Adaptive Whitening in Neural Populations with Gain-modulating Interneurons

Lyndon R. Duong [*1] **David Lipshutz** [*2] **David J. Heeger** [1] **Dmitri B. Chklovskii** [2 3] **Eero P. Simoncelli** [1 2]

Abstract

Statistical whitening transformations play a fundamental role in many computational systems, and may also play an important role in biological sensory systems. Existing neural circuit models of adaptive whitening operate by modifying synaptic interactions; however, such modifications would seem both too slow and insufficiently reversible. Motivated by the extensive neuroscience literature on gain modulation, we propose an alternative model that adaptively whitens its responses by modulating the gains of individual neurons. Starting from a novel whitening objective, we derive an online algorithm that whitens its outputs by adjusting the marginal variances of an *overcomplete* set of projections. We map the algorithm onto a recurrent neural network with fixed synaptic weights and gain-modulating interneurons. We demonstrate numerically that sign-constraining the gains improves robustness of the network to ill-conditioned inputs, and a generalization of the circuit achieves a form of local whitening in convolutional populations, such as those found throughout the visual or auditory systems.

1. Introduction

Statistical whitening transformations, in which multi-dimensional inputs are decorrelated and normalized to have unit variance, are common in signal processing and machine learning systems. For example, they are integral to many statistical factorization methods (Olshausen & Field, 1996; Bell & Sejnowski, 1996; Hyvärinen & Oja, 2000), they provide beneficial preprocessing during neural network training (Krizhevsky, 2009), and they can improve unsupervised feature learning (Coates et al., 2011). More recently, self-supervised learning methods have used decorrelation transformations such as whitening to prevent representational collapse (Ermolov et al., 2021; Zbontar et al., 2021; Hua et al., 2021; Bardes et al., 2022). While whitening has mostly been used for training neural networks in the offline setting, it is also of interest to develop adaptive (run-time) variants that can adjust to dynamically changing input statistics with minimal changes to the network (e.g. Mohan et al., 2021; Hu et al., 2022).

Single neurons in early sensory areas of many nervous systems rapidly adjust to changes in input statistics by scaling their input-output gains (Adrian & Matthews, 1928). This allows neurons to adaptively normalize the variance of their outputs (Bonin et al., 2006; Nagel & Doupe, 2006), maximizing information transmitted about sensory inputs (Barlow, 1961; Laughlin, 1981; Fairhall et al., 2001). At the neural *population* level, in addition to variance normalization, adaptive decorrelation and whitening transformations have been observed across species and sensory modalities, including: macaque retina (Atick & Redlich, 1992); cat primary visual cortex (Muller et al., 1999; Benucci et al., 2013); and the olfactory bulbs of zebrafish (Friedrich, 2013) and mice (Giridhar et al., 2011; Gschwend et al., 2015). These population-level adaptations reduce redundancy in addition to normalizing neuronal outputs, facilitating *dynamic* efficient multi-channel coding (Schwartz & Simoncelli, 2001; Barlow & Foldiak, 1989). However, the mechanisms underlying such adaptive whitening transformations remain unknown, and would seem to require coordinated synaptic adjustments amongst neurons, as opposed to the single neuron case which relies only on gain rescaling.

Here, we propose a novel recurrent network architecture for online statistical whitening that exclusively relies on gain modulation. Specifically, the primary contributions of our study are as follows:

1. We introduce a novel factorization of the (inverse) whitening matrix, using an *overcomplete, arbitrary, but fixed* basis, and a diagonal matrix with statistically optimized entries. This is in contrast with the conventional factorization using the eigendecomposition of the input covariance matrix.

[*] Equal contribution [1] Center for Neural Science, New York University; [2] Center for Computational Neuroscience, Flatiron Institute; [3] Neuroscience Institute, NYU School of Medicine. Correspondence to: Lyndon R. Duong <lyndon.duong@nyu.edu>, David Lipshutz <dlipshutz@flatironinstitute.org>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).

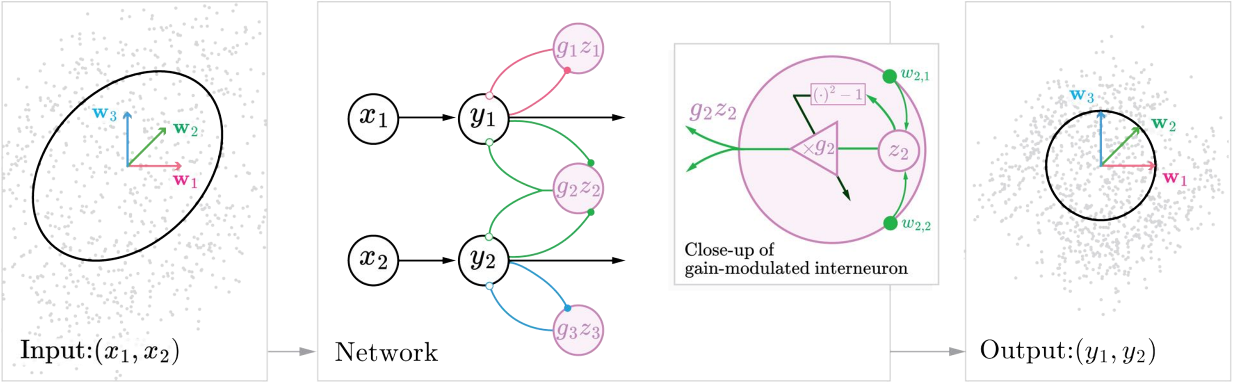

Figure 1. Schematic of a recurrent statistical whitening network with 2 primary neurons and 3 interneurons. **Left**: 2D Scatter plot of network inputs $\mathbf{x} = [x_1, x_2]^\top$ (e.g. post-synaptic currents), with covariance indicated by the ellipse. **Center**: Primary neurons, with outputs $\mathbf{y} = [y_1, y_2]^\top$, receive external feedforward inputs, $\mathbf{x}$, and recurrent feedback from an overcomplete population of interneurons, $-\sum_{i=1}^{3} g_i z_i \mathbf{w}_i$. Projection vectors $\{\mathbf{w}_1, \mathbf{w}_2, \mathbf{w}_3\} \in \mathbb{R}^2$ encode feedforward synaptic weights connecting primary neurons to interneuron $i = 1, 2, 3$, with *symmetric* feedback connections. Weight vectors are shown in the left and right panels with corresponding colors. In general, the network may require all-to-all connectivity between primary and interneurons; we use a reduced subset of connections here for diagram clarity. **Inset:** The i^{th} interneuron (e.g. here $i = 2$) receives input $z_i = \mathbf{w}_i^\top \mathbf{y}$, which is multiplied by its gain g_i to produce output $g_i z_i$. Its gain, g_i, is adjusted s.t. $\Delta g_i \propto z_i^2 - 1$. The dark arrow indicates that the gain update operates on a slower time scale. **Right:** Scatter plots of the whitened network outputs $\mathbf{y}$. Outputs have unit variance along all $\mathbf{w}_i$'s, which is equivalent to having identity covariance matrix, i.e., $\mathbf{C}_{yy} = \mathbf{I}_N$ (black circle).

2. We introduce an unsupervised online learning objective using this factorization to express the whitening objective solely in terms of the *marginal* variances within the overcomplete representation of the input signal.

3. We derive a recursive algorithm to optimize the objective, and show that it corresponds to an unsupervised recurrent neural network (RNN), comprised of primary neurons and an auxiliary overcomplete population of interneurons, whose synaptic weights are fixed, but whose gains are adaptively modulated. The network responses converge to the classical symmetric whitening solution without backpropagation.

4. We show how enforcing non-negativity on the gain modulation provides a novel approach for dealing with ill-conditioned or noisy data. Further, we relax the global whitening constraint in our objective and provide a method for *local* decorrelation of convolutional neural populations.

2. A Novel Objective for Symmetric Whitening

Consider a neural network with N primary neurons. For each $t = 1, 2, \dots$, let $\mathbf{x}_t$ and $\mathbf{y}_t$ be N-dimensional vectors whose components respectively denote the inputs (e.g. post-synaptic currents), and outputs of the primary neurons at time t (Figure 1). Without loss of generality, we assume the inputs $\mathbf{x}_t$ are centered.

2.1. Conventional objective

Statistical whitening aims to linearly transform inputs $\mathbf{x}_t$ so that the covariance of the outputs $\mathbf{y}_t$ is the identity, i.e.,

$$\mathbf{C}_{yy} = \langle \mathbf{y}_t \mathbf{y}_t^\top \rangle_t = \mathbf{I}_N, \tag{1}$$

where $\langle \cdot \rangle_t$ denotes the expectation operator over t, and $\mathbf{I}_N$ denotes the $N \times N$ identity matrix (see Appendix A for a list of notation used in this work).

It is well known that whitening is not unique: any orthogonal rotation of a random vector with identity covariance matrix also has identity covariance matrix. There are several common methods of resolving this rotational ambiguity, each with their own advantages (Kessy et al., 2018). Here, we focus on the symmetric whitening transformation, often referred to as Zero-phase Component Analysis (ZCA) whitening or Mahalanobis whitening, which minimizes the mean-squared error between the inputs and the whitened outputs (alternatively, the one whose transformation matrix is symmetric). The symmetric whitened outputs are the optimal solution to the minimization problem

$$\min_{\{\mathbf{y}_t\}} \langle \|\mathbf{x}_t - \mathbf{y}_t\|_2^2 \rangle_t \quad \text{s.t.} \quad \langle \mathbf{y}_t \mathbf{y}_t^\top \rangle_t = \mathbf{I}_N, \tag{2}$$

where $\|\cdot\|_2$ denotes the Euclidean norm on $\mathbb{R}^N$. Assuming the covariance of the inputs $\mathbf{C}_{xx} := \langle \mathbf{x}_t \mathbf{x}_t^\top \rangle_t$ is positive definite, the unique solution to the optimization problem in Equation 2 is $\mathbf{y}_t = \mathbf{C}_{xx}^{-1/2} \mathbf{x}_t$ for $t = 1, 2, \dots$, where

$\mathbf{C}_{xx}^{-1/2}$ is the symmetric inverse matrix square root of $\mathbf{C}_{xx}$ (see Appendix B).

Previous approaches to *online* symmetric whitening have optimized Equation 2 by deriving RNNs whose *synaptic weights* adaptively adjust to learn the eigendecomposition of the (inverse) whitening matrix, $\mathbf{C}_{xx}^{1/2} = \mathbf{V}\mathbf{\Lambda}^{1/2}\mathbf{V}^\top$, where $\mathbf{V}$ is an orthogonal matrix of eigenvectors and $\mathbf{\Lambda}$ is a diagonal matrix of eigenvalues (Pehlevan & Chklovskii, 2015). We propose an entirely different decomposition: $\mathbf{C}_{xx}^{1/2} = \mathbf{W}\,\mathrm{diag}\,(\mathbf{g})\,\mathbf{W}^\top + \mathbf{I}_N$, where $\mathbf{W}$ is a *fixed* overcomplete matrix of synaptic weights, and $\mathbf{g}$ is a vector of *gains* that adaptively adjust to match the whitening matrix.

2.2. A novel objective using *marginal* statistics

We formulate an objective for learning the symmetric whitening transform via gain modulation. Our innovation exploits the fact that a random vector has identity covariance matrix (i.e., Equation 1 holds) if and only if it has unit marginal variance along *all possible 1D projections* (a form of tomography; see Related Work). We derive a tighter statement for a finite but *overcomplete* set of at least $K \geq K_N := N(N+1)/2$ distinct axes ('overcomplete' means that the number of axes exceeds the dimensionality of the input, i.e., $K > N$). Intuitively, this equivalence holds because an $N \times N$ symmetric matrix has K_N degrees of freedom, so the marginal variances along $K \geq K_N$ distinct axes are sufficient to constrain an $N \times N$ covariance matrix. We formalize this equivalence in the following proposition, whose proof is provided in Appendix C.

