NBDT

TUTORIAL

Methods

arXiv:2512.12467v2 [q-bio.NC] 12 Nov 2026

Reduced-rank regression for neural communication: a tutorial for neuroscientists

Bichan Wu[1] | **Jonathan W. Pillow**[1]

[1]Princeton Neuroscience Institute, Princeton University, Princeton, NJ, 08544, USA

Correspondence
Jonathan W. Pillow, Princeton Neuroscience Institute, Princeton University, Princeton, NJ, 08544, USA
Email: pillow@princeton.edu

Reduced-rank regression (RRR) is a statistical method for finding a low-dimensional linear mapping between a set of high-dimensional inputs and outputs [1]. In recent years, RRR has found numerous applications in neuroscience, in particular for identifying "communication subspaces" governing the interactions between brain regions [2, 3]. This tutorial article seeks to provide an introduction to RRR and its mathematical foundations, with a particular emphasis on neural communication. We discuss RRR's relationship to alternate dimensionality reduction techniques such as singular value decomposition (SVD), principal components analysis (PCA), principal components regression (PCR), and canonical correlation analysis (CCA). We also derive important extensions to RRR, including "ridge" regularization and non-spherical noise. Finally, we introduce new metrics for quantifying communication strength as well as the alignment between communication axes and the principal modes of neural activity. By the end of this article, readers should have a clear understanding of RRR and the practical considerations involved in applying it to their own data.

KEYWORDS
reduced-rank regression, communication subspace, dimensionality reduction, neural populations, multi-region

recordings, tutorial

1 | INTRODUCTION

An important goal of neuroscience research is to understand how coordinated neural activity across brain regions gives rise to the complex forms of information processing. Although this problem remains daunting in general, recent work has shown that pairs of brain regions often communicate via a low-dimensional channel, commonly known as a *communication subspace* [2, 4, 5]. Intuitively, this means that only some fraction of the signals in a given brain region are communicated to other brain regions (see Fig. 1).

The most popular method for estimating communication subspaces is a statistical technique known as *reduced-rank regression* (RRR) [1, 6], which has also found application in a variety of other neural data analysis methods [7, 8, 9, 10]. Despite its widespread use in neuroscience, the mathematical underpinnings of RRR remain shrouded in a statistical literature that is not easily accessible to many neuroscientists. In this tutorial paper, we seek to address this gap, and to clarify the mathematical details underlying RRR, its extensions, and its interpretation.

Our first goal is to unpack the RRR estimator and its derivation. For example, the RRR estimator has a closed-form solution arising from an eigenvector equation—but where exactly does this eigenvector problem come from? And why is RRR not equivalent to the optimal low-rank approximation to the maximum likelihood or least-squares estimator? More generally, how does RRR differ from other regression-based dimensionality-reduction methods? By the end of this tutorial, the answers to these and related questions should become clear.

After reviewing the basic method, we will discuss several straightforward but underappreciated extensions to RRR. First, RRR can be regularized in closed form with an L_2 penalty, analogous to ridge regression, which is especially important in settings with a small number of samples relative to the number of neurons. Second, RRR can be adapted to incorporate non-spherical or non-independent noise distributions, which can be important when neurons have correlated variability or differing amounts of noise.

Next, we will turn to the problem of interpreting the communication channels identified by RRR. Neuroscientists may wish to quantify how much variability can be explained using RRR, or to understand how the dimensions along which two neural populations communicate relate to the principal components of neural activity. To address these problems, we introduce new metrics for quantifying both the strength of communication channels and their alignment with the dominant modes of neural activity in both input and output regions.

Finally, we will discuss practical as well as conceptual considerations in the use of RRR for analyzing multi-region recordings. We begin with several operational issues, including time binning, centering and conditioning of the data, and rank determination. We then turn to broader interpretational considerations, such as the modeling assumptions underlying RRR, recommended follow-up analyses, and the fundamentally correlational nature of the method.

Our hope is that, by the end of this article, RRR will feel less like a black box and more like an intuitive and accessible tool, which readers can readily apply to their own data. This article will assume a basic knowledge of linear algebra, though we will begin with a brief review of least squares regression and singular value decomposition (SVD) in Section 2. If readers are already comfortable with this background, they should feel free to skip ahead to sections that seem most relevant or interesting.

2 | MATHEMATICAL BASICS

2.1 | Linear-Gaussian communication model

We begin by introducing the linear-Gaussian regression model that forms the theoretical basis for reduced-rank regression. We consider neural data recorded from a pair of brain areas, one designated the *input region* and the other designated the *output region*. Although real neural communication in the brain is typically bidirectional—meaning that no region is ever purely input or output in terms of information flow—the RRR estimator arises from a regression model that seeks to predict neural activity in one area (i.e., the output region) using the activity of another (i.e., the input region). To characterize information flow in the opposite direction, one can simply apply RRR with input and output labels swapped.

To formalize the model, let $\mathbf{x}_t \in \mathbb{R}^m$ denote a vector of neural activity from m neurons in the input brain region at time t, and $\mathbf{y}_t \in \mathbb{R}^n$ denote a vector of activity from n neurons in the output region at time t. These vectors might contain spike counts, average calcium fluorescence, BOLD signal, or other measures of neural activity in a given time window. The linear-Gaussian model seeks to describe the output activity $\mathbf{y}_t$ as a linear function of the input activity $\mathbf{x}_t$, corrupted by Gaussian noise:

$$\mathbf{y}_t = W^\top \mathbf{x}_t + \epsilon_t, \tag{1}$$

where W is a $m \times n$ matrix of weights describing the influence between each input neuron and each output neuron, and ϵ_t represents independent, zero-mean Gaussian noise, which we assume (for now) to be identically distributed with variance σ^2:

$$\epsilon_t \sim \mathcal{N}(0, \sigma^2 I_n), \tag{2}$$

where I_n is the $n \times n$ identity matrix.

2.2 | Least Squares Regression

Given a set of measured activity patterns from our input and output populations, $\{(\mathbf{x}_t, \mathbf{y}_t)\}_{t=1}^T$, for time samples $t = 1, \ldots, T$, our goal is to estimate the linear weight matrix W, which describes the linear communication channel from $\mathbf{x}_t$ to $\mathbf{y}_t$. We can use the linear-Gaussian regression model (eq. 1) to define the log-likelihood function, which is the log of the conditional probability of output vectors $\{\mathbf{y}_t\}$ given the input vectors $\{\mathbf{x}_t\}$, considered as a function of W:

$$\log P(\{\mathbf{y}_t\} \mid \{\mathbf{x}_t\}, W) = \sum_{t=1}^{T} \sum_{i=1}^{n} -\frac{1}{2\sigma^2} (y_{it} - \mathbf{w}_i{}^\top \mathbf{x}_t)^2 + c, \tag{3}$$

where y_{it} is the response of the i'th neuron at time t, $\mathbf{w}_i$ is the i'th column of W, and c is a constant that does not depend on W. The maximum likelihood (ML) estimate can be obtained by maximizing this log-likelihood for W.

We can derive the ML estimate more easily if we first rewrite the log-likelihood function as a matrix equation, with inputs and outputs stacked into matrices as row vectors. Let $X \in \mathbb{R}^{T \times m}$ and $Y \in \mathbb{R}^{T \times n}$ denote matrices whose

rows are formed from inputs and outputs for a dataset containing T total samples:

$$X = \begin{bmatrix} — & \mathbf{x}_1{}^\top & — \\ & \vdots & \\ — & \mathbf{x}_T{}^\top & — \end{bmatrix}, \qquad Y = \begin{bmatrix} — & \mathbf{y}_1{}^\top & — \\ & \vdots & \\ — & \mathbf{y}_T{}^\top & — \end{bmatrix}. \tag{4}$$

The log-likelihood can then be rewritten:

$$\log P(Y|X, W) = -\frac{1}{2\sigma^2}||Y - XW||_F^2 + c, \tag{5}$$

where $||\cdot||_F^2$ denotes the squared Frobenius norm of a matrix, which is equal to the sum of squares of all elements in the matrix. The Frobenius norm term in (eq. 5) thus corresponds to the sum of squared residual errors between the outputs Y and the linear prediction XW. It can also be expressed as $\mathrm{Tr}\big[(Y - XW)^\top (Y - XW)\big]$, where $\mathrm{Tr}[\cdot]$ denotes the matrix trace.

We can obtain the maximum likelihood estimate by differentiating (eq. 5) with respect to W, setting it to zero, and solving for W. This leads to the well-known solution:

$$\hat{W}_{\mathrm{LS}} = (X^\top X)^{-1}(X^\top Y), \tag{6}$$

which is also known as the *ordinary least-squares* (OLS) estimate, since it minimizes $||Y - XW||_F^2$, the sum of squared errors between the output responses Y and the linear prediction XW. Note that the solution does not depend on the noise variance σ^2.

2.3 | Singular Value Decomposition (SVD)

Singular value decomposition (SVD) is an extremely useful tool from linear algebra for decomposing any matrix into a product of three structured matrices. Specifically, the singular value decomposition for a matrix $A \in \mathbb{R}^{m\times n}$ is given by:

$$A = USV^\top, \tag{7}$$

where $U \in \mathbb{R}^{m\times m}$ and $V \in \mathbb{R}^{n\times n}$ are orthogonal matrices, meaning that $U^\top U = UU^\top = I$ and $V^\top V = VV^\top = I$, and $S \in \mathbb{R}^{m\times n}$ is a rectangular diagonal matrix with nonnegative entries along the diagonal. The columns of U and V are called the left and right singular vectors, respectively, and the diagonal elements of S are called the singular values (which, by convention, are sorted from largest to smallest). If A is a symmetric, positive semi-definite matrix, then the SVD corresponds to eigendecomposition, meaning that the left and right singular vectors are identical ($U = V$) and correspond to eigenvectors, and the singular values (diagonal elements of S) are the eigenvalues.

Although a complete review of the SVD is beyond the scope of this article, a key fact relevant to this article is that the SVD can be used to obtain the optimal low-rank approximation to any matrix, in the sense of minimizing squared error between the matrix and its low-rank approximation. Consider an arbitrary $m \times n$ matrix A, and suppose that we wish to find the best rank-r approximation, denoted A_r, in the sense of minimizing $||A - A_r||_F^2$, the sum of squared error between the elements of A and A_r. Note that we are assuming $r < \min(n, m)$, since the matrix is only lower

rank if r is smaller than the number of rows or columns, whichever is smaller. The optimal solution for A_r is given by:

$$A_r = \sum_{i=1}^{r} s_i \mathbf{u}_i \mathbf{v}_i{}^\top = U_r S_r V_r{}^\top, \tag{8}$$

where $\mathbf{u}_i$ and $\mathbf{v}_i$ are the i'th left and right singular vectors of A, and s_i are the corresponding singular values. The rightmost expression results from forming U_r and V_r from the first r columns of U and V, respectively, and S_r from the top-left $r \times r$ block of S, where $A = USV^\top$ is the SVD of A.

2.4 | Principal Components Analysis (PCA)

Principal component analysis (PCA) is a dimensionality reduction method for finding the best low-dimensional approximation to a high dimensional dataset in terms of squared error. Specifically, it identifies an ordered set of orthogonal axes, ordered by how much variance they explain in the dataset.

The principal components can be computed by performing SVD on the sample covariance matrix of the data:

$$\Sigma_X = \tfrac{1}{T} X^\top X, \tag{9}$$

where T is the number of samples, and X is a data matrix formed as in (eq. 4), which we assume to be zero-centered for each column. Since Σ_X is symmetric and positive semi-definite, the SVD (which is also an eigen-decomposition) can be written:

$$\Sigma_X = USU^\top, \tag{10}$$

where the singular vectors or eigenvectors (given by the columns of U) represent the principal components—that is, unit vectors indicating the directions of maximal variance in X, sorted in decreasing order of variance. Incidentally, the singular values or eigenvalues (given by the diagonal elements of S) correspond to the amount of variance captured by each component.

In neuroscience settings, the principal components correspond to population activity patterns, and are commonly referred to as "modes" of neural activity. In many recordings of large-scale neural population activity, a large fraction of the total variability is captured by a relatively small number of principal components [11, 12, 13, 14, 15]. For a tutorial on PCA aimed at neuroscientists, see Shlens [16].

