. This leads to the same results as before, i.e.,

$\mathcal{P}_{\mathcal{C}}\left(z_j^{(i)}\right) = z_j^{(i)}$ and $sign\left(w^{(i+1)}\right) = sign\left(w_j^{(i)}\right) = sign\left(w_j^{(0)}\right) = sign\left(w_j^*\right)$.

As the choice of the index j was arbitrary, these results holds across all indices and therefore

$$sign\left(W^{(i)}\right) = sign\left(W^{(0)}\right) = sign\left(W^*\right)\ \forall i \in \mathbb{N}, \text{ and } \mathcal{P}_{\mathcal{C}}(w_j) = w_j \text{ for } j \in \{1, 2, ..., n\}.$$

□

Theorem (Convergence of Dale's Backpropagation)**.** *Let ℓ be a loss function satisfying the $\mu-$Polyak-Łojasiewicz condition, with gradients that are L-Lipschitz such that $L \geq \mu > 0$. Consider the sequence of weights $\{W_D^{(i)}\}$ generated according to the Dale's backpropagation update, with a step size of $\frac{1}{L}$. Given an optimal loss $\ell^* = \ell(W^*) = \operatorname{argmin} \ell(W_D)$ where W^* has the same sign pattern as all $W_D^{(i)}$ and a specific error $\varepsilon > 0$, it holds for the iteration i that*

$$\ell(W_D^{(i)}) - \ell^* \leq \varepsilon \text{ when } i \geq \frac{\log\left(\frac{\ell(W_D^{(0)}) - \ell^*}{\varepsilon}\right)}{\log\left(\frac{L}{L-\mu}\right)}$$

Proof. By Lemma 7 we note that the function $g(W)$ is convex, when it is defined as

$$g(W) = \frac{L}{2}\|W\|_2^2 - \ell(W)$$

Furthermore, by the first-order equivalence of convexity on $g(W)$, we have

$$g(W') \geq g(W) + \langle \nabla g(W), W' - W \rangle \ \ \forall\, W, W'$$

This subsequently implies that

$$\frac{L}{2}\|W'\|_2^2 - \ell(W') \geq -\frac{L}{2}\|W\|_2^2 - \ell(W) + L\langle W', W\rangle - \langle W' - W, \nabla\ell(W)\rangle$$

Rearranging terms, we have

$$\ell(W') \leq \ell(W) + \langle \nabla\ell(W), W' - W\rangle + \frac{L}{2}\|W' - W\|_2^2 \tag{7}$$

Setting $W' = W_D^{(i+1)}$ and $W = W_D^{(i)}$ in Eq. 7 while using the Dale's backprop update rule, we get

$$\ell\left(W_D^{(i+1)}\right) - \ell\left(W_D^{(i)}\right) \leq \left\langle \nabla\ell\left(W_D^{(i)}\right), W_D^{(i+1)} - W_D^{(i)}\right\rangle + \frac{L}{2}\left\|W_D^{(i+1)} - W_D^{(i)}\right\|_2^2 \tag{8}$$

Defining $z = W_D^{(i)} - \frac{1}{L}\nabla\ell\left(W_D^{(i)}\right)$ and the projection operator $\mathcal{P}_{\mathcal{C}}$ as before, Eq. 8 can be re-written as

$$\begin{aligned}\ell\left(W_D^{(i+1)}\right) - \ell\left(W_D^{(i)}\right) &\leq \left\langle \nabla\ell\left(W_D^{(i)}\right), \mathcal{P}_{\mathcal{C}}(z) - W_D^{(i)}\right\rangle + \frac{L}{2}\left\|\mathcal{P}_{\mathcal{C}}(z) - W_D^{(i)}\right\|_2^2 \\ &= \underbrace{\left\langle \nabla\ell\left(W_D^{(i)}\right), \mathcal{P}_{\mathcal{C}}(z) - z\right\rangle}_{\text{Term 1}} + \underbrace{\left\langle \nabla\ell\left(W_D^{(i)}\right), z - W_D^{(i)}\right\rangle}_{\text{Term 2}} + \underbrace{\frac{L}{2}\left\|\mathcal{P}_{\mathcal{C}}(z) - W_D^{(i)}\right\|_2^2}_{\text{Term 3}}\end{aligned}$$

By Lemma 1 we note that **Term 1** is always 0 since $\mathcal{P}_{\mathcal{C}}(z) = z$.

Re-substituting $z = W_D^{(i)} - \frac{1}{L}\nabla\ell\left(W_D^{(i)}\right)$ in **Term 2** simplifies it to $-\frac{1}{L}\left\|\nabla\ell\left(W_D^{(i)}\right)\right\|_2^2$

Due to the non-expansive property of metric projections onto convex sets (Theorem 1.2.1 of [98]) and the fact that $\mathcal{P}_{\mathcal{C}}\left(W_D^{(i)}\right) = W_D^{(i)}$ it holds for **Term 3** that

$$\left\|\mathcal{P}_{\mathcal{C}}(z) - \mathcal{P}_{\mathcal{C}}\left(W_D^{(i)}\right)\right\|_2 \leq \left\|z - W_D^{(i)}\right\|_2 = \frac{1}{L}\left\|\nabla\ell\left(W_D^{(i)}\right)\right\|_2$$

Combining the three terms, we get the bound

$$\ell\left(W_D^{(i+1)}\right) - \ell\left(W_D^{(i)}\right) \leq -\frac{1}{L}\left\|\nabla\ell\left(W_D^{(i)}\right)\right\|_2^2 + \frac{1}{2L}\left\|\nabla\ell\left(W_D^{(i)}\right)\right\|_2^2$$
$$= -\frac{1}{2L}\left\|\nabla\ell\left(W_D^{(i)}\right)\right\|_2^2$$