Proposition 2.1. *Fix $K \geq K_N$. Suppose $\mathbf{w}_1, \dots, \mathbf{w}_K \in \mathbb{R}^N$ are unit vectors*[1] *such that*

$$\mathit{span}(\{\mathbf{w}_1\mathbf{w}_1^\top, \dots, \mathbf{w}_K\mathbf{w}_K^\top\}) = \mathbb{S}^N, \tag{3}$$

where $\mathbb{S}^N$ denotes the K_N-dimensional vector space of $N \times N$ symmetric matrices. Then Equation 1 holds if and only if the projection of $\mathbf{y}_t$ onto each unit vector $\mathbf{w}_1, \dots, \mathbf{w}_K$ has unit variance, i.e.,

$$\langle(\mathbf{w}_i^\top \mathbf{y}_t)^2\rangle_t = 1 \quad \mathit{for} \quad i = 1, \dots, K. \tag{4}$$

Assuming Equation 3 holds, we can interpret the set of vectors $\{\mathbf{w}_1, \dots, \mathbf{w}_K\}$ as a *frame* (i.e., an overcomplete basis; Casazza et al., 2013) in $\mathbb{R}^N$ such that the covariance of the outputs $\mathbf{C}_{yy}$ can be computed from the variances of the K-dimensional projection of the outputs onto the set of frame vectors. Thus, we can replace the whitening constraint in Equation 2 with the equivalent *marginal variance* constraint to obtain the following objective:

$$\min_{\{\mathbf{y}_t\}} \langle \|\mathbf{x}_t - \mathbf{y}_t\|_2^2 \rangle_t \quad \text{s.t.} \quad \text{Equation 4 holds.} \tag{5}$$

3. An RNN with Gain Modulation for Adaptive Symmetric Whitening

In this section, we derive an online algorithm for solving the optimization problem in Equation 5 and map the algorithm onto an RNN with adaptive gain modulation. Assume we have an overcomplete frame $\{\mathbf{w}_1, \dots, \mathbf{w}_K\}$ in $\mathbb{R}^N$ satisfying Equation 3. We concatenate the frame vectors into an $N \times K$ synaptic weight matrix $\mathbf{W} := [\mathbf{w}_1, \dots, \mathbf{w}_K]$. In our network, primary neurons project onto a layer of K interneurons via the synaptic weight matrix to produce the K-dimensional vector $\mathbf{z}_t := \mathbf{W}^\top \mathbf{y}_t$, encoding the interneurons' post-synaptic inputs at time t (Figure 1). We emphasize that the synaptic weight matrix $\mathbf{W}$ remains *fixed*.

3.1. Enforcing the marginal variance constraints with scalar gains

We introduce Lagrange multipliers $g_1, \dots, g_K \in \mathbb{R}$ to enforce the K constraints in Equation 4. These are concatenated as the entries of a K-dimensional vector $\mathbf{g} := [g_1, \dots, g_K]^\top \in \mathbb{R}^K$, and express the whitening objective as a saddle point optimization:

$$\max_{\mathbf{g}} \min_{\{\mathbf{y}_t\}} \langle \ell(\mathbf{x}_t, \mathbf{y}_t, \mathbf{g}) \rangle_t, \tag{6}$$

$$\text{where } \ell(\mathbf{x}, \mathbf{y}, \mathbf{g}) := \|\mathbf{x} - \mathbf{y}\|_2^2 + \sum_{i=1}^K g_i \left\{ (\mathbf{w}_i^\top \mathbf{y})^2 - 1 \right\}.$$

Here, we have exchanged the order of maximization over $\mathbf{g}$ and minimization over $\mathbf{y}_t$, which is justified because $\ell(\mathbf{x}_t, \mathbf{y}_t, \mathbf{g})$ satisfies the saddle point property with respect to $\mathbf{y}$ and $\mathbf{g}$, see Appendix E.

In our RNN implementation, there are K interneurons and g_i corresponds to the multiplicative gain associated with the i^{th} interneuron, so that its output at time t is $g_i z_{i,t}$ (Figure 1, Inset). Equation 6, shows that the gain of the i^{th} interneuron, g_i, encourages the marginal variance of $\mathbf{y}_t$ along the axis spanned by $\mathbf{w}_i$ to be unity. Importantly, the gains are not hyper-parameters, but rather they are optimization variables which statistically whiten the outputs $\{\mathbf{y}_t\}$, preventing the neural outputs from trivially matching the inputs $\{\mathbf{x}_t\}$.

3.2. Deriving RNN neural dynamics and gain updates

To solve Equation 6 in the online setting, we assume there is a time-scale separation between 'fast' neural dynamics and 'slow' gain updates, so that at each time step the neural dynamics equilibrate before the gains are adjusted. This allows us to perform the inner minimization over $\{\mathbf{y}_t\}$ before the outer maximization over the gains $\mathbf{g}$. This is consistent with biological networks in which a given neuron's responses operate on a much faster time-scale than its intrinsic input-output gain, which is driven by slower processes such as changes in Ca^{2+} concentration gradients and Na^{+}-activated

[1] The unit-length assumption is imposed, without loss of generality, for notational convenience.

K+ channels (Wang et al., 2003; Ferguson & Cardin, 2020).

3.2.1. Fast neural activity dynamics

For each time step $t = 1, 2, \dots$, we minimize the objective $\ell(\mathbf{x}_t, \mathbf{y}_t, \mathbf{g})$ over $\mathbf{y}_t$ by recursively running gradient-descent steps to equilibrium:

$$\begin{aligned}\mathbf{y}_t &\leftarrow \mathbf{y}_t - \frac{\gamma}{2}\nabla_{\mathbf{y}}\ell(\mathbf{x}_t, \mathbf{y}_t(\tau), \mathbf{g}) \\ &= \mathbf{y}_t + \gamma\left\{\mathbf{x}_t - \mathbf{W}(\mathbf{g}\circ\mathbf{z}_t) - \mathbf{y}_t\right\}, \qquad (7)\end{aligned}$$

where $\gamma > 0$ is a small constant, $\mathbf{z}_t = \mathbf{W}^\top\mathbf{y}_t$, the circle '$\circ$' denotes the Hadamard (element-wise) product, $\mathbf{g}\circ\mathbf{z}_t$ is a vector of K gain-modulated interneuron outputs, and we assume the primary cell outputs are initialized at zero.

We see from the right-hand-side of Equation 7 that the 'fast' dynamics of the primary neurons are driven by three terms (within the curly braces): 1) constant feedforward external input $\mathbf{x}_t$; 2) recurrent gain-modulated feedback from interneurons $-\mathbf{W}(\mathbf{g}\circ\mathbf{z}_t)$; and 3) a leak term $-\mathbf{y}_t$. Because the neural activity dynamics are linear, we can analytically solve for their equilibrium (i.e. steady-state), $\bar{\mathbf{y}}_t$, by setting the update in Equation 7 to zero:

$$\begin{aligned}\bar{\mathbf{y}}_t &= \left[\mathbf{I}_N + \mathbf{W}\operatorname{diag}(\mathbf{g})\,\mathbf{W}^\top\right]^{-1}\mathbf{x}_t \\ &= \left[\mathbf{I}_N + \sum_{i=1}^{K} g_i\mathbf{w}_i\mathbf{w}_i^\top\right]^{-1}\mathbf{x}_t, \qquad (8)\end{aligned}$$

where $\operatorname{diag}(\mathbf{g})$ denotes the $K\times K$ diagonal matrix whose $(i,i)^{\text{th}}$ entry is g_i, for $i = 1, \dots, K$. The equilibrium feedforward interneuron inputs are then given by

$$\bar{\mathbf{z}}_t = \mathbf{W}^\top\bar{\mathbf{y}}_t. \qquad (9)$$

The gain-modulated outputs of the K interneurons, $\mathbf{g}\circ\mathbf{z}_t$, are then projected back onto the primary cells via symmetric weights, $-\mathbf{W}$ (Figure 1). After $\mathbf{g}$ adapts to optimize Equation 6 (provided Proposition 2.1 holds), the matrix within the brackets in Equation 8 will equal $\mathbf{C}_{xx}^{1/2}$, and the circuit's equilibrium responses are symmetrically whitened. The result is a novel *overcomplete* symmetric matrix factorization in which $\mathbf{W}$ is arbitrary and fixed, while $\mathbf{C}_{xx}^{1/2}$ is adaptively learned and encoded in the gains $\mathbf{g}$.

3.2.2. Slow gain dynamics

After the fast neural activities reach steady-state, the interneuron gains are updated with a stochastic gradient-ascent step with respect to $\mathbf{g}$:

$$\begin{aligned}\mathbf{g} &\leftarrow \mathbf{g} + \frac{\eta}{2}\nabla_{\mathbf{g}}\ell(\mathbf{x}_t, \bar{\mathbf{y}}_t, \mathbf{g}) \\ &= \mathbf{g} + \eta\left(\bar{\mathbf{z}}_t^{\circ 2} - \mathbf{1}\right), \qquad (10)\end{aligned}$$

where $\eta > 0$ is the learning rate, $\bar{\mathbf{z}}_t^{\circ 2} = [\bar{z}_{t,1}^2, \dots, \bar{z}_{t,K}^2]^\top$, and $\mathbf{1} = [1, \dots, 1]^\top$ is the K-dimensional vector of ones[2]. Remarkably, the update to the i^{th} interneuron's gain g_i (Equation 10) depends only on the online estimate of the *variance* of its equilibrium input $\bar{z}_{t,i}^2$, and its distance from 1 (i.e. the target variance). Since the interneurons adapt using local signals, this circuit is a suitable candidate for hardware implementations using low-power neuromorphic chips (Pehlevan & Chklovskii, 2019). Intuitively, each interneuron adjusts its gain to modulate the amount of suppressive (inhibitory) feedback onto the joint primary neuron responses. In Appendix D, we provide conditions under which $\mathbf{g}$ can be solved analytically. Thus, while statistical whitening inherently involves a transformation on a joint density, our solution operates solely using single neuron gain changes in response to *marginal* statistics of the joint density.

3.2.3. Online unsupervised algorithm

By combining Equations 7 and 10, we arrive at our online RNN algorithm for adaptive whitening via gain modulation (Algorithm 1). We also provide batched and offline versions of the algorithm in Appendix G.

Algorithm 1 Adaptive whitening via gain modulation

```
1:  Input: Centered inputs $\mathbf{x}_1, \mathbf{x}_2, \dots \in \mathbb{R}^N$
2:  Initialize: $\mathbf{W} \in \mathbb{R}^{N\times K}$; $\mathbf{g} \in \mathbb{R}^K$; $\eta, \gamma > 0$
3:  for $t = 1, 2, \dots$ do
4:      $\mathbf{y}_t \leftarrow \mathbf{0}$
5:      while not converged do
6:          $\mathbf{z}_t \leftarrow \mathbf{W}^\top\mathbf{y}_t$
7:          $\mathbf{y}_t \leftarrow \mathbf{y}_t + \gamma\left\{\mathbf{x}_t - \mathbf{W}(\mathbf{g}\circ\mathbf{z}_t) - \mathbf{y}_t\right\}$
8:      end while
9:      $\mathbf{g} \leftarrow \mathbf{g} + \eta\left(\mathbf{z}_t^{\circ 2} - \mathbf{1}\right)$
10: end for
```

There are two points worth noting about this network: 1) $\mathbf{W}$ remains *fixed* in Algorithm 1. Instead, $\mathbf{g}$ adapts to statistically whiten the outputs. 2) In practice, since network dynamics are linear, we can bypass the inner loop (the fast dynamics of the primary cells, lines 5–8), by directly computing $\bar{\mathbf{y}}_t$, and $\bar{\mathbf{z}}_t$ (Eqs. 8, 9).

4. Numerical Experiments and Applications

We provide different applications of our adaptive symmetric whitening network via gain modulation, emphasizing that gain adaptation is distinct from, and *complementary to*, synaptic weight learning (i.e. learning $\mathbf{W}$). We therefore side-step the goal of learning the frame $\mathbf{W}$, and assume it is fixed (for example, through longer time scale learning). This allows us to decouple and analyze the general properties of

[2] Appendix F generalizes the gain update to allowing for temporal-weighted averaging of the variance over past samples.

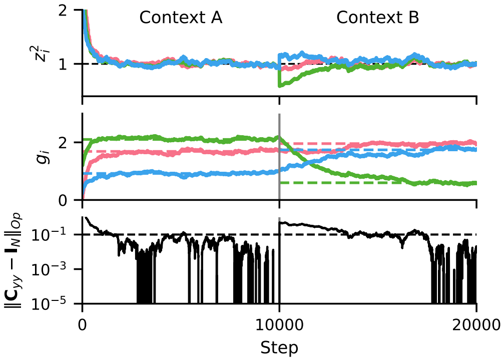

Figure 2. Network from Figure 1 (with corresponding colors; N=2, K=K_N=3, η=2E-3) adaptively whitening samples from two randomly generated statistical contexts online (10K steps each). **Top:** Marginal variances measured by interneurons approach 1 over time. **Middle:** Dynamics of interneuron gains, which are applied to z_i before feeding back onto the primary cells. Dashed lines are optimal gains (Appendix D). **Bottom:** Error over time, as measured by the maximal difference between the standard deviation along the principal axes of $\mathbf{C}_{yy}$ and unity.

our proposed gain modulation framework, independently of the choice of frame. Python code for this study can be located at `github.com/lyndond/frame_whitening`.

