Discrete and continuous latent states of neural activity in *Caenorhabditis elegans*

Scott W. Linderman, David M. Blei, and Liam Paninski
Columbia University

November 9, 2017

Abstract

Recent advances in neural recording technologies have enabled simultaneous measurements of the majority of head ganglia neurons in both immobilized and freely-behaving *C. elegans*. The dynamics of neural activity shed light on how *C. elegans* processes sensory information and generates motor activity. To understand these dynamics, we leverage *recurrent switching linear dynamical systems* [Linderman et al., 2017]—probabilistic models that decompose complex time-series into segments with simple, linear dynamics. We incorporate these models into a robust, hierarchical framework for combining information across whole-brain recordings of multiple worms. We use this framework to reveal latent states of population neural activity, along with the discrete modes of behavior that drive dynamics in this latent state space. We find characteristic transition patterns between these discrete modes, and we see that transition probabilities are determined, in part, by the current brain state. This probabilistic framework is a useful aid for neural identification, currently a laborious, manual task. Moreover, we find clusters of neurons that are similarly tuned in latent state space, and we show how the different dynamical modes activate these clusters. Finally, we find a significant overlap between our inferred modes and the manually-labeled states of Kato et al. [2015], which were shown to correspond to different worm behaviors. Our framework automatically discovers these behaviorally-meaningful regimes directly from neural activity.

1 Introduction

The nematode *C. elegans* is a uniquely accessible model organism of neuroscience. With recent advances in optical recording technologies [Schrödel et al., 2013, Prevedel et al., 2014, Nguyen et al., 2016], we now possess a mounting set of tools for measuring the activity of large fractions of these neurons simultaneously. These technologies provide exciting opportunities to probe neural dynamics and their link to sensory processing and behavior.

Recently, Kato et al. [2015] harnessed these methodological advances to study the coordinated activity of hundreds of neurons in head-fixed *C. elegans*. Across multiple organisms, they found that neural activity reliably traced out smooth trajectories in the subspace of the first three principal components. Upon closer evaluation, they found that different components of these trajectories corresponded to different elements of behavior, like forward and reverse crawling, dorsal and ventral turns, etc. These findings raise a number of interesting questions: can we learn a (potentially nonlinear) dynamical system that approximates these low-dimensional dynamics; is there a more appropriate subspace, or collection of subspaces, for capturing these dynamics; can we gain statistical power by partial, noisy recordings across separate worms and trials; and, can we segment these trajectories in an unsupervised manner to glean further insight into the relationship between low-dimensional state and behavior?

We develop a probabilistic framework for investigating these questions. We model these multi-neuronal recordings as a high-dimensional projection of a low-dimensional, dynamical latent state. We allow the dynamics of this latent state change over time, intuitively capturing different patterns of brain activity for different types of behavior. The *switching linear dynamical system* (SLDS) [Chang and Athans, 1978, Ackerson and Fu, 1970, Hamilton, 1990, Ghahramani and Hinton, 1996, Murphy, 1998] captures both of these properties: low-dimensional continuous latent states with different dynamical regimes. The SLDS is a probabilistic time series model characterized by co-evolving discrete and continuous latent states. We extend this class of models with a hierarchical Bayesian framework that captures many of the nuances of these whole-brain recordings. In fitting these models to recordings from multiple *C. elegans*, we characterize the globally nonlinear dynamics that govern neural activity, and we parse these recordings into an interpretable sequence of behavioral segments.

2 Data

The data comes in the form of a collection of matrices $\{\widetilde{Y}^{(w)}\}_{w=1}^{W}$ for each of W worms. The matrix $\widetilde{Y}^{(w)} \in \mathbb{R}^{T_w \times N_w}$ represents the calcium activity of the w-th worm, which consists of T_w time frames and N_w neurons.[1] *C. elegans* is special in that each neuron has a unique name. Assigning names to neurons in calcium imaging data is a challenging task, but typically, out of the ~ 100 neurons observed in any worm, we can label about $20-30$ with high certainty. As such, we choose to represent each matrix $\widetilde{Y}^{(w)}$ in *canonical form* as a tuple $(Y^{(w)}, M^{(w)})$, where $Y^{(w)} \in \mathbb{R}^{T_w \times N}$ is a matrix where each column corresponds to one of the N neurons that is identified in at least one of the W worms, and $M^{(w)} \in \{0,1\}^{T_w \times N}$ is a corresponding *mask* matrix that indicates which neurons were observed in worm w. If $M^{(w)}_{:,n} = 1$, neuron n was labeled in worm w and $Y^{(w)}_{:,n}$ was its observed activity. If $M^{(w)}_{:,n} = 0$, this neuron was not observed in worm w, and the corresponding activity is missing.[2]

3 Hierarchical Generative Model

We model the neural activity with a switching linear dynamical system (SLDS) with three new extensions: (i) a *hierarchical* model to share parameters across worms while also allowing for worm-to-worm variability; (ii) a *robust* model for dynamics noise to help address model misspecification; and (iii) a *recurrent* model to capture how continuous latent states influence discrete state transition probabilities. We will introduce the SLDS first and then present each of these extensions in turn.

3.1 Switching linear dynamical systems

Switching linear dynamical system models (SLDS) break down complex, nonlinear time series data into sequences of simpler, reused dynamical modes. By fitting an SLDS to data, we not only learn a flexible nonlinear generative model, but also learn to parse data sequences into coherent discrete units.

The generative model is as follows. At each time $t = 1, 2, \ldots, T$, for each worm w, there is a discrete latent state $z_t^{(w)} \in \{1, 2, \ldots, K\}$ that follows Markovian dynamics,

$$z_{t+1}^{(w)} \,|\, z_t^{(w)}, \{\pi_k\}_{k=1}^{K} \sim \pi_{z_t^{(w)}} \tag{1}$$

[1] Technically, this calcium activity is a matrix of first-order temporal differences of a bleaching-corrected $\Delta F/F$ signal. Unlike Kato et al. [2015], we do not smooth these derivatives. We work with the raw first-order temporal differences.

[2] The mask is defined in such a way that neurons can be missing for only subsets of time frames, but in our data neurons are either labeled or missing.

where $\{\pi_k\}_{k=1}^K$ is the Markov transition matrix and $\pi_k \in [0,1]^K$ is its kth row. In addition, a continuous latent state $x_t^{(w)} \in \mathbb{R}^D$ follows conditionally linear (or affine) dynamics, where the discrete state $z_t^{(w)}$ determines the linear dynamical system used at time t:

$$x_{t+1}^{(w)} = A_{z_{t+1}^{(w)}} x_t^{(w)} + b_{z_{t+1}^{(w)}} + u_t^{(w)}, \qquad u_t^{(w)} \overset{\text{iid}}{\sim} \mathcal{N}(0, Q_{z_{t+1}^{(w)}}), \tag{2}$$

for matrices $A_k, Q_k \in \mathbb{R}^{D\times D}$ and vectors $b_k \in \mathbb{R}^D$ for $k = 1, 2, \ldots, K$. Finally, at each time t a linear Gaussian observation $y_t^{(w)} \in \mathbb{R}^N$ (some entries of $y_t^{(w)}$ are masked off; recall Section 2) is generated from the corresponding latent continuous state,

$$y_t^{(w)} = C x_t^{(w)} + d + v_t^{(w)}, \qquad v_t^{(w)} \overset{\text{iid}}{\sim} \mathcal{N}(0, S), \tag{3}$$

for $C \in \mathbb{R}^{N\times D}$, $S \in \mathbb{R}^{N\times N}$, and $d \in \mathbb{R}^N$. We denote the rows of C by vectors c_n. For simplicity, we assume C, d, and S to be shared among all discrete states in our model. Moreover, we assume the observation noise is diagonal,

$$S = \operatorname{diag}([s_1, \ldots, s_N]).$$

The system parameters comprise the discrete Markov transition matrix, the library of linear dynamical system matrices, and the neuron-specific emission parameters, which we write as

$$\theta = \{(\pi_k, A_k, Q_k, b_k)\}_{k=1}^K \cup \{c_n, d_n, s_n\}_{n=1}^N.$$

To learn an SLDS using Bayesian inference, we place conjugate Dirichlet priors on each row of the transition matrix and conjugate matrix normal inverse Wishart (MNIW) priors on the linear dynamical system parameters, Gaussian priors on the rows of the emission matrix, and inverse gamma priors on the emission noise. We write this as,

$$\pi_k \mid \alpha \overset{\text{iid}}{\sim} \text{Dir}(\alpha), \qquad [A_k, b_k], Q_k \mid \lambda \overset{\text{iid}}{\sim} \text{MNIW}(\lambda),$$
$$[c_n, d_n] \mid \eta \overset{\text{iid}}{\sim} \mathcal{N}(\eta), \qquad s_n \mid \eta \overset{\text{iid}}{\sim} \text{IG}(\eta).$$

where $[\cdot,\cdot]$ denotes column concatenation, and α, λ, and η denote appropriate hyperparameters of the transitions, dynamics, and emissions, respectively.

3.2 Extensions of the standard SLDS

To better model the *C. elegans* data, we introduce three extensions to the standard SLDS.

Hierarchical model to allow limited variability across worms. The first extension captures shared patterns across worms while still allowing for worm-to-worm variability. We do this in two ways. First, we allow the dynamics parameters to vary slightly from worm to worm with the following model,

$$\begin{aligned} \text{vec}([A_k, b_k]) &\sim \mathcal{N}(\lambda), \\ \text{vec}([A_k^{(w)}, b_k^{(w)}]) &\sim \mathcal{N}(\text{vec}([A_k, b_k]), \sigma^2 I), \\ Q_k^{(w)} &\sim \text{IW}(\lambda). \end{aligned}$$

This allows worm-specific dynamics parameters $(A_k^{(w)}, b_k^{(w)})$ but requires that they not deviate too far from the global mean (A_k, b_k). It also allows for worm-specific dynamics noise.

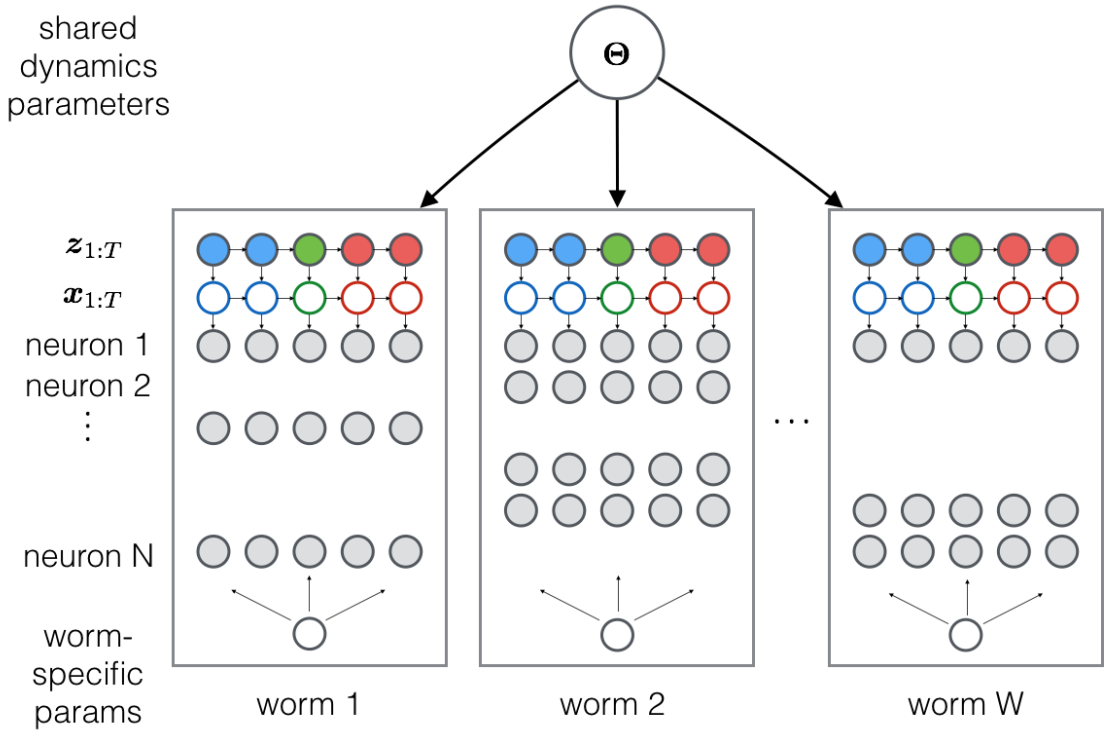

Figure 1: A schematic of the hierarchical SLDS for combining information across multiple worms in order to learn a shared dynamical system for *C. elegans*. We have a set of shared set of model parameters, θ, represented by the large node at the top of the graphical model. Each worm, w has its own set of discrete latent states (colored nodes) and continuous latent states (nodes with colored outlines) and observed neurons. For example, neuron 1 may appear in all neurons whereas neuron 2 may only be seen in worm 2. Thus, each worm provides information about the mapping from latent states to observations for only a subset of neurons. In terms of the model, this corresponds to a subset of rows in the matrix C and the vector d. Finally, we allow each worm to have local parameters that capture, for example, the amount of fluorescent protein expressed in a particular neuron for a given worm.

Second, we allow each worm to have neuron-specific emission variances,

$$s_n^{(w)} \overset{\text{iid}}{\sim} \text{IG}(\eta).$$

We may expect, for example, that calcium indicator expression varies from worm to worm, which in turn could give rise to different levels of observed activity. While this could be modeled as a scaling of the emission matrix, a simpler approach is to let the noise term capture these different fluctuations. We could share the global mean of the worm-specific variances, but we have little reason to expect the $s_n^{(w)}$ and $s_n^{(w')}$ to be correlated under the prior.

Figure 1 illustrates the hierarchical model. Each worm has a set of discrete states $z_{1:T}$, continuous states $x_{1:T}$, and observations for a subset of neurons. The worms share a set of global dynamics parameters and emission matrices, but they also have worm-specific perturbations of the global dynamics, as well as worm-specific observation variances.

Robust dynamics model. While the SLDS can capture nonlinear dynamics by composing simple linear pieces, it is still an approximation to the true data-generating process. To account for some of this model

misspecification, we introduce a *robust* extension of the SLDS by allowing for heavy-tailed noise in the dynamics. Specifically, we replace (2) with,

$$x_{t+1}^{(w)} = A_{z_{t+1}^{(w)}} x_t^{(w)} + b_{z_{t+1}^{(w)}} + u_t^{(w)}, \qquad u_t^{(w)} \overset{\text{iid}}{\sim} \text{t}(0, Q_{z_{t+1}^{(w)}}, \nu_{z_{t+1}^{(w)}}), \tag{4}$$

where $\text{t}(\cdot, \cdot, \cdot)$ denotes the multivariate-t distribution. This distribution has heavier tails than its Gaussian counterpart; indeed, it can be derived by a mixture of multivariate Gaussians with a χ^2-distributed scaling of the covariance matrix.

