SCALABLE BAYESIAN TIME WARPING AT SINGLE-NEURON RESOLUTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Neuroscience analyses often require trial-averaging to determine reliable neural signals. However, trial-to-trial heterogeneity in the timing of neural dynamics often limits the efficacy of trial-averaging. In many cases, this heterogeneity may be linked to latent events, such as decision time, which lack obvious behavioral correlates. Here, we formulate a scalable Bayesian approach to time warping to resolve this challenge. We leverage newly developed methods for efficient Gaussian process regression, introduce a prior over monotonic piecewise-linear functions, and develop a parallel, blocked Hamiltonian Monte Carlo-within-Gibbs sampling approach to learn a full posterior distribution over model parameters. Moreover, we show this approach is capable of identifying warping function posteriors for *individual neurons*. On synthetic data, this capability can be used to identify latent subpopulation structure invisible to existing methods. On data from rat orbitofrontal cortex during a temporal wagering task, population-level fits favor more complex warps than most individual neurons require, providing evidence for heterogeneous single-neuron alignment surrounding a latent decision. Additionally, fitting to individual neurons sharpens firing patterns obscured by a shared population warp. In all, our results illustrate that Bayesian time warping is a practical tool for neural data analysis that not only mitigates trial-to-trial heterogeneity, but can leverage it to reveal previously hidden structure in single-neuron dynamics.

1 INTRODUCTION

A standard paradigm in neuroscience is to average neural activity across repeated, nominally identical trials to identify reliable signals from noisy single-trial activity. However, these experiments often exhibit substantial trial-to-trial heterogeneity in the timing of neural activity due to alignment of these dynamics to latent events outside of direct experimental control (Willett et al., 2021; Starkweather et al., 2017; Aronov et al., 2017; Banerjee et al., 2024). This heterogeneity complicates the use of standard analysis approaches based on trial-averaging (Figure 1A). When studying neural systems close to the sensorimotor periphery, there are often reliable behavioral correlates to align the observed activity, such as sniff cycle for olfaction or reach onset for motor control (Shusterman et al., 2011; Churchland et al., 2012). Neural systems involved in abstract concepts or higher-order cognition typically lack these reliable correlates, forcing experimenters to rely on noisy-at-best behavioral readouts for alignment (Buzsáki & Tingley, 2018; Stine et al., 2023).

To address this problem, Williams et al. (2020) developed an unsupervised time warping approach to infer interpretable trial alignments and firing patterns solely from the observed neural activity. However, standard time warping does not report a measure of model uncertainty, which results in several limitations for modern neuroscience experiments. First, all downstream analyses must assume alignment certainty for each trial, which may be invalid in experiments with high behavioral variability. Second, the model is unable to report the reliability of alignments inferred by single units, resulting in the strong simplifying assumption that all recorded neurons in a population are aligned the same way on a given trial. Recent experimental evidence has shown that several regions exhibit heterogeneous structure with different neurons aligned to distinct task events, suggesting this simplifying assumption may be misspecified in many practical contexts (Dombeck et al., 2009; Hirokawa et al., 2019; Hocker et al., 2021; Posani et al., 2026).

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Here, we propose a scalable Bayesian approach to piecewise-linear time warping which infers a posterior distribution over possible trial alignments and firing patterns consistent with the observed neural data. This work builds on several prior methods discussed in section 2.3. Crucially, quantifying uncertainty via the posterior distribution makes reliable inference at the level of noisy single units feasible, as individual units often provide far less evidence for a single warping function than the full population. We showcase this capability on synthetic data with latent subpopulation structure and electrophysiological data from rat orbitofrontal cortex during a temporal wagering task.

Our contributions are threefold. First, we develop a fully probabilistic time warping model built on a novel prior over monotonic piecewise-linear warping functions and an extension of efficient Gaussian process regression techniques. Second, we implement an inference scheme which exploits conditional independence to parallelize sampling of multimodal distributions across neurons and trials on modern accelerator hardware. Finally, we demonstrate that single-neuron warping posteriors can reveal otherwise-latent heterogeneity in neural population dynamics.

2 METHOD

2.1 GENERATIVE MODEL

Let $y_{nk}(t)$ denote the neural activity observed at time t for neuron n on trial k. Without loss of generality, let $t \in [0, 1]$ denote the fraction of elapsed "clock" time through the trial. For spikes binned into T time bins for each trial, we denote time as a vector $\boldsymbol{t}$ such that index $t_i = \frac{i}{T}$, and $y_{nk}(t_i) \in \mathbb{Z}_{\geq 0}$ represents the spike count in bin i for neuron n on trial k. We assume that the observed neural activity can be decomposed into a **neuron-specific template function** $\tilde{\lambda}_n(\cdot) : [0, 1] \to \mathbb{R}_{\geq 0}$ which is corrupted by a **trial-specific warping function** $w_k(\cdot) : [0, 1] \to [0, 1]$ such that

$$\begin{aligned} \lambda_{nk}(t) &= \tilde{\lambda}_n(w_k(t)) \\ y_{nk}(t_i) &\sim \text{Poisson}(\lambda_{nk}(t_i)) \end{aligned} \tag{1}$$

In other words, the warping function maps clock time t on trial k to a standardized "template" time $\tau := w_k(t)$ which is then read out by the neuron's template function $\lambda_{nk}(t) = \tilde{\lambda}_n(\tau)$ (Figure 1B). To generalize this framework to the multi-trial setting, we assume that a neuron's template function is conserved across all trials. Likewise, to generalize this framework to the multi-neuron setting, we start by assuming that a trial's warping function is shared across all neurons (Figure 1C). We later relax this assumption by fitting the model on a single-neuron basis in Sections 3.2 and 3.3.

Our goal is to learn the posterior distribution over the template functions for all neurons and warping functions for all trials given the observed data:

$$p\left(\left\{\tilde{\lambda}_n\right\}_{n=1}^{N}, \{w_k\}_{k=1}^{K} \mid \mathbf{Y}\right) \propto p\left(\left\{\tilde{\lambda}_n\right\}_{n=1}^{N}\right) p\left(\{w_k\}_{k=1}^{K}\right) \prod_{n=1}^{N} \prod_{k=1}^{K} \prod_{i=1}^{T} p\left(y_{nk}(t_i) \mid \tilde{\lambda}_n, w_k\right) \tag{2}$$

where $\mathbf{Y} \in \mathbb{Z}_{\geq 0}^{T \times N \times K}$ denotes the spike count tensor across timebins, neurons, and trials. We specify priors over the template functions and warping functions in the following sections. Our approach here focuses on the modeling of discrete spike counts with Poisson noise, but we note that the time warping framework is general and can be adapted to any desired noise model.

2.1.1 TEMPLATE PRIOR

We assume that the observed data contains a dynamical motif of interest which is embedded within a longer time series on each trial. We model this by parameterizing the template functions as **constant-padded Gaussian processes** (Figure 1C). More precisely, we define two padding points $0 < \tilde{b}_{\text{initial}} < \tilde{b}_{\text{end}} < 1$ which define the duration of the dynamical motif of interest in template time. We treat these padding points as user-specified constants shared across all neurons. For $\tilde{b}_{\text{initial}} \leq \tau \leq \tilde{b}_{\text{end}}$, we define a Gaussian process (GP) with a squared exponential kernel function:

$$f_n \sim \mathcal{GP}\left(\mathbf{0}, k_n(\cdot, \cdot)\right) \tag{3}$$

$$k_n(\tau_1, \tau_2) = \sigma_n^2 \exp\left[-\frac{\|\tau_1 - \tau_2\|^2}{2\ell_n^2}\right] \tag{4}$$

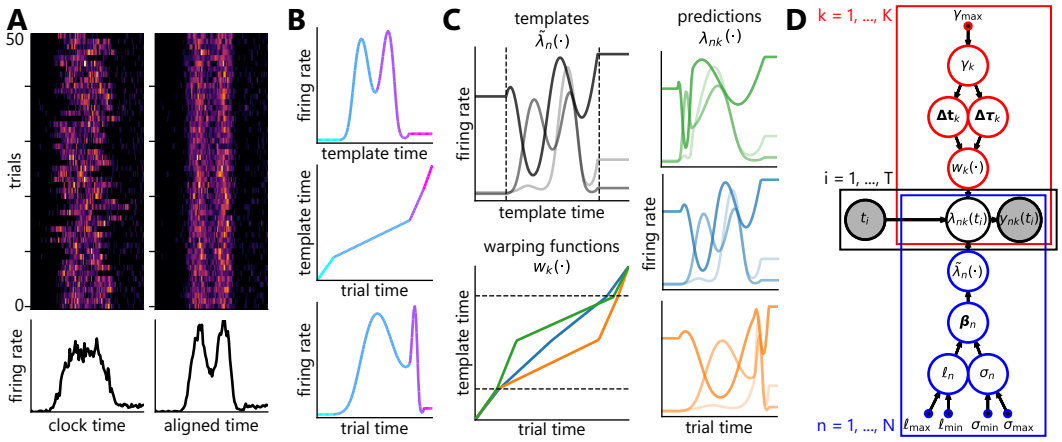

Figure 1: Model setup. A) Top: Activity across trials before (left) and after (right) temporal alignment. Bottom: Corresponding trial-averages. B) A template function (top) is corrupted by a warping function (middle) to generate the observed activity on a trial (bottom). Shared colors indicate corresponding regions. C) The templates are specific to each neuron (shading) and the warping functions are specific to each trial (color) to give rise to model predictions. Dashed lines indicate padding regions. D) Graphical model of Bayesian time warping.

with neuron-specific kernel hyperparameters σ_n and ℓ_n defining the output scale and length scale of the kernel, respectively (Rasmussen & Williams, 2008). The template function is then defined as:

$$\tilde{f}_n(\tau) = \text{softplus}(f_n(\tau)) \tag{5}$$

$$\tilde{\lambda}_n(\tau) = \begin{cases} \tilde{f}_n(\tau) & \text{if } \tilde{b}_{\text{initial}} \leq \tau \leq \tilde{b}_{\text{end}}, \\ \tilde{f}_n(\tilde{b}_{\text{initial}}) & \text{if } \tau < \tilde{b}_{\text{initial}}, \\ \tilde{f}_n(\tilde{b}_{\text{end}}) & \text{if } \tau > \tilde{b}_{\text{end}} \end{cases} \tag{6}$$

Crucially, this defines a prior distribution over *smooth* template functions that are constrained to only contain interesting dynamics between the padding points and *cannot* model any dynamics occurring in the padded regions.

To improve the scalability of GP inference, we leverage Equispaced Fourier Gaussian Processes (EFGP) to obtain an efficient and accurate weight-space approximation of the squared exponential kernel in Equation 4 (Greengard et al., 2025). A review of EFGP and related GP approximation methods is included in appendix A.1 for reference. Initial work on EFGP defined a weight-space approximation for a *fixed* squared exponential kernel. In order to learn posteriors over kernel hyperparameters for each neuron, we **extend EFGP to define a shared set of basis functions for a *family* of squared exponential kernels** with varying hyperparameters. To do so, we constrain the length scales such that $\ell_n \in [\ell_{\min}, \ell_{\max}]$ and define a set of basis functions that is guaranteed to approximate all kernels in this range with the user-specified tolerance or better. An informal sketch of this guarantee can be found in appendix A.1. We also constrain the output scales such that $\sigma_n \in [\sigma_{\min}, \sigma_{\max}]$ for consistency. This avoids the need for recomputation of the basis set with each kernel hyperparameter update and provides a fixed-size array for improved compatibility with accelerator hardware. Therefore, inference of the template function for neuron n involves learning ℓ_n, σ_n, and the basis function weights $\boldsymbol{\beta}_n$ (Figure 1D). Standard transformations are used to sample these parameters in an unconstrained space to improve the efficiency of Markov chain Monte Carlo (MCMC) inference. A full specification of the template prior can be found in appendix A.2.