Using the Polyak-Łojasiewicz inequality (Def. 8.1) we get

$$\ell\left(W_D^{(i+1)}\right) - \ell\left(W_D^{(i)}\right) \leq -\frac{\mu}{L}\left(\ell\left(W_D^{(i)}\right) - \ell^*\right)$$

Rearranging and subtracting ℓ^* from both sides gives us

$$\ell\left(W_D^{(i+1)}\right) - \ell^* \leq \left(1 - \frac{\mu}{L}\right)\left(\ell\left(W_D^{(i)}\right) - \ell^*\right) \tag{9}$$

Applying Eq. 9 recursively gives us the result

$$\ell\left(W_D^{(i)}\right) - \ell^* \leq \left(1 - \frac{\mu}{L}\right)^i\left(\ell\left(W_D^{(0)}\right) - \ell^*\right) \tag{10}$$

Let the error ε be defined as $\varepsilon = \ell\left(W_D^{(i)}\right) - \ell^*$ thereby simplifying Eq. 10 to

$$\varepsilon \leq \left(1 - \frac{\mu}{L}\right)^i\left(\ell\left(W_D^{(0)}\right) - \ell^*\right)$$

Taking the logarithm on both sides we get

$$\log(\varepsilon) \leq i \cdot \log\left(1 - \frac{\mu}{L}\right) + \log\left(\ell\left(W_D^{(0)}\right) - \ell^*\right)$$

Rearranging the terms finally gives us the bound

$$i \geq \frac{\log\left(\frac{\ell(W_D^{(0)}) - \ell^*}{\varepsilon}\right)}{\log\left(\frac{L}{L-\mu}\right)}$$

for the target error $\ell(W_D^{(i)}) - \ell^* \leq \varepsilon$.

□

Definition 8.1 (Polyak-Łojasiewicz condition)**.** *A loss function ℓ is said to satisfy the Polyak-Łojasiewicz condition if for some $\mu > 0$ it holds that:*

$$\frac{1}{2}\left\|\nabla\ell(W)\right\|_2^2 \geq \mu(\ell(W) - \ell^*)\ \forall\ W$$

where $\ell^* = \text{argmin}\ \ell(W)$ is the optimal loss attainable.

Lemma 7 (Convexity of transformed function, Lemma 11.1 of [65])**.** *If the gradient of a loss function $\ell(W)$ is L-Lipschitz, then the transformed function g is convex, where $g : \mathbb{R}^n \mapsto \mathbb{R}$ is defined as*

$$g(W) := \frac{L}{2}\left\|W\right\|_2^2 - \ell(W)$$

Proof. Since $\nabla\ell(W)$ is L-Lipschitz, we have

$$\left\|\nabla\ell(W) - \nabla\ell(W')\right\|_2 \leq L\left\|W - W'\right\|_2\ \forall\ W, W'$$

By the Cauchy-Schwarz inequality, we then have

$$\langle\nabla\ell(W) - \nabla\ell(W'), W - W'\rangle \leq L\left\|W - W'\right\|_2^2\ \forall\ W, W'$$

Rearranging terms,

$$0 \leq -\langle\nabla\ell(W) - \nabla\ell(W'), W - W'\rangle + L\left\|W - W'\right\|_2^2$$
$$= \langle W - W', L(W - W') - \nabla\ell(W) + \nabla\ell(W')\rangle$$

Substituting $g(W) = \frac{L}{2}\left\|W\right\|_2^2 - \ell(W)$ and $\nabla g(W) = LW - \nabla\ell(W)$, we get

$$0 \leq \langle W - W', \nabla g(W) - \nabla g(W')\rangle\ \forall\ W, W'$$

By the monotonicity of the gradient, $g(W)$ is convex. □

8.2 Analyzing Dale's backprop w.r.t. standard backpropagation

Lemma (Distance between learnt weights)**.** *Let $W^{(i)}$ and $W_D^{(i)}$ be the weights at iteration i for standard backpropagation and Dale's backpropagation, respectively. Assume the gradients $\nabla\ell(W)$ and $\nabla\ell(W_D)$ are upper bounded in magnitude by G and Lipschitz continuous with constant L. Then, the distance between the two sets of weights at any iteration i, denoted as $\|\delta^{(i)}\|_2 = \|W^{(i)} - W_D^{(i)}\|_2$, is bounded by:*

$$\|\delta^{(i)}\|_2 \leq \frac{G}{L}\left((1+\eta L)^i - 1\right)$$

where η is the learning rate.

Proof. Consider the case where the weights of the network are updated using gradient descent as the optimizer. This implies the following update rules at any iteration i:

1. Standard backpropagation update: $W^{(i)} = W^{(i-1)} - \eta\nabla\ell(W)^{(i-1)}$
2. Dale's backpropagation update: $W_D^{(i)} = \mathcal{P}_{\mathcal{C}}\left(W_D^{(i-1)} - \eta\nabla\ell(W_D)^{(i-1)}\right)$

Let $\|\delta^{(i)}\|_2 = \|W^{(i)} - W_D^{(i)}\|_2$ be the distance between the two sets of weights at iteration i. We can bound this as:

$$\begin{aligned}
\left\|\delta^{(i)}\right\|_2 &= \left\|W^{(i)} - W_D^{(i)}\right\|_2 \\
&= \left\|W^{(i)} - \left(W_D^{(i-1)} - \eta\nabla\ell(W_D)^{(i-1)}\right) + \left(W_D^{(i-1)} - \eta\nabla\ell(W_D)^{(i-1)}\right) - W_D^{(i)}\right\|_2 \\
&\leq \left\|W^{(i)} - \left(W_D^{(i-1)} - \eta\nabla\ell(W_D)^{(i-1)}\right)\right\|_2 + \left\|\left(W_D^{(i-1)} - \eta\nabla\ell(W_D)^{(i-1)}\right) - W_D^{(i)}\right\|_2 \\
&= \underbrace{\left\|W^{(i-1)} - \eta\nabla\ell(W)^{(i-1)} - \left(W_D^{(i-1)} - \eta\nabla\ell(W_D)^{(i-1)}\right)\right\|_2}_{\text{Term 1}} \\
&\quad + \underbrace{\left\|\left(W_D^{(i-1)} - \eta\nabla\ell(W_D)^{(i-1)}\right) - \mathcal{P}_{\mathcal{C}}\left(W_D^{(i-1)} - \eta\nabla\ell(W_D)^{(i-1)}\right)\right\|_2}_{\text{Term 2}}
\end{aligned}$$

We now bound each term separately.