Recurrent model of discrete state transition probabilities. Finally, we consider a third extension that allows for more complex transition probabilities. We replace (1) with,

$$z_{t+1}^{(w)} \mid z_t^{(w)}, x_t^{(w)} \sim \text{softmax}(r_{z_t^{(w)}} + R x_t^{(w)}), \tag{5}$$

where $r_k \in \mathbb{R}^K$ and $R \in \mathbb{R}^{K \times D}$ parameterize a map from previous discrete and continuous states to a distribution over next discrete states. The link function is the softmax function, which exponentiates and normalizes the K-dimensional argument $r_{z_t^{(w)}} + R x_t^{(w)}$.

4 Model Fitting

We have proposed a variety of Bayesian algorithms for learning the model parameters θ and performing joint inference of the latent states z and x [Linderman et al., 2017, Linderman and Johnson, 2017], but for simplicity, here we treat inference of x and z separately. That is, first we find low-dimensional continuous states to model the observations y, then we find a set of discrete states z that best explain the dynamics in x. We learn the corresponding parameters at both stages.

We will present full details of these inference procedures in the appendix, but for now, here are a few highlights.

To be more specific, first we fit linear dynamical systems to infer the continuous latent states x given the observed values of y. Since many entries of y are masked off, this requires inference with missing data. Under the assumption of diagonal observation covariance, this inference is straightforward. We will compare the standard LDS with the hierarchical LDS, which has worm- and neuron-specific variances.

Once we have inferred the continuous latent states, we will fit an autoregressive hidden Markov model (AR-HMM) to x in order to infer z and learn the dynamics parameters (A_k, b_k, Q_k). Here, again, we will consider all possible variants of hierarchical, robust, and recurrent models.

Both fitting procedures use Markov chain Monte Carlo, specifically block Gibbs sampling, to sample from the conditional distribution of latent states given parameters and of parameters given latent states. We run these algorithms for 1000 iterations, at which point we find them to have converged on the basis of held-out log likelihood.

5 Results

For this draft, I will present results one at a time, each on a separate page. Once we have settled on the story, we can add more exposition.

5.1 Neural activity is low-dimensional and has worm-specific variance

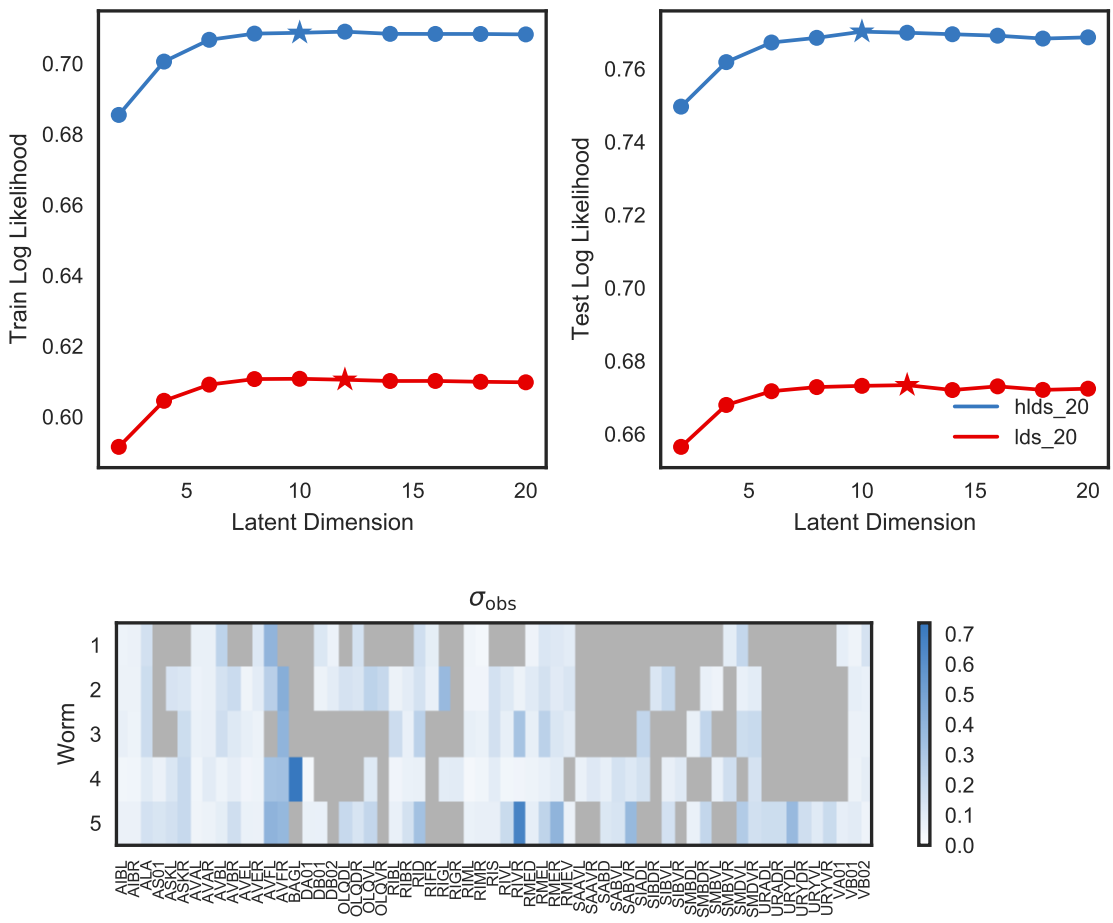

Figure 2: (top) Training and test likelihoods for the linear dynamical system and its hierarchical extension as a function of the dimensionality of the latent state, D (units: nats/observation over Gaussian baseline). On the basis of held-out test likelihood, we choose the hierarchical LDS with 10-dimensional latent states. *(bottom)* Inferred standard deviation $\sqrt{s_n^{(w)}}$ for each neuron and worm. Gray cells indicate neurons that were not observed in that worm.

We fit linear dynamical systems (LDS) and their hierarchical extensions with worm-specific variance (see Section 3.2) for a range of latent dimensions, D. We found that the 10-dimensional hierarchical LDS yielded the highest test likelihood (holding out the final 20% of time frames for each worm for testing, assuming stationarity). This is shown in Figure 2(top).

The hierarchical model has worm-specific observation variances. These are shown in Figure 2(bottom). For neurons that were not observed in a given worm, the corresponding cells are colored gray.

5.2 The hierarchical LDS smooths the data and predicts unobserved activity

Figures 3-7 show the observed activity y along with the smoothed activity under the hierarchical LDS. Formally, the blue lines are given by $C\mathbb{E}[x \mid y] + d$. Unobserved neurons are shown as dotted black lines. Though their activity is not observed, the hierarchical model can predict it based on correlations observed in other worms.

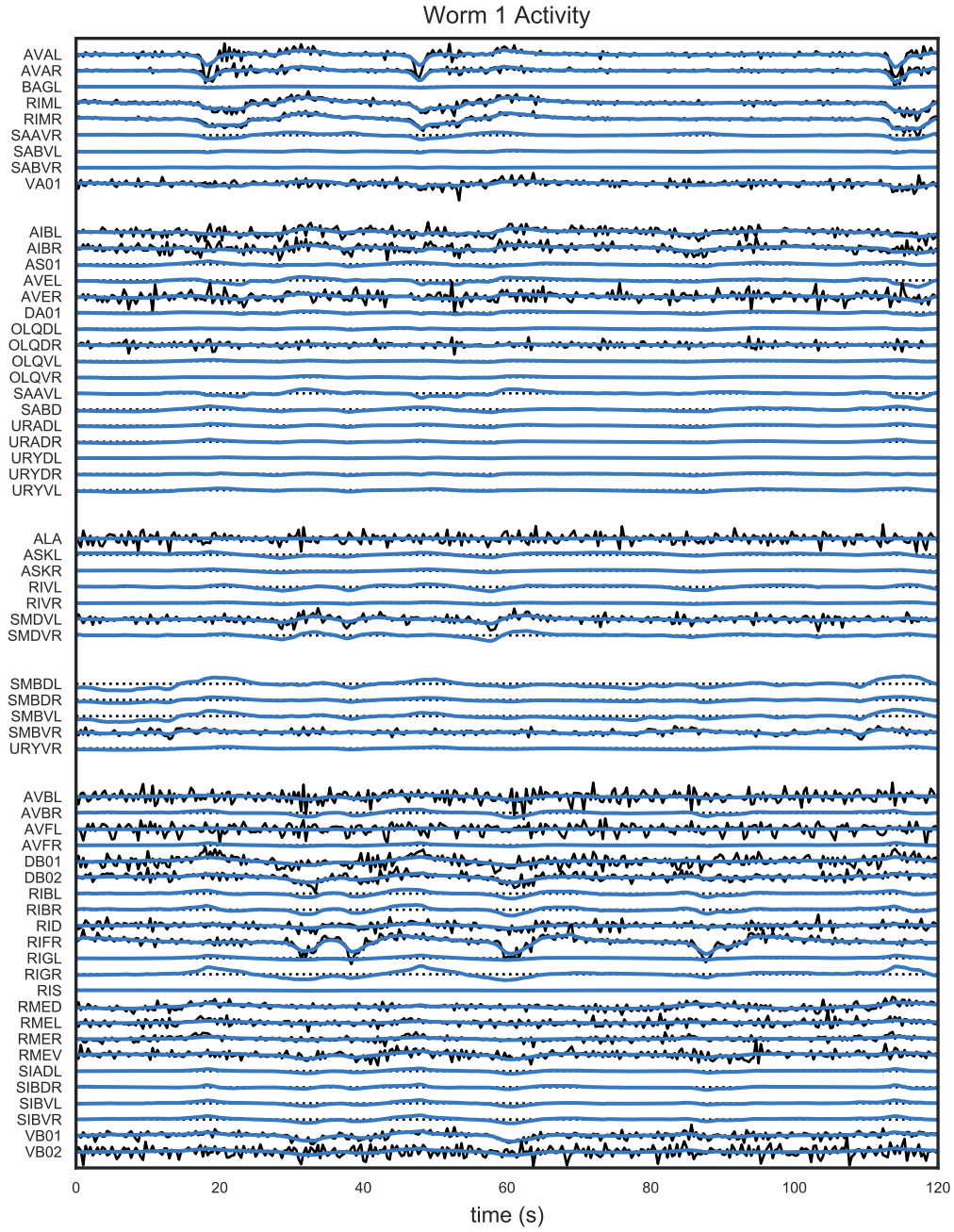

Figure 3: The observed activity y (i.e. the first-order differences of the calcium signal) are shown in black; the smoothed activity with the hierarchical LDS is shown in blue. The hierarchical LDS makes predictions about unobserved neurons—those with dotted black lines—based on the correlations found in other worms. We show two minutes of data here. The neurons have been grouped into five clusters, as discussed below. Within each cluster, the neurons are sorted alphabetically. Each neuron is individually scaled by the standard deviation of its observed activity to illustrate dynamics of low-amplitude neurons.