2.1.2 Warping Prior

To respect causality in the template and trial, the warping functions must be monotonically increasing. Following Williams et al. (2020), we additionally want the warping functions to be interpretable for each trial. We introduce a **distribution over monotonically increasing piecewise-linear functions**, ranging from simple shifts to complicated nonlinear warps with many segments (Figure 1C).

The complexity of the warping functions is determined by the number of knots J, where knot j for trial k is defined as the tuple $(t_k^{(j)}, \tau_k^{(j)})$. The warping function is then defined as the linear interpolation between knots:

$$w_k(t) = \tau_k^{(j)} + \frac{\tau_k^{(j+1)} - \tau_k^{(j)}}{t_k^{(j+1)} - t_k^{(j)}}(t - t_k^{(j)}) \quad \text{if } t \in [t_k^{(j)}, t_k^{(j+1)}] \tag{7}$$

for $j = 1, ..., J-1$. All warping functions are constrained such that $(t_k^{(1)}, \tau_k^{(1)}) = (0, 0)$ and $(t_k^{(J)}, \tau_k^{(J)}) = (1, 1)$. This ensures the warping function $w_k(\cdot)$ can be used to *align* the observed data from trial time into template time without introducing edge artifacts and enables the use of standard analyses based on trial-averaging (Figure 1A) (Williams et al., 2020). We additionally constrain all warping functions to reflect the padding assumptions made on the template in Section 2.1.1 by locking $\tau_k^{(2)} = \tilde{b}_{\text{initial}}$ and $\tau_k^{(J-1)} = \tilde{b}_{\text{end}}$ for all trials. These constraints specify that $J \geq 4$ for all warping functions.

To more easily specify a prior over monotonically increasing functions, we switch to reasoning about the *intervals* defined by the J knots. Note that J knots specify $J-1$ intervals in the t- and τ-axes. We collect these intervals into two vectors: $\boldsymbol{\Delta t}_k, \boldsymbol{\Delta \tau}_k \in \mathbb{R}^{J-1}$. To satisfy the monotonicity constraint, we constrain all entries of these vectors to be positive which guarantees that all segments of $w_k(\cdot)$ will have slope > 0. To satisfy the endpoint constraint that $(t_k^{(J)}, \tau_k^{(J)}) = (1, 1)$, both vectors must have unit-sum. Prior approaches sample $\boldsymbol{\Delta t}_k, \boldsymbol{\Delta \tau}_k$ directly from the $J-1$ simplex via a Dirichlet distribution, which satisfies both constraints (Cheng et al., 2016). However, a Dirichlet prior can create difficult-to-sample posterior geometry due to its inability to be represented via a non-centered parameterization (Appendix A.3).

To improve the posterior geometry, we can instead generate $\boldsymbol{\Delta t}_k, \boldsymbol{\Delta \tau}_k$ by sampling unconstrained vectors $\boldsymbol{dt}_k, \boldsymbol{d\tau}_k$ and passing each through a softmax. The unconstrained vectors are drawn from a multivariate Gaussian distribution with an isotropic covariance matrix scaled by a trial-specific warp scale parameter γ_k (Aitchison & Shen, 1980). Intuitively, as γ_k gets larger, the unconstrained variables will have higher variance which will result in more lopsided intervals after the softmax and more extreme warping functions. Conversely, as $\gamma_k \to 0$, the entries of the unconstrained vectors will be roughly equal, resulting in similarly-sized intervals after the softmax and warping functions closer to the identity. Therefore, inference of the warping function for trial k involves learning $\gamma_k, \boldsymbol{\Delta t}_k, \boldsymbol{\Delta \tau}_k$ (Figure 1D). A schematic can be found in figure S1 and further details can be found in appendix A.3.

For $J = 4$, all 3 elements of $\boldsymbol{\Delta \tau}_k$ are fully specified by the padding and unit sum constraints. A *shift* warping function can be implemented by further constraining $\Delta t_k^{(2)} = \Delta \tau_k^{(2)}$, meaning the clock time duration of the dynamical motif of interest cannot be changed. A *linear* warping function can be implemented by lifting this constraint to allow the clock time duration of the dynamical motif to change. A *piecewise-linear* warping function can be implemented by increasing $J > 4$, where the number of effective knots is given by $J - 4$. The number of knots J is shared by all trials and can be treated as a user-specified hyperparameter or chosen in a data-driven manner as explored in Section 2.2.3.

2.2 Inference

2.2.1 Efficient Sampling via Conditional Independence

To draw approximate samples from the posterior of interest (equation 2), we implement a Markov chain Monte Carlo (MCMC) algorithm (Geyer, 2026). Although gradient-based MCMC samplers, such as Hamiltonian Monte Carlo (HMC), are state of the art, we found inference of the full joint distribution in the large N and/or K regime typical of modern neuroscience datasets prohibitively memory- and time-intensive (Neal, 2026; Betancourt, 2018; Carpenter et al., 2017). Our key insight is that *conditioned on the warping functions*, the template functions for all neurons are independent. Similarly, *conditioned on the template functions*, the warping functions for all trials are independent. This enables a **parallel, blocked HMC-within-Gibbs** sampling approach to make MCMC tractable. We use HMC to sample each neuron's template function *in parallel* conditioned on the current estimate of the warping functions, sample each trial's warping function *in parallel* conditioned on

the samples of the template functions, and repeat. We implement our sampling scheme in `jax` which enables trivial usage of accelerator hardware, such as GPUs or TPUs, to scale this parallelization to the large N and/or K setting common in modern neuroscience (Bradbury et al., 2018). Runtime scaling behavior for N or K up to 500 on synthetic data shows less than a minute to fit the model on modern accelerator hardware (Figure S2). A detailed implementation of this inference scheme can be found in appendix A.4 and algorithm 1. Additionally, our use of MCMC for inference allows us to track standard diagnostic machinery (such as $\hat{R}$ and effective sample size) and raise flags for experimenters when there are signs of problematic inference, such as non-convergence between MCMC chains (Vehtari et al., 2021).

2.2.2 Handling Multimodality via Parallel Tempering

As noted by Williams et al. (2020), the warping likelihood is "complex and full of local minima" (see figure S3 for an example on real data). We therefore use **parallel tempering**, running many samplers in parallel, each at a different rung on a pre-specified temperature ladder (Geyer, 1991; Sambridge, 2014). Information is shared between temperature levels by proposing swaps between the parameter states on adjacent rungs. As these swaps propagate across rungs, samples from well-separated regions of parameter space initially drawn from the highest temperature rungs can be passed down to the base posterior sampler. Using the same conditional independence as in section 2.2.1, we maintain separate ladders for each neuron's template and each trial's warp. This keeps every swap low-dimensional and avoids the low acceptance rates of coordinated swaps in the joint distribution. Further details for the tempering procedure can be found in appendix A.4.

2.2.3 Model Selection via Pareto Smoothed Importance Sampling

The key hyperparameter is warp complexity. When the source of temporal variability is known *a priori*, the complexity can be set directly. For example, trial-to-trial jitter in sniff onsets can be modeled as shifts for aligning olfactory experiments (Shusterman et al., 2011). However, the cause of temporal heterogeneity is not always known *a priori*, especially in naturalistic behaviors or novel/complex experiments. In these cases, data-driven selection of the warping complexity can be beneficial. Unfortunately, naive deployment of V-fold cross-validation can be problematic for Bayesian models because posteriors fit to smaller training sets are intentionally more dominated by the prior, while exact leave-one-out (LOO) requires prohibitively many MCMC refits. We instead use **Pareto smoothed importance sampling** (PSIS) to estimate LOO predictive performance on held-out (neuron, trial) pairs from a single fit per complexity (Vehtari et al., 2017; 2024). We review this procedure in appendix A.5 for reference. This requires only one model fit per warping complexity, making data-driven hyperparameter tuning feasible. Critically, PSIS-LOO reports a measure of the difference in models' expected predictive performance on unseen data (relative elpd difference), *as well as their spread*. We select the simplest complexity whose expected log predictive density is within 2 SE of the best model's.

2.3 Related Work

Affine Warping. We use the method of Williams et al. (2020) as a deterministic baseline. This approach uses maximum likelihood to determine template functions and piecewise-linear warping functions. Therefore, it does not provide any estimates of uncertainty and is required to pool information across neurons for reliable estimation of the warping functions.

Time-warped svGPFA. Duncker & Sahani (2018) also use the Bayesian machinery of GPs to model neural firing rates along with trial-specific time warping. However, they use a GP prior on the warping functions which makes it difficult to guarantee monotonic warping functions. Additionally, their use of mean-field Gaussian variational approximations is fundamentally unable to represent complex, multimodal posteriors over warping functions often observed in real data (Figure S3). Finally, their model cannot identify single-neuron warping posteriors because it warps latent factors shared between all neurons on a trial; single-neuron warping posteriors are critical for our results in sections 3.2 and 3.3.

Bayesian Curve Registration. Cheng et al. (2016) is most similar in spirit to our approach, utilizing a GP prior on the functions undergoing temporal warping. Unlike our overparameterized, unconstrained prior on warping functions, they use a Dirichlet process prior on the intervals between knots

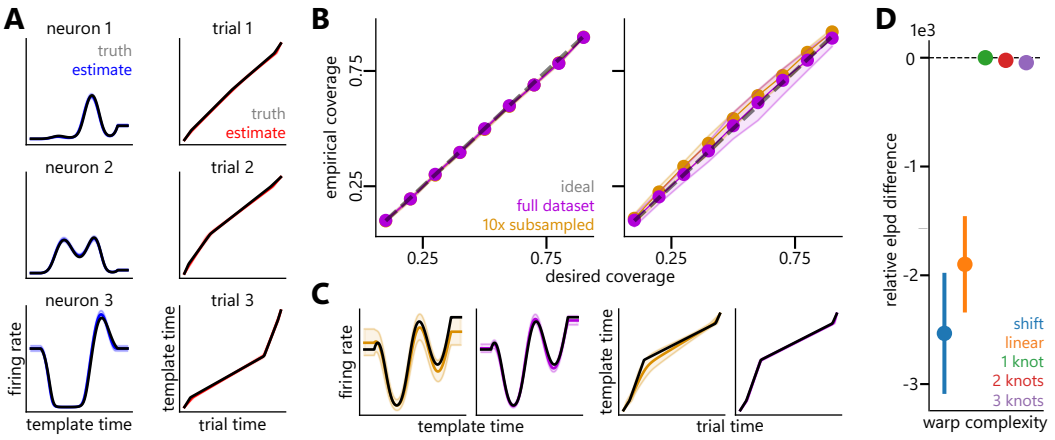

Figure 2: Synthetic validation. A) Recovery of ground truth template functions for example neurons (left) and warping functions for example trials (right). Solid line is posterior mean, shaded region is central 95% credible interval. B) Coverage sweep for central credible intervals of sampled template parameters (left) and warping parameters (right). Shading indicates 2×SE over 20 synthetic datasets. C) Effect of subsampling on an example template (left) and warping function (right). Color legend same as B. D) PSIS-LOO model selection recovers 1 knot as the ground truth warping complexity. Points are mean $\pm 2 \times$ SE over all (neuron, trial) tuples.

in piecewise linear warping functions (in the finite case, this reduces to a Dirichlet prior). They take the number of knots on the order of the number of timebins to model arbitrarily nonlinear warping functions. However, as noted in section 2.1.2 and in appendix A.3, the Dirichlet parameterization can lead to unfavorable posterior geometry for sampling. Instead, we implement an overparameterized logit-normal prior (Aitchison & Shen, 1980).