Bounding Term 1:

$$\begin{aligned}
&\left\|W^{(i-1)} - \eta\nabla\ell(W)^{(i-1)} - \left(W_D^{(i-1)} - \eta\nabla\ell(W_D)^{(i-1)}\right)\right\|_2 \\
&\leq \left\|W^{(i-1)} - W_D^{(i-1)}\right\|_2 + \left\|\eta\nabla\ell(W_D)^{(i-1)} - \eta\nabla\ell(W)^{(i-1)}\right\|_2 \text{ (by Triangle inequality)} \\
&= \left\|\delta^{(i-1)}\right\|_2 + \eta\left\|\nabla\ell(W_D)^{(i-1)} - \nabla\ell(W)^{(i-1)}\right\|_2 \\
&\leq \left\|\delta^{(i-1)}\right\|_2 + \eta L\left\|W_D^{(i-1)} - W^{(i-1)}\right\|_2 \quad \text{(by Lipschitz continuity)} \\
&= \left\|\delta^{(i-1)}\right\|_2 + \eta L\left\|\delta^{(i-1)}\right\|_2 \\
&= \left\|\delta^{(i-1)}\right\|_2 (1+\eta L)
\end{aligned}$$

Bounding Term 2: The difference between the update using gradient descent before and after the projection step $\mathcal{P}_{\mathcal{C}}$ at iteration $(i-1)$ will never exceed $\eta\nabla\ell(W_D)^{(i-1)}$ when the update pushes weights W_D outside the feasible region. Therefore,

$$\left\|\left(W_D^{(i-1)} - \eta\nabla\ell(W_D)^{(i-1)}\right) - \mathcal{P}_{\mathcal{C}}\left(W_D^{(i-1)} - \eta\nabla\ell(W_D)^{(i-1)}\right)\right\|_2 \leq \left\|\eta\nabla\ell(W_D)^{(i-1)}\right\|_2 \leq \eta G$$

Combining the two bounds, we get:

$$\left\|\delta^{(i)}\right\|_2 \leq \left\|\delta^{(i-1)}\right\|_2 (1+\eta L) + \eta G$$

This forms a recurrence relation as follows

$$\begin{aligned}\left\|\delta^{(1)}\right\|_2 &\leq \eta G \\ \left\|\delta^{(2)}\right\|_2 &\leq \eta G(1+\eta L)+\eta G \\ \left\|\delta^{(3)}\right\|_2 &\leq \eta G(1+\eta L)^2+\eta G(1+\eta L)+\eta G \\ &\vdots\end{aligned}$$

More generally, for any iteration i,

$$\left\|\delta^{(i)}\right\|_2 \leq \eta G \sum_{k=0}^{i-1}(1+\eta L)^k$$

This is a geometric series with ratio $(1+\eta L)$ and i terms. Summing the series we get

$$\left\|\delta^{(i)}\right\|_2 \leq \eta G \cdot \frac{(1+\eta L)^i-1}{(1+\eta L)-1} = \frac{G}{L}\left((1+\eta L)^i-1\right)$$

□

Theorem (Differences in errors between solutions)**.** *Let $f(W)$ be the function represented by a single-layer RNN unrolled over T timesteps, with weights W. Let W_D be the weights learnt using Dale's backpropagation, and W be the weights learnt using standard backpropagation. Assume the non-linearity ϕ is either tanh or ReLU. Then, the error of the solution found using Dale's backpropagation with respect to the ground truth y is bounded by:*

$$\|f(W_D)-y\|_2^2 \leq \delta^2 \sum_{t=1}^{T}(L_{f_t})^2 + \sum_{t=1}^{T}(\varepsilon_t^*)^2$$

where $\delta = \frac{G}{L}\left((1+\eta L)^K-1\right)$ after K training iterations, $L_{f_t} = \max\left(L_{f_t(W)}, L_{f_t(W_D)}\right)$ is the Lipschitz constant of the RNN at timestep t, and $\varepsilon_t^* = \|f_t(W)-y_t\|_2$ is the error of the solution found using conventional backpropagation at timestep t.

Proof. We begin by considering a single-layer RNN unrolled over T timesteps. Let $f(W)$ be the function represented by this network, where W are the weights. We can express $f(W)$ as a composition of functions for each timestep:

$$f(W,x) = (f_T \circ f_{T-1} \circ \cdots \circ f_2 \circ f_1)(x_t) \tag{11}$$

where each $f_t(W_t, x_t) = \phi(W_{hh}h_{t-1} + W_{hi}x_t)$ represents the function at timestep t.

