Fast Multigroup Gaussian Process Factor Models

Evren Gokcen
egokcen@cmu.edu
Department of Electrical and Computer Engineering, Carnegie Mellon University,
Pittsburgh, PA 15213, USA

Anna I. Jasper
anna.jasper@einsteinmed.edu
Dominick Purpura Department of Neuroscience, Albert Einstein College of Medicine,
Bronx, NY 10461, USA

Adam Kohn
adam.kohn@einsteinmed.edu
Dominick Purpura Department of Neuroscience, Department of Ophthalmology and
Visual Sciences, and Department of Systems and Computational Biology, Albert
Einstein College of Medicine, Bronx, NY 10461, USA

Christian K. Machens
christian.machens@neuro.fchampalimaud.org
Champalimaud Neuroscience Programme, Champalimaud Foundation, 1400-038
Lisbon, Portugal

Byron M. Yu
byronyu@cmu.edu
Department of Electrical and Computer Engineering and Department of Biomedical
Engineering, Carnegie Mellon University, Pittsburgh, PA 15213, USA

Gaussian processes are now commonly used in dimensionality reduction approaches tailored to neuroscience, especially to describe changes in high-dimensional neural activity over time. As recording capabilities expand to include neuronal populations across multiple brain areas, cortical layers, and cell types, interest in extending gaussian process factor models to characterize multipopulation interactions has grown. However, the cubic runtime scaling of current methods with the length of experimental trials and the number of recorded populations (groups) precludes their application to large-scale multipopulation recordings. Here, we improve this scaling from cubic to linear in both trial length

and group number. We present two approximate approaches to fitting multigroup gaussian process factor models based on inducing variables and the frequency domain. Empirically, both methods achieved orders of magnitude speed-up with minimal impact on statistical performance, in simulation and on neural recordings of hundreds of neurons across three brain areas. The frequency domain approach, in particular, consistently provided the greatest runtime benefits with the fewest trade-offs in statistical performance. We further characterize the estimation biases introduced by the frequency domain approach and demonstrate effective strategies to mitigate them. This work enables a powerful class of analysis techniques to keep pace with the growing scale of multipopulation recordings, opening new avenues for exploring brain function.

1 Introduction

With the proliferation of high-dimensional neuronal population recordings, dimensionality reduction has become a widely used class of multivariate statistical techniques (Cunningham & Yu, 2014). Increasingly, these recordings capture not only many neurons but many distinct neuronal populations, spanning brain areas, cortical layers, and cell types (Ahrens et al., 2012; Steinmetz et al., 2021; Yang & Yuste, 2017). Interest has grown, therefore, in dimensionality reduction methods capable of characterizing the interactions among these recorded populations (Kang & Druckmann, 2020; Keeley, Zoltowski, Aoi et al., 2020; Machado et al., 2022; Semedo et al., 2020).

Recent development efforts have focused, in particular, on methods that characterize multipopulation, or multigroup, interactions over time. These approaches incorporate latent variables, or factors, that follow a time series model, based typically on a dynamical system (state-space model; Bong et al., 2020; Glaser et al., 2020; Karniol-Tambour et al., 2024; Semedo et al., 2014) or a gaussian process (GP; Balzani et al., 2023; Gokcen et al., 2022; Gondur et al., 2024; Keeley, Aoi et al., 2020; Li, Li et al., 2024). GP-based approaches can be especially useful for exploratory analyses of neural recordings, where an appropriate parametric dynamical model is unknown a priori.

The delayed latents across groups (DLAG) framework (Gokcen et al., 2022) is a representative member of this class of multigroup GP factor models. DLAG leverages latent variables with time delays to disentangle the bidirectional, concurrent interactions between two neuronal populations. DLAG's recent extension to more than two populations, mDLAG (Gokcen et al., 2023), determines (1) the subset of populations described by each latent dimension, (2) the direction of signal flow among those populations, and (3) how those signals evolve over time within and across experimental trials. While multigroup GP factor models like mDLAG could advance the study of concurrent signaling throughout the brain, they are ultimately

limited by computational scalability. The multi-output GP kernels (Álvarez et al., 2012) at their core lead to runtimes that scale cubically in the number of time points per experimental trial and in the number of analyzed neuronal populations, or groups.

Addressing this computational challenge is thus critical for these statistical methods to remain a practical tool for analyzing recordings that already span dozens of brain areas (Steinmetz et al., 2019) and continue to grow in scale. Here, we develop two approximate methods to accelerate the fitting of multigroup GP factor models (specifically, mDLAG) based on inducing variables and the frequency domain. To our knowledge, these methods are the first for this class of models to achieve linear scaling in both trial length and group number. This work thus enables a broad class of dimensionality reduction techniques to keep pace with the rapidly growing scale of multi-population neural recordings.

We begin with a synthesis of the diverse array of existing approaches to accelerating GP inference and estimation (section 2). Then we provide a mathematical overview of the core methods considered and developed in this work: the baseline mDLAG formulation (section 3.1) and the two accelerated methods, mDLAG with inducing variables (section 3.2) and mDLAG in the frequency domain (section 3.3). We demonstrate empirically, in simulation and on neural recordings of hundreds of neurons across three brain areas, that both accelerated methods achieve orders of magnitude speed-up over baseline with minimal impact on statistical performance (sections 4.1–4.4). The frequency domain approach, in particular, consistently provided the greatest runtime benefits with the fewest trade-offs in statistical performance. We further characterize the estimation biases introduced by the frequency domain approach and demonstrate straightforward but effective strategies to mitigate those biases (section 4.5).

2 Related Work: Fast Methods for Gaussian Processes

The computational challenges associated with GP posterior inference and (hyper)parameter estimation are particularly pronounced when GPs are used in latent variable models. Model fitting is iterative, and GP inference steps therefore require hundreds to thousands of reevaluations. In turn, the potential runtime benefits are that much more pronounced for methods that accelerate those computationally intensive steps.

Methods for fast GP inference have an extensive history, spanning several decades and disciplines. Much of that work has origins in GP regression (Rasmussen & Williams, 2006), without any appeal to the class of multigroup latent variable models described above. In sections 2.1 to 2.4, we review a selection of approaches relevant to the scope of this work: multi-output GPs for time series (i.e., GPs with one input dimension), as an ingredient in latent variable models.

2.1 Numerical Methods That Exploit Matrix Structure. In specific applications of single-group latent variable models (and single-output GPs), the cubic scaling in the number of time points, T, can be alleviated by exploiting matrix structure. Stationary GP kernels, for instance (a standard modeling choice), give rise to a GP covariance matrix and its derivatives (the source of computational bottlenecks in GP inference and parameter estimation) with Toeplitz structure. Certain computations involving Toeplitz matrices (matrix inversion, matrix products) can be implemented in $\mathcal{O}(T^2)$ operations (Golub & Van Loan, 2013) and even $\mathcal{O}(T)$ storage (Zhang et al., 2005). The fast Fourier transform (FFT) can also be leveraged to reduce Toeplitz matrix products to $\mathcal{O}(T \log T)$ operations (Silverman, 1982). Many GP-based methods thus seek out advantageous matrix structure (e.g., Toeplitz, Kronecker), through careful model structural choices or a series of approximations during model fitting (Aoi & Pillow, 2017; Cunningham et al., 2008; Foreman-Mackey et al., 2017; Jensen et al., 2021). These bespoke approaches are not as broadly applicable as the approaches that follow, particularly in the multigroup context.

2.2 State-Space Representations. The first more general class of approaches to GP acceleration exploits the deep mathematical relationship between GPs and state-space models (i.e., linear dynamical systems). Any well-behaved stationary GP time series can be approximated by an appropriate linear-gaussian state-space model with a sufficiently expanded state dimension, p (Loper et al., 2021). Many paths can be taken to parameterize a GP as a state-space model. Approaches that rely on spectral factorization are particularly well known (Sayed & Kailath, 2001; Hartikainen & Särkkä, 2010) and have also been extended to the multi-output case (Li, Li et al., 2024). Alternative parameterizations can be obtained via the latent exponentially generated (LEG) family of state-space models (Loper et al., 2021) or regression between the state-space model parameters and the GP covariance function (kernel) itself (Li, Wang et al., 2024). Li, Li et al. (2024) and Li, Wang et al. (2024) have applied this state-space representation directly to DLAG. In a slightly different approach, Dowling et al. (2023) arrive at an arbitrarily expressive state-space representation by leveraging the Hida-Matérn class of GP covariance functions.

Regardless of the path one takes to a state-space formulation of a GP, the goal is the same: inference of the GP posterior can be carried out via dynamic programming, most commonly Kalman filtering (Kalman & Bucy, 1961) and Rauch–Tung–Striebel smoothing (Rauch et al., 1965) algorithms or information filtering (Dowling et al., 2023). The complexity of posterior inference is then reduced from $\mathcal{O}(T^3)$ operations and $\mathcal{O}(T^2)$ storage to $\mathcal{O}(p^3 T)$ operations and $\mathcal{O}(p^2 T)$ storage: linear in the number of time points per trial, T, at the added cost of cubic scaling in the number of expanded states in the state-space approximation, p. In practice, however, it appears that common GP covariance functions are well approximated by only a

handful of terms, p (Hartikainen & Särkkä, 2010; Li, Wang et al., 2024; Loper et al., 2021; Solin & Särkkä, 2014), and hence the added cost is effectively a modest constant factor. Additional approximations can ameliorate the cubic scaling in the number of state dimensions to quadratic, in both the single-output GP (Solin et al., 2018) and multi-output GP cases (Lim et al., 2021). As an ingredient in multigroup GP factor models, however, the state-space approach still leads to cubic scaling in the number of groups M—$\mathcal{O}(p^3 M^3 T)$ operations (Li, Wang et al., 2024)—where we would like to achieve linear scaling if possible.

2.3 Sparsity via Inducing Variables. The second general class of approaches relies on sparse approximations, specifically by inducing variables (Quiñonero-Candela & Rasmussen, 2005). Inducing variable methods leverage a straightforward insight: for sufficiently smooth signals, perhaps only some smaller number of samples, $T_{\text{ind}} < T$, are needed to carry out accurate inference. In a useful mathematical abstraction, these T_{ind} inducing variables need not belong to the set of T original samples, but can instead be defined at arbitrary inducing points or "pseudo-inputs" (Snelson & Ghahramani, 2005) in the same domain as that of the original samples (time, for time series data).

These approaches pass the computational burden to the smaller set of inducing points: $\mathcal{O}(T T_{\text{ind}}^2)$ operations and $\mathcal{O}(T T_{\text{ind}})$ storage—linear in the original number of samples (see also Hensman et al., 2013, for a formulation that achieves $\mathcal{O}(T_{\text{ind}}^3)$ operations). Titsias (2009) developed a variational formulation of inducing variables that facilitated their use in extended versions of the single-group dimensionality reduction method gaussian process factor analysis (GPFA; Duncker & Sahani, 2018; Yu et al., 2009). Álvarez et al. (2010) extended this approach to multi-output GP regression, suggesting that as an ingredient in multigroup GP factor models, inducing variables could produce $\mathcal{O}(M T T_{\text{ind}}^2)$ scaling—linear in both the number of time points T and the number of groups M. To date, the efficacy of assimilating inducing variables into multigroup GP factor models has not been explored. Here, we develop an mDLAG model with inducing variables (mDLAG-inducing) that achieves the desired linear scaling properties.

2.4 Frequency Domain Representations. The final class of approaches invokes the frequency domain (Fourier basis) representation of stationary GPs (these ideas can be combined with inducing variable approaches with some success; see, for example, Hensman et al., 2018). In the large T limit, any well-behaved stationary GP can be represented as a spectral process with independent frequency components (Kolmogorov et al., 1941; Priestley, 1981). In practical terms, this theoretical result suggests that if we work with a frequency domain representation of observed time series, we can perform approximate GP inference and parameter estimation with a diagonal GP covariance matrix rather than the dense GP covariance matrix in

the time domain (see section 3.3 for details). Computations involving this diagonal matrix could thus scale linearly in the number of time points, T, or at least $\mathcal{O}(T \log T)$ if a FFT of the data need be calculated first. In spectral estimation, this approach dates back at least to Whittle (1951). The Whittle likelihood (a "quasi-likelihood") is a biased approximation to gaussian likelihoods of time domain signals, computed efficiently from the periodogram of the data (Rao & Yang, 2021; Sykulski et al., 2019).

More recently, frequency domain GP representations have been leveraged for runtime benefits in several latent variable modeling applications. Applications include Bayesian smoothing for spatial modeling (Paciorek, 2007), modeling of natural sounds (Turner, 2010), and certain single-group GP factor models for neuroscience. For example, Keeley, Aoi et al. (2020) proposed a frequency domain quasi-likelihood for a GPFA model with Poisson observation noise, enabling $\mathcal{O}(T \log T)$ scaling. Similar ideas have been incorporated into multimodal GP variational autoencoders (Gondur et al., 2024). Dowling et al. (2023) explicitly incorporated the Whittle likelihood into GPFA-related models as a subroutine of fitting, to accelerate GP hyperpameter updates (like GP inference, naively $\mathcal{O}(T^3)$ operations) to $\mathcal{O}(T \log T)$ operations.

These studies were limited to GP factor models that employed single-output GPs. The cross-spectral factor analysis method of Gallagher et al. (2017) employed multi-output GPs to describe interactions across univariate recording channels (analogously, single neurons). Drawing directly from Ulrich et al. (2015), Gallagher et al. (2017) employed a frequency domain quasi-likelihood that produced linear scaling in T but still maintained cubic scaling in the number of recording channels (analogously, $\mathcal{O}(M^3 T)$ operations for M neuronal populations or groups). Here, we develop an approximate formulation of mDLAG (mDLAG-frequency) fully in the frequency domain, from the generative model to posterior inference and fitting. By performing posterior inference and fitting completely in the frequency domain, the approach effectively achieves linear scaling in both the number of time points per trial and the number of groups, $\mathcal{O}(MT)$.

3 Model and Algorithmic Overviews

3.1 Delayed Latents across Multiple Groups (mDLAG-Time). We start with an overview of the mDLAG model and fitting procedure from Gokcen et al. (2023), which will serve as the baseline for the other two approaches developed in sections 3.2 and 3.3. These sections constitute the core contributions of this work. Since the baseline mDLAG approach relies fully on the time domain (without invoking inducing variables), we refer to it as mDLAG-time, to disambiguate it from those that follow. Note that we explicitly define all variables and parameters as they appear, but for reference, we include an explanation of notation and a glossary of common variables and parameters in Tables 2 to 5.

In brief, analyzing multipopulation neural recordings presents two key challenges: identifying network-level interactions and disentangling concurrent signal flow. These interactions are difficult to unravel from raw neural activity but can be pinpointed along certain latent dimensions. mDLAG leverages that insight and addresses each of the two challenges with a dedicated model component. For the first, mDLAG employs automatic relevance determination (ARD) to promote group-wise sparsity for each estimated latent dimension and identify the groups involved in each interaction (see section 3.1.1). For the second, mDLAG recognizes that communication between neuronal populations is not instantaneous and estimates for each latent variable a set of time delays that describes the signal flow across the involved groups (see section 3.1.2).

3.1.1 Observation Model and Automatic Relevance Determination.
For group m (comprising q_m units) at time t on trial n, we define a linear relationship between observed activity, $\mathbf{y}_{n,t}^m \in \mathbb{R}^{q_m}$, and latent variables (latents), $\mathbf{x}_{n,t}^m \in \mathbb{R}^p$ (see Figure 1A):

$$\mathbf{y}_{n,t}^m = C^m \mathbf{x}_{n,t}^m + \mathbf{d}^m + \boldsymbol{\varepsilon}^m, \tag{3.1}$$

$$\boldsymbol{\varepsilon}^m \sim \mathcal{N}(\mathbf{0}, (\Phi^m)^{-1}), \tag{3.2}$$

where the loading matrix $C^m \in \mathbb{R}^{q_m \times p}$ and mean parameter $\mathbf{d}^m \in \mathbb{R}^{q_m}$ are model parameters. The vector $\boldsymbol{\varepsilon}^m$ is a zero-mean gaussian random variable with noise precision matrix $\Phi^m \in \mathbb{S}^{q_m \times q_m}$ ($\mathbb{S}^{q_m \times q_m}$ is the set of $q_m \times q_m$ symmetric matrices). We 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 unit. This constraint encourages the latents to explain as much of the shared variance among units as possible.

As we will describe, at time point t, latents $\mathbf{x}_{n,t}^m$, $m = 1, \ldots, M$ are coupled across groups, and thus each group has the same number of latents, p. Because we seek a low-dimensional description of observed activity, the number of latents is fewer than the total number of units, that is, $p < q$, where $q = \sum_m q_m$.

mDLAG seeks to identify multiple network-level interactions across the observed groups. To do so requires estimating the number of latents across all groups and which subset of groups each latent involves. This estimation problem needs to scale tractably with the number of groups. mDLAG therefore takes a Bayesian approach to the problem, where $\mathbf{d}^m$, Φ^m, and C^m are taken to be probabilistic parameters with prior distributions.

The parameter $\mathbf{d}^m$ describes the mean activity of each unit over time and trials. We set a gaussian prior over $\mathbf{d}^m$:

$$P(\mathbf{d}^m) = \mathcal{N}(\mathbf{d}^m \mid \mathbf{0}, \beta^{-1} I_{q_m}) \tag{3.3}$$
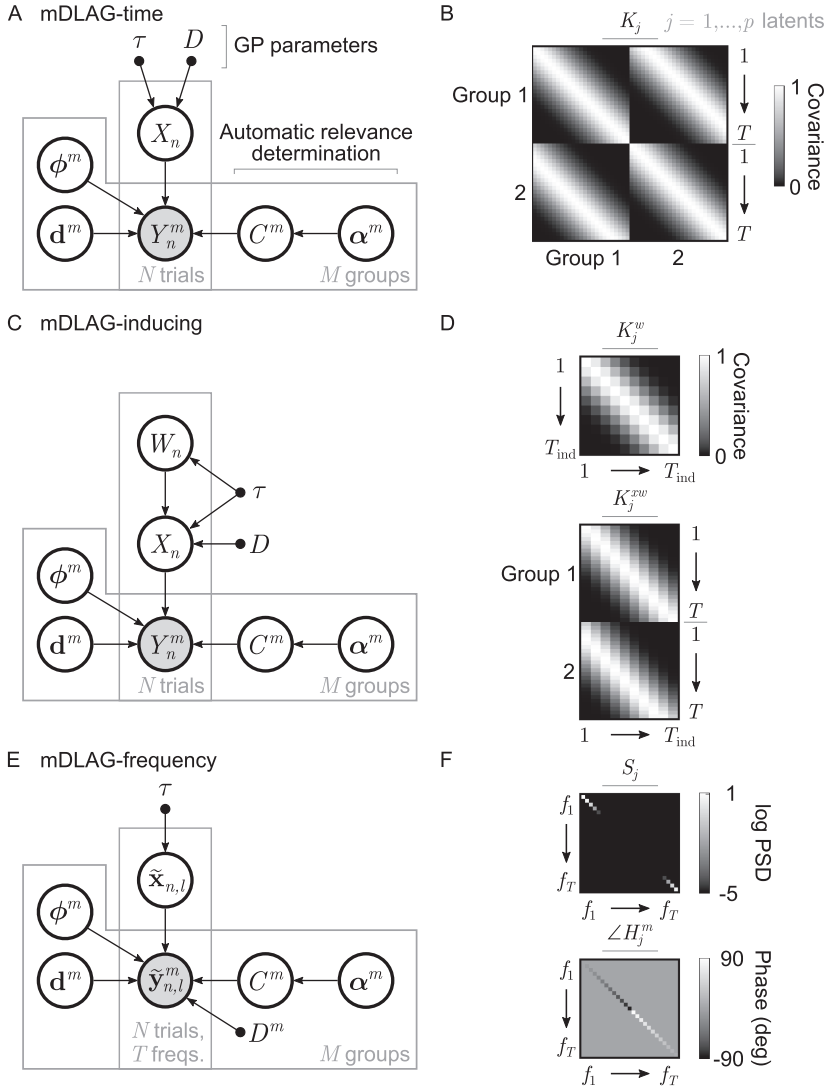
Figure 1: Summary of generative models and covariance structures. (A) Delayed latents across multiple groups (mDLAG) in the time domain (mDLAG-time; Gokcen et al., 2023). Filled circles represent observed variables. Unfilled circles represent probabilistic latent variables and parameters. Black dots represent deterministic parameters. Arrows indicate conditional dependence relationships between variables. Boxes indicate repetition of the enclosed variables or parameters over a particular index ($n = 1, \ldots, N$ trials or $m = 1, \ldots, M$ groups), where those repetitions are mutually independent. (B) An example GP covariance matrix (K_j, for a latent $j = 1, \ldots, p$),

where $\beta \in \mathbb{R}_{>0}$ is a hyperparameter and I_{q_m} is the $q_m \times q_m$ identity matrix. We set the conjugate gamma prior over each ϕ_r^m, for each unit $r = 1, \ldots, q_m$:

$$P(\phi_r^m) = \Gamma(\phi_r^m \mid a_\phi, b_\phi), \tag{3.4}$$

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

The loading matrix C^m linearly combines latents and maps them to observed neural activity. In particular, the jth column of C^m, $\mathbf{c}_j^m \in \mathbb{R}^{q_m}$, maps the jth latent $x_{n,j,t}^m$ to group m. To determine which subset of groups is described by each latent, mDLAG employs ARD (MacKay, 1994; Neal, 1995). Specifically, we define the following prior over the columns of each C^m (Klami et al., 2015):

$$P(\mathbf{c}_j^m \mid \alpha_j^m) = \mathcal{N}(\mathbf{c}_j^m \mid \mathbf{0}, (\alpha_j^m)^{-1} I_{q_m}), \tag{3.5}$$

$$P(\alpha_j^m) = \Gamma(\alpha_j^m \mid a_\alpha, b_\alpha), \tag{3.6}$$

where $\alpha_j^m \in \mathbb{R}_{>0}$ is the ARD parameter for latent j and group m, and $a_\alpha, b_\alpha \in \mathbb{R}_{>0}$ are hyperparameters. As α_j^m becomes large, the magnitude of $\mathbf{c}_j^m$ becomes concentrated around 0, and hence the jth latent $x_{n,j,t}^m$ will have a vanishing influence on group m. The ARD prior encourages group-wise sparsity for each latent during model fitting, where the loading matrix coefficients will be pushed toward zero for latents that explain an insignificant amount of shared variance within a group and remain nonzero otherwise.

---

corresponding to the mDLAG-time model. The matrix K_j was generated with squared exponential GP timescale $\tau = 100$ ms and time delay $D = 80$ ms, for trial length $T = 25$ time points (with 20 ms sampling period) and $M = 2$ groups. (C) mDLAG with inducing variables (mDLAG-inducing). Same conventions as in panel A. (D) Top: An example inducing variable covariance matrix (K_j^w). Bottom: Corresponding covariance matrix between latent j and its inducing variable (K_j^{xw}). Both K_j^w and K_j^{xw} were generated with the same GP parameters as K_j in panel A, with $T_{\text{ind}} = 10$ inducing points. (E) mDLAG in the frequency domain (mDLAG-frequency). Same conventions as in panel A. Note the box indicating independence across not only trials but also $l = 1, \ldots, T$ frequency components (freqs.). (F) Top: An example (diagonal) GP power spectral density (PSD) matrix (S_j, displayed on a log scale). Bottom: An example (diagonal) phase-shift matrix from latent j to an observed group m (H_j^m, displayed in degrees, $\angle H_j^m$). The PSD matrix S_j was generated with GP timescale $\tau = 100$ ms, for trial length $T = 25$ time points. The phase-shift matrix H_j^m was generated with time delay $D_j^m = 10$ ms. Frequency $f_1 = 0$ is the zero frequency, frequencies f_2 through f_{13} are increasing positive frequencies, and frequencies f_{14} through f_{25} (f_T) are negative frequencies increasing toward zero (see section 3.3.1).

Intuitively, dimensions that appear in two or more groups indicate the presence of correlated activity across those groups.

3.1.2 Gaussian Process State Model. For each latent, mDLAG seeks to characterize the direction of signal flow among the involved groups (determined by ARD) and how those signals evolve over time within and across trials. We therefore employ GPs (Rasmussen & Williams, 2006) and define a GP over all time points $t = 1, \ldots, T$ for each latent $j = 1, \ldots, p$ as follows (see Figure 1B):

$$
\begin{bmatrix} \mathbf{x}_{n,j,:}^1 \\ \vdots \\ \mathbf{x}_{n,j,:}^M \end{bmatrix} \sim \mathcal{N} \left(\mathbf{0}, \begin{bmatrix} K_{1,1,j} & \cdots & K_{1,M,j} \\ \vdots & \ddots & \vdots \\ K_{M,1,j} & \cdots & K_{M,M,j} \end{bmatrix} \right).
\tag{3.7}
$$

Under equation 3.7, latents are independent and identically distributed across trials. The diagonal blocks $K_{1,1,j} = \cdots = K_{M,M,j} \in \mathbb{S}^{T \times T}$ describe the autocovariance of each latent, and each T-by-T off-diagonal block describes the cross-covariance that couples two groups.

To define these matrices, we introduce additional notation. Specifically, we indicate groups 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 $\mathbf{x}_{n,j,:}^{m_1} \in \mathbb{R}^T$ in group m_1 and latent $\mathbf{x}_{n,j,:}^{m_2} \in \mathbb{R}^T$ in group m_2 on trial n. mDLAG is immediately compatible with any stationary covariance function. Here, we explore in depth the commonly used squared exponential function for GP covariances. Therefore, element (t_1, t_2) of each $K_{m_1,m_2,j}$ can be computed as follows (Gokcen et al., 2022; Lakshmanan et al., 2015):

$$
k_{m_1,m_2,j}(t_1, t_2) = \left(1 - \sigma_j^2\right) \exp\left(-\frac{(\Delta t)^2}{2\tau_j^2}\right) + \sigma_j^2 \cdot \delta_{\Delta t},
\tag{3.8}
$$

$$
\Delta t = \left(t_2 - D_j^{m_2}\right) - \left(t_1 - D_j^{m_1}\right),
\tag{3.9}
$$

where the characteristic timescale, $\tau_j \in \mathbb{R}_{>0}$ is a deterministic model parameter to be estimated from observed activity. The GP noise variance, $\sigma_j^2 \in (0, 1)$, is conventionally fixed to a small value (10^{-3}). $\delta_{\Delta t}$ is the Kronecker delta, which is 1 for $\Delta t = 0$ and 0 otherwise. 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 the latents and loading matrices C^m.

Two parameters, the time delay to group m_1, $D_j^{m_1} \in \mathbb{R}$, and the time delay to group m_2, $D_j^{m_2} \in \mathbb{R}$, are key to describing signal flow across groups. First notice that when computing the autocovariance for group m (i.e., when $m_1 = m_2 = m$; see Figure 1B, diagonal blocks of K_j), the time delay

parameters $D_j^{m_1}$ and $D_j^{m_2}$ are equal, and so Δt (see equation 3.9) reduces simply to the time difference $(t_2 - t_1)$. Time delays are therefore only relevant when computing the cross-covariance between distinct groups m_1 and m_2. The time delay to group m_1, $D_j^{m_1}$, and the time delay to group m_2, $D_j^{m_2}$, by themselves have no physically meaningful interpretation. Their difference, $D_j^{m_2} - D_j^{m_1}$, however, represents a well-defined, continuous-valued time delay from group m_1 to group m_2 (see Figure 1B, off-diagonal blocks of K_j). The sign of the relative time delay indicates the directionality of the lead-lag relationship between groups captured by latent j (positive: group m_1 leads group m_2; negative: group m_2 leads group m_1), which we interpret as a description of signal flow. Note that time delays need not be integer multiples of the sampling period or spike count bin width of recorded neural activity. Without loss of generality, we designate group $m = 1$ as the reference area and fix the delays for group 1 at 0 (i.e., $D_j^1 = 0$ for all latents $j = 1, \ldots, p$).

3.1.3 Posterior Inference, Fitting, and Computational Scaling. Let Y and X be collections of all observed and latent variables, respectively, across all time points and trials. Similarly, let $\mathbf{d}$, ϕ, C, $\mathcal{A}$, τ, and D be collections of the mean parameters, noise precisions, loading matrices, ARD parameters, GP timescales, and time delays, respectively. From the observed activity, we seek to estimate posterior distributions over the probabilistic model components $\theta = \{X, \mathbf{d}, \phi, C, \mathcal{A}\}$ and point estimates of the deterministic GP parameters $\Omega = \{\tau, D\}$.

We do so by employing a variational inference scheme (Bishop, 1999; Klami et al., 2015) in which we maximize the evidence lower bound (ELBO), $L(Q, \Omega)$, with respect to the approximate posterior distribution $Q(\theta)$ and the deterministic parameters Ω, where

$$\log P(Y) \geq L(Q, \Omega) = \mathbb{E}_Q[\log P(Y, \theta|\Omega)] - \mathbb{E}_Q[\log Q(\theta)]. \quad (3.10)$$

We constrain $Q(\theta)$ so that it factorizes over the elements of θ:

$$Q(\theta) = Q_x(X)Q_d(\mathbf{d})Q_\phi(\phi)Q_c(C)Q_\mathcal{A}(\mathcal{A}). \quad (3.11)$$

This factorization enables closed-form updates during optimization. The ELBO can then be iteratively maximized via coordinate ascent of the factors of $Q(\theta)$ and the deterministic parameters Ω. Here all hyperparameters were fixed to a very small value, $\beta, a_\phi, b_\phi, a_\alpha, b_\alpha = 10^{-12}$, to produce noninformative priors (Klami et al., 2015). Throughout this work, we take estimates of the latent variables and model parameters to be the corresponding means of the posterior distributions comprising equation 3.11.

