

# Appendix: The Symmetric Generalized Eigenvalue Problem as a Nash Equilibrium

## A PRELIMINARIES: GENERALIZED EIGENVALUE PROBLEM

The following known properties of the SGEP are useful for our analysis and broadening the scope of SGEP applications.

**Lemma 3** (*B-orthogonality*).  $v_i^\top B v_j = v_j^\top B v_i = 0$  for any distinct pair of generalized eigenvectors of  $Av = \lambda Bv$  where  $A$  is symmetric and  $B$  is symmetric positive definite.

*Proof.* Consider the eigenvalue problem  $B^{-\frac{1}{2}}AB^{-\frac{1}{2}}w = \lambda w$ . Let  $v$  be a generalized eigenvector of the generalized eigenvalue problem  $Av = \lambda'Bv$ . Then the former eigenvalue problem is solved by  $w = B^{\frac{1}{2}}v$ . By inspection,  $B^{-\frac{1}{2}}AB^{-\frac{1}{2}}w = B^{-\frac{1}{2}}AB^{-\frac{1}{2}}B^{\frac{1}{2}}v = B^{-\frac{1}{2}}Av = \lambda'B^{-\frac{1}{2}}Bv = \lambda'B^{\frac{1}{2}}v = \lambda'w$ . Direct computation of the Rayleigh quotients for both problems reveals  $\lambda = \lambda'$ . Note that  $B$  is positive definite, i.e., full-rank, establishing a bijection between  $v$  and  $w$ :  $v = B^{-\frac{1}{2}}w$ . Also, note that  $B^{-\frac{1}{2}}AB^{-\frac{1}{2}}$  is symmetric because  $A$  and  $B$  are symmetric, therefore,  $w$  may be chosen such that  $W^\top W = I$  which implies  $w_i^\top w_j = \delta_{ij} = v_i^\top B^{\frac{1}{2}}B^{\frac{1}{2}}v_j = v_i^\top Bv_j$ , i.e., the generalized eigenvectors are  $B$ -orthogonal.  $\square$

**Proposition 2** (*Similar Matrices*). Given symmetric matrices  $A$  and  $B \succ 0$ , consider the generalized eigenvalue problem  $Av = \lambda'Bv$  with  $\lambda'$  and  $v$  its corresponding generalized eigenvalues and eigenvectors. Then the eigenvectors and eigenvalues of the related eigenvalue problem  $B^{-\frac{1}{2}}AB^{-\frac{1}{2}}w = \lambda w$  are  $w = B^{\frac{1}{2}}v$  and  $\lambda = \lambda'$ .

*Proof.* The relationship between the eigenvectors of the two problems is proven in Lemma 3. The relationship between the eigenvalues can be proven by inspection after calculating the Rayleigh quotients for both problems:

$$\lambda' = \frac{v^\top Av}{v^\top Bv} \quad (8)$$

$$\lambda = \frac{w^\top B^{-\frac{1}{2}}AB^{-\frac{1}{2}}w}{w^\top w} \quad (9)$$

$$= \frac{v^\top B^{\frac{1}{2}}B^{-\frac{1}{2}}AB^{-\frac{1}{2}}B^{\frac{1}{2}}v}{v^\top Bv} \quad (10)$$

$$= \frac{v^\top Av}{v^\top Bv} \quad (11)$$

$$= \lambda'. \quad (12)$$

$\square$

### A.1 COMPUTING SUBSPACE ERROR FOR SGEP

Lemma 3 states that the generalized eigenvectors are  $B$ -orthogonal rather than orthogonal under the standard Euclidean basis. Therefore, we cannot compute subspace error in the same way as is typically done for e.g., singular value decomposition. However, we can exploit Lemma 2 to compute subspace error for the related eigenvalue problem  $B^{-\frac{1}{2}}AB^{-\frac{1}{2}}w = \lambda w$  which *does* have orthogonal eigenvectors due to its symmetry.

Formally, let  $v$  be a solution to the SGEP,  $Av = \lambda'Bv$ . Then by Lemma 2,  $w = B^{1/2}v$  is a solution to  $B^{-1/2}AB^{-1/2}w = \lambda w$ , with eigenvalue  $\lambda = \lambda'$ . Leveraging this equivalence, we can measure subspace error of the SGEP solution by first mapping it to the normalized case and computing subspace error there where  $W$  contains the top- $k$  eigenvectors of  $B^{-1/2}AB^{-1/2}$ . Also let  $\hat{W} = B^{1/2}\hat{V}$  where  $\hat{V}$  contains our top- $k$  approximations. Given the top- $k$  ground truth eigenvectors  $W$

and approximations  $W$ , normalized subspace error can then be computed as  $1 - \frac{1}{k} \text{tr}(U^* P) \in [0, 1]$  where  $U^* = WW^\top$  and  $P = \hat{W}\hat{W}^\top$  (Gemp et al., 2021; Tang, 2019).

## A.2 COURANT-FISCHER MIN-MAX PRINCIPLE

The Courant-Fischer Min-Max principle states that the  $i$ th largest generalized eigenvalue is given by the *minimum* possible Rayleigh quotient within the  $i$ -dimensional subspace  $S$  that captures *maximal* trace (Avron, 2008; Parlett, 1998):

$$v_i = \arg \min_{v_i \in S} \max_{\dim(S)=i} \frac{v_i^\top A v_i}{v_i^\top B v_i}. \quad (13)$$

Note this defines each eigenvalue of the SGEP as the value (at Nash equilibrium) of a min-max (two-player, zero-sum) game rather than the entire set of top- $k$  eigenvectors/eigenvalues as the Nash equilibrium / utility-at-Nash of a  $k$ -player, general-sum game.

## B $\Gamma$ -EIGENGAME IS WELL-POSED

First, we prove  $\Gamma$ -EigenGame suitably captures the top- $k$  SGEP.

**Lemma 1** (Well-posed Utilities). *Given exact parents and assuming the top- $k$  eigenvalues of  $B^{-1}A$  are distinct and positive, the maximizer of player  $i$ 's utility is the unique generalized eigenvector  $v_i$  (up to sign, i.e.,  $-v_i$  is also valid).*

*Proof. Approach:* We will represent each  $\hat{v}_i$  as a linear combination of the true eigenvectors,  $v_p$  for  $p \in \{1, d\}$ . We will then show that maximizing the utility for each player with exact parents is equivalent to solving a linear program. This resulting problem has a unique solution, which is the true eigenvector  $v_i$ .

Assume the parents have been learned exactly and let  $\hat{v}_i = \sum_p w_p v_p$  with  $\|\hat{v}_i\| = 1$  and where  $w_p$  are the weights of the linear combination. Expand and simplify the following expressions that appear in the utility definition with the knowledge that the generalized eigenvectors are guaranteed to be  $B$ -orthogonal, i.e.,  $v_i^\top B v_j = 0$  for all  $i \neq j$  (see Lemma 3 in appendix):

$$\langle \hat{v}_i, B \hat{v}_i \rangle = \left( \sum_p w_p v_p \right)^\top B \left( \sum_l w_l v_l \right) = \sum_p \sum_l w_p w_l v_p^\top B v_l = \sum_p w_p^2 \langle v_p, B v_p \rangle \quad (14)$$

$$\langle \hat{v}_i, A \hat{v}_i \rangle = \left( \sum_p w_p v_p \right)^\top A \left( \sum_l w_l v_l \right) = \sum_p \sum_l \lambda_l w_p w_l v_p^\top B v_l = \sum_p \lambda_p w_p^2 \langle v_p, B v_p \rangle \quad (15)$$

$$\langle \hat{v}_i, B v_j \rangle = \left( \sum_p w_p v_p \right)^\top B v_j = w_j \langle v_j, B v_j \rangle \quad (16)$$

$$\langle \hat{v}_i, A v_j \rangle = \left( \sum_p w_p v_p \right)^\top A v_j = \lambda_j w_j \langle v_j, B v_j \rangle. \quad (17)$$

Plugging these in to the utility function, we find

$$u_i(\hat{v}_i | v_{j < i}) = \frac{\langle \hat{v}_i, A\hat{v}_i \rangle}{\langle \hat{v}_i, B\hat{v}_i \rangle} - \sum_{j < i} \frac{\langle v_j, Av_j \rangle \langle \hat{v}_i, Bv_j \rangle^2}{\langle v_j, Bv_j \rangle^2 \langle \hat{v}_i, B\hat{v}_i \rangle} \quad (18)$$

$$= \frac{\langle \hat{v}_i, A\hat{v}_i \rangle}{\langle \hat{v}_i, B\hat{v}_i \rangle} - \sum_{j < i} \frac{\lambda_j \langle \hat{v}_i, Bv_j \rangle^2}{\langle v_j, Bv_j \rangle \langle \hat{v}_i, B\hat{v}_i \rangle} \quad \langle v_j, Av_j \rangle \rightarrow \langle v_j, \lambda_j Bv_j \rangle \quad (19)$$

$$= \frac{\langle \hat{v}_i, A\hat{v}_i \rangle}{\langle \hat{v}_i, B\hat{v}_i \rangle} - \sum_{j < i} \frac{\langle \hat{v}_i, Av_j \rangle \langle \hat{v}_i, Bv_j \rangle}{\langle v_j, Bv_j \rangle \langle \hat{v}_i, B\hat{v}_i \rangle} \quad \langle \hat{v}_i, \lambda_j Bv_j \rangle \rightarrow \langle \hat{v}_i, Av_j \rangle \quad (20)$$

$$= \frac{1}{\sum_p w_p^2 \langle v_p, Bv_p \rangle} \left[ \sum_l \lambda_l w_l^2 \langle v_l, Bv_l \rangle - \sum_{j < i} \frac{(\lambda_j w_j \langle v_j, Bv_j \rangle)(w_j \langle v_j, Bv_j \rangle)}{\langle v_j, Bv_j \rangle} \right] \quad (21)$$

$$= \sum_l \lambda_l z_l - \sum_{j < i} \lambda_j z_j = \sum_{j \geq i} \lambda_j z_j. \quad (22)$$

where

$$z_j = \frac{w_j^2 \langle v_j, Bv_j \rangle}{\sum_p w_p^2 \langle v_p, Bv_p \rangle} = \frac{w_j^2 b_j^2}{\sum_p w_p^2 b_p^2} = \frac{q_j^2}{\sum_p q_p^2} \quad (23)$$

and  $z \in \Delta^{d-1}$ .

This is a linear optimization problem over the simplex. Given that the eigenvalues are distinct and positive, we have that the unique solution is  $z = e_i$ , the onehot vector with a 1 at index  $i$ .

In order to prove uniqueness of  $w$  (up to sign), we apply Lemma 4, which proves a bijection (up to sign) between  $z$  and  $w$ , completing the proof.  $\square$

**Lemma 4.** Let  $z \in \Delta^{d-1}$  such that  $z_j = \frac{w_j^2 \langle v_j, Bv_j \rangle}{\sum_p w_p^2 \langle v_p, Bv_p \rangle}$  where  $w$  parameterizes the approximation  $\hat{v}_i = \sum_p w_p v_p \in \mathcal{S}^{d-1}$ . There exists a unique bijection (up to sign of  $w_j$ ) between  $z_j$  and  $w_j$ , i.e.,  $w_j = \pm g(z)_j$ .