Now, let's consider the Lipschitz constants of these functions. When the non-linearity ϕ is either tanh or ReLU (both of which are globally Lipschitz with $L_\phi = 1$), it holds by Lemmas 8 and 9 that for every individual layer f_t, we have:

$$L_{f_t(W)} \leq L_\phi \cdot L_{W_t} \tag{12}$$

By recursively substituting Eq. 12 in Eq. 11, we notice the following pattern

$$\begin{aligned}h_1 &= \phi\left(W_{hh}h_0 + W_{hi}x_1\right) \\ L_{f_1(W)} &\leq \|W_{hh}\|_2 + \|W_{hi}\|_2 \\ h_2 = \phi\left(W_{hh}h_1 + W_{hi}x_2\right) &= \phi\left(W_{hh}\phi\left(W_{hh}h_0 + W_{hi}x_1\right) + W_{hi}x_2\right) \\ L_{f_2(W)} &\leq \|W_{hh}\|_2^2 + \|W_{hh}\|_2\|W_{hi}\|_2 + \|W_{hi}\|_2 \\ &= \|W_{hh}\|_2^2 + \|W_{hi}\|_2\left(1+\|W_{hh}\|_2\right) \\ h_3 = \phi\left(W_{hh}h_2 + W_{hi}x_3\right) &= \phi\left(W_{hh}\phi\left(W_{hh}\phi\left(W_{hh}h_0 + W_{hi}x_1\right) + W_{hi}x_2\right) + W_{hi}x_3\right) \\ L_{f_3(W)} &\leq \|W_{hh}\|_2^3 + \|W_{hh}\|_2^2\|W_{hi}\|_2 + \|W_{hh}\|_2\|W_{hi}\|_2 + \|W_{hi}\|_2 \\ &= \|W_{hh}\|_2^3 + \|W_{hi}\|_2\left(1+\|W_{hh}\|_2 + \|W_{hh}\|_2^2\right)\end{aligned}$$

Generalizing the pattern, the Lipschitz constant $L_{f_t(W)}$ of the RNN at T timesteps is bounded by

$$L_{f_T(W)} \leq \|W_{hh}\|_2^T + \|W_{hi}\|_2 \left(\sum_{t=0}^{T-1} \|W_{hh}\|_2^t \right)$$

Summing the geometric series we get

$$L_{f_T(W)} \leq \begin{cases} \|W_{hh}\|_2^T + \|W_{hi}\|_2 \cdot \left(\frac{1-\|W_{hh}\|_2^T}{1-\|W_{hh}\|_2} \right) & \text{if } \|W_{hh}\|_2 \neq 1 \\ 1 + T \cdot \|W_{hi}\|_2 & \text{if } \|W_{hh}\|_2 = 1 \end{cases} \tag{13}$$

The Lipschitz constant $L_{f_t(W_D)}$ of the RNN with weights W_D can be bounded similarly.

Now, let's consider the difference between the outputs of the RNNs with weights W and W_D at any given timestep t. By Lipschitzness,

$$\begin{aligned} \|f_t(W) - f_t(W_D)\|_2 &\leq \max(L_{f_t(W)}, L_{f_t(W_D)}) \|W - W_D\|_2 \\ &= L_{f_t} \|W - W_D\|_2 \end{aligned} \tag{14}$$

where $L_{f_t} = \max\left(L_{f_t(W)}, L_{f_t(W_D)}\right)$.

Applying Lemma 3 after K training iterations, we get:

$$\|W - W_D\|_2 \leq \frac{G}{L}\left((1+\eta L)^K - 1\right) = \delta \tag{15}$$

Therefore, we can simplify our bound on the difference between the outputs:

$$\|f_t(W) - f_t(W_D)\|_2 \leq L_{f_t} \cdot \delta \tag{16}$$

Let y_t be the ground truth at timestep t. Applying the triangle inequality, we get:

$$\begin{aligned} \|f_t(W_D) - y_t\|_2 &= \|f_t(W_D) - f_t(W) + f_t(W) - y_t\|_2 \\ &\leq \|f_t(W_D) - f_t(W)\|_2 + \|f_t(W) - y_t\|_2 \end{aligned}$$

Let $\varepsilon_t^* = \|f_t(W) - y_t\|_2$ be the error of the solution found using conventional backpropagation at timestep t. Hence,

$$\|f_t(W_D) - y_t\|_2 \leq L_{f_t} \cdot \delta + \varepsilon_t^* \tag{17}$$

To get the overall error, we sum over all timesteps:

$$\begin{aligned} \|f(W_D) - y\|_2^2 &= \sum_{t=1}^{T} \|f_t(W_D) - y_t\|_2^2 \\ &\leq \sum_{t=1}^{T} (L_{f_t} \cdot \delta + \varepsilon_t^*)^2 \\ &= \sum_{t=1}^{T} (L_{f_t}^2 \cdot \delta^2 + 2L_{f_t} \cdot \delta \cdot \varepsilon_t^* + (\varepsilon_t^*)^2) \\ &= \delta^2 \sum_{t=1}^{T} L_{f_t}^2 + 2\delta \sum_{t=1}^{T} L_{f_t} \cdot \varepsilon_t^* + \sum_{t=1}^{T} (\varepsilon_t^*)^2 \\ &\leq 2 \cdot \left(\delta^2 \sum_{t=1}^{T} (L_{f_t})^2 + \sum_{t=1}^{T} (\varepsilon_t^*)^2 \right) \end{aligned}$$

where the last inequality follows from the fact that $2\delta \sum_{t=1}^{T} L_{f_t} \cdot \varepsilon_t^* \leq \delta^2 \sum_{t=1}^{T} (L_{f_t})^2 + \sum_{t=1}^{T} (\varepsilon_t^*)^2$ over $\mathbb{R}$. □

Lemma 8 (Lipschitz constant of matrix multiplication, [99, 100])**.** *For a linear transformation $f(x) = Wx$, the Lipschitz constant L_f is equal to the operator norm of W, i.e., $L_f = ||W||_{op}$.*