We evaluate the performance of our adaptive whitening algorithm using the matrix operator norm, $\|\cdot\|_{\text{Op}}$, which measures the largest eigenvalue,

$$\text{Error} := \|\mathbf{C}_{yy} - \mathbf{I}_N\|_{\text{Op}}.$$

As a performance criterion, we use $\|\mathbf{C}_{yy} - \mathbf{I}_N\|_{\text{Op}} \leq 0.1$, the point at which the principal axes of $\mathbf{C}_{yy}$ are within 0.1 of unity. Geometrically, this means the ellipsoid corresponding to the covariance matrix lies between the circles with radii 0.9 and 1.1.

For visualization of output covariance matrices, we plot 2D ellipses representing the 1-standard deviation probability level-set contour of the density. These ellipses are defined by the set of points $\{\|\mathbf{C}_{yy}^{1/2}\mathbf{v}\|\mathbf{v} : \|\mathbf{v}\| = 1\}$.

4.1. Adaptive symmetric whitening via gain modulation

We first demonstrate that our algorithm successfully whitens its outputs. We initialize a network with fixed interneuron weights, $\mathbf{W}$, corresponding to the frame illustrated in Figure 1 (N=2, K=K_N=3). Figure 2 shows the network adapting to inputs from two successively-presented contexts with randomly-generated underlying input covariances $\mathbf{C}_{xx}$ (10K gain update steps each). As update steps progress, all marginal variances converge to unity, as expected from the objective (top panel). Since the number of interneurons satisfies K=K_N, the optimal gains to achieve symmetric whitening can be solved analytically (Appendix D), and are shown in the middle panel (dashed lines).

Figure 2 illustrates the *online, adaptive* nature of the network; it whitens inputs from novel statistical contexts at run-time, without supervision. By Proposition 2.1, measuring unit variance along K_N unique axes, as in this example, guarantees that the underlying joint density is statistically white. Indeed, the whitening error (bottom panel), approaches zero as all K_N marginal variances approach 1. Thus, with interneurons monitoring their respective *marginal* input variances z_i^2, and re-scaling their gains to modulate feedback onto the primary neurons, the network adaptively whitens its outputs in each context.

4.2. Algorithmic convergence rate depends on W

Our model assumes that the frame, $\mathbf{W}$, is fixed and known (e.g., optimized via pre-training or development). This distinguishes our method from existing symmetric whitening methods, which typically operate by estimating and transforming to the eigenvector basis. By contrast, our network obviates learning the principal axes of the data altogether, and instead uses a statistical sampling approach along the fixed set of measurement axes spanned by $\mathbf{W}$. While the result expressed in Proposition 2.1 is exact, and the *optimal solution* to the whitening objective Equation 5 is independent of $\mathbf{W}$ (provided Equation 3 holds), we hypothesize that the *algorithmic convergence rate* would depend on $\mathbf{W}$.

Figure 3 summarizes an experiment assessing the convergence rate of different networks whitening inputs with a random covariance, $\mathbf{C}_{xx}$, with $N = 2$ (the results are consistent when $N > 2$). We initialize three kinds of frames $\mathbf{W} \in \mathbb{R}^{N \times K_N}$ with 100 repetitions each: ‘**Random**’, a frame with i.i.d. Gaussian entries; ‘**Optimized**’, a randomly initialized frame whose columns are then optimized to have minimum mutual coherence and cover the ambient space; and ‘**Spectral**’, a frame whose first N columns are the eigenvectors of the data and the remaining $K_N - N$ columns are zeros. For clarity, we remove the effects of input sampling stochasticity by running the offline version of our network, which assumes having direct access to the input covariance (Appendix G); the online version is qualitatively similar.

When the input distribution is known, then using the input covariance eigenvectors, as with the Spectral frame, defines a bound on achievable performance, converging faster, on average, than the Random and Optimized frames (Figure 3A,B). This is because the frame is aligned with the input covariance’s principal axes, and a simple gain scaling along those directions is sufficient to achieve a whitened response. We find that the networks with Optimized frames converge at similar rates to those with Spectral frames, despite the frame vectors not being aligned with the principal axes of the data (Figure 3B). Comparing the Random to

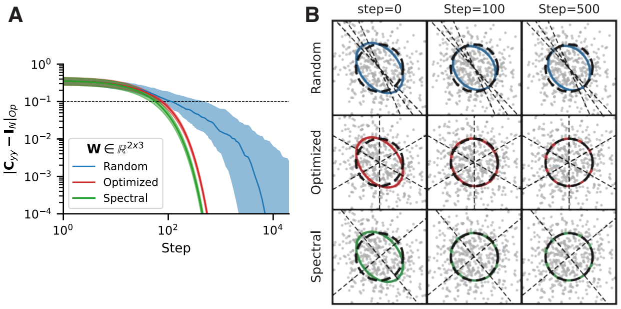

Figure 3. Convergence rate depends on structure of $\mathbf{W}$. For each network, η=1E-2. **A:** Error over time. Curves are median and [25%, 75%] quantile regions over 100 repeats. Dashed line indicates when the principal axes of 1-standard deviation ellipse representing $\mathbf{C}_{yy}$ are within 0.1 of unity. **B:** Scatter plots and covariance ellipses of $\mathbf{y}$ for a single experiment with each frame type at different steps. Gray dashed lines are axes spanned by $\mathbf{W}$.

Optimized frames gives a better understanding of how one might choose a frame in the more realistic scenario when the input distribution is unknown. The networks with Optimized frames systematically converge faster than Random frames. Thus, when the input distribution is unknown, we empirically find that the convergence rate of Algorithm 1 benefits from a frame that is optimized to splay the ambient space. Increased coverage of the space by the frame vectors facilitates whitening with our gain re-scaling mechanism. Sec. 4.5 elaborates on how underlying signal structure can be exploited to inform more efficient choices of frames.

4.3. Implicit sparse gating via gain modulation

Motivated by the findings in Sec 4.2, and concepts from sparse coding (Olshausen & Field, 1996), we explore how adaptive gain modulation can complement or augment a ‘pre-trained’ network with context-dependent weights. Figure 4 shows an experiment using either a pre-trained Spectral, or Random $\mathbf{W}$ (N=6, K=K_N=21) adaptively whitening inputs from two random, alternating statistical contexts, A and B, for 10K steps each. The first and second N columns of the Spectral frame are the eigenvectors of context A and B’s covariance matrix, respectively, and the remaining elements are random i.i.d. Gaussian; the Random frame has all i.i.d. Gaussian elements. Figure 4 (top panel) shows that both networks successfully adapt to whiten the inputs from each context, with the Spectral frame converging faster than the Random frame (as in Sec 4.2).

Inspecting the Spectral frame’s K interneuron gains during run-time (bottom panel) reveals that they sparsely ‘select’ the frame vectors corresponding to the eigenvectors of each respective condition (indicated by the blue/red intensity). This effect arises *without* a sparsity penalty or modifying the objective. Gain modulation thus *sparsely gates* context-dependent information without an explicit context signal.

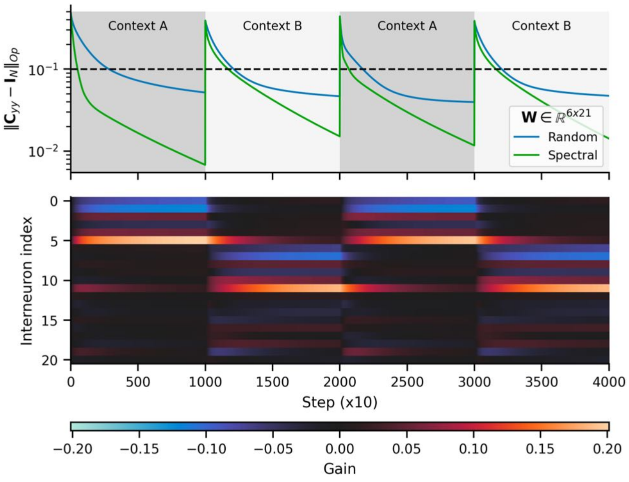

Figure 4. Gain modulation as a fast implicit sparse gating mechanism. **Top**: Error over time for Spectral vs. Random networks (N=6; K=K_N=21; η=1E-3) adapting to 2 alternating statistical contexts with different input covariances. Dashed line indicates when the principal axes of 1-standard deviation ellipsoid representing $\mathbf{C}_{yy}$ are within 0.1 of unity. **Bottom**: Gains act as implicit context switches, sparsely gating the respective eigenbases embedded in the Spectral frame to optimally whiten each context.

4.4. Normalizing ill-conditioned data

Foundational work by Atick & Redlich (1992) showed that neural populations in the retina may encode visual inputs by optimizing mutual information in the presence of noise. For natural images with $1/f$ spectra, the optimal transform is approximately a product of a whitening filter and a low-pass filter. This is a particularly effective solution because when inputs are low-rank, $\mathbf{C}_{xx}$ is ill-conditioned (Figure 5A), and classical whitening leads to noise amplification along axes with small variance. In this section, we show how a simple modification to the objective allows our gain-modulating network to handle these types of inputs.

We prevent amplification of inputs below a certain variance threshold by replacing the unit marginal variance equality constraints with upper bound constraints[3]:

$$\langle (\mathbf{w}_i^\top \mathbf{y}_t)^2 \rangle_t \leq 1 \quad \text{for} \quad i = 1, \ldots, K. \tag{11}$$

Our modified network objective then becomes

$$\min_{\{\mathbf{y}_t\}} \langle \|\mathbf{x}_t - \mathbf{y}_t\|_2^2 \rangle_t \quad \text{s.t.} \quad \text{Equation 11 holds.} \tag{12}$$

Intuitively, if the projected variance along a given direction is already less than or equal to unity, then it will not affect the overall loss. Interneuron gain should accordingly *stop adjusting* once the marginal variance along its projection axis is less than or equal to one. To enforce these upper

[3] We set the threshold to 1 to remain consistent with the whitening objective, but it can be any arbitrary variance.

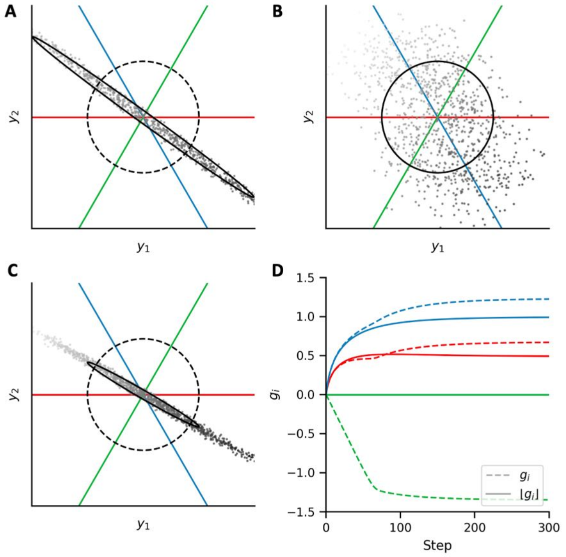

Figure 5. Two networks (N=2, K=3, η=0.02) whitening ill-conditioned inputs. **A:** Outputs without whitening. 2D scatterplot of a non-Gaussian density whose underlying signal lies close to a latent 1D axis. Many points lie outside of the axis limits in this panel. Signal magnitude along that axis is denoted by the grayscale gradient. The 1-standard deviation covariance matrix is depicted as a black ellipse. Colored lines are axes spanned by Optimal frame (see Sec 4.2). **B:** Symmetric whitening boosts noise along the uninformative direction. **C:** Modulating gains according to Eq. 14 rescales the data *without* amplifying noise. **D:** Gains updated with Eq. 10 vs. Eq. 14. Colors correspond to frame axes in panels A–C.

bound constraints, we introduce gains as Lagrange multipliers, but restrict the domain of $\mathbf{g}$ to be the non-negative orthant $\mathbb{R}^K_+$, resulting in non-negative optimal gains:

$$\max_{\mathbf{g}\in\mathbb{R}^K_+} \min_{\{\mathbf{y}_t\}} \langle \ell(\mathbf{x}_t, \mathbf{y}_t, \mathbf{g})\rangle_t, \tag{13}$$

where $\ell(\mathbf{x}, \mathbf{y}, \mathbf{g})$ is defined as in Equation 6. At each time step t, we optimize Equation 13 by first taking gradient-descent steps with respect to $\mathbf{y}_t$, resulting in the same neural dynamics (Equation 7) and equilibrium solution (Equation 8) as before. To update $\mathbf{g}$, we modify Equation 10 to take a *projected* gradient-ascent step with respect to $\mathbf{g}$:

$$\mathbf{g} \leftarrow \lfloor \mathbf{g} + \eta(\bar{\mathbf{z}}_t^{\circ 2} - \mathbf{1}) \rfloor \tag{14}$$

where $\lfloor\cdot\rfloor$ denotes the element-wise half-wave rectification operation that projects its inputs onto the non-negative orthant $\mathbb{R}^K_+$, i.e., $\lfloor \mathbf{v} \rfloor := [\max(v_1, 0), \ldots, \max(v_K, 0)]^\top$.