*Proof.* Let  $b_j = \langle v_j, Bv_j \rangle$  and  $q_j = w_j b_j$  so that  $w_j = q_j / b_j$ . Then  $\hat{v}_i = \sum_p \frac{q_p}{b_p} v_p$ . Also,  $q_j^2 = cz_j$  where  $c = \sum_p q_p^2$  so that  $q_j^2$  is uniquely defined up to a scalar multiple, i.e., its direction is immediately unique by this formula but not its magnitude. Recall  $\langle v_i, Bv_j \rangle = 0$  for all  $i \neq j$  which implies  $V^\top BV$  is diagonal. Therefore, the constraint  $\|\hat{v}_i\| = \|w\|_{V^\top V}^2 = 1$  translates to  $\| \frac{q}{b} \|_{V^\top V}^2 = q^\top (V^\top BV)^{-1/2} (V^\top V) (V^\top BV)^{-1/2} q = q^\top D^{-1} q = 1$ . In other words, whereas an approximate eigenvector for the standard eigenvalue problem can be modeled as choosing a vector on the unit-sphere, an approximate eigenvector for the generalized eigenvalue problem is modeled as choosing a vector on an ellipsoid ( $D$  is positive definite because  $V^\top V$  is symmetric positive definite assuming distinct eigenvalues, and we are given  $B$  is symmetric positive definite). This result uniquely defines a magnitude for  $q$ , therefore, combining it with the previous result uniquely defines  $w_j^2$  from  $q_j^2$  completing the bijection. The only degree of freedom that remains is the sign of  $w_j$  which is expected as both  $v_i$  and  $-v_i$  are valid eigenvectors.  $\square$

Although player  $i$ 's utility  $u_i$  appears abstruse, it actually has a simple explanation and structure.

**Proposition 1** (Utility Shape). *Each player's utility is periodic in the angular deviation ( $\theta$ ) along the sphere. Its shape is sinusoidal, but with its angular axis ( $\theta$ ) smoothly deformed as a function of  $B$ . Most importantly, every local maximum is a global maximum (see Figure 1 for an example).*

*Proof.* Lemma 1 proves each utility function can be represented as a linear function over a simplex  $z \in \Delta^{d-1}$ . Lemma 4 then proves this simplex can be parameterized by a variable  $q$  constrained to an ellipsoid with curvature  $D = \text{diag}(\dots, \langle v_i, Bv_i \rangle, \dots)$ . This matches the analysis of EigenGame exactly, except that  $D = I$  in that previous work. The implication is that each utility function as defined in equation (48) is also a cosine, but with its angular axis deformed according to  $D$ .  $\square$

As an example consider setting

$$A = \begin{bmatrix} 0.77759061 & 0.26842584 \\ 0.26842584 & 0.87788983 \end{bmatrix} \quad B = \begin{bmatrix} 0.2325605 & 0.06042127 \\ 0.06042127 & 0.03241424 \end{bmatrix} \quad (24)$$

and observe the utilities in Figure [1](#)

Our proposed utilities tie nicely back to previous work ([Gemp et al. \(2022\)](#), Appx. J.2) via their gradients and our derived update directions.

**Lemma 5** ( $\Gamma$ -EigenGame Gradient). *The gradient of player  $i$ 's utility with respect to  $\hat{v}_i$  is*

$$2 \times \left[ \frac{(\hat{v}_i^\top B \hat{v}_i) A \hat{v}_i - (\hat{v}_i^\top A \hat{v}_i) B \hat{v}_i}{\langle \hat{v}_i, B \hat{v}_i \rangle^2} - \sum_{j < i} \frac{\hat{\lambda}_j}{\langle \hat{v}_j, B \hat{v}_j \rangle} (\hat{v}_i^\top B \hat{v}_j) \frac{[\langle \hat{v}_i, B \hat{v}_i \rangle B \hat{v}_j - \langle \hat{v}_i, B \hat{v}_j \rangle B \hat{v}_i]}{\langle \hat{v}_i, B \hat{v}_i \rangle^2} \right]. \quad (25)$$

*Proof.* Recall player  $i$ 's utility function:

$$u_i(\hat{v}_i | \hat{v}_{j < i}) = \underbrace{\hat{\lambda}_i}_{\text{reward}} - \sum_{j < i} \underbrace{\hat{\lambda}_j \langle \hat{y}_i, B \hat{y}_j \rangle^2}_{\text{penalty}} \quad \text{where } \hat{y}_i = \frac{\hat{v}_i}{\|\hat{v}_i\|_B}, \quad (26)$$

$$\hat{\lambda}_i = \frac{\langle \hat{v}_i, A \hat{v}_i \rangle}{\langle \hat{v}_i, B \hat{v}_i \rangle}, \text{ and } \|z\|_B = \sqrt{\langle z, B z \rangle}.$$

We will address the gradient of each term in the chain rule in sequence. First consider  $\hat{\lambda}_i$ :

$$\nabla_{\hat{v}_i} \hat{\lambda}_i = \nabla_{\hat{v}_i} \left\{ \frac{\langle \hat{v}_i, A \hat{v}_i \rangle}{\langle \hat{v}_i, B \hat{v}_i \rangle} \right\} = \nabla_{\hat{v}_i} \left\{ \langle \hat{v}_i, A \hat{v}_i \rangle \langle \hat{v}_i, B \hat{v}_i \rangle^{-1} \right\} \quad (27)$$

$$= \frac{2(\hat{v}_i^\top B \hat{v}_i) A \hat{v}_i - 2(\hat{v}_i^\top A \hat{v}_i) B \hat{v}_i}{\langle \hat{v}_i, B \hat{v}_i \rangle^2}. \quad (28)$$

The next term that depends on  $\hat{v}_i$  is  $\langle \hat{y}_i, B \hat{y}_j \rangle^2$  through  $\hat{y}_i$ :

$$\nabla_{\hat{v}_i} \langle \hat{y}_i, B \hat{y}_j \rangle^2 = \nabla_{\hat{v}_i} \left\{ \langle \hat{v}_i, B \hat{y}_j \rangle^2 \langle \hat{v}_i, B \hat{v}_i \rangle^{-1} \right\} \quad (29)$$

$$= 2 \langle \hat{v}_i, B \hat{y}_j \rangle B \hat{y}_j \langle \hat{v}_i, B \hat{v}_i \rangle^{-1} - \langle \hat{v}_i, B \hat{y}_j \rangle^2 \langle \hat{v}_i, B \hat{v}_i \rangle^{-2} (2 B \hat{v}_i) \quad (30)$$

$$= \frac{2 \langle \hat{v}_i, B \hat{y}_j \rangle (\langle \hat{v}_i, B \hat{v}_i \rangle B \hat{y}_j - \langle \hat{v}_i, B \hat{y}_j \rangle B \hat{v}_i)}{\langle \hat{v}_i, B \hat{v}_i \rangle^2} \quad (31)$$

$$= \frac{2 \langle \hat{v}_i, B \hat{v}_j \rangle (\langle \hat{v}_i, B \hat{v}_i \rangle B \hat{v}_j - \langle \hat{v}_i, B \hat{v}_j \rangle B \hat{v}_i)}{\langle \hat{v}_i, B \hat{v}_j \rangle \langle \hat{v}_i, B \hat{v}_i \rangle^2} \quad (32)$$

where we have replaced all  $\hat{y}_j$  terms with  $\frac{\hat{v}_j}{\langle \hat{v}_j, B \hat{v}_j \rangle^{1/2}}$  terms and consolidated the denominators.

Combining these intermediate results, we find

$$\nabla_{\hat{v}_i} u_i = 2 \left[ \frac{(\hat{v}_i^\top B \hat{v}_i) A \hat{v}_i - (\hat{v}_i^\top A \hat{v}_i) B \hat{v}_i}{\langle \hat{v}_i, B \hat{v}_i \rangle^2} - \sum_{j < i} \frac{\hat{\lambda}_j}{\langle \hat{v}_j, B \hat{v}_j \rangle} (\hat{v}_i^\top B \hat{v}_j) \frac{[\langle \hat{v}_i, B \hat{v}_i \rangle B \hat{v}_j - \langle \hat{v}_i, B \hat{v}_j \rangle B \hat{v}_i]}{\langle \hat{v}_i, B \hat{v}_i \rangle^2} \right]. \quad (33)$$

□

**Proposition 3** (Equivalence to EigenGame Unloaded). *The generalized EigenGame pseudogradient in equation [\(72\)](#) is equivalent to the Riemannian gradient in [\(Gemp et al. 2021\)](#) when  $B = I$ .*

*Proof.* In order to compute the Riemannian update direction, we project player  $i$ 's direction onto the tangent space of the unit-sphere by left-multiplying with  $(I - \hat{v}_i \hat{v}_i^\top)$ . Starting with equation [\(12\)](#),

we find

$$\tilde{\nabla}_i = \overbrace{(\hat{v}_i^\top B \hat{v}_i) A \hat{v}_i - (\hat{v}_i^\top A \hat{v}_i) B \hat{v}_i}^{\text{reward}} - \sum_{j < i} \overbrace{(\hat{v}_i^\top A \hat{y}_j) [\langle \hat{v}_i, B \hat{v}_i \rangle B \hat{y}_j - \langle \hat{v}_i, B \hat{y}_j \rangle B \hat{v}_i]}^{\text{penalty}} \quad (34)$$

$$= A \hat{v}_i - (\hat{v}_i^\top A \hat{v}_i) \hat{v}_i - \sum_{j < i} (\hat{v}_i^\top A \hat{v}_j) [\hat{v}_j - \langle \hat{v}_i, \hat{v}_j \rangle \hat{v}_i] \quad (35)$$

$$= (I - \hat{v}_i \hat{v}_i^\top) [A \hat{v}_i - \sum_{j < i} (\hat{v}_i^\top A \hat{v}_j) \hat{v}_j] = (I - \hat{v}_i \hat{v}_i^\top) \tilde{\nabla}_i^{\mu-EG}. \quad (36)$$

□

**Proposition 4** ( $\Gamma$ -EigenGame Utilities as Deflated Rayleigh Quotients). *The generalized EigenGame utilities defined in equation (48) can also be derived from the perspective of maximizing the Rayleigh quotients of a deflated matrix assuming exact parents.*

*Proof.* Deflating a matrix means to modify the matrix such that the spectrum corresponding to a certain subspace of the matrix is zero. For example, in the case of the SGEP, the matrix  $A$  can be deflated to produce a matrix  $\tilde{A} = (I - B \frac{v_j v_j^\top}{\|v_j\|_B^2}) A$  such that any vector in the span of eigenvector  $v_j$  achieves zero eigenvalue:

$$\tilde{A}(w_j v_j) = w_j (I - B \frac{v_j v_j^\top}{\|v_j\|_B^2}) A v_j \quad (37)$$

$$= w_j A v_j - w_j B v_j \frac{v_j^\top A v_j}{\|v_j\|_B^2} \quad (38)$$

$$= w_j \lambda_j B v_j - w_j \lambda_j B v_j \frac{v_j^\top B v_j}{\|v_j\|_B^2} \text{ apply rule } A v_j = \lambda_j B v_j \quad (39)$$

$$= w_j \lambda_j B v_j (1 - \frac{v_j^\top B v_j}{\|v_j\|_B^2}) \quad (40)$$

$$= 0 \quad (41)$$

where  $\|v_j\|_B^2 = v_j^\top B v_j$  and  $w_j$  is an arbitrary scalar.