Lemma 9 (Lipschitzness of composable Lipschitz functions, [99, 100])**.** *Let g and h be two composable Lipschitz functions with constants L_g, L_h respectively. Then $g \circ h$ is also Lipschitz with the constant $L_{(g \circ h)} \leq L_g \cdot L_h$.*

9 Setting κ and sparsity values for pruning

9.1 Derivation of κ

According to our pruning rule, the probability that a particular edge w_{ji} is retained in the pruned set is $\kappa|w_{ji}|$. Noting the fact that all edges in the matrix $W \in \mathbb{R}^{m\times n}$ are sampled independently, the expected number of edges in the pruned matrix W^{sparse} is simply the sum of probabilities that each individual edge of W is retained, i.e.,

$$\mathbb{E}\left[||W^{sparse}||_0\right] = \sum_{i=1}^{m}\sum_{j=1}^{n} \kappa|w_{ji}|$$

Assuming we wish W^{sparse} to have a sparsity of s, the number of edges in the pruned matrix needs to be $(1-s)mn$, thus giving us the equality

$$(1-s)mn = \sum_{i=1}^{m}\sum_{j=1}^{n} \kappa|w_{ji}| = \kappa\sum_{i=1}^{m}\sum_{j=1}^{n} |w_{ji}| = \kappa||W||_{L^1} \implies \kappa = \frac{(1-s)mn}{||W||_{L^1}}$$

When the matrix W represents a recurrent circuit of N neurons, $m = n = N \implies \kappa = \frac{(1-s)N^2}{||W||_{L^1}}$.

9.2 Re-normalization of sampling probabilities

Since the initial probability estimates $\kappa|w_{ij}|$ may result in values greater than 1, applying them directly could lead to an overestimation for the probability of retaining certain elements, potentially causing the final sparsity level to deviate from the target. We therefore take a renormalization step to adjust the probabilities so that they sum appropriately, enabling the pruning rule to meet the desired sparsity while maintaining consistency with probabilistic interpretation.

Algorithm 3: Probability Re-normalization

Input: arr: Array of all probabilities computed as $\kappa|w_{ij}|$
Output: arr: Adjusted array with re-normalized probabilities

$r \leftarrow \sum_{i=1}^{n} arr[i] - 1$ if $arr[i] > 1$ else 0 ; `// Calculate initial total residue`
while $r > 0$ **do**
  $counts \leftarrow \sum_{i=1}^{n} 1$ if $arr[i] < 1$ else 0 ; `// Count number of probabilities less than 1`
  $\delta \leftarrow r/counts$; `// Estimate delta to be added per probability < 1`
  for $i \leftarrow 1$ **to** n **do**
    $arr[i] \leftarrow arr[i] + \delta$ if $arr[i] < 1$ else 1 ; `// Update array with delta added`
  end
  $r \leftarrow \sum_{i=1}^{n} arr[i] - 1$ if $arr[i] > 1$ else 0 ; `// Calculate new total residue`
end
return arr ; `// Return re-normalized probability array`

9.3 Adjusting sparsity values

When targeting a specific sparsity level s for a matrix (or block of weights), we may find that the matrix W already contains a certain number of zero entries, denoted by z_0. If these existing zeros are not accounted for, applying the desired sparsity s directly may result in a final sparsity that exceeds the intended level. Therefore, we adjust s to a new value s', which takes into account the current sparsity of W as follows:

$$s' = \frac{s \cdot N_{total} - z_0}{N_{total} - z_0}$$

10 Expected overlaps with MST across sampling methods

10.1 Lower bounding expected overlap: Top-prob pruned network vs dense MST

Here we establish a (loose) lower bound on the expected overlap between the weights kept when sparsifying an RNN using the top-prob pruning rule and the maximum spanning tree (MST) of the original network, which from the view point of persistent homology, encapsulates all the zeroth-order topological information of a (trained) network.

Consider the square connectivity matrix with N pre-synaptic and post-synaptic neurons each. For our purposes we will assume that every neuron always acts as both, a source and a target to at least one other (but not necessarily the same) neuron. The total number of weights in the connectivity matrix is then N^2 of which $(1-s)N^2$ will be sampled for the pruned network to have a target sparsity of s. The MST of such a weight matrix will have exactly $2N-1$ weights.

Following Kruskal's algorithm, the probability that the k^{th} largest weight by magnitude is in the MST of the bipartite graph can be lower bounded as follows:

In a bipartite graph, the smallest cycle must have at least 4 edges. Therefore, the largest three weights ($k \leq 3$) are always in the MST.

For $k > 3$ we can lower bound the probability that the k^{th} largest edge is in the MST. In particular we note that for a weight to lie in the MST, it must not form a cycle, meaning that it either joins two nodes that were both previously not connected to any other nodes in the graph, or joins at most one new node to another connected component on the graph. For the purposes of establishing a lower bound, we only look at the probability of the former.

Consequently, for $3 < k < N$

$$\mathbb{P}[k^{th} \text{ largest weight connects two isolated vertices}] \geq \frac{(N-(k-1))^2}{N^2-(k-1)}$$

Equivalently,

$$\mathbb{P}[k^{th} \text{ largest weight is in MST}] \geq \frac{(N-(k-1))^2}{N^2-(k-1)}$$

Combining the previous two statements we get the bound:

$$\mathbb{P}[k^{th} \text{ largest edge is in MST}] \geq \begin{cases} 1 & \text{for } k \leq 3 \\ \frac{(N-(k-1))^2}{N^2-(k-1)} & \text{for } 3 < k < N \\ 0 & \text{for } k \geq N \end{cases}$$

The expected number of weights that overlap with the MST is then simply bounded as

$$\begin{aligned} \mathbb{E}[N_{overlap}]_{top-prob} &\geq \sum_{k=1}^{N^2} \kappa|w_k| \cdot \mathbb{P}[k^{th} \text{ largest weight is in MST}] \\ &= \sum_{k=1}^{3} \kappa|w_k| + \sum_{k=4}^{N} \kappa|w_k| \cdot \left(\frac{(N-(k-1))^2}{N^2-(k-1)} \right) \end{aligned}$$

where $\kappa = \frac{(1-s)N^2}{||W||_{L^1}}$

10.2 Expected overlap with MST for random pruning

In the case of random sampling, quantifying the expected number of weights which overlap between the MST and sampled weights is equivalent to that between two arbitrarily chosen subsets, one with $2N-1$ weights (i.e., the same size as the MST) and the other with $(1-s)N^2$ weights (i.e., the same size as the sparsified connectivity matrix).