Figure 5 shows a simulation of a network whitening ill-conditioned inputs with an Optimized frame (N=2, K=K_N; see Sec. 4.2) where gains are either unconstrained (Equation 10), or rectified (Equation 14). We observe that these two models converge to two different solutions (Figure 5B, C). When g_i is unconstrained, the network achieves global whitening, as before, but in doing so it amplifies noise along the axis orthogonal to the latent signal axis. The gains constrained to be non-negative converged to different values than the unconstrained gains (Figure 5D), with one of them (green) converging to zero rather than becoming negative. In general, with constrained g_i, the whitening error network converges to a non-zero value (see Appendix H for details). Thus, with a non-negative constraint, the network normalizes the responses $\mathbf{y}$, and *does not amplify the noise*. In Appendix H we show additional cases that provide further geometric intuition on differences between symmetric whitening with and without non-negative constrained gains.

4.5. Gain modulation enables local spatial decorrelation

Requiring K_N interneurons to guarantee a statistically white output (Proposition 2.1) becomes prohibitively costly for high-dimensional inputs: the number of interneurons scales as $\mathcal{O}(N^2)$. This leads us to ask: how many interneurons are needed in practice? For natural sensory inputs such as images, it is well-known that inter-pixel correlation is highly structured, decaying as a function of distance. We simulate an experiment of visual gaze fixations and micro-saccadic eye movements using a Gaussian random walk, drawing 12×12 patch samples from a region of a natural image (Figure 6A; van Hateren & van der Schaaf, 1998); this can be interpreted as a form of video-streaming dataset where each frame is a patch sample. We repeat this for different randomly selected regions of the image (Figure 6A colors). The image content of each region is quite different, but the inter-pixel correlation within each context consistently falls rapidly with distance (Figure 6B).

We *relax* the $\mathcal{O}(N^2)$ marginal variance constraint to instead whiten *spatially local neighborhoods* of primary neurons whose inputs are the image patches. We construct a frame $\mathbf{W}$ that exploits spatial structure in the image patches, and spans $K < K_N$ axes in $\mathbb{R}^N$. $\mathbf{W}$ is convolutional, such that *overlapping* neighborhoods of 4×4 primary neurons are decorrelated, each by a population of interneurons that is 'overcomplete' with respect to that neighborhood (see Appendix I for details). Importantly, taking into account local structure dramatically reduces the interneuron complexity from $\mathcal{O}(N^2) \rightarrow \mathcal{O}(N)$, thereby making our framework practically feasible for high-resolution image inputs and video streams. This frame is still overcomplete ($K > N$), but because $K < K_N$, we no longer guarantee at equilibrium that $\mathbf{C}_{yy} = \mathbf{I}_N$ (Proposition 2.1).

After the network converges to the inputs drawn from the red context (Figure 6C): i) inter-pixel correlations drop within the region specified by the local neighborhood; and ii) surprisingly, correlations at longer-range (i.e. outside the window of the defined spatial neighborhood) are also dramatically reduced. Accordingly, the eigenspectrum of

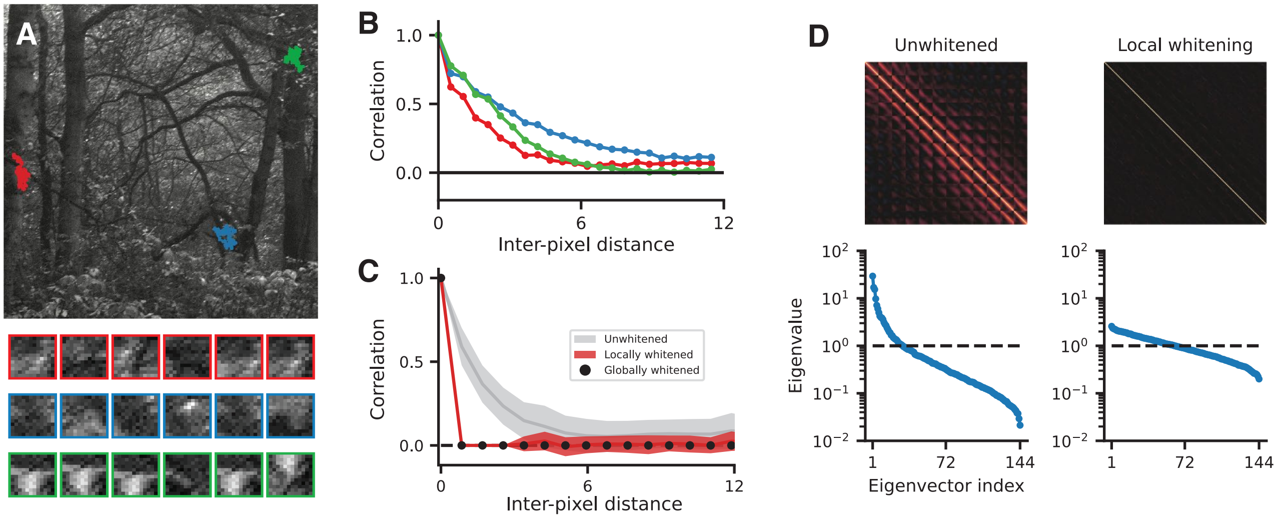

Figure 6. Local spatial whitening. **A)** Large grayscale image from which 12×12 image patch samples are drawn. Colors represent random-walk sampling from regions of the image corresponding to contexts with different underlying statistics. Six samples from each context are shown below. **B)** Without whitening, pixel correlations decay rapidly with spatial distance in each context, suggesting that local whitening may be effective. **C)** Binned pairwise output pixel correlation of patches from the red context before (gray) and after global (black dots) vs. local whitening with overlapping 4×4 neighborhoods (red). Shaded regions represent standard deviations. **D)** Top: Correlation matrices of flattened patches from the red context before whitening (left), and after local symmetric whitening (right). Both panels use the same color scale. Bottom: Corresponding covariance eigenspectra. Dashed lines are spectra after global whitening.

the locally whitened outputs is significantly flatter compared to the inputs (Figure 6D left vs. right columns). We also provide an example using 1D inputs in Appendix I. This empirical result is not obvious — that whitening individual *overlapping local* neighborhoods of neurons should produce a more *globally* whitened output covariance. Indeed, exactly how or when a globally whitened solution is possible from whitening of spatial overlapping neighborhoods of the inputs is a problem worth pursuing.

5. Related Work

5.1. Biologically plausible whitening networks

Biological circuits operate in the online setting and, due to physical constraints, must learn exclusively using local signals. Therefore, to plausibly model neural computation, a neural network model must operate in the online setting (i.e., streaming data) and use local learning rules (Pehlevan & Chklovskii, 2019). There are a few existing normative models of adaptive statistical whitening and related transformations; however, these models use synaptic plasticity mechanisms (i.e., changing $\mathbf{W}$) to adapt to changing input statistics (Pehlevan & Chklovskii, 2015; Westrick et al., 2016; Chapochnikov et al., 2021; Młynarski & Hermundstad, 2021; Lipshutz et al., 2023). Adaptation of neural population responses to changes in sensory input statistics occurs rapidly, on the order of hundreds of milliseconds to seconds (Muller et al., 1999; Wanner & Friedrich, 2020), so it could potentially arise from short-term synaptic plasticity, which operates on the timescale of tens of milliseconds to minutes (Zucker & Regehr, 2002), but not by long-term synaptic plasticity, which operates on the timescale of minutes or longer (Martin et al., 2000). Here, we have proposed an alternative hypothesis: that modulation of neural gains, which operates on the order of tens of milliseconds to minutes (Ferguson & Cardin, 2020), facilitates rapid adaptation of neural populations to changing input statistics.

5.2. Tomography and "sliced" density measurements

Leveraging 1D projections to compute the symmetric whitening transform is reminiscent of approaches taken in the field of tomography. Geometrically, our method represents an ellipsoid (i.e., the N dimensional covariance matrix) using noisy 1D projections of the ellipsoid onto axes spanned by frame vectors (i.e., estimates of the marginal variances). This is a special case of reconstruction problems studied in geometric tomography (Karl et al., 1994; Gardner, 1995). A distinction between tomography and our approach to symmetric whitening is that we are not reconstructing the multi-dimensional inputs; instead, we are utilizing the univariate measurements to transform an ellipsoid into a hyper-sphere.

In optimal transport, "sliced" methods offer a way to measure otherwise intractable p-Wasserstein distances in high dimensions (Bonneel et al., 2015), thereby enabling their use in optimization loss functions. Sliced methods estimate Wasserstein distance by taking series of 1D projections of two densities, then computing the expectation over all 1D Wasserstein distances, for which there exists an analytic solution. The 2-Wasserstein distance between a 1D zero-mean Gaussian with variance σ^2 and a standard normal density is

$$W_2\left(\mathcal{N}\left(0,\sigma^2\right);\mathcal{N}\left(0,1\right)\right) = \|\sigma - 1\|.$$

This is strikingly similar to Equation 10. However, distinguishing characteristics of our approach include: 1) minimizing distance between *variances* rather than standard deviations; 2) directions along which we compute slices are fixed, whereas sliced methods compute a new set of projections at each optimization step; 3) our network operates online, *without* backpropagation.

6. Discussion

Our study introduces a recurrent circuit for adaptive whitening using *gain modulation* to transform joint second-order statistics of their inputs based on *marginal* variance measurements. We demonstrate that, given sufficiently many marginal measurements along unique axes, the network produces symmetric whitened outputs. Our objective (Equation 5) provides a novel way to think about the classical problem of statistical whitening, and draws connections to old concepts from tomography and transport theory. This framework is *flexible and extensible*, with some possible generalizations explored in Appendix J. For example, we show that our model provides a way to prevent representational collapse in the analytically tractable example of online principal subspace learning (Appendix J.1). Additionally, by replacing the unity marginal variance constraint by a set of target variances differing from 1, the network can be used to transform its input density to one matching the corresponding (non-white) covariance (Appendix J.2).

6.1. Implications for machine learning

Decorrelation and whitening are canonical transformations in signal processing, widely used in compression and channel coding. Deep nets are generally not trained to whiten, although their response variances are generally normalized during training through batch normalization, and recent methods (e.g. Bardes et al., 2022) do impose global whitening properties in their objective functions. Modulating feature gains has proven effective in adapting pre-trained neural networks to novel inputs with out-of-training distribution statistics (Ballé et al., 2020; Duong et al., 2023; Mohan et al., 2021). Future architectures may benefit from adaptive run-time adjustments to changing input statistics (e.g. Hu et al., 2022). Our framework provides an unsupervised, online mechanism that avoids 'catastrophic forgetting' in neural networks during continual learning.

6.2. Implications for neuroscience

It has been known for nearly 100 years (Adrian & Matthews, 1928) that single neurons rapidly adjust their sensitivity (gain) adaptively, based on recent response history. Experiments suggest that neural populations *jointly* adapt, adjusting both the amplitude of their responses, as well as their correlations (e.g. Benucci et al., 2013; Friedrich, 2013) to confer dynamic, efficient multi-channel coding. The natural thought is that they achieve this by adjusting the strength of their interactions (synaptic weights). Our work provides a *fundamentally different* solution: these effects can arise solely through gain changes, thereby generalizing rapid and reversible single neuron adaptive gain modulation to the level of a neural population.

Support for our model will ultimately require careful experimental measurement and analysis of responses and gains of neurons in a circuit during adaptation (e.g. Wanner & Friedrich, 2020). Our model predicts: 1) Specific architectural constraints, such as reciprocally connected interneurons (Kepecs & Fishell, 2014), with consistency between their connectivity and population size (e.g. in the olfactory bulb). 2) Synaptic strengths that remain stable during adaptation, which would adjudicate between our model and more conventional adaptation models relying on synaptic plasticity (e.g. Lipshutz et al., 2023). 3) Interneurons that modulate their gains according to the difference between the variance of their post-synaptic inputs and some *target variance* (Equation 10; also see Appendix J.2). Experiments could assess whether interneuron input variances converge to the same values after adaptive whitening. 4) Interneurons that increase their gains with the variance of their inputs (i.e. $\bar{z}_{i,t}^2$). Input variance-dependent gain modulation may be mediated by changes in slow Na$^+$ currents (Kim & Rieke, 2003). This predicts a mechanistic role for interneurons during adaptation, and complements the observed gain effects found in excitatory neurons described in classical studies (Fairhall et al., 2001; Nagel & Doupe, 2006).

