Beyond Gradients: Noise Correlations Control Hebbian Plasticity to Shape Credit Assignment

Daniel N. Scott[1,3,*] and Michael J. Frank[2,3]

[1]Department of Neuroscience, Brown University
[2]Department of Cognitive, Linguistic, and Psychological Sciences, Brown University
[3]Carney Institute for Brain Science, Brown University
[*]Corresponding Author

Updated: March 22, 2023

Abstract

Learning involves synaptic plasticity, but it remains unclear how the brain determines which synapses should change and when. Plasticity is thus often considered as an approximation to gradient descent, despite drawbacks to the latter, interpretational difficulties, and seemingly contrary evidence. To address these issues, we introduce the "coordinated eligibility model", linking biology with function and generalizing neuroscientific theories. We show that gradients can be decomposed into factors related to correlated firing rate variability and dendritic inhibition, which control population response changes and receptive field re-weighting respectively. By determining plasticity according to products of these factors, our model computes directional derivatives of loss functions. These derivatives need not align with task gradients, allowing networks to overcome limitations of gradient descent like catastrophic interference, and they can facilitate important functions like compositional generalization. As such, our model introduces a powerful and widely-applicable framework for interpreting supervised, reinforcement-based, and unsupervised plasticity in nervous systems.

Introduction

When animals learn new skills they rarely get worse at previously learned ones, and new knowledge does not generally displace old knowledge. Instead, old learning is often brought to bear on new situations, refining it and speeding new learning as a result (see e.g. Dekker *et al.* 2022; Franklin & Frank 2018; Ghirlanda & Enquist 2003; Tenenbaum & Griffiths 2001; Shepard 1987). A theoretical and mechanistic understanding of how these two feats - the retention of old learning and its generalized use in new scenarios - can be jointly accomplished has been elusive however. Most research has treated each topic separately in fact, despite the fact that both characterize learning interactions between episodes; in the former situation, some set of specific counter-productive learning interactions should be avoided, and in the latter, productive interactions should be sought. These observations suggest seeking a unified account of learning interactions, which we refer to generically as "interference" below.

Most previous work on memory retention during new learning has taken place in computational cognitive neuroscience or computer science, under the topic of catastrophic forgetting. When neural networks are trained to solve one task at a time, they often retain only the most recently learned abilities or information, forgetting old information (McCloskey & Cohen 1989; Ratcliff 1990; Flesch, Balaguer, *et al.* 2018). Within machine learning, common approaches to mitigate this problem include interleaving training of different tasks or regularizing weight updates (Srivastava *et al.* 2014; Kirkpatrick *et al.* 2017), whereas specialized architectures are often considered in computational neuroscience instead (e.g., Schapiro *et al.* 2017; O'Reilly & Norman 2002; Mcclelland *et al.* 1995). For example, the hippocampus appears to promote memory retention (i.e., protect against over-writing) by ensuring that only a small fraction of neurons participate in learning at any given time (Mcclelland *et al.* 1995). Likewise, neuronal populations in the prefrontal cortex are regulated by "gating" processes subject to reinforcement learning, which likely avoid interference by determining which populations learn about what information (Frank & Badre 2012). Some work also exists near the intersection of these, showing that initial network connectivity can shape gradients to reduce catastrophic forgetting, for example (Flesch, Juechems, *et al.* 2021).

Work on generalization has also proceeded according to several methodological routes. The most common approach in computer science is simply exposing networks to diverse learning scenarios, which can allow them to extract latent regularities and thereby facilitate generalization. The dense connectivity of the cortex has long been

thought to play a related role, by representing diverse forms of information in highly overlapping cell ensembles (Mcclelland *et al.* 1995). In theory, such overlap may average over learning episodes to extract shared information. A second important approach in computer science is the use of restricted capacity learning elements, as in auto-encoders, which attempt to compress information and thereby force gradient descent to extract latent features of importance (Goodfellow *et al.* 2016). In neuroscience, related work has considered how reinforcement learning of "gating" policies may control information flow to prefrontal cortex and thereby produce selective information processing (Frank & Badre 2012). Such selectivity may then facilitate compositional generalization, and the transfer of old learning to new situations (Rougier *et al.* 2005; Collins & Frank 2013).

While worthwhile and informative, the approaches noted above investigate architectural constraints, regularizing procedures, and other auxiliary factors related to forgetting and generalization. By contrast, we consider here how the functional form of synaptic plasticity impacts interference. From a theoretical perspective, such a principled account is more desirable than a set of ad-hoc methods for managing it. Moreover, biological synaptic plasticity has been extensively characterized (Rudy 2015; Scott & Frank 2022; Roelfsema & Holtmaat 2018), and is both necessary and sufficient for many forms of learning (see, e.g., Lemke *et al.* 2021; Sehgal *et al.* 2021; Yang *et al.* 2014), suggesting that auxiliary methods for managing interference may represent only some aspects of biological function. And finally, while work in computational neuroscience has often assumed that synaptic weights evolve according to gradient descent (or some approximation thereof), exactly how known forms of plasticity are related to gradients remains an important open question (Scott & Frank 2022; Liu *et al.* 2021; Nayebi *et al.* 2020; Lillicrap *et al.* 2020; Bellec *et al.* 2020).

Elements of a theory are suggested by observations regarding coordinated plasticity. By the latter, we mean statistical dependence between non-proximal synapse changes, either within or across neurons. Such plasticity is widespread and functionally important (Cummings, Lacagnina, *et al.* 2021; Cichon & Gan 2015; Yang *et al.* 2014; Gambino *et al.* 2014; Antic *et al.* 2010), and as we show here, mathematically more general than typical formulations of gradient descent. Moreover, groups of neurons in cell ensembles frequently show coordinated plasticity (Cummings, Bayshtok, *et al.* 2022; Cummings, Lacagnina, *et al.* 2021), and dendritic branch specific plasticity often yokes learning across multiple synaptic inputs (L. Chen *et al.* 2020; Cichon & Gan 2015). Both have been shown to impact interference between learning episodes, in fact, (Cichon & Gan 2015; Yang *et al.* 2014). This lead us to consider how such mechanisms may relate to gradient descent, and how they may be impact interference.

To answer these questions, we analyzed neuromodulated Hebbian models of synaptic plasticity and their relationships to task gradients. We develop a theory we call the coordinated eligibility model (CEM), which generalizes and extends the closely related Bienenstock-Cooper-Munroe theory (Bienenstock *et al.* 1982; Cooper & Bear 2012), two-threshold calcium theories (Evans & Blackwell 2015), and models from reinforcement learning (Williams 1992; Fiete & Seung 2006; Izhikevich 2007; Farries & Fairhall 2007; Frémaux *et al.* 2013). The CEM links neural firing-rate variation, intracellular calcium dynamics, and neuromodulation to control both task improvement and between-task learning interactions. As we show, the synaptic weight changes which most directly improve performance on a task (i.e., gradients) can be written in terms of two components. We refer to the first as a population response (PR) change, because it represents a change in how neurons in a network layer respond to a given stimulus. We refer to the second as a receptive field (RF) change, because it reflects a change in what drives each neuron. If two neurons respond proportionally to a stimulus, for example, but one increases its firing rate relative to the other, we would refer to this as a population response change. If instead each begins responding to different input features, we would refer to these as receptive field changes for each neuron. Both PR and RF changes will often be referred to as "filter changes" below for brevity, by reference to linear filter theory.

Our results stem from the fact that any form of plasticity improving task performance involves filter changes with specific relations to those of task gradients. In general, there are many synaptic weight changes that are compatible with immediate task improvement. By modulating trial-wise neural (co)variability (frequently referred to as "noise correlation" in neuroscience), CEM provides additional constraints on PR and RF filters that primarily control other properties of learning, such as retention and generalization. Across multiple task settings commonly studied in reinforcement learning and continual learning, we show that CEM affords the opportunity for more robust performance compared to generic gradient descent. Moreover, we provide a formal account of how filter relations control these properties, and of how noise correlations play a critical role in trial-and-error gradient estimation.

Re-stated mathematically, our results follow from several observations. First, gradients of task performance with respect to layers of network weights have tensor-product decompositions into vector and dual vector components, representing population response and receptive field changes respectively. Second, task-learning interactions (i.e., interference) arise from inner product relationships between gradients, and these inner products therefore decompose into products between pairs of vectors and pairs of dual vectors. Since tensor products of projections of gradient PR and RF changes are themselves projections of gradients, the CEM computes directional derivatives. And because

directional derivatives can be interpreted as gradients on constraint surfaces, we note that they can impose inner product relations between pairs of PR and RF changes to control interference. Existing models of plasticity are compatible with these observations, and our coordinated eligibility model formalizes the relevant relationships.

Results

In the following sections, we first introduce the coordinated eligibility model, then show how it controls population responses and receptive fields. Next, we show why these are important for interference mathematically. These ideas are illustrated in figure 1, with PR and RF changes either removing or generating interference. We therefore present a sequence of simulations addressing these cases. The first two simulations show how destructive (catastrophic) interference can be avoided via adjustments to RFs and PRs, respectively. We then address the joint use of PR and RF eligibility to reduce interference involved in learning about compositional representations across layers. Two further simulations demonstrate how coarse-coded RF and PR changes facilitate generalization. Throughout all our simulations, we use a reinforcement learning paradigm called the contextual bandit problem, which we discuss below, and we compare our results with gradient descent. In our final sections, we show that the coordinated eligibility model is a generalization of the BCM, two-threshold calcium, and REINFORCE models.

The Coordinated Eligibility Model

Our coordinated eligibility model posits that changes in receptive fields and population responses are jointly learned by trial and error in the form of variable neural activity and concomitant plasticity. Coordinated firing rate variability is hypothesized to explore population representation changes, while coordinated control of dendritic calcium dynamics samples receptive field modifications. The degree to which these are converted into plasticity is assumed variable, and dependent on covariation with so-called third factors such as dopaminergic reward-prediction errors (reviewed in, e.g., Scott & Frank 2022). Importantly, trial, error, and reward in the model are formal notions that need not correspond with behavioral trial, error, and reward. Instead, they describe a variation-selection process controlling computational adaptation. Nonetheless, each can be interpreted literally in areas such as striatum, and so we address this case in our exposition.

Mathematically, we model each post-synaptic neuron's firing rate as the product of a variable gain term γ and a normalized current-frequency curve (i.e., point-wise non-linearity) f applied to summed synaptic input. In our simulations we take f to be rectification above some threshold θ, which we set at zero for simplicity. Plasticity in each neuron's set of input synapses w is then modelled as a function of the post-synaptic variables along with a neuromodulator concentration m and a vector of dendritic calcium variables d:

$$\Delta w = m\gamma\Delta f d^T \tag{1}$$

The vector d is itself some function of the synaptic inputs x, and the quantity Δf is a firing rate deviation $f - \bar{f}$, where $\bar{f}$ is a baseline. There may be biological or normative reasons to define this baseline differently depending on context, but we take it here to be the gain-normalized average firing rate over presentations of a given stimulus, denoted $\langle f \rangle$. Hence the baseline is stimulus-dependent, reflecting differentiated, relatively consistent, spatially specific calcium current inductions by distinct stimuli. This stimulus-dependence is consistent with other models such as REINFORCE, CHL, and XCAL, which also have baselines induced by stimulus evoked activity, and such baselines are used in conjunction with trial-wise activity deviations and performance signals to improve function.

Biologically, the model is motivated by two observations. First, firing rate gains are frequently controlled quasi-independently of glutamatergic inputs (Cohen-Kashi Malina *et al.* 2021; Garcia-Junco-Clemente *et al.* 2019). And second, dendritic calcium deviations are generally impacted by many factors in addition to input stimulus identity (Scott & Frank 2022). While most models take weighted sums of inputs to both determine firing rates and to define the Hebbian term governing weight changes, our coordinated eligibility model separates these. This separation allows receptive field plasticity to be a more general function of input, and coordinated firing-rate gain modulation to control the population response aspects of plasticity. As such, the CEM directly and independently controls updates to the input weighting and output properties of a group of neurons. These features allow the model to flexibly explore different potential learning outcomes, and they provide links between biology and mathematical aspects of learning which we examine in the following sections. Notably, the model reduces to various other algorithms as special cases, as we explain in our final results section and figure 7.

Trial-wise covariation in firing rates samples population responses

The coordinated eligibility model can be written in terms of vectors and matrices to summarize all synaptic changes between two layers of neurons simultaneously. This is simplest to illustrate when $d(x)$ is the same function for every neuron. Using subscripts to index neurons, the matrix of synapse changes is:

$$\Delta w = m\gamma\Delta f d^T = m \begin{bmatrix} \gamma_1\Delta f_1 \\ \gamma_2\Delta f_2 \\ ... \\ \gamma_n\Delta f_3 \end{bmatrix} \begin{bmatrix} d_1, d_2, ..., d_n \end{bmatrix} = m \begin{bmatrix} \gamma_1\Delta f_1 d_1 & \gamma_1\Delta f_1 d_2 & ... & \gamma_1\Delta f_1 d_n \\ \gamma_2\Delta f_2 d_1 & \gamma_2\Delta f_2 d_2 & ... & \gamma_2\Delta f_2 d_n \\ ... & ... & ... & ... \\ \gamma_n\Delta f_n d_1 & \gamma_n\Delta f_n d_2 & ... & \gamma_n\Delta f_n d_n \end{bmatrix}$$

Above, w now denotes the whole set of synaptic efficacies, i.e., the weight matrix, whereas f is a vector of post-synaptic firing rates and d is a vector of calcium elevations for each potential pre-synaptic input. The term $\gamma\Delta f d^T$ is an outer (or tensor) product, and denotes a matrix with $\gamma_i\Delta f_i d_j$ in the j^{th} column of the i^{th} row. This product is composed from two vectors, the vector $\gamma\Delta f$ on the left, and d on the right. Since $\gamma\Delta f$ specifies a change in firing rates across a layer, and d specifies a change in responsiveness to various inputs for each neuron, these vectors represent population-representation changes and receptive field re-weightings, respectively. As one can verify, the weight change Δw_{ij} depends on the post-synaptic activity $\gamma_i f_i$ and the dendritic calcium d_j as in equation 1.