The computational bottlenecks in mDLAG-time comprise three stages of the fitting procedure: (1) the update of the posterior distribution over

latents, $Q_x(X)$, (2) the updates of the GP parameters, Ω, via gradient ascent, and (3) evaluation of the ELBO, equation 3.10. From equation 3.7, let

$$K_j = \begin{bmatrix} K_{1,1,j} & \cdots & K_{1,M,j} \\ \vdots & \ddots & \vdots \\ K_{M,1,j} & \cdots & K_{M,M,j} \end{bmatrix} \in \mathbb{S}^{MT \times MT}. \tag{3.12}$$

Each of the three stages above requires the inversion of the $MT \times MT$ covariance matrix K_j and therefore require at least $\mathcal{O}(M^3 T^3)$ operations and $\mathcal{O}(M^2 T^2)$ storage. The update of $Q_x(X)$, in fact, requires the inversion of a larger $pMT \times pMT$ matrix (see below). The iterative nature of gradient ascent also requires potentially several reevaluations of K_j^{-1} within a single fitting iteration.

A complete set of update equations, including for the three stages above, can be found in Gokcen et al. (2023). For the sake of concision, we highlight just the update of the posterior distribution over latents, $Q_x(X)$. Let us first define several variables. Construct $\mathbf{y}_{n,t} = [\mathbf{y}_{n,t}^{1\top} \cdots \mathbf{y}_{n,t}^{M\top}]^\top \in \mathbb{R}^q$, $q = \sum_m q_m$, by vertically concatenating the observed activity of groups $m = 1, \ldots, M$ 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 observed activity $\mathbf{y}_{n,t}$ across all time points $t = 1, \ldots, T$. For latents, define $\mathbf{x}_{n,t} = [\mathbf{x}_{n,:,t}^{1\top} \cdots \mathbf{x}_{n,:,t}^{M\top}]^\top \in \mathbb{R}^{pM}$ by vertically concatenating the p latents of each group at time t on trial n. Then we vertically concatenate the latents $\mathbf{x}_{n,t}$ across all time points $t = 1, \ldots, T$ to give $\bar{\mathbf{x}}_n = [\mathbf{x}_{n,1}^\top \cdots \mathbf{x}_{n,T}^\top]^\top \in \mathbb{R}^{pMT}$. Finally, we collect the parameters C^m, Φ^m, and $\mathbf{d}^m$ across populations $m = 1, \ldots, M$ by defining $C = \text{diag}(C^1, \ldots, C^M) \in \mathbb{R}^{q \times pM}$, $\Phi = \text{diag}(\Phi^1, \ldots, \Phi^m) \in \mathbb{S}^{q \times q}$, and $\mathbf{d} = [\mathbf{d}^{1\top} \cdots \mathbf{d}^{M\top}]^\top \in \mathbb{R}^q$.

The update to the posterior distribution over latents, $Q_x(X)$, takes the same functional form as the prior distribution (see equation 3.7), a gaussian distribution,

$$Q_x(X) = \prod_{n=1}^N \mathcal{N}(\bar{\mathbf{x}}_n \mid \bar{\boldsymbol{\mu}}_{x_n}, \bar{\Sigma}_x), \tag{3.13}$$

with trial-dependent posterior mean $\bar{\boldsymbol{\mu}}_{x_n} \in \mathbb{R}^{pMT}$ and trial-independent posterior covariance $\bar{\Sigma}_x \in \mathbb{S}^{pMT \times pMT}$. The update for the posterior covariance is the computationally limiting step:

$$\bar{\Sigma}_x = \left(\bar{K}^{-1} + \langle \overline{C^\top \Phi C} \rangle\right)^{-1}. \tag{3.14}$$

Here we introduce the notation $\langle \cdot \rangle$ to indicate the expectation with respect to the approximate posterior distribution, $\mathbb{E}_Q[\cdot]$. The term $\overline{\langle C^\top \Phi C \rangle} \in \mathbb{R}^{pMT \times pMT}$ is a block diagonal matrix comprising T copies of the matrix $\langle C^\top \Phi C \rangle$. The elements of the prior covariance matrix $\bar{K} \in \mathbb{R}^{pMT \times pMT}$ follow the structure of $\bar{\mathbf{x}}_n$ and are computed using equations 3.8 and 3.9. Evaluating

equation 3.14 on each mDLAG fitting iteration thus requires the inversion of a $pMT \times pMT$ matrix, costing $\mathcal{O}(p^3M^3T^3)$ operations and $\mathcal{O}(p^2M^2T^2)$ storage. Ameliorating this computational cost and the remaining costs of the three bottleneck stages above is the focus of the rest of this work.

3.2 mDLAG with Inducing Variables (mDLAG-Inducing). The scaling challenges of mDLAG-time stem from the gaussian process state model (equation 3.7), namely, the $MT \times MT$ covariance matrix K_j and the computation of its inverse and determinant. Alternative formulations of the state model that mitigate or avoid these computations altogether are therefore desirable. We first consider an approach that incorporates inducing variables (Álvarez et al., 2010; Duncker et al., 2018; Titsias, 2009). In essence, we might be able to improve scalability if instead of the full set of latents across groups, we manipulate one latent representation common to all groups (thereby improving scaling with M) at a lower temporal resolution (thereby improving scaling with T). This approach, mDLAG-inducing, requires no changes to the mDLAG-time observation model or ARD components (equations 3.1–3.6), only the state model (see Figure 1C).

3.2.1 Gaussian Process State Model with Inducing Variables. For each latent j on trial n we define a corresponding inducing variable $\mathbf{w}_{n,j,:} = [w_{n,j,1} \cdots w_{n,j,T_{\text{ind}}}]^\top \in \mathbb{R}^{T_{\text{ind}}}$. Inducing variable values are defined over a set of inducing (time) points indexed by $t \in \{1, \ldots, T_{\text{ind}}\}$. The number of inducing points per trial, T_{ind}, is a hyperparameter, and in this work we choose T_{ind} to be no greater than the number of time points per trial, T, so that $T_{\text{ind}} \leq T$ (in principle, T_{ind} can also be chosen separately for each latent j, but we choose not to do so here). While the number of inducing points is discrete, the times at which the inducing variables are defined are real-valued. In detail, inducing variable $w_{n,j,t}$ at inducing point t, $t \in \{1, \ldots, T_{\text{ind}}\}$ is defined at time $\xi_t \in \mathbb{R}$.

These locations in time are themselves a design choice and can even be treated as learnable model parameters, just like the GP parameters (Duncker & Sahani, 2018; Snelson & Ghahramani, 2005; Titsias, 2009). Throughout this work, we will fix the inducing points on a uniformly spaced grid, with the first and last inducing points fixed at the beginning and end of each trial—$\xi_1 = 1$ and $\xi_{T_{\text{ind}}} = T$. We do not explore inducing points as learnable model parameters here to facilitate comparison with the other two methods (updating the inducing points adds a significant computational cost to the model-fitting procedure), but we do include that option in accompanying code (see appendix E).

We then define a GP over all inducing points $\xi_1, \ldots, \xi_{T_{\text{ind}}}$ for each latent $j = 1, \ldots, p$ as follows:

$$\mathbf{w}_{n,j,:} \sim \mathcal{N}\left(\mathbf{0}, K_j^w\right), \tag{3.15}$$

where $K_j^w \in \mathbb{S}^{T_{\text{ind}} \times T_{\text{ind}}}$ is the inducing variable covariance matrix (see Figure 1D, top, K_j^w). Continuing with the squared exponential GP covariance function, element (t_1, t_2) of K_j^w is computed according to

$$k_j^w(\xi_{t_1}, \xi_{t_2}) = \left(1 - \sigma_j^2\right) \exp\left(-\frac{(\Delta t)^2}{2\tau_j^2}\right) + \sigma_j^2 \cdot \delta_{\Delta t} \tag{3.16}$$

$$\Delta t = \xi_{t_2} - \xi_{t_1}. \tag{3.17}$$

The characteristic timescale, $\tau_j \in \mathbb{R}_{>0}$, and the GP noise variance, $\sigma_j^2 \in (0, 1)$, are defined as in equation 3.8.

We then define each latent j across groups $m = 1, \ldots, M$, $\mathbf{x}_{n,j,:} = [\mathbf{x}_{n,j,:}^{1\top} \cdots \mathbf{x}_{n,j,:}^{M\top}]^\top \in \mathbb{R}^{MT}$, in terms of the common inducing variable, $\mathbf{w}_{n,j,:}$, via the following conditional gaussian distribution:

$$P\left(\mathbf{x}_{n,j,:} \mid \mathbf{w}_{n,j,:}\right) = \mathcal{N}\left(\mathbf{x}_{n,j,:} \mid K_j^{xw}(K_j^w)^{-1}\mathbf{w}_{n,j,:}, K_j - K_j^{xw}(K_j^w)^{-1}K_j^{wx}\right), \tag{3.18}$$

where $K_j \in \mathbb{S}^{MT \times MT}$ is the covariance matrix defined in equation 3.12 (see Figure 1B), and $K_j^{xw} = (K_j^{wx})^\top \in \mathbb{R}^{MT \times T_{\text{ind}}}$ is the cross-covariance matrix between latent j and its inducing variable (see Figure 1D, bottom, K_j^{xw}). If we define $K_{m,j}^{xw} \in \mathbb{R}^{T \times T_{\text{ind}}}$ to be the mth block of $K_j^{xw} = [K_{1,j}^{xw\top} \cdots K_{M,j}^{xw\top}]^\top$, then element (t_1, t_2) of $K_{m,j}^{xw}$, where $t_1 \in \{1, \ldots, T\}$ and $t_2 \in \{1, \ldots, T_{\text{ind}}\}$, can be computed according to the covariance function,

$$k_j^{xw}(t_1, \xi_{t_2}) = \left(1 - \sigma_j^2\right) \exp\left(-\frac{(\Delta t)^2}{2\tau_j^2}\right) + \sigma_j^2 \cdot \delta_{\Delta t} \tag{3.19}$$

$$\Delta t = \xi_{t_2} - (t_1 - D_j^m), \tag{3.20}$$

with the same timescale parameter τ_j and GP noise variance σ_j^2 as defined in equation 3.16. The time delay parameter $D_j^m \in \mathbb{R}$ is defined in equation 3.9. Under this formulation, each $\mathbf{x}_{n,j,:}^m$ for latent j and group m can be viewed as an interpolated (upsampled) and time-delayed version of the common inducing variable $\mathbf{w}_{n,j,:}$.

3.2.2 Posterior Inference, Fitting, and Computational Scaling. Let Y, X, and W be collections of all observed variables, latent variables, and inducing variables, respectively, across all time points (or inducing points) and trials. As with mDLAG-time, let $\mathbf{d}$, ϕ, C, $\mathcal{A}$, τ, and D be collections of the mean parameters, noise precisions, loading matrices, ARD parameters, GP timescales, and time delays, respectively. From the observed activity, we seek to estimate posterior distributions over the probabilistic model

components $\theta = \{X, W, \mathbf{d}, \phi, C, \mathcal{A}\}$, which now includes the inducing variables W, and point estimates of the deterministic GP parameters $\Omega = \{\tau, D\}$.

We again employ variational inference, in which we maximize the ELBO, $L(Q, \Omega)$ (equation 3.10), with respect to the approximate posterior distribution $Q(\theta)$ and the deterministic parameters Ω. For mDLAG-inducing, however, we follow Titsias (2009) and constrain $Q(\theta)$ so that it factorizes over the elements of θ as follows:

$$Q(\theta) = Q_{xw}(X, W)Q_d(\mathbf{d})Q_\phi(\phi)Q_c(C)Q_{\mathcal{A}}(\mathcal{A}), \qquad (3.21)$$

$$= P(X|W)Q_w(W)Q_d(\mathbf{d})Q_\phi(\phi)Q_c(C)Q_{\mathcal{A}}(\mathcal{A}). \qquad (3.22)$$

Here we have constrained the joint approximate posterior distribution over the latents and their inducing variables to factorize as $Q_{xw}(X, W) = P(X|W)Q_w(W)$, with generic distribution $Q_w(W)$ over the inducing variables and the conditional prior distribution $P(X|W)$ over the latents themselves. From equation 3.18, $P(X|W) = \prod_j \prod_n P(\mathbf{x}_{n,j,:} \mid \mathbf{w}_{n,j,:})$. Then as with mDLAG-time, this factorization enables closed-form updates during optimization. The ELBO can be iteratively maximized via coordinate ascent of the factors of $Q(\theta)$ and the deterministic parameters Ω.

As with mDLAG-time in section 3.1.3, the three key stages of the mDLAG-inducing fitting procedure are (1) the update of the posterior distribution over the latents and their inducing variables, $Q_{xw}(X, W)$; (2) the updates of the GP parameters, Ω, via gradient ascent; and (3) evaluation of the variational lower bound, equation 3.10. A complete set of update equations, including for these three key stages, is provided in appendix B. At all three stages, mDLAG-inducing achieves linear scaling in both the number of groups M and the number of time points per trial T, albeit with superlinear scaling in the number of inducing points T_{ind}. For example, the limiting computation in the update of $Q_{xw}(X, W)$ costs $\mathcal{O}(p^3 M T T_{\text{ind}}^2)$ operations (see appendix B).

3.3 mDLAG in the Frequency Domain (mDLAG-Frequency). Consider now the unitary discrete Fourier transform (DFT) of the time series of observations for unit r in group m on trial n: $\widetilde{\mathbf{y}}_{n,r,:}^m = U_T \mathbf{y}_{n,r,:}^m$, where $U_T \in \mathbb{C}^{T \times T}$ is the unitary DFT matrix. Our goal is to develop a generative model for these observations while remaining entirely in the frequency domain (for mDLAG-frequency; see Figure 1E), with the hope of uncovering computational benefits. Note that the conversion of each unit's activity from the time domain to the frequency domain has time complexity $\mathcal{O}(T \log T)$, assuming the DFT is implemented using a fast Fourier transform (FFT) algorithm. Developing a generative model entirely in the frequency domain, and thereby procedures for posterior inference and fitting entirely in the frequency domain, requires this computation only once, as a preprocessing

step. Then, as we will show, each iteration of the fitting procedure will scale linearly in T and M.

3.3.1 Gaussian Process State Model.

For the frequency domain to provide those computational benefits, it must improve the scalability of the limiting mDLAG-time model component, the GP state model. We first consider the case with $M = 1$ group. Taking the unitary DFT of the time course of latent j on trial n, $\mathbf{x}_{n, j, :}$, we get

$$U_T \mathbf{x}_{n, j, :} \sim \mathcal{N}\left(\mathbf{0}, U_T K_j U_T^{\mathsf{H}}\right), \tag{3.23}$$

a linear transformation of the state model (U_T^{H} is the conjugate transpose of U_T). The key quantity in equation 3.23 is the transformed covariance matrix $\widetilde{K}_j = U_T K_j U_T^{\mathsf{H}}$. Unfortunately, like the time domain covariance matrix K_j, the frequency domain covariance matrix $\widetilde{K}_j$ is, in general, dense. Consequently, any fitting procedure based on this state model will still scale cubically in the number of time points per trial. We will get no computational benefit over the bottleneck mDLAG-time updates (see section 3.1.3).

But critically, for large T (and for a stationary GP), the transformed covariance matrix $\widetilde{K}_j$ approaches a diagonal matrix, a well-established result in spectral analysis (Kolmogorov et al., 1941; Priestley, 1981). The diagonal elements of that limiting matrix are given by the power spectral density (PSD) of the gaussian process, a function of frequency. By the Wiener–Khinchin theorem, the PSD function of latent j, s_j, is given by the continuous-time Fourier transform of its covariance function, k_j (treated as a function of the time difference $t_2 - t_1$). For the squared exponential covariance function in equation 3.8, the PSD function takes the following closed form,

$$s_j(f_l) = \left(1 - \sigma_j^2\right) \sqrt{2\pi} \tau_j \exp\left(-\tfrac{1}{2} \left(2\pi f_l\right)^2 \tau_j^2\right) + \sigma_j^2, \tag{3.24}$$

where $f_l \in \mathbb{R}$ is a frequency value (l indexes the components of a discretely sampled signal; see below). The characteristic timescale, $\tau_j \in \mathbb{R}_{>0}$, and the GP noise variance, $\sigma_j^2 \in (0, 1)$, are defined as in equation 3.8.

A generative model for a GP with a diagonal covariance matrix would lead to significant runtime advantages. Toward that end, let $\widetilde{\mathbf{x}}_{n, j, :} \in \mathbb{C}^T$ be the values of latent j across frequencies $l = 1, \ldots, T$ on trial n. Following the symmetry properties of the DFT, the index $l = 1$ corresponds to the zero frequency, $f_1 = 0$. For even values of T, the indices $l = 2, \ldots, \frac{T}{2} + 1$ correspond to increasing positive frequencies, where $l = \frac{T}{2} + 1$ indexes the Nyquist frequency ($f_{\frac{T}{2}+1}$ is half the sampling rate of observations). Indices $l = \frac{T}{2} + 2, \ldots, T$ correspond to negative frequencies increasing

toward zero, with symmetry about the Nyquist rate such that $f_{\frac{T}{2}+2} = -f_{\frac{T}{2}}$, $f_{\frac{T}{2}+3} = -f_{\frac{T}{2}-1}$, up to $f_T = -f_2$. Odd values of T lead to similar frequency ordering but with subtle differences in bookkeeping.

Then as an approximation to the GP prior defined in equation 3.23, we define a (complex-valued) GP over all frequencies for each frequency domain latent $j = 1, \ldots, p$, where the covariance matrix is diagonal by construction (see Figure 1F, top, S_j):

$$\widetilde{\mathbf{x}}_{n,j,:} \sim \mathcal{N}\left(\mathbf{0}, S_j\right). \tag{3.25}$$

The lth element of the diagonal "PSD matrix" $S_j \in \mathbb{S}^{T \times T}$ is given by the PSD function of the GP (equation 3.24). We explore the consequences of this approximation in section 4.5. For now, we continue to define the frequency domain generative model under this approximation and extract any computational benefit that arises.

3.3.2 Observation Model. From here, we could define a linear observation model from latents to observed activity, akin to equation 3.1. However, we have omitted in the development of the model thus far any notion of time delays. Let us revisit, for the multigroup ($M > 1$) case, the continuous-time Fourier transform of the cross-covariance function $k_{m_1,m_2,j}$ (equation 3.8) between groups m_1 and m_2:

$$s_{m_1,m_2,j}(f_l) = \left(1 - \sigma_j^2\right)\sqrt{2\pi}\,\tau_j \exp\left(-\frac{1}{2}\,(2\pi f_l)^2\,\tau_j^2\right)$$

$$\cdot \exp\left(-i2\pi f_l(D_j^{m_1} - D_j^{m_2})\right) + \sigma_j^2. \tag{3.26}$$

Here $i = \sqrt{-1}$ is the imaginary number. This cross-spectral density (CSD) function is nearly identical in form to the PSD function of equation 3.24, but with the addition of a multiplicative complex exponential term, $\exp(-i2\pi f_l(D_j^{m_1} - D_j^{m_2}))$. This term isolates the time-delay parameters $D_j^{m_1}$ and $D_j^{m_2}$, representing them as a relative phase shift, $2\pi f_l(D_j^{m_1} - D_j^{m_2})$, at frequency f_l. Note the separability of the complex exponential term by groups:

$$\exp\left(-i2\pi f_l(D_j^{m_1} - D_j^{m_2})\right) = \exp\left(-i2\pi f_l D_j^{m_1}\right) \cdot \exp\left(i2\pi f_l D_j^{m_2}\right)$$

$$= \exp\left(-i2\pi f_l D_j^{m_1}\right) \cdot \exp\left(-i2\pi f_l D_j^{m_2}\right)^*$$

$$= h_{j,l}^{m_1} \cdot (h_{j,l}^{m_2})^*, \tag{3.27}$$

where we have defined $h_{j,l}^m = \exp(-i2\pi f_l D_j^m)$ to be the phase shift term from latent j to observed group m at frequency l.

We can then construct a frequency domain analog, $\widetilde{\mathbf{x}}_{n,j,:}^m \in \mathbb{C}^T$, to the time delayed latent to group m, $\mathbf{x}_{n,j,:}^m \in \mathbb{R}^T$. Collect the phase-shift terms $h_{j,l}^m$ across all frequencies $l = 1, \ldots, T$ into the diagonal matrix $H_j^m = \text{diag}(h_{j,1}^m, \ldots h_{j,T}^m) \in \mathbb{C}^{T \times T}$ (see Figure 1F, bottom, H_j^m). Then we get $\widetilde{\mathbf{x}}_{n,j,:}^m$ by multiplication of the diagonal phase shift matrix H_j^m with the frequency domain latent j defined in equation 3.25: $\widetilde{\mathbf{x}}_{n,j,:}^m = H_j^m \widetilde{\mathbf{x}}_{n,j,:}$.

We now have the conceptual basis to define a frequency domain observation model, from latents to observed activity. For group m (comprising q_m units) at frequency l on trial n, we define a linear relationship between observed activity, $\widetilde{\mathbf{y}}_{n,l}^m \in \mathbb{C}^{q_m}$, and the set of latents common to all groups, $\widetilde{\mathbf{x}}_{n,t} \in \mathbb{C}^p$ (see Figure 1E):

$$\widetilde{\mathbf{y}}_{n,l}^m = C^m H_l^m \widetilde{\mathbf{x}}_{n,l} + \widetilde{\mathbf{d}}_l^m + \boldsymbol{\varepsilon}^m, \tag{3.28}$$

$$\boldsymbol{\varepsilon}^m \sim \mathcal{N}(\mathbf{0}, (\Phi^m)^{-1}), \tag{3.29}$$

Here the diagonal matrix $H_l^m \in \mathbb{C}^{p \times p}$ is a different collection of the phase shift terms $h_{j,l}^m$, across all latents $j = 1, \ldots, p$ at the single frequency l: $H_l^m = \text{diag}(h_{1,l}^m, \ldots h_{p,l}^m)$. The mean parameter term $\widetilde{\mathbf{d}}_l^m \in \mathbb{R}^{q_m}$ is shorthand for $\delta_{l-1} \cdot \sqrt{T}\mathbf{d}^m$, which is $\sqrt{T}\mathbf{d}^m$ for the zero frequency ($f_1 = 0$ for index $l = 1$), and $\mathbf{0}$ otherwise.

All other parameters are the same as defined in the mDLAG-time observation model (equations 3.1 and 3.2). We further set the same prior distributions over the mean parameters $\mathbf{d}^m$ (equation 3.3) and noise precision parameters (equation 3.4), and we maintain the ARD prior over the columns of each loading matrix C^m (equations 3.5 and 3.6). Note that the PSD function given by equation 3.24 is normalized so that $\int_{-\infty}^{+\infty} s_j(f)df = 1$, thereby removing—as was the case for mDLAG-time—model redundancy in the scaling of the latents and loading matrices C^m.

3.3.3 Posterior Inference, Fitting, and Computational Scaling. Fitting the mDLAG-frequency model proceeds analogous to fitting for mDLAG-time. Let $\widetilde{Y}$ and $\widetilde{X}$ be collections of all observed and latent variables, respectively, across all frequencies and trials. As before, let $\mathbf{d}$, ϕ, C, $\mathcal{A}$, τ, and D be collections of the mean parameters, noise precisions, loading matrices, ARD parameters, GP timescales, and time delays, respectively. From frequency domain observations, we seek to estimate posterior distributions over the probabilistic model components $\theta = \{\widetilde{X}, \mathbf{d}, \phi, C, \mathcal{A}\}$ and point estimates of the deterministic GP parameters $\Omega = \{\tau, D\}$.

We again employ variational inference, in which we maximize the lower bound $\widetilde{L}(\widetilde{Q}, \Omega)$ to the log quasi-likelihood $\log P(\widetilde{Y})$, with respect to the approximate posterior distribution $\widetilde{Q}(\theta)$ and the deterministic parameters Ω:

$$\log P(\widetilde{Y}) \geq \widetilde{L}(\widetilde{Q}, \Omega) = \mathbb{E}_{\widetilde{Q}}[\log P(\widetilde{Y}, \theta | \Omega)] - \mathbb{E}_{\widetilde{Q}}[\log \widetilde{Q}(\theta)]. \qquad (3.30)$$

The quasi-likelihood $P(\widetilde{Y})$ is meant to approximate the time domain likelihood $P(Y)$ (equation 3.10). The two quantities converge as the number of time points per trial, T, becomes large (Whittle, 1951).

We again constrain $\widetilde{Q}(\theta)$ so that it factorizes over the elements of θ:

$$\widetilde{Q}(\theta) = Q_{\tilde{x}}(\widetilde{X})Q_d(\mathbf{d})Q_\phi(\phi)Q_c(C)Q_{\mathcal{A}}(\mathcal{A}). \qquad (3.31)$$

In the same manner as mDLAG-time, this factorization enables closed-form updates during optimization, and the lower bound $\widetilde{L}(\widetilde{Q}, \Omega)$ can be iteratively maximized with coordinate ascent of the factors of $\widetilde{Q}(\theta)$ and the deterministic parameters Ω.

As with the previous two methods, the three key stages of the mDLAG-frequency fitting procedure are (1) updating the posterior distribution over latents, $Q_{\tilde{x}}(\widetilde{X})$; (2) updating the GP parameters, Ω, via gradient ascent; and (3) evaluating the variational lower bound, equation 3.30. At all three stages, mDLAG-frequency achieves linear scaling in both the number of groups M and the number of time points per trial T. A complete set of update equations, including for these three key stages, is provided in appendix C.

Here, we highlight just the update of the posterior distribution over latents, $Q_{\tilde{x}}(\widetilde{X})$. The update to $Q_{\tilde{x}}(\widetilde{X})$ takes the same functional form as the prior distribution (equation 3.25), a gaussian distribution,

$$Q_{\tilde{x}}(\widetilde{X}) = \prod_{n=1}^{N} \prod_{l=1}^{T} \mathcal{N}(\widetilde{\mathbf{x}}_{n,l} \mid \widetilde{\boldsymbol{\mu}}_{x_{n,l}}, \widetilde{\Sigma}_{x,l}), \qquad (3.32)$$

with trial- and frequency-dependent posterior mean $\widetilde{\boldsymbol{\mu}}_{x_{n,l}} \in \mathbb{C}^p$ and frequency-dependent but trial-independent posterior PSD matrix $\widetilde{\Sigma}_{x,l} \in \mathbb{C}^{p \times p}$. Note the independence in equation 3.32 not only across trials, but also across frequencies. This independence emerges naturally: we impose only the factorization in equation 3.31. By contrast, the analogous updates for mDLAG-time are independent across trials only, not time points (equation 3.13). We can thus update the posterior PSD matrix, $\widetilde{\Sigma}_{x,l}$, separately for each frequency l:

$$\widetilde{\Sigma}_{x,l} = \left(S_l^{-1} + \sum_{m=1}^{M} (H_l^m)^{\mathsf{H}} \langle (C^m)^\top \Phi^m C^m \rangle H_l^m \right)^{-1}. \qquad (3.33)$$

The elements of the diagonal PSD matrix $S_l = \mathrm{diag}(s_1(f_l), \ldots, s_p(f_l)) \in \mathbb{R}^{p \times p}$ are computed using equation 3.24. The diagonal phase shift matrix $H_l^m \in \mathbb{C}^{p \times p}$ is defined as in equation 3.28.

Evaluation of equation 3.33 for all frequencies $l = 1, \ldots, T$ is significantly more efficient than the evaluation of the analogous equation 3.14 for

Table 1: Summary of Simulated Data Sets.

Data Set	Demo	Scaling, T	Scaling, M	Model selection
Independent runs	1	20	20	10 per SNR
Training trials	100	100	100	50
Time points	100	10–500	50	10–200
Samp. period (ms)	20	20	20	20
Trial duration (s)	2	0.2–10	1	0.2–4
Groups	2	2	1–24	1
Units per group	10	12	24–1	1
Latents	4	1	1	4
GP timescales (ms)	20–120	100	100	50
GP time delays (ms)	+12, −23	10	0–20	—
Signal-to-noise ratio	0.2	0.2	0.2	0.01–10.0

mDLAG-time. In the first term, the PSD matrix S_l is diagonal, and thus its inversion scales linearly in the number of latents, p. In the second term, H_l^m is diagonal, and thus the sum over groups $m = 1, \ldots, M$ costs $\mathcal{O}(p^2 M)$, and $\mathcal{O}(p^2 MT)$ in total when evaluated for all frequencies. The final computation is the inversion of a $p \times p$ matrix, costing $\mathcal{O}(p^3 T)$ operations and $\mathcal{O}(p^2 T)$ storage, when evaluated for all frequencies. Thus in total, posterior inference over the latents scales linearly in both the number of time points per trial T and the number of groups M.