6.3. Conclusion

Whitening is an effective constraint for preventing feature collapse in representation learning (Zbontar et al., 2021; Ermolov et al., 2021). The networks developed here provide a whitening solution that is particularly well-suited for applications prioritizing streaming data and low-power consumption.

Acknowledgements

We thank Pierre-Étienne Fiquet, Sina Tootoonian, Michael Landy, and the ICML reviewers for their helpful feedback.

References

Adrian, E. D. and Matthews, R. The action of light on the eye: Part III. The interaction of retinal neurones. *The Journal of Physiology*, 65(3):273, 1928.

Atick, J. J. and Redlich, A. N. What does the retina know about natural scenes? *Neural Computation*, 4:196–210, 1992.

Ballé, J., Chou, P. A., Minnen, D., Singh, S., Johnston, N., Agustsson, E., Hwang, S. J., and Toderici, G. Nonlinear transform coding. *IEEE Journal of Selected Topics in Signal Processing*, 15(2):339–353, 2020.

Bardes, A., Ponce, J., and LeCun, Y. VICReg: Variance-invariance-covariance regularization for self-supervised learning. *International Conference on Learning Representations*, 2022.

Barlow, H. B. Possible Principles Underlying the Transformations of Sensory Messages. In *Sensory Communication*, pp. 216–234. The MIT Press, 1961.

Barlow, H. B. and Foldiak, P. Adaptation and decorrelation in the cortex. In *The Computing Neuron*, pp. 54–72. Addison-Wesley, 1989.

Bell, A. J. and Sejnowski, T. J. The "independent components" of natural scenes are edge filters. *Vision Research*, 37:3327–3338, 1996.

Benucci, A., Saleem, A. B., and Carandini, M. Adaptation maintains population homeostasis in primary visual cortex. *Nature Neuroscience*, 16(6):724–729, 2013.

Bonin, V., Mante, V., and Carandini, M. The statistical computation underlying contrast gain control. *The Journal of Neuroscience*, 26(23):6346–6353, 2006.

Bonneel, N., Rabin, J., Peyré, G., and Pfister, H. Sliced and Radon Wasserstein Barycenters of Measures. *Journal of Mathematical Imaging and Vision*, 51(1):22–45, January 2015.

Boyd, S. and Vandenberghe, L. *Convex Optimization*. Cambridge University Press, 2004.

Carlsson, M. von neumann's trace inequality for Hilbert-Schmidt operators. *Expositiones Mathematicae*, 39(1): 149–157, 2021.

Casazza, P. G., Kutyniok, G., and Philipp, F. Introduction to Finite Frame Theory. In Casazza, P. G. and Kutyniok, G. (eds.), *Finite Frames*, pp. 1–53. Birkhäuser Boston, 2013.

Chapochnikov, N. M., Pehlevan, C., and Chklovskii, D. B. Normative and mechanistic model of an adaptive circuit for efficient encoding and feature extraction. *bioRxiv*, 2021.

Coates, A., Ng, A., and Lee, H. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pp. 215–223. JMLR Workshop and Conference Proceedings, 2011.

Duong, L. R., Li, B., Chen, C., and Han, J. Multi-rate adaptive transform coding for video compression. *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 2023.

Ermolov, A., Siarohin, A., Sangineto, E., and Sebe, N. Whitening for self-supervised representation learning. In *International Conference on Machine Learning*, pp. 3015–3024. PMLR, 2021.

Fairhall, A. L., Lewen, G. D., and Bialek, W. Efficiency and ambiguity in an adaptive neural code. *Nature*, 412: 787–792, 2001.

Ferguson, K. A. and Cardin, J. A. Mechanisms underlying gain modulation in the cortex. *Nature Reviews Neuroscience*, 21(2):80–92, 2020.

Friedrich, R. W. Neuronal computations in the olfactory system of zebrafish. *Annual Review of Neuroscience*, 36: 383–402, 2013.

Gardner, R. J. *Geometric Tomography*, volume 58. Cambridge University Press Cambridge, 1995.

Giridhar, S., Doiron, B., and Urban, N. N. Timescale-dependent shaping of correlation by olfactory bulb lateral inhibition. *Proceedings of the National Academy of Sciences*, 108(14):5843–5848, 2011.

Gschwend, O., Abraham, N. M., Lagier, S., Begnaud, F., Rodriguez, I., and Carleton, A. Neuronal pattern separation in the olfactory bulb improves odor discrimination learning. *Nature Neuroscience*, 18(10):1474–1482, 2015.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. Lora: Low-rank adaptation of large language models. *International Conference on Learning Representations*, 2022.

Hua, T., Wang, W., Xue, Z., Ren, S., Wang, Y., and Zhao, H. On feature decorrelation in self-supervised learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9598–9608, 2021.

Hyvärinen, A. and Oja, E. Independent component analysis: algorithms and applications. *Neural Networks*, 13(4-5): 411–430, 2000.

Karl, W. C., Verghese, G. C., and Willsky, A. S. Reconstructing Ellipsoids from Projections. *CVGIP: Graphical Models and Image Processing*, 56(2):124–139, 1994.

Kepecs, A. and Fishell, G. Interneuron cell types are fit to function. *Nature*, 505:318–326, 2014.

Kessy, A., Lewin, A., and Strimmer, K. Optimal whitening and decorrelation. *The American Statistician*, 72(4):309–314, 2018.

Kim, K. J. and Rieke, F. Slow Na$^+$ inactivation and variance adaptation in salamander retinal ganglion cells. *Journal of Neuroscience*, 23(4):1506–1516, 2003.

Krizhevsky, A. Learning multiple layers of features from tiny images. Master's thesis, University of Toronto, 2009.

Laughlin, S. A Simple Coding Procedure Enhances a Neuron's Information Capacity. *Zeitschrift fur Naturforschung. C, Journal of Biosciences*, pp. 910–2, 1981.

Lipshutz, D., Pehlevan, C., and Chklovskii, D. B. Interneurons accelerate learning dynamics in recurrent neural networks for statistical adaptation. *International Conference on Learning Representations*, 2023.

Martin, S., Grimwood, P. D., and Morris, R. G. Synaptic plasticity and memory: an evaluation of the hypothesis. *Annual Review of Neuroscience*, 23(1):649–711, 2000.

Mohan, S., Vincent, J. L., Manzorro, R., Crozier, P., Fernandez-Granda, C., and Simoncelli, E. Adaptive denoising via gaintuning. *Advances in Neural Information Processing Systems*, 34:23727–23740, 2021.

Muller, J. R., Metha, A. B., Krauskopf, J., and Lennie, P. Rapid adaptation in visual cortex to the structure of images. *Science*, 285(5432):1405–1408, 1999.

Młynarski, W. F. and Hermundstad, A. M. Efficient and adaptive sensory codes. *Nature Neuroscience*, 24(7):998–1009, 2021.

Nagel, K. I. and Doupe, A. J. Temporal Processing and Adaptation in the Songbird Auditory Forebrain. *Neuron*, 51(6):845–859, 2006.

Olshausen, B. and Field, D. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381:607–609, 1996.

Pehlevan, C. and Chklovskii, D. B. A normative theory of adaptive dimensionality reduction in neural networks. *Advances in Neural Information Processing Systems*, 28, 2015.

Pehlevan, C. and Chklovskii, D. B. Neuroscience-Inspired Online Unsupervised Learning Algorithms: Artificial Neural Networks. *IEEE Signal Processing Magazine*, 36 (6):88–96, 2019.

Schwartz, O. and Simoncelli, E. P. Natural signal statistics and sensory gain control. *Nature Neuroscience*, 4(8): 819–825, August 2001.

van Hateren, J. and van der Schaaf, A. Independent component filters of natural images compared with simple cells in primary visual cortex. *Proceedings: Biological Sciences*, 265(1394):359–366, Mar 1998.

Wang, X.-J., Liu, Y., Sanchez-Vives, M. V., and McCormick, D. A. Adaptation and temporal decorrelation by single neurons in the primary visual cortex. *Journal of Neurophysiology*, 89(6):3279–3293, 2003.

Wanner, A. A. and Friedrich, R. W. Whitening of odor representations by the wiring diagram of the olfactory bulb. *Nature Neuroscience*, 23(3):433–442, 2020.

Westrick, Z. M., Heeger, D. J., and Landy, M. S. Pattern adaptation and normalization reweighting. *Journal of Neuroscience*, 36(38):9805–9816, 2016.

Zbontar, J., Jing, L., Misra, I., LeCun, Y., and Deny, S. Barlow twins: Self-supervised learning via redundancy reduction. In *International Conference on Machine Learning*, pp. 12310–12320. PMLR, 2021.

Zucker, R. S. and Regehr, W. G. Short-term synaptic plasticity. *Annual Review of Physiology*, 64(1):355–405, 2002.

A. Notation

For $N \geq 2$, let $K_N := N(N+1)/2$. Let $\mathbb{R}^N$ denote N-dimensional Euclidean space equipped with the Euclidean norm, denoted $\|\cdot\|_2$. Let $\mathbb{R}^N_+$ denote the non-negative orthant in $\mathbb{R}^N$. Given $K \geq 2$, let $\mathbb{R}^{N\times K}$ denote the set of $N \times K$ real-valued matrices. Let $\mathbb{S}^N$ denote the set of $N \times N$ symmetric matrices and let $\mathbb{S}^N_{++}$ denote the set of $N \times N$ symmetric positive definite matrices.

Matrices are denoted using bold uppercase letters (e.g., $\mathbf{M}$) and vectors are denoted using bold lowercase letters (e.g., $\mathbf{v}$). Given a matrix $\mathbf{M}$, M_{ij} denotes the entry of $\mathbf{M}$ located at the i^{th} row and j^{th} column. Let $\mathbf{1} = [1, \ldots, 1]^\top$ denote the N-dimensional vector of ones. Let $\mathbf{I}_N$ denote the $N \times N$ identity matrix.

Given vectors $\mathbf{v}, \mathbf{w} \in \mathbb{R}^N$, define their Hadamard product by $\mathbf{v} \circ \mathbf{w} := (v_1 w_1, \ldots, v_N w_N) \in \mathbb{R}^N$. Define $\mathbf{v}^{\circ 2} := (v_1^2, \ldots, v_N^2) \in \mathbb{R}^N$.

Let $\langle \cdot \rangle_t$ denote expectation over $t = 1, 2, \ldots$.

The $\text{diag}(\cdot)$ operator, similar to `numpy.diag()` or MATLAB's `diag()`, can either: 1) map a vector in $\mathbb{R}^K$ to the diagonal of a $K \times K$ zeros matrix; or 2) map the diagonal entries of a $K \times K$ matrix to a vector in $\mathbb{R}^K$. The specific operation being used should be clear by context. For example, given a vector $\mathbf{v} \in \mathbb{R}^K$, define diag($\mathbf{v}$) to be the $K \times K$ diagonal matrix whose $(i,i)^{\text{th}}$ entry is equal to v_i, for $i = 1, \ldots, K$. Alternatively, given a sqaure matrix $\mathbf{M} \in \mathbb{R}^{K\times K}$, define diag($\mathbf{M}$) to be the K-dimensional vector whose i^{th} entry is equal to M_{ii}, for $i = 1, \ldots, K$.

B. Optimal Solution to Symmetric Whitening Objective

In this section, we prove that the optimal solution to the optimization problem in equation 2 is given by $\mathbf{y}_t = \mathbf{C}_{xx}^{-1/2}\mathbf{x}_t$ for $t = 1, \ldots, T$ (we treat the case that $T < \infty$).

We first recall Von Neumann's trace inequality (see, e.g., Carlsson, 2021, Theorem 3.1).

