Disentangling communication across populations of neurons

Submitted in partial fulfillment of the requirements for

the degree of

Doctor of Philosophy

in

Electrical and Computer Engineering

Evren A. Gokcen

B.S., Electrical Engineering, California Institute of Technology

M.S., Electrical and Computer Engineering, Carnegie Mellon University

Carnegie Mellon University
Pittsburgh, PA

April 2023

Thesis committee:

Byron M. Yu, chair

Adam Kohn

Christian K. Machens

Pulkit Grover

Nuo Li

Acknowledgements

A Ph.D. is an inherently lonely pursuit. Yet it takes a village to write a thesis, and I have that village to thank for getting me over the finish line.

First, thanks to Byron, Adam, and Christian for their guidance and especially their trust over these many years. The core idea of this thesis occurred to me one Winter Break, a year-and-a-half into my Ph.D. I returned from break working ardently on a completely new project, and to my continued amazement, Byron, Adam, and Christian let me keep going. For better or for worse, here are the results of that trust.

Thanks to my remaining committee members, Pulkit Grover and Nuo Li. Their discussions helped shape my thinking for this dissertation and beyond.

My work throughout my Ph.D. has been generously supported with a CIT Bertucci Fellowship, CMU BrainHub Presidential Fellowship, CIT Dowd Fellowship, ECE Benjamin Garver Lamme/Westinghouse Graduate Fellowship, as well as funding from Byron, Adam, and Christian through the Simons Foundation, NIH, and NSF.

Thanks to the many members of the Yu, Kohn, Machens, Batista, Chase, and Smith labs, who have provided valuable comments, both critical and snarky, on this work over the years. Special thanks to João for being a mentor and a role model, and to Anna, Aravind, and Joana for tolerating my pixels nearly every week for the last 4 to 6 years.

Thanks to many dear friends in the Explorer's Club of Pittsburgh for the long carpools, Alpine Starts, and icy bushwhacks.

Thanks to Kevin, Pedro, and Valentin for a decade of misadventures.

Thanks to Ashley for the punny snacks, baked goods, and companionship that fueled this final sprint.

And thanks *so much* to my family. To Ajda and Yasemin, with whom I can be my goofiest self. To Mom, for the steady Stauf's supply and the uniquely motherly power to motivate me to finish this dissertation. And to Dad, for teaching me to find my tree.

Abstract

Technological advances now allow us to record from large populations of neurons across multiple brain areas. These recordings may illuminate how communication between areas contributes to brain function, yet a substantial barrier remains: how do we disentangle the concurrent, bidirectional flow of signals between populations of neurons? Here we develop a dimensionality reduction framework, delayed latents across groups (DLAG), that disentangles signals relayed in each direction, identifies how these signals are represented by each population, and characterizes how they evolve within and across trials. We systematically validate DLAG in simulation, demonstrating that it performs well over a wide range of simulated conditions, including synthetic datasets similar in scale to current neurophysiological recordings. We also demonstrate its robustness to mild deviations from its model assumptions. Then we use DLAG to study bidirectional communication between neuronal populations in (1) visual areas V1 and V2, recorded simultaneously in anesthetized macaques, and (2) V1 and V4, recorded simultaneously in an awake, passively fixating macaque. In both studies, DLAG revealed signatures of bidirectional yet selective communication. To support the interpretation of DLAG models fit to these neural recordings, we develop descriptive and inferential statistics. Finally, we extend the DLAG framework to include an arbitrary number of neuronal populations (that is, three or more), and validate the extended method with simulated neural activity. This work lays a foundation for dissecting the intricate flow of signals across populations of neurons, and how this signaling contributes to cortical computation.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Simultaneous recordings from large populations of neurons across multiple brain areas are growing in availability[1–4]. These recordings present opportunities to illuminate how inter-areal communication enables brain function[5], but they also present substantial conceptual and statistical challenges. Brain areas involved in sensory[6–9], cognitive[10], and motor functions[11] are often reciprocally connected: signals are relayed not only from one area to the next, but bidirectionally, and likely concurrently. The raw recordings, however, provide only a tangled view of this concurrent communication (Fig. 1.1, top): individual neurons simultaneously reflect an area's inputs, outputs, and ongoing internal computations[12].

Determining the flow of signals between brain areas is therefore a nontrivial task. To dissect the direction of signal flow, one can leverage the fact that inter-areal communication is not instantaneous. The physiological properties of axons and synapses introduce delays in signal transmission. These delays provide a working definition of signal flow: the appearance of a signal first in area A, and later in area B, is consistent with signal flow from A to B (though this apparent flow could be due to common input from a third area).

Adopting this conception, several inter-areal studies have compared the timing of the onset of neural responses[13–15] or of the emergence of selectivity attributable to top-down processes[16–20] across areas following the presentation of a stimulus. Other studies, leveraging simultaneous recordings, have measured temporal delays between two areas through pairwise spiking correlations[21–26] and information-theoretic measures[27]. Similarly, inter-areal phase delays of local field potentials (LFPs) have been measured[28–31]. These timing-based approaches have significantly advanced our understanding of how signals propagate across brain areas. However, because these approaches focus largely on pairs of neurons or aggregate measures of neural activity, much remains unknown about how neuronal populations coordinate their activity to accomplish inter-areal signaling.

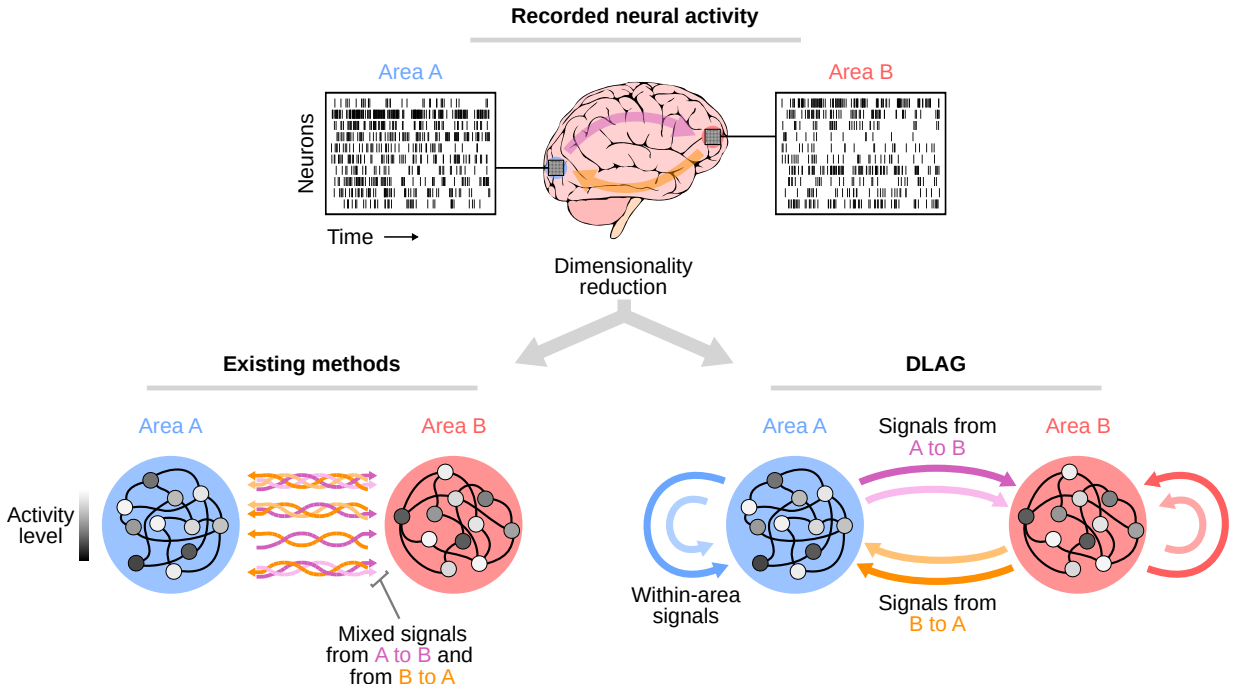

Figure 1.1: Disentangling the flow of signals between populations of neurons. Top: Recorded neural activity provides only a tangled view of the bidirectional, concurrent interactions between brain areas (illustrated by the thick translucent arrows; magenta: signals directed from area A to area B; orange: signals directed from area B to area A). Bottom left: Existing dimensionality reduction methods identify correlated population activity across areas (each correlated population activity pattern is represented by a braid of multi-colored arrows; four different activity patterns are shown). Each activity pattern likely reflects a mixture of signals relayed in each direction. Within each activity pattern, individual arrows represent a directed interaction; color depicts the direction of signal flow (magenta: A to B; orange: B to A), and shading (light vs. dark) distinguishes distinct signals. Bottom right: Delayed Latents Across Groups (DLAG) identifies both within- and across-area population signals (indicated by color and source/target of each arrow; blue: within-A; red: within-B; magenta/orange: across-area). Importantly, DLAG disentangles signals relayed in each direction. The color of each arrow depicts the direction of signal flow (magenta: A to B; orange: B to A) associated with a population activity pattern, and shading (light vs. dark) distinguishes distinct signals.

To characterize inter-areal signal flow at the level of neuronal populations is a challenging high-dimensional problem. Dimensionality reduction techniques capable of identifying low-dimensional latent variables that describe activity shared by two or more recorded areas are thus increasingly used[32–34]. These techniques have driven new proposals for population-level mechanisms of gating between motor cortex output and muscle movement[35,36]; selective communication between cortical areas[37,38]; enhanced communication of stimulus information with attention[39,40]; and the robustness of local computations to perturbations upstream[41,42].

The relationship between the correlated activity across areas identified in these studies and the flow of inter-areal signals, however, remains unclear. Specifically, does the correlated activity across areas reflect the flow of activity from area A to B, from B to A, or in both directions concurrently? If communica-

tion were to occur in one direction at a time, then existing dimensionality reduction methods could, in principle, identify the direction of population-level signal flow. If two areas were to communicate in both directions concurrently, however, then existing methods would only identify the dominant direction of signal flow[43]. Disentangling the concurrent flow of signals between populations remains a substantial barrier in neuroscience (Fig. 1.1, bottom left).

In this dissertation, we develop a novel dimensionality reduction framework: delayed latents across groups, or DLAG (Fig. 1.1, bottom right). DLAG disentangles signals relayed in each direction, identifies how these signals are represented by each population, and characterizes how they evolve over time within and across trials. We begin in Chapter 2 with a technical overview of dimensionality approaches, particularly those used to study interactions between neuronal populations. Then in Chapter 3 we introduce the DLAG model and its accompanying fitting and model selection procedures. We further discuss DLAG's interpretation as a low-rank decomposition of the covariance matrix of a time series.

In Chapter 4, we systematically validate DLAG in simulation. We first demonstrate that DLAG performs well on synthetic datasets similar in scale to state-of-the-art neurophysiological recordings from multiple brain areas. Then, we consider additional datasets covering a wider range of experimental conditions, and characterize both DLAG's performance and runtime. We also consider more challenging synthetic scenarios to demonstrate DLAG's robustness to mild deviations from its modeling assumptions. Finally, we demonstrate that DLAG disentangles concurrent signaling where existing dimensionality reduction methods cannot.

In Chapter 5, we use DLAG to study bidirectional interactions among early and midlevel visual cortical areas. We first study simultaneously recorded populations in visual areas V1 and V2 of anesthetized macaques, where DLAG revealed that V1-V2 interactions are selective and bidirectional. Then we study interactions between a second pair of brain regions, V1 and V4, in an awake animal, where DLAG uncovered differences in V1-V4 interaction that depended on the complexity of the stimulus presented. To facilitate the interpretation of DLAG models fit to these neural recordings, we develop supporting descriptive and inferential statistics. We conclude the chapter with empirical comparisons of DLAG to other statistical methods.

In Chapter 6, we motivate the problem of studying interactions between many (more than two) neuronal populations. We introduce a promising approach to the problem via group factor analysis, a static dimensionality reduction method. Then we build upon this approach to extend the DLAG framework to an arbitrary number of neuronal populations. We demonstrate the viability of the new method on simulated neural activity.

Finally, in Chapter 7 we summarize the main contributions of this dissertation and conclude with a

broader discussion. Contributions from this thesis have led to the following publications:

Chapter 2

Semedo, J. D.,* Gokcen, E.,* Machens, C. K., Kohn, A. & Yu, B. M. Statistical methods for dissecting interactions between brain areas. *Current Opinion in Neurobiology* **65**, 59–69 (2020).

Kohn, A., Jasper, A. I., Semedo, J. D., Gokcen, E., Machens, C. K. & Yu, B. M. Principles of Corticocortical Communication: Proposed Schemes and Design Considerations. *Trends in Neurosciences* **43**, 725–737 (2020).

Chapter 3–5

Gokcen, E., Jasper, A. I., Semedo, J. D., Kohn, A., Machens, C. K. & Yu, B. M. Disentangling the flow of signals between populations of neurons. *Nature Computational Science* **2**, 512-525 (2022).

Gokcen, E. egokcen/DLAG: v1.0.0. *Zenodo* `https://doi.org/10.5281/zenodo.6654831` (2022).

Chapter 6

Gokcen, E.,* Jasper, A. I.,* Xu, A., Yu, B. M., Machens, C. K., & Kohn, A. Dissecting multi-population interactions across cortical areas and layers. Cosyne Abstracts 2023.

Chapter 2

Background: Dimensionality reduction

Dimensionality reduction methods summarize high-dimensional population activity with a smaller number of latent variables. These methods have been used extensively in studies of neuronal populations in single areas[44], and are being increasingly used to study multi-area recordings[32–34]. Here, we provide a technical overview of the dimensionality reduction methods required to understand the concepts throughout the rest of this dissertation. We begin with single-area methods, and then build to methods that consider interactions between pairs of areas. Note that all of the methods presented here are static: they do not consider the flow of time. The need for multi-area dimensionality reduction methods that do consider the flow of time is a core motivation underlying the rest of this dissertation.

2.1 Methods for single areas

Suppose we simultaneously record the activity (for example, spike counts) of q neurons across N trials, given by $Y \in \mathbb{R}^{q \times N}$ (for the ensuing discussion we will assume this matrix is zero-centered: the mean activity across trials has been subtracted from each neuron). We can represent this activity geometrically in a high-dimensional population activity space, where each axis represents the activity of a single neuron, and each point in this space represents the population activity on a single trial (Fig. 2.1a).

Principal component analysis (PCA)

The goal of principal component analysis (PCA), a foundational dimensionality reduction approach, is to identify dimensions within the population space, $\mathbf{u} \in \mathbb{R}^q$, along which the variance of the neural activity is maximized. Concretely, PCA solves the following maximization problem:

$$\max \frac{\mathbf{u}^\top \hat{\Sigma} \mathbf{u}}{\mathbf{u}^\top \mathbf{u}} \tag{2.1}$$

where $\hat{\Sigma} \in \mathbb{S}^{q \times q}$ ($\mathbb{S}^{q \times q}$ is the set of $q \times q$ symmetric matrices) is the sample covariance matrix. The numerator of the objective function, $\mathbf{u}^\top \hat{\Sigma} \mathbf{u}$, is precisely the sample variance of neural activity projected onto $\mathbf{u}$, $\mathbf{u}^\top Y$. The denominator is a normalization factor that ensures solutions $\mathbf{u}$ are unit vectors.

PCA can equivalently be defined as a minimization problem, in which the goal is to identify dimensions that minimize the error in the reconstruction of neural activity from projected activity:

$$\begin{aligned} &\min \|Y - \mathbf{u}\mathbf{u}^\top Y\|_F^2 \\ &\text{s.t.} \quad \mathbf{u}^\top \mathbf{u} = 1 \end{aligned} \tag{2.2}$$

where the constraint $\mathbf{u}^\top \mathbf{u} = 1$ again ensures that solutions $\mathbf{u}$ are unit vectors.

Both of these problems can be solved for a set of $p < q$ dimensions via the eigendecomposition of the sample covariance matrix, $\hat{\Sigma}$[45]:

$$\hat{\Sigma} = UDU^\top \tag{2.3}$$

where the columns of $U \in \mathbb{R}^{q \times q}$ are eigenvectors and $D \in \mathbb{S}^{q \times q}$ is the diagonal matrix of eigenvalues. The first p eigenvectors, $U_p \in \mathbb{R}^{q \times p}$—the top p principal components—form an orthonormal basis that defines a low-dimensional subspace within the population activity space (Fig. 2.1a, blue-shaded plane). This subspace represents patterns of population activity that exhibit the greatest variance. The variance along each principal component is given by the corresponding diagonal element of D.

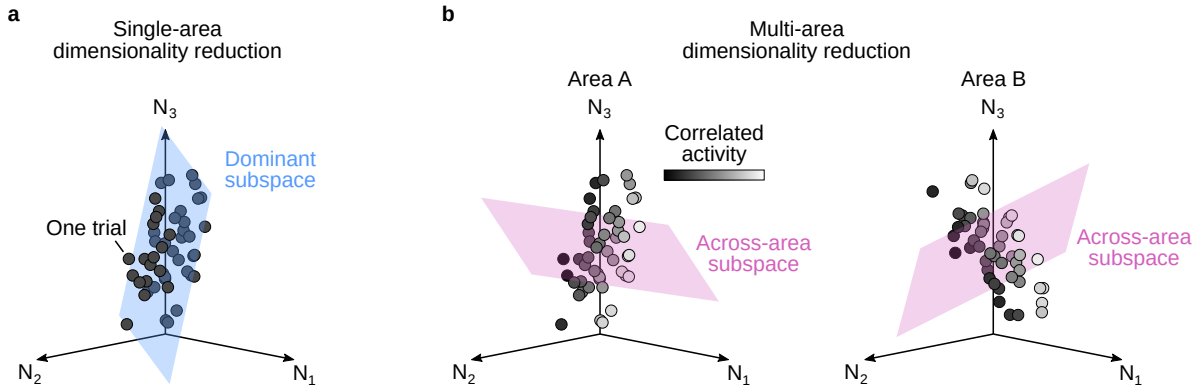

Figure 2.1: Geometric view of single- and multi-area dimensionality reduction methods. (**a**) The activity of a neural population can be represented in a population activity space, where each axis represents the activity of a single neuron (N_1, N_2, N_3). Each point in population space represents the population activity on a single trial. Single-area dimensionality reduction methods (e.g., PCA, FA) identify a low-dimensional subspace that captures the dominant trial-to-trial fluctuations shared across neurons (blue-shaded plane, "Dominant subspace"). (**b**) Multi-area dimensionality reduction methods (e.g., CCA, RRR, PLS) identify jointly a low-dimensional subspace in each recorded area (magenta-shaded plane, "Across-area subspace"). These subspaces capture trial-to-trial fluctuations that are correlated across areas (indicated by the shading of each point).

Probabilistic interpretation of PCA

PCA can alternatively be posed as a probabilistic latent variable model. Probabilistic models offer multiple advantages over their non-probabilistic counterparts, particularly the ability to define an explicit noise model, and a more principled framework for selecting the number of latent variables used (for example, through cross-validation).

Probabilistic PCA[46] defines a linear-Gaussian relationship between observed activity, $\mathbf{y} \in \mathbb{R}^q$, and latent variables, $\mathbf{x} \in \mathbb{R}^p$:

$$\mathbf{x} \sim \mathcal{N}(\mathbf{0}, I) \tag{2.4}$$

$$\mathbf{y}|\mathbf{x} \sim \mathcal{N}(C\mathbf{x} + \mathbf{d}, \sigma^2 I) \tag{2.5}$$

where $C \in \mathbb{R}^{q \times p}$, $\mathbf{d} \in \mathbb{R}^q$, and $\sigma^2 \in \mathbb{R}_{>0}$ are model parameters to be estimated from data. The loading matrix C linearly combines latent variables and maps them to observed neural activity. The parameter $\mathbf{d}$ can be thought of as the mean firing rate of each neuron. Under this model, observation noise is independent for each neuron and of the same variance σ^2. The latent variables and model parameters are estimated from the neural activity by maximizing the data likelihood, $P(\mathbf{y})$, via the Expectation-Maximization (EM) algorithm.

The columns of the matrix C, and the latent variables to which they correspond, are generally neither ordered nor mutually orthogonal. With an additional *post hoc* operation, however, they can be directly connected to the non-probabilistic PCA solution (equations (2.1), (2.2)). The singular value decomposition of C is given by $C = USV^\top$ where $U \in \mathbb{R}^{q \times p}$, $S \in \mathbb{S}^{p \times p}$, and $V \in \mathbb{R}^{p \times p}$. The columns of U are now orthonormal, ordered, and correspond to the top p principal components of the neural activity. Projections of neural activity onto to these principal components are given by $\mathbf{z} = SV^\top \mathbb{E}[\mathbf{x}|\mathbf{y}] \in \mathbb{R}^p$.

Factor analysis (FA)

Probabilistic PCA attributes equal independent variance to each neuron (i.e., an isotropic noise model). A neuronal population, however, can have wide-ranging mean firing rates and hence wide-ranging variances. The principal subspace therefore tends to be biased in the direction of neurons with high-firing rates, at the expense of dimensions that capture shared trial-to-trial fluctuations across the neuronal population[47].

A more appropriate model for neural activity can be obtained through a simple change to the proba-

bilistic PCA observation noise model (equation (2.5)), resulting in the method factor analysis (FA)[48]:

$$\mathbf{x} \sim \mathcal{N}(\mathbf{0}, I) \tag{2.6}$$

$$\mathbf{y}|\mathbf{x} \sim \mathcal{N}(C\mathbf{x} + \mathbf{d}, R) \tag{2.7}$$

where $R \in \mathbb{S}^{q \times q}$ is a general diagonal matrix. All other variables are defined as before. The independent variance attributed to each neuron may thus be different (i.e., an anisotropic noise model), thereby encouraging the latent variables to explain as much of the shared variance among neurons as possible. The same orthonormalization procedure described above, applied to an estimated FA model, produces a matrix U that defines a "dominant subspace" (Fig. 2.1a). This dominant subspace represents patterns of population activity that exhibit the greatest shared variance.

2.2 Methods for pairs of areas

We now consider the problem of studying interactions between pairs of neuronal populations, for example in different brain areas. Suppose we simultaneously record the activity of q_1 neurons in area A and q_2 neurons in area B across N trials, given by $Y_1 \in \mathbb{R}^{q_1 \times N}$ and $Y_2 \in \mathbb{R}^{q_2 \times N}$, respectively (as before, we assume each matrix is zero-centered). We can now represent this activity geometrically with two population activity spaces, one for each area (Fig. 2.1b).

In principle, the single-area methods introduced in the previous section could be used to study interactions between areas A and B. For example, one could first apply PCA or FA to area A. Then, the corresponding latent variables in area A could be regressed with the activity of each neuron in area B, leading to principal component regression (PCR) and factor regression (FR), respectively (Fig. 2.2a).

PCR and FR are advantageous over, say, multivariate linear regression because they define a more concise relationship between the two areas, thus improving interpretability. However, since PCR and FR identify latent variables using only activity within area A, it is possible for some non-dominant activity patterns in area A that are correlated with activity in area B to be left out during the dimensionality reduction stage[32].

Thus for the remainder of this chapter, we consider methods that jointly perform dimensionality reduction and relate activity across areas. Three methods in particular form a foundation for understanding the class of multi-area dimensionality reduction methods: partial least squares[49], canonical correlation analysis[50], and reduced-rank regression[51]. Each of these methods identifies latent variables with subtly different interpretations. Our primary goal here, however, is to emphasize their unifying trait: all can be seen as decompositions of the (normalized) cross-covariance matrix between areas. See Table 2.1

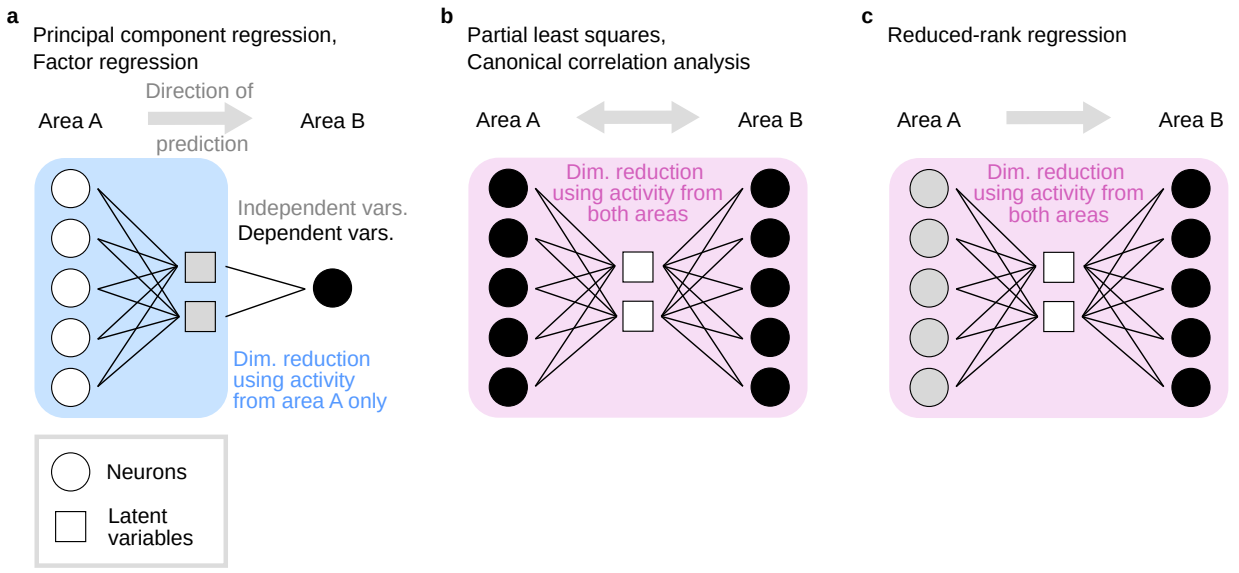

Figure 2.2: Graphical depiction of multi-area dimensionality reduction methods. (**a**) Principal component regression first identifies latent variables using activity in area A only (gray squares). Each latent variable represents a population activity pattern that explains the most variance in area A. Latent variables are then treated as independent variables and used to predict activity of a neuron in area B (the dependent variable; black circle). Factor regression is similar; however, each latent variable represents a population activity pattern that explains the most shared variance within area A. Here we show a single area B neuron being predicted, but this class of methods can be applied to a population of neurons in area B by repeating the same process for each neuron in area B. (**b**) Partial least squares uses population activity in both areas to identify latent variables. It treats populations symmetrically (i.e., all neurons are treated as dependent variables), and identifies a common set of latent variables for both areas A and B. Each latent variable represents jointly a population activity pattern in each area that describes large activity covariance across areas. Canonical correlation analysis is similar; however, each latent variable represents jointly a population activity pattern in area A and a population activity pattern in area B that are highly correlated. (**c**) Reduced-rank regression uses population activity in both areas to identify latent variables. Neurons in area A are treated as independent variables (gray circles), while neurons in area B are treated as dependent variables (black circles). Latent variables represent population activity patterns in area A that are most predictive of population activity in area B. In (a)–(c), the activity of the neurons (circles) in both areas is observed, whereas the latent variables (squares) are inferred from the observed neural activity. Boxes (blue and magenta shading) indicate which neurons are used to identify latent variables. Symbols are colored gray to indicate independent variables, and black to indicate dependent variables, when relating activity across areas.

for a comparison of these three methods, posed equivalently as maximization problems, minimization problems, singular value decompositions, and probabilistic graphical models.

Partial least squares (PLS)

The goal of partial least squares (PLS)[49] (Fig. 2.2b) is to identify pairs of dimensions, $\mathbf{u}_1 \in \mathbb{R}^{q_1}$ in area A and $\mathbf{u}_2 \in \mathbb{R}^{q_2}$ in area B, along which the covariance across areas is maximized. Concretely, PLS solves the

Table 2.1: Connecting classical multi-area dimensionality reduction methods.

Method	**Partial least squares (PLS)**	**Canonical correlation analysis (CCA)**	**Reduced-rank regression (RRR)**
Max problem	$\max \frac{\mathbf{u}_1^\top \hat{\Sigma}_{12} \mathbf{u}_2}{\sqrt{\mathbf{u}_1^\top \mathbf{u}_1}\sqrt{\mathbf{u}_2^\top \mathbf{u}_2}}$	$\max \frac{\mathbf{u}_1^\top \hat{\Sigma}_{12} \mathbf{u}_2}{\sqrt{\mathbf{u}_1^\top \hat{\Sigma}_{11} \mathbf{u}_1}\sqrt{\mathbf{u}_2^\top \hat{\Sigma}_{22} \mathbf{u}_2}}$	$\max \frac{\mathbf{u}_1^\top \hat{\Sigma}_{12} \mathbf{u}_2}{\sqrt{\mathbf{u}_1^\top \hat{\Sigma}_{11} \mathbf{u}_1}\sqrt{\mathbf{u}_2^\top \mathbf{u}_2}}$
Min problem	$\min \lVert \mathbf{u}_1^\top Y_1 - \mathbf{u}_2^\top Y_2 \rVert_F^2$ s.t. $\mathbf{u}_1^\top \mathbf{u}_1 = 1$ $\mathbf{u}_2^\top \mathbf{u}_2 = 1$	$\min \lVert \mathbf{u}_1^\top Y_1 - \mathbf{u}_2^\top Y_2 \rVert_F^2$ s.t. $\mathbf{u}_1^\top \hat{\Sigma}_{11} \mathbf{u}_1 = 1$ $\mathbf{u}_2^\top \hat{\Sigma}_{22} \mathbf{u}_2 = 1$	$\min \lVert \mathbf{u}_1^\top Y_1 - \mathbf{u}_2^\top Y_2 \rVert_F^2$ s.t. $\mathbf{u}_1^\top \hat{\Sigma}_{11} \mathbf{u}_1 = 1$ $\mathbf{u}_2^\top \mathbf{u}_2 = 1$
SVD	$\hat{\Sigma}_{12} = U_1 S U_2^\top$ $U_1^\top U_1 = I;\ U_2^\top U_2 = I$	$\hat{\Sigma}_{11}^{-\frac{1}{2}} \hat{\Sigma}_{12} \hat{\Sigma}_{22}^{-\frac{1}{2}} = V_1 S V_2^\top$ $V_1^\top V_1 = I; \quad V_2^\top V_2 = I$ $U_1 = \hat{\Sigma}_{11}^{-\frac{1}{2}} V_1; \quad U_2 = \hat{\Sigma}_{22}^{-\frac{1}{2}} V_2$ $U_1^\top \hat{\Sigma}_{11} U_1 = I;\ U_2^\top \hat{\Sigma}_{22} U_2 = I$	$\hat{\Sigma}_{11}^{-\frac{1}{2}} \hat{\Sigma}_{12} = V_1 S U_2^\top$ $V_1^\top V_1 = I; \quad U_2^\top U_2 = I$ $U_1 = \hat{\Sigma}_{11}^{-\frac{1}{2}} V_1$ $U_1^\top \hat{\Sigma}_{11} U_1 = I$
Graphical model	$\mathbf{x} \sim \mathcal{N}(\mathbf{0}, I)$ $\mathbf{y}_1 \lvert \mathbf{x} \sim \mathcal{N}(C_1 \mathbf{x} + \mathbf{d}_1, \sigma_1^2 I)$ $\mathbf{y}_2 \lvert \mathbf{x} \sim \mathcal{N}(C_2 \mathbf{x} + \mathbf{d}_2, \sigma_2^2 I)$	$\mathbf{x} \sim \mathcal{N}(\mathbf{0}, I)$ $\mathbf{y}_1 \lvert \mathbf{x} \sim \mathcal{N}(C_1 \mathbf{x} + \mathbf{d}_1, R_1)$ $\mathbf{y}_2 \lvert \mathbf{x} \sim \mathcal{N}(C_2 \mathbf{x} + \mathbf{d}_2, R_2)$	$\mathbf{x} \sim \mathcal{N}(\mathbf{0}, I)$ $\mathbf{y}_1 \lvert \mathbf{x} \sim \mathcal{N}(C_1 \mathbf{x} + \mathbf{d}_1, R_1)$ $\mathbf{y}_2 \lvert \mathbf{x} \sim \mathcal{N}(C_2 \mathbf{x} + \mathbf{d}_2, \sigma_2^2 I)$

following maximization problem:

$$\max \frac{\mathbf{u}_1^\top \hat{\Sigma}_{12} \mathbf{u}_2}{\sqrt{\mathbf{u}_1^\top \mathbf{u}_1}\sqrt{\mathbf{u}_2^\top \mathbf{u}_2}} \tag{2.8}$$

where $\hat{\Sigma}_{12} \in \mathbb{R}^{q_1 \times q_2}$ is the sample cross-covariance matrix between area A and area B. The numerator of the objective function, $\mathbf{u}_1^\top \hat{\Sigma}_{12} \mathbf{u}_2$, is precisely the sample cross-covariance of area A activity projected onto $\mathbf{u}_1$, $\mathbf{u}_1^\top Y_1$, with area B activity projected onto $\mathbf{u}_2$, $\mathbf{u}_2^\top Y_2$. The denominator is a normalization factor that ensures that solutions $\mathbf{u}_1$ and $\mathbf{u}_2$ are unit vectors.

PLS can equivalently be defined as a minimization problem, in which the goal is to identify dimensions

that minimize the error between projected activity in each area:

$$\begin{aligned} &\min \|\mathbf{u}_1^\top Y_1 - \mathbf{u}_2^\top Y_2\|_F^2 \\ &\text{s.t.} \quad \mathbf{u}_1^\top \mathbf{u}_1 = 1 \\ &\qquad \mathbf{u}_2^\top \mathbf{u}_2 = 1 \end{aligned} \tag{2.9}$$

where the constraints $\mathbf{u}_1^\top \mathbf{u}_1 = 1$ and $\mathbf{u}_2^\top \mathbf{u}_2 = 1$ again enforce that solutions $\mathbf{u}_1$ and $\mathbf{u}_2$ are unit vectors.

Both of these problems can be solved for a number of pairs $p < \min(q_1, q_2)$ of dimensions via the singular value decomposition of the sample cross-covariance matrix[45]:

$$\hat{\Sigma}_{12} = U_1 S U_2^\top \tag{2.10}$$

where $U_1 \in \mathbb{R}^{q_1 \times q_1}$, $S \in \mathbb{R}^{q_1 \times q_2}$, and $U_2 \in \mathbb{R}^{q_2 \times q_2}$. The first p columns of U_1, $U_{1p} \in \mathbb{R}^{q_1 \times p}$, paired with the first p columns of U_2, $U_{2p} \in \mathbb{R}^{q_2 \times p}$, define a low-dimensional subspace within each area's population activity space (Fig. 2.1b, magenta-shaded planes). Each subspace represents patterns of population activity that exhibit the greatest covariance between areas. The cross-covariance associated with each pair of dimensions is given by the corresponding diagonal element of S.

Probabilistic interpretation of PLS

A probabilistic interpretation of PLS can be defined by the following linear-Gaussian relationship between observed activity in both areas A and B, $\mathbf{y}_1 \in \mathbb{R}^{q_1}$ and $\mathbf{y}_2 \in \mathbb{R}^{q_2}$, respectively, and latent variables $\mathbf{x} \in \mathbb{R}^p$:

$$\mathbf{x} \sim \mathcal{N}(\mathbf{0}, I) \tag{2.11}$$

$$\begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \end{bmatrix} \mid \mathbf{x} \sim \mathcal{N}\left(\begin{bmatrix} C_1 \\ C_2 \end{bmatrix} \mathbf{x} + \begin{bmatrix} \mathbf{d}_1 \\ \mathbf{d}_2 \end{bmatrix}, \begin{bmatrix} \sigma_1^2 I & \mathbf{0} \\ \mathbf{0} & \sigma_2^2 I \end{bmatrix} \right) \tag{2.12}$$

where, for each area $m = 1, 2$, $C_m \in \mathbb{R}^{q_m \times p}$, $\mathbf{d}_m \in \mathbb{R}^{q_m}$, and $\sigma_m^2 \in \mathbb{R}_{>0}$ are model parameters to be estimated from data. The loading matrix C_m linearly combines latent variables and maps them to observed neural activity in area m. The parameter $\mathbf{d}_m$ can be thought of as the mean firing rate of each neuron in area m. The noise variance σ_m^2 in each area is isotropic, but may be a different magnitude for each area. The latent variables and model parameters can be estimated from the neural activity by maximizing the data likelihood, $P(\mathbf{y}_1, \mathbf{y}_2)$, via the Expectation-Maximization (EM) algorithm.

With additional *post hoc* operations, probabilistic PLS solutions can be connected to their non-probabilistic counterparts (equations (2.8), (2.9)). The singular value decomposition of the cross-covariance $C_1 C_2^\top$ is given by $C_1 C_2^\top = U_1 S U_2^\top$ where $U_1 \in \mathbb{R}^{q_1 \times p}$, $S \in \mathbb{S}^{p \times p}$, and $U_2 \in \mathbb{R}^{q_2 \times p}$. Then, U_m corresponds to the top p PLS dimensions in area m. Projections of neural activity onto to these dimensions can be computed as $\mathbf{z}_m = U_m^\top C_m \mathbb{E}[\mathbf{x} | \mathbf{y}_1, \mathbf{y}_2] \in \mathbb{R}^p$.

Canonical correlation analysis (CCA)

The goal of canonical correlation analysis (CCA)[50] (Fig. 2.2b) is to identify pairs of dimensions, $\mathbf{u}_1 \in \mathbb{R}^{q_1}$ in area A and $\mathbf{u}_2 \in \mathbb{R}^{q_2}$ in area B, along which the correlation across areas is maximized. Concretely, CCA solves the following maximization problem:

$$\max \frac{\mathbf{u}_1^\top \hat{\Sigma}_{12} \mathbf{u}_2}{\sqrt{\mathbf{u}_1^\top \hat{\Sigma}_{11} \mathbf{u}_1} \sqrt{\mathbf{u}_2^\top \hat{\Sigma}_{22} \mathbf{u}_2}} \tag{2.13}$$

where $\hat{\Sigma}_{12} \in \mathbb{R}^{q_1 \times q_2}$ is the sample cross-covariance matrix between area A and area B. $\hat{\Sigma}_{11} \in \mathbb{S}^{q_1 \times q_1}$ and $\hat{\Sigma}_{22} \in \mathbb{S}^{q_2 \times q_2}$ are covariance matrices within each area. The numerator of the objective function, $\mathbf{u}_1^\top \hat{\Sigma}_{12} \mathbf{u}_2$, is precisely the sample cross-correlation of area A activity projected onto $\mathbf{u}_1$, $\mathbf{u}_1^\top Y_1$, with area B activity projected onto $\mathbf{u}_2$, $\mathbf{u}_2^\top Y_2$. The denominator is a normalization factor that ensures that projections onto solutions $\mathbf{u}_1$ and $\mathbf{u}_2$ within each area have unit correlation. CCA, in contrast with PLS, is thus scale invariant: rescaling observed dimensions of Y_1 or Y_2 does not alter the value of the objective.

CCA can equivalently be defined as a minimization problem, in which the goal is to identify dimensions that minimize the error between projected activity in each area:

$$\begin{aligned} &\min \|\mathbf{u}_1^\top Y_1 - \mathbf{u}_2^\top Y_2\|_F^2 \\ &\text{s.t.} \quad \mathbf{u}_1^\top \hat{\Sigma}_{11} \mathbf{u}_1 = 1 \\ &\qquad \mathbf{u}_2^\top \hat{\Sigma}_{22} \mathbf{u}_2 = 1 \end{aligned} \tag{2.14}$$

where the constraints $\mathbf{u}_1^\top \hat{\Sigma}_{11} \mathbf{u}_1 = 1$ and $\mathbf{u}_2^\top \hat{\Sigma}_{22} \mathbf{u}_2 = 1$ again enforce the scale invariance of solutions.

Both of these problems can be solved for a number of pairs $p < \min(q_1, q_2)$ of canonical dimensions via the singular value decomposition of the sample cross-correlation matrix[52]:

$$\hat{\Sigma}_{11}^{-\frac{1}{2}} \hat{\Sigma}_{12} \hat{\Sigma}_{22}^{-\frac{1}{2}} = V_1 S V_2^\top \tag{2.15}$$

where $V_1 \in \mathbb{R}^{q_1 \times q_1}$, $S \in \mathbb{R}^{q_1 \times q_2}$, and $V_2 \in \mathbb{R}^{q_2 \times q_2}$. Then let $U_1 = \hat{\Sigma}_{11}^{-\frac{1}{2}} V_1 \in \mathbb{R}^{q_1 \times q_1}$, and let $U_2 = \hat{\Sigma}_{22}^{-\frac{1}{2}} V_2 \in \mathbb{R}^{q_2 \times q_2}$. The first p columns of U_1, $U_{1p} \in \mathbb{R}^{q_1 \times p}$, paired with the first p columns of U_2, $U_{2p} \in \mathbb{R}^{q_2 \times p}$, are the top p canonical pairs. These canonical dimensions form an uncorrelated (and generally not orthogonal) basis that defines a low-dimensional subspace within each area's population activity space (Fig. 2.1b, magenta-shaded planes). Each subspace represents patterns of population activity that exhibit the greatest correlation between areas. The canonical correlation associated with each canonical pair is given by the corresponding diagonal element of S, which lies between 0 and 1.

Probabilistic interpretation of CCA

Probabilistic canonical correlation analysis[52] defines a linear-Gaussian relationship between observed activity in both areas A and B, $\mathbf{y}_1 \in \mathbb{R}^{q_1}$ and $\mathbf{y}_2 \in \mathbb{R}^{q_2}$, respectively, and latent variables $\mathbf{x} \in \mathbb{R}^p$:

$$\mathbf{x} \sim \mathcal{N}(\mathbf{0}, I) \tag{2.16}$$

$$\begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \end{bmatrix} \mid \mathbf{x} \sim \mathcal{N}\left(\begin{bmatrix} C_1 \\ C_2 \end{bmatrix} \mathbf{x} + \begin{bmatrix} \mathbf{d}_1 \\ \mathbf{d}_2 \end{bmatrix}, \begin{bmatrix} R_1 & \mathbf{0} \\ \mathbf{0} & R_2 \end{bmatrix} \right) \tag{2.17}$$

where, for each area $m = 1, 2$, $C_m \in \mathbb{R}^{q_m \times p}$, $\mathbf{d}_m \in \mathbb{R}^{q_m}$, and $R_m \in \mathbb{S}_{q_m \times q_m}$ are model parameters to be estimated from data. C_m and $\mathbf{d}_m$ are defined as for PLS. The noise covariance matrix R_m, however, is neither isotropic, as in probabilistic PLS, nor diagonal, as in FA. It can be any covariance matrix, thus leading instead to a block-diagonal observation noise matrix when looking across both areas. This constraint encourages the loading matrices C_1 and C_2 to capture as much shared covariance between areas as possible, and any remaining variability local to one area (including variability independent to each neuron) is explained by the observation noise matrices R_1 and R_2. The relationship between PLS and CCA is thus analogous to the relationship between PCA and FA. PLS solutions are biased toward dimensions of high variance in an area, whereas CCA solutions seek dimensions with correlated activity between areas, regardless of the within-area variance along those dimensions.

With additional *post hoc* operations, probabilistic CCA solutions can be connected to their non-probabilistic counterparts (equations (2.13), (2.14)). First let $\Sigma_{12} = C_1 C_2^\top$, $\Sigma_{11} = C_1 C_1^\top + R_1$, and $\Sigma_{22} = C_2 C_2^\top + R_2$, the cross-covariance and covariance matrices in each area, respectively. The singular value decomposition of the cross-correlation matrix $\Sigma_{11}^{-\frac{1}{2}} \Sigma_{12} \Sigma_{22}^{-\frac{1}{2}}$ is given by $\Sigma_{11}^{-\frac{1}{2}} \Sigma_{12} \Sigma_{22}^{-\frac{1}{2}} = V_1 S V_2^\top$ where $V_1 \in \mathbb{R}^{q_1 \times p}$, $S \in \mathbb{S}^{p \times p}$, and $V_2 \in \mathbb{R}^{q_2 \times p}$. Then for area m, let $U_m = \Sigma_{mm}^{-\frac{1}{2}} V_m$. U_m corresponds to the top p canonical dimensions in area m. Projections of neural activity onto to these dimensions can be computed as $\mathbf{z}_m = U_m^\top C_m \mathbb{E}[\mathbf{x} | \mathbf{y}_1, \mathbf{y}_2] \in \mathbb{R}^p$.

Reduced-rank regression (RRR)

Reduced-rank regression (RRR)[51] differs from PLS and CCA in that it treats the two areas asymmetrically. Neurons in area A, for example, are treated as independent variables, while neurons in area B are treated as dependent variables (Fig. 2.2c). The goal, then, as RRR is typically introduced[45], is to identify dimensions in area A (the source area) such that projections of area A's activity onto those dimensions are maximally predictive of the activity in area B (the target area). Here, we will give an unconventional introduction of RRR with the goal of highlighting its connections to PLS and CCA.

The equivalent goal of RRR is to identify pairs of dimensions, $\mathbf{u}_1 \in \mathbb{R}^{q_1}$ in area A and $\mathbf{u}_2 \in \mathbb{R}^{q_2}$ in area B, such that projections onto $\mathbf{u}_1$ in area A are maximally predictive of projections onto $\mathbf{u}_2$ in area B. Concretely, RRR solves the following maximization problem:

$$\max \frac{\mathbf{u}_1^\top \hat{\Sigma}_{12} \mathbf{u}_2}{\sqrt{\mathbf{u}_1^\top \hat{\Sigma}_{11} \mathbf{u}_1} \sqrt{\mathbf{u}_2^\top \mathbf{u}_2}} \tag{2.18}$$

where $\hat{\Sigma}_{12} \in \mathbb{R}^{q_1 \times q_2}$ is the sample cross-covariance matrix between area A and area B. $\hat{\Sigma}_{11} \in \mathbb{S}^{q_1 \times q_1}$ is the covariance matrix within area A. The numerator of the objective function, $\mathbf{u}_1^\top \hat{\Sigma}_{12} \mathbf{u}_2$, is precisely the sample cross-correlation of area A activity projected onto $\mathbf{u}_1$, $\mathbf{u}_1^\top Y_1$, with area B activity projected onto $\mathbf{u}_2$, $\mathbf{u}_2^\top Y_2$. The denominator is a normalization factor that ensures that projections onto solutions $\mathbf{u}_1$ within area A have unit correlation, and solutions $\mathbf{u}_2$ in area B are unit vectors. RRR is thus scale invariant with respect to source activity, but not with respect to target activity. These constraints encourage RRR to find source activity patterns that predict variance in the target population, regardless of whether that variance is shared among multiple target neurons or not.

RRR can equivalently be defined as a minimization problem, in which the goal is to identify dimensions that minimize the error between projected activity in each area:

$$\begin{aligned} &\min \|\mathbf{u}_1^\top Y_1 - \mathbf{u}_2^\top Y_2\|_F^2 \\ &\text{s.t.} \quad \mathbf{u}_1^\top \hat{\Sigma}_{11} \mathbf{u}_1 = 1 \\ &\qquad \mathbf{u}_2^\top \mathbf{u}_2 = 1 \end{aligned} \tag{2.19}$$

where the constraints $\mathbf{u}_1^\top \hat{\Sigma}_{11} \mathbf{u}_1 = 1$ and $\mathbf{u}_2^\top \mathbf{u}_2 = 1$ again enforce the same scaling properties as described above.

Both of these problems can be solved for a number of pairs $p < \min(q_1, q_2)$ of predictive dimensions via the singular value decomposition of the matrix $\hat{\Sigma}_{11}^{-\frac{1}{2}} \hat{\Sigma}_{12}$ (related to the ordinary least squares estimator)[45]:

$$\hat{\Sigma}_{11}^{-\frac{1}{2}} \hat{\Sigma}_{12} = V_1 S U_2^\top \tag{2.20}$$

where $V_1 \in \mathbb{R}^{q_1 \times q_1}$, $S \in \mathbb{R}^{q_1 \times q_2}$, and $U_2 \in \mathbb{R}^{q_2 \times q_2}$. (Equivalently, RRR solutions can be derived from the eigendecomposition of the covariance of optimal ordinary least squares predictions, $\hat{\Sigma}_{21} \hat{\Sigma}_{11}^{-1} \hat{\Sigma}_{12}$.) Then let $U_1 = \hat{\Sigma}_{11}^{-\frac{1}{2}} V_1 \in \mathbb{R}^{q_1 \times q_1}$. The first p columns of U_1, $U_{1p} \in \mathbb{R}^{q_1 \times p}$, paired with the first p columns of U_2, $U_{2p} \in \mathbb{R}^{q_2 \times p}$, are the top p predictive pairs. The predictive dimensions U_{1p} form an uncorrelated (and generally not orthogonal) basis that defines a low-dimensional subspace in the source area's population space. In contrast, the predictive dimensions U_{2p} form an orthonormal basis that defines a low-dimensional subspace in the target area's population space (Fig. 2.1b, magenta-shaded planes). Activity in the source predictive subspace is maximally predictive of activity in the target predictive subspace.

The predictive power from the source area to the target area is given by the diagonal elements of S, and variance explained in the target area (R^2) along the j^{th} predictive dimension can be computed according to $R^2_j = (S^2_{jj})/\text{tr}(\hat{\Sigma}_{22})$.

Probabilistic interpretation of RRR

A probabilistic interpretation of RRR can be defined by the following linear-Gaussian relationship between observed activity in both areas A and B, $\mathbf{y}_1 \in \mathbb{R}^{q_1}$ and $\mathbf{y}_2 \in \mathbb{R}^{q_2}$, respectively, and latent variables $\mathbf{x} \in \mathbb{R}^p$:

$$\mathbf{x} \sim \mathcal{N}(\mathbf{0}, I) \tag{2.21}$$

$$\begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \end{bmatrix} \mid \mathbf{x} \sim \mathcal{N}\left(\begin{bmatrix} C_1 \\ C_2 \end{bmatrix} \mathbf{x} + \begin{bmatrix} \mathbf{d}_1 \\ \mathbf{d}_2 \end{bmatrix}, \begin{bmatrix} R_1 & \mathbf{0} \\ \mathbf{0} & \sigma_2^2 I \end{bmatrix} \right) \tag{2.22}$$

where, for each area $m = 1, 2$, $C_m \in \mathbb{R}^{q_m \times p}$ and $\mathbf{d}_m \in \mathbb{R}^{q_m}$ have the same definitions as for PLS and CCA. $R_1 \in \mathbb{S}_{q_1 \times q_1}$ is not constrained to be diagonal, as in probabilistic CCA. The noise variance in area B, $\sigma_2^2 I$, however, is isotropic. The structure of the probabilistic RRR model is thus "in between" CCA and PLS. C_1 is encouraged to explain covariance shared between areas A and B. C_2 is encouraged to explain as much variance in area B as possible.

With additional *post hoc* operations, probabilistic RRR solutions can be connected to their non-probabilistic counterparts (equations (2.18), (2.19)). First let $\Sigma_{12} = C_1 C_2^\top$ and $\Sigma_{11} = C_1 C_1^\top + R_1$, the cross-covariance matrix and covariance matrix in area A, respectively. The singular value decomposition of the matrix $\Sigma_{11}^{-\frac{1}{2}} \Sigma_{12}$ is given by $\Sigma_{11}^{-\frac{1}{2}} \Sigma_{12} = V_1 S U_2^\top$ where $V_1 \in \mathbb{R}^{q_1 \times p}$, $S \in \mathbb{S}^{p \times p}$, and $U_2 \in \mathbb{R}^{q_2 \times p}$. Then for area A, let $U_1 = \Sigma_{11}^{-\frac{1}{2}} V_1$. U_m corresponds to the top p predictive dimensions in area m. Projections of neural activity onto to these dimensions can be computed as $\mathbf{z}_m = U_m^\top C_m \mathbb{E}[\mathbf{x} | \mathbf{y}_1, \mathbf{y}_2] \in \mathbb{R}^p$. Predictions of activity in the target area, given activity in the source area, can be computed according to $\mathbb{E}[\mathbf{y}_2 | \mathbf{y}_1] = C_2 C_1^\top (C_1 C_1^\top + R_1)^{-1} (\mathbf{y}_1 - \mathbf{d}_1) + \mathbf{d}_2 = U_2 S U_1^\top (\mathbf{y}_1 - \mathbf{d}_1) + \mathbf{d}_2$.

Chapter 3

Delayed latents across groups (DLAG)

In this chapter, we introduce the DLAG model (Sections 3.1–3.4) and its accompanying fitting (Section 3.5) and model selection procedures (Section 3.6). We conclude with a mathematical discussion of DLAG's interpretation as a low-rank decomposition of the covariance matrix of a time series (Section 3.7).

3.1 DLAG Model Overview

Consider recording the activity of two populations of neurons (Fig. 3.1, left column), measured as, for example, the number of spikes counted within nonoverlapping time bins. Here we will take these populations as belonging to two different brain areas, A and B. In principle, they can belong to any meaningful groups, such as cortical layers or cell types.

DLAG dissects the recorded population activity in each area on individual trials into a linear combination (weighted sum) of two types of latent variables (Fig. 3.1, center column). The first type of latent variable, *across-area* variables, describes population activity that is correlated across areas (illustrated by the magenta box spanning both areas in Fig. 3.1). The second type of latent variable, *within-area* variables, describes population activity in one area that is not related to population activity in the other area (Fig. 3.1; blue: within A; red: within B). Whether or not the within-area variables are a subject of scientific study, they are critical to the correct estimation of across-area variables (see Section 3.7).

The temporal structure of within- and across-area variables are both described by relating each latent variable at different time points through Gaussian processes. Each Gaussian process is associated with its own characteristic timescale that controls the temporal smoothing of neural activity. Across-area variables are defined in pairs, where the elements of each pair correspond to the two areas and covary with each other according to a common Gaussian process. Importantly, the elements of each across-area pair are time-delayed relative to each other (Fig. 3.1, D_1 between the first pair and D_2 between the second pair).

