Beyond gradients: Noise correlations control Hebbian plasticity to shape credit assignment

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

[1,2]Department of Cognitive, Linguistic, and Psychological Sciences, Brown University, Providence, Rhode Island, U.S.A.
[1,2]Carney Institute for Brain Sciences, Brown University, Providence, Rhode Island, U.S.A.
[*]Corresponding Author

Updated: November 20, 2021

Abstract

Two key problems that span biological and industrial neural network research are how networks can be trained to generalize well and to minimize destructive interference between tasks. Both hinge on credit assignment, the targeting of specific network weights for change. In artificial networks, credit assignment is typically governed by gradient descent. Biological learning is thus often analyzed as a means to approximate gradients. We take the complementary perspective that biological learning rules likely confer advantages when they aren't gradient approximations. Further, we hypothesized that noise correlations, often considered detrimental, could usefully shape this learning. Indeed, we show that noise and three-factor plasticity interact to compute directional derivatives of reward, which can improve generalization, robustness to interference, and multi-task learning. This interaction also provides a method for routing learning quasi-independently of activity and connectivity, and demonstrates how biologically inspired inductive biases can be fruitfully embedded in learning algorithms.

Keywords: Neural network, policy gradient, credit assignment, Hebbian plasticity, three factor rule, neuromodulation, noise correlations, generalization, interference

Contents

1 Introduction

The success of gradient descent (GD) in neural network training has made it a standard across research settings. Much work has accordingly asked how biological learning might either approximate GD or recapitulate it's properties (e.g. Williams 1992; O'Reilly 1996; Xie and Seung 2004; Fiete and Seung 2006; Guerguiev et al. 2017; Zenke and Ganguli 2018; Bellec et al. 2019; Moldwin et al. 2021). Approximating gradients may not always enable the most desirable forms of plasticity however, and systematic differences between biological learning rules and gradients may perform important functions (e.g., O'reilly 2001, Vasilaki et al. 2009). It is thus also plausible that incorporating aspects of biological learning into artificial neural networks will both improve network performance and advance our understanding of neural computation (e.g. Schrimpf et al. 2018; Linsley et al. 2020; Jaskir and Michael J. Frank 2021).

One setting where we might expect biological rules to outperform gradient

descent is multi-task learning; GD often produces contradictory weight changes between tasks, which can result in the reduction of performance on one while learning another (McCloskey and Cohen 1989; Ratcliff 1990; Flesch, Balaguer, et al. 2018). By contrast, animals frequently learn without losing their previous knowledge or ability. Standard workarounds for this problem with gradients include interleaving training of different tasks or applying regularizing procedures to weight changes. Examples of the latter include inducing sparsity (Srivastava et al. 2014), freezing previous learning (Kirkpatrick et al. 2017) or shaping gradients through initial connectivity (Flesch, Juechems, et al. 2021). In computational neuroscience, however, multi-task learning is often addressed with specialized architecture, by mimicking hippocampal pattern separation (Mcclelland et al. 1995) or by gating activity in the prefrontal cortex, for example (Rougier et al. 2005; Collins and Michael J. Frank 2013).

Such considerations prompted us to analyze biological learning rules under conditions in which they would not approximate gradients. We considered reward-modulated Hebbian learning rules, specifically, for several reasons. First, they are known to perform GD in particular circumstances (Williams 1992; Fiete and Seung 2006; Fremaux et al. 2010; Frémaux, Sprekeler, et al. 2013; Frémaux and Gerstner 2016). Next, they fall into a class of empirically well established forms of plasticity (Bi and Poo 1998; Dan and Poo 2004; Seol et al. 2007; Ruan et al. 2014;). Third, when implemented in neural networks, such rules can recapitulate empirical data linking physiology to behavioral learning (Michael J. Frank 2005; Franklin and Michael J Frank 2015; Gurney et al. 2015). Finally, because Hebbian rules are activity-dependent, they are tied to another realm of ongoing investigation, that of so-called noise-correlations (Gawne and Richmond 1993; Shadlen and Newsome 1994; Zohary et al. 1994; Averbeck et al. 2006; Adam Kohn et al. 2016). As we show below, these observations can be unified, expanded upon, and applied to control basic properties of credit assignment, whereby noise correlations route and shape learning signals. This, in turn, can allow networks to avoid interference between learning episodes and to render representations more or less mutually available for learning.

Specifically, our results are as follows: (i) We provide a mathematical expansion of reward-modulated Hebbian plasticity, allowing us to decompose learning into gradient-like and unsupervised weight updates. (ii) We show how the gradient-like terms can be interpreted as directional derivatives of the network loss, indicating that network noise correlations can construct gradient projections. We then develop measures of task interference and a classification of interference categories, which we use to produce a generative model of tasks with avoidable interference. We show that directional derivative weight updates (stemming from reward-modulated plasticity with adapting noise) can solve these tasks more efficiently than (vanilla) gradient descent. (iii) As another application, we ask how information encoded in existing network weights via prior learning might be used constructively in new tasks. We find that feedforward noise can be used to prioritize worthwhile search dimensions in a network's weight space, mitigating the tendency of gradients to de-specialize representations. (iv) Finally, we show that statistically dependent noise correlations

can modularize and gate learning signals, which allows for credit-assignment restrictions within and between feature groups. This "eligibility through dependence" mechanism can enhance learning according to priors about relevance, and provides a means of encoding equivalent biases in gradient algorithms.

2 Results

2.1 Three factor rules compute directional derivatives

We consider feed-forward networks $x_r = W_r W_{h_1} ... W_{h_n} W_i x$, where the W variables encode synaptic connectivity between layers. Here W_r corresponds to the "readout" from a final cortical layer, and the W_{h*} correspond to "hidden layers". Thus we are considering linear rate-coded point-neuron models without recurrence. Linearity provides tractability, a means to develop intuitions, and a locally valid description of nonlinear cases (e.g. Saxe et al. 2014). We consider task dependent readouts, which occur when processing is state- (e.g. motivation or attention) dependent. This makes the collection of networks a nonlinear system. Each multi-layer network is equivalent to a 3 layer network $x_r = W_r W_h W_i x$, where W_r, W_h, and W_i are relabeled to denote products of the original factors. We label this way because we will take derivatives with respect to W_h to compute a gradient of reward, and we presented the expanded form to emphasize that the choice of particular hidden layer is arbitrary.

For each stimulus, activity is applied trial-wise by $x(t) = \mu(t) + \xi(t)$ at the input. Here μ is a constant, ξ is mean zero noise, and t is the trial number, which we suppress below. Neurons in other layers j receive additional noise, and each layer is described by this equation (see figure 1A). The learning problem is to match readout activity to some target by to modifying the weights W_h. We consider reward r to be the negative mean squared error of an output, and we let δ_r denote the target prediction error used to compute this. That is, $\delta_r = \mu_r^* - \mu_r$, given the target activity μ_r^*. Subscripts (including r) denote the layers variables are associated with. The gradient of expected reward with respect to hidden weights (derived in an appendix) is then:

$$\frac{\partial \langle r \rangle}{\partial W_h} = 2 W_r^T \delta_r \mu_i^T - 2 W_r^T \langle \xi_r \xi_i^T \rangle \tag{1}$$

We compare the properties of weight updates given by this gradient to those driven by reward-modulated Hebbian plasticity:

$$\Delta W_h = \alpha (r - \langle r \rangle)(x_h - c_h \langle x_h \rangle)(x_i - c_i \langle x_i \rangle)^T \tag{2}$$

The first term of the Hebbian equation applies reward modulation, and the others comprise the Hebbian product of pre- and post-synaptic activity. These latter terms are computed relative to stimulus-specific homeostatic set-points proportional to their average expected firing rates. The c_h and c_I parameters control set-points use, and α is a learning rate. Such rules enjoy broad empirical support as models of synaptic learning; examples include dopaminergic

modulation of spike-time-dependent plasticity (Shen et al. 2008; Ruan et al. 2014; Frémaux and Gerstner 2016) and plasticity in basal ganglia models of behavioral reinforcement learning (Michael J. Frank 2005; Gurney et al. 2015; Franklin and Michael J Frank 2015).

Prior work has shown (2) is equivalent in expectation to (1) under specific conditions (Williams 1992), whereas the general case of (3) was not considered. We are interested in this general case, and we show below how the resulting updates function analytically. In later sections, we demonstrate the implications of our results via simulation. Additional details on our calculations can generally be found in our supplements.

Expanding (2) into terms, taking an expectation over trials, and defining $\beta = 1 - c$ for compactness allows us to write $\langle \Delta W_h \rangle$ as a sum over products:

$$\langle \Delta W_h \rangle = \alpha \sum_{jk} \langle a_j b_k \rangle \tag{3}$$

The terms a_j and b_k come from two sets, and the sum ranges over all pairs:

$$a_j \in \{2\delta_r^T \xi_r,\ -\xi_r^2,\ \langle \xi_r^2 \rangle\}$$
$$b_k \in \{\beta_h \beta_i \mu_h \mu_i^T,\ \beta_h \mu_h \xi_i^T,\ \beta_i \xi_h \mu_i^T,\ \xi_h \xi_i^T\}$$

These equations generate a number of interesting algorithms as special cases. For example, if we take $\beta_h = 0$, $\beta_i = 1$, and specify that the noise is Gaussian, then the sum is comprised of terms $\{a_1, a_2, a_3\} \times \{b_3, b_4\}$, which simplify to give:

$$\langle \Delta W_h \rangle = 2\alpha \langle \delta_r^T \xi_r \xi_h \mu_i^T \rangle - 2\alpha \langle \xi_r^2 \rangle \langle \xi_h \xi_i^T \rangle \tag{4}$$

This equation has a similar structure to (1), where the first term contains the output prediction error δ_r and the average input μ_i^T, and the second is determined by noise alone. It simplifies to the first term there when higher order noise components are neglected, input and endogenous output noise are zero, and the noise is full rank, isotropic, and uncorrelated. This special case was shown by other means in Williams 1992.

The noted special case follows a gradient because noise performs a sampling-based exploration of possible weight changes. Updates orthogonal to the gradient then cancel one another. This observation leads us to consider how structured noise can focus weight updates along useful dimensions. To do so, we decompose each layer's noise into components originating there (λ_n) and those originating in preceding layers (ϕ_{nk}):

$$\xi_n = \lambda_n + \sum_k \phi_{nk}$$

The weight update (3) then includes a term $\langle a_1(\lambda_h) b_3(\lambda_h) \rangle = \langle \delta_r^T \phi_{rh} \lambda_h \mu_i^T \rangle$, where we have further split the a_j and b_k according noise origin. If λ_h is isotropic in the subspace it spans, this term implements gradient descent in the subspace of the network weights given by $\{s_h \mu_i^T | s_h \in \text{Range}(\Sigma_{\lambda_h})\}$. That is, when λ_h

is not full rank noise, but does have equal variance in its nonzero dimensions, this term implements a projection of the gradient onto the noise subspace. The projective effect is graded if we relax the isotropic assumption; non-isotropic noise interpolates between sampling within a subspace and full-rank sampling, and is graded according to the eigenvalues of the noise covariance (figure 1C). Importantly, the term in question can comprise the majority of the weight update if we choose β and λ terms appropriately. We show below that this can be used to orthogonalize learning across tasks and avoid interference.

For concreteness, we refer the reader to figure 1, illustrating how terms we discuss are related. We define d to be the column-vector component of the Hebbian update (i.e. $\langle \delta_r^T \phi_{rh} \lambda_h \rangle$), which is (geometrically) a combination of the gradient g and the noise covariance matrix Σ_h. The direction d points is determined by taking g and "squeezing" it according to the covariance matrix, focusing updates on dimensions spanned by the noise. As shown in an appendix, for a network to effectively learn via these restricted Hebbian updates, it must be possible to move along d to the solution manifold of the task, S. If noise covariance lies in the kernel of some other task's readout, $K(W_r^2)$, then we avoid interference with that task. If networks retain knowledge of earlier tasks or have foreknowledge of later ones, noise variance can be shrunk along their gradients. This is reminiscent of Sanger's classic Hebbian work (Sanger 1989) in the sense that both orthogonalize a statistic of activity (average vs variance) in a way that's plausibly accomplished by inhibitory circuits.

While we've focused on the column spaces of synaptic weight matrices so far, similar arguments apply to sampling the input components of the weight update via the term $\langle a_1 b_2 \rangle = \langle \delta_r^T \phi_{ri} \mu_h \lambda_i^T \rangle$. Moreover, we can obtain projected-gradient algorithms with varying input and output filters via the term $\langle \delta_r^T \phi_{ri} \lambda_h \lambda_i^T \rangle$. In the latter, non-independence of noise across layers can isolate components of inputs and hidden representations that should be mutually "available" for learning. Top-down activation via prefrontal cortex could be used to do so, for example, focusing learning on specific representations' features (e.g., Michael J. Frank and Badre 2012; Niv et al. 2015; Stalnaker et al. 2016). Such features could be chosen to minimize interference between multiple modalities in associative cortex, even as full stimulus information is transmitted in firing rates. Before addressing these cases, we return to the simple learning rule with $\beta_h = 0$, $\beta_i = 1$, for which the input filters μ^T of the weight updates are independent of noise. These provide the flexibility to avoid certain types of interference in multi-task learning.