4 Results

4.1 Demonstration in Simulation. We start with an illustrative simulation to demonstrate, across our three fitting methods, (1) their basic correctness, (2) their relative runtime performance, and (3) themes explored more deeply in subsequent sections. Data set characteristics are summarized in Table 1.

We generated simulated activity from two groups ($M = 2$) according to the mDLAG generative model (see equations 3.1 to 3.9 and section 6.1). For illustration, we set 10 units in each group ($q_m = 10$). We designed the loading matrices C^m so that all types of intergroup interactions were represented (see Figure 2A, left, ground truth): a feedforward interaction (A to B), a feedback interaction (B to A), and an interaction local to each group. We also scaled the observation noise precision matrices Φ^m so that noise levels were representative of realistic neural activity. Specifically, activity due to single-unit observation noise was five times stronger than activity due to latents (the signal-to-noise ratio, or SNR, $\mathrm{tr}(C^m C^{m\top})/\mathrm{tr}((\Phi^m)^{-1}) = 0.2$ for each group). Gaussian process timescales and time delays were also chosen within realistic ranges (see Table 1, Demo). With all model parameters specified, we then generated $N = 100$ independent and identically distributed
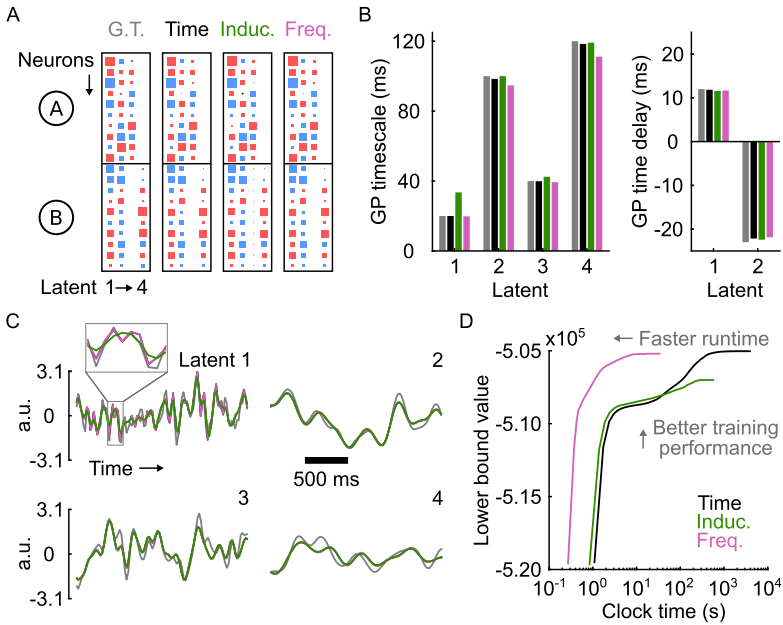
Figure 2: Demonstrative simulation. (A) Loading matrix estimates. From left to right: Ground truth (G.T.), mDLAG-time (Time), mDLAG-inducing (Induc.), mDLAG-frequency (Freq.). Here the loading matrices C^1 and C^2, for groups A and B, respectively, have been concatenated vertically. Each element of each 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). Note that the sign and ordering of each loading matrix column are 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) Gaussian process (GP) parameter estimates. Left: GP timescales. Right: GP time delays. Time delays are only relevant for latents 1 and 2. Latents 3 and 4 are local to groups A and B, respectively. (C) Latent time course estimates for an example training trial. Each panel includes the ground truth and estimated time course—for each method—of a single latent variable. All estimated time courses, regardless of the method used for model fitting, were computed using the mDLAG-time inference equation 3.13. Inset: zoomed-in view of latent 1 time courses. Note that mDLAG-time and mDLAG-frequency estimates are visually indistinguishable, and thus mDLAG-time estimates (black) are occluded by mDLAG-frequency estimates (magenta). For concision, for latents 1 and 2 we show only latents corresponding to group A ($x_{n,j,:}^m$); the latents corresponding to group B are time-shifted versions of those shown here. a.u.: arbitrary units. (D) Lower-bound value versus elapsed clock time over the course of model fitting. Note that each method optimizes a slightly different lower bound. Lower-bound values are therefore not directly comparable across methods. Color scheme is consistent across panels A to D. Gray: ground truth; black: mDLAG-time; green: mDLAG-inducing; magenta: mDLAG-frequency.

trials. Each trial was 2 s in length, comprising $T = 100$ time points with a sampling period of 20 ms, or a rate of 50 Hz.

We then fit three mDLAG models to the simulated activity using each of the three fitting methods: mDLAG-time, mDLAG-inducing, and mDLAG-frequency. For all fitting methods, we set the initial number of latents ($p = 8$) to be greater than the ground-truth number ($p = 4$) to verify that these additional latents would be pruned during fitting (see section 6.2). For mDLAG-inducing we chose the number of inducing points to be half of the number of time points per trial ($T_{\text{ind}} = 50$; we elaborate on the choice of this hyperparameter in subsequent sections).

All fitting methods produced similar estimates. The number of latent variables ($p = 4$) was correctly identified in all cases (see section 6.3). The loading matrices (C) and their group-wise sparsity patterns were also recovered with high accuracy (see Figure 2A; normalized Frobenius norm between ground-truth loading matrix and estimates: mDLAG-time, 0.0462; mDLAG-inducing, 0.0538; mDLAG-frequency, 0.0467). GP timescale (see Figure 2B, left) and time delay (Figure 2B, right) estimates were all within 10% of the ground truth, with one exception: mDLAG-inducing produced a significant overestimate of the timescale for latent 1 (see Figure 2B, left, latent 1; compare green to gray). Consequently, the trial-to-trial time course estimates of latent 1, based on the mDLAG-inducing fit, were overly smooth (see Figure 2C, latent 1; see inset), leading to lower reconstruction accuracy than the other two fitting methods (R^2 between ground truth and estimated time courses: mDLAG-time, 0.8653; mDLAG-inducing, 0.8357; mDLAG-frequency, 0.8652). We elaborate on this phenomenon in subsequent sections. The mDLAG-time and mDLAG-frequency methods produced practically indistinguishable estimates of latent time courses (see Figure 2C, black and magenta traces coincide).

The runtime for mDLAG-frequency, however, was significantly faster than for the other two fitting methods (see Figure 2D). Per fitting iteration, mDLAG-frequency ran more than an order of magnitude faster than either mDLAG-time or mDLAG-inducing (wall clock time per iteration: mDLAG-time, 0.616 s; mDLAG-inducing, 0.464 s; mDLAG-frequency, 0.032 s; see appendix E for details on computing resources used). Compared to mDLAG-time, mDLAG-frequency also required significantly fewer iterations to reach convergence (mDLAG-time, 6,952; mDLAG-inducing, 1,282; mDLAG-frequency, 1,091). mDLAG-frequency appears to converge in fewer iterations than mDLAG-time because, at least in part, the GP parameters converge more quickly to their optimal values (see supplementary Figure 1). The total runtime to convergence of mDLAG-frequency was thus faster by a factor of over 100 (total wall clock time: mDLAG-time, 4,283 s; mDLAG-inducing, 595 s; mDLAG-frequency, 35 s).

4.2 Scaling with Number of Time Points per Trial. We next performed a more exhaustive characterization of the runtime performance of each

fitting method, starting with runtime scaling as a function of the number of time points per trial. We again generated simulated activity from two groups ($M = 2$) according to the mDLAG generative model (section 6.1), with a few key differences from above (see Table 1, Scaling, T). Without loss of generality, we fixed one global latent variable, shared across both groups ($p = 1$), and fixed the associated GP timescale and time delay throughout all experiments (100 ms and 10 ms for the GP timescale and time delay, respectively). We generated data sets comprising $T = 500$ time points per trial (10 s in length). Beyond $T = 500$, mDLAG-time was prohibitively expensive to run, precluding method-to-method comparisons (but see below for an application of mDLAG-frequency to longer trials). We then fit mDLAG models to increasingly long trial epochs (lengths spaced equally on a log scale from 10 to 500 time points per trial). Here we chose the same number of estimated latent variables as the ground truth ($p = 1$). Overall, we ran 20 independent experiments ("runs" on independent data sets produced by different randomly generated mDLAG models) to gauge variability in performance.

Across nearly all trial lengths, T, the three fitting methods gave practically indistinguishable statistical performance (see Figure 3A: black, green, and magenta traces visually indistinguishable). All methods exhibited a leave-unit-out test R^2 value (see section D.5) close to the noise ceiling (see Figure 3A, black dashed line, $R^2 = 0.167$). Below 50 time points per trial (1 s trial duration), mDLAG-frequency exhibited slightly worse performance (see Figure 3A, inset; in the worst case, R^2 for mDLAG-frequency was 1% worse than the other two methods at $T = 10$ time points, or 200 ms duration, per trial).

However, mDLAG-frequency exhibited dramatically better runtime scaling with the number of time points per trial. In agreement with theory, per fitting iteration, mDLAG-time exhibited cubic scaling with T (see Figure 3B: compare mDLAG-time, black trace, to the cubic reference slope, gray dotted line), while mDLAG-frequency exhibited linear scaling or better (Figure 3B: compare mDLAG-frequency, magenta solid trace, to the linear reference slope, gray dashed line). Fitting with mDLAG-frequency thus produced a $100\times$ speed-up per iteration at $T = 500$ (mean clock time per iteration: mDLAG-time, 6.030 ± 0.048 s; mDLAG-inducing, 0.524 ± 0.005 s; mDLAG-frequency, 0.059 ± 0.002 s). Because mDLAG-time also exhibited an increase in the number of iterations required for convergence (Figure 3C, at $T = 500$: mDLAG-time, $3,822 \pm 209$; mDLAG-inducing, $2,242 \pm 91$; mDLAG-frequency, 424 ± 79; see also supplementary Figure 1), the computational advantage of mDLAG-frequency compounded: at $T = 500$, the overall runtime of mDLAG-frequency was 1,000x faster than that of mDLAG-time (see Figure 3D, total clock time: mDLAG-time, $22,871 \pm 1,128$ s; mDLAG-inducing, $1,167 \pm 37.0$ s; mDLAG-frequency, 23.3 ± 3.4 s).

Note that mDLAG-inducing scaled superlinearly with T (see Figure 3B: compare mDLAG-inducing, green trace, to the linear reference slope, gray
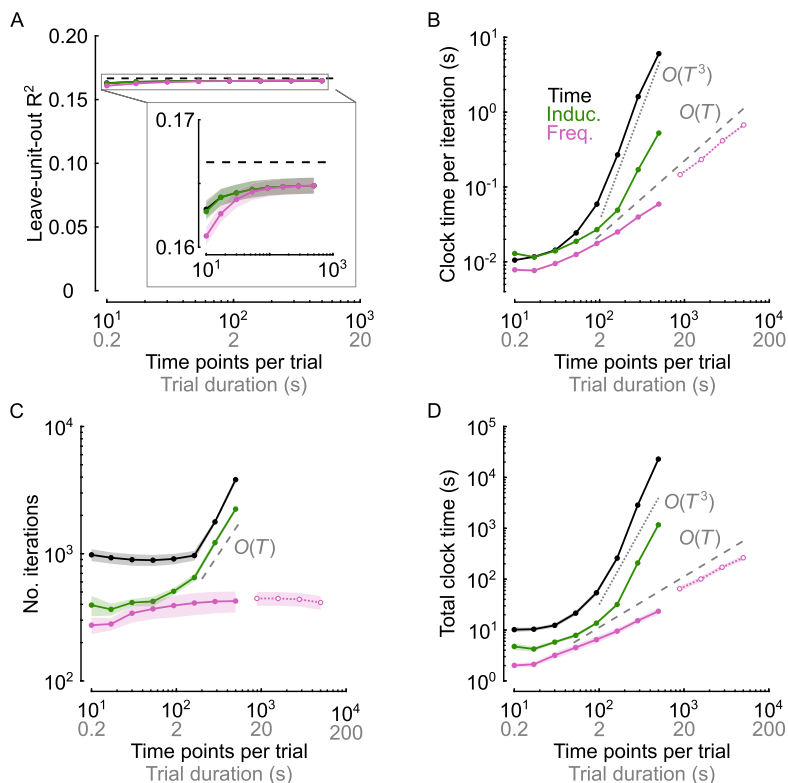
Figure 3: Scaling of performance and runtime with number of time points per trial. (A) Leave-unit-out R^2 versus number of time points per trial. For each run, performance was evaluated on 100 held-out test trials. The performance curves for each method are visually indistinguishable from one another. Black dashed line: noise ceiling ($R^2 = 0.167$ for these simulations). Inset: same data, but with vertical axis magnified to show the difference between methods. (B) Mean (across fitting iterations) clock time per iteration versus number of time points per trial. Gray dashed line: reference slope for a linear scaling law, $O(T)$. Gray dotted line: reference slope for a cubic scaling law, $O(T^3)$. (C) Number of iterations to convergence versus number of time points per trial. (D) Total clock time to convergence versus number of time points per trial. Across panels A to D, solid traces indicate the mean, and shading (where visible) indicates standard error of the mean, computed across 20 runs. Color scheme is consistent across panels. Black: mDLAG-time (Time); green: mDLAG-inducing (Induc.); magenta: mDLAG-frequency (Freq.). Across panels B to D, dotted magenta traces with open circles show results from applying mDLAG-frequency to a separate, additional set of 20 runs with extended trial lengths, up to $T = 5000$ time points (100 s in duration). The slight discontinuity between solid magenta and dotted magenta in panel B is due to increased memory overhead for the larger data sets.

dashed line). In theory, for a fixed number of inducing points, T_{ind}, mDLAG-inducing should scale linearly with the number of time points per trial. Here, however, as T increased, we also increased the number of inducing points to maintain it above the approximate Nyquist rate of the underlying latent time course (for a squared exponential GP with 100 ms timescale, a sampling rate of 12.5 Hz or greater captures essentially 100% of the frequency content). Without doing so, the performance of mDLAG-inducing suffers significantly (see supplementary Figure 2; see also Figure 2C, latent 1). Consequently, the runtime benefit (with increasing T) of mDLAG-inducing over mDLAG-time is less significant than that of mDLAG-frequency.

The linear runtime scaling of mDLAG-frequency allowed us to apply it to data sets with even longer trials. We generated an additional 20 independent data sets with $T = 5,000$ time points per trial (100 s in length). All other data set characteristics remained the same as above (see Table 1, Scaling, T). We then fit models via mDLAG-frequency to increasingly long trial epochs, from $T = 890$ to $T = 5,000$ time points in length, to demonstrate mDLAG-frequency's extended runtime scaling (see Figures 3B–D, magenta dotted traces and open circles). mDLAG-frequency not only maintained linear scaling with T up to $T = 5,000$ (Figures 3B and 3D: compare magenta dotted trace to the linear reference slope, gray dashed line), but remained faster in total runtime than mDLAG-time and mDLAG-inducing applied to $T = 500$ time points per trial.

4.3 Scaling with Number of Groups. We performed a similar characterization of runtime scaling as a function of the number of observation groups (see Table 1, Scaling, M). Here, we fixed the total number of units throughout experiments at $q = 24$, but then considered a varying number of groups, from $M = 1$ (with 24 units belonging to one group) to $M = 24$ (with one "group" comprising one unit). We again fixed one global latent variable ($p = 1$), with the same GP timescale as in section 4.2 (100 ms), but we selected the magnitudes of pairwise time delays, uniformly at random, between 0 ms and 20 ms. The number of time points per trial was fixed at $T = 50$ (1 s in length). The number of estimated latent variables was again chosen to match the ground truth ($p = 1$). For mDLAG-inducing, we chose $T_{ind} = 13$ inducing points, the smallest number that maintained the approximate Nyquist rate of the latent signals (12–13 Hz for a squared exponential GP with 100 ms timescale). Overall, we ran 20 independent experiments at each group number (M) to gauge variability in performance.

Across all group numbers, M, the three fitting methods gave practically indistinguishable statistical performance (see Figure 4A: black, green, and magenta traces visually indistinguishable). Both mDLAG-frequency and mDLAG-inducing, however, scaled significantly better than mDLAG-time. In agreement with theory, per fitting iteration, mDLAG-time exhibited near-cubic scaling with the number of groups (Figure 4B: compare
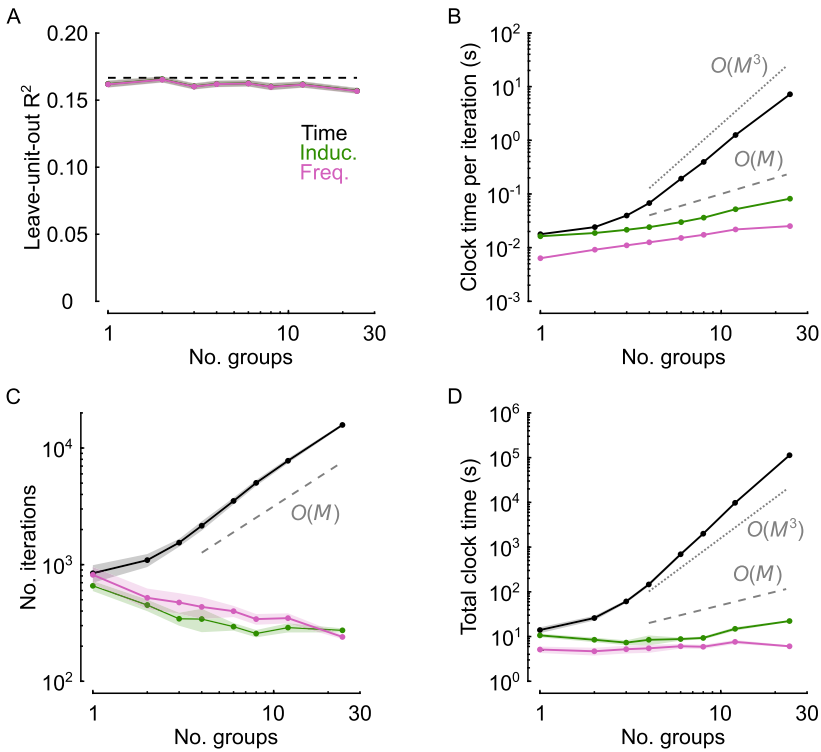
Figure 4: Scaling of performance and runtime with number of groups. (A) Leave-unit-out R^2 versus number of groups. For each run, performance was evaluated on 100 held-out test trials. The performance curves for each method are visually indistinguishable from one another. Black dashed line: noise ceiling ($R^2 = 0.167$ for these simulations). (B) Mean (across fitting iterations) clock time per iteration versus number of groups. Gray dashed line: reference slope for a linear scaling law, $O(M)$. Gray dotted line: reference slope for a cubic scaling law, $O(M^3)$. (C) Number of iterations to convergence versus number of groups. (D) Total clock time to convergence versus number of groups. Across panels A to D, solid traces indicate the mean, and shading (where visible) indicates standard error of the mean, computed across 20 runs. Color scheme is consistent across panels. Black: mDLAG-time (Time); green: mDLAG-inducing (Induc.); magenta: mDLAG-frequency (Freq.).

mDLAG-time, black trace, to the cubic reference slope, gray dotted line), while both mDLAG-frequency and mDLAG-inducing exhibited linear scaling or better (Figure 4B: compare mDLAG-frequency, magenta trace, and mDLAG-inducing, green trace, to the linear reference slope, gray dashed line). Fitting with mDLAG-frequency and mDLAG-inducing

thus produced a 285× and a 88× speed-up per iteration, respectively, at $M = 24$ (mean clock time per iteration: mDLAG-time, 7.200 ± 0.091 s; mDLAG-inducing, 0.0816 ± 0.0013 s; mDLAG-frequency, 0.0252 ± 0.0004 s). The number of iterations required for mDLAG-time convergence also increased with the number of groups (Figure 4C, number of iterations at $M = 24$: mDLAG-time, $15,770 \pm 387$; mDLAG-inducing, 273 ± 11; mDLAG-frequency, 240 ± 10). The computational advantages of mDLAG-frequency and mDLAG-inducing thus compounded, resulting in overall runtime speed-ups of over 18,000× and 5,000× for mDLAG-frequency and mDLAG-inducing, respectively, at $M = 24$ (Figure 4D, total clock time: mDLAG-time, $113,087 \pm 1,987$ s; mDLAG-inducing, 22.20 ± 0.80 s; mDLAG-frequency, 6.05 ± 0.29 s).

4.4 Neuropixels Recordings of Three Visual Cortical Areas. We next sought to validate our accelerated methods beyond a simulated environment on state-of-the-art neural recordings. We considered electrophysiological recordings from neuronal populations in anesthetized macaque visual cortex, recorded using multiple Neuropixels probes (see section 6.4). These recordings encompassed hundreds of neurons spanning three brain areas: V1, V2, and V3d (V1: 55–191 neurons across recording sessions; V2: 63–166; V3d: 8–111). Each brain area was treated as a separate group throughout our analyses below (i.e., $M = 3$). Within a recording session, on each trial, animals were visually presented with a drifting sinusoidal grating (one of two possible orientations, 90° apart). Each orientation was presented on 300 trials, for a total of 600 trials per recording session.

Overall, we analyzed seven recording sessions from three animals. We further treated each grating stimulus orientation separately, giving a total of 14 "data sets" (a subset of which were previously analyzed in Gokcen et al., 2023). For each data set, we allocated at random 225 trials as a training set and 75 trials as a test set on which to measure model performance (see appendix D). We applied each fitting method (mDLAG-time, mDLAG-inducing, mDLAG-frequency) to spike counts (taken in 20 ms nonoverlapping time bins) measured during the first 1,000 ms after stimulus onset (see Figure 5A, trial-averaged responses for an example data set). As a preprocessing step, for each neuron, we subtracted the mean spike count across time bins within each trial to remove slow fluctuations beyond the timescale of a trial (Cowley et al., 2020). We started all fitting methods from the same initialization, with $p = 30$ latents (see section 6.2). In terms of scale, these recordings included many more trials and neurons than the simulations in sections 4.2 and 4.3, but the number of time points per trial ($T = 50$) and groups ($M = 3$) are within the ranges considered above.

mDLAG-frequency performed statistically as well as mDLAG-time (see Figure 5B, magenta points lie on the diagonal) and achieved a median 2.9× speed-up per fitting iteration and a 25× speed-up overall (Figure 5C, median clock time per iteration across data sets—mDLAG-time, 5.88 s;
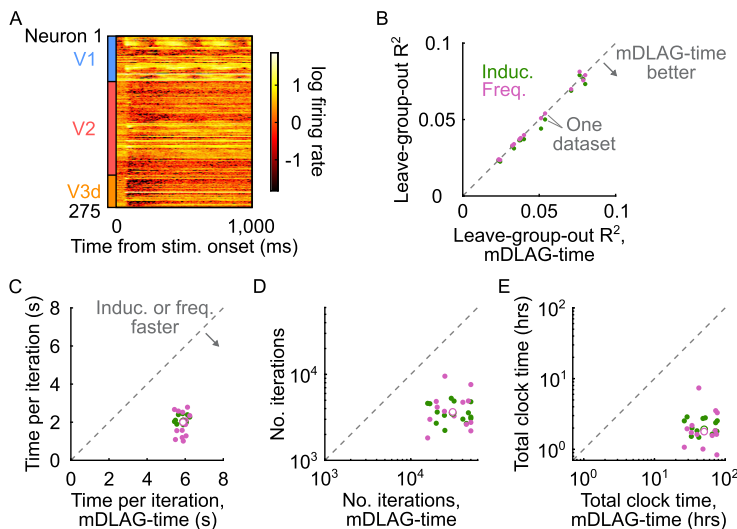
Figure 5: Performance and runtime on Neuropixels recordings from macaque visual cortex. Here $M = 3$ groups, corresponding to the three brain areas V1, V2, and V3d. (A) Temporally smoothed peristimulus time histograms during the stimulus presentation period, for an example session and stimulus condition. (B) Performance (leave-group-out R^2) of mDLAG-inducing (green points; Induc.) or mDLAG-frequency (magenta points; Freq.) versus performance of mDLAG-time. Each data point represents one data set. For each data set, performance was evaluated on 75 held-out test trials. mDLAG-inducing (with $T_{ind} = 20$ inducing points) was significantly outperformed by mDLAG-time (one-sided paired sign test: mDLAG-time better than mDLAG-inducing, $p = 0.0065$), whereas mDLAG-frequency performed as well as mDLAG-time (one-sided paired sign test: mDLAG-time better than mDLAG-frequency, $p = 0.7880$). (C) Mean clock time per iteration, mDLAG-inducing or mDLAG-frequency versus mDLAG-time. As in panel B, each data point represents one data set. White-filled circles indicate the median of the magenta points (circle with magenta border) or of the green points (circle with green border, occluded by the magenta median). (D) Number of iterations to convergence, mDLAG-inducing or mDLAG-frequency versus mDLAG-time. Same conventions as in panel C. Median over green points is occluded by median over magenta points. (E) Total clock time to convergence, mDLAG-inducing or mDLAG-frequency versus mDLAG-time. Same conventions as in panel C.

mDLAG-frequency, 2.00 s; Figure 5D, median number of iterations across data sets: mDLAG-time, 30,737; mDLAG-frequency, 3,659; Figure 5E, median total clock time across data sets: mDLAG-time, 50.4 hr; mDLAG-frequency, 1.8 hr). These speed-ups are of the same order of magnitude as or better than the speed-ups seen in simulation (see section 4.3). For example,

in simulated data with $M = 3$ groups and $T = 50$ time points per trial, mDLAG-frequency achieved a $3.6\times$ speed-up per fitting iteration (see Figure 4B) and a $12\times$ speed-up overall (see Figure 4D) relative to mDLAG-time.

mDLAG-inducing, by contrast, presented a performance trade-off that depended on the choice of the number of inducing points. At $T_{\text{ind}} = 20$ inducing points, mDLAG-inducing achieved a runtime similar to mDLAG-frequency (Figure 5C, median clock time per iteration across data sets: 2.03 s; Figure 5D, median number of iterations across data sets: 3,457; Figure 5E, median total clock time across data sets: 1.9 hr). However, $T_{\text{ind}} = 20$ inducing points corresponds to an effective sampling rate of 20 Hz. At a 20 Hz sampling rate, aliasing becomes a potential issue for latents with (squared exponential) GP timescales of about 50 ms or shorter. For instance, by mDLAG-time estimates, about 25% of the GP timescales encountered across Neuropixels data sets were 35 ms or shorter. By contrast, no mDLAG-inducing timescale estimates were shorter than 35 ms (see supplementary Figure 3).

Models fit by mDLAG-inducing consequently performed significantly worse than models fit by the other two methods (see Figure 5B, one-sided paired sign tests, leave-group-out R^2: mDLAG-time better than mDLAG-inducing, $p = 0.0065$; mDLAG-frequency better than mDLAG-inducing, $p = 0.0009$). This gap in mDLAG-inducing's statistical performance could be closed by increasing the number of inducing points (see supplementary Figure 4A: mDLAG-inducing performance when we chose $T_{\text{ind}} = 32$ inducing points, corresponding to an approximate Nyquist rate for GP timescales of 30 ms or longer), but at the expense of runtime benefit (see supplementary Figure 4B: mDLAG-frequency 55% faster than mDLAG-inducing per iteration, and 72% faster overall).

Finally, we sought to characterize further the extent to which mDLAG-frequency for these neural recordings maintained the runtime scaling properties exhibited in simulation (see Figures 3 and 4). Importantly, mDLAG-frequency maintained linear scaling or better with both the number of time points per trial (see supplementary Figure 5, up to $T = 500$) and the number of analyzed groups (see supplementary Figure 6, up to $M = 7$, where each group represents a distinct laminar compartment of sampled V1, V2, and V3d recordings).

4.5 Exploration of Biases Introduced by Frequency Domain Fitting.
Thus far, we have shown that the frequency domain fitting approach provides orders of magnitude runtime speed-up with seemingly minimal loss in statistical performance relative to baseline. Of course, no approximation comes without a cost. We now explore the biases introduced by the frequency domain approach, particularly in the estimation of GP parameters and latent dimensionality.