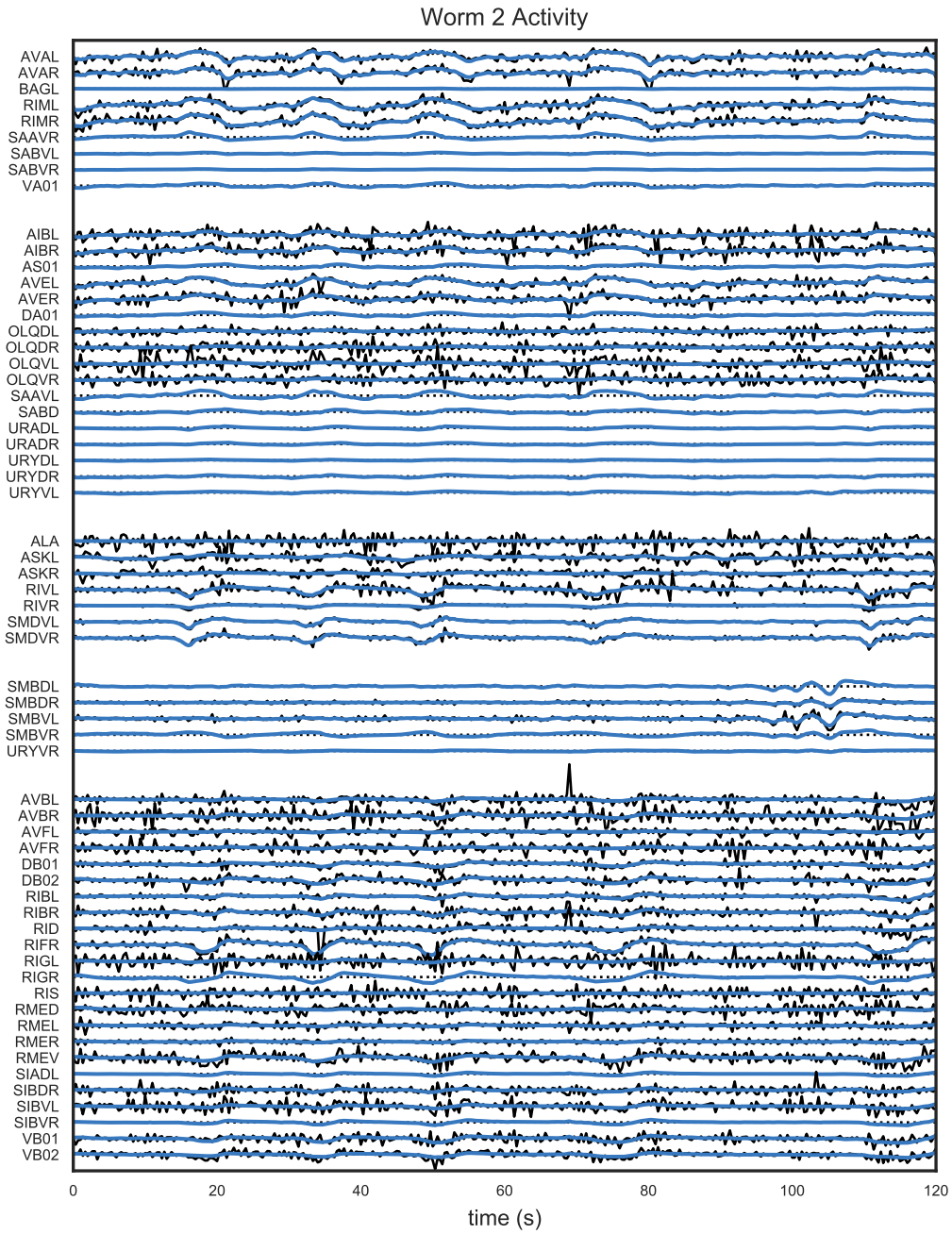

Figure 4: See caption of Figure 3

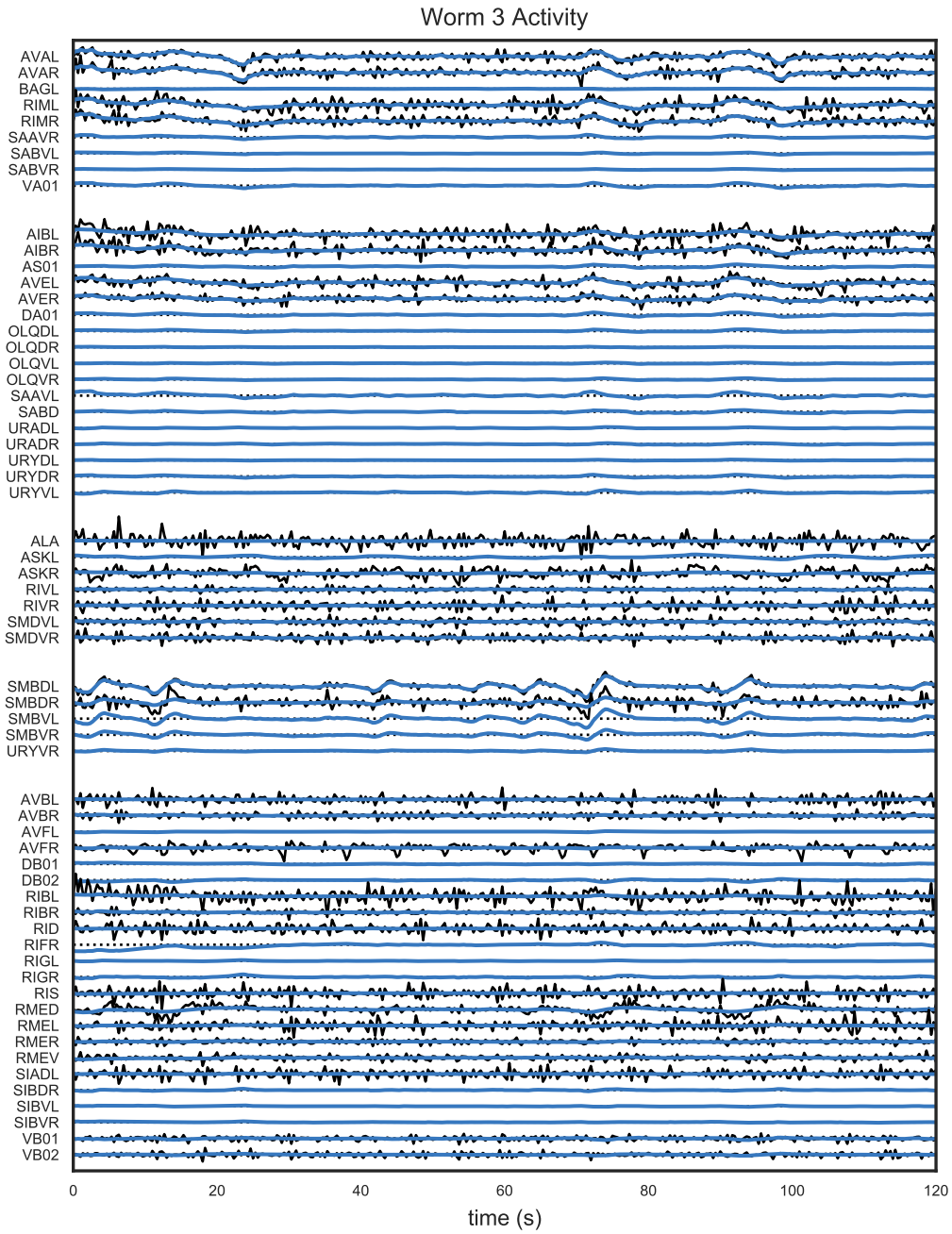

Figure 5: See caption of Figure 3

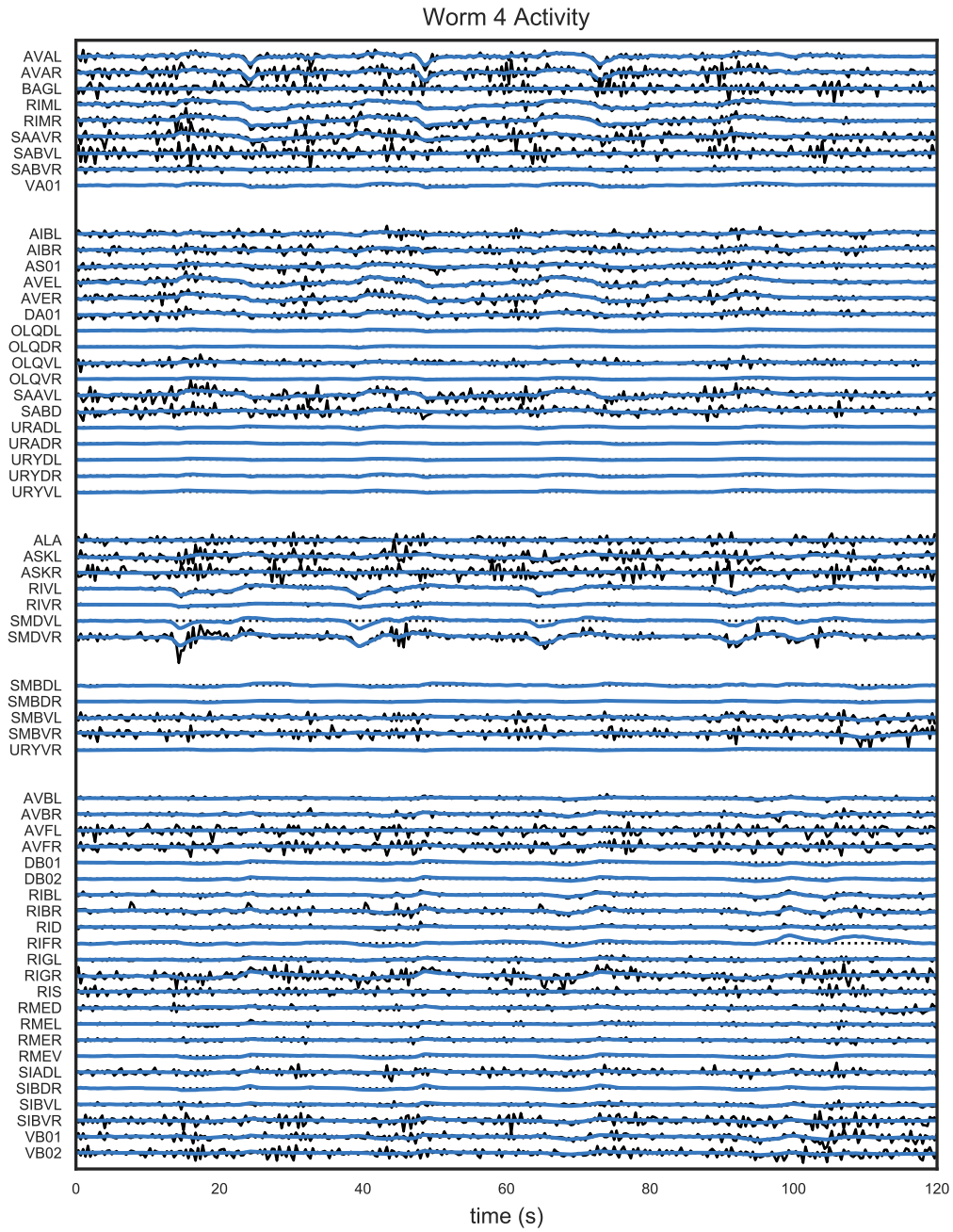

Figure 6: See caption of Figure 3

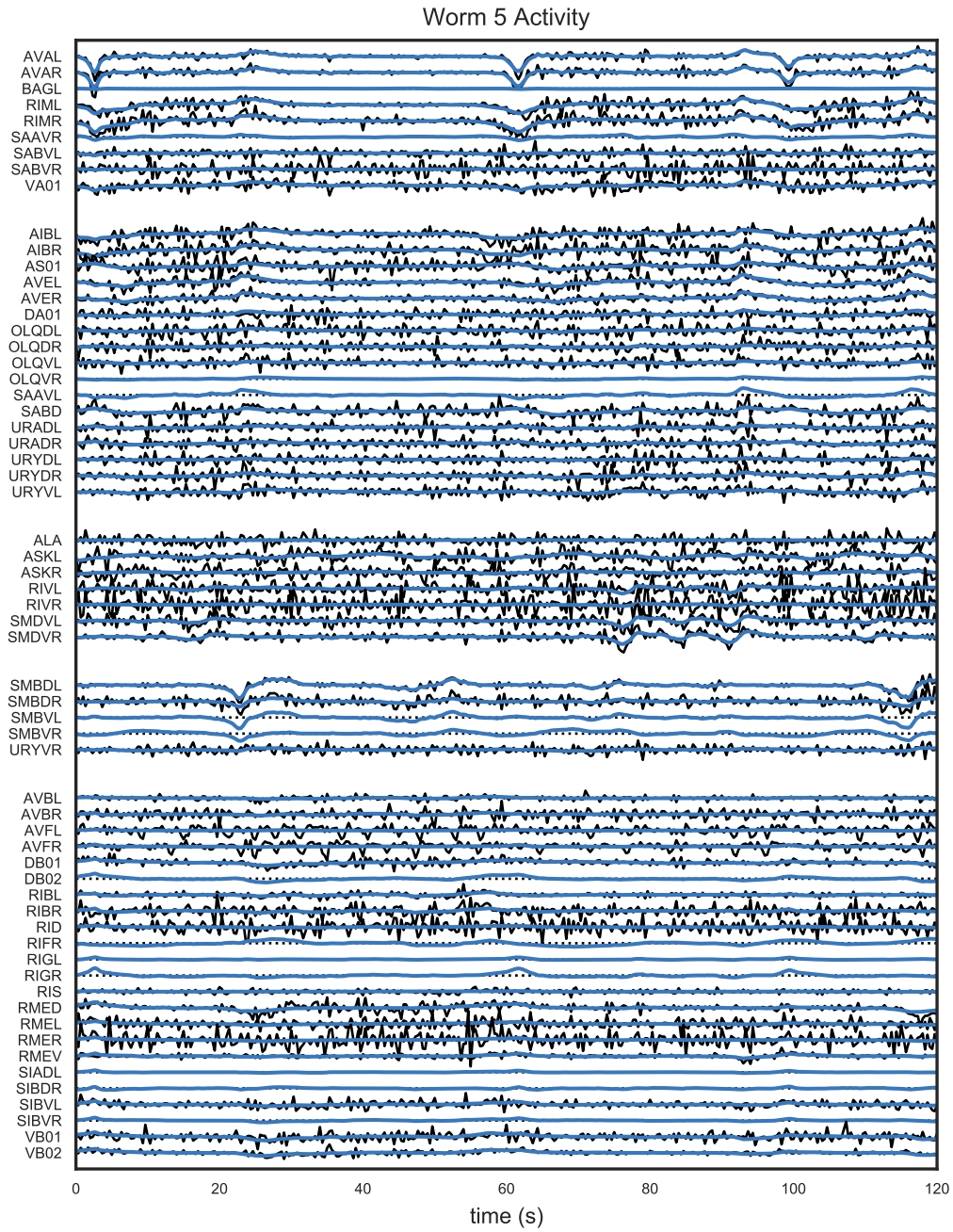

Figure 7: See caption of Figure 3

5.3 The emission parameters aid in neural identification

One of the challenges in modeling this data is that most neurons cannot be labeled with certainty. In other words, we can detect a neuron in the calcium imaging data, but we cannot determine its identity (AVAL, AVAR, etc.). The emission parameters of the SLDS are a useful aid for determining these identities, as we show here.

Across the dataset of five worms, 61 unique neurons were identified in at least one worm. In any given worm, we also observe many other neurons that we cannot accurately label. Our goal here is to assign labels to the unlabeled neurons. Of course, if a label, say AVAL, has already been identified in worm 1, then we cannot assign an unlabeled neuron to AVAL. So the problem is to find the most likely matching of unlabeled neurons to remaining, available labels. Since there are more unlabeled neurons than possible labels, our result will only be a partial matching.

The rows of the emission matrix c_n serve as "embeddings" of the neurons. These embeddings can aid in this matching. Specifically, given an activity trace $y_{n^*}^{(w)} \in \mathbb{R}^T$ for an unlabeled neuron n^* in worm w, along with the set of continuous latent states $x^{(w)} \in \mathbb{R}^{T \times D}$ inferred from the labeled neurons, we perform a linear regression to estimate $c_{n^*} \in \mathbb{R}^D$, the "embedding" of this unlabeled neuron. We then compare this vector c_{n^*} to the embeddings of the labeled neurons, $\{c_n\}_{n=1}^N$, which were learned with the SLDS. We use the cosine similarity between two vectors as a measure of likeness.

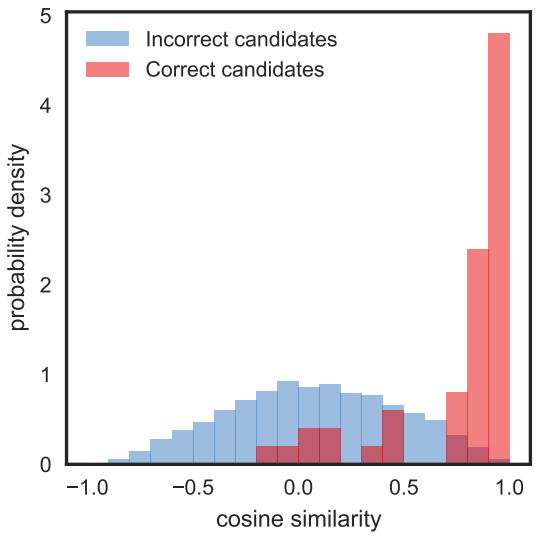

Figure 8: The emission parameters aid in neural identification. In this task, we view unlabeled neurons as *candidates* for a particular label. For example, suppose we are trying to determine the candidate neuron that is most likely to be AVAL. For this task, we have artificially held-out this label so that we know which candidates truly is AVAL. We use the neural embeddings c_{AVAL}, which was inferred from other worms, to aid in this matching. We measure the cosine similarity between c_{AVAL} and the coefficients of linear regressions between x and the activity of the unlabeled neurons' activity. This histogram shows the results of performing this experiment with 10 held-out identities for each of the five worms. We see that the correct candidates frequently have much higher cosine similarity than incorrect candidates. This means that we can use this similarity measure to infer neuron identities.