2.2 Interference arises from rank-one relationships

Interference is a property of sets of tasks whereby changing performance on one task leads to changing performance on another (McCloskey and Cohen 1989). To systematically evaluate the impact of weight updates on multi-task learning, we chose to quantify the interference introduced by plasticity on several scales. Thus we define "microscopic interference" to occur when the same individual weights are updated in response to multiple inputs. Changes can be of equal

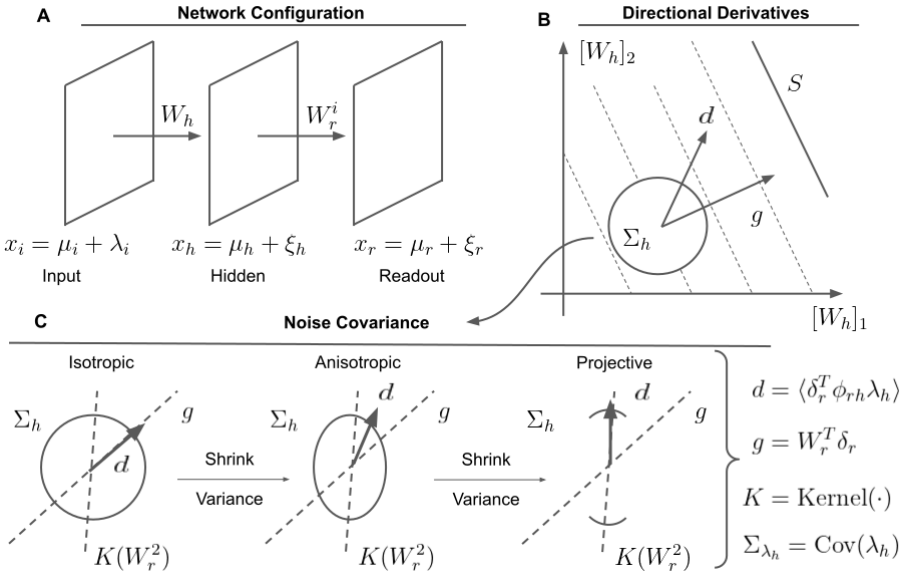

Figure 1: Noise covariance squeezes gradients along particular dimensions. (A) Network configuration. The networks we consider are composed of input, hidden, and readout layers. The x terms denote layer activities, the terms λ_i, ξ_h, and ξ_r are noise, and mean stimulus responses are denoted by μ terms. Layer activation is linearly transferred via weights W_h and W_r^i. (B) Contour plot of an example weight update scenario. A gradient of reward with respect to network weights, g, is contrasted with a directional derivative of reward, d. The gradient points directly to the problem's solution manifold S, but updates along d can also be used to push weights to S. The update direction depends on the covariance Σ_h of the noise λ_h. The update based on g would be used by gradient descent, whereas the update based on d is used by our reward-modulated Hebbian rules. (C) Elaboration on noise. When the covariance Σ_h is isotropic, the modulated Hebbian algorithms estimate gradients. When it is rank-deficient, updates are projections of the gradient onto the subspaces containing noise. Anisotropic noise interpolates these cases. When the noise covariance is relatively large in a subspace which doesn't impact another task (the kernel of the readout matrix for that task, $K(W_r^i)$), interference between tasks is diminished.

sign (constructive) or antagonistic (destructive). We define "macroscopic interference" to occur when the directions of two sets of weight updates, considered as vectors, make an angle different from 90 degrees. Macroscopic destructive interference occurs when this angle is obtuse and macroscopic constructive interference occurs when it is acute. We track interference in our simulations below, along with response errors, which are cumulative indicators thereof.

To formalize interference, we need to elaborate our notion of a task. We define a task $\mathcal{T}^i$ to be a set of inputs X^i and associated target outputs X^{*i}, along with a readout prescription W_r^i. That is, as a tuple (X^i, X^{i*}, W_r^i). For more considerations regarding tasks, see our supplements on interference and solution manifolds. We define a macroscopic interference matrix M as:

$$M \equiv [M_{jk}] \equiv [(U_j, U_k)_F] = [\mathrm{vec}(U_j)^T \mathrm{vec}(U_k)] \tag{5}$$

The notation $(\cdot,\cdot)_F$ denotes the sum of the elements in the entry-wise product of two matrices. The $\mathrm{vec}(\cdot)$ operation denotes vectorization. Subscripts j and k denote arbitrary U matrices, and we have refrained from writing explicit functional dependencies throughout.

This allows us to show that interference between weight updates arises from overlapping input filters, overlapping output filters, or both, i.e., rank-one relationships; since our weight updates are outer products, we can rewrite (5). Let z denote the output filter for a given update U (meaning $z = W_r^T \delta_r$ or $z = \langle \delta_r^T \xi_r \xi_h \rangle$) and let u be the input filter (meaning $u = \mu$ for the $\beta_h = 0$, $\beta_i = 1$ case). We find:

$$M = [(U_j, U_k)_F] = [\mathrm{tr}(U_j^T U_k)] = [z_j^T z_k u_j^T u_k] \tag{6}$$

The first source of interference is thus input filter overlap ($u_j^T u_k$), and the second is overlap between target output errors ($z_j^T z_k$). Intuitively, interference only occurs if weight updates modify connections from a shared set of inputs or to a shared set of outputs, and it depends on the angles of these updates. Equation (6) shows that, to mitigate interference, one of these pairs of vectors must be orthogonalized. Since noise subspaces determine z_j, z_k, u_j and u_k, this can often be done. When certain task set characteristics are known a-priori, e.g. that the action of turning on a light switch is insensitive to elbow angle, then such knowledge can be used. If foreknowledge is not available, noise subspaces can be adapted online during task exposure in an iterative way, mirroring the gradient accumulation procedure (elaborated in the supplement on interference).

2.3 Noise tuning can avoid interference

To illustrate our results, we generated and solved random task sets with inessential (avoidable) interference. This required developing an interference taxonomy and determining conditions for producing avoidable interference, then developing a generative model of task sets to meet those conditions. These points are addressed in appendices.

Specifically, we simulated ensembles composed of 1000 random networks with eight tasks comprising one input-output pair each. Hidden and input layers were 10 dimensional and readouts scalar. Our task-set choices yield simple situations for which it is straightforward to prove the various interference properties and develop intuitions, but more complicated linear cases are similar. Trial blocks were ordered in an arbitrary, predefined sequence, with each task appearing 10-15 times. These simulations allowed us to compare training with gradient descent against training with the projective algorithm described above. For demonstration, we used fixed update step sizes and oracle noise covariance matrices, although we verified that online schemes estimating covariance during learning produce similar results (see supplement). Similarly, we used algorithmic gradients and projections, rather than sampled ones, yielding a conservative estimate of improvement based on geometry alone, since sample complexity improves with noise anisotropy. Again, we verified that sample-based weight updates produce very similar results. Both algorithmic simplifications serve to remove subtle dependencies between parameters while leaving clear the geometric nature of our claims.

High accuracy, meaning an MSE for each task remaining under 0.01, was generally reached within 2000 trials, given a learning rate of 0.01 and 25 trials per epoch. Mean cumulative error, calculated at every trial for all tasks, was reduced by an average of 25% relative to gradient descent when using projective updates (figure 2, panels A-D). Example learning curves and error trajectories for a smaller network can be seen in figure 2, panels (i-iii). The qualitative features of each task's learning curves are the same, but error increments during the other tasks' training are smaller by a factor of 5. This occurs because we used an anisotropy parameter $P = 0.8$, meaning that weight updates were 4 parts projective and 1 part gradient.

The generality of our approach is illustrated by performance surfaces computed across different parameter sets. The first performance surface explores the impact of P, anisotropy strength, on relative error (figure 2C), and hence the improvement arising from the use of directional derivatives. Increasing P from zero improves performance linearly until a value of approximately 0.8, after which performance improvements begin to reverse. The reversal arises from our fixed step-size; a purely projective algorithm must sometimes move at a high angle to the gradient, inducing a performance trade off between interference avoidance and distance moved.

Performance improvement systematically grows when baseline interference is increased by manipulating various factors (figure 2 D-F). One such factor is input overlap; improvement over GD increases from 20% to 60% as overlap becomes total. Another factor is the number of tasks performed relative to the number of dimensions in hidden and input layers. When the former is low relative to the latter, target outputs are few and high dimensional, hence roughly orthogonal. As the number of tasks increases, performance improvement grows. It is maximal when every dimension is used by some readout, because only one dimension is then "free" to solve each task. Gradients never point in this dimension, whereas the projective updates do by construction. A final factor

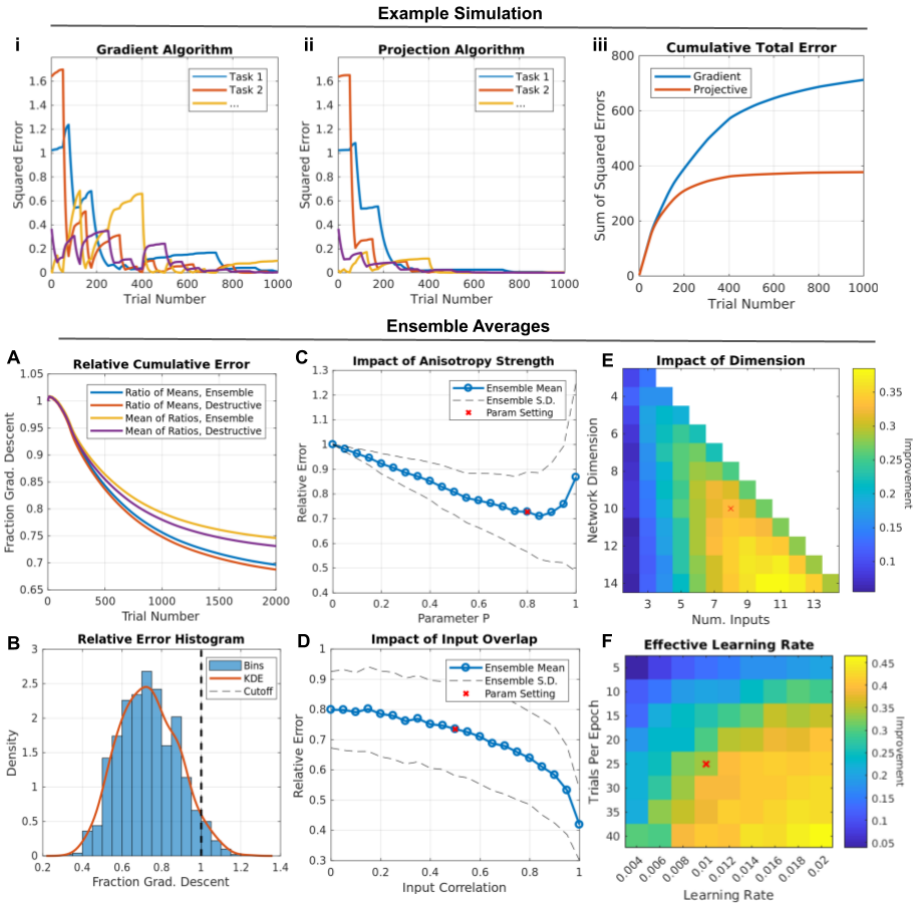

Figure 2: Performance improvement relative to gradient descent in randomly generated networks with inessential macroscopic interference. (i - iii) Example learning curves and cumulative errors. (A) Ensemble mean (across random task configurations) cumulative error curves for gradient descent and directional derivative approaches. (B) Histogram and KDE of relative performance across tasks. Most random task sets have inherently destructive interference, mitigated by projective updates. (C) Relative performance as a function of anisotropic noise strength P. Red indicates the value used in other panels (P=0.8), which is near optimal for the given number of dimensions and inputs. (D,E,F) Relative improvement of the projective method increases as a function of destructive interference induced by input pattern overlap (D), number of inputs relative to network size/capacity (E), and amount of training on any single task before switching (F).

combines learning rates and repetitions per epoch; their product defines an effective number of trials per epoch. When low, the tasks are effectively interleaved, and improvement is negligible. When high, gradient updates produce large conflicting weight changes, which are mitigated by directional updates.

2.4 Propagating noise reduces de-specialization

Inessential interference is a general phenomenon, which doesn't require any particular interpretation of solution manifold geometries. In ecological scenarios, interpretations are often natural however. Tasks frequently have structure such as hierarchy, which we may expect to re-use neural representations. For example, oranges are simultaneously in the class of oranges, the class of citruses, and the class of fruits. To respond "yes" when asked if an orange is a fruit requires an "or" over class elements, because one should also reply "yes" when asked about an apple. But disjunctions propagate gradients along all backward paths from a target output through hidden layers in a network, even when those paths intersect representations which should remain separate. In the fruit example, if one suspects an apple is a fruit, and similarly regards oranges, then one can believe apples are fruits more strongly by increasing one's belief that an apple is a citrus. Gradient rules drive logically equivalent weight changes, such that GD tends to de-specialize representations.

Specifically, under gradient descent, all connections are eligible for change at all times. One way to constrain the set of weight updates is to change the nature of the network noise. Up until now, we have considered noise in the hidden layer independently of noise in earlier layers. However, one important potential application of our directional derivative arguments occurs when updates take their direction from existing feed-forward weights. In particular, the network weights W_h provide a natural set of directional updates to consider based on ϕ_{hi}, the noise in the hidden layer that is carried forward from the inputs. The same formulas for d and D can be considered with ϕ_{hi} taking the place of λ_h, and can be used to avoid the de-specialization just described.