To do so we can directly use the expression for the probability mass function of a hypergeometric distribution, where the probability of a random variable X having k successes (random draws for which the object drawn has a specified feature) in n independent draws (without replacement) from a finite population of size M objects that contains exactly K objects with that feature is given as

$$p_X(k) = \mathbb{P}(X=k) = \frac{\binom{K}{k} \cdot \binom{M-K}{n-k}}{\binom{M}{n}}$$

Noting that the MST is a fixed set of $2N-1$ weights for any instantiation of the connectivity matrix, the probability that it has exactly k weights overlapping with the randomly sampled set of size $(1-s)N^2$ out of a possible set of N^2 weights is

$$\mathbb{P}\left[N_{overlap}\right]_{random} = \frac{\binom{2N-1}{k} \cdot \binom{N^2-2N+1}{(1-s)N^2-k}}{\binom{N^2}{(1-s)N^2}}$$

The expected number of sampled weights that overlap with the MST then is simply

$$\begin{aligned}\mathbb{E}\left[N_{overlap}\right]_{random} &= \sum_{k=1}^{n} k \cdot \mathbb{P}\left[N_{overlap}\right]_{random} \\ &= \sum_{k=1}^{n} k \cdot \frac{\binom{2N-1}{k} \cdot \binom{N^2-2N+1}{(1-s)N^2-k}}{\binom{N^2}{(1-s)N^2}}\end{aligned}$$

where $n = \min(2N-1, (1-s)N^2)$.

11 Data pre-processing: Imaging depths

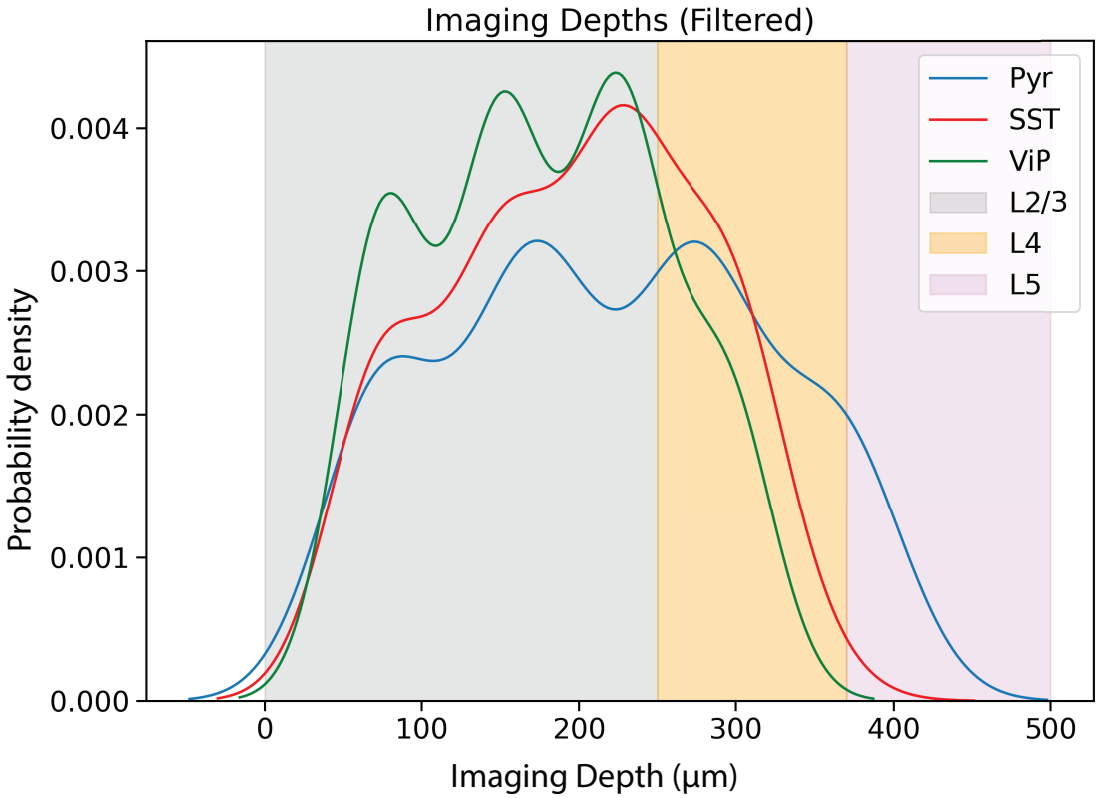

Figure 6: *Imaging depths.* Distribution of experiments (Y-axis) with respect to imaging depth (X-axis). Shaded grey, yellow and pink panels represent depths corresponding to cortical layers 2/3, 4, and 5 respectively.