But first, why might there be biases in the first place? The approximation to take the frequency domain PSD matrix as diagonal is equivalent to
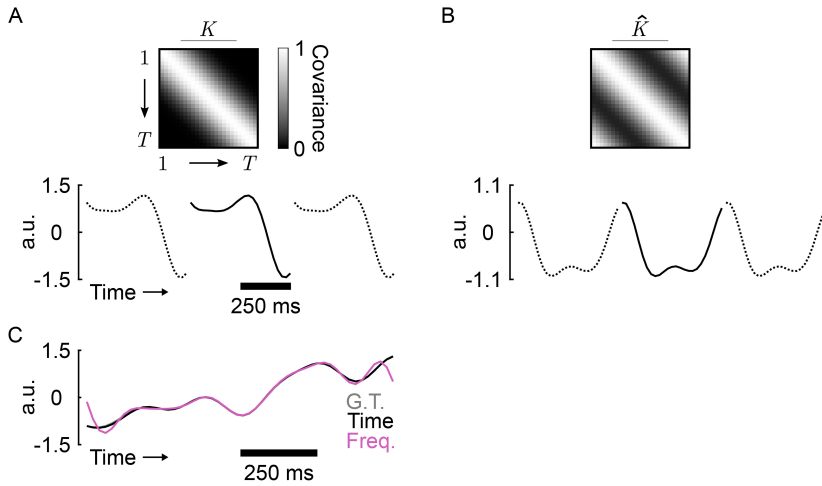
Figure 6: The circulant approximation induced by the frequency domain approach and its effects on generation and inference. (A) Generation via the time domain. Top: An example gaussian process (GP) covariance matrix (K), corresponding to the mDLAG-time model. Bottom: An example GP time course (black solid trace) generated via the mDLAG-time model. To emphasize the (lack of) boundary conditions, copies of that time course are displayed to its left and right (black dotted traces). (B) Generation via the frequency domain. Top: The circulant approximation of K in panel A ($\hat{K}$), arising implicitly from the mDLAG-frequency model. Both K and $\hat{K}$ shown here were generated with the same squared exponential GP timescale (100 ms) for trial length $T = 25$ time points and $M = 1$ group. Bottom: An example GP time course (black solid trace) generated via the mDLAG-frequency model. The signal was generated in the frequency domain and then transformed into the time domain via the discrete Fourier transform (DFT). To emphasize the periodic boundary conditions, copies of that time course are displayed to its left and right (black dotted traces). (C) Inference via the time domain versus the frequency domain. A ground-truth latent time course (G.T., gray; occluded by the black trace) was generated via the mDLAG-time model (trial length $T = 50$ time points, $M = 1$ group, and GP timescale 100 ms). Then, time courses were inferred via the mDLAG-time model (Time, black; equation 3.13) or the mDLAG-frequency model (Freq., magenta; equation 3.32, followed by DFT). In either case, the ground-truth parameters were used for inference. a.u.: arbitrary units.

replacing the time domain covariance matrix with a circulant matrix (see Figure 6A, top: time domain covariance matrix; Figure 6B, top: circulant approximation arising implicitly from the mDLAG-frequency model). This circulant approximation affects both the generative model and posterior inference. Latent time courses generated via the approximate frequency

domain model (equation 3.25, followed by an inverse DFT) are putatively periodic (Figure 6B, bottom: note the continuity between the solid trace and its copies, the dotted traces), with correlations between the beginning of the trial and the end of the trial (Figure 6B, top: note the large magnitude of covariance in the upper right and lower left quadrants of the circulant covariance matrix $\hat{K}$).

Like frequency-domain-generated time courses, frequency-domain-inferred time courses are similarly periodic, leading to edge effects (Figure 6C). In the middle of a trial, time domain and frequency domain inference produce practically indistinguishable estimates (Figure 6C: compare black and magenta traces). However, the estimates start to visibly deviate toward the beginning and end of the trial as the frequency domain estimate curves away from the time domain estimate to satisfy its periodic boundary conditions. The extent of edge effects, in both generation and inference, increases as the length of the GP timescale increases relative to the length of the trial.

4.5.1 Biases in Estimation of Gaussian Process Parameters. To characterize the impact of the frequency domain approximation on GP parameter estimation, we returned to our simulated data sets from above. Here we focus on bias as a function of trial length (see Table 1, Scaling, T), as the departure of the frequency domain approximation from the original time domain specification is ultimately a finite trial length effect. We also explore biases as a function of group number (see supplementary Figure 7A), sampling rate (supplementary Table 1, Sampling rate; supplementary Figure 7B), and number of units (supplementary Table 1, Scaling, q; supplementary Figure 7C).

Comparing the GP timescales and time delays estimated via mDLAG-frequency to the ground-truth values, we found that the magnitudes of both parameters were consistently underestimated (see Figures 7A and 7B, magenta traces). Biases were more severe for shorter trial lengths and approached the ground truth for longer trial lengths. In separate analyses, we refit models via mDLAG-frequency while keeping either the GP timescales or the GP time delays fixed throughout fitting. Regardless of which parameter was held fixed, the bias remained in estimates of the other parameter, suggesting that these biases are decoupled effects (see supplementary Figure 8).

Intuition for this effect can be gained by considering the time domain covariance matrix and its (approximate) circulant counterpart. Consider, for example, a ground-truth latent variable with covariance matrix as depicted in Figure 6A. Consider also a frequency domain (circulant in the time domain) estimate with GP timescale that matches the ground-truth value (Figure 6B, top). If we consider the error between the ground-truth covariance matrix and the circulant estimate, then the error for elements close to the main diagonal will be low, while the error for elements in the
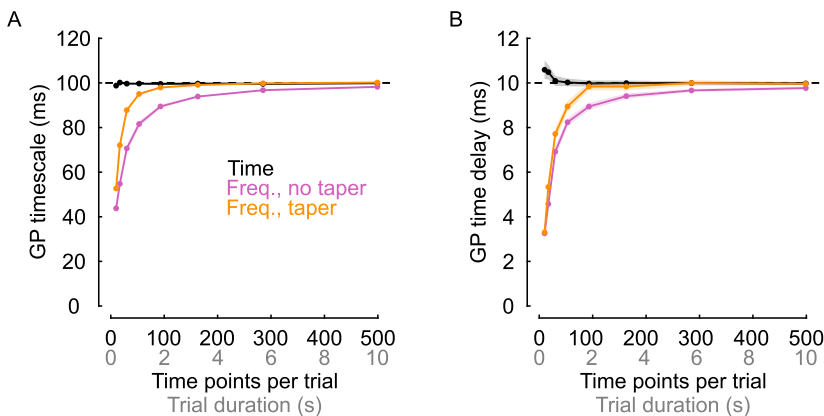
Figure 7: Bias in gaussian process (GP) parameter estimation with number of time points per trial. (A) GP timescale estimates versus number of time points per trial. (B) GP time delay estimates versus number of time points per trial. Across panels A and B, black dashed traces indicate ground-truth parameter values. Solid traces indicate the mean of estimates, and shading (where visible) indicates standard error of the mean, computed across 20 runs. Black: mDLAG-time (Time); magenta: mDLAG-frequency without tapering (Freq., no taper); gold: mDLAG-frequency with tapering (Freq., taper).

upper right and lower left quadrants will be relatively high. The error accumulated in these upper right and lower left quadrants could be reduced if the GP timescale estimate was shorter, but at the expense of added error for elements close to the main diagonal. Ultimately, the optimal timescale balances these sources of error, settling on an estimate that is shorter than the ground truth but not too short. We validated this intuition empirically in simulation (see supplementary Figure 9). Importantly, as the length of the trial increases, the bias approaches zero (see Figures 7A and 7B, magenta traces): the bias becomes less concerning precisely as the runtime of mDLAG-frequency becomes more advantageous (see Figure 3).

Why does this bias not lead to more severe negative impacts on model performance (see Figures 3A, 4A, and 5B)? An underestimate of GP timescale means that higher frequencies are overrepresented a priori. In principle, then, posterior estimates of the latent time courses are encouraged to maintain potentially spurious high-frequency content in held-out test trials, leading to overfitting. In practice, however, given the consistently similar performance between mDLAG-frequency and mDLAG-time, overfitting was evidently not an issue for the simulated data sets and neural recordings considered here. An overestimate of GP timescale, by contrast, would encourage underfitting, as meaningful high-frequency content is suppressed. For the neural recordings considered here, that type of

underfitting appeared to be the more salient issue, as evidenced by the relatively poor performance of mDLAG-inducing model fits (see Figure 5B) paired with their consistent overestimates of GP timescales (see supplementary Figure 3).

4.5.2 Biases in Estimation of Dimensionality. We next explored potential biases in the estimation of latent dimensionality, that is, model selection. In the case of methods like GPFA (Yu et al., 2009) and DLAG (Gokcen et al., 2022), model selection is accomplished via cross-validation and grid search to find optimal latent dimensionalities. In the case of mDLAG, estimation of latent dimensionality is incorporated into the model fitting process via the ARD prior, which encourages group-wise sparsity of the loadings (equations 3.5 and 3.6). Here we focus on estimation of latent dimensionality via ARD (see section 6.3).

As we will demonstrate, the results of model selection are noise-level dependent. We therefore generated (via the mDLAG generative model; see section 6.1) 10 independent data sets at four noise levels (SNR = 0.01, 0.1, 1.0, 10.0), for a total of 40 data sets (see Table 1, Model Selection). Each data set comprised $N = 50$ trials, with 200 time points per trial (400 ms in length). Here we were also interested in bias as a function of trial length, and thus fit mDLAG models to increasingly long trial epochs (lengths spaced equally on a log scale from 10 to 200 time points per trial). For simplicity, we generated activity for just one group with 24 units. The themes from model selection for one group are representative of themes that underlie model selection in multigroup contexts. We set the ground-truth number of latents to $p = 4$, but initialized all models with $p = 8$ latents to allow dimensionality to be freely estimated. All ground-truth GP timescales were set to 50 ms.

We first considered trends in dimensionality estimates via mDLAG-time as a baseline (see Figure 8A). Biases in these estimates depended on trial length. At low SNRs (Figure 8A, red, SNR = 0.01), mDLAG-time produced conservative dimensionality estimates—underestimates that approached the ground truth with increasing trial length (and increasing training data set size). At midlevel SNRs (Figure 8A, gold and magenta, SNR = 0.1, 1.0, respectively), dimensionality estimates were accurate at all trial lengths. But interestingly, at high SNRs (Figure 8A, green, SNR = 10.0), mDLAG-time produced overestimates of dimensionality at long trial lengths (large training data set sizes). This behavior has been reported previously for simpler dimensionality reduction approaches like factor analysis, in conjunction with cross-validation (Williamson et al., 2016): in low-noise, data-rich regimes, the effects of overfitting are small, and cross-validated performance remains high even for models with dimensionalities larger than the ground truth.

We then sought to characterize the extent to which dimensionality estimates via mDLAG-frequency matched or deviated from the trends described above (Figure 8B). At the lowest two SNRs (Figure 8B, red and gold,
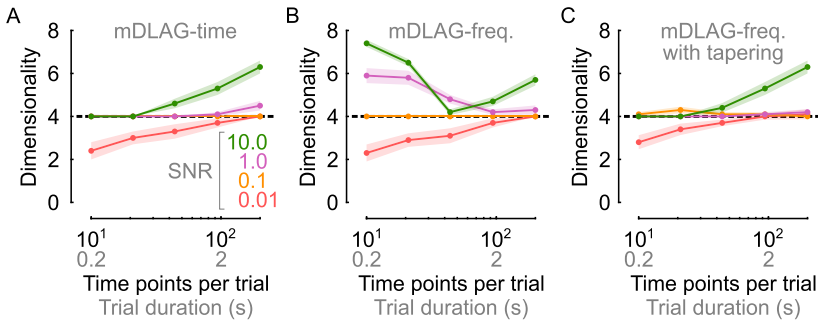
Figure 8: Bias in estimation of latent dimensionality with number of time points per trial. (A) mDLAG-time estimates. (B) mDLAG-frequency (mDLAG-freq.) estimates. (C) mDLAG-frequency estimates with tapering as a preprocessing step. Across panels A to C, black dashed lines indicate ground-truth dimensionality ($p = 4$). Solid traces indicate the mean of estimates, and shading (where visible) indicates standard error of the mean, computed across 10 runs. SNR: Signal-to-noise ratio. Green: SNR = 10.0; magenta: SNR = 1.0; gold: SNR = 0.1; red: SNR = 0.01. In panels A and C, the gold and magenta curves almost fully coincide along the ground-truth line.

SNR = 0.01, 0.1, respectively), mDLAG-frequency estimates were qualitatively indistinguishable from estimates by mDLAG-time (compare to Figure 8A). At the highest two SNRs (Figure 8B, magenta and green, SNR = 1.0, 10.0, respectively), trends for the two fitting methods were notably different, at least for shorter trial lengths ($T < 44$). In contrast with mDLAG-time estimates, mDLAG-frequency estimates were biased toward dimensionality values that were larger than the ground truth at shorter trial lengths and approached the ground truth as trial length increased (up to $T = 44$). Then as trial length continued to grow, the trends in mDLAG-frequency estimates matched the trends of mDLAG-time estimates, growing in magnitude above the ground-truth value (for $T = 44$ and larger).

The trend for short trial lengths can be explained by the bias in GP timescale estimation (see section 4.5.1). mDLAG-frequency tends to produce underestimates of GP timescales at short trial lengths—a mismatch with the data. By employing additional latents, mDLAG-frequency could possibly compensate for any explanatory power lost by this parameter bias. As trial length increases, the bias in GP timescale decreases; mDLAG-frequency estimates behave increasingly like mDLAG-time estimates.

4.5.3 Bias Mitigation Strategies. We have demonstrated empirically that the biases explored above have a minimal impact on model performance and that such biases diminish as the runtime benefit of mDLAG-frequency grows. Still, it might be of practical and scientific importance to mitigate

these biases during the model-fitting process. Bias mitigation is a mature topic of study in spectral estimation, with many strategies to choose from (Priestley, 1981). To demonstrate that this body of work is applicable to the present context, we considered one such strategy: tapering.

We repeated all simulated experiments with mDLAG-frequency (see Figures 7, 8, and supplementary Figure 7). Here, however, as a preprocessing step, we applied a taper (we chose a periodic Hamming window function, see section 6.5) to each observed unit on each trial. Tapering smoothly attenuates each observed time series toward zero at the beginning and end of each trial, effectively enforcing the periodic boundary conditions assumed by the frequency domain approach.

Across experiments, tapering was an effective bias mitigation strategy. For instance, in trial length experiments, mDLAG-frequency (without tapering) achieved within 10% error of the ground-truth GP timescale by $T = 163$ (Figure 7A, magenta). With tapering, that error was achieved by $T = 53$ (Figure 7A, gold). In model selection experiments (Figure 8C), tapering removed the tendency of mDLAG-frequency to overestimate dimensionality at short trial lengths (compare Figure 8C to Figure 8B, for trial lengths $T = 44$ and shorter). Tapering was also effective for bias mitigation in our experiments scaling group number (supplementary Figure 7A), sampling rate (supplementary Figure 7B), and number of neurons (supplementary Figure 7C).

As an alternative to tapering, one could use mDLAG-frequency to fit an initial set of model parameters and then fine-tune those parameters using mDLAG-time for a limited number of iterations. For example, we employed this approach on one of the Neuropixels data sets (see supplementary Figure 10). After 500 iterations of fine-tuning with mDLAG-time, GP timescales and time delays settled into slightly different values from the initial mDLAG-frequency estimates.

5 Discussion

In this work, we developed two methods to accelerate model fitting and inference in multigroup GP factor models. Our methods incorporate multigroup extensions of techniques drawn from two broader classes of approaches in the GP and spectral estimation fields: sparsity via inducing variables (Álvarez et al., 2010; Duncker & Sahani, 2018; Titsias, 2009) and frequency domain representations (Dowling et al., 2023; Keeley, Aoi et al., 2020; Paciorek, 2007; Ulrich et al., 2015; Whittle, 1951). We demonstrated that mDLAG-inducing and mDLAG-frequency can achieve linear scaling in both the number of time points per trial T and the number of groups M, translating to orders of magnitude runtime speed-ups over the baseline approach, mDLAG-time, across simulations and neural recordings. Notably, in neural recordings of hundreds of neurons across three brain areas, mDLAG-frequency produced a 25× speed-up over mDLAG-time

without compromising statistical performance. Prior to this work, the analysis of increasingly available recordings that span many brain areas, cortical layers, and cell types was prohibitive for the class of multigroup GP factor models. With the methods developed here, that analysis is now feasible, opening the door to questions about multipopulation interactions that were previously inaccessible.

In contrast with mDLAG-frequency, mDLAG-inducing often presented a trade-off between runtime and statistical performance. Applied to neural recordings, mDLAG-inducing could achieve either equal runtime performance to mDLAG-frequency (see Figure 5) or equal statistical performance to mDLAG-frequency (see supplementary Figure 4), but not both. In theory, mDLAG-inducing scales linearly in T so long as the number of inducing points T_{ind} is held fixed. In practice, T_{ind} must be kept large enough to avoid aliasing and excessively overestimating latent timescales (see Figure 2B, supplementary Figure 2, supplementary Figure 3), hence tempering the runtime benefit of the approach (see Figures 2D and 3). The need to search for the optimal hyperparameter T_{ind}, when the (potentially wide-ranging) latent timescales are not known a priori tempers that runtime benefit further.

Of course, mDLAG-frequency is not without trade-offs. For small trial lengths, we demonstrated that mDLAG-frequency exhibits a tendency to underestimate the magnitudes of GP timescales and time delays (see Figure 7 and supplementary Figure 3), and a tendency to overestimate dimensionality (Figure 8B). Both of these biases could leave models fit via mDLAG-frequency prone to overfitting. For the simulated data and neural recordings considered here, however, overfitting did not appear to be an issue. Importantly, as trial length increases, mDLAG-frequency's bias diminishes (see Figures 7 and 8) precisely as its runtime benefit grows (see Figure 3). For instance, in the simulations with $T = 500$ time points per trial, a 1000× runtime speed-up (see Figure 3D) with less than 2% bias in GP parameter estimates (see Figure 7) represents a remarkably good trade-off.

We explored tapering and fine-tuning in the time domain as straightforward yet effective bias mitigation strategies (see section 4.5.3). While these strategies worked well for the simulated and neural data sets considered here, they might not always be the best strategy for the problem at hand. In other contexts, other strategies could be explored, such as zero-padding (Aoi & Pillow, 2017) or even debiased modifications to the variational lower bound (Sykulski et al., 2019), though likely at the expense of increased computational cost.

In our Neuropixels analyses (see section 4.4), neurons were straightforwardly grouped by brain area. In other scenarios, delineating groups (e.g., by layer or cell type) might be challenging. Control analyses are critical for validating interactions across the chosen groups. Within-group controls (Gokcen et al., 2022; Semedo et al., 2019) help confirm that the across-group interactions of interest are meaningful. Analyses based on

random subsamples of neurons also serve as valuable controls (Gokcen et al., 2022). For example, given two groups of neurons, one could randomly subsample neurons from group 1 and then randomly subsample neurons from group 2. mDLAG models could then be refit to these two subselected groups. Highly inconsistent results across repetitions of this procedure might suggest the need to reevaluate group definitions.

Future work could adapt mDLAG for group discovery (Buesing et al., 2014; Klami et al., 2015). When groups are unknown a priori, one could forsake assigning groups altogether and take each individual neuron to be its own "group." Then one could study the fitted mDLAG models in an exploratory fashion to search for meaningful group structure. This analysis, previously computationally prohibitive with mDLAG-time, is now feasible with mDLAG-frequency or mDLAG-inducing (see, e.g., Figure 4, at $M = 24$ groups).

We have focused here on approaches to accelerating one particular dimensionality reduction method, mDLAG; but in fact, mDLAG is representative of a general class of dimensionality reduction methods. Our approaches are therefore quite general, applicable to a wide range of dimensionality reduction methods commonly used in neuroscience. In the case of two groups ($M = 2$), mDLAG is equivalent to a Bayesian formulation of DLAG (Gokcen et al., 2022) and closely related to other multigroup approaches that incorporated GP state models (Balzani et al., 2023; Keeley, Aoi et al., 2020). In the case of one group ($M = 1$) and when all time delays are fixed to zero ($D_j^m = 0$), mDLAG becomes equivalent to a Bayesian formulation of gaussian process factor analysis (GPFA; Duncker & Sahani, 2018; Jensen et al., 2021; Yu et al., 2009). When each group comprises one unit ($M = q$), as noted above, mDLAG resembles a time-delay GPFA (Lakshmanan et al., 2015) or cross-spectral factor analysis (Gallagher et al., 2017) model with a sparsity prior on each coefficient of the loading matrix, C. Finally, by removing temporal smoothing (i.e., in the limit as all GP timescale parameters τ_j approach 0) mDLAG becomes equivalent to the static method group factor analysis (Klami et al., 2015).

The accelerated fitting methods developed here, in the Bayesian mDLAG setting, are readily applicable to non-Bayesian methods like GPFA and DLAG. For these non-Bayesian methods, development would largely follow sections 3.2 and 3.3, but the mean parameter $\mathbf{d}^m$, the loading matrix C^m, and the noise precision matrix Φ^m would be treated as deterministic model parameters for which we seek point estimates rather than probabilistic model parameters for which we seek posterior distributions. In fact, we include an implementation of these methods in accompanying code (see appendix E). We note also that we restricted our development here to models with a linear-gaussian relationship between latents and observations (equations 3.1 and 3.2). Our approaches could be adapted to nonlinear regimes (Duncker et al., 2023; Gondur et al., 2024; Wu et al., 2017) and non-gaussian noise models (Duncker & Sahani, 2018; Keeley,

Zoltowski, Yu et al., 2020; Zhao & Park, 2017), though again likely at the expense of increased computational cost.

Our investigations focused on the commonly used squared exponential GP covariance function with time delays (equation 3.8). We emphasize, however, that all three fitting methods examined here are compatible with any stationary GP covariance function, while possessing the same computational scaling properties (see Figures 3 and 4). The mDLAG-frequency mathematical development, specifically, need not depend on the particular parameterization of the GP cross-spectral density (CSD) function given in equation 3.26. Rather, any GP CSD function that can be written in terms of an amplitude function and a phase function is compatible with the approach. Concretely, for latent j and groups m_1 and m_2, we can write the GP CSD function $s_{m_1,m_2,j}(f_l)$, evaluated at frequency f_l, in terms of the amplitude function $\zeta_j(f_l)$ and phase function $\eta_{m_1,m_2,j}(f_l)$:

$$s_{m_1,m_2,j}(f_l) = \zeta_j(f_l) \cdot \exp\left(-i\eta_{m_1,m_2,j}(f_l)\right). \tag{5.1}$$

For the squared exponential GP covariance function (equation 3.8), the corresponding amplitude function $\zeta_j(f_l)$ is the squared exponential in equation 3.24. The phase function is a linear function of frequency, $\eta_{m_1,m_2,j}(f_l) = 2\pi f_l(D_j^{m_1} - D_j^{m_2})$, where the slope is determined by the difference between time delay parameters $D_j^{m_1}$ and $D_j^{m_2}$. Alternative choices for the amplitude function can be drawn directly from the single-output GP literature (Wilson & Adams, 2013). In tandem, alternative choices of the phase function would lead to alternative multi-output GP kernels, for instance, constant (as in a simple phase shift; Ulrich et al., 2015) or affine (Parra & Tobar, 2017) functions of frequency. Different choices of GP kernel would lead to different behavior in terms of statistical performance and biases in estimation. These differences ought to be explored on a case-by-case basis. Still, the themes raised in section 4.5 will be relevant to any alternative GP kernel choices.

Multigroup dimensionality reduction methods have historically faced a number of scaling challenges, from model selection to model fitting. The challenge of model selection requires identifying both the number of latents across all observed groups and which subset of groups each latent involves. For example, in the case of $M = 2$ groups, the non-Bayesian DLAG (Gokcen et al., 2022), and methods with similar group structure (e.g., Balzani et al., 2023; Keeley, Aoi et al., 2020), has three hyperparameters: one for the number of shared latents and two for the number of latents local to each group. Grid search over just 10 candidate values for each of these three hyperparameters would result in 10^3 candidate models to fit, and k-fold cross-validation would inflate that number further. To generalize this model selection approach to M groups would require the fitting of $p^{2^M-1}k$ candidate models, for a search over p candidate values for each of $2^M - 1$ hyperparameters with k-fold cross-validation. With the incorporation of automatic

relevance determination (Klami et al., 2015), mDLAG reduced the number of model fits to one. The methods developed here then accelerate that model fit to overcome remaining computational barriers, thereby enabling a broad class of dimensionality reduction techniques to keep pace with the rapidly growing scale of multipopulation neural recordings.

6 Methods

6.1 Synthetic Data Generation via the Time and Frequency Domains. Throughout our simulated experiments (see Table 1 and supplementary Table 1), we generated the simulated data sets using the mDLAG generative model. In section 4.1 (see Figure 2 and Table 1, Demo), generation could be performed straightforwardly via the mDLAG-time formulation, equations 3.1 to 3.9. For the remaining larger-scale experiments (see Table 1: Scaling, T; Scaling, M; Model selection; see also supplementary Table 1: Sampling rate; Scaling, q), data generation via the mDLAG-time formulation was prohibitively expensive. Sampling from the multivariate gaussian distribution in equation 3.7 requires $\mathcal{O}(M^3T^3)$ operations, cubic in the number of groups, M, and the number of time points per trial, T.

For these remaining experiments, we therefore generated simulated data sets using the mDLAG-frequency formulation, equations 3.24–3.25, 3.28–3.29, and 3.3–3.6 (Dietrich & Newsam, 1997). The computational cost of sampling frequency domain observations is ameliorated to $\mathcal{O}(MT)$ operations. Then we convert these observations to the time domain by taking the inverse unitary DFT of the frequency domain activity for each unit $r = 1, \ldots, q_m$ in group m on each trial n: $\mathbf{y}_{n,r,:}^m = U_T^H \tilde{\mathbf{y}}_{n,r,:}^m$. Assuming the DFT is carried out using an FFT algorithm, this operation scales $\mathcal{O}(T \log T)$ in the number of time points per trial.

Yet time courses generated via mDLAG-frequency exhibit edge effects (see section 4.5 and Figure 6B). To mitigate the influence of these edge effects on our experimental results, we conservatively first generated trials that were three times longer than desired ($3T$ time points in length, when we desired length-T trials). Then we took only the middle third of each trial and threw out the first and final thirds. For our simulated experiments, in which latent time courses were generated with a squared exponential GP timescale of 100 ms, taking only the middle third of the trial was sufficient for edge effects to be rendered negligible. Even with the generation of extra time points per trial, data generation via the mDLAG-frequency formulation was still markedly faster than data generation via the mDLAG-time formulation.

6.2 Parameter Initialization. We initialized each of the three methods, mDLAG-time, mDLAG-inducing, and mDLAG-frequency, in a similar fashion. For any of the methods, we first specified the number of latents, p. Through automatic relevance determination, insignificant latents

are then effectively pruned. Therefore, in general, p should be chosen to be as small as possible (to minimize runtime) yet large enough that at least one of the initial latents is deemed insignificant (according to shared variance explained; see equation 6.2), thereby ensuring that dimensionalities are not underestimated. We indicate our chosen value of p for each experiment throughout section 4.

To initialize the rest of the fitting procedure, we specified initial values for only the moments of the posterior factors $Q_d(\mathbf{d})$, $Q_\phi(\phi)$, $Q_c(C)$, and $Q_\mathcal{A}(\mathcal{A})$ (see, e.g., equations B.18–B.21) that were needed to begin iteration. The posterior distribution over the latents $Q_x(X)$ (for mDLAG-time, equation 3.13; $Q_w(W)$ over inducing variables for mDLAG-inducing, equation B.17; $Q_{\tilde{x}}(\tilde{X})$ over frequency domain latents for mDLAG-frequency, equation 3.32) 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 (Klami et al., 2015), $\beta, a_\phi, b_\phi, a_\alpha, b_\alpha = 10^{-12}$.

For $Q_d(\mathbf{d})$, we initialized $\boldsymbol{\mu}_d^m$ at the sample mean of observed activity across all trials and time points. For $Q_\phi(\phi)$, we initialized $\langle \phi_r^m \rangle^{-1}$ for each unit r in group m to the sample variance of that unit across all trials and time points. For $Q_c(C)$, we first randomly initialized all first moments $\bar{\boldsymbol{\mu}}_{c_r}^m \in \mathbb{R}^p$ (for the rth row of C^m in group m, see equation B.36) with entries drawn from a zero-mean gaussian distribution with variance chosen to match the scale of the data. Then we initialized the second moments $\langle \bar{\mathbf{c}}_r^m (\bar{\mathbf{c}}_r^m)^\top \rangle$ to the outer product of first moments $\bar{\boldsymbol{\mu}}_{c_r}^m \bar{\boldsymbol{\mu}}_{c_r}^{m\top}$. For $Q_\mathcal{A}(\mathcal{A})$, we initialized $\langle \alpha_j^m \rangle$ for each latent j in group m to $q_m / \langle \|\mathbf{c}_j^m\|_2^2 \rangle$, which stems from equations B.37 and B.38.

We initialized all delay parameters to zero and all gaussian process timescale parameters to the same value, equal to twice the sampling period or spike count bin width of the neural activity. All gaussian process noise variances were fixed to small values (10^{-3}), as in Gokcen et al. (2023). Finally, for mDLAG-inducing specifically, we specified the number of inducing points T_{ind} and fixed the inducing points on a uniformly spaced grid, with the first and last inducing points fixed at the beginning and end of each trial, that is, $\xi_1 = 1$ and $\xi_{T_{\text{ind}}} = T$. Fitting via all three methods proceeded iteratively, until their respective lower bounds improved from one iteration to the next by less than a present tolerance (here we used 10^{-8}; see appendixes B and C).

6.3 Choosing the Number of Significant Latents in Each Group. mD-LAG incorporates automatic relevance determination (ARD) to automatically determine, during model fitting, both the total number of latents and the subset of groups that each latent involves. We sought an intuitive measure of the significance of each latent variable within a population, postfitting, based on the amount of shared variance each latent explains. The shared variance of latent j in group m is given by $\langle \|\mathbf{c}_j^m\|_2^2 \rangle$, the expected

squared magnitude of the jth column of the loading matrix C^m, with respect to the posterior distribution $Q_c(C)$. Since the total shared variance can vary widely across observed groups, we considered a normalized metric, the fraction of shared variance explained by latent j in group m:

$$\nu_j^m = \frac{\langle ||\mathbf{c}_j^m||_2^2 \rangle}{\sum_{k=1}^p \langle ||\mathbf{c}_k^m||_2^2 \rangle}. \tag{6.1}$$

For small ARD hyperparameters, a_α and b_α (as we have chosen in this work), the fraction of shared variance can equivalently be computed using the estimated ARD parameters (see, e.g., equations B.37 and B.38, appendix B):

$$\nu_j^m \approx \frac{\langle \alpha_j^m \rangle^{-1}}{\sum_{k=1}^p \langle \alpha_k^m \rangle^{-1}}. \tag{6.2}$$

If a latent does not significantly explain activity in a group, then ν_j^m will be close to zero. Throughout this work, across all methods considered, we reported a latent as significant in a group if it explained at least 2% of the shared variance within that group ($\nu_j^m \geq 0.02$).