This is useful because it allows us to construct a top- $k$  solver by induction: repeatedly deflate a matrix to ignore the top- $(j < i)$  eigenvectors and then deploy a top-1 solver on the deflated matrix to find the  $i$ th eigenvector. To that end, we can construct the following deflation matrix:

$$\tilde{A}_i = (I - \sum_{j < i} B \frac{v_j v_j^\top}{\|v_j\|_B^2}) A. \quad (42)$$

Note that this definition assumes the parents eigenvectors are exact. If they are approximate, this may not act as a deflation in the precise sense. For example, consider defining  $\tilde{A}$  with approximate  $\hat{v}_j$  instead of exact  $v_j$ . Now let  $\hat{v}_j = v_1$  for all  $j$ . If one repeats the analysis above for a vector in the span of  $v_1$ , they would find that equation (40) becomes  $w_1 \lambda_1 B v_1 (1 - (i-1) \frac{v_1^\top B v_1}{\|v_1\|_B^2}) = w_1 \lambda_1 B v_1 (2-i)$ , i.e., it results in an eigenvalue of  $(2-i)\lambda_1$ . This is why the effect of this matrix is more accurately described via penalties. We will clarify this connection next.

With the above preliminaries taken care of, we will now show how to derive our utilities via a deflation perspective. Initially, we will assume exact parents,  $\hat{v}_{j<i} = v_{j<i}$ .

$$u_i(\hat{v}_i | v_{j<i}) = \frac{\langle \hat{v}_i, \tilde{A}_i \hat{v}_i \rangle}{\langle \hat{v}_i, B \hat{v}_i \rangle} \quad (43)$$

$$= \frac{\langle \hat{v}_i, (I - \sum_{j<i} B \frac{v_j v_j^\top}{\|v_j\|_B^2}) A \hat{v}_i \rangle}{\langle \hat{v}_i, B \hat{v}_i \rangle} \quad (44)$$

$$= \frac{\langle \hat{v}_i, A \hat{v}_i \rangle}{\langle \hat{v}_i, B \hat{v}_i \rangle} - \sum_{j<i} \frac{\langle \hat{v}_i, B v_j v_j^\top A \hat{v}_i \rangle}{\|v_j\|_B^2 \langle \hat{v}_i, B \hat{v}_i \rangle} \text{ expand sum} \quad (45)$$

$$= \frac{\langle \hat{v}_i, A \hat{v}_i \rangle}{\langle \hat{v}_i, B \hat{v}_i \rangle} - \sum_{j<i} \frac{\langle \hat{v}_i, B v_j \rangle \langle \hat{v}_i, A v_j \rangle}{\langle v_j, B v_j \rangle \langle \hat{v}_i, B \hat{v}_i \rangle} \text{ split \& transpose inner product} \quad (46)$$

$$= \frac{\langle \hat{v}_i, A \hat{v}_i \rangle}{\langle \hat{v}_i, B \hat{v}_i \rangle} - \sum_{j<i} \frac{\lambda_j \langle \hat{v}_i, B v_j \rangle^2}{\langle v_j, B v_j \rangle \langle \hat{v}_i, B \hat{v}_i \rangle} \text{ apply rule } A v_j = \lambda_j B v_j \quad (47)$$

$$= \underbrace{\hat{\lambda}_i}_{\text{reward}} - \sum_{j<i} \underbrace{\lambda_j \langle \hat{y}_i, B y_j \rangle^2}_{\text{penalty}} \quad \text{where } \hat{y}_i = \frac{\hat{v}_i}{\|\hat{v}_i\|_B}, \quad (48)$$

$$\hat{\lambda}_i = \frac{\langle \hat{v}_i, A \hat{v}_i \rangle}{\langle \hat{v}_i, B \hat{v}_i \rangle}, \text{ and } \|z\|_B = \sqrt{\langle z, B z \rangle}.$$

If we then relax our assumption and allow  $v_j$  to be approximate ( $v_j \rightarrow \hat{v}_j$ ) we recover our utilities in equation (48).  $\square$

## C SMOOTH AND UNBIASED

In order to prove asymptotic convergence of  $\gamma$ -EigenGame in the deterministic setting, we establish the following lemmas.

**Lemma 6.** *The update  $\tilde{\nabla}_i$  in equation (7) is smooth.*

*Proof.* The reward terms are polynomial in  $\hat{v}_i$  and therefore smooth (analytic). The numerators of the penalty terms are also polynomial in  $\hat{v}_i$  and  $\hat{v}_j$ , however, the denominator includes a scalar  $\langle \hat{v}_j, B \hat{v}_j \rangle$ . Given  $B \succ 0$ , this term is guaranteed to be greater than the minimum eigenvalue of  $B$  (which is positive), thereby ensuring the penalty terms are non-singular. So these terms are also smooth.  $\square$

Instead of proving the following lemmas directly for Algorithm 1 (the deterministic variant), we prove them for Algorithm 2, which subsumes Algorithm 1 ( $\rho = 0, \gamma_t = 1, b = n, M = 1$ ).

The following two lemmas are proven in the single proof below. Note Lemma 7 is essentially a restatement of Lemma 2 from the main body.

**Lemma 7.** *The unique stable fixed point (up to sign of  $\hat{v}_j$ ) of Algorithm 2 run with exact expectations (e.g.,  $n' = n$  where  $n$  is the full dataset size) is  $\hat{v}_j = v_j$  for all  $j \in \{1, \dots, k\}$ , i.e., the top- $k$  generalized eigenvectors.*

**Lemma 8.** *Algorithm 2's updates are asymptotically unbiased.*

*Proof.* The proof is constructed sequentially by proving each update process has a unique stable fixed point conditioned on the previous updates' fixed points defined by the hierarchy imposed on the players. We explain how we are able to address constructing unbiased estimates of each update as well, thereby supporting a stochastic, asymptotic convergence proof.

The proof begins by considering the updates of the first player on  $\hat{v}_1$ . Player 1 is unique in that it pays no penalties for aligning with other players. Its update consists of the reward terms only, which comprises an unbiased estimate assuming independent, unbiased estimates for  $A$  and  $B$  (i.e., these are constructed with independent minibatches). Player 1's update simply performs Riemannian

gradient ascent on its utility function. Proposition 1 proves that every maximum of this function is a global maximum (in addition, it contains no saddle points). Therefore, the only stable fixed point for  $\hat{v}_1$  is  $v_1$ .

Next, we consider player 1's update to  $[B\hat{v}]_1$ . Given we just showed  $\hat{v}_1$ 's stable fixed point is  $v_1$ , and this update is simply a running average, its unique stable fixed point is  $Bv_1$ .

We now consider player 2's update, which includes penalty terms. Plugging  $[B\hat{v}]_1$ 's unique stable fixed point into these penalty terms, and again assuming independent, unbiased estimates for  $A$  and  $B$ , allows us to construct an unbiased estimate of the penalty terms. Similarly to player 1's analysis, player 2's update performs Riemannian gradient ascent on a utility function with a unique stable fixed point,  $\hat{v}_2 = v_2$ .

The proof then proceeds repeating the same arguments, alternating between proving the unique stable fixed point of each  $[B\hat{v}]_j = Bv_j$  and  $\hat{v}_j = v_j$ .  $\square$

## D ERROR PROPAGATION

An error propagation analysis is necessary to rule out the scenario where an arbitrary (unbounded) level of precision is required by the parents to ensure any progress towards the true solution can be made by the children. In other words, we show that as the parents near their true solutions, the children may near theirs as well.

As before in Appx. C we will perform this analysis for the stochastic version of the algorithm (Algorithm 2), but note that this subsumes the analysis for the deterministic version (where  $[B\hat{v}]_j$  is replaced by the exact  $Bv_j$  with zero error).

Player 1's update of  $\hat{v}_1$  is unbiased without any assumptions on the state of any of the other player vectors  $\hat{v}_{j>1}$  or auxiliary variables  $[B\hat{v}]_{j\geq 1}$ . We would like to understand how transient error in  $\hat{v}_1$  propagates through to these other variables. The updates of player 1's children depend on  $[B\hat{v}]_1$ , so we analyze the effect on it first. Note that the error propagation analysis naturally repeats as we progress down the hierarchy of players, so we analyze how error in  $\hat{v}_i$  propagates through to  $[B\hat{v}]_i$  and then onto  $\hat{v}_{j>i}$ . Interestingly, step 2 of the following proof suggests the error in the  $\hat{v}_i$  must fall below  $\frac{1}{\kappa}$  before any increase in accuracy of the parents helps to improve accuracy in the children. This result mirrors that of (Gemp et al., 2021) (see their Appendix F).

**Theorem 3.** *An  $\mathcal{O}(\epsilon)$  angular error in the parent propagates to an  $\mathcal{O}(\epsilon^{\frac{1}{2}})$  upper bound on the angular error of the child's solution.*

*Proof.* The proof proceeds in three steps:

1.  $\mathcal{O}(\epsilon)$  angular error of parent  $v_i \implies \mathcal{O}(\epsilon)$  Euclidean error of parent  $v_i$
2.  $\mathcal{O}(\epsilon)$  Euclidean error of parent  $v_i \implies \mathcal{O}(\epsilon)$  Euclidean error of norm of child  $v_{j>i}$ 's gradient
3.  $\mathcal{O}(\epsilon)$  Euclidean error of norm child  $v_{j>i}$ 's gradient + instability of minima at  $v_{k\neq j} \implies \mathcal{O}(\epsilon)$  angular error of child  $v_{j>i}$ 's solution assuming  $B = I$ .
4.  $\mathcal{O}(\epsilon)$  angular error of child  $v_{j>i}$ 's solution assuming  $B = I \implies \mathcal{O}(\epsilon^{\frac{1}{2}})$  angular error of child  $v_{j>i}$ 's solution for a general  $B \succ 0$ .

**1** As in  $\mu$ -EigenGame, an *angular error* of  $\epsilon$  in the parent translates to  $\epsilon$  *Euclidean error*. The proof is exactly the same as in (Gemp et al., 2021), repeated here for convenience. Angular error in the parent can be converted to Euclidean error by considering the chord length between the mis-specified parent and the true parent direction. The two vectors plus the chord form an isosceles triangle with the relation that chord length  $l = 2 \sin(2\epsilon)$  is  $\mathcal{O}(\epsilon)$  for  $\epsilon \ll 1$ .