We demonstrate this identification in a synthetic task where we artificially withheld the labels of ten neurons per worm. Suppose one of the held-out neurons in worm 1 was known to be AVAL. We iterate through all the unlabeled neurons in worm 1—*candidates* for the label AVAL—and measure the cosine similarity between their embeddings c_{n^*} and the embedding c_{AVAL}, which was inferred from worms 2-5.

Figure 8 shows the results of this task. The embeddings of the held-out neurons were much more similar to the embedding of their true identity than were the other unlabeled neurons.

To further quantify these results, we sort these similarities and find the index of the true held-out neuron in the sorted list. If the held-out AVAL neuron was the most similar to c_{AVAL}, its rank would be 1; if it was the least similar, its rank would be $N_{\mathsf{unlabeled}}$, the number of unlabeled neurons for this worm. Table 1 shows the ranks of the similarities of held-out neurons to their true identities, compared to all other unlabeled neurons. If the held-out neurons were always the most similar to their true embeddings, the matching columns would all be 1's, and we would easily be able to identify neurons with this information. We see that in most cases the held-out neurons are within the top three most similar candidates, demonstrating that embeddings are useful for identification.

Finally, we perform an even harder task. Using the Hungarian algorithm, we find the partial matching that maximizes the sum of similarities. We compute the accuracy of this matching by counting the fraction of held-out neurons that were assigned to their true label. (Since the Hungarian algorithm is assigning all the neurons at once, in general the number of correctly assigned neurons is not equal to the sum of 1's in the rank columns.) Still, we see that in this challenging task we assign many neurons to their true identities.

Table 1: Artificial neuron identification task results. In each worm we have N_{labeled} labeled neurons and $N_{\mathsf{unlabeled}}$ unlabeled neurons. Of the unlabeled neurons, 10 had known labels that were held out for this task. We compare the cosine similarity of the unlabeled neuron embeddings to the embeddings of the labeled neurons, sort the similarities, and compute the rank of each held-out neuron to its true label (most similar = 1; least similar = $N_{\mathsf{unlabeled}}$). We use the Hungarian algorithm to find the most likely matching of unlabeled neurons to possible labels, and we report the fraction of the 10 held-out neurons that were correctly matched ("Acc.").

	N_{labeled}	$N_{\mathsf{unlabeled}}$	Matching Rank (10 Held-out Neurons)										Acc.
Worm 1	14	95	10	25	1	1	66	2	1	1	2	1	0.30
Worm 2	31	76	3	1	1	1	1	39	1	1	1	4	0.50
Worm 3	20	111	5	53	3	22	1	2	1	1	1	1	0.50
Worm 4	33	93	2	1	26	2	1	56	1	1	1	1	0.30
Worm 5	38	91	1	1	1	1	1	2	1	26	1	67	0.70

5.4 Neural embeddings reveal clusters of similarly tuned neurons

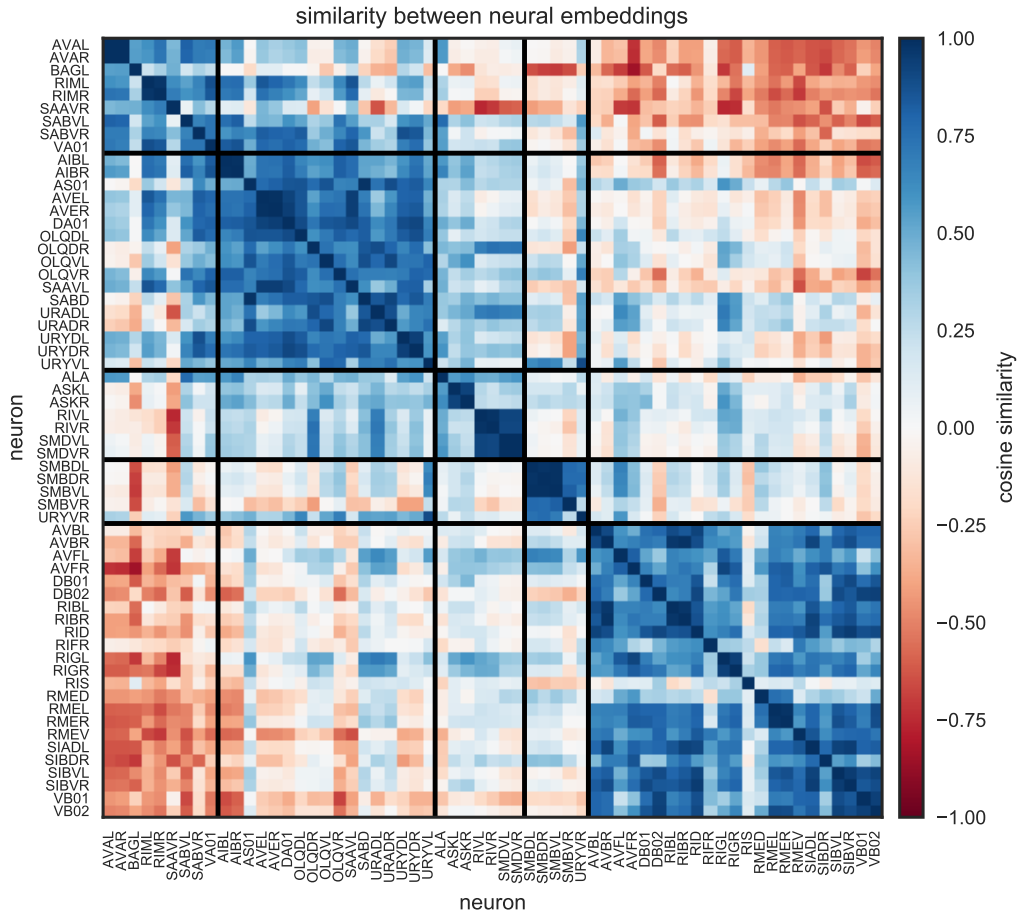

Figure 9: Cosine similarity between embedding vectors c_n suggest clusters of similarly tuned neurons.

The rows of the emission matrix c_n (i.e. the embedding vectors) also suggest that neurons cluster into groups of similarly tuned cells. Figure 9 shows the $N \times N$ matrix of cosine similarity for each pair of embedding vectors. We used k-means to find clusters of embedding vectors and have permuted the similarity matrix to visualize these groups. We see that many clusters contain lateral pairs (e.g. AVAL, AVAR in the first cluster). We will use these clusters to better understand the dynamics of the SLDS.

Note that this clustering is not so simple without a hierarchical model. Since we never see all 61 neurons in a single worm, we need some means of combining information across worms in order to cluster their activity. The hierarchical LDS does exactly this. We could consider simpler hierarchical models, e.g. a hierarchical factor analysis model that does not include assumptions about temporal dynamics; such a model would probably yield similar clusters. The key point is that we must combine information across worms.

5.5 Latent dynamics are hierarchical, robust, and recurrent

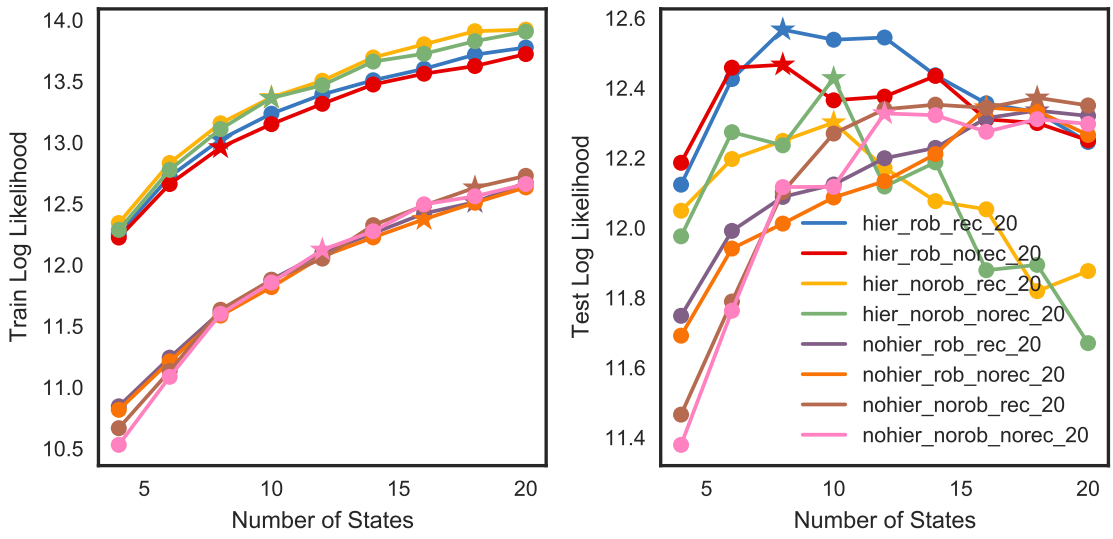

Figure 10: Training and test likelihood of the hierarchical autoregressive hidden Markov models (a special case of the SLDS) for all combinations of hierarchical, robust, and recurrent extensions. We evaluate test likelihoods for a range of numbers of latent states, and we find that the hierarchical, robust, recurrent model with $K = 8$ latent states performs best.

As with the LDS, we fit autoregressive hidden Markov models (AR-HMMs) with all combinations of hierarchical, robust, and recurrent extensions described in Section 3.2. For each possible model, we swept from $K = 2$ to $K = 20$ discrete latent states in increments of two.

Overall, the hierarchical models (red, blue, yellow, green) performed best across all K in training data. This is not surprising as the hierarchical models effectively have $W \cdot K$ dynamics parameters—one version of each state for each worm—whereas the non-hierarchical models have only K. Of these hierarchical models, only those with the robust extension showed good generalization. The non-robust models (with standard Gaussian noise; green, yellow) severely overfit the training data. Of the robust hierarchical models, the recurrent model showed substantial improvement over the non-recurrent model, and obtained peak performance with $K = 8$ states. Likelihoods are measured in units of nats/time step.

5.6 The AR-HMM automatically segments low-dimensional states

Since the continuous latent states are 10 dimensional, we use only the three dimensions of activity for visualization. We choose the three dimensions that are most correlated with the observed neural activity. We color code these three-dimensional trajectories according to the discrete states z, which correspond to different linear dynamical regimes of the continuous states x. Compare these plots to the state space plots of Kato et al. [2015]. These trajectories are not as smooth (which is expected given that we are modeling the first-order differences directly), but they capture the same qualitative patterns. We will quantify this correspondence shortly.

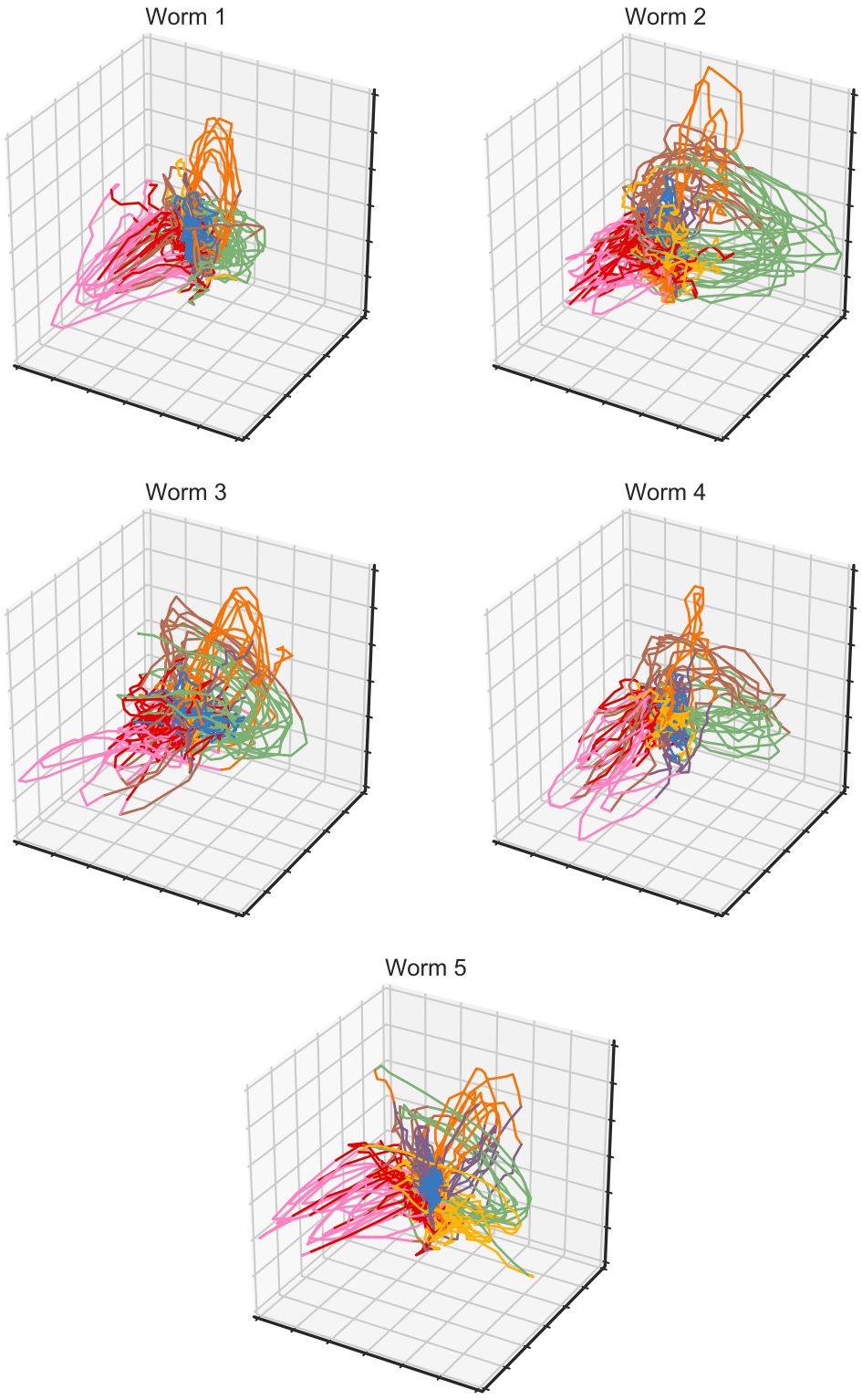

Figure 11: Three dimensional projections of the 10 dimensional latent states x. The trajectories are color coded according to the inferred discrete states z. We find that discrete states correspond to different "loops" in latent space, and that they are, to large extent, used in separate regions of x-space. This is consistent with the recurrent model.