All DLAG model parameters, including the Gaussian process timescales and time delays, are estimated from the neural activity using an exact expectation-maximization (EM) algorithm. After the DLAG model parameters are estimated from the neural activity, the time courses of within- and across-area latent variables can be studied on a trial-to-trial basis. Conceptually, DLAG can be viewed as a time series extension of probabilistic canonical correlation analysis (pCCA)[52,53] or a multi-area extension of Gaussian process factor analysis (GPFA)[47,54] with the added ability to estimate time delays between two areas.

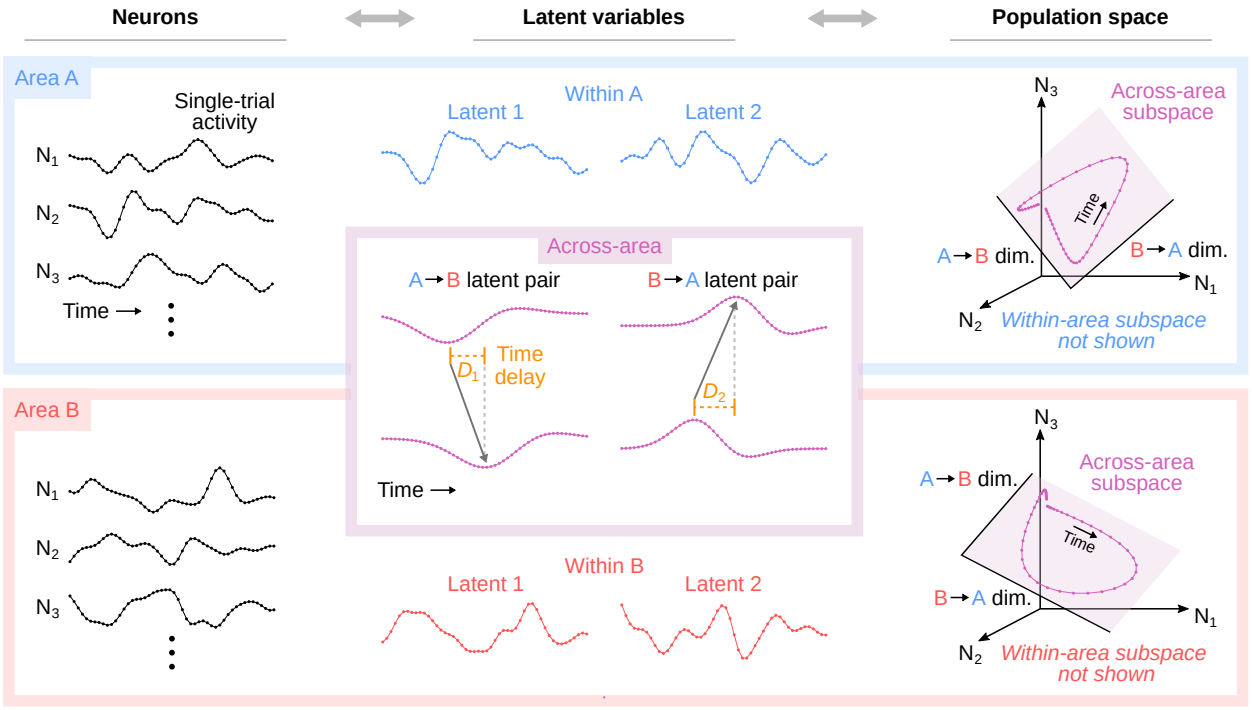

Figure 3.1: DLAG conceptual illustration. From left to right: neurons, latent variables, and population activity space representations in two recorded brain areas analyzed by DLAG (top row / blue box: area A; bottom row / red box: area B). Left column: Single-trial activity of neurons simultaneously recorded in each area. Only three neurons (N_1, N_2, N_3) are shown in each area for clarity. Center column: Within-area variables are shown in the color corresponding to the area in which they belong (Within A: blue; Within B: red). For clarity, only two within-area variables are shown in each area, but in principle there may be a greater number, as determined by DLAG from the recorded activity. Across-area variables are shown in magenta. The magenta box inset overlaps the blue and red boxes for area A and B, respectively, to indicate that across-area variables are shared among neurons in both areas. The location of each across-area variable (i.e., within the bounds of Area A's box or area B's box) indicates which area's activity it reflects. Between area A and area B, across-area variables are vertically paired. The time courses of each pair are related after a time delay (D_1: delay between the left pair; D_2: delay between the right pair). The sign of this delay allows each pair to be associated with a directed interaction (A to B or B to A), which is indicated by gray arrows. For clarity, only two across-area variable pairs are shown. Right column: The activity of each neural population can be represented in a population activity space, where each axis represents the activity of a single neuron (N_1, N_2, N_3). Each point in population space represents the population activity at a particular time, and the points trace out a trajectory over time (magenta curve). DLAG identifies two linearly independent subspaces in each area: a within-area subspace (not shown, for clarity) and an across-area subspace (magenta-shaded plane). Each dimension ('dim.') of the across-area subspace is associated with a directed interaction.

Intuitively, if a particular time course is reflected in the population activity of area A, and a similar time course is reflected in the population activity of area B, but after a time delay, then an across-area variable pair can describe the apparent flow of that signal from A to B. And if, concurrently, a time course is first seen in area B, followed by area A, a second across-area variable pair can also describe the flow of that inter-areal signal. The key to disambiguating the first and second across-area variable pairs is that they involve different population activity patterns (i.e., a "loading" vector indicating how the activity of each neuron relates to the latent variable). In fact, DLAG can identify many across-area variable pairs, each with a delay of its own sign and magnitude, to capture multiple concurrent streams of signal flow between the two populations at different timescales.

The relationship between within- and across-area latent variables and observed population activity in each area can be represented geometrically with the concept of a population activity space (Fig. 3.1, right column). For each area, we can define a high-dimensional population activity space where each axis represents the activity of one neuron. Each point in the space represents the population activity at a particular time, and the points trace out a trajectory over time. DLAG's two types of latent variables each define the axes (dimensions) of a low-dimensional subspace within this population activity space (in Fig. 3.1, we show only the across-area subspaces for visual clarity). Each dimension of these subspaces represents a population activity pattern.

3.2 Mathematical notation

To disambiguate each variable or parameter in the DLAG model, we need to keep track of up to four labels that indicate their associated (1) subpopulation (for example, brain area); (2) neuron or latent variable index; (3) time point; or (4) designation as within- or across-area. We indicate the first three labels via subscripts, where subpopulations (areas) are indexed by $m = 1, 2$; neurons or latent variables are indexed by j (we'll indicate the upper bound as appropriate); and time is indexed by $t = 1, \ldots, T$. For example, we define the observed activity of neuron j (out of q_m) in area m at time t as $y_{m,j,t} \in \mathbb{R}$. To indicate a collection of all variables along a particular index, we replace that index with the ':' symbol. Hence we represent the simultaneous activity of the population of q_m neurons observed in area m at time t as the vector $\mathbf{y}_{m,:,t} \in \mathbb{R}^{q_m}$. For concision, where a particular index is either not applicable or not immediately relevant, we omit it. The identities of the remaining indices should be clear from context. For example, throughout this work we consider only the activity of a full population, and not of single neurons, so we rewrite $\mathbf{y}_{m,:,t}$ as $\mathbf{y}_{m,t}$. Finally, we indicate a latent variable's or parameter's designation as within- or across-area via a superscript, where 'w' indicates within-area, and 'a' indicates across-area. For example,

we define across-area latent variable j (out of p^a) in area m at time t as $x^a_{m,j,t} \in \mathbb{R}$, and the collection of all p^a latent variables as the vector $\mathbf{x}^a_{m,:,t} \in \mathbb{R}^{p^a}$. We similarly define within-area latent variable j (out of p^w_m) in area m at time t as $x^w_{m,j,t} \in \mathbb{R}$, and the collection of all p^w_m latent variables as the vector $\mathbf{x}^w_{m,:,t} \in \mathbb{R}^{p^w_m}$.

It is conceptually helpful to understand the above notation for observed ($\mathbf{y}$) and latent ($\mathbf{x}$) variables as taking cross-sections of matrices. For example, observed activity in area m can be grouped into the matrix $Y_m = [\mathbf{y}_{m,1} \cdots \mathbf{y}_{m,T}] \in \mathbb{R}^{q_m \times T}$. Then, each $\mathbf{y}_{m,t}$ is a column of Y_m. Similarly, across-area latent variables in area m can be grouped into the matrix $X^a_m = [\mathbf{x}^a_{m,:,1} \cdots \mathbf{x}^a_{m,:,T}] \in \mathbb{R}^{p^a \times T}$. Each $\mathbf{x}^a_{m,:,t}$ is a column of X^a_m. Similarly, we represent a row of X^a_m (i.e., the values of a single latent variable j at all time points) as $\mathbf{x}^a_{m,j,:} \in \mathbb{R}^T$. Within-area latent variables can be understood analogously from the matrix $X^w_m = [\mathbf{x}^w_{m,:,1} \cdots \mathbf{x}^w_{m,:,T}] \in \mathbb{R}^{p^w_m \times T}$. Finally, we note that there is a separate set of observed and latent variables (Y_m, X^a_m, X^w_m) for each trial, while there is a single set of DLAG model parameters shared across trials. For concision, we index trial number only as needed, and omit the trial index otherwise.

We will explicitly define all other variables and parameters as they appear, but for reference, we list common variables and parameters below:

Observed neural activity

- q_m – number of neurons observed in area m
- Y_m – $q_m \times T$ matrix of observed activity in area m
- $\mathbf{y}_{m,t}$ – $q_m \times 1$ vector of observed activity in area m at time t; the t^{th} column of Y_m

Latent variables

- p^a – number of across-area variables (same for both areas)
- X^a_m – $p^a \times T$ matrix of across-area variables in area m
- $\mathbf{x}^a_{m,:,t}$ – $p^a \times 1$ vector of across-area variables in area m at time t; the t^{th} column of X^a_m
- $\mathbf{x}^a_{m,j,:}$ – $T \times 1$ vector of values of across-area variable j in area m over time; the j^{th} row of X^a_m
- p^w_m – number of within-area variables in area m
- X^w_m – $p^w_m \times T$ matrix of within-area variables in area m
- $\mathbf{x}^w_{m,:,t}$ – $p^w_m \times 1$ vector of within-area variables in area m at time t; the t^{th} column of X^w_m
- $\mathbf{x}^w_{m,j,:}$ – $T \times 1$ vector of values of within-area variable j in area m over time; the j^{th} row of X^w_i

Model parameters

- C^a_m – $q_m \times p^a$ across-area loading matrix for area m
- C^w_m – $q_m \times p^w_m$ within-area loading matrix for area m
- $\mathbf{d}_m$ – $q_m \times 1$ mean parameter for area m
- R_m – $q_m \times q_m$ observation noise covariance matrix for area m
- $D_{m,j}$ – time delay parameter between area m and across-area variable j
- D_j – relative time delay associated with across-area variable j; $D_j = D_{2,j} - D_{1,j}$
- τ^a_j – Gaussian process timescale for across-area variable j
- σ^a_j – Gaussian process noise parameter for across-area variable j
- $\tau^w_{m,j}$ – Gaussian process timescale for within-area variable j in area m
- $\sigma^w_{m,j}$ – Gaussian process noise parameter for within-area variable j in area m

Gaussian process covariances

- $K^a_{m_1,m_2,j}$ – $T \times T$ covariance matrix for across-area variable j, between areas m_1 and m_2
- $k^a_{m_1,m_2,j}$ – covariance function for across-area variable j, between areas m_1 and m_2
- $K^w_{m,j}$ – $T \times T$ covariance matrix for within-area variable j in area m
- $k^w_{m,j}$ – covariance function for within-area variable j in area m

3.3 DLAG observation model

For area m at time t, we define a linear-Gaussian relationship between observed activity, $\mathbf{y}_{m,t}$, and latent variables, $\mathbf{x}^a_{m,:,t}$ and $\mathbf{x}^w_{m,:,t}$[53]:

$$\mathbf{y}_{m,t} = C^a_m \mathbf{x}^a_{m,:,t} + C^w_m \mathbf{x}^w_{m,:,t} + \mathbf{d}_m + \boldsymbol{\varepsilon}_m \tag{3.1}$$

$$\boldsymbol{\varepsilon}_m \sim \mathcal{N}(\mathbf{0}, R_m) \tag{3.2}$$

where $C^a_m \in \mathbb{R}^{q_m \times p^a}$, $C^w_m \in \mathbb{R}^{q_m \times p^w_m}$, $\mathbf{d}_m \in \mathbb{R}^{q_m}$, and $R_m \in \mathbb{S}^{q_m \times q_m}$ ($\mathbb{S}^{q_m \times q_m}$ is the set of $q_m \times q_m$ symmetric matrices) are model parameters to be estimated from data. The relationship between observed and latent variables is illustrated graphically in Fig. 3.2. The loading matrices C^a_m and C^w_m linearly combine latent variables and map them to observed neural activity. The parameter $\mathbf{d}_m$ can be thought of as the mean firing rate of each neuron. $\boldsymbol{\varepsilon}_m$ is a zero-mean Gaussian random variable, where we constrain the covariance matrix R_m to be diagonal, as in factor analysis (FA)[48] and Gaussian process factor analysis (GPFA)[47], to capture variance that is independent to each neuron. This constraint encourages the latent variables to explain as much of the shared variance among neurons as possible.

As we will describe, at time point t, across-area variables $\mathbf{x}^a_{1,:,t}$ and $\mathbf{x}^a_{2,:,t}$ in area 1 and area 2, respectively, are coupled with each other, and thus each area has the same number of across-area variables, p^a. Within-area variables are not coupled across areas, on the other hand, and thus each area m may have a different number of within-area variables, p^w_m. Because we seek a low-dimensional description of neural activity in each area, the combined number of across- and within-area variables is less than the number of neurons, i.e., $p^a + p^w_m < q_m$, where p^a and p^w_m are determined by the data (see Section 3.6).

The parameters C^w_m and C^a_m have an intuitive geometric interpretation (Fig. 3.1, right column). Each element of $\mathbf{y}_{m,t}$, the activity of each neuron in area m, can be represented as an axis in a high-dimensional population activity space. Then the columns of C^a_m, the across-area loading matrix for area m, define a subspace in this population activity space, where each dimension corresponds to a distinct across-area latent variable. This across-area subspace represents patterns of population activity that is correlated across areas. Analogously, the columns of C^w_m define a within-area subspace, which represents patterns of population activity that is shared only among neurons within area m. Additionally, as we will discuss

below, since the j^{th} pair of across-area variables ($\mathbf{x}^a_{1,j,:}$, $\mathbf{x}^a_{2,j,:}$) is associated with a direction of population signal flow (Fig. 3.1, center column), so too are the corresponding columns in C^a_1 and C^a_2. The across-area subspace can thus be partitioned further based on the nominal directionality of activity patterns (area 1 to area 2, or area 2 to area 1). Finally, note that the columns of C^a_m are linearly independent but not, in general, orthogonal. Likewise, the columns of C^w_m are linearly independent but not, in general, orthogonal. The across- and within-area subspaces in area m (spanned by the columns of C^a_m and by the columns of C^w_m, respectively) are also linearly independent but not, in general, orthogonal. The ordering of the columns of each loading matrix, and of the corresponding latent variables, is arbitrary.

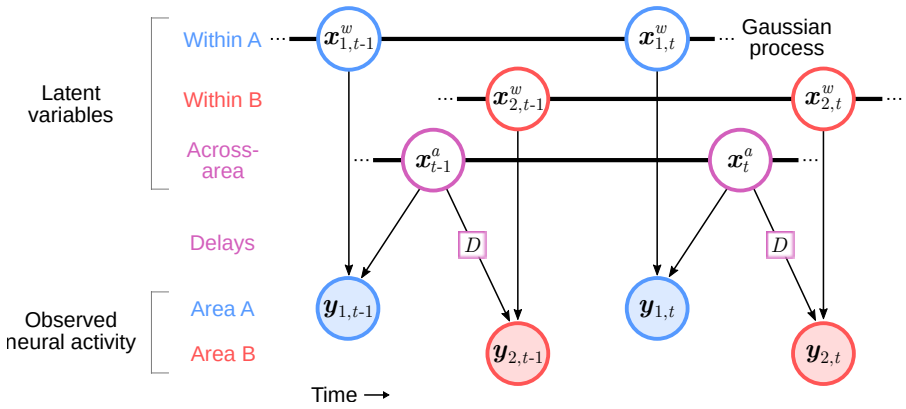

Figure 3.2: DLAG directed graphical model representation. Filled circles represent observed variables (i.e., observed neural activity in each area), where $\mathbf{y}_{1,t}$ and $\mathbf{y}_{2,t}$ are the observed neural activity in area A and B, respectively, at time t. Unfilled circles represent latent variables, where $\mathbf{x}^a_t$ are across-area variables at time t; $\mathbf{x}^w_{1,t}$ and $\mathbf{x}^w_{2,t}$ are within-area variables in area A and B, respectively, at time t. D represents the set of relative time delay parameters between the two areas. Color indicates a variable's or parameter's association with area A (blue), area B (red), or both (magenta). Arrows indicate conditional dependence relationships between variables. In particular, the arrows point from latent variables to observed neural activity, framing DLAG as a generative model. Thick black lines indicate that variables are related in time via a Gaussian process. Here two time steps are shown ($t-1$ and t), and time evolves from left to right.

3.4 DLAG state model

We seek to extract smooth, single-trial latent time courses, where the degree of smoothing is determined by the neural activity (as described below). The time course of each within-area and across-area latent variable is described by a Gaussian process (GP)[55].

Within-area latent variables For each within-area variable $j = 1, \ldots, p^w_m$ in brain area m, we define a separate GP as follows[47]:

$$\mathbf{x}^w_{m,j,:} \sim \mathcal{N}(\mathbf{0}, K^w_{m,j}) \tag{3.3}$$

where $K^w_{m,j} \in \mathbb{S}^{T\times T}$ is the covariance matrix for within-area variable j of area m. DLAG is compatible with any valid form of GP covariance, but for the present work, we choose the commonly used squared exponential (SE) function. Then, element (t_1, t_2) of $K^w_{m,j}$, the covariance between samples of the within-area variable at times t_1 and t_2, can be computed according to:

$$k^w_{m,j}(t_1, t_2) = \left(1 - (\sigma^w_{m,j})^2\right) \exp\left(-\frac{(\Delta t)^2}{2(\tau^w_{m,j})^2}\right) + (\sigma^w_{m,j})^2 \cdot \delta_{\Delta t} \tag{3.4}$$

$$\Delta t = t_2 - t_1 \tag{3.5}$$

where the characteristic timescale, $\tau^w_{m,j} \in \mathbb{R}_{>0}$, and GP noise variance, $(\sigma^w_{m,j})^2 \in (0, 1)$, are model parameters. $\delta_{\Delta t}$ is the kronecker delta, which is 1 for $\Delta t = 0$ (equivalently, $t_1 = t_2$) and 0 otherwise.

Notice that $k^w_{m,j}$ is stationary: the SE function depends only on the time difference $(t_2 - t_1)$ (Fig. 3.3a). This stationarity gives the covariance matrix $K^w_{m,j}$ a characteristic banded structure (Fig. 3.3b). The characteristic timescale, $\tau^w_{m,j}$, dictates the width of $k^w_{m,j}(t_1, t_2)$, or equivalently, how rapidly the latent variable changes over time. The $\tau^w_{m,j}$ parameters are estimated from the neural activity, together with the other DLAG parameters (see Section 3.5). We follow the same conventions as in [47], and fix $(\sigma^w_{m,j})^2$ to a small value (10^{-3}). Note also that, under this definition, the process is normalized so that $k^w_{m,j}(t_1, t_2) = 1$ for $t_1 = t_2$. Thus, the prior distribution of within-area latent variables $\mathbf{x}^w_{m,:,t}$ in area m at each time t follows the standard normal distribution, $\mathcal{N}(\mathbf{0}, I)$. This normalization removes model redundancy in the scaling of X^w_m and C^w_m.

Beyond describing within-area interactions, within-area variables are critical to the interpretability of across-area variables. As we will define below, across-area variables describe the activity of neurons in both areas. Within-area variables could, in principle, be formulated as a special case of across-area variables, where the loading coefficients to one area (the appropriate columns of C^a_1 or C^a_2 in equation (3.1)) are identically zero. If the model does not allow for within-area variables, then across-area variables must explain within-area activity in addition to across-area activity. Across-area variables could thus reflect a mixture of within- and across-area activity in this case, obfuscating their interpretation as representing population activity patterns that are correlated across areas. The presence of within-area variables allows the across-area variables to isolate activity that is truly correlated across areas. This statistical phenomenon applies to other statistical models, and is not specific to DLAG[32,56]. See Section 3.7 for further mathematical discussion.

Across-area latent variables We next describe across-area temporal structure. Across-area variables are different from within-area variables in two respects: (1) across-area variables are defined in pairs, where the elements of each pair correspond to the two areas, and (2) the elements of each pair are time-delayed

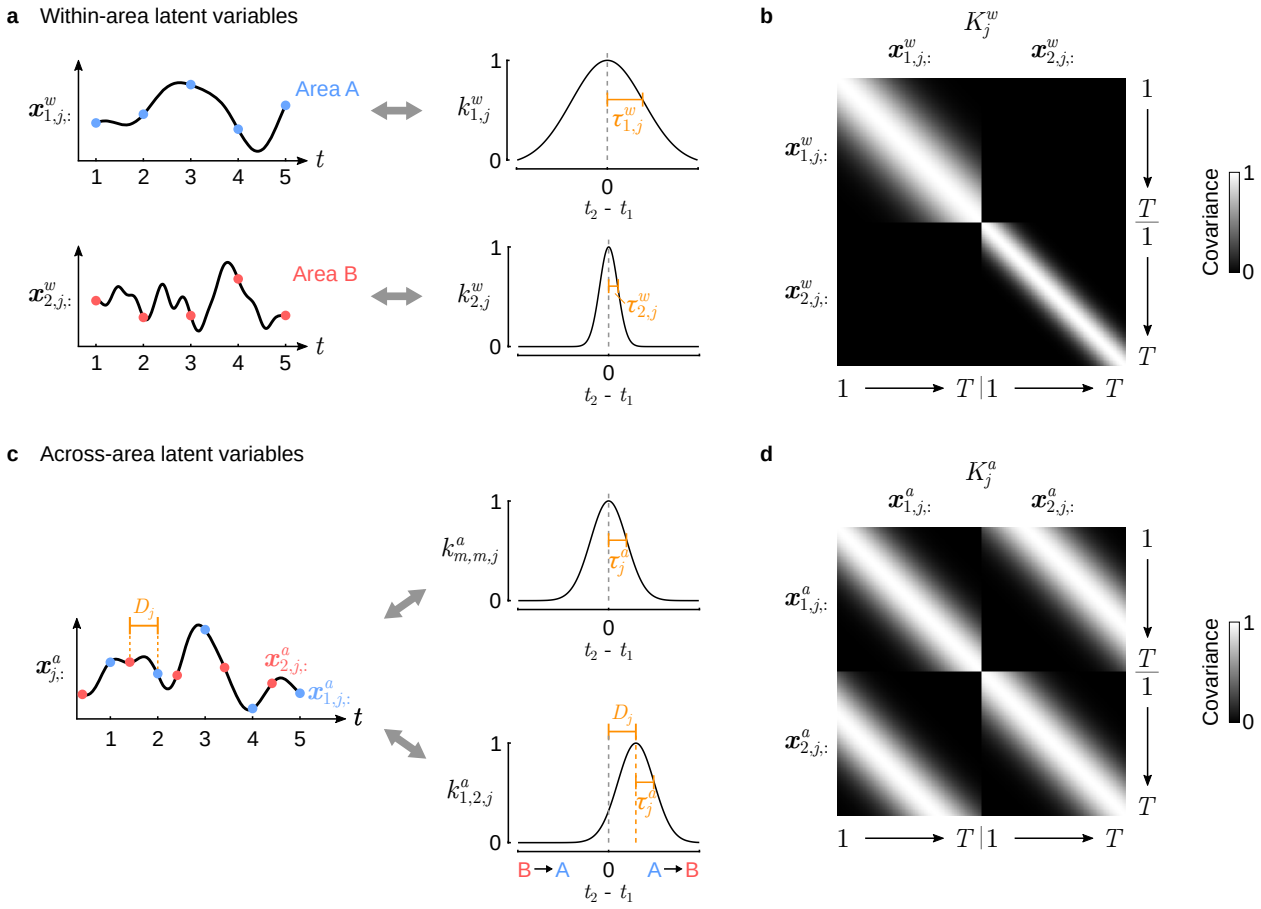

Figure 3.3: The use of Gaussian processes in the DLAG state model. (**a**) Within-area state model. Left column: Within-area time courses (area A: $\mathbf{x}^w_{1,j,:}$, blue points; area B: $\mathbf{x}^w_{2,j,:}$, red points) can be described as a finite number of samples drawn from a Gaussian process (GP) for each area and each j. Right column: The temporal structure of each within-area GP is governed by a covariance function (area A: $k^w_{1,j}$; area B: $k^w_{2,j}$). The squared exponential (SE) function, chosen for the present work, is defined by a timescale parameter ($\tau^w_{1,j}$, $\tau^w_{2,j}$), which controls the width of the covariance kernel, or equivalently, how quickly the latent variable changes over time. (**b**) An example set of within-area GP covariance matrices (K^w_j). The banded structure emerges from the choice of squared exponential function and stationarity of the GP covariance. Note the independence of within-area latent variables across areas: each latent variable has its own characteristic timescale, and cross-covariance terms are all zero. (**c**) Across-area state model. Left column: Like within-area time courses, across-area time courses can also be described as a finite number of samples drawn from a GP. In contrast to the within-area time courses, which are independent across areas, across-area time courses are coupled across areas, drawn from a common GP ($\mathbf{x}^a_{j,:}$). The sampling grid of area A (blue) is shifted by a time delay (D_j) relative to that of area B (red). Right column: The temporal structure of the common GP is governed by a SE covariance function. The width of the auto- and cross-covariances ($k^a_{m,m,j}$ and $k^a_{1,2,j}$, respectively) is controlled by a timescale parameter (τ^a_j). The center of the cross-covariance is controlled by the delay parameter D_j (positive delays: A leads B; negative delays: B leads A). (**d**) An example across-area GP covariance matrix (K^a_j). The banded structure emerges from the choice of squared exponential function and stationarity of the GP covariance. Note the non-zero cross-covariance terms in the off-diagonal blocks of K^a_j: the banded structure is shifted from the diagonal of each off-diagonal block by the delay parameter D_j.

relative to each other (Fig. 3.1, center column). Thus in contrast to our definition of within-area variables, in which we considered each area separately, we now consider across-area variables in both areas together: $\mathbf{x}^a_{1,j,:} \in \mathbb{R}^T$ and $\mathbf{x}^a_{2,j,:} \in \mathbb{R}^T$, the j^{th} rows of X^a_1 and X^a_2, respectively, for the j^{th} across-area variable.

The across-area latent variables of area 1 ($\mathbf{x}^a_{1,j,:}$) and area 2 ($\mathbf{x}^a_{2,j,:}$) belong to the same GP (Fig. 3.3c). The $\mathbf{x}^a_{1,j,:}$ are values of the GP sampled on a time grid. The $\mathbf{x}^a_{2,j,:}$ are values of the same GP, also sampled on a time grid, but offset from the time grid of area 1 by a time delay. We define the GP for each across-area variable $j = 1, \ldots, p^a$ as follows:

$$\begin{bmatrix} \mathbf{x}^a_{1,j,:} \\ \mathbf{x}^a_{2,j,:} \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} K^a_{1,1,j} & K^a_{1,2,j} \\ K^a_{2,1,j} & K^a_{2,2,j} \end{bmatrix} \right) \tag{3.6}$$

where $K^a_{1,1,j} = K^a_{2,2,j}$ describe the autocovariance of each across-area variable, and $K^a_{1,2,j} = K^{a\top}_{2,1,j}$ describe the cross-covariance that couples the two areas (Fig. 3.3d).

To express the auto- and cross-covariance functions, we introduce additional notation. Specifically, we indicate brain areas with two subscripts, $m_1 = 1, 2$ and $m_2 = 1, 2$. Then, we define $K^a_{m_1,m_2,j} \in \mathbb{R}^{T\times T}$ to be either the auto- or cross-covariance matrix between across-area variable $\mathbf{x}^a_{m_1,j,:}$ in area m_1 and across-area variable $\mathbf{x}^a_{m_2,j,:}$ in area m_2. We again choose to use the SE function for GP covariances. Therefore, element (t_1, t_2) of each $K^a_{m_1,m_2,j}$ can be computed as follows[54]:

$$k^a_{m_1,m_2,j}(t_1, t_2) = \left(1 - (\sigma^a_j)^2\right) \exp\left(-\frac{(\Delta t)^2}{2(\tau^a_j)^2} \right) + (\sigma^a_j)^2 \cdot \delta_{\Delta t} \tag{3.7}$$

$$\Delta t = \left(t_2 - D_{m_2,j}\right) - \left(t_1 - D_{m_1,j}\right) \tag{3.8}$$

where the characteristic timescale, $\tau^a_j \in \mathbb{R}_{>0}$, and the GP noise variance, $(\sigma^a_j)^2 \in (0, 1)$, are model parameters. $\delta_{\Delta t}$ is the kronecker delta, which is 1 for $\Delta t = 0$ and 0 otherwise.

We also introduce two new parameters: the time delay to area m_1, $D_{m_1,j} \in \mathbb{R}$, and the time delay to area m_2, $D_{m_2,j} \in \mathbb{R}$. Notice that, when computing the autocovariance for area m (i.e., $m_1 = m_2 = m$), the time delay parameters $D_{m_1,j}$ and $D_{m_2,j}$ are equal, and so Δt (equation (3.8)) reduces simply to the time difference $(t_2 - t_1)$, as in the within-area case (equation (3.5)). Time delays are therefore only relevant when computing the cross-covariance between area 1 and area 2. The time delay to area 1, $D_{1,j}$, and the time delay to area 2, $D_{2,j}$, by themselves have no physically meaningful interpretation. Their difference $D_j = D_{2,j} - D_{1,j}$, however, represents a well-defined, continuous-valued time delay from area 1 to area 2. The sign of the relative time delay D_j indicates the directionality of the lead-lag relationship between areas captured by latent variable j (positive: area 1 leads area 2; negative: area 2 leads area 1), which we interpret as a description of inter-areal signal flow.

Both the characteristic timescales τ_j^a and relative delays D_j are estimated from the neural activity, together with the other DLAG parameters (see Section 3.5). More specifically, to ensure identifiability of time delay parameters, we designate area 1 as the reference area, and fix the delays for area 1 at 0, that is, $D_{1,j} = 0$ for all across-area variables $j = 1, \ldots, p^a$. Then, each relative time delay D_j is simply the time delay parameter to area 2, $D_{2,j}$. As in the within-area case, the across-area GP noise variance, $(\sigma_j^a)^2$, is set to a small value (10^{-3}). Furthermore, the across-area GP is also normalized so that $k^a_{m_1,m_2,j}(t_1, t_2) = 1$ if $\Delta t = 0$, thereby removing model redundancy in the scaling of X_m^a and C_m^a.

Note that D_j need not be an integer multiple of the sampling period or spike count bin width of the neural activity. Because latent time courses and time delays are continuous-valued, DLAG can leverage the correlated activity of the neuronal populations to recover delays that are smaller than the sampling period or spike count bin width. This feature of the DLAG model distinguishes it from other time series modeling approaches (see Discussion). For intuition, consider the case of reconstructing a pair of time-delayed (noiseless) band-limited signals from a set of discrete samples. So long as these signals are sampled at a sufficiently high rate (i.e., the Nyquist rate), they can be perfectly reconstructed. The relative time delay between the paired signals can then be estimated precisely by identifying the peak of the cross-correlation function between the reconstructed signals. DLAG's process for the estimation of latent time courses and time delays is analogous. We systematically characterize the effects of various data attributes on DLAG's time delay estimates in the next chapter.

DLAG special cases Finally, we consider some special cases of the DLAG model that illustrate its relationship to other dimensionality reduction methods. First, by fixing all time delays to zero ($D_j = 0$), and by removing within-area latent variables ($p_1^w = p_2^w = 0$), DLAG becomes equivalent to Gaussian process factor analysis (GPFA)[47] applied to both areas jointly. By removing instead the across-area latent variables ($p^a = 0$), and keeping the within-area latent variables intact, DLAG becomes equivalent to GPFA applied to each area independently. And finally, by removing temporal smoothing (i.e., in the limit as all GP noise parameters σ_j^a, $\sigma_{m,j}^w$ approach 1), while keeping both within- and across-area latent variables, DLAG becomes similar to probabilistic canonical correlation analysis (pCCA)[52,53]. Whereas pCCA describes within-area activity via observation noise covariance matrices (R_m; see equation (3.37)), this special-case DLAG model would describe within-area activity via low-dimensional latent variables (the class of static methods that include within-area latent variables is sometimes referred to as inter-battery factor analysis[57]).

3.5 Fitting the DLAG model

Equations (3.1)–(3.8) provide a full definition of the DLAG model. In this section, we describe how DLAG model parameters are fit using exact Expectation Maximization (EM), where the parameters are

$$\theta = \left\{C,\ \mathbf{d},\ R,\ \{D_j\}_{j=1}^{p^a},\ \{\tau_j^a\}_{j=1}^{p^a},\ \{\tau_{1,j}^w\}_{j=1}^{p_1^w},\ \{\tau_{2,j}^w\}_{j=1}^{p_2^w}\right\} \tag{3.9}$$

Toward that end, we first write the DLAG observation model more compactly as follows. Define the joint activity of neurons in all brain areas by vertically concatenating the observations in each area, $\mathbf{y}_{1,t}$ and $\mathbf{y}_{2,t}$:

$$\mathbf{y}_t = \begin{bmatrix} \mathbf{y}_{1,t} \\ \mathbf{y}_{2,t} \end{bmatrix} \in \mathbb{R}^q \tag{3.10}$$

where $q = q_1 + q_2$. Next we group together the across- and within-area latent variables for the m^{th} brain area to define $\mathbf{x}_{m,t} = [\mathbf{x}_{m,:,t}^{a\top}\ \mathbf{x}_{m,:,t}^{w\top}]^\top \in \mathbb{R}^{p_m}$, where $p_m = p^a + p_m^w$. We then vertically concatenate the latent variables in each area:

$$\mathbf{x}_t = \begin{bmatrix} \mathbf{x}_{1,t} \\ \mathbf{x}_{2,t} \end{bmatrix} \in \mathbb{R}^p \tag{3.11}$$

where $p = p_1 + p_2$. We also define the following structured matrices. First define $C_m = [C_m^a\ C_m^w] \in \mathbb{R}^{q_m \times p_m}$ by horizontally concatenating C_m^a and C_m^w. Then, we collect the C_m into a block-diagonal matrix as follows:

$$C = \begin{bmatrix} C_1 & \mathbf{0} \\ \mathbf{0} & C_2 \end{bmatrix} \in \mathbb{R}^{q \times p} \tag{3.12}$$

Similarly, define

$$R = \begin{bmatrix} R_1 & \mathbf{0} \\ \mathbf{0} & R_2 \end{bmatrix} \in \mathbb{R}^{q \times q}, \tag{3.13}$$

$$\mathbf{d} = \begin{bmatrix} \mathbf{d}_1 \\ \mathbf{d}_2 \end{bmatrix} \in \mathbb{R}^q \tag{3.14}$$

We can then write the DLAG observation model compactly as follows:

$$\mathbf{y}_t \mid \mathbf{x}_t \sim \mathcal{N}(C\mathbf{x}_t + \mathbf{d}, R) \tag{3.15}$$

The observation model expressed in equation (3.15) defines a distribution for neural activity at a single time point, but to properly fit the DLAG model, we must consider the distribution over all time points. Thus we define $\bar{\mathbf{y}} = [\mathbf{y}_1^\top \cdots \mathbf{y}_T^\top]^\top \in \mathbb{R}^{qT}$ and $\bar{\mathbf{x}} = [\mathbf{x}_1^\top \cdots \mathbf{x}_T^\top]^\top \in \mathbb{R}^{pT}$, obtained by vertically concatenating the observed variables $\mathbf{y}_t$ and latent variables $\mathbf{x}_t$, respectively, across all $t = 1, \ldots, T$. Then, we rewrite the

state and observation models as follows:

$$\bar{\mathbf{x}} \sim \mathcal{N}(\mathbf{0}, \bar{K}) \tag{3.16}$$

$$\bar{\mathbf{y}} \mid \bar{\mathbf{x}} \sim \mathcal{N}(\bar{C}\bar{\mathbf{x}} + \bar{\mathbf{d}}, \bar{R}), \tag{3.17}$$

where $\bar{C} \in \mathbb{R}^{qT \times pT}$ and $\bar{R} \in \mathbb{S}^{qT \times qT}$ are block diagonal matrices comprising T copies of the matrices C and R, respectively. $\bar{\mathbf{d}} \in \mathbb{R}^{qT}$ is constructed by vertically concatenating T copies of $\mathbf{d}$. The elements of $\bar{K} \in \mathbb{R}^{pT \times pT}$ are computed using equations (3.3)–(3.8). Then, the joint distribution over observed and latent variables is given by

$$\begin{bmatrix} \bar{\mathbf{x}} \\ \bar{\mathbf{y}} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mathbf{0} \\ \bar{\mathbf{d}} \end{bmatrix}, \begin{bmatrix} \bar{K} & \bar{K}\bar{C}^\top \\ \bar{C}\bar{K} & \bar{C}\bar{K}\bar{C}^\top + \bar{R} \end{bmatrix} \right) \tag{3.18}$$

E-step In the E-step, our goal is to compute the posterior distribution of the latent variables $\bar{\mathbf{x}}$ given the recorded neural activity $\bar{\mathbf{y}}$, $P(\bar{\mathbf{x}}|\bar{\mathbf{y}})$, using the most recent parameter estimates θ. Using basic results of conditioning for jointly Gaussian random variables, we get

$$\bar{\mathbf{x}} \mid \bar{\mathbf{y}} \sim \mathcal{N}\left(\bar{K}\bar{C}^\top \left(\bar{C}\bar{K}\bar{C}^\top + \bar{R} \right)^{-1} \left(\bar{\mathbf{y}} - \bar{\mathbf{d}} \right), \bar{K} - \bar{K}\bar{C}^\top \left(\bar{C}\bar{K}\bar{C}^\top + \bar{R} \right)^{-1} \bar{C}\bar{K} \right) \tag{3.19}$$

Thus, posterior estimates of latent variables are given by

$$\mathbb{E}\left[\bar{\mathbf{x}}|\bar{\mathbf{y}}\right] = \bar{K}\bar{C}^\top \left(\bar{C}\bar{K}\bar{C}^\top + \bar{R} \right)^{-1} \left(\bar{\mathbf{y}} - \bar{\mathbf{d}} \right) \tag{3.20}$$

The marginal likelihood of the observed neural activity can be computed as

$$\bar{\mathbf{y}} \sim \mathcal{N}\left(\bar{\mathbf{d}}, \bar{C}\bar{K}\bar{C}^\top + \bar{R} \right) \tag{3.21}$$

M-step In the M-step, our goal is to maximize $\mathcal{E}(\theta) = \mathbb{E}\left[\log P(\bar{\mathbf{x}}, \bar{\mathbf{y}})|\theta\right]$ with respect to θ, using the latest inference of the latent variables, computed in the E-step. As in [47, 54], we adopt the following notation. Given a vector $\mathbf{v}$,

$$\langle \mathbf{v} \rangle = \mathbb{E}\left[\mathbf{v}|\bar{\mathbf{y}}\right] \tag{3.22}$$

$$\langle \mathbf{v}\mathbf{v}^\top \rangle = \mathbb{E}\left[\mathbf{v}\mathbf{v}^\top|\bar{\mathbf{y}}\right] \tag{3.23}$$

The appropriate expectations can be found using equation (3.19).

Maximizing $\mathcal{E}(\theta)$ with respect to C, $\mathbf{d}$ yields the following closed-form update for the m^{th} brain area:

$$\begin{bmatrix} C_m & \mathbf{d}_m \end{bmatrix} = \left(\sum_{t=1}^{T} \mathbf{y}_{m,t} \cdot \begin{bmatrix} \langle \mathbf{x}_{m,t} \rangle^\top & 1 \end{bmatrix} \right) \left(\sum_{t=1}^{T} \begin{bmatrix} \langle \mathbf{x}_{m,t}\mathbf{x}_{m,t}^\top \rangle & \langle \mathbf{x}_{m,t} \rangle \\ \langle \mathbf{x}_{m,t} \rangle^\top & 1 \end{bmatrix} \right)^{-1} \tag{3.24}$$

After performing the update for each area separately, we collect all updated values into C and $\mathbf{d}$. Then we update R for both brain areas together, as follows:

$$R = \frac{1}{T}\text{diag}\left\{\sum_{t=1}^{T}\left((\mathbf{y}_t - \mathbf{d})(\mathbf{y}_t - \mathbf{d})^\top - (\mathbf{y}_t - \mathbf{d})\langle\mathbf{x}_t\rangle^\top C^\top - C\langle\mathbf{x}_t\rangle(\mathbf{y}_t - \mathbf{d})^\top + C\langle\mathbf{x}_t\mathbf{x}_t^\top\rangle C^\top\right)\right\} \tag{3.25}$$

There are no closed-form solutions for the Gaussian process parameter updates, but we can compute gradients and perform gradient ascent. Note that, for this work, we choose not to fit the Gaussian process noise variances, but rather, we set them to small values (10^{-3}), as in [47]. Within-area timescale gradients for the m^{th} brain area and j^{th} within-area latent variable are given by

$$\frac{\partial\mathcal{E}(\theta)}{\partial\tau^w_{m,j}} = \text{tr}\left(\left(\frac{\partial\mathcal{E}(\theta)}{\partial K^w_{m,j}}\right)^\top\left(\frac{\partial K^w_{m,j}}{\partial\tau^w_{m,j}}\right)\right) \tag{3.26}$$

where

$$\frac{\partial\mathcal{E}(\theta)}{\partial K^w_{m,j}} = -\frac{1}{2}(K^w_{m,j})^{-1} + \frac{1}{2}\left((K^w_{m,j})^{-1}\langle\mathbf{x}^w_{m,j,:}\mathbf{x}^{w\top}_{m,j,:}\rangle(K^w_{m,j})^{-1}\right) \tag{3.27}$$

and element (t_1, t_2) of $\partial K^w_{m,j}/\partial\tau^w_{m,j}$ is given by

$$\frac{\partial k^w_{m,j}(t_1,t_2)}{\partial\tau^w_{m,j}} = \left(1 - (\sigma^w_{m,j})^2\right)\frac{(t_2 - t_1)^2}{(\tau^w_{m,j})^3}\exp\left(-\frac{(t_2 - t_1)^2}{2(\tau^w_{m,j})^2}\right) \tag{3.28}$$

To express the across-area timescale and delay parameter gradients, we introduce more compact notation for the variables in equation (3.6). Let $\mathbf{x}^a_{j,:} = [\mathbf{x}^{a\top}_{1,j,:}\mathbf{x}^{a\top}_{2,j,:}]^\top \in \mathbb{R}^{2T}$ for the j^{th} across-area latent variable, and

$$K^a_j = \begin{bmatrix} K^a_{1,1,j} & K^a_{1,2,j} \\ K^a_{2,1,j} & K^a_{2,2,j} \end{bmatrix} \in \mathbb{S}^{2T\times 2T} \tag{3.29}$$

Then, across-area timescale gradients are given by

$$\frac{\partial\mathcal{E}(\theta)}{\partial\tau^a_j} = \text{tr}\left(\left(\frac{\partial\mathcal{E}(\theta)}{\partial K^a_j}\right)^\top\left(\frac{\partial K^a_j}{\partial\tau^a_j}\right)\right) \tag{3.30}$$

where

$$\frac{\partial\mathcal{E}(\theta)}{\partial K^a_j} = -\frac{1}{2}(K^a_j)^{-1} + \frac{1}{2}\left((K^a_j)^{-1}\langle\mathbf{x}^a_{j,:}\mathbf{x}^{a\top}_{j,:}\rangle(K^a_j)^{-1}\right) \tag{3.31}$$

and each element of $\partial K^a_j/\partial\tau^a_j$ is given by

$$\frac{\partial k^a_{m_1,m_2,j}(t_1,t_2)}{\partial\tau^a_j} = \left(1 - (\sigma^a_j)^2\right)\frac{(\Delta t)^2}{(\tau^a_j)^3}\exp\left(-\frac{(\Delta t)^2}{2(\tau^a_j)^2}\right) \tag{3.32}$$

where Δt is defined as in equation (3.8). To optimize the timescales while respecting non-negativity constraints, we perform a change of variables, and then perform unconstrained gradient ascent with respect to $\log\tau^w_{m,j}$ or $\log\tau^a_j$.

Next, delay gradients for brain area m and across-area latent variable j are given by

$$\frac{\partial \mathcal{E}(\theta)}{\partial D_{m,j}} = \mathrm{tr}\left(\left(\frac{\partial \mathcal{E}(\theta)}{\partial K_j^a}\right)^\top \left(\frac{\partial K_j^a}{\partial D_{m,j}}\right)\right) \tag{3.33}$$

where $\frac{\partial \mathcal{E}(\theta)}{\partial K_j^a}$ is defined as in equation (3.31), and each element of $\partial K_j^a / \partial D_{m,j}$ is given by

$$\frac{\partial k^a_{m_1,m_2,j}(t_1,t_2)}{\partial D_{m,j}} = \left(1-(\sigma_j^a)^2\right)\frac{\Delta t}{(\tau_j^a)^2}\exp\left(-\frac{(\Delta t)^2}{2(\tau_j^a)^2}\right)\frac{\partial\,(\Delta t)}{\partial D_{m,j}} \tag{3.34}$$

$$\frac{\partial(\Delta t)}{D_{m,j}} = \begin{cases} 1 & \text{if} \quad m = m_2 \\ -1 & \text{if} \quad m = m_1 \end{cases} \tag{3.35}$$

where Δt, m_1, and m_2 are defined as in equation (3.8). In practice, we fix all delay parameters for area 1 at 0 to ensure identifiability. As with the timescales, one might wish to constrain the delays within some physically realistic range, such as the length of an experimental trial, so that $-D_{\max} \leq D_{m,j} \leq D_{\max}$. Toward that end, we make the change of variables $D_{m,j} = D_{\max}\frac{1-e^{-D^*_{m,j}}}{1+e^{-D^*_{m,j}}}$ and perform unconstrained gradient ascent with respect to $D^*_{m,j}$. Here we chose $D_{\max}$ to be half the length of a trial. No delays came close to these constraints in our results (Fig. 5.3, Fig. 5.4).

Finally, note that all of these EM updates are derived for a single sequence, or trial. It is straightforward to extend these equations to N independent trials (each with a potentially different number of time steps, T) by maximizing $\frac{\partial}{\partial\theta}\left[\sum_{n=1}^N \mathcal{E}_n(\theta)\right]$, where trial is indexed by $n = 1, \ldots, N$.

Parameter initialization To initialize the DLAG observation model parameters to reasonable values prior to fitting with the EM algorithm, we first fit a probabilistic canonical correlation analysis (pCCA)[52] model to the neural activity, with the same number of across-area latent variables as the desired DLAG model (see Section 3.6). pCCA is defined by the following state and observation models:

$$\mathbf{x}_t^a \sim \mathcal{N}(\mathbf{0}, I) \tag{3.36}$$

$$\mathbf{y}_{m,t} \mid \mathbf{x}_t^a \sim \mathcal{N}(C_m^a \mathbf{x}_t^a + \mathbf{d}_m, R_m) \tag{3.37}$$

where $C_m^a \in \mathbb{R}^{q_m \times p^a}$ maps the p^a-dimensional across-area latent variables $\mathbf{x}_t^a \in \mathbb{R}^{p^a}$ to the neural activity of area m, $\mathbf{d}_m \in \mathbb{R}^{q_m}$ is a mean parameter, and $R_m \in \mathbb{S}^{q_m \times q_m}$ is the observation noise covariance matrix. R_m is not constrained to be diagonal. The fitted values for C_m^a and $\mathbf{d}_m$ are used as initial values for their DLAG analogues. We take only the diagonal elements of R_m to initialize its DLAG analogue.

pCCA does not incorporate within-area latent variables. Therefore, we initialized each DLAG within-area loading matrix C_m^w so that its columns spanned a subspace uncorrelated with that spanned by the columns of C_m^a, returned by pCCA. Such a subspace can be computed as follows. Let $\Sigma_m \in \mathbb{S}^{q_m \times q_m}$ be the

sample covariance matrix of activity in area m. Then define $W_m = C_m^{a\top} \Sigma_m \in \mathbb{R}^{p^a \times q_m}$. The singular value decomposition of W_m is given by $W_m = U_m S_m V_m^\top$, where $U_m \in \mathbb{R}^{p^a \times p^a}$, $S_m \in \mathbb{R}^{p^a \times q_m}$, and $V_m \in \mathbb{R}^{q_m \times q_m}$. The first p^a columns of V_m span the same across-area subspace spanned by the columns of C_m^a. The remaining $q_m - p^a$ columns form an orthonormal basis for the subspace uncorrelated with this across-area subspace. We initialized C_m^w with the first p_m^w of these uncorrelated basis vectors. Finally, we initialized all delays to zero, and all within- and across-area Gaussian process timescales to the same value, equal to twice the sampling period or spike count bin width of the neural activity.

3.6 Selecting the number of within- and across-area latent variables

DLAG has three hyperparameters: p^a, the number of across-area latent variables; and p_1^w and p_2^w, the number of within-area latent variables for each area. Model selection therefore poses a significant scaling challenge. Grid search over even a small range of within- and across-area dimensionalities can result in a large number of models that need to be fitted and validated. For example, considering just 10 possibilities for each type of latent variable would result in 1,000 candidate models. Thus, exhaustive search for the optimal DLAG model is impractical.

We therefore developed a streamlined cross-validation procedure that significantly improves scalability. In brief, our model selection procedure occurs in two stages. First, we consider each area separately, and—using factor analysis (FA)[48]—we find the number of latent variables needed to explain the shared variance among neurons within each area. We reasoned that, while there is not a direct correspondence between the optimal number of latent variables in DLAG and FA models (because of temporal smoothing and other differences in model structure), it is unlikely that the total number of within- and across-area latent variables extracted by DLAG will exceed the FA dimensionality for an area (such a case would imply that there exists a neuron in, for example, area A that covaries with one or more neurons in area B, but no other neurons in area A). Hence we believe this approach to be reasonable given the significant computational benefits. We then use the FA dimensionality in each area to reduce the space of DLAG model candidates to a practical size.

In greater detail, we first applied FA to each area independently, and identified the optimal FA dimensionality through K-fold cross-validation (here we chose $K = 4$). We randomly split all trials into K equally sized partitions. For the k^{th} cross-validation fold ($k = 1, \ldots, K$), we held out the k^{th} partition of trials and fit FA model parameters to the trials in the remaining $K - 1$ partitions. Using the fitted parameters, we evaluated the data log-likelihood on the held-out trials. We repeated this procedure for each of the K folds and summed the held-out data log-likelihoods computed for each fold. We refer to this value as the

cross-validated data (log)-likelihood. The FA model with the highest cross-validated data likelihood was taken as "optimal."

We then used the optimal FA dimensionalities ($p_m^{FA}, m = 1, 2$) to constrain the space of DLAG model candidates. In particular, we consider only DLAG models that satisfy $p^a + p_m^w = p_m^{FA}$, for $m = 1, 2$; and $p^a \leq \min(p_1^{FA}, p_2^{FA})$. In words, we consider only DLAG models such that the number of within- and across-area latent variables in each area sum to that area's optimal FA dimensionality. Furthermore, the number of across-area latent variables is limited by the area with the smallest optimal FA dimensionality. Not only does this streamlined cross-validation approach provide an upper limit on the possible number of within- and across-area latent variables, it also effectively collapses the DLAG hyperparameter space from three free hyperparameters to one (across-area dimensionality, p^a), drastically improving scalability.

Among the model candidates within this constrained search range, we selected models that exhibited the largest cross-validated data likelihood, using the same K-fold cross-validation scheme as for FA. For each of the K folds, we evaluated (the log of) equation (3.21) on held-out trials using DLAG model parameters fit to all remaining trials. We then took the cross-validated data log-likelihood to be the sum (across the K folds) of held-out data log-likelihoods. To further reduce runtime, we limited the number of EM iterations during cross-validation to 1,000. The optimal DLAG model was then re-fit to full convergence, where the data log-likelihood improved from one iteration to the next by less than a preset tolerance (here we used 10^{-8}).

We also note that throughout this work, we explicitly considered model candidates for which across-area dimensionality was zero ($p^a = 0$): the two areas are independent, and any correlations between neurons are purely within-area. Similarly, we explicitly considered model candidates for which within-area dimensionalities were zero ($p_1^w = 0$ or $p_2^w = 0$): all variance shared among neurons in one area is attributed to their interactions with neurons in the other area. The case where all dimensionalities are zero ($p^a = p_1^w = p_2^w = 0$) is equivalent to fitting a multivariate Gaussian distribution to the data with diagonal covariance (i.e., all neurons are treated as independent). We similarly considered zero-dimensional FA models ($p_1^{FA} = 0$ or $p_2^{FA} = 0$) during the first stage of our model selection procedure, equivalent to fitting a multivariate Gaussian distribution with diagonal covariance to observations in the respective area. The inclusion of these zero-dimensionality model candidates protects against the identification of spurious interactions across or within areas.

3.7 Statistical tradeoffs between within- and across-area latent variables

Thus far, we have described how DLAG decomposes observed neural activity into a linear combination of within- and across-area latent variables. Equivalently, DLAG partitions each area's population space into distinct within- and across-area subspaces, which represent characteristic ways in which the neurons covary (Fig. 3.1). Here we investigate more deeply why the within-area latent variables are a necessary model component, even if across-area activity is of primary scientific interest. Toward that end, we will consider an alternative interpretational perspective: namely, that DLAG performs a low-rank decomposition of the covariance matrix of a time series. This alternative perspective also illuminates a general statistical phenomenon—not specific to DLAG—that any multi-area time series method must consider.