3 Results

3.1 Synthetic Validation

To validate our approach, we simulate synthetic data from the generative model with $N = 20$, $K = 40$, and piecewise-linear warps with 1 knot. Simulation details can be found in appendix B.1 and table 1. We assess the model's ability to recover the ground truth parameters at three levels: estimation, uncertainty, and model selection.

First, we show that the inferred posterior means of the template functions and warping functions are qualitatively accurate (Figure 2A). However, the posterior mean can deviate from ground truth for a variety of reasons not indicative of model failure. Therefore, we choose *uncertainty* to quantify the correctness of model inference on synthetic data. For a well-specified Bayesian model, we expect the posterior credible intervals to behave like frequentist confidence intervals with appropriate coverage guarantees (Gelman et al., 2020; Talts et al., 2020). For each of 20 simulated datasets and each inferred parameter, we compute whether the true value falls within the central credible interval containing x fraction of the posterior density. We sweep the desired coverage x from 0 to 1 and report the empirical coverage averaged over template parameters and warping parameters separately, along with its spread across datasets. When fit to the full dataset, the templates and warps exhibit ideal average coverage behavior (Figure 2B). As an additional validation for our model, we refit the model using 4 of 40 trials to assess template coverage, and separately using 2 of 20 neurons to assess warp coverage. This tests coverage as the evidence per parameter shrinks by 10x. Even in this subsampled regime, we observe near-optimal average coverage behavior of our inference procedure (Figure 2B). To verify that this coverage does not come at the cost of completely uninformative estimates, we compare posterior estimates of the template functions and warping functions under subsampling to those found by fitting the whole dataset and observe that the posterior degrades gracefully when fit to subsampled data (Figure 2C). These observations suggest that single-neuron posterior estimates

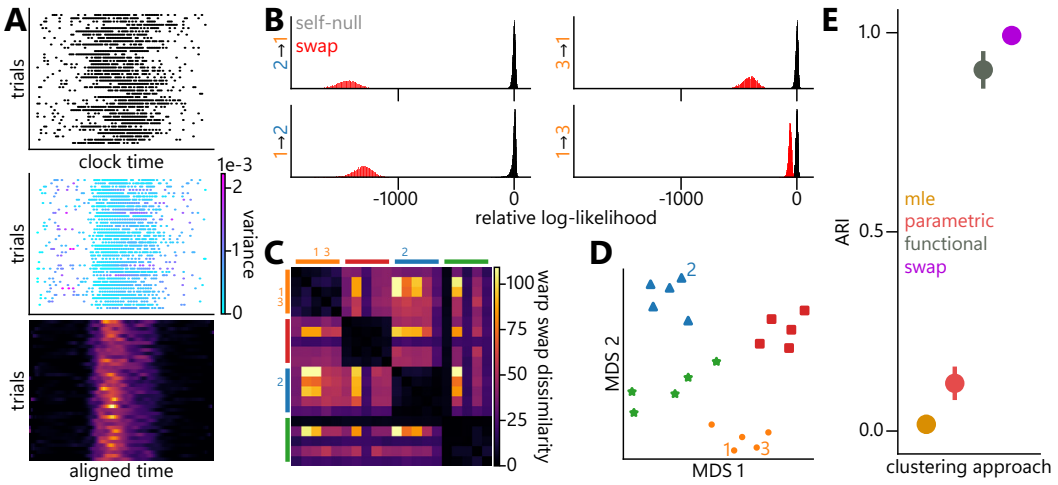

Figure 3: Subpopulation identification in synthetic data. A) Example raw data (top) along with aligned raster colored by posterior variance in aligned spike times (middle) and posterior-smoothed binned spike counts (bottom) from model fit to this single neuron. B) Likelihood effect of swapping warps between two neurons in different clusters (left) and the same cluster (right). C) Warp swap dissimilarity matrix between all neurons. Details for warp swap dissimilarity in text. Colored bars indicate ground truth clusters. D) Non-metric MDS embedding implied by C. Shapes indicate ground truth clusters, colors indicate clusters identified by hierarchical clustering. Numbered neurons are examples used in B. E) Adjusted Rand index (ARI) between ground truth clusters and identified clusters for 4 different clustering approaches. Points are mean $\pm 2 \times$ SE over 20 synthetic datasets.

for the warping function may be informative, which motivates sections 3.2 and 3.3. Finally, we validate that the PSIS-LOO procedure for model selection from Section 2.2.3 recovers the ground truth warping complexity used to generate the data. On the synthetic data used in panel A, the procedure identified the 1 knot model as the best-performing complexity, and correctly assigned lower expected predictive densities to the simpler shift and linear models (Figure 2D). Although the 2 knot and 3 knot models had similar performance to the 1 knot model, our heuristic would choose 1 knot as the most parsimonious explanation for the data.

3.2 Subpopulation Identification

Building on the subsampling results from section 3.1, we explore the utility of fitting the model to *single neurons* independently. We simulate 4 subpopulations of 5 neurons each from the generative model. Simulation details can be found in appendix B.2 and table 1. We fit the model to each neuron independently to determine the single-neuron optimal warping functions. An example alignment for a single neuron is shown in Figure 3A. We highlight two approaches to propagate model uncertainty into downstream analyses. First, we can utilize all samples from the posterior to align the observed spike times and report the mean aligned spike time as well as its associated variance as a raster. Second, we can align the binned spike counts directly, and averaging this binned alignment over posterior samples corresponds to *smoothing* the observed spike counts by the posterior uncertainty. These smoothed spike counts can be used for downstream modeling analyses, similar to standard spike smoothing.

It can be difficult to naively compare the posterior mean of the warping functions inferred by individual neurons because different neurons will provide varying amounts of information to constrain the warping functions relative to the prior. To robustly compare the warping functions of two neurons, we define a **warp swap dissimilarity** (Figure 3B). Intuitively, if neuron m and neuron n have different ground truth warping functions, then using samples from the warping function posterior of neuron m on the template inferred by neuron n should decrease the likelihood relative to using samples from the warping function posterior of neuron n (and vice versa). However, if neuron m and

neuron n share the same ground truth warping function, the warp swap should lead to a relatively small decrease in the likelihood. We formalize this intuition in A.6 to define a positive, symmetric dissimilarity score. We can construct a dissimilarity matrix and low-dimensional embedding using this approach, as well as use the dissimilarity matrix for hierarchical clustering (Figure 3C-D) (Kruskal, 1964).

For 20 simulated datasets with the subpopulation structure outlined above, we compare the adjusted Rand index (ARI) between the ground truth and the clusters identified by four approaches (Figure 3E) (Hubert & Arabie, 1985). First, we fit maximum likelihood-based affine warping to each neuron and cluster the neurons together based on the distance between their fitted warping functions (Williams et al., 2020). Second, we fit Bayesian time warping to each neuron and cluster the neurons together based on the distance between the posterior means of their sampled warping parameters. Third, we cluster the neurons together based on the distance between the posterior means of their fitted warping functions. Finally, we cluster neurons together based on the warp swap dissimilarity. We find that deterministic warping is entirely unable to reliably identify the ground truth clustering (mean: 0.0019, min: -0.0106, max: 0.0842). Examples highlighting the per-neuron failure of deterministic warping are shown in figure S4. However, using the sampled parameters of the Bayesian model is also prone to misclustering (mean: 0.1224, min: -0.0144, max: 0.3019). In contrast, clustering the induced warping functions between neurons leads to a dramatic improvement in clustering (mean: 0.9067, min: 0.7072, max: 1.0). Further improvements in clustering accuracy are achieved by the warp swap dissimilarity (mean: 0.9929, min: 0.8588, max: 1.0) which significantly outperforms all other methods (Wilcoxon signed-rank test p-values - swap vs. MLE: 6.84×10^{-5}; vs. parametric: 8.83×10^{-5}; vs. functional: 0.0057).

3.3 Rat Decision-Making

We apply the concept of heterogeneous single-neuron warping to electrophysiological data from rat orbitofrontal cortex (OFC) during a temporal wagering task (Schiereck et al., 2026). The task, in short, involves waiting for an offered reward amount until the reward is delivered or opting out early to initiate a new trial (Figure 4A). Critically, the animal's decision to opt out is an unobserved latent event and can only be behaviorally registered once the rat pokes into an opt-out port. Here, we focus on the 2 seconds preceding and the 0.5 second following the measured opt-out time. Further details of the dataset and sampling procedure are available in appendix B.3 and table 1.

We first fit the model to the full neural population and perform model selection as described in section 2.2.3 (Figure 4B). This procedure selected a warp complexity of 3 knots as optimal. Next, we fit the model to each neuron individually and perform model selection at the single-neuron level (Figure 4C). We identified heterogeneous warping complexities preferred by individual neurons, with the majority (39/48 neurons) of neurons preferring shift warping. There were also neurons which preferred linear (4/48 neurons), 1 knot (2/48 neurons), 2 knots (2/48 neurons), and 3 knots (1/48 neurons). To determine if this is an artifact of applying our model selection procedure at the single-neuron level, we performed two synthetic controls and verified that PSIS-LOO is able to identify the ground truth single-neuron warping complexities with $\geq 65\%$ accuracy when a 1-knot warp is shared between all neurons and when there are subpopulations of neurons each with different 1-knot warps (Figure S5). Critically, the heterogeneous control exhibits a similar mismatch to the real data, with the population warp preferring a higher warp complexity to model the data than was used for each subpopulation's warping function.

We then compared the alignment identified by the optimal population warping complexity (3 knots) to those fit to single-neurons at their optimal complexity (Figure 4D). This reveals interesting post-alignment structure that may be smeared in the population fit (such as for neurons 1 and 3), or completely hidden by the population fit (such as the 4th peak for neurons 4 and 5). To quantify this effect further, we compute the trial-split Poisson log-likelihood of the raw data, the population fit with 3-knots, and the single-neuron optimal fits across all neurons (Appendix B.3 and Figure S6). We find that the single-neuron fits for neurons with optimal warps of 1, 2, or 3 knots outperform the population fit with 3 knots (Wilcoxon signed-rank test p-value 0.0244). Combined with the heterogeneity in optimal warping complexities biased towards simpler warping functions than those used by the full population fit, these results suggest that OFC neurons are aligned to heterogeneous signals surrounding the labeled opt-out time. Interestingly, we did not find convincing evidence of clustering in this dataset despite our ability to reliably identify clusters in synthetic data with the