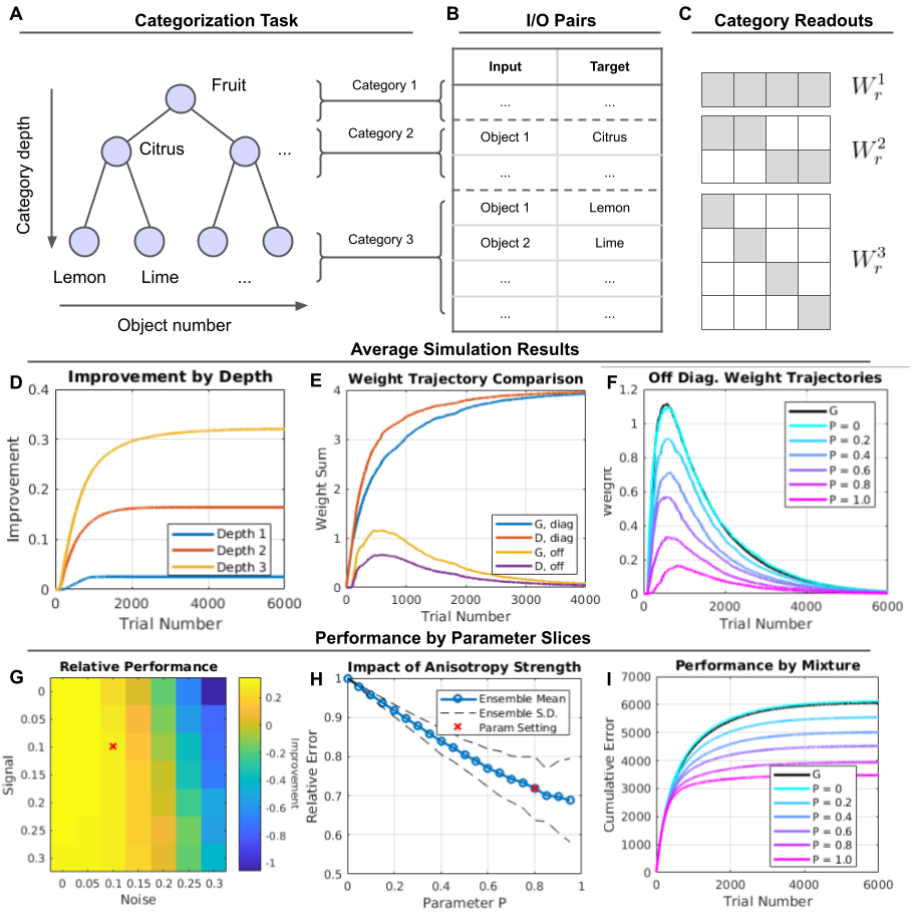

Figure 3: Avoiding de-specialization. Task gradients in sets with hierarchical structure degrade previously segregated representations. This is avoided when plasticity follows a Hebbian rule with forward-propagating noise. (A) Example categorization task. The lowest level categories are individual items, and categories form a binary tree. (B) Each categorization resolution defines a task for which the readout matrix is constant. Task 3 can be thought of as "identify this object", specified by the tuple (object 1, lemon, I_4). The readout matrix is the identity here because each object is its own object-level category. (C) Readout weight matrices for the 4-leaf binary tree example. (D-E) Simulation results for the example task. (D) Fractional cumulative improvement in task performance by category depth using feed-forward noise. (E) Weight divergence between networks using gradient and directional updates. (F) Off-diagonal weight trajectories as a function of P. (G) Improvement in task performance relative to baseline for a range of signal and noise values in the initial weights. (H,I) Performance relative to gradient descent given different mixture parameters P.

To demonstrate this application, we ran a series of network simulations with hierarchical task structures. Groups of hidden units were taken by the readouts to encode objects in the input, and we defined readouts for superordinate categories including multiple objects, thereby instantiating the "or" operations discussed above. We further defined non-overlapping object representations as binary vectors in the input, and initialized network weights to small Gaussian random values. The learning task was then to correctly respond to both category queries and object identification queries. Lastly, because forward-propagating noise extrapolates from extant information in the weight structure, we trained the network for several iterations on individual items using GD.

When the network continues learning with gradients, the category conditions tend to mix processing in the hidden layer before ultimately removing these mixed representations (figure 4, E and F). Notably, training the network with forward-propagating noise maintains the compositional character of the hidden representations (panels E,F), improving performance (panels D,G, and H).

As above, our illustration depends on various parameters. Most salient are the signal and noise in initial weights and the anisotropy. Here again, a purely projective approach is not ideal. Graded anisotropy, combining feed-forward noise with an independent isotropic component, imparts useful flexibility in the weight updates (figure 4, F,H) while avoiding interference between subordinate categories (panel D). Regarding the initial weights, we find that increasing absolute noise degrades performance, because the initial gradient-based training has limited ability to remove it (panel G).

2.5 Noise correlations set eligibility through dependence

Representation de-specialization is closely related to the broader question of how learning can be biased toward compositionality. Most objects can be described as bundles of features, with varying statistical interdependence, and when learning a task, only some subset of these might be relevant. Nausea is more easily associated with food characteristics than environmental cues like time of day, for example. Generally, specific features can be preferentially attended to support directed exploration of policy space, enhancing learning and generalization (Michael J. Frank and Badre 2012; Niv et al. 2015). In this section, we consider how feature-dependent noise can work in tandem with modulated Hebbian plasticity to facilitate such preferential learning.

When average responses are subtracted from both the input and output vectors comprising Hebbian plasticity, input noise becomes the basis of the input filter rather than the stimulus itself. Noise independence between pairs of input and output features then makes their relation invisible to the Hebbian learning rule, whereas dependent-noise pairs produce candidate weight updates. Combined with layer-wise representation decompositions, networks can therefore "tag" features with varying levels of eligibility for mutual learning, even when parallel representations are active.

Mathematically, feature eligibility can be manipulated by expanding the $\beta_h = 0$, $\beta_i = 0$ plasticity case according to feature-based noise terms. For a set

of input features p_i and hidden features q_j, we can set input noise to $\sum_i \eta_{ij}^2 p_j$, and hidden noise to $\sum_j \eta_{ij} q_i$. Then private, independent noise terms η_{ij} link each p_j to each q_i. Limiting exploration of the policy space corresponds to removing (or weakening) some of these links, e.g., by setting noise between *a priori* unrelated feature categories to zero.

Technically, the square term η_{ij}^2 violates the mean-zero noise assumption, but this can be absorbed by a slightly different choice of baseline ($c_i = 1 - \langle \eta_{ij}^2 \rangle \oslash \langle x_i \rangle$, where $\oslash$ is elementwise division), as discussed in the supplement. Other means can also accomplish the same ends. Similarly, "setting the noise" can be construed as either a subtle violation of linearity or via the interpretation that there are multiple forms of activity with different transfer properties in the same network, and an update rule that depends on some of these but not others. This latter idea happens to be empirically true, since biological plasticity operates on a complex mixture of action potentials, spike rates, calcium signals, etc. Again, such considerations are elaborated in the supplement, and we return to our more heuristic discussion now.

The reward-modulated Hebbian algorithm describing our situation arises (roughly) from the $\langle a_1 b_4 \rangle$ term in equation (3). Expanding the update according to the above noise decomposition gives:

$$\langle \Delta W_h \rangle \approx 2\alpha \langle a_1 b_4 \rangle \tag{7}$$

$$= 2\alpha \langle \delta_r^T \xi_r \xi_h \xi_u^T \rangle \tag{8}$$

$$= 2\alpha \sum_{ij} \delta_r^T W_r q_i p_j^T \langle \eta_{ij}^4 \rangle \tag{9}$$

The approximate equality here comes from neglecting higher order (bias) terms. Updates thus decompose as a sum of gradient-like terms operating on input and output feature vectors, and can be considered a set of simultaneous line-searches, each defined by a "legitimate" or "matched" input-output pair (figure 4, A,B).

We simulated simple compositional task sets for illustration. Tasks were constructed by generating random basis decompositions for network input and hidden layers, then 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 likewise for target outputs. The number of input features per stimulus was determined with a compositionality parameter $C \in \{1, ..., n\}$, where n is the network width. This yields a set of n-choose-C potential inputs, which grows rapidly when C is not approximately 1 or n. Therefore, we selected stimuli to form minimal spanning sets for the features.

To perform graded feature-matching, we defined a parameter L, the "linking number". Cross-layer feature dependencies form a bipartite graph, with links between layers set by the non-zero η_{ij} terms. Whereas GD operates on the complete feature graph, the best projective algorithm operates only on those input-to-hidden feature links required for the task, reducing dimensionality. The linking number L interpolates between GD and this best projective algorithm

by setting the in- and out-degrees of each feature vector in the hidden and input layers, respectively.

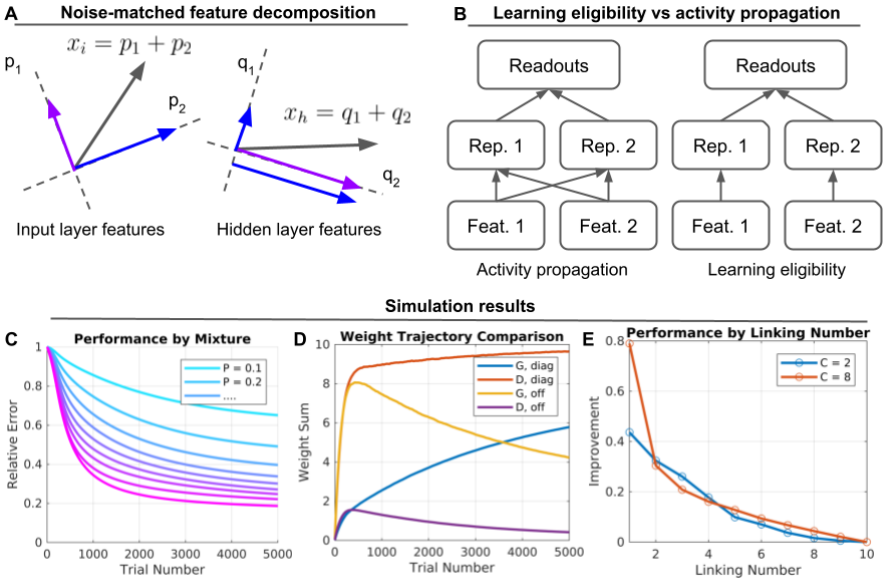

Figure 4: Eligibility through dependence (A) Representations can be decomposed according to orthogonal feature sets, and subsets of features in both layers can have dependent or independent noise with respect to one another. Here q_2 and p_1 are statistically dependent, and likewise for q_1, q_2, and p_2. (B) Elgibility segregation. Eligibility for gradient descent is defined by activity and can occur across any combination of features. Eligibility in the modulated Hebbian case can be limited by noise dependence. (C) Hebbian algorithm performance. Improvement results from both the large decrease in dimensionality of the learning problem and the orthogonalization of features. Parameters are $C = 8$, $L = 1$, and $n = 10$. (D) Average weight trajectories connecting features for the gradient and projective updates, using the same parameters as C and taking $P = 0.8$. "Diag" and "off" refer to weight groups which connect input features to their target outputs and inappropriate ones, respectively. GD learns cross-modal connections then removes them, whereas the projective algorithm avoids them. (E) Performance gains increase with compositionality and linking. GD is equivalent to the projective algorithm when the linking number is maximal (10 here). The projective algorithm improves superlinearly to its maximum, achieved when learning is completely factorized (linking = 1).

Our simulations show improved performance, as expected from the dimensionality reduction imparted by projective filtering (figure 4, panels C-E). In particular, the projective algorithm generates much less interference, solves the example task set more quickly, and accrues less error than GD. The impact

of linking number shows that any reduction in dimensionality improves performance across compositionality parameters C (fig 4E). Thus, feature matching via noise covariance and Hebbian plasticity can drastically improve learning.

3 Discussion

3.1 Review

Biological learning, as noted above, is often discussed as an approximation to gradient descent. Here we have developed the alternative perspective that biological phenomena, such as modulated Hebbian plasticity and noise correlations, can endow networks with valuable mechanisms to adaptively shape credit assignment. Whereas gradient descent can be considered optimal from the perspective of minimizing error in any given task, we have shown that inductive biases afforded by noise and Hebbian plasticity can be leveraged to minimize interference between tasks, to prevent de-specialization of representations, and to bias learning toward features that are likely to be relevant in new tasks.

Specifically, we showed how noise variance interacts with synaptic plasticity to produce different input and output filters for stimulus information in networks. By developing a characterization of task interference and a generative model for tasks with certain interference properties, we showed that it is possible to shape the modulated Hebbian rules to selectively explore non-interfering synaptic update dimensions, inducing a bias that can be manipulated independently from network structure. Moreover, we showed that this selective exploration can result from adaptive noise tuning, whereby the degree of anisotropic noise can be used to focus learning on weight subspaces orthogonal to (or commensurate with) those needed to solve additional tasks. We also provided a thorough analysis of the task factors and network parameters in which such performance gains are most likely to be realized.

Subsequently, we demonstrated that noise originating from feed-forward activity could be adaptive in tasks that demand representational specialization. In particular, we showed how the hierarchy of nested "or" functions exhibited by a categorization task induces gradients which tend to de-specialize neural representations of subordinate categories. We showed that this de-specialization is a function of the noise subspace available for estimating a gradient, and we used information encoded in existing weights to limit the effective dimension of that space. As such, we demonstrated one instance in which destructive task interference could be avoided even without orthogonalizing gradients. More generally, we demonstrated the potential value of, and a mechanism for, segregating learning across feed-forward pathways when there is reason to believe those pathways should remain segregated.

Finally, we showed how the specific interaction of learning rule and noise structure suggests an "eligibility through dependence" function. By decomposing representations within both sending and receiving layers of a network, the statistical dependence of the noise could be used to make subsets of features in

the input and output mutually "available" or "unavailable" for learning. Of key importance here was the compatibility of this description with the decomposition of a Hebbian learning rule into gradient projections operating on feature pairs. By routing learning eligibility according to a-priori relevance, such a segregation results in substantially reduced learning dimensionality and can greatly improve performance.

3.2 Relation to other work

Our results provide an unexpected generalization of related noise correlation work. Nassar et al. 2021 found that by fixing signal to noise ratios and varying noise correlation in a two-alternative forced choice task, they could improve learning speed and weight homogeneity. Our first principles approach to studying the interaction between Hebbian plasticity and noise has converged on a new framework for understanding their results. In the terms we developed here, their manipulation consisted of tuning anisotropy to maximize constructive interference between two sub-tasks with a special geometry (elaborated in the supplement). Our framework generalizes their results and applies to other phenomena, including general interference, multi-task learning and eligibility through dependence.

At a broader, conceptual scale, our learning algorithm is complementary to other strategies in computational neuroscience, whereby interference can be managed by changes in the network architecture (e.g., hippocampus and cortex; Mcclelland et al. 1995; O'Reilly and Norman 2002; Schapiro et al. 2017). Such strategies posit a division of labor for orchestrating complementary biases among sub-networks of a learning agent. These sub-networks may support targeted sampling via noise, and our geometric analyses may describe many neural systems. For example, prefrontal cortical networks involving the basal ganglia can "gate" stimulus dimensions, providing top-down biases onto cortical and striatal representations, thereby changing their eligibility for learning (Rougier et al. 2005; Michael J. Frank and Badre 2012; Collins and Michael J. Frank 2013; Franklin and Michael J Frank 2015; Stalnaker et al. 2016). However, because these gating strategies recruit distinct neural populations across tasks, they are unable to capitalize on possible constructive components or shared abstractions that facilitate learning (Musslick et al. 2020, November 16). Thus, integrating our projective algorithm into a gating framework might make both approaches more powerful.