DLAG performs a low-rank covariance decomposition

Let us first express the DLAG model not only for a single time point, as in equation (3.15), but for all time points in a sequence. In particular, we will collect observed and latent variables in a manner that highlights group structure (i.e., organized differently than in equations (3.16) and (3.17)). We define $\widetilde{\mathbf{y}}_1 = [\mathbf{y}_{1,1}^\top \cdots \mathbf{y}_{1,T}^\top]^\top \in \mathbb{R}^{q_1 T}$ and $\widetilde{\mathbf{y}}_2 = [\mathbf{y}_{2,1}^\top \cdots \mathbf{y}_{2,T}^\top]^\top \in \mathbb{R}^{q_2 T}$, obtained by vertically concatenating the observed neural activity $\mathbf{y}_{1,t}$ and $\mathbf{y}_{2,t}$ in areas 1 and 2, respectively, across all times $t = 1, \ldots, T$. We collect the across- and within-area latent variables for each area similarly. Let $\widetilde{\mathbf{x}}_1^a = [\mathbf{x}_{1,:,1}^{a\top} \cdots \mathbf{x}_{1,:,T}^{a\top}]^\top \in \mathbb{R}^{p^a T}$, $\widetilde{\mathbf{x}}_1^w = [\mathbf{x}_{1,:,1}^{w\top} \cdots \mathbf{x}_{1,:,T}^{w\top}]^\top \in \mathbb{R}^{p_1^w T}$, $\widetilde{\mathbf{x}}_2^a = [\mathbf{x}_{2,:,1}^{a\top} \cdots \mathbf{x}_{2,:,T}^{a\top}]^\top \in \mathbb{R}^{p^a T}$, and $\widetilde{\mathbf{x}}_2^w = [\mathbf{x}_{2,:,1}^{w\top} \cdots \mathbf{x}_{2,:,T}^{w\top}]^\top \in \mathbb{R}^{p_2^w T}$.

Then, we rewrite the state and observation models as follows:

$$\begin{bmatrix} \widetilde{\mathbf{x}}_1^a \\ \widetilde{\mathbf{x}}_1^w \\ \widetilde{\mathbf{x}}_2^a \\ \widetilde{\mathbf{x}}_2^w \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \begin{bmatrix} \widetilde{K}_{1,1}^a & \mathbf{0} & \widetilde{K}_{1,2}^a & \mathbf{0} \\ \mathbf{0} & \widetilde{K}_1^w & \mathbf{0} & \mathbf{0} \\ \widetilde{K}_{2,1}^a & \mathbf{0} & \widetilde{K}_{2,2}^a & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \widetilde{K}_2^w \end{bmatrix} \right) \tag{3.38}$$

$$\begin{bmatrix} \widetilde{\mathbf{y}}_1 \\ \widetilde{\mathbf{y}}_2 \end{bmatrix} \mid \begin{bmatrix} \widetilde{\mathbf{x}}_1^a \\ \widetilde{\mathbf{x}}_1^w \\ \widetilde{\mathbf{x}}_2^a \\ \widetilde{\mathbf{x}}_2^w \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \widetilde{C}_1^a & \widetilde{C}_1^w & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \widetilde{C}_2^a & \widetilde{C}_2^w \end{bmatrix} \begin{bmatrix} \widetilde{\mathbf{x}}_1^a \\ \widetilde{\mathbf{x}}_1^w \\ \widetilde{\mathbf{x}}_2^a \\ \widetilde{\mathbf{x}}_2^w \end{bmatrix} + \begin{bmatrix} \widetilde{\mathbf{d}}_1 \\ \widetilde{\mathbf{d}}_2 \end{bmatrix}, \begin{bmatrix} \widetilde{R}_1 & \mathbf{0} \\ \mathbf{0} & \widetilde{R}_2 \end{bmatrix} \right) \tag{3.39}$$

where $\widetilde{C}_1^a \in \mathbb{R}^{q_1 T \times p^a T}$, $\widetilde{C}_1^w \in \mathbb{R}^{q_1 T \times p_1^w T}$, $\widetilde{C}_2^a \in \mathbb{R}^{q_2 T \times p^a T}$, $\widetilde{C}_2^w \in \mathbb{R}^{q_2 T \times p_2^w T}$, $\widetilde{R}_1 \in \mathbb{S}^{q_1 T \times q_1 T}$, and $\widetilde{R}_2 \in \mathbb{S}^{q_2 T \times q_2 T}$ are all block diagonal matrices comprising T copies of the loading matrices C_1^a, C_1^w, C_2^a, and C_2^w, and observation noise covariance matrices R_1 and R_2, respectively. $\widetilde{\mathbf{d}}_1 \in \mathbb{R}^{q_1 T}$ and $\widetilde{\mathbf{d}}_2 \in \mathbb{R}^{q_2 T}$ are constructed by vertically concatenating T copies of mean parameters $\mathbf{d}_1$ and $\mathbf{d}_2$, respectively. Note that equations (3.38) and (3.39) above are equivalent to equations (3.16) and (3.17), but with variables rearranged.

Each within-area covariance matrix $\widetilde{K}^w_m \in \mathbb{S}^{p^w_m T \times p^w_m T}$, for area $m = 1, 2$ has the following block structure:

$$\widetilde{K}^w_m = \begin{bmatrix} \widetilde{K}^w_m(1,1) & \cdots & \widetilde{K}^w_m(1,T) \\ \vdots & \ddots & \vdots \\ \widetilde{K}^w_m(T,1) & \cdots & \widetilde{K}^w_m(T,T) \end{bmatrix} \tag{3.40}$$

where each block $\widetilde{K}^w_m(t_1,t_2) = \mathrm{diag}(k^w_{m,1}(t_1,t_2),\ldots,k^w_{m,p^w_m}(t_1,t_2)) \in \mathbb{S}^{p^w_m \times p^w_m}, t_1, t_2 \in \{1,\ldots,T\}$ is a diagonal matrix whose elements are computed according to the covariance function defined in equations (3.4) and (3.5).

Each across-area auto- or cross-covariance matrix $\widetilde{K}^a_{m_1,m_2} \in \mathbb{R}^{p^a T \times p^a T}$, for areas $m_1, m_2 \in \{1,2\}$ has analogous structure:

$$\widetilde{K}^a_{m_1,m_2} = \begin{bmatrix} \widetilde{K}^a_{m_1,m_2}(1,1) & \cdots & \widetilde{K}^a_{m_1,m_2}(1,T) \\ \vdots & \ddots & \vdots \\ \widetilde{K}^a_{m_1,m_2}(T,1) & \cdots & \widetilde{K}^a_{m_1,m_2}(T,T) \end{bmatrix} \tag{3.41}$$

where each block $\widetilde{K}^a_{m_1,m_2}(t_1,t_2) = \mathrm{diag}(k^a_{m_1,m_2,1}(t_1,t_2),\ldots,k^a_{m_1,m_2,p^a}(t_1,t_2)) \in \mathbb{S}^{p^a \times p^a}, t_1, t_2 \in \{1,\ldots,T\}$ is a diagonal matrix whose elements are computed according to the covariance function defined in equations (3.7) and (3.8). Note that the cross-covariance matrices are transposes of one another, i.e., $\widetilde{K}^a_{m_1,m_2} = \widetilde{K}^{a\top}_{m_2,m_1}$.

Upon inspection of equation (3.38), the statistical dependency between latent variables becomes clear. However, the statistical dependency between observed neural activity in each area, $\widetilde{\mathbf{y}}_1$ and $\widetilde{\mathbf{y}}_2$, is not obvious, since the structure of equation (3.39) suggests that they might be decoupled. The relationship between observed areas becomes clear when we consider their joint distribution, after marginalizing out the latent variables:

$$\begin{bmatrix} \widetilde{\mathbf{y}}_1 \\ \widetilde{\mathbf{y}}_2 \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \widetilde{\mathbf{d}}_1 \\ \widetilde{\mathbf{d}}_2 \end{bmatrix}, \widetilde{\Sigma} \right) \tag{3.42}$$

where

$$\widetilde{\Sigma} = \begin{bmatrix} \widetilde{C}^a_1 \widetilde{K}^a_{1,1} \widetilde{C}^{a\top}_1 + \widetilde{C}^w_1 \widetilde{K}^w_1 \widetilde{C}^{w\top}_1 + \widetilde{R}_1 & \widetilde{C}^a_1 \widetilde{K}^a_{1,2} \widetilde{C}^{a\top}_2 \\ \widetilde{C}^a_2 \widetilde{K}^a_{2,1} \widetilde{C}^{a\top}_1 & \widetilde{C}^a_2 \widetilde{K}^a_{2,2} \widetilde{C}^{a\top}_2 + \widetilde{C}^w_2 \widetilde{K}^w_2 \widetilde{C}^{w\top}_2 + \widetilde{R}_2 \end{bmatrix} \tag{3.43}$$

Equation (3.43) makes explicit the alternative interpretational perspective of DLAG: DLAG performs a low-rank decomposition of the covariance matrix $\widetilde{\Sigma}$. This decomposition is illustrated graphically in Fig. 3.4a. For simplicity, we illustrate a covariance matrix for areas with three neurons each, over two time points. The shading of blocks of the covariance matrix illustrate which type of DLAG parameter is responsible for explaining that particular portion of covariance (magenta: across-area; blue/red: within-area; gray: independent single-neuron variability). Regions of overlap (i.e., where both blue/magenta or red/magenta shading are present) illustrate portions of covariance that both within- and across-area

variables are responsible for explaining. Any regions of white indicate that no model parameters explain that portion of covariance.

The across-area parameters (note the fully magenta-shaded across-area covariance component in Fig. 3.4a) serve to explain covariance among all neurons, in both areas. Within-area parameters (blue and red shading, for areas 1 and 2, respectively) serve to explain covariance among neurons within each area, but not across areas (note the white across-area blocks for the within-area covariance component). Importantly, the only parameters in the DLAG model capable of explaining covariance across areas are the across-area parameters (only magenta shading is present in the across-area blocks of $\widetilde{\Sigma}$). And interestingly, within-area components fully overlap across-area components in the within-area blocks of $\widetilde{\Sigma}$, suggesting a potential redundancy. However, as we will discuss below, the overall structure of the decomposition shown in Fig. 3.4a is critical to the interpretation of across-area variables—that they isolate neural interactions *across* areas (and minimally reflect purely within-area interactions).

A time series within-area model must accompany a time series across-area model

To build further intuition, let us consider the scenario where within- and across-area covariances are modeled statically—without considering the flow of time (Fig. 3.4b). Static covariance decompositions result, for example, from the probabilistic canonical correlation analysis (pCCA) model[52], which includes static across-area latent variables and no within-area latent variables (within-area covariance is instead captured using full observation noise covariance matrices, R_1 and R_2). The covariance matrix $\widetilde{\Sigma}$ still decomposes into across- and within-area components; however, covariances at non-zero time lags (i.e., the covariance between neural activity at a time point t_1 and a different time point $t_2 \neq t_1$, indicated by the white-shaded blocks of $\widetilde{\Sigma}$ in Fig. 3.4b) are all zero, by definition. Just like the DLAG case (Fig. 3.4a), only the across-area parameters can explain across-area covariance, and within-area components fully overlap across-area components in the within-area blocks of $\widetilde{\Sigma}$ (to understand why this covariance structure is important, see case below). Across-area activity is successfully isolated by across-area variables.

The problematic case arises when we use a time series model to describe across-area interactions, but use a static model to describe within-area interactions (Fig. 3.4c). For example, what if we proposed a version of DLAG that simply adopted the same observation model as pCCA (i.e., full observation noise covariance matrices, R_1 and R_2) to model within-area interactions? In this case, although the within-area model components do explain covariance among neurons within each area, they fail to capture any within-area covariance across time points, by definition. This shortcoming forces the across-area variables to explain within-area covariance across time points. Visually, all within-area blocks of the covariance matrix $\widetilde{\Sigma}$ representing relationships across time points have solely magenta shading (these problematic

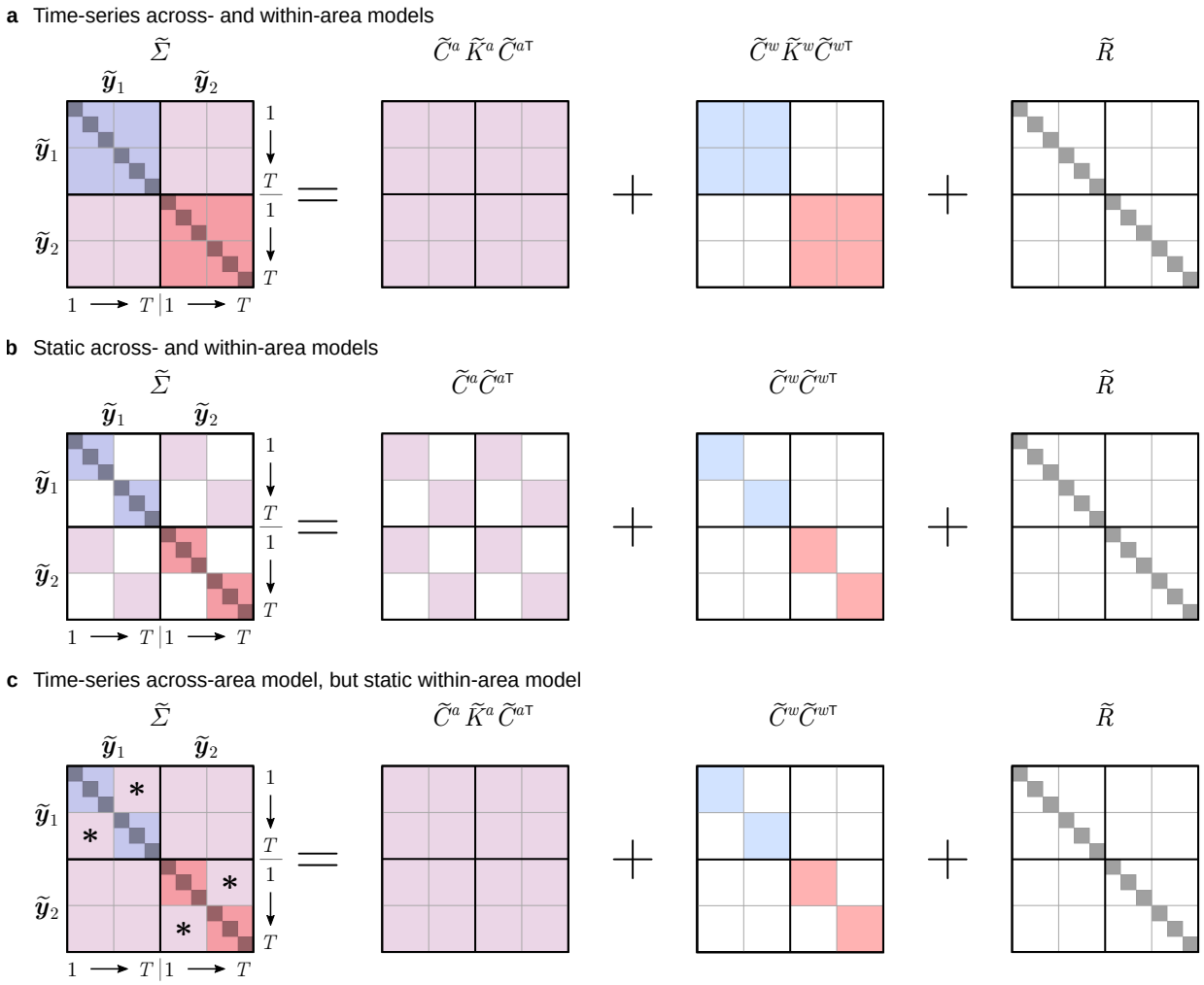

Figure 3.4: Full-sequence (trial) covariance matrix decompositions. For simplicity, in (a)-(c), we illustrate a covariance matrix for areas with three neurons each, over two time points. From left to right, panels represent the overall covariance matrix, its across-area component, its within-area component, and a component representing variance independent to each neuron. Across-area parameters (magenta shading) are solely responsible for explaining across-area covariance over time (i.e., there is no overlap of magenta with blue, red, or gray in the across-area off-diagonal blocks of the overall covariance matrix, on the left). (**a**) DLAG decomposes the covariance of a full sequence (trial) into low-rank components. Covariance among neurons within an area that cannot be explained by across-area covariance is captured by within-area parameters (area A: blue; area B: red). (**b**) Models such as probabilistic canonical correlation analysis (pCCA), for example, similarly decompose the overall covariance matrix into across- and within-area components, but make no attempt to model covariance across time points, either across or within areas (indicated by blocks with white shading). (**c**) If one is using a time series across-area model, then in the absence of a time series within-area model, across-area parameters are forced to explain within-area covariance over time. This problem is illustrated by the within-area blocks of the overall covariance matrix that have only magenta shading (indicated by the '*' symbols).

blocks are highlighted by the '*' symbols in Fig. 3.4c). In contrast, the true DLAG model and fully static models avoid this pitfall. These successful models (Fig. 3.4a,b) do not have any blocks of $\widetilde{\Sigma}$ for which across-area parameters are solely responsible for explaining within-area covariance. This statistical

phenomenon applies to any multi-area time series method, and is not specific to DLAG[32,56].

Chapter 4

Validating DLAG in simulation

Before applying DLAG to experimental data, it was critical to characterize its performance in simulated experiments in which the ground truth was known. In this chapter, we first demonstrate that DLAG performs well on synthetic datasets similar in scale to state-of-the-art neurophysiological recordings from multiple brain areas (Section 4.1). Then, we consider additional datasets covering a wider range of experimental conditions, and characterize both DLAG's performance and runtime (Section 4.2). We also consider more challenging synthetic scenarios to demonstrate DLAG's robustness to mild deviations from its modeling assumptions (Section 4.3). Finally, we demonstrate that DLAG disentangles concurrent signaling where existing methods like CCA cannot (Section 4.4).

4.1 Validation on realistic-scale synthetic data

We first characterized DLAG's performance on synthetic datasets similar in scale to state-of-the-art neurophysiological recordings from multiple brain areas. In brief, informed by our recordings in macaque V1 and V2[26,37] (see Section 5.1), we simulated independent datasets with representative numbers of neurons (area A: 80; area B: 20), trial counts (100), trial lengths (1,000 ms), and levels of noise, where noise is defined as the variance independent to each neuron.

4.1.1 Simulating data from the DLAG generative model

In greater detail, we generated synthetic datasets according to the DLAG generative model, so that we could leverage known ground truth to evaluate the accuracy of estimates. We started by randomly generating the set of model parameters, θ (equation (3.9)), subject to constraints informed by experimental data. For all datasets, we chose the numbers of neurons in each area based on our V1-V2 recordings (area A: $q_1 = 80$; area B: $q_2 = 20$). We set the combined total dimensionality in each area to representative

values (area A: $p^a + p_1^w = 10$; area B: $p^a + p_2^w = 5$), but varied the relative number of within- and across-area latent variables across datasets. Generating 20 datasets at each of six configurations ($p^a = 0, \ldots, 5$; $p_1^w = 5, \ldots, 10$; $p_2^w = 0, \ldots, 5$) resulted in a total of 120 independent datasets. Importantly, among these datasets, we included datasets without across- or within-area structure (i.e., datasets for which across- or within-area dimensionality was zero), to test if our framework could identify such cases.

To ensure that synthetic datasets exhibited realistic noise levels, we first evaluated the strength of latent variables relative to the strength of single-neuron variability exhibited in the V1-V2 recordings. Specifically, we computed the "signal-to-noise" ratio (where "signal" is defined as the shared activity described by latent variables), $\mathrm{tr}(C_m C_m^\top)/\mathrm{tr}(R_m)$, for V1 and V2 using the parameters of the optimal DLAG models fit to each V1-V2 dataset. Representative values were 0.3 and 0.2 for V1 and V2, respectively. Then for each dataset, we generated our synthetic observation model parameters, C_m and R_m, as follows. We first drew the elements of C_m and a diagonal matrix $\Phi_m \in \mathbb{R}^{q_m \times q_m}$ from the standard normal distribution $\mathcal{N}(0, 1)$. Then, we set $R_m = \Phi_m \Phi_m^\top$ (so that R_m was a valid covariance matrix) and rescaled R_m such that area m exhibited the correct signal-to-noise ratio. The elements of the mean parameter $\mathbf{d}$ were also drawn from the standard normal distribution.

Finally, we drew all timescales ($\{\tau_j^a\}_{j=1}^{p^a}$, $\{\tau_{1,j}^w\}_{j=1}^{p_1^w}$, $\{\tau_{2,j}^w\}_{j=1}^{p_2^w}$) uniformly from $U(\tau_{\min}, \tau_{\max})$, with $\tau_{\min} =$ 10 ms and $\tau_{\max} = 150$ ms. We drew all delays ($\{D_1, \ldots, D_{p^a}\}$) uniformly from $U(D_{\min}, D_{\max})$, with $D_{\min} = -30$ ms and $D_{\max} = +30$ ms. All Gaussian process noise variances ($\{(\sigma_j^a)^2\}_{j=1}^{p^a}$, $\{(\sigma_{1,j}^w)^2\}_{j=1}^{p_1^w}$, $\{(\sigma_{2,j}^w)^2\}_{j=1}^{p_2^w}$) were fixed at 10^{-3}. With all model parameters specified, we then generated $N = 100$ independent and identically distributed trials ($\bar{\mathbf{x}}_n, \bar{\mathbf{y}}_n$, $n = 1, \ldots, N$) according to equations (3.16) and (3.17). Each trial comprised $T = 50$ time points, corresponding to 1,000 ms sequences sampled with a period of 20 ms, to mimick the 20 ms spike count time bins used to analyze the experimental data.

4.1.2 Synthetic data performance metrics

To quantify DLAG's performance across all synthetic datasets, we employed a variety of metrics. We first consider the estimation of DLAG's observation model parameters. To assess the accuracy of loading matrix estimation (C_m^a, C_m^w; reported in Fig. 4.1, Fig. 4.2, Fig. 4.7), we computed a normalized subspace error[58]:

$$e_{\mathrm{sub}} = \frac{\|(I - \hat{W}(\hat{W}^\top \hat{W})^{-1} \hat{W}^\top) W\|_F}{\|W\|_F} \tag{4.1}$$

where W is the appropriate ground truth parameter, $\hat{W}$ is the corresponding estimate, and $\|\cdot\|_F$ is the Frobenius norm. e_{sub} quantifies the magnitude of the projection of the column space of W onto the null space of $\hat{W}$. A value of 1 indicates that the column space of W lies completely in the null space of $\hat{W}$, and

therefore the estimate captures no component of the ground truth. A value of 0 indicates that the column space of $\hat{W}$ contains the full column space of W, and therefore the estimate captures all components of the ground truth. This metric offers two advantages: (1) it does not require that the columns of W and $\hat{W}$ are ordered in any way (the ordering of DLAG latent variables is arbitrary); and (2) it does not require that W and $\hat{W}$ have the same number of columns, so it can be used to compare the performance of models with different numbers of latent variables. We report the accuracy of loading matrix estimation as $1 - e_{\text{sub}}$ (Fig. 4.1). To assess the accuracy of estimating $\mathbf{d}$ and R (reported in Fig. 4.2, Fig. 4.7), we computed the normalized error

$$e_{\text{vec}} = \frac{\|\mathbf{v} - \hat{\mathbf{v}}\|_2}{\|\mathbf{v}\|_2} \tag{4.2}$$

where $\mathbf{v}$ is either $\mathbf{d}$ or $\text{diag}(R)$, and $\hat{\mathbf{v}}$ is the corresponding estimate.

We next consider the estimation of DLAG's state model parameters. Reporting the accuracy of delay and timescale estimates (Fig. 4.1, Fig. 4.2, Fig. 4.3, Fig. 4.7) required explicitly matching estimated latent variables to the ground truth. Given the large number of synthetic datasets presented here, we automated this matching process as follows. First, for each area m, we took the unordered across- and within-area latent variable estimates, $\hat{\mathbf{x}}_m^a$ and $\hat{\mathbf{x}}_m^w$, and computed the pairwise correlation between each estimated latent variable and each ground truth latent variable, $\mathbf{x}_m^a$ and $\mathbf{x}_m^w$, across all time points and trials. We then reordered the estimated latent variables to match the ground truth latent variables with which they showed the highest magnitude of correlation. To report delay and timescale estimation performance, we computed the absolute error between ground truth and (matched) estimated parameters, to express the error in units of time (ms).

Finally, we consider the moment-by-moment estimation of latent variables. As with the loading matrix, delay, and timescale estimates, quantifying the accuracy of latent variable estimates requires care since the sign and ordering of latent variables is arbitrary and will not, in general, match between estimates and the ground truth. First, let $\widetilde{\mathbf{x}}_m^a = [\mathbf{x}_{m,:,1}^{a\top} \cdots \mathbf{x}_{m,:,T}^{a\top}]^\top \in \mathbb{R}^{p^a T}$ be a collection of all (ground truth) across-area variables at all time points in area m. Similarly, let $\widetilde{\mathbf{x}}_m^w = [\mathbf{x}_{m,:,1}^{w\top} \cdots \mathbf{x}_{m,:,T}^{w\top}]^\top \in \mathbb{R}^{p_m^w T}$ be a collection of all (ground truth) within-area variables at all time points in area m. Finally, define $\widetilde{C}_m^a \in \mathbb{R}^{q_m T \times p^a T}$ and $\widetilde{C}_m^w \in \mathbb{R}^{q_m T \times p_m^w T}$ to be block diagonal matrices comprising T copies of the (ground truth) matrices C_m^a and C_m^w, respectively; and define $\widetilde{\mathbf{d}}_m \in \mathbb{R}^{q_m T}$ by vertically concatenating T copies of (the ground truth) $\mathbf{d}_m$. We'll denote the estimates of each of these values by $\hat{\widetilde{\mathbf{x}}}_m^a$, $\hat{\widetilde{\mathbf{x}}}_m^w$, $\hat{\widetilde{C}}_m^a$, $\hat{\widetilde{C}}_m^w$, and $\hat{\widetilde{\mathbf{d}}}_m$. The estimates $\hat{\widetilde{\mathbf{x}}}_m^a$ and $\hat{\widetilde{\mathbf{x}}}_m^w$ are posterior means, computed according to equation (3.20).

Then, to separate the accuracy of across-area variable estimation from the accuracy of within-area variable estimation (as reported in Fig. 4.1, Fig. 4.2), we estimated denoised (smoothed) observations,

using only across-area or only within-area latent variable estimates:

$$\hat{\tilde{\mathbf{y}}}^*_m = \hat{\tilde{C}}^*_m \hat{\tilde{\mathbf{x}}}^*_m + \hat{\tilde{\mathbf{d}}}_m \tag{4.3}$$

where $\hat{\tilde{\mathbf{y}}}^*_m = [\hat{\mathbf{y}}^{*\top}_{m,1} \cdots \hat{\mathbf{y}}^{*\top}_{m,T}]^\top \in \mathbb{R}^{q_m T}$. Here, the '$*$' symbol is used to indicate either a or w as a superscript, where observations have been denoised using only across- or within-area variable estimates, respectively. We then collect the denoised sequences on all N trials, $\hat{\tilde{\mathbf{y}}}^*_{m,n}$, $n = 1, \ldots, N$, into the matrix $\hat{Y}^*_m = [\hat{\tilde{\mathbf{y}}}^*_{m,1} \cdots \hat{\tilde{\mathbf{y}}}^*_{m,N}] \in \mathbb{R}^{q_m T \times N}$. Analogously, define $Y^*_m \in \mathbb{R}^{q_m T \times N}$ to be the set of ground truth sequences generated prior to adding noise (i.e., the noise term $\boldsymbol{\varepsilon}_m$, defined in equation (3.2)).

We then computed the R^2 value between estimated and (noiseless) ground truth sequences:

$$R^2 = 1 - \frac{\|Y^*_m - \hat{Y}^*_m\|^2_F}{\|Y^*_m - \bar{Y}^*_m\|^2_F} \tag{4.4}$$

where $\bar{Y}^*_m = [\bar{\mathbf{y}}^*_m \cdots \bar{\mathbf{y}}^*_m] \in \mathbb{R}^{q_m T \times N}$ is constructed by horizontally concatenating N copies of the sample mean for each neuron in the ground truth Y^*_m, taken over all time points and trials ($\bar{\mathbf{y}}^*_m \in \mathbb{R}^{q_m T}$). Note that, in the multivariate case, $R^2 \in (-\infty, 1]$, where a negative value implies that estimates predict the ground truth less accurately than simply the sample mean.

4.1.3 Performance

Across all datasets, within- and across-area latent time courses (Fig. 4.1a; see legend for quantification), across-area parameters (Fig. 4.1b, dimensionalities; Fig. 4.1c, delays; Fig. 4.1d, Gaussian process timescales), and within-area parameters (Fig. 4.1e, dimensionalities; Fig. 4.1f,g, Gaussian process timescales) were all consistently and accurately estimated. We highlight, in particular, DLAG's ability to estimate time delays between the two areas (Fig. 4.1c). Delay error was 1.3±0.1 ms (mean and SEM across all delays; max error 7.0 ms), despite observations occurring at 20 ms time steps. This accuracy emphasizes an important feature of the DLAG model that distinguishes it from other time series modeling approaches. Because latent time courses and time delays are continuous-valued, DLAG can leverage the correlated activity of the neuronal populations to recover delays that are smaller than the sampling period (i.e., spike count bin width, in the case of spiking activity). We note also that our streamlined cross-validation procedure proved highly accurate. Across all datasets—including those with no across- or within-area structure—the selected dimensionalities matched the ground truth (Fig. 4.1b, across-area; Fig. 4.1e, within-area).

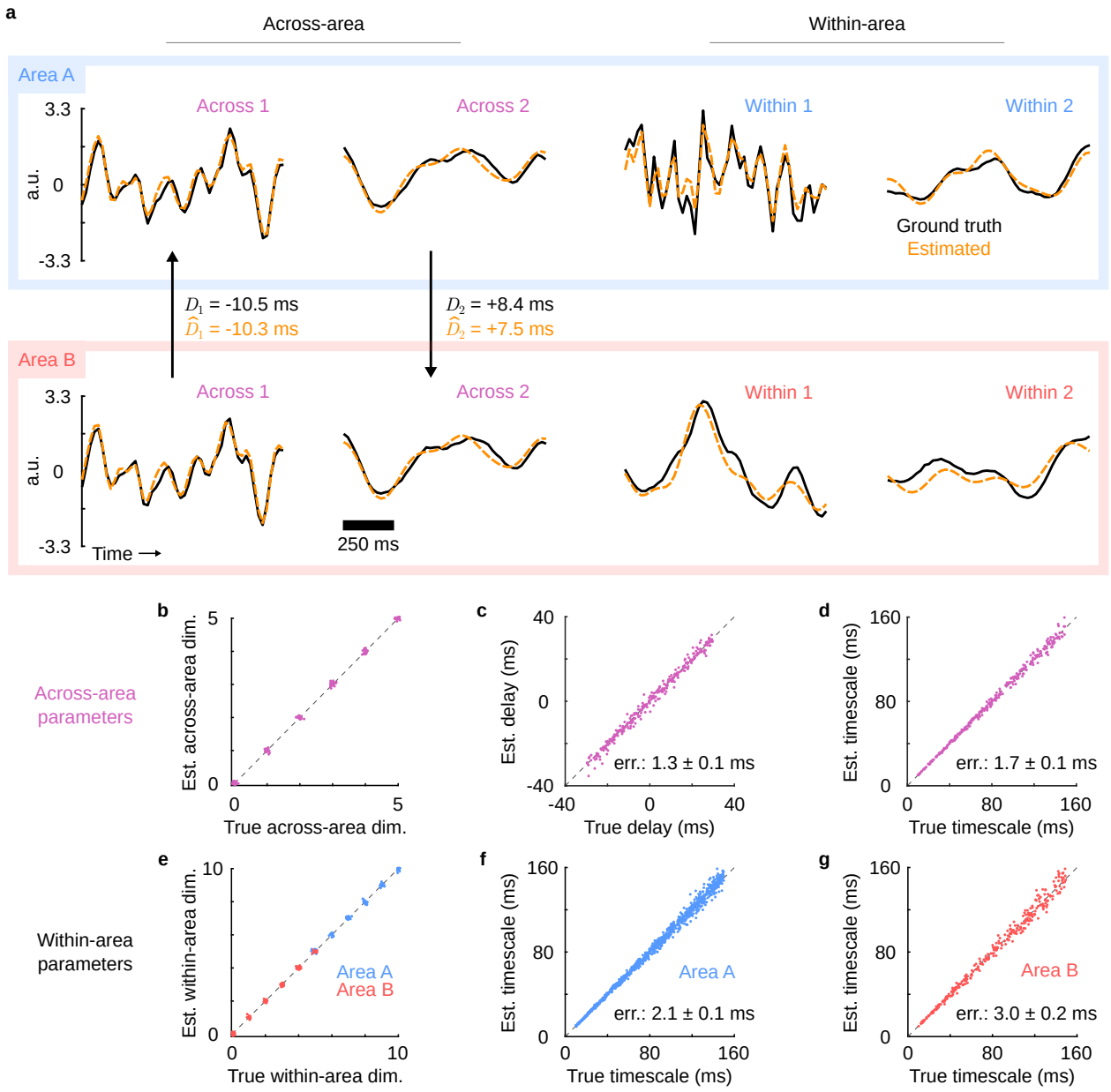

Figure 4.1. DLAG accurately estimates within- and across-area time courses and their parameters in synthetic data. (**a**) Single-trial latent-variable time course estimates for a representative synthetic dataset. Top row / blue box: area A; bottom row / red box: area B. For visual clarity, two latent variables of each type are shown (left: across-area; right: within-area). Orange dashed traces: DLAG estimates; black solid traces: ground truth. a.u.: arbitrary units. Across all synthetic datasets for which across- or within-area dimensionality was non-zero (across: 100 datasets; within A: 120 datasets; within B: 100 datasets), mean accuracy (R^2) of time course estimation was as follows: area A, across – 0.90; area B, across – 0.91; area A, within – 0.88; area B, within – 0.82 (all SEM values less than 0.01). Similarly, mean accuracy of subspace (loading matrix) estimation was as follows: C_1^a – 0.89; C_2^a – 0.93; C_1^w – 0.92; C_2^w – 0.94 (where a value of 1 implies that the ground truth is fully captured by estimates; all SEM values less than 0.01). (**b**) Across-area dimensionality estimates versus the ground truth for all 120 synthetic datasets. Data points are integer-valued, but randomly jittered to show points that overlap. (**c**) Delay estimates versus the ground truth. Displayed error ('err.') indicates mean absolute error and SEM reported across 300 across-area variables. (**d**) Across-area Gaussian process (GP) timescale estimates versus the ground truth. Displayed

error ('err.') indicates mean absolute error and SEM reported across 300 across-area variables. (**e**) Within-area dimensionality estimates versus the ground truth for all 120 synthetic datasets (blue: within-area A; red: within-area B). Data points are integer-valued, but randomly jittered to show points that overlap. (**f**) Within-area A GP timescale estimates versus the ground truth. Displayed error ('err.') indicates mean absolute error and SEM reported across 900 within-area variables in area A. (**g**) Within-area B GP timescale estimates versus the ground truth. Displayed error ('err.') indicates mean absolute error and SEM reported across 300 within-area variables in area B.

4.2 Performance and runtime over a range of simulated conditions

The synthetic datasets presented above were generated with a variety of parameters representative of realistic data, but we also verified that DLAG performed well over a wider range of simulated conditions. Specifically, we systematically characterized DLAG's performance as a function of number of trials, number of neurons, latent dimensionality, and noise level (Fig. 4.2), as well as latent timescale (Fig. 4.3). We also characterized the runtime of the DLAG fitting procedure as a function of number of trials, number of neurons, trial length, and latent dimensionality (Fig. 4.4).

For each performance analysis (Fig. 4.2a–d, Fig. 4.3b), we synthesized 25 datasets (via the DLAG generative model, see Section 4.1.1). Unless specified otherwise, the datasets used for each analysis had the following fixed characteristics: $N = 100$ trials; $q_1 = q_2 = 50$ neurons per area; 500 ms trial lengths with 20 ms sampling period (for $T = 25$ samples per trial); latent dimensionalities $p^a = p_1^w = p_2^w = 5$; signal-to-noise ratios $\mathrm{tr}(C_1 C_1^\top)/\mathrm{tr}(R_1) = \mathrm{tr}(C_2 C_2^\top)/\mathrm{tr}(R_2) = 0.3$; GP timescales $\tau^a, \tau_1^w, \tau_2^w \in [10, 150]$ ms; and delays $D \in [-30, 30]$ ms. Then for each analysis, we varied one of these characteristics to study how it affected DLAG's performance. We found that DLAG performs well over a wide range of simulated conditions, and that DLAG's performance improves with increasing number of trials (Fig. 4.2a), increasing ratio of neurons to latent dimensionlity (Fig. 4.2b,c), and increasing signal-to-noise ratio (Fig. 4.2d).

During the realistic-scale synthetic experiments (Section 4.1), we observed that the variance of both GP timescale and delay estimates increases as the underlying ground truth GP timescale increases (Fig. 4.3a). For intuition, consider the extreme case of a latent variable whose time course is constant, or equivalently, whose autocovariance function (Fig. 3.3) is flat (i.e., has a very long timescale). Then, a range of DLAG models with any delay and any sufficiently long GP timescale could explain the data equally well, particularly in the presence of noise. We systematically verified this trend with additional simulations (Fig. 4.3b). The lowest error in timescale and delay estimation was achieved when GP timescales were equal to the sampling period of observations. For GP timescales larger than the sampling period, the error increases according to the intuition outlined above. For GP timescales less than the sampling period, error increases because a particular delay can be difficult to estimate if its magnitude is large relative to the correspond-

ing GP timescale: the cross-covariance function (Fig. 3.3) decays quickly enough that observed activity appears uncorrelated across areas in that latent dimension.

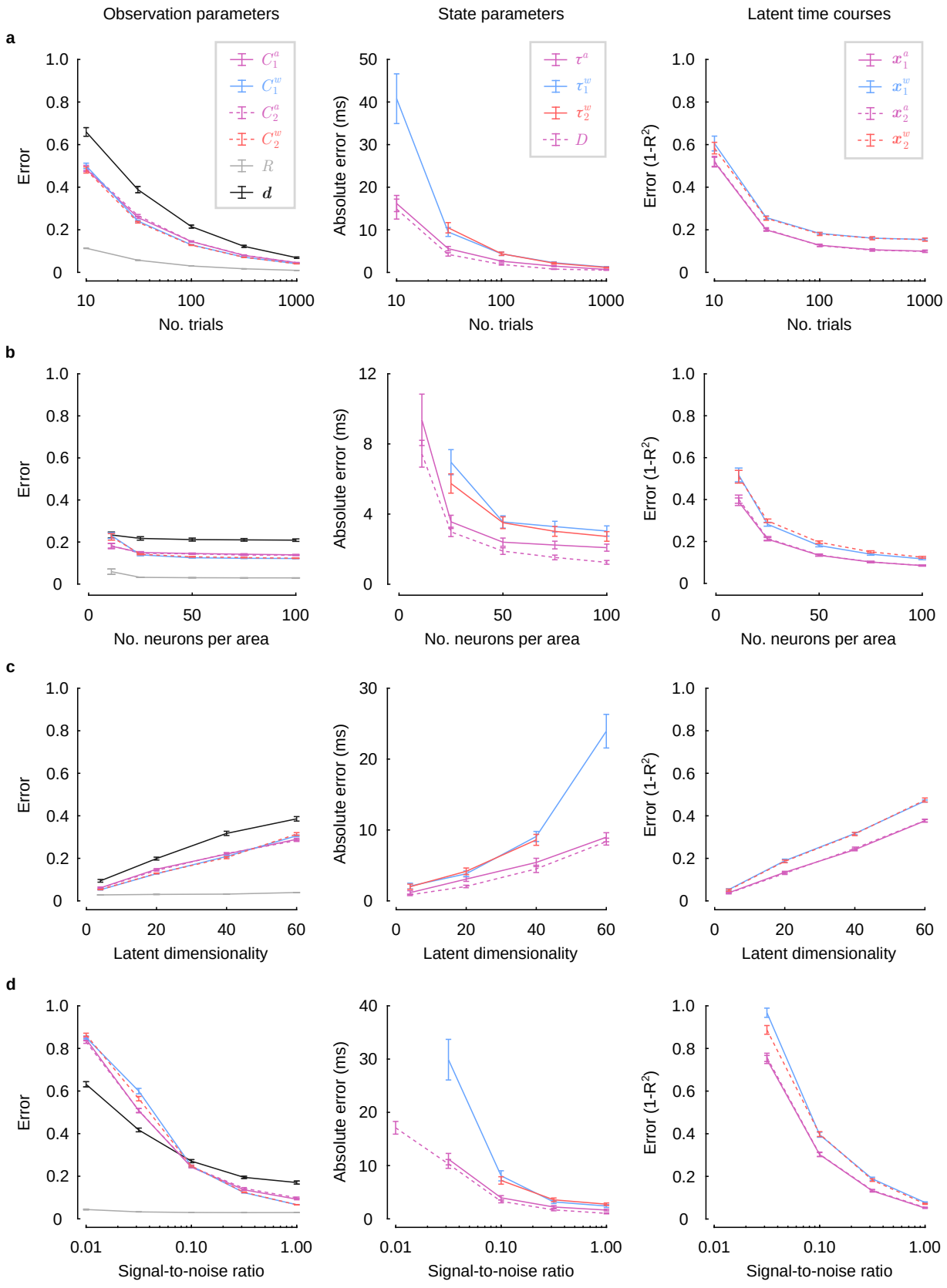
Observation parameters
State parameters
Latent time courses
a
b
c
d
Error
Absolute error (ms)
Error (1-R²)
No. trials
No. neurons per area
Latent dimensionality
Signal-to-noise ratio
C_1^a
C_1^w
C_2^a
C_2^w
R
d
τ^a
τ_1^w
τ_2^w
D
x_1^a
x_1^w
x_2^a
x_2^w

Figure 4.2. DLAG performance as a function of number of trials, number of neurons, latent dimensionality, and signal-to-noise ratio. All panels follow the same plotting conventions: the left column shows the error of observation model parameter estimates (C_1^a: solid magenta; C_2^a: dashed magenta; C_1^w: solid blue; C_2^w: dashed red; R: light gray; $\mathbf{d}$: dark gray); the center column shows the absolute error (in ms) of state model parameter estimates (τ^a: magenta; τ_1^w: blue; τ_2^w: red; D: dashed magenta); the right column shows the error ($1 - R^2$) of latent variable time course estimates ($\mathbf{x}_1^a$: solid magenta; $\mathbf{x}_2^a$: dashed magenta; $\mathbf{x}_1^w$: solid blue; $\mathbf{x}_2^w$: dashed red). (**a**) DLAG performance improves with increasing number of trials. We generated datasets that comprised $N = 1000$ trials. We then took subsets of trials from these datasets, and fit DLAG to increasingly large subsets (sizes equally spaced on a log scale from 10 to 1000 trials). Left: Error bars represent SEM across 25 independent simulated datasets. Center: Error of within-area timescale estimates (τ_2^w) have been omitted for values of 10 trials, where absolute error was 212.1±174.4 ms (mean and SEM across all within-area timescales). Given insufficient statistical power, some GP timescale estimates (likely for latent dimensions that explain little shared variance within an area) become large (i.e., larger than the length of a trial)—to the point where smoothed population activity in the corresponding dimension is effectively constant within a trial. Error bars represent SEM across 125 latent variables. Right: Error bars represent SEM across 25 independent simulated datasets. (**b**) DLAG performance improves with increasing number of neurons (and fixed latent dimensionality). We generated datasets with $q_1 = q_2 = 100$ neurons per area. We then took subsets of neurons from these datasets, and fit DLAG to increasingly large subsets (11, 25, 50, 75, and 100 neurons in each area). Left: Error bars represent SEM across 25 independent simulated datasets. Center: Error of within-area timescale estimates have been omitted for values of 11 neurons per area, where absolute error was 60.1±39.2 ms for τ_1^w and 93.7±46.3 ms for τ_2^w (mean and SEM across all within-area timescales). Error bars represent SEM across 125 latent variables. Right: Error bars represent SEM across 25 independent simulated datasets. (**c**) DLAG performance declines with increasing latent dimensionality (and fixed number of neurons). We considered four settings of across- and within-area dimensionalities ($p^a = p_1^w = p_2^w = 1$, 5, 10, 15). For each setting, we synthesized 25 independent datasets. Here we define the total latent dimensionality (the horizontal axis in each panel) as $2p^a + p_1^w + p_2^w$. Left: Error bars represent SEM across 25 independent simulated datasets. Center: Error of within-area timescale estimates (τ_2^w) have been omitted for values of 60 total latent dimensions, where absolute error was 171.3±91.7 ms (mean and SEM across all within-area timescales). Error bars represent SEM across all across- or within-area latent variables, across all datasets of a given latent dimensionality setting (i.e., across 25, 125, 250, and 375 latent variables for each respective setting). Right: Error bars represent SEM across 25 independent simulated datasets. (**d**) DLAG performance improves with increasing signal-to-noise ratio. We considered five settings for the signal-to-noise ratio (signal-to-noise ratios were the same for both areas; values were spaced equally on a log scale from 0.01 to 1.0). For each setting, we synthesized 25 independent datasets. Left: Error bars represent SEM across 25 independent simulated datasets. Center: Error of GP timescale estimates have been omitted for values of 10^{-2} and $10^{-1.5}$, where absolute errors were greater than 100 ms. Error bars represent SEM across 125 latent variables. Right: Error of latent time course estimates have been omitted for values of 10^{-2}, where average R^2 values were less than 0 (and hence error values were greater than 1). Error bars represent SEM across 25 independent simulated datasets.

4.3 Robustness to violation of model assumptions

In all of the synthetic experiments conducted thus far, data were generated according to the DLAG model itself, and performance was characterized under conditions in which estimates of dimensionality matched the ground truth. We therefore sought to explore DLAG's robustness under several more challenging synthetic scenarios. Here we demonstrate that DLAG's parameter and latent variable estimates remain stable in instances where we induced imperfect estimates of dimensionality (Fig. 4.5, Fig. 4.6). Further-

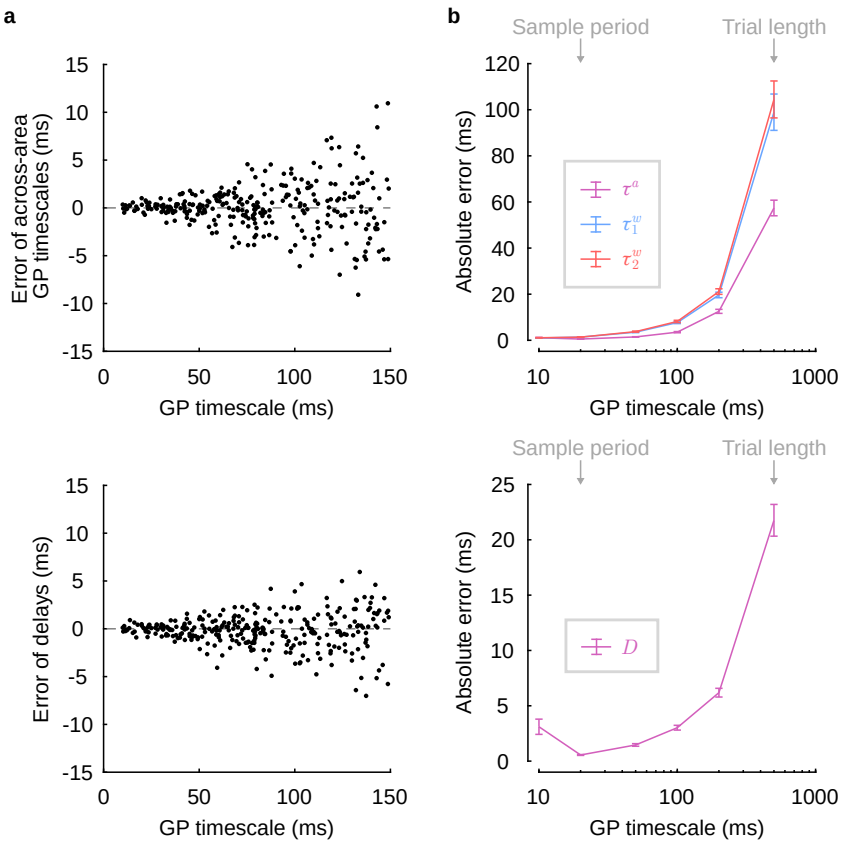

Figure 4.3: Uncertainty of DLAG timescale and delay estimates increases with increasing latent timescale. (**a**) Error (in ms; estimate minus ground truth value) of across-area GP timescale (top) and delay (bottom) estimates for each latent variable shown in Fig. 4.1c,d. The variance of both GP timescale and delay estimates increases as the underlying ground truth GP timescale increases. (**b**) To verify the trend in (a), we systematically characterized the accuracy of GP timescale and delay parameter estimates as a function of ground truth GP timescale. We synthesized additional datasets (via the DLAG generative model) with the following characteristics: $N = 100$ trials; $q_1 = q_2 = 50$ neurons per area; 500 ms trial lengths with 20 ms sampling period (for $T = 25$ samples per trial); latent dimensionalities $p^a = p_1^w = p_2^w = 5$; signal-to-noise ratios $\mathrm{tr}(C_1 C_1^\top)/\mathrm{tr}(R_1) = \mathrm{tr}(C_2 C_2^\top)/\mathrm{tr}(R_2) = 0.3$; and delays $D \in [-30, 30]$ ms. Each dataset's within- and across-area latent variables were given the same GP timescale; and across 150 datasets, we considered six different timescales (25 datasets synthesized for each timescale), ranging in length from half the sampling period to the length of the trial (10 ms, 20 ms, 50 ms, 100 ms, 200 ms, 500 ms). Top: Absolute error (in ms) of across- and within-area GP timescale estimates increases as underlying GP timescale increases (τ^a: magenta; τ_1^w: blue; τ_2^w: red). Bottom: Absolute error (in ms) of delay parameter estimates increases as underlying GP timescale increases. Error bars represent SEM across 125 latent variables.

more, we demonstrate that DLAG shows robustness to mild deviations from its assumptions of linearity and Gaussian observation noise (Fig. 4.7; synthetic datasets were generated via a linear-nonlinear-Poisson model) and its assumption that neural activity follows a Gaussian process (Fig. 4.8).

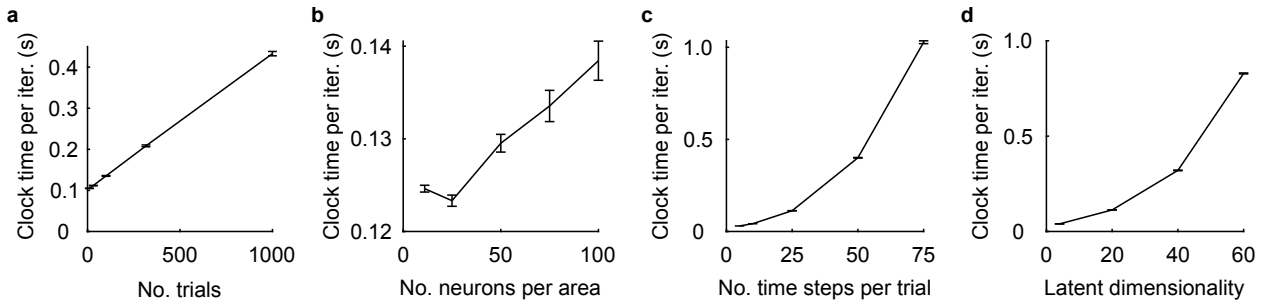

Figure 4.4: DLAG runtime as a function of number of trials, number of neurons, trial length, and latent dimensionality. (**a**) The average clock time (in seconds) per DLAG EM iteration scales (approximately) linearly with the number of trials. These runtime analyses were carried out on synthetic datasets with $q_1 = q_2 = 50$ neurons in each area; $T = 25$ time steps per trial; and latent dimensionalities $p^a = p_1^w = p_2^w = 5$ (total number of latent dimensions $2p^a + p_1^w + p_2^w = 20$). (**b**) The average clock time (in seconds) per DLAG EM iteration scales (approximately) linearly with the number of neurons per area. These runtime analyses were carried out on synthetic datasets with $N = 100$ trials; $T = 25$ time steps per trial; and latent dimensionalities $p^a = p_1^w = p_2^w = 5$ (total number of latent dimensions $2p^a + p_1^w + p_2^w = 20$). (**c**) The average clock time (in seconds) per DLAG EM iteration scales (approximately) quadratically with the number of time steps per trial. Runtime scales quadratically, rather than linearly (as in (a)), because DLAG describes the temporal structure within each trial via Gaussian processes. These runtime analyses were carried out on synthetic datasets with $N = 100$ trials; $q_1 = q_2 = 50$ neurons in each area; and latent dimensionalities $p^a = p_1^w = p_2^w = 5$ (total number of latent dimensions $2p^a + p_1^w + p_2^w = 20$). (**d**) The average clock time (in seconds) per DLAG EM iteration scales (approximately) quadratically with the total number of latent dimensions ($2p^a + p_1^w + p_2^w$). These runtime analyses were carried out on synthetic datasets with $N = 100$ trials; $q_1 = q_2 = 50$ neurons in each area; and $T = 25$ time steps per trial. In (a)-(d), error bars represent SEM across 25 independent simulated datasets. Results were obtained on a Red Hat Enterprise Linux machine (release 7.9, 64-bit) with 250GB of RAM running Matlab (R2019a), on an Intel Xeon CPU (E5-2695 v3, 2.3 GHz).

4.3.1 Stability under imperfect dimensionality estimates

While the results in Fig. 4.1b,e suggest that our model selection procedure performs well on realistic-scale synthetic data, we additionally sought to explore the impact of imperfect dimensionality estimates—inevitable in real data—on the estimation and interpretation of DLAG's parameters and latent variables following fitting. With the goal of inducing dimensionality misestimates, we therefore repeated the analyses in Fig. 4.1b,e with 120 additional datasets generated from the DLAG generative model, but we lowered the signal-to-noise ratio, $\mathrm{tr}(C_m C_m^\top)/\mathrm{tr}(R_m)$, to 0.1 for each area m (compared to 0.3 and 0.2 in area A and area B, respectively, in the original synthetic datasets; see Section 4.1.1). All other data characteristics remained the same as in the original data. Model selection remained accurate overall: estimated across- and within-area dimensionalities never deviated from the ground truth by more than one (results not shown). Any inaccuracy primarily originated from the initial factor analysis (FA) stage of model selection, rather than the second stage involving DLAG.