**2** Next, write the mis-specified parents as  $\hat{v}_i = v_i + w_i$  where  $\|w_i\|$  is  $\mathcal{O}(\epsilon_i)$  as we just explained.

Now consider the fixed point of the auxiliary variable's update:  $[Bv]_i = B(v_i + w_i) = Bv_i + Bw_i$ . Hence any mis-specification in the parent  $\hat{v}_i$  appears as a mis-specification of the auxiliary variable's

fixed point by  $Bw_i$ , which is  $\mathcal{O}(\lambda_{max}\epsilon)$  where  $\lambda_{max}$  is the maximum eigenvalue (spectral radius) of  $B$ . Assume the auxiliary variable is mis-specified by an additional error ( $q_i$  where  $\|q_i\|$  is  $\mathcal{O}(\epsilon'_i)$ ) representing failure to precisely reach the perturbed fixed point  $B(v_i + w_i)$ , i.e.,  $[B\hat{v}]_i = B(v_i + w_i) + q_i$ .

The auxiliary variable impacts the update of  $\hat{v}_{j>i}$  through  $\hat{y}_i$  and similarly  $[B\hat{y}]_i$ :

$$\hat{y}_i = \frac{\hat{v}_i}{\sqrt{[\langle \hat{v}_i, [B\hat{v}]_i \rangle]_\rho}} \quad (49)$$

$$= \frac{v_i + w_i}{\sqrt{[\langle \hat{v}_i, Bv_i \rangle + \langle \hat{v}_i, Bw_i \rangle + \langle \hat{v}_i, q_i \rangle]_\rho}} \quad (50)$$

$$= \frac{v_i + w_i}{\sqrt{[\langle v_i, Bv_i \rangle + 2\langle v_i, Bw_i \rangle + \langle w_i, Bw_i \rangle + \langle v_i, q_i \rangle + \langle w_i, q_i \rangle]_\rho}} \quad (51)$$

$$= cy_i + \frac{w_i}{\sqrt{[\langle v_i, Bv_i \rangle + 2\langle v_i, Bw_i \rangle + \langle w_i, Bw_i \rangle + \langle v_i, q_i \rangle + \langle w_i, q_i \rangle]_\rho}} \quad (52)$$

$$= cy_i + e_i \quad (53)$$

In order to bound this term, we make a few mild assumptions.

- Assume  $\rho$  is less than  $\lambda_{min}$  as stated earlier and, in particular, less than the lower bound.
- Assume  $\epsilon'_i$  is  $\mathcal{O}(\epsilon_i)$  to ease the exposition.
- Also, w.l.o.g., assume  $\lambda_{max} > 1$ ; if not, we can simply scale the problem such that it is true.

Let  $\kappa = \frac{\lambda_{max}}{\lambda_{min}}$  be the condition number of  $B$ , and note that the error term in the denominator is bounded by the spectrum of  $B$ :

$$[\langle \hat{v}_i, [B\hat{v}]_i \rangle]_\rho \leq \langle v_i, Bv_i \rangle + 2\lambda_{max}\epsilon_i + \lambda_{max}\epsilon_i^2 + \epsilon'_i + \epsilon_i\epsilon'_i \quad (54)$$

$$\stackrel{1 \leq \lambda_{max} \leq \kappa \langle v_i, Bv_i \rangle}{\leq} \langle v_i, Bv_i \rangle (1 + 2\kappa\epsilon_i + \kappa\epsilon_i^2 + \kappa\epsilon'_i + \kappa\epsilon_i\epsilon'_i) \quad (55)$$

$$\stackrel{\epsilon' \text{ is } \mathcal{O}(\epsilon)}{\leq} \langle v_i, Bv_i \rangle (1 + 3\kappa\epsilon_i + 2\kappa\epsilon_i^2) \quad (56)$$

and vice versa for the lower bound, which implies

$$(57)$$

$$[\langle \hat{v}_i, [B\hat{v}]_i \rangle]_\rho \in \langle v_i, [Bv]_i \rangle \left[ (1 - 3\kappa\epsilon_i - 2\kappa\epsilon_i^2), (1 + 3\kappa\epsilon_i + 2\kappa\epsilon_i^2) \right]. \quad (58)$$

Then

$$c \in \left[ \frac{1}{\sqrt{1 + 3\kappa\epsilon_i + 2\kappa\epsilon_i^2}}, \frac{1}{\sqrt{1 - 3\kappa\epsilon_i - 2\kappa\epsilon_i^2}} \right]. \quad (59)$$

Note that if  $\epsilon_i \ll \frac{1}{\kappa}$ , then  $c$  is  $\mathcal{O}(1)$ . Note this condition also implies  $e_i$  is  $\mathcal{O}(\epsilon'_i)$ . We will use these facts later.

Now we are prepared to consider the norm of the difference,  $d_j$ , between the Riemannian<sup>4</sup> update to  $\hat{v}_j$  with exact parents and auxiliary variables versus the actual inexact Riemannian update. Let  $\Delta_j^R$  be defined as in line 12 of Algorithm 2. Define  $\hat{\Delta}_j^R$  to be the same except with  $\hat{y}_l$  and  $[B\hat{y}]_l$  terms replaced by their true solution counterparts  $y_l$  and  $[By]_l$ . Note that  $\Delta_j^R$  and  $\hat{\Delta}_j^R$  already live in the tangent space of the unit-sphere at  $\hat{v}_j$ . Then the norm of the difference between the two update

<sup>4</sup>The Riemannian update projects the vanilla update onto the tangent space of the sphere.



directions is upper bounded as

$$\|d_j\| = \|\Delta_j^R - \bar{\Delta}_j^R\| \quad (60)$$

$$\leq \left\| \sum_{l < i} \left[ (\hat{v}_j^\top A \hat{y}_l) [\langle \hat{v}_j, B \hat{v}_j \rangle [B \hat{y}]_l - \langle \hat{v}_j, [B \hat{y}]_l \rangle B \hat{v}_j] - \dots \right] \right\| \quad (61)$$

$$\leq \sum_{l < i} \left\| \left[ (\hat{v}_j^\top A \hat{y}_l) [\langle \hat{v}_j, B \hat{v}_j \rangle [B \hat{y}]_l - \langle \hat{v}_j, [B \hat{y}]_l \rangle B \hat{v}_j] - \dots \right] \right\|. \quad (62)$$

Recall  $q_i$  is the error associated with suboptimality of  $[B \hat{v}]_i$  and propoagates to  $[B \hat{y}]_i$  as defined on line 9 of Algorithm 2. Let

$$p_i = \frac{q_i}{\sqrt{[\langle v_i, B v_i \rangle + 2 \langle v_i, B w_i \rangle + \langle w_i, B w_i \rangle + \langle v_i, q_i \rangle + \langle w_i, q_i \rangle]_\rho}}. \quad (63)$$

By similar arguments used to bound  $e_i$ ,  $\|p_i\|$  is  $\mathcal{O}(\epsilon'_i)$  if  $\epsilon_i \ll \frac{1}{\kappa}$ .

Bounding the summand in equation (62), we find

$$\|(\hat{v}_j^\top A \hat{y}_l) [\langle \hat{v}_j, B \hat{v}_j \rangle [B \hat{y}]_l - \langle \hat{v}_j, [B \hat{y}]_l \rangle B \hat{v}_j] - (\hat{v}_j^\top A y_l) [\langle \hat{v}_j, B \hat{v}_j \rangle [B y]_l - \langle \hat{v}_j, [B y]_l \rangle B \hat{v}_j]\| \quad (64)$$

$$= \|(c \hat{v}_j^\top A y_l + \hat{v}_j^\top A e_l) [\langle \hat{v}_j, B \hat{v}_j \rangle (c B y_l + B e_l + p_l) - \langle \hat{v}_j, (c B y_l + B e_l + p_l) \rangle B \hat{v}_j] - \dots\| \quad (65)$$

$$= \|(c^2 - 1)(\hat{v}_j^\top A y_l) [\langle \hat{v}_j, B \hat{v}_j \rangle B y_l - \langle \hat{v}_j, B y_l \rangle B \hat{v}_j] + \mathcal{O}(\epsilon)\|. \quad (66)$$

Recall equation (59) and note that

$$c^2 - 1 \in \left\{ \frac{-3\kappa\epsilon_i - 2\kappa\epsilon_i^2}{1 + 3\kappa\epsilon_i + 2\kappa\epsilon_i^2}, \frac{3\kappa\epsilon_i + 2\kappa\epsilon_i^2}{1 - 3\kappa\epsilon_i - 2\kappa\epsilon_i^2} \right\} \quad (67)$$

which has norm  $|c^2 - 1| = \mathcal{O}(\kappa\epsilon_i)$ . Therefore, taking into account the impact of the other  $A$  and  $B$  terms,  $\|d_j\|$  is upper bounded by  $\mathcal{O}(i\kappa\sigma(A)\lambda_{\max}^2\epsilon_i)$  where  $\sigma(A)$  is the spectral radius of  $A$ .

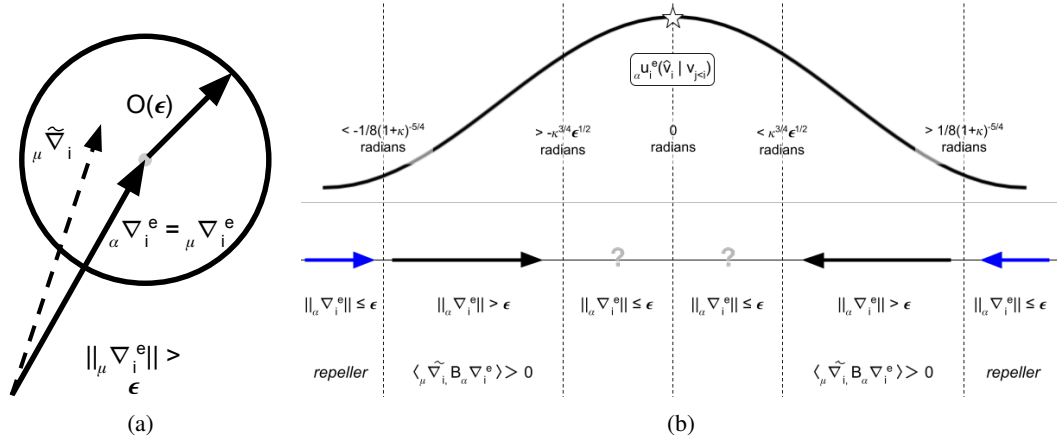


Figure 5: (a) Close in Euclidean distance can imply close in angular distance if the vectors are long enough (reprinted with permission from (Gemp et al., 2021)). (b) The stable region may consist of an  $\mathcal{O}(\kappa^{3/4}\epsilon^{1/2})$  ball around the true optimum as  $\epsilon \rightarrow 0$ .