6.4 Neural Recordings. Animal procedures have been reported in previous work (Smith & Kohn, 2008; Zandvakili & Kohn, 2015). Briefly, animals (*Macaca fascicularis*, 2–5 years old) were anesthetized with ketamine (10 mg kg^{-1}) and maintained on isoflurane during surgery. All recordings were performed under sufentanil (6-18 μg kg^{-1} hr^{-1}) anesthesia. Vecuronium bromide (150 μg kg^{-1} hr^{-1}) was used to prevent eye movements. All procedures were approved by the Institutional Animal Care and Use Committee of the Albert Einstein College of Medicine.

Recordings were performed using up to four Neuropixels 1.0 (IMEC, Belgium) probes spaced primarily in the mediolateral direction, with the most anterior probe placed roughly 3 mm posterior to the lunate sulcus. After the probes were lowered, the brain surface was covered with agar or Dura-Gel (Cambridge Neurotech LTD, U.K.) to prevent desiccation. We carried out seven recording sessions from three animals. In one animal, we recorded from both hemispheres (in separate sessions). For the analysis of each recording session, we excluded neurons that fired fewer than 0.5 spikes s^{-1}, on average, across all trials and all grating orientations, or exhibited a Fano factor greater than 5. Average analyzed population sizes per recording session were 125 for V1 (range 55–191), 124 for V2 (range 63–166), and 53 for V3d (range 8–111).

6.5 Bias Mitigation: Tapering. Tapering is a well-established technique for bias mitigation in spectral estimation (Priestley, 1981). In this work (see section 4.5.3 and Figures 7 and 8C), we used tapering as a preprocessing step

to bring observed time series (in training data only; we left test data unaltered to faithfully measure performance) in line with the periodic boundary conditions assumed by the frequency domain approach. Toward that end, let $v_t \in \mathbb{R}$, $t = 1, \ldots, T$, be a set of weights, to be applied to a time series of length T. We chose the commonly used periodic Hamming window function to compute each weight according to

$$v_t = 0.54 - 0.46 \cos \left(2\pi \cdot \frac{t-1}{T} \right).$$ (6.3)

The act of tapering then involves the multiplication of each weight v_t with each $y_{n,r,t} \in \mathbb{R}$, the observation at time t on trial n of unit r (here, group identity does not matter, and we suppress any dependency). The same weights v_t are used for all trials and units.

We took additional care to preserve the sample mean and variance (over time points and trials) of each unit's activity pre- and post-tapering. Let $\mu_r \in \mathbb{R}$ be the sample mean of the observations for unit r, and let $\sigma_r \in \mathbb{R}_{>0}$ be its standard deviation. Then we apply the taper to normalized observations as follows:

$$y'_{n,r,t} = v_t \cdot \frac{y_{n,r,t} - \mu_r}{\sigma_r}.$$ (6.4)

The sample mean and standard deviation (over time points and trials) of the modified observations $y'_{n,r,t}$, μ'_r and σ'_r, are no longer the same as the original sample mean and standard deviation, μ_r and σ_r. We restore these original statistics with an additional processing step:

$$y''_{n,r,t} = \frac{\sigma_r}{\sigma'_r} \cdot (y'_{n,r,t} - \mu'_r) + \mu_r.$$ (6.5)

Models were then fit via mDLAG-frequency to the fully preprocessed observations $y''_{n,r,t}$.

Appendix A: Mathematical Notation

To disambiguate each variable or parameter across the three models we consider here, we need to keep track of up to four labels that indicate their associated (1) trial, (2) unit (neuron) or latent variable index, (3) sample in time or frequency, or (4) group or subpopulation (see Table 2). We indicate the first three labels with subscripts. Trials are indexed by $n = 1, \ldots, N$. Units (neurons) are indexed by $r = 1, \ldots, q$, and latent variables (latents) are indexed by $j = 1, \ldots, p$. In the time domain, we index a trial of length T samples by $t = 1, \ldots, T$. In the frequency domain, we must also consider T frequency components, but to emphasize the indexing over frequency rather than time, we switch to the subscript $l = 1, \ldots, T$. Where relevant,

Table 2: Data Shape, Indices, and Other Constants.

Symbol	Description
N	Number of trials
n	Index over trials, $n = 1, \ldots, N$
T	Number of time points (or frequencies) per trial
t	Index over time points or inducing points, $t = 1, \ldots, T$
l	Index over frequency components, $l = 1, \ldots, T$
f_l	The real-valued frequency of component l
M	Number of observed groups
m	Index over groups, $m = 1, \ldots, M$
q_m	Number of units (neurons) in group m
q	Total number of units across all groups, $\sum_m q_m$
r	Index over units, $r = 1, \ldots, q_m$
p	Number of latent variables (same for all groups)
j	Index over latent variables, $j = 1, \ldots, p$
i	The imaginary number, $\sqrt{-1}$

we indicate the group (subpopulation) to which a variable or parameter pertains via $m = 1, \ldots, M$ (most commonly a superscript).

Putting these four labels together, we define the observed activity of unit r (out of q_m) in group m at time t on trial n as $y_{n,r,t}^m \in \mathbb{R}$ (see Table 3). Similarly, we define latent j (out of p) in group m at time t on trial n as $x_{n,j,t}^m \in \mathbb{R}$. 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 units observed in group m at time t on trial n as the vector $\mathbf{y}_{n,:,t}^m \in \mathbb{R}^{q_m}$. Similarly, we represent the collection of all p latents in group m at time t on trial n as the vector $\mathbf{x}_{n,:,t}^m \in \mathbb{R}^p$. 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 $\mathbf{y}_{n,:,t}^m$ as $\mathbf{y}_{n,t}^m$.

It is conceptually helpful to understand the above notation as taking cross-sections of three-dimensional arrays (see Tables 3–5). For example, observed activity in group m on trial n can be collected into the matrix (two-dimensional array) $Y_n^m = [\mathbf{y}_{n,1}^m \cdots \mathbf{y}_{n,T}^m] \in \mathbb{R}^{q_m \times T}$. Hence each $\mathbf{y}_{n,t}^m$ is a column of Y_n^m. Then we can form the three-dimensional array Y^m by concatenating the matrices $Y_1^m, \ldots, Y_N^m$ across trials along a third dimension. Similarly, the latents in group m on trial n can be collected into the matrix $X_n^m = [\mathbf{x}_{n,:,1}^m \cdots \mathbf{x}_{n,:,T}^m] \in \mathbb{R}^{p \times T}$. We represent a row of X_n^m (i.e., the values of a single latent j at all time points on trial n) as $\mathbf{x}_{n,j,:}^m \in \mathbb{R}^T$. Then we can form the three-dimensional array X^m by concatenating the matrices $X_1^m, \ldots, X_N^m$ across trials along a third dimension.

For all variables with a frequency domain counterpart, we indicate that counterpart using a tilde (see Table 5). The complex-valued $\widetilde{y}_{n,r,l}^m \in \mathbb{C}$ is thus the activity of unit r in group m at frequency l on trial n. Similarly, $\widetilde{x}_{n,j,l}^m \in \mathbb{C}$ is the value of latent j at frequency l on trial n (in the frequency domain,

Table 3: mDLAG-Time and Symbols Common across Methods.

Symbol	Description
Y_n^m	$q_m \times T$ matrix of time domain observations in group m on trial n
$\mathbf{y}_{n,t}^m$	$q_m \times 1$ vector of observations in group m at time t on trial n; the tth column of Y_n^m
$\mathbf{y}_{n,r,:}^m$	$T \times 1$ vector of observations of unit r in group m over time on trial n; the rth row of Y_n^m
$y_{n,r,t}^m$	The activity of unit r in group m at time t on trial n; the rth element of $\mathbf{y}_{n,t}^m$
X_n^m	$p \times T$ matrix of time domain latents in group m on trial n
X_n	The collection, across groups, of time domain latents on trial n, $\{X_n^1, \dots, X_n^M\}$
$\mathbf{x}_{n,:,t}^m$	$p \times 1$ vector of latents in group m at time t on trial n; the tth column of X_n^m
$\mathbf{x}_{n,j,:}^m$	$T \times 1$ vector of values of latent j in group m over time on trial n; the jth row of X_n^m
$x_{n,j,t}^m$	The value of latent j in group m at time t on trial n; the jth element of $\mathbf{x}_{n,:,t}^m$ and the tth element of $\mathbf{x}_{n,j,:}^m$
C^m	$q_m \times p$ loading matrix for group m
α_j^m	automatic relevance determination (ARD) parameter for group m and latent j
$\mathbf{d}^m$	$q_m \times 1$ mean parameter for group m
ϕ^m	$q_m \times 1$ observation noise precision parameter for group m, $\phi^m = [\phi_1^m \cdots \phi_{q_m}^m]^\top$
β	precision (hyper)parameter of the gaussian prior over each mean parameter $\mathbf{d}^m$; fixed to a small value
a_ϕ, b_ϕ	shape and rate (hyper)parameters, respectively, of the gamma prior over each noise precision parameter ϕ_r^m for unit r in group m; fixed to small values
a_α, b_α	shape and rate (hyper)parameters, respectively, of the gamma prior over each ARD parameter α_j^m for group m and latent j; fixed to small values
D_j^m	GP time delay between group m and latent j
τ_j	GP timescale for latent j
σ_j	GP noise parameter for latent j; fixed to a small value
$K_{m_1,m_2,j}$	$T \times T$ covariance matrix for latent j, between groups m_1 and m_2
$k_{m_1,m_2,j}$	covariance function for latent j, between groups m_1 and m_2

we do not assign latents to each group m; see section 3.3). We collect these frequency domain quantities into structures analogous to their time domain counterparts—for example, $\widetilde{\mathbf{y}}_{n,l}^m \in \mathbb{C}^{q_m}$ as the analog to $\mathbf{y}_{n,t}^m$, defined above.

Appendix B: Posterior Inference and Fitting via mDLAG-Inducing

B.1 Variational Inference. Let Y, X, and W be collections of all observed variables, latent variables, and inducing variables, respectively, across all

Table 4: mDLAG-Inducing.

Symbol	Description
T_{ind}	Number of inducing points per trial
W_n	$p \times T_{\text{ind}}$ matrix of inducing variables (common to all groups) on trial n
$\mathbf{w}_{n,j,:}$	$T_{\text{ind}} \times 1$ vector of inducing variable values for latent j (common to all groups) on trial n; the jth row of W_n
$w_{n,j,t}$	Inducing variable for latent j at inducing point t on trial n; the tth element of $\mathbf{w}_{n,j,:}$
ξ_t	inducing point; the real-valued time at which inducing variable $w_{n,j,t}$ is defined
K_j^w	$T_{\text{ind}} \times T_{\text{ind}}$ inducing variable covariance matrix for latent j
k_j^w	covariance function for the inducing variable of latent j
K_j^{xw}	$MT \times T_{\text{ind}}$ covariance matrix between latent j and its inducing variable
k_j^{xw}	covariance function between latent j and its inducing variable

Table 5: mDLAG-Frequency.

Symbol	Description
$\widetilde{Y}_n^m$	$q_m \times T$ matrix of frequency domain observations in group m on trial n
$\widetilde{\mathbf{y}}_{n,l}^m$	$q_m \times 1$ vector of observations in group m at frequency l on trial n; the lth column of $\widetilde{Y}_n^m$
$\widetilde{\mathbf{y}}_{n,r,:}^m$	$T \times 1$ vector of observations of unit r in group m across frequencies on trial n; the rth row of $\widetilde{Y}_n^m$
$\widetilde{y}_{n,r,l}^m$	The value of unit r in group m at frequency l on trial n; the rth element of $\widetilde{\mathbf{y}}_{n,l}^m$ and the lth element of $\widetilde{\mathbf{y}}_{n,r,:}^m$
$\widetilde{X}_n$	$p \times T$ matrix of frequency domain latents (common to all groups) on trial n
$\widetilde{\mathbf{x}}_{n,:,l}$	$p \times 1$ vector of latents (common to all groups) at frequency l on trial n; the lth column of $\widetilde{X}_n$
$\widetilde{\mathbf{x}}_{n,j,:}$	$T \times 1$ vector of values of latent j (common to all groups) across frequencies on trial n; the jth row of $\widetilde{X}_n$
$\widetilde{x}_{n,j,l}$	The value of latent j at frequency l on trial n; the jth element of $\widetilde{\mathbf{x}}_{n,:,l}$ and the lth element of $\widetilde{\mathbf{x}}_{n,j,:}$
$\widetilde{\mathbf{d}}_l^m$	shorthand for $\delta_{l-1} \cdot \sqrt{T}\mathbf{d}^m$: $\sqrt{T}\mathbf{d}^m$ for the zero frequency, and $\mathbf{0}$ otherwise
H_j^m	$T \times T$ diagonal phase-shift operator matrix, from latent j to group m
H_l^m	$p \times p$ diagonal phase-shift operator matrix at frequency l, from latents to group m
S_j	$T \times T$ diagonal power spectral density (PSD) matrix for latent j
S_l	$p \times p$ diagonal PSD matrix at frequency l
s_j	PSD function for latent j
U_T	$T \times T$ unitary discrete Fourier transform matrix

time points (or inducing points) and trials. Similarly, let $\mathbf{d}, \phi, C, \mathcal{A}, \tau$, and D be collections of the mean parameters, noise precisions, loading matrices, ARD parameters, GP timescales, and time delays, respectively. From the observed activity, we seek to estimate posterior distributions over X, W, and the remaining probabilistic model components,

$$\theta = \left\{ \mathbf{d}, \ \phi, \ C, \ \mathcal{A} \right\}, \tag{B.1}$$

and point estimates of the deterministic GP parameters $\Omega = \{\tau, \ D\}$.

In the case of methods like GPFA (Yu et al., 2009) and DLAG (Gokcen et al., 2022), the linear-gaussian structure of the model enables an exact expectation-maximization (EM) algorithm. With the introduction of prior distributions over model parameters (to enable automatic relevance determination), the family of mDLAG models, including mDLAG-inducing, loses this property. The complete likelihood of the mDLAG-inducing model (see Figure 1C),

$$P(Y, X, W, \theta | \Omega) = P(\mathbf{d})P(\phi)P(C|\mathcal{A})P(\mathcal{A})P(Y|X, C, \mathbf{d}, \phi)P(X|W, \Omega)P(W|\tau)$$

$$= \prod_{m=1}^{M} \left[P(\mathbf{d}^m) \left[\prod_{r=1}^{q_m} P(\phi_r^m) \right] \left[\prod_{j=1}^{p} P(\mathbf{c}_j^m \mid \alpha_j^m)P(\alpha_j^m) \right] \right.$$

$$\left. \cdot \left[\prod_{n=1}^{N} \prod_{t=1}^{T} P(\mathbf{y}_{n,t}^m \mid \mathbf{x}_{n,t}^m, C^m, \mathbf{d}^m, \phi^m) \right] \right]$$

$$\cdot \prod_{n=1}^{N} \prod_{j=1}^{p} \left[P(\mathbf{x}_{n,j,:} \mid \mathbf{w}_{n,j,:}, \tau_j, \{D_j^m\}_{m=1}^{M})P(\mathbf{w}_{n,j,:} \mid \tau_j) \right], \tag{B.2}$$

is no longer gaussian. Then a hypothetical EM E-step (evaluation of the posterior distribution $P(X, W, \theta | Y, \Omega)$) becomes prohibitive, as it relies on the analytically intractable marginalization of equation B.2 with respect to X, W, and θ.

We therefore employ a variational inference scheme (Bishop, 1999; Gokcen et al., 2023; Klami et al., 2015), in which we maximize the lower-bound $L(Q, \Omega)$ to the log-likelihood $\log P(Y)$, with respect to the approximate posterior distribution $Q(X, W, \theta)$ and the deterministic parameters Ω:

$$\log P(Y) \geq L(Q, \Omega) = \mathbb{E}_Q[\log P(Y, X, W, \theta | \Omega)] - \mathbb{E}_Q[\log Q(X, W, \theta)]. \tag{B.3}$$

We constrain $Q(X, W, \theta)$ so that it factorizes over X, W, and the elements of θ:

$$Q(X, W, \theta) = Q_{xw}(X, W)Q_d(\mathbf{d})Q_\phi(\phi)Q_c(C)Q_\mathcal{A}(\mathcal{A}). \tag{B.4}$$

We then follow Titsias (2009), and further constrain $Q_{xw}(X, W)$, the joint approximate posterior distribution over the latents and their inducing variables, to factorize as $Q_{xw}(X, W) = P(X|W, \Omega)Q_w(W)$, with generic distribution $Q_w(W)$ over the inducing variables and the conditional prior distribution $P(X|W, \Omega)$ (equation B.2) over the latents themselves. The final constrained form of $Q(X, W, \theta)$ is thus

$$Q(X, W, \theta) = P(X|W, \Omega)Q_w(W)Q_d(\mathbf{d})Q_\phi(\phi)Q_c(C)Q_A(\mathcal{A}). \tag{B.5}$$

This particular choice of factorization leads to an important simplification of the lower-bound $L(Q, \Omega)$, ultimately enabling efficient (and closed-form) updates of the latents X and inducing variables W during optimization. We first rewrite the lower bound by explicitly writing out the definition of the expectation:

$$L(Q, \Omega) = \mathbb{E}_Q[\log P(Y, X, W, \theta|\Omega)] - \mathbb{E}_Q[\log Q(X, W, \theta)] \tag{B.6}$$

$$= \int Q(X, W, \theta) \log P(Y, X, W, \theta|\Omega)dXdWd\theta$$

$$- \int Q(X, W, \theta) \log Q(X, W, \theta)dXdWd\theta \tag{B.7}$$

$$= \int Q(X, W, \theta) \log \frac{P(Y, X, W, \theta|\Omega)}{Q(X, W, \theta)}dXdWd\theta \tag{B.8}$$

$$= \int Q(X, W, \theta)$$

$$\cdot \log \frac{P(\mathbf{d})P(\phi)P(C|\mathcal{A})P(\mathcal{A})P(Y|X, C, \mathbf{d}, \phi)P(X|W, \Omega)P(W|\tau)}{P(X|W, \Omega)Q_w(W)Q_d(\mathbf{d})Q_\phi(\phi)Q_c(C)Q_A(\mathcal{A})}$$

$$\cdot dXdWd\theta, \tag{B.9}$$

where in the last line, we substituted equations B.2 and B.5 into the numerator and denominator, respectively, of the log term. Notice the common term $P(X|W, \Omega)$ between the numerator and denominator. This term cancels, resulting in the following simplified expression for the lower bound:

$$L(Q, \Omega) = \int Q(X, W, \theta)$$

$$\cdot \log \frac{P(\mathbf{d})P(\phi)P(C|\mathcal{A})P(\mathcal{A})P(Y|X, C, \mathbf{d}, \phi)P(W|\tau)}{Q_w(W)Q_d(\mathbf{d})Q_\phi(\phi)Q_c(C)Q_A(\mathcal{A})}dX\,dW\,d\theta \tag{B.10}$$

To simplify notation further, define the joint prior distribution $P(W, \theta|\tau) = P(\mathbf{d})P(\phi)P(C|\mathcal{A})P(\mathcal{A})P(W|\tau)$ and the joint posterior distribution $Q_{w\theta}(W, \theta) = Q_w(W)Q_d(\mathbf{d})Q_\phi(\phi)Q_c(C)Q_A(\mathcal{A})$. Then substituting in

these expressions, along with $Q(X, W, \theta) = P(X|W, \Omega)Q_{w\theta}(W, \theta)$, we get

$$L(Q, \Omega) = \int P(X|W, \Omega)Q_{w\theta}(W, \theta) \log \frac{P(Y|X, C, \mathbf{d}, \phi)P(W, \theta|\tau)}{Q_{w\theta}(W, \theta)} dX \, dW \, d\theta \tag{B.11}$$

$$= \int Q_{w\theta}(W, \theta) \left[P(X|W, \Omega) \cdot \log P(Y|X, C, \mathbf{d}, \phi) dX \right] dW \, d\theta$$

$$+ \int Q_{w\theta}(W, \theta) \left[P(X|W, \Omega) \cdot \log \frac{P(W, \theta|\tau)}{Q_{w\theta}(W, \theta)} dX \right] dW \, d\theta \tag{B.12}$$

Now let $\mathbb{E}_{P_{x|w}}[\cdot]$ and $\mathbb{E}_{Q_{w\theta}}[\cdot]$ be expectations with respect to the conditional prior distribution $P(X|W, \Omega)$ and joint posterior distribution $Q_{w\theta}(W, \theta)$, respectively. We can then rewrite the lower bound as

$$L(Q, \Omega) = \mathbb{E}_{Q_{w\theta}} \left[\mathbb{E}_{P_{x|w}} \left[\log P(Y|X, C, \mathbf{d}, \phi) \right] \right]$$

$$+ \mathbb{E}_{Q_{w\theta}} \left[\mathbb{E}_{P_{x|w}} \left[\log \frac{P(W, \theta|\tau)}{Q_{w\theta}(W, \theta)} \right] \right] \tag{B.13}$$

$$= \mathbb{E}_{Q_{w\theta}} \left[\mathbb{E}_{P_{x|w}} \left[\log P(Y|X, C, \mathbf{d}, \phi) \right] \right] + \mathbb{E}_{Q_{w\theta}} \left[\log \frac{P(W, \theta|\tau)}{Q_{w\theta}(W, \theta)} \right], \tag{B.14}$$

The term $\mathbb{E}_{P_{x|w}}[\log P(Y|X, C, \mathbf{d}, \phi)]$ integrates out any dependence on the latents X and is therefore only a function of observations Y, inducing variables W, and parameters in θ and Ω. To emphasize these dependencies, we define $\log G(Y, W, \theta|\Omega) = \mathbb{E}_{P_{x|w}}[\log P(Y|X, C, \mathbf{d}, \phi)]$. Furthermore, note that the second term in equation B.14 is the negative KL-divergence between the joint posterior distribution $Q_{w\theta}(W, \theta)$ and the joint prior distribution $P(W, \theta|\tau)$.

We then arrive at the final reexpression of the lower bound:

$$L(Q, \Omega) = \mathbb{E}_{Q_{w\theta}} \left[\log G(Y, W, \theta|\Omega) \right] - \mathrm{KL}\big(Q_{w\theta}(W, \theta) \| P(W, \theta|\tau)\big). \tag{B.15}$$

This reexpression of the lower bound can then be iteratively maximized via coordinate ascent of the factors of $Q_{w\theta}(W, \theta)$ 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 lower bound, which is guaranteed to be nondecreasing, improves from one iteration to the next by less than a present tolerance (here we used 10^{-8}).

B.1.1 Posterior Distribution Updates. Maximizing the lower bound, $L(Q, \Omega)$, with respect to the kth factor of $Q_{w\theta}(W, \theta)$, $\hat{Q}_k$, results in the following update rule:

$$\log \hat{Q}_k = \langle \log G(Y, W, \theta|\Omega) + \log P(W, \theta|\tau) \rangle_{-k} + \text{const.} \tag{B.16}$$

Here we introduce the notation $\langle \cdot \rangle$ to indicate the expectation, $\mathbb{E}_{Q_{w\theta}}[\cdot]$, with respect to the posterior distribution $Q_{w\theta}(W, \theta)$, and $\langle \cdot \rangle_{-k}$ specifically indicates the expectation with respect to all but the kth factor of $Q_{w\theta}(W, \theta)$.

Because of the choice of gaussian and conjugate gamma priors in section 3.2, evaluation of equation B.16 for each factor $\hat{Q}_k$ leads to factors with the same functional form as their corresponding priors (equations 3.15 and 3.1–3.6):

$$Q_w(W) = \prod_{n=1}^{N} \mathcal{N}(\bar{\mathbf{w}}_n \mid \bar{\boldsymbol{\mu}}_{w_n}, \bar{\Sigma}_w), \tag{B.17}$$

$$Q_d(\mathbf{d}) = \prod_{m=1}^{M} \mathcal{N}(\mathbf{d}^m \mid \boldsymbol{\mu}_d^m, \Sigma_d^m), \tag{B.18}$$

$$Q_\phi(\phi) = \prod_{m=1}^{M} \prod_{r=1}^{q_m} \Gamma(\phi_r^m \mid \bar{a}_\phi, \bar{b}_{\phi,r}^m), \tag{B.19}$$

$$Q_c(C) = \prod_{m=1}^{M} \prod_{r=1}^{q_m} \mathcal{N}(\bar{\mathbf{c}}_r^m \mid \bar{\boldsymbol{\mu}}_{c_r}^m, \Sigma_{c_r}^m), \tag{B.20}$$

$$Q_{\mathcal{A}}(\mathcal{A}) = \prod_{m=1}^{M} \prod_{j=1}^{p} \Gamma(\alpha_j^m \mid \bar{a}_\alpha^m, \bar{b}_{\alpha,j}^m). \tag{B.21}$$

Here $\bar{\mathbf{w}}_n \in \mathbb{R}^{pT_{\text{ind}}}$ is a collection of all inducing variables on trial n (see below), and $\bar{\mathbf{c}}_r^m \in \mathbb{R}^p$ is the rth row of C^m, the loading matrix for group m. Any additional factorization in equations B.17 to B.21 emerge naturally; we only impose the factorization in equation B.5.

To express the updates for $Q_w(W)$, we first define several variables. First construct $\bar{\mathbf{w}}_n = [\mathbf{w}_{n,1,:}^\top \cdots \mathbf{w}_{n,p,:}^\top]^\top \in \mathbb{R}^{pT_{\text{ind}}}$ by vertically concatenating the inducing variables $\mathbf{w}_{n,j,:}$ for latents $j = 1, \ldots, p$ on trial n. Collect the inducing variable covariance matrices K_j^w for $j = 1, \ldots, p$ (equation 3.15; see Figure 1D, top) into the block diagonal matrix $\bar{K}^w = \text{diag}(K_1^w, \ldots, K_p^w) \in \mathbb{S}^{pT_{\text{ind}} \times pT_{\text{ind}}}$. Then let $\mathbf{k}_{m,j,t}^{xw} \in \mathbb{R}^{T_{\text{ind}}}$ be the tth row of $K_{m,j}^{xw} \in \mathbb{R}^{T \times T_{\text{ind}}}$, the mth block (for group m) of the covariance matrix between latent j and its inducing variable, K_j^{xw} (equation 3.18; Figure 1D, bottom). For time point t, collect each $\mathbf{k}_{m,j,t}^{xw}$ for latents $j = 1, \ldots, p$ into the block diagonal matrix $\bar{K}_{m,t}^{xw} = \text{diag}(\mathbf{k}_{m,1,t}^{xw\top}, \ldots, \mathbf{k}_{m,p,t}^{xw\top}) \in \mathbb{R}^{p \times pT_{\text{ind}}}$.

Posterior estimates of the inducing variables W are independent across trials. We can thus update $Q_w(W)$ by evaluating the posterior covariance,

$\bar{\Sigma}_w \in \mathbb{S}^{pT_{\text{ind}} \times pT_{\text{ind}}}$, and mean, $\bar{\boldsymbol{\mu}}_{w_n} \in \mathbb{R}^{pT_{\text{ind}}}$, of $\bar{\mathbf{w}}_n$, for each trial n:

$$\bar{\Sigma}_w = \left((\bar{K}^w)^{-1} + (\bar{K}^w)^{-1} \left[\sum_{m=1}^{M} \sum_{t=1}^{T} \bar{K}_{m,t}^{wx} \langle (C^m)^\top \Phi^m C^m \rangle \bar{K}_{m,t}^{xw} \right] (\bar{K}^w)^{-1} \right)^{-1}$$

(B.22)

$$\bar{\boldsymbol{\mu}}_{w_n} = \bar{\Sigma}_w (\bar{K}^w)^{-1} \left[\sum_{m=1}^{M} \sum_{t=1}^{T} \bar{K}_{m,t}^{wx} \langle C^m \rangle^\top \langle \Phi^m \rangle \left(\mathbf{y}_{n,t}^m - \langle \mathbf{d}^m \rangle \right) \right],$$

(B.23)

where $\bar{K}_{m,t}^{wx} = (\bar{K}_{m,t}^{xw})^\top$. Note that (1) the update for the posterior covariance, $\bar{\Sigma}_w$, is identical for trials of the same length. This computation can therefore be reused efficiently across trials. (2) Under the posterior distribution, the inducing variables for latents $j = 1, \ldots, p$ are no longer independent, as they are under the prior distribution (equations 3.15, B.2).