3 | REDUCED-RANK REGRESSION (RRR)

We are now ready to move on to the main topic of this article: the RRR estimator. The problem of interest is to find the best linear weight matrix W for the linear-Gaussian model (eq. 1), in the sense of maximum likelihood or least squares, subject to a rank constraint on W. The communication channel between input and output is low-dimensional or *low rank* when the $m \times n$ weight matrix W can be factored into a pair of rank-r matrices:

$$W = UV^\top, \tag{11}$$

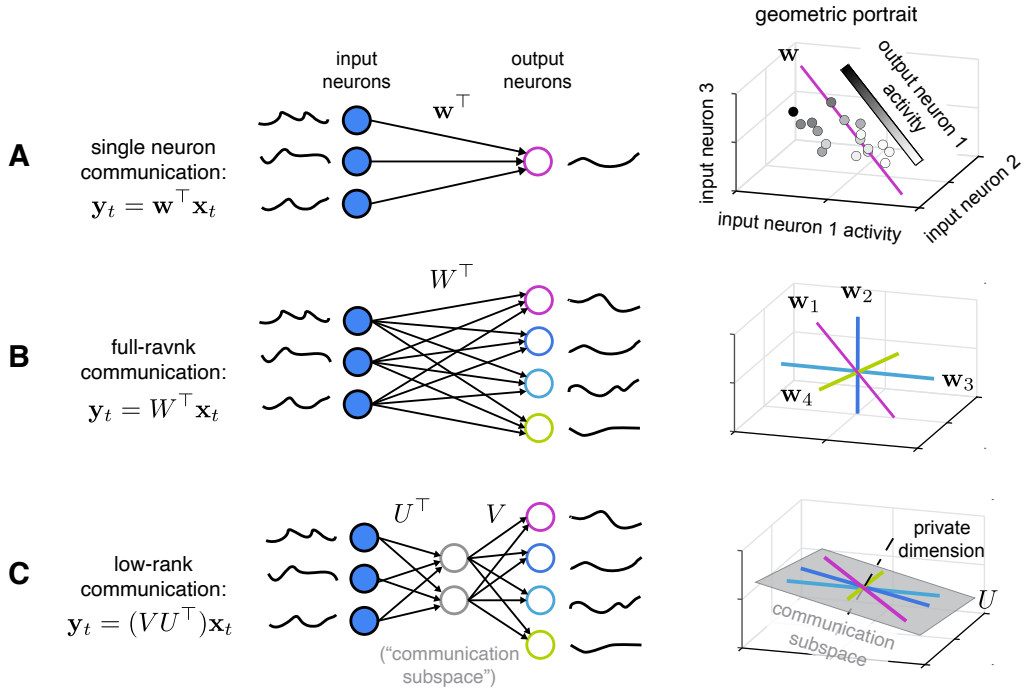


FIGURE 1 Illustration of full-rank and low-rank linear communication (adapted from Semedo et al. [2]). **(A)** To build intuition, we start by considering a linear map from 3 input neurons to 1 output neuron. In this case, the linear weights **w** define a 1D subspace (magenta line) within the 3D space of input neuron activity (right). The activity of the output neuron depends only on the linear projection of input activity vector onto this subspace, and is unaffected by perturbations orthogonal to it. **(B)** For a recording with 3 input neurons and 4 output neurons, the columns of the weight matrix W define four different linear communication axes. If these vectors span the full 3D input space, the communication is full rank. **(C)** If the weight matrix is rank 2, then we can factor it as $W^\top = VU^\top$, where U maps the 3D input vector to a 2D latent variable, and $V^\top$ maps this 2D variable to a 4D output vector. Geometrically, this means that the dimensions of input activity space that can affect the output neurons are confined to a 2D subspace, while the third dimension represents a "private dimension" that does not affect the output population.

where $U \in \mathbb{R}^{m \times r}$ and $V \in \mathbb{R}^{n \times r}$ both have $r < \min(m, n)$ linearly independent columns. In this case, the regression model (eq. 1) can be rewritten:

$$\mathbf{y}_t = VU^\top \mathbf{x}_t + \epsilon_t. \tag{12}$$

Intuitively, we can think of the columns of U as representing patterns of neural activity in the input space that can affect the output population, and we refer to them as *input axes*. Patterns of activity orthogonal to the columns of U represent "private dimensions" in the input population, which do not affect the outputs (see Fig. 1) [2]. Similarly, the columns of V represent patterns of neural activity in the output population that can be modulated by the input population, and we refer to these as *output axes*. Activity patterns orthogonal to the columns of V cannot be modulated by the input population.

The log-likelihood function for the low-rank communication model can thus be written:

$$\log P(Y \mid X, U, V) = -\frac{1}{2\sigma^2}||Y - XUV^\top||_F^2 + c. \tag{13}$$

Fitting the model to data thus requires finding the matrices U and V that maximize this log-likelihood, or equivalently, minimize $||Y - XUV^\top||_F^2$, the squared error between Y and the linear prediction $XUV^\top$.

3.1 | The RRR estimator

Based on what we have seen so far, it would be natural to guess that the RRR estimator can be obtained by finding the best rank-r approximation to the least-squares estimate $\hat{W}_{\text{LS}}$ (eq. 6) using SVD (eq. 8). Unfortunately, this guess turns out to be wrong! To understand why, we will provide a detailed comparison between RRR and the SVD of the least-squares estimate in Sec. 4.1. But first, we will introduce the RRR estimator itself and provide a simple derivation.

Given a dataset consisting of input responses X and output responses Y, the RRR estimator of rank r can be obtained as follows:

1. Compute the least-squares estimate of the weights:

$$\hat{W}_{\text{LS}} = (X^\top X)^{-1} X^\top Y; \tag{14}$$

2. Perform PCA on the linear prediction $X\hat{W}_{\text{LS}}$. Specifically, compute $(\hat{W}_{\text{LS}}{}^\top X^\top X \hat{W}_{\text{LS}})$ and take its top r eigenvectors to form the matrix

$$V_r = \begin{bmatrix} \mathbf{v}_1 \cdots \mathbf{v}_r \end{bmatrix}. \tag{15}$$

3. Finally, project the least squares solution onto the subspace spanned by the columns of V_r:

$$\hat{W}_{\text{RRR}} = \hat{W}_{\text{LS}} V_r V_r{}^\top. \tag{16}$$

If desired, note that we can express this estimate as the product of an $m \times r$ matrix U and an $n \times r$ matrix V as in (eq. 11) using the factorization:

$$\hat{U}_{\text{RRR}} = (\hat{W}_{\text{LS}} V_r), \qquad \hat{V}_{\text{RRR}} = V_r, \tag{17}$$

which gives us $\hat{W}_{\text{RRR}} = \hat{U}_{\text{RRR}} \hat{V}_{\text{RRR}}{}^\top$ as in (eq. 16). For this factorization, the matrix $\hat{V}_{\text{RRR}}$ is semi-orthogonal—all columns are orthogonal unit vectors—while columns of $X\hat{U}_{\text{RRR}}$ are orthogonal. Next, we will provide a simple derivation of this result.

3.2 | Derivation of RRR (rank 1 case)

For simplicity, we will focus on the rank-1 case, where the RRR estimate can be written as an outer product $\hat{W}_{\text{RRR}} = \mathbf{u}\mathbf{v}^\top$, for some column vectors $\mathbf{u}$ and $\mathbf{v}$. (See Appendix B for a derivation for the more general case with arbitrary rank).

The RRR estimate corresponds to the minimum of the least-squares loss function across all rank-1 matrices. This

loss is given by the negative of the log-likelihood (eq. 13), neglecting the factor of $(1/2\sigma^2)$ and the additive constant c, both of which can be safely ignored. For the rank-1 case, the least-squares loss can be written:

$$||Y - X\mathbf{u}\mathbf{v}^\top||_F^2 = \mathrm{Tr}[Y^\top Y] - 2\mathbf{u}^\top X^\top Y\mathbf{v} + \mathrm{Tr}[\mathbf{u}^\top X^\top X\mathbf{u}\mathbf{v}^\top\mathbf{v}], \tag{18}$$

where we have used the definition of the Frobenius norm and the cyclic property of the trace to obtain the right-hand side. Without loss of generality, we can assume that $\mathbf{v}$ is a unit vector:

$$\mathbf{v}^\top\mathbf{v} = 1, \tag{19}$$

which allows us to eliminate the $\mathbf{v}^\top\mathbf{v}$ term in (eq. 18). (In this case, $\mathbf{u}$ can take on arbitrary norm; see Appendix A for a discussion on identifiability).

To solve the resulting constrained optimization problem, we add a term to the loss function consisting of a Lagrange multiplier λ, multiplied by an expression that is zero when the constraint $\mathbf{v}^\top\mathbf{v} = 1$ is satisfied. The resulting loss function for $\mathbf{u}$ and $\mathbf{v}$ is given by:

$$L(\mathbf{u}, \mathbf{v}) = ||Y - X\mathbf{u}\mathbf{v}^\top||_F^2 + \lambda(\mathbf{v}^\top\mathbf{v} - 1). \tag{20}$$

To find the RRR estimate, we will simply differentiate this loss with respect to the parameters $\mathbf{u}$ and $\mathbf{v}$, set them to zero, and solve. Using identities for matrix derivatives [e.g., 17] we obtain the following partial derivatives:

$$\frac{\partial L}{\partial \mathbf{u}} = -2X^\top Y\mathbf{v} + 2X^\top X\mathbf{u} \tag{21}$$

$$\frac{\partial L}{\partial \mathbf{v}} = -2Y^\top X\mathbf{u} + 2(\mathbf{u}^\top X^\top X\mathbf{u})\,\mathbf{v} + 2\lambda\mathbf{v}. \tag{22}$$

Setting (eq. 21) to zero leads to a solution for $\mathbf{u}$ in terms of $\mathbf{v}$:

$$\hat{\mathbf{u}} = (X^\top X)^{-1}X^\top Y\mathbf{v} = \hat{W}_{\mathrm{LS}}\mathbf{v}, \tag{23}$$

where $\hat{W}_{\mathrm{LS}}$ denotes the least-squares solution (eq. 6). We can then substitute this expression for $\hat{\mathbf{u}}$ into (eq. 22) and set it to zero, which gives us the equation:

$$(Y^\top X\hat{W}_{\mathrm{LS}})\mathbf{v} = \mu\mathbf{v}, \qquad \mu = \lambda + \mathbf{u}^\top X^\top X\mathbf{u}. \tag{24}$$

The matrix $Y^\top X\hat{W}_{\mathrm{LS}} = Y^\top X(X^\top X)^{-1}X^\top Y$, which can be seen to be equal to $\hat{W}_{\mathrm{LS}}{}^\top X^\top X\hat{W}_{\mathrm{LS}}$, is therefore positive-semidefinite. Equation (24) is therefore an eigenvector equation, since it shows that a symmetric matrix times $\mathbf{v}$ is equal to a scalar μ times $\mathbf{v}$. We can conclude that the solution $\hat{\mathbf{v}}$ must be an eigenvector of $(\hat{W}_{\mathrm{LS}}{}^\top X^\top X\hat{W}_{\mathrm{LS}})$. With a little more work, we can show that the optimum is obtained with $\hat{\mathbf{v}}$ set to the top eigenvector, i.e., the eigenvector with largest eigenvalue (see Appendix D), which is also the top right eigenvector of the linear prediction $X\hat{W}_{\mathrm{LS}}$. Plugging this solution back into (eq. 23) gives

$$\hat{\mathbf{u}} = \hat{W}_{\mathrm{LS}}\hat{\mathbf{v}}, \tag{25}$$

concluding our derivation.

3.3 | Geometric intuition

To gain geometric intuition for RRR, it is helpful to view the RRR estimation procedure as a sequence of orthogonal projections. First, the least-squares fit (eq. 6) is obtained by orthogonally projecting each column of the response matrix Y onto the column space of the design matrix X:

$$\hat{\mathbf{y}}_{i,\mathrm{LS}} = X\mathbf{w}_{i,\mathrm{LS}} = X(X^\top X)^{-1}X^\top \mathbf{y}_i, \tag{26}$$

where $\mathbf{y}_i$ and $\mathbf{w}_{i,\mathrm{LS}}$ represent corresponding columns of Y and $\hat{W}_{\mathrm{LS}}$ (Fig. 2A). Collecting these projections across response variables gives the least-squares prediction $\hat{Y}_{\mathrm{LS}} = X\hat{W}_{\mathrm{LS}}$.

The key geometric insight behind RRR is that its rank constraint can then be imposed directly on these least-squares predictions. To see this, we can rewrite the RRR loss in terms of $\hat{Y}_{\mathrm{LS}}$:

$$||Y - X\mathbf{u}\mathbf{v}^\top||_F^2 = ||(Y - \hat{Y}_{\mathrm{LS}}) + (\hat{Y}_{\mathrm{LS}} - X\mathbf{u}\mathbf{v}^\top)||_F^2 \tag{27}$$

$$= ||Y - \hat{Y}_{\mathrm{LS}}||_F^2 + ||\hat{Y}_{\mathrm{LS}} - X\mathbf{u}\mathbf{v}^\top||_F^2 + 2\mathrm{Tr}[(Y - \hat{Y}_{\mathrm{LS}})(\hat{Y}_{\mathrm{LS}} - X\mathbf{u}\mathbf{v}^\top)^\top]. \tag{28}$$

The least-squares residual $Y - \hat{Y}_{\mathrm{LS}}$ is orthogonal to the column space of X, while every column of both $\hat{Y}_{\mathrm{LS}}$ and $X\mathbf{u}\mathbf{v}^\top$ lies within this space. The cross term therefore vanishes, giving

$$||Y - X\mathbf{u}\mathbf{v}^\top||_F^2 = ||Y - \hat{Y}_{\mathrm{LS}}||_F^2 + ||\hat{Y}_{\mathrm{LS}} - X\mathbf{u}\mathbf{v}^\top||_F^2. \tag{29}$$

The first term is simply the least-squares error and is independent of $\mathbf{u}$ and $\mathbf{v}$. Thus, minimizing the RRR loss is equivalent to finding the best low-rank approximation to $\hat{Y}_{\mathrm{LS}}$.