In basal ganglia models, opponent striatal populations, which respond in opposite ways to dopaminergic Hebbian plasticity (Michael J. Frank 2005; Gurney et al. 2015), exhibit advantages over classical RL algorithms (Jaskir and Michael J. Frank 2021) and might be fruitfully investigated as a means of geometric interference avoidance. Moreover, recent data indicate that dopamine conveys more than just scalar reward prediction errors (Langdon et al. 2018; Engelhard et al. 2019; Hamid et al. 2021). We expect these graded or vector-like RPE signals to impact policy gradients collinearly with the noise variance changes we discuss, since both can be interpreted as assigning credit to a subset

of dimensions (or "experts" in the mixture-of-experts interpretation of Hamid et al. 2021).

Additionally, we expect work on key biological phenomena, such as the large principle component of approximately uniform firing in noise correlations (Ecker et al. 2016; Kanashiro et al. 2017; Ni et al. 2018) to determine whether and how faithfully our descriptions here apply to real systems. In the supplement we elaborate testable predictions arising from our framework across several domains of study, including work decomposing noise variance and stimulus responses (A. Kohn 2005; A. Luczak et al. 2007; Artur Luczak et al. 2009), work on inhibitory control of network properties (Isaacson and Scanziani 2011; Sippy and Yuste 2013), and work on choice and stimulus probabilities (Haefner et al. 2013; Voelcker and Peron 2021, September 17; Yang et al. 2016).

3.3 Acknowledgements

For helpful discussion, commentary, and feedback, we thank Mathew Nassar, Gabriel Provencher Langlois, Apoorva Bhandari, Rex Liu, Christopher I. Moore, Christopher Deister, 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.

3.4 Author contributions

D.N.S. and M.J.F. developed the research topic. D.N.S. conceived and developed the mathematical analyses, wrote code, and performed simulations. M.J.F. provided extensive feedback at all project stages. D.N.S. and M.J.F wrote the manuscript and prepared it for submission.

3.5 Declaration of interests

The authors declare no competing interests.

References

Averbeck, B. B., Latham, P. E., & Pouget, A. (2006). Neural correlations, population coding and computation. *Nature Reviews Neuroscience*, *7*(5), 358–366. https://doi.org/10.1038/nrn1888

Bellec, G., Scherr, F., Hajek, E., Salaj, D., Legenstein, R., & Maass, W. (2019). Biologically inspired alternatives to backpropagation through time for learning in recurrent neural nets. *arXiv:1901.09049 [cs]*. http://arxiv.org/abs/1901.09049

Bi, G.-q., & Poo, M.-m. (1998). Synaptic modifications in cultured hippocampal neurons: Dependence on spike timing, synaptic strength, and postsynaptic cell type. *The Journal of Neuroscience*, *18*(24), 10464–10472. https://doi.org/10.1523/JNEUROSCI.18-24-10464.1998

Collins, A. G. E., & Frank, M. J. [Michael J.]. (2013). Cognitive control over learning: Creating, clustering, and generalizing task-set structure. *Psychological Review*, *120*(1), 190–229. https://doi.org/10.1037/a0030852

Dan, Y., & Poo, M.-m. (2004). Spike timing-dependent plasticity of neural circuits. *Neuron*, *44*(1), 23–30. https://doi.org/10.1016/j.neuron.2004.09.007

Ecker, A. S., Denfield, G. H., Bethge, M., & Tolias, A. S. (2016). On the structure of neuronal population activity under fluctuations in attentional state. *The Journal of Neuroscience*, *36*(5), 1775–1789. https://doi.org/10.1523/JNEUROSCI.2044-15.2016

Engelhard, B., Finkelstein, J., Cox, J., Fleming, W., Jang, H. J., Ornelas, S., Koay, S. A., Thiberge, S. Y., Daw, N. D., Tank, D. W., & Witten, I. B. (2019). Specialized coding of sensory, motor and cognitive variables in VTA dopamine neurons. *Nature*, *570*(7762), 509–513. https://doi.org/10.1038/s41586-019-1261-9

Fiete, I. R., & Seung, H. S. (2006). Gradient learning in spiking neural networks by dynamic perturbation of conductances. *Physical Review Letters*, *97*(4), 048104. https://doi.org/10.1103/PhysRevLett.97.048104

Flesch, T., Balaguer, J., Dekker, R., Nili, H., & Summerfield, C. (2018). Comparing continual task learning in minds and machines [Publisher: National Academy of Sciences Section: PNAS Plus]. *Proceedings of the National Academy of Sciences*, *115*(44), E10313–E10322. https://doi.org/10.1073/pnas.1800755115

Flesch, T., Juechems, K., Dumbalska, T., Saxe, A., & Summerfield, C. (2021). Rich and lazy learning of task representations in brains and neural networks [Publisher: Cold Spring Harbor Laboratory Section: New Results]. *bioRxiv*, 2021.04.23.441128. https://doi.org/10.1101/2021.04.23.441128

Frank, M. J. [Michael J.]. (2005). Dynamic dopamine modulation in the basal ganglia: A neurocomputational account of cognitive deficits in medicated and nonmedicated parkinsonism. *Journal of Cognitive Neuroscience*, *17*(1), 51–72. https://doi.org/10.1162/0898929052880093

Frank, M. J. [Michael J.], & Badre, D. (2012). Mechanisms of hierarchical reinforcement learning in corticostriatal circuits 1: Computational analysis. *Cerebral Cortex*, *22*(3), 509–526. https://doi.org/10.1093/cercor/bhr114

Franklin, N. T., & Frank, M. J. [Michael J]. (2015). A cholinergic feedback circuit to regulate striatal population uncertainty and optimize reinforcement learning (U. S. Bhalla, Ed.) [Publisher: eLife Sciences Publications, Ltd]. *eLife*, *4*, e12029. https://doi.org/10.7554/eLife.12029

Fremaux, N., Sprekeler, H., & Gerstner, W. (2010). Functional requirements for reward-modulated spike-timing-dependent plasticity. *Journal of Neuro-*

science, *30*(40), 13326–13337. https://doi.org/10.1523/JNEUROSCI.6249-09.2010

Frémaux, N., & Gerstner, W. (2016). Neuromodulated spike-timing-dependent plasticity, and theory of three-factor learning rules. *Frontiers in Neural Circuits*, *9*. https://doi.org/10.3389/fncir.2015.00085

Frémaux, N., Sprekeler, H., & Gerstner, W. (2013). Reinforcement learning using a continuous time actor-critic framework with spiking neurons [Publisher: Public Library of Science]. *PLOS Computational Biology*, *9*(4), e1003024. https://doi.org/10.1371/journal.pcbi.1003024

Gawne, T. J., & Richmond, B. J. (1993). How independent are the messages carried by adjacent inferior temporal cortical neurons? [Publisher: Society for Neuroscience Section: Articles]. *Journal of Neuroscience*, *13*(7), 2758–2771. https://doi.org/10.1523/JNEUROSCI.13-07-02758.1993

Guerguiev, J., Lillicrap, T. P., & Richards, B. A. (2017). Towards deep learning with segregated dendrites. *eLife*, *6*.

Gurney, K. N., Humphries, M. D., & Redgrave, P. (2015). A new framework for cortico-striatal plasticity: Behavioural theory meets in vitro data at the reinforcement-action interface [Publisher: Public Library of Science]. *PLOS Biology*, *13*(1), e1002034. https://doi.org/10.1371/journal.pbio.1002034

Haefner, R. M., Gerwinn, S., Macke, J. H., & Bethge, M. (2013). Inferring decoding strategies from choice probabilities in the presence of correlated variability. *Nature Neuroscience*, *16*(2), 235–242. https://doi.org/10.1038/nn.3309

Hamid, A. A., Frank, M. J., & Moore, C. I. (2021). Wave-like dopamine dynamics as a mechanism for spatiotemporal credit assignment. *Cell*, *184*(10), 2733–2749.e16. https://doi.org/10.1016/j.cell.2021.03.046

Isaacson, J. S., & Scanziani, M. (2011). How inhibition shapes cortical activity. *Neuron*, *72*(2), 231–243. https://doi.org/10.1016/j.neuron.2011.09.027

Jaskir, A., & Frank, M. J. [Michael J.]. (2021). On the normative advantages of basal ganglia opponency in decision-making. *bioRxiv*.

Kanashiro, T., Ocker, G. K., Cohen, M. R., & Doiron, B. (2017). Attentional modulation of neuronal variability in circuit models of cortex. *eLife*, *6*, e23978. https://doi.org/10.7554/eLife.23978

Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A. A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., Hassabis, D., Clopath, C., Kumaran, D., & Hadsell, R. (2017). Overcoming catastrophic forgetting in neural networks [Publisher: National Academy of Sciences Section: Biological Sciences]. *Proceedings of the National Academy of Sciences*, *114*(13), 3521–3526. https://doi.org/10.1073/pnas.1611835114

Kohn, A. [A.]. (2005). Stimulus dependence of neuronal correlation in primary visual cortex of the macaque. *Journal of Neuroscience*, *25*(14), 3661–3673. https://doi.org/10.1523/JNEUROSCI.5106-04.2005

Kohn, A. [Adam], Coen-Cagli, R., Kanitscheider, I., & Pouget, A. (2016). Correlations and neuronal population information. *Annual Review of Neu-*

roscience, *39*(1), 237–256. https://doi.org/10.1146/annurev-neuro-070815-013851

Langdon, A. J., Sharpe, M. J., Schoenbaum, G., & Niv, Y. (2018). Model-based predictions for dopamine. *Current Opinion in Neurobiology*, *49*, 1–7. https://doi.org/10.1016/j.conb.2017.10.006

Linsley, D., Kim, J., Berson, D., & Serre, T. (2020). Robust neural circuit reconstruction from serial electron microscopy with convolutional recurrent networks [version: 3]. *arXiv:1811.11356 [cs]*. http://arxiv.org/abs/1811.11356

Luczak, A. [A.], Bartho, P., Marguet, S. L., Buzsaki, G., & Harris, K. D. (2007). Sequential structure of neocortical spontaneous activity in vivo. *Proceedings of the National Academy of Sciences*, *104*(1), 347–352. https://doi.org/10.1073/pnas.0605643104

Luczak, A. [Artur], Barthó, P., & Harris, K. D. (2009). Spontaneous events outline the realm of possible sensory responses in neocortical populations [Publisher: Elsevier]. *Neuron*, *62*(3), 413–425. https://doi.org/10.1016/j.neuron.2009.03.014

Mcclelland, J. L., Mcnaughton, B. L., & O'Reilly, R. C. (1995). Why there are complementary learning systems in the hippocampus and neocortex: Insights from the successes and failures of connectionist models of learning and memory. *Psychological Review*, *102*(3), 419–457.

McCloskey, M., & Cohen, N. J. (1989). Catastrophic interference in connectionist networks: The sequential learning problem. *Psychology of learning and motivation* (pp. 109–165). Elsevier.

Moldwin, T., Kalmenson, M., & Segev, I. (2021). The gradient clusteron: A model neuron that learns to solve classification tasks via dendritic nonlinearities, structural plasticity, and gradient descent [Publisher: Public Library of Science]. *PLOS Computational Biology*, *17*(5), e1009015. https://doi.org/10.1371/journal.pcbi.1009015

Musslick, S., Saxe, A., Hoskin, A. N., Reichman, D., & Cohen, J. D. (2020, November 16). *On the rational boundedness of cognitive control: Shared versus separated representations* (preprint). PsyArXiv. https://doi.org/10.31234/osf.io/jkhdf

Nassar, M. R., Scott, D., & Bhandari, A. (2021). Noise correlations for faster and more robust learning. *The Journal of Neuroscience*, *41*(31), 6740–6752. https://doi.org/10.1523/JNEUROSCI.3045-20.2021

Ni, A. M., Ruff, D. A., Alberts, J. J., Symmonds, J., & Cohen, M. R. (2018). Learning and attention reveal a general relationship between population activity and behavior [Publisher: American Association for the Advancement of Science Section: Report]. *Science*, *359*(6374), 463–465. https://doi.org/10.1126/science.aao0284

Niv, Y., Daniel, R., Geana, A., Gershman, S. J., Leong, Y. C., Radulescu, A., & Wilson, R. C. (2015). Reinforcement learning in multidimensional environments relies on attention mechanisms [Publisher: Society for Neuroscience Section: Articles]. *Journal of Neuroscience*, *35*(21), 8145–8157. https://doi.org/10.1523/JNEUROSCI.2978-14.2015

O'Reilly, R. C. (1996). Biologically plausible error-driven learning using local activation differences: The generalized recirculation algorithm [Place: US Publisher: MIT Press]. *Neural Computation*, *8*(5), 895–938. https://doi.org/10.1162/neco.1996.8.5.895

O'reilly, R. C. (2001). Generalization in interactive networks: The benefits of inhibitory competition and hebbian learning. *Neural Computation*, *13*, 1199–1242.

O'Reilly, R. C., & Norman, K. A. (2002). Hippocampal and neocortical contributions to memory: Advances in the complementary learning systems framework. *Trends in cognitive sciences*, *6*(12), 505–510. http://www.sciencedirect.com/science/article/pii/S1364661302020053

Ratcliff, R. (1990). Connectionist models of recognition memory: Constraints imposed by learning and forgetting functions. *Psychological Review*, 285–308.