The trial-average weight change is equal to the gradient of reward with respect to network weights under particular conditions. These require that m represents reward prediction error, γ is constant, $d(x) = x$ for all neurons, the firing rate transfer function is the identity, and firing rate variability is mean-zero and isotropic. In this case the model is equivalent to REINFORCE Williams 1992. The mean-zero isotropic condition occurs when firing rate noise is normally distributed with no correlation, for example, meaning $\gamma\Delta f \sim \mathcal{N}(O, I)$. Using g to denote the ideal population-response change, the average weight update is then:

$$\langle\Delta w\rangle = gx^T \tag{2}$$

This says that every neuron should change its receptive field to weigh inputs by strength, reflected in the x^T term, and that each neuron should increase or decrease its firing rate such that the whole population response changes by g. A proof of this weight update relation can be found in our appendices, reproducing (Williams 1992).

Importantly, the average weight update is the result of summing a number of individual ones, as with stochastic gradient descent. The assumption that $\gamma\Delta f \sim \mathcal{N}(O, I)$ means that potential updates $\gamma\Delta f x^T$ perform exhaustive trial-and-error search for the best dimensions of change, being reinforced or discouraged by reward. Each individual neuron gradually increases or decreases its firing rate independently, according to RPE-transmitted feedback dependent on network outputs. By inspection, one can note that it would be more efficient if only one sample population change was tried, with all neurons varying together, such that the vector of changes was g. The sought-after average would then be computed in one step, and conflicting population response changes would not need to be added together, generally cancelling one another out to ultimately generate gx^T.

This line of reasoning has two important implications. First, the sample complexity of gradient estimation can be improved by considering smaller hypothesis spaces of population-response changes. Sampling along g alone cannot generally be enacted because is is not known beforehand, but the space of possibilities to consider may often be pruned using prior information or biases. Indeed, previous work showed that noise correlations reduce sample complexity during learning in a model of plasticity related to that considered here (Nassar *et al.* 2021). The CEM generalizes this insight via the additional observation that noise correlations specifically produce directional derivatives (see figure 1). Introducing covariance in general dimensions, rather than necessarily along g, produces population response updates that need not align with g as they sum. Importantly, such alignment is often unnecessary, even for completely and efficiently solving tasks, because neural networks typically have many more degrees of freedom than they do training data. As a result, the CEM can use population-response changes to flexibly satisfy multiple constraints, such as avoiding interference during task improvement. We illustrate a simple example of this in figure 1, panels G-H. A formal treatment of this result, including its proof, can be found in the supplement.

Dendritic shaping determines receptive fields

Taking expected values above resulted in an expectation over $m\gamma\Delta f$, which yielded the optimal population response change g when particular (i.e., REINFORCE) conditions were met, and directional derivatives otherwise. In a more general setting, d is variable and is included in the average. The simplest example differing from gradient descent takes d as identical across neurons but independent from m, γ, and Δf. Then for some v representing the population-response change we just replace x with d in the equations above:

$$\langle\Delta w\rangle = vd^T \tag{3}$$

For example, suppose that $d = [x_1, 0, ...0]$. Then this weight change $\langle \Delta w \rangle$ implies that the whole population will respond more with the representation v whenever x_1 is present (subject to additional impacts of the firing-rate non-linearity). More generally, if $d = x$, then the population will respond more like v whenever the inputs match elements of the vector x. More realistically, these receptive field plasticity selections almost certainly vary across neurons, in which case the population change v may be as above, but each neuron would modify its receptive field to attend to different aspects of the layer's input. The total weight change matrix is then a sum of terms similar to (3) with different d functions depending on the term in the sum. These cases are also directional derivatives, because they are also projections of the gradient gx^T. We expect these to be computed biologically by a sampling mechanism like the one discussed for population responses. Work showing, for example, that dendritic-calcium shaping SOM+ interneurons form covarying ensembles just as PV+, pyramidal, and medium-spiny neurons do, supports such a hypothesis (Cummings, Bayshtok, *et al.* 2022). The resulting RF description is mathematically similar to the PR description (shown alongside one biological interpretation in figure 1), so we refer interested readers to our supplementary material and proceed to address interference as a function of PR and RF changes.

Interference decomposes as representation and receptive field change

As noted above, many outcomes of learning, such as learning-induced forgetting or generalization, can be cast in terms of interference. Interference is a property of tasks and the learning algorithms used to solve them, and it refers to the situation whereby modifying performance on one task leads to altered performance on another (McCloskey & Cohen 1989). This applies to broad definitions of "task" and "learning algorithm" in the sense that if one quantifies an aspect of network activity and alters both this and another measure by changing the network, then these "tasks" interfere given this "learning algorithm". In this section we quantify how interference manifests in terms of RF and PR changes and their products.

Importantly, performance change only occurs when weights in a network move uphill or downhill on a task error surface, meaning non-orthogonally relative to a gradient. Interference is thus a result of gradient relationships. When reward depends on the activity in some layer of a network, its gradient with respect to network weights is always an outer product gx^T, where x represents input to the network layer and g is the ideal population-response change. Gradients for two tasks can then be seen to interfere according to relationships between their representation and receptive field changes.

Using $(\cdot, \cdot)$ to denote the usual matrix inner product, and examining gradients with respect to two reward functions r_1 and r_2, we find that:

$$(\frac{dr_1}{dw}, \frac{dr_2}{dw}) = (\Delta_1 w, \Delta_2 w) = (g_1^T g_2)(x_1^T x_2) \tag{4}$$

For networks with multiple layers, the same is true of each layer, so that the inner product of two gradients taken with respect to all weights simultaneously also decomposes into terms such as these. Each inner product is zero if either $x_1^T x_2 = 0$ or $g_1^T g_2 = 0$, meaning one gradient has no impact on the other task's performance if either the RF changes or the PR changes are orthogonal. In the extreme case of two sub-networks being either active or inactive conditional on each of two tasks, there is clearly no interference because changes in both RFs and PRs involve different neurons. The equation above generalizes this to the case where tasks use the same neurons. Because neural networks are typically very high dimensional, RF and PR changes generally have unconstrained degrees of freedom available to potentially manage such interference.

In the coordinated eligibility model, the population response covariance C_1, gradient PR change g_1, and selective receptive field plasticity d_1 associated with task 1 jointly impact performance on task 2 if:

$$(\Delta_1 w, \Delta_2 w) = (g_1^T C_1^T g_2)(d_1^T x_2) \neq 0 \tag{5}$$

That is, learning task 1 has no impact on task 2's performance if the candidate representation changes explored for task 1 do not "overlap" with those impacting performance on task 2, or if dendritic control of receptive field plasticity ignores elements of synaptic input shared between tasks. The result is slightly more subtle than this in the sense that interference will still be avoided if change associated with task 1 involves counterbalanced updates, rather than simply non-overlapping ones, with respect to task 2. For example, if task 2 involves summing activity from several elements, then increasing RF responses to one and decreasing them to another could leave their sum, and hence performance, unchanged.

Importantly, equation (5) is a recipe for producing generalization as well as avoiding destructive interference. Generalization is the opposite of destructive interference in the sense that rather than having new learning degrade previously achieved performance, either new learning improves it (which we would call retroactive constructive interference) or learning one task decreases the need for new learning during another, providing proactive constructive

interference. These outcomes also require RF and PR changes to have significant overlap with their counterparts in gradients, with the inner products being positive rather than negative. Networks can therefore seek interference via PR and RF changes in order to produce learning generalization or transfer.

Finally, it is important to note that the representation-modifying terms g contain information about downstream processing, such that overlap therein acts similarly to overlap in stimulus inputs to the network. Even if two tasks require driving different effectors, RF overlap for the neurons controlling them can generate interference just as overlapping input stimuli do. There is a fundamental symmetry between input overlap and readout overlap, which generate the two sources of interference in equations (4) and (5). This symmetry can be broken when reward depends on a set of effectors shared across tasks, however, because the reward or error computation applied to all network outputs acts analogously to a final, single-neuron processing stage. As a result, the common machine-learning practice of using "multiple heads", which is analogous to context-dependent processing in real neural networks, is important for realizing the full potential of any particular network architecture to determine interference.

To summarize, the preceding subsections have introduced the coordinated eligibility model, which explicitly controls population response and receptive field plasticity via neural firing rate covariance and a generic model of dendritic calcium shaping. Candidate population response changes form a hypothesis space of potential computational changes, given any particular input, and the model selectively samples this space according to firing rate covariance. A similar logic applies to candidate receptive field changes, although we have left the details unspecified, because the interaction between the two would increase the scope of this paper substantially. These PR and RF changes control interference, because task gradients decompose into PR and RF terms, implying that interference calculations do as well. We now proceed to demonstrate these results and some of their implications in neural network simulations.

Contextual bandit tasks and network model

Perhaps the most well established example of neuromodulated Hebbian plasticity, of which the coordinated eligibility model is an example, is corticostriatal reinforcement learning. In this context, dopaminergic signaling of RPEs modulates synaptic plasticity in striatal medium spiny neurons (MSNs) to improve behavior (Schultz *et al.* 1997; Frank 2005; Shen *et al.* 2008; Scott & Frank 2022). We therefore considered how coordinating eligibility might facilitate adaptive striatal learning. The striatum contains several interneuron types, including parvalbumin-positive, somatostatin-positive, and cholinergic subtypes which modulate spiny neuron excitability and learning (see, e.g., Monteiro *et al.* 2018; Cattaneo *et al.* 2019; Reynolds *et al.* 2022). In addition, MSNs express voltage-gated calcium channels, exhibit depolarized dendritic "up-states", have their firing rates modulated by both dopaminergic signaling and CREB activity, and develop recurrent collaterals targeting both somatic and dendritic regions of other MSNs. As a result, there are numerous mechanisms that could generate the basic functional subdivisions of gain, firing rate, and dendritic calcium control required by our coordinated eligibility model. One such candidate division of labor is illustrated in figure 1, panel C.

We therefore simulated simplified three-layer neural networks representing cortico-basal-ganglia interactions to explore how plasticity rules (which are downstream of the biology discussed above) relate to interference, abstracting away from the complex disinhibitory architecture of the BG in doing so. These used coordinated eligibility to solve reinforcement learning tasks, and we compared our results with policy gradient based learning. Each of our simulations used 200 input (cortical) neurons, 200 hidden (striatal) neurons, and one output neuron for each action, which competed for selection via a softmax function. Neurons were modelled as rectified-linear units, input-to-hidden-layer weights were initialized according to log-normal distributions. and all weights were constrained to be positive. The tasks we modelled were "contextual bandit" problems as described below, which networks learned over repeated blocks of 20 training trials each. Our demonstrations used algorithmic gradients (rather than sampled ones) to control for sample complexity, and they therefore illustrate improvement based on filter geometry alone. Nevertheless, we verify that online schemes estimating projections during learning produce very similar results (see the supplement on oracle vs sample-based quantities).

The contextual bandit tasks describe the common process of learning to take different particular actions depending on the state (or context) one is in. This is typically framed as visiting different casinos, wherein one has a set of slot machines to choose between playing; the context is the casino, the action is the slot machine play, and the reinforcement learning problem is choosing the best machine in each casino based on reward receipt. This setup is the reinforcement learning analogue of supervised input-output learning, such as labeling images. Contexts are analogous to images and actions to labels. The supervised case and the reinforcement one differ only in that actions must be tried and compared in order to determine the correct response for each input. This sampling occurs via noise in network firing rates and the resulting actions.

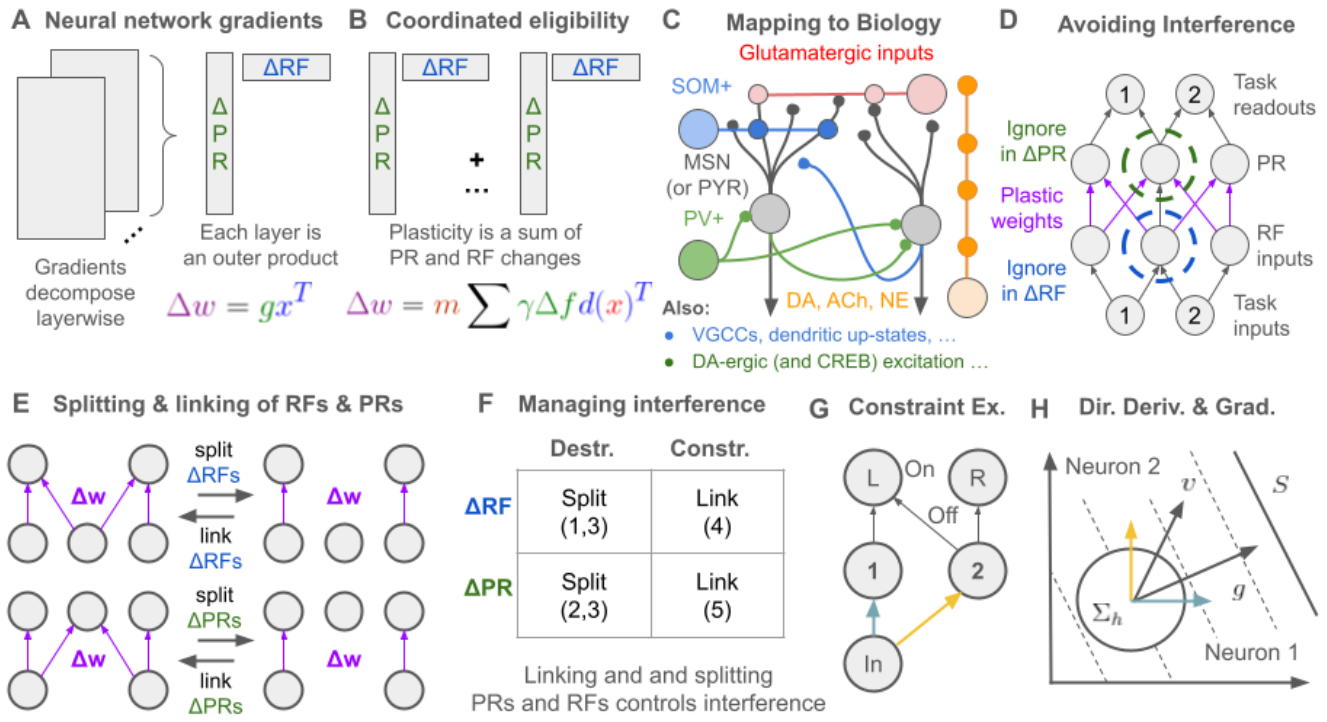

Figure 1: Gradients, interference, and the coordinated eligibility model. (A) Neural network loss gradients decompose into population representation changes and receptive field changes by layer. (B) In the CEM, plasticity is a sum of outer products of these. (C) Example mapping to striatal biology. A similar mapping to cortical biology, with pyramidal neurons rather than MSNs as the principal cells is also plausible. (D) Interference arises when population response changes involved in learning one task also impact another task, and similarly for receptive field changes. Orthogonalizing either population response changes or receptive field ones removes interference. Merging them creates interference, which can be used to generalize. (E) Diagram of merging and splitting eligibility for RFs (top row) and PRs (bottom). In the top row, an input is either included or left out of two candidate receptive field changes occurring in downstream neurons. In the bottom row, a neuron is either included or left out of the population response change associated with two inputs. (F) There are 2x2 cases in E, organized as a table here. One can either split or link both RF and PR changes. Any combination of table elements can theoretically occur in a single layer of weights. (G) A simple interference removal example. The same input "In" is applied given two readouts L, and R. The loss specifies that L should be on, and R should be off. Potentiating the left pathway (teal) but not the right pathway (yellow) will accomplish this goal. A continual learning scenario involving only L will tend to drive both neurons 1 and 2, by contrast. (H) L's activity in panel H is constant on a manifold in the firing rate space reached by driving 1, 2, or both. Which change occurs depends on their noise covariance, yielding activity change in some direction v, if noise is not isotropic, or in direction g if it is. If only one neuron varies at a time, such as unit 1, then only the yellow dimension of potential population response changes will be eligible for trial-and-error updating, and therefore subsequently "moved along" until reaching S.