3 We can reuse the analysis of (Gemp et al., 2021) to understand how a change in the norm of the vector field relates to a change in the location of the fixed point. This is because the Riemannian update direction of our proposed method with exact parents and  $B = I$  is equivalent to the Riemannian update direction in (Gemp et al., 2021) (simply left-multiply their equation (4) by  $(I - \hat{v}_i \hat{v}_i^\top)$  to compute their Riemannian update). Therefore, as in this prior work, an error in the gradient norm translates to the same order of error in angular distance to the true fixed point (inflated by a finite scaling dependent on the spectrum of  $B$ —accounted for in Step 4 next).

Also, the region around any generalized eigenvector  $v_{l \neq j}$  is unstable and this is because the Riemmanian Hessian at that point is positive (this implies instability because we are maximizing). We can reason that the Riemmanian Hessian is positive by appealing to the fact that the Riemmanian Hessian of our generalized EVP utilities is related to the Hessian of prior work by a warping defined by the positive definite matrix  $B$  (for a visual, see Figure 1; for math, see Lemmas 1 and 4).

In contrast to this prior work, the generalized eigenvectors are more generally,  $B$ -orthogonal (see Lemma 3). By Lemma 9 the angular distance between generalized eigenvectors is finite and depends on the condition number of  $B$ . Therefore, there exists a small enough  $\epsilon$  such that an  $\epsilon$ -ball around any *unstable* region and an  $\epsilon$ -ball around the *stable* region no longer overlap.

4 Lastly, Lemma 10 proves that an  $\epsilon_i < 1$  angular error assuming  $B = I$  can be increased to at most  $\kappa^{\frac{3}{4}} \epsilon_i^{\frac{1}{2}}$  if  $B$  is relaxed to be any symmetric positive definite matrix with condition number  $\kappa$ .  $\square$

**Lemma 9.** *The angle between a pair of orthonormal vectors when instead measured under a general positive definite matrix  $C$  is lower bounded by  $\frac{1}{8}(1 + \kappa)^{-\frac{5}{4}}$  where  $\kappa$  is the condition number of  $C$ , i.e., if  $\langle v_i, v_j \rangle = 0$ , then  $|\theta| = \arccos \left( \frac{|\langle C^{\frac{1}{2}} v_i, C^{\frac{1}{2}} v_j \rangle|}{\|C^{\frac{1}{2}} v_i\| \|C^{\frac{1}{2}} v_j\|} \right) > \frac{1}{8}(1 + \kappa)^{-\frac{5}{4}}$  radians.*

*Proof.* The angle between two vectors is a function of their relation to each other in the two-dimensional plane defined by their pair. Therefore, without loss of generality, consider two vectors  $u = \begin{bmatrix} 1 & 0 \end{bmatrix}$  and  $v = \begin{bmatrix} 0 & 1 \end{bmatrix}$  and consider the effect of an arbitrary positive definite matrix  $\hat{C}$  on their angle. For ease of exposition, denote  $\tau = \kappa^{\frac{1}{2}}$  the condition number of  $C^{\frac{1}{2}}$ .

Let  $\hat{C}^{\frac{1}{2}} = \begin{bmatrix} a & c \\ c & b \end{bmatrix}$  be the unique positive definite square root of  $\hat{C}$  where  $a$  and  $b$  are positive and the determinant  $ab - c^2 > \gamma > 0$ . We aim to show that the magnitude of the angle between  $u$  and  $v$  under the generalized inner product  $\langle \cdot, \cdot \rangle_C$  is lower bounded by a finite, positive quantity dependent on the properties of  $C$ .

Consider

$$\hat{C}^{\frac{1}{2}} u = \begin{bmatrix} a \\ c \end{bmatrix}, \quad \hat{C}^{\frac{1}{2}} v = \begin{bmatrix} c \\ b \end{bmatrix}, \quad (68)$$

$$\|\hat{C}^{\frac{1}{2}} u\| = \sqrt{a^2 + c^2}, \quad \|\hat{C}^{\frac{1}{2}} v\| = \sqrt{c^2 + b^2}, \quad (69)$$

$$\langle \hat{C}^{\frac{1}{2}} u, \hat{C}^{\frac{1}{2}} v \rangle = c(a + b). \quad (70)$$

Then

$$\frac{\langle \hat{C}^{\frac{1}{2}} u, \hat{C}^{\frac{1}{2}} v \rangle}{\|C^{\frac{1}{2}} u\| \|C^{\frac{1}{2}} v\|} = \frac{c(a + b)}{\sqrt{(a^2 + c^2)(c^2 + b^2)}} \quad (71)$$

$$\begin{aligned} & \text{ratio is inc. in } c \text{ for } ab > c^2 \\ & \quad \underbrace{\quad}_{<} \quad \frac{\sqrt{ab - \gamma}(a + b)}{\sqrt{(a^2 + ab - \gamma)(ab + b^2 - \gamma)}} \end{aligned} \quad (72)$$

$$\begin{aligned} & \text{div. num. \& den. by } \frac{1}{a^2} \\ & \quad \underbrace{\quad}_{=} \quad \frac{\sqrt{\frac{b}{a} - \frac{\gamma}{a^2}}(1 + \frac{b}{a})}{\sqrt{(1 + \frac{b}{a} - \frac{\gamma}{a^2})(\frac{b}{a} + (\frac{b}{a})^2 - \frac{\gamma}{a^2})}} \end{aligned} \quad (73)$$

$$\begin{aligned} & \text{ratio is inc. in } \frac{b}{a} \\ & \quad \underbrace{\quad}_{\leq} \quad \frac{\sqrt{\tau - \frac{\gamma}{a^2}}(1 + \tau)}{\sqrt{(1 + \tau - \frac{\gamma}{a^2})(\tau + \tau^2 - \frac{\gamma}{a^2})}} \end{aligned} \quad (74)$$

$$\begin{aligned} & \text{ratio is inc. in } \tau - \frac{\gamma}{a^2} \\ & \quad \underbrace{\quad}_{<} \quad \frac{\sqrt{\tau - \frac{\gamma}{a^2}}(1 + \tau)}{\sqrt{(1 + \tau - \frac{\gamma}{a^2})(\tau + \tau^2 - \frac{\gamma}{a^2})}}. \end{aligned} \quad (75)$$

Note that  $\frac{b}{a}$  is a lower bound on the condition number of  $\hat{C}$ ;  $\frac{b}{a}$  is equal to the condition number of  $\hat{C}^{\frac{1}{2}}$  when  $c = 0$  (assuming  $b > a$ , otherwise,  $\frac{b}{a} < \frac{a}{b}$  clearly), and the condition number can only increase as  $c$  deviates from 0. Lastly, note that the condition number of  $\hat{C}^{\frac{1}{2}}$  is upper bounded by  $\tau$ , the condition number of  $C^{\frac{1}{2}}$ . Recall, by replacing  $\frac{b}{a}$  with a larger number  $\tau$  and  $\frac{\gamma}{a^2}$  with a strictly smaller number  $\frac{\gamma}{tr^2}$ ,  $\frac{b}{a} - \frac{\gamma}{a^2}$  implies that  $\tau > \frac{\gamma}{tr^2}$ .

Now consider

$$\left( \frac{\langle C^{\frac{1}{2}}u, C^{\frac{1}{2}}v \rangle}{\|C^{\frac{1}{2}}u\| \|C^{\frac{1}{2}}v\|} \right)^2 < \frac{(\tau - \frac{\gamma}{tr^2})(1 + \tau)^2}{(1 + \tau - \frac{\gamma}{tr^2})(\tau + \tau^2 - \frac{\gamma}{tr^2})} \quad (76)$$

$$= \frac{(\tau - \frac{\gamma}{tr^2})(1 + \tau)^2}{(\tau - \frac{\gamma}{tr^2})(1 + \tau)^2 + (\frac{\gamma}{tr^2})^2} \quad (77)$$

$$= 1 - \frac{(\frac{\gamma}{tr^2})^2}{(\tau - \frac{\gamma}{tr^2})(1 + \tau)^2 + (\frac{\gamma}{tr^2})^2} \quad (78)$$

$$\leq 1 - \frac{(\frac{\gamma}{tr^2})^2}{(\tau)(1 + \tau)^2 + (1 + \tau)^2} \quad (79)$$

$$= 1 - \frac{(\frac{\gamma}{tr^2})^2}{(1 + \tau)^3}. \quad (80)$$

We can also simplify the fraction  $\frac{\gamma}{tr^2}$ .  $\gamma$  is a lower bound on the determinant, which is equal to the product of eigenvalues of  $\hat{C}^{\frac{1}{2}}$ . This is lower bounded by  $\lambda_{min}^2$  for any two-dimensional subspace, therefore,  $\lambda_{min}^2 < \gamma$ . Furthermore, the trace of the matrix is equal to the sum of its eigenvalues. This is upper bounded by  $2\lambda_{max}$  for any two-dimensional subspace, therefore  $\frac{\gamma}{tr^2} > \frac{\lambda_{min}^2}{4\lambda_{max}^2} = \frac{1}{4\tau^2} > \frac{1}{4(\tau+1)^2}$ .

Note that  $|\theta| \geq |\sin(\theta)|$ . Therefore,

$$|\theta| \geq |\sin(\theta)| = \sqrt{1 - \cos^2(\theta)} \quad (81)$$

$$> \sqrt{\frac{(\frac{\gamma}{tr^2})^2}{(1 + \tau)^3}} \quad (82)$$

$$> \frac{1}{2} \sqrt{(1 + \tau)^{-5}} \quad (83)$$

$$= \frac{1}{2} (1 + \tau)^{-\frac{5}{2}}. \quad (84)$$

Clearly this bound is loose as it implies  $\theta$  is only greater than  $2^{-\frac{7}{2}}$  for  $C = I$  ( $\tau = 1$ ) whereas we know in that case that  $\theta = \frac{\pi}{2}$ . However, this bound serves its purpose of establishing a finite impact of  $C$  on the orthogonality of the original two vectors, i.e., if  $u$  and  $v$  are orthogonal, their angle as measured under a generalized inner product by  $C \succ 0$  cannot be 0.

Replacing  $\tau$  with  $\kappa^{\frac{1}{2}}$ , we can further simplify the bound to

$$|\theta| > \frac{1}{2} (1 + \tau)^{-\frac{5}{2}} \quad (85)$$

$$\geq \frac{1}{8} (1 + \kappa)^{-\frac{5}{4}}. \quad (86)$$

□

**Lemma 10.** *The angle between a pair of nearly parallel vectors ( $|\theta|$  is  $\mathcal{O}(\epsilon)$  with  $\epsilon \ll 1$ ) when instead measured under a general positive definite matrix  $C$  is upper bounded by  $\mathcal{O}(\kappa^{\frac{3}{4}} \epsilon^{\frac{1}{2}})$ .*

*Proof.* Let  $C^{\frac{1}{2}} = \begin{bmatrix} a & c \\ c & b \end{bmatrix}$  be the unique positive definite square root of  $C$  where  $a$  and  $b$  are positive and the determinant  $ab - c^2 > 0$ . Consider a vector  $u = \begin{bmatrix} 1 & 0 \end{bmatrix}$  and another vector

$v = \begin{bmatrix} \sqrt{1-\epsilon^2} & \epsilon \end{bmatrix}$  that is nearly parallel to  $u$ , i.e.,  $\epsilon \ll 1$  (implies  $|\theta|$  is  $\mathcal{O}(\epsilon)$ ). We aim to show that the angle between these two vectors under the generalized inner product  $\langle \cdot, \cdot \rangle_C$  is upper bounded by a constant multiple of  $\epsilon$  given a small enough  $\epsilon$ .