12 Connection probabilities within and across cell populations

Connection Probabilities	L2/3 Pyr	L2/3 SST	L2/3 VIP	L4 Pyr	L4 SST	L4 VIP	L5 Pyr	L5 SST	L5 VIP
L2/3 Pyr	0.06	0.23	0.05	0.07	0.33	0.00	0.00	0.15	0.00
L2/3 SST	0.3	0.05	0.14	0.04	0.05	0.25	0.00	0.05	0.00
L2/3 VIP	0.16	0.30	0.01	0.02	0.21	0.00	0.00	0.13	0.00
L4 Pyr	0.05	0.04	0.00	0.10	0.22	0.00	0.00	0.19	0.00
L4 SST	0.17	0.00	0.14	0.04	0.01	0.14	0.08	0.04	0.2
L4 VIP	0.00	0.13	0.05	0.00	0.22	0.03	0.02	0.14	0.00
L5 Pyr	0.08	0.00	0.00	0.00	0.08	0.00	0.04	0.15	0.00
L5 SST	0.04	0.00	0.00	0.03	0.04	0.11	0.10	0.03	0.05
L5 VIP	0.00	0.00	0.04	0.11	0.07	0.04	0.02	0.1	0.03

Table 1: Connection probabilities amongst different cell types and populations as per Campagnola et al. [85]. **Columns** = Pre-synaptic population **(source)**, **Rows** = Post-synaptic population **(target)**.

13 Connectivity differences across varying degrees of spatial resolution and types of error

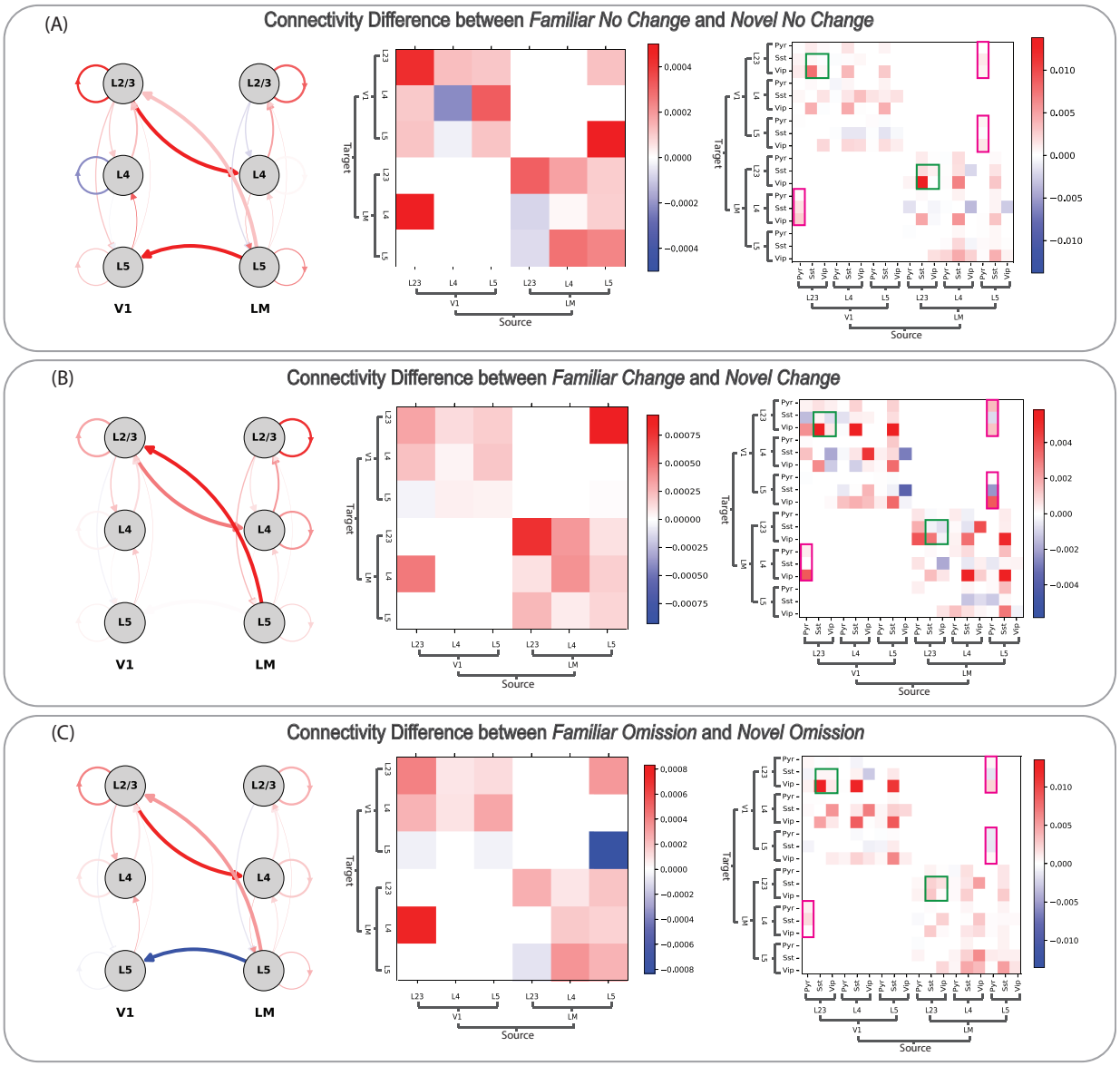

Figure 7: *Connectivity differences between familiar and novel conditions.* **(A)** Familiar no change vs. Novel no change. **(B)** Familiar change vs. Novel change. **(C)** Familiar omission vs. Novel omission. All plots are from the full presentations condition. All differences are computed as `Second condition - First condition`; Blue implies higher weights in the first condition, while red indicates higher weights in the second. In all cases, the left and middle plots are a graphical representation of the weights averaged across layers, while the rightmost plot averages weights by cell-type within each layer. Magenta boxes highlight the feedforward and feedback connections, i.e., those originating at V1 L2/3 and LM L5 respectively. Green boxes highlight Sst-ViP interactions in L2/3 of V1 and LM.