5.7 The AR-HMM learns dynamical systems for each discrete state

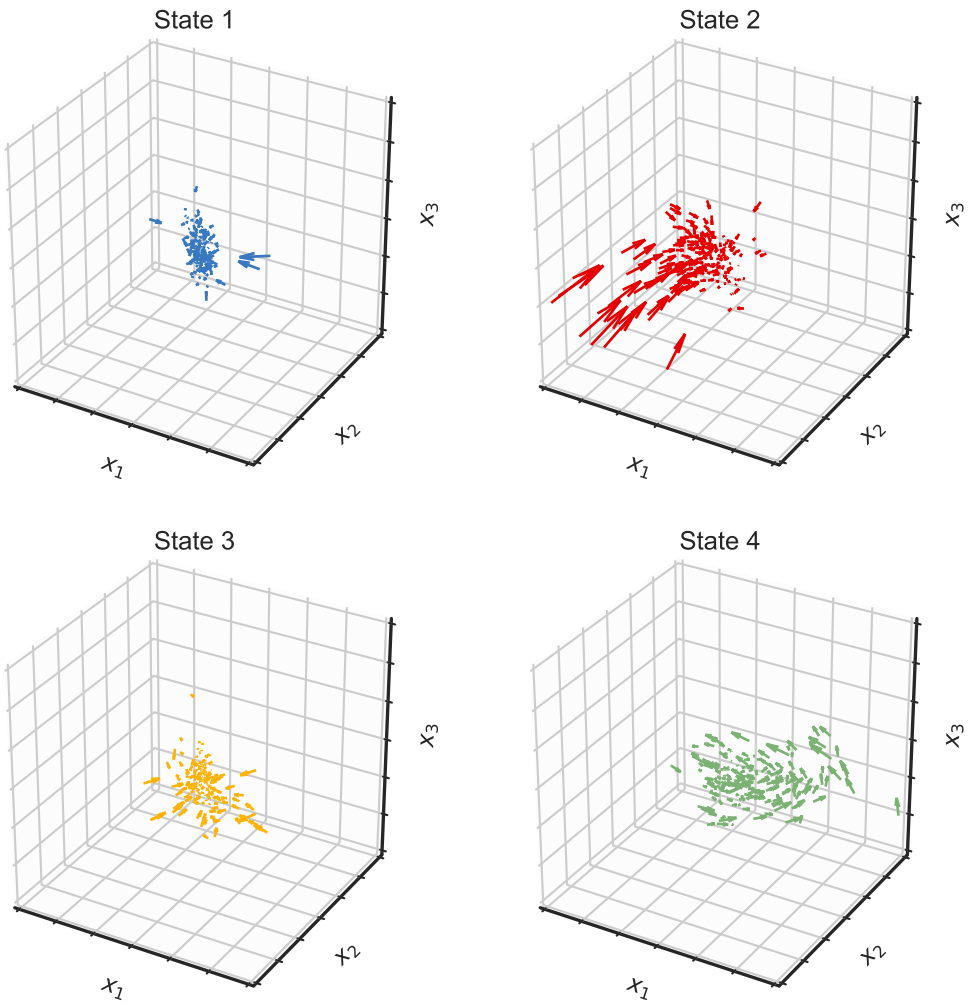

Figure 12: Each discrete state has a corresponding set of linear dynamics parameters (A_k, b_k), which we visualize here as a vector field. Each arrow shows where a point x_t would be mapped to by the linear dynamics of a given state. We visualize points x_t where the states were actually used. We see a number of states related to "returning to the origin" (e.g. states 1, 2, 3, 5), as well as states corresponding to loops (e.g. states 4, 6, 7).

Figures 12 and 13 illustrate the global dynamics parameters for each of the K discrete states. Specifically, we show where a point x_t would be mapped to by the operation $A_k x_t + b_k$. We choose a subset of starting points x_t where the discrete state was used. Many of the states correspond to returning to the origin, which in turn corresponds to constant neural activity (since we are modeling the first-order differences in the calcium signal). Other states correspond to loops, which give rise to increases in the activity of some clusters of neurons and decreases in others.

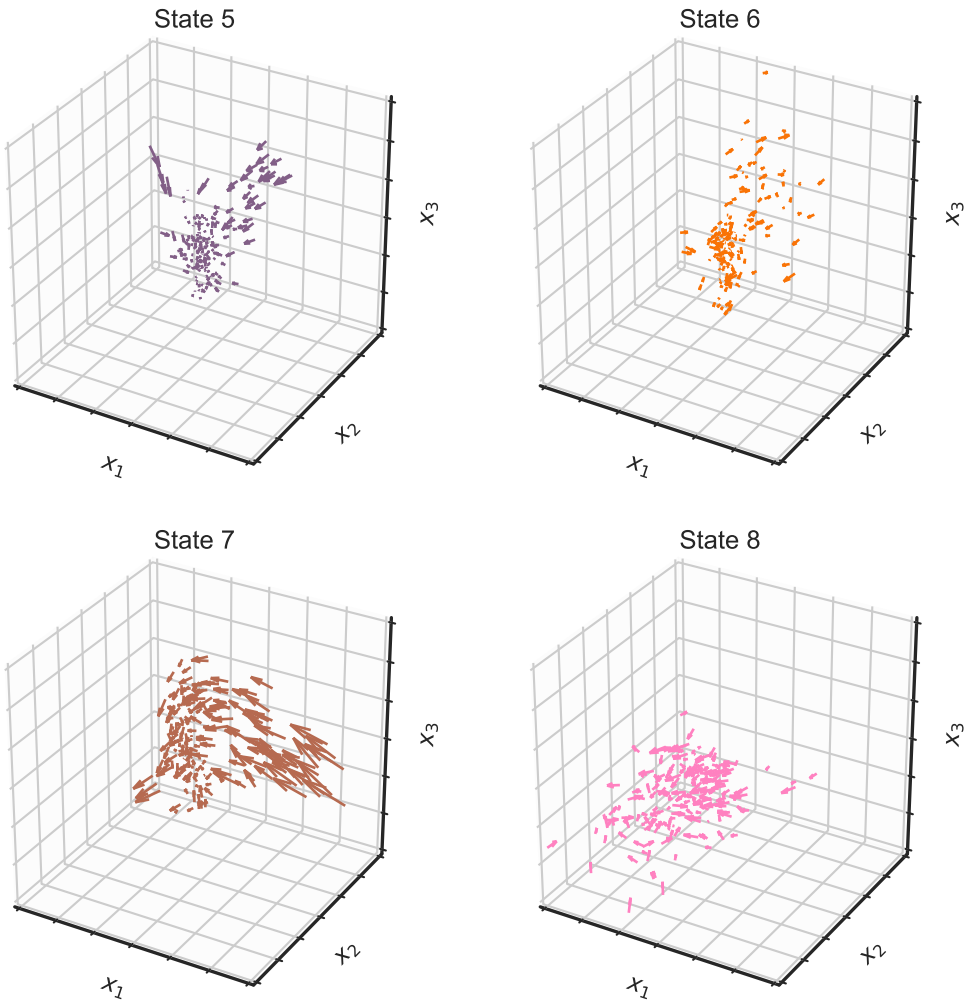

Figure 13: See caption for Figure 12

5.8 Inferred states correspond to activation of clusters of neurons

Figures 14, 15, 16, and 17 show how trajectories in continuous latent state space for each of the K states manifest in the activity of the neural clusters identified in Figure 9. The black lines with circle, triangle, square, and pentagon starting points are forward simulated trajectories using the inferred dynamics of state k with four different initial conditions. The initial conditions are values of x_t where we had inferred an entry into state k. The bottom rows show how the activity the forward simulated trajectories activate each of the five clusters of neurons.

The colored lines in the top panel are segments of x that were inferred to be in state k. Recall that these continuous latent states were found by first fitting a hierarchical LDS. As such, they correspond to the conditional mean of the posterior distribution under the hierarchical LDS, $\mathbb{E}[x \mid y]$.

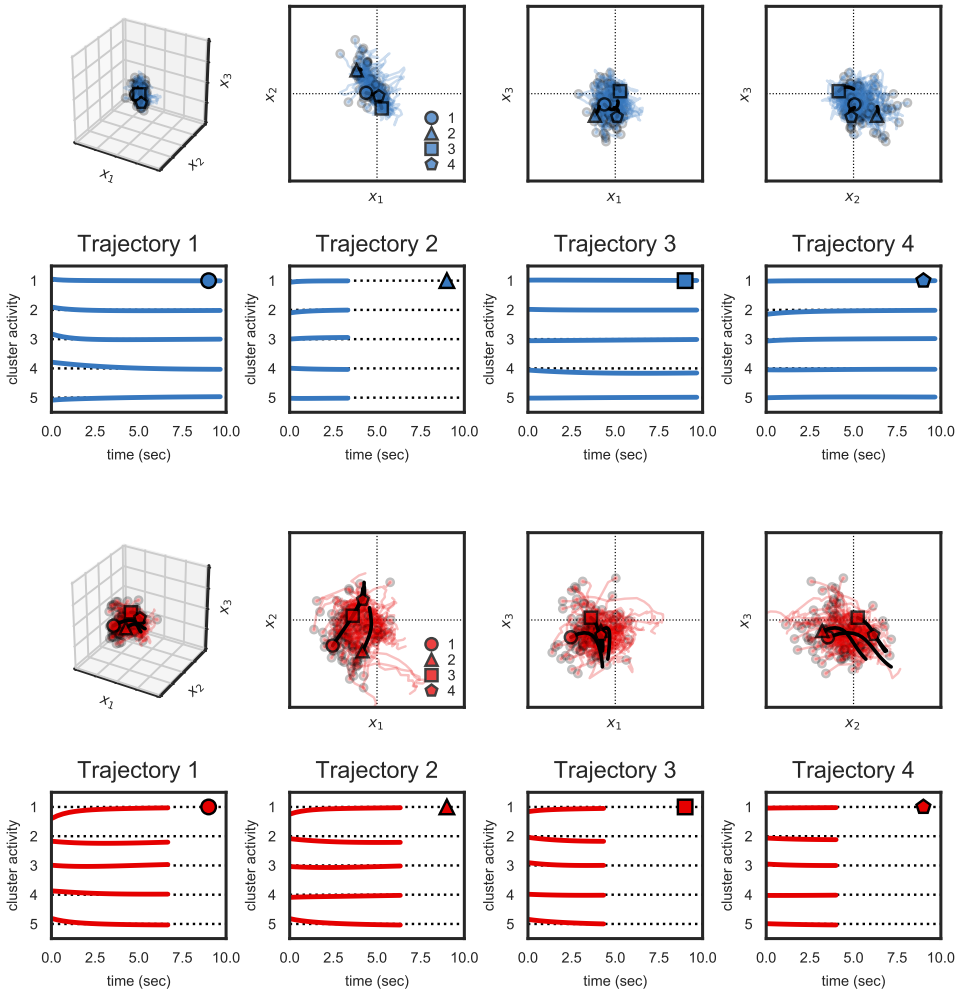

Figure 14: (top) Colored traces show a number of real-data trajectories that were inferred to be in states 1 (blue) and 2 (red). Then we simulate from our model's inferred linear dynamical system for these states, and show simulated trajectories with four different starting points (circle, triangle, square, and pentagon) in black lines. The top left panel shows the top 3 dimensions of these trajectories in latent state space. The top right panels show 2D views of the same trajectories. (bottom) These four simulated trajectories give rise to different patterns of neural activity in the five inferred clusters. For example, the first trajectory of the second state (red circle) corresponds to activity that decays toward the baseline for each of the clusters, with a starting value of below-baseline activity in the first cluster.

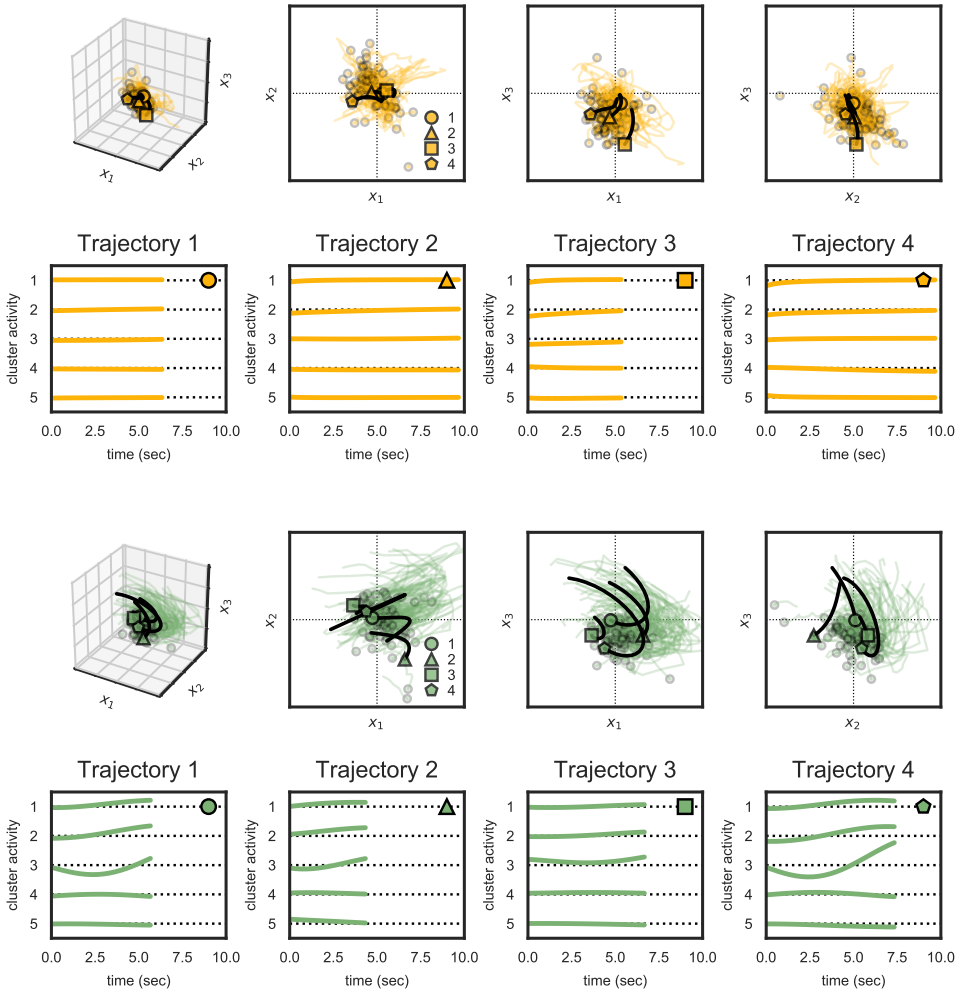

Figure 15: See Figure 14 caption.