Any sensible RL model can solve contextual bandit tasks, but changing contexts generally induces experience-dependent forgetting in networks if learning is done online and in a blocked fashion. Furthermore, the typical learning approach of following a policy gradient, which is the equivalent of gradient descent for image classification, also does nothing to manage generalization of learning from one context to another. Therefore, we demonstrate the coordinated eligibility model controlling both such forms of interference below.

Avoiding forgetting via selective receptive field plasticity

When a feature is shared by multiple stimuli, it may improperly suggest taking particular actions, leading to interference. Perhaps the most intuitive form of interference control is thus the capacity to ignore those stimulus features shared across stimuli when changing responses to them. Selective receptive field plasticity, controlled by the d terms in our CEM equations (1), can accomplish this. To illustrate this mechanism computationally, we simulated our network model solving a contextual bandit task involving 20 contextual inputs, each of which required taking one of 5 alternative actions. (These numbers are purely for illustration purposes - nothing about the mathematics of our results changes with scale.) The inputs were vectors of Bernoulli random numbers with P(success)=0.15 (figure 2C) and therefore included moderate amounts of feature (i.e. neuron activation) overlap (figure 2D). We also guaranteed each input had at least 1 unique feature, for reasons discussed below. Learning was grouped hierarchically by "days" and "blocks". Days referred to one pass through the training environment, in which the network was presented with 20 trials in context 1 (the first "block"), 20 in context 2 (a second "block"), etc. Once a block of trials for context 20 was completed a "day" therefore ended and the next "day" began.

For our model comparison, receptive field plasticity vectors d were restricted in the CEM to be orthogonal to previously encountered gradient ΔRFs. This was accomplished via projection, with projection matrices for each context being updated during other contexts. That is, each context had an associated noise-covariance matrix which progressively lost dimensionality according to other task's inputs (details in the supplement). Whereas the standard policy-gradient algorithm exhibited substantial interference between contexts, the CEM significantly improved learning times and task performance (figure 2, panels E-H). Specifically, at every return to a previously learned-about context, performance was worse for gradient descent than it had been when that context was last encountered (panels E,F). This is abolished by the CEM (also shown in panels E,F), and total cumulative error, also referred to as regret, was diminished by about 20 percentage points (panel G).

We expected improvement under the CEM to result in part from decreased learning about irrelevant dimensions in network inputs, and we verified this. To do so, we computed the principal components of the network weight dynamics (panels I,J) to examine their frequency content, on one hand, and we isolated the biasing impacts of shared inputs on the other (panels K,L). The primary principle component of the weight dynamics in both simulations was strongly aligned with the direction from the initial to the final weights ($\rho \approx 0.9$), whereas every other significant dimension was primarily oscillatory, with a period of 1 day (panel J). This periodicity is natural, because the inputs cycle daily. We also therefore computed the effective weights from the average input (taken over all contexts) to each readout at every trial during learning. These weights are just entries of $W_r\text{ReLu}(W_h x_{avg})$ for x_{avg} the average input, W_r the readout matrix, and W_h the hidden weights. This verified (panel K) that gradient descent repeatedly learns and unlearns associations between shared input and each readout. These bias the network to respond to non-diagnostic input cyclically, an effect that is largely abolished by the CEM. We also observed (not shown) that computing noise updates to orthogonalize ΔRFs against one another, in addition to the orthogonalization against other inputs, was even more effective at removing interference. This occurs when ΔRFs are restricted to unique subsets of neurons coding unique input features, for example. In such a scenario, weight changes are incapable of conflicting with one another, not just in how much they introduce functional conflict. As such, the network is effectively segregated into non-interacting sub-networks during learning.

Avoiding forgetting with population response eligibility

Our second avenue for managing interference involves context-dependent processing, manifest here as contextual differences in action readouts. (Note that the sense of "context" here differs from the contextual-bandit sense, which refers to input.) Each task in our second simulation therefore had its own set of readouts "listening" to network activity. This could occur when frontal circuits change motor or pre-motor responsiveness to different striatal populations, and is common in neural network research, for example. Appropriate responding therefore requires some indication of context, but in general this may be absent from network inputs. Realistic context signals may be internally generated, or available via a combination of memory and latent state inference, however. Interference arises in these settings because changing representations to drive one set of readouts will generically either introduce noise or aliasing (i.e. differential, specific drive) in other readouts.

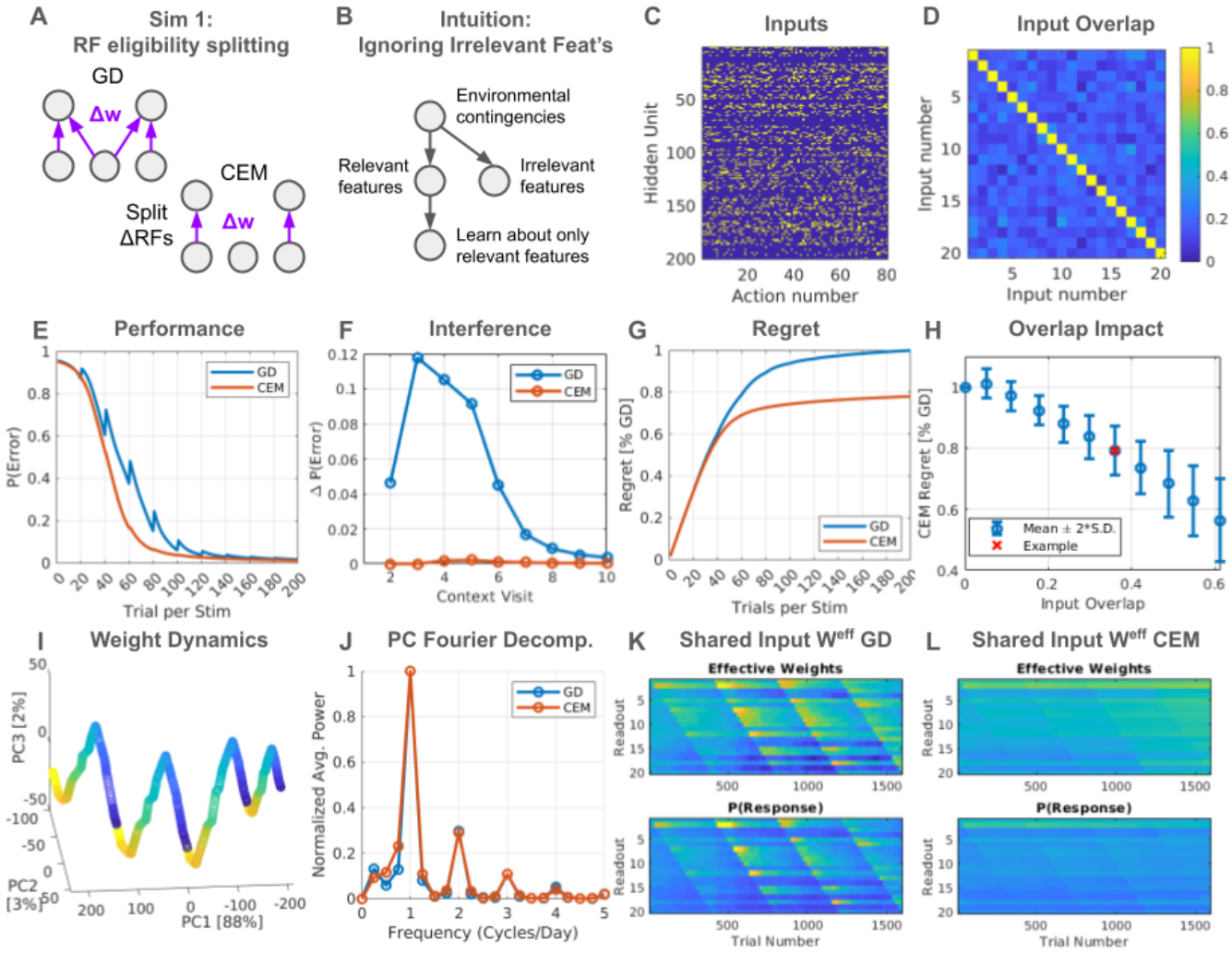

Figure 2: Minimizing interference with selective receptive field plasticity. (A) Comparison of gradient and CEM eligibility in this simulation. Receptive field changes to hidden layer neurons are limited to orthogonal stimulus features in the CEM. (B) RF splitting here is similar to ignoring non-diagnostic input features. (C) Inputs to the network are random feature vectors. (D) These have modest average pairwise overlap, stemming from their many shared non-diagnostic features. (E) Comparison of performance between gradient descent (GD) and the coordinated eligibility model (CEM). Each stimulus has a learning curve which specifies, for each trial of stimulus presentation, the probability that the network responds correctly. These are averaged and displayed. Because stimuli are presented in 20-trial blocks ("days"), and are subsequently returned to after learning occurs for other stimuli, GD exhibits decreased performance at these return times relative to the previous context visit, whereas the CEM does not. (F) Change in performance at context return induced by interference. Non-monotonicity arises because there is limited range for performance loss at the initial 20-trial mark. (G) The CEM generates less regret (cumulative error) than GD by avoiding performance decrements. (H) Interference is greater or lesser, such that improvement of the CEM over GD is greater or lesser, according to input overlap. (I) PCA of network weight dynamics under GD. (J) Averaged, normalized Fourier decompositions of the 5 principal components following the first. All substantial PCs except PC1 are oscillatory. (K, top) Impact of shared input on readout activity for first four days of GD weight history. The network repeatedly learns and unlearns readout associations to the shared inputs. (K, bottom) softmax of the above. Shared, irrelevant input cyclically biases the network towards whatever response is currently being learned. (L) Same as (K) but with CEM, on the same color scales. Oscillatory activity is strongly damped.

Tuning receptive fields to avoid interference in purely context-dependent processing is not possible, however, because there will be (by definition) no unique input features. However, firing rate variability in a network's hidden layer determines which aspects of population responses are subject to change. By ensuring that changes involved in driving one set of context-dependent readouts are orthogonal to other contexts' readouts, network plasticity based on the CEM is therefore able to avoid interference. As with the ΔRF case, this orthogonalization can be computed online by removing dimensions from the noise variance as tasks are encountered, or it can be based on an a-priori partition of the population response space. Biologically, autonomous CREB-mediated excitability drift in amygdala neurons has been shown to recruit partially overlapping, partially distinct collections of neurons into memory encoding (Zhou *et al.* 2009; Yiu *et al.* 2014), for example, and CREB similarly mediates excitability in striatum (Dong *et al.* 2006). For simplicity, we assume here that since the readouts are known a-priori, the noise correlations which orthogonalize sampling between them are as well.

Concretely, we model this contextual-processing scenario by presenting networks with one stimulus and using multiple overlapping sets of readouts to select actions. We simulated 20 such contexts, with five actions available in each. In analogy with the RF simulations, readouts were generated as Bernoulli random vectors with P(success)=0.15 (figure 3C). This generated moderate amounts of readout overlap, as seen in figure 3D. Simulations were done using the same trial counts, blocking parameters, and neuron parameters as the previous section, but we used a mixture model for gradient project that interpolated between isotropic and projective noise (our "anisotropy" parameter below). We found that the CEM reduced interference considerably, with cumulative regret in the example reduced by roughly 25 percentage points compared to gradient descent (figure 3G). The extent of improvement depended on the anisotropy parameter, as illustrated in panel H. With zero anisotropy, the network performed gradient descent, whereas with a parameter of 1, it orthogonalized updates completely. The best performance occurs between these values, when weight updates are at intermediate (rather than potentially high) angles to gradients.

As with our RF simulation, we sought to explain the performance effects in more detail on the basis of weight dynamics. To do so, we again computed effective weights, this time segregated into "correct" and "other" categories and averaged (figure 3, panels I,J). These indicated that for gradient descent, like the previous simulation, effective weights into correct readouts oscillated over training (panel I). Similarly, competitor readouts saw their weights decrease over training in an oscillatory manner (panel J). The CEM abolished both of these effects, and the probability of choosing a correct action therefore increased monotonically (for the CEM) over training (panel K). Nonetheless, as with our RF simulation, we observed that the underlying weight trajectory retained substantial oscillation (panel L), which would be removed by a yet more stringent sampling-orthogonalization procedure as discussed in the RF simulation section.

Coordinating eligibility for compositional stimulus-action relationships

As a final application to interference avoidance we consider tasks with multidimensional, compositional inputs that require compositional responses. Many objects can be described as bundles of features of varying statistical interdependence, and when learning a task, subsets of these features might dictate separate elements of an appropriate response (figure 4B). Brain regions with convergent input from diverse pre-synaptic partners would be well served by plasticity mechanisms that could manage responsiveness to the compositional building blocks in such mixed inputs, and the resulting compositionality would also facilitate combinatorial generalization (as in e.g. O'reilly 2001). While there are circuit level architectural mechanisms that can gate attention to specific features in order to govern responding (Frank & Badre 2012; Niv *et al.* 2015), we consider here how such functions can arise from learning rules themselves. As we show, feature-dependent population-response plasticity can work in tandem with receptive field plasticity to produce compositional learning.