We first present a case study from one of the synthetic datasets described above, in which the total

dimensionality of area B was underestimated during the initial factor analysis (FA) model selection stage, and across-area dimensionality was underestimated in the second stage (Fig. 4.5). For reference, we first fit a DLAG model with the correct number of within- and across-area latent variables, i.e, no model selection was performed (Fig. 4.5a). With real data, we would not have access to this information, but here we use it to understand the scenario in Fig. 4.5b. It is worth noting that even in the weak-shared variance regime, estimates are qualitatively close to the ground truth.

We next consider the model chosen through model selection, as we would with real data (Fig. 4.5b). The estimated number of latent variables in area B and the estimated number of across-area variables were each one fewer than the respective ground truth. Qualitatively, time course estimates closely match those of the model in Fig. 4.5a, in which the correct number of within- and across-area variables was used (compare estimated latent variables with the same index across Fig. 4.5a and Fig. 4.5b). Furthermore, delay estimates are only slightly affected. By inspection, the third across-area latent variable pair (marked by the asterisks in Fig. 4.5a) now appears as the eighth within-area A latent variable (also marked by an asterisk in Fig. 4.5b).

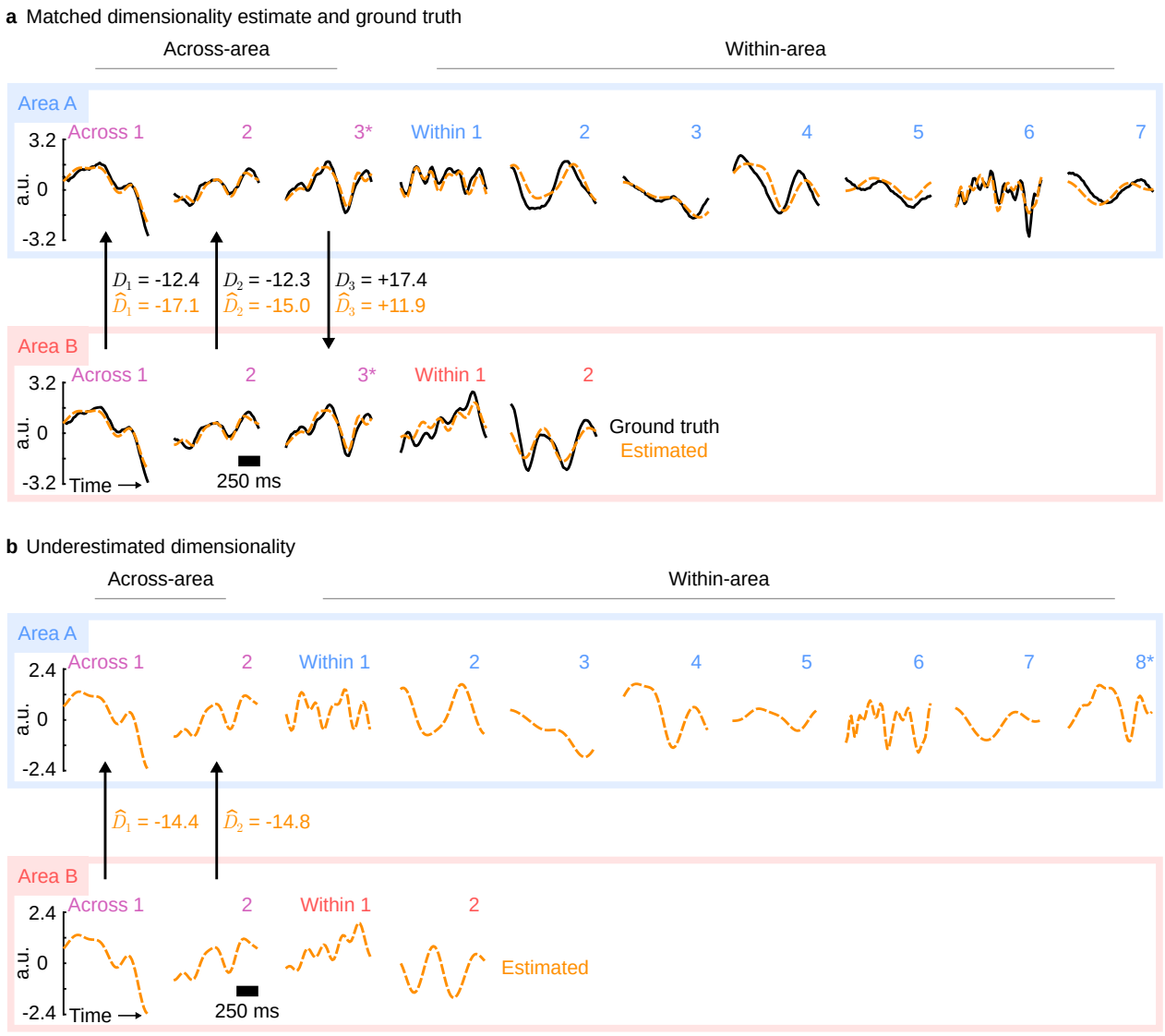

Figure 4.5. DLAG's parameter and latent variable estimates remained stable when dimensionality was underestimated. (**a**) For reference, we first fit a DLAG model with the correct number of within- and across-area latent variables, i.e, no model selection was performed. Shown are single-trial latent-variable time course estimates produced by the fitted model along with the ground truth (one example trial shown). Top row / blue box: area A; bottom row / red box: area B. Left: across-area; right: within-area. Orange dashed traces: DLAG estimates; black solid traces: ground truth. a.u.: arbitrary units. Delays reported in ms. The asterisks ('*') are intended to highlight the third across-area latent variable for each area, which becomes mistaken as a within-area A latent variable when area B's dimensionality is underestimated (see within-area A latent variable 8 in (b)). (**b**) We next consider the model chosen through model selection, as we would with real data. Shown are single-trial latent-variable time course estimates produced by this model (same trial shown as in (a)). Qualitatively, time course estimates closely match those of the model in (a), in which the correct number of within- and across-area variables was used (compare estimated latent variables with the same index across (a) and (b)). By inspection, the third across-area latent variable pair (marked by the asterisks in (a)) now appears as the eighth within-area A latent variable (also marked by an asterisk). Note that the ordering of latent variables is arbitrary; we have ordered the latent variables here to facilitate visual illustration.

Next, we present a second case study from one of the synthetic datasets described above, in which the total dimensionality of area B was overestimated during the initial factor analysis (FA) model selection stage, and across-area dimensionality was overestimated in the second stage (Fig. 4.6). Again for reference, we first fit a DLAG model with the correct number of within- and across-area latent variables, i.e., no model selection was performed (Fig. 4.6a). With real data, we would not have access to this information, but here we use it to understand the scenario in Fig. 4.6b. Like the previous case study, even in the weak-shared variance regime, estimates are qualitatively close to the ground truth.

We next consider the model chosen through model selection, as we would with real data (Fig. 4.6b). The estimated number of latent variables in area B and the estimated number of across-area variables were each one more than the respective ground truth. Qualitatively, time course estimates closely match those of the model in Fig. 4.6a, in which the correct number of within- and across-area variables was used (compare estimated latent variables with the same index across Fig. 4.6a and Fig. 4.6b). By inspection, the eighth within-area A latent variable (marked by the asterisk in Fig. 4.6a) now appears as the third across-area latent variable (also marked by asterisks in Fig. 4.6b). This phenomenon is straightforward to diagnose. Upon additionally scaling each latent variable by the fraction of shared variance it explains within its respective area, it becomes clear that the third across-area latent variable explains little shared variance in area B, consistent with the ground truth.

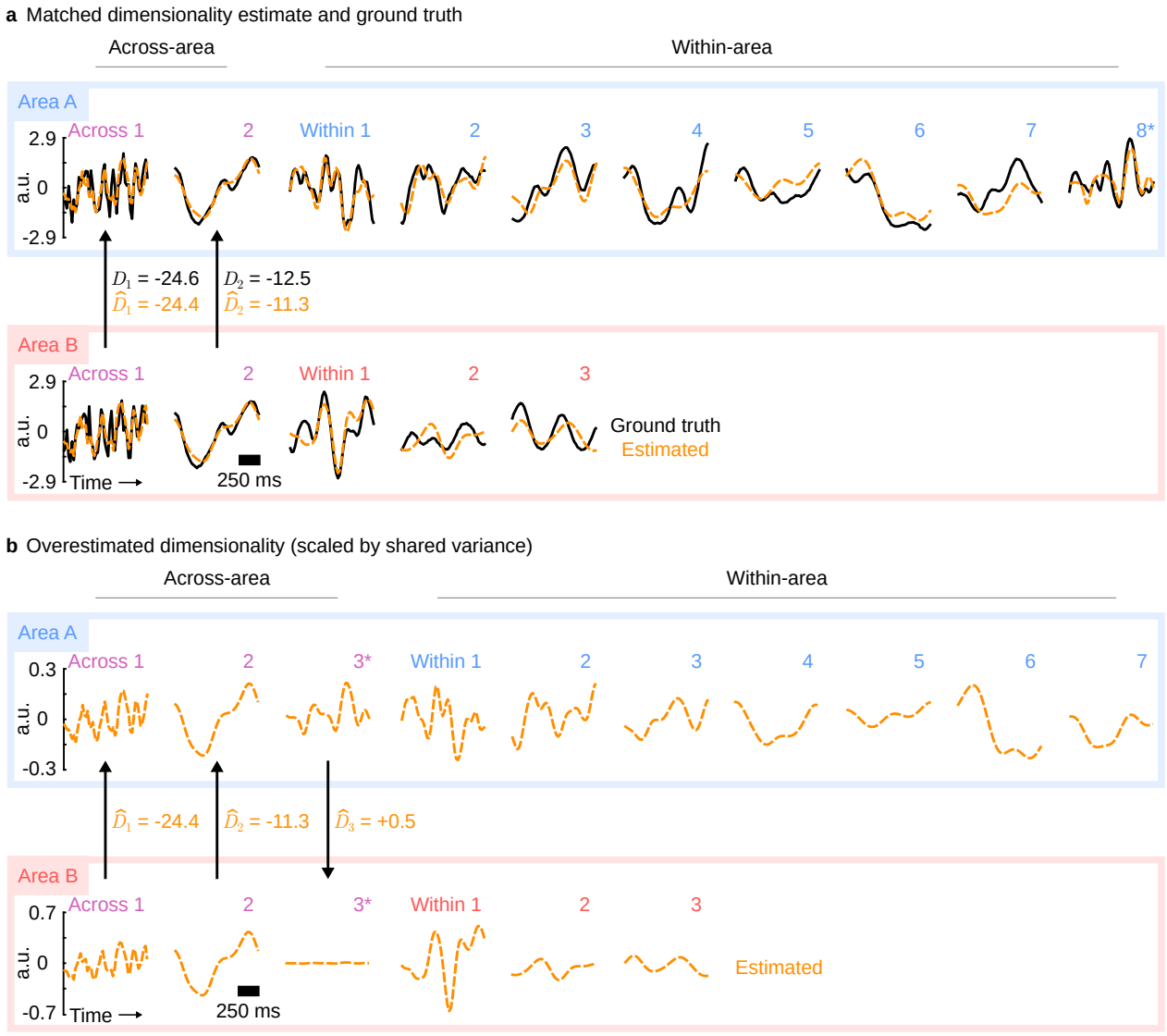

Figure 4.6. DLAG's parameter and latent variable estimates remained stable when dimensionality was overestimated. (**a**) For reference, we first fit a DLAG model with the correct number of within- and across-area latent variables, i.e., no model selection was performed. Shown are single-trial latent-variable time course estimates produced by the fitted model along with the ground truth. Same conventions as in Fig. 4.5. The asterisk ('*') is intended to highlight the eighth within-area A latent variable, which becomes mistaken as an across-area variable when area B's dimensionality is overestimated (see across-area variable 3 in (b)). (**b**) We next consider the model chosen through model selection, as we would with real data. The estimated number of latent variables in area B and the estimated number of across-area variables were each one more than the respective ground truth. Shown are single-trial latent-variable time course estimates produced by this model (same trial shown as in (a)). Qualitatively, time course estimates closely match those of the model in (a), in which the correct number of within- and across-area variables was used (compare estimated latent variables with the same index across (a) and (b)). By inspection, the eighth within-area A latent variable (marked by the asterisk in (a)) now appears as the third across-area latent variable (also marked by asterisks). This phenomenon is straightforward to diagnose: here, we have additionally scaled each latent variable by the fraction of shared variance it explains within its respective area. The third across-area latent variable explains little shared variance in area B, consistent with the ground truth. Note that the ordering of latent variables is arbitrary; we have ordered the latent variables

here to facilitate visual illustration.

4.3.2 Robustness to violations of the linear-Gaussian assumption

We sought to understand how the results in Fig. 4.1 might change if we applied DLAG to synthetic data in which the linear and Gaussian assumptions of the DLAG observation model, equations (3.1) and (3.2), are violated. Toward that end, we generated additional synthetic datasets from the following linear-nonlinear-Poisson (LNP) generative model. For a given dataset, on each trial, we generated within- and across-area latent variable time courses according to the DLAG state model, equations (3.3)–(3.8). Hence each latent variable time course followed a Gaussian process (GP) with squared exponential (SE) covariance function, and across-area latent variables included time delays across areas.

For area m with q_m neurons, we then generated neural firing rates, $\boldsymbol{\lambda}_{m,t} \in \mathbb{R}^{q_m}$, during time bin t of width Δ according to the following model:

$$\boldsymbol{\lambda}_{m,t} = \log\left(1 + \exp\left(C^a_m \mathbf{x}^a_{m,:,t} + C^w_m \mathbf{x}^w_{m,:,t} + \mathbf{d}_m\right)\right) \cdot \Delta \tag{4.5}$$

The function $\log(1 + \exp(\cdot))$ is the commonly used softplus function (applied element-wise to its arguments), a smooth analogue of the rectified linear function. The parameters $C^a_m \in \mathbb{R}^{q_m \times p^a}$, $C^w_m \in \mathbb{R}^{q_m \times p^w_m}$, and $\mathbf{d}_m \in \mathbb{R}^{q_m}$ have similar interpretations as in equations (3.1) and (3.2) of the DLAG observation model. We then generated observed spike counts for neuron j in area m during time bin t, $y_{m,j,t}$, according to a Poisson distribution with rate parameter $\lambda_{m,j,t}$ (the j^{th} element of $\boldsymbol{\lambda}_{m,t}$):

$$y_{m,j,t} \mid \mathbf{x}^a_{m,:,t}, \mathbf{x}^w_{m,:,t} \sim \text{Poisson}(\lambda_{m,j,t}) \tag{4.6}$$

Note that this generative model can be interpreted as describing nonlinear interactions across areas since the conditional distributions $P(\mathbf{y}_{2,t} \mid \mathbf{y}_{1,t})$ and $P(\mathbf{y}_{1,t} \mid \mathbf{y}_{2,t})$ describe nonlinear relationships between the observed neural activity in each area, $\mathbf{y}_{1,t}$ and $\mathbf{y}_{2,t}$.

As we did for the synthetic datasets underlying Fig. 4.1 (see Section 4.1.1), we generated synthetic datasets from the LNP generative model that were informed by experimental recordings. For all datasets, we chose the numbers of neurons in each area based on our V1-V2 recordings (area A: $q_1 = 80$; area B: $q_2 = 20$). We set the combined total dimensionality in each area to representative values (area A: $p^a + p^w_1 = 10$; area B: $p^a + p^w_2 = 5$), but varied the relative number of within- and across-area latent variables across datasets. Generating 20 datasets at each of six configurations ($p^a = 0, \ldots, 5$; $p^w_1 = 5, \ldots, 10$; $p^w_2 = 0, \ldots, 5$) resulted in a total of 120 independent datasets.

We generated the mean parameter for each area m, $\mathbf{d}_m$, so that the distribution of mean firing rates over time and trials was qualitatively similar to typical mean firing rate distributions encountered in V1

and V2 recordings. Specifically, we drew each element of $\mathbf{d}_m$ from an exponential distribution with mean 20 spikes/second and 10 spikes/second in area A and area B, respectively. To ensure that the synthetic datasets exhibited realistic noise levels, we manually tuned the loading matrix parameters for each area, C_m, so that the signal-to-noise ratios according to DLAG model estimates, $\mathrm{tr}(\hat{C}_m\hat{C}_m^\top)/\mathrm{tr}(\hat{R}_m)$, were similar to those encountered in V1 and V2 (0.3 in area A; 0.2 in area B).

Finally, we drew all timescales ($\{\tau_j^a\}_{j=1}^{p^a}$, $\{\tau_{1,j}^w\}_{j=1}^{p_1^w}$, $\{\tau_{2,j}^w\}_{j=1}^{p_2^w}$) uniformly from $U(\tau_{\min}, \tau_{\max})$, with $\tau_{\min} = 10$ ms and $\tau_{\max} = 150$ ms. We drew all delays ($\{D_1, \ldots, D_{p^a}\}$) uniformly from $U(D_{\min}, D_{\max})$, with $D_{\min} = -30$ ms and $D_{\max} = +30$ ms. All Gaussian process noise variances ($\{(\sigma_j^a)^2\}_{j=1}^{p^a}$, $\{(\sigma_{1,j}^w)^2\}_{j=1}^{p_1^w}$, $\{(\sigma_{2,j}^w)^2\}_{j=1}^{p_2^w}\}$) were fixed at 10^{-3}. With all model parameters specified, we then generated $N = 100$ independent and identically distributed trials according to the LNP generative model described above. Each trial was $1,000$ ms in length, comprising spike counts in $T = 50$ time bins of width 20 ms, the same spike count bin width used to analyze the V1-V2 recordings. Fig. 4.7a, c, d, f, and g demonstrate DLAG's ability to estimate the ground truth latent variable time courses and parameters of the LNP generative models when the correct within- and across-area dimensionalities are assumed. Fig. 4.7b and e show the results of estimating across- and within-area dimensionalitities from the data.

Overall, these results suggest that, for firing rates similar to those encountered in the experimental recordings we consider in this work, DLAG is largely robust when the neural activity is not generated according to the linear and Gaussian assumptions of the DLAG observation model. Across the neuronal populations, firing rates are sufficiently high that neural activity is essentially operating in the linear regime of the softplus function, and a Gaussian noise model can still suffice for Poisson-distributed spike counts (we explore low-firing-rate regimes in Fig. 4.7h).

The LNP-generated activity appears to have the greatest impact on the estimation of across- and within-area dimensionalities, shown in Fig. 4.7b,e. During the first stage of our model selection procedure, the optimal factor analysis (FA) dimensionality was larger than the ground truth in at least one area in 115 of 120 datasets. Consequently, estimated within-area dimensionalities also tend to be higher than the ground truth. Interestingly, across-area dimensionality estimates remained highly accurate, matching the ground truth in 107 of 120 datasets (across-area latent activity is shared among a larger number of neurons, leading to greater statistical power). We have already explored the consequences of misestimates of dimensionality in Fig. 4.5 and Fig. 4.6; those results still hold here. Quantifying the shared variance explained by each latent variable provides safeguards against the overestimation of dimensionality.

To probe the limits of DLAG's performance as a function of firing rate (Fig. 4.7h), we synthesized additional datasets from the LNP generative model defined above, with the following characteristics: $N = 100$ trials; $q_1 = q_2 = 50$ neurons per area; 500 ms trial lengths with 20 ms spike count bin widths

(for $T = 25$ bins per trial); latent dimensionalities $p^a = p_1^w = p_2^w = 5$; GP timescales $\tau^a, \tau_1^w, \tau_2^w \in [10, 150]$ ms; and delays $D \in [-30, 30]$ ms. We systematically varied the mean parameter, $\mathbf{d}$, of the models used to generate each dataset (equally spaced on a log scale from 1 to 100 spikes/second). All neurons had the same mean parameter value, so that mean firing rates over time and trials were nearly the same for all neurons. We manually tuned the loading matrix parameters for each area, C_m, so that the signal-to-noise ratios according to DLAG model estimates, $\mathrm{tr}(\hat{C}_m \hat{C}_m^\top)/\mathrm{tr}(\hat{R}_m)$, were no greater than 0.2 for all firing rate settings. For Poisson-distributed spike counts, the estimated signal-to-noise ratio is inextricably linked to firing rate: in the lowest firing rate setting, 1.0 spikes/second, estimated signal-to-noise ratios were about 0.04. We generated 25 independent datasets for each firing rate setting.

Overall, the smooth degradation of performance as mean firing rates decrease (Fig. 4.7h) is an expected trend: neural activity increasingly inhabits the nonlinear regime of the softplus function, and DLAG's Gaussian noise model becomes a poorer description of the Poisson-distributed spike counts. Importantly, however, DLAG's performance remains stable over a wide range of firing rates, from 100 spikes/second (50 spikes/trial) to as low as 3 spikes/second (1.5 spikes/trial).

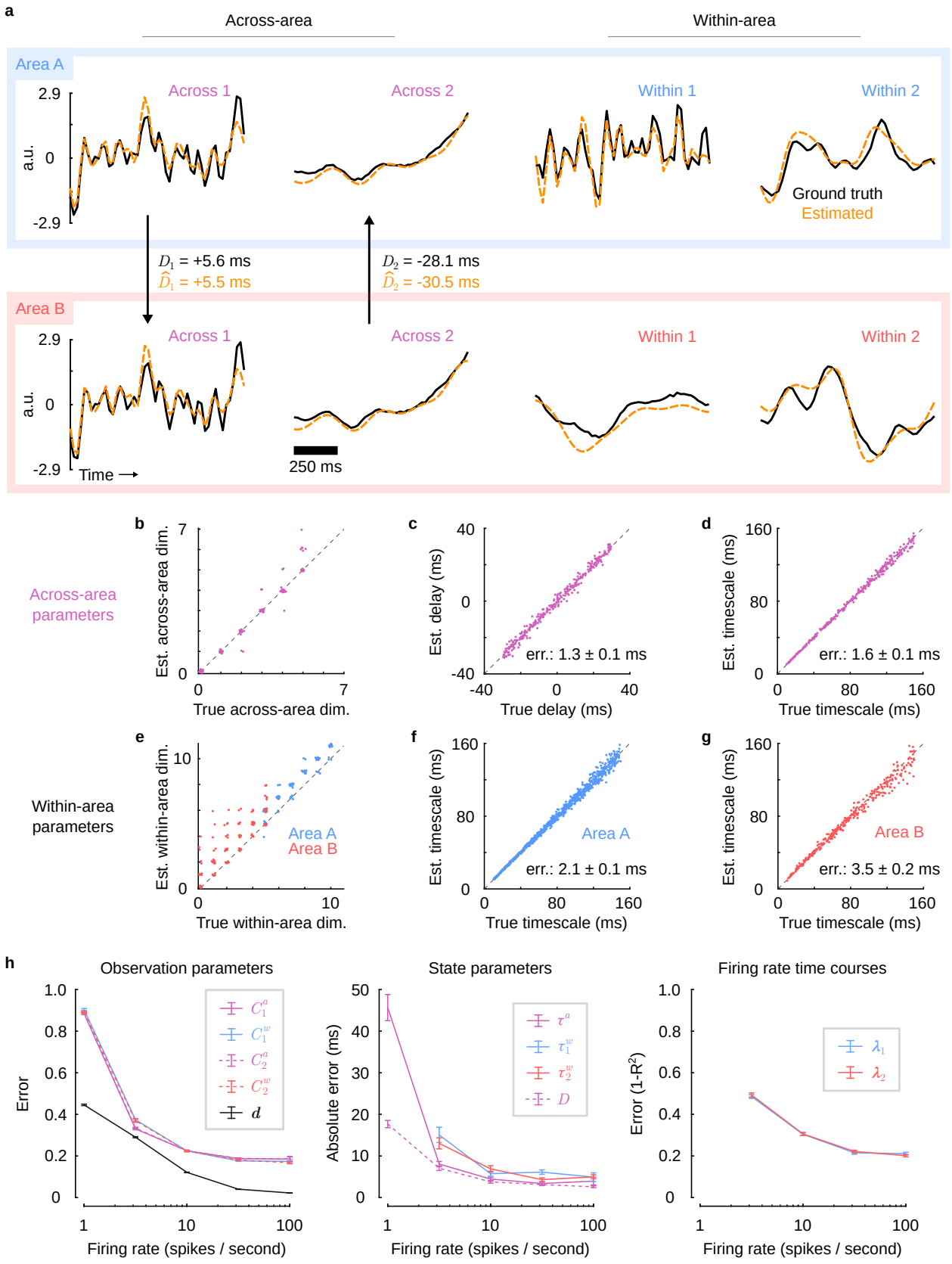
a
Across-area
Within-area
Area A
Across 1
Across 2
Within 1
Within 2
2.9
0
-2.9
a.u.
Ground truth
Estimated
D_1 = +5.6 ms
$\widehat{D}_1$ = +5.5 ms
D_2 = -28.1 ms
$\widehat{D}_2$ = -30.5 ms
Area B
Time
250 ms
b
Across-area parameters
Est. across-area dim.
True across-area dim.
c
Est. delay (ms)
True delay (ms)
err.: 1.3 ± 0.1 ms
d
Est. timescale (ms)
True timescale (ms)
err.: 1.6 ± 0.1 ms
e
Within-area parameters
Est. within-area dim.
True within-area dim.
Area A
Area B
f
err.: 2.1 ± 0.1 ms
g
err.: 3.5 ± 0.2 ms
h
Observation parameters
State parameters
Firing rate time courses
Error
Absolute error (ms)
Error (1-R²)
Firing rate (spikes / second)

Figure 4.7. DLAG accurately estimates within- and across-area time courses and their parameters in synthetic data generated by a linear-nonlinear-Poisson model. (**a**) Single-trial latent-variable time course estimates for a representative synthetic dataset. Same conventions as in Fig. 4.1a. Across all synthetic datasets for which across- or within-area dimensionality was non-zero (across: 100 datasets; within A: 120 datasets; within B: 100 datasets), mean accuracy (R^2) of firing rate estimation was as follows: area A – 0.81; area B – 0.76 (all SEM values less than 0.01). Similarly, mean accuracy of subspace (loading matrix) estimation was as follows: C_1^a – 0.77; C_2^a – 0.83; C_1^w – 0.79; C_2^w – 0.83 (where a value of 1 implies that the ground truth is fully captured by estimates; all SEM values less than 0.01). (**b**) Across-area dimensionality estimates versus the ground truth for all 120 synthetic datasets. Data points are integer-valued, but randomly jittered to show points that overlap. (**c**) Delay estimates versus the ground truth. Displayed error ('err.') indicates mean absolute error and SEM reported across 300 across-area variables. (**d**) Across-area Gaussian process (GP) timescale estimates versus the ground truth. Displayed error ('err.') indicates mean absolute error and SEM reported across 300 across-area variables. (**e**) Within-area dimensionality estimates versus the ground truth for all 120 synthetic datasets (blue: within-area A; red: within-area B). Data points are integer-valued, but randomly jittered to show points that overlap. (**f**) Within-area A GP timescale estimates versus the ground truth. Displayed error ('err.') indicates mean absolute error and SEM reported across 900 within-area variables in area A. (**g**) Within-area B GP timescale estimates versus the ground truth. Displayed error ('err.') indicates mean absolute error and SEM reported across 300 within-area variables in area B. (**h**) DLAG performance remains stable over a range of realistic firing rates. Left: Error of observation model parameter estimates decreases with increasing firing rate, d (C_1^a: solid magenta; C_2^a: dashed magenta; C_1^w: solid blue; C_2^w: dashed red; **d**: dark gray). Error bars represent SEM across 25 independent simulated datasets. Center: Absolute error (in ms) of state model parameter estimates decreases as firing rate increases (τ^a: magenta; τ_1^w: blue; τ_2^w: red; D: dashed magenta). Error of within-area timescale estimates have been omitted for values of 1 spike/second, where absolute error was 685±236 ms for τ_1^w and 1089±339 ms for τ_2^w (mean and SEM across all within-area timescales). Given insufficient statistical power, some GP timescale estimates (likely for latent dimensions that explain little shared variance within an area) become large (i.e., larger than the length of a trial)—to the point where smoothed population activity in the corresponding dimension is effectively constant within a trial. Error bars represent SEM across 125 latent variables. Right: Error ($1 - R^2$) of firing rate time course estimates decreases as mean firing rate increases (λ_1: blue; λ_2: red). Error values have been omitted for values of 1 spike/second, where R^2 values were less than 0 (and hence error values were greater than 1). Error bars represent SEM across 25 independent simulated datasets.

4.3.3 Robustness to violations of the Gaussian process state model assumption

We next sought to investigate the effects of violations to DLAG's Gaussian process state model assumptions (see also Appendix B). We therefore explored a case study in which the latent time courses of the linear-nonlinear-Poisson (LNP) generative model, described inSection 4.3.2, were inspired by the V1-V2 neural recordings, rather than generated via Gaussian processes.

We generated ground truth across-area latent time courses as follows. (For simplicity, we did not consider within-area latent variables in this case study.) First, we applied canonical correlation analysis (CCA) to spike trains (i.e., neuronal spikes counted in 1 ms time bins) from the same V1-V2 dataset as analyzed in Fig. 5.2. Hence the data consisted of 400 trials, each 1280 ms in length. CCA produces two sets of canonical basis vectors (dimensions)—one for V1 and one for V2. We took the top three canonical dimensions in V1, and projected observed V1 spike trains on each trial onto these canonical dimensions.

Then, we averaged the projected activity in each canonical dimension over trials, to produce a single set of trial-averaged "template" time courses. For each template time course, we took activity in a 1000 ms time window—these snippets became the across-area latent time courses for our simulated area A. We then took another 1000 ms snippet from each template, time-shifted relative to the snippets used for area A—these snippets became the time-delayed across-area latent time courses for our simulated area B.

Next, we generated an observed spike train on each simulated trial from the LNP observation model defined in equations (4.5) and (4.6). The same latent time courses were used on each trial, hence all sources of trial-to-trial variability in these simulations arise from Poisson-distributed noise that is independent across neurons. Before applying DLAG, we counted spikes in 20 ms time bins, as we did for the V1-V2 recordings. Remaining dataset characteristics were as follows: $N = 100$ trials; $q_1 = q_2 = 50$ neurons per area. We drew each element of the mean parameter for area m, $\mathbf{d}_m$, from an exponential distribution with mean 20 spikes/second (same for area A and area B). We manually tuned the loading matrix parameters for each area, C_m, so that the signal-to-noise ratios according to DLAG model estimates, $\mathrm{tr}(\hat{C}_m\hat{C}_m^\top)/\mathrm{tr}(\hat{R}_m)$, was 0.3 for both areas.

Notice how the assumptions of the DLAG state model (i.e., that latent time courses follow a zero-mean Gaussian process) no longer hold for these simulated data (Fig. 4.8a). First, latent time courses are no longer zero-mean. Second, latent time courses no longer covary according to a squared exponential function. For instance, all three ground truth across-area latent variable pairs exhibit strong periodic structure. Furthermore, each latent time course comprises multiple timescales: notably, fast transient activity at the beginning of each trial, and slower timescales as the trial progresses.

To focus first on the effects these violated assumptions had on DLAG's estimation of latent time courses, without being concerned about model selection, we fit a DLAG model with the same number of across-area latent variables as the ground truth (Fig. 4.8b). Importantly, DLAG's estimates recapitulated the key qualitative features of the ground truth, including the fast increase in activity at the beginning of each trial (see "Across 1" in Fig. 4.8b) and the periodic structure throughout each trial. Time delays were also accurately estimated. The latent time courses estimated by DLAG were qualitatively smoother than the ground truth (particularly during the first 60 ms of each trial), a consequence originating from two sources: (1) temporal smoothing via the SE kernel, and (2) counting spikes in 20 ms time bins.

Next, we assumed no prior knowledge of the ground truth dimensionality—as would be the case with real neural recordings—and estimated the across-area dimensionality. Interestingly, the optimal across-area dimensionality, selected via cross-validated data log-likelihood, was 6, greater than the ground truth value. We investigated the latent time courses extracted by this 6-dimensional model (Fig. 4.8c).

The first three across-area variables still recapitulated the main features of the ground truth. Relative

to the 3-dimensional DLAG model (Fig. 4.8b), the delay estimates of the 6-dimensional DLAG model differed by a few ms. Close inspection of the first 60 ms of each trial suggests that the first three latent variables of the 6-dimensional DLAG model smooth over the fast transient activity to a greater degree than the 3-dimensional DLAG model. Indeed, Across 1 in Fig. 4.8c has a slightly longer GP timescale (43 ms) than Across 1 in Fig. 4.8b (34 ms).

The remaining latent variables, Across 4–6, are used by DLAG to account for the multiple timescales present in the ground truth. Across 4 combines with Across 1 to account for the fast rise in activity. Across 5 accounts for slower temporal structure throughout the trial, present in all ground truth time courses. Across 6 is periodic with twice the temporal frequency of Across 3, and hence a harmonic signal. We note that we did not rescale latent variable amplitudes here, to best highlight the temporal structure of each latent variable; however, these "extra" latent variables explained little shared variance relative to the first three latent variables (Across 4–6 cumulatively explained only 12% and 9% of the shared variance in area A and in area B, respectively). Still, the model selection results (i.e., that 6 dimensions was deemed optimal) suggest that these extra latent variables do improve DLAG's ability to capture the temporal structure of this simulated neural activity. See Appendix A for further discussion.

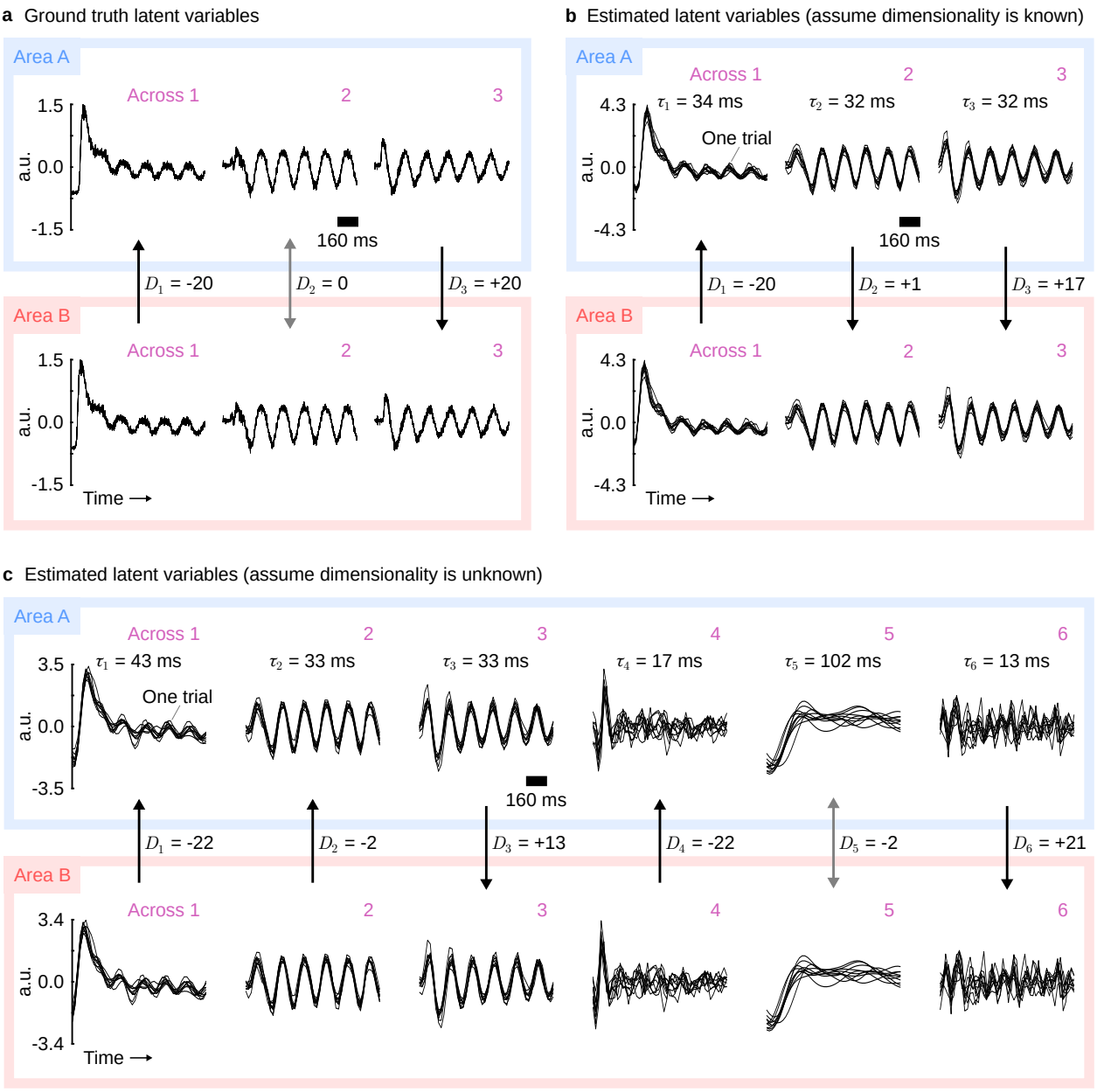

Figure 4.8. DLAG performance when state model, in addition to observation model, assumptions are violated. (**a**) Ground truth latent variable time courses. Top row / blue box: area A; bottom row / red box: area B. Across-area variables are paired vertically; vertical arrows point in the direction of signal flow, as defined by the sign of the delay next to each arrow (all delay values are in units of ms). a.u.: arbitrary units. Ground truth latent time courses are the same on every trial. (**b**) Single-trial latent time courses for a DLAG model fit with the same number of across-area latent variables as the ground truth. Each black trace corresponds to one trial; for clarity, only 10 of 100 are shown. To facilitate comparison with panel (c), the estimated GP timescale is displayed for each latent variable (τ_1, τ_2, τ_3). All other conventions are the same as in panel (a). (**c**) Same conventions as in panel (b) for the 6-dimensional model chosen through cross-validation.

4.4 DLAG disentangles concurrent signaling where CCA cannot

Here we leverage simulations to demonstrate where a static method like CCA is unable to disentangle concurrent signaling. In brief, we synthesized two additional datasets from the linear-nonlinear-Poisson (LNP) generative model defined in Section 4.3.2. The two datasets were nearly identical, with one difference: in the first dataset (Fig. 4.9a), across-area latent variables had different strengths; in the second dataset (Fig. 4.9b), across-area latent variables had equal strengths. Note that this difference between datasets cannot be seen in Fig. 4.9, since the amplitudes of latent time courses are normalized.

In detail, we first generated latent time courses for $p^a = 2$ across-area variables. For simplicity, we did not include within-area latent variables. One across-area variable (Across 1) was assigned a delay of $+25$ ms (so that area A leads area B; observe the relative time-shift in Across 1 between black traces in area A versus area B); the second across-area variable (Across 2) was assigned a delay of -25 ms (so that area B leads area A; observe the relative time-shift in Across 2 between black traces in area A versus area B). Both across-area variables had the same Gaussian process (GP) timescale, 60 ms. In this demonstration, we wanted to isolate the consequences of the CCA model definition from issues like overfitting. We therefore simulated a data-rich scenario by generating $N = 1,000$ independent trials, each 500 ms in length. On each trial, we generated a different set of across-area latent time courses, X_n. Let $X = \{X_1, \ldots, X_N\}$ be the set of latent time courses over all N trials.

For both datasets, we generated spike trains (see equations (4.5) and (4.6)) at 1 ms resolution for $q_1 = q_2 = 50$ neurons per area from the common set of latent time courses, X. All neurons had the same mean parameter value ($\mathbf{d}$, defined in equation (4.5)) of 20 spikes/second, so that mean firing rates over time and trials were nearly the same for all neurons. The loading matrix parameters for each area, C_m^a, were manually tuned so that the signal-to-noise ratios according to DLAG model estimates, $\mathrm{tr}(\hat{C}^a{}_m \hat{C}^a{}_m^\top)/\mathrm{tr}(\hat{R}_m)$, were 0.2. We counted spikes in 20 ms time bins, and then fit both a CCA model and a DLAG model to each dataset.

The difference between the two datasets was as follows. For the first dataset, we scaled the columns of C_m^a (for each area m) so that the magnitude of the column associated with the $+25$ ms latent variable was twice the magnitude of the column associated with the -25 ms latent variable. For the second dataset, we took the same C_m^a that was used for the first dataset, but rescaled the columns of C_m^a (for each area m) so that both columns had equal magnitude. We performed this rescaling such that signal-to-noise ratios remained the same across both datasets. Thus these two datasets allowed us to isolate the effects of the relative strengths of feedforward versus feedback signals on CCA's (and DLAG's) ability to disentangle those signals.

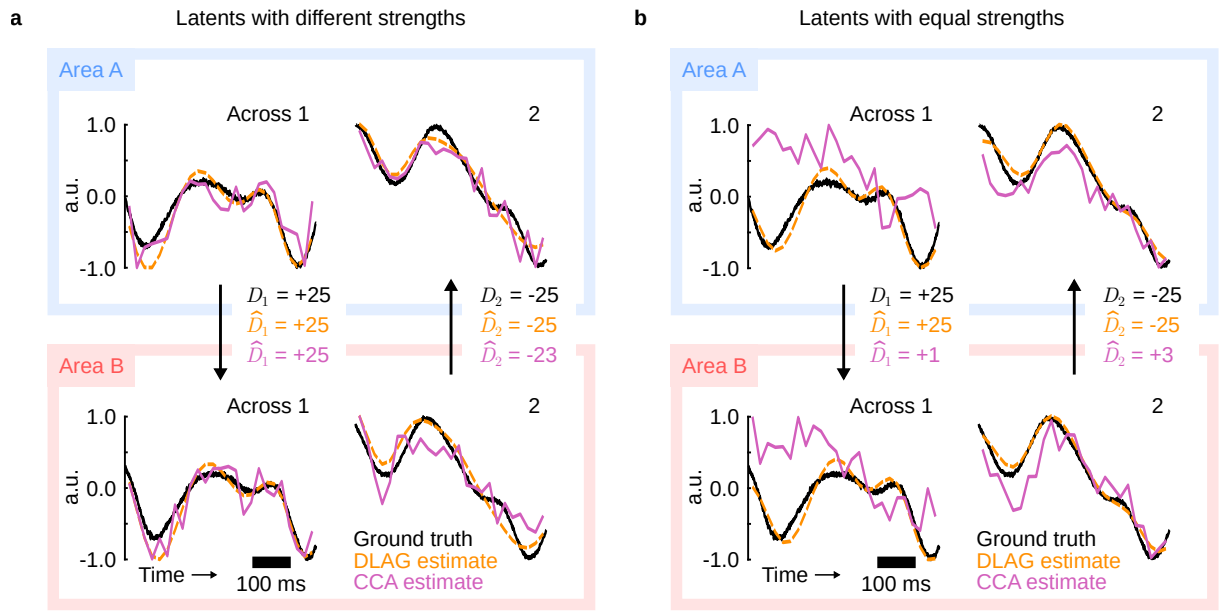

Figure 4.9: Canonical correlation analysis (CCA) cannot disentangle signals that are relayed concurrently and with similar strength. (**a**) Each canonical dimension can reflect a directed interaction if the signals in each direction have different strengths. Top row / blue box: area A; bottom row / red box: area B. Black solid traces: ground truth across-area latent time courses on a representative trial. Orange dashed traces: DLAG estimates. Magenta solid traces: CCA estimates. a.u.: arbitrary units. Black arrows indicate the direction of signal flow between area A and area B, given by the ground truth delay value. Ground truth and estimated delay values (in ms) are shown to the right of each arrow (top, black: ground truth; center, orange: DLAG estimate; bottom, magenta: CCA estimate). Canonical pairs are sorted from left to right, in descending order, based on the value of their canonical correlation. (**b**) Canonical dimensions reflect a mixture of signals relayed in each direction if those signals have similar strengths. Same conventions as in panel (a).

Time delays are not inherently built into the CCA model. To estimate a time delay for each pair of fitted canonical dimensions, we identified the time delay at which projections of area A activity and projections of area B activity had maximum cross-correlation. The cross-correlation function between area A and area B projections was computed with 1 ms resolution, from -40 ms (B leads A) to +40 ms (A leads B). In detail, we first took a fixed window of activity in area A, 420 ms in length, from 40 ms to 460 ms into the trial. For each trial, we counted spikes within this window in 20 ms nonoverlapping time bins, and projected this activity onto each canonical dimension in area A. For area B, we employed a sliding window of length 420 ms, which we advanced in 1 ms increments, from the beginning of the trial to 80 ms into the trial. At each increment, we counted spikes within the window in 20 ms nonoverlapping time bins, and projected this activity (on each trial) onto each canonical dimension in area B. For each canonical pair, we computed the Pearson correlation between the projected area A activity and the projected area B activity. This correlation value gave one element of a cross-correlation function: repeating this procedure at each increment of the sliding window in area B produced a cross-correlation function from -40 ms to

+40 ms. We then identified the time delay at which the cross-correlation function for each canonical pair was maximum.

In general, the first canonical pair returned by CCA is the pair of dimensions along which projections of simultaneously observed activity exhibit the greatest correlation across areas. Projections onto the second canonical pair exhibit the second greatest correlation across areas, and so on. In the first dataset, projections of simultaneously observed activity onto Across 1 exhibit greater across-area correlation than do projections onto Across 2, by design. Thus the first and second canonical pairs (Fig. 4.9a, magenta traces) indeed reasonably reflected each direction of signal flow. DLAG estimates closely match the ground truth (Fig. 4.9a, orange dashed traces).

The second dataset leads to dramatically different results (Fig. 4.9b). Because the latent variables in the second dataset have similar strengths, the canonical pairs do not provide a faithful description of each direction of signal flow. The CCA-estimated time courses and time delays deviate significantly from the ground truth (Fig. 4.9b, magenta traces). DLAG estimates, on the other hand, still closely match the ground truth (Fig. 4.9b, orange dashed traces).

Overall, these two scenarios demonstrate that CCA can identify directions of signal flow if signals in one direction are dominant (Fig. 4.9a), but not if signals in both directions have similar strengths (Fig. 4.9b). DLAG successfully disentangles concurrent signaling in both scenarios.

Chapter 5

Dissecting bidirectional interactions among early and midlevel visual cortical areas

5.1 Dissecting interactions between V1 and V2

We first used DLAG to study interactions between two areas in the early visual system: V1 and V2. V1 and V2 share strong reciprocal connections[59,60] and show correlated activity[22–24,26,37], but the bidirectional nature of their interactions is not yet well understood. We simultaneously recorded the activity of neuronal populations in the superficial (output) layers of V1 (61 to 122 neurons; mean 86.3), and the middle (input) layers of V2 (15 to 32 neurons; mean 19.6) in three anesthetized monkeys (Fig. 5.1a; data reported previously in [26, 37]). Recording locations were selected to maximize the probability that the recorded V1 and V2 populations interact by ensuring spatial receptive field alignment. We analyzed neuronal responses measured during the 1.28 second presentation of drifting sinusoidal gratings of different orientations, and counted spikes in 20 ms time bins. The periodic nature of the drifting gratings (160 ms per cycle) is evident in peristimulus time histograms (PSTHs) for an example recording session and grating orientation (Fig. 5.1b). In total, we fit DLAG models separately to 40 "datasets," corresponding to five recording sessions, each with eight different orientations. For comparison, on each dataset we also randomly split V1 into two equally sized subpopulations (termed V1a and V1b; Fig. 5.1c), and then applied DLAG to study V1a-V1b interactions in a manner identical to V1-V2.

V1-V2 interactions are selective and are more prominent in V2 than in V1 We first used DLAG to study whether V1 and V2 interact selectively: in addition to fluctuations shared between V1 and V2, are there fluctuations that are not shared between the two areas? Selective inter-areal communication may be a hallmark of cortical computation that remains to be fully understood, particularly at the level of

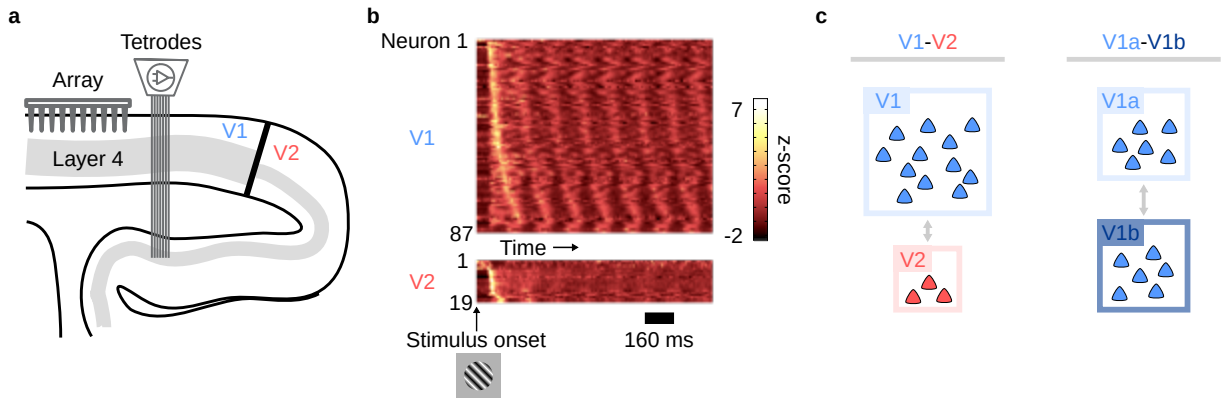

Figure 5.1: Simultaneous population recordings in V1 and V2. (**a**) Schematic showing a sagittal section of occipital cortex and the recording setup. V1 population activity was recorded using a 96-channel Utah array. V2 population activity was recorded using a set of movable electrodes and tetrodes. (**b**) Peristimulus time histograms during the stimulus presentation period, for an example session and stimulus condition. For visualization purposes, neuronal spike trains were first smoothed using a sliding Gaussian window of width 20 ms, and then z-scored to produce normalized firing rates. Neurons are ordered from top to bottom (separately for V1 and V2) according to the time at which their peak firing rate occurs. (**c**) Inter- and intra-areal comparisons. (Left) We applied DLAG to spike counts in V1 (light blue) and V2 (red). (Right) For comparison, we applied DLAG to two equally sized V1 subpopulations (V1a, light blue; V1b, dark blue), randomly selected from the V1 population. Each triangle represents a neuron. Box sizes illustrate typical relative population sizes.

neuronal populations[5]. Indeed, significant across- and within-area latent variables (i.e., latent variables that were selected via cross-validation) were identified consistently across datasets (Fig. 5.2a: single-trial latent time courses from a representative dataset; Fig. 5.3a, top: dimensionalities across all datasets; median dimensionality across areas: 3; within-V1: 14; within-V2: 2).

We further sought to characterize the strength (in addition to the dimensionality) of across- versus within-area activity in each area. We therefore considered the latent variables in V1 and in V2 separately, and computed the fraction of shared variance that each latent variable explained in its corresponding area (see Section 5.4.1; in Fig. 5.2, the amplitude of each latent time course is scaled by this value). Across-area variables explained only a portion of the shared variance in V1 and in V2 (Fig. 5.3b, top; median across-area strengths: 34% in V1; 76% in V2). Interestingly, across-area activity explained more of the shared variance in V2 than in V1 (Fig. 5.3b, top, points above the diagonal). This observation could not be fully attributed to differences in recorded population size or in the total dimensionality of each area (Fig. 5.4). This difference in across-area strength might be a consequence of the cortical layers from which we recorded: much of the activity in the middle layers of V2 is likely driven by V1. The superficial layers of V1, on the other hand, receive input from other sources that do not also project to the middle layers of V2.

Collectively, these observations (Fig. 5.3a,b, top) are consistent with the presence of a communication

subspace between V1 and V2[37], through which only a subset of population activity patterns are shared between the two areas. Our results further suggest that not only does there exist activity in V1 that is not shared with V2 (as reported in [37]), but there also exists activity in V2 that is not shared with V1. By contrast, V1a and V1b do not interact selectively. V1a-V1b "across-population" activity was higher-dimensional than "within-population" activity and V1-V2 across-area activity (Fig. 5.3a, bottom; median dimensionality across populations 11; within-V1a: 2; within-V1b: 1), and accounted for nearly all of the shared variance in V1a and in V1b (Fig. 5.3b, bottom; median across-population strengths: 96% in V1a; 98% in V1b; note also the small amplitudes of the within-population latent time courses in Fig. 5.2b).