Consider

$$C^{\frac{1}{2}}u = \begin{bmatrix} a \\ c \end{bmatrix}, \quad C^{\frac{1}{2}}v = \begin{bmatrix} a\sqrt{1-\epsilon^2} + c\epsilon \\ c\sqrt{1-\epsilon^2} + b\epsilon \end{bmatrix}, \quad (87)$$

$$\|C^{\frac{1}{2}}u\| = \sqrt{a^2 + c^2} \quad \|C^{\frac{1}{2}}v\| = \sqrt{a^2(1-\epsilon^2) + c^2\epsilon^2 + b^2\epsilon^2 + c^2(1-\epsilon^2)} \quad (88)$$

$$= a\sqrt{1 + (c/a)^2}, \quad = \sqrt{a^2 + c^2 + \epsilon^2(b^2 - a^2)} \quad (89)$$

$$= a\sqrt{1 + (c/a)^2 + \epsilon^2((b/a)^2 - 1)}, \quad (90)$$

and

$$\langle C^{\frac{1}{2}}u, C^{\frac{1}{2}}v \rangle = a^2\sqrt{1-\epsilon^2} + ac\epsilon + c^2\sqrt{1-\epsilon^2}b\epsilon \quad (91)$$

$$= (a^2 + c^2)\sqrt{1-\epsilon^2} + (a+b)c\epsilon \quad (92)$$

$$= a^2(1 + (c/a)^2)\sqrt{1-\epsilon^2} + a^2(1 + (b/a))(c/a)\epsilon \quad (93)$$

$$\stackrel{|\epsilon| < 1}{\geq} a(1 + (c/a)^2)(1 - \epsilon^2) + a^2(1 + (b/a))(c/a)\epsilon. \quad (94)$$

Then

$$\frac{\langle C^{\frac{1}{2}}u, C^{\frac{1}{2}}v \rangle}{\|C^{\frac{1}{2}}u\| \|C^{\frac{1}{2}}v\|} \geq \frac{(1 + (c/a)^2)(1 - \epsilon^2) + (1 + (b/a))(c/a)\epsilon}{(1 + (c/a)^2)\sqrt{1 + \epsilon^2 \frac{(b/a)^2 - 1}{(c/a)^2 + 1}}} \quad (95)$$

$$= \frac{1}{\sqrt{1 + \epsilon^2 \frac{(b/a)^2 - 1}{(c/a)^2 + 1}}} + \frac{(1 + (b/a))(c/a)\epsilon - (1 + (c/a)^2)\epsilon^2}{(1 + (c/a)^2)\sqrt{1 + \epsilon^2 \frac{(b/a)^2 - 1}{(c/a)^2 + 1}}} \quad (96)$$

$$\stackrel{b/a \leq \kappa}{\geq} \frac{1}{\sqrt{1 + \epsilon^2 \kappa^2}} + \frac{(1 + (b/a))(c/a)\epsilon - (1 + (c/a)^2)\epsilon^2}{(1 + (c/a)^2)\sqrt{1 + \epsilon^2 \frac{(b/a)^2 - 1}{(c/a)^2 + 1}}} \quad (97)$$

$$= 1 - (1 - \frac{1}{\sqrt{1 + \epsilon^2 \kappa^2}}) + \frac{(1 + (b/a))(c/a)\epsilon - (1 + (c/a)^2)\epsilon^2}{(1 + (c/a)^2)\sqrt{1 + \epsilon^2 \frac{(b/a)^2 - 1}{(c/a)^2 + 1}}} \quad (98)$$

$$\stackrel{c \geq -\sqrt{ab}}{\geq} 1 - (1 - \frac{1}{\sqrt{1 + \epsilon^2 \kappa^2}}) - \frac{(1 + (b/a))\sqrt{(b/a)}|\epsilon| - (1 + (c/a)^2)\epsilon^2}{(1 + (c/a)^2)\sqrt{1 + \epsilon^2 \frac{(b/a)^2 - 1}{(c/a)^2 + 1}}} \quad (99)$$

$$= 1 - (1 - \frac{1}{\sqrt{1 + \epsilon^2 \kappa^2}}) - \frac{(1 + (b/a))\sqrt{(b/a)}|\epsilon|}{(1 + (c/a)^2)\sqrt{1 + \epsilon^2 \frac{(b/a)^2 - 1}{(c/a)^2 + 1}}} - \frac{\epsilon^2}{\sqrt{1 + \epsilon^2 \frac{(b/a)^2 - 1}{(c/a)^2 + 1}}} \quad (100)$$

$$\stackrel{b/a \leq \kappa}{\geq} 1 - (1 - \frac{1}{\sqrt{1 + \epsilon^2 \kappa^2}}) - \frac{(1 + \kappa)\sqrt{\kappa}|\epsilon|}{\sqrt{1 - \epsilon^2}} - \frac{\epsilon^2}{\sqrt{1 - \epsilon^2}} \quad (101)$$

$$\stackrel{1 - \frac{1}{\sqrt{1 + \epsilon^2 \kappa^2}} \leq \frac{\epsilon}{2}, \epsilon^2 < \frac{1}{\kappa^2}}{\geq} 1 - \frac{\epsilon^2}{2\kappa^2} - \frac{(1 + \kappa)\sqrt{\kappa}|\epsilon|}{\sqrt{1 - \epsilon^2}} - \frac{\epsilon^2}{\sqrt{1 - \epsilon^2}} \quad (102)$$

$$\stackrel{\epsilon < \frac{1}{2}}{\geq} 1 - 2(1 + \kappa)\sqrt{\kappa}|\epsilon| - \frac{\epsilon^2}{2}(\kappa^2 - 1) - 2\epsilon^2. \quad (103)$$

Note that  $\cos(\theta) \leq 1 - \frac{1}{8}\theta^2$  for  $|\theta| \leq \pi$ . Then

$$|\theta| \leq 2\sqrt{2}\sqrt{1 - \cos(\theta)} \leq 2\sqrt{2}\sqrt{2(1 + \kappa)\sqrt{\kappa}|\epsilon| + (\frac{\kappa^2}{2} + 2)\epsilon^2} \quad (104)$$

$$\leq 4\sqrt{(1 + \kappa)\sqrt{\kappa}|\epsilon| + (\frac{\kappa^2}{4} + 1)\epsilon^2}. \quad (105)$$

Therefore, for  $\epsilon < \min(\frac{1}{\kappa}, \frac{1}{2})$ ,  $\arccos(\frac{|\langle C^{\frac{1}{2}}u, C^{\frac{1}{2}}v \rangle|}{\|C^{\frac{1}{2}}u\| \|C^{\frac{1}{2}}v\|})$  is upper bounded by  $\mathcal{O}(\epsilon^{\frac{1}{2}}\kappa^{\frac{3}{4}})$ .

□

## E ASYMPTOTIC CONVERGENCE

We carryout a proof of convergence of Algorithm 1 in the deterministic setting and a partial proof of Algorithm 2 with further discussion.

### E.1 ASYMPTOTIC CONVERGENCE OF DETERMINISTIC UPDATE

We now give the convergence proof of Algorithm 1 using the theoretical results established above.

**Theorem 2** (Deterministic / Full-batch Global Convergence). *Given a symmetric matrix  $A$  and symmetric positive definite matrix  $B$  where the top- $k$  eigengaps of  $B^{-1}A$  are positive along with a square-summable, not summable step size sequence  $\eta_t$  (e.g.,  $1/t$ ), Algorithm 1 converges to the top- $k$  eigenvectors asymptotically ( $\lim_{T \rightarrow \infty}$ ) with probability 1.*

*Proof.* Assume none of the  $\hat{v}_i$  are initialized to an angle exactly at the minimum of their utility. This is a set of vectors with Lebesgue measure 0, therefore, the assumption holds w.p.1.

Denote the “update field”  $H(\hat{V})$  to match the work of (Shah, 2019).  $H(\hat{V})$  is simply the concatenation of all players’ Riemannian update rules, i.e., all players updating in parallel using their Riemannian updates:

$$H(\hat{V}) = [\Delta_1, \dots, \Delta_k] : \mathbb{R}^{kd} \rightarrow \mathbb{R}^{kd} \quad (106)$$

where  $\Delta_i$  is defined in equation (7) and  $\hat{V}$  represents the set of all  $\hat{v}_i$ .

A Riemannian gradient ascent step (with retractions) is then given by the following update step:

$$\hat{V}(t+1) \leftarrow \hat{V}(t) + \eta_t H(\hat{V}(t)) \quad (107)$$

$$\hat{v}_i(t+1) \leftarrow \hat{v}_i(t+1) / \|\hat{v}_i(t+1)\| \quad \forall i. \quad (108)$$

By Lemma 7,  $v_1$  is the unique fixed point of  $\hat{v}_1$ ’s update. And by Theorem 3, convergence of  $v_1$  to within  $\mathcal{O}(\epsilon)$  of its fixed point contributes to a mis-specification of children  $\hat{v}_{j>1}$ ’s fixed point by  $\mathcal{O}(\sqrt{\epsilon})$ . Critically, this mis-specification is shrinking in  $\epsilon$  so that as  $\hat{v}_1$  nears its fixed point, so may its children. This chain of reasoning applies for all  $\hat{v}_i$ .

The result is then obtained by applying Theorem 7 of (Shah, 2019) with the following information: **A0**) the unit-sphere is a compact manifold with an injectivity radius of  $\pi$  which implies the injectivity radius of the manifold of the game (the product space of  $k$  unit-spheres) is also finite, **A1**) the update field is smooth (analytic) by Lemma 6, **A2**) we assume a square-summable, not summable step size, **A3**) we assume the full-batch (noiseless) setting so the update “noise” clearly constitutes a bounded martingale difference sequence, and **A4**) the iterates remain bounded because they are constrained to the unit-sphere.