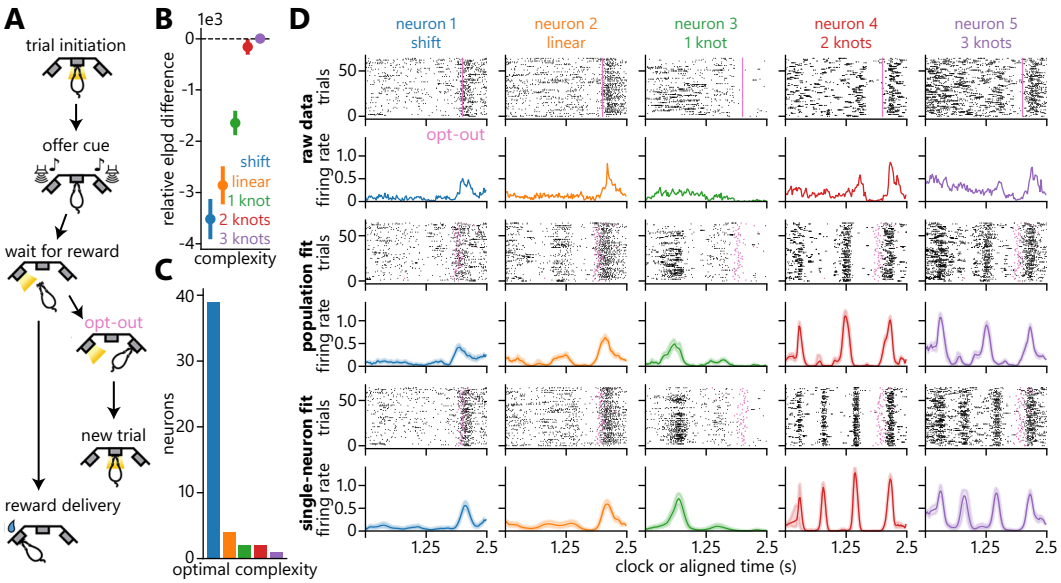

Figure 4: Application to rat OFC data. A) Task schematic, see text for details. B) PSIS-LOO model selection when fit to the entire population. C) Histogram of optimal warping complexities when PSIS-LOO model selection is done for individual neurons. Color legend same as B. D) Comparison between raw data raster and trial average (rows 1-2), population aligned raster and trial average (rows 3-4), and single-neuron aligned raster and trial average (rows 5-6) across neurons with varying optimal single-neuron warping complexities (columns). Aligned trial averages show posterior mean in solid line and central 90% credible interval in shading.

swap test. Future work will investigate possible behavioral correlates contributing to these neural responses.

4 Discussion

We introduced a scalable Bayesian formulation of time warping for neural data. The practical payoff of posterior uncertainty is that inference of single-unit warping function posteriors becomes meaningful. This provides a new way to group neurons via similarity in their trial-to-trial alignments, which we validate in synthetic data with subpopulation structure. Crucially, the clustering induced by the warp swap dissimilarity is driven solely by similarity in *timing*, rather than similarity in overall response profile (amplitude and timing) used by typical neural data analysis approaches (Williams et al., 2018; Yu et al., 2009). On electrophysiological data from rat OFC, we identified a mismatch in complexity between the population-level and single-neuron fits. We suggest that this mismatch in complexity may occur because the **shared alignment assumption common to all current neuroscientific time warping models is misspecified**. Consistent with this, single-neuron fits recover structure smeared or invisible in the population fit and improve the trial-average's fit to individual trials. These heterogeneous single-neuron alignments are a novel object for future analysis, as the possible behavioral correlates underlying these activity patterns remain unknown.

There are several limitations of our approach. First, we assume the template is conserved across all trials, which cannot model trial-to-trial variability in the response profile (Williams et al., 2020). While a multiplicative per-trial gain term would be relatively straightforward to add, more complicated modulations of the template may be out of scope of this model class. Second, the number of knots must be shared across all trials. This may be misspecified in experiments with very high degrees of behavioral variability and ill-defined trial structure, such as naturalistic behavior.

Taken together, our results support that trial-to-trial temporal heterogeneity is an underexplored signal of neural population dynamics, rather than solely noise to be removed before analysis.

AI USE STATEMENT

In this work, we used generative AI tools to provide feedback on research methodology and to support data analysis. We have not used generative AI tools for generating synthetic datasets, developing the conceptual framework, formulating mathematical claims, proposing hypotheses, implementation of methods, or interpreting results. Proving mathematical claims, assisting with translation, and reformatting datasets are not applicable to this work. Additionally, we used generative AI tools to edit the text for clarity and summarize relevant existing literature. We have reviewed all AI-assisted work. All LLM-suggested code was reviewed by an author before being re-implemented by hand and a manual literature review was performed to supplement LLM-summarized relevant literature. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

An open-source implementation of the Bayesian time warping method will be made available on GitHub upon acceptance. We have included high-level implementation details in sections 2.1.1, 2.1.2, 2.2.1, 2.2.2, and 2.2.3. Additionally, more specific implementation details relevant for re-implementing the MCMC sampling procedure can be found in appendices A.1, A.2, A.3, A.4 and algorithm 1. Experimental details are contained in appendices B.1, B.2, B.3, and model parameters for each experiment are shown in table 1.

REFERENCES

J. Aitchison and S. M. Shen. Logistic-Normal Distributions: Some Properties and Uses. *Biometrika*, 67(2):261–272, 1980. ISSN 0006-3444. doi: 10.2307/2335470. URL https://www.jstor.org/stable/2335470.

Dmitriy Aronov, Rhino Nevers, and David W. Tank. Mapping of a non-spatial dimension by the hippocampal–entorhinal circuit. *Nature*, 543(7647):719–722, March 2017. ISSN 1476-4687. doi: 10.1038/nature21692. URL https://www.nature.com/articles/nature21692.

Arkarup Banerjee, Feng Chen, Shaul Druckmann, and Michael A. Long. Temporal scaling of motor cortical dynamics reveals hierarchical control of vocal production. *Nature Neuroscience*, 27(3): 527–535, March 2024. ISSN 1546-1726. doi: 10.1038/s41593-023-01556-5. URL https://www.nature.com/articles/s41593-023-01556-5.

Michael Betancourt. A Conceptual Introduction to Hamiltonian Monte Carlo, July 2018. URL http://arxiv.org/abs/1701.02434. arXiv:1701.02434 [stat.ME].

James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Yash Katariya, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. JAX: composable transformations of Python+NumPy programs, 2018. URL http://github.com/jax-ml/jax.

György Buzsáki and David Tingley. Space and Time: The Hippocampus as a Sequence Generator. *Trends in Cognitive Sciences*, 22(10):853–869, October 2018. ISSN 1879-307X. doi: 10.1016/j.tics.2018.07.006.

Bob Carpenter, Andrew Gelman, Matthew D. Hoffman, Daniel Lee, Ben Goodrich, Michael Betancourt, Marcus Brubaker, Jiqiang Guo, Peter Li, and Allen Riddell. Stan: A Probabilistic Programming Language. *Journal of Statistical Software*, 76:1–32, January 2017. ISSN 1548-7660. doi: 10.18637/jss.v076.i01. URL https://doi.org/10.18637/jss.v076.i01.

Wen Cheng, Ian L. Dryden, and Xianzheng Huang. Bayesian Registration of Functions and Curves. *Bayesian Analysis*, 11(2):447–475, June 2016. ISSN 1936-0975, 1931-6690. doi: 10.1214/15-BA957. URL https://projecteuclid.org/journals/bayesian-analysis/volume-11/issue-2/Bayesian-Registration-of-Functions-and-Curves/10.1214/15-BA957.full.

Mark M. Churchland, John P. Cunningham, Matthew T. Kaufman, Justin D. Foster, Paul Nuyujukian, Stephen I. Ryu, and Krishna V. Shenoy. Neural population dynamics during reaching. *Nature*, 487(7405):51–56, July 2012. ISSN 1476-4687. doi: 10.1038/nature11129. URL https://www.nature.com/articles/nature11129.

Daniel A. Dombeck, Michael S. Graziano, and David W. Tank. Functional Clustering of Neurons in Motor Cortex Determined by Cellular Resolution Imaging in Awake Behaving Mice. *Journal of Neuroscience*, 29(44):13751–13760, November 2009. ISSN 0270-6474, 1529-2401. doi: 10.1523/JNEUROSCI.2985-09.2009. URL https://www.jneurosci.org/content/29/44/13751.

Lea Duncker and Maneesh Sahani. Temporal alignment and latent Gaussian process factor inference in population spike trains. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/file/d1ff1ec86b62cd5f3903ff19c3a326b2-Paper.pdf.

Andrew Gelman, Aki Vehtari, Daniel Simpson, Charles C. Margossian, Bob Carpenter, Yuling Yao, Lauren Kennedy, Jonah Gabry, Paul-Christian Bürkner, and Martin Modrák. Bayesian Workflow, November 2020. URL http://arxiv.org/abs/2011.01808. arXiv:2011.01808 [stat.ME].

Charles J Geyer. Markov Chain Monte Carlo Maximum Likelihood. 1991.

Charles J. Geyer. Introduction to MCMC. In *Handbook of Markov Chain Monte Carlo*. Chapman and Hall/CRC, 2 edition, 2026. Num Pages: 46.

Philip Greengard, Manas Rachh, and Alex H. Barnett. Equispaced Fourier Representations for Efficient Gaussian Process Regression from a Billion Data Points. *SIAM/ASA Journal on Uncertainty Quantification*, 13(1):63–89, March 2025. ISSN 2166-2525. doi: 10.1137/23M1565310. URL https://epubs.siam.org/doi/10.1137/23M1565310.

Junya Hirokawa, Alexander Vaughan, Paul Masset, Torben Ott, and Adam Kepecs. Frontal cortex neuron types categorically encode single decision variables. *Nature*, 576(7787):446–451, December 2019. ISSN 1476-4687. doi: 10.1038/s41586-019-1816-9. URL https://www.nature.com/articles/s41586-019-1816-9.

David L Hocker, Carlos D Brody, Cristina Savin, and Christine M Constantinople. Subpopulations of neurons in lOFC encode previous and current rewards at time of choice. *eLife*, 10:e70129, October 2021. ISSN 2050-084X. doi: 10.7554/eLife.70129. URL https://doi.org/10.7554/eLife.70129.

Lawrence Hubert and Phipps Arabie. Comparing partitions. *Journal of Classification*, 2(1):193–218, December 1985. ISSN 1432-1343. doi: 10.1007/BF01908075. URL https://doi.org/10.1007/BF01908075.

J. B. Kruskal. Nonmetric multidimensional scaling: A numerical method. *Psychometrika*, 29(2):115–129, June 1964. ISSN 1860-0980. doi: 10.1007/BF02289694. URL https://doi.org/10.1007/BF02289694.

Radford M. Neal. MCMC Using Hamiltonian Dynamics. In *Handbook of Markov Chain Monte Carlo*. Chapman and Hall/CRC, 2 edition, 2026. Num Pages: 49.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Andreas Müller, Joel Nothman, Gilles Louppe, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Scikit-learn: Machine Learning in Python, June 2018. URL http://arxiv.org/abs/1201.0490. arXiv:1201.0490 [cs.LG].

Lorenzo Posani, Shuqi Wang, Samuel P. Muscinelli, Liam Paninski, and Stefano Fusi. Rarely categorical, highly separable representations along the cortical hierarchy. *Nature*, pp. 1–12, July 2026. ISSN 1476-4687. doi: 10.1038/s41586-026-10668-4. URL https://www.nature.com/articles/s41586-026-10668-4.

Ali Rahimi and Benjamin Recht. Random Features for Large-Scale Kernel Machines. In J. Platt, D. Koller, Y. Singer, and S. Roweis (eds.), *Advances in Neural Information Processing Systems*, volume 20. Curran Associates, Inc., 2007. URL https://proceedings.neurips.cc/paper_files/paper/2007/file/013a006f03dbc5392effeb8f18fda755-Paper.pdf.

Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian processes for machine learning*. Adaptive computation and machine learning. MIT Press, Cambridge, Mass., 3. print edition, 2008. ISBN 978-0-262-18253-9.