This gives the second projection in our geometric picture. The best rank-1 approximation is obtained by identifying the leading principal component of $\hat{Y}_{\mathrm{LS}}$ in response space (Fig. 2B) and projecting the least-squares predictions onto this axis (Fig. 2C). More generally, rank-r RRR projects $\hat{Y}_{\mathrm{LS}}$ onto the subspace spanned by its top r principal components. Thus, RRR can be summarized as two successive orthogonal projections:

$$Y \xrightarrow[\text{column-wise}]{\text{project onto col}(X)} \hat{Y}_{\mathrm{LS}} \xrightarrow[\text{response space}]{\text{project onto top } r \text{ PCs}} \hat{Y}_{\mathrm{RRR}}. \tag{30}$$

The first projection retains the component of each response that is predictable from X, while the second retains the dominant r-dimensional structure within these predictions.

4 | GAINING INSIGHT INTO RRR

Although we have now seen a formal derivation of RRR using Lagrange multipliers, the reader may still be curious about how or why the RRR estimator differs from the optimal low-rank approximation of $\hat{W}_{\mathrm{LS}}$ using SVD. To provide more intuition, we examine the relationship between RRR and a variety of related methods (summarized in Box 1). In the following, we will provide a brief discussion of each method, along with the conditions under which it is identical to RRR.

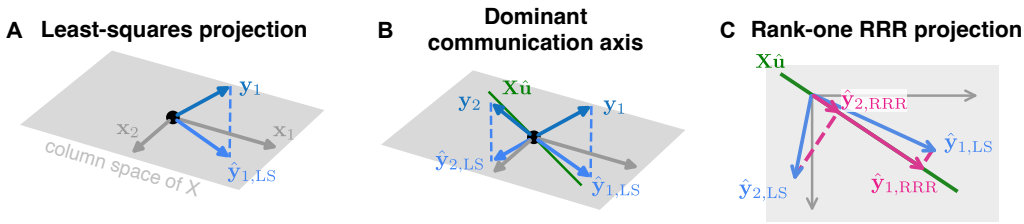


FIGURE 2 Geometric interpretation of rank-1 RRR. **(A)** Least-squares fitting projects the observed output responses Y onto the column space of X, yielding $\hat{Y}_{\text{LS}}$. **(B)** The dominant output-space communication axis $\hat{\mathbf{v}}$ determines the corresponding input-space direction $X\hat{\mathbf{u}} = X\hat{W}_{\text{LS}}\hat{\mathbf{v}}$. **(C)** Projecting $\hat{Y}_{\text{LS}}$ onto $X\hat{\mathbf{u}}$ yields the rank-1 RRR fitted responses $\hat{Y}_{\text{RRR}}$.

4.1 | Relationship to SVD

Given that the SVD can be used to find the optimal low-rank approximation to any matrix (Sec. 2.3), readers might naturally wonder why the RRR estimator is not equivalent to the optimal rank-r approximation to the least-squares estimate $\hat{W}_{\text{LS}}$. For starters, we can observe that the two estimates *are* identical when the covariance of the input population is isotropic or spherical, or equivalently, when $X^\top X = aI$ for some scalar a.

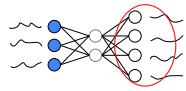
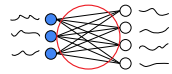
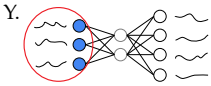
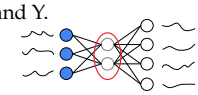
To see this, the least-squares estimate in this case is equal to $\hat{W}_{\text{LS}} = (X^\top X)^{-1}X^\top Y = (1/a)X^\top Y$. We can compute the SVD of this matrix to obtain $\hat{W}_{\text{LS}} = USV^\top$, and then extract the top r singular vectors and values to find the optimal rank-r approximation, which we write as $U_r S_r {V_r}^\top$. For RRR, the row-space matrix $\hat{V}_{\text{RRR}}$ is given by the top r eigenvectors of $\hat{W}_{\text{LS}}{}^\top X^\top X \hat{W}_{\text{LS}} = a\hat{W}_{\text{LS}}{}^\top \hat{W}_{\text{LS}} = aVS^2V^\top$, which is clearly equal to V_r. Given that $\hat{V}_{\text{RRR}} = V_r$, the final RRR estimate is then given by $\hat{W}_{\text{RRR}} = \hat{W}_{\text{LS}}V_r{V_r}^\top = (USV^\top)V_r{V_r}^\top = U_rS_r{V_r}^\top$, which is the same as the SVD-based estimate given above. Thus, if the input population activity is spherical, meaning that it has equal variance in all directions, RRR and SVD give the same estimate of the low-dimensional communication channel. (See Fig. 3A for an illustration).

By contrast, when the input activity distribution is non-spherical, the RRR estimate may differ from the best low-rank approximation to the linear weight matrix (see Fig. 3B-C). The reason for this is that the amount of explained variance along any output dimension is the amount of variance along the corresponding input dimension times the square of the linear weight connecting the input dimension to output dimension.

We can illustrate this with a simple example. Consider a rank-2 weight matrix with SVD given by $W = s_1\mathbf{u}_1{\mathbf{v}_1}^\top + s_2\mathbf{u}_2{\mathbf{v}_2}^\top$, where $s_1 > s_2$ are the singular values of W, and $(\mathbf{u}_1, \mathbf{u}_2)$ and $(\mathbf{v}_1, \mathbf{v}_2)$ are left and right singular vectors, respectively. In this case, the best rank-1 approximation to W in a least squares sense is $\hat{W}_{\text{SVD}} = s_1\mathbf{u}_1{\mathbf{v}_1}^\top$. However, if we consider an input population X that has variance σ_1^2 along the $\mathbf{u}_1$ axis and variance σ_2^2 along the $\mathbf{u}_2$ axis, then the amount of variance explained by the two rank-1 components will be $\sigma_1^2 s_1^2$ and $\sigma_2^2 s_2^2$, respectively. If $\sigma_1^2 s_1^2 < \sigma_2^2 s_2^2$, the rank-1 RRR estimate will correspond to $\hat{W}_{\text{RRR}} = s_2\mathbf{u}_2{\mathbf{v}_2}^\top$, differing from the rank-1 approximation $\hat{W}_{\text{SVD}}$. See Fig. 3 for a two-dimensional illustration of this phenomenon.

4.2 | Relationship to Principal Component Regression (PCR)

Principal Component Regression (PCR) provides another method for finding a low-dimensional communication channel between a pair of brain regions. However, PCR uses the principal components of the input region to identify the input dimensions, rather than taking into account the neural activity from both brain regions. Once the input dimen-

Method	Loss function	Description
Reduced-Rank Regression (RRR)	$\|Y - XUV^\top\|^2$	Seek to capture the most variance in Y. 
SVD of least-squares estimates	$\left\|\hat{W}_{LS} - UV^\top\right\|^2$	Seeks to capture most variance in $\hat{W}_{LS}$ 
Principal Components Regression (PCR)	$\left\|Y - X\hat{U}_{pca}V^\top\right\|^2$	Uses maximum-variance dimensions of X to predict Y. 
Canonical Correlation Analysis (CCA)	$1 - \mathrm{corr}(Y\mathbf{v}, X\mathbf{u})$	Seeks to find maximally correlated projections of X and Y. 

box 1 Comparison of RRR with other regression and dimensionality-reduction methods.

sions are determined, PCR performs least-squares regression to find a linear mapping to the output space. The PCR weights can thus be written in factorized form:

$$\hat{W}_{\text{PCR}} = \hat{U}_{\text{PCA}}\hat{V}_{\text{PCR}}{}^\top, \tag{31}$$

where $\hat{U}_{\text{PCA}}$ represents the principal components of the input population (i.e., the eigenvectors of $X^\top X$), and we use least-squares to obtain the mapping to Y:

$$\hat{V}_{\text{PCR}} = (\tilde{X}^\top\tilde{X})^{-1}\tilde{X}^\top Y, \tag{32}$$

where $\tilde{X} = X\hat{U}_{\text{PCA}}$ is the projection of X onto the top r principal components defined by $\hat{U}_{\text{PCA}}$.

The PCR estimate thus assumes that the leading components of the input population are also the most predictive of the output population—a premise that, as we will discuss in Sec. 6, does not always hold.

4.3 | Relationship to Canonical Correlation Analysis (CCA)

Canonical Correlation Analysis (CCA) [18] is another dimensionality-reduction method that has been used to model communication between brain regions [19]. However, CCA differs from RRR in that it does not arise from a regression model, meaning that it does not seek to predict the activity in one brain region from activity in the other. Instead, CCA attempts to identify subspaces of X and Y in which the responses are maximally correlated. For a 1D subspace,

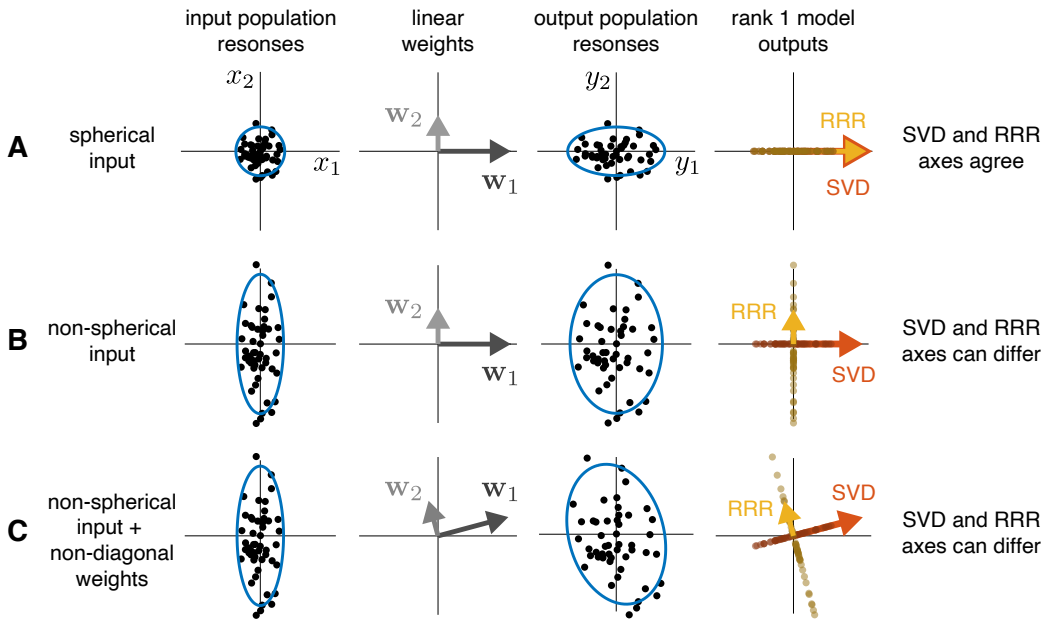


FIGURE 3 Geometric illustration of differences between RRR and singular value decomposition (SVD). Each row shows a different example of (noiseless) linear communication between a 2D input population and a 2D output population via a 2×2 weight matrix: $\mathbf{y} = W^\top \mathbf{x} = \mathbf{w}_1 x_1 + \mathbf{w}_2 x_2$, where $\mathbf{w}_1$ and $\mathbf{w}_2$ denote the columns of $W^\top$. In all three examples, we set $\mathbf{w}_1$ and $\mathbf{w}_2$ to be orthogonal, with $||\mathbf{w}_1|| = 2||\mathbf{w}_2||$. This means that the rank-1 reconstruction of the weights using SVD is always $\hat{W}_{\text{SVD}}{}^\top = [\mathbf{w}_1\ 0]$, preserving the $\mathbf{w}_1$ axis and discarding $\mathbf{w}_2$. However, the rank-1 RRR estimate can vary depending on the input distribution. **(A)** Spherical input distribution. Here, the principal axes of the outputs $\mathbf{y}$ (3rd column) are determined entirely by $\mathbf{w}_1$ and $\mathbf{w}_2$, and the rank-1 RRR and SVD axes (arrows) and predictions (dots) are always identical (4th column). **(B)** Non-spherical input distribution with 9-times larger variance in the x_2 than the x_1 component. Here, the $\mathbf{w}_2$ weight vector captures more of the variance in $\mathbf{y}$ than $\mathbf{w}_1$ (yellow dots vs. red dots) and the RRR estimate is $\hat{W}_{\text{RRR}}{}^\top = [0\ \mathbf{w}_2]$, preserving a subspace orthogonal to $\hat{W}_{\text{SVD}}$. **(C)** Analogous to (B), except the weights $\mathbf{w}_1$ and $\mathbf{w}_2$ are rotated by 15°. Once again, the output distribution has greater predictable variance along the $\mathbf{w}_2$ axis than along the $\mathbf{w}_1$ axis, so the rank-1 RRR estimate and SVD axes disagree.

the CCA objective is to maximize

$$\text{corr}(Y\mathbf{v}, X\mathbf{u}), \tag{33}$$

for vectors $\mathbf{u}$ and $\mathbf{v}$. Let $\mathbf{a}$ and $\mathbf{b}$ be the top left and right singular vectors, respectively, of the matrix

$$(Y^\top Y)^{-\frac{1}{2}} Y^\top X (X^\top X)^{-\frac{1}{2}}. \tag{34}$$

The CCA solution is $\mathbf{v} = (Y^\top Y)^{-1/2}\mathbf{a}$ and $\mathbf{u} = (X^\top X)^{-1/2}\mathbf{b}$, and the top singular value equals the maximal canonical correlation. The method can be extended to higher-dimensional subspaces by preserving additional right and left singular vectors (see [20]).