Lemma B.1 (Von Neumann's trace inequality)**.** *Suppose* $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{n\times m}$ *with* $n \leq m$*. Let* $\sigma_1^A \geq \cdots \geq \sigma_n^A \geq 0$ *and* $\sigma_1^B \geq \cdots \geq \sigma_n^B \geq 0$ *denote the respective singular values of* $\mathbf{A}$ *and* $\mathbf{B}$*. Then*

$$\text{Tr}(\mathbf{A}\mathbf{B}^\top) \leq \sum_{i=1}^n \sigma_i^A \sigma_i^B.$$

Furthermore, equality holds if and only if $\mathbf{A}$ *and* $\mathbf{B}$ *share left and right singular vectors.*

We can now proceed with the proof of our result. We first concatenate the inputs and outputs into data matrices $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_T] \in \mathbb{R}^{N\times T}$ and $\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_T] \in \mathbb{R}^{N\times T}$. We can write equation 2 as follows:

$$\min_{\mathbf{Y}} \|\mathbf{X} - \mathbf{Y}\|_F^2 \qquad \text{subject to} \qquad \mathbf{Y}\mathbf{Y}^\top = T\mathbf{I}_N.$$

Expanding, substituting in with the constraint $\mathbf{Y}\mathbf{Y}^\top = T\mathbf{I}_N$ and dropping terms that do not depend on $\mathbf{Y}$ results in the objective

$$\max_{\mathbf{Y}} \text{Tr}(\mathbf{X}\mathbf{Y}^\top) \qquad \text{subject to} \qquad \mathbf{Y}\mathbf{Y}^\top = T\mathbf{I}_N.$$

By Von Neumann's trace inequality, the trace is maximized when the singular vectors of $\mathbf{Y}$ are aligned with the singular vectors of $\mathbf{X}$. In particular, if the SVD of $\mathbf{X}$ is given by $\mathbf{X} = \mathbf{U}_x\mathbf{S}_x\mathbf{V}_x^\top$, then the optimal $\mathbf{Y}$ is given by $\mathbf{Y} = \sqrt{T}\mathbf{U}_x\mathbf{V}_x^\top$, which is precisely $\mathbf{C}_{xx}^{-1/2}\mathbf{X}$, where $\mathbf{C}_{xx} := \frac{1}{T}\mathbf{X}\mathbf{X}^\top = \mathbf{U}_x\mathbf{S}_x^2\mathbf{U}_x^\top$.

C. Proof of Proposition 2.1

Proof of Proposition 2.1. Suppose Equation 1 holds. Then, for $i = 1, \ldots, K$,

$$\langle(\mathbf{w}_i^\top\mathbf{y}_t)^2\rangle_t = \langle\mathbf{w}_i^\top\mathbf{y}_t\mathbf{y}_t^\top\mathbf{w}_i\rangle_t = \mathbf{w}_i^\top\mathbf{w}_i = 1.$$

Therefore, Equation 4 holds.

Now suppose Equation 4 holds. Let $\mathbf{v} \in \mathbb{R}^N$ be an arbitrary unit vector. Then $\mathbf{v}\mathbf{v}^\top \in \mathbb{S}^N$ and by Equation 3, there exist $g_1, \dots, g_K \in \mathbb{R}$ such that

$$\mathbf{v}\mathbf{v}^\top = g_1 \mathbf{w}_1 \mathbf{w}_1^\top + \cdots + g_K \mathbf{w}_K \mathbf{w}_K^\top. \tag{15}$$

We have

$$\mathbf{v}^\top \langle \mathbf{y}_t \mathbf{y}_t^\top \rangle_t \mathbf{v} = \operatorname{Tr}(\mathbf{v}\mathbf{v}^\top \langle \mathbf{y}_t \mathbf{y}_t^\top \rangle_t) = \sum_{i=1}^K g_i \operatorname{Tr}(\mathbf{w}_i \mathbf{w}_i^\top \langle \mathbf{y}_t \mathbf{y}_t^\top \rangle_t) = \sum_{i=1}^K g_i \operatorname{Tr}(\mathbf{w}_i \mathbf{w}_i^\top) = \operatorname{Tr}(\mathbf{v}\mathbf{v}^\top) = 1. \tag{16}$$

The first equality is a property of the trace operator. The second and fourth equalities follow from Equation 15 and the linearity of the trace operator. The third equality follows from Equation 4, the cyclic property of the trace, and the fact that each $\mathbf{w}_i$ is a unit vector. The final equality holds because $\mathbf{v}$ is a unit vector. Since Equation 16 holds for every unit vector $\mathbf{v} \in \mathbb{R}^N$, Equation 1 holds. □

D. Frame Factorizations of Symmetric Matrices

D.1. Analytic solution for the optimal gains

Recall that the optimal solution of the symmetric objective in Equation 5 is given by $\mathbf{y}_t = \mathbf{C}_{xx}^{-1/2}\mathbf{x}_t$ for $t = 1, 2, \dots$. In our neural circuit with interneurons and gain control, the outputs of the primary neurons at equilibrium is (given in Equation 8, but repeated here for clarity),

$$\bar{\mathbf{y}}_t = \left[\mathbf{I}_N + \mathbf{W} \operatorname{diag}(\mathbf{g}) \mathbf{W}^\top\right]^{-1} \mathbf{x}_t,$$

where $\mathbf{W} \in \mathbb{R}^{N\times K}$ is overcomplete, arbitrary (provided Equation 3 holds), and *fixed*; and elements of $\mathbf{g} \in \mathbb{R}^K$ can be interpreted as learnable scalar gains. The circuit performs symmetric whitening when the gains $\mathbf{g}$ satisfy the relation

$$\mathbf{I}_N + \mathbf{W} \operatorname{diag}(\mathbf{g}) \mathbf{W}^\top = \mathbf{C}_{xx}^{1/2}. \tag{17}$$

It is informative to contrast this with conventional approaches to symmetric whitening, which rely on eigendecompositions,

$$\mathbf{V} \operatorname{diag}(\boldsymbol{\lambda})^{1/2} \mathbf{V}^\top = \mathbf{C}_{xx}^{1/2},$$

where $\mathbf{V} \in \mathbb{R}^{N\times N}$ and $\boldsymbol{\lambda}$ are the eigenvectors and eigenvalues of $\mathbf{C}_{xx}$, respectively. Note that in this eigenvector formulation, both vector quantities (columns of $\mathbf{V}$) and scalar quantities (elements of $\boldsymbol{\lambda}$) need to be learned, whereas in our formulation (Equation 17), *only scalars* need to be learned (elements of $\mathbf{g}$).

When $K \geq N(N+1)/2$, we can explicitly solve for the optimal gains $\mathbf{g}^*$ (derived in the next subsection):

$$\mathbf{g}^* = \left[\left(\mathbf{W}^\top \mathbf{W}\right)^{\circ 2}\right]^\dagger \left[\mathbf{w}_1^\top \mathbf{C}_{xx}^{1/2} \mathbf{w}_1 - 1, \dots, \mathbf{w}_K^\top \mathbf{C}_{xx}^{1/2} \mathbf{w}_K - 1\right]^\top. \tag{18}$$

D.2. Isolating g embedded in a diagonal matrix

In the upcoming subsection, our variable of interest, $\mathbf{g}$, is embedded along the diagonal of a matrix, then wedged between two fixed matrices, i.e. $\mathbf{A}_1 \operatorname{diag}(\mathbf{g}) \mathbf{A}_2$. We employ the following identity to isolate $\mathbf{g}$,

$$\operatorname{diag}(\mathbf{A}_1 \operatorname{diag}(\mathbf{g}) \mathbf{A}_2) = \left(\mathbf{A}_1 \circ \mathbf{A}_2^\top\right) \mathbf{g}, \tag{19}$$

where, on the left-hand-side, the inner $\operatorname{diag}(\cdot)$ forms a diagonal matrix from a vector, the outer $\operatorname{diag}(\cdot)$ returns the diagonal of a matrix as a vector, and $\circ$ is the element-wise Hadamard product.

D.3. Deriving optimal gains

Let $\mathbf{C} \in \mathbb{S}^N$, where $\mathbb{S}^N$ is the set of symmetric $N \times N$ matrices. Suppose $\mathbf{g} \in \mathbb{R}^K$ is such that the following holds:

$$\mathbf{W} \operatorname{diag}(\mathbf{g}) \mathbf{W}^\top = \mathbf{C} \tag{20}$$

where $\mathbf{W} \in \mathbb{R}^{N \times K}$ is some fixed, arbitrary, frame with $K \geq \frac{N(N+1)}{2}$ (i.e. a representation that is $\mathcal{O}(N^2)$ overcomplete). To solve for $\mathbf{g}$, we multiply both sides of Equation 20 from the left and right by $\mathbf{W}^\top$ and $\mathbf{W}$, respectively, then take the diagonal[4] of the resultant matrices,

$$\operatorname{diag}\left(\mathbf{W}^\top \mathbf{W} \operatorname{diag}(\mathbf{g}) \mathbf{W}^\top \mathbf{W}\right) = \operatorname{diag}\left(\mathbf{W}^\top \mathbf{C} \mathbf{W}\right). \tag{21}$$

Finally, employing the identity in Equation 19 yields

$$(\mathbf{W}^\top \mathbf{W})^{\circ 2} \mathbf{g} = \operatorname{diag}\left(\mathbf{W}^\top \mathbf{C} \mathbf{W}\right), \tag{22}$$

$$\mathbf{g} = \left[(\mathbf{W}^\top \mathbf{W})^{\circ 2}\right]^\dagger \operatorname{diag}\left(\mathbf{W}^\top \mathbf{C} \mathbf{W}\right), \tag{23}$$

where $(\cdot)^{\circ 2}$ denotes element-wise squaring, $(\mathbf{W}^\top \mathbf{W})^{\circ 2}$ is positive semidefinite by the Schur product theorem and $(\cdot)^\dagger$ denotes the Moore-Penrose pseudoinverse. Thus, *any* $N \times N$ symmetric matrix, can be encoded as a vector, $\mathbf{g}$, with respect to an arbitrary fixed frame, $\mathbf{W}$, by solving a standard linear system of K equations of the form $\mathbf{A}\mathbf{g} = \mathbf{b}$. Importantly, when $K = \frac{N(N+1)}{2}$ and the columns of $\mathbf{W}$ are not collinear, we have empirically found the matrix on the LHS, $(\mathbf{W}^\top \mathbf{W})^{\circ 2}$, to be positive definite, so the vector $\mathbf{g}$ is uniquely defined.

Without loss of generality, assume that the columns of $\mathbf{W}$ are unit-norm (otherwise, we can always normalize them by absorbing their lengths into the elements of $\mathbf{g}$). Furthermore, assume without loss of generality that $\mathbf{C} \in \mathbb{S}^N_{++}$, the set of all symmetric positive definite matrices (e.g. covariance, precision, PSD square roots, etc.). When $\mathbf{C}$ is a covariance matrix, then $\operatorname{diag}\left(\mathbf{W}^\top \mathbf{C} \mathbf{W}\right)$ can be interpreted as a vector of projected variances of $\mathbf{C}$ along each axis spanned by $\mathbf{W}$. Therefore, Equation 22 states that the vector $\mathbf{g}$ is linearly related to the vector of projected variances via the element-wise squared frame Gramian, $(\mathbf{W}^\top \mathbf{W})^{\circ 2}$.

E. Saddle Point Property

In this section, we prove the following minimax property (for the case $t = 1, \ldots, T$ with T finite):

$$\min_{\{\mathbf{y}_t\}} \max_{\mathbf{g}} \langle \ell(\mathbf{x}_t, \mathbf{y}_t, \mathbf{g}) \rangle_t = \max_{\mathbf{g}} \min_{\{\mathbf{y}_t\}} \langle \ell(\mathbf{x}_t, \mathbf{y}_t, \mathbf{g}) \rangle_t. \tag{24}$$

The proof relies on the following minimax property for a function that satisfies the saddle point property (Boyd & Vandenberghe, 2004, section 5.4).

Theorem E.1. *Let $V \subseteq \mathbb{R}^n$, $W \subseteq \mathbb{R}^m$ and $f : V \times W \to \mathbb{R}$. Suppose f satisfies the saddle point property; that is, there exists $(\mathbf{a}^*, \mathbf{b}^*) \in V \times W$ such that*

$$f(\mathbf{a}^*, \mathbf{b}) \leq f(\mathbf{a}^*, \mathbf{b}^*) \leq f(\mathbf{a}, \mathbf{b}^*), \qquad \textit{for all } (\mathbf{a}, \mathbf{b}) \in V \times W.$$

Then

$$\min_{\mathbf{a} \in V} \max_{\mathbf{b} \in W} f(\mathbf{a}, \mathbf{b}) = \max_{\mathbf{b} \in W} \min_{\mathbf{a} \in V} f(\mathbf{a}, \mathbf{b}) = f(\mathbf{a}^*, \mathbf{b}^*).$$

In view of Theorem E.1, it suffices to show there exists $(\mathbf{y}_1^*, \ldots, \mathbf{y}_T^*, \mathbf{g}^*)$ such that

$$\ell(\mathbf{y}_1^*, \ldots, \mathbf{y}_T^*, \mathbf{g}) \leq \ell(\mathbf{y}_1^*, \ldots, \mathbf{y}_T^*, \mathbf{g}^*) \leq \ell(\mathbf{y}_1, \ldots, \mathbf{y}_T, \mathbf{g}^*), \qquad \text{for all } \mathbf{y}_1, \ldots, \mathbf{y}_T \in \mathbb{R}^N \text{ and } \mathbf{g} \in \mathbb{R}^K. \tag{25}$$

Define $\mathbf{y}_t^* := \mathbf{C}_{xx}^{-1/2} \mathbf{x}_t$ for all $t = 1, \ldots, T$ and define $\mathbf{g}^*$ as in equation 18 so that equation 17 holds. Then, for all $\mathbf{g} \in \mathbb{R}^K$,

$$\ell(\mathbf{y}_1^*, \ldots, \mathbf{y}_T^*, \mathbf{g}) = \frac{1}{T} \sum_{t=1}^{T} \|\mathbf{x}_t - \mathbf{y}_t^*\|_2^2.$$

[4] Similar to commonly-used matrix libraries, the $\operatorname{diag}(\cdot)$ operator here is overloaded and can map a vector to a matrix or vice versa. See Appendix A for details.