Gabriel Riutort-Mayol, Paul-Christian Bürkner, Michael R. Andersen, Arno Solin, and Aki Vehtari. Practical Hilbert space approximate Bayesian Gaussian processes for probabilistic programming, March 2022. URL http://arxiv.org/abs/2004.11408. arXiv:2004.11408 [stat.CO].

Malcolm Sambridge. A Parallel Tempering algorithm for probabilistic sampling and multimodal optimization. *Geophysical Journal International*, 196(1):357–374, January 2014. ISSN 1365-246X, 0956-540X. doi: 10.1093/gji/ggt342. URL https://academic.oup.com/gji/article/196/1/357/585739.

Shannon S. Schiereck, Danilo Trinidad Pérez-Rivera, Andrew Mah, Margaret L. DeMaegd, David Hocker, Royall McMahon Ward, Cristina Savin, and Christine M. Constantinople. The orbitofrontal cortex updates beliefs for state inference. *Neuron*, 114(3):507–520.e8, February 2026. ISSN 0896-6273. doi: 10.1016/j.neuron.2025.10.024. URL https://www.cell.com/neuron/abstract/S0896-6273(25)00805-0.

Roman Shusterman, Matthew C. Smear, Alexei A. Koulakov, and Dmitry Rinberg. Precise olfactory responses tile the sniff cycle. *Nature Neuroscience*, 14(8):1039–1044, August 2011. ISSN 1546-1726. doi: 10.1038/nn.2877. URL https://www.nature.com/articles/nn.2877.

Arno Solin and Simo Särkkä. Hilbert space methods for reduced-rank Gaussian process regression. *Statistics and Computing*, 30(2):419–446, March 2020. ISSN 1573-1375. doi: 10.1007/s11222-019-09886-w. URL https://doi.org/10.1007/s11222-019-09886-w.

Pavel Sountsov, Colin Carroll, and Matthew D. Hoffman. Running Markov Chain Monte Carlo on Modern Hardware and Software. In *Handbook of Markov Chain Monte Carlo*. Chapman and Hall/CRC, 2 edition, 2026. Num Pages: 22.

Clara Kwon Starkweather, Benedicte M. Babayan, Naoshige Uchida, and Samuel J. Gershman. Dopamine reward prediction errors reflect hidden-state inference across time. *Nature Neuroscience*, 20(4):581–589, April 2017. ISSN 1546-1726. doi: 10.1038/nn.4520. URL https://www.nature.com/articles/nn.4520.

Gabriel M. Stine, Eric M. Trautmann, Danique Jeurissen, and Michael N. Shadlen. A neural mechanism for terminating decisions. *Neuron*, 111(16):2601–2613.e5, August 2023. ISSN 0896-6273. doi: 10.1016/j.neuron.2023.05.028. URL https://www.sciencedirect.com/science/article/pii/S0896627323004002.

Saifuddin Syed, Vittorio Romaniello, Trevor Campbell, and Alexandre Bouchard-Côté. Parallel Tempering on Optimized Paths, June 2021. URL http://arxiv.org/abs/2102.07720. arXiv:2102.07720 [stat.CO].

Sean Talts, Michael Betancourt, Daniel Simpson, Aki Vehtari, and Andrew Gelman. Validating Bayesian Inference Algorithms with Simulation-Based Calibration, October 2020. URL http://arxiv.org/abs/1804.06788. arXiv:1804.06788 [stat.ME].

Aki Vehtari, Andrew Gelman, and Jonah Gabry. Practical Bayesian model evaluation using leave-one-out cross-validation and WAIC. *Statistics and Computing*, 27(5):1413–1432, September 2017. ISSN 1573-1375. doi: 10.1007/s11222-016-9696-4. URL https://doi.org/10.1007/s11222-016-9696-4.

Aki Vehtari, Andrew Gelman, Daniel Simpson, Bob Carpenter, and Paul-Christian Bürkner. Rank-Normalization, Folding, and Localization: An Improved R^ for Assessing Convergence of MCMC (with Discussion). *Bayesian Analysis*, 16(2):667–718, June

2021. ISSN 1936-0975, 1931-6690. doi: 10.1214/20-BA1221. URL https://projecteuclid.org/journals/bayesian-analysis/volume-16/issue-2/Rank-Normalization-Folding-and-Localization--An-Improved-R%cb%86-for/10.1214/20-BA1221.full.

Aki Vehtari, Daniel Simpson, Andrew Gelman, Yuling Yao, and Jonah Gabry. Pareto Smoothed Importance Sampling, March 2024. URL http://arxiv.org/abs/1507.02646. arXiv:1507.02646 [stat.CO].

Francis R. Willett, Donald T. Avansino, Leigh R. Hochberg, Jaimie M. Henderson, and Krishna V. Shenoy. High-performance brain-to-text communication via handwriting. *Nature*, 593(7858): 249–254, May 2021. ISSN 1476-4687. doi: 10.1038/s41586-021-03506-2. URL https://www.nature.com/articles/s41586-021-03506-2.

Alex H. Williams, Tony Hyun Kim, Forea Wang, Saurabh Vyas, Stephen I. Ryu, Krishna V. Shenoy, Mark Schnitzer, Tamara G. Kolda, and Surya Ganguli. Unsupervised Discovery of Demixed, Low-Dimensional Neural Dynamics across Multiple Timescales through Tensor Component Analysis. *Neuron*, 98(6):1099–1115.e8, June 2018. ISSN 0896-6273. doi: 10.1016/j.neuron.2018.05.015. URL https://www.sciencedirect.com/science/article/pii/S0896627318303878.

Alex H. Williams, Ben Poole, Niru Maheswaranathan, Ashesh K. Dhawale, Tucker Fisher, Christopher D. Wilson, David H. Brann, Eric M. Trautmann, Stephen Ryu, Roman Shusterman, Dmitry Rinberg, Bence P. Ölveczky, Krishna V. Shenoy, and Surya Ganguli. Discovering Precise Temporal Patterns in Large-Scale Neural Recordings through Robust and Interpretable Time Warping. *Neuron*, 105(2):246–259.e8, January 2020. ISSN 0896-6273. doi: 10.1016/j.neuron.2019.10.020. URL https://www.sciencedirect.com/science/article/pii/S0896627319308943.

Byron M. Yu, John P. Cunningham, Gopal Santhanam, Stephen I. Ryu, Krishna V. Shenoy, and Maneesh Sahani. Gaussian-Process Factor Analysis for Low-Dimensional Single-Trial Analysis of Neural Population Activity. *Journal of Neurophysiology*, 102(1):614–635, July 2009. ISSN 0022-3077. doi: 10.1152/jn.90941.2008. URL https://journals.physiology.org/doi/full/10.1152/jn.90941.2008.

A Method Details

A.1 Extending EFGP

Weight-space approximation methods reduce the $\mathcal{O}(T^3)$ computational burden of standard GP regression on T data points by converting the problem to Bayesian linear regression with $M \ll T$ nonlinear basis functions at the expense of incurring some approximation error. The key goal is to choose basis functions which provide low error for a given budget M. A seminal work in this area was Random Fourier Features (RFFs), which leveraged Fourier duality between a translation-invariant kernel (such as the squared exponential kernel) and its spectral density in frequency space (Rahimi & Recht, 2007; Rasmussen & Williams, 2008). Specifically, for a positive-definite translation-invariant kernel $k(x, y) = k(x - y) = k(r)$, we can write the following equalities:

$$k(r) = \int S(s) e^{2\pi i s^\top r} ds \tag{8}$$

$$S(s) = \int k(r) e^{-2\pi i r^\top s} dr \tag{9}$$

RFFs leverage the fact that equation 8 can be rewritten as the following expectation:

$$k(r) = \mathbb{E}_{s\sim S}[e^{2\pi i s^\top r}] \tag{10}$$

and approximate this expectation over frequencies via simple Monte Carlo. This inherits the standard Monte Carlo error rate of $\mathcal{O}\left(M^{-1/2}\right)$. The spectral density $S(s)$ for a 1-dimensional squared-exponential kernel is well-characterized:

$$S(s) = \ell\sqrt{2\pi} e^{-2\pi^2 \ell^2 \|s\|^2} \tag{11}$$

For our use case, we are doing 1-dimensional GP regression to learn the template functions. This means the integral of equation 8 is also 1-dimensional and can be approximated using quadrature methods rather than a Monte Carlo estimate; EFGP is a recently developed fast algorithm which defines a quadrature scheme *guaranteed* to point-wise approximate the desired squared exponential kernel within a user-specified tolerance ϵ (Greengard et al., 2025). This is an equispaced quadrature scheme, meaning it is wholly specified by the spacing between quadrature nodes h, and the (symmetric) truncation bound L. This results in a set of Fourier basis functions, reweighted by their spectral density, which can be used for template function inference. As the tolerance decreases, the spacing h decreases and the truncation bound L increases to define more basis functions and perform higher-fidelity quadrature of this integral. However, all prior work on EFGP assumed approximation to a single squared exponential kernel with fixed length scale ℓ and output scale σ.

Here, we provide an informal sketch to extend EFGP for efficient hyperparameter sampling. Let the tuple (h, L) denote the EFGP quadrature scheme for a given ℓ. As we decrease ℓ, note that the spectral density in equation 11 becomes wider and flatter; intuitively, lower length scales generate higher frequency components of the observed data. Therefore, for a fixed ϵ, the EFGP quadrature scheme will tend to *increase* h and L as ℓ decreases. As we increase ℓ, the spectral density becomes narrower and taller and must be sampled more finely to resolve it; higher length scales concentrate towards lower frequency fluctuations in the data. In this setting, the EFGP quadrature scheme will tend to *decrease* h and L as ℓ increases to maintain a fixed error tolerance. Say we have a bounded prior over ℓ so that we know $\ell \in [\ell_{\text{min}}, \ell_{\text{max}}]$. In this case, let $(h_{\text{min}}, L_{\text{min}})$ denote the EFGP quadrature rule for ℓ_{min} and $(h_{\text{max}}, L_{\text{max}})$ denote the rule for ℓ_{max}, both at the same ϵ. Following the previous argument, we know $L_{\text{min}} \geq L_{\text{max}}$ and $h_{\text{max}} \leq h_{\text{min}}$. Therefore, we define a *worst-case* quadrature scheme $(h_{\text{max}}, L_{\text{min}})$ which is guaranteed to approximate *better* than the user-specified tolerance for all $\ell_{\text{min}} \leq \ell \leq \ell_{\text{max}}$, as it is either truncating further than necessary or spacing nodes closer than necessary to achieve ϵ error. Each ℓ update step simply rescales the fixed basis set based on the spectral density defined for the new ℓ. Empirically, we find that even when the bounds on the length scale are fairly loose (an order of magnitude separating $\ell_{\text{min}} = 0.05$ and $\ell_{\text{max}} = 0.5$), this worst-case scheme still results in a tractable number of basis functions ($M = 74$ for $\epsilon = 0.1$), avoids the need for recomputation with each parameter update, and can be shared across all neurons. Although we implemented this approach mainly for ease-of-compatibility with accelerator hardware along with approximation guarantees, future work will extend the worst-case quadrature scheme to incorporate explicit knowledge about the prior apart from the bounds and formally benchmark relative to recomputation with each hyperparameter update.