Thus, CCA differs from RRR in that it aims to find dimensions of correlated activity without maximizing variance

explained in either region. This means that, for example, it is possible to have a CCA axis along which responses are highly correlated, but which does not explain very much variability in the output region.

It also follows that, when Y is whitened, CCA and RRR yield equivalent solutions. Conceptually, RRR assigns equal weight to every dimension of the whitened output space, matching CCA. Algebraically, when $Y^\top Y = I$, the left singular vectors of the matrix in (eq. 34) coincide with the right singular vectors of $X\hat{W}_{\text{LS}}$.

5 | EXTENDING RRR

In this section, we discuss two powerful extensions to the standard RRR estimator. Both were described in previous papers [1, 21], but remain underappreciated. The first extension involves regularization with an L_2 or "ridge" penalty, which is helpful in noisy or high-dimensional settings. The second involves incorporating non-spherical noise, which may be important when response noise is correlated or unequal across neurons.

5.1 | Ridge-RRR: incorporating regularization

When a dataset has a low signal-to-noise ratio, which can arise when sample size is small relative to the number of neurons, or when response noise is large, estimation performance can be substantially improved by regularization. Regularization generally refers to a wide variety of techniques for constraining or penalizing a model's parameters to reduce over-fitting [22, 23].

The low-rank constraint inherent to RRR can be seen as one useful form of regularization, since a low-rank weight matrix has fewer degrees of freedom than the full-rank least-squares estimate (eq. 6). However, it is possible to further regularize the RRR estimate using a penalty on the squared L_2 norm of the regression weights, an approach commonly known as ridge regression [24, 21]. Even in the low-rank setting, the ridge penalty corresponds to a penalty on the sum of squares of all elements in the weight matrix $W = UV^\top$:

$$||W||_F^2 = \text{Tr}[W^\top W] = \text{Tr}[VU^\top UV^\top] = \text{Tr}[U^\top UV^\top V] = \text{Tr}[U^\top U] = ||U||_F^2, \tag{35}$$

where we have exploited the cyclic property of the trace and the fact that we constrained V so that $V^\top V = I$ (see (eq. 19) and Appendix B). Thus, the ridge penalty can be simplified to a penalty on the sum of squared elements in U.

We can derive the "ridge-RRR" estimator by minimizing the following penalized loss function:

$$L_{\text{ridge}}(U, V) = ||Y - XUV^\top||_F^2 + \lambda_{\text{ridge}}\text{Tr}[U^\top U], \tag{36}$$

where λ_{ridge} is the ridge hyperparameter penalizing the L_2 norm of the weights. This hyperparameter determines how much the squared elements of W are penalized, and thus sets the trade-off between minimizing error in Y and keeping the weights W small. In practice, λ_{ridge} is typically set via cross-validation.

The derivation of the ridge-RRR (RRRR) estimate follows similar steps to those in Sec. 3.2, and is provided in Appendix C.1. The solution is given by $\hat{W}_{\text{RRRR}} = \hat{U}_{\text{RRRR}}\hat{V}_{\text{RRRR}}{}^\top$, with

$$\hat{V}_{\text{RRRR}} = \text{eig}(\hat{W}_{\text{ridge}}{}^\top X^\top X\hat{W}_{\text{ridge}}) \tag{37}$$

$$\hat{U}_{\text{RRRR}} = \hat{W}_{\text{ridge}}\hat{V}_{\text{RRRR}}, \tag{38}$$

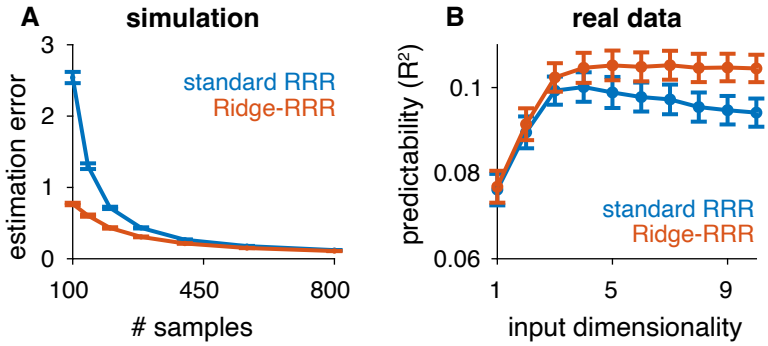


FIGURE 4 Comparison of standard and "ridge" regularized RRR. **(A)** For each sample size, we performed 50 repetitions of a simulated experiment with $m = 50$ input neurons and $n = 50$ output neurons interacting via a communication channel of rank $r = 2$. Traces show the average error (±1 SEM) in recovering the true weights, showing that a ridge penalty can substantially reduce error in low-sample regimes. **(B)** Analysis of an example recording in primate visual cortex [2], in which we varied the rank of an estimated communication channel from V1 to V2 using 111 V1 and 37 V2 neurons. Traces show prediction performance (±1 SD over 10 folds of cross-validation) for both estimators. Ridge-RRR (red) outperformed standard RRR (blue) for all ranks, with larger improvements for higher-rank models.

where $\hat{W}_{\text{ridge}} = (X^\top X + \lambda_{\text{ridge}} I)^{-1} X^\top Y$ is the standard (full-rank) ridge regression estimate for W. Thus, we have the nice result that the ridge-RRR estimate is computed using exactly the same steps as the RRR estimate, only with the ridge estimate $\hat{W}_{\text{ridge}}$ in place of the ordinary least squares estimate $\hat{W}_{\text{LS}}$.

To explore the benefits of the ridge penalty, we performed simulated experiments to compare errors in standard and ridge RRR when generating data from a ground-truth model. We simulated responses from a population of $m = 50$ input neurons and $n = 50$ output neurons with a communication channel of rank $r = 2$, whose weights $W = UV^\top$ were obtained by sampling $U \in \mathbb{R}^{50\times 2}$ and $V \in \mathbb{R}^{50\times 2}$ from a standard normal distribution (see Appendix E for details). The activities of the input population $\{\mathbf{x}_t\}$ were then drawn from a standard normal distribution, and the output activities were generated via $\mathbf{y}_t = W^\top \mathbf{x}_t + \epsilon_t$, with noise $\epsilon_t \sim \mathcal{N}(0, 2500I)$. We set the ridge penalty via cross-validation, obtaining an optimum at $\lambda_{\text{ridge}} = 100$. We then varied the number of samples used for estimation, and evaluated performance in recovering the true weights W via both standard RRR and ridge-RRR. This confirmed that regularization using a ridge penalty can dramatically improve performance in low-sample settings (Fig. 4A).

We then examined applicability to real neural data using an example session from Semedo et al. [2], consisting of simultaneous recordings from primate V1 and V2. We estimated a low-rank communication channel from a population of 111 recorded V1 neurons to 37 recorded V2 neurons, varying the rank of the inferred weights from 1 to 10. For each rank, we computed test error using 10-fold cross-validation. To set the ridge penalty, we considered values of λ_{ridge} ranging from 0 to 10^4 and selected the value that yielded the lowest test error ($\lambda_{\text{ridge}} = 5000$). The ridge-RRR estimate outperformed the standard RRR across all ranks in terms of R^2 on held-out test data (Fig. 4B). Moreover, it allowed us to identify a higher-dimensional communication channel: the standard RRR estimate achieved maximal performance at rank 4, whereas the ridge-RRR estimate continued improving up to rank 7. This shows that the use of regularization—in addition to improving estimation of the regression weights—can allow for the recovery of additional communication dimensions that would otherwise be subsumed by noise.

5.2 | Full-covariance RRR: incorporating non-spherical noise

In standard RRR, we set out to minimize the sum of squared errors between Y and a low-rank linear prediction XW, which is equivalent to a maximum likelihood estimate under a linear model with spherical Gaussian noise (eqs. 1–2). However, this model thus assumes all the output neurons have the same amount of (independent) noise in their activities, which may be inaccurate in real neural recordings. To relax this assumption, we might wish to allow for noise in $\mathbf{y}_t$ that is correlated, or of unequal variance across neurons. Let us therefore consider the more general linear-Gaussian response model:

$$\mathbf{y}_t \mid \mathbf{x}_t \sim \mathcal{N}(W^\top \mathbf{x}_t, \Sigma), \tag{39}$$

where Σ denotes an arbitrary covariance matrix governing the noise in $\mathbf{y}_t$. In this case, the maximum likelihood estimator can instead be written as the minimization of the following objective function, which arises from the negative log-likelihood of a multivariate normal distribution:

$$L_{\text{fcRRR}}(U, V) = \text{Tr}\big[(Y - XUV^\top)\Sigma^{-1}(Y - XUV^\top)^\top\big], \tag{40}$$

The minimum of this objective, which we refer to as the full-covariance RRR ("fcRRR") estimator, is given by $\hat{W}_{\text{fcRRR}} = \hat{U}_{\text{fcRRR}}\hat{V}_{\text{fcRRR}}{}^\top$, with

$$\hat{V}_{\text{fcRRR}} = \Sigma^{\frac{1}{2}}\,\text{eig}(\Sigma^{-\frac{1}{2}}\hat{W}_{\text{LS}}{}^\top X^\top X \hat{W}_{\text{LS}}\Sigma^{-\frac{1}{2}}) \tag{41}$$

$$\hat{U}_{\text{fcRRR}} = \left(\hat{W}_{\text{LS}}\Sigma^{-\frac{1}{2}}\right)\Sigma^{-\frac{1}{2}}\hat{V}_{\text{fcRRR}}. \tag{42}$$

This is equivalent to performing PCA on the linear projection $X\hat{W}_{\text{LS}}$, but whitened by the inverse of the noise covariance Σ. (See Appendix C.2 for full derivation). Note however that $\hat{U}_{\text{fcRRR}}$ and $\hat{V}_{\text{fcRRR}}$ are not in general semi-orthogonal, since the matrix of orthogonal eigenvectors in (eq. 41) is transformed by $\Sigma^{\frac{1}{2}}$ to obtain $\hat{V}_{\text{fcRRR}}$. If an orthogonal set of basis vectors is preferred, we could certainly obtain one by taking the SVD of $\hat{W}_{\text{fcRRR}}$ after we have computed it.

In practice, the noise covariance Σ is unknown *a priori*, and must be inferred from data. To apply full-covariance RRR to data, we can thus perform coordinate ascent of the log-likelihood, in which we initialize Σ to the identity, then alternate between estimating $\hat{W}_{\text{fcRRR}}$ given the current estimate of noise covariance $\hat{\Sigma}$, and updating $\hat{\Sigma}$ using the maximum likelihood estimator under the current weights, which is given by the sample covariance of the residuals $(Y - X\hat{W}_{\text{fcRRR}})$. We repeat this iterative procedure until convergence (typically 1-10 iterations in our experiments).

To illustrate the benefits of full-covariance RRR, we performed simulated experiments comparing standard and full-covariance RRR for populations with increasing degrees of anisotropy in the output noise (Fig. 5A). We set up these simulations identically to those in the previous section ($m = n = 50$, $r = 2$), except that the noise was no longer spherical. Specifically, we considered a noise covariance with increased variance along one dimension: $\Sigma = \text{diag}(1, \ldots, 1, \sigma^2_{\text{max}})$, while varying the value of σ^2_{max}. This revealed that the larger the departure from sphericity, the greater the relative improvement of the full-covariance RRR estimator.

We also applied full-covariance RRR to the same example primate V1-V2 recording [2] considered in Fig. 4B. Using the same procedure—predicting activity in 37 V2 neurons from 111 V1 neurons with 10-fold cross-validation—we again observed an improvement in predictive performance relative to standard RRR (Fig. 5B), although the effect is smaller compared to ridge-RRR. We think this smaller effect may be explained by the estimated covariance deviating from the true noise covariance with limited samples.