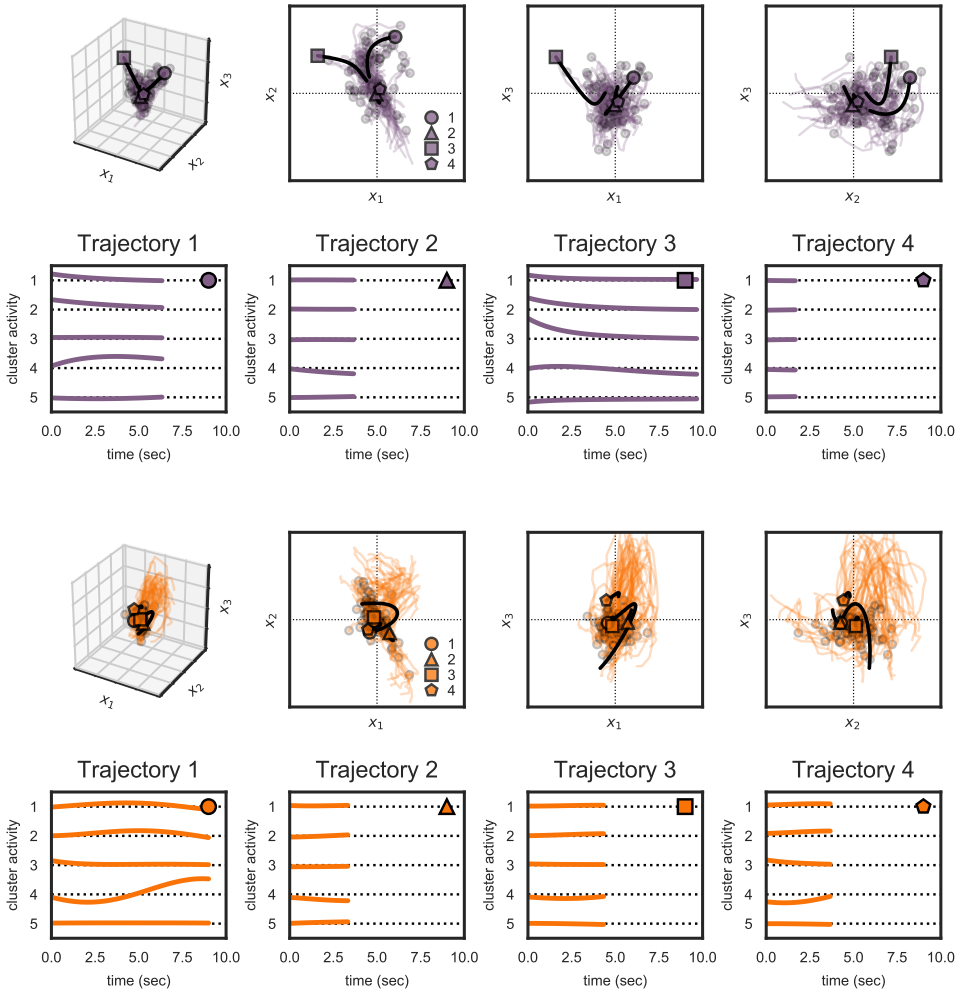

Figure 16: See Figure 14 caption.

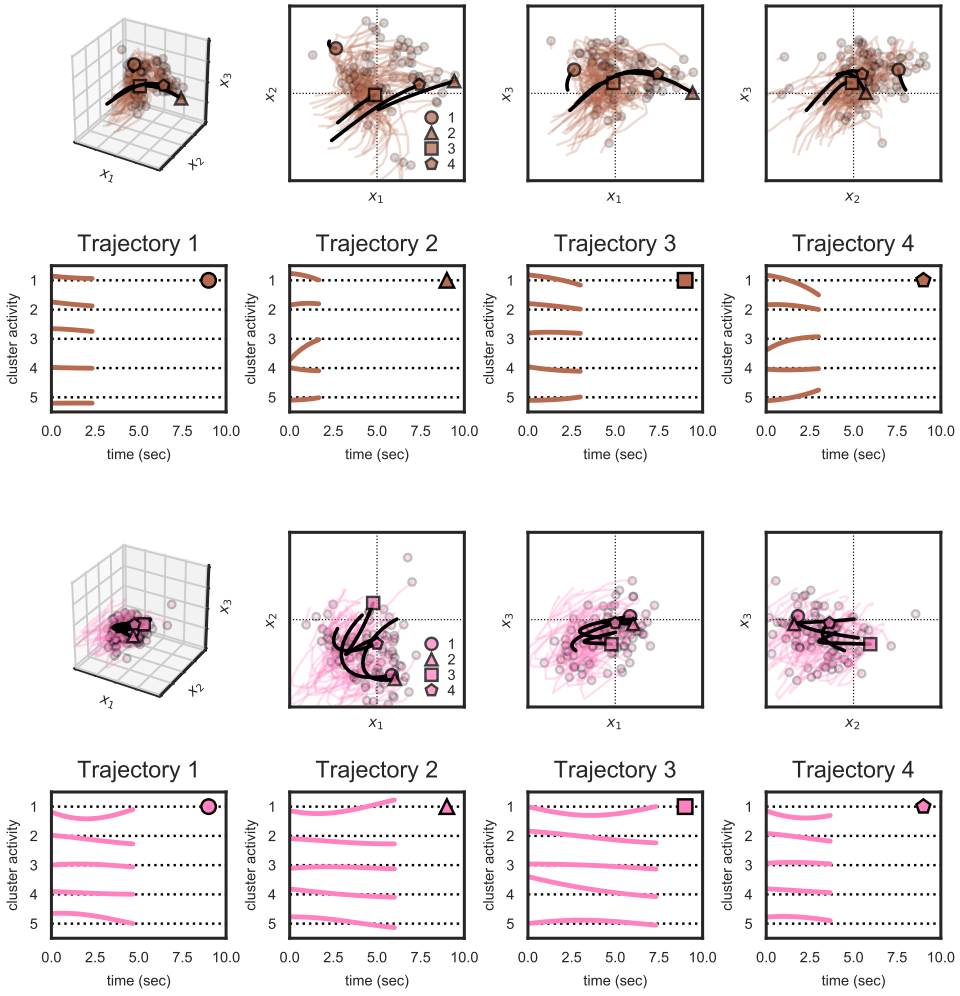

Figure 17: See Figure 14 caption.

5.9 States are employed roughly equally by these worms

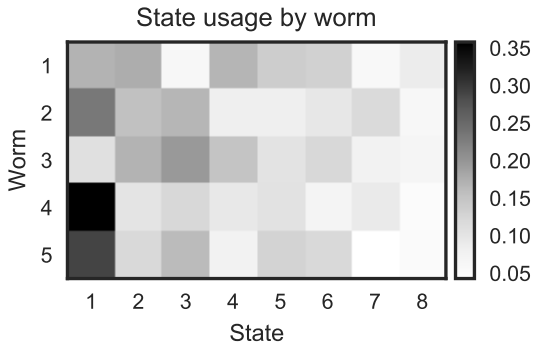

Figure 18: Proportion of time frames spent in each state, broken down per-worm.

The worms appear to use these states with roughly equal proportion, as shown in Figure 18. The states have been ordered to overall population usage, but within worms we see that some states are used slightly more often than others. For example, worm 3 uses state 1 less often than the others, and instead seems to have a higher usage of state 3. This particular difference is probably not important, but it points to how these types of methods could be useful for distinguishing between, for example, different genetic variants of worms.

5.10 Transition structure is largely stereotyped across worms as well

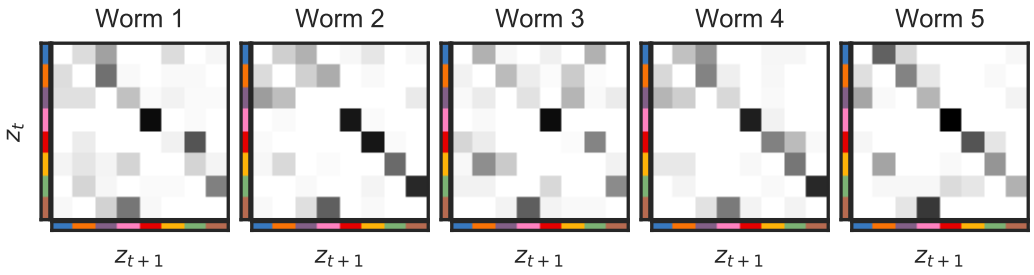

Figure 19: Empirical transition probabilities for each of the worms. By far the most common transition is into the same state. For visualization, we have removed this diagonal component and renormalized so that the k-th row of each transition matrix shows the conditional distribution over z_{t+1} given that $z_t = k$ and $z_{t+1} \neq z_t$.

Figure 19 shows the empirical transition matrices for each of the worms. We see a few transitions appear across all worms:

- Pink transitions to red with high probability in all worms. This is evident from the state trajectories above.
- Brown transitions to pink or, with lower probability, to purple.
- Orange transitions to purple, pink, or blue.
- Purple and orange transition to blue for most worms, except in worm three, which uses yellow in place of blue (see above).
- Green transitions to brown with high probability across all worms.
- For worms 2, 4, and 5, red often transitions into yellow and yellow to green or orange for most worms.
- Blue (the "origin" state), tends to transition to purple or orange in worms 2, 4, and 5.

These aren't meaningful yet, but as we show next, these discrete states closely match the states identified in [Kato et al., 2015].

5.11 Recurrent weights influence transition probabilities

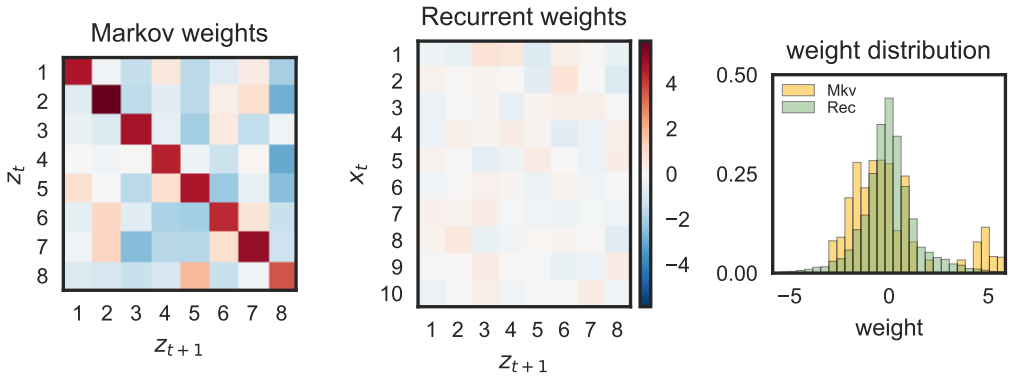

Figure 20: Weights of the recurrent transition model. The largest weights come from the Markov terms for z_t, but the weights for the continuous location x_t are non-trivial. When we compute the effective weights that come from r_{z_t} and the "recurrent" terms $R^\top x_t$, we see that the recurrent terms are a significant fraction of the Markovian terms.

Figure 20 shows the two terms that go into the distribution over next states: the "Markov" term $\{r_k\}_{k=1}^K$ and the "recurrent" weights R, as defined in Eq. (5). Since the impact of the recurrent weights depends on the scale of the continuous latent states, we multiply the weight matrix R by the standard deviation of x (since x is roughly zero mean, this is like z-scoring x so that it is comparable with the Markov term). The largest weights come from the self-transitions, i.e. the diagonal of the Markov terms. Still, the recurrent weights are non-negligible. The histogram of Markov weights (r_{z_t}) and the recurrent weights ($R^\top x_t$) at the times of transitions (right panel) shows that the recurrent weights are comparable to the Markov weights, and therefore must play a role in determining transition probabilities. This must be the case as the recurrent models yield significant improvement in held-out log likelihoods, as shown in Figure 10.

5.12 The recurrent model captures non-Markovian state durations

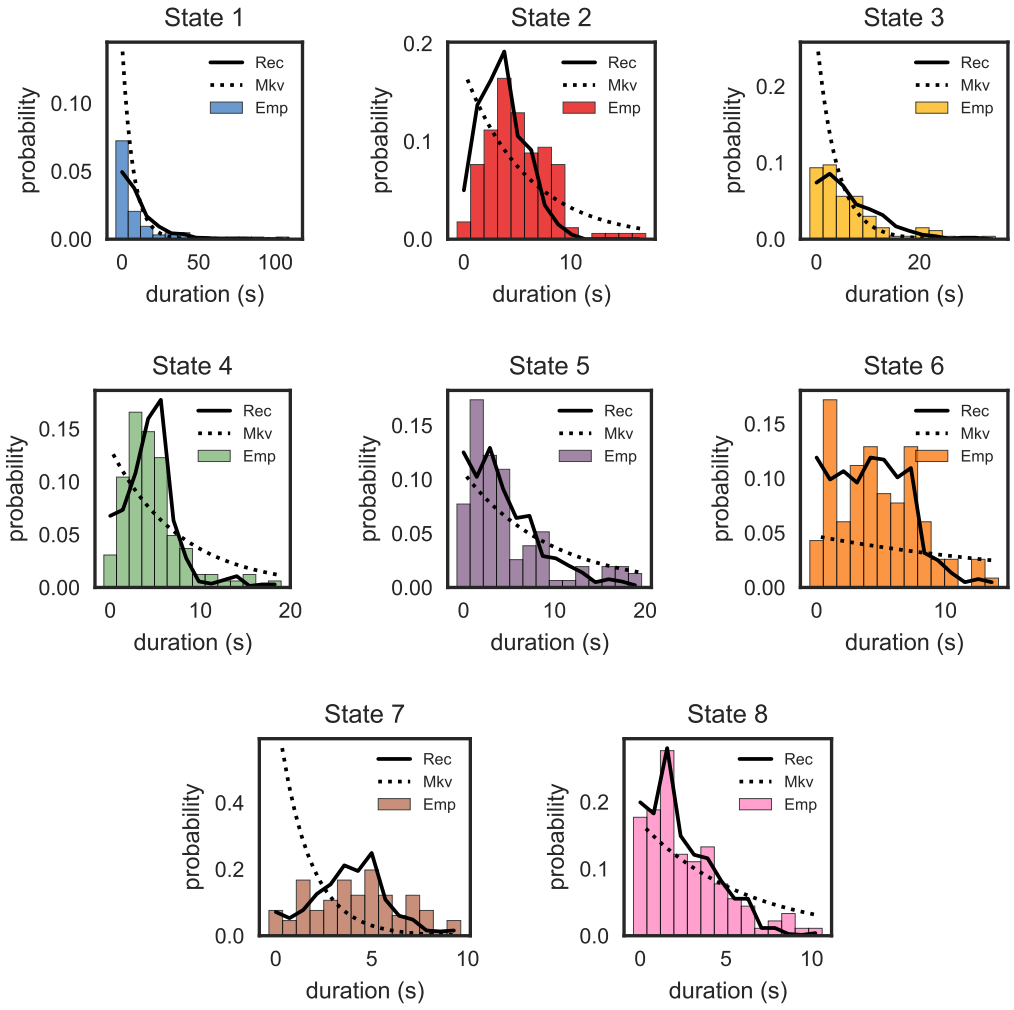

Figure 21: Empirical distribution of state durations (histogram) compared to the geometric distribution implied by the Markovian weights of the transition model. Some state durations are well-approximated by the geometric distribution, but others are peaked away from zero (e.g. states 2, 4, 6, and 7). The recurrent transition model allows the AR-HMM to capture this non-Markovianity by modeling transitions as a function of location in latent state space x_t. Solid lines show the distribution of simulated durations under the recurrent AR-HMM; dashed lines show the geometric duration distribution under the Markovian model.