Rougier, N. P., Noelle, D. C., Braver, T. S., Cohen, J. D., & O'Reilly, R. C. (2005). Prefrontal cortex and flexible cognitive control: Rules without symbols [Publisher: National Academy of Sciences Section: Biological Sciences]. *Proceedings of the National Academy of Sciences*, *102*(20), 7338–7343. https://doi.org/10.1073/pnas.0502455102

Ruan, H., Saur, T., & Yao, W.-D. (2014). Dopamine-enabled anti-hebbian timing-dependent plasticity in prefrontal circuitry [Publisher: Frontiers]. *Frontiers in Neural Circuits*, *8*. https://doi.org/10.3389/fncir.2014.00038

Sanger, T. D. (1989). Optimal unsupervised learning in a single-layer linear feedforward neural network. *Neural Networks*, *2*(6), 459–473. https://doi.org/10.1016/0893-6080(89)90044-0

Saxe, A. M., McClelland, J. L., & Ganguli, S. (2014). Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. *arXiv:1312.6120 [cond-mat, q-bio, stat]*. http://arxiv.org/abs/1312.6120

Schapiro, A. C., Turk-Browne, N. B., Botvinick, M. M., & Norman, K. A. (2017). Complementary learning systems within the hippocampus: A neural network modelling approach to reconciling episodic memory with statistical learning. *Philosophical Transactions of the Royal Society B: Biological Sciences*, *372*(1711), 20160049. https://doi.org/10.1098/rstb.2016.0049

Schrimpf, M., Kubilius, J., Hong, H., Majaj, N. J., Rajalingham, R., Issa, E. B., Kar, K., Bashivan, P., Prescott-Roy, J., Geiger, F., Schmidt, K., Yamins, D. L. K., & DiCarlo, J. J. (2018). *Brain-score: Which artificial neural network for object recognition is most brain-like?* (preprint). Neuroscience. https://doi.org/10.1101/407007

Seol, G. H., Ziburkus, J., Huang, S., Song, L., Kim, I. T., Takamiya, K., Huganir, R. L., Lee, H.-K., & Kirkwood, A. (2007). Neuromodulators control the polarity of spike-timing-dependent synaptic plasticity. *Neuron*, *55*(6), 919–929. https://doi.org/10.1016/j.neuron.2007.08.013

Shadlen, M. N., & Newsome, W. T. (1994). Noise, neural codes and cortical organization. *Current Opinion in Neurobiology*, *4*(4), 569–579. https://doi.org/10.1016/0959-4388(94)90059-0

Shen, W., Flajolet, M., Greengard, P., & Surmeier, D. J. (2008). Dichotomous dopaminergic control of striatal synaptic plasticity [Publisher: American Association for the Advancement of Science]. *Science*, *321*(5890), 848–851. https://doi.org/10.1126/science.1160575

Sippy, T., & Yuste, R. (2013). Decorrelating action of inhibition in neocortical networks [Publisher: Society for Neuroscience Section: Articles]. *Journal of Neuroscience*, *33*(23), 9813–9830. https://doi.org/10.1523/JNEUROSCI.4579-12.2013

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting [Publisher: JMLR. org]. *The journal of machine learning research*, *15*(1), 1929–1958.

Stalnaker, T. A., Berg, B., Aujla, N., & Schoenbaum, G. (2016). Cholinergic interneurons use orbitofrontal input to track beliefs about current state [Publisher: Society for Neuroscience Section: Articles]. *Journal of Neuroscience*, *36*(23), 6242–6257. https://doi.org/10.1523/JNEUROSCI.0157-16.2016

Vasilaki, E., Frémaux, N., Urbanczik, R., Senn, W., & Gerstner, W. (2009). Spike-based reinforcement learning in continuous state and action space: When policy gradient methods fail. *PLoS Computational Biology*, *5*(12). https://doi.org/10.1371/journal.pcbi.1000586

Voelcker, B., & Peron, S. (2021, September 17). *Transformation of primary sensory cortical representations from layer 4 to layer 2* (preprint). Neuroscience. https://doi.org/10.1101/2021.09.17.460780

Williams, R. J. (1992). Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, *8*(3-4), 229–256.

Xie, X., & Seung, H. S. (2004). Learning in neural networks by reinforcement of irregular spiking. *Physical Review E*, *69*(4), 041909. https://doi.org/10.1103/PhysRevE.69.041909

Yang, H., Kwon, S. E., Severson, K. S., & O'Connor, D. H. (2016). Origins of choice-related activity in mouse somatosensory cortex [Bandiera_abtest: a Cg_type: Nature Research Journals Number: 1 Primary_atype: Research Publisher: Nature Publishing Group Subject_term: Barrel cortex;Cortex;Neural circuits;Sensory processing;Whisker system Subject_term_id: barrel-cortex;cortex;neural-circuit;sensory-processing;whisker-system]. *Nature Neuroscience*, *19*(1), 127–134. https://doi.org/10.1038/nn.4183

Zenke, F., & Ganguli, S. (2018). SuperSpike: Supervised learning in multilayer spiking neural networks. *Neural Computation*, *30*(6), 1514–1541. https://doi.org/10.1162/neco_a_01086

Zohary, E., Shadlen, M. N., & Newsome, W. T. (1994). Correlated neuronal discharge rate and its implications for psychophysical performance. *Nature*, *370*(6485), 140–143. https://doi.org/10.1038/370140a0

4 Supplementary material

4.1 Hebbian theories, interneurons, and noise control

Our results require some form of either online or prescriptive control of network noise covariance, depending on how much task knowledge is taken to be known beforehand (and its content). Because covariance is a macroscopic property, i.e. a property of the system of neurons rather than some small subset of them, the noise controller must have access to (and control of) most or all of the neurons' activities. In cortex, this naturally implicates inhibitory interneurons, which are in a unique position to integrate network information; interneurons have high interconnectivity and exert tight control over pyramidal neurons (Gibson et al. 1999; Fino and Yuste 2011; Fino, Packer, et al. 2012; King et al. 2013; Pi et al. 2013; Yavorska and Wehr 2016). Moreover, some interneurons are sensitive to dopaminergic modulation, as the present analysis (hypothesizing adaptive, interneuron based noise control) would predict. Vasoactive intestinal peptide (VIP) expressing neurons, for example, specifically respond to reward signals by disinhibiting functional pyramidal populations (Pi et al. 2013).

Similar ideas may apply in the striatum, where dopamine-modulated plasticity is most well established (Shen et al. 2008; Yagishita et al. 2014; Ruan et al. 2014). Local cholinergic interneurons (TANs) integrate information from surrounding medium spiny neurons (MSNs) and reciprocally influence the degree to which they are eligible for plasticity by DA signals (eg Morris et al. 2004; Franklin and Michael J Frank 2015; Cragg 2006). Orbitofrontal cortex is reported to send top-down state input to TANS (Stalnaker et al. 2016), which may therefore act on this information by modifying noise covariance in the MSNs. Indeed, doing so could provide a complementary, dimensional elaboration of TANs' hypothesized entropy-response characteristics (Franklin and Michael J Frank 2015). Furthermore, sampling across D1 and D2 populations could operate independently on opponent representations of costs and benefits of alternative actions. Future work can consider how it might be constructive to sample noise in such an opponent scheme so as to facilitate and suppress action relationships with distinct input features.

While interneuron control of plasticity has been postulated in classical Hebbian literature before, this has been in the unsupervised setting, and towards somewhat different ends. The naive Hebbian algorithm (raw pre x post synaptic activity) requires some form of normalization to keep the weights from growing in magnitude indefinitely, but under loose assumptions the weights converge to a matched filter for the first principal component of the data. That is, they converge to $W \propto q_1 p_1^T$, where p_1 is the first principal component of the input covariance, $q_1 = W_0 p_1$, and W_0 is the initial weight matrix. Sanger's rule (Sanger 1989) suggests using feedback inhibition from the second layer onto the first in order to perform Grahm-Schmidt orthogonalization over learning, and thereby sequentially extract additional principal components. Sanger also explored the use of competitive lateral inhibition (Rumelhart and Zipser 1985) for separating outputs. The former idea was itself based on earlier work by

(Oja 1982), and an adaptive covariance matrix estimation algorithm developed by Karhunen (Karhunen 1984). An important difference between this classic work and our predictions, however, is that we suggest inhibition shapes noise, whereas earlier work simply required noiseless inhibition to "turn down" activity on certain units in a consistent, prescribed, and non-exploratory manner.

Hebbian algorithms like ours have the salient requirement of "knowing" what average activity and reward rates are for a given input. Previous work on contrastive Hebbian learning (Ackley et al. 1985), and the subsequently developed GeneRec and XCAL algorithms (O'Reilly 1996; O'reilly 2001), provide a potential means of addressing this. These latter rules are supervised algorithms based on Boltzmann machine learning (Ackley et al. 1985). They function by comparing evoked and "clamped" (i.e., supervised) activity in a recurrent network to generate a gradient estimate. The idea that early activity could be used to generate a network prediction, and that later activity would provide a differential learning signal relative to this, could also be used in an algorithm such as ours. That is, networks could generate the activity and reward baselines necessary for comparison with trial-based outcomes at trial-time. For example, this comparison between activity levels during expectation and outcome is amplified by dopaminergic dynamics within striatal models of RL, driving learning between contrasting attractor states (Michael J. Frank 2005; Franklin and Michael J Frank 2015). Integrated with our results, this would suggest late-emerging trial-wise noise correlations, because the early phase of the response would be stereotyped and the latter phase would be exploratory. This has the intriguing possibility of accounting for the recent finding that choice-probability increases over time in rodent S1, following an early, transient, and strong stimulus-probability (Voelcker and Peron 2021, September 17).

Finally, our work is related to research examining the integration of Hebbian and non-Hebbian algorithms. In particular, other authors have often addressed mixed supervised and unsupervised learning schemes (O'reilly 2001; O'Reilly et al. 2012; Krotov and Hopfield 2019). These generally operate as regularization schemes or as biases forcing networks to develop broadly useful and re-usable representations. While Hebbian rules do not perform unsupervised learning in our work, one could naturally interpolate between our case and unsupervised learning by grading the stimulus selectivity of activity baseline comparisons and reward modulation.

4.2 Relations to cognitive neuroscience

Differential impacts of reward signals via modulated Hebbian plasticity are also prominent in the distinction between so called "go" and "no-go" striatal neurons, which exhibit D1 and D2 dopamine receptors. In such a scheme, striatal neurons learn from RPEs relative to their baseline expectation, much like the REINFORCE algorithm, but in opposite directions. The opponency of this system is thought to allow D1 neurons to specialize in representing benefits of an action given the current input, whereas the D2 neurons come to represent the cost of that action (Michael J. Frank 2005; Franklin and Michael J Frank 2015).

The original model explored the utility of the opponency for learning complex probabilistic classification tasks (e.g., the weather prediction task), which may be interpretable in terms of the internal segregation of credit assignment we've discussed here. Moreover, it may be productive to consider in future work how noise can be differentially allocated to the inputs to D1 and D2 neurons to explore a range of policies in which actions can be facilitated or suppressed under different stimulus configurations.

At a systems level, the tuning of noise parameters could be accomplished by meta reinforcement learning (Wang et al. 2018), by gating of prefrontal subpopulations in working memory (Michael J. Frank and Badre 2012) or by attentional mechanisms (Niv et al. 2015). Each of these could provide a means to focus reinforcement learning along only those subspaces in a learning scenario that are relevant to an agent (or which satisfy e.g. non-interference criteria). In turn, the tools we have developed to discuss and characterize task relationships, and our characterization of policy gradient manipulations, should be broadly portable to analyses of these systemic processes, regardless of any particular commitment to implementational theories.

4.3 Relation to Nassar, Scott, and Bhandari 2021

As noted in the text, Nassar et al. 2021 find that noise correlations focused on the signal dimension of a two-alternative forced choice discrimination task improve learning speed and enhance the homogeneity of the learned network weights. (There are other findings as well, but these are the relevant ones here.) The learning rule used was a rate-coded, uncorrected Hebbian rule, which fits into our framework by taking $\beta_h = 1$ and $\beta_i = 1$, and which itself had been introduced in Law and Gold 2009. The discrimination task performed involved two completely overlapping inputs (evidence for stimulus 1 and against stimulus 2, vs evidence for stimulus 2 and against stimulus 1) and two completely overlapping readouts (antagonistic responding). These inputs and outputs can be considered as two tasks with essential interference and identical solution manifolds.

Notably, the paper's aim was to explore a previously understudied aspect of noise correlations (signal to noise ratio) and to look into Hebbian mechanisms for generating correlation. These differ from our goals here, of understanding the general relationship between noise, Hebbian rules, and gradients. As we discovered in this work, the paper's relevant manipulation was equivalent to varying the projective parameter P discussed above, in order to maximize constructive interference. This is itself equivalent to promoting sample based gradient estimation along the true gradient dimension relative to others. In the projective limit, every weight update then occurs strictly along the gradient, so that every trial constitutes a gradient step (i.e. has no orthogonal component) down the loss. Hence the improved learning speed, and weight distribution homogeneity. Nassar et al. 2021 did not characterize general features of Hebbian rules or of noise distributions, but were interested instead in several particular network and noise configurations that appeared to warrant consideration given prior literature. This meant, for example, that there was no general treatment

of any of the main elements of this manuscript, such as interference, solution manifold geometries, uses of noise based on the Hebbian equations, etc.

4.4 Predictions

Our work suggests the following predictions and analyses: (i) Noise should be partially exploratory, (ii) it should be factorizable, (iii) the components should align with learning dimensions, (iv) variances and representations should co-evolve, and (v) learning dimensions should reflect task knowledge. Additionally, (vi) interneuron networks should control learning dimension, (vii) directed sampling should reflect minimum energy network excitations, and (viii) consolidation and sampling mechanisms should cooperate dimensionally. Points (i) - (v) reflect the update-sampling logic of the noise. (vi) - (vii) reflect the high interconnectivity of interneuron networks, which suggests they can integrate network information and determine activity dimensions. (viii) is a normative claim for the mutual constraint of noise and consolidation, given (i)-(iii).

Elaborating, we expect activity, weights, and noise to co-evolve in a generalized Hebbian sense, with noise driving weights driving activity in rewarded dimensions. The alternative is factorized learning across neurons. We also expect learned dimensions will be consolidated, and predict this will move representations in or out of the dimensions noise variance drives learning in (depending on expectations of future needs). Moreover, sampling dimensions should be determined by a combination of low-energy network excitations (defined by network weights and topology) and top-down input characteristics. The former presumably reflect prior experience and the latter should represent task knowledge. Both likely participate in tuning noise to maximize learning speed, minimize sampling, avoid interference, and match features. Each point above requires analysis of noise factorizations, dynamics, relations with representations, and relations with task knowledge.