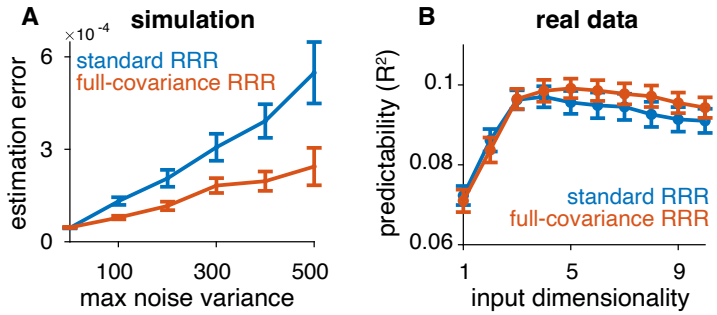


FIGURE 5 Comparison of standard and full-covariance RRR. **(A)** We varied non-sphericity of the response noise by increasing its variance along a single dimension, setting the noise covariance to $\Sigma = \mathrm{diag}(1, \ldots, 1, \sigma^2_{\max})$ for varying values of $\sigma^2_{\max}$. For each noise covariance, we performed 50 repeated experiments using a rank-2 communication channel between $m = 50$ input neurons and $n = 50$ output neurons. We then inferred the low-rank communication weights from a dataset of $T = 1000$ samples. Traces show the mean error (±1 SEM) in recovering the true weights as a function of $\sigma^2_{\max}$, confirming that the full-covariance estimator (red) dramatically outperforms standard RRR (blue) as the noise becomes more non-spherical. **(B)** Analysis of the same example recording as in Fig. 4 from Semedo et al. [2], in which we varied the rank of an estimated communication channel from V1 to V2 using 111 V1 and 37 V2 neurons. Traces show prediction performance (±1 SD over 10 folds of cross-validation) for both estimators. Full-covariance RRR (red) outperformed standard RRR (blue) for all ranks higher than 3, although the improvement is smaller than that obtained by adding a ridge penalty.

6 | CHARACTERIZING COMMUNICATION ALIGNMENT

Despite widespread interest in low-rank neural communication, there has been little work on the problem of quantifying the relationship between communication axes and the dominant modes of neural activity. This problem arises somewhat separately for input and output populations. On the input side, one can ask whether the dominant modes of activity—as defined by principal components of the input population—are closely aligned with the communication subspace, meaning that the dominant modes provide the primary drive to the output population. (In this case, we would expect principal components regression (PCR) to do perfectly well at characterizing the communication channel). Or conversely, do the dominant modes of the input population primarily constitute private dimensions, with only smaller modes of activity providing input to the communication channel? On the output side, we can similarly ask whether the output of the communication channel aligns with the dominant or the minor modes of the output population activity.

Fundamentally, these questions revolve around the relationship between the principal components of the input and output populations and the subspaces defined by U and V matrices, while taking into account that different dimensions of the communication channel may be weighted unequally. Here we introduce metrics that seek to simplify this comparison by quantifying the degree to which communication is aligned with maximal or minimal modes of input and output populations, respectively.

6.1 | Quantifying input alignment

First, we introduce an *input alignment index* to quantify the degree to which communication is maximally aligned with the dominant modes or PCs of the input population. Conceptually, we say that they are "aligned" if communication

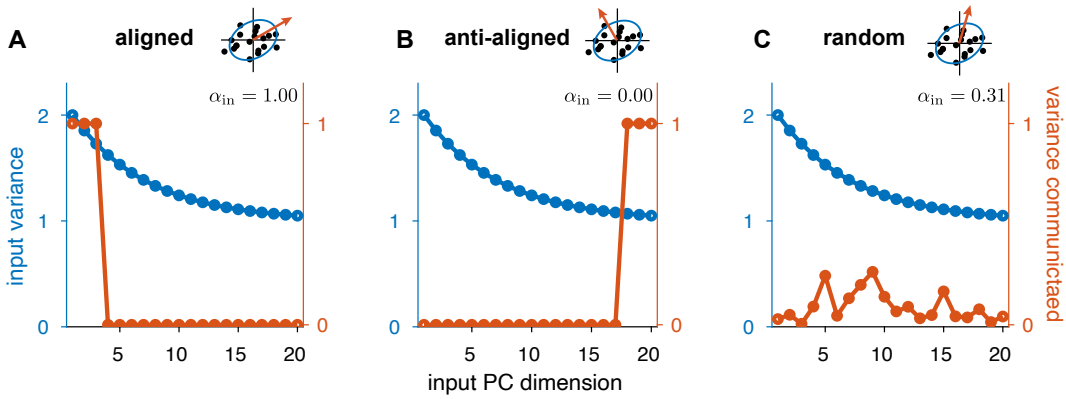


FIGURE 6 Examples of input variance and variance communicated when input is aligned (left), anti-aligned (middle), and randomly oriented (right) to the communication axes. X-axis shows the input activity principal component dimensions. Left y-axis (blue) is the input variance distribution across the dimensions; right y-axis (orange) is the normalized variance communicated to the output region.

subspace is aligned with the dominant modes of input population activity (Fig. 6, left); and "anti-aligned" if it is more aligned with the smallest-variance modes of the input population (Fig. 6, middle).

We can start by computing the variance of the communication channel output (or "communication variance") by projecting the input covariance onto the inferred communication weights W, giving:

$$\alpha_{\text{in}}^{\text{raw}} = \text{Tr}[\, W^\top \Sigma_X W \,], \tag{43}$$

where $\Sigma_X = \frac{1}{T} X^\top X$ denotes the covariance of the input population.

We can then compare this quantity to the *maximal* amount of variance that could be obtained if the communication channel were maximally aligned with the dominant modes of the input population. This corresponds to permuting and rotating the communication weights so that the left singular vectors, ordered from largest to smallest singular value, are aligned with the dominant principal components of the input population. In this case, the communication variance would be given by:

$$\alpha_{\text{in}}^{\text{max}} = \sum_{i=1}^{r} \lambda_i^2 \sigma_i^2 \tag{44}$$

where r is the communication rank, λ_i denotes the i'th singular value of W, and σ_i^2 is the i'th eigenvalue of Σ_X (corresponding to the variance along the i'th input mode).

Similarly, the *minimum* possible communication variance would be obtained by rotating W so that its dominant singular vectors are aligned with the *smallest* modes of the input population activity. In this case, the communication variance would be given by:

$$\alpha_{\text{in}}^{\text{min}} = \sum_{i=1}^{r} \lambda_i^2 \sigma_{m+1-i}^2, \tag{45}$$

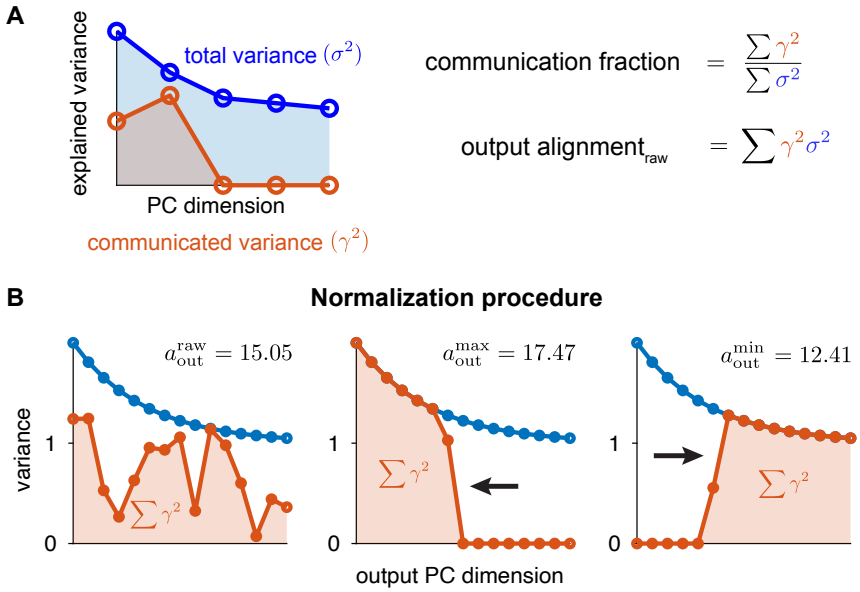


FIGURE 7 Illustration of two indices for quantifying the relationship between the low-rank communication channel and the output population activity: *communication fraction* and *output alignment index*. **(A)** Schematics showing *communication fraction*: the ratio of communication variance to total output-population variance and *output alignment index*: measures how strongly the communication channel aligns with dominant modes of the output population activity. **(B)** Example variance profiles and resulting index values when variance is randomly oriented (left), aligned (middle), anti-aligned (right) with respect to the communication axes.

where $\sigma^2_{m+1-r}, \ldots, \sigma^2_m$ correspond to the variance along the r smallest modes of the input population activity, sorted from largest to smallest.

To obtain an alignment index that is normalized between 0 (minimal alignment) and 1 (maximal alignment), we define the input alignment index as:

$$\alpha_{\text{in}} = \frac{\alpha_{\text{in}}^{\text{raw}} - \alpha_{\text{in}}^{\text{min}}}{\alpha_{\text{in}}^{\text{max}} - \alpha_{\text{in}}^{\text{min}}}. \tag{46}$$

To illustrate the input alignment index obtained by different forms of communication channel alignment, we calculated the input alignment index for the three cases shown in Fig. 6. As expected, $\alpha_{\text{in}} = 1$ when the communication subspace is aligned with the activity modes, $\alpha_{\text{in}} = 0$ when they are anti-aligned, and it falls between 0 and 1 in the random case (values shown in the top-right corner of each subplot).

6.2 | Quantifying output alignment

Before considering the problem of output alignment, it is natural to ask: How *much* of the output region variance can be explained by the communication channel? We can imagine one extreme where the output population can be perfectly explained by a linear transformation of input region activity, $Y = XW$, with no residual error. In this case,

the communication variance accounts for the entirety of the output variance. At the other extreme, one can imagine cases where there is very little communication, so the residual error $(Y - XW)$ is large relative to the variance of the linear prediction XW. In this case, communication variance makes up a tiny fraction of total output variance. To quantify such differences, we introduce the *communication fraction* (CF), defined as the ratio of the total communication variance to the total variance in the output population:

$$CF = \frac{\mathrm{Tr}[W^\top \Sigma_X W]}{\mathrm{Tr}[\Sigma_Y]}, \tag{47}$$

where $\mathrm{Tr}[W^\top \Sigma_X W]$ is the total communication variance and $\Sigma_Y = \frac{1}{T} Y^\top Y$ is the covariance of the output region activity. It is equivalent to $\|XW\|_F^2 / \|Y\|_F^2$. When applied to the training data used to fit W, this fraction is bounded between 0 (no communication) and 1 (maximal communication).

Next, we turn to the problem of quantifying the alignment between the communication channel and output population activity, which poses a slightly different problem than the input alignment case. In the input region, the communication axes can have an arbitrary relationship to the input population covariance, so it is possible to consider how the communication variance would change if we rotated them. In the output region, however, the communicated variance along each dimension is upper-bounded by the total variance of the output population along that dimension. Rotating the right singular vectors of W would change the covariance of the output population, making it impossible to take the same approach for computing an alignment index.

To define an *output alignment index*, we instead examine how much of the total communicated variance falls along large vs. small modes of output population activity. Let $\{(\boldsymbol{\mu}_j, \sigma_j^2)\}_{j=1}^n$ denote the sorted eigenvectors and eigenvalues of Σ_Y, which define the principal components and their associated variances. Then the communicated variance along output mode j is given by

$$\gamma_j^2 = \boldsymbol{\mu}_j{}^\top (W^\top \Sigma_X W) \boldsymbol{\mu}_j. \tag{48}$$

Intuitively, we wish to compare the communicated variances γ_j^2 with the variance along each output mode σ_j^2, which upper bounds them. If γ_j^2 is close to σ_j^2 for early modes (i.e., small indices j), and close to 0 for later modes (larger j) then we would say that the communication channel is strongly aligned with the dominant modes of the output population. On the other hand, if γ_j^2 values are near zero for small j and approach σ_j^2 for large j, then we would say that the communication channel is predominantly aligned with the minor modes of output population activity, and thus mis-aligned with dominant modes. Note that the communication fraction (eq. 47), which quantifies total communication variance relative to total output population variance, can also be written as $CF = (\sum \gamma_j^2)/(\sum \sigma_j^2)$, as illustrated in Fig. 7A.

To define an output alignment index, we simply compute the dot product between the communicated variances and the output variances (i.e., eigenvalues) along those same axes:

$$\alpha_{\mathrm{out}}^{\mathrm{raw}} = \sum_{j=1}^{n} \gamma_j^2 \sigma_j^2. \tag{49}$$

We can then normalize this quantity so that it lies between zero and one by computing the maximal and minimal values of this dot product for the same total communication variance. These extrema would be obtained if the communication variances γ_j^2 were adjusted so that they were maximally aligned with early or late eigenvalues, while preserving their sum.