Figure 21 shows histograms of the empirical distribution of inferred state durations for each state. For comparison, the solid lines show the distribution of *simulated* state distributions from our recurrent AR-HMM. We see a close correspondence.

This correspondence is a direct result of the recurrent nature of the model. Without the recurrent terms, the state duration distributions would necessarily follow the geometric distribution shown in dotted black lines. These distributions decay monotonically and cannot capture the peaks away from zero that we find in the data. This is due to the fact that the distribution over durations d_k of state k under a Markov model

(ignoring the recurrent weights) is given by,

$$p_{k\to k} = [\text{softmax}(r_k)]_k,$$
$$d_k \sim \text{Geom}(1 - p_{k\to k}).$$

While some states (e.g. states 1, 5, and 8) seem to follow the Markov assumption quite well, others (states 2, 4, and especially 7), are far from Markovian. The recurrent model incorporates the continuous latent state x_t into the transition probabilities via the additive term Rx_t in the softmax and thereby captures the non-Markovianity.

5.13 Inferred states overlap with those of Kato et al. [2015]

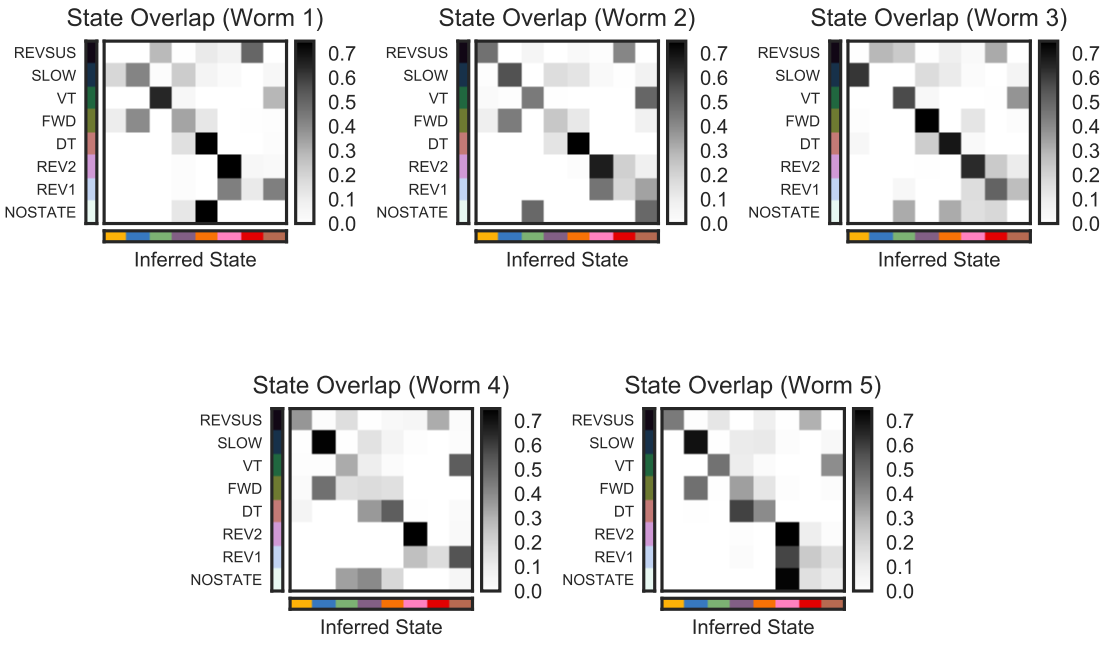

Figure 22: Overlap between the manually labeled states of Kato et al. [2015] and the inferred states of the AR-HMM. Each row is a normalized probability distribution over which of the K discrete states each worm was inferred to be in for the corresponding manual label.

Figure 22 shows a close correspondence between manually labeled states and inferred discrete states, indicating that the switching linear dynamics assumption is a formal specification of the different dynamical modes reported by Kato et al. [2015]. Specifically, different "behaviors" correspond to different linear dynamical regimes in continuous latent space.

A few patterns emerge here:

- Worm 3 differs from the rest, and seems to have a separate matching of inferred states and true states. This list focuses on patterns in Worms 1, 2, 4, and 5.
- REVSUS — the most common manually labeled state — tends to overlap with the red and yellow inferred states.
- SLOW — the second most common — and FWD often map onto the blue state.
- VT (ventral turns) map to the green state, which has a rise and fall of activity in cluster 3 (neurons ALA, ASKL, ASKR, RIVL, RIVR, SMDVL and SMDVR). This seems consistent with the roles of sensory neurons ASK*, interneurons RIV*, and motor neurons SMDV* in local search behavior.
- Skipping ahead, DT (dorsal turns) often map onto the orange and, to a much lesser extent, the purple state. Both states encourage upswings in cluster 4 (SMBDL, SMBDR, SMBVL, SMBVR, and URYVR). The first four of these neurons are motor neurons involved in local search behavior.
- In addition to the blue state, FWD maps onto the green, purple, and orange states as well, though much less significantly.

- REV2 and REV1 map primarily onto the pink state. This state is characterized by an excursion of activity along cluster 1 (AVAL, AVAR, BAGL, RIML, RIMR, SAAVR, SABVL, SABVR, VA01. The AVA* interneurons innervate motor neurons. RIM* are motor neurons that modulate reversals. SABV* are interneurons and sublateral motor neurons. VA01 is a motor neuron.

  We also note that REV1 maps onto the brown state, which is also associated with ventral turns. Judging by the dynamics plots above, it seems ventral turns are often followed by type-1 reversals.

- NOSTATE seems to often map in different ways across worms. This is by far the least common of the manually-labeled states.

Combining these overlap figures with the transition matrices above, I find a slightly different transition structure from what was reported in Kato et al Figure 4E. I find that the transitions between REVSUS and FWD are largely mediated by the SLOW state without an intervening turn. I find that ventral turns are often exercised in between REVSUS and REV1, and that dorsal turns are often used between episodes of forward crawling. Does this make any sense?

5.14 Inferred states change-points correspond with changes in dynamics

Figures 23 through 27 show an interval of latent state trajectories x along with the segmentation from the inferred discrete states z (top). For comparison, we also show the manual segmentation of [Kato et al., 2015] (bottom). The inferred segmentation seems to identify change-points when one or more of the latent continuous states changes dynamics. Indeed, this is the assumption of the model. By contrast, the segmentation of [Kato et al., 2015] is based on a read-out of a single neuron, and while it is largely consistent with the inferred segmentation at the population level, it misses some salient changes in dynamics. For example, see the inferred change-point around time 700 in worm 1. This change-point was not identified in the manual data.

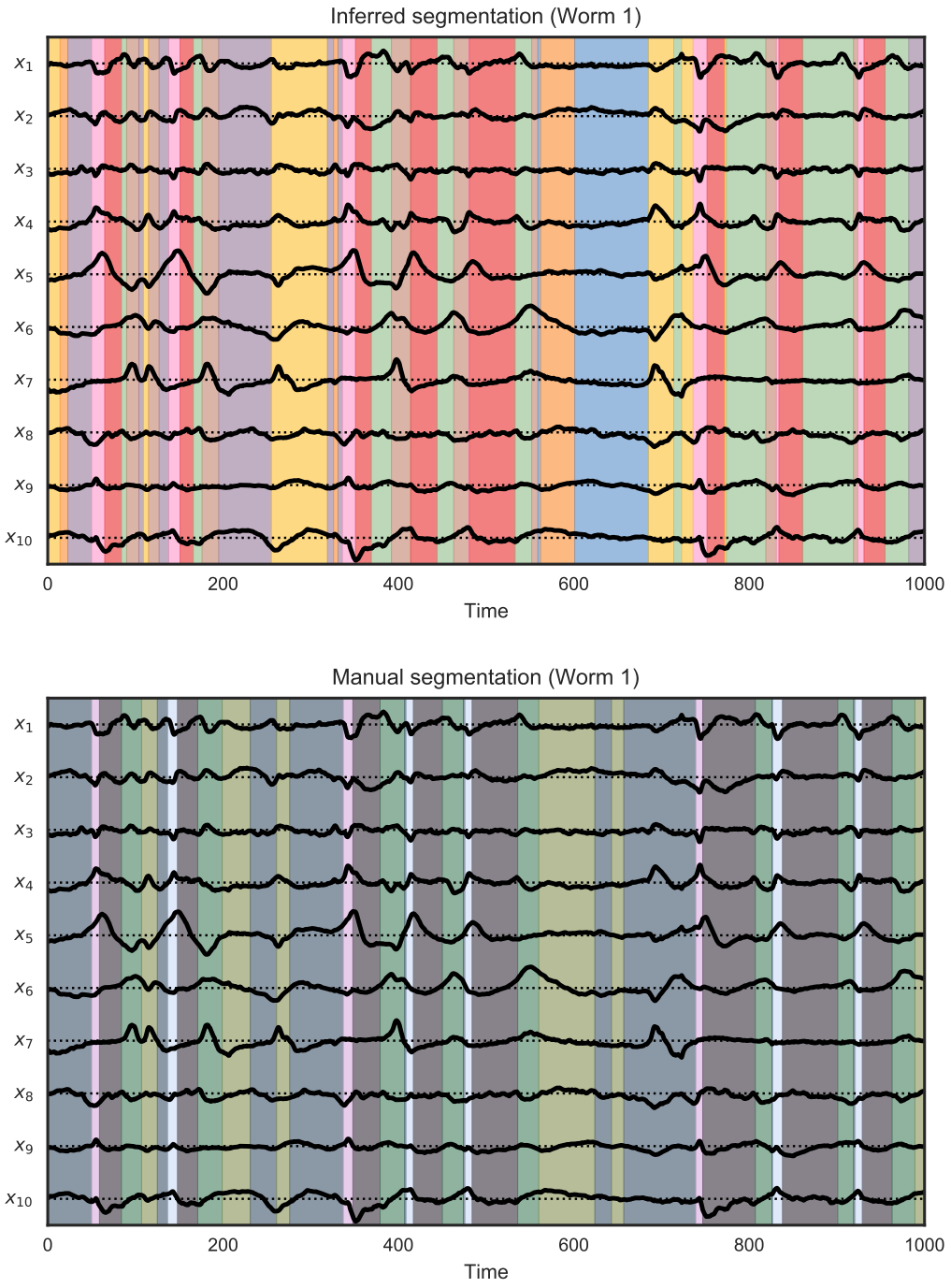

Figure 23: Latent state trajectories superimposed on the inferred segmentation (top) and the manual segmentation (bottom).

