Supplement to "Biological Learning of Irreducible Representations of Commuting Transformations"

Alexander Genkin[*] **David Lipshutz**[†] **Siavash Golkar**[†]

Tiberiu Teşileanu[†] **Dmitri B. Chklovskii**[*,†]

[*]Neuroscience Institute, NYU Langone Medical School
[†]Center for Computational Neuroscience, Flatiron Institute

alexander.genkin@gmail.com
{dlipshutz,sgolkar,ttesileanu,dchklovskii}@flatironinstitute.org

A Proof of Lemma 1

Define the $d \times d$ orthogonal matrix $\mathbf{P}$ by

$$\mathbf{P} := \text{blockdiag}\left(\mathbf{J}, \dots, \mathbf{J}\right), \qquad \mathbf{J} := \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}.$$

Since $\mathbf{P}$ is orthogonal, we have

$$\mathbf{B} = \mathbf{Q}\mathbf{G}\mathbf{Q}^\top = \mathbf{Q}(\mathbf{G}\mathbf{P})(\mathbf{Q}\mathbf{P})^\top = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top,$$

where $\mathbf{U} := \mathbf{Q}$, $\mathbf{\Sigma} := \text{diag}(g_1, g_1, \dots, g_k, g_k)$ and $\mathbf{V} := \mathbf{Q}\mathbf{P}$. We assume, without loss of generality, that $g_1 \geq \cdots \geq g_k \geq 0$. By definition, the $(2i-1)^{\text{st}}$ column vectors of $\mathbf{U}$ and $\mathbf{V}$ and the $2i^{\text{th}}$ column vectors of $\mathbf{U}$ and $\mathbf{V}$ satisfy the relations

$$(\mathbf{u}_{2i-1}, \mathbf{v}_{2i-1}) = (\mathbf{q}_{2i-1}, -\mathbf{q}_{2i}), \qquad (\mathbf{u}_{2i}, \mathbf{v}_{2i}) = (\mathbf{q}_{2i}, \mathbf{q}_{2i+1}), \qquad i = 1, \dots, k.$$

B Convergence proof for SVD algorithms

Assume $\mathbf{B} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$ is SVD, singular values in Σ are non-increasing, and $\mathbf{B}$ is full rank. Singular values of anti-symmetric matrix come in pairs, but we will assume for simplicity that there are no two equal pairs. We prove the convergence by induction, similar to the proof for GHA [2], so we start from the top singular vector pair. Differential equations for the iterations in Eqs. 8 from the main text are

$$\frac{d\mathbf{u}}{dt} = \mathbf{B}\mathbf{v} - (\mathbf{u}^\top \mathbf{B}\mathbf{v})\mathbf{u}, \qquad \frac{d\mathbf{v}}{dt} = -\mathbf{B}\mathbf{u} + (\mathbf{v}^\top \mathbf{B}\mathbf{u})\mathbf{v}. \tag{1}$$

The variables $\mathbf{u}, \mathbf{v}$ can be expanded in the column spaces of orthogonal matrices $\mathbf{U}$ and $\mathbf{V}$: $\mathbf{u} = \mathbf{U}\mathbf{p}$, $\mathbf{v} = \mathbf{V}\mathbf{q}$, where $\mathbf{p}$ and $\mathbf{q}$ are the coefficients of the expansion. Substituting into Eq. (1), we obtain differential equations in the variables $\mathbf{p}, \mathbf{q}$:

$$\frac{d\mathbf{p}}{dt} = \mathbf{\Sigma}\mathbf{q} - \mathbf{p}\mathbf{p}^\top \mathbf{\Sigma}\mathbf{q} \tag{2}$$

$$\frac{d\mathbf{q}}{dt} = \mathbf{\Sigma}\mathbf{p} - \mathbf{q}\mathbf{q}^\top \mathbf{\Sigma}\mathbf{p} \tag{3}$$

For the vector $\mathbf{p}$ we follow the evolution of the ratio of the component p_1 corresponding to the top singular value σ_1 and other components p_k corresponding to singular values $\sigma_k < \sigma_1$. From the

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

equations above:

$$\frac{d(p_k/p_1)}{dt} = \frac{p_k}{p_1}\left(\sigma_k \frac{q_k}{p_k} - \sigma_1 \frac{q_1}{p_1}\right). \tag{4}$$