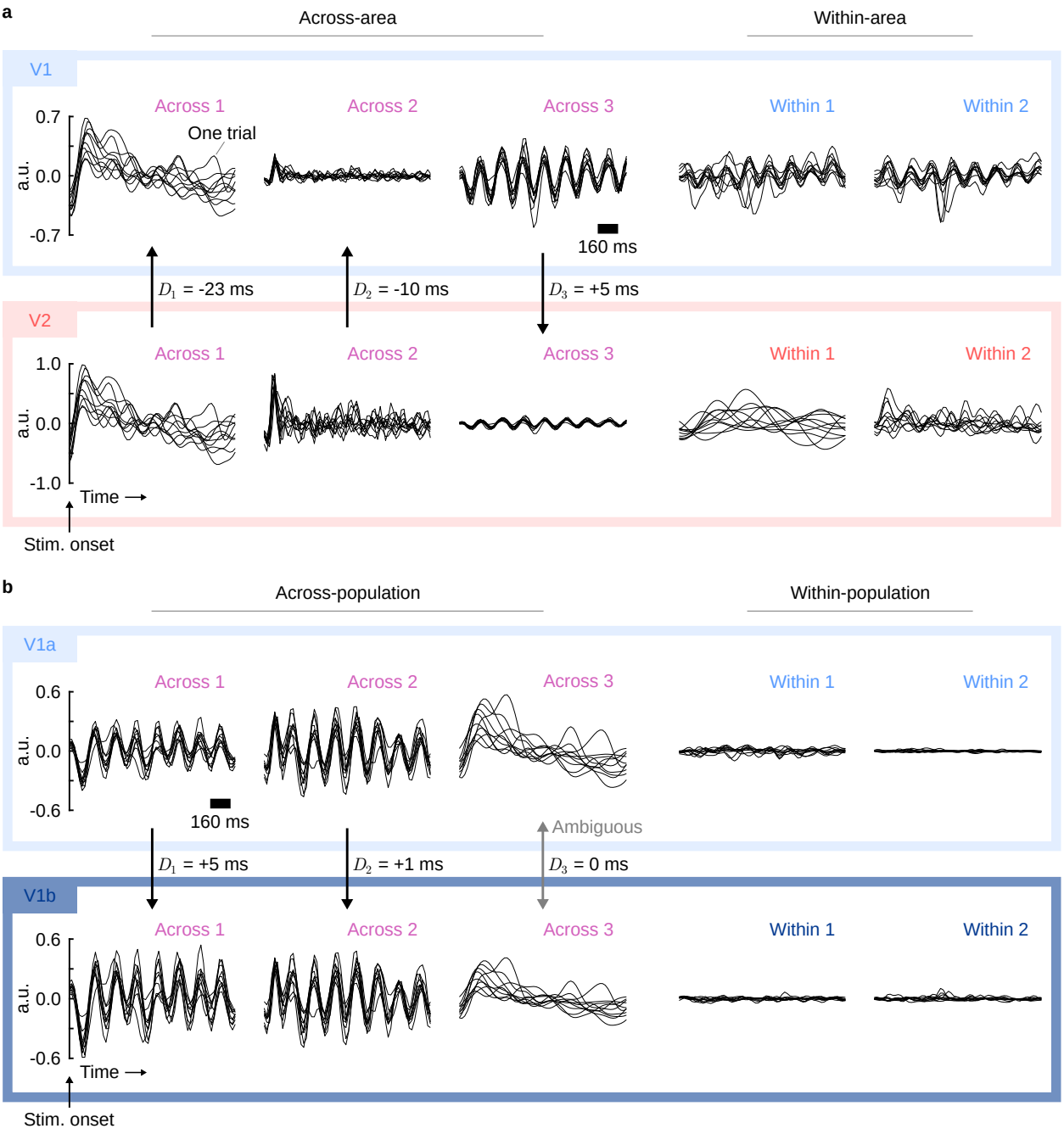

Figure 5.2. Representative DLAG time courses for inter- and intra-areal analyses. (**a**) V1-V2 time courses. Left: Across-area time courses. Right: Within-area time courses. Top row / light blue box: V1. Bottom row / red box: V2. Each panel corresponds to the single-trial time courses of a latent variable. All time courses are aligned to stimulus onset. a.u.: arbitrary units. Each black trace corresponds to one trial; for clarity, only 10 of 400 are shown. Note that the polarity of traces is arbitrary, as long as it is consistent with the polarity of C_i^a or C_i^w. Across-area variables are paired vertically; vertical arrows point in the direction of the identified signal flow, as determined by the sign of the delay next to each arrow. All delays for the displayed dataset were deemed significantly different from zero (see Section 5.4.2). For visualization purposes, latent variables have been scaled and ordered by the fraction of shared variance they explain (across- and within-area variables are sorted separately; across-area variables are sorted according to shared variance explained in V2). All across-area variables and within-V2 variables uncovered by DLAG

are shown here. The top 2 of 14 within-V1 variables are displayed, which explain 46% of V1's within-area shared variance. (**b**) V1a-V1b time courses. Left: Across-population time courses. Right: Within-population time courses. Top row / light blue box: V1a. Bottom row / dark blue box: V1b. All other conventions the same as in (a). Here, the delay for the third across-population variable (Across 3) was deemed to have an ambiguous sign, indicated by the bidirectional gray arrow. All other delays for the displayed dataset were deemed significantly different from zero, indicated by the unidirectional black arrows. Three of 10 across-population variables uncovered by DLAG are shown here, which explain 23% and 17% of V1a's and V1b's total shared variance, respectively. All uncovered within-V1a variables are shown, and 2 of 5 within-V1b variables are shown. Within-population variables (including those not shown here) explained 5% and 7% of V1a's and V1b's total shared shared variance, respectively.

DLAG's latent variables enabled further qualitative characterization of the moment-to-moment nature of within- and across-area activity on individual trials. For instance, stereotyped periodic signals, whose periods matched the period of the drifting grating presented, appeared strongly within V1 (Fig. 5.2a, top, "Across 3", "Within 1", and "Within 2") and only weakly in V2 (Fig. 5.2a, bottom, "Across 3"). The prominence of this stimulus-related periodic structure in V1 relative to V2 is consistent with the stimulus response properties of neurons in each area[61], evident in the neuronal PSTHs (Fig. 5.1b). Care should be taken, however, when interpreting these latent variables as across-area interactions (see Discussion). By contrast, periodic signals were not evident in V1a or V1b within-population variables, but were evident in the activity shared between V1a and V1b (Fig. 5.2b, "Across 1" and "Across 2"). Other latent variables, particularly within V2, exhibited additional trial-to-trial variability whose connection to the presented stimulus is less apparent (for example, Fig. 5.2a, bottom, "Within 1" and "Within 2").

V1-V2 interactions are bidirectional and asymmetric We next used DLAG to study the bidirectional nature of interactions between V1 and V2. Each of DLAG's across-area latent variables is associated with a time delay that indicates a feedforward (positive delay: V1 to V2) or feedback (negative delay: V2 to V1) interaction. For example, the first representative V1-V2 across-area variable (Fig. 5.2a, "Across 1") was associated with a -23 ms delay, implying a feedback interaction. In contrast, the visually similar V1a-V1b across-population variable (Fig. 5.2b, "Across 3") was associated with a 0 ms delay. A V1a-V1b delay at or near zero is expected, given that the V1a and V1b populations belong to the same area, and likely receive common inputs with similar latencies (in contrast to the populations in distinct areas V1 and V2).

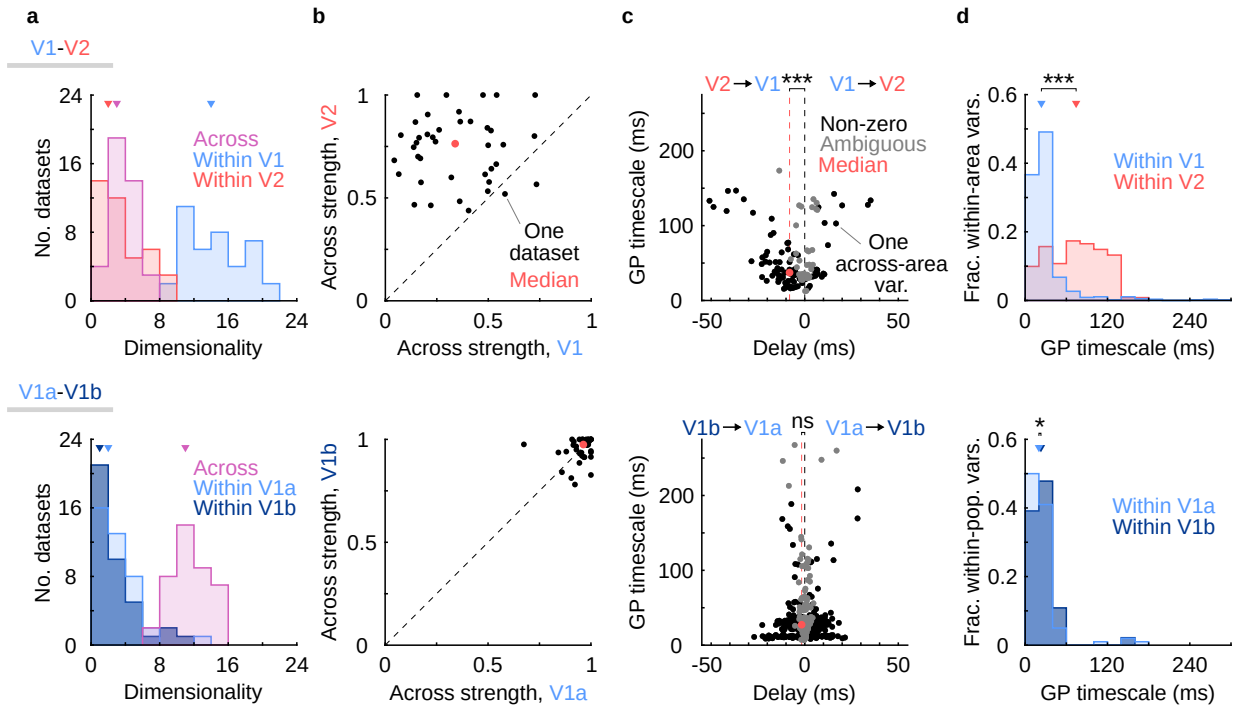

Figure 5.3. DLAG reveals that V1-V2 interactions are selective and asymmetric. (**a**) Within- and across-area dimensionalities (determined via cross-validation). Top: V1-V2 results. Distribution of within-V1, within-V2, and across-area dimensionalities across 40 datasets. Triangles indicate the median of each distribution. Bottom: V1a-V1b results; same format. (**b**) Fraction of shared variance of each area explained by across-area latent variables. Top: V1-V2 results. Across-area strength is significantly greater in V2 than in V1 (one-sided paired sign test; $p = 7.5 \times 10^{-10}$). Bottom: V1a-V1b results; same format. Across-population strength is not significantly greater in one population or the other (two-sided paired sign test; $p = 0.868$). (**c**) Gaussian process (GP) timescale vs. time delay for across-area latent variables. Top: V1-V2 results. Each point represents one across-area latent variable. Black points: across-area latent variables for which the delays were deemed significantly non-zero (see Section 5.4.2; 95 of 135 across-area variables across all 40 datasets). Gray points: across-area latent variables for which delays were deemed ambiguous (not significantly positive or negative; 40 of 135 across-area variables across all 40 datasets). '***': delays are significantly less than zero, representing feedback interactions from V2 to V1 (one-sided one-sample sign test on 'non-zero' delays, $p = 2.4 \times 10^{-7}$). Bottom: V1a-V1b results; same format. Out of 437 across-population latent variables uncovered across all 40 datasets, 316 delays were deemed significantly non-zero, while 121 delays were deemed ambiguous. 'ns': delays are not significantly negative (one-sided one-sample sign test on 'non-zero' delays, $p = 0.08$). (**d**) GP timescales for within-area latent variables. Top: V1-V2 results. Normalized distribution of within-V1 and within-V2 GP timescales across all 40 datasets (total within-V1 latent variables: 562; total within-V2 latent variables: 121). Triangles indicate the median of each distribution. '***': within-V2 GP timescales are significantly longer than within-V1 GP timescales (one-sided Wilcoxon rank sum test, $p = 1.6 \times 10^{-31}$). Bottom: V1a-V1b results; same format (total within-V1a latent variables: 100; total within-V1b latent variables: 92). '*': within-V1b GP timescales are significantly longer than within-V1a GP timescales (one-sided Wilcoxon rank sum test, $p = 0.039$), even though the magnitude of the difference is small (as expected for randomly assigned subpopulations).

We developed a statistical procedure to test whether such delays significantly deviate from zero. In brief, we assessed whether setting the delay to 0 ms resulted in a significant reduction in model performance; if so, the delay was deemed significant (i.e., "non-zero"; see Section 5.4.2). Indeed, the direc-

tionality of this latent variable ("Across 3" for V1a-V1b) was identified as statistically "ambiguous" (i.e. not significantly different from zero, indicated by the bidirectional gray arrow in Fig. 5.2b). In separate analyses, we also verified that V1-V2 interactions are better described by DLAG models with time delays than without time delays (Section 5.5; Fig. 5.9).

Delays across all datasets reflected bidirectional interactions between V1 and V2 (Fig. 5.3c, top). Notably, the delays between V1 and V2 exhibited a striking asymmetry. The interactions across these areas were predominantly directed from V2 to V1 (Fig. 5.3c, top; median over "non-zero" delays: -8 ms; median over all delays: -5 ms). Among the across-area latent variables with statistically significant delays, 76% were associated with a negative delay. This asymmetry remained even when we subsampled the V1 population to match V2 in size, and re-applied DLAG (Fig. 5.4). Like the strength of across-area activity observed in V1 and in V2 (Fig. 5.3b, top), the magnitudes of the delays might also reflect the cortical layers from which we recorded. The positive delays tended to be short (Fig. 5.3c, top; median across significant positive delays: +7 ms), consistent with the fact that the superficial layers of V1 directly project to the middle layers of V2[24,26]. The negative delays tended to be longer (Fig. 5.3c, top; median across significant negative delays: -11 ms), consistent with a multi-synaptic path from the middle layers of V2 back to the superficial layers of V1. We also found that the strongest across-area interactions in V1 were nominally feedforward (V1 to V2), while the strongest across-area interactions in V2 were nominally feedback (V2 to V1) (Fig. 5.5).

By contrast, V1a-V1b interactions were symmetric (Fig. 5.3c, bottom; median over "non-zero" delays: -2 ms; median over all delays: 0 ms; neither median significantly different from zero; 54% of "non-zero" delays were negative; see also Fig. 5.5). This centering of the delay distribution around zero is expected, given that the neurons in V1a and V1b were randomly chosen and belong to the same area. Still, the magnitudes of V1a-V1b delays were not universally zero. These non-zero delays likely reflect aggregate differences in the stimulus response properties of the randomly chosen V1a and V1b subpopulations. For example, inspection of PSTHs (Fig. 5.1b) suggests that the phase of trial-averaged periodic structure can vary by tens of ms between individual V1 neurons.

Finally, we examined the timescales of neural activity identified by DLAG within V1 and V2. Within-V2 Gaussian process (GP) timescales were longer than within-V1 GP timescales (Fig. 5.3d, top; median within-V1: 24 ms; within-V2: 74 ms). Within-V1a and within-V1b GP timescales, on the other hand, were nearly the same (Fig. 5.3d, bottom; median within-V1a: 20 ms; within-V1b: 23 ms). These observations are consistent with previous evidence that timescales increase for areas higher up the cortical hierarchy[62–64].

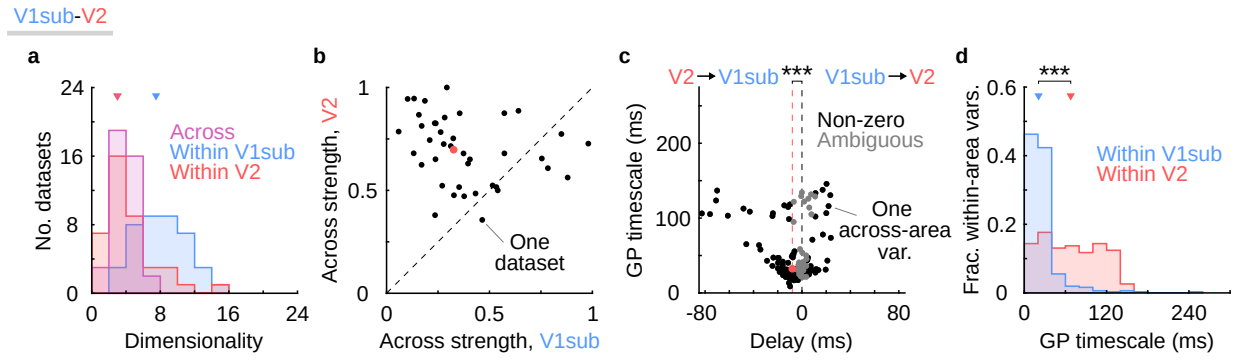

Figure 5.4. V1-V2 results are preserved when V1 is subsampled to match V2 in population size. Same conventions as in Fig. 5.3. We sought to understand the extent to which the results reported in Fig. 5.3 were driven by the fact that V1 populations were larger than V2 populations. All else being equal, more neurons allows one to reliably identify more latent dimensions[65]. For each dataset, we thus randomly subsampled the V1 population ('V1sub') to match the size of the V2 population. We then applied DLAG to each subsampled dataset in the same manner as in Fig. 5.3. (**a**) V1sub-V2 within- and across-area dimensionalities. Compared to Fig. 5.3, median across-area dimensionality (3) was the same. As a consequence of the smaller population size, median within-V1sub dimensionality (7.5) decreased, but remained higher than median across-area and median within-V2 (3) dimensionalities. Within-V2 dimensionality was 0 in 1 of 40 datasets. (**b**) Fraction of shared variance of each area explained by across-area latent variables in V1sub and in V2. Despite population sizes now being the same, across-area strength is still significantly greater in V2 than in V1sub (median V1sub: 0.33; median V2: 0.70; one-sided paired sign test; $p < 0.001$), as in Fig. 5.3. Even after controlling for V1 population size, the within-area dimensionality of V1sub and V2 are not equal. It is possible that the difference in across-area strength seen in (b) is implied by, and therefore redundant with, the difference in within-area dimensionalities seen in (a). Specifically, the weaker across-area strength in V1 relative to V2 might be implied by the greater number of within-V1sub dimensions relative to the number of across-area dimensions. To test this possibility, we recomputed the median across-area strengths for V1 and V2, considering only datasets such that the distributions of within-V1sub and within-V2 dimensionalities were the same. Sixteen datasets remained after this distribution-matching procedure (the 16 datasets for V1 were not necessarily the same 16 datasets as for V2). The medians in V1 and V2 were nearly unchanged (V1sub: 0.33; V2: 0.67). Across-area strengths therefore convey a difference in the properties of V1 versus V2 activity that could not be seen from differences in dimensionality alone. (**c**) Gaussian process (GP) timescale vs. time delay for across-area latent variables. Across all 40 datasets, the delays of 97 of 136 across-area variables were deemed significantly non-zero, and the remaining 39 delays were deemed ambiguous. These values are nearly identical to those reported in Fig. 5.3. Similarly, delays remained significantly less than zero, representing feedback interactions from V2 to V1sub (median delay across all significantly non-zero across-area variables: -8 ms; '***': one-sided one-sample sign test on 'non-zero' delays, $p < 0.001$). Among the significantly non-zero delays, 67% were negative. The magnitude of significant negative delays (median: -12 ms) remained greater than the magnitude of significant positive delays (median: +8ms). (**d**) GP timescales for within-area latent variables. GP timescales within V1sub and within V2 are similar to those reported in Fig. 5.3 (median across 307 within-V1sub latent variables: 21 ms; median across 153 within-V2 latent variables: 68 ms). Furthermore, as in Fig. 5.3, within-V2 GP timescales are significantly longer than within-V1sub GP timescales ('***': one-sided Wilcoxon rank sum test, $p < 0.001$).

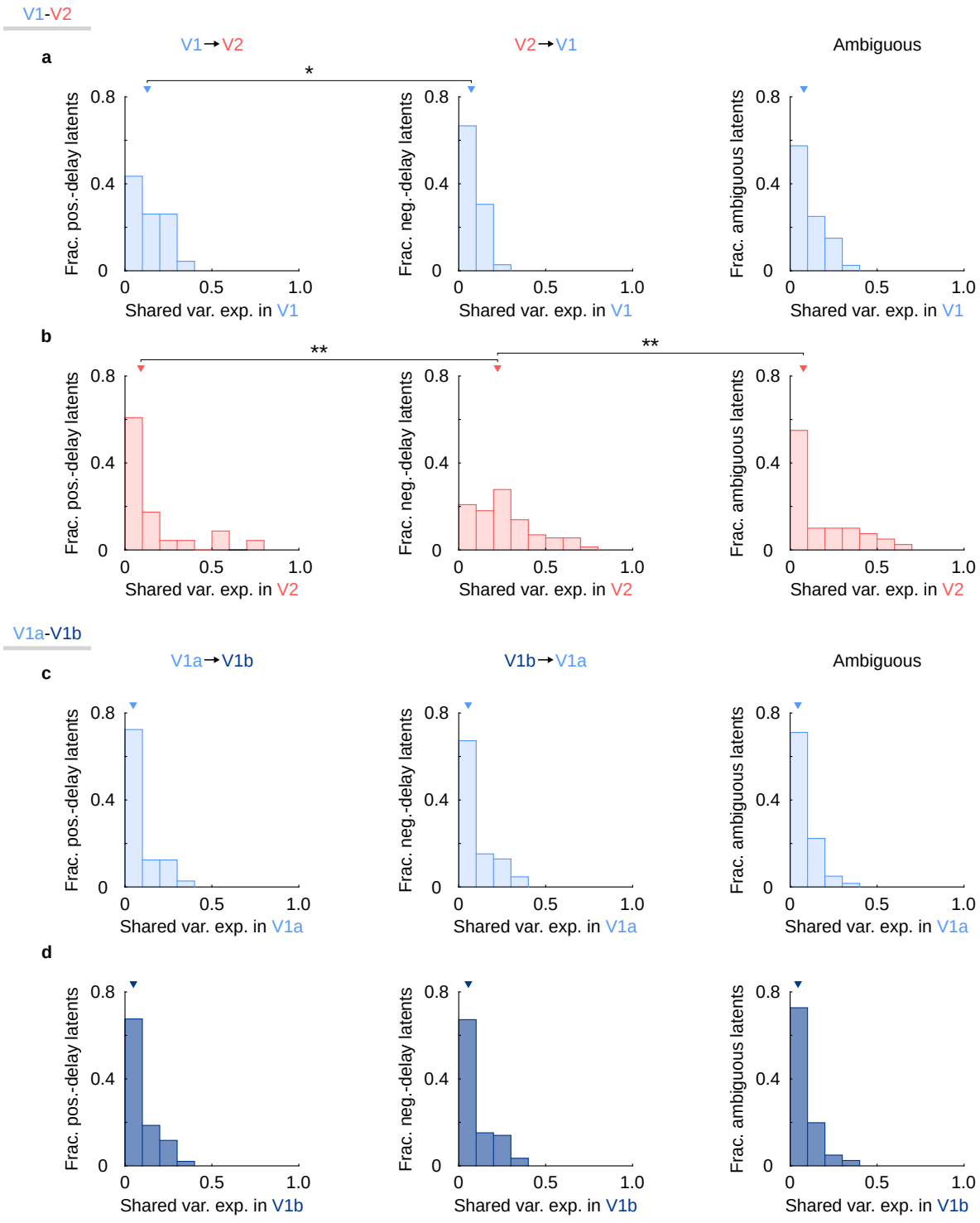

Figure 5.5. The strongest across-area interactions in V1 are nominally feedforward (V1 to V2), while the strongest across-area interactions in V2 are nominally feedback (V2 to V1). (**a**) Normalized distributions of the fraction of shared variance explained in V1 ('Shared var. exp. in V1') by individual across-area latent variables across all 40 datasets. Left: All across-area latent variables with a significant positive delay (V1 to V2). 'Frac. pos.-delay latents': Fraction of positive-delay latent variables. Center: All across-area latent variables with a significant negative delay (V2 to V1). 'Frac. neg.-delay latents': Fraction of negative-delay latent variables. Right: All across-area latent variables with an ambiguous delay (not significantly positive

or negative). '*': Individual positive-delay latent variables explained more shared variance in V1 than individual negative-delay latent variables (one-sided Wilcoxon rank sum test, $p = 0.042$). (**b**) Normalized distributions of the fraction of shared variance explained in V2 ('Shared var. exp. in V2') by individual across-area latent variables across all 40 datasets. Same conventions as in (a). '**': Individual negative-delay latent variables explained more shared variance in V2 than individual positive-delay latent variables and individual latent variables with ambiguous delays (one-sided Wilcoxon rank sum test, $p < 0.01$). (**c**) Normalized distributions of the fraction of shared variance explained in V1a ('Shared var. exp. in V1a') by individual across-population latent variables across all 40 datasets. Left: All across-population latent variables with a significant positive delay (V1a to V1b). Center: All across-population latent variables with a significant negative delay (V1b to V1a). Right: All across-population latent variables with an ambiguous delay (not significantly positive or negative). No type of latent variable explained more or less shared variance in V1a than any other type of latent variable (two-sided Wilcoxon rank sum test, $p > 0.05$ in all cases). (**d**) Normalized distributions of the fraction of shared variance explained in V1b ('Shared var. exp. in V1b') by individual across-population latent variables across all 40 datasets. Same conventions as in (c). No type of latent variable explained more or less shared variance in V1b than any other type of latent variable (two-sided Wilcoxon rank sum test, $p > 0.05$ in all cases).

5.2 Dissecting interactions between V1 and V4

We next used DLAG to study interactions between a second pair of brain regions (visual areas V1 and V4) in an awake animal. In particular, we sought to explore if DLAG, when used to study V1-V4 interactions, was sensitive to the type of stimulus presented: oriented gratings versus naturalistic textures. Previous work has shown that responsivity to higher order statistics of visual stimuli develops gradually along the ventral visual stream. V2 and V4 respond to the higher order statistics present in textures, whereas V1 does not—selective primarily to the spectral content of textures[66,67].

To better understand the effect of stimulus complexity on inter-areal communication, we recorded simultaneous V1 and V4 population responses to gratings and textures while an awake animal was passively fixating (Fig. 5.6a). Array locations were chosen so that receptive fields were largely overlapping for the V1 and V4 populations (see [43]). During recording sessions, two sets of stimuli were presented: a set of sinusoidal gratings and a set of naturalistic textures. Sets of gratings included four stimuli, comprising two spatial frequencies one octave apart and two orientations 90° apart. Sets of textures included four naturalistic texture stimuli (see Section 5.3.2).

Trials began with the animal fixating on a small spot in the center of the screen. After a delay of 300 ms, a random sequence of two stimuli, both from either the grating set or the texture set, appeared on the screen. Each stimulus presentation lasted for 300 ms. The inter stimulus interval was 400 ms (gray screen). After the second stimulus presentation, the animal had to maintain fixation for an additional 300 ms (gray screen) and was then positively reinforced with a liquid reward if fixation was maintained throughout the trial.

When applying DLAG, we treated the two stimulus presentation periods as independent "trials." Our analysis included on average 262±4 presentations per stimulus (four grating stimuli, four texture stimuli) per session. We recorded neural activity for three sessions. For each recording session, we grouped together all trials in which oriented grating stimuli were presented (regardless of orientation or spatial frequency; a "grating stimulus set"), and all trials in which texture stimuli were presented (regardless of texture sample; a "texture stimulus set"). We analyzed 480 ms time windows, from 30 ms after stimulus onset to 210 ms after stimulus offset (hence the analysis time window included some spontaneous neural activity; Fig. 5.6b). We counted spikes in 20 ms time bins during this analysis time window.

We analyzed neuronal responses from one V1 array (60–83 neurons) and from one V4 array (37–54 neurons) that showed the greatest visual receptive field overlap with V1. Note that, for each recording session, V1 and V4 neurons were the same across grating and texture stimulus sets. Both populations responded robustly to each stimulus set (Fig. 5.6b).

Finally, throughout our analyses, we sought to assess the variability of DLAG's estimates within each recording session and stimulus set. For each recording session, we randomly subsampled 20 V1 neurons and 20 V4 neurons from the overall pool of neurons described above. We repeated this subsampling procedure 10 times (starting from the same overall pool of neurons in V1 and in V4). We then applied DLAG separately to each subsample, resulting in 60 separate analyses across the three recording sessions, each with one grating stimulus set and one texture stimulus set. Importantly, the subsampled V1 and V4 neurons were the same across grating and texture stimulus sets, enabling direct comparison between DLAG models. From here on, we refer to these paired grating/texture stimulus sets as simply "stimulus sets."

Indeed, DLAG was sensitive to the type of stimulus presented (Fig. 5.6c–e). In two of three stimulus sets, V1-V4 across-area dimensionality was significantly lower during presentations of texture stimuli than during presentations of oriented grating stimuli (Fig. 5.6c). Furthermore, in all three stimulus sets, V1-V4 across-area prediction (see Section 5.4.3) appeared to be weaker during presentations of texture stimuli than during presentations of oriented grating stimuli (Fig. 5.6d). We then sought to uncover any stimulus dependence in the temporal structure of V1-V4 interactions (Fig. 5.6e). Relative to grating sets, texture sets exhibited a marked absence of across-area GP timescales in the 30–45 ms range. The time delays for across-area variables with GP timescales in the 45–80 ms range appeared to depend on the set of textures presented (Fig. 5.6e, right; points in this range cluster according to texture set).

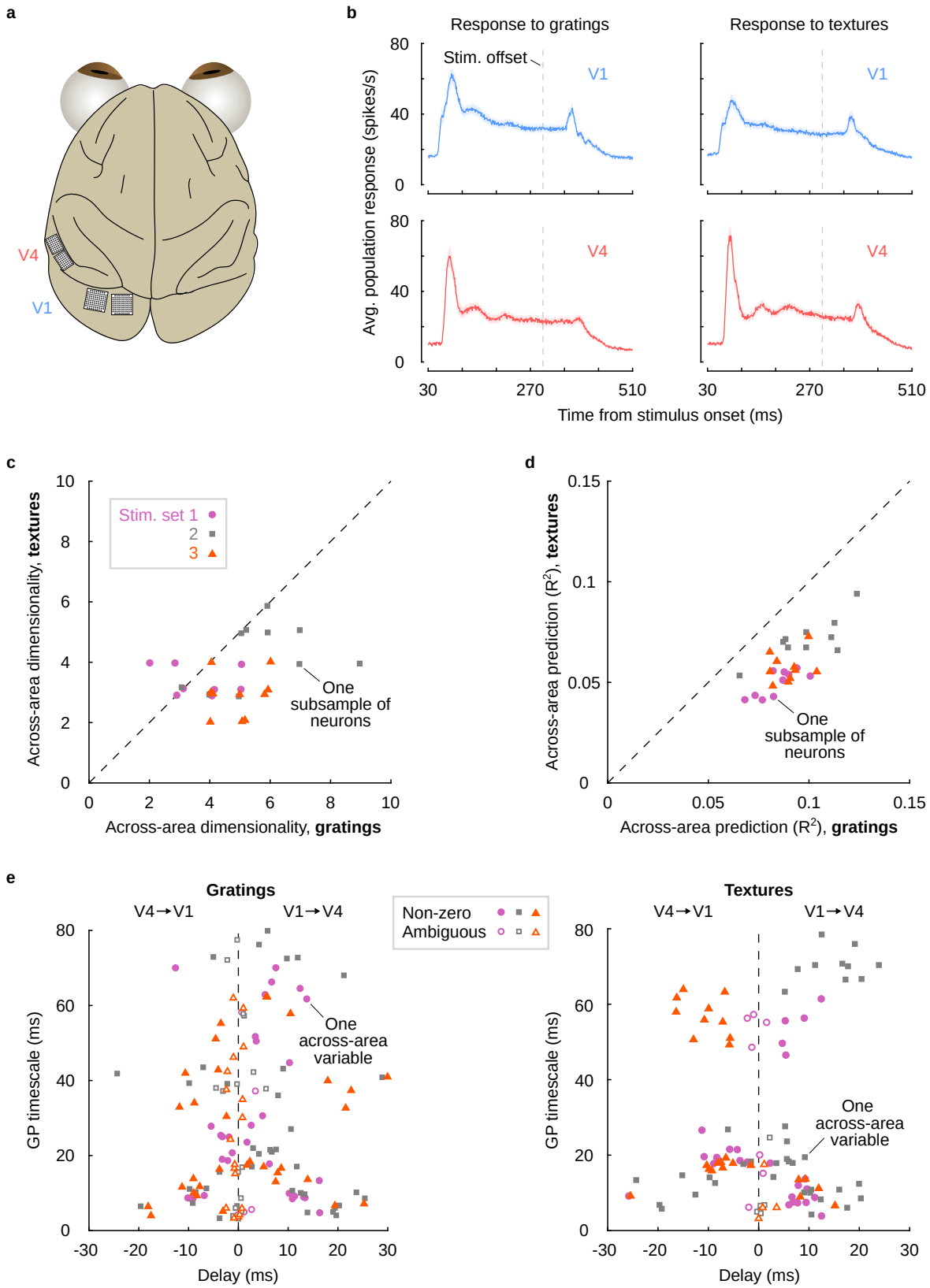

Figure 5.6. DLAG shows that V1-V4 interactions depend on the type of visual stimulus presented. (**a**) Schematic of recording setup. Utah arrays (0.4 mm spacing; 1 mm electrode length, Blackrock, UT) were

implanted in V1 and V4: two 96 channel arrays in V1 and two 48 channel arrays in V4. (**b**) Average population activity in V1 (top row) and V4 (bottom row) in response to an example grating stimulus set (left column) and in response to an example texture stimulus set (right column). These grating and texture stimulus sets correspond to Stimulus Set 2 (gray squares) in panels (c)–(e). Shaded regions indicate ± one SEM, where the mean is taken over peristimulus time histograms (PSTHs) of individual neurons (83 in V1; 54 in V4). The recorded V1 and V4 neurons are the same across the left and right columns. (**c**) V1-V4 across-area dimensionality during the presentation of texture stimuli versus oriented grating stimuli. Each point represents results for a single subsample of V1 and V4 neurons. Data points are integer-valued, but randomly jittered to show points that overlap. In two of three stimulus sets, V1-V4 across-area dimensionality was significantly lower during presentations of texture stimuli than during presentations of oriented grating stimuli (one-sided paired sign test; stimulus set 1, magenta circles: $p = 0.144$; stimulus set 2, gray squares: $p = 0.016$; stimulus set 3, orange triangles: $p = 0.002$). (**d**) Cross-validated across-area prediction (leave-group-out R^2; see Section 5.4.3) between V1 and V4 during the presentation of texture stimuli versus oriented grating stimuli. Each point represents results for a single subsample of V1 and V4 neurons. In all three stimulus sets, V1-V4 across-area prediction appears weaker during presentations of texture stimuli than during presentations of oriented grating stimuli (one-sided paired sign test; for all stimulus sets, $p < 0.001$). (**e**) Gaussian process (GP) timescale vs. time delay for across-area latent variables uncovered during presentations of oriented grating stimuli (left) and during presentations of texture stimuli (right). Each point represents one across-area variable. Filled points: across-area latent variables for which the delays were deemed significantly non-zero. Unfilled points: across-area latent variables for which delays were deemed ambiguous (not significantly positive or negative). In (c)–(e), "Stim. set" refers the paired grating/texture stimulus sets (see Section 5.3.2).

5.3 Experimental methods

5.3.1 V1-V2 anesthetized recordings

Visual stimuli and neural recordings

Animal procedures and recording details have been described in previous work[26,68]. Briefly, animals (macaca fascicularis, young adult males) were anesthetized with ketamine (10 mg/kg) and maintained on isoflurane (1%-2%) during surgery. Recordings were performed under sufentanil (typically 6-18 mg/kg/hr) anesthesia. Vecuronium bromide (150 mg/kg/hr) was used to prevent eye movements. The duration of each experiment (which comprised multiple recording sessions) varied from 5 to 7 days. All procedures were approved by the IACUC of the Albert Einstein College of Medicine.

The data analyzed here are those reported in [37, 43], and a subset of recording sessions reported in [26]. Activity in V1 output layers was recorded using a 96 channel Utah array (400 micron inter-electrode spacing, 1 mm length, inserted to a nominal depth of 600 microns; Blackrock, UT). We recorded V2 activity using a set of electrodes/tetrodes (interelectrode spacing 300 microns) whose depth could be controlled independently (Thomas Recording, Germany). These electrodes were lowered through V1, the underlying white matter, and then into V2. Within V2, we targeted neurons in the input layers. We verified the recordings were performed in the input layers using measurements of the depth in V2 cortex,

histological confirmation (in a subset of recordings), and correlation measurements. For complete details see [68] and [26]. Voltage snippets that exceeded a user-defined threshold were digitized and sorted offline. The sampled neurons had spatial receptive fields within 2-4° of the fovea, in the lower visual field.

We measured responses evoked by drifting sinusoidal gratings (1-1.1 cyc/°; drift rate of 6.25 Hz; 2.6-4.95° in diameter; full contrast, defined as Michelson contrast, $(L_{\max} - L_{\min})/(L_{\max} + L_{\min})$, where $L_{\min}$ is 0 cd/m^2 and $L_{\max}$ is 80 cd/m^2) at 8 different orientations (22.5° steps), on a calibrated CRT monitor placed 110 cm from the animal (1024 × 768 pixel resolution at a 100 Hz refresh rate; Expo: `http://sites.google.com/a/nyu.edu/expo`). Each stimulus was presented 400 times for 1.28 seconds. Each presentation was preceded by an interstimulus interval of 1.5 seconds during which a gray screen was presented.

We recorded neuronal activity in three animals. In two of the animals, we recorded in two different but nearby locations in V2, providing distinct middle-layer populations, yielding a total of five recording sessions. We treated responses to each of the 8 stimuli in each session separately, yielding a total of 40 "datasets."

Data preprocessing

We counted spikes in 20 ms time bins during the 1.28 second stimulus presentation period (64 bins per trial). For all analyses corresponding to each recording session, we excluded neurons that fired fewer than 0.5 spikes/second, on average, across all trials and all grating orientations. Because we were interested in V1-V2 interactions on timescales within a trial, we subtracted the mean across time bins within each trial from each neuron. This step removed activity that fluctuated on slow timescales from one stimulus presentation to the next[69]. We then applied DLAG to each dataset separately.

Intra-areal and subsampled population comparisons

To contrast with the V1-V2 results, we also used DLAG to characterize the interactions between two V1 subpopulations. For each dataset, we randomly split V1 into two equally sized subpopulations (for datasets with an odd number of V1 neurons, we discarded one neuron at random). Each subpopulation was labeled arbitrarily as either "V1a" or "V1b" (Fig. 5.1c). We then applied DLAG to dissect these V1a-V1b interactions in a manner identical to V1-V2 (Fig. 5.2, Fig. 5.3).

We also sought to understand the extent to which the V1-V2 results were driven by disparities in population size between V1 and V2 (Fig. 5.4). For each dataset, we therefore randomly subsampled the V1 population to match the size of the V2 population. We then applied DLAG to each subsampled dataset in the same manner as above.

5.3.2 V1-V4 awake recordings

Visual stimuli and neural recordings

Animal procedures and recording details have been described in previous work[43,70]. Briefly, one male adult cynomolgus macaque was trained to maintain fixation on a small spot (0.2° × 0.2°, 80 cd/m^2) on a gray background (40 cd/m^2) within a 1.4° diameter fixation window. Eye-position was monitored using a video tracking system (Eyelink II, SR research, ON, Canada) with a sampling rate of 500 Hz. Stimuli were presented on a calibrated monitor 64 cm away from the animal (1400 × 1050 pixel resolution; 100 Hz refresh rate).

After training, Utah arrays (0.4 mm spacing; 1 mm electrode length, Blackrock, UT) were implanted in V1 and V4: two 96 channel arrays in V1 and two 48 channel arrays in V4 (see [43]). All procedures were approved by the IACUC of the Albert Einstein College of Medicine. We targeted the arrays to have matching retinotopic locations in V1 and V4 by relying on anatomical markers and previous mapping studies. Receptive fields were in the lower right visual hemifield and largely overlapping for V1 and V4 populations (see [43]). Extracellular voltage signals were amplified and band-pass filtered between 250 and 7.5 kHz using commercial acquisition software (Blackrock Microsystems, UT and Grapevine, Ripple, UT). Voltage snippets that exceeded a user-defined threshold were digitized and sorted offline.

Visual stimuli and task contingencies were presented using custom openGL software (Expo: `http://sites.google.com/a/nyu.edu/expo`). During recording sessions, two sets of stimuli were presented: a set of sinusoidal gratings and a set of naturalistic textures, which included noise stimuli whose spectra were matched to that of a texture. Sets of gratings included four full contrast stimuli, comprising two spatial frequencies one octave apart (1.2-2.4 cyc/°) and two orientations 90° apart (e.g., 1.2 cyc/°, 45°; 2.4 cyc/°, 45°; 1.2 cyc/°, 135°; 2.4 cyc/°, 135°). Sets of textures included six stimuli (four naturalistic texture stimuli, two spectrally matched noise stimuli), generated as follows. Two textures were selected from the Multiband Texture Database (`http://multibandtexture.recherche.usherbrooke.ca/original_brodatz.html`) and Salzburg Texture Image Database (`https://wavelab.at/sources/STex`). The two textures were first down sampled to 256 × 256 pixels and matched in contrast. Then two distinct samples (each 512 × 512 pixels in size) were synthesized for each texture using the Portilla-Simoncelli algorithm[71]. One sample of spectrally matched noise was synthesized for each of the two textures. All stimuli were presented in a 4.7° square aperture.

Trials began with the animal fixating on a small spot in the center of the screen. After a delay of 300 ms, a random sequence of two stimuli, both from either the grating set or the texture set, appeared on the screen. Each stimulus presentation lasted for 300 ms. The inter stimulus interval was 400 ms (gray screen).

After the second stimulus presentation, the animal had to maintain fixation for an additional 300 ms (gray screen) and was then positively reinforced with a liquid reward if fixation was maintained throughout the trial. The animal performed on average 1307±15 trials per session. We recorded neural activity for three sessions.

Data preprocessing

We were interested in observing whether DLAG was sensitive to the presentation of grating versus texture stimuli. Hence for further analysis, we excluded presentations of spectrally matched noise stimuli. As stated above, each trial comprised two stimulus presentation periods: we treated these periods as independent "trials" when applying DLAG. Our analysis included on average 262±4 presentations per stimulus (four grating stimuli, four texture stimuli) per session. For each recording session, we grouped together all trials in which oriented grating stimuli were presented (regardless of orientation or spatial frequency; a "grating stimulus set"), and all trials in which texture stimuli were presented (regardless of texture sample; a "texture stimulus set"). We analyzed 480 ms time windows, from 30 ms after stimulus onset to 210 ms after stimulus offset (hence the analysis time window included some spontaneous neural activity). We counted spikes in 20 ms time bins during this analysis time window.

We analyzed neuronal responses from one V1 array and from one V4 array that showed the greatest visual receptive field overlap with V1. For each recording session, we excluded neurons that fired fewer than 0.5 spikes/second, on average, for any given stimulus condition. We also excluded neurons with a Fano factor greater than 1.6, on average, across all stimulus conditions (Fano factor was computed across trials of one stimulus condition at a time). Following these screening steps, sessions 1, 2, and 3 contained pools of 60, 83, and 77 neurons, respectively, in V1, and pools of 44, 54, and 37 neurons, respectively, in V4. Note that, for each recording session, V1 and V4 neurons were the same across grating and texture stimulus sets. Because we were interested in V1-V4 interactions on timescales within a trial, we subtracted the mean across time bins within each trial from each neuron. This step removed activity that fluctuated on slow timescales from one stimulus presentation to the next.

Subsampling of neuronal populations

Finally, throughout our analyses, we sought to assess the variability of DLAG's estimates within each recording session and stimulus set. For each recording session, we randomly subsampled 20 V1 neurons and 20 V4 neurons from the overall pool of neurons described above. We repeated this subsampling procedure 10 times (starting from the same overall pool of neurons in V1 and in V4). We then applied DLAG separately to each subsample, resulting in 60 separate analyses across the three recording sessions,

each with one grating stimulus set and one texture stimulus set. Importantly, the subsampled V1 and V4 neurons were the same across grating and texture stimulus sets, enabling direct comparison between DLAG models.

5.4 DLAG-derived descriptive and inferential statistics

5.4.1 Variance explained by DLAG latent variables

After fitting a DLAG model to each experimental dataset, we sought to compare the relative strengths of across- or within-area latent variables extracted from the same dataset (as in Fig. 5.2) and across different datasets (as in Fig. 5.3b, Fig. 5.4b). To quantify these comparisons, we computed the variance each latent variable explained, as derived from fitted model parameters. From equation (3.1), the total variance in area m simplifies to

$$\mathrm{var}_{\mathrm{total}} = \mathrm{tr}\left(C_m^a C_m^{a\top} + C_m^w C_m^{w\top} + R_m\right) \tag{5.1}$$

By inspection, the total variance decomposes into three separable components: $\mathrm{tr}(C_m^a C_m^{a\top})$, the variance due to across-area activity; $\mathrm{tr}(C_m^w C_m^{w\top})$, the variance due to within-area activity; and $\mathrm{tr}(R_m)$, the variance that is independent to each neuron. In fact, the across-area and within-area components can be decomposed further into contributions by individual latent variables. Let $\mathbf{c}_{m,j}^a \in \mathbb{R}^{q_m}$ be the j^{th} column of C_m^a, and $\mathbf{c}_{m,j}^w \in \mathbb{R}^{q_m}$ be the j^{th} column of C_m^w. Then, $\mathrm{tr}(C_m^a C_m^{a\top}) = \sum_{j=1}^{p^a} \|\mathbf{c}_{m,j}^a\|_2^2$, and $\mathrm{tr}(C_m^w C_m^{w\top}) = \sum_{j=1}^{p_m^w} \|\mathbf{c}_{m,j}^w\|_2^2$.

Because we were interested in variance shared among neurons, rather than independent to each neuron, we focused on the variance components involving C_m^a and C_m^w, rather than R_m. Furthermore, since the total variance of recorded neural activity may vary widely across animals, stimuli, and recording sessions, we computed two normalized metrics to facilitate comparison of these shared variance components across datasets. First, let $\mathbf{c}_{m,j}$ be the j^{th} column of C_m, where $C_m = [C_m^a \; C_m^w]$ is the same as in equation (3.12). To visualize the relative strength of latent variables in each area (Fig. 5.2), we computed

$$\alpha_{m,j} = \frac{\|\mathbf{c}_{m,j}\|_2^2}{\mathrm{tr}\left(C_m^a C_m^{a\top} + C_m^w C_m^{w\top}\right)} \tag{5.2}$$

that is, the fraction of shared variance explained by latent variable j in area m. We then displayed latent time courses multiplied by the appropriate $\alpha_{m,j}$ at each time point. Similarly, to quantify the strength of across-area activity (relative to within-area activity) in each area (Fig. 5.3b), we computed

$$\alpha_m^a = \frac{\mathrm{tr}\left(C_m^a C_m^{a\top}\right)}{\mathrm{tr}\left(C_m^a C_m^{a\top} + C_m^w C_m^{w\top}\right)} \tag{5.3}$$

that is, the fraction of shared variance explained by all across-area latent variables in area m.

5.4.2 Uncertainty of estimated delays

DLAG's performance on the synthetic data presented here suggests that time delays are estimated with high accuracy and precision. For our neural recordings, however, where no "ground truth" is accessible, we sought to assess the certainty with which fitted delay parameters were indeed positive or negative—indicating a particular direction of inter-areal signal flow. We therefore developed the following nonparametric bootstrap procedure.

First, consider a DLAG model that has been fit to a particular dataset with N trials. We construct a bootstrap sample $b = 1, \ldots, B$ from this dataset by selecting N trials uniformly at random with replacement (here we used $B = 1,000$). Then, let ℓ_b be the data log-likelihood of the DLAG model evaluated on bootstrap sample b. And let $\ell_{b,j=0}$ be the data log-likelihood of the same DLAG model evaluated on bootstrap sample b, but for which D_j, the delay for across-area latent variable j, has been set to zero (all other model parameters remain unaltered).

To compare the performance of this "zero-delay" model to the performance of the original model, we define the following statistic:

$$\Delta\ell_{b,j=0} = \ell_b - \ell_{b,j=0} \tag{5.4}$$

If the zero-delay model performed at least as well as the original DLAG model (equivalently, $\Delta\ell_{b,j=0} \leq 0$) on 5% or more of the bootstrap samples, then we could not say, with sufficient certainty, that the delay for across-area variable j was strictly positive or strictly negative. Otherwise, we took the magnitude of the delay for across-area variable j to differ significantly from zero.

For each of our V1-V2 datasets, then, this procedure allowed us to label some delays as "ambiguous," where the corresponding population signal could not be confidently categorized as flowing in one direction or the other (Fig. 5.3c, Fig. 5.4c, Fig. 5.6e). Finally, note that the concept of ambiguity defined here is distinct from the concept of a variable's importance in describing observed neural activity: for example, an across-area variable with an ambiguous time delay between areas could, in principle, still explain a large portion of an area's shared variance.

5.4.3 Across-area prediction

As described in Section 3.6, we selected the number of within- and across-area latent variables for DLAG models using cross-validated data log-likelihood (from equation (3.21)). Cross-validated data log-likelihood (LL) offers a principled performance metric, as it is precisely the (training) data LL that a fitted DLAG model maximizes, and it fits within DLAG's probabilistic framework. However, interpretation of the relative performance differences between models can be difficult given the scale of LL values.

Furthermore, LL values can vary dramatically from dataset to dataset, often by orders of magnitude. We therefore sought an alternative metric that facilitates more intuitive comparison between models/methods (see Section 5.5, Fig. 5.7, Fig. 5.9) and across datasets (Fig. 5.6).

Toward that end, we developed a leave-group-out prediction procedure that measures a model's ability to capture interactions across areas (similar to the leave-neuron-out prediction procedure in [47]). Our goal, therefore, is to use a fitted model to predict the unobserved activity of held-out neurons in one area, given the observed activity of neurons in the other area. Let us first collect observed variables (for one trial) in a manner that highlights group structure. We define $\widetilde{\mathbf{y}}_1 = [\mathbf{y}_{1,1}^\top \cdots \mathbf{y}_{1,T}^\top]^\top \in \mathbb{R}^{q_1 T}$ and $\widetilde{\mathbf{y}}_2 = [\mathbf{y}_{2,1}^\top \cdots \mathbf{y}_{2,T}^\top]^\top \in \mathbb{R}^{q_2 T}$, obtained by vertically concatenating the observed neural activity $\mathbf{y}_{1,t}$ and $\mathbf{y}_{2,t}$ in areas 1 and 2, respectively, across all times $t = 1, \ldots, T$.

To predict $\widetilde{\mathbf{y}}_2$ from $\widetilde{\mathbf{y}}_1$, we use the conditional distribution of $\widetilde{\mathbf{y}}_2$ given $\widetilde{\mathbf{y}}_1$, $P(\widetilde{\mathbf{y}}_2|\widetilde{\mathbf{y}}_1)$, which can be obtained from the joint distribution $P(\widetilde{\mathbf{y}}_1, \widetilde{\mathbf{y}}_2)$. For a derivation and discussion of the joint distribution, $P(\widetilde{\mathbf{y}}_1, \widetilde{\mathbf{y}}_2)$, see Section 3.7 (equation (3.42)). From the conditional distribution, $P(\widetilde{\mathbf{y}}_2|\widetilde{\mathbf{y}}_1)$, we take predictions to be the expected value of activity in area 2 given activity in area 1:

$$\hat{\widetilde{\mathbf{y}}}_2 = \mathbb{E}\left[\widetilde{\mathbf{y}}_2 \mid \widetilde{\mathbf{y}}_1\right] = \widetilde{C}_2^a \widetilde{K}_{2,1}^a \widetilde{C}_1^{a\top} (\widetilde{C}_1^a \widetilde{K}_{1,1}^a \widetilde{C}_1^{a\top} + \widetilde{C}_1^w \widetilde{K}_1^w \widetilde{C}_1^{w\top} + \widetilde{R}_1)^{-1} (\widetilde{\mathbf{y}}_1 - \widetilde{\mathbf{d}}_1) + \widetilde{\mathbf{d}}_2 \tag{5.5}$$

where $\widetilde{C}_1^a \in \mathbb{R}^{q_1 T \times p^a T}$, $\widetilde{C}_1^w \in \mathbb{R}^{q_1 T \times p_1^w T}$, $\widetilde{C}_2^a \in \mathbb{R}^{q_2 T \times p^a T}$, $\widetilde{C}_2^w \in \mathbb{R}^{q_2 T \times p_2^w T}$, $\widetilde{R}_1 \in \mathbb{S}^{q_1 T \times q_1 T}$, and $\widetilde{R}_2 \in \mathbb{S}^{q_2 T \times q_2 T}$ are all block diagonal matrices comprising T copies of the loading matrices C_1^a, C_1^w, C_2^a, and C_2^w, and observation noise covariance matrices R_1 and R_2, respectively. $\widetilde{\mathbf{d}}_1 \in \mathbb{R}^{q_1 T}$ and $\widetilde{\mathbf{d}}_2 \in \mathbb{R}^{q_2 T}$ are constructed by vertically concatenating T copies of mean parameters $\mathbf{d}_1$ and $\mathbf{d}_2$, respectively. The Gaussian process covariance matrices $\widetilde{K}_1^w \in \mathbb{S}^{p_1^w T \times p_1^w T}$, $\widetilde{K}_{1,1}^a \in \mathbb{R}^{p^a T \times p^a T}$, and $\widetilde{K}_{2,1}^a \in \mathbb{R}^{p^a T \times p^a T}$ are defined in equations (3.40) and (3.41) of Section 3.7. We similarly predict $\widetilde{\mathbf{y}}_1$ from $\widetilde{\mathbf{y}}_2$ using $\mathbb{E}\left[\widetilde{\mathbf{y}}_1|\widetilde{\mathbf{y}}_2\right]$.

We next use equation (5.5) to define a cross-validated measure of a model's across-area predictive performance. Assume we are given the parameters of a DLAG model fit to training data (equation (3.9)). Then let $\widetilde{\mathbf{y}}_{m,n}$ be the activity of area m on trial n of a held-out validation set, and let $\hat{\widetilde{\mathbf{y}}}_{m,n}$ be its predicted value given by equation (5.5). Collect these values across all $n = 1, \ldots, N$ held-out validation set trials into the respective matrices $Y_m = [\widetilde{\mathbf{y}}_{m,1} \cdots \widetilde{\mathbf{y}}_{m,N}] \in \mathbb{R}^{q_m T \times N}$ and $\hat{Y}_m = [\hat{\widetilde{\mathbf{y}}}_{m,1} \cdots \hat{\widetilde{\mathbf{y}}}_{m,N}] \in \mathbb{R}^{q_m T \times N}$. We then define a leave-group-out R^2 value as follows:

$$R^2_{\text{lgo}} = 1 - \frac{\|Y_1 - \hat{Y}_1\|_F^2 + \|Y_2 - \hat{Y}_2\|_F^2}{\|Y_1 - \bar{Y}_1\|_F^2 + \|Y_2 - \bar{Y}_2\|_F^2} \tag{5.6}$$

where $\bar{Y}_m = [\bar{\mathbf{y}}_m \cdots \bar{\mathbf{y}}_m] \in \mathbb{R}^{q_m T \times N}$ is constructed by horizontally concatenating N copies of the sample mean for each neuron in observations Y_m, taken over all time points and trials ($\bar{\mathbf{y}}_m \in \mathbb{R}^{q_m T}$). In K-fold cross-validation, we evaluate R^2_{lgo} on each of the K validation sets, and report the average value over all K.