4.5 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:

$$\Delta w_{ij} = \alpha(r - \bar{r})(y_i - \bar{y}_i)x_j \tag{10}$$

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 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. While Williams' proof is done fairly abstractly, the direct equivalence for exponential families can be seen in our appendix entries on modulated plasticity and analytic policy gradients, where it is derived via linear algebra and matrix calculus rather than the log-derivative trick.

4.6 Modulated plasticity with quadratic loss RPEs

In this section we derive the expected weight update under reward-modulated Hebbian plasticity. That is, we compute the expected value of equation (2) over trials. Equation (2) was:

$$\Delta W_h = \alpha(r - \langle r \rangle)(x_h - c_1\langle x_h \rangle)(x_i - c_2\langle x_i \rangle)^T$$

With reward based on mean squared error and a constant offset a, we have:

$$\begin{aligned} r &= a - (x_r^* - x_r)^2 \\ &= a - (x_r^* - \mu_r - \xi_r)^2 \\ &\equiv a - (\delta_r - \xi_r)^2 \end{aligned}$$

We have used the last equation to define δ_r, which is the same definition found in the text. The mean of this quantity is:

$$\begin{aligned} \langle r \rangle &= a - \langle (\delta_r - \xi_r)^2 \rangle \\ &= a - \langle (\delta_r^2 - 2\delta_r^T \xi_r - \xi_r^2) \rangle \\ &= a - (\delta_r^2 - \langle \xi_r^T \xi_r \rangle) \end{aligned}$$

Hence, cancelling a terms, δ_r^2 terms, and the zero-expectation cross term in $\langle r \rangle$, the reward prediction error can be written:

$$r - \langle r \rangle = 2\delta_r^T \xi_r - \xi_r^T \xi_r + \langle \xi_r^T \xi_r \rangle$$

These are the a_i terms in the factored expectation, equation (3). This can be interpreted as saying that the reward prediction error will be positive if either the noise pushes the response unit activity in the right direction (via $\delta_r^T \xi_r$), or the magnitude of the noise in the readouts $\xi_r^T \xi_r$ is smaller than the expected

amount $\langle \xi_r^T \xi_r \rangle$. This motivates the definition of the term $\zeta_r = \langle \xi_r^T \xi_r \rangle - \xi_r^T \xi_r$, which measures the relative magnitude of readout noise, giving:

$$r - \langle r \rangle = 2\delta_r^T \xi_r - \zeta_r$$

The Hebbian term in the weight update is:

$$\begin{aligned}(x_h - c_h\langle x_h \rangle)(x_i - c_i\langle x_i \rangle)^T &= (\mu_h + \xi_h - c_h\mu_h)(\mu_i + \xi_i - c_i\mu_i)^T \\ &= \beta_h\beta_i\mu_h\mu_i^T + \beta_i\xi_h\mu_i^T + \beta_h\mu_h\xi_i^T + \xi_h\xi_i^T\end{aligned}$$

These are the b_j terms in equation (3). Therefore, the expected weight update is the set of $\langle a_i b_j \rangle$ terms. If we specialize to the case of $\beta_h = 0$, $\beta_i = 1$ and take the noise to be Gaussian, we reduce the equation to:

$$\begin{aligned}\langle \Delta W_h \rangle &= \langle \alpha(r - \langle r \rangle)(x_h - \langle x_h \rangle)x_i^T \rangle \\ &= \alpha\langle (2\delta_r^T \xi_r - \zeta_r)(\xi_h \mu_i^T + \xi_h \xi_i^T) \rangle \\ &= 2\alpha\langle \delta_r^T \xi_r \xi_h \mu_i^T \rangle - \alpha\langle \zeta_r \xi_h \xi_i^T \rangle \\ &= 2\alpha\langle \delta_r^T \xi_r \xi_h \mu_i^T \rangle - 2\alpha\langle \xi_r^T \xi_r \rangle \langle \xi_h \xi_i^T \rangle\end{aligned}$$

This is equation (4) in the text. In moving from the second to third equation we've made use of the symmetry of Gaussian distributions, so that third moments are zero. In moving from the third to the fourth, we've used the fact that fourth order statistics of Gaussians can be re-written as products of second order statistics. Before showing the conditions under which this reduces to a gradient (section 4.8), we derive the gradient in question (section 4.7).

4.7 Analytic policy gradient for a linear network

In the simplest noise free case of a two layer network $y = Wx$ we can perform the following manipulations. First we expand the loss:

$$\begin{aligned}\mathcal{L}(y) &= (y - y^*)^T(y - y^*) \\ &= (Wx - y^*)^T(Wx - y^*) \\ &= x^T W^T W x - y^{*T} W x - x^T W^T y^* + y^{*T} y^*\end{aligned}$$

Then, with recourse to some well known formulas we find:

$$\begin{aligned}\nabla_W \mathcal{L} &= W(xx^T + xx^T) - y^* x^T - y^* x^T \\ &= 2(Wx - y^*)x^T \\ &= 2(y - y^*)x^T\end{aligned}$$

Thus one obtains the generalization of a one dimensional quadratic gradient to a quadratic form. For the case involving derivatives of matrices inside products (e.g. $\partial_A CADA^T$), it is easier to make use of a formulation of matrix calculus than to get things indirectly with well known basic formulas. There are several

ways to do this, i.e. to calculate gradients of arrays with respect to other arrays. We use the so-called "narrow" or α-derivative, defined as $DF(x) = \partial \text{vec}F / \partial (\text{vec}F)^T$. Since we are taking the derivative of a scalar (reward) with respect to a 2D array (a weight matrix), the α-derivative will be a vector we can naturally reshape back into a matrix of equal dimensions with the weights. The quadratic form defining reward is:

$$\langle r \rangle = \langle a - (\delta_r - \xi_r)^T (\delta_r - \xi_r) \rangle$$

We can use the chain rule and the inner-product derivative to write:

$$d\langle r \rangle = -2 \langle (\delta_r - \xi_r)^T (d\delta_r - d\xi_r) \rangle$$

Hence we must compute two component differentials. To do so we make use of the Kronecker product ($\otimes$) relation:

$$AB = (I \otimes A)\text{vec}(B) = (B^T \otimes I)\text{vec}(A)$$

Taking the differential of the average response error we have:

$$\begin{aligned} d\delta_r &= d(x_r^* - W_r W_h W_i \mu) \\ &= -W_r d\text{vec}(W_h \mu_i) \\ &= -W_r (\mu_i^T \otimes I) d\text{vec} W_h \end{aligned}$$

For the response noise we have we have:

$$\begin{aligned} d\xi_r &= d(W_r W_h W_i \xi) \\ &= d(W_r (\xi_i^T \otimes I)\text{vec} W_h) \\ &= W_r (\xi_i^T \otimes I) d\text{vec} W_h \end{aligned}$$

In both derivations we have made use of the Kronecker product to rewrite the matrix product with the free variables (i.e. the terms $[W_h]_{ij}$) arranged as a vector. Assembling these pieces we have that:

$$\begin{aligned} \frac{\partial \langle r \rangle}{\partial (\text{vec} W_h)^T} &= -2 \langle (\delta_r - \xi_r)^T W_r (-\mu_i^T \otimes I - \xi_i^T \otimes I) \rangle \\ (\text{vec} \frac{\partial \langle r \rangle}{\partial W_h})^T &= 2 \langle \delta_r^T W_r (\mu_i^T \otimes I) \rangle - 2 \langle \xi_r^T W_r (\xi_i^T \otimes I) \rangle \\ \text{vec} \frac{\partial \langle r \rangle}{\partial W_h} &= 2 \langle (\mu_i \otimes I) W_r^T \delta_r \rangle - 2 \langle (\xi_i \otimes I) W_r^T \xi_r \rangle \end{aligned}$$

Un-vectorizing this using the same Kronecker product relation gives equation (1) in the text, which we repeat here:

$$\frac{\partial \langle r \rangle}{\partial W_h} = 2 W_r^T \delta_r \mu_i^T - 2 W_r^T \langle \xi_r \xi_i^T \rangle$$

4.8 Noise decomposition and gradient projection

Important special cases of the modulated Hebbian updates occur when we take noise in each layer to be a sum of a feed-forward term ϕ and layer-of-origin terms λ. We can then obtain an expression for the weight change by splitting the component expectations in $\langle \Delta W_h \rangle$. We take ϕ_{nk} to be the forward-propagated noise from layer k to layer n and write the general form:

$$\xi_n = \lambda_n + \sum_k \phi_{nk}$$

The weight update expression is:

$$\begin{aligned}
\langle \Delta W_h \rangle &= \alpha \langle (r - \langle r \rangle)(x_h - c_h \langle x_h \rangle)(x_i - c_i \langle x_i \rangle)^T \rangle \\
&= \alpha \langle (2\delta_r^T \xi_r - \zeta_r)(\mu_h + \xi_h - c_h \mu_h)(\mu_i + \xi_i - c_i \mu_i)^T \\
&= \alpha (2\delta_r^T \xi_r - \zeta_r))(\beta_h \beta_i \mu_h \mu_i^T + \beta_i \xi_h \mu_i^T + \beta_h \mu_h \xi_i^T + \xi_h \xi_i^T)
\end{aligned}$$

There are 4 monomials here which depend on reward prediction error. Taking $\beta_h = 0$ and $\beta_i = 1$ and letting the noise be Gaussian (which simplifies the ζ term) gives:

$$\begin{aligned}
\langle \Delta W_h \rangle &= 2\alpha \langle \delta_r^T (\lambda_r + \phi_{rh} + \phi_{ri})(\lambda_h + \phi_{hi}) \mu_i^T \rangle \\
&\quad + 2\alpha \langle (\lambda_r + \phi_{rh} + \phi_{ri})^T (\lambda_r + \phi_{rh} + \phi_{ri}) \rangle \langle (\lambda_h + \phi_{hi})(\lambda_i)^T \rangle
\end{aligned}$$

If we continue with the assumption that the layer-endogenous noise terms are independent, then expectations involving different sending (right) indices are zero. Furthermore, we can consider the case without input noise or endogenous response noise to get:

$$\begin{aligned}
\langle \Delta W_h \rangle &= \langle \delta_r^T \phi_{rh} \lambda_h \mu_i^T \rangle \\
&= \langle \delta_r^T W_r \lambda_h \lambda_h \mu_i^T \rangle \\
&= \langle \lambda_h^T W_r^T \delta_r \lambda_h \mu_i^T \rangle \\
&= \langle \lambda_h \lambda_h^T \rangle W_r^T \delta_r \mu_i^T
\end{aligned}$$

Taking λ_h to be full rank and isotropic makes $\langle \lambda_h \lambda_h^T \rangle = I$, so that we have:

$$\langle \Delta W_h \rangle = W_r^T \delta_r \mu_i^T$$

This provides an alternative derivation of the REINFORCE algorithm, making use of various linear algebraic manipulations in such a way as to clarify how it can be generalized. In particular, the matrix $\langle \lambda_h \lambda_h^T \rangle$ is a noise covariance matrix which, interpreted as a transformation of weight space, operates to project out any dimensions in which it is rank-deficient, and to stretch or compress other dimensions according their variances. The various terms we have neglected along the way can be interpreted according to the same principles - we only neglected them because they are numerous.

If we take $\beta_i = 0$ instead, and ignore the ζ term, we have:

$$\langle \Delta W_h \rangle = 2\alpha \langle \delta_r^T (W_r \lambda_h + \phi_{ri})(\lambda_h + \phi_{hi}) \lambda_i^T \rangle$$

Neglecting the feed-forward impacts of the input noise (as could be instantiated, for example, by having different frequency-based transfer of activity through different cortical layers), we can perform a similar manipulation as above:

$$\begin{aligned} \langle \Delta W_h \rangle &= 2\alpha \langle \delta_r^T W_r \lambda_h \lambda_h \lambda_i^T \rangle \\ &= \langle \lambda_h \lambda_h^T W_r^T \delta_r \lambda_i^T \rangle \end{aligned}$$

Now consider a decomposition of λ_h into components $\eta_{ij} q_i$, and λ_i into components $\gamma_{ij} p_j$, with η_{ij} and γ_{ij} dependent only when the index pairs are the same. Then:

$$\begin{aligned} \langle \Delta W_h \rangle &= 2\alpha \langle \lambda_h \lambda_h^T W_r^T \delta_r \lambda_i^T \rangle \\ &= 2\alpha \sum_{ij} \langle \eta_{ij}^2 \gamma_{ij} \rangle q_i q_i^T W_r^T \delta_r p_j^T \\ &\propto 2\alpha \sum_{ij} \langle \eta_{ij}^2 \gamma_{ij} \rangle (q_i q_i^T) W_r^T \delta_r \mu_i^T (p_j p_j^T) \end{aligned}$$

The last line shows that both input and output filters are now subject to projection, according to $q_i q_i^T$ and $p_j p_j^T$ respectively, and that each projected gradient has it's own learning rate according to the noise yoking feature pairs. Noise can no longer be Gaussian for the $\langle \eta_{ij}^2 \gamma_{ij} \rangle$ term to be non-zero, however. If we let $\gamma_{ij} = \eta_{ij}^2$, the problem of vanishing skewness is resolved in exchange for a violation of the mean-zero requirement. Intuitively, we can resolve this by adding a correction term to the $\beta_i = 0$ baseline. This effectively re-centers the noise, and we calculate it by requiring $x_i - c_i \langle x_i \rangle = \lambda_i = \sum \eta_{ij}^2 p_j$. This gives $c_i = 1 - \sum \langle \eta_{ij}^2 p_j \rangle \oslash \langle x_i \rangle$, or $\beta_i = \sum \langle \eta_{ij}^2 p_j \rangle \oslash \langle x_i \rangle$, and hence $c_i \langle x_i \rangle$ must now also be interpreted as a Hadamard product. ($\oslash$ denotes element-wise division.) Then we find the following, as desired:

$$\begin{aligned} x_i - c_i \langle x_i \rangle &= \mu_i + (\sum \eta_{ij}^2 p_j - \sum \langle \eta_{ij}^2 \rangle p_j) - c_i \mu_i \\ &= \mu_i + (\sum \eta_{ij}^2 p_j - \sum \langle \eta_{ij}^2 \rangle p_j) - (1 - \sum \langle \eta_{ij}^2 \rangle p_j \oslash \mu_i) \odot \mu_i \\ &= \sum \eta_{ij}^2 p_j - \sum \langle \eta_{ij}^2 \rangle p_j + \sum \langle \eta_{ij}^2 \rangle p_j \oslash \mu_i \odot \mu_i \\ &= \sum \eta_{ij}^2 p_j \end{aligned}$$

The first line splits x_i consistently with our previous discussion, as a mean activity plus mean zero noise, the second line substitutes our derived baseline, the third equation results from distributing the Hadamard product of the mean, and the fourth results from cancelling division by multiplication, element-wise.