For multiple groups ($M > 1$) and a sufficiently small number of inducing points T_{ind}, evaluation of equation B.22 is significantly more efficient than the evaluation of the analogous equation 3.14 for mDLAG-time. The quantity $\bar{K}_{m,t}^{wx} \langle (C^m)^\top \Phi^m C^m \rangle \bar{K}_{m,t}^{xw}$ costs $\mathcal{O}(p^3 T_{\text{ind}}^2)$ operations, for a total cost of $\mathcal{O}(p^3 M T T_{\text{ind}}^2)$ when evaluated for all groups and time points. The inversion of the block diagonal $\bar{K}^w$ costs $O(pT_{\text{ind}}^3)$ operations. Then the final inversion of a $pT_{\text{ind}} \times pT_{\text{ind}}$ matrix costs $\mathcal{O}(p^3 T_{\text{ind}}^3)$ operations and $\mathcal{O}(p^2 T_{\text{ind}}^2)$ storage. Thus, in total, posterior inference over the inducing variables scales linearly in both the number of time points per trial T and the number of groups M, albeit superlinearly in the number of inducing points T_{ind}.

The posterior mean and covariance of the latents, X, are computed from their corresponding inducing variables using the conditional prior distribution $P(X|W, \Omega)$ (equation 3.18). The posterior mean of latent j across groups $m = 1, \ldots, M$, $\mathbf{x}_{n,j,:} = [\mathbf{x}_{n,j,:}^{1\top} \cdots \mathbf{x}_{n,j,:}^{M\top}]^\top \in \mathbb{R}^{MT}$ is given by

$$\langle \mathbf{x}_{n,j,:} \rangle = \mathbb{E}_{Q_{w\theta}} \left[\mathbb{E}_{P_{x|w}} [\mathbf{x}_{n,j,:}] \right]$$

(B.24)

$$= \mathbb{E}_{Q_{w\theta}} \left[K_j^{xw} (K_j^w)^{-1} \mathbf{w}_{n,j,:} \right]$$

(B.25)

$$= K_j^{xw} (K_j^w)^{-1} \langle \mathbf{w}_{n,j,:} \rangle.$$

(B.26)

This computation requires $\mathcal{O}(T_{\text{ind}}^3)$ operations for the inversion of K_j^w, followed by $\mathcal{O}(M T T_{\text{ind}} + T_{\text{ind}}^2)$ operations for the remaining matrix multiplications.

We next construct $\hat{\mathbf{x}}_n = [\mathbf{x}_{n,1,:}^\top \cdots \mathbf{x}_{n,p,:}^\top]^\top \in \mathbb{R}^{pMT}$ by vertically concatenating latents $j = 1, \ldots, p$ on trial n. Also collect the covariance matrices between each latent $j = 1, \ldots, p$ and its inducing variable, $K_j^{xw} \in \mathbb{R}^{MT \times T_{\text{ind}}}$ (equation 3.18; Figure 1D, bottom), into the block diagonal matrix $\bar{K}^{xw} = \text{diag}(K_1^{xw}, \ldots, K_p^{xw}) \in \mathbb{R}^{pMT \times pT_{\text{ind}}}$. Similarly, collect the latent covariance matrices, $K_j \in \mathbb{S}^{MT \times MT}$ (equation 3.12; Figure 1B), into the block

diagonal matrix $\hat{K} = \mathrm{diag}(K_1, \ldots, K_p) \in \mathbb{S}^{pMT \times pMT}$ ($\hat{K}$ is simply a reorganization of $\bar{K}$ in equation 3.14). Then the full posterior covariance of the latents, $\hat{\Sigma}_x \in \mathbb{S}^{pMT \times pMT}$ is given by

$$\hat{\Sigma}_x = \mathbb{E}_{Q_{w\theta}}\big[\mathbb{E}_{P_{x|w}}[\hat{\mathbf{x}}_n \hat{\mathbf{x}}_n^\top]\big] - \mathbb{E}_{Q_{w\theta}}\big[\mathbb{E}_{P_{x|w}}[\hat{\mathbf{x}}_n]\big]\mathbb{E}_{Q_{w\theta}}\big[\mathbb{E}_{P_{x|w}}[\hat{\mathbf{x}}_n]\big]^\top \tag{B.27}$$

$$= \langle \hat{\mathbf{x}}_n \hat{\mathbf{x}}_n^\top \rangle - \langle \hat{\mathbf{x}}_n \rangle \langle \hat{\mathbf{x}}_n \rangle^\top \tag{B.28}$$

$$= \hat{K} - \bar{K}^{xw}(\bar{K}^w)^{-1}\bar{K}^{wx} + \bar{K}^{xw}(\bar{K}^w)^{-1}\langle \bar{\mathbf{w}}_n \bar{\mathbf{w}}_n^\top \rangle(\bar{K}^w)^{-1}\bar{K}^{wx}$$
$$\quad - \bar{K}^{xw}(\bar{K}^w)^{-1}\langle \bar{\mathbf{w}}_n \rangle\langle \bar{\mathbf{w}}_n \rangle^\top(\bar{K}^w)^{-1}\bar{K}^{wx} \tag{B.29}$$

$$= \hat{K} - \bar{K}^{xw}(\bar{K}^w)^{-1}\bar{K}^{wx} + \bar{K}^{xw}(\bar{K}^w)^{-1}\bar{\Sigma}_w(\bar{K}^w)^{-1}\bar{K}^{wx}, \tag{B.30}$$

where the inducing variable covariance matrix $\bar{\Sigma}_w$ is defined in equation B.22.

The prior covariance matrix $\hat{K}$, the posterior covariance matrix $\hat{\Sigma}_x$, or the posterior second moment $\langle \hat{\mathbf{x}}_n \hat{\mathbf{x}}_n^\top \rangle$ never need to be computed in full during the mDLAG-inducing fitting procedure. Only subsets of these quantities are needed, thereby maintaining linear scaling in M and T and runtime gains over mDLAG-time. From here, the updates for $Q_d(\mathbf{d})$, $Q_\phi(\phi)$, $Q_c(C)$, and $Q_\mathcal{A}(\mathcal{A})$ look identical in form to the analogous updates in mDLAG-time (Gokcen et al., 2023). We include them here for completeness.

Posterior estimates of the mean parameters $\mathbf{d}$ are independent across groups (and, in fact, units). 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 group m:

$$\Sigma_d^m = \big(\beta I_{q_m} + NT\langle \Phi^m \rangle\big)^{-1}, \tag{B.31}$$

$$\boldsymbol{\mu}_d^m = \Sigma_d^m \langle \Phi^m \rangle \sum_{n=1}^N \sum_{t=1}^T \big(\mathbf{y}_{n,t}^m - \langle C^m \rangle\langle \mathbf{x}_{n,t}^m \rangle\big). \tag{B.32}$$

Posterior estimates of precision parameters ϕ are independent across groups and units. We can thus update $Q_\phi(\phi)$ by evaluating the posterior parameters $\bar{a}_\phi$ and $\bar{b}_{\phi,r}^m$ of parameter ϕ_r^m for each unit r in group m:

$$\bar{a}_\phi = a_\phi + \frac{NT}{2}, \tag{B.33}$$

$$\bar{b}_{\phi,r}^m = b_\phi + \frac{1}{2}\sum_{n=1}^N \sum_{t=1}^T \bigg[(y_{n,r,t}^m)^2 + \langle (d_r^m)^2 \rangle + \mathrm{tr}\big(\langle \bar{\mathbf{c}}_r^m(\bar{\mathbf{c}}_r^m)^\top \rangle\langle \mathbf{x}_{n,t}^m(\mathbf{x}_{n,t}^m)^\top \rangle\big)$$

$$\quad - 2\langle \bar{\mathbf{c}}_r^m \rangle^\top \langle \mathbf{x}_{n,t}^m \rangle\big(y_{n,r,t}^m - \langle d_r^m \rangle\big) - 2y_{n,r,t}^m \langle d_r^m \rangle \bigg]. \tag{B.34}$$

Here $\bar{\mathbf{c}}_r^m \in \mathbb{R}^p$ is again the rth row of C^m, the loading matrix for group m.

Posterior estimates of loading matrices C are independent across groups and units, that is, across the rows of each C^m. We can thus update $Q_C(C)$ by evaluating the posterior covariance, $\Sigma_{c_r}^m \in \mathbb{S}^{p \times p}$, and mean, $\bar{\boldsymbol{\mu}}_{c_r}^m \in \mathbb{R}^p$, of the rth row of C^m:

$$\Sigma_{c_r}^m = \left(\langle \mathcal{A}^m \rangle + \langle \phi_r^m \rangle \sum_{n=1}^{N} \sum_{t=1}^{T} \langle \mathbf{x}_{n,t}^m (\mathbf{x}_{n,t}^m)^\top \rangle \right)^{-1}, \tag{B.35}$$

$$\bar{\boldsymbol{\mu}}_{c_r}^m = \Sigma_{c_r}^m \langle \phi_r^m \rangle \sum_{n=1}^{N} \sum_{t=1}^{T} \langle \mathbf{x}_{n,t}^m \rangle \left(y_{n,r,t}^m - \langle d_r^m \rangle \right). \tag{B.36}$$

Here $\mathcal{A}^m = \text{diag}(\alpha_1^m, \ldots, \alpha_p^m)$. Note that the posterior independence over the rows of each C^m contrasts with the prior independence over the columns of each C^m (equations 3.5, B.2).

Finally, posterior estimates of ARD parameters $\mathcal{A}$ are independent across groups and latents. We can thus update $Q_{\mathcal{A}}(\mathcal{A})$ by evaluating the posterior parameters $\bar{a}_\alpha^m$ and $\bar{b}_{\alpha,j}^m$ of parameter α_j^m for each group m and latent j:

$$\bar{a}_\alpha^m = a_\alpha + \frac{q_m}{2}, \tag{B.37}$$

$$\bar{b}_{\alpha,j}^m = b_\alpha + \frac{1}{2} \left\langle \|\mathbf{c}_j^m\|_2^2 \right\rangle, \tag{B.38}$$

where $\mathbf{c}_j^m \in \mathbb{R}^{q_m}$ is the jth column of the loading matrix C^m. All moments $\langle \cdot \rangle$ can be readily computed from the approximate posterior distributions given in equations B.17 to B.21.

B.1.2 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. To optimize timescales and time delays, we rewrite the lower bound to show the terms that depend on the covariance matrices that define the latents and their inducing variables, K_j, K_j^{xw}, and K_j^w for latents $j = 1, \ldots, p$. First we define several intermediate variables. Construct $\hat{\mathbf{y}}_n^m = [\mathbf{y}_{n,1,:}^{m\top} \cdots \mathbf{y}_{n,q_m,:}^{m\top}]^\top \in \mathbb{R}^{q_m T}$ by vertically concatenating the observed activity of each unit $r = 1, \ldots, q_m$ in group m across time on trial n, $\mathbf{y}_{n,r,:}^m \in \mathbb{R}^T$. Then construct $\hat{\mathbf{y}}_n = [\hat{\mathbf{y}}_n^{1\top} \cdots \hat{\mathbf{y}}_n^{M\top}]^\top \in \mathbb{R}^{qT}$, $q = \sum_m q_m$, by vertically concatenating the $\hat{\mathbf{y}}_n^m$ across groups $m = 1, \ldots, M$. Vertically concatenate all mean parameters $\mathbf{d}^m \in \mathbb{R}^{q_m}$ across groups $m = 1, \ldots, M$ to give the vector $\mathbf{d} = [\mathbf{d}^{1\top} \cdots \mathbf{d}^{M\top}]^\top \in \mathbb{R}^q$. And finally, let $\hat{C}_j = \text{diag}(\mathbf{c}_j^1, \ldots, \mathbf{c}_j^M) \in \mathbb{R}^{q \times M}$ be the block diagonal matrix where each block is the jth column of the loading matrix to group m, $\mathbf{c}_j^m \in \mathbb{R}^q$.

Then the terms of the lower-bound $L(Q, \Omega)$ that depend on the jth latent are given by

$$
\begin{aligned}
L_j = \sum_{n=1}^{N} \Bigg[&\frac{1}{2} \log |(K_j^w)^{-1}| - \frac{1}{2} \mathrm{tr}\left((K_j^w)^{-1} \langle \mathbf{w}_{n,j,:} \mathbf{w}_{n,j,:}^\top \rangle \right) \\
&+ \langle \mathbf{w}_{n,j,:} \rangle^\top (K_j^w)^{-1} K_j^{wx} \left(\langle \hat{C}_j \rangle^\top \langle \Phi \rangle \otimes I_T \right) \left(\hat{\mathbf{y}}_n - \langle \mathbf{d} \rangle \otimes \mathbf{1}_T \right) \\
&- \frac{1}{2} \mathrm{tr}\left(\left(\langle \hat{C}_j^\top \Phi \hat{C}_j \rangle \otimes I_T \right) \left(K_j - K_j^{xw} (K_j^w)^{-1} K_j^{wx} \right) \right) \\
&- \frac{1}{2} \mathrm{tr}\left(\left(\langle \hat{C}_j^\top \Phi \hat{C}_j \rangle \otimes I_T \right) K_j^{xw} (K_j^w)^{-1} \langle \mathbf{w}_{n,j,:} \mathbf{w}_{n,j,:}^\top \rangle (K_j^w)^{-1} K_j^{wx} \right) \\
&- \sum_{k \neq j} \mathrm{tr}\left(\left(\langle \hat{C}_k^\top \Phi \hat{C}_j \rangle \otimes I_T \right) K_j^{xw} (K_j^w)^{-1} \langle \mathbf{w}_{n,j,:} \mathbf{w}_{n,k,:}^\top \rangle (K_k^w)^{-1} K_k^{wx} \right) \Bigg],
\end{aligned}
$$
(B.39)

where $\otimes$ is the Kronecker product, $I_T \in \mathbb{S}^{T \times T}$ is the $T \times T$ identity matrix, and $\mathbf{1}_T \in \mathbb{R}^T$ is the length-T column vector of ones. The overall lower bound can then be reexpressed as

$$
L(Q, \Omega) = \sum_{j=1}^{p} L_j + \mathrm{const}.
$$
(B.40)

From here we omit explicit expressions for the gradients of the lower-bound $L(Q, \Omega)$ with respect to each of the timescale and time delay parameters. However, we do write out intermediate expressions for the gradients starting from the chain rule to illustrate the dependencies of each gradient update on the covariance matrices K_j, K_j^{xw}, and K_j^w. The gradients with respect to timescale τ_j for latent j are given by

$$
\begin{aligned}
\frac{\partial L}{\partial \tau_j} = {}& \mathrm{tr}\left(\left(\frac{\partial L_j}{\partial K_j} \right)^\top \left(\frac{\partial K_j}{\partial \tau_j} \right) \right) \\
&+ \mathrm{tr}\left(\left(\frac{\partial L_j}{\partial K_j^{xw}} \right)^\top \left(\frac{\partial K_j^{xw}}{\partial \tau_j} \right) \right) + \mathrm{tr}\left(\left(\frac{\partial L_j}{\partial K_j^{wx}} \right)^\top \left(\frac{\partial K_j^{wx}}{\partial \tau_j} \right) \right) \\
&+ \mathrm{tr}\left(\left(\frac{\partial L_j}{\partial K_j^w} \right)^\top \left(\frac{\partial K_j^w}{\partial \tau_j} \right) \right).
\end{aligned}
$$
(B.41)

However, because of special structure in the partial derivatives $\frac{\partial L_j}{\partial K_j}$ (diagonal) and $\frac{\partial K_j}{\partial \tau_j}$ (zero along the diagonal), the first trace term evaluates exactly

to zero. Furthermore, because $K_j^{wx} = (K_j^{xw})^\top$, we can simplify

$$\frac{\partial L}{\partial \tau_j} = 2 \cdot \text{tr}\left(\left(\frac{\partial L_j}{\partial K_j^{xw}}\right)^\top \left(\frac{\partial K_j^{xw}}{\partial \tau_j}\right)\right) + \text{tr}\left(\left(\frac{\partial L_j}{\partial K_j^{w}}\right)^\top \left(\frac{\partial K_j^{w}}{\partial \tau_j}\right)\right), \quad \text{(B.42)}$$

leaving dependence only on the smaller covariance matrices K_j^{xw} and K_j^{w}, and not on the large covariance matrix K_j. The gradients with respect to time delay D_j^m for latent j and group m can be similarly simplified:

$$\frac{\partial L}{\partial D_j^m} = 2 \cdot \text{tr}\left(\left(\frac{\partial L_j}{\partial K_j^{xw}}\right)^\top \left(\frac{\partial K_j^{xw}}{\partial D_j^m}\right)\right). \quad \text{(B.43)}$$

Since K_j^{w} does not involve time delays (equation 3.17), the partial derivative $\frac{\partial L}{\partial D_j^m}$ does not depend on K_j^{w}.

B.2 Evaluation of the Lower Bound. To monitor the progress of the fitting procedure, we evaluate the variational lower bound on each iteration:

$$L(Q, \Omega) = \mathbb{E}_{Q_{w\theta}}\left[\log G(Y, W, \theta|\Omega)\right] - \text{KL}(Q_{w\theta}(W, \theta)\|P(W, \theta|\tau)). \quad \text{(B.44)}$$

Due to the factorized forms of $Q_{w\theta}(W, \theta)$ and $P(W, \theta|\tau)$, $L(Q, \Omega)$ becomes

$$\begin{aligned} L(Q, \Omega) = {} & \mathbb{E}_{Q_{w\theta}}\left[\log G(Y, W, \theta|\Omega)\right] - \text{KL}(Q_w(W)\|P(W|\tau)) \\ & - \text{KL}(Q_c(C)\|P(C|\mathcal{A})) - \text{KL}(Q_{\mathcal{A}}(\mathcal{A})\|P(\mathcal{A})) \\ & - \text{KL}(Q_\phi(\phi)\|P(\phi)) - \text{KL}(Q_d(\mathbf{d})\|P(\mathbf{d})). \end{aligned} \quad \text{(B.45)}$$

This form of the lower bound provides insight into the nature of the optimization procedure for fitting mDLAG-inducing models. The first term is essentially an expected log-likelihood (with respect to the approximate posterior $Q_{w\theta}(W, \theta)$) of the observations Y, given the latest estimates of the inducing variables W and model parameters θ and Ω. This term encourages mDLAG-inducing models to explain the observed 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 B.1.1 and the prior definitions in section 3.2, each term of the lower bound can be computed as follows:

$$\begin{aligned} \mathbb{E}_{Q_{w\theta}}\left[\log G(Y, W, \theta|\Omega)\right] = {} & -\frac{qNT}{2}\log(2\pi) + \frac{NT}{2}\sum_{m=1}^{M}\sum_{r=1}^{q_m}\langle\log\phi_r^m\rangle \\ & - \sum_{m=1}^{M}\sum_{r=1}^{q_m}(\bar{a}_\phi - \langle\phi_r^m\rangle b_\phi) \end{aligned} \quad \text{(B.46)}$$

$$-\text{KL}(Q_w(W)\|P(W|\tau)) = \frac{pNT_{\text{ind}}}{2} + \frac{1}{2}\sum_{n=1}^{N}\Bigg[\log|\bar{\Sigma}_w|$$
$$-\sum_{j=1}^{p}\Big[\log|K_j^w| + \text{tr}((K_j^w)^{-1}\langle\mathbf{w}_{n,j,:}\mathbf{w}_{n,j,:}^{\top}\rangle)\Big]\Bigg] \tag{B.47}$$

$$-\text{KL}(Q_c(C)\|P(C|\mathcal{A})) = \frac{qp}{2} + \sum_{m=1}^{M}\Bigg[\frac{1}{2}\sum_{r=1}^{q_m}\log|\Sigma_{c_r}^m|$$
$$+ \sum_{j=1}^{p}\Big[\frac{q_m}{2}\langle\log\alpha_j^m\rangle - \frac{1}{2}\langle\alpha_j^m\rangle\langle\|\mathbf{c}_j^m\|_2^2\rangle\Big]\Bigg] \tag{B.48}$$

$$-\text{KL}(Q_{\mathcal{A}}(\mathcal{A})\|P(\mathcal{A})) = \sum_{m=1}^{M}\sum_{j=1}^{p}\Bigg[-\bar{a}_\alpha^m\log\bar{b}_{\alpha,j}^m + a_\alpha\log b_\alpha + \log\frac{\Gamma(\bar{a}_\alpha^m)}{\Gamma(a_\alpha)}$$
$$- b_\alpha\langle\alpha_j^m\rangle + \bar{a}_\alpha^m + (a_\alpha - \bar{a}_\alpha^m)(\Psi(\bar{a}_\alpha^m) - \log\bar{b}_{\alpha,j}^m)\Bigg] \tag{B.49}$$

$$-\text{KL}(Q_\phi(\phi)\|P(\phi)) = \sum_{m=1}^{M}\sum_{r=1}^{q_m}\Bigg[-\bar{a}_\phi\log\bar{b}_{\phi,r}^m + a_\phi\log b_\phi + \log\frac{\Gamma(\bar{a}_\phi)}{\Gamma(a_\phi)}$$
$$- b_\phi\langle\phi_r^m\rangle + \bar{a}_\phi + (a_\phi - \bar{a}_\phi)(\Psi(\bar{a}_\phi) - \log\bar{b}_{\phi,r}^m)\Bigg] \tag{B.50}$$

$$-\text{KL}(Q_d(\mathbf{d})\|P(\mathbf{d})) = \frac{q}{2} + \frac{q}{2}\log\beta + \frac{1}{2}\log|\Sigma_d| - \frac{1}{2}\beta\langle\|\mathbf{d}\|_2^2\rangle. \tag{B.51}$$

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 B.17 to B.21.

Appendix C: Posterior Inference and Fitting via mDLAG-Frequency ___

C.1 Variational Inference. Let $\widetilde{Y}$ and $\widetilde{X}$ be collections of all observed neural activity and latent variables, respectively, across all frequencies and trials. Similarly, let $\mathbf{d}, \phi, C, \mathcal{A}, \tau$, and D be collections of the mean parameters, noise precisions, loading matrices, ARD parameters, GP timescales, and time delays, respectively. From frequency domain observations, we seek to estimate posterior distributions over the probabilistic model components,

$$\theta = \{\widetilde{X}, \mathbf{d}, \phi, C, \mathcal{A}\}, \tag{C.1}$$

and point estimates of the deterministic GP parameters $\Omega = \{\tau, D\}$.

In the case of methods like GPFA (Yu et al., 2009) and DLAG (Gokcen et al., 2022), the linear-gaussian structure of the model enables an exact expectation-maximization (EM) algorithm. With the introduction of prior distributions over model parameters (to enable automatic relevance determination), the family of mDLAG models, including mDLAG-frequency, loses this property. The complete quasi-likelihood of the mDLAG-frequency model (Figure 1E),

$$
P(\widetilde{Y}, \theta | \Omega) = P(\mathbf{d}) P(\phi) P(C|\mathcal{A}) P(\mathcal{A}) P\left(\widetilde{Y}|\widetilde{X}, C, \mathbf{d}, \phi, D\right) P\left(\widetilde{X}|\tau\right)
$$

$$
= \prod_{m=1}^{M}\left[P(\mathbf{d}^m)\left[\prod_{r=1}^{q_m} P(\phi_r^m)\right]\left[\prod_{j=1}^{p} P(\mathbf{c}_j^m \mid \alpha_j^m)P(\alpha_j^m)\right]\right]
$$

$$
\cdot \prod_{n=1}^{N}\prod_{l=1}^{T}\left[\left[\prod_{m=1}^{M} P(\widetilde{\mathbf{y}}_{n,l}^m \mid \widetilde{\mathbf{x}}_{n,l}, C^m, \mathbf{d}^m, \phi^m, \{D_j^m\}_{j=1}^p)\right]\right.
$$

$$
\left.\cdot\left[\prod_{j=1}^{p} P(x_{n,j,l} \mid \tau_j)\right]\right], \tag{C.2}
$$

is no longer gaussian. Then a hypothetical EM E-step (evaluation of the posterior distribution $P(\theta|\widetilde{Y}, \Omega)$) becomes prohibitive, as it relies on the analytically intractable marginalization of equation C.2 with respect to θ.

We therefore employ instead a variational inference scheme (Bishop, 1999; Gokcen et al., 2023; Klami et al., 2015) in which we maximize the lower-bound $\widetilde{L}(\widetilde{Q}, \Omega)$ to the log quasi-likelihood $\log P(\widetilde{Y})$, with respect to the approximate posterior distribution $\widetilde{Q}(\theta)$ and the deterministic parameters Ω:

$$
\log P(\widetilde{Y}) \geq \widetilde{L}(\widetilde{Q}, \Omega) = \mathbb{E}_{\widetilde{Q}}[\log P(\widetilde{Y}, \theta|\Omega)] - \mathbb{E}_{\widetilde{Q}}[\log \widetilde{Q}(\theta)]. \tag{C.3}
$$

The quasi-likelihood $P(\widetilde{Y})$ is meant to approximate the time domain likelihood $P(Y)$ (equation 3.10). The two quantities converge as the number of time points per trial, T, becomes large (Whittle, 1951).

We constrain $\widetilde{Q}(\theta)$ so that it factorizes over the elements of θ:

$$
\widetilde{Q}(\theta) = Q_{\widetilde{x}}(\widetilde{X})Q_d(\mathbf{d})Q_\phi(\phi)Q_c(C)Q_{\mathcal{A}}(\mathcal{A}). \tag{C.4}
$$

This factorization enables closed-form updates during optimization (see below). The lower-bound $\widetilde{L}(\widetilde{Q}, \Omega)$ can then be iteratively maximized via coordinate ascent of the factors of $\widetilde{Q}(\theta)$ 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 lower bound, which is guaranteed to be nondecreasing, improves from one iteration to the next by less than a present tolerance (here we used 10^{-8}).

C.1.1 Posterior Distribution Updates. Maximizing the lower bound, $\widetilde{L}(\widetilde{Q}, \Omega)$, with respect to the kth factor of $\widetilde{Q}$, $\hat{Q}_k$, results in the following update rule (Bishop, 1999):

$$\log \hat{Q}_k(\theta_k) = \langle \log P(\widetilde{Y}, \theta | \Omega) \rangle_{-k} + \text{const.} \tag{C.5}$$

Here we introduce the notation $\langle \cdot \rangle$ to indicate the expectation with respect to the approximate posterior distribution, $\mathbb{E}_{\widetilde{Q}}[\cdot]$, and $\langle \log P(\widetilde{Y}, \theta | \Omega) \rangle_{-k}$ specifically indicates the expectation of the complete log likelihood with respect to all but the kth factor of $\widetilde{Q}$.

Because of the choice of gaussian and conjugate gamma priors in section 3.3, evaluation of equation C.5 for each factor $\hat{Q}_k$ leads to factors with the same functional form as their corresponding priors (equations 3.25, 3.3–3.6):

$$Q_{\widetilde{x}}(\widetilde{X}) = \prod_{n=1}^{N} \prod_{l=1}^{T} \mathcal{N}(\widetilde{\mathbf{x}}_{n,l} \mid \widetilde{\boldsymbol{\mu}}_{x_{n,l}}, \widetilde{\Sigma}_{x,l}), \tag{C.6}$$

$$Q_d(\mathbf{d}) = \prod_{m=1}^{M} \mathcal{N}(\mathbf{d}^m \mid \boldsymbol{\mu}_d^m, \Sigma_d^m), \tag{C.7}$$

$$Q_\phi(\phi) = \prod_{m=1}^{M} \prod_{r=1}^{q_m} \Gamma(\phi_r^m \mid \bar{a}_\phi, \bar{b}_{\phi,r}^m), \tag{C.8}$$

$$Q_c(C) = \prod_{m=1}^{M} \prod_{r=1}^{q_m} \mathcal{N}(\bar{\mathbf{c}}_r^m \mid \bar{\boldsymbol{\mu}}_{c_r}^m, \Sigma_{c_r}^m), \tag{C.9}$$

$$Q_{\mathcal{A}}(\mathcal{A}) = \prod_{m=1}^{M} \prod_{j=1}^{p} \Gamma(\alpha_j^m \mid \bar{a}_\alpha^m, \bar{b}_{\alpha,j}^m). \tag{C.10}$$

Here $\bar{\mathbf{c}}_r^m \in \mathbb{R}^p$ is the rth row of C^m, the loading matrix for group m. Any additional factorization in equations C.6 to C.10 emerge naturally; we impose only the factorization in equation C.4.