In a typical multivariate regression setting, R^2 is an asymmetric measure of predictive performance: prediction of $\widetilde{\mathbf{y}}_2$ from $\widetilde{\mathbf{y}}_1$ yields a different R^2 value than does prediction of $\widetilde{\mathbf{y}}_1$ from $\widetilde{\mathbf{y}}_2$. In contrast, R^2_{lgo} is a symmetric measure that aggregates predictions in both directions. Like R^2, $R^2_{lgo} \in (-\infty, 1]$, where a value of 1 implies perfect prediction of neural activity, and a negative value implies that estimates predict neural activity less accurately than simply the sample mean. R^2_{lgo} is normalized by the total variance of neural activity within each dataset, thereby facilitating comparison across datasets, in which the variance of neural activity could vary widely. This more intuitive comparison across datasets (compared to LL) comes at the expense of a principled characterization of performance within DLAG's probabilistic framework, and we emphasize that across-area prediction is not the objective that a fitted DLAG model is designed to maximize.

5.5 Empirical comparisons of DLAG to other statistical methods

5.5.1 Quantitative comparison of DLAG to pCCA

To demonstrate the advantages of modeling the temporal structure of neuronal interactions within and across areas, we applied probabilistic canonical correlation analysis (pCCA)[52] to the same V1-V2 datasets as in Fig. 5.3. pCCA is a static dimensionality reduction method that includes across-area latent variables, but not within-area latent variables (see Section 3.5, equations (3.36) and (3.37)). For each of the 40 V1-V2 datasets, we identified the number of pCCA latent variables through K-fold cross-validation (here we chose $K = 4$, as was done for DLAG cross-validation). The pCCA model with the highest cross-validated data likelihood was taken as optimal.

We first compared the optimal across-area dimensionalties of each method. pCCA and DLAG estimates of across-area dimensionality were modestly correlated (Fig. 5.7a, top; Pearson correlation coefficient, $r = 0.48$), and pCCA estimates were slightly higher than DLAG estimates (Fig. 5.7a, bottom; median difference across datasets: 0.5; one-sided paired sign test: $p = 0.0494$).

We then compared the optimal pCCA model to the optimal DLAG model on each dataset (each selected through cross-validation) via two performance metrics: cross-validated data log-likelihood (LL; Fig. 5.7b) and cross-validated leave-group-out R^2 (Fig. 5.7c; see Section 5.4.3). Cross-validated LL offers the most principled comparison, as it is precisely the data log-likelihood that the two probabilistic methods are intended to maximize. However, interpretation of the relative performance differences between methods can be difficult given the scale of LL values. Furthermore, LL values can vary dramatically from dataset to dataset, often by orders of magnitude. Hence leave-group-out R^2 facilitates more intuitive comparison between methods and across datasets, at the expense of a principled characterization of performance

within each method's probabilistic framework. DLAG significantly outperformed pCCA across datasets (Fig. 5.7b,c; one-sided paired sign test, $p < 0.001$).

DLAG's better performance can be attributed to multiple differences between the DLAG and pCCA models. First, DLAG includes the addition of low-dimensional within-area latent variables. pCCA models within-area activity via full-rank observation noise covariance matrices (see equation (3.37)). Fig. 5.3a suggests that within-area activity in both V1 and in V2 is well-described as low-dimensional. Second, the number of parameters in the DLAG model scales linearly with the number of neurons in each area, whereas the number of parameters in the pCCA model scales quadratically with the number of neurons in each area, lending pCCA to be more prone to overfitting. Third, DLAG accounts for the temporal structure of within- and across-area interactions (using Gaussian processes), whereas pCCA does not. Fourth, DLAG accounts for time delays in across-area interactions, whereas pCCA does not.

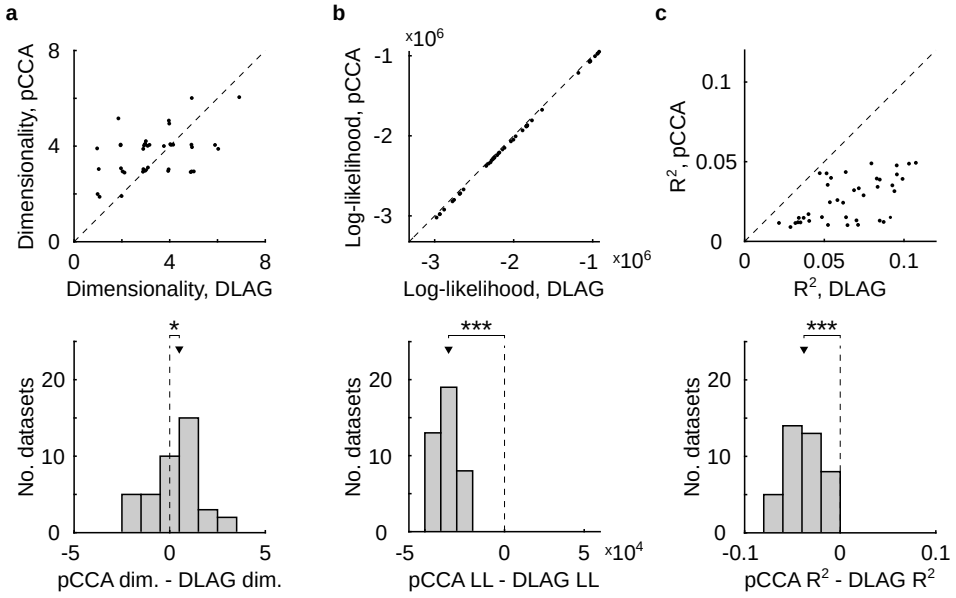

Figure 5.7: V1-V2 interactions are better described by DLAG than by probabilistic canonical correlation analysis. (**a**) Comparison of pCCA and DLAG across-area dimensionality estimates. Top: Estimated pCCA dimensionality versus estimated DLAG across-area dimensionality. Each data point represents one V1-V2 dataset. Data points are integer-valued, but randomly jittered to show points that overlap. pCCA and DLAG estimates of across-area dimensionality are modestly correlated (Pearson correlation coefficient, $r = 0.48$). Bottom: Distribution of the differences between pCCA and DLAG across-area dimensionality ('dim.') estimates on each dataset. pCCA estimates are slightly higher than DLAG estimates (black triangle indicates the median difference across datasets: 0.5; '*': one-sided paired sign test; $p = 0.0494$). (**b**)–(**c**) DLAG outperforms pCCA according to multiple metrics (panel (b): LL; panel (c): leave-group-out R^2). Top panels: pCCA performance versus DLAG performance. Each data point represents one V1-V2 dataset. Bottom panels: Distribution of differences between pCCA and DLAG performance on each dataset. DLAG significantly outperforms pCCA across datasets (black triangles indicate the median difference across datasets; '***': one-sided paired sign test; $p < 0.001$).

5.5.2 Qualitative comparison of DLAG to pCCA

Here we consider the V1-V2 recordings, and explore the qualitative differences between a static method like CCA and DLAG, particularly in their descriptions of inter-areal signal flow. We thus considered the same V1-V2 dataset as presented in Fig. 5.2, and studied the projections of V1-V2 neural activity onto the across-area dimensions obtained via CCA.

The top canonical variable is dominated by feedback (V2 leads V1) activity, even if CCA is fit to V1-V2 activity with a nominal feedforward (V1 leads V2) time-shift. One approach to using CCA to identify the direction of inter-areal signal flow was recently proposed in [43]. There, a sliding window scheme was used, in which observations of V2 activity were first time-shifted relative to observations of V1 activity, and then CCA was fit to this time-shifted V1-V2 activity. CCA was fit anew for each incremental advance of the sliding window throughout the course of the trial, thereby producing a different set of canonical dimensions for each relative time shift between V1 and V2 activity. The top canonical dimensions were then studied at various time delays and at various time points throughout the trial to identify periods of feedforward- and feedback-dominated activity.

In Fig. 4.9, we showed that the top canonical dimension—when fit to simultaneous observations—reflects either the dominant direction of interaction (Fig. 4.9a) or a mixture of signals relayed in both directions (Fig. 4.9b). Could one tease apart concurrent feedforward and feedback signals by instead fitting the top canonical dimension to time-shifted V1-V2 activity, as in [43]? One might expect, for example, that a feedforward interaction becomes dominant in V1-V2 activity after imposing a "feedforward" time shift. Then in principle, the top canonical dimension identified from this time-shifted activity could reflect such a feedforward interaction (resembling, for example, DLAG's Across 3 in Fig. 5.2a, a nominally feedforward latent variable). One could analogously find the top canonical dimension for "feedback-shifted" V1-V2 activity to reveal a feedback interaction (resembling, for example, DLAG's Across 1 in Fig. 5.2a, a nominally feedback latent variable).

To investigate whether this expectation holds in the V1-V2 recordings, we employed a scheme similar to that of [43] (but modified to better facilitate comparison with DLAG), and studied how projections of V1 and V2 activity onto the top canonical dimension qualitatively change as CCA is fit to V1-V2 activity with different relative time shifts. Specifically, we first took a fixed window of activity in V1, 1240 ms in length, from 20 ms to 1260 ms after stimulus onset. We counted spikes within this window in 20 ms nonoverlapping time bins. For V2, we considered three different (overlapping) time windows, each 1240 ms in length: from 0 ms to 1240 ms after stimulus onset, from 20 ms to 1260 ms after stimulus onset, and from 40 ms to 1280 ms after stimulus onset. In each of these windows, we counted spikes in 20 ms

nonoverlapping time bins. We then fit a separate CCA model between the fixed window of activity in V1 and each of the three windows of activity in V2. Then for each fitted model, we projected V1 and V2 neural activity onto the top canonical pair of dimensions. We found that the projected time courses showed no appreciable differences across the three time-shifted model fits (Fig. 5.8a).

Even though the time courses in each of the three cases look similar, do they reflect signal flow in different directions? To address this question, we estimated a time delay for each pair of fitted canonical dimensions using the same procedure as in Fig. 4.9: we identified the time delay at which projections of V1 activity and projections of V2 activity had maximum cross-correlation. The cross-correlation function between V1 and V2 projections was computed with 1 ms resolution, from -40 ms (V2 leads V1) to +40 ms (V1 leads V2). In detail, we first took a fixed window of activity in V1, 1200 ms in length, from 40 ms to 1240 ms after stimulus onset. For each trial, we counted spikes within this window in 20 ms nonoverlapping time bins, and projected this activity onto each canonical dimension in V1. For V2, we employed a sliding window of length 1200 ms, which we advanced in 1 ms increments, from 0 ms to 1280 ms after stimulus onset. At each increment, we counted spikes within the window in 20 ms nonoverlapping time bins, and projected this activity (on each trial) onto each canonical dimension in V2. For each canonical pair, we computed the Pearson correlation between the projected V1 activity and the projected V2 activity. This correlation value gave one element of a cross-correlation function: repeating this procedure at each increment of the sliding window in V2 produced a cross-correlation function from -40 ms to +40 ms. We then identified the time delay at which the cross-correlation function for each canonical pair was maximum.

For the canonical pair fit to V1-V2 observations with a -20 ms time shift (Fig. 5.8a, left panel), the identified time delay is indeed negative—but so are the time delays identified in the other two cases. Here, a feedback interaction is dominant, and its cross-correlation (a function of the relative time lag between V1 and V2) decays sufficiently slowly that it remains dominant over a wide range of time lags. Thus the top canonical pair reflects this dominant feedback interaction even when fit to feedforward-shifted V1-V2 activity (Fig. 5.8a, right panel). This phenomenon demonstrates the challenge of using a static method like CCA, as we have done here (see also [43]), to disentangle concurrent, bidirectional interactions across areas. We note that, in contrast to the results demonstrated here, [43] found bidirectional (though not concurrent) signals because a much smaller analysis time window was used (80 ms), which enabled the characterization of feedforward- and feedback-dominated trial periods. The concepts demonstrated here still apply within each of those trial periods.

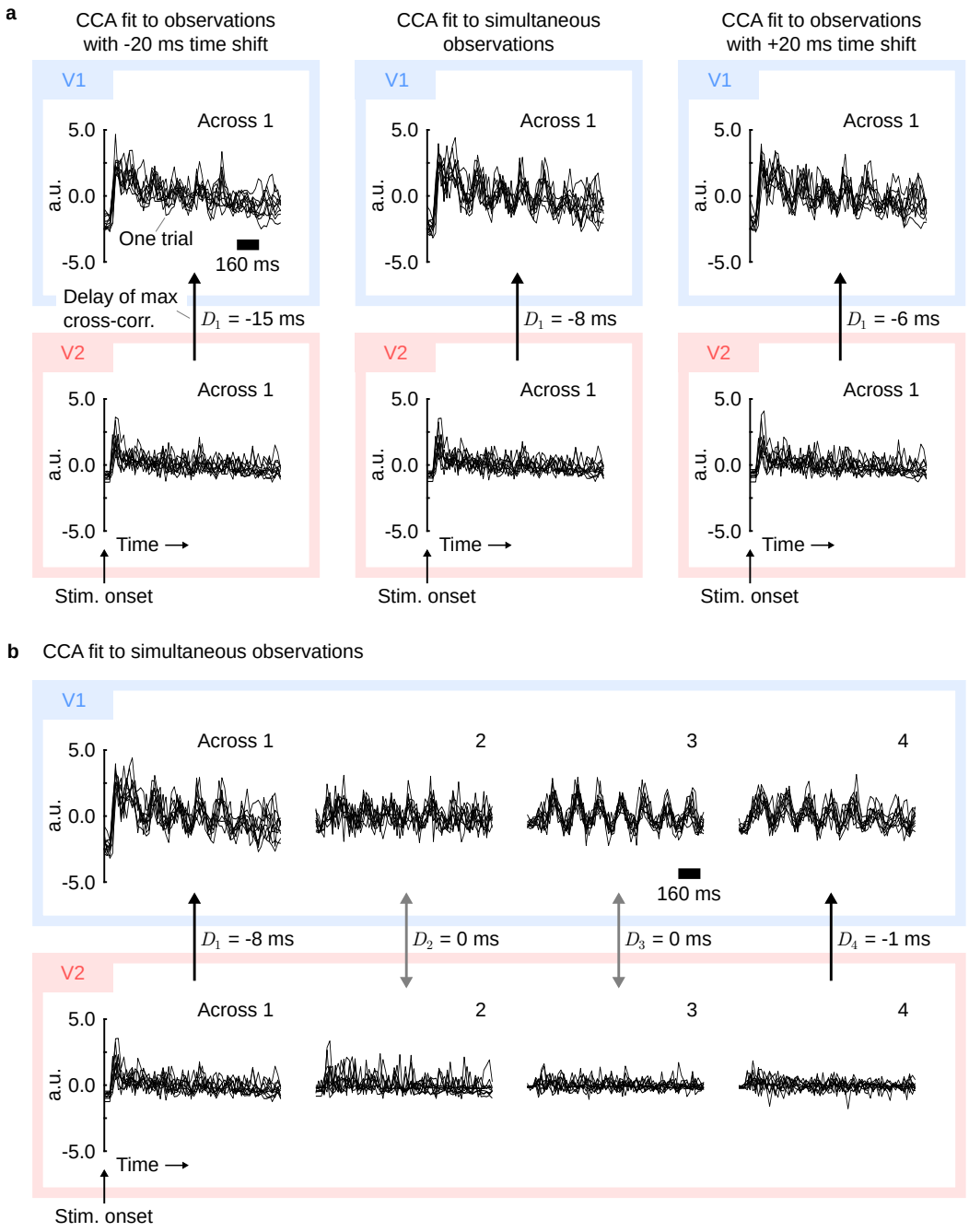

Figure 5.8. Canonical correlation analysis (CCA) provides a description of V1-V2 signal flow that is qualitatively different from that of DLAG. (**a**) The top canonical variable is dominated by feedback (V2 leads V1) activity, even if CCA is fit to V1-V2 activity with a nominal feedforward (V1 leads V2) time-shift. Left: A CCA model was fit to time-shifted activity, in which V2 activity was shifted to lead V1 activity by 20 ms (-20 ms delay). Center: A CCA model was fit to simultaneously observed V1 and V2 activity. Right: A CCA model was fit to time-shifted activity, in which V2 activity was shifted to lag V1 activity by 20 ms (+20 ms delay). Top row / blue box: V1. Bottom row / red box: V2. Each black trace corresponds to one trial; for clarity, only 10 of 400 are shown. All time courses are aligned to stimulus onset. a.u.: arbitrary units. (**b**) Canonical variables fit to simultaneously observed activity indicate only predominant feedback (V2 to V1) activity or zero-lag activity. Canonical variables are paired vertically, and ordered from left to right according to descending canonical correlation value. All other conventions are the same as in (a).

Canonical variables fit to simultaneously observed activity indicate only predominant feedback (V2 to V1) activity or zero-lag activity. We again considered the CCA model fit to simultaneously observed activity (Fig. 5.8a, center), and sought to assess the direction of signal flow associated with all significant canonical pairs selected via cross-validation (see Fig. 5.7). We estimated a time delay for each canonical pair using the same procedure as described for Fig. 4.9 and for Fig. 5.8a.

The first canonical pair (Fig. 5.8b, Across 1) was associated with a negative (V2 to V1) delay, similar to DLAG's Across 1 and Across 2 in Fig. 5.2a. But notably, the remaining canonical pairs were associated with time delays at or near 0 ms, whereas DLAG identified a similarly periodic signal with a time delay of $+5$ ms (Across 3 in Fig. 5.2a). The qualitative discrepancy between CCA and DLAG could be due to two possible sources: (1) Given the same data, CCA has less statistical power than DLAG, and (2) The mathematical definition of CCA limits its ability to disentangle concurrent signals, irrespective of the amount of available data (as illustrated in Fig. 4.9).

5.5.3 Demonstrating the empirical benefit of time delays

To demonstrate the benefit of including time delays in the statistical model, we re-applied DLAG to the V1-V2 datasets presented in Fig. 5.3, but forced all time delay parameters to be zero throughout model selection and fitting. We abbreviate these constrained models as 'DLAG-0' from here on and in Fig. 5.9. For each of the 40 V1-V2 datasets, we identified the number of within- and across-area latent variables for DLAG-0 models using the same two-stage model selection procedure as for the DLAG models (see Section 3.6). Hence estimates for DLAG-0 and DLAG dimensionalities were based on the same first-stage factor analysis (FA) estimates of dimensionality.

DLAG-0 and DLAG estimates of across-area dimensionality were highly correlated (Fig. 5.9a, top; Pearson correlation coefficient, $r = 0.81$), and not significantly different across datasets (Fig. 5.9a, bottom; median difference across datasets: 0; one-sided paired sign test: $p = 0.0946$). Whether or not the ability to fit time delays leads to higher or lower estimates of across-area dimensionality depends on the idiosyncrasies of the neural activity being analyzed. Greater model flexibility provided by time delays could lead to fewer identified dimensions[54]. However, the ability to capture time-delayed interactions could also lead to the discovery of additional dimensions that contain significant (time-lagged) cross-area correlations—correlations that would have gone otherwise undetected by a method that could not account for time delays.

We then compared the optimal DLAG-0 model to the optimal DLAG model on each dataset (each selected through cross-validation) via two performance metrics: cross-validated data log-likelihood (LL;

Fig. 5.9b) and cross-validated leave-group-out R^2 (Fig. 5.9c; see Section 5.4.3). DLAG significantly outperformed DLAG-0 across datasets (Fig. 5.9b,c; one-sided paired sign test, $p < 0.001$). DLAG-0 did outperform DLAG on some datasets (2 of 40 datasets according to LL; 7 of 40 datasets according to leave-group-out R^2), not inconsistent with the results presented in Fig. 5.3c, in which many "ambiguous" time delays were identified, whose magnitudes did not significantly deviate from zero (see also Section 5.4.2). Preferably, one would assess the significance of time delay estimates on a case-by-case basis, as we have done throughout this work.

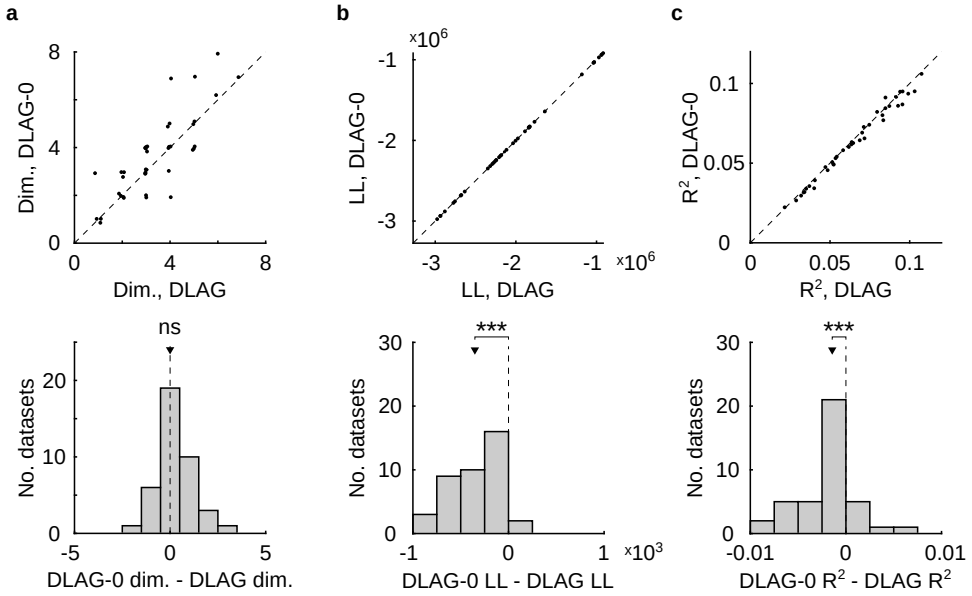

Figure 5.9: V1-V2 interactions are better described by DLAG models with time delays than without time delays. (**a**) Comparison of DLAG-0 and DLAG across-area dimensionality estimates. Top: Estimated DLAG-0 across-area dimensionality ('dim.') versus estimated DLAG across-area dimensionality. Each data point represents one V1-V2 dataset. Data points are integer-valued, but randomly jittered to show points that overlap. DLAG-0 and DLAG estimates of across-area dimensionality are highly correlated (Pearson correlation coefficient, $r = 0.81$). Bottom: Distribution of the differences between DLAG-0 and DLAG across-area dimensionality estimates on each dataset. 'ns': across-area dimensionality estimates are not significantly different across datasets (one-sided paired sign test: $p = 0.0946$; black triangle indicates the median difference across datasets: 0). (**b**)–(**c**) DLAG outperforms DLAG-0 according to multiple metrics (panel (b): LL; panel (c): leave-group-out R^2). Top panels: DLAG-0 performance versus DLAG performance. Each data point represents one V1-V2 dataset. Bottom panels: Distribution of differences between DLAG-0 and DLAG performance on each dataset. DLAG significantly outperforms DLAG-0 across datasets (black triangles indicate the median difference across datasets; '***': one-sided paired sign test; $p < 0.001$).

Chapter 6

Extending DLAG to multiple (more than two) populations

6.1 Motivation

In Chapters 2–5, we addressed the challenge of disentangling concurrent signaling between two neuronal populations. Of course, cortical circuits involve feedforward, feedback, and horizontal connections between many populations that span distinct areas and layers. As recording techniques continue to scale to allow us to record from many neurons across these populations, the need for new conceptual and statistical frameworks grows as well.

Consider the following motivating example. Suppose we wish to study the interactions of three recorded populations, A, B, and C. We might then consider applying a two-area method such as CCA or DLAG to each pair of populations. However, we would encounter the following interpretational ambiguity. Suppose that populations A and B exhibit shared activity fluctuations, and populations A and C also exhibit shared fluctuations. Do populations A, B, and C all co-fluctuate together? Or do A and B co-fluctuate in a way that is uncorrelated with the way in which A and C co-fluctuate? Only by analyzing all populations together can we differentiate these possibilities.

We require a dimensionality reduction method, then, that looks across all populations and determines from the neural activity (1) the number of latent variables needed to describe interactions between populations, and (2) for each latent, which subset of populations is involved. It must do so, furthermore, in a manner that tractably scales with the number of populations. In this chapter, we will first introduce the static dimensionality reduction method group factor analysis (GFA)[72], which solves this problem via automatic relevance determination (ARD). Then, we will build upon this approach to extend the DLAG

framework to include multiple (more than two) neuronal populations (termed multi-population DLAG, or mDLAG). mDLAG inherits the desirable properties of both GFA and DLAG, capable of not only determining the subset of populations involved in an interaction, but also characterizing the flow of signals among those populations and how those signals evolve over time within and across trials.

6.2 Mathematical notation

Introducing the mDLAG model requires different notation from that used in previous chapters. Here we redefine key notation that will be used throughout Chapter 6.

To disambiguate each variable or parameter in the mDLAG model, we need to keep track of up to four labels that indicate their associated (1) trial; (2) neuron or latent state index; (3) time point; or (4) subpopulation (for example, brain area). We indicate the first three labels via subscripts. Trials are indexed by $n = 1, \ldots, N$; neurons are indexed by $i = 1, \ldots, q$; latent states are indexed by $j = 1, \ldots, p$; and time is indexed by $t = 1, \ldots, T$. Where relevant, we indicate the population to which a variable or parameter pertains via a superscript, where populations are indexed by $m = 1, \ldots, M$. For example, we define the observed activity of neuron i (out of q_m) in population m at time t on trial n as $y^m_{n,i,t} \in \mathbb{R}$. To indicate a collection of all variables along a particular index, we replace that index with a colon. Hence we represent the simultaneous activity of q_m neurons observed in population m at time t on trial n as the vector $\mathbf{y}^m_{n,:,t} \in \mathbb{R}^{q_m}$. For concision, where a particular index is either not applicable or not immediately relevant, we omit it. The identities of the remaining indices should be clear from context. For example, we might rewrite $\mathbf{y}^m_{n,:,t}$ as $\mathbf{y}^m_{n,t}$.

It is conceptually helpful to understand the notation for observed variables ($\mathbf{y}$) and latent states ($\mathbf{x}$, see below) as taking cross-sections of three-dimensional arrays. For example, observed activity in population m on trial n can be grouped into the matrix (two-dimensional array) $Y^m_n = [\mathbf{y}^m_{n,1} \cdots \mathbf{y}^m_{n,T}] \in \mathbb{R}^{q_m \times T}$. Hence each $\mathbf{y}^m_{n,t}$ is a column of Y^m_n. Then we can form the three-dimensional array Y^m by concatenating the matrices $Y^m_1, \ldots, Y^m_N$ across trials along a third dimension.

We will explicitly define all other variables and parameters as they appear, but for reference, we list common variables and parameters below:

Data characteristics

- N – total number of trials
- T – number of time points

Observed neural activity

- q_m – number of neurons observed in population m
- Y^m_n – $q_m \times T$ matrix of observed activity in population m on trial n
- $\mathbf{y}^m_{n,t}$ – $q_m \times 1$ vector of observed activity in population m at time t on trial n; the t^{th} column of Y^m_n

Latent state variables

- p – number of latent states (same for all populations)
- X_n^m – $p \times T$ matrix of latent states in population m on trial n
- $\mathbf{x}_{n,:,t}^m$ – $p \times 1$ vector of latent states in population m at time t on trial n; the t^{th} column of X_n^m
- $\mathbf{x}_{n,j,:}^m$ – $T \times 1$ vector of values of latent state j in population m over time on trial n; the j^{th} row of X_n^m

Probabilistic model parameters

- C^m – $q_m \times p$ loading matrix for population m
- α_j^m – automatic relevance determination (ARD) parameter for population m and latent state j
- $\mathbf{d}^m$ – $q_m \times 1$ mean parameter for population m
- $\boldsymbol{\phi}^m$ – $q_m \times 1$ observation noise precision parameter for population m

Deterministic model parameters

- $D_{m,j}$ – time delay parameter between population m and latent state j
- τ_j – Gaussian process timescale for latent state j
- σ_j – Gaussian process noise parameter for latent state j

Gaussian process covariances

- $K_{m_1,m_2,j}$ – $T \times T$ covariance matrix for latent state j, between populations m_1 and m_2
- $k_{m_1,m_2,j}$ – covariance function for latent state j, between populations m_1 and m_2

6.3 Background: Group factor analysis (GFA)

Here we introduce a slightly modified version of the static dimensionality reduction method group factor analysis (GFA)[72]. For population m on trial n, define a linear relationship between observed neural activity, $\mathbf{y}_n^m \in \mathbb{R}^{q_m}$, and latent state variables, $\mathbf{x}_n \in \mathbb{R}^p$:

$$\mathbf{y}_n^m = C^m \mathbf{x}_n + \mathbf{d}^m + \boldsymbol{\varepsilon}^m \tag{6.1}$$

$$\boldsymbol{\varepsilon}^m \sim \mathcal{N}(\mathbf{0}, (\Phi^m)^{-1}) \tag{6.2}$$

where $C^m \in \mathbb{R}^{q_m \times p}$, $\mathbf{d}^m \in \mathbb{R}^{q_m}$, and $\Phi^m \in \mathbb{S}^{q_m \times q_m}$ ($\mathbb{S}^{q_m \times q_m}$ is the set of $q_m \times q_m$ symmetric matrices) are probabilistic model parameters with prior distributions, defined below.

The parameter $\mathbf{d}^m$ can be thought of as the mean firing rate of each neuron in population m. Each $\mathbf{d}^m$ is defined to have a Gaussian prior:

$$P(\mathbf{d}^m) = \mathcal{N}(\mathbf{d}^m \mid \mathbf{0}, \beta^{-1} I_{q_m}) \tag{6.3}$$

where $\beta \in \mathbb{R}_{>0}$ is a hyperparameter and I_{q_m} is the $q_m \times q_m$ identity matrix. $\boldsymbol{\varepsilon}^m$ is a zero-mean Gaussian random variable, where—here—we will constrain the precision matrix $\Phi^m = \text{diag}(\phi_1^m, \ldots, \phi_{q_m}^m)$ to be diagonal to capture variance that is independent to each neuron (in [72], the precision matrix is defined as τI_{q_m}, so that the noise variance is the same for all neurons, $\tau^{-1} \in \mathbb{R}_{>0}$). This constraint encourages the

latent state variables to explain as much of the shared variance among neurons as possible. We set the conjugate Gamma prior over each ϕ_i^m:

$$P(\phi_i^m) = \Gamma(\phi_i^m \mid a_\phi, b_\phi) \tag{6.4}$$

where $a_\phi, b_\phi \in \mathbb{R}_{>0}$ are hyperparameters.

The loading matrix C^m linearly combines latent state variables and maps them to observed neural activity. The automatic selection of the number of latent states, and of the number of populations a particular latent state involves, is accomplished through an automatic relevance determination (ARD) framework (see also [73]). Specifically, each column of C^m is defined by the following prior:

$$P(\mathbf{c}_j^m \mid \alpha_j^m) = \mathcal{N}(\mathbf{c}_j^m \mid \mathbf{0}, (\alpha_j^m)^{-1} I_{q_m}) \tag{6.5}$$

$$P(\alpha_j^m) = \Gamma(\alpha_j^m \mid a_\alpha, b_\alpha) \tag{6.6}$$

where $\mathbf{c}_j^m \in \mathbb{R}^{q_m}$ is the j^{th} column of C^m, $\alpha_j^m \in \mathbb{R}_{>0}$ is the ARD parameter for latent state j and population m, and $a_\alpha, b_\alpha \in \mathbb{R}_{>0}$ are hyperparameters. As α_j^m becomes very large, the magnitude of $\mathbf{c}_j^m$ becomes increasingly concentrated around 0, and hence the j^{th} latent state $x_{n,j}$ will have a vanishing influence on population m. The ARD prior thus encourages population-wise sparsity for each latent state variable.

Finally, latent state variables $\mathbf{x}_n$ are defined by a standard Normal prior:

$$P(\mathbf{x}_n) = \mathcal{N}(\mathbf{x}_n \mid \mathbf{0}, I_p) \tag{6.7}$$

where I_p is the $p \times p$ identity matrix.

The posterior distributions over the latent state variables and model parameters are estimated from the neural activity. However, as a departure from the other probabilistic methods we have discussed thus far (for example FA, pCCA, DLAG), the addition of prior distributions over the model parameters (equations (6.3)–(6.7)) precludes the use of an exact EM algorithm. GFA models are instead fit using approximate inference: posterior estimates maximize a variational lower bound on the data likelihood, and are constrained to follow a particular factorized form (see Section 6.5).

To our knowledge, GFA has not previously been applied to electrophysiological recordings. We have therefore validated GFA's ability to recover multi-population interactions in simulated and real spiking neural activity[74]. GFA not only performed well on realistic-scale simulated neural activity, but also reproduced key results from a prior study of areas V1 and V2: that the two areas interact via a communication subspace[37]. We then used GFA to study interactions across select laminar compartments of macaque visual areas V1, V2, and V3d, recorded simultaneously with multiple Neuropixels probes[75]. GFA uncovered intriguing receptive-field dependent signatures of selective communication across V1, V2, V3d, and their layers. These initial results establish a foundation for the development of mDLAG.

6.4 mDLAG model definition

Observation model and automatic relevance determination For population m at time t on trial n, we define a linear relationship between observed activity, $\mathbf{y}_{n,t}^m$, and the latent state variables, $\mathbf{x}_{n,t}^m$ (Fig. 6.1a):

$$\mathbf{y}_{n,t}^m = C^m \mathbf{x}_{n,t}^m + \mathbf{d}^m + \boldsymbol{\varepsilon}^m \tag{6.8}$$

$$\boldsymbol{\varepsilon}^m \sim \mathcal{N}(\mathbf{0}, (\Phi^m)^{-1}) \tag{6.9}$$

where $C^m \in \mathbb{R}^{q_m \times p}$, $\mathbf{d}^m \in \mathbb{R}^{q_m}$, and $\Phi^m \in \mathbb{S}^{q_m \times q_m}$ ($\mathbb{S}^{q_m \times q_m}$ is the set of $q_m \times q_m$ symmetric matrices) are probabilistic model parameters with prior distributions, defined below.

The parameter $\mathbf{d}^m$ can be thought of as the mean firing rate of each neuron. We set a Gaussian prior over $\mathbf{d}^m$:

$$P(\mathbf{d}^m) = \mathcal{N}(\mathbf{d}^m \mid \mathbf{0}, \beta^{-1} I_{q_m}) \tag{6.10}$$

where $\beta \in \mathbb{R}_{>0}$ is a hyperparameter and I_{q_m} is the $q_m \times q_m$ identity matrix. $\boldsymbol{\varepsilon}^m$ is a zero-mean Gaussian random variable, where we constrain the precision matrix $\Phi^m = \mathrm{diag}(\phi_1^m, \ldots, \phi_{q_m}^m)$ to be diagonal to capture variance that is independent to each neuron. This constraint encourages the latent variables to explain as much of the shared variance among neurons as possible. We set the conjugate Gamma prior over each ϕ_i^m:

$$P(\phi_i^m) = \Gamma(\phi_i^m \mid a_\phi, b_\phi) \tag{6.11}$$

where $a_\phi, b_\phi \in \mathbb{R}_{>0}$ are hyperparameters.

As we will describe, at time point t, latent state variables $\mathbf{x}_{n,t}^m$, $m = 1, \ldots, M$ are coupled across populations, and thus each population has the same number of latent states, p. Because we seek a low-dimensional description of neural activity, the number of latent states is less than the number of neurons, i.e., $p < q$, where $q = \sum_m q_m$.

The loading matrix C^m linearly combines latent states and maps them to observed neural activity. The automatic selection of the number of latent states, and of the number of areas a particular latent state involves, is accomplished through ARD. Specifically, we define the following prior over the columns of each C^m:

$$P(\mathbf{c}_j^m \mid \alpha_j^m) = \mathcal{N}(\mathbf{c}_j^m \mid \mathbf{0}, (\alpha_j^m)^{-1} I_{q_m}) \tag{6.12}$$

$$P(\alpha_j^m) = \Gamma(\alpha_j^m \mid a_\alpha, b_\alpha) \tag{6.13}$$

where $\mathbf{c}_j^m \in \mathbb{R}^{q_m}$ is the j^{th} column of C^m, $\alpha_j^m \in \mathbb{R}_{>0}$ is the ARD parameter for latent state j and population m, and $a_\alpha, b_\alpha \in \mathbb{R}_{>0}$ are hyperparameters. As α_j^m becomes very large, the magnitude of $\mathbf{c}_j^m$ becomes

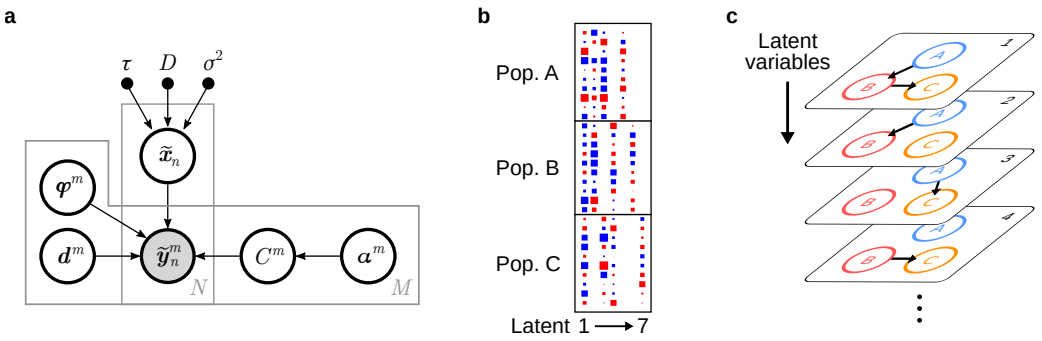

Figure 6.1: DLAG for multiple neuronal populations (mDLAG). (**a**) mDLAG directed graphical model representation. Filled circles represent observed variables. Unfilled circles represent latent variables. Small black circles represent deterministic model paramters. Arrows indicate conditional dependence relationships between variables. (**b**) Example mDLAG loading matrix (here the loading matrices for individual populations, C^1, C^2, and C^3 have been concatenated vertically). Each element of the matrix is represented by a square: magnitude is represented by the square's area, and sign is represented by the square's color (red: positive; blue: negative). Each column represents the population activity pattern represented by a latent variable. Note the population-wise sparsity pattern of each latent variable. (**c**) Each latent variable (depicted by a panel) describes which subset of populations is involved in an interaction, and the direction of signal flow among the populations in that subset (indicated by the presence and direction of black arrows). Multiple latent variables can be employed to describe concurrent signaling across various subnetworks.

increasingly concentrated around 0, and hence the j^{th} latent state $x^m_{n,j,t}$ will have a vanishing influence on population m.

The ARD prior encourages population-wise sparsity for each latent state variable (Fig. 6.1b). Additionally, as we will discuss below, since the j^{th} latent state ($\mathbf{x}^m_{n,j,:}$) is associated with a direction of population signal flow, so too is the corresponding column in C^m. The sparsity structure of C^m and the latent states $\mathbf{x}^m_{n,j,:}$ therefore combine to describe which subset of populations is involved in an interaction, and the direction of signal flow among the populations in that subset (Fig. 6.1c). Multiple latent variables can be employed to describe concurrent signaling across various subnetworks.

The parameter C^m also has an intuitive geometric interpretation. Each element of $\mathbf{y}^m_{n,t}$, the activity of each neuron in population m on trial n, can be represented as an axis in a high-dimensional population activity space. Then the columns of C^m define a subspace in this population activity space, where each dimension corresponds to a distinct latent state. This subspace represents patterns of population activity that is correlated across populations, and the subspace can be partitioned further based on the nominal directionality of activity patterns. Finally, note that the columns of C^m (and the subspaces they define) are linearly independent; but they are not, in general, orthogonal. The ordering of these columns, and of the corresponding latent state variables, is arbitrary.

State model We seek to extract smooth, single-trial latent time courses, where the degree of smoothing is determined by the neural activity (see Section 6.5). The time course of each latent state is described by a Gaussian process (GP)[55]. We define a multi-output GP for each latent state variable $j = 1, \ldots, p$ as follows (Fig. 6.2a):

$$\begin{bmatrix} \mathbf{x}^1_{n,j,:} \\ \vdots \\ \mathbf{x}^M_{n,j,:} \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} K_{1,1,j} & \cdots & K_{1,M,j} \\ \vdots & \ddots & \vdots \\ K_{M,1,j} & \cdots & K_{M,M,j} \end{bmatrix} \right) \tag{6.14}$$

The diagonal blocks $K_{1,1,j} = \cdots = K_{M,M,j} \in \mathbb{S}^{T \times T}$ describe the autocovariance of each latent state, and each T-by-T off-diagonal block describes the cross-covariance that couples two populations.

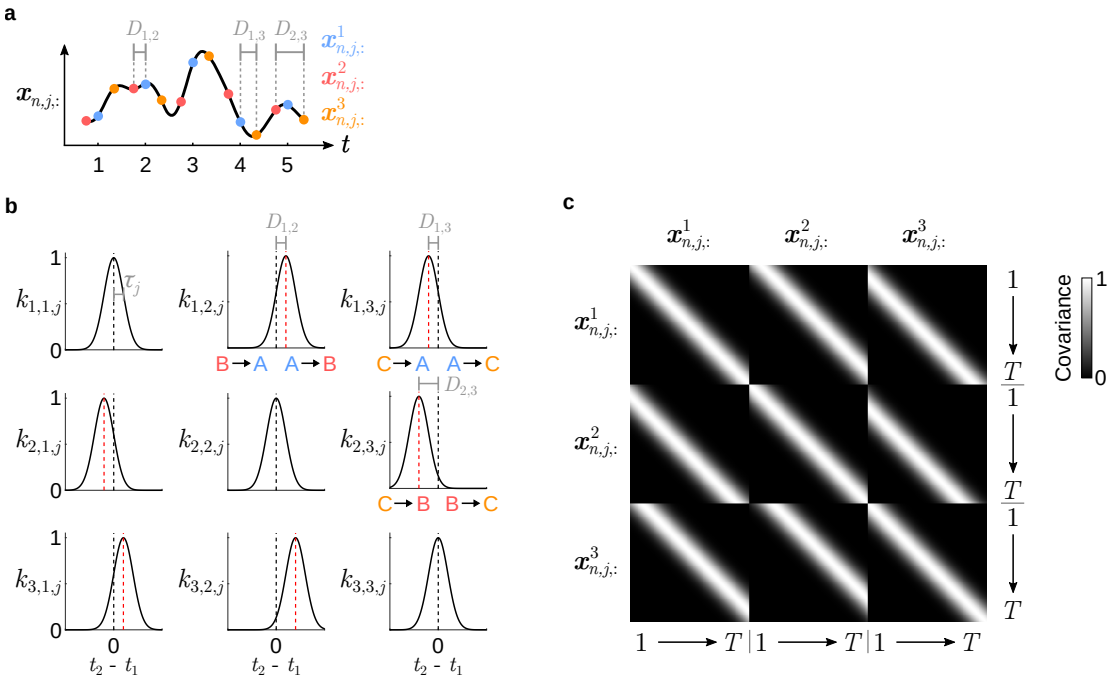

Figure 6.2: The use of Gaussian processes in the mDLAG state model. (**a**) Latent time courses on the n^{th} trial can be described as a finite number of samples drawn from a common GP ($\mathbf{x}_{n,j,:}$). The sampling grid of populations A (blue), B (red), and C (gold) are shifted by time delays ($D_{1,2}$, $D_{1,3}$, $D_{2,3}$) relative to each other. (**b**) The temporal structure of the common GP is governed by a squared exponential covariance function. The width of the auto- and cross-covariances ($k_{m_1,m_2,j}$) is controlled by a timescale parameter (τ_j). The center of the cross-covariance between populations m_1 and m_2 is controlled by the delay parameter D_{m_1,m_2}. (**c**) An example GP covariance matrix (K_j). The banded structure emerges from the choice of squared exponential function and stationarity of the GP covariance. Note the non-zero cross-covariance terms in the off-diagonal blocks of K_j: the banded structure is shifted from the diagonal of each off-diagonal block by the delay parameter $D_{m_1,m_2,j}$.

To express the auto- and cross-covariance functions, we introduce additional notation. Specifically, we indicate populations with two subscripts, $m_1 = 1, \ldots, M$ and $m_2 = 1, \ldots, M$. Then, we define $K_{m_1,m_2,j} \in \mathbb{R}^{T \times T}$ to be either the auto- or cross-covariance matrix between latent state $\mathbf{x}^{m_1}_{n,j,:}$ in population m_1 and

latent state $\mathbf{x}_{n,j,:}^{m_2}$ in population m_2 on trial n. We choose to use the squared exponential function for GP covariances (Fig. 6.2b). Therefore, element (t_1, t_2) of each $K_{m_1,m_2,j}$ (Fig. 6.2c) can be computed as follows[54,76]:

$$k_{m_1,m_2,j}(t_1,t_2) = \left(1-(\sigma_j)^2\right)\exp\left(-\frac{(\Delta t)^2}{2(\tau_j)^2}\right) + (\sigma_j)^2 \cdot \delta_{\Delta t} \tag{6.15}$$

$$\Delta t = \left(t_2 - D_{m_2,j}\right) - \left(t_1 - D_{m_1,j}\right) \tag{6.16}$$

where the characteristic timescale, $\tau_j \in \mathbb{R}_{>0}$, and the GP noise variance, $(\sigma_j)^2 \in (0,1)$, are deterministic model parameters to be estimated from neural activity. $\delta_{\Delta t}$ is the kronecker delta, which is 1 for $\Delta t = 0$ and 0 otherwise.

We also introduce two new parameters: the time delay to population m_1, $D_{m_1,j} \in \mathbb{R}$, and the time delay to population m_2, $D_{m_2,j} \in \mathbb{R}$. Notice that, when computing the auto-covariance for population m (i.e., $m_1 = m_2 = m$), the time delay parameters $D_{m_1,j}$ and $D_{m_2,j}$ are equal, and so Δt (equation (6.16)) reduces simply to the time difference $(t_2 - t_1)$. Time delays are therefore only relevant when computing the cross-covariance between distinct populations m_1 and m_2. The time delay to population m_1, $D_{m_1,j}$, and the time delay to population m_2, $D_{m_2,j}$, by themselves have no physically meaningful interpretation. Their difference $D_{m_2,j} - D_{m_1,j}$, however, represents a well-defined, continuous-valued time delay from population m_1 to population m_2. The sign of the relative time delay indicates the directionality of the lead-lag relationship between populations captured by latent variable j (positive: population m_1 leads population m_2; negative: population m_2 leads population m_1), which we interpret as a description of signal flow.

Both the characteristic timescales τ_j and time delays $D_{m,j}$ are estimated from the neural activity, together with the other mDLAG parameters (see Section 6.5). More specifically, to ensure identifiability of time delay parameters, we designate population $m = 1$ as the reference area, and fix the delays for population 1 at 0, that is, $D_{1,j} = 0$ for all latent state variables $j = 1, \ldots, p$. Note that time delays need not be an integer multiples of the sampling period or spike count bin width of the neural activity. We follow the same conventions as in [47, 76], and fix $(\sigma_j)^2$ to a small value (10^{-3}). Furthermore, the GP is normalized so that $k_{m_1,m_2,j}(t_1,t_2) = 1$ if $\Delta t = 0$, thereby removing model redundancy in the scaling of X^m and C^m.

mDLAG special cases Finally, we consider some special cases of the mDLAG model that illustrate its relationship to other dimensionality reduction methods. First, in the case of two populations ($M = 2$), mDLAG is equivalent to a Bayesian DLAG formulation. In the case of one population ($M = 1$), and when all time delays are fixed to zero ($D_{m,j} = 0$), mDLAG becomes equivalent to a Bayesian Gaussian process

factor analysis (GPFA) formulation[77]. By removing temporal smoothing (i.e., in the limit as all GP noise parameters σ_j approach 1) mDLAG becomes equivalent to GFA.

6.5 Posterior inference and fitting the mDLAG model

6.5.1 Variational inference

Let Y and X be collections of all observed neural activity and latent state variables, respectively, across all time points and trials. Similarly, let $\mathbf{d}$, $\boldsymbol{\phi}$, C, $\mathcal{A}$, and D be collections of the mean parameters, noise precisions, loading matrices, ARD parameters, and time delays, respectively. From the neural activity, we seek to estimate posterior distributions over the probabilistic variables and parameters

$$\theta = \{X,\ \mathbf{d},\ \boldsymbol{\phi},\ C,\ \mathcal{A}\} \tag{6.17}$$

and point estimates of the deterministic GP parameters $\Omega = \left\{D,\ \{\tau_j\}_{j=1}^{p}\right\}$.

In the case of DLAG, the linear-Gaussian structure of the model enabled an exact EM algorithm. With the introduction of prior distributions over model parameters, mDLAG loses this property. The complete likelihood of the mDLAG model,

$$\begin{aligned} P(Y,\theta|\Omega) &= P(\mathbf{d})P(\boldsymbol{\phi})P(C|\mathcal{A})P(\mathcal{A})P(Y|X,C,\mathbf{d},\boldsymbol{\phi})P(X|\Omega) \\ &= \prod_{m=1}^{M}\left[P(\mathbf{d}^m)\left[\prod_{i=1}^{q_m}P(\phi_i^m)\right]\left[\prod_{j=1}^{p}P(\mathbf{c}_j^m \mid \alpha_j^m)P(\alpha_j^m)\right]\left[\prod_{n=1}^{N}\prod_{t=1}^{T}P(\mathbf{y}_{n,t}^m|\mathbf{x}_{n,t}^m,C^m,\mathbf{d}^m,\boldsymbol{\phi}^m)\right]\right] \\ &\cdot\left[\prod_{n=1}^{N}\prod_{j=1}^{p}P(\mathbf{x}_{n,j,:}|\{D_{m,j}\}_{m=1}^{M},\tau_j)\right] \end{aligned} \tag{6.18}$$

is no longer Gaussian. Then a hypothetical EM E-step (evaluation of the posterior distribution $P(\theta|Y,\Omega)$) becomes prohibitive, as it relies on the analytically intractable marginalization of equation (6.18) with respect to θ.

We therefore employ instead a variational inference scheme[72,73], in which we maximize the evidence lower bound (ELBO), $L(Q,\Omega)$, where

$$\log P(Y) \geq L(Q,\Omega) = \mathbb{E}_Q[\log P(Y,\theta|\Omega)] - \mathbb{E}_Q[\log Q(\theta)] \tag{6.19}$$

with respect to the approximate posterior distribution $Q(\theta)$ and the deterministic parameters Ω. We constrain $Q(\theta)$ so that it factorizes over the elements of θ:

$$Q(\theta) = Q_x(X)Q_d(\mathbf{d})Q_\phi(\boldsymbol{\phi})Q_c(C)Q_\mathcal{A}(\mathcal{A}) \tag{6.20}$$

This factorization enables closed-form updates during optimization (see below). The ELBO can then be iteratively maximized via coordinate ascent of the factors of $Q(\theta)$ and the deterministic parameters Ω:

each factor or deterministic parameter is updated in turn while the remaining factors or parameters are held fixed. These updates are repeated until the ELBO improves from one iteration to the next by less than a present tolerance (here we used 10^{-8}).

Posterior distribution updates

Updates of the i^{th} factor of Q, Q_i^*, are given by[73]

$$\log Q_i^*(\theta_i) = \langle \log P(Y, \theta) \rangle_{k \neq i} + \text{const.} \tag{6.21}$$

Here we introduce the notation $\langle \cdot \rangle$ to indicate the expectation with respect to the approximate posterior distribution, $\mathbb{E}_Q[\cdot]$, and $\langle \log P(Y, \theta) \rangle_{k \neq i}$ specifically indicates the expecation of the complete log likelihood with respect to all but the i^{th} factor of Q. We impose no further constraints on Q or its factors. However, because of the choice of Gaussian and conjugate Gamma priors in Section 6.4, evaluation of equation (6.21) leads to factors with the same functional form as their corresponding priors:

$$Q_x(X) = \prod_{n=1}^{N} \mathcal{N}(\bar{\mathbf{x}}_n \mid \bar{\boldsymbol{\mu}}_{x_n}, \bar{\Sigma}_x) \tag{6.22}$$

$$Q_d(\mathbf{d}) = \prod_{m=1}^{M} \mathcal{N}(\mathbf{d}^m \mid \boldsymbol{\mu}_d^m, \Sigma_d^m) \tag{6.23}$$

$$Q_\phi(\boldsymbol{\phi}) = \prod_{m=1}^{M} \prod_{i=1}^{q_m} \Gamma(\phi_i^m \mid \widetilde{a}_\phi, \widetilde{b}_{\phi,i}^m) \tag{6.24}$$

$$Q_c(C) = \prod_{m=1}^{M} \prod_{i=1}^{q_m} \mathcal{N}(\widetilde{\mathbf{c}}_i^m \mid \widetilde{\boldsymbol{\mu}}_{c_i}^m, \Sigma_{c_i}^m) \tag{6.25}$$

$$Q_{\mathcal{A}}(\mathcal{A}) = \prod_{m=1}^{M} \prod_{j=1}^{p} \Gamma(\alpha_j^m \mid \widetilde{a}_\alpha^m, \widetilde{b}_{\alpha,j}^m) \tag{6.26}$$

Any additional factorization in equations (6.22)–(6.26) also emerge naturally—we impose only the factorization in equation (6.20).