For every k^{th} component of vectors $\mathbf{p}, \mathbf{q}$ we can calculate:

$$\frac{d(q_k/p_k)}{dt} = \sigma_k \left(1 - (q_k/p_k)^2\right). \tag{5}$$

The solution is

$$\frac{q_k(t)}{p_k(t)} = -\frac{1 + C\exp(2\sigma_k t)}{1 - C\exp(2\sigma_k t)}, \tag{6}$$

and converges to 1 when t goes to infinity since $\sigma_k > 0$. Using this, the expression in parenthesis in Eq. (4) converges to $\sigma_k - \sigma_1$, which is negative, so p_k/p_1 goes to zero for every k where $\sigma_k < \sigma_1$. We are left to show that the norm of the vector is maintained at 1. For that we follow the evolution of $z := \mathbf{u}^\top \mathbf{u}$:

$$\frac{dz}{dt} = 2\mathbf{p}^\top \frac{d\mathbf{p}}{dt} = 2\mathbf{p}^\top(\mathbf{I} - \mathbf{p}\mathbf{p}^\top)\mathbf{B}\mathbf{q} = 2(1-z)\mathbf{p}^\top \mathbf{B}\mathbf{q} = 0$$

since we initialized $z = 1$. We conclude that $\mathbf{u}$ converges to $\mathbf{U}_1$, top left singular vector. Similarly, $\mathbf{v}$ converges to $\mathbf{V}_1$, top right singular vector.

Having the base for the induction, for the induction step i we assume that for all $k < i$ the variables $\mathbf{u}_k, \mathbf{v}_k$ converge correspondingly to $\mathbf{U}_{2k-1}, \mathbf{V}_{2k-1}$, columns of $\mathbf{U}$ and $\mathbf{V}$. Now need to prove that $\mathbf{u}_i, \mathbf{v}_i$ converge to $\mathbf{U}_{2i-1}, \mathbf{V}_{2i-1}$. Differential equations corresponding to Algorithm 1 from the main text are:

$$\frac{d\mathbf{u}_i}{dt} = \mathbf{B}\mathbf{v}_i - \mathbf{u}_i\mathbf{u}_i^\top \mathbf{B}\mathbf{v}_i - \sum_{k<i}\left(\mathbf{u}_k\mathbf{u}_k^\top + \mathbf{v}_k\mathbf{v}_k^\top\right)\mathbf{B}\mathbf{v}_i \tag{7}$$

$$\frac{d\mathbf{v}_i}{dt} = -\mathbf{B}\mathbf{u}_i + \mathbf{v}_i\mathbf{v}_i^\top \mathbf{B}\mathbf{u}_i + \sum_{k<i}\left(\mathbf{u}_k\mathbf{u}_k^\top + \mathbf{v}_k\mathbf{v}_k^\top\right)\mathbf{B}\mathbf{u}_i \tag{8}$$

Substituting as before $\mathbf{u}_i = \mathbf{U}\mathbf{p}_i, \mathbf{v}_i = \mathbf{V}\mathbf{q}_i$ and multiplying from the left the first equation by $\mathbf{U}^\top$ and the second – by $\mathbf{V}^\top$ gives:

$$\frac{d\mathbf{p}_i}{dt} = \mathbf{\Sigma}\mathbf{q}_i - \mathbf{p}_i\mathbf{p}_i^\top \mathbf{\Sigma}\mathbf{q}_i - \sum_{k<i}\left(\mathbf{p}_k\mathbf{p}_k^\top + \mathbf{J}\mathbf{q}_k\mathbf{q}_k^\top\mathbf{J}^\top\right)\mathbf{\Sigma}\mathbf{q}_i \tag{9}$$

$$\frac{d\mathbf{q}_i}{dt} = \mathbf{\Sigma}\mathbf{p}_i - \mathbf{q}_i\mathbf{q}_i^\top \mathbf{\Sigma}\mathbf{p}_i - \sum_{k<i}\left(\mathbf{q}_k\mathbf{q}_k^\top + \mathbf{J}^\top\mathbf{p}_k\mathbf{p}_k^\top\mathbf{J}\right)\mathbf{\Sigma}\mathbf{p}_i \tag{10}$$

where $\mathbf{J} := \mathbf{U}^\top\mathbf{V}$ is a block-diagonal matrix with blocks: $\begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$. The task now is to prove that p_i and q_i converge to unit vector with 1 at the $(2i-1)$-st component, the rest being zeros.