Posterior estimates of the frequency domain latents $\widetilde{X}$ are independent across trials and frequencies. We can thus update $Q_{\widetilde{x}}(\widetilde{X})$ by evaluating the posterior covariance, $\widetilde{\Sigma}_{x,l} \in \mathbb{C}^{p \times p}$, and mean, $\widetilde{\boldsymbol{\mu}}_{x_{n,l}} \in \mathbb{C}^p$, of $\widetilde{\mathbf{x}}_{n,l}$ for each trial n and frequency l:

$$\widetilde{\Sigma}_{x,l} = \left(S_l^{-1} + \sum_{m=1}^{M} (H_l^m)^{\mathsf{H}} \langle (C^m)^\top \Phi^m C^m \rangle H_l^m \right)^{-1} \tag{C.11}$$

$$\widetilde{\boldsymbol{\mu}}_{x_{n,l}} = \widetilde{\Sigma}_{x,l} \sum_{m=1}^{M} (H_l^m)^{\mathsf{H}} \langle C^m \rangle^{\top} \langle \Phi^m \rangle \big(\widetilde{\mathbf{y}}_{n,l}^m - \langle \widetilde{\mathbf{d}}_l^m \rangle \big). \tag{C.12}$$

Recall from equation 3.28 that the diagonal matrix $H_l^m \in \mathbb{C}^{p \times p}$ is a collection of the phase shift terms $h_{j,l}^m = \exp(-i2\pi f_l D_j^m)$, across all latents $j = 1, \ldots, p$ at the single frequency l: $H_l^m = \mathrm{diag}(h_{1,l}^m, \ldots h_{p,l}^m)$. The mean parameter term $\widetilde{\mathbf{d}}_l^m \in \mathbb{R}^{q_m}$ is shorthand for $\delta_{l-1} \cdot \sqrt{T} \mathbf{d}^m$, which is $\sqrt{T} \mathbf{d}^m$ for the zero frequency, and $\mathbf{0}$ otherwise. The elements of the diagonal PSD matrix $S_l = \mathrm{diag}(s_1(f_l), \ldots, s_p(f_l)) \in \mathbb{R}^{p \times p}$ are computed using equation 3.24. Note that (1) the update for the posterior covariance, $\widetilde{\Sigma}_{x,l}$, is identical for trials of the same length. This computation can therefore be reused efficiently across trials. (2) Under the posterior distribution, latent variables $j = 1, \ldots, p$ are no longer independent, as they are under the prior distribution (equations 3.25, C.2).

Posterior estimates of the mean parameters $\mathbf{d}$ are independent across groups (and, in fact, observed units). 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:

$$\Sigma_d^m = \big(\beta I_{q_m} + NT \langle \Phi^m \rangle \big)^{-1}, \tag{C.13}$$

$$\boldsymbol{\mu}_d^m = \Sigma_d^m \langle \Phi^m \rangle \sum_{n=1}^{N} \sqrt{T} \big(\widetilde{\mathbf{y}}_{n,1}^m - \langle C^m \rangle \langle \widetilde{\mathbf{x}}_{n,1} \rangle \big). \tag{C.14}$$

Note that updates to the mean, $\boldsymbol{\mu}_d^m$, depend only on observed and latent variables at the zero frequency, at index $l = 1$.

Posterior estimates of precision parameters ϕ are independent across groups and units. We can thus update $Q_\phi(\phi)$ by evaluating the posterior parameters $\bar{a}_\phi$ and $\bar{b}_{\phi,r}^m$ of parameter ϕ_r^m for each unit r in group m:

$$\widetilde{a}_\phi = a_\phi + \frac{NT}{2}, \tag{C.15}$$

$$\begin{aligned}
\widetilde{b}_{\phi,i}^m = b_\phi + \frac{1}{2} \sum_{n=1}^{N} \sum_{l=1}^{T} \Re \bigg\{ \bigg[|\widetilde{y}_{n,r,l}^m|^2 + \delta_{l-1} T \langle (d_r^m)^2 \rangle \\
+ \mathrm{tr} \big(\langle \bar{\mathbf{c}}_r^m (\bar{\mathbf{c}}_r^m)^{\top} \rangle H_l^m \langle \widetilde{\mathbf{x}}_{n,l} \widetilde{\mathbf{x}}_{n,l}^{\mathsf{H}} \rangle H_l^{m\mathsf{H}} \big) \\
- 2 \langle \bar{\mathbf{c}}_r^m \rangle^{\top} H_l^m \langle \widetilde{\mathbf{x}}_{n,l} \rangle \big(\widetilde{y}_{n,r,l}^m - \delta_{l-1} \sqrt{T} \langle d_r^m \rangle \big)^* - 2 \delta_{l-1} \sqrt{T} \widetilde{y}_{n,r,l}^m \langle d_r^m \rangle \bigg] \bigg\}.
\end{aligned} \tag{C.16}$$

Here $\bar{\mathbf{c}}_r^m \in \mathbb{R}^p$ is again, the rth row of C^m, the loading matrix for group m. The term δ_{l-1} is the Kronecker delta: 1 for the zero frequency ($l = 1$) and 0 otherwise. $\Re\{\cdot\}$ indicates the real part of the enclosed expression.

Posterior estimates of loading matrices C are independent across groups and units, that is, across the rows of each C^m. We can thus update $Q_c(C)$ by evaluating the posterior covariance, $\Sigma_{c_r}^m \in \mathbb{S}^{p \times p}$, and mean, $\bar{\boldsymbol{\mu}}_{c_r}^m \in \mathbb{R}^p$, of the rth row of C^m:

$$\Sigma_{c_r}^m = \left(\langle \mathcal{A}^m \rangle + \langle \phi_r^m \rangle \Re \left\{ \sum_{n=1}^N \sum_{l=1}^T H_l^m \langle \widetilde{\mathbf{x}}_{n,l} \widetilde{\mathbf{x}}_{n,l}^{\mathsf{H}} \rangle H_l^{m\mathsf{H}} \right\} \right)^{-1}, \tag{C.17}$$

$$\bar{\boldsymbol{\mu}}_{c_r}^m = \Sigma_{c_r}^m \langle \phi_r^m \rangle \Re \left\{ \sum_{n=1}^N \sum_{l=1}^T H_l^m \langle \widetilde{\mathbf{x}}_{n,l} \rangle \left(\widetilde{y}_{n,r,l}^m - \delta_{l-1} \sqrt{T} \langle d_r^m \rangle \right)^* \right\}. \tag{C.18}$$

Here $\mathcal{A}^m = \mathrm{diag}(\alpha_1^m, \ldots, \alpha_p^m)$. Note that the posterior independence over the rows of each C^m contrasts with the prior independence over the columns of each C^m (equations 3.5, C.2).

Finally, posterior estimates of ARD parameters $\mathcal{A}$ are independent across groups and latents. We can thus update $Q_{\mathcal{A}}(\mathcal{A})$ by evaluating the posterior parameters $\bar{a}_\alpha^m$ and $\bar{b}_{\alpha,j}^m$ of parameter α_j^m for each group m and latent j:

$$\bar{a}_\alpha^m = a_\alpha + \frac{q_m}{2}, \tag{C.19}$$

$$\bar{b}_{\alpha,j}^m = b_\alpha + \frac{1}{2} \left\langle \|\mathbf{c}_j^m\|_2^2 \right\rangle, \tag{C.20}$$

where $\mathbf{c}_j^m \in \mathbb{R}^{q_m}$ is the jth column of the loading matrix C^m. All moments $\langle \cdot \rangle$ can be readily computed from the approximate posterior distributions given in equations C.6 to C.10.

C.1.2 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. To optimize timescales, we rewrite the lower bound, $\widetilde{L}(\widetilde{Q}, \Omega)$, to show the terms that depend on the PSD matrix S_j (from equation 3.25). Let

$$\widetilde{L}(\widetilde{Q}, \Omega) = \sum_{j=1}^p \left[\frac{N}{2} \log \left| S_j^{-1} \right| - \frac{1}{2} \sum_{n=1}^N \mathrm{tr} \left(S_j^{-1} \left\langle \widetilde{\mathbf{x}}_{n,j,:} \widetilde{\mathbf{x}}_{n,j,:}^{\mathsf{H}} \right\rangle \right) \right] + \text{const.} \tag{C.21}$$

We further make the change of variables $\gamma_j = 1/\tau_j^2$. The variable γ_j is simpler to work with. We then optimize with respect to γ_j. The γ_j gradients for latent j are then given by the chain rule,

$$\frac{\partial \widetilde{L}}{\partial \gamma_j} = \mathrm{tr} \left(\left(\frac{\partial \widetilde{L}}{\partial S_j} \right)^\top \left(\frac{\partial S_j}{\partial \gamma_j} \right) \right), \tag{C.22}$$

where

$$\frac{\partial \widetilde{L}}{\partial S_j} = -\frac{N}{2} S_j^{-1} + \frac{1}{2} S_j^{-1} \left(\sum_{n=1}^{N} \langle \widetilde{\mathbf{x}}_{n,j,:} \widetilde{\mathbf{x}}_{n,j,:}^{\mathsf{H}} \rangle \right) S_j^{-1}. \tag{C.23}$$

Because S_j is a diagonal matrix, so is $\partial S_j / \partial \gamma_j$. The lth element along the diagonal of $\partial S_j / \partial \gamma_j$ is then given by

$$\frac{\partial s_j(f_l)}{\partial \gamma_j} = \left(1 - \sigma_j^2 \right) \sqrt{\frac{\pi}{2}} \exp \left(\frac{-(2\pi f_l)^2}{2\gamma_j} \right) \left[(2\pi f_l)^2 \gamma_j^{-5/2} - \gamma_j^{-3/2} \right]. \tag{C.24}$$

To optimize γ_j while respecting nonnegativity constraints, we perform the change of variables $\gamma_j = \exp(\hat{\gamma}_j)$ and then perform unconstrained gradient ascent with respect to $\hat{\gamma}_j$.

Naive computation of the posterior second moment $\langle \widetilde{\mathbf{x}}_{n,j,:} \widetilde{\mathbf{x}}_{n,j,:}^{\mathsf{H}} \rangle$ in equation C.23 would lead to a gradient evaluation (equation C.22) that costs $\mathcal{O}(T^2)$ operations. This cost can be improved to $\mathcal{O}(T)$ by exploiting the diagonal structure of S_j and its gradient $\partial S_j / \partial \gamma_j$. Substituting equation C.23 into equation C.22, we get

$$\frac{\partial \widetilde{L}}{\partial \gamma_j} = \mathrm{tr} \left(\left(\left(-\frac{N}{2} S_j^{-1} + \frac{1}{2} S_j^{-1} \left(\sum_{n=1}^{N} \langle \widetilde{\mathbf{x}}_{n,j,:} \widetilde{\mathbf{x}}_{n,j,:}^{\mathsf{H}} \rangle \right) S_j^{-1} \right)^{\top} \left(\frac{\partial S_j}{\partial \gamma_j} \right) \right),$$

$$= -\frac{N}{2} \mathrm{tr} \left(S_j^{-1} \frac{\partial S_j}{\partial \gamma_j} \right) + \frac{1}{2} \mathrm{tr} \left(\sum_{n=1}^{N} \langle \widetilde{\mathbf{x}}_{n,j,:} \widetilde{\mathbf{x}}_{n,j,:}^{\mathsf{H}} \rangle^{\top} S_j^{-1} \frac{\partial S_j}{\partial \gamma_j} S_j^{-1} \right). \tag{C.25}$$

Because of the diagonal structure of S_j and its gradient $\partial S_j / \partial \gamma_j$, however, both trace terms can be computed efficiently, and only the diagonal elements of $\langle \widetilde{\mathbf{x}}_{n,j,:} \widetilde{\mathbf{x}}_{n,j,:}^{\mathsf{H}} \rangle$ are needed:

$$\frac{\partial \widetilde{L}}{\partial \gamma_j} = \sum_{l=1}^{T} \left(s_j^{-1}(f_l) \frac{\partial s_j(f_l)}{\partial \gamma_j} + \left(\sum_{n=1}^{N} \langle |\widetilde{x}_{n,j,l}|^2 \rangle \right) s_j^{-1}(f_l) \frac{\partial s_j(f_l)}{\partial \gamma_j} s_j^{-1}(f_l) \right). \tag{C.26}$$

The evaluation of GP timescale gradients therefore requires only $\mathcal{O}(T)$ operations.

The diagonal structure of the phase shift matrix H_l^m can be similarly exploited to produce efficient time delay parameter updates. To optimize time delays, we rewrite the lower bound to show the terms that depend on H_l^m. Let

$$\widetilde{L}_{n,l}^m = -\frac{1}{2} \mathrm{tr} \left(\langle C^{m\top} \Phi^m C^m \rangle H_l^m \langle \widetilde{\mathbf{x}}_{n,l} \widetilde{\mathbf{x}}_{n,l}^{\mathsf{H}} \rangle H_l^{m\mathsf{H}} \right)$$

$$+ \frac{1}{2}\mathrm{tr}\left(H_l^m \langle \widetilde{\mathbf{x}}_{n,l} \rangle (\widetilde{\mathbf{y}}_{n,l}^m - \langle \widetilde{\mathbf{d}}_l^m \rangle)^{\mathsf{H}} \langle \Phi^m \rangle \langle C^m \rangle \right)$$

$$+ \frac{1}{2}\mathrm{tr}\left(\langle C^m \rangle^{\top} \langle \Phi^m \rangle (\widetilde{\mathbf{y}}_{n,l}^m - \langle \widetilde{\mathbf{d}}_l^m \rangle) \langle \widetilde{\mathbf{x}}_{n,l} \rangle^{\mathsf{H}} H_l^{m\mathsf{H}} \right). \tag{C.27}$$

Then

$$\widetilde{L}(\widetilde{Q}, \Omega) = \sum_{n=1}^{N} \sum_{l=1}^{T} \sum_{m=1}^{M} \widetilde{L}_{n,l}^m + \text{const.} \tag{C.28}$$

Since H_l^m is generally complex valued, we follow the Wirtinger calculus (Schreier & Scharf, 2010; Wirtinger, 1927), and treat the lower-bound $\widetilde{L}(\widetilde{Q}, \Omega)$ formally as a function of both H_l^m and its complex conjugate $(H_l^m)^*$. Then delay gradients for group m and latent j are given by the Wirtinger chain rule, which requires partial derivatives with respect to both H_l^m and $(H_l^m)^*$:

$$\frac{\partial \widetilde{L}}{\partial D_j^m} = \sum_{n=1}^{N} \sum_{l=1}^{T} \frac{\partial \widetilde{L}_{n,l}^m}{\partial D_j^m} \tag{C.29}$$

$$= \sum_{n=1}^{N} \sum_{l=1}^{T} \left[\mathrm{tr}\left(\left(\frac{\partial \widetilde{L}_{n,l}^m}{\partial H_l^m} \right)^{\top} \left(\frac{\partial H_l^m}{\partial D_j^m} \right) \right) + \mathrm{tr}\left(\left(\frac{\partial \widetilde{L}_{n,l}^m}{\partial (H_l^m)^*} \right)^{\top} \left(\frac{\partial (H_l^m)^*}{\partial D_j^m} \right) \right) \right], \tag{C.30}$$

The partial derivatives $\frac{\partial H_l^m}{\partial D_j^m} \in \mathbb{C}^{p \times p}$ and $\frac{\partial (H_l^m)^*}{\partial D_j^m} \in \mathbb{C}^{p \times p}$ are each single-entry matrices with

$$\frac{\partial H_l^m}{\partial D_j^m} = \mathrm{diag}\left(0, \ldots, -i2\pi f_l \cdot h_{j,l}^m, \ldots, 0 \right) \tag{C.31}$$

and

$$\frac{\partial (H_l^m)^*}{\partial D_j^m} = \mathrm{diag}\left(0, \ldots, i2\pi f_l \cdot (h_{j,l}^m)^*, \ldots, 0 \right), \tag{C.32}$$

where $h_{j,l}^m = \exp(-i2\pi f_l D_j^m)$. Each $\frac{\partial \widetilde{L}_{n,l}^m}{\partial H_l^m} \in \mathbb{C}^{p \times p}$ and $\frac{\partial \widetilde{L}_{n,l}^m}{\partial (H_l^m)^*} \in \mathbb{C}^{p \times p}$ are given by

$$\frac{\partial \widetilde{L}_{n,l}^m}{\partial H_l^m} = -\frac{1}{2} \langle C^{m\top} \Phi^m C^m \rangle (H_l^m)^* \langle \widetilde{\mathbf{x}}_{n,l} \widetilde{\mathbf{x}}_{n,l}^{\mathsf{H}} \rangle^{\top}$$

$$+ \frac{1}{2} \langle C^m \rangle^{\top} \langle \Phi^m \rangle \left(\widetilde{\mathbf{y}}_{n,l}^m - \langle \widetilde{\mathbf{d}}_l^m \rangle \right)^* \langle \widetilde{\mathbf{x}}_{n,l} \rangle^{\top} \tag{C.33}$$

$$\frac{\partial \widetilde{L}_{n,l}^m}{\partial (H_l^m)^*} = -\frac{1}{2} \langle \widetilde{\mathbf{x}}_{n,l} \widetilde{\mathbf{x}}_{n,l}^{\mathsf{H}} \rangle^{\top} H_l^m \langle C^{m\top} \Phi^m C^m \rangle$$

$$+ \frac{1}{2} \langle \widetilde{\mathbf{x}}_{n,l} \rangle^* (\widetilde{\mathbf{y}}_{n,l}^m - \langle \widetilde{\mathbf{d}}_l^m \rangle)^{\top} \langle \Phi^m \rangle \langle C^m \rangle. \tag{C.34}$$

The overall delay gradient $\frac{\partial \widetilde{L}}{\partial D_j^m}$ then evaluates to

$$\frac{\partial \widetilde{L}}{\partial D_j^m} = \sum_{n=1}^{N} \sum_{l=1}^{T} \Re \Bigg\{ i2\pi f_l \cdot h_{j,l}^m \cdot \Bigg(\sum_{k=1}^{p} \langle \widetilde{x}_{n,j,l} \widetilde{x}_{n,k,l}^* \rangle \cdot (h_{k,l}^m)^* \cdot \langle \mathbf{c}_k^{m\top} \Phi^m \mathbf{c}_j^m \rangle$$

$$- \langle \widetilde{x}_{n,j,l} \rangle (\widetilde{\mathbf{y}}_{n,l}^m - \langle \widetilde{\mathbf{d}}_l^m \rangle)^{\mathsf{H}} \langle \Phi^m \rangle \langle \mathbf{c}_j^m \rangle \Bigg) \Bigg\}, \tag{C.35}$$

which costs $\mathcal{O}(T)$ operations.

In practice, we fix all delay parameters for group 1 at 0 to ensure identifiability. Similar to the timescales, one might wish to constrain the delays within some physically realistic range, such as the length of an experimental trial, so that $-D_{\max} \leq D_j^m \leq D_{\max}$. Toward that end, we make the change of variables $D_j^m = D_{\max} \cdot \tanh(\frac{\hat{D}_j^m}{2})$ and perform unconstrained gradient ascent with respect to $\hat{D}_j^m$. Here we chose $D_{\max}$ to be half the length of a trial.

C.2 Evaluation of the Lower Bound. To monitor the progress of the fitting procedure, we evaluate the lower bound, $\widetilde{L}(\widetilde{Q}, \Omega)$, on each iteration. To evaluate the lower bound, we can rewrite it as follows:

$$\widetilde{L}(\widetilde{Q}, \Omega) = \mathbb{E}_{\widetilde{Q}}[\log P(\widetilde{Y}|\theta, \Omega)] - \mathrm{KL}(\widetilde{Q}(\theta) \| P(\theta|\Omega)). \tag{C.36}$$

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

$$\widetilde{L}(\widetilde{Q}, \Omega) = \mathbb{E}_{\widetilde{Q}}[\log P(\widetilde{Y}|\theta, \Omega)] - \mathrm{KL}(Q_{\widetilde{x}}(\widetilde{X}) \| P(\widetilde{X}|\Omega)) - \mathrm{KL}(Q_c(C) \| P(C|\mathcal{A}))$$

$$- \mathrm{KL}(Q_{\mathcal{A}}(\mathcal{A}) \| P(\mathcal{A})) - \mathrm{KL}(Q_\phi(\phi) \| P(\phi)) - \mathrm{KL}(Q_d(\mathbf{d}) \| P(\mathbf{d})). \tag{C.37}$$

This form of the ELBO provides insight into the nature of the optimization procedure for fitting via mDLAG-frequency. The first term is the expected log-quasi-likelihood (with respect to the approximate posterior $\widetilde{Q}(\theta)$) of the frequency domain observations, $\widetilde{Y}$, given the latest model parameters, θ and Ω. This term encourages mDLAG-frequency models to explain the frequency domain observations 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 C.1.1 and the prior definitions in section 3.3, each term of the lower bound can be computed as follows:

$$
\mathbb{E}_{\widetilde{Q}}[\log P(\widetilde{Y}|\theta, \Omega)] = -\frac{qNT}{2}\log(2\pi) + \frac{NT}{2}\sum_{m=1}^{M}\sum_{r=1}^{q_m}\langle\log\phi_r^m\rangle
$$
$$
- \sum_{m=1}^{M}\sum_{r=1}^{q_m}(\bar{a}_\phi - \langle\phi_r^m\rangle b_\phi) \tag{C.38}
$$

$$
-\mathrm{KL}(Q_{\widetilde{x}}(\widetilde{X})\|P(\widetilde{X}|\Omega)) = \frac{pNT}{2} + \frac{N}{2}\sum_{l=1}^{T}\log|\widetilde{\Sigma}_{x,l}| - \frac{N}{2}\sum_{l=1}^{T}\sum_{j=1}^{p}\log(s_j(f_l))
$$
$$
- \frac{1}{2}\sum_{l=1}^{T}\sum_{j=1}^{p}s_j^{-1}(f_l)\sum_{n=1}^{N}\langle|\widetilde{x}_{n,j,l}|^2\rangle \tag{C.39}
$$

$$
-\mathrm{KL}(Q_c(C)\|P(C|\mathcal{A})) = \frac{qp}{2} + \sum_{m=1}^{M}\left[\frac{1}{2}\sum_{r=1}^{q_m}\log|\Sigma_{c_r}^m|\right.
$$
$$
\left. + \sum_{j=1}^{p}\left[\frac{q_m}{2}\langle\log\alpha_j^m\rangle - \frac{1}{2}\langle\alpha_j^m\rangle\langle\|\mathbf{c}_j^m\|_2^2\rangle\right]\right] \tag{C.40}
$$

$$
-\mathrm{KL}(Q_{\mathcal{A}}(\mathcal{A})\|P(\mathcal{A})) = \sum_{m=1}^{M}\sum_{j=1}^{p}\left[-\bar{a}_\alpha^m\log\bar{b}_{\alpha,j}^m + a_\alpha\log b_\alpha + \log\frac{\Gamma(\bar{a}_\alpha^m)}{\Gamma(a_\alpha)}\right.
$$
$$
\left. - b_\alpha\langle\alpha_j^m\rangle + \bar{a}_\alpha^m + (a_\alpha - \bar{a}_\alpha^m)(\Psi(\bar{a}_\alpha^m) - \log\bar{b}_{\alpha,j}^m)\right] \tag{C.41}
$$

$$
-\mathrm{KL}(Q_\phi(\phi)\|P(\phi)) = \sum_{m=1}^{M}\sum_{r=1}^{q_m}\left[-\bar{a}_\phi\log\bar{b}_{\phi,r}^m + a_\phi\log b_\phi + \log\frac{\Gamma(\bar{a}_\phi)}{\Gamma(a_\phi)}\right.
$$
$$
\left. - b_\phi\langle\phi_r^m\rangle + \bar{a}_\phi + (a_\phi - \bar{a}_\phi)(\Psi(\bar{a}_\phi) - \log\bar{b}_{\phi,r}^m)\right] \tag{C.42}
$$

$$
-\mathrm{KL}(Q_d(\mathbf{d})\|P(\mathbf{d})) = \frac{q}{2} + \frac{q}{2}\log\beta + \frac{1}{2}\log|\Sigma_d| - \frac{1}{2}\beta\langle\|\mathbf{d}\|_2^2\rangle. \tag{C.43}
$$

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 C.6 to C.10.

Appendix D: Leave-Group-Out Prediction

Each of the three methods we considered here is designed to optimize a slightly different objective function (mDLAG-time: equation 3.10; mDLAG-inducing: equation B.3, mDLAG-frequency: equation C.3). We therefore developed a common performance metric to facilitate comparison across these methods: leave-group-out prediction (Gokcen et al., 2023; Klami et al., 2015). Leave-group-out prediction measures an mDLAG model's ability to capture interactions across observed groups. In brief, we used a model fit to training data to predict (on held-out test trials) the unobserved activity of held-out units in one group, given the observed activity of units in the remaining groups. Our three methods each give rise to a distinct way to perform this prediction.

D.1 Prediction via mDLAG-Time. We first collect observed variables (for one trial) in a manner that highlights group structure. We collect observations in group m on trial n in $\check{\mathbf{y}}_n^m = [\mathbf{y}_{n,:,1}^{m\top} \cdots \mathbf{y}_{n,:,T}^{m\top}]^\top \in \mathbb{R}^{q_m T}$ by vertically concatenating the observed activity $\mathbf{y}_{n,:,t}^m \in \mathbb{R}^{q_m}$ in group m across all times $t = 1, \ldots, T$. Then we collect the observations for the remaining $M - 1$ groups in $\check{\mathbf{y}}_n^{-m} \in \mathbb{R}^{\sum_{k \neq m} q_k T}$, obtained by vertically concatenating the ordered set of observations $\{\check{\mathbf{y}}_n^k\}_{k \neq m}$.

Our goal is to predict $\check{\mathbf{y}}_n^m$ given $\check{\mathbf{y}}_n^{-m}$. We do so by inferring the latents given observations $\check{\mathbf{y}}_n^{-m}$ and then predicting the held-out activity $\check{\mathbf{y}}_n^m$ from the inferred latents. Toward that end, we similarly collect the latents for group m on trial n in $\check{\mathbf{x}}_n^m = [\mathbf{x}_{n,:,1}^{m\top} \cdots \mathbf{x}_{n,:,T}^{m\top}]^\top \in \mathbb{R}^{pT}$ by vertically concatenating the latents $\mathbf{x}_{n,:,t}^m \in \mathbb{R}^p$ across all times $t = 1, \ldots, T$. Then the latents for the remaining $M - 1$ groups can be collected in $\check{\mathbf{x}}_n^{-m} \in \mathbb{R}^{(M-1)pT}$, obtained by vertically concatenating the ordered set of latents $\{\check{\mathbf{x}}_n^k\}_{k \neq m}$. This variable reorganization then allows us to rewrite the mDLAG-time state model as

$$\begin{bmatrix} \check{\mathbf{x}}_n^m \\ \check{\mathbf{x}}_n^{-m} \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} \check{K}_{m,m} & \check{K}_{m,-m} \\ \check{K}_{-m,m} & \check{K}_{-m,-m} \end{bmatrix} \right), \tag{D.1}$$

where the elements of the GP covariance matrices $\check{K}_{m,m} \in \mathbb{S}^{pT \times pT}$, $\check{K}_{m,-m} = \check{K}_{-m,m}^\top \in \mathbb{R}^{pT \times (M-1)pT}$, and $\check{K}_{-m,-m} \in \mathbb{S}^{(M-1)pT \times (M-1)pT}$ are computed using equations 3.8 and 3.9.

Next, for each group m, define $\langle\check{C}^m\rangle \in \mathbb{R}^{q_m T \times pT}$, $\langle\check{\Phi}^m\rangle \in \mathbb{S}^{q_m T \times q_m T}$, and $\langle\check{R}^m\rangle \in \mathbb{R}^{pT \times pT}$ as block diagonal matrices comprising T copies of the matrices $\langle C^m\rangle$, $\langle\Phi^m\rangle$, and $\langle R^m\rangle = \langle C^{m\top}\Phi^m C^m\rangle$, respectively. Define also $\langle\check{\mathbf{d}}^m\rangle \in \mathbb{R}^{q_m T}$ by vertically concatenating T copies of $\langle\mathbf{d}^m\rangle$. The parameters corresponding to the remaining $M - 1$ populations can then be collected into the block diagonal matrices $\langle\check{C}^{-m}\rangle = \text{diag}(\{\langle\check{C}^k\rangle\}_{k \neq m}) \in \mathbb{R}^{\sum_{k \neq m} q_k T \times (M-1)pT}$, $\langle\check{\Phi}^{-m}\rangle = \text{diag}(\{\langle\check{\Phi}^k\rangle\}_{k \neq m}) \in \mathbb{R}^{\sum_{k \neq m} q_k T \times \sum_{k \neq m} q_k T}$, $\langle\check{R}^{-m}\rangle = \text{diag}(\{\langle\check{R}^k\rangle\}_{k \neq m}) \in$

$\mathbb{R}^{(M-1)pT \times (M-1)pT}$, and the vector $\langle \check{\mathbf{d}}^{-m} \rangle \in \mathbb{R}^{\sum_{k \neq m} q_k T}$, obtained by vertically concatenating the elements of the set $\{\langle \check{\mathbf{d}}^k \rangle\}_{k \neq m}$.

Similar to the update equations for the posterior distribution over latents ($Q_x(X)$, equation 3.13), we compute the inferred latents given only the observations $\check{\mathbf{y}}_n^{-m}$ according to

$$\check{\Sigma}_x^{-m} = \left(\check{K}_{-m,-m}^{-1} + \langle \check{R}^{-m} \rangle \right)^{-1}, \tag{D.2}$$

$$\check{\boldsymbol{\mu}}_{x_n}^{-m} = \check{\Sigma}_x^{-m} \langle \check{C}^{-m} \rangle^\top \langle \check{\Phi}^{-m} \rangle \left(\check{\mathbf{y}}_n^{-m} - \langle \check{\mathbf{d}}^{-m} \rangle \right). \tag{D.3}$$

We then use equation D.1 to infer the latents for group m according to

$$\check{\boldsymbol{\mu}}_{x_n}^m = \check{K}_{m,-m} (\check{K}_{-m})^{-1} \check{\boldsymbol{\mu}}_{x_n}^{-m} \tag{D.4}$$

and take predictions of the observations in group m to be

$$\overset{\star}{\mathbf{y}}_n^m = \langle \check{C}^m \rangle \check{\boldsymbol{\mu}}_{x_n}^m + \langle \check{\mathbf{d}}^m \rangle. \tag{D.5}$$

We employed this approach for models fit via mDLAG-time and for models fit via mDLAG-frequency in Figure 5B and supplementary Figure 4A.