Therefore, the first inequality in equation 25 holds (in fact it is an equality for all $\mathbf{g}$). Next, we have

$$\begin{aligned}
\ell(\mathbf{y}_1,\dots,\mathbf{y}_T,\mathbf{g}^*) &= \frac{1}{T}\sum_{t=1}^{T}\|\mathbf{x}_t-\mathbf{y}_t\|_2^2+\frac{1}{T}\sum_{t=1}^{T}\mathrm{Tr}\left[\mathbf{W}\mathrm{diag}(\mathbf{g}^*)\mathbf{W}^\top(\mathbf{y}_t\mathbf{y}_t^\top-\mathbf{I}_N)\right]\\
&= \frac{1}{T}\sum_{t=1}^{T}(\mathbf{x}_t^\top\mathbf{x}_t-2\mathbf{x}_t^\top\mathbf{y}_t)+\frac{1}{T}\sum_{t=1}^{T}\mathrm{Tr}\left[(\mathbf{I}_N+\mathbf{W}\mathrm{diag}(\mathbf{g}^*)\mathbf{W}^\top)(\mathbf{y}_t\mathbf{y}_t^\top-\mathbf{I}_N)\right]\\
&= \frac{1}{T}\sum_{t=1}^{T}(\mathbf{x}_t^\top\mathbf{x}_t-2\mathbf{x}_t^\top\mathbf{y}_t)+\frac{1}{T}\sum_{t=1}^{T}\mathrm{Tr}\left[\mathbf{C}_{xx}^{1/2}(\mathbf{y}_t\mathbf{y}_t^\top-\mathbf{I}_N)\right]
\end{aligned}$$

Since $\mathbf{C}_{xx}^{1/2}$ is positive definite, $\ell(\mathbf{y}_1,\dots,\mathbf{y}_T,\mathbf{g}^*)$ is strictly convex in $(\mathbf{y}_1,\dots,\mathbf{y}_T)$ with its unique minimum obtained at $\mathbf{y}_t=\mathbf{C}_{xx}^{-1/2}\mathbf{x}_t$ for all $t=1,\dots,T$ (to see this, differentiate with respect to $\mathbf{y}_1,\dots,\mathbf{y}_T$, set the derivatives equal to zero and solve for $\mathbf{y}_1,\dots,\mathbf{y}_T$). This establishes the second inequality in equation 25 holds. Therefore, by Theorem E.1, equation 24 holds.

F. Weighted Average Update Rule for $\mathbf{g}$

The update for $\mathbf{g}$ in Equation 10 can be generalized to allow for a weighted average over past samples. In particular, the general update is given by

$$\mathbf{g}\leftarrow\mathbf{g}+\eta\left(\frac{1}{Z}\sum_{s=1}^{t}\gamma^{t-s}\mathbf{z}_s^{\circ 2}-\mathbf{1}\right),$$

where $\gamma\in[0,1]$ determines the decay rate and $Z:=1+\gamma+\cdots+\gamma^{t-1}$ is a normalizing factor.

G. Batched and Offline Algorithms for Whitening with RNNs via Gain Modulation

In addition to the fully-online algorithm provided in the main text (Algorithm 1), we also provide two variants below. In many applications, streaming inputs arrive in batches rather than one at a time (e.g. video streaming frames). Similarly for conventional offline stochastic gradient descent training, data is sampled in batches. Algorithm 2 would be one way to accomplish this in our framework, where the main difference between the fully online version is taking the mean across samples in the batch to yield average gain update $\Delta\mathbf{g}$ term. Furthermore, in the fully offline setting when the covariance of the inputs, $\mathbf{C}_{xx}$ is known, Algorithm 3 presents a way to whiten the covariance directly.

Algorithm 2 Batched symmetric whitening

```
Input: Data matrix $\mathbf{X}\in\mathbb{R}^{N\times T}$ (centered)
Initialize: $\mathbf{W}\in\mathbb{R}^{N\times K}$; $\mathbf{g}\in\mathbb{R}^K$; $\eta$; batch size $B$
while not converged do
    $\mathbf{X}_B \leftarrow$ sample_batch($\mathbf{X}, B$)
    $\mathbf{Y}_B \leftarrow [\mathbf{I}_N+\mathbf{W}\,\mathrm{diag}\,(\mathbf{g})\,\mathbf{W}^\top]^{-1}\mathbf{X}_B$
    $\mathbf{Z}_B \leftarrow \mathbf{W}^\top\mathbf{Y}_B$
    $\Delta\mathbf{g} \leftarrow \frac{1}{T}\mathrm{diag}(\mathbf{Z}_B\mathbf{Z}_B^\top)-\mathbf{1}$
    $\mathbf{g}\leftarrow\mathbf{g}+\eta$ mean($\Delta\mathbf{g}$, axis=1)
end while
```

Algorithm 3 Offline symmetric whitening

```
Input: Input covariance $\mathbf{C}_{xx}$
Initialize: $\mathbf{W}\in\mathbb{R}^{N\times K}$; $\mathbf{g}\in\mathbb{R}^K$; $\eta$
while not converged do
    $\mathbf{M}\leftarrow[\mathbf{I}_N+\mathbf{W}\,\mathrm{diag}\,(\mathbf{g})\,\mathbf{W}^\top]^{-1}$
    $\mathbf{C}_{yy}\leftarrow\mathbf{M}\mathbf{C}_{xx}\mathbf{M}$
    $\Delta\mathbf{g}\leftarrow\mathrm{diag}\left(\mathbf{W}^\top\mathbf{C}_{yy}\mathbf{W}\right)-\mathbf{1}$
    $\mathbf{g}\leftarrow\mathbf{g}+\eta\Delta\mathbf{g}$
end while
```

H. Normalizing Ill-conditioned Inputs with Non-negative Constrained Gains

H.1. Quantifying whitening error

Whitening with non-negative gains does not, in general, produce an output with identity covariance matrix; therefore, quantifying algorithm performance with the error defined in the main text would not be informative. Because this extension shares similarities with ideas of regularized whitening, in which principal axes whose eigenvalues are below a certain threshold are unaffected by the whitening transform, we quantify algorithmic performance using thresholded Spectral Error,

$$\text{Spectral Error} := \frac{1}{N}\sum_{i}^{N} \max(\lambda_i - 1, 0)^2,$$

where λ_i is the i^{th} eigenvalue of $\mathbf{C}_{yy}$. Here, as in the main text, we set the threshold to 1. Figure 7 shows that this network reduces spectral error. Importantly, the converged solution depends on the initial choice of frame (see next subsection).

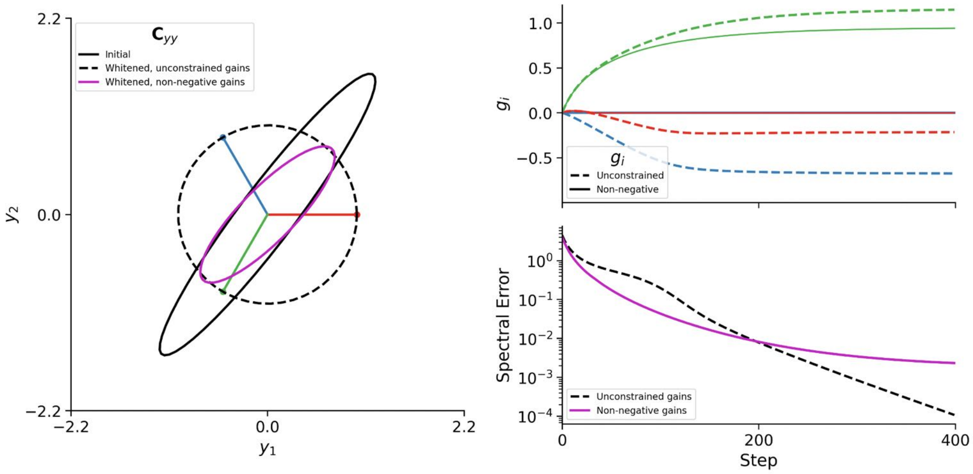

Figure 7. Whitening ill-conditioned inputs with non-negative gains. **A)** An equi-angular frame (red, blue, green; see Sec. 4.2) whitening ill-conditioned inputs. **B)** Gains as algorithm progresses, using updates with either rectified or unrectified constraints. **C)** Spectral Error (see text).

H.2. Geometric intuition behind thresholded whitening with non-negative gains

In general, the modified objective with rectified gains (Equation 14) does not statistically whiten the inputs $\mathbf{x}_1, \mathbf{x}_2, \ldots$, but rather adapts the non-negative gains $g_1, \ldots, g_K$ to ensure that the variances of the outputs $\mathbf{y}_1, \mathbf{y}_2, \ldots$ in the directions spanned by the frame vectors $\{\mathbf{w}_1, \ldots, \mathbf{w}_K\}$ are bounded above by unity (Figure 8). This one-sided normalization carries interesting implications for how and when the circuit statistically whitens its outputs, which can be compared with experimental observations. For instance, the circuit performs symmetric whitening if and only if there are non-negative gains such that Equation 17 holds (see, e.g., the top right example in Figure 8), which corresponds to cases such that the matrix $\mathbf{C}_{xx}^{1/2}$ is an element of the following cone (with its vertex translated by $\mathbf{I}_N$):

$$\left\{ \mathbf{I}_N + \sum_{i=1}^{K} g_i \mathbf{w}_i \mathbf{w}_i^\top : \mathbf{g} \in \mathbb{R}_+^K \right\}.$$

On the other hand, if the variance of an input projection is less than unity — i.e., $\mathbf{w}_i^\top \mathbf{C}_{xx} \mathbf{w}_i \leq 1$ for some i — then the corresponding gain g_i remains zero. When this is true for all $i = 1, \ldots, K$, the gains all remain zero and the circuit output is equal to its input (see, e.g., the bottom middle panel of Figure 8).

I. Whitening Spatially Local Neighborhoods

I.1. Spatially local whitening in 1D

For an N-dimensional input, we consider a network that whitens spatially local neighborhoods of size $M < N$. To this end, we can construct N filters of the form

$$\mathbf{w}_i = \mathbf{e}_i, \qquad i = 1, \ldots, N$$

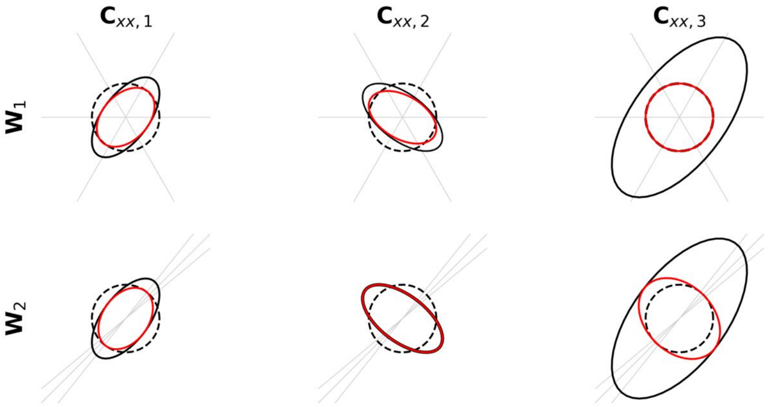

Figure 8. Geometric intuition of whitening with/without inequality constraint. Whitening efficacy using non-negative gains depends on $\mathbf{W}$ and $\mathbf{C}_{xx}$. For $N = 2$ and $K = 3$, examples of covariance matrices $\mathbf{C}_{yy}$ (red ellipses) corresponding to optimal solutions $\mathbf{y}$ of objective 12, for varying input covariance matrices $\mathbf{C}_{xx}$ (black ellipses) and frames $\mathbf{W}$ (spanning axes denoted by gray lines). Unit circles, which correspond to the identity matrix target covariance, are shown with dashed lines. Each row corresponds to a different frame $\mathbf{W}$ and each column corresponds to a different input covariance $\mathbf{C}_{xx}$.

and $M(N - \frac{M+1}{2})$ filters of the form

$$\mathbf{w}_{ij} = \frac{\mathbf{e}_i + \mathbf{e}_j}{\sqrt{2}}, \qquad i, j = 1, \ldots, N, \qquad 1 \leq |i - j| \leq M.$$

The total number of filters is $(M + 1)(N - \frac{M}{2})$, so for fixed M the number of filters scales linearly in N rather than quadratically.