We note that, for the squared exponential kernel, an alternative argument for using Fourier basis functions involves choosing eigenfunctions of the Laplacian on a bounded domain as the optimal low-dimensional basis to project the infinite-dimensional kernel onto, as formalized in Hilbert Space GPs (HSGPs) (Solin & Särkkä, 2020; Riutort-Mayol et al., 2022). However, to our knowledge, the number of basis functions and domain for HSGPs are typically chosen via empirical heuristics which may require multiple MCMC fits to ensure approximation error does not degrade (Riutort-Mayol et al., 2022). Our extension to EFGP instead inherits an explicit pointwise error guarantee that holds over $\ell \in [\ell_{\text{min}}, \ell_{\text{max}}]$ by construction and does not require multiple MCMC refit loops.

A.2 Template Prior Specification

To improve MCMC efficiency, we sample the kernel hyperparameters in an unconstrained space and transform to the constrained domain using the logistic sigmoid followed by appropriate scaling and shifting (Carpenter et al., 2017). Additionally, we place weakly informative Gaussian priors on the unconstrained parameters which results in a logit-normal distribution in the constrained space centered on the midpoint of the bounds:

$$
\begin{aligned}
\sigma_n^\dagger &\sim \mathcal{N}(0,1) \\
\ell_n^\dagger &\sim \mathcal{N}(0,1) \\
\ell_n &= \ell_{\text{min}} + (\ell_{\text{max}} - \ell_{\text{min}}) \frac{1}{1+\exp(-\ell_n^\dagger)} \\
\sigma_n &= \sigma_{\text{min}} + (\sigma_{\text{max}} - \sigma_{\text{min}}) \frac{1}{1+\exp(-\sigma_n^\dagger)}
\end{aligned}
$$

Because the prior is specified directly on the unconstrained parameters, no Jacobian adjustment is required when sampling, which simplifies inference.

These hyperparameters combine with the EFGP approximation from appendix A.1 to generate a non-centered parameterization of the Gaussian process:

$$\beta_n^{(m)} \sim \mathcal{N}(0,1) \tag{12}$$

$$\alpha_n^{(m)} = \sigma_n \sqrt{h S_{\ell_n}(s_m)} \tag{13}$$

$$f_n(\tau) = \sum_{m=1}^{M} \beta_n^{(m)} \alpha_n^{(m)} \psi_m(\tau) \tag{14}$$

where $a^{(i)}$ indicates index i of the vector $\boldsymbol{a}$, h denotes the grid spacing of the EFGP quadrature rule, $S_{\ell_n}(s_m)$ denotes the evaluation of the spectral density associated with length scale ℓ_n at the frequency s_m of sinusoidal basis function $\psi_m(\cdot)$. We parameterize the ψ_m such that each quadrature node (frequency) is associated with a real pair of padded sine and cosine (padding defined as specified in 2.1.1). This allows for simpler conversion to the desired functional form in equation 6.

A.3 Warping Prior Specification

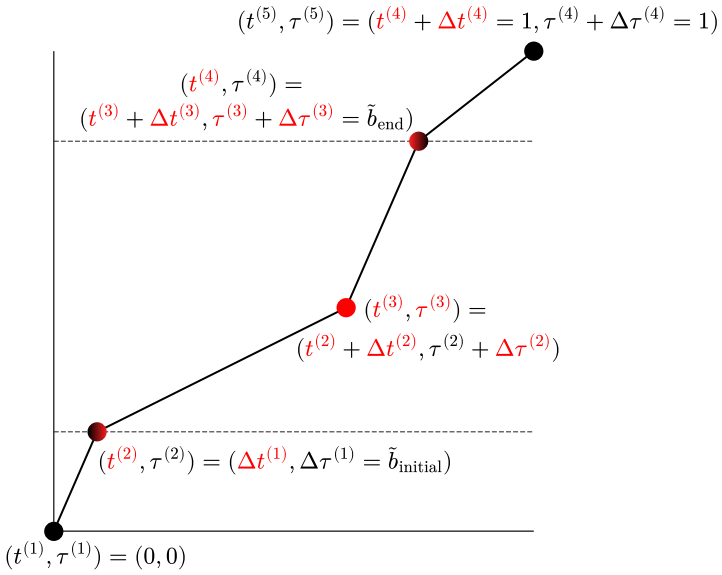

Figure S1: Annotated schematic of example warping function that is piecewise-linear with 1 knot ($J = 5$). All quantities in red are learned, whereas all quantities in black are fixed. Dashed lines indicate padding regions.

Similar to the template scale parameters discussed in section 2.1.1 and appendix A.2, we constrain the warp scale such that $\gamma_k \in [0, \gamma_{\text{max}}]$, sample in an unconstrained space with a weakly informative Gaussian prior, and transform back to the constrained domain with the logistic sigmoid followed by scaling:

$$\gamma_k^{\dagger} \sim \mathcal{N}(0,1)$$

$$\gamma_k = \gamma_{\text{max}} \frac{1}{1 + \exp(-\gamma_k^{\dagger})}$$

The warp scale is used for a non-centered parameterization of the unconstrained intervals which are then passed through an augmented softmax to satisfy padding constraints and approach identity

warping as $\gamma_k \to 0$:

$$\Delta\tau_k^{(1)} = \tilde{b}_{\text{initial}} \tag{15}$$

$$\Delta\tau_k^{(J-1)} = 1 - \tilde{b}_{\text{end}} \tag{16}$$

$$\boldsymbol{d\tau}_k \in \mathbb{R}^{J-3} \tag{17}$$

$$\boldsymbol{d\tau}_k \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}) \tag{18}$$

$$\Delta\tau_k^{(1+i)} = (\tilde{b}_{\text{end}} - \tilde{b}_{\text{initial}}) \frac{\exp(\sqrt{\gamma_k} d\tau_k^{(i)})}{\sum_{j=1}^{J-3} \exp(\sqrt{\gamma_k} d\tau_k^{(j)})} \tag{19}$$

$$\boldsymbol{dt}_k \in \mathbb{R}^{J-1} \tag{20}$$

$$\boldsymbol{dt}_k \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}) \tag{21}$$

$$\Delta t_k^{(i)} = \frac{\exp(\log \Delta\tau_k^{(i)} + \sqrt{\gamma_k} dt_k^{(i)})}{\sum_{j=1}^{J-1} \exp(\log \Delta\tau_k^{(j)} + \sqrt{\gamma_k} dt_k^{(j)})} \tag{22}$$

where $a^{(i)}$ indicates index i of the vector $\boldsymbol{a}$. If a shift is desired, we fix $\Delta t_k^{(2)} = \tilde{b}_{\text{end}} - \tilde{b}_{\text{initial}}$ and appropriately rescale the softmax to define $\Delta t_k^{(1)}$ and $\Delta t_k^{(3)}$ via a softmax over $\boldsymbol{dt}_k \in \mathbb{R}^2$. We can then generate the J knots by taking the cumulative sum across each interval vector, pairing entries together into tuples, and appending $(t_k^{(1)}, \tau_k^{(1)}) = (0, 0)$ to the set. Finally, these knots can be used directly in equation 7 to compute the warping function. Notably, this non-centered parameterization in the space of unconstrained parameters *cannot* easily be achieved with a Dirichlet prior, even if sampling in an unconstrained space. This can create funnel-like geometry when trying to infer the Dirichlet concentration parameters which limits the efficiency of gradient-based samplers such as HMC.

A.4 Inference

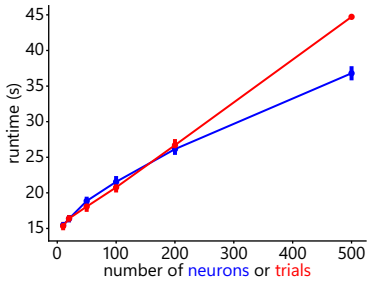

Figure S2: Runtime scaling behavior of model across varying neuron count or trial count in synthetic data. Points are mean runtime $\pm 2 \times SD$ across 5 runs. Synthetic data parameters were $T = 100$, $N = 20$ for varying K and vice versa. Sampler parameters were fixed to those used on the real data fit (Table 1).

We assume familiarity with standard Hamiltonian Monte Carlo (HMC) techniques (Neal, 2026; Betancourt, 2018; Carpenter et al., 2017). We choose vanilla Hamiltonian Monte Carlo rather than the commonly used No-U-Turn Sampler (NUTS) as recent work has shown HMC to have better parallelization properties on modern accelerator hardware by avoiding variable tree depths between parallel NUTS samplers (Sountsov et al., 2026). We still adapt the step size and a diagonal mass matrix during the warm-up period as is standard for NUTS (Carpenter et al., 2017). The sampling procedure (Algorithm 1) is easily parallelizable across chains, so we include support for sharding multi-chain inference across multiple GPUs to allow for more samples to be drawn given access to multi-GPU workflows.

Parallel tempering is performed between a user-specified number of rungs R at log-spaced temperatures ranging from 1 to a user-specified max tempering factor F_{max}. On even iterations, swaps are proposed between all adjacent pairs in parallel starting with $r = 0$ and $r = 1$, $r = 2$ and $r = 3$,

and so on. On odd iterations, swaps are proposed from the other set of adjacent pairs $r = 1$ and $r = 2$, $r = 3$ and $r = 4$, and so on. This follows standard practice to ensure information rapidly spreads from the tempered distributions to the base distribution (Syed et al., 2021). As stated in the main text, these swaps are performed block-wise exactly analogous to the HMC steps, meaning template swaps are proposed independently and in parallel across neurons and warp swaps are proposed independently and in parallel across trials.

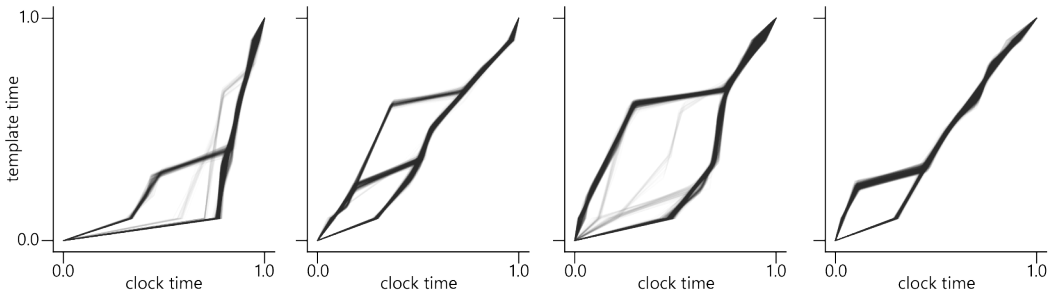

Figure S3: Full population warping function posteriors exhibit substantial multimodality for four example trials from the rat OFC data used in section 3.3. Two thousand samples from the posterior visualized per trial. Further data details in B.3 and sampling parameters in table 1.

A.5 Review of PSIS-LOO

Cross-validation estimates fitted models' predictive ability on unseen test data as a heuristic to choose the best model from the set. It does so by repeatedly fitting to different subsets of the training data, using the validation sets to quantify expected performance on test data, and picking the model which maximizes this expected performance. A desirable approach is leave-one-out (LOO) cross-validation, which sweeps over each individual training point as the validation set.

Our goal is to compute the log leave-one-out predictive density summed over all data points (elpd_{loo}):

$$\text{elpd}_{\text{loo}} = \sum_{i=1}^{N} \log p(y_i \mid y_{-i}) \tag{23}$$

where $p(y_i \mid y_{-i})$ denotes the model's posterior predictive density for data point i given all other data points (denoted by y_{-i}). Given two models, we compute the expectation of the paired differences between the models' predictive densities $\text{elpd}_{\text{diff}}$ as well as the standard error of this difference SE_{diff}. Given many models, we can find the model which maximizes elpd_{loo} and use this model's predictions as a baseline to compute $\text{elpd}_{\text{diff}}$ against each other model. Based on this set of $\text{elpd}_{\text{diff}}$ values, a practitioner can make an informed decision for how to proceed with model selection. For our use case, we choose the simplest model that contains 0 within the interval defined by $\text{elpd}_{\text{diff}} \pm 2\text{SE}_{\text{diff}}$ as the most parsimonious explanation consistent with the observed data (Vehtari et al., 2017). We choose to hold out (neuron, trial)-tuples as the unit of LOO accuracy.