By the assumption of the induction step, $\mathbf{p}_k, \mathbf{q}_k$ for $k < i$ converge to corresponding unit vectors. Then simple matrix calculations show that we have the following convergence, as fast as the slowest of $\mathbf{p}_k, \mathbf{q}_k$:

$$\sum_{k<i}\left(\mathbf{p}_k\mathbf{p}_k^\top + \mathbf{J}\mathbf{q}_k\mathbf{q}_k^\top\mathbf{J}^\top\right) \to \mathbf{\Gamma}, \quad \sum_{k<i}\left(\mathbf{q}_k\mathbf{q}_k^\top + \mathbf{J}^\top\mathbf{p}_k\mathbf{p}_k^\top\mathbf{J}\right) \to \mathbf{\Gamma}, \tag{11}$$

where $\mathbf{\Gamma}$ is diagonal matrix, with $2(i-1)$ ones and the rest zeros. The differential equations now take the form:

$$\begin{aligned} \frac{d\mathbf{p}_i}{dt} &= \mathbf{\Sigma}(\mathbf{I} - \mathbf{\Gamma})\mathbf{q}_i - \mathbf{p}_i\mathbf{p}_i^\top \mathbf{\Sigma}\mathbf{q}_i \\ \frac{d\mathbf{q}_i}{dt} &= \mathbf{\Sigma}(\mathbf{I} - \mathbf{\Gamma})\mathbf{p}_i - \mathbf{q}_i\mathbf{q}_i^\top \mathbf{\Sigma}\mathbf{p}_i \end{aligned} \tag{12}$$

These are somewhat similar to 2, 3, and we proceed similarly to observe:

$$\begin{aligned} \frac{d(q_{ik}/p_{ik})}{dt} &= \sigma_k(1-\gamma_k)\left(1 - (q_{ik}/p_{ik})^2\right) \\ \frac{d(p_{ik}/p_{im})}{dt} &= \frac{p_{ik}}{p_{im}}\left(\sigma_k(1-\gamma_k)\frac{q_{ik}}{p_{ik}} - \sigma_m(1-\gamma_m)\frac{q_{im}}{p_{im}}\right). \end{aligned} \tag{13}$$

From the first of these equations we conclude that q_{ik}/p_{ik} converges to 1 when $k \geq 2i-1$, i.e. $\gamma_k = 0$. Then for $k > 2i-1$ from the second equation:

$$\frac{d(p_{ik}/p_{i,2i-1})}{dt} = \frac{p_{ik}}{p_{i,2i-1}} (\sigma_k - \sigma_{2i-1}), \tag{14}$$

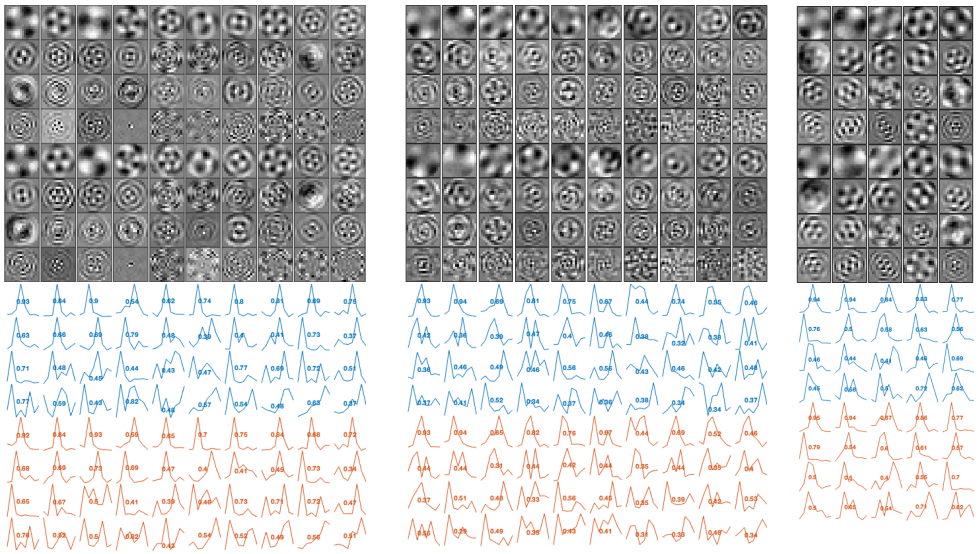
which converges to zero because of the ordering of singular values. For $k < 2i-1$ we have:

$$\frac{d(p_{ik}/p_{i,2i-1})}{dt} = -\sigma_{2i-1} \frac{p_{ik}}{p_{i,2i-1}}, \tag{15}$$

which converges to zero because singular values are positive. As before, we can show that the norm of vector $\mathbf{p}_i$ is maintained at 1, so $p_{i,2i-1}$ converges to 1. The proof for $\mathbf{q}_i$ goes along the same lines.