When pushing all communicated variance $\Gamma = \sum_{j=1}^{n} \gamma_j^2$ towards early eigenvalues (Fig. 7B, middle), we can "fill up" first k eigenvalues with the communicated variance so that $\gamma_j^2 = \sigma_j^2$ for $j = 1, \ldots, k$, where k is the largest integer satisfying $\sum_{j=1}^{k} \sigma_j^2 < \Gamma$. The remaining variance is allocated to the $(k+1)$-th direction, such that $\gamma_{k+1}^2 = \Gamma - \sum_{j=1}^{k} \sigma_j^2$. This allocation yields the maximum possible output alignment index:

$$\alpha_{\text{out}}^{\max} = \sum_{j=1}^{k} \sigma_j^4 + \sigma_{k+1}^2 \left(\Gamma - \sum_{j=1}^{k} \sigma_j^2 \right). \tag{50}$$

Conversely, to minimize the output alignment index, we allocate communicated variance to the last eigenvalue directions (Fig. 7B, right). We set $\gamma_j^2 = \sigma_j^2$ for $j = l, \ldots, n$, where l is the smallest integer satisfying $\sum_{j=l}^{n} \sigma_j^2 < \Gamma$, "filling up" the last $n - l + 1$ eigenvalues. The remaining variance is allocated to the $(l-1)$-th direction. This allocation yields the minimum possible output alignment index:

$$\alpha_{\text{out}}^{\min} = \sum_{j=l}^{n} \sigma_j^4 + \sigma_{l-1}^2 \left(\Gamma - \sum_{j=l}^{n} \sigma_j^2 \right). \tag{51}$$

Normalizing the output alignment index to the range $[0, 1]$ then proceeds as above:

$$\alpha_{\text{out}} = \frac{\alpha_{\text{out}}^{\text{raw}} - \alpha_{\text{out}}^{\min}}{\alpha_{\text{out}}^{\max} - \alpha_{\text{out}}^{\min}}. \tag{52}$$

6.3 | Quantifying communication dimensionality

Although the dimensionality of communication is naively equal to the rank of the RRR model weights, communication may not be evenly distributed across these dimensions. In some cases, a large fraction of the predictive information may be captured by a small number of dimensions, with the remaining dimensions making only weak contributions. For example, a rank-10 RRR model might capture 95% of its variance in the first two dimensions, with the remaining eight dimensions accounting for only 5%. In this case, although the communication subspace is formally 10-dimensional, it may be useful to describe its effective dimensionality as being close to two.

One metric for quantifying this effective dimensionality is the participation ratio (PR), which summarizes how evenly the signal variance is distributed across dimensions:

$$\text{PR} = \frac{\left(\sum_{i=1}^{n} \lambda_i \right)^2}{\sum_{i=1}^{n} \lambda_i^2}, \tag{53}$$

where λ_i is the i-th eigenvalue of the covariance matrix of the communication channel output XW. The PR ranges from 1, when all predictable variance is concentrated in a single dimension, to r (dimensionality of W, $r \leq n$), when predictable variance is distributed equally across all r dimensions. Thus, unlike rank, the PR provides a graded measure of dimensionality that is sensitive to the relative strength of communication along different dimensions.

The participation ratio has been widely used to quantify the effective dimensionality of neural population activity [25]. Here, applying it to the predictable variance provides an analogous measure of the effective dimensionality of communication between populations.

7 | PRACTICAL CONSIDERATIONS

Now that we have reviewed the mathematical foundations of RRR and its relationship to other methods, we turn to practical considerations around its application to neural datasets. In particular, we will discuss issues related to the pre-processing of response data and methods for determining rank.

7.1 | Time binning

Given raw recordings from a pair of brain regions, several stages of pre-processing are required in order to form the matrices X and Y that serve as inputs to the RRR estimator. First, one must decide on the size of the time bins for discretizing (for spike trains) or averaging (for continuous responses like calcium fluorescence) the responses. Larger time bins will provide more reliable measurements of the mean activity in each bin, but will obscure signal fluctuations on faster timescales. Thus, time bin size should be set as large as possible without exceeding the timescale of signal fluctuations. Semedo et al. [2] used 100 ms time bins to analyze communication between V1 and V2 in response to drifting sinusoidal gratings, but smaller bins might be considered for transient or rapidly fluctuating stimuli, or for responses from the sensory or motor periphery. Conversely, larger bins might be sufficient for responses in higher-order cortical regions with slower dynamics.

If there is a signal delay between the two brain regions, it may be desirable to offset the time bins of the input and output region data so that the rows of X have maximal predictive performance for the rows of Y. A simple approach would be to select the time lag at which the cross-correlation between input and output region activity is maximized. Alternatively, one could compute the RRR estimate for a grid of time lags and select the value with maximal predictive performance.

7.2 | Centering and conditioning

Once the data are binned, the response of each neuron is typically centered to have a mean of zero, which is equivalent to subtracting off the mean of each column of the X and Y matrices. This ensures that RRR reflects shared fluctuations as opposed to the mean response pattern in each region. However, there is nothing wrong in principle with preserving a non-zero mean—in this case, the communication variance will also reflect contributions of the squared mean, and the rank of the communication channel will increase by 1 (unless the means lie in the span of the input and output subspaces inferred from the de-meaned data).

In Semedo et al. [2], the authors centered the data separately for each time bin relative to stimulus onset. This is equivalent to removing the peri-stimulus time histogram (PSTH), or the average time-varying stimulus component of each neuron's response before constructing the X and Y response matrices, which means that different portions of X and Y are centered independently. Under this approach, the inferred communication channel depends only on the residual fluctuations around the time-varying mean responses in each area. This choice seems appropriate if one assumes that the stimulus information arrives directly at the output region, bypassing the input region, as opposed to assuming that stimulus-related information must be passed from input to output region via the communication channel.

Another design choice of Semedo et al. [2] was to apply RRR separately to the data from each stimulus condition. Thus, for example, the authors inferred different communication channels for responses to 0° and 90° gratings. This approach allows researchers to assess whether the communication dimensions change across stimulus conditions, and provides greater flexibility if a single linear transformation cannot capture the patterns of shared variability that

arise across all stimuli. However, we note that a finding of low-dimensional communication for each stimulus or task condition may nevertheless be consistent with high-dimensional communication when considering all conditions.

Overall, we feel there is no "correct" choice about whether or not to subtract off time-varying mean responses, or to analyze stimulus conditions jointly or separately. Researchers should carefully consider the differences in interpretation that may result from these choices, and be sure to discuss them when communicating results.

7.3 | Determining rank

So far we have generally treated the rank of the RRR estimator as known or pre-specified. In practical settings, of course, the rank is not known *a priori*, and must therefore be inferred from the data.

One method for setting the rank is cross-validation, in which the reduced-rank weights are computed on a training set and then used to evaluate prediction performance on a held-out test set. We used this approach in Figs. 4B and 5B, computing test performance (R^2) as a function of rank for an example dataset from primate visual cortex [2]. These curves exhibit the expected pattern of rising to a maximum and then gradually decreasing. This decrease indicates that any increase in prediction performance conferred by adding dimensions is outweighed by the deleterious effects of overfitting. The location of the maximum can thus be taken as the optimal dimensionality of the inferred communication channel, or equivalently, the rank of the RRR weights. For this dataset, we found that standard RRR and ridge-RRR reached maximal performance at ranks of 4 and 7, respectively (Fig. 4B). This shows that appropriate regularization can allow for the recovery of higher-dimensional communication channels due to reductions in overfitting.

An alternate approach to setting the rank is to rely on heuristic criteria. For example, Semedo et al. [2] developed an approach that involved performing 10-fold cross-validation and then selecting rank using the smallest rank to achieve a performance level within one standard-error-of-the-mean (SEM) across folds of the maximum. This approach has the effect of selecting a rank near where the cross-validation curve flattens out, removing dimensions that capture small amounts of variance. This method also produces a rank estimate that is less variable across different splits of the data, since the exact location of the maximum may vary substantially across splits.

8 | INTERPRETING RRR

Finally, we raise several issues related to the interpretation of RRR estimates, useful follow-up analyses, and conclusions that can or cannot be drawn from them.

8.1 | Modeling assumptions

Although we have derived RRR as the maximum likelihood estimator for a low-rank linear communication model corrupted by Gaussian noise, we wish to emphasize that the estimator itself makes no assumption of Gaussianity. Another equally valid formulation is to say that the RRR estimator provides the low-rank weight estimate W that achieves the minimum mean squared error in linearly predicting outputs $\{\mathbf{y}_t\}$ from inputs $\{\mathbf{x}_t\}$. This holds for any noise distribution corrupting $\{\mathbf{y}_t\}$, so we are free to apply RRR even when the noise is clearly non-Gaussian, as in the case of integer spike counts. (Technically, RRR no longer corresponds to a maximum likelihood estimator in such settings, and we might achieve lower asymptotic performance using a different estimator, but RRR still achieves lowest possible MSE for the training data).

Additionally, we can apply RRR even when the putative mapping from input to output responses is nonlinear. In this case, RRR cannot hope to recover the true (nonlinear) communication channel, but it provides the best linear approximation in a least squares sense.

8.2 | Follow-up analyses

One useful direction for analyzing a communication channel inferred using RRR is to examine its relationship to dimensions of neural activity that encode particular task covariates, as estimated with standard encoding or decoding methods. In sensory areas, one might examine whether communication dimensions from the input population—defined by the column space of W—are aligned with the dimensions encoding sensory stimuli. Indeed, recent studies have shown that the primary communication axes are often aligned with dimensions that encode task-relevant variables in a context-dependent manner, suggesting that task-relevant information is effectively transmitted from input to output regions [5, 26].

Conversely, previous work in motor cortex has argued that neural activity during the "preparatory" period (i.e., the period before the onset of movement) is orthogonal to the dimensions that drive motor signals [27]. This suggests that preparatory activity is not aligned with communication dimensions, although—to the best of our knowledge—this phenomenon has not yet been examined using RRR specifically.

8.3 | Correlation, not causation

Finally, we feel it is important to emphasize that RRR is a statistical method that relies on correlations between $\{\mathbf{x}_t\}$ and $\{\mathbf{y}_t\}$. It identifies *statistical dependencies* between brain regions, but there is no guarantee that these dependencies reflect causal interactions [28, 29].

We would suggest that the term "communication subspace" is itself misleading insofar as it implies measurement of (causal) signal propagation between brain regions. On the contrary, it is entirely possible that the RRR estimate reflects correlations due to shared common input from a third region, or from signals propagating in the reverse direction from output to input. This caveat is important to consider for any analysis of neural communication using RRR.

9 | DISCUSSION

Reduced-rank regression (RRR) provides a principled way to infer a low-dimensional linear mapping between high-dimensional inputs and outputs. In this paper, we presented a detailed introduction to the RRR estimator—its motivation, mathematical origins, and its relationship to other dimensionality reduction methods—with the aim of making these tools accessible to neuroscience researchers.

We discussed several extensions of RRR, including ridge regularization and models with non-spherical noise, which can improve estimation in regimes with low signal-to-noise ratio or correlated or unequal-variance noise. We then introduced new metrics for quantifying both the strength of communication and the alignment between communication channels and population activity spaces. Finally, we reviewed practical considerations for applying RRR to neural data and interpreting its results.

We emphasize that—despite its popularity for identifying "communication subspaces"—RRR is still fundamentally a regression method. It relies on correlation, not causation. As such, it inherits the interpretational pitfalls of regression

more broadly and cannot, on its own, establish the true direction of signal flow within neural circuits.

Beyond this central limitation, several additional shortcomings of RRR point toward promising directions for future work. First, RRR is inherently a *pairwise* method, modeling communication from one input region to one output region. This is a substantial simplification in real neural systems, where most regions both send and receive signals from multiple partners. Extending RRR-style communication models to handle multi-region, low-dimensional interactions represents an important challenge [30]. Second, RRR assumes a linear mapping between regions. Nonlinear approaches, including deep neural networks [31, 32, 33] and nonlinear latent-variable models [34, 35, 36], have shown considerable promise as flexible alternatives. Third, future work could incorporate cell-type structure into RRR-like frameworks, allowing communication channels to be resolved across genetically or molecularly defined subpopulations. Recent advances in cell-type–aware dynamical systems models [37, 38, 39] provide a promising avenue for such an approach.

Despite these limitations, RRR remains a powerful and practical tool for studying interactions between neural populations. It is simple, interpretable, computationally efficient, and often captures a substantial proportion of shared variance between brain regions. More widespread application of RRR and its emerging extensions stands to significantly advance our understanding of brain-wide neural activity and the distributed computations that support cognition and behavior.

Software Implementation

We have provided software implementations in both Python and MATLAB for ridge-RRR, full-covariance RRR, and alignment indices at `https://github.com/bichanw/RRR`.

Acknowledgments

This work was supported by grants from the Simons Collaboration on the Global Brain (SCGB AWD543027), the NIH BRAIN initiative (9R01DA056404-04), a U19 NIH-NINDS BRAIN Initiative Award (U19NS104648, U19NS123716), and an NIH R01 (1R01EY033064-01A1).

References

[1] Izenman AJ. Reduced-rank regression for the multivariate linear model. Journal of multivariate analysis 1975;5(2):248–264.

[2] Semedo JD, Zandvakili A, Machens CK, Byron MY, Kohn A. Cortical areas interact through a communication subspace. Neuron 2019;102(1):249–259.

[3] Srinath R, Ruff DA, Cohen MR. Attention improves information flow between neuronal populations without changing the communication subspace. Current Biology 2021;31(23):5299–5313.