The result is the weight update we sought in our "$\beta_i = 0$" discussion:

$$\begin{aligned}\langle \Delta W_h \rangle &= \alpha \langle (r - \langle r \rangle)(x_h - c_h \langle x_h \rangle)(x_i - c_i \langle x_i \rangle)^T \rangle \\ &= 2\alpha \langle \delta_r^T W_r \lambda_h \lambda_h \lambda_i^T \rangle + \text{bias} \\ &\approx \langle \lambda_h \lambda_h^T W_r^T \delta_r \lambda_i^T \rangle \\ &\propto 2\alpha \sum_{ij} \langle \eta_{ij}^4 \rangle (q_i q_i^T) W_r^T \delta_r \mu_i^T (p_j p_j^T)\end{aligned}$$

This final line shows that the skewness problem is resolved by the quadratic noise and baseline correction, as claimed. Notably, this was just one route to achieving such an end, however. For example, we could have used a basic rule with a baseline that wasn't multiplied by average activity (i.e., we could have used c_i rather than $c_i \langle x_i \rangle$ in order to side-step this technicality. We developed this particular formulation to show that even without making seemingly unmotivated modifications to the basic rule (as applying absolute values or departing from the traditional mean-baseline might be construed), biological processes could arrive at the same ultimate mathematical description.

Returning briefly to the question of "setting noise", we regard the manipulations we have made as plausibly directed by e.g. top-down sources of layer "endogenous" noise. Such noise could easily be "formatted" to propagate from the hidden layer onward, but not from the input layer to the hidden layer, such as via the frequency dependent transfer noted above. The Hebbian rule must obviously then listen to both noise processes in their given formats. While this is a fairly specific hypothesis, we reiterate that the purpose of this paper is the elaboration of the geometric relations inherent in the Hebbian weight update and their potential uses. A general enumeration of routes to the same filter analyses, or of biological mechanisms which may be suitable for generating these routes, we leave for future work.

4.9 Interference categories from solution geometries

For a task's solution to be reachable by directed exploration along yu^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. Note that there are solvable tasks with essential interference, however. These definitions are illustrated in figure 5.

To be solvable, all component tasks' solution manifolds must intersect. In our case, these manifolds are flat because the networks are linear. This is convenient analytically, and it permits extrapolating local gradient information

into global information, but nonlinear networks are subject to similar qualitative considerations. Here, solution manifolds are induced by 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 thereby improve while moving within the other readouts' kernels. Denoting solution manifolds for each task S_i, these conditions are:

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

The first is the definition of a solution manifold; the output filter associated with input i, denoted S_i, comprises any activation of the hidden layer x_h^{i*} producing zero error, plus 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 5 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 space weight space. This partitions it into regions of constant interference. 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 5B illustrates this.

Given the points above, we ask how noise can sample along interference minimizing dimensions. Prescriptive sampling requires choosing noise dimensions in readout weight kernels, whereas an online approach can simply orthogonalize current noise against all previously encountered gradients (which we also verified in simulation). 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 could tune noise at the network level via their high inter-connectivity. Regardless of how control is accomplished biologically, we make use of oracle kernels for each W_r^i in our simulations here.

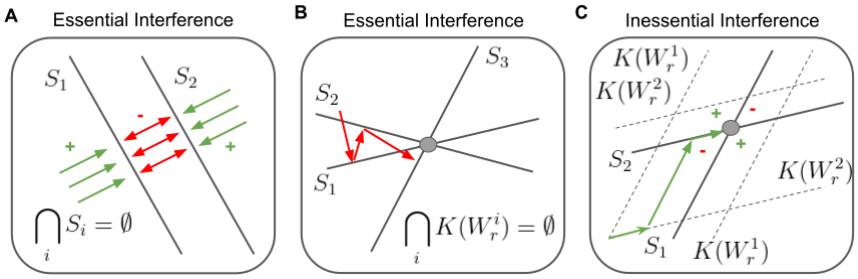

Figure 5: Sketches of interference cases. (A) Tasks which aren't simultaneously solvable interfere in regions of the solution space that lie between their solution manifolds (at minimum). Red arrows denote opposite directions of updates in this region of the plane based on which task is being learned 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.

4.10 Solution manifold geometry

To determine when we have different essential and inessential interference conditions, we inspect the intersection properties of generic flats in Euclidean space (also called affine subspaces, such as lines and planes which may or may not pass through the origin). The flats we consider are the null spaces of the readout weight matrices, which are the spans of all vectors which a given readout matrix sends to zero (i.e. "doesn't care about"). Every kernel 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 $m = [m_1, m_2, ...m_n]$ to count the number of readout kernels of each dimension, starting with 1 and ending with n. That is, 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), then 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:

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

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 the solution. 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.

4.11 Generative model for inessential interference

The simplest form of inessential interference arises when we consider sets of tasks which each have only one input-output pair. In this case, dimensional considerations dictate various quantities. If we let $K_i = \text{Ker}(W_r^i)$, n be the number of dimensions of the hidden layer, and m_k be the number of kernels K_i with dimension k, then for the intersection of the kernels to be non-empty the following inequality must be satisfied:

$$\sum_{k=1}^{n-1} m_k k - (\sum_{k=1}^{n-1} m_k - 1)n \geq 0 \tag{11}$$

To construct a task, we construct W_h^* recursively, making use of random subspace elements $a_i \in \cap_{j \neq i} \text{Ker}(W_r^j)$. If we use a recursion index $l \in \{1, ..., N\}$,

and use a random set of inputs x_l then the procedure is:

$$W_h^*(0) = 0 \tag{12}$$

$$W_h^*(l) = W_h^*(l-1) + a_l x_l^T \tag{13}$$

We then set $t_r^i = W_r^i W_h^* x_i$, which completes the task specification for each $\mathcal{T}^i$.

For illustration, we would like tasks generated using this procedure to have significant amounts of interference, and to have similar squared error magnitudes given an initialization of the weights near zero. To this end, we take the random a_l vectors to be unit length, and the x_l vectors to be distributed randomly over the unit sphere with pairwise correlations of 0.5. This means the x_l vectors are multivariate Gaussian random variables with an appropriate (non-diagonal) covariance matrix, which we subsequently normalize.

4.12 Generative model for compositional tasks

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 accumulating Euclidean basis vectors e_i into vectors v_i according to the composition parameter C and 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 this would yield:

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

Juxtaposition here represents matrix multiplication, and parenthesis denote vectors rather than indexing. Matching the inputs and outputs then indicates that the target weight matrix $W_h* = BA^T$, as shown in the text, because stimuli must be back-transformed into the feature basis for the input and forward-transformed into the euclidean basis for the hidden layer.

Linking numbers were free parameters which we used to generate projectors. The input layer projector 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 (out of all those included in a compositional feature vector). This matrix corresponds to $(p_i p_i^T)$ in the final equation of section 4.4 (describing input projections). A matrix for the outputs, Q was constructed and applied similarly.

4.13 Note on step-size normalization

With respect to the biological (and indeed, in silico) implementations of our results, it is important to note that the reward-modulated Hebbian rules are being used to set relative weight update strengths which are subsequently made intercomparable with gradient updates by fixing step sizes. While synaptic

biology, with its diverse set of molecular dependencies, undoubtedly has enough degrees of freedom to accomplish this, real weight updates are unlikely to be fixed in magnitude. The existing literature on STDP does not appear sufficient for determining how much the fixed step-size approximation matters however. Nonetheless, it is also unlikely that step size depends quadratically on noise, as simple Hebbian algorithms would dictate, because this description corresponds with a discretized dynamical system having finite escape time to infinity. That is, the naive algorithm is also of limited biological realism in the same regard as ours. This of course means that both will deviate from the qualitative properties induced by actual biological constraints to some extent. While these deviations will even produce learning rate distortions, the approximation being made here (and in all Hebbian work) is that they are not induced locally to the homeostatic operating points of real networks.

From a computational point of view, fixed step sizes are highly desirable here. This allows inter-comparability with gradient descent and avoids the pathological dynamics of super-linear state feedback. However, there is no theory-free inter-comparison to be made between gradients and projective updates in 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 is more likely to be descriptive of spatially well-separated pairs of neurons, whereas the Frobenious norm is more likely to be descriptive of neurons 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. Results are qualitatively similar under either norm.

4.14 Note on interference with zero weights

Our interference definitions ideally need to account for the case when a task's weight updates are all zero. When this occurs we can define interference as the constraint correction applied to keep weight motion on the solution manifold for the task with zero gradient. This recapitulates the classical mechanical interpretation of Lagrangian optimization, and should generally be considered

when discussing interference, but it is not especially relevant for our simulations.

4.15 Oracle vs. online sample-based quantities

To demonstrate the advantage of the directional derivative approach, in the main text we used algorithmically computed gradients and gradient projections (i.e., based on an oracle), which most cleanly assess how noise covariance and Hebbian rules can be productively combined. Because real learning scenarios may or may not have access to these projections *a priori*, we verified the consistency of our results using numerical experiments with sample-based gradients and with online computations of noise covariance. This was accomplished by computing coloring matrices from the desired noise covariances, transforming Gaussian white noise appropriately, and using the equations developed throughout the text to sample Hebbian input and output filters on a trial-by-trial basis. Technically speaking, we replaced trial level computations performed with oracles by block-accumulation loops of "sub-trials" which accumulated Hebbian weight updates. In theory, such "sub-trials" could be promoted to "trials", accumulated with a causal kernel such as an exponential moving average, and interleaved (such that the task ceased to be sequential) without changing our results. We chose the present implementation because (1) it removes complications such as overlapping kernel time constants and (2) we are naturally concerned with sequential tasks, which manifest interference in training error over and above that in interleaved tasks. 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. Oracle covariance matrices were replaced with online adaptive versions which only orthogonalized noise according to previously encountered tasks (i.e., had no foreknowledge). Simulation results recapitulate the findings in the main text with only minor quantitative differences.

Specifically, to 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 as in simulation 1 / figure 2 from the text. Average differences between the oracle algorithm and the sample based algorithm were 0.5% +- 0.007% (mean +- SEM) of cumulative error for gradient filter accumulation, and 0.5% +- 0.004% (mean +- SEM) for projective filter accumulation, using an accumulation loop of 1000 "sub-trials". These along with other diagnostics are shown in figures below. Lastly, we produced a version of figure 2B below without the use of an oracle covariance matrix. This shows a 4% point decrease in the ensemble mean improvement, and indicates that the basic elements of our results are not especially sensitive to whether noise dimension is set online or proactively.

In future work, we intend to provide a detailed examination of the sample complexity based learning trade off which arises from the simplicity of obtaining accurate low-dimensional estimators relative to high-dimensional ones.

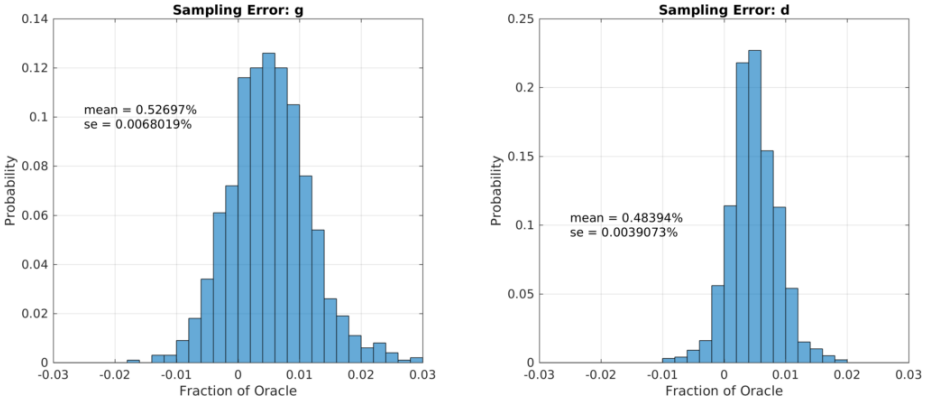

Figure 6: Sample-based cumulative simulation error relative to oracle cumulative simulation error for gradient (g) and projective (d) output filter construction. Converting to and from the use of a sample based oracle introduces negligible differences in simulation results, given a-priori reasonable numbers of samples for gradient estimation via the Hebbian updates, here 1k samples per gradient. Average error induced in results such as those presented in figure 1 is less than 1%.