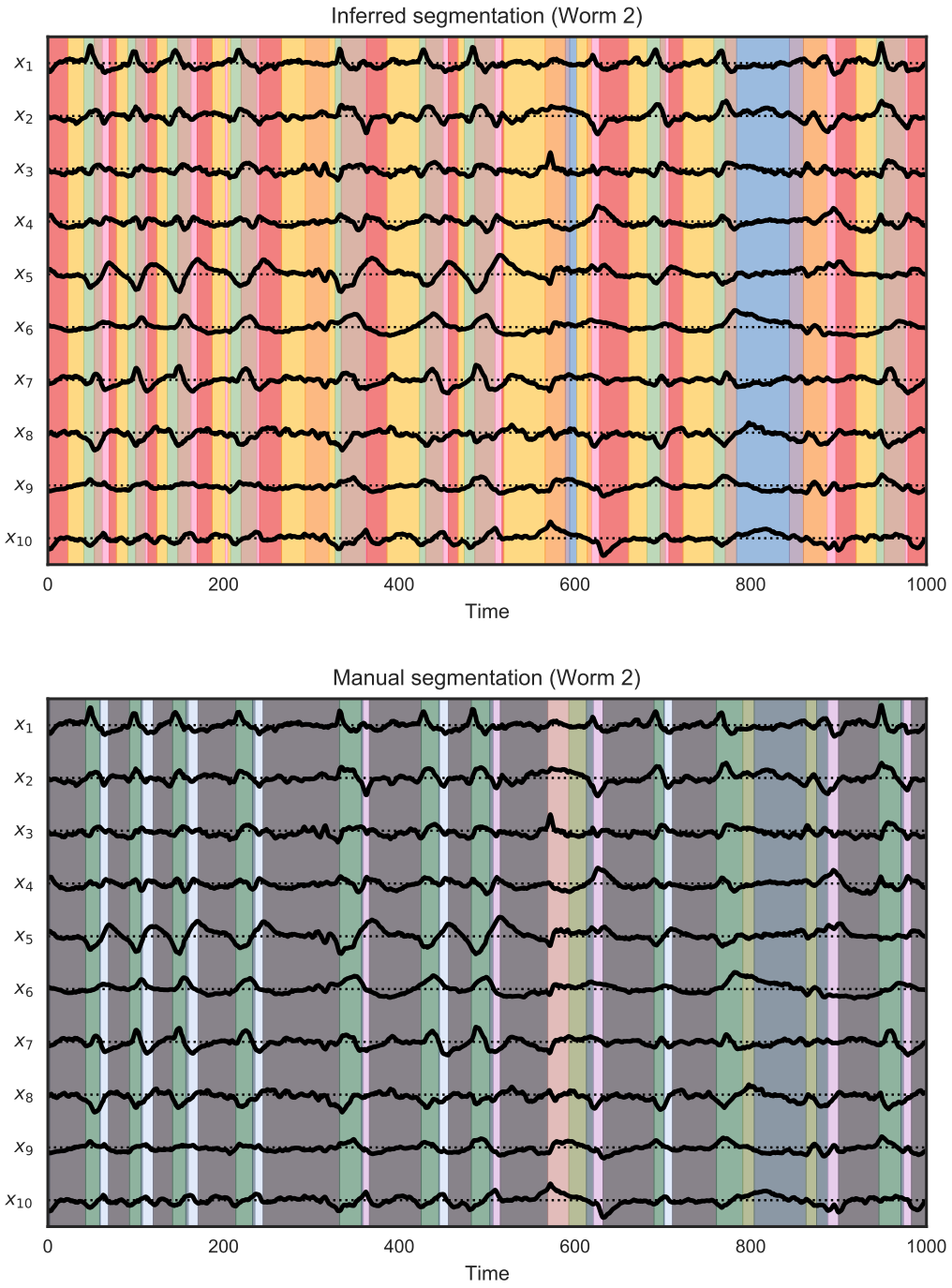

Figure 24: See caption of Figure 23

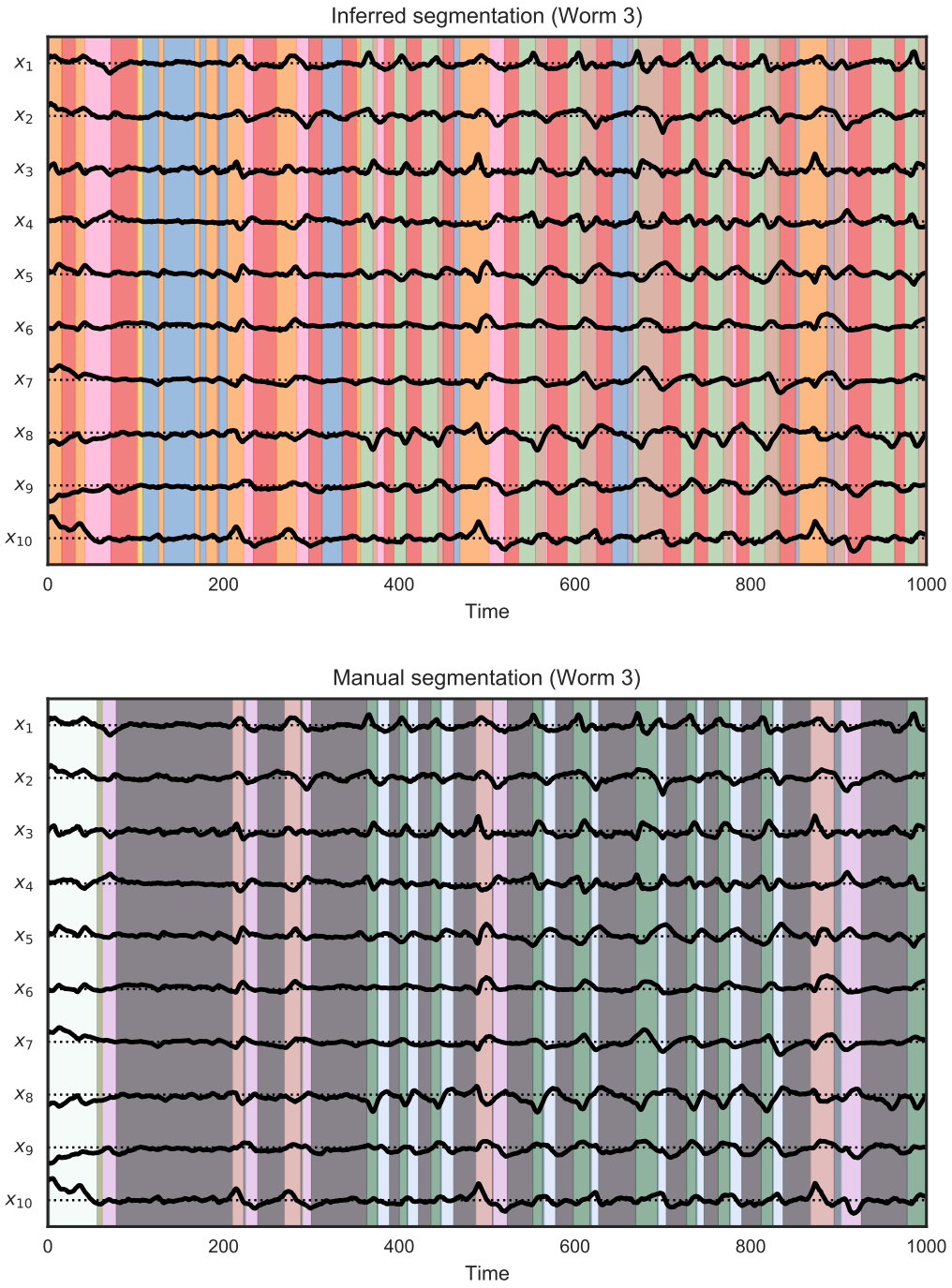

Figure 25: See caption of Figure 23

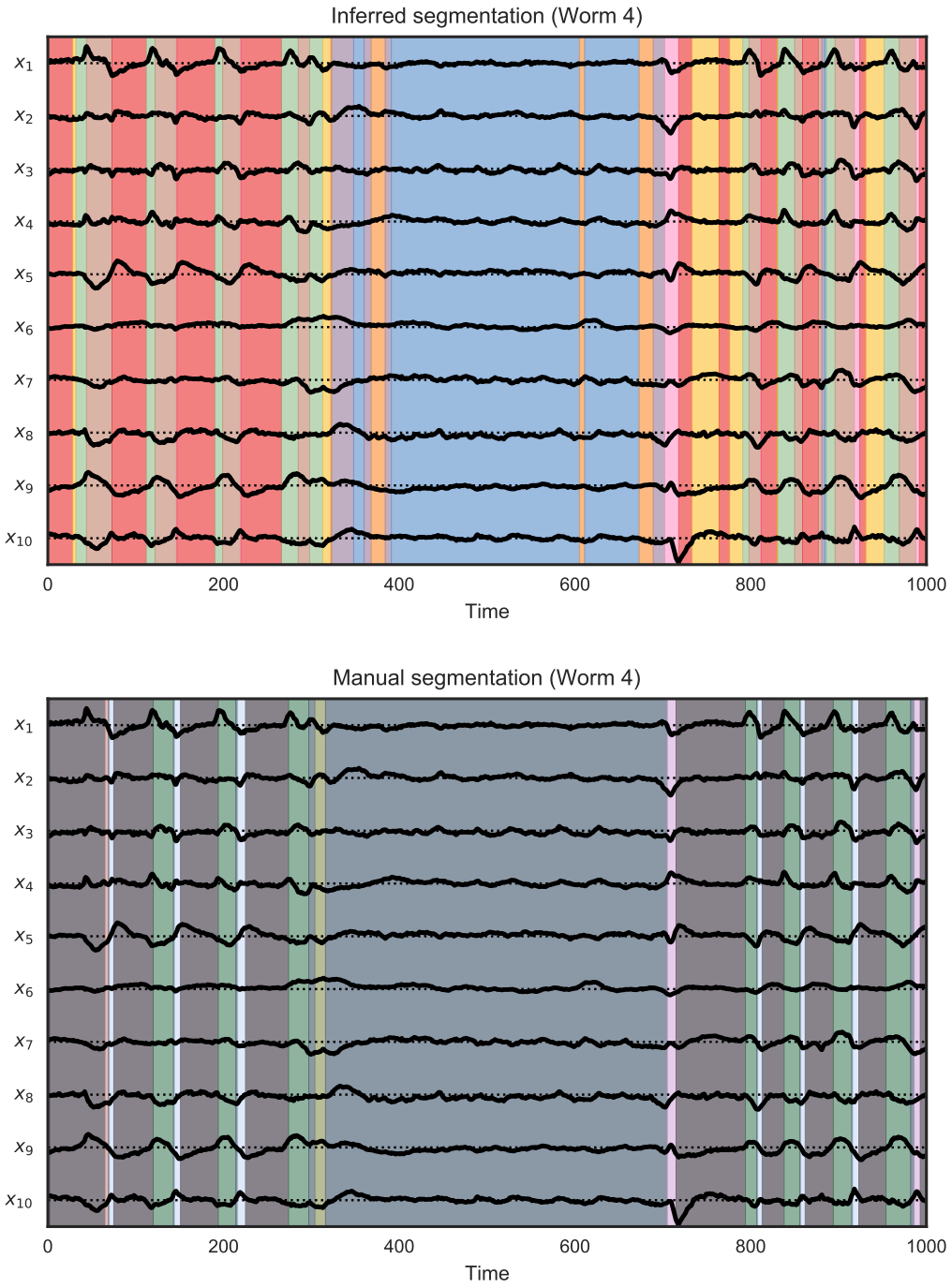

Figure 26: See caption of Figure 23

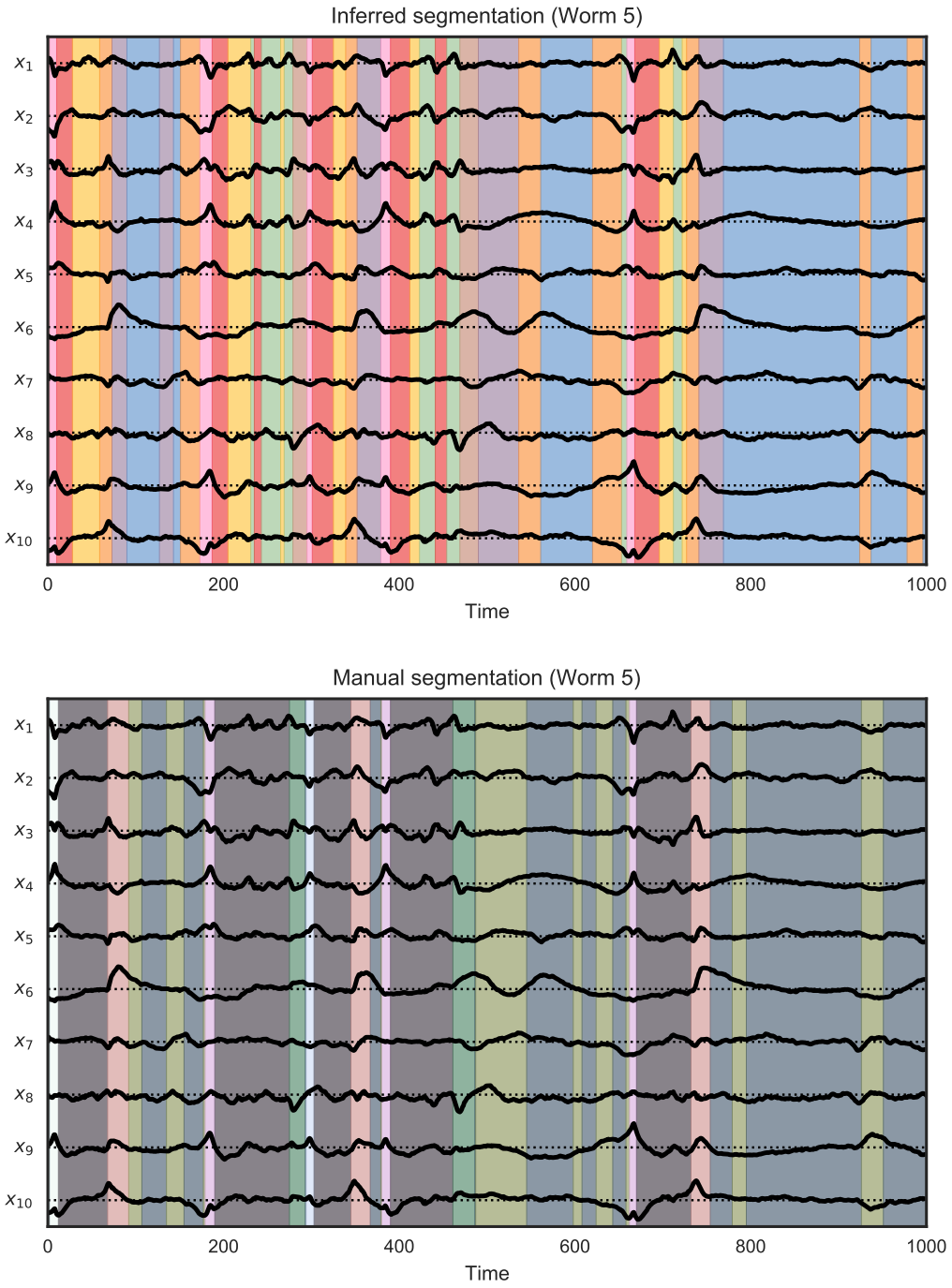

Figure 27: See caption of Figure 23

References

G. A. Ackerson and K.-S. Fu. On state estimation in switching environments. *IEEE Transactions on Automatic Control*, 15(1):10–17, 1970.

C.-B. Chang and M. Athans. State estimation for discrete systems with switching parameters. *IEEE Transactions on Aerospace and Electronic Systems*, (3):418–425, 1978.

Z. Ghahramani and G. E. Hinton. Switching state-space models. Technical report, University of Toronto, 1996.

J. D. Hamilton. Analysis of time series subject to changes in regime. *Journal of econometrics*, 45(1):39–70, 1990.

S. Kato, H. S. Kaplan, T. Schrödel, S. Skora, T. H. Lindsay, E. Yemini, S. Lockery, and M. Zimmer. Global brain dynamics embed the motor command sequence of Caenorhabditis elegans. *Cell*, 163(3):656–669, 2015.

S. W. Linderman and M. J. Johnson. Structure-exploiting variational inference for recurrent switching linear dynamical systems. *IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (To appear)*, 2017.

S. W. Linderman, M. J. Johnson, A. C. Miller, R. P. Adams, D. M. Blei, and L. Paninski. Recurrent switching linear dynamical systems. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2017.

K. P. Murphy. Switching Kalman filters. Technical report, Compaq Cambridge Research, 1998.

J. P. Nguyen, F. B. Shipley, A. N. Linder, G. S. Plummer, M. Liu, S. U. Setru, J. W. Shaevitz, and A. M. Leifer. Whole-brain calcium imaging with cellular resolution in freely behaving Caenorhabditis elegans. *Proceedings of the National Academy of Sciences*, 113(8):E1074–E1081, 2016.

R. Prevedel, Y.-G. Yoon, M. Hoffmann, N. Pak, G. Wetzstein, S. Kato, T. Schrödel, R. Raskar, M. Zimmer, E. S. Boyden, et al. Simultaneous whole-animal 3d imaging of neuronal activity using light-field microscopy. *Nature Methods*, 11(7):727–730, 2014.

T. Schrödel, R. Prevedel, K. Aumayr, M. Zimmer, and A. Vaziri. Brain-wide 3d imaging of neuronal activity in Caenorhabditis elegans with sculpted light. *Nature methods*, 10(10):1013–1020, 2013.