Our basic insight is that exploratory population response noise need not be independent of dendritic selection mechanisms, and indeed likely isn't in general. A candidate population response change q can therefore be considered in conjunction with a receptive field change p, leading to a synapse update matrix qp^T, yoking compositional building blocks of representation changes to those of receptive field changes. Multiple such pairs of changes can be sampled during the same learning episode, and the result constitutes a form of dimensionality reduction; instead of performing sampling-based gradient (or directional derivative) estimation on the complete bipartite graph of representations and receptive field updates, some smaller graph is assumed. For example, striatal neurons multiplexing sensory information from different areas may have different population-level representations related to different modalities, and it may be useful to explore holistic changes in these representations in a way that keeps processing changes separated by (and internal to) different modalities. This would imply yoking a representation change associated with one modality to a set of receptive field changes "listening" to that same modality. As above, we are not suggesting other mechanisms cannot accomplish similar goals. Instead, we show that synaptic plasticity could be

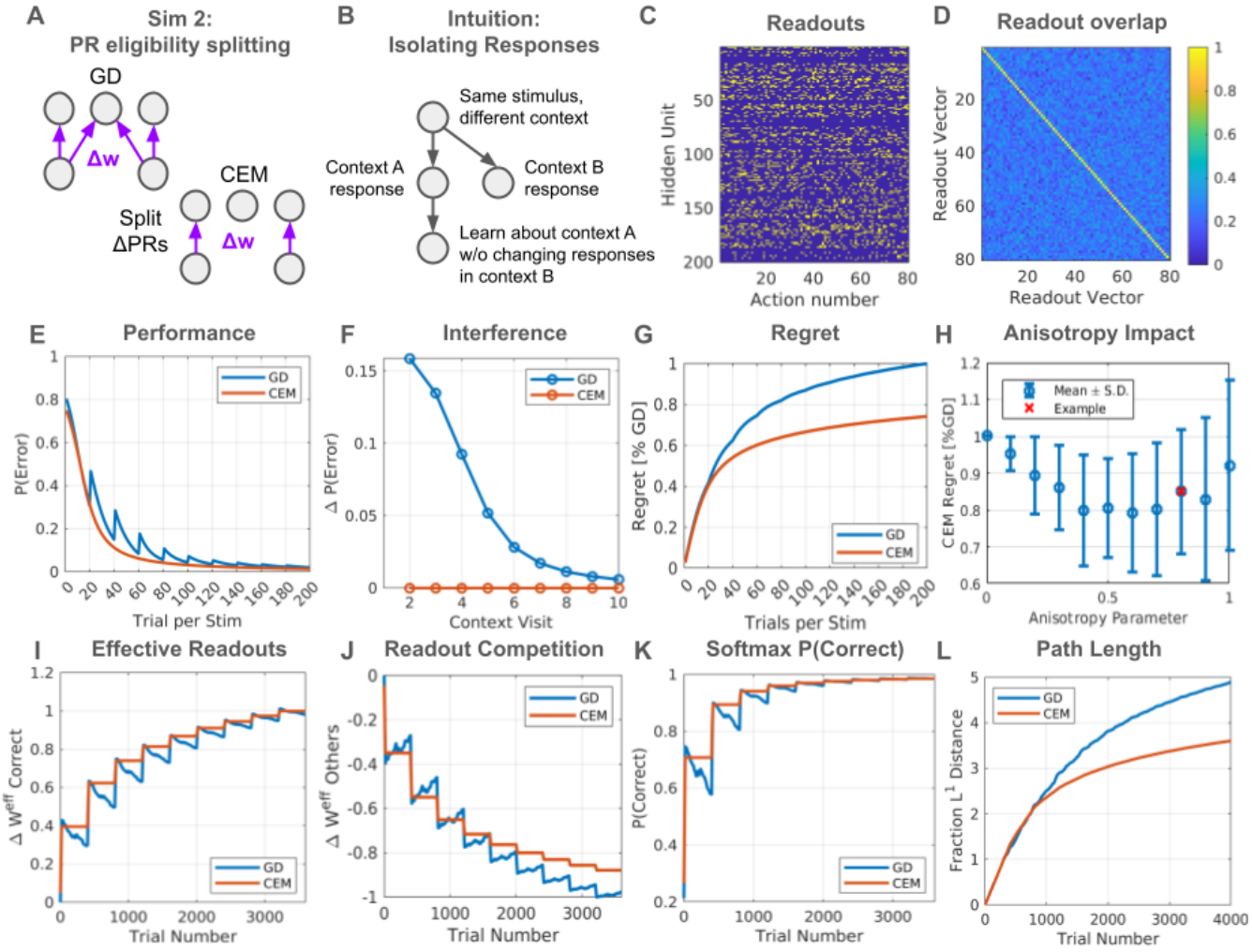

Figure 3: Minimizing interference with selective population response plasticity. (A) Comparison of gradient and CEM eligibility in this simulation. In the CEM, population response changes are sampled via orthogonal noise dimensions, segregating learning across contextual decoding subspaces. This is illustrated by the middle neuron being removed from PR change eligibility. (B) Intuitively, this simulation models isolating stimulus response changes when multiple contextual responses make use of shared aspects of internal representations. (C) Random readouts used in the task. (D) Readout vector overlap. (E) Comparison of GD and CEM performance. GD has discontinuous performance when returning to a context (every 20-trial mark here) because learning in other contexts changes representations that are read out by other contexts' readouts. The CEM model avoids this. (F) Average change in performance on block entry relative to exiting the previous block of the same context. (G) Interference leads to worse performance under GD. (H) Performance improvement depends on how strongly orthogonalized the CEM PR changes are. (I) Effective readout weights from the (single, shared, constant) input to each contexts correct answer rise monotonically under the CEM, but rise and fall under GD as each new context suppresses weights which aren't aligned to the current readout. (J) Competition from incorrect readouts falls when each stimulus is trained, then rises again under GD. This effect is abolished under the CEM. (K) The probability of providing a correct answer rises and falls under GD in line with (I and J). (L) As one would expect, the CEM moves more directly from the initial weight configuration to its final one, as shown by the L^1 path length (summed absolute weight increments and decrements) as a fraction of the L^1 distance from initial to final weights. Both GD and CEM relative path-lengths exceed 1 because both exhibit non-monotonicity in their weight changes. Non-monotonicity in the CEM case is orthogonal to readouts however, and therefore does not manifest in performance non-monotonicity. A strictly better algorithm might also remove this "invisible interference".

either pre-configured or tuned online to perform them in some circuits.

To explore this idea, we simulated simple compositional task sets. Tasks were constructed by generating random mathematical bases for network inputs and hidden layer responses, and associating elements of each basis with elements of the other. The desired input-to-hidden transformation was thus an orthogonal matrix $W_h^* = BA^T$, with input feature vectors encoded in A and hidden ones in B. Stimuli were generated as compositions of the basic features, and target outputs were generated as compositions of graded responses to these input features. The strength of an input feature therefore determined the extent to which its associated output was required. The number of input features per stimulus was determined with a compositionality parameter $C \in \{1, ..., n\}$, where n is the network width in feature groups. This yielded a set of n-choose-C potential inputs, which grows rapidly when C is not approximately 1 or n. Therefore, we selected stimuli to include all n of the features with equal frequency. For example, with $C = 2$ and $n = 4$, neurons in the pairs {1,2}, {2,3}, {3,4}, and {4,1} would be taken to be active to varying degrees as inputs.

To perform graded feature-matching, we defined a parameter L, a "linking number". As noted above, cross-layer feature dependencies form a bipartite graph, with links between layers set by the non-zero qp^T combinations discussed above. Whereas gradient descent operates on the complete bipartite graph, involving every possible (q, p) pair, the best coordinated eligibility model operates only on those input-to-hidden feature links required for the task, reducing dimensionality substantially. The linking number L interpolates between GD and this best model by setting the in- and out-degrees of each feature vector in the hidden and input layers' dependencies, respectively.

Our results are illustrated in figure 4, showing improved performance of the CEM relative to GD. This results from reducing the number of candidate network changes sampled by the plasticity rule. The reduced set of candidate weight changes avoids interference that would otherwise arise from incorrectly associating input features with output features that appear jointly with the inputs' associated targets. As a result, the coordinated eligibility model solves the example task set more quickly, with less interference, and accrues less total error than GD, as with our other simulations. The impact of linking number shows that reductions in dimensionality improve performance monotonically, as one would expect (figure 4H). Thus, feature matching via coordinated receptive field and response eligibility can drastically improve learning in compositional tasks relative to gradient descent.

Linking receptive fields for generalization

We now consider constructive interference, in the form of generalization or learning transfer. As in our previous simulations, this can be accomplished with either selective receptive field plasticity or selective population response plasticity. In the former case, by contrast to the increased specificity of receptive field changes used to avoid interference, networks can also coarse-code (or decrease the specificity of) plasticity in order to change responses to multiple inputs based on feedback related to any single input. Notably, this is not the same as coarse-coding representations, which roughly refers to neurons responding to many inputs, because coarse coding of plasticity does not inherently result in coarse codes for activity. But by updating neural responses to multiple inputs simultaneously, such plasticity can generalize learning from one input to another.

This type of generalization can also be construed as a form of inference. For example, co-occurring features in a network's input might signal an object's identity, such that response changes to subsets of features ought to be generalized to the whole partially observed object. In particular, this means that when a different subset of object features signal the object's identity, updated behaviors should still apply. Whether inferences like these are appropriate will depend on many factors, but in cases where they are, coarse-coded receptive-field plasticity is one potential mechanism for supporting them. As with tailored RF plasticity for interference avoidance, dendritic features such as spine clustering, active electrical processes, and branch-level excitability gating are all mechanisms that could reasonably serve this function.

To illustrate this, we ran network simulations in which pairs of inputs were associated pairwise with the same target outputs. One network learned the resulting contextual bandit task using gradient descent, while in a second network, receptive field plasticity was yoked across paired inputs. The results of the two simulations can be seen in figure (5), showing that learning speed is roughly doubled under the given coordinated eligibility model, as one would expect. Transfer of learning from one stimulus to its paired input can be seen directly in panels J and K, in comparison to the absence of transfer visible in F and G. These simulations therefore demonstrate one mechanism by which similarity based clustering or other procedures could tune learning rules to generalize, in a manner consistent with solving tasks via gradient-like improvement.

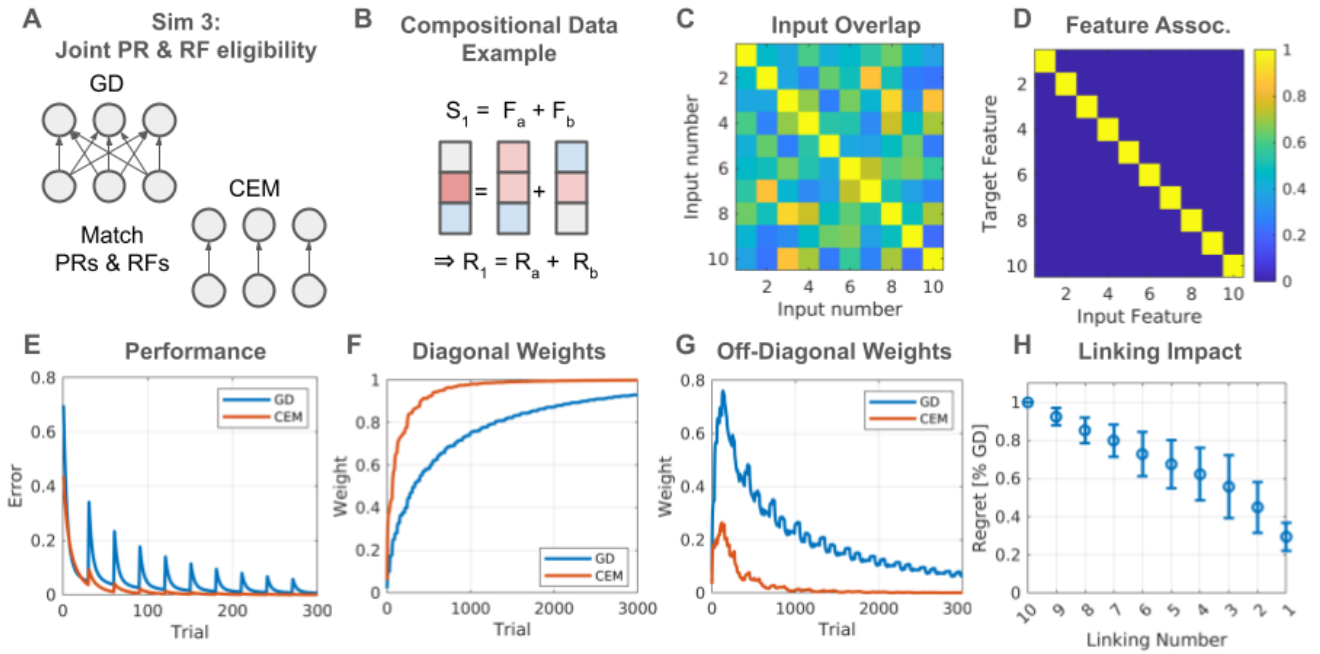

Figure 4: Coordinating receptive field and population response eligibility to de-mix plasticity for compositional learning. (A) Comparison of GD and CEM. When network connectivity is dense, GD learns dense updates, whereas CEM can learn sparse, targeted updates. This can be useful in cases where e.g. multi-modal information is multiplexed in neurons' activities, but learning should respect modality. (B) Example of such compositional data: A stimulus is composed of several features, and each feature independently dictates an element of the network's response. (C) Input second moments for 10 inputs. (D) Feature associations. Each input is a composition of features, and each output is a composition of the input features' individual output associations. Each input feature dictates one unique output feature. (D) Context-average performance over time. GD continuously produces interference between inputs because it both wrongly associates multiple input features with each output feature, and wrongly associates multiple output features with each input feature. (E) Summed weights connecting input features to their "true" output features. The CEM here is a mixture-model with $P = 0.8$, so that performance closes 80% of the gap between GD and ideal behavior (which would be seen with $P = 1$). (F) Summed weights connecting erroneously associated input and target features, as a fraction of the true "diagonal" ones. GD dedicates a majority of it's synapse changes to erroneous associations early on, diminishing to roughly 15% towards the simulation end. This is diminished by roughly a factor of 5 (i.e. $1/(1-P)$) in the CEM mixture model, whereas incorrect associations are abolished in the case of $P = 1$ (not shown). (H) Impact of linking number, by compositionality. Improvement over GD increases monotonically as eligibility is narrowed from all candidate features (linking = 10) to only the truly associated feature (linking = 1). This has a greater impact for more compositional sets of inputs (here comp. = 10) in which each input has more features.