To express the updates for $Q_x(X)$, let us first define several variables. Construct $\mathbf{y}_{n,t} = [\mathbf{y}_{n,t}^{1\top} \cdots \mathbf{y}_{n,t}^{M\top}]^\top \in \mathbb{R}^q$ by vertically concatenating the neural activity of populations $m = 1, \ldots, M$ at time t on trial n. Then construct $\bar{\mathbf{y}}_n = [\mathbf{y}_{n,1}^\top \cdots \mathbf{y}_{n,T}^\top]^\top \in \mathbb{R}^{qT}$ by vertically concatenating the neural activity $\mathbf{y}_{n,t}$ across all time points $t = 1, \ldots, T$. For latent state variables, define $\mathbf{x}_{n,t} = [\mathbf{x}_{n,:,t}^{1\top} \cdots \mathbf{x}_{n,:,t}^{M\top}]^\top \in \mathbb{R}^{Mp}$ by vertically concatenating the p latent states of each population at time t on trial n. Then we vertically concatenate the latent states $\mathbf{x}_{n,t}$ across all time points $t = 1, \ldots, T$ to give $\bar{\mathbf{x}}_n = [\mathbf{x}_{n,1}^\top \cdots \mathbf{x}_{n,T}^\top]^\top \in \mathbb{R}^{MpT}$. Finally, we collect the parameters C^m, Φ^m, and $\mathbf{d}^m$ across populations $m = 1, \ldots, M$ by defining $C = \text{diag}(C^1, \ldots, C^M) \in \mathbb{R}^{q \times Mp}$, $\Phi = \text{diag}(\Phi^1, \ldots, \Phi^m) \in \mathbb{S}^{q \times q}$, and $\mathbf{d} = [\mathbf{d}^{1\top} \cdots \mathbf{d}^{M\top}]^\top \in \mathbb{R}^q$.

Posterior estimates of the latent state variables X are independent across trials. We can thus update $Q_x(X)$ by evaluating the posterior covariance, $\bar{\Sigma}_x \in \mathbb{S}^{MpT \times MpT}$, and mean, $\bar{\boldsymbol{\mu}}_{x_n} \in \mathbb{R}^{MpT}$, of $\bar{\mathbf{x}}_n$ for each

trial n:

$$\bar{\Sigma}_x = (\bar{K}^{-1} + \langle \overline{C^\top \Phi C} \rangle)^{-1} \tag{6.27}$$

$$\bar{\boldsymbol{\mu}}_{x_n} = \bar{\Sigma}_x \langle \bar{C} \rangle^\top \langle \bar{\Phi} \rangle (\bar{\mathbf{y}}_n - \langle \bar{\mathbf{d}} \rangle) \tag{6.28}$$

where $\langle \bar{C} \rangle \in \mathbb{R}^{qT \times MpT}$, $\langle \bar{\Phi} \rangle \in \mathbb{S}^{qT \times qT}$, and $\langle \overline{C^\top \Phi C} \rangle \in \mathbb{R}^{MpT \times MpT}$ are block diagonal matrices comprising T copies of the matrices $\langle C \rangle$, $\langle \Phi \rangle$, and $\langle C^\top \Phi C \rangle$, respectively. $\bar{\mathbf{d}} \in \mathbb{R}^{qT}$ is constructed by vertically concatenating T copies of $\mathbf{d}$. The elements of $\bar{K} \in \mathbb{R}^{MpT \times MpT}$ are computed using equations (6.15) and (6.16).

Posterior estimates of the mean parameters $\mathbf{d}$ are independent across populations (and, in fact, neurons). We can thus update $Q_d(\mathbf{d})$ by evaluating the posterior covariance, $\Sigma_d^m \in \mathbb{S}^{q_m \times q_m}$, and mean, $\boldsymbol{\mu}_d^m \in \mathbb{R}^{q_m}$, of mean parameter $\mathbf{d}^m$ for each population m:

$$\Sigma_d^m = (\beta I_{q_m} + NT \langle \Phi^m \rangle)^{-1} \tag{6.29}$$

$$\boldsymbol{\mu}_d^m = \Sigma_d^m \langle \Phi^m \rangle \sum_{n=1}^{N} \sum_{t=1}^{T} (\mathbf{y}_{n,t}^m - \langle C^m \rangle \langle \mathbf{x}_{n,t}^m \rangle) \tag{6.30}$$

Posterior estimates of precision parameters $\boldsymbol{\phi}$ are independent across populations and neurons. We can thus update $Q_\phi(\boldsymbol{\phi})$ by evaluating the posterior parameters $\widetilde{a}_\phi$ and $\widetilde{b}_{\phi,i}^m$ of parameter ϕ_i^m for each neuron i in population m:

$$\widetilde{a}_\phi = a_\phi + \frac{NT}{2} \tag{6.31}$$

$$\begin{aligned} \widetilde{b}_{\phi,i}^m = b_\phi + \frac{1}{2} \sum_{n=1}^{N} \sum_{t=1}^{T} [& (y_{n,i,t}^m)^2 + \langle (d_i^m)^2 \rangle + \operatorname{tr} \left(\langle \widetilde{\mathbf{c}}_i^m (\widetilde{\mathbf{c}}_i^m)^\top \rangle \langle \mathbf{x}_{n,t}^m (\mathbf{x}_{n,t}^m)^\top \rangle \right) \\ & - 2 \langle \widetilde{\mathbf{c}}_i^m \rangle \langle \mathbf{x}_{n,t}^m \rangle (y_{n,i,t}^m - \langle d_i^m \rangle) - 2 y_{n,i,t}^m \langle d_i^m \rangle] \end{aligned} \tag{6.32}$$

Here $\widetilde{\mathbf{c}}_i^m \in \mathbb{R}^p$ is the i^{th} row of C^m, the loading matrix for population m.

Posterior estimates of loading matrices C are independent across populations and neurons, i.e., across the rows of each C^m. We can thus update $Q_c(C)$ by evaluating the posterior covariance, $\Sigma_{c_i}^m \in \mathbb{S}^{p \times p}$, and mean, $\widetilde{\boldsymbol{\mu}}_{c_i}^m \in \mathbb{R}^p$, of the i^{th} row of C^m:

$$\Sigma_{c_i}^m = (\langle \mathcal{A}^m \rangle + \langle \phi_i^m \rangle \sum_{n=1}^{N} \sum_{t=1}^{T} \langle \mathbf{x}_{n,t}^m (\mathbf{x}_{n,t}^m)^\top \rangle)^{-1} \tag{6.33}$$

$$\widetilde{\boldsymbol{\mu}}_{c_i}^m = \Sigma_{c_i}^m \langle \phi_i^m \rangle \sum_{n=1}^{N} \sum_{t=1}^{T} \langle \mathbf{x}_{n,t}^m \rangle (y_{n,i,t}^m - \langle d_i^m \rangle) \tag{6.34}$$

Here $\mathcal{A}^m = \operatorname{diag}(\alpha_1^m, \ldots, \alpha_p^m)$.

Finally, posterior estimates of ARD parameters $\mathcal{A}$ are independent across populations and latent state variables. We can thus update $Q_\mathcal{A}(\mathcal{A})$ by evaluating the posterior parameters $\widetilde{a}_\alpha^m$ and $\widetilde{b}_{\alpha,i}^m$ of parameter α_j^m

for each population m and latent state variable j:

$$\widetilde{a}_{\alpha}^{m} = a_{\alpha} + \frac{q_m}{2} \tag{6.35}$$

$$\widetilde{b}_{\alpha,j}^{m} = b_{\alpha} + \frac{1}{2}\langle \|\mathbf{c}_j^m\|_2^2 \rangle \tag{6.36}$$

All moments $\langle \cdot \rangle$ can be readily computed from the approximate posterior distributions given in equations (6.22)–(6.26).

Gaussian process parameter updates

There are no closed-form solutions for the Gaussian process parameter updates, but we can compute gradients and perform gradient ascent. Note that, for this work, we choose not to fit the Gaussian process noise variances, but rather, we set them to small values (10^{-3}), as in [47, 76].

To express the timescale and delay parameter gradients, we introduce more compact notation for the variables in equation (6.14). Let $\mathbf{x}_{n,j,:} = [\mathbf{x}_{n,j,:}^{1\top} \cdots \mathbf{x}_{n,j,:}^{M\top}]^\top \in \mathbb{R}^{MT}$ for the j^{th} latent state, and

$$K_j = \begin{bmatrix} K_{1,1,j} & \cdots & K_{1,M,j} \\ \vdots & \ddots & \vdots \\ K_{M,1,j} & \cdots & K_{M,M,j} \end{bmatrix} \in \mathbb{S}^{MT \times MT} \tag{6.37}$$

Rewrite the ELBO to show the terms that depend on K_j:

$$L(Q, \Omega) = \sum_{n=1}^{N} \sum_{j=1}^{p} \left[\frac{1}{2} \log |K_j^{-1}| - \frac{1}{2} \mathrm{tr}(K_j^{-1} \langle \mathbf{x}_{n,j,:} \mathbf{x}_{n,j,:}^\top \rangle) \right] + \text{const.} \tag{6.38}$$

Then, let $L_n = \sum_{j=1}^{p} \left[\frac{1}{2} \log |K_j^{-1}| - \frac{1}{2} \mathrm{tr}(K_j^{-1} \langle \mathbf{x}_{n,j,:} \mathbf{x}_{n,j,:}^\top \rangle) \right]$.

To optimize timescales, we first make the change of variables $\gamma_j = 1/\tau_j^2$. γ_j is simpler to work with. We then optimize with respect to γ_j. The γ_j gradients are given by

$$\frac{\partial L}{\partial \gamma_j} = \sum_{n=1}^{N} \mathrm{tr} \left(\left(\frac{\partial L_n}{\partial K_j} \right)^\top \left(\frac{\partial K_j}{\partial \gamma_j} \right) \right) \tag{6.39}$$

where

$$\frac{\partial L_n}{\partial K_j} = -\frac{1}{2} K_j^{-1} + \frac{1}{2} \left(K_j^{-1} \langle \mathbf{x}_{n,j,:} \mathbf{x}_{n,j,:}^\top \rangle K_j^{-1} \right) \tag{6.40}$$

and each element of $\partial K_j / \partial \gamma_j$ is given by

$$\frac{\partial k_{m_1,m_2,j}(t_1, t_2)}{\partial \gamma_j} = -\frac{1}{2} (\Delta t)^2 \left(1 - \sigma_j^2 \right) \exp \left(-\frac{1}{2} \gamma_j (\Delta t)^2 \right) \tag{6.41}$$

where Δt is defined as in equation (6.16). To optimize γ_j while respecting non-negativity constraints, we perform a change of variables, and then perform unconstrained gradient ascent with respect to $\log \gamma_j$.

Next, delay gradients for population m and latent variable j are given by

$$\frac{\partial L}{\partial D_{m,j}} = \sum_{n=1}^{N} \operatorname{tr}\left(\left(\frac{\partial L_n}{\partial K_j}\right)^{\top}\left(\frac{\partial K_j}{\partial D_{m,j}}\right)\right) \tag{6.42}$$

where $\frac{\partial L_n}{\partial K_j}$ is defined as in equation (6.40), and each element of $\partial K_j / \partial D_{m,j}$ is given by

$$\frac{\partial k_{m_1,m_2,j}(t_1,t_2)}{\partial D_{m,j}} = -\gamma_j(\Delta t)\left(1-\sigma_j^2\right)\exp\left(-\frac{1}{2}\gamma_j(\Delta t)^2\right)\frac{\partial(\Delta t)}{\partial D_{m,j}} \tag{6.43}$$

$$\frac{\partial(\Delta t)}{D_{m,j}} = \begin{cases} 1 & \text{if} \quad m = m_1 \\ -1 & \text{if} \quad m = m_2 \\ 0 & \text{otherwise} \end{cases} \tag{6.44}$$

where Δt, m_1, and m_2 are defined as in equation (6.16). In practice, we fix all delay parameters for population 1 at 0 to ensure identifiability. One might wish to constrain the delays within some physically realistic range, such as the length of an experimental trial, so that $-D_{\max} \leq D_{m,j} \leq D_{\max}$. Toward that end, we make the change of variables $D_{m,j} = D_{\max} \cdot \tanh(\frac{D^*_{m,j}}{2})$ and perform unconstrained gradient ascent with respect to $D^*_{m,j}$. Here we chose $D_{\max}$ to be half the length of a trial.

6.5.2 Evaluation of the lower bound

To evaluate the ELBO, we can rewrite it as follows:

$$L(Q,\Omega) = \mathbb{E}_Q[\log P(Y|\theta,\Omega)] - \mathrm{KL}(Q(\theta)\|P(\theta|\Omega)) \tag{6.45}$$

$\mathrm{KL}(Q(\theta)\|P(\theta|\Omega))$ is the KL-divergence between the approximate posterior distribution $Q(\theta)$ and prior distribution $P(\theta|\Omega)$. Due to the factorized forms of $Q(\theta)$ and $P(\theta|\Omega)$, $L(Q,\Omega)$ becomes

$$\begin{aligned} L(Q,\Omega) = \mathbb{E}_Q[\log P(Y|\theta,\Omega)] &- \mathrm{KL}(Q_x(X)\|P(X|\Omega)) - \mathrm{KL}(Q_c(C)\|P(C|\mathcal{A})) \\ &- \mathrm{KL}(Q_{\mathcal{A}}(\mathcal{A})\|P(\mathcal{A})) - \mathrm{KL}(Q_\phi(\boldsymbol{\phi})\|P(\boldsymbol{\phi})) - \mathrm{KL}(Q_d(\mathbf{d})\|P(\mathbf{d})) \end{aligned} \tag{6.46}$$

This form of the ELBO provides insight into the nature of the optimization procedure for fitting mDLAG models. The first term is the expected log-likelihood (with respect to the approximate posterior $Q(\theta)$) of the observed neural activity, Y, given the latest model parameters, θ and Ω. This term encourages mDLAG models to explain the observed neural activity as well as possible. The KL-divergence terms, on the other hand, penalize deviations of each factor of the fitted posterior from its corresponding prior distribution, and hence act as a form of regularization.

Using the posterior updates in Section 6.5.1 and the prior definitions in Section 6.4, each term of the ELBO can be computed as follows:

$$\mathbb{E}_Q[\log P(Y|\theta,\Omega)] = -\frac{qNT}{2}\log(2\pi) + \frac{NT}{2}\sum_{m=1}^{M}\sum_{i=1}^{q_m}\langle \log \phi_i^m\rangle - \sum_{m=1}^{M}\sum_{i=1}^{q_m}(\widetilde{a}_\phi - \langle\phi_i^m\rangle b_\phi) \tag{6.47}$$

$$-\mathrm{KL}(Q_x(X)\|P(X|\Omega)) = \frac{MpNT}{2} + \frac{1}{2}\sum_{n=1}^{N}\left[\log|\bar{\Sigma}_x| - \sum_{j=1}^{p}\left[\log|K_j| + \mathrm{tr}(K_j^{-1}\langle \mathbf{x}_{n,j,:}\mathbf{x}_{n,j,:}^\top\rangle)\right]\right] \tag{6.48}$$

$$-\mathrm{KL}(Q_c(C)\|P(C|\mathcal{A})) = \sum_{m=1}^{M}\left[\frac{q_m}{2}\sum_{j=1}^{p}\langle\log\alpha_j^m\rangle + \frac{1}{2}\sum_{i=1}^{q_m}[\log|\Sigma_{c_i}^m| + \mathrm{tr}(I_p - \langle\widetilde{\mathbf{c}}_i^m(\widetilde{\mathbf{c}}_i^m)^\top\rangle\langle\mathcal{A}^m\rangle)]\right] \tag{6.49}$$

$$\begin{aligned}-\mathrm{KL}(Q_{\mathcal{A}}(\mathcal{A})\|P(\mathcal{A})) = \sum_{m=1}^{M}\sum_{j=1}^{p}[&-\widetilde{a}_\alpha^m\log\widetilde{b}_{\alpha,j}^m + a_\alpha\log b_\alpha + \log\frac{\Gamma(\widetilde{a}_\alpha^m)}{\Gamma(a_\alpha)} - b_\alpha\langle\alpha_j^m\rangle + \widetilde{a}_\alpha^m \\ &+ (a_\alpha - \widetilde{a}_\alpha^m)(\Psi(\widetilde{a}_\alpha^m) - \log\widetilde{b}_{\alpha,j}^m)]\end{aligned} \tag{6.50}$$

$$\begin{aligned}-\mathrm{KL}(Q_\phi(\boldsymbol{\phi})\|P(\boldsymbol{\phi})) = \sum_{m=1}^{M}\sum_{i=1}^{q_m}[&-\widetilde{a}_\phi\log\widetilde{b}_{\phi,i}^m + a_\phi\log b_\phi + \log\frac{\Gamma(\widetilde{a}_\phi)}{\Gamma(a_\phi)} - b_\phi\langle\phi_i^m\rangle + \widetilde{a}_\phi \\ &+ (a_\phi - \widetilde{a}_\phi)(\Psi(\widetilde{a}_\phi) - \log\widetilde{b}_{\phi,i}^m)]\end{aligned} \tag{6.51}$$

$$-\mathrm{KL}(Q_d(\mathbf{d})\|P(\mathbf{d})) = \frac{q}{2} + \frac{q}{2}\log\beta + \frac{1}{2}\log|\Sigma_d| - \frac{1}{2}\beta\langle\|\mathbf{d}\|_2^2\rangle \tag{6.52}$$

Here, $\Gamma(\cdot)$ is the gamma function, and $\Psi(\cdot)$ is the digamma function. All moments $\langle\cdot\rangle$ can be readily computed from the approximate posterior distributions given in equations (6.22)–(6.26).

6.5.3 Parameter initialization and removal of insignificant latent state variables

To initialize the mDLAG fitting procedure, we first specified an initial number of latent state variables, p. Through automatic relevance determination, mDLAG effectively prunes insignificant latent state variables. We leveraged this feature to improve the computational efficiency (with respect to both speed and memory) of the fitting procedure as follows. Each iteration, we evaluated the sample second moment of the estimated latent state variables, $\frac{1}{N}\sum_n \bar{\boldsymbol{\mu}}_{x_n}^2$. If the sample second moment of a latent state variable was not larger than some threshold, ϵ, for at least one population, then we removed it from the mDLAG model (and its associated parameters in θ and Ω)[72]. Here, we chose $\epsilon = 10^{-7}$. We chose an initial p to be as small as possible (to minimize runtime) yet large enough that at least one of the initial latent state variables would be deemed insignificant, thus ensuring that dimensionalities were not underestimated.

To initialize the rest of the mDLAG fitting procedure, we specified initial values for needed moments of the posterior factors $Q_d(\mathbf{d})$, $Q_\phi(\boldsymbol{\phi})$, $Q_c(C)$, and $Q_{\mathcal{A}}(\mathcal{A})$ (equations (6.23)–(6.26)). $Q_x(X)$ was then the first factor to be updated each iteration of the fitting procedure. We specified noninformative priors by fixing all hyperparameters to a very small value[72], $\beta, a_\phi, b_\phi, a_\alpha, b_\alpha = 10^{-12}$. For $Q_d(\mathbf{d})$, we initialized $\boldsymbol{\mu}_d^m$ at the sample mean of neural activity across all trials and time points. For $Q_\phi(\boldsymbol{\phi})$, we initialized $\langle\phi_i^m\rangle^{-1}$ for

each neuron i in population m to the sample variance of that neuron across all trials and time points. For $Q_c(C)$, we first randomly initialized all first moments $\widetilde{\boldsymbol{\mu}}_{c_i}^m$ with entries drawn from a zero-mean Gaussian distribution with variance chosen to match the scale of the data (the ratio $|\hat{\Sigma}_y|/p$ is reasonable in practice, where $\hat{\Sigma}_y$ is the sample covariance matrix of the neural activity). Then, we initialized the second moments $\langle \widetilde{\mathbf{c}}_i^m (\widetilde{\mathbf{c}}_i^m)^\top \rangle$ to the outer product of first moments $\widetilde{\boldsymbol{\mu}}_{c_i}^m \widetilde{\boldsymbol{\mu}}_{c_i}^{m\top}$. For $Q_{\mathcal{A}}(\mathcal{A})$, we initialized $\langle \alpha_j^m \rangle$ for each latent j in population m to $\frac{1}{2}\langle \|\mathbf{c}_j^m\|_2^2 \rangle / q_m$, which stems from equations (6.35) and (6.36). Finally, we initialized all delays to zero, and all Gaussian process timescales to the same value, equal to twice the sampling period or spike count bin width of the neural activity.

6.6 Validating mDLAG with an example simulated dataset

To validate the mDLAG model and fitting procedure described above, we generated simulated neural activity from the following linear-nonlinear-Poisson (LNP) generative model. On each trial, we generated latent state variable time courses according to the mDLAG state model, equation (6.16). Hence each latent variable time course followed a Gaussian process (GP) with squared exponential (SE) covariance function, and latent state variables included time delays across populations.

For population m with q_m neurons we then generated neural firing rates, $\boldsymbol{\lambda}_{n,t}^m \in \mathbb{R}^{q_m}$, during time bin t of width Δ according to the following model:

$$\boldsymbol{\lambda}_{n,t}^m = \log\left(1 + \exp\left(C^m \mathbf{x}_{n,t}^m + \mathbf{d}^m\right)\right) \cdot \Delta \tag{6.53}$$

The function $\log(1+\exp(\cdot))$ is the softplus function (applied element-wise to its arguments). The parameters $C^m \in \mathbb{R}^{q_m \times p}$ and $\mathbf{d}^m \in \mathbb{R}^{q_m}$ have similar interpretations as in equation (6.8) of the mDLAG observation model. We then generated observed spike counts for neuron i in population m during time bin t of trial n, $y_{n,i,t}^m$, according to a Poisson distribution with rate parameter $\lambda_{n,i,t}^m$ (the i^{th} element of $\boldsymbol{\lambda}_{n,t}^m$):

$$y_{n,j,t}^m \mid \mathbf{x}_{n,t}^m \sim \text{Poisson}(\lambda_{n,i,t}^m) \tag{6.54}$$

We simulated activity for $M = 3$ populations, each with $q^m = 10$ neurons. We chose the mean parameters $\mathbf{d}^m$ and loading matrices C^m so that average neural firing rates (5 spikes per second across neurons in each population) and noise levels (activity due to single-neuron observation noise was 10 times stronger than activity due to latent states) were representative of realistic neural activity. Importantly, we included in our simulated activity all types of inter-population interactions (Fig. 6.3a, left): interactions shared globally, unique to each pair, and local to one population. Finally, we selected Gaussian process timescales and (relative) time delays that ranged between 20 ms to 150 ms and between 15 ms and 40 ms, respectively. With all model parameters specified, we then generated $N = 100$ independent and

identically distributed trials according to the LNP generative model described above. Each trial was 500 ms in length, comprising spike counts in $T = 25$ time bins of width 20 ms.

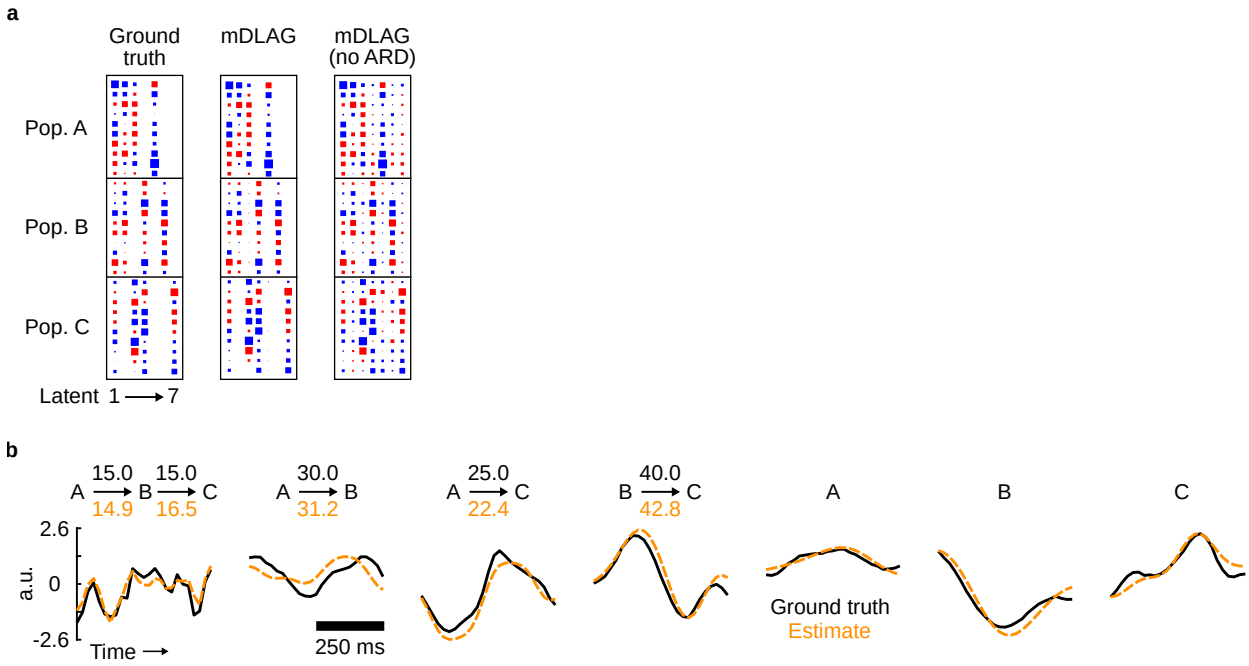

Figure 6.3: Validating mDLAG with an example simulated dataset. (**a**) Loading matrix estimates. Left: Ground truth loading matrix. Center: mDLAG estimate. Right: mDLAG estimate where automatic relevance determination (ARD) was not used, that is, no population-wise sparsity priors were placed on the loading matrix. Same conventions as in Fig. 6.1b. Note that the sign and ordering of each loading matrix column is, in general, arbitrary. We have therefore reordered and flipped the signs of the columns of the estimates to facilitate comparison with the ground truth. (**b**) Single-trial latent-variable time course estimates. Each panel corresponds to the ground truth and estimated time course of a single latent variable. For concision, only the latent variables as they appear in population A are shown ($\mathbf{x}^1_{n,j,:}$). Inset above each latent variable are the involved populations along with the signal flow and magnitude of time delays between populations. Delays are given in ms. Orange: mDLAG estimates; black: ground truth. a.u.: arbitrary units.

We then fit an mDLAG model to the simulated neural activity. To demonstrate the benefit of ARD in the mDLAG model, we also fit a modified mDLAG model that did not use ARD. Specifically, we fit (via an exact EM algorithm) a modified model with state model defined by equations (6.14)–(6.16) and observation model defined by equations (6.8) and (6.9), but with C^m, $\mathbf{d}^m$, and Φ^m defined as deterministic parameters (as in DLAG).

The mDLAG model with ARD recovered the ground truth interactions—particularly the population-wise sparsity structure—with high accuracy (Fig. 6.3a, center). The mDLAG model without ARD, however, produced an estimate of the loading matrix with mostly non-zero elements (Fig. 6.3a, right): had we not known the ground truth in advance, it would be difficult to interpret which population subsets are involved in which interactions. The mDLAG model with ARD also estimated the latent state variable

time courses and time delays with high accuracy (Fig. 6.3b). These results demonstrate that DLAG can be readily extended to incorporate any number of populations, laying a foundation for dissecting the multi-dimensional flow of signals across many interacting populations, such as cortical areas and layers.

Chapter 7

Discussion

In this dissertation, we developed a dimensionality reduction framework, delayed latents across groups (DLAG), which provides a novel description of bidirectional signal flow between populations of neurons. By leveraging the correlated activity across the two populations, DLAG can disentangle concurrent signals relayed in each direction and characterize how those signals evolve over time within and across trials. We demonstrated that DLAG performs well over a wide range of simulated conditions, including those datasets similar in scale to current neurophysiological recordings. Then we used DLAG to study bidirectional interactions between pairs of early and midlevel areas in the macaque visual cortex, in both anesthetized and awake animals. Finally, we developed an extension of DLAG to study interactions across many (more than two) neuronal populations.

Although we applied DLAG to the spiking activity of populations of neurons in distinct brain areas, DLAG is applicable to any high-dimensional time series data, including other neural recording modalities (e.g., calcium imaging, subject to the temporal resolution inherent to the recording technology). It can also be used to study the interactions across populations of neurons in different cortical layers or of different cell types. DLAG can even be used to study the relationship between a neuronal population and a dynamic stimulus or behavioral variables.

Bidirectional interactions between V1 and V2

To our knowledge, DLAG has enabled for the first time the identification of bidirectional, concurrent interactions between brain areas from spiking activity of neuronal populations. DLAG uncovered signatures of inter- and intra-areal interaction that are consistent with previous work, such as the selectivity with which V1 and V2 interact[37], as well as an increase in timescale moving up the cortical hierarchy from V1 to V2[62–64]. In addition, DLAG provided a novel ability to study the bidirectional nature of interactions

between these areas, and characterize these interactions on a moment-to-moment basis. DLAG identified population-level interactions in both directions, whose strengths and associated time delays appear to reflect the cortical layers from which we recorded. One might have expected DLAG to identify at least as many feedforward (V1 to V2) interactions as feedback (V2 to V1): generally, feedback inter-cortical connections equal feedforward connections in number; and, specific to our recording arrangement, feedback connections do not originate in the input layers of V2[59,60]. Surprisingly, DLAG revealed a marked asymmetry, such that a majority of across-area latent variables were associated with a feedback interaction. This apparent disparity presents an opportunity for future study.

Recently, feedforward and feedback signaling was studied in the same V1-V2 recordings analyzed here[43]. Canonical correlation analysis (CCA) was used in a sliding window scheme to identify trial epochs dominated by either feedforward or feedback signaling. V1-V2 (and V1-V4) interactions were found to involve distinct population activity patterns during feedforward- versus feedback-dominated trial epochs. This statistical approach, however, could not be used to study the concurrent nature of feedforward and feedback signaling (see Fig. 4.9 and Fig. 5.8 for further discussion). Here, we provided a complementary view of V1-V2 interactions, using DLAG to identify concurrent, distinct feedforward and feedback activity patterns that characterize the stimulus presentation period as a whole. Future work could characterize how the activity patterns uncovered by DLAG and their associated time delays might change during the course of a trial (see below).

Relation to previous statistical methods

DLAG shares commonalities with several other methods. For instance, static dimensionality reduction methods such as CCA, sparse structured CCA, and their probabilistic variants[57,78] identify across- and/or within-area latent variables, but do not characterize inter-areal interactions over time or the directionality of signal flow (but see [43], discussed above). Multivariate time-series methods such as Granger causal modeling[79–81], Generalized Linear Models[38,82,83], or recurrent neural networks[84] characterize the directionality of signal flow, but not in a low-dimensional manner. Time series methods that provide a low-dimensional description of across-area activity do not provide a low-dimensional description of within-area activity, should low-dimensional within-area activity be of scientific interest[85,86], or they or do not characterize time delays between areas[87]. In contrast with all of these methods, DLAG jointly reduces dimensionality and characterizes the directionality of signal flow by estimating across- and within-area latent variables with time delays and timescales.

DLAG offers unique advantages when characterizing the temporal structure of activity within and across areas. Applied to V1 and V2, DLAG uncovered latent variables with diverse temporal profiles and

timescales. The ability to capture diverse dynamical motifs stems from DLAG's definition via Gaussian processes[47]: beyond temporal smoothness, DLAG makes no additional assumptions about the form of dynamics within or across areas. In contrast, multi-area methods proposed by [56] and [88], for instance, describe interactions between areas according to a parametric dynamical model. Gaussian processes provide DLAG with another advantage: the ability to discover wide-ranging delays with high precision[54]. Existing multi-area methods (nearly all of which, above, are defined in discrete-time) are limited to delays restricted to be integer multiples of the sampling period or spike count bin width of neural activity.

With the conceptual and statistical advantages described above, DLAG is a powerful tool for exploratory data analysis. For example, after performing a new experiment, one can use DLAG to generate data-driven hypotheses about plausible dynamical motifs within and across areas. Then, one can test these hypotheses using a dynamical system-based approach, for example, data-constrained recurrent networks[56,84,88].

Common or unobserved input

One might interpret the population activity patterns represented by DLAG's across-area variables as distinct "channels" with which two areas communicate[43,89]. As with any statistical method, however, interpretation of the features extracted by DLAG is subject to ambiguities, particularly when not all relevant brain areas and neurons are recorded[32,90]. An across-area latent variable, for instance, could reflect an interaction between areas A and B that is direct or indirect, mediated by a third (unobserved) area C. Similarly, a within-area latent variable could reflect activity internal to one area, or it could reflect inputs sent from unrecorded neurons to one area but not the other.

The sign and magnitude of DLAG's time delays can, however, narrow the set of hypotheses consistent with the data. We might reasonably suspect, for example, that short positive (V1 to V2) delays identified by DLAG reflect direct interactions from the output layers of V1 to the input layers of V2 (the layers from which we recorded)[24,26]. Larger negative (V2 to V1) delays might instead indicate indirect interactions, given that the path from the input layers of V2 to the output layers of V1 involves multiple synapses. Some across-area latent variables were associated with delays statistically indistinguishable from zero (i.e., "ambiguous"), and could indicate either tight recurrent interactions or common input from an unobserved source.

A phenomenon widely recognized by cross-correlation studies[21–26] is the presence of correlations across areas due simply to common stimulus drive, rather than an inter-areal interaction. For DLAG, these stimulus driven effects can appear as an across-area variable. The stereotyped periodic signals evident in V1-V2 across-area latent variables (Fig. 5.2a; "Across 3") are a likely example. If desired, one could

control for these effects with straightforward preprocessing steps, such as the subtraction of PSTHs from single-trial responses, thereby emphasizing trial-to-trial fluctuations correlated across areas[37].

Finally, we note that the issue of common input was a key motivation for the development of mDLAG. Neural recordings will also continue to include increasingly many populations. Combining these recordings, methods like mDLAG that look across all populations, and experimental interventions will better resolve ambiguities.

Variability of time delays across trials, time, and neurons

DLAG treats time delays as constant parameters. However, the direction of interaction associated with a dimension of population activity might not be constant across different trial epochs or different experimental (e.g., stimulus) conditions. Thus, we interpret a delay as a summary of this direction of interaction throughout the course of an experiment. Similarly, neurons within the same area can respond to a common input with different latencies (evident in, for example, Fig. 5.1b). An estimated delay hence also represents a summary across neurons[54]. One could fit DLAG to subsets of trials, subsets of neurons, or to separate trial epochs to understand how DLAG's estimates depend on these elements of the neural recordings. We have already employed some of these strategies here (Fig. 5.3, Fig. 5.4, Fig. 5.6), and could continue to build upon that foundation.

Feasible extensions of the DLAG framework might better accommodate these sources of variability. mDLAG, for instance, could address the challenge of heterogeneity in the delays and timescales of individual neurons. One could, for example, set the number of populations, M, in the mDLAG model definition equal to q, the number of recorded neurons, thereby achieving a Bayesian version of time-delay GPFA (TD-GPFA)[54]. Then, the time delays and timescales of individual neurons could be modeled explicitly (but see below for a discussion of scaling). Finally, we note that DLAG is compatible with any Gaussian process covariance function[55]. The squared exponential covariance function explored here is only one member of a rich class of stationary and non-stationary covariance functions. More expressive covariance functions could be readily employed to better capture more temporally complex, non-stationary inter-areal interactions (see Appendix B).

Nonlinearity

DLAG is a linear dimensionality reduction method. However, many signals are likely represented within a neuronal population nonlinearly, and nonlinearly transformed from one brain area to the next. The potential effects of nonlinearity on DLAG estimates fall into two categories: spatial and temporal. A "spatial" nonlinearity could arise from the tuning properties of neurons in V1 and V2: the orientation of drifting

grating stimuli, for example, is represented in these two populations by a characteristic ring structure in population space (see [91] for an example). While this structure could be parsimoniously described by a one-dimensional function (the cosine of the orientation angle), two dimensions of population space are required to describe this structure. For DLAG, this common tuning-related structure between V1 and V2 might therefore require two across-area variables, rather than one.

A "temporal" nonlinearity could arise from the nonlinear filtration of signals from a source area to a target area. In general, nonlinear filtration of an input signal can lead to new frequencies—or equivalently, new timescales—in the output signal. Estimated DLAG models might therefore identify extra within-area variables to capture these additional timescales seen in the target area but not the source area. Systematically applying DLAG to neural activity simulated from models that introduce specific spatial and temporal nonlinearities will further inform interpretation of DLAG models fit to real neural recordings. Such investigations could also guide potential extensions of the DLAG framework to incorporate nonlinearities[92,93].

Scaling to many neuronal populations

mDLAG is a promising step toward studying the growing number of recordings from three or more neuronal populations. Scaling the approach to large numbers of populations, however, presents computational and conceptual challenges. Computationally, the mDLAG fitting procedure requires the inversion of a matrix with dimensions $MpT \times MpT$ (equation (6.27)), where M is the number of populations, p is the number of latent state variables, and T is the number of time points. Hence to study large-scale multi-population recordings, it will be critical to explore the many approaches to improving the scalability of Gaussian process methods[77,87,91,94].

Conceptually, multiple interpretational challenges arise when considering three or more populations. Suppose, for example, that mDLAG identifies an interaction across populations A, B, and C, with a 10 ms delay between A and B, and a 10 ms delay between B and C. These delays are consistent with a description of signal flow from A to B to C. However, they are also consistent with a configuration in which A is a common input to B (with a 10 ms delay) and to C (with a 20 ms delay). Regardless of this statistical ambiguity, there is the broader matter of conceptual scale: How do we decipher multi-dimensional, concurrent interactions across networks of four or five populations, let alone dozens[4]? New conceptual frameworks will be needed to provide insight into these large network interactions.

Toward a deeper perspective on inter-areal computation

If we take the DLAG state model at face value (Fig. 3.1, center; Fig. 3.3), then DLAG would appear to describe a limited type of interaction: the transmission of copies of a signal with some time delay. Simple transmission might reasonably describe some instances of inter-areal signaling[14,69], but certainly not most. After all, the brain computes: signals are transformed from one area to the next, not merely propagated[5].

With the right representation, however, we can see that DLAG describes a rich (albeit linear) spatiotemporal transformation across areas: "spatial" in the sense that a low-rank linear transformation takes place (see equation (5.5), the expected value of one area's activity given another's activity), and "temporal" in the sense that across-area signals are not merely propagated with a time delay, but also nontrivially filtered (see Appendix A). Concepts throughout this dissertation (Section 3.7, Section 5.4.3, Appendix A) lay a foundation for continued development of a deeper theoretical and computational framework for describing—and discovering—properties of inter-areal signal transformations. Applying this framework synergistically with (nonlinear, recurrent) network modeling will lead to both improved interpretation of models fit to multi-area activity and richer insights into inter-areal computation.

Bibliography

1. Ahrens, M. B. *et al.* Brain-wide neuronal dynamics during motor adaptation in zebrafish. *Nature* **485,** 471–477 (2012).
2. Yang, W. & Yuste, R. In vivo imaging of neural activity. *Nature Methods* **14,** 349–359 (2017).
3. Jun, J. J. *et al.* Fully integrated silicon probes for high-density recording of neural activity. *Nature* **551,** 232–236 (2017).
4. Steinmetz, N. A., Zatka-Haas, P., Carandini, M. & Harris, K. D. Distributed coding of choice, action and engagement across the mouse brain. *Nature* **576,** 266–273 (2019).
5. Kohn, A. *et al.* Principles of Corticocortical Communication: Proposed Schemes and Design Considerations. *Trends in Neurosciences* **43,** 725–737 (2020).
6. Lamme, V. A., Supèr, H. & Spekreijse, H. Feedforward, horizontal, and feedback processing in the visual cortex. *Current Opinion in Neurobiology* **8,** 529–535 (1998).
7. Angelucci, A. & Bressloff, P. C. Contribution of feedforward, lateral and feedback connections to the classical receptive field center and extra-classical receptive field surround of primate V1 neurons. *Progress in Brain Research* **154,** 93–120 (2006).
8. Gilbert, C. D. & Li, W. Top-down influences on visual processing. *Nature Reviews Neuroscience* **14,** 350–363 (2013).
9. Harris, K. D. & Mrsic-Flogel, T. D. Cortical connectivity and sensory coding. *Nature* **503,** 51–58 (2013).
10. Miller, E. K., Lundqvist, M. & Bastos, A. M. Working Memory 2.0. *Neuron* **100,** 463–475 (2018).
11. Shadmehr, R. & Krakauer, J. W. A computational neuroanatomy for motor control. *Experimental Brain Research* **185,** 359–381 (2008).
12. Keemink, S. W. & Machens, C. K. Decoding and encoding (de)mixed population responses. *Current Opinion in Neurobiology* **58,** 112–121 (2019).

13. Schmolesky, M. T. *et al.* Signal timing across the macaque visual system. *Journal of Neurophysiology* **79,** 3272–3278 (1998).

14. Hernández, A. *et al.* Decoding a Perceptual Decision Process across Cortex. *Neuron* **66,** 300–314 (2010).

15. Siegel, M., Buschman, T. J. & Miller, E. K. Cortical information flow during flexible sensorimotor decisions. *Science* **348,** 1352–1355 (2015).

16. Supèr, H., Spekreijse, H. & Lamme, V. A. F. Two distinct modes of sensory processing observed in monkey primary visual cortex (V1). *Nature Neuroscience* **4,** 304–310 (2001).

17. Pooresmaeili, A., Poort, J. & Roelfsema, P. R. Simultaneous selection by object-based attention in visual and frontal cortex. *Proceedings of the National Academy of Sciences* **111,** 6467–6472 (2014).

18. Chen, M. *et al.* Incremental integration of global contours through interplay between visual cortical areas. *Neuron* **82,** 682–694 (2014).

19. Schwiedrzik, C. M. & Freiwald, W. A. High-Level Prediction Signals in a Low-Level Area of the Macaque Face-Processing Hierarchy. *Neuron* **96,** 89–97 (2017).

20. Issa, E. B., Cadieu, C. F. & DiCarlo, J. J. Neural dynamics at successive stages of the ventral visual stream are consistent with hierarchical error signals. *eLife* **7,** e42870 (2018).

21. Reid, R. C. & Alonso, J. M. Specificity of monosynaptic connections from thalamus to visual cortex. *Nature* **378,** 281–284 (1995).

22. Roe, A. W. & Ts'o, D. Y. Specificity of Color Connectivity Between Primate V1 and V2. *Journal of Neurophysiology* **82,** 2719–2730 (1999).

23. Nowak, L. G., Munk, M., James, A. C., Girard, P. & Bullier, J. Cross-Correlation Study of the Temporal Interactions Between Areas V1 and V2 of the Macaque Monkey. *Journal of Neurophysiology* **81,** 1057–1074 (1999).

24. Jia, X., Tanabe, S. & Kohn, A. Gamma and the Coordination of Spiking Activity in Early Visual Cortex. *Neuron* **77,** 762–774 (2013).

25. Oemisch, M., Westendorff, S., Everling, S. & Womelsdorf, T. Interareal Spike-Train Correlations of Anterior Cingulate and Dorsal Prefrontal Cortex during Attention Shifts. *Journal of Neuroscience* **35,** 13076–13089 (2015).

26. Zandvakili, A. & Kohn, A. Coordinated Neuronal Activity Enhances Corticocortical Communication. *Neuron* **87,** 827–839 (2015).

27. Campo, A. T. *et al.* Feed-forward information and zero-lag synchronization in the sensory thalamocortical circuit are modulated during stimulus perception. *Proceedings of the National Academy of Sciences* **116,** 7513–7522 (2019).

28. Gregoriou, G. G., Gotts, S. J., Zhou, H. & Desimone, R. High-frequency, long-range coupling between prefrontal and visual cortex during attention. *Science* **324,** 1207–1210 (2009).

29. Salazar, R. F., Dotson, N. M., Bressler, S. L. & Gray, C. M. Content-Specific Fronto-Parietal Synchronization During Visual Working Memory. *Science* **338,** 1097–1100 (2012).

30. van Kerkoerle, T. *et al.* Alpha and gamma oscillations characterize feedback and feedforward processing in monkey visual cortex. *Proceedings of the National Academy of Sciences* **111,** 14332–14341 (2014).

31. Bastos, A. M., Vezoli, J. & Fries, P. Communication through coherence with inter-areal delays. *Current Opinion in Neurobiology* **31,** 173–180 (2015).

32. Semedo, J. D., Gokcen, E., Machens, C. K., Kohn, A. & Yu, B. M. Statistical methods for dissecting interactions between brain areas. *Current Opinion in Neurobiology* **65,** 59–69 (2020).

33. Kang, B. & Druckmann, S. Approaches to inferring multi-regional interactions from simultaneous population recordings. *Current Opinion in Neurobiology* **65,** 108–119 (2020).

34. Keeley, S. L., Zoltowski, D. M., Aoi, M. C. & Pillow, J. W. Modeling statistical dependencies in multi-region spike train data. *Current Opinion in Neurobiology* **65,** 194–202 (2020).

35. Kaufman, M. T., Churchland, M. M., Ryu, S. I. & Shenoy, K. V. Cortical activity in the null space: permitting preparation without movement. *Nature Neuroscience* **17,** 440–448 (2014).

36. Ames, K. C. & Churchland, M. M. Motor cortex signals for each arm are mixed across hemispheres and neurons yet partitioned within the population response. *eLife* **8,** e46159 (2019).

37. Semedo, J. D., Zandvakili, A., Machens, C. K., Yu, B. M. & Kohn, A. Cortical Areas Interact through a Communication Subspace. *Neuron* **102,** 249–259 (2019).

38. Perich, M. G., Gallego, J. A. & Miller, L. E. A Neural Population Mechanism for Rapid Learning. *Neuron* **100,** 964–976 (2018).

39. Ruff, D. A. & Cohen, M. R. Simultaneous multi-area recordings suggest that attention improves performance by reshaping stimulus representations. *Nature Neuroscience* **22,** 1669–1676 (2019).

40. Srinath, R., Ruff, D. A. & Cohen, M. R. Attention improves information flow between neuronal populations without changing the communication subspace. *Current Biology* **31,** 5299–5313 (2021).

41. Veuthey, T. L., Derosier, K., Kondapavulur, S. & Ganguly, K. Single-trial cross-area neural population dynamics during long-term skill learning. *Nature Communications* **11,** 4057 (2020).

42. Chen, G., Kang, B., Lindsey, J., Druckmann, S. & Li, N. Modularity and robustness of frontal cortical networks. *Cell* **184,** 3717–3730 (2021).

43. Semedo, J. D. *et al.* Feedforward and feedback interactions between visual cortical areas use different population activity patterns. *Nature Communications* **13,** 1099 (2022).

44. Cunningham, J. P. & Yu, B. M. Dimensionality reduction for large-scale neural recordings. *Nature Neuroscience* **17,** 1500–1509 (2014).

45. Hastie, T., Tibshirani, R. & Wainwright, M. *Statistical Learning with Sparsity: The Lasso and Generalizations* (CRC Press, 2016).

46. Tipping, M. E. & Bishop, C. M. Probabilistic Principal Component Analysis. en. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **61,** 611–622 (1999).

47. Yu, B. M. *et al.* Gaussian-Process Factor Analysis for Low-Dimensional Single-Trial Analysis of Neural Population Activity. *Journal of Neurophysiology* **102,** 614–635 (2009).

48. Everett, B. *An Introduction to Latent Variable Models* (Springer Netherlands, 1984).

49. Wold, H. Soft Modelling by Latent Variables: The Non-Linear Iterative Partial Least Squares (NIPALS) Approach. *Journal of Applied Probability* **12,** 117–142 (1975).

50. Hotelling, H. Relations Between Two Sets of Variates. *Biometrika* **28,** 321–377 (1936).

51. Izenman, A. J. Reduced-rank regression for the multivariate linear model. *Journal of Multivariate Analysis* **5,** 248–264 (1975).

52. Bach, F. R. & Jordan, M. I. A Probabilistic Interpretation of Canonical Correlation Analysis. Technical Report No. 688, Department of Statistics, University of California, Berkeley (2005).

53. Archambeau, C. & Bach, F. Sparse probabilistic projections. *Advances in Neural Information Processing Systems* **21,** 73–80 (2008).

54. Lakshmanan, K. C., Sadtler, P. T., Tyler-Kabara, E. C., Batista, A. P. & Yu, B. M. Extracting Low-Dimensional Latent Structure from Time Series in the Presence of Delays. *Neural Computation* **27,** 1825–1856 (2015).

55. Rasmussen, C. E. & Williams, C. K. I. *Gaussian processes for machine learning* (MIT Press, Cambridge, MA, 2006).

56. Semedo, J., Zandvakili, A., Kohn, A., Machens, C. K. & Yu, B. M. Extracting Latent Structure From Multiple Interacting Neural Populations. *Advances in Neural Information Processing Systems* **27,** 2942–2950 (2014).

57. Klami, A., Virtanen, S. & Kaski, S. Bayesian Canonical Correlation Analysis. *Journal of Machine Learning Research* **14,** 965–1003 (2013).

58. Golub, G. H. & Van Loan, C. F. *Matrix computations* Fourth edition (The Johns Hopkins University Press, Baltimore, 2013).

59. Felleman, D. J. & Van Essen, D. C. Distributed hierarchical processing in the primate cerebral cortex. *Cerebral cortex* (1991).

60. Markov, N. T. *et al.* Cortical High-Density Counterstream Architectures. *Science* **342,** 1238406 (2013).

61. Smith, M. A., Kohn, A. & Movshon, J. A. Glass pattern responses in macaque V2 neurons. *Journal of Vision* **7,** 5 (2007).

62. Murray, J. D. *et al.* A hierarchy of intrinsic timescales across primate cortex. *Nature Neuroscience* **17,** 1661–1663 (2014).

63. Runyan, C. A., Piasini, E., Panzeri, S. & Harvey, C. D. Distinct timescales of population coding across cortex. *Nature* **548,** 92–96 (2017).

64. Siegle, J. H. *et al.* Survey of spiking in the mouse visual system reveals functional hierarchy. *Nature* **592,** 86–92 (2021).

65. Williamson, R. C. *et al.* Scaling Properties of Dimensionality Reduction for Neural Populations and Network Models. *PLOS Computational Biology* **12,** e1005141 (2016).

66. Freeman, J., Ziemba, C. M., Heeger, D. J., Simoncelli, E. P. & Movshon, J. A. A functional and perceptual signature of the second visual area in primates. *Nature Neuroscience* **16,** 974–981 (2013).

67. Okazawa, G., Tajima, S. & Komatsu, H. Gradual Development of Visual Texture-Selective Properties Between Macaque Areas V2 and V4. *Cerebral Cortex* **27,** 4867–4880 (2017).

68. Smith, M. A. & Kohn, A. Spatial and Temporal Scales of Neuronal Correlation in Primary Visual Cortex. *Journal of Neuroscience* **28,** 12591–12603 (2008).

69. Cowley, B. R. *et al.* Slow Drift of Neural Activity as a Signature of Impulsivity in Macaque Visual and Prefrontal Cortex. *Neuron* **108,** 551–567 (2020).

70. Jasper, A. I., Tanabe, S. & Kohn, A. Predicting Perceptual Decisions Using Visual Cortical Population Responses and Choice History. *Journal of Neuroscience* **39,** 6714–6727 (2019).

71. Portilla, J. & Simoncelli, E. P. A Parametric Texture Model Based on Joint Statistics of Complex Wavelet Coefficients. *International Journal of Computer Vision* **40,** 49–70 (2000).

72. Klami, A., Virtanen, S., Leppäaho, E. & Kaski, S. Group Factor Analysis. *IEEE Transactions on Neural Networks and Learning Systems* **26,** 2136–2147 (2015).

73. Bishop, C. Variational principal components. *9th International Conference on Artificial Neural Networks,* 509–514 (1999).

74. Gokcen, E. *et al.* Dissecting multi-population interactions across cortical areas and layers. *Cosyne Abstracts* (2023).

75. Jasper, A. I., Xu, A., Machens, C. K., Yu, B. M. & Kohn, A. Early and midlevel visual areas interact via distinct communication subspaces. *Society for Neuroscience Abstract* (2022).

76. Gokcen, E. *et al.* Disentangling the flow of signals between populations of neurons. *Nature Computational Science* **2,** 512–525 (2022).

77. Jensen, K., Kao, T.-C., Stone, J. & Hennequin, G. Scalable Bayesian GPFA with automatic relevance determination and discrete noise models. *Advances in Neural Information Processing Systems* **34,** 10613–10626 (2021).

78. Zhuang, X., Yang, Z. & Cordes, D. A technical review of canonical correlation analysis for neuroscience applications. *Human Brain Mapping* **41,** 3807–3833 (2020).

79. Kamiński, M., Ding, M., Truccolo, W. A. & Bressler, S. L. Evaluating causal relations in neural systems: Granger causality, directed transfer function and statistical assessment of significance. *Biological Cybernetics* **85,** 145–157 (2001).

80. Quinn, C. J., Coleman, T. P., Kiyavash, N. & Hatsopoulos, N. G. Estimating the directed information to infer causal relationships in ensemble neural spike train recordings. *Journal of Computational Neuroscience* **30,** 17–44 (2011).

81. Kim, S., Putrino, D., Ghosh, S. & Brown, E. N. A Granger Causality Measure for Point Process Models of Ensemble Neural Spiking Activity. *PLOS Computational Biology* **7,** e1001110 (2011).

82. Pillow, J. W. *et al.* Spatio-temporal correlations and visual signalling in a complete neuronal population. *Nature* **454,** 995–999 (2008).

83. Truccolo, W., Hochberg, L. R. & Donoghue, J. P. Collective dynamics in human and monkey sensorimotor cortex: predicting single neuron spikes. *Nature Neuroscience* **13,** 105–111 (2010).

84. Perich, M. G. *et al.* Inferring brain-wide interactions using data-constrained recurrent neural network models. Preprint at https://doi.org/10.1101/2020.12.18.423348 (2021).

85. Rodu, J., Klein, N., Brincat, S. L., Miller, E. K. & Kass, R. E. Detecting multivariate cross-correlation between brain regions. *Journal of Neurophysiology* **120,** 1962–1972 (2018).

86. Bong, H. *et al.* Latent Dynamic Factor Analysis of High-Dimensional Neural Recordings. *Advances in Neural Information Processing Systems* **33,** 16446–16456 (2020).