We simulated a network comprising $N = 10$ primary neurons, and a convolutional weight matrix connecting each interneuron to spatial neighborhoods of three primary neurons. Given input data with covariance $\mathbf{C}_{xx}$ illustrated in Figure 9A (left panel), this modified network succeeded to statistically whiten local neighborhoods of size of primary 3 neurons (right panel). Notably, the eigenspectrum (Figure 9B) after local whitening is much closer to being equalized. Furthermore, while the global whitening solution produced a flat spectrum as expected, the local whitening network did not amplify the axis with very low-magnitude eigenvalues (Figure 9B right panel).

I.2. Filter bank construction in 2D

Here, we describe one way of constructing a set of convolutional weights for overlapping spatial neighborhoods (e.g. image patches) of neurons. Given an $n \times m$ input and overlapping neighborhoods of size $h \ \times w$ to be statistically whitened, the samples are therefore matrices $X \in \mathbb{R}^{n \times m}$. In this case, filters $\mathbf{w} \in \mathbb{R}^{1 \times n \times m}$ can be indexed by pairs of pixels that are in the same patch:

$$((i, j), (k, \ell)), \qquad 1 \leq i \leq n, \qquad 1 \leq j \leq m, \qquad 0 \leq |i - k| \leq h, \qquad 0 \leq |j - \ell| \leq w$$

We can then construct the filters as,

$$\mathbf{w}_{(i,j),(k,\ell)}(X) = \begin{cases} x_{i,j} & \text{if } (i, j) = (k, \ell), \\ \frac{x_{i,j} + x_{k,\ell}}{\sqrt{2}} & \text{if } (i, j) \neq (k, \ell). \end{cases}$$

In this case there are

$$nm + wh \left[(n - w)(m - h) + (n - w)\frac{(h + 1)}{2} + (m - h)\frac{(w + 1)}{2} + (h + 1)\frac{(w + 1)}{2} \right]$$

such filters, so the number of filters required scales linearly with nm rather than quadratically.

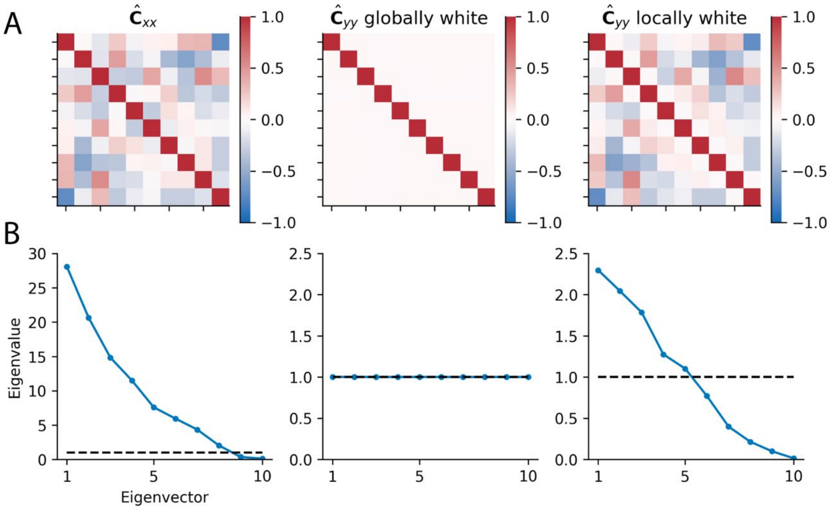

Figure 9. Statistically adapting local neighborhoods of neurons. **A)** $\hat{\mathbf{C}}_{xx}$ denotes correlation matrix, which are shown here for display purposes only, to facilitate comparisons. Network with 10-dimensional input correlation (left) 10-dimensional output correlation matrix after global whitening (middle); and output correlation matrix after statistically whitening local neighborhoods of size 3. The output correlation matrix of the locally adapted circuit has block-identity structure along the diagonal. **B)** Corresponding eigenspectra of *covariance* matrices of unwhitened (left), global whitened (middle), and locally whitened (right) network outputs. The y-axis limits of the middle and right columns are the same, but different than the left column. The black dashed line denotes unity.

J. Additional Applications

J.1. Preventing representational collapse in online principal subspace learning

Here, similar to Lipshutz et al. (2023), we show how whitening can prevent representational collapse using the analytically tractable example of online principal subspace learning. Recent approaches to self-supervised learning have used decorrelation transforms such as whitening to prevent collapse during training (e.g. Zbontar et al., 2021). Future architectures may benefit from online, adaptive whitening to allow for continual learning and test-time adaptation.

Consider a primary neuron whose *pre-synaptic* input at time t is $\mathbf{s}_t \in \mathbb{R}^D$, and corresponding output is $y_t := \mathbf{v}^\top \mathbf{s}_t$, where $\mathbf{v} \in \mathbb{R}^D$ are the synaptic weights connecting the inputs to the neuron. An online variant of power iteration algorithm learns the top principal component of the inputs by updating the vector $\mathbf{v}$ as follows:

$$\mathbf{v} \leftarrow \mathrm{v} + \zeta \left(y_t \mathbf{s}_t - y_t^2 \mathbf{v}\right)$$
$$\mathbf{v} \leftarrow \frac{1}{\|\mathbf{v}\|}\mathbf{v}$$

where $\zeta > 0$ is small.

Next, consider a population of $2 \leq N \leq D$ primary neurons with outputs $\mathbf{y}_t \in \mathbb{R}^N$ and feedforward synaptic weight vectors $\mathbf{v}_1, \ldots, \mathbf{v}_N \in \mathbb{R}^D$ connecting the pre-synaptic inputs $\mathbf{s}_t$ to the N neurons. Running N parallel instances of the power iteration algorithm defined above *without* a decorrelation process results in representational collapse, because each synaptic weight vector $\mathbf{v}_i$ converges to the top principal component (Figure 10, orange). We demonstrate that our whitening algorithm via gain modulation readily solves this problem. Here, it is important that the whitening happen on a faster timescale than the principal subspace learning, to avoid collapse (see Lipshutz et al., 2023, for details).

For this simulation, we set $D = 3, N = 2$ and randomly sample i.i.d. pre-synaptic inputs $\mathbf{s}_t \sim \mathcal{N}(\mathbf{0}, \mathrm{diag}(5, 2, 1))$. We randomly initialize two vectors $\mathbf{v}_1, \mathbf{v}_2 \in \mathbb{R}^3$ with i.i.d. Gaussian entries. At each time step t, we project pre-synaptic inputs to form the post-synaptic primary neuron inputs, $\mathbf{x}_t := \left[\mathbf{v}_1^\top \mathbf{s}_t, \mathbf{v}_2^\top \mathbf{s}_t\right]^\top$, forming the input to Algorithm 1. Let $\mathbf{y}_t$ be the primary neuron steady-state output; that is, $\mathbf{y}_t = \left(\mathbf{I}_N + \mathbf{W}\,\mathrm{diag}\,(\mathbf{g})\,\mathbf{W}^\top\right)^{-1} \mathbf{x}_t$ (Equation 8). For $i = 1, 2$, we update $\mathbf{v}_i$ according to the above-defined update rules, with $\zeta = 10^{-3}$. We update the gains $\mathbf{g}$ according to Algorithm 1 with $\eta = 10\zeta$.

To measure the online subspace learning performance, we define

$$\text{Subspace error } := \left\| \mathbf{V}\left(\mathbf{V}^\top\mathbf{V}\right)^{-1}\mathbf{V}^\top - \operatorname{diag}\left([1,1,0]\right)\right\|^2_{\text{Frob}}, \quad \mathbf{V} := [\mathbf{v}_1, \mathbf{v}_2] \in \mathbb{R}^{3\times 2}$$

Figure 10 (blue) shows that our adaptive whitening algorithm with gain modulation successfully facilitates subspace learning and prevents representational collapse.

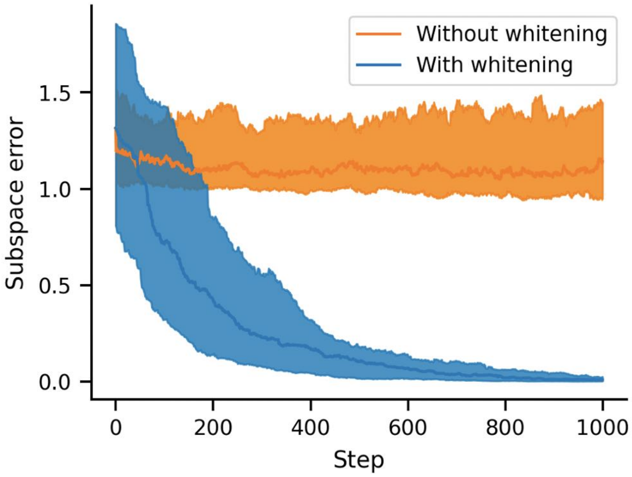

Figure 10. Adaptive symmetric whitening with gain modulation prevents representational collapse during online principal subspace learning. Without whitening, subspace error stabilizes at a non-zero value, indicating that the network has converged to a collapsed representation. Shaded curves are median and [25%, 75%] quantiles over 50 random intializations.

J.2. Generalized adaptive covariance transformations

Our framework for adaptive whitening via gain modulation can easily be generalized to adaptively transform a signal with some initial covariance matrix to one with *any target covariance* (i.e. not just the identity matrix). This demonstrates that our adaptive gain modulation framework has implications beyond statistical whitening. This could, for example, allow online systems to stably maintain some initial/target (non-white) output covariance under changing input statistics (i.e. covariance homeostasis, Westrick et al., 2016; Benucci et al., 2013). The key insight, similar to the main text, is that a full-rank covariance matrix has K_N degrees of freedom, and therefore marginal measurements along K_N distinct axes is necessary and sufficient to represent the matrix (Karl et al., 1994).

Let $\mathbf{C}_{\text{target}}$ be some arbitrary target covariance matrix. Then the general objective is

$$\min_{\{\mathbf{y}_t\}} \langle \|\mathbf{x}_t - \mathbf{y}_t\|_2^2\rangle_t \quad \text{s.t.} \quad \langle \mathbf{y}_t\mathbf{y}_t^\top\rangle_t = \mathbf{C}_{\text{target}}. \tag{26}$$

Following the same logic as in the main text, the Lagrangian becomes

$$\max_{\mathbf{g}} \min_{\{\mathbf{y}_t\}} \langle \ell(\mathbf{x}_t, \mathbf{y}_t, \mathbf{g})\rangle_t, \tag{27}$$

$$\text{where } \ell(\mathbf{x}, \mathbf{y}, \mathbf{g}) := \|\mathbf{x} - \mathbf{y}\|_2^2 + \sum_{i=1}^{K} g_i \left\{(\mathbf{w}_i^\top \mathbf{y})^2 - \sigma_i^2\right\},$$

where $\sigma_i^2 = \mathbf{w}_i^\top \mathbf{C}_{\text{target}} \mathbf{w}_i$ is the marginal variance along the axis spanned by $\mathbf{w}_i$. When $\mathbf{C}_{\text{target}} = \mathbf{I}_N$, then $\sigma_i^2 = 1$ for all i, and this reduces to our original overcomplete whitening objective (Equation 5). The only difference in the recursive algorithm optimizing this generalized objective is the gain update rule,

$$\begin{aligned} g_i &\leftarrow g_i + \frac{\eta}{2}\nabla_{g_i}\ell(\mathbf{x}_t, \bar{\mathbf{y}}_t, \mathbf{g}) \\ &= g_i + \eta\left(\bar{z}_{i,t}^2 - \sigma_i^2\right). \end{aligned} \tag{28}$$

We can interpret this formulation as each interneuron having a pre-determined target input variance (perhaps learned over long time-scales), and adjusting its gains to modulate the joint responses of the primary neurons until its input variance matches the target.