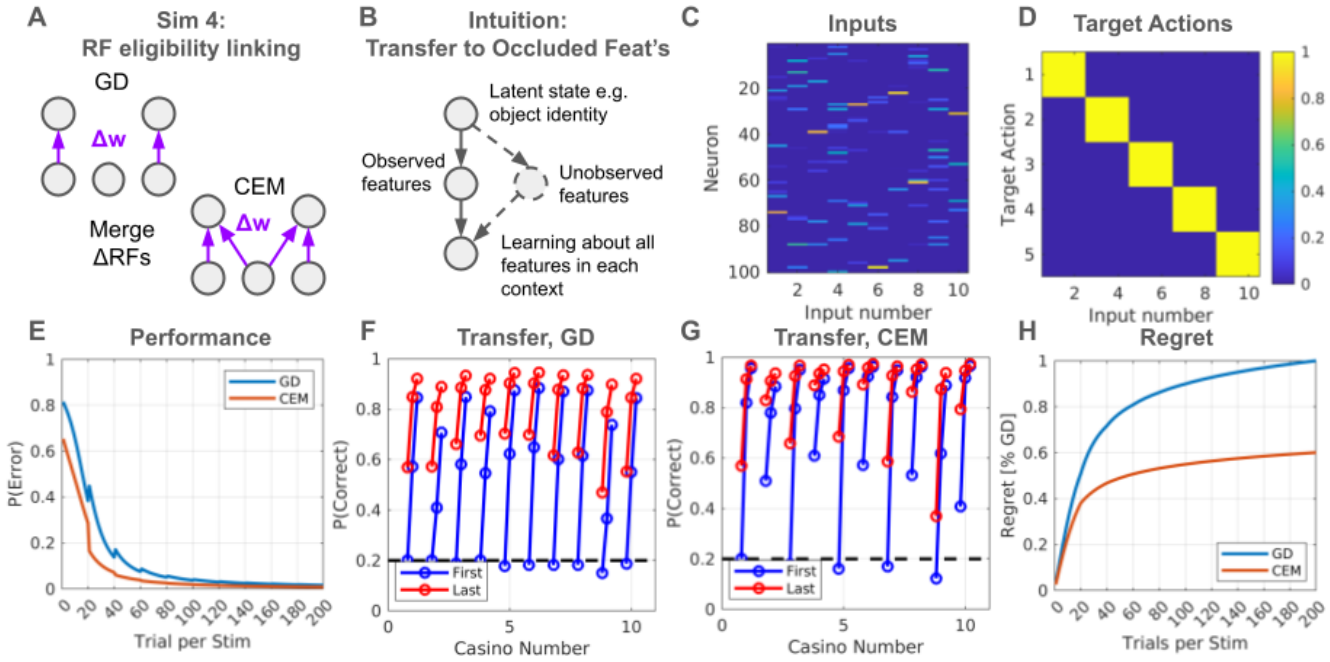

Figure 5: Generalizing with selective receptive field plasticity. (A) Comparison of gradient and CEM eligibility in this simulation. In the CEM, receptive field changes are coarse coded over inputs which share targets. (B) Coarse coding input plasticity can be interpreted as yoking learning over observed and inferred unobserved features. (C) Inputs are sparse positive random vectors. (D) Five target actions are allocated to 10 inputs. Inputs 1 and 2 share a target action, as do 3 and 4, etc. (E) Performance of GD and the CEM. Whereas some minor destructive interference is visible in the GD curve (at block switches), the CEM has a prominent improvement at the first block switch (20 trial mark). This occurs because performance on each stimulus is better on a new context block than when it left the previous context block for a stimulus, a point illustrated further in F and G. (F) First and last trial performance within a context ("casino") for GD, for the first three blocks ("days") of learning. Blue curves are performance on the first trials of each block and red curves are performance on the final trials. The first trials in context 2, 4, etc show performance around chance (20%), and learning proceeds independently over days for different contexts. (G) Same as F, but for the CEM. The first trial in each context shows performance comparable to the last trial in its paired context. For example, the first trial on day 1 for context 2 has performance comparable to that of the last trial on day one for context 1, which was presented immediately prior. Generalization can also be observed for later days and contexts. (H) Difference in regret (cumulative error) between the two models. Total error is (1 - 0.5*0.8)*GD, i.e. 80% of being halved, because our anisotropy parameter was set to 0.8.

Linking representations via overlap for generalization

As with receptive-field plasticity, coarse-coding population-representation eligibility can also produce generalization. In this case, different contexts might make use of different representations (or different aspects of "the same" representation) for information about any particular stimulus, but representational plasticity associated with one context would be yoked to representational plasticity associated with the other.

We therefore simulated a version of this representation-based generalization using eligibility for population plasticity that was distributed over multiple actions. Specifically, we simulated a network in which five inputs were observed in each of two contexts, the analogue of our RF linking simulation. This could represent performing two binary classifications for each such input, with the classification prompt determining which contextual readouts are applied to the network. For example, it could model being asked, for each of five animals, if the given animal is a mammal or bird, and whether it is small or large. Known or inferred shared information between these answers can be used within the plasticity rule to generalize learning, such that if (e.g.) all mammals previously observed are large, then learning to respond "mammal" can produce a learned response of "large" as well. Trial-wise covariation of populations driving these two classifications thereby serve to generalize learning.

Our simulation results in figure 6 illustrate these points, showing that gradient descent does not transfer information, whereas yoking receptive field plasticity across responses reduces total learning time substantially. As with our RF simulation, panels J and K directly illustrate transfer from one context to its pair in the CEM. The absence of transfer under gradient descent is shown in panels F and G. Hence, regardless of the mechanisms that might exist to infer the appropriateness of such processes, or the specific architectural divisions of labor between brain areas which might support their natural pre-configuration, we have shown how coordinated eligibility in gradient-like learning can support generalization via the geometry of population-response plasticity.

Coordinated eligibility generalizes earlier models

Having discussed the functional aspects of our coordinated eligiblity model, we now explain how it generalizes other important formulations. The Beinenstock-Cooper-Munroe (BCM) rule, two-threshold Ca^{2+} model of plasticity, and REINFORCE algorithm are all equivalent to one another under certain circumstances, and they can thus be jointly generalized. Our coordinated eligibility model incorporates simple formulations of each as special cases. Basic information about these models is provided in figure 7.

The version of the BCM model we consider takes synaptic plasticity as:

$$\Delta w = y(y - \theta)x \tag{6}$$

Post-synaptic neuronal activity, in the form of a spike rate, is denoted here by the variable y, whereas the pre-synaptic neuron's activity is represented by x. The synapse strength is represented by the variable w, and its change over time is denoted by the Δw. The term θ denotes a "floating threshold" which measures activity in the post-synaptic neuron over time. Biologically, this roughly summarizes the calcium dynamics in post-synaptic cells, which play a large role in determining plasticity. Large calcium fluxes tend to induce LTP, whereas small ones tend to induce LTD, and the boundary between these is adapted over time. Theta represents this boundary, and NMDA receptor subunit compositions are known to modify Ca^{2+} fluxes in response to activity, partially determining such a threshold (Scott & Frank 2022 provides a recent review). Firing rates in this model are non-negative, implying the sign of $y(y - \theta)$ is determined by $y - \theta$. When y is greater than θ LTP is induced, whereas LTD is induced otherwise.

The BCM model leaves out various important factors controlling plasticity. First, there are prominent impacts of neuromodulators such as dopamine, acetylcholine, and norepinephrine in many systems. Calcium flux is also mediated by NMDAR and voltage-gated calicum-channel (VGCC) composition in a synapse local way, suggesting that the floating threshold separating LTD and LTP should be local too. And finally, synaptic eligibility for change is known to be modified by factors such as shunting fluxes on spines and dendrites, which pro-actively modify calcium dynamics rather than being passively driven by their inputs. The two-threshold model of calcium induced plasticity and the REINFORCE model incorporate some of these aspects of spine-local and neuromodulatory impacts, and our coordinated eligibility equations can functionally model all of them.

The simplest two-threshold calcium model of plasticity is piecewise constant, with three regimes. In this model, high calcium flux elicits LTP, intermediate fluxes induce LTD, and small fluxes have no impact. Such a model can be formulated to have linear behavior around the LTP to LTD transition, so that being slightly above threshold yields minor LTP and being slightly below yields minor LTD. If this calcium flux is set by the convergence of a variable post-synaptic response amplitude and a fixed input activity then, denoting the LTD-LTP threshold θ_h and

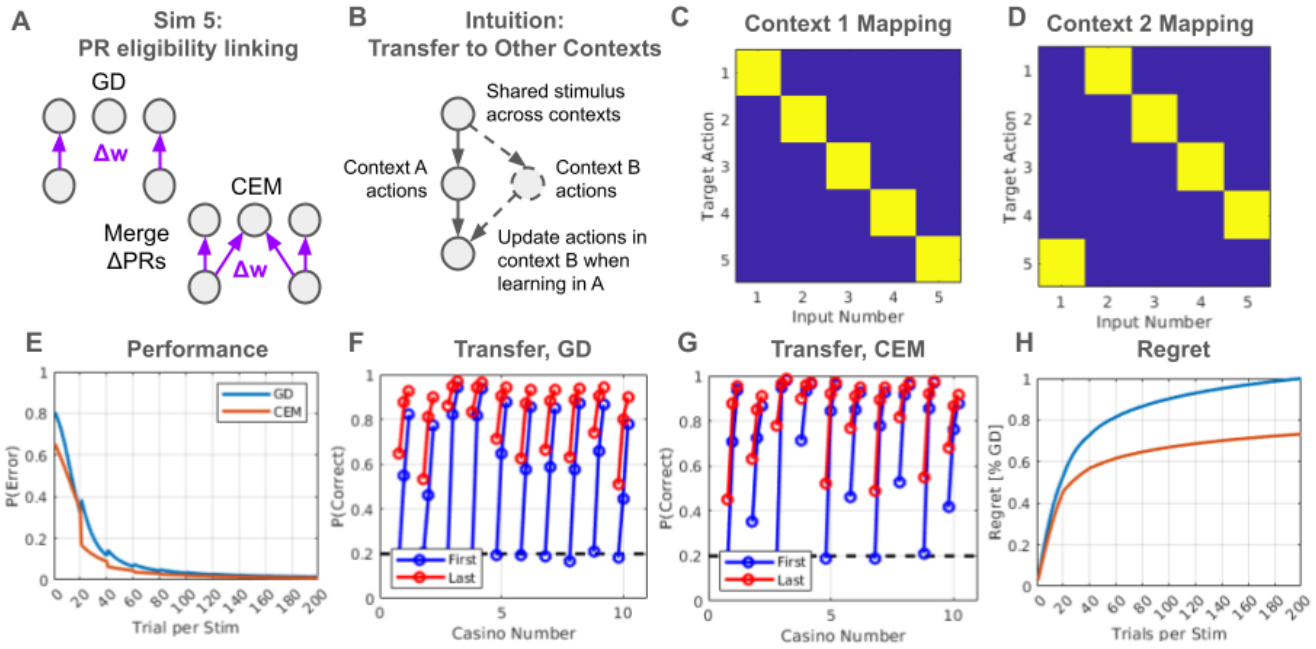

Figure 6: Generalizing with selective population response eligibility. (A) Comparison of gradient and CEM eligibility in this simulation. In the CEM, population response changes are coarse coded over target actions. (B) This simulation models situations in which learning to change one type of behavior related to a stimulus should result in changes to other behaviors related to the same stimulus. (C) Each of 5 inputs is associated with a unique action in one context. (D) The same 5 inputs are associated with a second action in a different context. (There is no inherent relation between action 1 in context 1 and action 1 in context 2, they are arbitrary labels associated with different readouts vectors) Inputs 1 and 2 share a target, as do 3 and 4, etc. (E) Performance for GD and the CEM. (F,G) First and last trials within a context for GD and the CEM. The first trial in each context shows performance comparable to the last trial in its paired one for the CEM model, whereas there is no shared learning under GD. (H) Reduction in cumulative loss attributable to generalization. In this simulation, differences in learning speeds for different associations under GD are heterogeneous (see panel F), such that regret improvement under the CEM depends on which random contexts are harder or easier to learn about. As with the previous simulation, expected improvement is 0.5*0.8*GD.

the LTD-no-change threshold θ_l, we have:

$$\Delta w = \begin{cases} 0 & y < \theta_l \\ (y - \theta_h)x & y \approx \theta_h \end{cases} \tag{7}$$

If we identify the θ_h here with θ from BCM, and consider neurons which are significantly excited by a stimulus to have only small fluctuations in firing rate around θ, then we can see that this is a local BCM model (in the sense of a Taylor expansion), restricted to significantly active synapses. Linear two-threshold calcium models therefore generalize BCM in the sense that the lower threshold for plasticity is some number greater than zero, and behavior around the higher threshold is (locally) equivalent to BCM. A piecewise polynomial model in which the low-to-high threshold transition is quadratic would be an even closer approximation. The linear case makes an important connection with gradient descent, however, because gradients reflect local linear relationships.

Setting aside for a moment those postsynaptic neurons which are inactive, the linear two threshold model and the BCM model are special cases of a third equation, which was introduced independently in the reinforcement learning literature. Specifically, REINFORCE algorithms update weights according to a family of related equations, some of which can be written:

$$\Delta w = (r - \bar{r})(y - \bar{y})x \tag{8}$$

The quantity $(r - \bar{r})$ here represents a reward prediction error (RPE) resulting from the activity in the network, and $\bar{y}$ is the trial-average firing rate of the post-synaptic neuron. It is an example of neuromodulated Hebbian plasticity, because the Hebbian product of $(y - \bar{y})$ and x is further multiplied by a neuromodulatory factor.