87. Keeley, S., Aoi, M., Yu, Y., Smith, S. & Pillow, J. W. Identifying signal and noise structure in neural population activity with Gaussian process factor models. *Advances in Neural Information Processing Systems* **33,** 13795–13805 (2020).

88. Glaser, J., Whiteway, M., Cunningham, J. P., Paninski, L. & Linderman, S. Recurrent Switching Dynamical Systems Models for Multiple Interacting Neural Populations. *Advances in Neural Information Processing Systems* **33,** 14867–14878 (2020).

89. Pesaran, B., Hagan, M., Qiao, S. & Shewcraft, R. Multiregional communication and the channel modulation hypothesis. *Current Opinion in Neurobiology. Developmental Neuroscience* **66,** 250–257 (2021).

90. Reid, A. T. *et al.* Advancing functional connectivity research from association to causation. *Nature Neuroscience* **22,** 1751–1760 (2019).

91. Zhao, Y. & Park, I. M. Variational Latent Gaussian Process for Recovering Single-Trial Dynamics from Population Spike Trains. *Neural Computation* **29,** 1293–1316 (2017).

92. Wu, A., Roy, N. A., Keeley, S. & Pillow, J. W. Gaussian process based nonlinear latent structure discovery in multivariate spike train data. *Advances in Neural Information Processing Systems* **30** (2017).

93. Duncker, L., Ruda, K. M., Field, G. D. & Pillow, J. W. Scalable variational inference for low-rank spatio-temporal receptive fields. Preprint at https://doi.org/10.1101/2022.08.12.503812 (2022).

94. Duncker, L. & Sahani, M. Temporal alignment and latent Gaussian process factor inference in population spike trains. *Advances in Neural Information Processing Systems* **31,** 10445–10455 (2018).

Appendix A

Linear transformations of DLAG latent variables

In Chapter 3, we introduced dual interpretational perspectives of the DLAG model as (1) a partitioning of each area's population space into within- and across-area subspaces (Fig. 3.1) and (2) a low-rank decomposition of the covariance matrix, $\widetilde{\Sigma}$ (equation (3.43)). Here we continue to build theoretical insight into the model by introducing linear transformations of DLAG's latent variables into ordered "modes" of population activity. These modes are valuable tools for addressing several interrelated questions:

1. How do we interpret the dimensionality of fitted DLAG models?
2. How do fitted DLAG models work to describe complex temporal structure (i.e., temporal structure that is more complicated than a Gaussian process with squared exponential covariance function)?
3. Does DLAG merely describe the delayed transmission of signals across areas, or does it describe a nontrivial transformation of these signals?

We will demonstrate these new concepts on the example dataset from our V1-V2 recordings (Fig. 5.2a).

A.1 Dominant modes within an area

Recall that the columns of the across- and within-area loading matrices, C_m^a and C_m^w, are linearly independent but not, in general, orthogonal. Furthermore, the ordering of the columns of each loading matrix, and of the corresponding latent variables, is arbitrary (see Section 3.3). The statistics derived in Section 5.4.1 (equations (5.1)–(5.3)) were thus critical to support the interpretation of latent variables estimated from real neural data (Secton 5.1).

A common alternative practice, in conjunction with single-area latent variable models like factor analysis (FA)[65] or Gaussian process factor analysis (GPFA)[47], is to transform the latent variables *post*

hoc to a more interpretable basis. We can take that approach here for DLAG as follows. First define $C_m = [C_m^a \; C_m^w] \in \mathbb{R}^{q_m \times p_m}$ by horizontally concatenating C_m^a and C_m^w for area m ($p_m = p^a + p_m^w$). Then, the singular value decomposition of C_m is given by $C_m = U_m S_m V_m^\top$ where $U_m \in \mathbb{R}^{q_m \times p_m}$, $S_m \in \mathbb{S}^{p_m \times p_m}$, and $V_m \in \mathbb{R}^{p_m \times p_m}$. Next we group together the across- and within-area latent variables at time t for the m^{th} brain area to define $\mathbf{x}_{m,t} = [\mathbf{x}_{m,:,t}^{a\top} \; \mathbf{x}_{m,:,t}^{w\top}]^\top \in \mathbb{R}^{p_m}$, and from the DLAG observation model (equation (3.1)) we can write

$$\begin{aligned}
\mathbb{E}[\mathbf{y}_{m,t} \mid \mathbf{x}_{m,t}] &= C_m \mathbf{x}_{m,t} + \mathbf{d}_m && \text{(A.1)} \\
&= U_m S_m V_m^\top \mathbf{x}_{m,t} + \mathbf{d}_m && \text{(A.2)} \\
&= U_m \mathbf{z}_{m,t} + \mathbf{d}_m && \text{(A.3)}
\end{aligned}$$

where we have defined a transformed set of latent variables $\mathbf{z}_{m,t} = S_m V_m^\top \mathbf{x}_{m,t} \in \mathbb{R}^{p_m}$. These transformed latent variables have two desirable properties: (1) they lie in an orthonormal subspace defined by the columns of U_m and (2) they are ordered according to shared variance explained within area m.

To see the second property, recall the within-area covariance matrix $\widetilde{K}_m^w \in \mathbb{S}^{p_m^w T \times p_m^w T}$ (defined in equation (3.40)) and the across-area auto-covariance matrix $\widetilde{K}_{m,m}^a \in \mathbb{S}^{p^a T \times p^a T}$ (defined in equation (3.41)). Collect these matrices into the block-diagonal matrix

$$\widetilde{K}_m = \begin{bmatrix} \widetilde{K}_{m,m}^a & \mathbf{0} \\ \mathbf{0} & \widetilde{K}_m^w \end{bmatrix} \in \mathbb{R}^{p_m T \times p_m T} \tag{A.4}$$

See also the definition of the DLAG state model given in equation (3.38). Then we can write the covariance matrix of the transformed latent variables $\mathbf{z}_{m,t}$ as

$$\begin{aligned}
\operatorname{cov}(\mathbf{z}_{m,t}, \mathbf{z}_{m,t}) &= \mathbb{E}[\mathbf{z}_{m,t} \mathbf{z}_{m,t}^\top] && \text{(A.5)} \\
&= \mathbb{E}[S_m V_m^\top \mathbf{x}_{m,t} \mathbf{x}_{m,t}^\top V_m S_m] && \text{(A.6)} \\
&= S_m V_m^\top \mathbb{E}[\mathbf{x}_{m,t} \mathbf{x}_{m,t}^\top] V_m S_m && \text{(A.7)} \\
&= S_m V_m^\top \widetilde{K}_m(t,t) V_m S_m && \text{(A.8)} \\
&= S_m V_m^\top I_{p_m} V_m S_m && \text{(A.9)} \\
&= S_m V_m^\top V_m S_m && \text{(A.10)} \\
&= (S_m)^2 && \text{(A.11)}
\end{aligned}$$

$\widetilde{K}_m(t,t) = I_{p_m}$ is block (t,t) of $\widetilde{K}_m$. Because the matrix S_m is diagonal, the latent variables $\mathbf{z}_{m,t}$ are independent of one another, and the variance of each $z_{m,j,t}$, $j = 1, \ldots, p_m$ is given by the squared diagonal elements of S_m, or equivalently, the eigenvalues of the shared covariance matrix $C_m \widetilde{K}_m(t,t) C_m^\top = C_m C_m^\top$.

We will refer to the columns of U_m from here on as "dominant modes,", i.e., the modes that explain the greatest shared variance in area m.

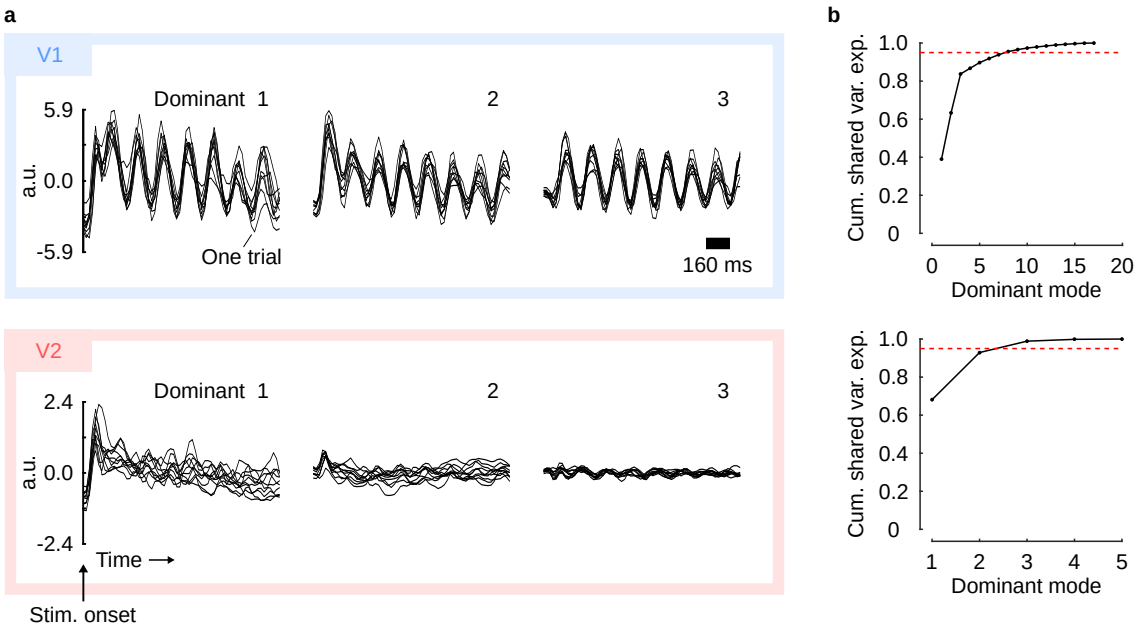

Figure A.1: Dominant modes in V1 and V2. (**a**) Time courses of activity along the dominant modes of V1 and V2 (same dataset as in Fig. 5.2). Top row / blue box: V1 dominant activity. Bottom row / red box: V2 dominant activity. Each panel corresponds to the single-trial time courses along a single dominant mode. All time courses are aligned to stimulus onset. a.u.: arbitrary units. Each black trace corresponds to one trial; for clarity, only 10 of 400 are shown. Note that the polarity of traces is arbitrary, as long as it is consistent with the polarity of U_m. In V1, the top 3 of 17 dominant modes are displayed. In V2, the top 3 of 5 dominant modes are displayed. (**b**) Cumulative shared variance explained as a function of number of dominant modes. Top: V1. Bottom: V2. Red dashed lines indicate 95% threshold.

Revisiting the V1-V2 recordings Revisiting our example V1-V2 dataset (across- and within-area latent variables shown in Fig. 5.2a), we computed time courses along the top three dominant modes in both V1 (Fig. A.1a, top) and V2 (Fig. A.1a, bottom). From these time courses its apparent just how dominant the periodic structure of the drifting grating stimulus is in V1, but not in V2. To further characterize the structure of the dominant subspace in each area, we computed the cumulative shared variance explained in each mode (Fig. A.1b; computed from equation (A.11)). Model selection led to a DLAG model with 17 total latent variables for V1 and 5 total latent variables for V2. However, only 8 dominant modes in V1 and 3 dominant modes in V2 were needed to explain at least 95% of the shared variance in their respective areas. Part of the discrepancy between the number of selected latent variables and the number of significant dominant modes could be due to a similar phenomenon as observed for FA in data-rich regimes[65]. However, as we will continue to discuss below, DLAG might also employ "extra" latent variables to account for complex temporal structure[47].

A.2 Modes across areas

The dominant modes characterize activity in each area independently. We now seek to derive modes with a direct connection to across-area interaction and that have desirable properties analogous to those of the dominant modes. To do so, we can draw inspiration from the classical methods partial least squares (PLS), canonical correlation analysis (CCA), and reduced-rank regression (RRR), which can all be seen as singular value decompositions of the (normalized) cross-covariance matrix between areas (see Chapter 2). These methods produce modes defined in pairs, where the elements of each pair correspond to dimensions in the population spaces of each area.

Covariant modes We begin with PLS-like "covariant" modes, which have the following properties: (1) modes are paired across areas; (2) modes within an area form an orthonormal basis in that area; (3) mode pairs are ordered according to shared covariance across areas; and (4) a mode in one area shares zero covariance with a mode outside of its pair in the other area. First, consider the shared cross-covariance matrix (evaluated at zero-lag) given by $\Sigma_{12} = C_1 \widetilde{K}^a_{1,2}(t,t) C_2^\top \in \mathbb{R}^{q_1 \times q_2}$. Note that, because of time delays across areas, the diagonal cross-covariance matrix $\widetilde{K}^a_{1,2}(t,t)$ is not the identity matrix. Then the singular value decomposition is given by $\Sigma_{12} = V_1 S V_2^\top$ where $V_1 \in \mathbb{R}^{q_1 \times p^a}$, $S \in \mathbb{S}^{p^a \times p^a}$, and $V_2 \in \mathbb{R}^{q_2 \times p^a}$.

Next, define the following transformed variables for area $m = 1, 2$:

$$\mathbf{z}_{m,t} = V_m^\top C_m^a \mathbf{x}_{m,t}^a \in \mathbb{R}^{p^a} \tag{A.12}$$

These latent variables lie in an orthonormal subspace of area m's population space defined by the columns of V_m. And from the cross-covariance between transformed variables, we can see the final two of four properties outlined above:

$$\begin{aligned}
\operatorname{cov}(\mathbf{z}_{1,t}, \mathbf{z}_{2,t}) &= \mathbb{E}[\mathbf{z}_{1,t} \mathbf{z}_{2,t}^\top] && \text{(A.13)} \\
&= \mathbb{E}[V_1^\top C^a \mathbf{x}_{1,t}^a \mathbf{x}_{2,t}^{a\top} C_2^{a\top} V_2] && \text{(A.14)} \\
&= V_1^\top C^a \mathbb{E}[\mathbf{x}_{1,t}^a \mathbf{x}_{2,t}^{a\top}] C_2^{a\top} V_2 && \text{(A.15)} \\
&= V_1^\top C^a \widetilde{K}^a_{1,2}(t,t) C_2^{a\top} V_2 && \text{(A.16)} \\
&= V_1^\top \Sigma_{12} V_2 && \text{(A.17)} \\
&= V_1^\top V_1 S V_2^\top V_2 && \text{(A.18)} \\
&= S && \text{(A.19)}
\end{aligned}$$

Properties (3) and (4) above follow from the diagonal structure of S. The covariance of each pair ($z_{1,j,t}$, $z_{2,j,t}$), $j = 1, \ldots, p_m$ is given by the diagonal elements of S.

Correlative modes Following a very similar procedure, we can construct CCA-like "correlative" modes. These correlative modes have the following properties: (1) modes are paired across areas; (2) modes within an area form an uncorrelated (not necessarily orthogonal) basis in that area; (3) mode pairs are ordered according to correlation across areas; and (4) a mode in one area is uncorrelated with a mode outside of its pair in the other area. First, consider the shared cross-correlation matrix (evaluated at zero-lag) given by $\Sigma_{11}^{-\frac{1}{2}}\Sigma_{12}\Sigma_{22}^{-\frac{1}{2}} = (C_1^a C_1^{a\top} + C_1^w C_1^{w\top} + R_1)^{-\frac{1}{2}}(C_1\widetilde{K}_{1,2}^a(t,t)C_2^\top)(C_2^a C_2^{a\top} + C_2^w C_2^{w\top} + R_2)^{-\frac{1}{2}} \in \mathbb{R}^{q_1\times q_2}$. Then the singular value decomposition is given by $\Sigma_{11}^{-\frac{1}{2}}\Sigma_{12}\Sigma_{22}^{-\frac{1}{2}} = V_1 S V_2^\top$ where $V_1 \in \mathbb{R}^{q_1\times p^a}$, $S \in \mathbb{S}^{p^a\times p^a}$, and $V_2 \in \mathbb{R}^{q_2\times p^a}$.

Next for area $m = 1, 2$, let $U_m = \Sigma_{mm}^{-\frac{1}{2}} V_m$ and define the following transformed variables:

$$\mathbf{z}_{m,t} = U_m^\top C_m^a \mathbf{x}_{m,t}^a \in \mathbb{R}^{p^a} \tag{A.20}$$

These latent variables lie in a subspace of area m's population space defined by the columns of U_m, which are uncorrelated but not necessarily orthogonal. And from the cross-covariance between transformed variables, we can see the final two of four properties outlined above:

$$\begin{aligned}
\operatorname{cov}(\mathbf{z}_{1,t}, \mathbf{z}_{2,t}) &= \mathbb{E}[\mathbf{z}_{1,t}\mathbf{z}_{2,t}^\top] && \text{(A.21)}\\
&= \mathbb{E}[U_1^\top C^a \mathbf{x}_{1,t}^a \mathbf{x}_{2,t}^{a\top} C_2^{a\top} U_2] && \text{(A.22)}\\
&= U_1^\top C^a \mathbb{E}[\mathbf{x}_{1,t}^a \mathbf{x}_{2,t}^{a\top}] C_2^{a\top} U_2 && \text{(A.23)}\\
&= U_1^\top C^a \widetilde{K}_{1,2}^a(t,t) C_2^{a\top} U_2 && \text{(A.24)}\\
&= V_1^\top \Sigma_{11}^{-\frac{1}{2}} \Sigma_{12} \Sigma_{22}^{-\frac{1}{2}} V_2 && \text{(A.25)}\\
&= V_1^\top V_1 S V_2^\top V_2 && \text{(A.26)}\\
&= S && \text{(A.27)}
\end{aligned}$$

Properties (3) and (4) above follow from the diagonal structure of S. The cross-correlation of each pair $(z_{1,j,t}, z_{2,j,t})$, $j = 1, \ldots, p_m$ is given by the diagonal elements of S.

Predictive modes Finally, we can construct RRR-like "predictive" modes. Whereas the covariant and correlative modes defined above can be thought of as symmetric, the predictive modes cast one area as the source and the other area as the target. These predictive modes have the following properties: (1) modes are paired across areas; (2) modes within the source area form an uncorrelated (not necessarily orthogonal) basis in that area; (3) modes within the target area form an orthonormal basis in that area; (4) mode pairs are ordered according to predictive power, from source to target; and (5) a mode in the source area has no predictive power versus a mode outside of its pair in the target area. Without loss

of generality, let area 1 be the source area and let area 2 be the target area. Then, consider the matrix $\Sigma_{11}^{-\frac{1}{2}}\Sigma_{12} = (C_1^a C_1^{a\top} + C_1^w C_1^{w\top} + R_1)^{-\frac{1}{2}}(C_1 \widetilde{K}_{1,2}^a(t,t) C_2^\top) \in \mathbb{R}^{q_1 \times q_2}$. Its singular value decomposition is given by $\Sigma_{11}^{-\frac{1}{2}}\Sigma_{12} = V_1 S V_2^\top$ where $V_1 \in \mathbb{R}^{q_1 \times p^a}$, $S \in \mathbb{S}^{p^a \times p^a}$, and $V_2 \in \mathbb{R}^{q_2 \times p^a}$.

Next let $U_1 = \Sigma_{11}^{-\frac{1}{2}} V_1$ and define the following set of transformed variables:

$$\mathbf{z}_{1,t} = U_1^\top C_1^a \mathbf{x}_{1,t}^a \in \mathbb{R}^{p^a} \tag{A.28}$$

$$\mathbf{z}_{2,t} = V_2^\top C_2^a \mathbf{x}_{2,t}^a \in \mathbb{R}^{p^a} \tag{A.29}$$

The latent variables $\mathbf{z}_{1,t}$ lie in a subspace of area 1's population space defined by the columns of U_1, which are uncorrelated but not necessarily orthogonal. The latent variables $\mathbf{z}_{2,t}$ lie in an orthonormal subspace of area 2's population space defined by the columns of V_2. And from the cross-covariance between transformed variables, we can see the final two properties outlined above:

$$\begin{aligned}
\operatorname{cov}(\mathbf{z}_{1,t}, \mathbf{z}_{2,t}) &= \mathbb{E}[\mathbf{z}_{1,t}\mathbf{z}_{2,t}^\top] && \text{(A.30)}\\
&= \mathbb{E}[U_1^\top C^a \mathbf{x}_{1,t}^a \mathbf{x}_{2,t}^{a\top} C_2^{a\top} V_2] && \text{(A.31)}\\
&= U_1^\top C^a \mathbb{E}[\mathbf{x}_{1,t}^a \mathbf{x}_{2,t}^{a\top}] C_2^{a\top} V_2 && \text{(A.32)}\\
&= U_1^\top C^a \widetilde{K}_{1,2}^a(t,t) C_2^{a\top} V_2 && \text{(A.33)}\\
&= V_1^\top \Sigma_{11}^{-\frac{1}{2}} \Sigma_{12} V_2 && \text{(A.34)}\\
&= V_1^\top V_1 S V_2^\top V_2 && \text{(A.35)}\\
&= S && \text{(A.36)}
\end{aligned}$$

Properties (3) and (4) above follow from the diagonal structure of S. The predictive power from $z_{1,j,t}$ to $z_{2,j,t}$, $j = 1, \ldots, p_m$ is given by the diagonal elements of S, and variance explained in the target area along each mode j, akin to an R^2 value, can be computed according to $R_j^2 = (S_{jj}^2)/\operatorname{tr}(\Sigma_{22})$.

Revisiting the V1-V2 recordings Returning to our example V1-V2 dataset, we computed time courses along the three covariant modes in V1 (Fig. A.2a, top) and in V2 (Fig. A.2a, bottom) (considering instead the correlative or predictive modes leads to comparable results). Activity along the covariant modes is qualitatively different from that along the dominant modes in each area (compare Fig. A.2a to Fig. A.1a). For example, V1 activity features the sinusoidal stimulus signal less strongly along its top covariant mode than along its top dominant mode. Inspection of the top covariant mode in V2 suggests that such a signal is not a prominent component of V1-V2 interaction. To further characterize the structure of the covariant modes, we computed the cumulative shared covariance explained in each mode (Fig. A.2b; computed from equation (A.19)). Model selection led to a DLAG model with 3 across-area latent variables. However, only the top 2 covariant modes are needed to explain at least 95% of the shared covariance across areas.

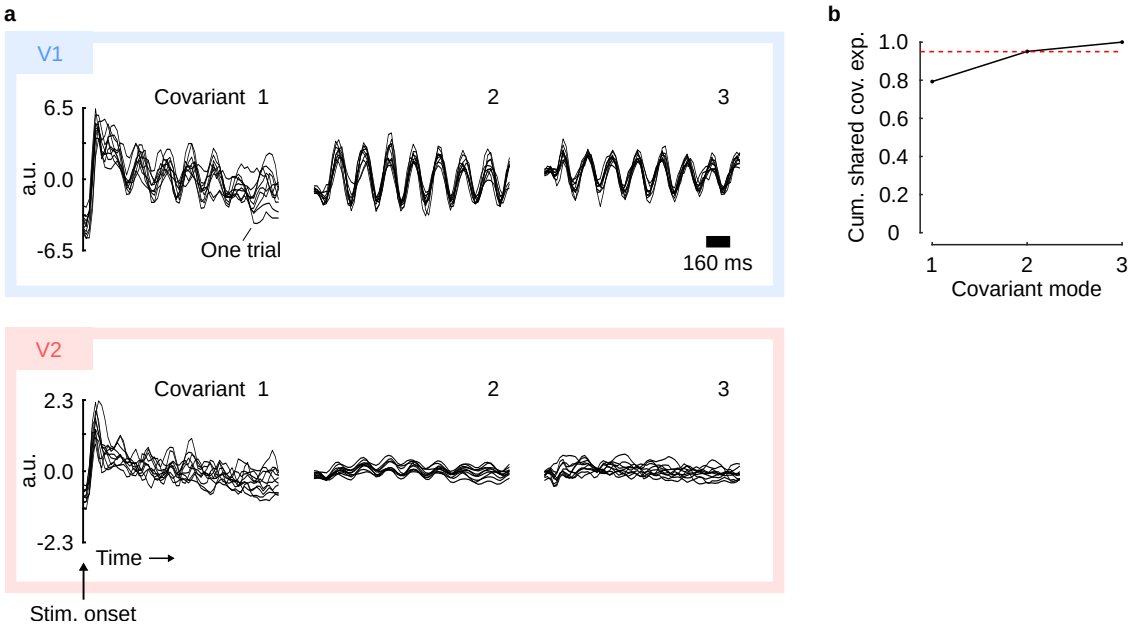

Figure A.2: Covariant modes across V1 and V2. (**a**) Time courses of activity along the covariant modes of V1 and V2 (same dataset as in Fig. 5.2). Top row / blue box: V1 covariant activity. Bottom row / red box: V2 covariant activity. Each panel corresponds to the single-trial time courses along a single covariant mode. Covariant modes are vertically paired across areas. All time courses are aligned to stimulus onset. a.u.: arbitrary units. Each black trace corresponds to one trial; for clarity, only 10 of 400 are shown. Note that the polarity of traces is arbitrary, as long as it is consistent with the polarity of V_m. (**b**) Cumulative shared cross-covariance explained as a function of number of covariant modes. Red dashed lines indicate 95% threshold.

A.3 Transformed Gaussian process covariance functions

With the development of these modes within and across areas, we have gained the interpretational benefits of ordered, orthonormal (or uncorrelated) bases. The conceptual power of DLAG, however, lies in its temporal features, namely the GP timescales and time delays. Thus our remaining goal is to characterize the temporal structure of activity along any given mode.

Let $V_1 \in \mathbb{R}^{q_1 \times r}$, $V_2 \in \mathbb{R}^{q_2 \times r}$ be coupled basis sets for each area (these could be, for example, dominant modes, covariant modes, etc.). Here we study the temporal properties of projections of neural activity onto these basis sets, $V_1^\top \mathbf{y}_{1,t}$ and $V_2^\top \mathbf{y}_{2,t}$. Therefore we will need to consider all time points in a sequence. Recall from Section 3.7 the definitions $\widetilde{\mathbf{y}}_1 = [\mathbf{y}_{1,1}^\top \cdots \mathbf{y}_{1,T}^\top]^\top \in \mathbb{R}^{q_1 T}$ and $\widetilde{\mathbf{y}}_2 = [\mathbf{y}_{2,1}^\top \cdots \mathbf{y}_{2,T}^\top]^\top \in \mathbb{R}^{q_2 T}$, obtained by vertically concatenating the observed neural activity $\mathbf{y}_{1,t}$ and $\mathbf{y}_{2,t}$ in areas 1 and 2, respectively, across all times $t = 1, \ldots, T$. Then define $\widetilde{V}_1 \in \mathbb{R}^{q_1 T \times rT}$ and $\widetilde{V}_2 \in \mathbb{R}^{q_2 T \times rT}$ to be block diagonal matrices comprising T copies of V_1 and V_2, respectively. The projected activity at all time points is then given by $\widetilde{V}_1^\top \widetilde{\mathbf{y}}_1 \in \mathbb{R}^{rT}$ and $\widetilde{V}_2^\top \widetilde{\mathbf{y}}_2 \in \mathbb{R}^{rT}$, and we can express their joint distribution as (following from equations (3.42) and

(3.43)):

$$\begin{bmatrix} \widetilde{V}_1^\top \widetilde{\mathbf{y}}_1 \\ \widetilde{V}_2^\top \widetilde{\mathbf{y}}_2 \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \widetilde{V}_1^\top \widetilde{\mathbf{d}}_1 \\ \widetilde{V}_2^\top \widetilde{\mathbf{d}}_2 \end{bmatrix}, \begin{bmatrix} \widetilde{V}_1^\top \widetilde{\Sigma}_{11} \widetilde{V}_1 & \widetilde{V}_1^\top \widetilde{\Sigma}_{12} \widetilde{V}_2 \\ \widetilde{V}_2^\top \widetilde{\Sigma}_{21} \widetilde{V}_1 & \widetilde{V}_2^\top \widetilde{\Sigma}_{22} \widetilde{V}_2 \end{bmatrix} \right) \tag{A.37}$$

where

$$\begin{bmatrix} \widetilde{\Sigma}_{11} & \widetilde{\Sigma}_{12} \\ \widetilde{\Sigma}_{21} & \widetilde{\Sigma}_{22} \end{bmatrix} = \begin{bmatrix} \widetilde{C}_1^a \widetilde{K}_{1,1}^a \widetilde{C}_1^{a\top} + \widetilde{C}_1^w \widetilde{K}_1^w \widetilde{C}_1^{w\top} + \widetilde{R}_1 & \widetilde{C}_1^a \widetilde{K}_{1,2}^a \widetilde{C}_2^{a\top} \\ \widetilde{C}_2^a \widetilde{K}_{2,1}^a \widetilde{C}_1^{a\top} & \widetilde{C}_2^a \widetilde{K}_{2,2}^a \widetilde{C}_2^{a\top} + \widetilde{C}_2^w \widetilde{K}_2^w \widetilde{C}_2^{w\top} + \widetilde{R}_2 \end{bmatrix} \tag{A.38}$$

The projected activity, like the latent variables in the original DLAG model definition, follow a Gaussian process. Our goal is to find expressions for the auto- and cross-covariance functions $k^v_{m,m,i}(t_1, t_2)$ and $k^v_{m_1,m_2,i}(t_1, t_2)$, respectively, for mode $i = 1, \ldots, r$. Consider block (t_1, t_2) of the within-area covariance for projections in area m:

$$(\widetilde{V}_m^\top \widetilde{\Sigma}_{mm} \widetilde{V}_m)_{t_1,t_2} = V_m^\top \left[C_m^a \widetilde{K}_{m,m}^a(t_1, t_2) C_m^{a\top} + C_m^w \widetilde{K}_m^w(t_1, t_2) C_m^{w\top} + \delta_{\Delta t} \cdot R_m \right] V_m \tag{A.39}$$

$$= V_m^\top \left[\sum_{j=1}^{p^a} \mathbf{c}_{m,j}^a \mathbf{c}_{m,j}^{a\top} \cdot k_{m,m,j}^a(t_1, t_2) + \sum_{j=1}^{p_m^w} \mathbf{c}_{m,j}^w \mathbf{c}_{m,j}^{w\top} \cdot k_{m,j}^w(t_1, t_2) + \delta_{\Delta t} \cdot R_m \right] V_m \tag{A.40}$$

$\delta_{\Delta t}$ is the kronecker delta (1 for $t_1 = t_2$, 0 otherwise), and $k_{m,j}^w$ and $k_{m,m,j}^a$ are the within- and across-area GP covariance functions defined in equations (3.4) and (3.7), respectively.

From here, the auto-covariance function of projected activity along basis vector i in area m, $\mathbf{v}_{m,i}$ is given by

$$k^v_{m,m,i}(t_1, t_2) = \mathbf{v}_{m,i}^\top \left[\sum_{j=1}^{p^a} \mathbf{c}_{m,j}^a \mathbf{c}_{m,j}^{a\top} \cdot k_{m,m,j}^a(t_1, t_2) + \sum_{j=1}^{p_m^w} \mathbf{c}_{m,j}^w \mathbf{c}_{m,j}^{w\top} \cdot k_{m,j}^w(t_1, t_2) + \delta_{\Delta t} R_m \right] \mathbf{v}_{m,i} \tag{A.41}$$

$$= \sum_{j=1}^{p^a} \mathbf{v}_{m,i}^\top \mathbf{c}_{m,j}^a \mathbf{c}_{m,j}^{a\top} \mathbf{v}_{m,i} \cdot k_{m,m,j}^a(t_1, t_2) + \sum_{j=1}^{p_m^w} \mathbf{v}_{m,i}^\top \mathbf{c}_{m,j}^w \mathbf{c}_{m,j}^{w\top} \mathbf{v}_{m,i} \cdot k_{m,j}^w(t_1, t_2) + \delta_{\Delta t} \cdot \mathbf{v}_{m,i}^\top R_m \mathbf{v}_{m,i} \tag{A.42}$$

$$= \sum_{j=1}^{p^a} \alpha_j \cdot k_{m,m,j}^a(t_1, t_2) + \sum_{j=1}^{p_m^w} \beta_j \cdot k_{m,j}^w(t_1, t_2) + \delta_{\Delta t} \cdot \mathbf{v}_{m,i}^\top R_m \mathbf{v}_{m,i} \tag{A.43}$$

where we've defined scalars $\alpha_j = \mathbf{v}_{m,i}^\top \mathbf{c}_{m,j}^a \mathbf{c}_{m,j}^{a\top} \mathbf{v}_{m,i}$ and $\beta_j = \mathbf{v}_{m,i}^\top \mathbf{c}_{m,j}^w \mathbf{c}_{m,j}^{w\top} \mathbf{v}_{m,i}$ to emphasize the key structure of the covariance function $k^v_{m,m,i}(t_1, t_2)$: it is a linear mixture of the original within- and across-area GP covariance functions $k_{m,j}^w$ and $k_{m,m,j}^a$.

By similar logic, the cross-covariance function between areas m_1 and m_2 is given by

$$k^v_{m_1,m_2,i}(t_1, t_2) = \sum_{j=1}^{p^a} \mathbf{v}_{m_1,i}^\top \mathbf{c}_{m_1,j}^a \mathbf{c}_{m_2,j}^{a\top} \mathbf{v}_{m_2,i} \cdot k_{m_1,m_2,j}^a(t_1, t_2) \tag{A.44}$$

$$= \sum_{j=1}^{p^a} \gamma_j \cdot k_{m_1,m_2,j}^a(t_1, t_2) \tag{A.45}$$

that is, a linear mixture of the across-area GP cross-covariance functions $k_{m_1,m_2,j}^a$. Note that within-area GP covariance functions do not contribute.

$k^v_{m,m,i}(t_1, t_2)$ and $k^v_{m_1,m_2,i}(t_1, t_2)$ are stationary, depending only on the time difference $(t_2 - t_1)$, since $k^w_{m,j}$ and $k^a_{m_1,m_2,j}$ are stationary. We obtain normalized (cross)-correlation functions via

$$\rho^v_{m_1,m_2,i}(t_2 - t_1) = \frac{k^v_{m_1,m_2,i}(t_2 - t_1)}{\sqrt{k^v_{m_1,m_1,i}(0)} \cdot \sqrt{k^v_{m_2,m_2,i}(0)}} \tag{A.46}$$

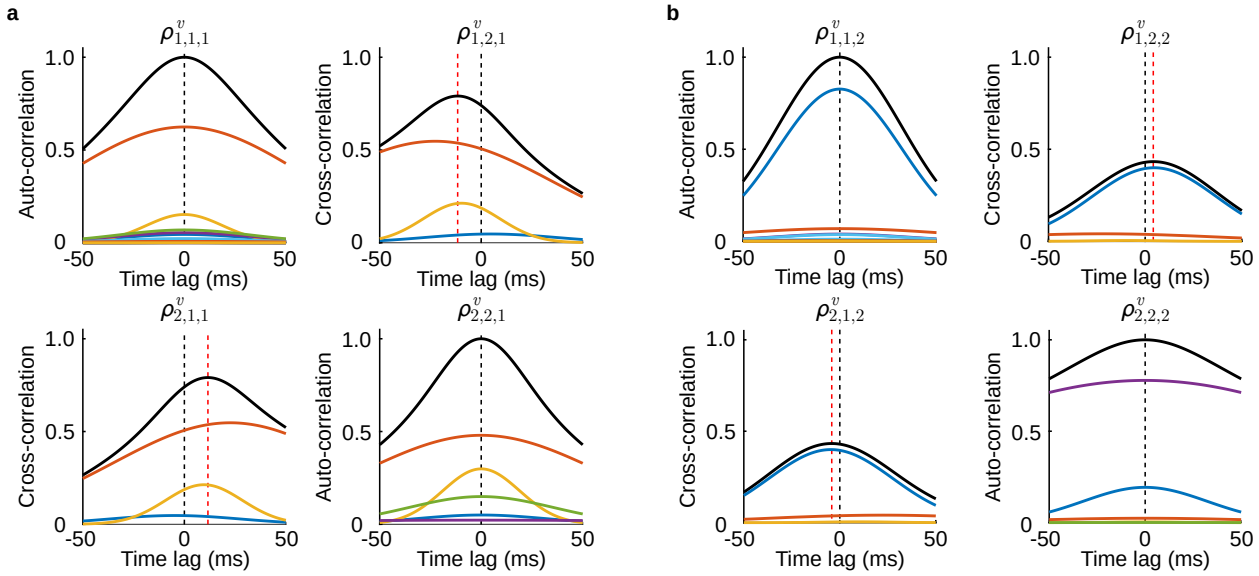

Figure A.3: GP correlation functions of V1-V2 covariant modes and their mixture components. (**a**) First covariant mode. (**b**) Second covariant mode. These modes correspond to those shown in Fig. A.2. In either panel, diagonal plots show auto-correlation functions (within an area). Off-diagonal plots show cross-correlation functions (across areas). Black: true value of the cross-correlation function, given by equation (A.46). All other colors show the mixture components that add up to the black curves (see equations (A.43) and (A.45)). Colors are consistent across all plots within a panel, so that the orange curve always corresponds to the same latent variable, and so on. Black dashed line indicates zero-lag. Red dashed line indicates time lag at which each cross-correlation function is maximized. Thus covariant mode 1 (a) implies a feedback interaction from V2 to V1 with time delay 12 ms. Covariant mode 2 (b) implies a feedforward interaction from V1 to V2 with time delay 4 ms.

Revisiting the V1-V2 recordings

Returning for a final time to our example V1-V2 dataset, we computed the auto- and cross-correlation functions (equation (A.46)) of the top two covariant modes shown in Fig. A.2 (Fig. A.3, black curves). Inspection of the cross-correlation functions reveals that the first covariant mode reflects a predominantly feedback interaction, from V2 to V1, with time delay 12 ms (Fig. A.3a, top right panel, red dashed line). Concurrently, the second covariant mode reflects a predominantly feedforward interaction, from V1 to V2, with time delay 4 ms (Fig. A.3b, top right panel, red dashed line). The ordering of the covariant modes conveys that the feedback interaction is more prominent than the feedforward interaction (see also the relative covariances in Fig. A.2b). Furthermore, these two covariant modes are orthogonal, by definition,

in both V1 and V2. This property suggests, intriguingly, that these bidirectional signals are represented by V1 and V2 population activity patterns in such a way that they do not interfere with each other.

To investigate how the original within- and across-area latent variables mix and contribute to the covariant modes, we also computed (and normalized) the individual terms of summation in equations (A.43) and (A.45) (Fig. A.3, colored curves). The auto-correlation functions for covariant modes in V1 are a linear combination of 17 symmetric (i.e., zero-centered) squared exponential components (3 across-area variables, 14 within-area variables), and the auto-correlation functions for covariant modes in V2 are a linear combination of 5 symmetric squared exponential components (3 across-area variables, 2 within-area variables). The cross-correlation functions are a linear combination of 3 asymmetric (i.e., delay-shifted) squared exponential components (the 3 across-area variables).

A few observations are particularly instructive. First, the cross-correlation function of the top covariant mode is primarily a mixture of the two negative-delay across-area variables (Fig. A.3a, top right panel, orange and yellow curves). It thus includes a mixture of two timescales—a long timescale and a short timescale—and a mixture of two time delays (-23 ms and -10 ms; see also Fig. 5.2a). Similarly, the auto-correlation functions include a mixture of many timescales of different lengths. Next, notice that, for either mode, the auto-correlation function for V1 has a different shape than the auto-correlation function for V2 (Fig. A.3a,b; compare black curves in the diagonal plots). This fact contrasts with the definition of the DLAG across-area state model, in which GP auto-covariance functions are the same in both areas (Fig. 3.3c,d). For covariant mode auto-correlation functions, across-area variables contribute differently in different areas (for example, observe the different heights of the orange and yellow curves between diagonal plots in Fig. A.3a), and within-area variables produce additional unique contributions (see the green curve in the bottom right plot of Fig. A.3a or the purple curve in the bottom right plot of Fig. A.3b). This phenomenon implies that aspects of the activity along the covariant modes are nontrivially filtered as they propagate between areas.

An updated perspective on DLAG

We are now able to provide answers to our three questions posed at the outset:

1. The number of latent variables in a fitted DLAG model reflects two sources of complexity in the neural activity: "spatial," i.e., the number of dimensions occupied in the population space, and "temporal," i.e., the timescales and time delays that describe the time course of activity within and across areas.

2. Even for an interaction that approximately occupies one "spatial" dimension of population space,

DLAG might employ multiple latent variables, with different timescales and/or time delays, to better capture the temporal structure of that interaction.

3. With the right representation, we can see that DLAG describes a spatiotemporal transformation across areas: spatial in the sense that a low-rank linear transformation takes place (see equation (5.5)), and temporal in the sense that across-area signals are not merely propagated with a time delay, but also nontrivially filtered.

Appendix B

Effects of Gaussian process covariance mismatch

Throughout this work, we have chosen to use the squared exponential function to describe the Gaussian process covariances of DLAG's latent variables (equations (3.4), (3.7), and (6.15)). Here, we investigate the question of how estimates of the Gaussian process timescales and time delays behave when the temporal structure of latent activity does not, in fact, follow a Gaussian process with squared exponential covariance function. In particular, we will consider two illustrative case studies, where (1) the time course of a latent variable is sinusoidal, and (2) the time course of a latent variable reflects a bidirectional interaction.

B.1 A latent variable with sinusoidal temporal structure

We first generated simulated neural activity from two areas, $q_1 = q_2 = 10$ neurons each, from a DLAG generative model. We generated 100 trials. Each trial comprised $T = 25$ time points, corresponding to 500 ms sequences sampled with a period of 20 ms. The activity in both areas comprised a single across-area latent variable, whose time course on each trial was sinusoidal (Fig. B.1a, black traces). To produce this temporal structure, we modified the across-area Gaussian process covariance function (equation (3.7)) as follows:

$$k^a_{m_1,m_2,j}(t_1, t_2) = \cos\left(2\pi\nu_j \Delta t\right) \tag{B.1}$$

$$\Delta t = (t_2 - t_1) - D_j \tag{B.2}$$

where $\nu_j \in \mathbb{R}_{>0}$ is a parameter that determines the frequency of the sinusoid, in Hz. We set $\nu_1 = 6.25$ Hz, corresponding to the same 160 ms temporal period as the drifting grating stimuli considered in Section 5.1. We set the relative delay between areas $D_1 = +30$ ms (area A leads area B). To isolate the effects of this temporal structure on the estimation of the latent time courses and GP covariance functions, we set the signal-to-noise ratio to be high, $\mathrm{tr}(C_m C_m^\top)/\mathrm{tr}(R_m) = 10.0$.

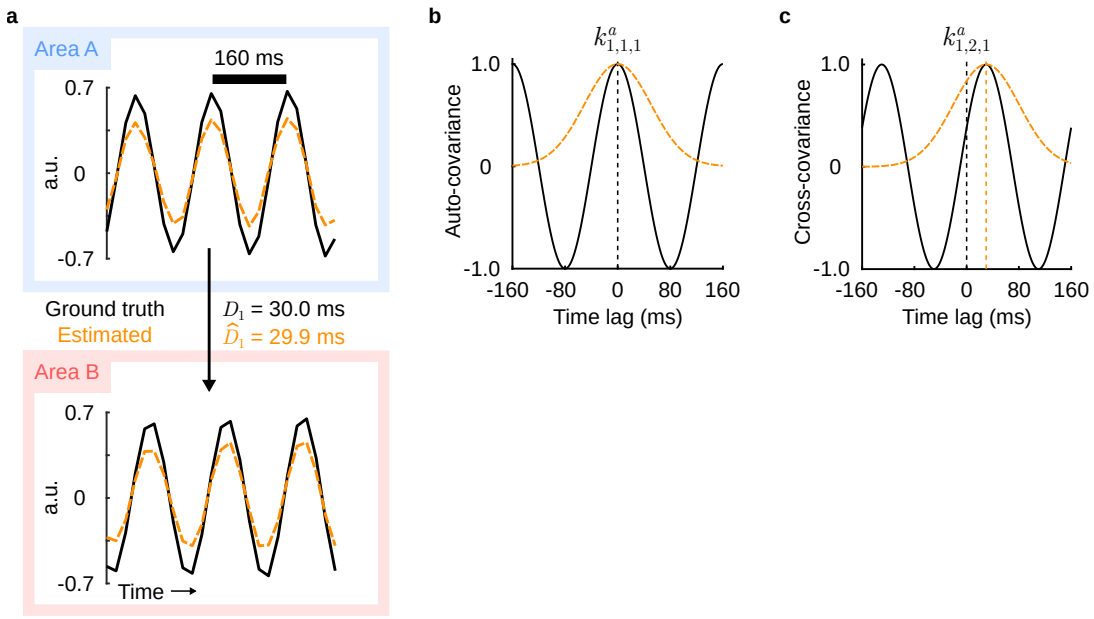

Figure B.1: Estimating a sinusoidal covariance function with a squared exponential function. (**a**) Latent-variable time course estimates for a representative trial. Top row / blue box: area A; bottom row / red box: area B. Orange dashed traces: DLAG estimates; black solid traces: ground truth. a.u.: arbitrary units. (**b**) Gaussian process auto-covariance functions. Orange dashed traces: DLAG estimates; black solid traces: ground truth. Black dashed vertical line indicates zero-lag. (**c**) Gaussian process cross-covariance functions. Orange dashed vertical line indicates estimated time delay parameter. All other conventions as in (b).

We then fit a DLAG model with squared exponential covariance (equation (3.7)) to this simulated neural activity. DLAG estimates of latent time courses accurately reflected the underlying sinusoidal temporal structure (Fig. B.1a, orange dashed traces). The estimated time delay was also accurate (delay estimate: 29.9 ms; ground truth: 30.0 ms). However, unsurprisingly, the estimated GP auto- and cross-covariances did not accurately reflect this sinusoidal structure (Fig. B.1b, auto-covariance; Fig. B.1c:, cross-covariance). The estimated squared exponential functions were relatively wide (compared to the curvature of the cosine functions; timescale estimate: 50.5 ms) in an attempt to capture the (periodically negative) correlation induced at long time lags by the periodic sinusoidal activity.

B.2 A latent variable that reflects a bidirectional interaction

We next generated simulated neural activity with the same characteristics as described in Section B.1, except we modified the temporal structure of the across-area variable so that it comprised the sum of two interactions: one in which area A leads area B, and another in which area B leads area A. To produce this bidirectional interaction structure, we modified the across-area Gaussian process covariance function

(equation (3.7)) as follows:

$$k^a_{m_1,m_2,j}(t_1,t_2) = \frac{1}{2}\exp\left(-\frac{(\Delta t^+)^2}{2(\tau^a_j)^2}\right) + \frac{1}{2}\exp\left(-\frac{(\Delta t^-)^2}{2(\tau^a_j)^2}\right) \tag{B.3}$$

$$\Delta t^+ = (t_2 - t_1) - D^+_j \tag{B.4}$$

$$\Delta t^- = (t_2 - t_1) - D^-_j \tag{B.5}$$

where $D^+_j \in \mathbb{R}_{>0}$ is a relative time delay that reflects the interaction in which area A leads area B, and $D^-_j \in \mathbb{R}_{<0}$ is a relative time delay that reflects the interaction in which area B leads area A. Here we chose $D^+_1 = +40$ ms, $D^-_1 = -40$ ms, and $\tau^a_1 = 25$ ms, which resulted in a bimodal GP cross-covariance function (Fig. B.2c, black trace).

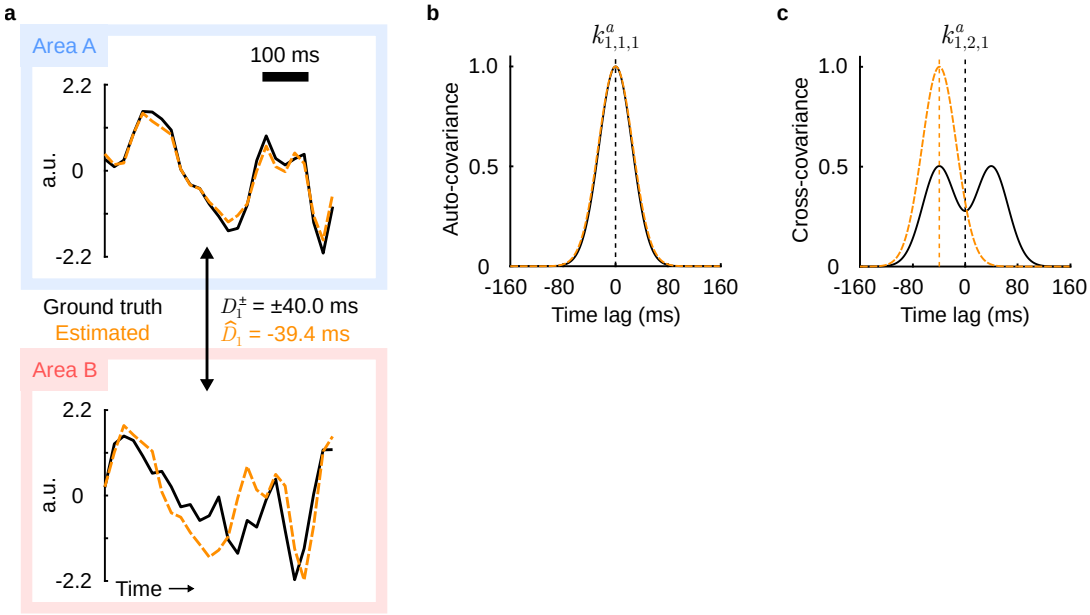

Figure B.2: Estimating a bimodal covariance function with a squared exponential function (high SNR). (**a**) Latent-variable time course estimates for a representative trial. (**b**) Gaussian process auto-covariance functions. (**c**) Gaussian process cross-covariance functions. Same conventions as in Fig. B.1.

We then fit a DLAG model with squared exponential covariance (equation (3.7)) to this simulated neural activity. Interestingly, while estimates of the latent time courses (Fig. B.2a, orange dashed traces) and GP auto-covariance (Fig. B.2b, orange dashed trace) were reasonably accurate, estimates of the GP cross-covariance and time delay appeared to capture primarily one direction of interaction (Fig. B.2c, orange dashed trace; delay estimate: -39.4 ms; timescale estimate: 26.1 ms). Note that different random initializations of the DLAG fitting procedure could lead to estimates that focused on the other direction of interaction (not shown).

This behavior reflects a compromise by a DLAG model that can only describe unimodal GP cross-

covariances. In principle, the fitted model could have captured more of the positive time lag interactions (the right half of Fig. B.2c) by employing a larger GP timescale, hence widening both the GP auto- and cross-covariance functions. However, this choice would result in an overly wide GP auto-covariance function.

To see if we could induce this alternative type of solution, we re-performed the above analysis; however, we significantly lowered the signal-to-noise ratio of the simulated activity from 10.0 to 0.1. Indeed, the DLAG model fit to these data produced estimates that attempted to balance capturing the interactions in both directions (Fig. B.3). The estimated GP cross-covariance function was centered close to zero and was wide enough to capture both positive and negative time lag interactions (Fig. B.3c, orange dashed trace; delay estimate: -5.8 ms; timescale estimate: 43.0 ms). Consequently, the estimated GP auto-covariance function was overly wide (Fig. B.3b, orange dashed trace), leading to overly smooth latent time courses (Fig. B.3a, orange dashed traces).

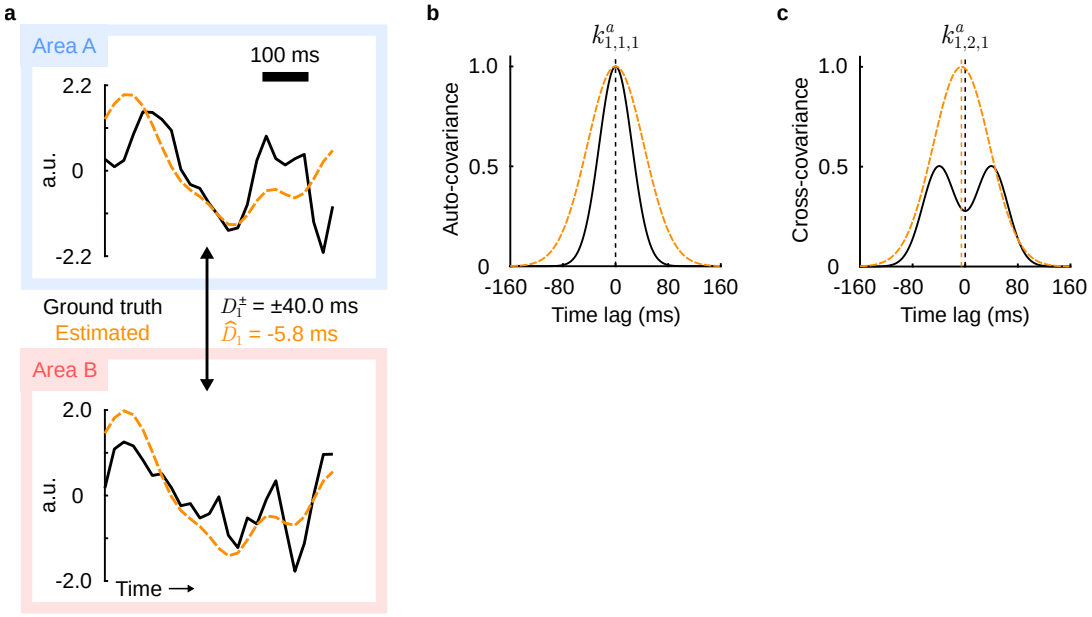

Figure B.3: Estimating a bimodal covariance function with a squared exponential function (low SNR). (**a**) Latent-variable time course estimates for a representative trial. (**b**) Gaussian process auto-covariance functions. (**c**) Gaussian process cross-covariance functions. Same conventions as in Fig. B.1.

In conclusion, we demonstrated here the effects of attempting to fit DLAG models with squared exponential GP covariances to neural activity with temporal structure that significantly deviates from that assumption. These DLAG models misestimated the underlying latent time courses and GP covariances in predictable ways. But importantly, these investigations demonstrate two strengths of the DLAG framework: (1) The explicit assumptions provided by the chosen GP covariance function allow for principled

tests of where and how DLAG's estimates succeed and fail. (2) The GP covariance function is a modular component of the DLAG model. If we desired to more accurately capture the various temporal structures considered here, we need only modify the GP covariance function according to equations (B.1) or (B.3). Many more GP covariance functions could be feasibly employed[55].