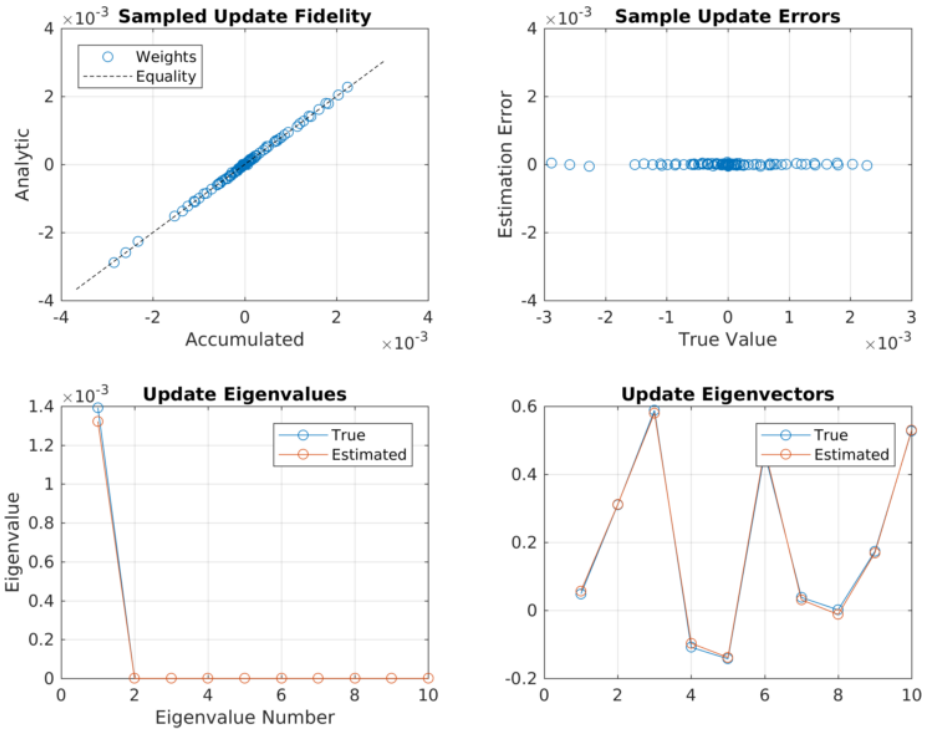

Figure 7: Diagnostics comparing a random weight update based on oracle computation with the same update based on a Hebbian sample accumulation, for simulation 1. Because the input filter does not change, one only expects the output filter to be distorted by sampling, and the eigendecomposition should reproduce this vector (since the column space is one dimensional). Nonetheless we computed the eigendecomposition because technically the output filters are not accumulated in a disaggregated way 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 in simulation 1, but a negligible fraction of total weight change, and therefore had no impact, as suggested by figure 6.

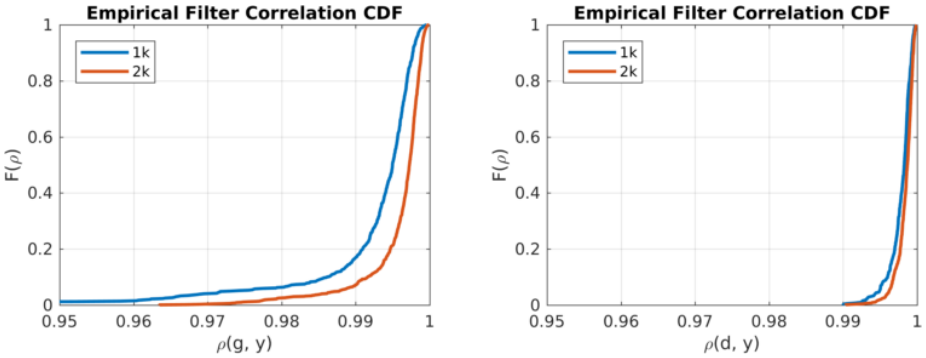

Figure 8: Empirical filter correlation CDFs comparing sample and oracle based output filters, for the weights contributing to 95% of the total weight change. The least important 5% are excluded because they are essentially zero, and are therefore relatively unconstrained in addition to being unimportant. Correlations are extremely high, in agreement with figures 6 and 7. 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.

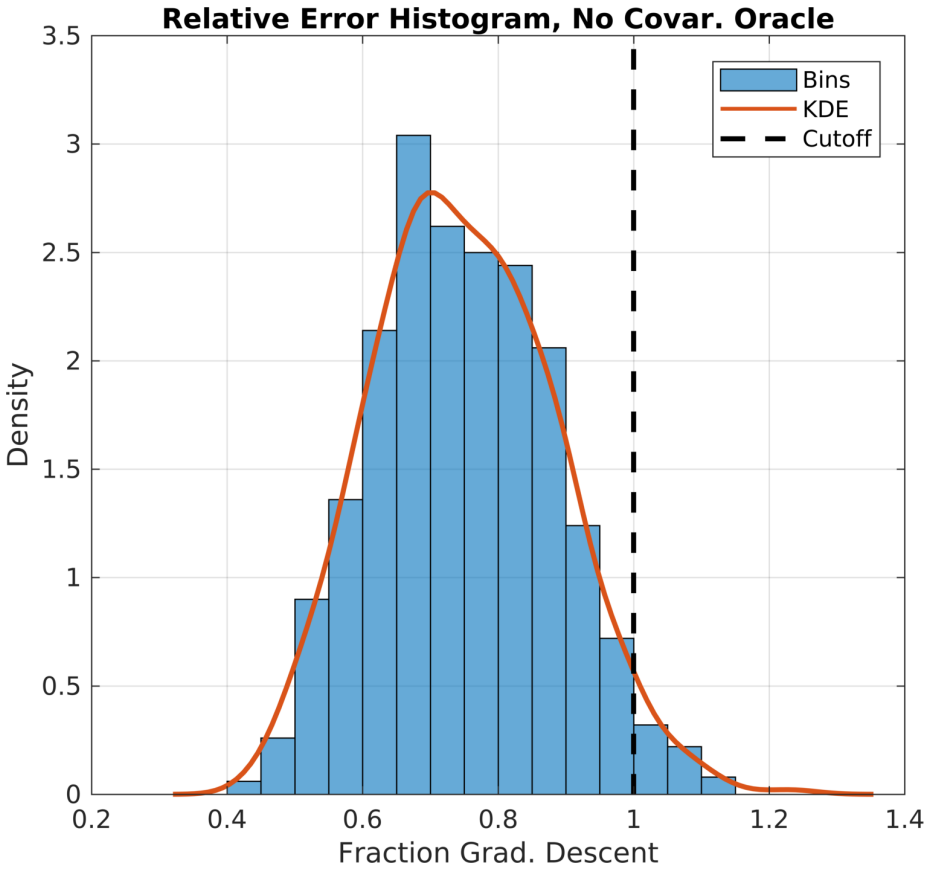

Figure 9: Figure 2B, recomputed without the use of a covariance oracle. That is, task noise covariances were computed online. The mean is 4% points higher than the equivalent panel in the text.

References

Ackley, D. H., Hinton, G. E., & Sejnowski, T. J. (1985). A learning algorithm for boltzmann machines* [_eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1207/s15516709cog0901 *Cognitive Science*, *9*(1), 147–169. https://doi.org/10.1207/s15516709cog0901_7

Cragg, S. J. (2006). Meaningful silences: How dopamine listens to the ACh pause. *Trends in Neurosciences*, *29*(3), 125–131. https://doi.org/10.1016/j.tins.2006.01.003

Fino, E., Packer, A. M., & Yuste, R. (2012). The logic of inhibitory connectivity in the neocortex: [Publisher: SAGE PublicationsSage CA: Los Angeles, CA]. *The Neuroscientist*. https://doi.org/10.1177/1073858412456743

Fino, E., & Yuste, R. (2011). Dense inhibitory connectivity in neocortex [Publisher: Elsevier]. *Neuron*, *69*(6), 1188–1203. https://doi.org/10.1016/j.neuron.2011.02.025

Frank, M. J. [Michael J.]. (2005). Dynamic dopamine modulation in the basal ganglia: A neurocomputational account of cognitive deficits in medicated and nonmedicated parkinsonism. *Journal of Cognitive Neuroscience*, *17*(1), 51–72. https://doi.org/10.1162/0898929052880093

Frank, M. J. [Michael J.], & Badre, D. (2012). Mechanisms of hierarchical reinforcement learning in corticostriatal circuits 1: Computational analysis. *Cerebral Cortex*, *22*(3), 509–526. https://doi.org/10.1093/cercor/bhr114

Franklin, N. T., & Frank, M. J. [Michael J]. (2015). A cholinergic feedback circuit to regulate striatal population uncertainty and optimize reinforcement learning (U. S. Bhalla, Ed.) [Publisher: eLife Sciences Publications, Ltd]. *eLife*, *4*, e12029. https://doi.org/10.7554/eLife.12029

Gibson, J. R., Beierlein, M., & Connors, B. W. (1999). Two networks of electrically coupled inhibitory neurons in neocortex [Number: 6757 Publisher: Nature Publishing Group]. *Nature*, *402*(6757), 75–79. https://doi.org/10.1038/47035

Karhunen, J. (1984). Adaptive algorithms for estimating eigenvectors of correlation type matrices. *ICASSP '84. IEEE International Conference on Acoustics, Speech, and Signal Processing*, *9*, 592–595. https://doi.org/10.1109/ICASSP.1984.1172323

King, P. D., Zylberberg, J., & DeWeese, M. R. (2013). Inhibitory interneurons decorrelate excitatory cells to drive sparse code formation in a spiking model of v1 [Publisher: Society for Neuroscience Section: Articles]. *Journal of Neuroscience*, *33*(13), 5475–5485. https://doi.org/10.1523/JNEUROSCI.4188-12.2013

Krotov, D., & Hopfield, J. J. (2019). Unsupervised learning by competing hidden units [Publisher: National Academy of Sciences Section: PNAS Plus]. *Proceedings of the National Academy of Sciences*, *116*(16), 7723–7731. https://doi.org/10.1073/pnas.1820458116

Law, C.-T., & Gold, J. I. (2009). Reinforcement learning can account for associative and perceptual learning on a visual-decision task [Number: 5

Publisher: Nature Publishing Group]. *Nature Neuroscience*, *12*(5), 655–663. https://doi.org/10.1038/nn.2304

Morris, G., Arkadir, D., Nevet, A., Vaadia, E., & Bergman, H. (2004). Coincident but distinct messages of midbrain dopamine and striatal tonically active neurons [Publisher: Elsevier]. *Neuron*, *43*(1), 133–143. https://doi.org/10.1016/j.neuron.2004.06.012

Nassar, M. R., Scott, D., & Bhandari, A. (2021). Noise correlations for faster and more robust learning. *The Journal of Neuroscience*, *41*(31), 6740–6752. https://doi.org/10.1523/JNEUROSCI.3045-20.2021

Niv, Y., Daniel, R., Geana, A., Gershman, S. J., Leong, Y. C., Radulescu, A., & Wilson, R. C. (2015). Reinforcement learning in multidimensional environments relies on attention mechanisms [Publisher: Society for Neuroscience Section: Articles]. *Journal of Neuroscience*, *35*(21), 8145–8157. https://doi.org/10.1523/JNEUROSCI.2978-14.2015

Oja, E. (1982). Simplified neuron model as a principal component analyzer. *Journal of Mathematical Biology*, *15*(3), 267–273. https://doi.org/10.1007/BF00275687

O'Reilly, R. C. (1996). Biologically plausible error-driven learning using local activation differences: The generalized recirculation algorithm [Place: US Publisher: MIT Press]. *Neural Computation*, *8*(5), 895–938. https://doi.org/10.1162/neco.1996.8.5.895

O'reilly, R. C. (2001). Generalization in interactive networks: The benefits of inhibitory competition and hebbian learning. *Neural Computation*, *13*, 1199–1242.

O'Reilly, R. C., Munakata, Y., Frank, M. J., Hazy, T. E., & Contributors. (2012). *Computational cognitive neuroscience*. Online Book, 4th Edition, URL: https://CompCogNeuro.org. https://github.com/CompCogNeuro/ed4

Pi, H.-J., Hangya, B., Kvitsiani, D., Sanders, J. I., Huang, Z. J., & Kepecs, A. (2013). Cortical interneurons that specialize in disinhibitory control [Number: 7477 Publisher: Nature Publishing Group]. *Nature*, *503*(7477), 521–524. https://doi.org/10.1038/nature12676

Ruan, H., Saur, T., & Yao, W.-D. (2014). Dopamine-enabled anti-hebbian timing-dependent plasticity in prefrontal circuitry [Publisher: Frontiers]. *Frontiers in Neural Circuits*, *8*. https://doi.org/10.3389/fncir.2014.00038

Rumelhart, D. E., & Zipser, D. (1985). Feature discovery by competitive learning* [_eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1207/s15516709cog0901_5]. *Cognitive Science*, *9*(1), 75–112. https://doi.org/10.1207/s15516709cog0901_5

Sanger, T. D. (1989). Optimal unsupervised learning in a single-layer linear feedforward neural network. *Neural Networks*, *2*(6), 459–473. https://doi.org/10.1016/0893-6080(89)90044-0

Shen, W., Flajolet, M., Greengard, P., & Surmeier, D. J. (2008). Dichotomous dopaminergic control of striatal synaptic plasticity [Publisher: American Association for the Advancement of Science]. *Science*, *321*(5890), 848–851. https://doi.org/10.1126/science.1160575

Stalnaker, T. A., Berg, B., Aujla, N., & Schoenbaum, G. (2016). Cholinergic interneurons use orbitofrontal input to track beliefs about current state [Publisher: Society for Neuroscience Section: Articles]. *Journal of Neuroscience*, *36*(23), 6242–6257. https://doi.org/10.1523/JNEUROSCI.0157-16.2016

Voelcker, B., & Peron, S. (2021, September 17). *Transformation of primary sensory cortical representations from layer 4 to layer 2* (preprint). Neuroscience. https://doi.org/10.1101/2021.09.17.460780

Wang, J. X., Kurth-Nelson, Z., Kumaran, D., Tirumala, D., Soyer, H., Leibo, J. Z., Hassabis, D., & Botvinick, M. (2018). Prefrontal cortex as a meta-reinforcement learning system. *Nature Neuroscience*, *21*(6), 860–868. https://doi.org/10.1038/s41593-018-0147-8

Yagishita, S., Hayashi-Takagi, A., Ellis-Davies, G. C., Urakubo, H., Ishii, S., & Kasai, H. (2014). A critical time window for dopamine actions on the structural plasticity of dendritic spines [Publisher: American Association for the Advancement of Science]. *Science*, *345*(6204), 1616–1620. https://doi.org/10.1126/science.1255514

Yavorska, I., & Wehr, M. (2016). Somatostatin-expressing inhibitory interneurons in cortical circuits [Publisher: Frontiers]. *Frontiers in Neural Circuits*, *10*. https://doi.org/10.3389/fncir.2016.00076