The model (8) also differs from BCM in the critical respect that the post-synaptic plasticity threshold θ, which is taken to be $\bar{y}$ here, is computed on a stimulus-by-stimulus basis. This conforms with biology insofar as different stimuli consistently elicit activity in specific spine populations. The expected value of the weight update Δw is then the gradient of reward with respect to the network weights, under the assumptions of linear neurons and Gaussian, isotropic noise, for example, meaning the model performs gradient descent. These conditions and related ones for different network models are critical, in the sense that REINFORCE algorithms are defined as gradient estimators, not with reference to the specific equation above. This is a key distinction from the coordinated eligibility model, which does not compute generic task gradients, does not generally have isotropic firing rate variation, and does not need to be redefined in network specific ways for networks with different firing rate properties, for example. Our results are therefore general and flexible in ways that the REINFORCE model is not.

In particular, the coordinated eligibility model is a generalization of REINFORCE in the sense that it reduces to equation 8 when: $m = (r - \bar{r})$; every neuron takes f as the identity function; every neuron uses the same dendritic calcium shaping $d(x) = x$, and the same gain γ; all neurons are subject to i.i.d. Gaussian variability; and the post-synaptic baselines are all $\langle f \rangle$. In this case, the neurons are linear, plasticity is proportional to synaptic input, neuromodulation reflects reward prediction errors, and all neurons exhibit identical uncorrelated variability. This in turn reduces to the two-threshold model if the RPEs are all the same and the high threshold is set as the average firing rate. Finally, if neurons are always in the vicinity of the high threshold, and stimuli are indistinguishable by the post-synaptic neuron, we recover a local BCM rule.

Discussion

While biological learning is often considered as an approximation to gradient descent, we have developed the alternative perspective that biological phenomena – specifically active receptive field structuring via dendritic calcium manipulations and heterogeneous firing rate variability – might endow neuromodulated Hebbian forms of plasticity with important ways to control synaptic credit assignment. We introduced a plasticity rule, which we referred to as the coordinated eligibility model, and showed that the inductive biases afforded by the biological elements of our model can be leveraged to manipulate task-learning relationships in useful and fundamental ways.

While our coordinated eligibility model is essentially mathematical, we have emphasized links with various microcircuit elements. In the cortex, for example, population response plasticity may be orchestrated by parvalbumin expressing interneurons such as basket and chandelier cells, via their impacts on firing rate variability, and MSN collaterals may mediate similar coordinated changes in striatal circuits. Receptive field plasticity properties, on the other hand, are known to be responsive to dendritic calcium manipulations, which are impacted by somatostatin interneuron activity in some contexts (Higley 2014; Naka *et al.* 2019; Kecskés *et al.* 2020). Empirical work has shown that experimental dendritic calcium manipulations do impact interference between tasks during learning (Yang *et al.* 2014; Cichon & Gan 2015; Sehgal *et al.* 2021), and furthermore, that subpopulations of PV+ and SOM+ interneurons in superficial layers of sensory cortex are responsive to both input from VIP+ interneurons

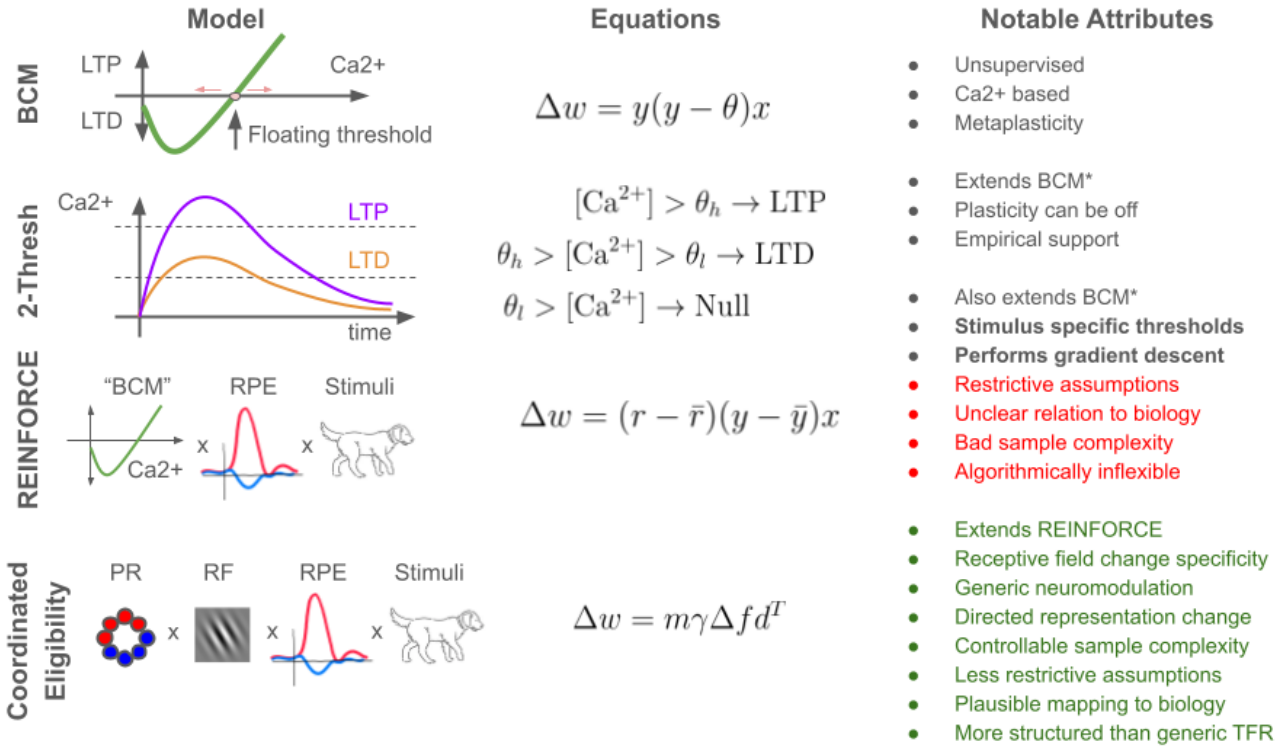

Figure 7: Plasticity model relationships. BCM and the two-threshold calcium model capture important biological aspects of plasticity, while the REINFORCE model and the coordinated eligibility model generalize these in certain senses. The REINFORCE model can be considered as a stimulus-specific, RPE gated linear BCM model, whereas the coordinated eligibility model is a generalization of REINFORCE in which receptive fields and population responses are subject to flexible control, rather than being determined by the gradient of reward with respect to network weights. The term $\gamma\Delta f$ is analogous to $(y - \bar{y})$, which we stress has substructure relative to REINFORCE and need not comply with key stipulations of REINFORCE, such as isotropic firing rate variability, baselining by post-synaptic average activity, or even perfect stimulus discriminability. This distinction is important to make because REINFORCE refers to a family of gradient-descent algorithms, implemented through a particular stochastic policy gradient estimate. The equation above is one form, specific to one type of network and reliant on other assumptions, whereas the model we call coordinated eligibility here encompasses a partially overlapping class of models which are always given by the equation above. These may not be REINFORCE algorithms in the contexts in which they are applied, and in general won't be.

and other cortical regions (Pfeffer *et al.* 2013). As such, our model predicts that the specific forms of coordination among these players during learning will control both the sample complexity of plasticity and the geometries of population responses and receptive fields. Feedback signals acting as "third factors" to plasticity are expected to covary with potential changes in these PRs and RFs to gate structured eligibility into realized plasticity. In this manuscript, we considered dopaminergic RPEs to play this role, but many other circuit elements could provide such third factors.

Our work also suggests that firing rate variability should be partially exploratory, that it should be factorizable into components representing potential functional changes, and that these components should align with learning dimensions. If in-vivo networks behave like this, we further expect that network state changes involving norepinephrine, acetylcholine, and top down input de-correlate activity in the sense of removing significant observable nuisance variance, rather than in the sense of making all neurons maximally independent, for example. That is, the increases in signal to noise ratios and cortical state "decorrelation" as discussed in optimal-cortical state literature are predicted to facilitate meaningful exploration of population-response changes by removing confounding or noisy features of neural responses, rather than by suppressing ensemble-based variation in cell groupings involved in relevant population coding. In these ways, learning dimensions would be expected to align with either information communicated by task knowledge or with inherent processing primitives embedded in the network. Complete decorrelation would be expected only when fine-tuning of population responses required truly high-dimensional hypothesis testing of potential functional changes.

Existing literature suggests fine grained coordination like this may indeed be the case. For example, L2/3 mediated cortical state decorrelation seems to involve some, but not other, populations of both SOM+ and PV+ cells (Pi *et al.* 2013; Pfeffer *et al.* 2013; Fu *et al.* 2014; Eggermann *et al.* 2014; N. Chen *et al.* 2015; Gasselin *et al.* 2021). Distinguishing between different "decorrelated" network states during learning is likely to be difficult however, because ensemble sizes or coordination scales in microcircuits are not generally known. Work on so-called critical cortical dynamics may provide an interesting connection in this regard however, given that a growing literature examines how network states facilitate information maintenance and transfer across scales (Ma *et al.* 2019; Karimipanah *et al.* 2017; Shew, Clawson, *et al.* 2015; Priesemann 2014; Shew & Plenz 2013; Beggs 2008). With luck, data on these and related phenomena will allow us to translate the mathematical features of our model into any number of more local circuit-specific hypotheses.

Finally, we note that our work is closely related to other developments in computational neuroscience, particularly in the recurrent neural network (RNN) literature. Work on RNN dynamics has increasingly focused on low dimensional processing, as embedded in networks by either hand-construction (Mastrogiuseppe & Ostojic 2018), or via regularization procedures applied to gradient-descent through loss-function modifications (Beiran *et al.* 2023). Our findings here indicate how biological plasticity may naturally support such low-dimensional learning - and the attendant computational advantages - via the functional properties of concrete cellular and molecular phenomena.

Acknowledgements

For helpful discussion, commentary, and feedback, we thank Mathew Nassar, Apoorva Bhandari, Rex Liu, Christopher I. Moore, Ian A. More, David Badre, Scott Susi, and the Frank lab. Daniel Scott was supported by NIMH training grant T32MH115895 (PI's: Frank, Badre, Moore). The project was supported by NIMH R01 MH084840-08A1. Computing hardware was supported by NIH Office of the Director grant S10OD025181.

Author contributions

D.N.S. and M.J.F. developed the research topic. D.N.S. conceived and developed the mathematical analyses, functional and biological interpretations, wrote code, performed simulations, and drafted the manuscript. M.J.F. provided extensive feedback at all project stages and on all topics. D.N.S. and M.J.F edited and redrafted the manuscript and prepared it for submission. D.N.S. and M.J.F revised it upon receiving feedback.

Declaration of interests

The authors declare no competing interests.

Supplementary material

The REINFORCE algorithm

Our main results are very closely related to the classic REINFORCE algorithm. For reference, we reproduce the specific relevant findings from Williams' 1992 paper here. REINFORCE was originally formulated for two-layer neural networks with weights w_{ij}, inputs x, and outputs y, by the pair of equations:

$$\Delta w_{ij} = \alpha_{ij}(r - b_{ij})\frac{\partial \ln(g_i)}{\partial w_{ij}}$$
$$g_i = P(y_i = \xi | w, x)$$

The quantities α_{ij}, r, and b_{ij} here are a learning rate, a "reward" and a "reward baseline". Two special cases are especially relevant to neuroscience. First, when y is a vector of Bernouli random variables ("spikes") with the probability of emission determined as a logistic function applied to Wx, then (with a few other details determined) the algorithm takes the form:

$$\Delta w_{ij} = \alpha(r - \bar{r})(y_i - \bar{y}_i)x_j \tag{9}$$

Second, when $y \sim \mathcal{N}(Wx, \Sigma)$, or in fact has any linear exponential family distribution, one arrives at (up to proportionality) the same conclusion. Hence, the results for either network construction are Hebbian algorithms, in the sense that they depend on the Hebbian product $(y - \bar{y})x^T$. Williams established that these algorithms are also policy gradient algorithms. In particular, he showed that the gradient of expected reward, with respect to the weights, is equal to the expected value of the weight updates themselves. Hence, updating the weights via a REINFORCE algorithm performs gradient descent on a reinforcement learning problem's loss. Williams' proof used something called the REINFORCE or "log-derivative" trick, which makes it difficult to see how more general noise cases relate to his result. Node perturbation, which functions by the same logic, also assumes a convenient noise form, whereas investigating the general case using linear algebra and (tensor) calculus yields our results below.

Gradients and the CEM

Using Einstein notation, basis vectors e_n, dual basis vectors e^m, h as the element-wise derivative of f, and D_h as matrix with h on the diagonal, the derivative of reward with respect to weights W is:

$$\begin{aligned}\frac{dr}{dW} &= \frac{dr}{dy_i}\frac{dy_i}{dW} = \frac{dr}{dy_i}\frac{dy_i}{df_j}\frac{df_j}{dz_k}\frac{dz_k}{dW} \\ &= \frac{dr}{dy_j}\frac{df_j}{dz_k}\frac{d}{dW}e^k W x = \frac{dr}{dy_j}(h_j e^j \otimes e_k)e_k x^T \\ &= \frac{dr}{dy_j}h_j e^j x^T = D_h b x^T \equiv g x^T\end{aligned}$$

On the last line, we have defined b as $(dr/dy)^T$ and g, the gradient output filter or gradient population response change, as $D_h b$. Compare this with the expected value of the reward-modulated Hebbian update, using the local linear form of r around $r(\bar{y})$, and denoting firing rate noise by λ_y:

$$\begin{aligned}\langle \Delta W \rangle &= \langle (r - \bar{r})(y - \bar{y})x^T \rangle \\ &= \langle (r(\bar{y}) + b^T(y - \bar{y}) - r(\bar{y}) - b^T(\bar{y} - \bar{y}))(y - \bar{y})x^T \rangle \\ &= \langle b^T(y - \bar{y})(y - \bar{y})x^T \rangle = \langle b^T(y - \bar{y})(y - \bar{y}) \rangle x^T \\ &= \langle (y - \bar{y})(y - \bar{y})^T b \rangle x^T = \langle \lambda_y \lambda_y^T \rangle b x^T = C_y b x^T\end{aligned}$$

This indicates that when the output firing rate covariance C_y is proportional to the matrix of firing rate slopes D_h, the neuromodulated Hebbian update is equal to the gradient. This specific conclusion is the same as that of Williams' REINFORCE paper. More interesting, however, is the general reliance on C_y, which was not calculated in that work (or in the node perturbation paper). In particular, for $D_h \approx I$, the Hebbian algorithm implements a transform of the gradient filter g by the approximately PSD matrix $C_y D_h^{-1}$. This warps the gradient along different dimensions, and in the case where $C_y D_h^{-1}$ is not full rank, it projects those dimensions out. As such this noise controls the inner product between different tasks' gradient filters g and the their directional response changes $C_y b$.