Formally, for any  $T > 0$ ,

$$\lim_{s \rightarrow \infty} \sup_{t \in [s, s+T]} d(\hat{V}(t), \hat{V}^s(t)) \rightarrow a.s., \quad (109)$$

where  $d(\cdot, \cdot)$  is the Riemannian distance on the (product space of) sphere and  $\hat{V}^s(t)$  denotes the continuous time trajectory of  $H(\hat{V})$  starting from  $\hat{V}(s)$ . □

## E.2 ASYMPTOTIC CONVERGENCE OF STOCHASTIC UPDATE

There are two primary issues with extending the asymptotic convergence guarantee in Theorem 2 to Algorithm 2. The first is that the joint parameter space includes  $\hat{v}_i \in \mathcal{S}^{d-1}$  and  $[B\hat{v}]_i \in \mathbb{R}^d$ . The unit-sphere,  $\mathcal{S}^{d-1}$ , is a compact Riemannian manifold. While  $\mathbb{R}^d$  is a Riemannian manifold, it is not compact. This violates assumptions A0 and A4 above. The second issue is that the vector field  $H(\hat{V}; [B\hat{V}])$  is not smooth due to the clipped denominator terms of  $y_j$  (see line 8 of Algorithm 2). We can easily fix this issue with a change of variables, defining  $\langle \hat{v}_i, B\hat{v}_i \rangle$  in log-space. This results in Algorithm 3.

Specifically, define  $[\langle v, Bv \rangle]_j = e^{\log(\rho) + (\log(\nu) - \log(\rho)) \text{sigmoid}(z_j)}$  where  $z_j$  is a newly introduced auxiliary variable. The relevant gradient with respect  $z_j$  is  $\nabla_{z_j} = -(\langle v_j, Bv_j \rangle - [\langle v, Bv \rangle]_j) \cdot [\langle v, Bv \rangle]_j \cdot (\log \nu - \log \rho) \cdot \text{sigmoid}(z_j) \cdot (1 - \text{sigmoid}(z_j))$ .

Regarding the still unresolved first issue, we could constraint  $[B\hat{v}]_i$  to a ball with radius  $\lambda_{\max}(B)$  centered at the origin, which is a convex set. Note that while a ball in  $\mathbb{R}^d$  is compact, it is not a Riemannian manifold anymore. A few works have developed theory for the setting that mixes convex and Riemannian optimization (Liu et al., 2017; Goyal & Shetty, 2019). Intuitively, we do not expect issues arising in our setting from the mixture of feasible sets, however, progress towards theoretic results takes time. We conjecture that Algorithm 3 is provably asymptotically convergent, although Algorithm 2 defined with clipping is a bit more practical.

---

### Algorithm 3 Smooth Stochastic $\gamma$ -EigenGame

---

- 1: Given: paired data streams  $X_t \in \mathbb{R}^{b \times d_x}$  and  $Y_t \in \mathbb{R}^{b \times d_y}$ , number of parallel machines  $M$  per player (minibatch size per machine  $b' = \frac{b}{M}$ ), step size sequence  $\eta_t$ , scalar  $\rho$  lower bounding  $\lambda_{\min}(B)$ , scalar,  $\nu$  upper bounding  $\lambda_{\max}(B)$ , and number of iterations  $T$ .
  - 2:  $\hat{v}_i \sim \mathcal{S}^{d-1}$ , i.e.,  $\hat{v}_i \sim \mathcal{N}(\mathbf{0}_d, \mathbf{I}_d)$ ;  $\hat{v}_i \leftarrow \hat{v}_i / \|\hat{v}_i\|$  for all  $i$
  - 3:  $[B\hat{v}]_i \in \mathbb{R}^d \leftarrow \hat{v}_i$  for all  $i$
  - 4:  $z_i^0 \in \mathbb{R} \leftarrow 0$  for all  $i$
  - 5: **for**  $t = 1 : T$  **do**
  - 6:   **parfor**  $i = 1 : k$  **do**
  - 7:     **parfor**  $m = 1 : M$  **do**
  - 8:       Construct  $A_{tm}$  and  $B_{tm}$
  - 9:        $[\langle v, Bv \rangle]_j = e^{\log(\rho) + (\log(\nu) - \log(\rho)) \sigma(z_j)}$
  - 10:        $\hat{y}_j = \frac{\hat{v}_j}{\sqrt{[\langle v, Bv \rangle]_j}}$
  - 11:        $[B\hat{y}]_j = \frac{[B\hat{v}]_j}{\sqrt{[\langle v, Bv \rangle]_j}}$
  - 12:       rewards  $\leftarrow (\hat{v}_i^\top B_{tm} \hat{v}_i) A_{tm} \hat{v}_i - (\hat{v}_i^\top A_{tm} \hat{v}_i) B_{tm} \hat{v}_i$
  - 13:       penalties  $\leftarrow \sum_{j < i} (\hat{v}_i^\top A_{tm} \hat{y}_j) [\langle \hat{v}_i, B_{tm} \hat{v}_i \rangle [B\hat{y}]_j - \langle \hat{v}_i, [B\hat{y}]_j \rangle B_{tm} \hat{v}_i]$
  - 14:        $\tilde{\nabla}_{im} \leftarrow \text{rewards} - \text{penalties}$
  - 15:        $\nabla_{im}^{Bv} = (B_{tm} \hat{v}_i - [B\hat{v}]_i)$
  - 16:        $\nabla_{im}^z = (\langle v_i, [Bv]_i \rangle - [\langle v, Bv \rangle]_i) \cdot [\langle v, Bv \rangle]_i \cdot (\log \nu - \log \rho) \cdot \sigma(z_i) \cdot (1 - \sigma(z_i))$
  - 17:     **end parfor**
  - 18:      $\tilde{\nabla}_i \leftarrow \frac{1}{M} \sum_m [\tilde{\nabla}_{im}]$
  - 19:      $\hat{v}'_i \leftarrow \hat{v}_i + \eta_t \tilde{\nabla}_i$
  - 20:      $\hat{v}_i \leftarrow \frac{\hat{v}'_i}{\|\hat{v}'_i\|}$
  - 21:      $\nabla_i^{Bv} \leftarrow \frac{1}{M} \sum_m [\nabla_{im}^{Bv}]$
  - 22:      $[B\hat{v}]_i \leftarrow [B\hat{v}]_i + \gamma_t \nabla_i^{Bv}$
  - 23:      $\nabla_i^z \leftarrow \frac{1}{M} \sum_m [\nabla_{im}^z]$
  - 24:      $z_i \leftarrow z_i + \gamma_t \nabla_i^z$
  - 25:   **end parfor**
  - 26: **end for**
  - 27: **return** all  $\hat{v}_i$
-

## F ALTERNATIVE PARALLELIZED IMPLEMENTATION

As mentioned in Section 5, we plan to open source our implementation, specifically the implementation used to conduct the neural CCA experiments described in Section 5. The specific parallelization we used was different than that implied by Algorithm 2. Instead, we parallelized the estimation of the matrix-vector products  $Av_i$  and  $Bv_i$  for all  $i$  and then aggregated this information across machines. Figure 6 provides a diagram illustrating how data and algorithmic operations are distributed.

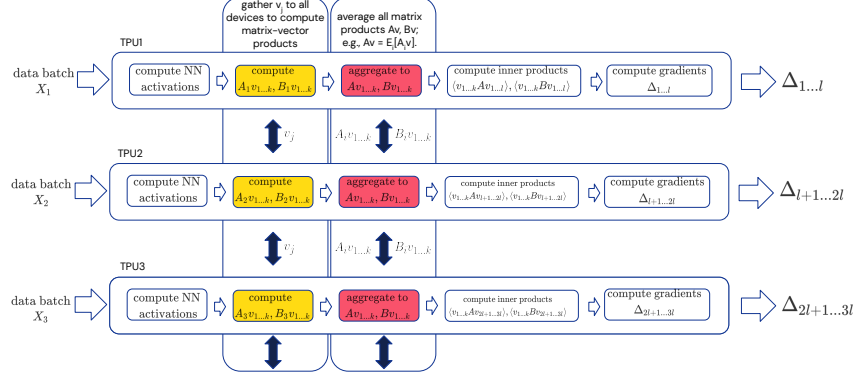


Figure 6: Parallelization of  $\gamma$ -EigenGame implementation for neural CCA experiments in Section 5

## G ADDITIONAL EXPERIMENTS

### G.1 REGULARIZATION EFFECTS OF STOCHASTIC APPROXIMATION

In pursuit of understanding the regularization benefits of our stochastic approximation approach with a fixed step size, we explore various ways of regularizing the matrices  $A$  and  $B$  prior to calling `scipy.linalg.eigh(A, B)` to see if we can achieve a similar solution. Figure 7 explores various parameter settings, but none adequately recover the original signals.

We parameterize our regularizations as follows. Let  $C = \mathbb{E}_t[x_t x_t^\top]$ . Then

$$A = \mathbb{E}_t[\langle x_t, x_t \rangle x_t x_t^\top] - \text{tr}(C + \epsilon)(C + \epsilon) - 2(C + \epsilon)^2 \quad B = C + \epsilon'. \quad (110)$$

### G.2 PARALLEL VS SEQUENTIAL LEARNING

The parallel approach provides a few advantages over the sequential approach. 1) Intuitively, the parallel approach is similar to the sequential approach but with “warm starting”. Child eigenvectors are allowed to learn while their parents are learning, which puts them in a good position to reach their correct directions once their parents have learned. In contrast, a sequential approach would randomly initialize the child once the parents have converged, leaving the child to traverse a longer geodesic to reach its true destination. 2) How do you know when the parents are done learning? You would need to measure convergence of the eigenvectors and this is difficult in the stochastic setting. You could use a running mean of the Riemannian gradient norm or of the difference in successive Rayleigh quotients, but this is approximate. This is an interesting challenge for future research.

In Figure 8, we assume we know the true eigenvalues and use this information to decide when to deflate. In this experiment, the first two eigenvalues are approximated well enough, but this level of accuracy is not high enough to allow learning the third eigenvector accurately. This supports our argument when knowing when to stop learning and deflate is a difficult problem because the accuracy of parents affects the learning of children in ways that depend on the spectrum (which is unknown in any practical setting).