[4] Steinmetz NA, Zatka-Haas P, Carandini M, Harris KD. Distributed coding of choice, action and engagement across the mouse brain. Nature 2019 Nov;`https://doi.org/10.1038/s41586-019-1787-x`.

[5] MacDowell CJ, Libby A, Jahn CI, Tafazoli S, Ardalan A, Buschman TJ. Multiplexed subspaces route neural activity across brain-wide networks. Nature Communications 2025 Apr;16(1):3359. `https://doi.org/10.1038/s41467-025-58698-2`.

[6] Reinsel GC, Velu RP. Multivariate reduced-rank regression: theory and applications. Springer New York; 1998.

[7] Vounou M, Nichols TE, Montana G, Initiative ADN, et al. Discovering genetic associations with high-dimensional neuroimaging phenotypes: A sparse reduced-rank regression approach. Neuroimage 2010;53(3):1147–1159.

[8] Kobak D, Brendel W, Constantinidis C, Feierstein CE, Kepecs A, Mainen ZF, et al. Demixed principal component analysis of neural population data. Elife 2016;5:e10989.

[9] Mante V, Sussillo D, Shenoy KV, Newsome WT. Context-dependent computation by recurrent dynamics in prefrontal cortex. Nature 2013;503(7474):78–84.

[10] Aoi M, Pillow JW. Model-based targeted dimensionality reduction for neuronal population data. In: Bengio S, Wallach H, Larochelle H, Grauman K, Cesa-Bianchi N, Garnett R, editors. Advances in Neural Information Processing Systems 31 Curran Associates, Inc.; 2018.p. 6689–6698.

[11] Chapin JK, Nicolelis MA. Principal component analysis of neuronal ensemble activity reveals multidimensional somatosensory representations. Journal of neuroscience methods 1999;94(1):121–140.

[12] Cunningham JP, Yu BM. Dimensionality reduction for large-scale neural recordings. Nature Neuroscience 2014 Nov;17(11):1500–1509.

[13] Gallego JA, Perich MG, Miller LE, Solla SA. Neural manifolds for the control of movement. Neuron 2017;94(5):978–984.

[14] Stringer C, Pachitariu M, Steinmetz N, Carandini M, Harris KD. High-dimensional geometry of population responses in visual cortex. Nature 2019;p. 1.

[15] Pospisil DA, Pillow JW. Revisiting the high-dimensional geometry of population responses in visual cortex. bioRxiv 2024;https://www.biorxiv.org/content/early/2024/02/21/2024.02.16.580726.

[16] Shlens J, A Tutorial on Principal Component Analysis; 2014. https://arxiv.org/abs/1404.1100.

[17] Petersen KB, Pedersen MS. The Matrix Cookbook. Technical University of Denmark 2008;7(15):510. http://www2.imm.dtu.dk/pubdb/p.php?3274.

[18] Hotelling H. Relations between two sets of variates. Biometrika 1936;28:321–377.

[19] Semedo JD, Jasper AI, Zandvakili A, Krishna A, Aschner A, Machens CK, et al. Feedforward and feedback interactions between visual cortical areas use different population activity patterns. Nature Communications 2022 Dec;13(1):1099. https://www.nature.com/articles/s41467-022-28552-w.

[20] Cunningham JP, Ghahramani Z. Linear Dimensionality Reduction: Survey, Insights, and Generalizations. Journal of Machine Learning Research 2015;16(89):2859–2900. http://jmlr.org/papers/v16/cunningham15a.html.

[21] Mukherjee A, Zhu J. Reduced rank ridge regression and its kernel extensions. Statistical Analysis and Data Mining: The ASA Data Science Journal 2011 Dec;4(6):612–622. https://doi.org/10.1002/sam.10138, publisher: John Wiley & Sons, Ltd.

[22] Bishop CM. Pattern recognition and machine learning. Springer New York:; 2006.

[23] Murphy KP. Probabilistic Machine Learning: An introduction. MIT Press; 2022. probml.ai.

[24] Hoerl AE, Kennard RW. Ridge regression: Biased estimation for nonorthogonal problems. Technometrics 1970;12(1):55–67.

[25] Gao P, Trautmann E, Yu B, Santhanam G, Ryu S, Shenoy K, et al. A theory of multineuronal dimensionality, dynamics and measurement. bioRxiv 2017 Nov;http://biorxiv.org/lookup/doi/10.1101/214262.

[26] Barbosa J, Proville R, Rodgers CC, DeWeese MR, Ostojic S, Boubenec Y. Early selection of task-relevant features through population gating. Nature Communications 2023 Oct;14(1):6837. https://doi.org/10.1038/s41467-023-42519-5.

[27] Kaufman MT, Churchland MM, Ryu SI, Shenoy KV. Cortical activity in the null space: permitting preparation without movement. Nature Neuroscience 2014 Mar;17(3):440–448. https://doi.org/10.1038/nn.3643.

[28] Keeley SL, Zoltowski DM, Aoi MC, Pillow JW. Modeling statistical dependencies in multi-region spike train data. Current Opinion in Neurobiology 2020;65:194 – 202. http://www.sciencedirect.com/science/article/pii/S0959438820301720, whole-brain interactions between neural circuits.

[29] Semedo JD, Gokcen E, Machens CK, Kohn A, Yu BM. Statistical methods for dissecting interactions between brain areas. Current Opinion in Neurobiology 2020;65:59–69. https://www.sciencedirect.com/science/article/pii/S0959438820301367, whole-brain interactions between neural circuits.

[30] Gokcen E, Jasper A, Xu A, Kohn A, Machens CK, Yu BM. Uncovering motifs of concurrent signaling across multiple neuronal populations. In: Advances in Neural Information Processing Systems, vol. 36 Curran Associates, Inc.; 2023. p. 34711–34722. https://proceedings.neurips.cc/paper_files/paper/2023/file/6cf7a37e761f55b642cf0939b4c64bb8-Paper-Conference.pdf.

[31] Koukuntla S, Julian JB, Kaminsky JC, Schottdorf M, Tank DW, Brody CD, et al., Unsupervised discovery of the shared and private geometry in multi-view data. arXiv; 2024. http://arxiv.org/abs/2408.12091, arXiv:2408.12091 [cs, q-bio].

[32] Liu B, Sacks J, Golub MD, Accurate identification of communication between multiple interacting neural populations. arXiv; 2025. http://arxiv.org/abs/2506.19094, arXiv:2506.19094 [q-bio].

[33] Perich MG, Rajan K. Rethinking brain-wide interactions through multi-region 'network of networks' models. Current Opinion in Neurobiology 2020 Dec;65:146–151. http://www.sciencedirect.com/science/article/pii/S0959438820301707.

[34] Dowling M, Savin C. Nonlinear multiregion neural dynamics with parametric impulse response communication channels. In: The Thirteenth International Conference on Learning Representations; 2025. https://openreview.net/forum?id=LbgIZpSUCe.

[35] Keeley S, Aoi M, Yu Y, Smith S, Pillow JW. Identifying signal and noise structure in neural population activity with Gaussian process factor models. In: Advances in Neural Information Processing Systems, vol. 33 Curran Associates, Inc.; 2020. p. 13795–13805. https://proceedings.neurips.cc/paper_files/paper/2020/hash/9eed867b73ab1eab60583c9d4a789b1b-Abstract.html.

[36] Glaser J, Whiteway M, Cunningham JP, Paninski L, Linderman S. Recurrent Switching Dynamical Systems Models for Multiple Interacting Neural Populations. In: Advances in Neural Information Processing Systems, vol. 33 Curran Associates, Inc.; 2020. p. 14867–14878. https://proceedings.neurips.cc/paper/2020/hash/aa1f5f73327ba40d47ebce155e785aaf-Abstract.html.

[37] O'Shea DJ, Duncker L, Goo W, Sun X, Vyas S, Trautmann EM, et al. Direct neural perturbations reveal a dynamical mechanism for robust computation. bioRxiv 2022;p. 2022–12.

[38] Javadzadeh M, Schimel M, Hofer SB, Ahmadian Y, Hennequin G. Dynamic consensus-building between neocortical areas via long-range connections. bioRxiv 2024;https://www.biorxiv.org/content/early/2024/12/04/2024.11.27.625691.

[39] Jha A, Gupta D, Brody CD, Pillow JW. Disentangling the Roles of Distinct Cell Classes with Cell-Type Dynamical Systems. In: The Thirty-eighth Annual Conference on Neural Information Processing Systems; 2024. https://openreview.net/forum?id=9sP4oejtjB.

A | IDENTIFIABILITY OF THE LOW-RANK FACTORIZATION OF MODEL WEIGHTS

Note that a factorized low-rank parameterization of a weight matrix, $W = UV^\top$, suffers from the limitation that the factors U and V are not unique. For example, we could scale U by an arbitrary nonzero scalar a and scale V by $1/a$ without changing the product $UV^\top$. More generally, U and V can be transformed by an arbitrary invertible matrix. To see this, consider an invertible matrix L, and let $\tilde{U} = (UL)$ and $\tilde{V} = (VL^{-\top})$. Then $\tilde{U}\tilde{V}^\top = ULL^{-1}V^\top = UV^\top$, showing that the two matrix products are identical.

To resolve this ambiguity, we can require that V be semi-orthogonal, so $V^\top V = I$, and require U to have orthogonal columns sorted by decreasing L_2 norm. We can obtain such a solution using the singular value decomposition (SVD) (Sec. 2.3).

First, given an arbitrary $\hat{U}_0$ and $\hat{V}_0$ such that the weights $\hat{W} = \hat{U}_0\hat{V}_0{}^\top$ is a rank r matrix, perform SVD on $\hat{W}$ to obtain $\hat{W} = USV^\top$. Then, set $\hat{V}_1$ to the first r columns of V, and $\hat{U}_1$ to the first r columns of (US). Clearly, the resulting matrices satisfy $\hat{W} = \hat{U}_1\hat{V}_1{}^\top$, since $\hat{W}$ is rank r by definition, $\hat{V}_1$ is semi-orthogonal since its columns come from V, and $\hat{U}_1$ inherits the orthogonal columns of U, scaled by the corresponding singular value in S. Thus, for any solution $\hat{W}$ to the RRR problem, we can always transform it using SVD to obtain a unique set of factors $\hat{U}$ and $\hat{V}$ up to column sign flips.

B | DERIVATION OF RRR FOR ARBITRARY RANK

The derivation of RRR for arbitrary rank is largely similar to the rank-1 case in Sec. 3.2. Instead of optimizing with respect to vectors **u** and **v**, we optimize with respect to matrices U and V. The loss function can be written as:

$$||Y - XUV^\top||_F^2 = \mathrm{Tr}\big[(Y - XUV^\top)^\top (Y - XUV^\top)\big] \tag{54}$$

We can assume without loss of generality that V is semi-orthogonal (meaning its columns consist of orthogonal unit vectors), so $V^\top V = I_r$. Adding Lagrange multipliers for the constraint, the loss function becomes:

$$L(U, V) = \mathrm{Tr}\big[(Y - XUV^\top)^\top (Y - XUV^\top)\big] + \mathrm{Tr}[\Lambda(V^\top V - I_r)], \tag{55}$$

$$= \mathrm{Tr}[Y^\top Y] - 2\mathrm{Tr}[VU^\top X^\top Y] + \mathrm{Tr}[U^\top X^\top XU] + \mathrm{Tr}[\Lambda(V^\top V - I_r)]. \tag{56}$$

where Λ is an $r \times r$ symmetric matrix of Lagrange multipliers for the constraint that $V^\top V = I$.

To find the optimum, we first differentiate with respect to U and V, which yields:

$$\frac{\partial L}{\partial U} = -2X^\top YV + 2X^\top XU \tag{57}$$

$$\frac{\partial L}{\partial V} = -2Y^\top XU + 2V\Lambda. \tag{58}$$

Setting (eq. 57) to zero leads to a solution for U in terms of V:

$$U = (X^\top X)^{-1} X^\top YV = \hat{W}_{\mathrm{LS}} V, \tag{59}$$

where $\hat{W}_{\mathrm{LS}} = (X^\top X)^{-1} X^\top Y$ denotes the classic least squares regression solution. We can then substitute for U in

(eq. 58) and set it to zero, which gives:

$$Y^\top X \hat{W}_{\text{LS}} V = V\Lambda. \tag{60}$$

While our Lagrange multiplier formulation is defined for symmetric Λ, it can be shown that any solution $\hat{\Lambda}$ has an equivalent diagonal form: For every symmetric Λ, there exists an eigendecomposition $\Lambda = U_\Lambda S_\Lambda U_\Lambda{}^\top$, so (eq. 60) can be transformed into:

$$Y^\top X \hat{W}_{\text{LS}} V = V U_\Lambda S_\Lambda U_\Lambda{}^\top \tag{61}$$

Multiplying both sides on the right by U_Λ gives:

$$Y^\top X \hat{W}_{\text{LS}} (V U_\Lambda) = (V U_\Lambda) S_\Lambda \tag{62}$$

Since both V and U_Λ are orthonormal, (VU_Λ) is still orthonormal, and S_Λ is a diagonal matrix, which will be our new Lagrange multiplier.