The firing rate covariance can also be rewritten in terms of the covariance of summed input responses, giving an interpretation of the noise as arising upstream of action potential generation:

$$\begin{aligned}\langle \Delta W \rangle &= \langle (h \odot \lambda_z)(h \odot \lambda_z)^T \rangle b x^T \\ &= D_h \langle \lambda_z \lambda_z^T \rangle D_h b x^T \\ &= D_h C_z D_h b x^T\end{aligned}$$

This indicates how somato-dendritic integration variability, which could easily be enacted by GABAergic shaping inputs on perisomatic regions independently of gain control, could yield similar results.

Finally, consider dependence on the output of the network. The gradient of reward with respect to a layer of rates y_i is of course the gradient with respect to y_n (the output) times the Jacobian $\frac{dy_n}{dy_i}$:

$$\frac{dr}{dy_i} = \frac{dr}{dy_n}\frac{dy_n}{dy_i}$$

The Jacobian is easy to compute by reference to the network architecture. We can introduce effective weights Ω^j for each layer j in the network to take the local slope of the f-I curves into account and write:

$$\begin{aligned}y_n &= f \circ W_n \circ f \circ W_{n-1} ... f \circ W_{i+1} y_i \\ \frac{dy_n}{dy_i} &= D_h^n W_n D_h^{n-1} W_{n-1} ... D_h^{i+1} W_{i+1} \\ \frac{dy_n}{dy_i} &= \Omega^n \Omega^{n-1} ... \Omega^{i+1} \equiv \Omega^{(n,i)}\end{aligned}$$

Here we have defined $\Omega^{(n,i)}$ as a set of weights from y_i to y_n in the local linearization. Since reward is a function of these outputs, there is some locally optimal direction in which those outputs should change, defining a locally optimal target activity. We denote this with a tilde, as $\tilde{y}_n$. This in turn defines a local target prediction error, $(\tilde{y}_n - y_n)$ which we label δ_n. We then observe that:

$$\frac{dr}{dy_i} = \frac{dr}{dy_n}\frac{dy_n}{dy_i} = (\tilde{y}_n - y_n)^T \Omega^{(n,i)} = \delta_n^T \Omega^{(n,i)}$$

This shows the intuitive fact that the locally-optimal target prediction error for the outputs can be "pulled backward" through the local linear approximation to the network to give a locally optimal target prediction error for the activities in y_i.. This can be used to rewrite the gradient step as a target-prediction-error minimization step, telling us what the relation between RL and the locally optimal supervised learning case is; RPEs are transmitting the only thing about the downstream weights that matters, namely their aggregate effect. Averaging RPE-weighted $(y_i - \bar{y}_i)$ values converts reward prediction error into a local activity prediction error. Notably, this solves both the non-locality problem of gradient updates and the low resolution of scalar signals with a particular space-vs-time trade-off, by swapping storage of weight information for its accumulation over time.

The activity-error relationship is now something we can substitute into the results above, to get a complete picture of how the neuromodulated Hebbian updates and the gradients work, respectively. This gives:

$$\langle \Delta W \rangle = C_y \Omega^{(n,i)T} \delta_n x^T \quad \text{and} \quad \frac{dr}{dW} = D_h^i \Omega^{(n,i)T} \delta_n x^T$$

The weight update can also be pre-multiplied with a covariance matrix, and the result interpreted in terms of noise, to yield projections of both the input and output filters.

Specializing to a bandit task, the expected reward is defined by the probability of choosing each action, weighted by the probable (expected) reward from that action. If p is the vector of reward probabilities, σ denotes the softmax function determining probabilities of each action, and y_n is the output layer of the network, then:

$$\langle r \rangle = p^T \sigma(y_n)$$

The gradient output filter can be computed for weights into layer y_i as:

$$\frac{d\langle r \rangle}{dy_i} = \frac{d\langle r \rangle}{d\sigma}\frac{d\sigma}{dy_n}\frac{dy_n}{dy_i} = p^T(\beta D_\sigma - \beta \sigma \sigma^T)\Omega^{(n,i)}$$

Therefore, the policy gradient for a given context is:

$$\frac{d\langle r\rangle}{dW} = \beta D_h \Omega^{(n,i)T}(D_\sigma - \sigma\sigma^T)px^T$$

The three three layer case we consider has $\Omega^{(n,i)} = W_r$, where W_r denotes the set of task readouts. ReLu units have D_h matrices with ones or zeros on the diagonal.

PR and RF projections

Simulation 1

In simulation 1, we project receptive fields by updating noise covariance matrices online. That is, the network starts with an input noise covariance C_i for each context (input) i. During training on input j, each C_k is updated as $C_k(t) = C_k(t-1)(I - u_j u_j^T)$, where u_j is the vector $C_k x_j$ scaled to have unit norm. C_k therefore becomes as a product of orthogonal projection matrices, and as a result, orthogonal noise dimensions are repeatedly removed. Input filters for future tasks are therefore orthogonalized against input filter plasticity for the current task. In realistic scenarios, we would expect there to be some decay-time for these projections, such that the matrices C gradually relax back to the identity, but that is not necessary for our simulations.

Simulation 3

Projections in simulation 2 are pre-computed based on readout weights. For each task i, we stack the readout weights for all other tasks into a matrix, then find a basis for the row-space, V, and a basis for the kernel, U, using the QR decomposition (i.e. using columns of Q). We set the covariance C_i as

$$C_i = [U, V]D[U, V]^T$$

where $[U, V]$ is the horizontal concatenation of U and V, and D is a diagonal matrix with one minus the anisotropy parameter p (i.e. $1 - p$) in rows corresponding to V. As such, when $p = 0$, C_i is the identity matrix, and when $p = 1$, C_i is a projection matrix equivalent to $I - VV^T$ which removes activity in the row-spaces of W_r matrices for other tasks. As a result, $C_i g_j$ will be in the kernel of the ith task's readout matrix, $W_r^{(i)}$, for arbitrary g_j, when $p = 1$.

Simulation 3

For simulation 3, we generated orthogonal bases A and B for the input and hidden layers by randomly sampling multivariate normal distributions and applying the Gram-Schmidt procedure to orthogonalize them. We regard the columns of A as feature vectors $f_1 ... f_n$ and the columns of B as feature vectors $g_1 ... g_n$. We generated compositional input stimuli by circularly shifting and accumulating Euclidean basis vectors e_i into vectors v_i according to the composition parameter C, then applying the basis trasnformation A. We did the same for outputs. For example, if stimuli are denoted s_i, then in the $n = 3$, $C = 2$ case:

$$s_1 = A(1, 1, 0)^T, s_2 = A(0, 1, 1)^T, s_3 = A(1, 0, 1)^T$$

Juxtaposition here represents matrix multiplication, and parenthesis denote vectors rather than indexing. Matching the inputs and outputs then indicates that the target weight matrix $W_h* = BA^T$, because stimuli must be transformed by A^{-1}, which is equal to A^T, into the feature basis for the input, then transformed into the euclidean basis for the hidden layer.

Linking numbers were free parameters that we used to generate projectors for input and output components of gradients. The input layer projector, which specified was constructed as $P = I - (AI(:, [i]))(AI(:, [i]))^T$, where I was the identity matrix and $[i]$ was the set of all indices to remove. For linking number one, $[i]$ would be every index other than that associated with the current feature under consideration. This matrix corresponds to $(p_i p_i^T)$ above. A matrix for the outputs, Q was constructed and applied similarly.

Simulations 4 and 5

Both of these simulations also used pre-computed gradient transforms for simplicity. In the receptive field stimulation, updates $\Delta W \propto gd^T$ took $d = x_i + x_{i+1}$ for paired inputs x_i and x_{i+1} to be associated with the same targets. This corresponds to projecting x_i or x_{i+1} onto their sum and renormalizing the result. Analogously, each task's correct readout in simulation 5 had an associate in another task, and population response filters were projected onto the dimension corresponding to the sum of the matched filters for the pair.

Taxonomy of interference categories

For a task's solution to be reachable by directed exploration along yd^T for some y, certain conditions must be met, which induce a taxonomy of interference categories. We define a set of tasks to have "essential interference" if there is no choice of subspaces along which to sample to avoid interference. We say a set of tasks has "inessential interference" if the task set does not have essential interference but the gradients themselves do interfere. And we call a set "non-interfering" if the gradients are all orthogonal. By our definition, unsolvable task sets exhibit essential interference, because minimizing error in one input-output pair implies increasing error in another. There are solvable tasks with essential interference, however, as illustrated (along with other cases) in figure 8.

To be solvable, all component tasks' solution manifolds must intersect. In the linear case these manifolds are "flats" or submanifolds of zero curvature. This is convenient analytically, and it permits extrapolating local gradient information into global information, but nonlinear networks with smooth solution manifolds are subject to the same qualitative considerations discussed here, because intersection relations are topological in nature. In either case, solution manifolds are induced by (effective) readout weight kernels. Several input-output pairs then have inessential interference when each item's solution space intersects with the intersection of all other items' readout weight kernels. Weights can then be changed to improve performance while moving within the other readouts' kernels. Denoting solution manifolds for each task S_i, these conditions are:

$$S_i = y_h^{i*} + \ker(W_r^i)$$
$$S_i \cap (\cap_j \ker(W_r^j)) \neq \emptyset$$

The first is the definition of a solution manifold; the acceptable output associated with input i, denoted S_i, comprises any activation of the hidden layer y_h^{i*} producing zero error, plus (infinitesimally) any vector from the set which current readouts W_r^i ignore. The second condition states that this solution manifold must intersect the kernels of the other tasks' readout weights; then it can be found by moving in a direction that doesn't interact with those tasks.

Inessential interference is illustrated in figure 8 panel C. Solution manifolds and readout weight kernels are shown, along with a path (green arrows) which moves towards each task's solution space within the other task's kernel. Performing gradient descent for one task would result in motion directly towards that task's solution space. This would not lie in the kernel of the other, impacting performance on the second task. Panels A and B show violations of the second condition above, meaning both exhibit essential interference.

Note that interference depends on the current weight configuration of a network. When readouts are one dimensional, for example, each solution manifold bisects the weight space. This partitions it into regions where the sign of interference between two tasks is constant. Actually observed interference is also a property of task order. The gradient of reward is always pointing towards some current solution flat, which may not be the closest one. When network weights move from one cell of the partitioned space into another, by virtue of following a weight update towards a non-proximal solution manifold, the gradient for the just-crossed solution manifold changes sign. Constructive interference with this second task then becomes destructive. Figure 8B illustrates this.

In task sets with inessential interference, noise can sample dimensions locally which are in the kernels of other tasks' readouts. This can be implemented online by removing dimensions from a networks' noise variability which have been seen in other tasks, at the point of a task switch. This noise variance decrement accumulation mirrors the gradient estimate accumulation inherent in reinforce like rules, is memory efficient, and is biologically plausible. Additionally, it accords with the facts that real neural responses to novel stimuli show decreasing variance over time, and that networks of interneurons are well positioned to tune noise at the network level via their high interconnectivity.

Linear solution manifold geometry example

To determine when linear networks have different essential and inessential interference conditions, one can inspect the intersection properties of generic flats in Euclidean space. The flats we consider are the null spaces of the readout weight matrices, which are the spans of all vectors that a given readout matrix sends to zero (i.e. "doesn't care about"). Each of these kernels can be written in terms of a basis, and the intersection of kernels can therefore be written in terms of basis relationships.

Let n denote the number of units in a network's hidden layer. Then inputs to the network's readout weights have dimension n, and the basis for a given readout's null space is given in terms of vectors of dimension n. Let $\{v_1^i, v_2^i, ...v_j^i\}$ denote a set of basis vectors for each W_r^i which are organized as columns in a matrix V^i, so that the dimensions of V^j are $(n \times j)$. Consider j to be arbitrary, meaning variable over the sets V^i, and the vector

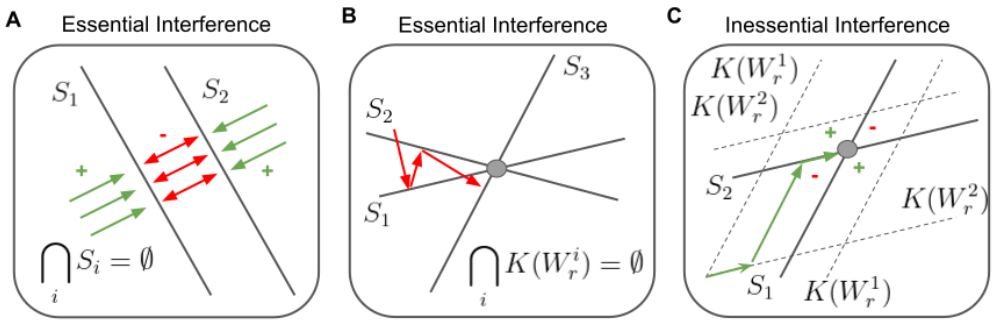

Figure 8: Interference cases. (A) Tasks which aren't simultaneously solvable interfere in regions of the solution space that lie between their solution manifolds. Red arrows denote opposite directions of updates in this region of the plane based on which task is being learned at a given time. Green arrows denote constructive interference for regions of space in which both solution manifolds lie in directions with an inner angle up to 90 degrees. (B) In some situations, updates cannot move towards one solution manifold while moving parallel to the others. The red arrows are an example of gradient descent moving the weights towards solution manifold S_1, then back towards S_2, and finally towards S_3. This illustrates how order matters, as well as how interference is destructive in regions where angles between solution manifolds are less than 90 degrees. (C) Inessential interference occurs when there is a dimension in the intersection of relevant kernels along which weights can move via directional derivatives while solving the current task. In this example, a directional derivative algorithm can follow $K(W_r^2)$ towards S_1, then $K(W_r^1)$ towards S_2 during consecutive training epochs in order to avoid interference. The green arrows illustrate such a trajectory. (A,C) For any pair of solution manifolds, there is a partition of the weight space into regions where gradients are destructive (solution space angles less than 90 degrees), constructive (solution space angles more than 90 degrees), or neither. These regions are denoted by red minus signs and green plus signs here, respectively.