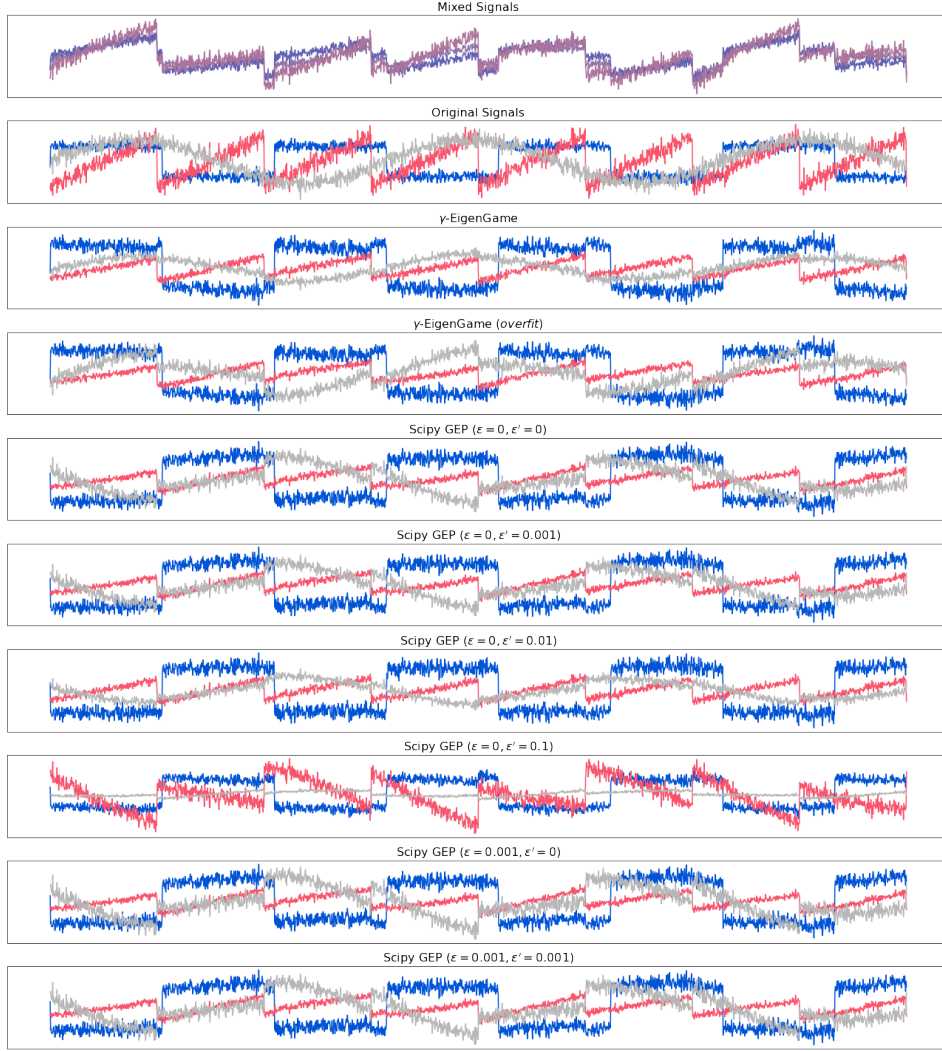


Figure 7: Figure 2 repeated with additional regularized versions of `scipy.linalg.eigh`.

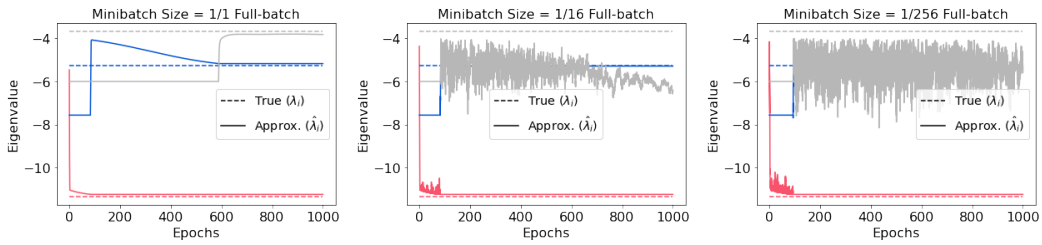


Figure 8: In contrast to the parallel learning approach we propose in Algorithm 2, here, we explore a sequential, deflation-inspired approach where each eigenvector completes learning only once its Rayleigh quotient (eigenvalue) is within 0.1 of the true value (we chose this value of 0.1 because full batch EigenGame (Figure 2 (left)) reaches at least 0.1 accuracy for all 3 eigenvalues by the end of training). Once this eigenvector has completed learning, the next eigenvector then begins learning. This sequential approach fails to learn the third eigenvalue to within the 0.1 threshold in both minibatch settings.



## H HYPERPARAMETERS AND EXPERIMENT DETAILS

### H.1 ICA

Details of the unmixing experiment we run can be found on the scikit-learn website: [./auto\\_examples/decomposition/plot\\_ica\\_blind\\_source\\_separation.html](https://auto-examples.decomposition/plot_ica_blind_source_separation.html) (Pedregosa et al., 2011). We solve for top-3 SGEP formulation of ICA using  $n = 2000$  samples taken from the time series data. Minimal hyperparameter tuning was performed. Learning rates were searched over orders of magnitude (e.g., 0.01, 0.1, 1.0, ...).

#### H.1.1 COMPARISON

The parameters used for Algorithm 2 ( $\gamma$ -EigenGame) in Figure 2 are listed in Table 1. Those for overfitting  $\gamma$ -EigenGame to the data are listed in Table 2.

Algorithm Parameters	
batch size $b$	$\frac{n}{4}$
$M$	1
# of iterations (T)	$10^3 \cdot \frac{n}{b}$
$\eta_t$	$10^{-2} \cdot \frac{b}{n}$
$\beta_t$	$1 \cdot \frac{b}{n}$

Table 1: Algorithm 2 hyperparameters for  $\gamma$ -EigenGame in Figure 2.

Algorithm Parameters	
batch size $b$	$n$
$M$	1
# of iterations (T)	$10^5$
$\eta_t$	$10^{-3}$
$\beta_t$	1

Table 2: Algorithm 2 hyperparameters for  $\gamma$ -EigenGame (*overfit*) in Figure 2.

#### H.1.2 UNBIASED

Each of the plots in Figure 3 uses a different minibatch size  $b$ ; the hyperparameters used for Algorithm 2 are listed in Table 3 as a function of  $b$ .

Algorithm Parameters	
$M$	1
# of iterations (T)	$10^3 \cdot \frac{n}{b}$
$\eta_t$	$10^{-2} \cdot \frac{b}{n}$
$\beta_t$	$1 \cdot \frac{b}{n}$

Table 3: Algorithm 2 hyperparameters for Figure 3.

## H.2 CCA

Details of both CCA experiments can be found below.

### H.2.1 COMPARISON

Hyperparameters for the algorithm proposed in (Meng et al., 2021) are the same as in their paper. Their code is available on github at [.../zihangm/riemannian-streaming-cca](https://github.com/zihangm/riemannian-streaming-cca). We ran experiments 10 times to produce the means and standard deviation shading in Figure 2. We run PCA first on the data to remove the subspaces in  $X$  and  $Y$  with zero variance. We then solve the top-4 SGEP formulation of CCA. Hyperparameters were tuned manually, searching over orders of magnitude (e.g., 0.01, 0.1, 1.0 and in some cases 0.05, 0.5, 5.0). We set  $\rho = 10^{-10}$ .

Shared parameters	
$b$	100
$M$	1
$T$	$\frac{n}{b}$
MNIST	
$\eta_t$	0.1
$\beta_t$	1.0
Mediamill	
$\eta_t$	50
$\beta_t$	5
CIFAR-10	
$\eta_t$	0.1
$\beta_t$	0.1

Table 4: Algorithm 2 hyperparameters for Figure 2

### H.2.2 NEURAL NETWORK ANALYSIS

We trained two CNNs for the  $d > 10^3$  and  $d > 10^5$  CCA (top-1024 SGEP) experiments in Figure 4. Details of both architectures are listed in Table 5.

We used the hyperparameters listed in Table 6 for running  $\gamma$ -EigenGame.

Adam( $b1 = 0.9, b2 = 0.999, \epsilon = 10^{-8}$ ) was used for learning  $\hat{v}_i$ . We pair Adam with a learning rate schedule that consists of separate warmup and harmonic decay phases. We have a warmup period for the eigenvectors while the auxiliary variables and mean estimates are learned. During this period, the learning rate increases linearly until it reaches the base learning rate after the period ends (iteration  $t_c$ ). This is followed by a decaying learning rate ( $\propto \frac{1}{t+\Delta t}$ ) which reaches the final learning rate at iteration  $T$ .

For the purpose of generating the plots, we estimated Rayleigh quotients with a larger batch size than that used to estimate the eigenvectors themselves; specifically, we used 2048 for evaluation vs 256 for training.

Both experiments were 100% input bound, meaning the bottleneck in speed was computing neural network activations and passing them in minibatches to our algorithm. Therefore, our current runtimes are not indicative of the complexity of our proposed update rule. Alternatively, one could precompute all activations and save them to disk, however, this is memory intensive and we chose not to do this. For completeness, the  $d > 10^3$  dimensional experiment ran at 4.7 ms per step on average, and the  $10^5$  experiment at 30.2 ms per step.

$d = 2048 > 10^3$  - Figure 4 (left)

Activations Harvested	Last convolutional layer and dense layer
Conv Layer Output Channels	[64, 32]
Conv Strides	[2, 1]
Dense Layer Sizes	[512]
Total Activations	$d_x = d_y = 1024$

$d = 116736 > 10^5$  - Figure 4 (right)

Activations Harvested	All convolutional layers and dense layer
Conv Layer Output Channels	[128, 256, 512]
Conv Strides	[1, 1, 1]
Dense Layer Sizes	[1024]
Total Activations	$d_x = d_y = 58368$

Table 5: CNN architecture parameters for CIFAR-10 Neural CCA experiment.

Algorithm Parameters

$b$	2048 (256 per device)
$M$	8 ( $2 \times 2$ TPU = 4 chips, 2 devices/chip)
$T$	$10^7$
$\eta_t$	$t_c = 10^5, \eta_0 = 10^{-4}, \eta_T = 10^{-6}$
$\beta_t$	$10^{-3}$
$\epsilon$	$10^{-4}$
$\rho$	$10^{-6}$

Table 6: Algorithm 2 (with parallelism modifications from Section F) hyperparameters for Figure 4.

We do not have a breakdown of the runtime that separates CIFAR-10 data loading from neural network evaluation (computing activations) from EigenGame update from other processes. However, we can share that overall, the  $d > 10^3$  dimensional experiment ran at 4.7 ms per step on average, and the  $10^5$  experiment at 30.2 ms per step.

## I RUNTIME COMPLEXITY

Algorithm 2 states under line 1 "Given" that it expects "number of parallel machines  $M$  per player". There are  $k$  players, established in Section 2. This means Algorithm 2 expects  $p = kM$  processors. This can also be inferred by noticing the two parallel for-loops (parfor) on lines 5 and 6 of Algorithm 2. The paragraph on "Computational Complexity and Parallelization" then goes on to consider the case where "each player (model) parallelizes over  $M = b$  machines" where  $b$  is the batch size. Again, referring back to Algorithm 2 it is written "minibatch size per machine  $b' = \frac{b}{M}$ ", therefore,  $b' = 1$ . We can now consider the computational cost of each line of Algorithm 2, which is dominated by the matrix-vector products on lines 8-11. Recall that  $A_{tm}$  and  $B_{tm}$  are both formed as outer products, e.g.,  $B_{tm} = X_{tm}^\top X_{tm}$  in ICA with  $X_{tm} \in \mathbb{R}^{b' \times d}$ . Given that we are considering the case where  $b' = 1$ ,  $B_{tm}$  can be rewritten as  $x_{tm}x_{tm}^\top$  where  $x_{tm} \in \mathbb{R}^d$  to make it clear that the  $1 \times d$  matrices are vectors. Therefore, all matrix-vector products (and inner products) on lines 8-11

cost  $\mathcal{O}(d)$ . There is 1 on line 8, 1 on line 9, 4 on line 10 (rewards), and  $4k$  on line 11 (penalties) making for a total cost of  $\mathcal{O}(dk)$ .