A key quantity required for this model selection procedure is $\log p(y_i \mid y_{-i})$ for every data point i. Naively, this requires N model refits to compute, which becomes prohibitive in the setting of Bayesian inference applied to large datasets. We use Pareto-Smoothed Importance Sampling (PSIS) to efficiently compute an approximation of this leave-one-out density for each data point from a *single* model fit (Vehtari et al., 2024). For intuition behind PSIS, note that data point i enters into the full posterior formulation solely through its likelihood $p(y_i \mid \theta)$. Therefore, we can use importance sampling to approximate the leave-one-out posterior by dividing the full-data posterior by the likelihood of data point i. These reciprocal likelihoods tend to have unstable tails, so we smooth the tails of the weights by replacing them with the best-fitting generalized Pareto distribution. Overall, PSIS-LOO provides a principled yet tractable approach for Bayesian model comparison between a discrete set of alternatives.

A.6 WARP SWAP DISSIMILARITY

To compare neurons based on similarities or differences in their timing, we first compute the single-neuron posteriors over warping functions. For neuron n, let $p_n(w_k)$ denote its single-neuron posterior distribution over warping functions for trial k and $p_n(\tilde{\lambda})$ denote its posterior distribution over its template function. We observe S approximate samples from the single-neuron posterior distributions via MCMC:

$$w_{nk}^{(i)} \sim p_n(w_k)$$
$$\tilde{\lambda}_n^{(i)} \sim p_n(\tilde{\lambda})$$

for $1 \leq i \leq S$. We can quantify the effect on the log-likelihood for random shuffling of the sample order by drawing C pairs of (i, j) uniformly from $1 \leq i, j \leq S$ and computing the likelihood of each sample pair:

$$\mathcal{L}_{nk}^{(c)} = \sum_{t=1}^{T} \log p(y_{nk}(t) \mid w_{nk}^{(i_c)}, \tilde{\lambda}_n^{(j_c)}) \tag{24}$$

for $1 \leq c \leq C$. We can take this sum across trials and center the values:

$$\tilde{\mathcal{L}}_n^{(c)} = \sum_{k=1}^{K} \mathcal{L}_{nk}^{(c)} \tag{25}$$

$$\mathcal{L}_n^{(c)} = \tilde{\mathcal{L}}_n^{(c)} - \frac{1}{C} \sum_{a=1}^{C} \tilde{\mathcal{L}}_n^{(a)} \tag{26}$$

for $1 \leq c \leq C$. The ensemble of these $\mathcal{L}_n^{(c)}$ values defines what we call the *self-null* log-likelihood distribution, or the distribution of log-likelihoods caused by random shuffling of sample numbers.

We can also define a distribution for neuron n as a function of neuron m which computes the log-likelihood of observing the data for neuron n given template samples from neuron n and warp samples from neuron m:

$$\mathcal{S}_{nk}^{(c)}(m) = \sum_{t=1}^{T} \log p(y_{nk}(t) \mid w_{mk}^{(i_c)}, \tilde{\lambda}_n^{(j_c)}) \tag{27}$$

for $1 \leq c \leq C$. Similarly, we take the sum across trials and center the values with respect to the self-null distribution:

$$\tilde{\mathcal{S}}_n^{(c)}(m) = \sum_{k=1}^{K} \mathcal{S}_{nk}^{(c)}(m) \tag{28}$$

$$\mathcal{S}_n^{(c)}(m) = \tilde{\mathcal{S}}_n^{(c)}(m) - \frac{1}{C} \sum_{a=1}^{C} \tilde{\mathcal{L}}_n^{(a)} \tag{29}$$

for $1 \leq c \leq C$. The ensemble of these $\mathcal{S}_n^{(c)}(m)$ values defines what we call the *warp swap* log-likelihood distribution, or the distribution of log-likelihoods for the data of neuron n caused by swapping of warps from neuron m onto neuron n.

Intuitively, if neurons m and n are warped differently from each other, the warp swap distribution $\mathcal{S}_n^{(c)}(m)$ will be left-shifted (lower log-likelihoods) compared to the self-null distribution $\mathcal{L}_n^{(c)}$. In this case, we would expect the other direction of the swap to also exhibit this behavior: the warp swap distribution $\mathcal{S}_m^{(c)}(n)$ will be left-shifted compared to the self-null distribution $\mathcal{L}_m^{(c)}$. However, if neurons m and n are warped similarly to each other, we expect these distributions to be closer, and one of these directions may actually exhibit the warp swap distribution being right-shifted compared to the self-null. This can happen if neuron m has a lower firing rate than neuron n (or vice versa), causing its posterior samples to be drawn closer to the prior. Because the swap and null distributions only compute the likelihood, swapping the warps from neuron n onto neuron m may increase the likelihood of the observed data from neuron m.

Therefore, we propose that quantifying the degree of *left-shift* from the self-null distribution to the swap distribution is a useful signal for warping similarity. We take the paired differences between samples of the self-null and the swap distribution, rectify to only penalize left-shifts, take the mean, and normalize by the standard deviation ξ of the corresponding self-null:

$$\tilde{d}_n(m) = \frac{1}{C\xi_n} \sum_{c=1}^{C} \max(\mathcal{L}_n^{(c)} - \mathcal{S}_n^{(c)}(m), 0) \tag{30}$$

To symmetrize this into a proper dissimilarity, we take the minimum of the pairwise evaluations of equation 30:

$$d(n, m) = \min(\tilde{d}_n(m), \tilde{d}_m(n)) \tag{31}$$

The intuition for using the minimum instead of the average is to account for situations as described above, where the warps of neuron n largely improve the likelihood of neuron m. In these cases, it is likely that $\tilde{d}_n(m) \gg 0$ whereas $\tilde{d}_m(n) \approx 0$. It naturally follows that taking the minimum of the two would correctly result in these two neurons being treated as similar, whereas taking the average would artificially inflate the dissimilarity.

B Experimental Details

All experiments were conducted on 2 NVIDIA H100-80GB GPUs.

B.1 Synthetic Validation

Synthetic datasets were generated from the generative model using the settings in table 1. For each of 20 datasets, the template hyperparameters (ℓ_n, σ_n), basis weights β_n, warp scales γ_k, and unconstrained interval vectors ($d\mathbf{t}_k$, $d\boldsymbol{\tau}_k$) were drawn from their priors, and Poisson spike counts were sampled for each time bin. Warping functions were piecewise-linear with one interior knot ($J = 5$).

As the strictest form of the coverage check, we compute coverage on the raw sampled parameters rather than the induced template/warping functions, as this includes weakly-informed parameters such as the basis function weights $\boldsymbol{\beta}_n$ or the warp scale γ_k. For each dataset and each target quantity, we computed central credible intervals containing a fraction x of the posterior mass for $x \in \{0.10, 0.20, \ldots, 0.90\}$ and recorded whether the ground-truth value fell inside each interval. Empirical coverage was averaged separately over template and warp quantities within each dataset. Figure 2B reports the mean and ± 2 SE across datasets. For the subsampled fits, we assessed template coverage by refitting on 4 of 40 trials (all neurons) and warp coverage by refitting on 2 of 20 neurons (all trials), with subsets drawn uniformly at random per dataset and all other settings unchanged.

For the dataset shown in figure 2A, we fit shift, linear, 1-, 2-, and 3-knot models and estimated PSIS-LOO over all $N \times K = 800$ (neuron, trial) pairs.

B.2 Subpopulation Identification

Synthetic datasets were generated from the generative model using the settings in table 1. Each synthetic dataset contained 4 subpopulations of 5 neurons each. On each trial, all neurons within a subpopulation shared a single 1-knot warping function, and warps were drawn independently across subpopulations from the warp prior. Each of the 20 neurons was then fit independently ($N = 1$, $K = 40$).

Aligned spike times were obtained by mapping each observed spike time through $w_k^{(s)}$ for every posterior sample s and reporting the across-sample mean and variance (Figure 3A). Posterior-smoothed spike counts were obtained by aligning the binned counts under each posterior sample and averaging over samples (Figure 3A).

Warp swap dissimilarities (see appendix A.6) were computed from $C = 10{,}000$ sample pairs. For every approach, neurons were grouped by hierarchical agglomerative clustering with average linkage and cut at 4 clusters (the ground truth). The two-dimensional non-metric MDS embedding was constructed using `sklearn` (figure 3D) (Pedregosa et al., 2018). There were 3 baselines. First,

affine warping fit to each neuron, using the Frobenius norm of the difference between evaluated warping function matrices ($K \times T$ matrix containing the warping function for each trial evaluated at each timebin) for each neuron. Second, the sum of Euclidean distances between posterior means of the sampled warp parameters across all trials. Third, the Frobenius norm of the difference between the posterior means of the evaluated warping function matrices for each neuron. Agreement with the ground truth was measured by the adjusted Rand index for 20 datasets, and methods were compared with paired (across datasets) Wilcoxon signed-rank tests.

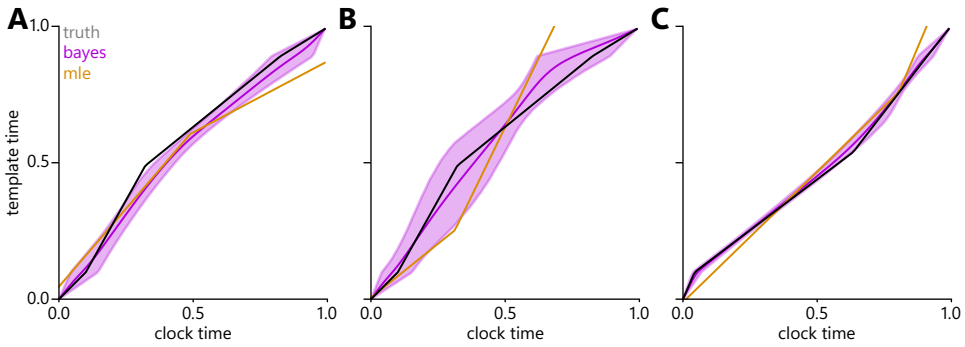

Figure S4: Deterministic warping does not readily generalize to single-neuron warping estimates on synthetic data with known subpopulation structure. A) Single-neuron fits to a warping function from an example trial for an example neuron. B) Single-neuron fits on the same trial for a different neuron in the same ground truth subpopulation as the neuron in panel A. C) Single-neuron fits on the same trial for another neuron in a different ground truth subpopulation as the neurons in panels A-B. Bayes fits represent posterior mean in solid line and central 90% credible interval in shading.

B.3 Rat Decision-Making

Data were collected by Schiereck et al. (2026) from Long-Evans rats performing a temporal wagering task. After the rat initiated a trial, an auditory cue indicating the offered reward volume; the rat could either wait at the reward port until the reward was delivered after an exponential delay, or opt out by poking the opt-out port, which initiated a new trial. On 25% of trials, reward was withheld entirely, forcing rats to exercise the opt-out action. Neural activity was recorded from OFC using Neuropixels 1.0 probes. We analyzed $K = 65$ opt-out trials from session S057_2023-08-03. On all trials, the rat waited at least 2s before opting out so the analysis window excludes the offer cue. Spikes were binned at 20ms over a window from 2s before to 0.5s after the opt-out poke ($T = 125$). We analyze the activity from all $N = 48$ simultaneously recorded neurons in this session.