$m = [m_1, m_2, ...m_n]$ to count the number of readout kernels of each dimension, starting with 1 and ending with n. Then if we have a 3 dimensional hidden layer with two readouts, one of which has a 1 dimensional null space and the second of which as a 3 dimensional null space (i.e. is the zero matrix), the vector $m = [1, 0, 1]$.

Now note that if there exists an intersection of these kernels, it has a coordinate in each basis. Denote the coordinate vector for basis i by a_i, and let $l = \text{sum}(m)$. Then the existence of the intersection is equivalent to the claim that we can represent it in each basis, i.e.:

$$V^1 a_1 = V^2 a_2 = ... = V^l a_l$$

Each equality between vectors here is an equality between n unknowns (the coordinates), so that there are $(-1 + \sum_k m_k)n$ equality constraints. On the other hand, there are $\sum_k m_k k$ unknown coordinates, because every kernel of dimension k contributes k of them. An underdetermined system of linear equations admits a set of solutions with as many degrees of freedoms as there are unconstrained free variables. Hence, the set of equations above admits a solution which can be parameterized by n_f such that:

$$n_f = \sum_k m_k k - (-1 + \sum_k m_k)n$$

When n_f is greater than or equal to zero, such a solution generically exists. When n_f is less than zero, a solution may still exist but is not generic, meaning any small displacement of a subspace will remove it. For example, $m = [3, 0]$ is a system of three lines in the plane. Random choices of these lines will not intersect, and indeed $n_f = 3 - 2 \times 2 = -1$. On the other hand if $m = [2, 0]$ we have a random pair of lines, which will generally intersect in a point, which is a subspace with $n_f = 0$ degrees of freedom.

Step-size normalization

The reward-modulated Hebbian rules used here are made inter-comparable with gradient updates by fixing step sizes. In particular, projective operations change learning-rates, whereas our interest here is in isolating the geometric aspects of weight updates. To do so, we have generally ignored both the substantial improvement in sampling complexity associated with lower-dimensional search, and the thorny and substantial issues related to setting learning rates. For example, real weight updates are unlikely to be fixed in magnitude, but it is not clear what sets update

sizes biologically. Moreover, the idea that gradients are a reasonable point of comparison is inherently speculative as well, and biologically, weight updates are unlikely to depend quadratically on noise as e.g. REINFORCE or simple Hebbian algorithms would dictate, because quadratic dependencies produces very poorly behaved dynamical systems. As such, naive Hebbian algorithms are also of limited biological realism. Both the CEM and GD are therefore expected to deviate from the metric properties biological weight changes. We leave resolving these sorts of questions to future work.

It is also important to note that there is no theory-free inter-comparison to be made between gradients and projective updates in our compositional learning simulation (simulation 3), because different matrix norms yield different step sizes for projective update matrices of rank greater than one. The gradient update is always rank one, because it is an outer product, whereas the projective update is generally higher rank because it is a sum of such products. Two candidate matrix norms for use with the projective algorithm are the Frobenious norm and the max norm. The two norms agree on rank one updates, and therefore agree on the gradient update. The max norm is potentially appropriate because it can be used to bound every rank-1 feature-pair's update to be at most equal in step size to the gradient update. On the other hand, the Frobenious norm is also potentially appropriate because it splits the total step size across feature-pair terms such that their sum of squares is equal to the length of the gradient update. The degree to which synaptic update magnitudes interact across e.g. a cortical column is not known with enough precision to adjudicate between these options. The max norm seems more likely to be descriptive of spatially well-separated pairs of neurons, whereas the Frobenious norm may be more likely to describe single neurons, or those with tight local inhibitory coupling. Nonetheless, the use of the Frobenious norm is conservative, whereas the max norm is "permissive" or "optimistic". In simulation 3, we display results using the max norm, because they don't induce a transient early non-monotonicity in the performance curve, which requires a detailed unpacking of matrix norms to explain.

Oracle versus sample-based quantities

To demonstrate the advantages of the CEM, we used algorithmically computed gradients in the main text. These are appropriate because they separate geometric impacts on learning from sample complexity differences associated with estimating derivatives in spaces of different dimension. Because real learning scenarios will not necessarily involve access to precomputed gradients, we verified the consistency of our results in simulations with sample-based ones.

Specifically, we ran simulations in which we replaced trials with loops of "sub-trials" which accumulated reward-modulated Hebbian weight updates. In theory, such "sub-trials" could be promoted to "trials", accumulated with a causal kernel such as an exponential moving average without qualitatively changing our results. We chose the present implementation because it removes complications such as overlapping integration time constants. We found, as expected, that we were able to reproduce gradient based learning curves using the appropriate isotropic noise based gradient-estimators, and likewise for projective update curves and estimators.

To further verify the interchangeable character of our gradient oracle and Hebbian sample-based weight updates in simulation, we inspected their relationships on a trial-by-trial basis, and we computed the difference in cumulative error at simulation end for an ensemble of 1000 random tasks in a pared-down version of simulation 2. Average differences between the oracle algorithm and the sample based algorithm were $0.5\% \pm 0.007\%$ (mean $\pm$ SEM) of cumulative error for gradient filter accumulation, and $0.5\% \pm 0.004\%$ (mean $\pm$ SEM) for projective filter accumulation, using an accumulation loop of 1000 "sub-trials". These are shown along with other diagnostics below. In future work, we intend to provide more detailed examinations of the sample complexity and learning rate based trade-offs in relation to gradient descent which we have bracketed here.

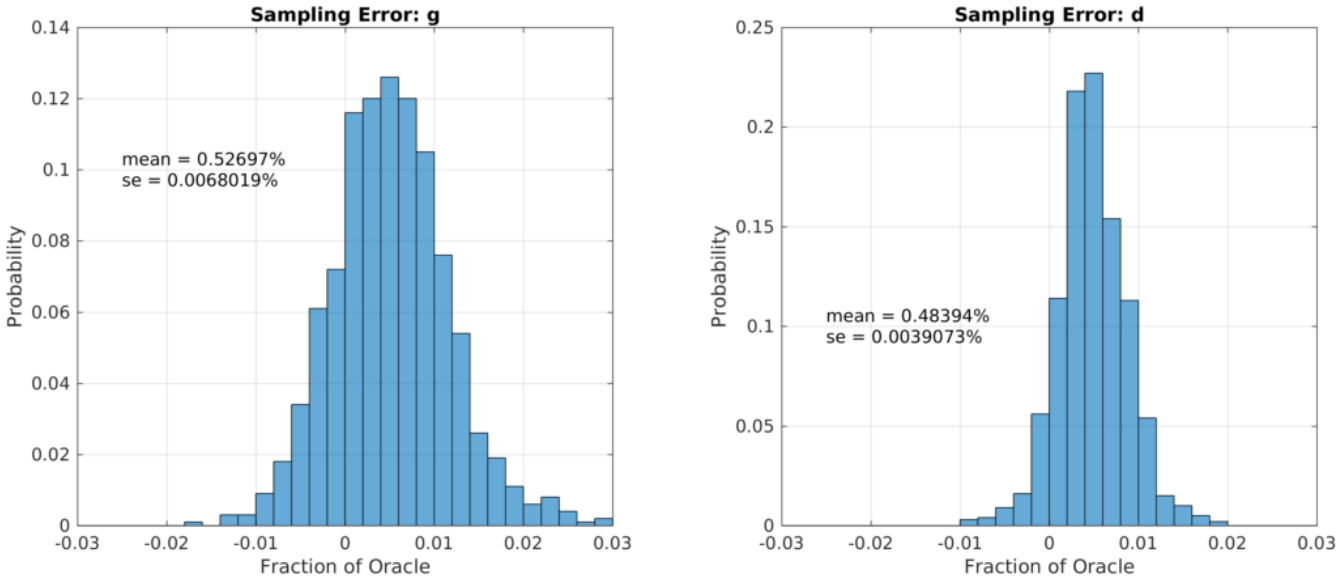

Figure 9: Sample-based cumulative simulation error relative to oracle cumulative simulation error for gradient (g) and projective (d) output filter construction. In general, sampling error is lesser for the CEM than for the full REINFORCE estimate, given a fixed sample count, because the dimensionality is lower. Here we used 1k samples per output filter estimate.

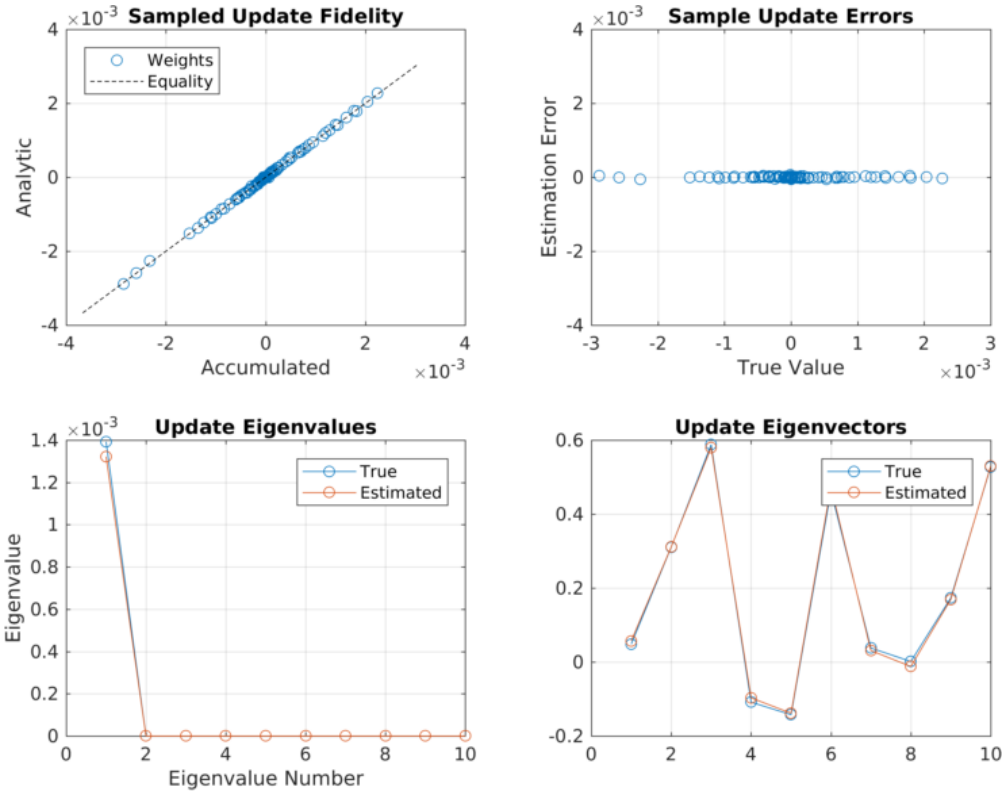

Figure 10: Diagnostics comparing a random weight update based on oracle computation with the same update based on a Hebbian sample accumulation, for a lower-dimensional version of simulation 2. Because the input filter does not change in this simulation, one only expects the output filter to be distorted by sampling. We computed and inspected the eigendecomposition of the weights, because the output filters are not accumulated independently from the (static) input filters. As can be seen above, typical errors were negligible, the output filters were closely aligned, and the eigenvalues were very generally very close as well. The exception to this occurred where gradients were effectively zero, and alignment between sampled and oracle computations diverged. These accounted for a significant fraction of the total number of updates but a negligible fraction of total weight change (because once the network has converged, every update is minimal).

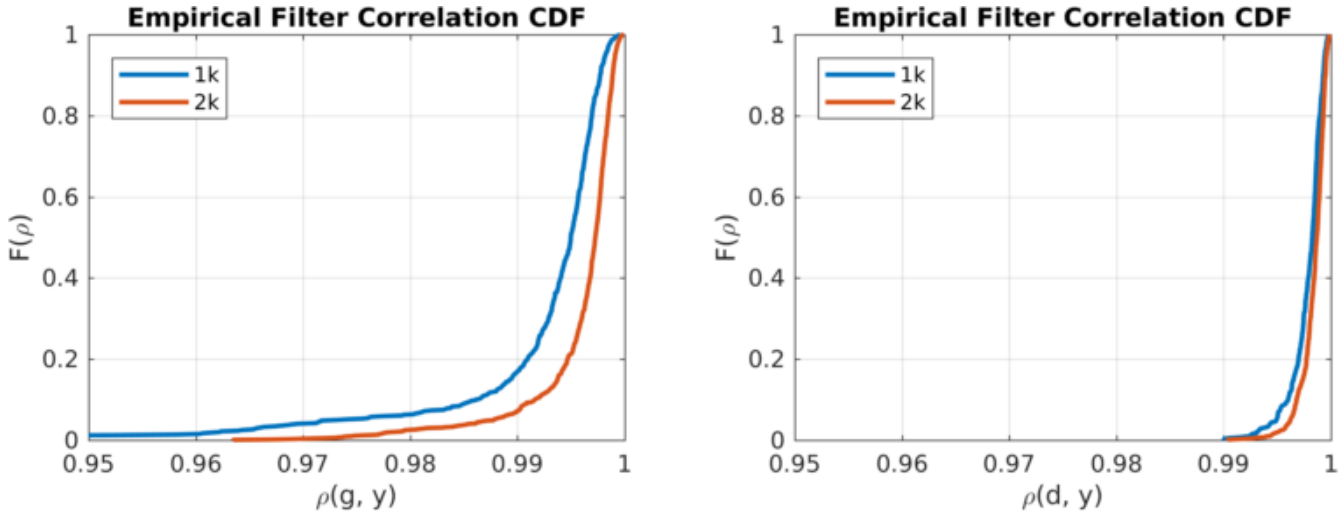

Figure 11: Empirical filter correlation CDFs comparing sample and oracle based output filters, for the weights contributing to 95% of the total weight change. The least important 5% are excluded because they are essentially zero, and are therefore relatively unconstrained in addition to being unimportant. Left panel is for gradient output filters g, and right panel is for projective output filters d. The improved sample complexity of the lower dimensional filters is apparent in the difference between these plots. Lines indicate simulations with 1k and 2k block-accumulation sub-trial loops, for comparison.