C Experiments on image rotations and evaluation

Here we present extended experiments on image rotations to give a better idea of the factors influencing the quality of the results. First, we experimented on random images, as in [1]. Images of size 16×16 were generated and, again, rotated by random angles using a bi-linear approximation and ignoring pixels outside the central circle. The top 40 filter pairs obtained from the SVD algorithm trained on a total of 2×10^6 image pairs are shown in Fig. 1(a). We also repeated the experiment on natural images as in the main text, but this time seeking the top 40 pairs of filters; results in Fig. 1(b). A duplicate of 20 top pairs results from the main text are included for visual comparison in Fig. 1(c). As expected, the results on natural images look less clean that those on random images, but visually similar. Also, filters obtained when only 20 of them were sought for in the algorithm run, are better than top 20 from the run where 40 filters were sought. This comparison is confirmed by quantitative evaluation, as described below. All experiments on image rotations were performed on a MacBook Pro with 3.5 GHz Dual-Core Intel Core i7 processor, each took within 10 minutes.



(a) (b) (c)

Figure 1: Filters obtained by SVD algorithm from rotations of 2D images: (a) On random images, 40 top filter pairs; (b) On natural images, 40 top filter pairs; (c) On natural images as in the main text, 20 top filter pairs. Top row: filters as images, corresponding left and right singular vector estimates on the upper and lower half of each chart, respectively. Bottom row: power spectrum of angular component of each filter; numbers indicate how well each filter factorizes into a radial and an angular component. See text for details.

Next we provide a quantitative evaluation of the obtained filters versus theoretical predictions. The subtlety here is that the left and right singular vectors are not uniquely determined—there is a degeneracy due to the fact that there are multiple modes corresponding to the same singular value.

Indeed, the rotation operator in two dimensions has eigenvectors that factorize between a radial part $g(r)$, that is unconstrained, and an angular part which needs to have a sinusoidal shape of a single frequency commensurate with the 2π periodicity of the angle θ:

$$\psi(r, \theta) = g(r)e^{2\pi ik\theta}, \qquad \text{where } k \in \mathbb{Z}. \tag{16}$$

The singular vectors that our algorithm learns are the real and imaginary parts of ψ.

To assess to what extent our filters are of the form (16), we need to separate the radial and angular parts. To do so, we first convert them from Cartesian to polar coordinates, obtaining a matrix Ψ where the row index corresponds to the radial direction and the column index corresponds to the angle. The factorization from eq. (16) implies that Ψ should be rank-1, and thus expressible as a product $\Psi = R\Theta$ of a purely radial component R and a purely angular component Θ.

To assess whether our filters approximate the expected factorization, we perform an SVD of Ψ and calculate the fraction of the variance explained by the top component (that is, the ratio between the square of the top singular value and the sum of the squares of all singular values). A perfect factorization as in eq. (16) would lead to a fraction of explained variance equal to 1, corresponding to a rank-1 Ψ. We show these fractions for the learned filters in the bottom row of Fig. 1.

Finally, we check whether the angular component Θ indeed corresponds to a single sinusoid by taking the Fourier transform of this angular part. We plot the power spectrum obtained from the Fourier transform in the bottom row of Fig. 1. A perfect filter would have a sharp peak at a single frequency in this spectrum.

Not surprisingly, the top filters, corresponding to higher singular values, have sharper spectra and higher explained-variance numbers. Overall, these results confirm what was observed by visual inspection of the filters: the quality of the filters from natural images is somewhat weaker than those from random images, and using a smaller number of filters in the simulation improves the quality of the top filters.

D Reconstruction of transformed image using network outputs

In this section, we verify that the learned representation of our algorithm is informative of the transformation by performing reconstruction of the transformed image. The idea is to learn a function that can predict the transformed image using the initial image together with the output of our algorithm. We can then apply this function to a completely different image. If the learned function performs the expected transformation on this new image, we can conclude that the output of our algorithm is indeed informative of the transformation.

In more detail, as in Appendix C, we run Algorithm 1 from the main text on naturalistic images, setting output dimension to $k = 20$ so that we only recover a strict subset of the filters. Now we train a multi-layer perceptron (MLP) with two hidden layers of rectified linear units. We take the input to be the concatenation of $\mathbf{x}_{t-1}$ and $\hat{\boldsymbol{\theta}}_t$ from our algorithm. We train the MLP to reconstruct $\mathbf{x}_t$; specifically, to minimize the mean-squared error

$$L = \sum_t |f(\mathbf{x}_{t-1}, \hat{\boldsymbol{\theta}}_t) - \mathbf{x}_t|^2.$$

Having trained this MLP, we verify that the transformations are correctly learned. To do so, we apply the transformations f to new images that were not seen during training as $\mathbf{x}_{t-1}$ argument and varying angle as θ_t argument, to obtain a sequence $\tilde{\mathbf{x}}_t$. We use MNIST images for easy visualization of the transformation, and take for ground truth the rotations performed by generic image processing algorithm. The results of this experiment can be seen in Fig. 2. Here, the top row represents $f(\tilde{\mathbf{x}}_{t-1}, \hat{\boldsymbol{\theta}}_t)$ and the bottom row represents the ground truth $\tilde{\mathbf{x}}_t$. Even though we do not use the full set of filters, we see that a reasonably good reconstruction is achieved.