In all three cases we see that the presentation of a novel image (which is a type of expectation violation) increases the inter-areal feedforward connectivity V1 L2/3 $\rightarrow$ LM L4 in the microcircuit. We see an increase in the feedback connectivity V1 $\leftarrow$ LM as well, but the specific feedback pathway that is engaged changes with the type of error; In particular, compared to the Familiar No Change condition, during the Novel No Change condition, an increase in V1 L5 $\leftarrow$ LM 5 dominates, while in the case of Familiar Change vs. Novel Change, V1 L2/3 $\leftarrow$ LM 5 is the dominant form of increased inter-areal feedback. Across both sets of comparisons, we notice that the novel case leads to increased activity in V1 Vip neurons in both layers 2/3 and 5. In the case of Familiar Omission vs. Novel Omission, we note that

there is an increase in the feedback V1 L2/3 ← LM L5 during the Novel Omission condition, as with the other two conditions. However in this case the projection V1 L5 ← LM L5 is lower during the Novel Omission condition, once again speaking to the specificity of feedback projections depending on the type of novelty. As before, we notice that the novel case leads to increased activity in V1 Vip neurons in both layers 2/3 and 5.

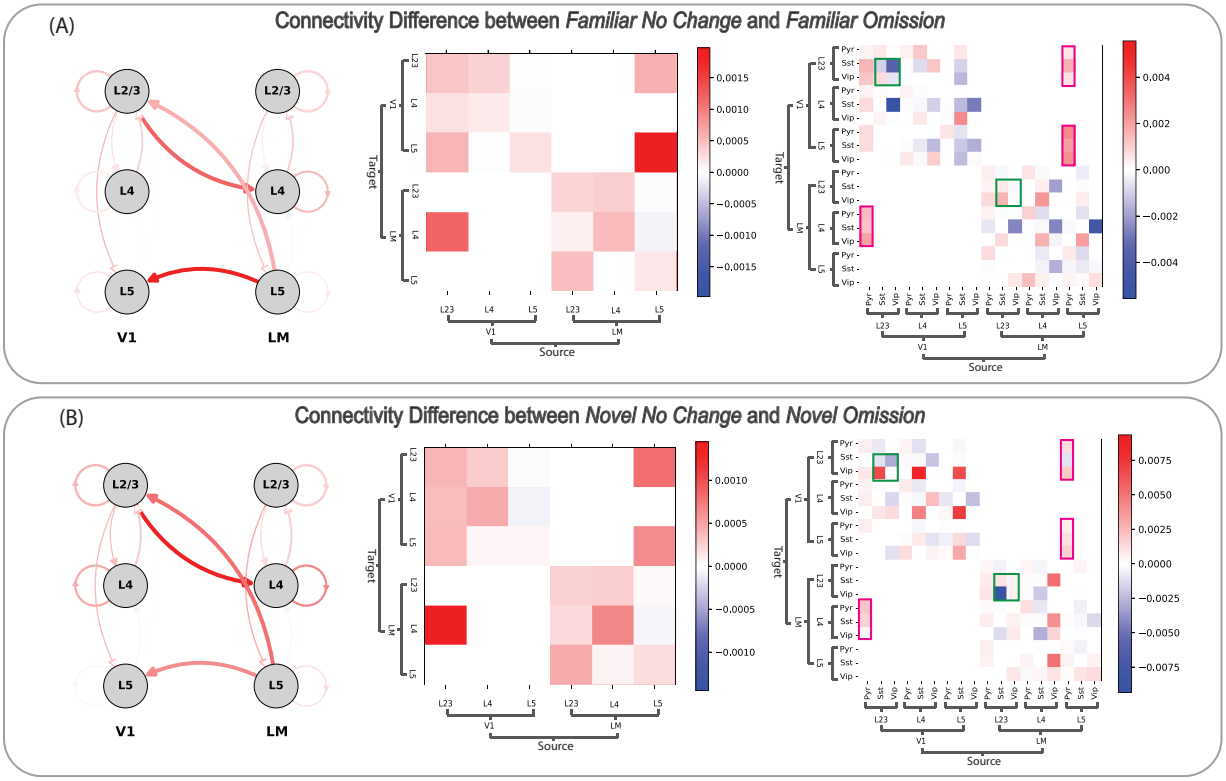

Figure 8: *Connectivity differences between no change and omission conditions.* **(A)** Familiar No Change vs. Familiar Omission. **(B)** Novel No Change vs. Novel Omission. Both plots are from the full presentations condition. All plots are from the full presentations condition. Differences are computed as `Second condition - First condition`; Blue implies higher weights in the first condition, while red indicates higher weights in the second. In both cases, the left and middle plots are a graphical representation of the weights averaged across layers, while the rightmost plot averages weights by cell-type within each layer. Magenta boxes highlight the feedforward and feedback connections, i.e., those originating at V1 L2/3 and LM L5 respectively. Green boxes highlight Sst-ViP interactions in L2/3 of V1 and LM.

In both cases we see that the omission condition increases the inter-areal feedforward connectivity V1 L2/3 → LM L4 as well as both forms of inter-areal V1 ← LM feedback in the microcircuit. Broadly, both no-change vs. omission conditions seem to induce similar connectivty across the various cell-type populations and layers, indicating that the omission, i.e., type of violation of the stimulus, strongly affects the connectivity and subsequent in the microcircuit.

However, at the cell-type level, we see that the strength of interaction from Sst to Vip in L2/3 increases under conditions of novelty and omission, which initially seems at odds with the activity-based findings in [56, 79] that state Vip response is increased during these conditions and Sst activity is reduced. One explanation is that the inferred connectivity is correlative rather than causative, and given that this circuit motif is itself embedded in a larger network with other cell-type populations whose activities we do not have access to in this set of experiments (e.g., PV cells), their absence might skew our results at this finer level when studying inferred connectivity.