Population and single-neuron fits were run at each of the five complexities with the settings in table 1. For single-neuron fits, PSIS-LOO was computed over that neuron's K trials. To test whether single-neuron model selection is biased towards simpler warping complexities, we simulated data with $N = 20, K = 40, T = 100$. In the *homogeneous* control, all neurons shared the same 1-knot warp on each trial. In the *heterogeneous* control, neurons were divided into 4 subpopulations of 5 neurons each with different 1-knot warps for each subpopulation. Single-neuron PSIS-LOO was performed and accuracy is reported as the fraction of neurons whose selected complexity matched the ground truth (1 knot) (Figure S5).

Quality of model fits was assessed with trial-split Poisson log-likelihood: we take the trial average of a neuron's activity on even trials and call it the "template", warp this template with the model's fitted warping functions on odd trials to generate predictions, and compute the log-likelihood of the observed data on odd trials given a Poisson likelihood centered at the predictions. From this log-likelihood, we subtract the log-likelihood of a null model (a constant firing rate set by the mean activity of the neuron on odd trials). We do this for the raw data (assuming identity warping functions), the activity after alignment with the warping functions fit to the population at all warping complexities, and the activity after alignment with the warping functions fit to single neurons at their optimal complexity. Because the alignments give samples from the posterior, we compute this quantity for each sample and take the average across samples for comparison (Figure S6). The trial-split log-likelihood values for the raw data, single-neuron fits, and population fits at matching

complexity were compared with paired Wilcoxon signed-rank tests for neurons with complexity higher than linear.

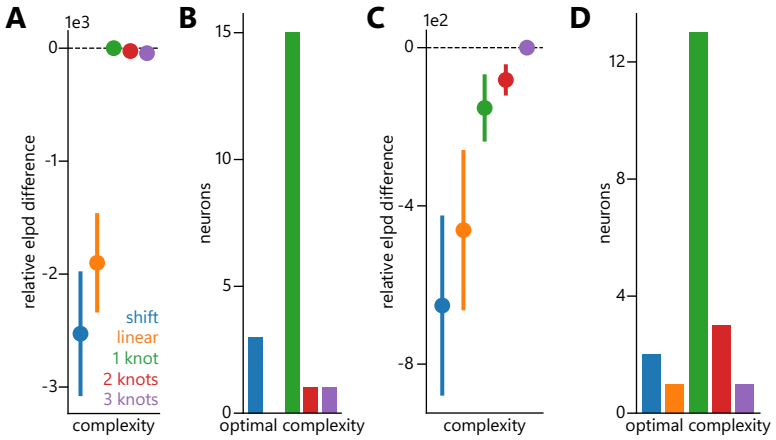

Figure S5: Homogeneous (A-B) and heterogeneous (C-D) population warping controls on synthetic data for PSIS-LOO. A) Population PSIS-LOO model selection on synthetic data with a 1-knot warp shared across the population correctly selects 1-knot. B) Single-neuron PSIS-LOO model selection correctly selects 1-knot on 15 of 20 neurons. C) Population PSIS-LOO model selection on synthetic data with 4 subpopulations with different 1 knot warps incorrectly selects 3 knots. D) Single-neuron PSIS-LOO model selection correctly selects 1-knot on 14 of 20 neurons.

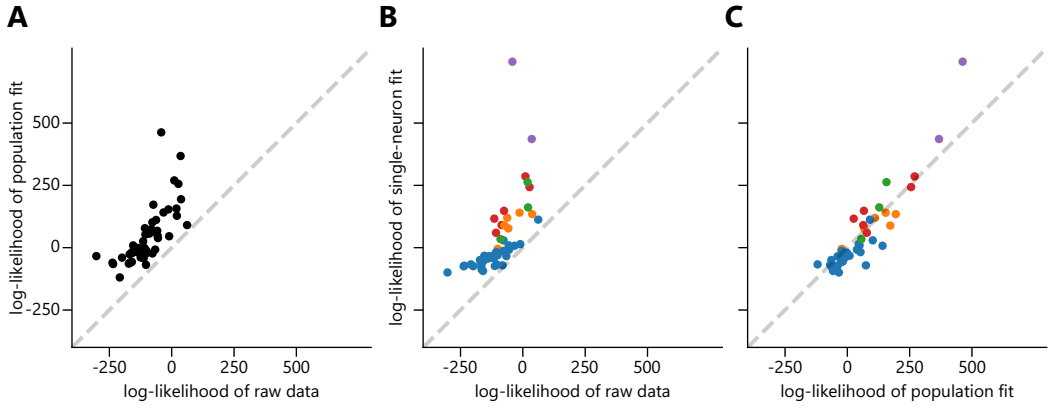

Figure S6: Trial-average log-likelihood comparisons for individual neurons. A) Population fit with 3-knots compared to raw data. B) Single-neuron optimal fits compared to raw data. Color legend same as figure 4. C) Single-neuron optimal fits compared to population fit with 3-knots.

Algorithm 1 parallel, blocked HMC-within-Gibbs with parallel tempering

Input: spike count data $\mathbf{Y}$, iteration i, current template parameters $\boldsymbol{\theta}^{(i)}_{\text{template}}$, current warp parameters $\boldsymbol{\theta}^{(i)}_{\text{warp}}$, current sampling parameters $\boldsymbol{\theta}^{(i)}_{\text{sampler}}$, hyperparameters Ω

procedure HMC-WITHIN-GIBBS($\mathbf{Y}, i, \boldsymbol{\theta}^{(i)}_{\text{template}}, \boldsymbol{\theta}^{(i)}_{\text{warp}}, \boldsymbol{\theta}^{(i)}_{\text{sampler}}, \Omega$)
  $\boldsymbol{W}^{(i)} \leftarrow$ params2warps($\boldsymbol{\theta}^{(i)}_{\text{warp}}, \Omega$) ▷ matrix of evaluated warping functions
  $\boldsymbol{\theta}^{(i+1)}_{\text{template}} \leftarrow$ template_hmc($\mathbf{Y}, \boldsymbol{\theta}^{(i)}_{\text{template}}, \boldsymbol{W}^{(i)}, \boldsymbol{\theta}^{(i)}_{\text{sampler}}, \Omega$) ▷ parallelized across neurons
  $\tilde{\boldsymbol{\Lambda}}^{(i+1)} \leftarrow$ params2templates($\boldsymbol{\theta}^{(i+1)}_{\text{template}}, \Omega$) ▷ matrix of evaluated template functions
  $\boldsymbol{\theta}^{(i+1)}_{\text{warp}} \leftarrow$ warp_hmc($\mathbf{Y}, \boldsymbol{\theta}^{(i)}_{\text{warp}}, \tilde{\boldsymbol{\Lambda}}^{(i+1)}, \boldsymbol{\theta}^{(i)}_{\text{sampler}}, \Omega$) ▷ parallelized across trials
  warmup_flag $\leftarrow$ is_warmup(i, Ω)
  if warmup_flag **then**
    $\boldsymbol{\theta}^{(i+1)}_{\text{sampler}} \leftarrow$ sampler_adaptation($\boldsymbol{\theta}^{(i+1)}_{\text{template}}, \boldsymbol{\theta}^{(i+1)}_{\text{warp}}, \boldsymbol{\theta}^{(i)}_{\text{sampler}}, \Omega$)
  end if
  return $\boldsymbol{\theta}^{(i+1)}_{\text{template}}, \boldsymbol{\theta}^{(i+1)}_{\text{warp}}, \boldsymbol{\theta}^{(i+1)}_{\text{sampler}}$
end procedure

procedure TEMPERING($\mathbf{Y}, \boldsymbol{\theta}^{(i)}_{\text{template}}, \boldsymbol{\theta}^{(i)}_{\text{warp}}, \Omega$)
  $\boldsymbol{W}^{(i)} \leftarrow$ params2warps($\boldsymbol{\theta}^{(i)}_{\text{warp}}, \Omega$)
  $\boldsymbol{\theta}^{(i+1)}_{\text{template}} \leftarrow$ template_rung_swap($\mathbf{Y}, \boldsymbol{\theta}^{(i)}_{\text{template}}, \boldsymbol{W}^{(i)}, \Omega$) ▷ parallelized across neurons
  $\tilde{\boldsymbol{\Lambda}}^{(i+1)} \leftarrow$ params2templates($\boldsymbol{\theta}^{(i+1)}_{\text{template}}, \Omega$)
  $\boldsymbol{\theta}^{(i+1)}_{\text{warp}} \leftarrow$ warp_rung_swap($\mathbf{Y}, \boldsymbol{\theta}^{(i)}_{\text{warp}}, \tilde{\boldsymbol{\Lambda}}^{(i+1)}, \Omega$) ▷ parallelized across trials
  return $\boldsymbol{\theta}^{(i+1)}_{\text{template}}, \boldsymbol{\theta}^{(i+1)}_{\text{warp}}$
end procedure

procedure SAMPLE($\mathbf{Y}, i, \boldsymbol{\theta}^{(i)}_{\text{template}}, \boldsymbol{\theta}^{(i)}_{\text{warp}}, \boldsymbol{\theta}^{(i)}_{\text{sampler}}, \Omega$)
  $\boldsymbol{\theta}^{*}_{\text{template}}, \boldsymbol{\theta}^{*}_{\text{warp}}, \boldsymbol{\theta}^{(i+1)}_{\text{sampler}} \leftarrow$ HMC-WITHIN-GIBBS($\mathbf{Y}, i, \boldsymbol{\theta}^{(i)}_{\text{template}}, \boldsymbol{\theta}^{(i)}_{\text{warp}}, \boldsymbol{\theta}^{(i)}_{\text{sampler}}, \Omega$)
  $\boldsymbol{\theta}^{(i+1)}_{\text{template}}, \boldsymbol{\theta}^{(i+1)}_{\text{warp}} \leftarrow$ TEMPERING($\mathbf{Y}, \boldsymbol{\theta}^{*}_{\text{template}}, \boldsymbol{\theta}^{*}_{\text{warp}}, \Omega$)
  return $\boldsymbol{\theta}^{(i+1)}_{\text{template}}, \boldsymbol{\theta}^{(i+1)}_{\text{warp}}, \boldsymbol{\theta}^{(i+1)}_{\text{sampler}}$
end procedure

Table 1: Parameters for experiments.

	Section 3.1	Section 3.2	Section 3.3
Data			
neurons N	20	20 (5 per subpopulation)	48
trials K	40	40	65
time bins T	100	100	125 (20ms bins)
Template Prior			
kernel tolerance ϵ	0.1	0.1	0.1
ℓ_{min}	0.05	0.05	0.05
ℓ_{max}	0.5	0.5	0.5
σ^2_{min}	100	100	100
σ^2_{max}	1000	1000	1000
Warp Prior			
warping complexity	1 knot ($J = 5$)	1 knot ($J = 5$)	varied
initial padding $\tilde{b}_{\text{initial}}$	0.1	0.1	0.1
end padding $\tilde{b}_{\text{end}}$	0.9	0.9	0.9
γ_{max}	0.3	0.5	0.3
Sampler			
chains	4	4	2
samples per chain	4000	2000	2000
warm-up fraction	0.1	0.1	0.5
base step size	0.001	0.001	0.001
target acceptance rate	0.9	0.9	0.9
leapfrog steps	5	5	5
temperature rungs	30	30	10
max tempering factor	100	100	100