The training of this network was carried out on an NVIDIA Quadro RTX 6000 GPU and took 45 seconds to complete.

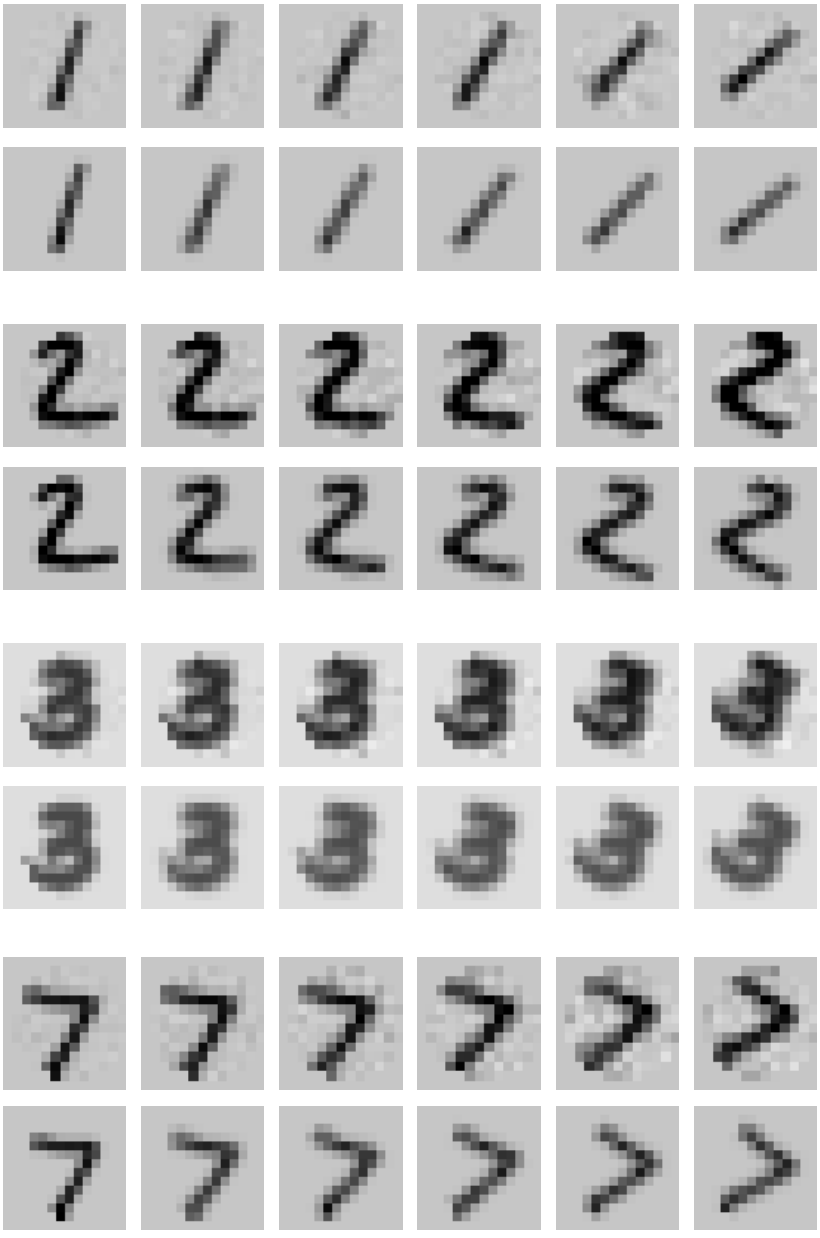


Figure 2: For each digit, the top row is the reconstruction of the transformed image and the bottom row is the ground truth.

References

[1] Taco Cohen and Max Welling. Learning the irreducible representations of commutative lie groups. In *International Conference on Machine Learning*, pages 1755–1763. PMLR, 2014.

[2] Terence D. Sanger. Optimal unsupervised learning in a single-layer linear feedforward neural network. *Neural Networks*, 2(6):459–473, 1989. ISSN 0893-6080. doi: https://doi.org/10.1016/0893-6080(89)90044-0. URL https://www.sciencedirect.com/science/article/pii/0893608089900440.