When searching in the space where Λ is diagonal, (eq. 60) can be seen as an eigenvector equation, where the optimal solution is to set V to the top eigenvectors of $Y^\top X \hat{W}_{\text{LS}}$, and Λ is a diagonal matrix with corresponding eigenvalues along the diagonal. A rigorous proof is not trivial, and we dedicate another appendix section for it (see Appendix D).

As the final result, we have

$$\hat{V} = \text{eig}(Y^\top X (X^\top X)^{-1} X^\top Y) = \text{eig}(Y^\top X \hat{W}_{\text{LS}}) \tag{63}$$

$$\hat{U} = \hat{W}_{\text{LS}} \hat{V} \tag{64}$$

$$\hat{W} = \hat{W}_{\text{LS}} \hat{V} \hat{V}^\top \tag{65}$$

This is equivalent to the commonly used RRR estimator in Sec. 3.1. We can write out the PCA solution:

$$\begin{aligned} \hat{V}_{\text{RRR}} &= \text{eig}[\hat{W}_{\text{LS}}{}^\top (X^\top X) \hat{W}_{\text{LS}}] \\ &= \text{eig}[Y^\top X (X^\top X)^{-1} (X^\top X) \hat{W}_{\text{LS}}] \\ &= \text{eig}[Y^\top X \hat{W}_{\text{LS}}] = \hat{V} \end{aligned}$$

C | DERIVATIONS OF EXTENSIONS OF RRR

C.1 | Ridge-RRR

The loss function for RRR with ridge regularization is:

$$L_{\text{ridge}}(U, V) = \text{Tr}\big[(Y - XUV^\top)^\top (Y - XUV^\top)\big] + \lambda_{\text{ridge}} \text{Tr}[U^\top U], \tag{66}$$

Following similar steps to the derivation of ordinary RRR above, we obtain:

$$\frac{\partial L_{\text{ridge}}}{\partial U} = -2X^\top Y V + 2(X^\top X + \lambda_{\text{ridge}} I) U \tag{67}$$

$$\frac{\partial L_{\text{ridge}}}{\partial V} = -2Y^\top XU + 2V\Lambda, \tag{68}$$

and thus

$$U = (X^\top X + \lambda_{\text{ridge}} I)^{-1} X^\top YV = \hat{W}_{\text{ridge}} V, \tag{69}$$

where $\hat{W}_{\text{ridge}} = (X^\top X + \lambda_{\text{ridge}} I)^{-1} X^\top Y$ is the classic ridge regression estimate for W. Following the same steps of the above derivation, we obtain:

$$Y^\top X \hat{W}_{\text{ridge}} V = V\Lambda, \tag{70}$$

and the final solution is:

$$\hat{V}_{\text{RRRR}} = \text{eig}(Y^\top X (X^\top X + \lambda_{\text{ridge}} I)^{-1} X^\top Y) = \text{eig}(Y^\top X \hat{W}_{\text{ridge}}) \tag{71}$$

$$\hat{U}_{\text{RRRR}} = \hat{W}_{\text{ridge}} \hat{V}_{\text{RRRR}}. \tag{72}$$

C.2 | Non-spherical noise

The loss function for RRR with non-spherical noise is:

$$L_{\text{fcRRR}}(U, V) = \text{Tr}\big[(Y - XUV^\top)\Sigma^{-1}(Y - XUV^\top)^\top\big], \tag{73}$$

To optimize for U and V, we will make the following change of variables:

$$\tilde{V} = \Sigma^{-\frac{1}{2}} V, \tag{74}$$

where $\Sigma^{-\frac{1}{2}}$ denotes the inverse matrix square root of Σ. We will then assume that $\tilde{V}^\top \tilde{V} = I$.

Substituting into the above loss function gives us:

$$L_{\text{fcRRR}}(U, \tilde{V}) = \text{Tr}[Y\Sigma^{-1}Y^\top] - 2\text{Tr}[XU\tilde{V}^\top \Sigma^{-\frac{1}{2}} Y^\top] + \text{Tr}[XUU^\top X^\top], \tag{75}$$

where the third term follows from the fact that $V^\top \Sigma^{-1} V = \tilde{V}^\top \tilde{V} = I$.

If we now add the Lagrange multiplier term to enforce the semi-orthogonality of $\tilde{V}$ and take derivatives, we obtain:

$$\frac{\partial L_{\text{fcRRR}}}{\partial U} = -2X^\top Y \Sigma^{-\frac{1}{2}} \tilde{V} + 2(X^\top X)U \tag{76}$$

$$\frac{\partial L_{\text{fcRRR}}}{\partial \tilde{V}} = -2\Sigma^{-\frac{1}{2}} Y^\top XU + 2\tilde{V}\Lambda. \tag{77}$$

Setting the first partial derivative to zero leads to:

$$U = (X^\top X)^{-1} X^\top Y \Sigma^{-\frac{1}{2}} \tilde{V}, \tag{78}$$

and substituting into the second yields

$$\Sigma^{-\frac{1}{2}} Y^\top X (X^\top X)^{-1} X^\top Y \Sigma^{-\frac{1}{2}} \tilde{V} = \tilde{V} \Lambda. \tag{79}$$

This is an eigenvector equation, which implies

$$\hat{\tilde{V}} = \mathrm{eig}(\Sigma^{-\frac{1}{2}} Y^\top X (X^\top X)^{-1} X^\top Y \Sigma^{-\frac{1}{2}}) \tag{80}$$

$$\hat{U} = \left((X^\top X)^{-1} X^\top Y \Sigma^{-\frac{1}{2}}\right) \hat{\tilde{V}}. \tag{81}$$

Translating back to our original variables yields:

$$\hat{V} = \Sigma^{\frac{1}{2}} \,\mathrm{eig}(\Sigma^{-\frac{1}{2}} Y^\top X (X^\top X)^{-1} X^\top Y \Sigma^{-\frac{1}{2}}) \tag{82}$$

$$\hat{U} = \left((X^\top X)^{-1} X^\top Y \Sigma^{-\frac{1}{2}}\right) \Sigma^{-\frac{1}{2}} \hat{V}. \tag{83}$$

D | PROOF THAT THE OPTIMAL SOLUTIONS ARE EIGENVECTORS

This section is for the proof that the optimal solution for (eq. 54) is indeed (eq. 63). We adapted the proof from [6]. The goal is to prove that for the following condition:

$$\arg\min_V \mathrm{Tr}\left[(Y - X\hat{W}_{\mathrm{LS}} V V^\top)^\top (Y - X\hat{W}_{\mathrm{LS}} V V^\top)\right] \tag{84}$$

The solution is:

$$V = \text{first } r \text{ eigenvectors of } Y^\top X (X^\top X)^{-1} X^\top Y \,(\text{or equivalently } Y^\top X \hat{W}_{\mathrm{LS}}) \tag{85}$$

1. **Lemma 1** A is an $m \times m$ symmetric matrix with eigenvalues $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_m$. $X \in \mathbb{R}^{m \times r}$, $r \leq m$ is an orthogonal matrix. The maximum of $\mathrm{Tr}[X^\top A X]$ is equal to the sum of the largest r eigenvalues of A and is attained when columns of X are the first r eigenvectors of A.
   Proof Let $\mathbf{x}_i$ be the i-th column of X, and $\mathbf{x}_i = P\mathbf{c}_i$ where $P \in \mathbb{R}^{m \times m}$ is the matrix of eigenvectors of A (so $AP = P\Lambda$, $\Lambda = \mathrm{diag}(\lambda_1, \cdots, \lambda_m)$), and $\mathbf{c}_i \in \mathbb{R}^m$ is the coefficient vector. Because X is a semi-orthogonal matrix, we have:

$$\mathbf{x}_i{}^\top \mathbf{x}_i = \mathbf{c}_i{}^\top P^\top P \mathbf{c}_i = \mathbf{c}_i{}^\top \mathbf{c}_i = 1 \tag{86}$$

$$\mathbf{x}_i{}^\top \mathbf{x}_j = \mathbf{c}_i{}^\top P^\top P \mathbf{c}_j = \mathbf{c}_i{}^\top \mathbf{c}_j = 0 \tag{87}$$

   Then we have the following:

$$\mathrm{Tr}[X^\top A X] = \sum_{i=1}^{r} \mathbf{x}_i{}^\top A \mathbf{x}_i = \sum_{i=1}^{r} \mathbf{c}_i{}^\top P^\top A P \mathbf{c}_i = \sum_{i=1}^{r} \mathbf{c}_i{}^\top \Lambda \mathbf{c}_i = \sum_{j=1}^{m} \lambda_j \left(\sum_{i=1}^{r} c_{ij}^2\right) \tag{88}$$

   We know $\sum_{j=1}^{m} \sum_{i=1}^{r} c_{ij}^2 = \sum_{i=1}^{r} \mathbf{c}_i{}^\top \mathbf{c}_i = r$, to maximize (eq. 88), we should set $c_{ij} = 0$ for $j > r$, and $\sum_{i=1}^{r} c_{ij}^2 = 1$ for

$j \leq r$. So the maximum of $\text{Tr}[X^\top AX]$ is $\sum_{i=1}^{r} \lambda_i$, and is attained when the columns of X are the first r eigenvectors of A.

2. **Theorem 1** $S \in \mathbb{R}^{m\times n}$ and is of rank m. $\text{Tr}\left[(S-P)(S-P)^\top\right]$ is minimum among matrices P of lower rank at rank r, when $P = MM^\top S$ where $M \in \mathbb{R}^{m\times r}$ and the columns of M are the first r eigenvectors of $SS^\top$.
Proof Let $P = MN$, $M \in \mathbb{R}^{m\times r}$, $N \in \mathbb{R}^{r\times n}$; for a given M, it's easy to see that $N = (M^\top M)^{-1}M^\top S$, then we have:

$$\text{Tr}\left[(S-P)(S-P)^\top\right] = \text{Tr}\left[SS^\top\left(I_m - MM^\top\right)\right] = \text{Tr}\left[SS^\top\right] - \text{Tr}\left[M^\top SS^\top M\right] \tag{89}$$

which based on Lemma 1 shows that the optimal M is indeed the first r eigenvectors of $SS^\top$.

3. **Theorem 2** We want to prove that to minimize:

$$\text{Tr}\left[(Y - X\hat{W}_{\text{LS}}VV^\top)^\top(Y - X\hat{W}_{\text{LS}}VV^\top)\right], \tag{90}$$

the solution is:

$$V = \text{first } r \text{ eigenvectors of } Y^\top X(X^\top X)^{-1}X^\top Y \tag{91}$$

The proof is achieved by setting $\tilde{S} = Y^\top X(X^\top X)^{-\frac{1}{2}}$ and $\tilde{P} = VV^\top\tilde{S}$, and writing (eq. 90) as:

$$\begin{aligned}
&\text{Tr}\left[\left(Y - X(X^\top X)^{-1}X^\top YVV^\top\right)^\top\left(Y - X(X^\top X)^{-1}X^\top YVV^\top\right)\right] && (92)\\
=&\text{Tr}\left[Y^\top Y - 2Y^\top X(X^\top X)^{-1}X^\top YVV^\top + VV^\top Y^\top X(X^\top X)^{-1}X^\top YVV^\top\right] && (93)\\
=&\text{Tr}\left[Y^\top Y - Y^\top X(X^\top X)^{-1}X^\top Y\right] && (94)\\
&+\text{Tr}\left[\left(Y^\top X(X^\top X)^{-\frac{1}{2}} - VV^\top Y^\top X(X^\top X)^{-\frac{1}{2}}\right)\left(Y^\top X(X^\top X)^{-\frac{1}{2}} - VV^\top Y^\top X(X^\top X)^{-\frac{1}{2}}\right)^\top\right] && (95)\\
=&\text{const} + \text{Tr}\left[(\tilde{S}-\tilde{P})(\tilde{S}-\tilde{P})^\top\right] && (96)
\end{aligned}$$

The first term is constant, and based on Theorem 1, the second term is minimized when V are the top eigenvectors of $\tilde{S}\tilde{S}^\top = Y^\top X(X^\top X)^{-1}X^\top Y$.

E | SIMULATION PARAMETERS

The code used to reproduce the simulation results shown in Fig. 4A and Fig. 5A is available at `https://github.com/bichanw/RRR`. In both panels, we used the parameters $m = 50$, $n = 50$, $n_{\text{sim}} = 50$, and $r = 2$.

For the ridge-regression simulations, we set $\sigma = 50$; for the full-covariance RRR simulations, we used $T = 1000$.

In each simulation, the elements of $X \in \mathbb{R}^{T\times m}$, $U \in \mathbb{R}^{m\times r}$, and $V \in \mathbb{R}^{n\times r}$ were drawn independently from a standard normal distribution. The response matrix $Y \in \mathbb{R}^{T\times n}$ was then generated from a Gaussian distribution with mean $XUV^\top$ and a covariance structure defined by the respective simulation condition.