D.2 Prediction via mDLAG-Inducing. Predictions via inducing variables take the same form as equation D.5. However, the predicted latents in group m, $\check{\boldsymbol{\mu}}_{x_n}^m$, are computed differently than those in equation D.4. Similar to the updates of the posterior distribution over the inducing variables, $Q_w(W)$ (equations B.22 and B.23), we compute the inferred inducing variables given observations in all but the held-out mth group.

Toward that end, we first define several variables. First construct $\bar{\mathbf{w}}_n = [\mathbf{w}_{n,1,:}^\top \cdots \mathbf{w}_{n,p,:}^\top]^\top \in \mathbb{R}^{pT_{\text{ind}}}$ by vertically concatenating the inducing variables $\mathbf{w}_{n,j,:}$ for latents $j = 1, \ldots, p$ on trial n. Collect the inducing variable covariance matrices K_j^w for $j = 1, \ldots, p$ (equation 3.15; Figure 1D, top) into the block diagonal matrix $\bar{K}^w = \text{diag}(K_1^w, \ldots, K_p^w) \in \mathbb{S}^{pT_{\text{ind}} \times pT_{\text{ind}}}$. Then let $\mathbf{k}_{m,j,t}^{xw} \in \mathbb{R}^{T_{\text{ind}}}$ be the tth row of $K_{m,j}^{xw} \in \mathbb{R}^{T \times T_{\text{ind}}}$, the mth block (for group m) of the covariance matrix between latent j and its inducing variable, K_j^{xw} (equation 3.18; Figure 1D, bottom). For time point t, collect each $\mathbf{k}_{m,j,t}^{xw}$ for latents $j = 1, \ldots, p$ into the block diagonal matrix $\bar{K}_{m,t}^{xw} = \text{diag}(\mathbf{k}_{m,1,t}^{xw\top}, \ldots, \mathbf{k}_{m,p,t}^{xw\top}) \in \mathbb{R}^{p \times pT_{\text{ind}}}$.

Given observations in all but the mth group, we can then evaluate the posterior covariance, $\bar{\Sigma}_w^{-m} \in \mathbb{S}^{pT_{\text{ind}} \times pT_{\text{ind}}}$, and mean, $\bar{\boldsymbol{\mu}}_{w_n}^{-m} \in \mathbb{R}^{pT_{\text{ind}}}$, of $\bar{\mathbf{w}}_n$ for

each trial n:

$$\bar{\Sigma}_w^{-m} = \left((\bar{K}^w)^{-1} + (\bar{K}^w)^{-1} \left[\sum_{k \neq m} \sum_{t=1}^{T} \bar{K}_{k,t}^{wx} \langle (C^k)^\top \Phi^k C^k \rangle \bar{K}_{k,t}^{xw} \right] (\bar{K}^w)^{-1} \right)^{-1},$$

(D.6)

$$\bar{\mu}_{w_n}^{-m} = \bar{\Sigma}_w (\bar{K}^w)^{-1} \left[\sum_{k \neq m} \sum_{t=1}^{T} \bar{K}_{k,t}^{wx} \langle C^k \rangle^\top \langle \Phi^k \rangle \left(\mathbf{y}_{n,t}^k - \langle \mathbf{d}^k \rangle \right) \right].$$

(D.7)

If we collect the covariance matrices $K_{m,j}^{xw}$ for group m and for latents $j = 1, \ldots, p$ into the block diagonal matrix $\bar{K}_m^{xw} = \mathrm{diag}(K_{m,1}^{xw}, \ldots, K_{m,p}^{xw}) \in \mathbb{R}^{pT \times pT_{\mathrm{ind}}}$, then we can compute the predicted latents in group m based on equation B.26:

$$\bar{\mu}_{x_n}^m = \bar{K}_m^{xw} (\bar{K}^w)^{-1} \bar{\mu}_{w_n}^{-m}.$$

(D.8)

The vector of predicted latents in equation D.5, $\check{\mu}_{x_n}^m$, can be obtained by a reorganization (permutation) of the elements in $\bar{\mu}_{x_n}^m$, above. Then predicted observations can be obtained directly from equation D.5. We employed this approach for models fit via mDLAG-inducing in Figure 5B and supplementary Figure 4A.

D.3 Prediction via mDLAG-Frequency. As in section 3.3, we now consider the unitary DFT of the time series of observations for unit r in group m on trial n: $\widetilde{\mathbf{y}}_{n,r,:}^m := U_T \mathbf{y}_{n,r,:}^m$, where $U_T \in \mathbb{C}^{T \times T}$ is the unitary DFT matrix. Then let $\widetilde{\mathbf{y}}_{n,l}^m \in \mathbb{R}^{q_m}$ be the frequency domain observations of group m on trial n and frequency l, and further collect the observations for the remaining $M - 1$ groups on trial n and frequency l in the set $\widetilde{Y}_{n,l}^{-m} = \{\widetilde{\mathbf{y}}_{n,l}^k\}_{k \neq m}$. Ultimately, our goal is still to predict the time domain observations $\mathbf{y}_{n,r,:}^m$ for all units $r = 1, \ldots, q_m$ in group m. We will do so, however, by using the frequency domain observations $\widetilde{Y}_{n,l}^{-m}$ and performing inference of the latents in the frequency domain.

Similar to the updates of the posterior distribution over the frequency domain latents, $Q_{\widetilde{x}}(\widetilde{X})$ (equations C.11 and C.12), we compute the inferred latents given the frequency domain observations $\widetilde{Y}_{n,l}^{-m}$ according to

$$\widetilde{\Sigma}_{x,l}^{-m} = \left(S_l^{-1} + \sum_{k \neq m} (H_l^k)^{\mathsf{H}} \langle (C^k)^\top \Phi^k C^k \rangle H_l^k \right)^{-1},$$

(D.9)

$$\widetilde{\mu}_{x_{n,l}}^{-m} = \widetilde{\Sigma}_{x,l}^{-m} \sum_{k \neq m} (H_l^k)^{\mathsf{H}} \langle C^k \rangle^\top \langle \Phi^k \rangle \left(\widetilde{\mathbf{y}}_{n,l}^k - \langle \widetilde{\mathbf{d}}_l^k \rangle \right).$$

(D.10)

Recall the diagonal phase shift matrix $H_l^m \in \mathbb{C}^{p \times p}$, defined in equation 3.28. We then take predictions of the frequency domain observations in group m to be

$$\overset{\star}{\widetilde{\mathbf{y}}}_{n,l}^m = \langle C^m \rangle H_l^m \widetilde{\boldsymbol{\mu}}_{x_{n,l}}^{-m} + \langle \widetilde{\mathbf{d}}_l^m \rangle. \tag{D.11}$$

To convert these predictions from the frequency domain to the time domain, we take the inverse unitary DFT of the frequency domain activity for each unit $r = 1, \ldots, q_m$ in group m on each trial n: $\overset{\star}{\mathbf{y}}_{n,r,:}^m = U_T^{\mathsf{H}} \overset{\star}{\widetilde{\mathbf{y}}}_{n,r,:}^m$. We employed this approach in Figures 3A and 4A; see section D.5.

D.4 A Common Performance Metric: Leave-Group-Out R^2. Let $\mathbf{y}_{n,t}^m \in \mathbb{R}^{q_m}$ be the observed activity for group m at time t on trial n of a held-out test set, and let $\overset{\star}{\mathbf{y}}_{n,t}^m \in \mathbb{R}^{q_m}$ be its predicted value. This prediction may come from any of the methods above (sections D.1–D.3). Furthermore, let $\boldsymbol{\mu}_y^m \in \mathbb{R}^{q_m}$ be the sample mean for each unit in group m, taken over all time points and trials.

We then define a leave-group-out R^2 value as follows:

$$R_{\text{lgo}}^2 = 1 - \frac{\sum_{m=1}^M \sum_{n=1}^N \sum_{t=1}^T ||\mathbf{y}_{n,t}^m - \overset{\star}{\mathbf{y}}_{n,t}^m||_F^2}{\sum_{m=1}^M \sum_{n=1}^N \sum_{t=1}^T ||\mathbf{y}_{n,t}^m - \boldsymbol{\mu}_y^m||_F^2}, \tag{D.12}$$

where $|| \cdot ||_F$ is the Frobenius norm. The value $R_{lgo}^2 \in (-\infty, 1]$, where a value of 1 implies perfect prediction of held-out observations and a negative value implies that estimates predict these held-out observations less accurately than simply the sample mean. The R_{lgo}^2 metric is normalized by the total variance of observed activity within each data set, thereby facilitating comparison across data sets, in which the variance of observed activity could vary widely.

D.5 Leave-Unit-Out Prediction. In the group number scaling experiments (see section 4.3, Figure 4), the number of groups, M, varied across simulated data sets while the number of units in total, q, remained fixed (see Table 1, Scaling, M). A fairer metric for comparison across data sets (as in Figure 4A), then, would involve not leave-group-out prediction but rather leave-unit-out prediction. To create a leave-unit-out prediction metric, we leveraged the development of leave-group-out prediction.

Suppose we have a fitted mDLAG model, for M groups and q_m units per group, where $m = 1, \ldots, M$. We can then simply re-index the parameters of this model so that each unit $r = 1, \ldots, q$, with $q = \sum_m q_m$, is treated as its own "group." The same equations developed for any of the leave-group-out prediction methods, above (equations D.5, D.8, or D.11 for generating

predictions, and equation D.12 for measuring performance) can then be used for leave-unit-out prediction: replace the index $m = 1, \ldots, M$ over groups with the index $r = 1, \ldots, q$ over individual units.

Generating leave-unit-out predictions via mDLAG-time (section D.1), however, was prohibitively expensive for the group number scaling experiments (see Figure 4A). Computationally, predictions require $\mathcal{O}(q^3)$ operations. We therefore generated predictions via the more efficient mDLAG-frequency (see section D.3; $\mathcal{O}(q)$ operations), regardless of which method (mDLAG-time, mDLAG-inducing, mDLAG-frequency) was used to fit the given model parameters.

Yet predictions via mDLAG-frequency exhibit edge effects (see section 4.5 and Figure 6C). To mitigate the impact of these edge effects on our measures of performance, we evaluated equation D.12 for only a middle portion of the test trials. For squared exponential GP timescales of 100 ms, removing the first and final 10 time points (200 ms) from evaluation was sufficient for edge effects to be rendered neglible. For the group number scaling experiments, where there were $T = 50$ time points per test trial, we therefore used only the middle 30 time points ($t = 11$ through 40). We also employed leave-unit-out prediction via mDLAG-frequency in the trial-length scaling experiments (section 4.2 and Figure 3), for consistency across simulated experiments and for computational benefit. Test trials comprised $T = 500$ time points, but we similarly used only the middle 480 time points ($t = 11$ through 490).

Appendix E: Reproducibility, Code Availability, and Data Availability. All numerical and statistical analyses described in this work were performed in Matlab (MathWorks; version 2024a). Numerical results were obtained on a Red Hat Enterprise Linux machine (release 7.9, 64-bit) with 219.68 GB of RAM, on an Intel Xeon Gold 6140 CPU (2.3 GHz). An implementation of mDLAG-time is available on GitHub at http://github.com/egokcen/mDLAG and on Zenodo at https://doi.org/10.5281/zenodo.10048163 (Gokcen, 2023). Implementations of mDLAG-inducing and mDLAG-frequency, as well as code and data to reproduce figures and any reported p-values, are available on GitHub at https://github.com/egokcen/fast-mDLAG and on Zenodo at https://doi.org/10.5281/zenodo.15305287 (Gokcen, 2025b). A frequency domain implementation of the non-Bayesian DLAG is also available on GitHub at https://github.com/egokcen/DLAG and on Zenodo at https://doi.org/10.5281/zenodo.15304817 (Gokcen, 2025a).

Acknowledgments

Author Contributions

E.G.: conceptualization, data curation (simulations), formal analysis, investigation, methodology, software, validation, visualization, writing— original draft. A.I.J.: data curation (Neuropixels recordings), resources (Neuropixels recordings), writing—review and editing. A.K.: data curation (Neuropixels recordings), funding acquisition, resources (Neuropixels recordings), supervision, writing—review and editing. C.K.M.: funding acquisition, supervision, writing—review and editing. B.M.Y: funding acquisition, resources (computing), supervision, writing—review and editing.

Competing Interests

We declare no competing interests.

References

Ahrens, M. B., Li, J. M., Orger, M. B., Robson, D. N., Schier, A. F., Engert, F., & Portugues, R. (2012). Brain-wide neuronal dynamics during motor adaptation in zebrafish. *Nature*, *485*(7399), 471–477. 10.1038/nature11057

Álvarez, M., Luengo, D., Titsias, M., & Lawrence, N. D. (2010). Efficient multi-output gaussian processes through variational inducing kernels. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics* (pp. 25–32).

Álvarez, M., Rosasco, L., & Lawrence, N. D. (2012). Kernels for vector-valued functions: A review. *Foundations and Trends in Machine Learning*, *4*(3), 195–266.

Aoi, M. C., & Pillow, J. W. (2017). *Scalable Bayesian inference for high- dimensional neural receptive fields*. bioRxiv.

Balzani, E., Noel, J.-P. G., Herrero-Vidal, P., Angelaki, D. E., & Savin, C. (2023). A probabilistic framework for task-aligned intra- and inter-area neural manifold estimation. In *Eleventh International Conference on Learning Representations*.

Bishop, C. (1999). Variational principal components. In *Proceedings of the 9th International Conference on Artificial Neural Networks*, (pp. 509–514).

Bong, H., Liu, Z., Ren, Z., Smith, M., Ventura, V., & Robert, K. E. (2020). Latent dynamic factor analysis of high-dimensional neural recordings. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, & H. Lin (Eds.), *Advances in neural information processing systems*, *33* (pp. 16446–16456). Curran.

Buesing, L., Machado, T. A., Cunningham, J. P., & Paninski, L. (2014). Clustered factor analysis of multineuronal spike data. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, & K. Q. Weinberger (Eds.), *Advances in neural information processing systems*, *27*(pp. 3500–3508). Curran.

Cowley, B. R., Snyder, A. C., Acar, K., Williamson, R. C., Yu, B. M., & Smith, M. A. (2020). Slow drift of neural activity as a signature of impulsivity in macaque visual and prefrontal cortex. *Neuron*, *108*(3), 551–567. 10.1016/j.neuron.2020.07.021

Cunningham, J. P., Shenoy, K. V., & Sahani, M. (2008). Fast gaussian process methods for point process intensity estimation. In *Proceedings of the 25th International Conference on Machine Learning* (pp. 192–199).

Cunningham, J. P., & Yu, B. M. (2014). Dimensionality reduction for large- scale neural recordings. *Nature Neuroscience*, *17*(11), 1500–1509. 10.1038/nn.3776

Dietrich, C. R., & Newsam, G. N. (1997). Fast and exact simulation of stationary gaussian processes through circulant embedding of the covariance matrix. *SIAM Journal on Scientific Computing*, *18*(4), 1088–1107. 10.1137/S1064827592240555

Dowling, M., Zhao, Y., & Park, I. M. (2023). Linear time GPs for inferring latent trajectories from neural spike trains. In *Proceedings of the 40th International Conference on Machine Learning* (pp. 8424–8448).

Duncker, L., Ruda, K. M., Field, G. D., & Pillow, J. W. (2023). Scalable variational inference for low-rank spatiotemporal receptive fields. *Neural Computation*, *35*(6), 995–1027. 10.1162/neco_a_01584

Duncker, L., & Sahani, M. (2018). Temporal alignment and latent gaussian process factor inference in population spike trains. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, & R. Garnett (Eds.), *Advances in neural information processing systems*, *31* (pp. 10445–10455). Curran.

Foreman-Mackey, D., Agol, E., Ambikasaran, S., & Angus, R. (2017). Fast and scalable gaussian process modeling with applications to astronomical time series. *Astronomical Journal*, *154*(6), 220. 10.3847/1538-3881/aa9332

Gallagher, N., Ulrich, K. R., Talbot, A., Dzirasa, K., Carin, L., & Carlson, D. E. (2017). Cross-spectral factor analysis. In A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. Gomez, . . . I. Polosukhin (Eds.), *Advances in neural information processing systems*, *30* (pp. 6842–6852).

Glaser, J., Whiteway, M., Cunningham, J. P., Paninski, L., & Linderman, S. (2020). Recurrent switching dynamical systems models for multiple interacting neural populations. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, & H. Lin (Eds.), *Advances in neural information processing systems*, *33* (pp. 14867–14878). Curran.

Gokcen, E. (2023). mDLAG (v1.0.0). *Zenodo*. 10.5281/zenodo.10048162.

Gokcen, E. (2025a). DLAG (v2.0.0). *Zenodo*. 10.5281/zenodo.15304817.

Gokcen, E. (2025b). Fast mDLAG (v1.0.0). *Zenodo*. 10.5281/zenodo.15305287.

Gokcen, E., Jasper, A. I., Semedo, J. D., Zandvakili, A., Kohn, A., Machens, C. K., & Yu, B. M. (2022). Disentangling the flow of signals between populations of neurons. *Nature Computational Science*, *2*(8), 512–525. 10.1038/s43588-022-00282-5

Gokcen, E., Jasper, A. I., Xu, A., Kohn, A., Machens, C. K., & Yu, B. M. (2023). Uncovering motifs of concurrent signaling across multiple neuronal populations. In A. Ohm, T. Naumann, A. Globerson, K. Saenko, M. Hardt, & S. Levine (Eds.), *Advances in neural information processing systems*, *36* (pp. 34711–34722). Curran.

Golub, G. H., & Van Loan, C. F. (2013). *Matrix computations* (4th ed.). Johns Hopkins University Press.

Gondur, R., Sikandar, U. B., Schaffer, E., Aoi, M. C., & Keeley, S. L. (2024). Multimodal gaussian process variational autoencoders for neural and behavioral data. In *Proceedings of the Twelfth International Conference on Learning Representations*.

Hartikainen, J., & Särkkä, S. (2010). Kalman filtering and smoothing solutions to temporal gaussian process regression models. In *Proceedings of the 2010 IEEE International Workshop on Machine Learning for Signal Processing* (pp. 379–384).

Hensman, J., Durrande, N., & Solin, A. (2018). Variational Fourier features for gaussian processes. *Journal of Machine Learning Research*, *18*(151), 1–52.

Hensman, J., Fusi, N., & Lawrence, N. D. (2013). Gaussian processes for big data. In *Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence* (pp. 282–290).

Jensen, K., Kao, T.-C., Stone, J., & Hennequin, G. (2021). Scalable Bayesian GPFA with automatic relevance determination and discrete noise models. In M. A. Beygelzimer, Y. Dauphin, P. S. Liang, & J. Wortman Vaughan (Eds.), *Advances in neural information processing systems*, *34* (pp. 10613–10626). Curran.

Kalman, R. E., & Bucy, R. S. (1961). New results in linear filtering and prediction theory. *Transactions of the ASME, Journal of Basic Engineering*, *83*(1), 95–108. 10.1115/1.3658902

Kang, B., & Druckmann, S. (2020). Approaches to inferring multi-regional interactions from simultaneous population recordings. *Current Opinion in Neurobiology*, *65*, 108–119. 10.1016/j.conb.2020.10.004

Karniol-Tambour, O., Zoltowski, D. M., Diamanti, E. M., Pinto, L., Brody, C. D., Tank, D. W., & Pillow, J. W. (2024). Modeling state-dependent communication between brain regions with switching nonlinear dynamical systems. In *Proceedings of the Twelfth International Conference on Learning Representations*.

Keeley, S., Aoi, M., Yu, Y., Smith, S., & Pillow, J. W. (2020). Identifying signal and noise structure in neural population activity with gaussian process factor models. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, & H. Lin (Eds.), *Advances in neural information processing systems*, *33* (pp. 13795–13805). Curran.

Keeley, S., Zoltowski, D. M., Aoi, M. C., & Pillow, J. W. (2020). Modeling statistical dependencies in multi-region spike train data. *Current Opinion in Neurobiology*, *65*, 194–202. 10.1016/j.conb.2020.11.005

Keeley, S., Zoltowski, D. M., Yu, Y., Yates, J. L., Smith, S. L., & Pillow, J. W. (2020). Efficient non-conjugate gaussian process factor models for spike count data using polynomial approximations. In *Proceedings of the 37th International Conference on Machine Learning*, *119* (pp. 5177–5186).

Klami, A., Virtanen, S., Leppäaho, E., & Kaski, S. (2015). Group factor analysis. *IEEE Transactions on Neural Networks and Learning Systems*, *26*(9), 2136–2147. 10.1109/TNNLS.2014.2376974

Kolmogorov, A. (1941). Stationary sequences in Hilbert space. *Moscow University Mathematics Bulletin*, *2*(6).

Lakshmanan, K. C., Sadtler, P. T., Tyler-Kabara, E. C., Batista, A. P., & Yu, B. M. (2015). Extracting low-dimensional latent structure from time series in the presence of delays. *Neural Computation*, *27*(9), 1825–1856. 10.1162/NECO_a_00759

Li, W., Li, C., Wang, Y., & Wu, A. (2024). Multi-region Markovian gaussian process: An efficient method to discover directional communications across multiple brain regions. In *Proceedings of the 41st International Conference on Machine Learning* (pp. 28112–28131).

Li, W., Wang, Y., Li, C., & Wu, A. (2024). *Markovian gaussian process: A universal state-space representation for stationary temporal gaussian process*. arXiv :2407.00397.

Lim, J., Park, J., Nah, S., & Choi, J. (2021). Multi-output infinite horizon gaussian processes. In *Proceedings of the 2021 IEEE International Conference on Robotics and Automation* (pp. 1542–1549).

Loper, J., Blei, D., Cunningham, J. P., & Paninski, L. (2021). A general linear-time inference method for gaussian processes on one dimension. *Journal of Machine Learning Research*, 22(234), 1–36.

Machado, T. A., Kauvar, I. V., & Deisseroth, K. (2022). Multiregion neuronal activity: The forest and the trees. *Nature Reviews Neuroscience*, 23(11), 683–704. 10.1038/s41583-022-00634-0

MacKay, D. J. (1994). Bayesian non-linear modeling for the energy prediction competition. *ASHRAE Transactions*, 100(2), 1053–1062.

Neal, R. M. (1995). *Bayesian learning for neural networks*. PhD diss., University of Toronto.

Paciorek, C. J. (2007). Bayesian smoothing with gaussian processes using Fourier basis functions in the spectral GP package. *Journal of Statistical Software*, 19, 1–38. 10.18637/jss.v019.i02

Parra, G., & Tobar, F. (2017). Spectral mixture kernels for multi-output gaussian processes. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, & R. Garnett (Eds.), *Advances in neural information processing systems*, 30 (pp. 6681–6690). Curran.

Priestley, M. B. (1981). *Spectral analysis and time series* (Vol. 1). Elsevier.

Quiñonero-Candela, J., & Rasmussen, C. E. (2005). A unifying view of sparseapproximate gaussian process regression. *Journal of Machine Learning Research*, 6(65), 1939–1959.

Rao, S. S., & Yang, J. (2021). Reconciling the gaussian and Whittle likelihood with an application to estimation in the frequency domain. *Annals of Statistics*, 49(5), 2774–2802.

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

Rauch, H. E., Tung, F., & Striebel, C. T. (1965). Maximum likelihood estimates of linear dynamic systems. *AIAA Journal*, 3(8), 1445–1450. 10.2514/3.3166

Sayed, A. H., & Kailath, T. (2001). A survey of spectral factorization methods. *Numerical Linear Algebra with Applications*, 8(6–7), 467–496. 10.1002/nla.250

Schreier, P. J., & Scharf, L. L. (2010). *Statistical signal processing of complex- valued data: The theory of improper and noncircular signals*. Cambridge University Press.

Semedo, J. D., Gokcen, E., Machens, C. K., Kohn, A., & Yu, B. M. (2020). Statistical methods for dissecting interactions between brain areas. *Current Opinion in Neurobiology*, 65, 59–69. 10.1016/j.conb.2020.09.009

Semedo, J. D., Zandvakili, A., Kohn, A., Machens, C. K., & Yu, B. M. (2014). Extracting latent structure from multiple interacting neural populations. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, & K. Q. Weinberger (Eds.), *Advances in neural information processing systems*, 27 (pp. 2942–2950). Curran.

Semedo, J. D., Zandvakili, A., Machens, C. K., Yu, B. M., & Kohn, A. (2019). Cortical areas interact through a communication subspace. *Neuron*, 102(1), 249–259. 10.1016/j.neuron.2019.01.026

Silverman, B. W. (1982). Kernel density estimation using the fast Fourier transform. *Journal of the Royal Statistical Society Series C: Applied Statistics*, 31(1), 93–99.

Smith, M. A., & Kohn, A. (2008). Spatial and temporal scales of neuronal correlation in primary visual cortex. *Journal of Neuroscience*, *28*(48), 12591–12603. 10.1523/JNEUROSCI.2929-08.2008

Snelson, E., & Ghahramani, Z. (2005). Sparse gaussian processes using pseudo- inputs. In Y. Weiss, B. Schölkopf, & J. Platt (Eds.), *Advances in neural information processing systems*, *18* (pp. 1257–1264). Curran.

Solin, A., Hensman, J., & Turner, R. E. (2018). Infinite-horizon gaussian processes. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, & R. Garnett, (Eds.), *Advances in neural information processing systems*, *31* (pp. 3486–3495). Curran.

Solin, A., & Särkkä, S. (2014). Explicit link between periodic covariance functions and state space models. In Y. Weiss, B. Schölkopf, & J. Platt (Eds.), *Advances in neural information processing systems*, *18* (pp. 904–912). Curran.

Steinmetz, N. A., Aydin, C., Lebedeva, A., Okun, M., Pachitariu, M., Bauza, M., . . . Harris, T. D. (2021). Neuropixels 2.0: A miniaturized high-density probe for stable, long-term brain recordings. *Science*, *372*(6539), eabf4588. 10.1126/science.abf4588

Steinmetz, N. A., Zatka-Haas, P., Carandini, M., & Harris, K. D. (2019). Distributed coding of choice, action and engagement across the mouse brain. *Nature*, *576*(7786), 266–273. 10.1038/s41586-019-1787-x

Sykulski, A. M., Olhede, S. C., Guillaumin, A. P., Lilly, J. M., & Early, J. J. (2019). The debiased Whittle likelihood. *Biometrika*, *106*(2), 251–266. 10.1093/biomet/asy071

Titsias, M. (2009). Variational learning of inducing variables in sparse gaussian processes. *Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics* (pp. 567–574).

Turner, R. E. (2010). *Statistical models for natural sounds*. PhD diss., University College London.

Ulrich, K. R., Carlson, D. E., Dzirasa, K., & Carin, L. (2015). GP kernels for cross-spectrum analysis. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, & R. Garnett (Eds.), *Advances in neural information processing systems*, *28*, 1999–2007.

Whittle, P. (1951). *Hypothesis testing in time series analysis*. Almqvist & Wiksells.

Williamson, R. C., Cowley, B. R., Litwin-Kumar, A., Doiron, B., Kohn, A., Smith, M. A., & Yu, B. M. (2016). Scaling properties of dimensionality reduction for neural populations and network models. *PLOS Computational Biology*, *12*(12), e1005141. 10.1371/journal.pcbi.1005141

Wilson, A., & Adams, R. (2013). Gaussian process kernels for pattern discovery and extrapolation. In *Proceedings of the 30th International Conference on Machine Learning* (pp. 1067–1075).

Wirtinger, W. (1927). Zur formalen Theorie der Funktionen von mehr komplexen Veränderlichen. *Mathematische Annalen*, *97*, 357–376. 10.1007/BF01447872

Wu, A., Roy, N. A., Keeley, S., & Pillow, J. W. (2017). Gaussian process–based nonlinear latent structure discovery in multivariate spike train data. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, & R. Garnett (Eds.), *Advances in neural information processing systems*, *30* (pp. 3496–3505). Curran.

Yang, W., & Yuste, R. (2017). In vivo imaging of neural activity. *Nature Methods*, *14*(4), 349–359. 10.1038/nmeth.4230

Yu, B. M., Cunningham, J. P., Santhanam, G., Ryu, S. I., Shenoy, K. V., & Sahani, M. (2009). Gaussian-process factor analysis for low-dimensional single-trial analysis of neural population activity. *Journal of Neurophysiology*, *102*(1), 614–635. 10.1152/jn.90941.2008

Zandvakili, A., & Kohn, A. (2015). Coordinated neuronal activity enhances cortico-cortical communication. *Neuron*, *87*(4), 827–839. 10.1016/j.neuron.2015.07.026

Zhang, Y., Leithead, W., & Leith, D. (2005). Time-series gaussian process regression based on Toeplitz computation of O(N2) operations and O(N)-level storage. In *Proceedings of the 44th IEEE Conference on Decision and Control* (pp. 3711–3716).

Zhao, Y., & Park, I. M. (2017). Variational latent gaussian process for recovering single-trial dynamics from population spike trains. *Neural Computation*, *29*(5), 1293–1316. 10.1162/NECO_a_00953