Dynamical Wasserstein Barycenters for Time-series Modeling

Kevin C. Cheng
Tufts University
kevin.cheng@tufts.edu

Shuchin Aeron
Tufts University
shuchin.aeron@tufts.edu

Michael C. Hughes
Tufts University
michael.hughes@tufts.edu

Eric L. Miller
Tufts University
eric.miller@tufts.edu

Abstract

Many time series can be modeled as a sequence of segments representing high-level discrete states, such as running and walking in a human activity application. Flexible models should describe the system state and observations in stationary "pure-state" periods as well as transition periods between adjacent segments, such as a gradual slowdown between running and walking. However, most prior work assumes instantaneous transitions between pure discrete states. We propose a dynamical Wasserstein barycentric (DWB) model that estimates the system state over time as well as the data-generating distributions of pure states in an unsupervised manner. Our model assumes each pure state generates data from a multivariate normal distribution, and characterizes transitions between states via displacement-interpolation specified by the Wasserstein barycenter. The system state is represented by a barycentric weight vector which evolves over time via a random walk on the simplex. Parameter learning leverages the natural Riemannian geometry of Gaussian distributions under the Wasserstein distance, which leads to improved convergence speeds. Experiments on several human activity datasets show that our proposed DWB model accurately learns the generating distribution of pure states while improving state estimation for transition periods compared to the commonly used linear interpolation mixture models.

1 Introduction

We consider the problem of estimating the dynamically evolving state of a system from time-series data.[1] The notion of "state" in such contexts typically is modeled in one of two ways. For many problems, the system state is a vector of continuous quantities (Kalman, 1960; Krishnan et al., 2016), perhaps constrained in some manner. Alternatively, discrete-state models take on one of a countable number of options at each point in time, as exemplified by hidden Markov models (HMMs) (Rabiner, 1989) or "switching-state" extensions (Ghahramani and Hinton, 2000; Linderman et al., 2017).

Many time-series characterization problems of current interest warrant a hybrid of continuous and discrete state representation approaches, where the system gradually transitions in a continuous manner among a finite collection of "pure" discrete states. For example, in human activity recognition using accelerometer sensor data (Bi et al., 2021), some segments of data do correspond to distinct activities (run, sit, walk, etc.), suggesting a discrete state representation. However, when using sensors with high-enough sampling rates, *transition* periods when the system is evolving from one state to another (e.g. the individual accelerates from standing to running over a few seconds) can be well

[1] Code available at https://github.com/kevin-c-cheng/DynamicalWassBarycenters_Gaussian

35th Conference on Neural Information Processing Systems (NeurIPS 2021).

resolved. Gradual evolution between pure states can also be observed in other domains of time series data, such as economics (Chang et al., 2016) or climate science (Chang et al., 2020). Characterizing these systems requires a model with a continuous state space to capture the gradual evolution of the system among the discrete set of pure states.

Motivated by this class of applications, we consider models for time series in which the system's dynamical state is specified by a vector of convex combination weights for mixing a set of data-generating distributions that define the individual pure states. Many approaches, such as mixture models, interpret such a simplex-constrained state vector (Rudin, 1976) as assignment probabilities; that is, the system is assumed to be in a pure state with uncertainty as to which. As a result, the data-generating distribution at moments of transition is a convex, *linear* combination of the pure-state emission distributions. While useful in many applications, such linear interpolation does not capture the gradual transitions among pure states in the time series of interest to us.

To illustrate the shortcomings of linear interpolation, consider the toy data task in Fig. 1, where a system gradually transitions between three pure states over time. During the transition periods (e.g. at times 600 and 1400), the linear interpolation method infers a data-generating distribution that is *multi-modal*, shown in Fig. 1(b). If we refer to our pure states as "walk" and "run," this approach models the walk-to-run transition as sometimes walk and sometimes run. This does not intuitively capture the gradual nature of accelerating from walk to run in our intended applications.

To overcome this limited representation, we consider another way to mix together pure-state distributions: *displacement-interpolation* (McCann, 1997), which is related to the Wasserstein distance (Peyré and Cuturi, 2019), a metric over the space of probability distributions (Sriperumbudur et al., 2010). While the work of McCann (1997) is limited to combining two distributions, it is extended to multiple distributions using the notion of a *Wasserstein barycenter* (Agueh and Carlier, 2011). Fig. 1(c) shows how a Wasserstein barycenter approach to time-series modeling infers data-generating distributions during transitions that are not multi-modal but instead place mass *in between* where the two pure-state distributions do. This intuitively captures *gradual transition* between two pure states.

Inspired by this framework, in this work we develop a dynamical Wasserstein barycentric (DWB) model for time series intended to explain data arising as a system evolves between pure states. Our model uses a barycentric weight vector to represent the system state. Given an observed multivariate time-series and a desired number of states K, all parameters are estimated in an unsupervised way. Estimation simultaneously learns the data-generating distributions of K pure discrete states as well as the K-simplex valued barycentric weight vector state at each timestep.

Given the nature of our model, we require that the state lie in the simplex at every timestep, a constraint not respected by the Gaussian noise that drives common continuous-state processes (Welch, 1997). Building on work by Nguyen and Volkov (2020), we employ a random walk where the driving noise comes from independent, identically distributed (IID) draws from a mixture of two Beta distributions, representing stationary and transitional dynamics. By blending the current state and a mixture-of-Betas draw in a convex manner, we construct a new state that lies in the simplex.

To specify the emission distributions of our model, we assume that each pure state generates data from a multivariate Gaussian. While a Gaussian model may not be suitable in all applications, this choice allows us to exploit useful properties of Gaussian densities under the Wasserstein distance (Takatsu, 2011). Specifically, a closed-form expression exists for the Wasserstein distance between Gaussians, the Wasserstein barycenter among Gaussians can be computed via a simple recursion, and the estimation of the Gaussian mean vectors and covariance matrices can be performed conveniently over a Riemannian product manifold. Empirically, we find our proposed DWB model with Gaussian pure-states performs well on human activity datasets, accurately characterizing both pure-states emission distributions and capturing the system state in pure states and transition periods.

Contributions: We introduce a *displacement-interpolation model for time series* where the data-generating distribution is given by the weighted Wasserstein barycenter of a set of pure-state emission distributions and a time-varying state vector. We propose a *simplex-valued random walk* with flexible dynamical structure to model the system state. We exploit the *Riemannian structure of Gaussian distributions* under the Wasserstein distance for parameter estimation for faster convergence speed. We evaluate on *human activity data* and demonstrate the ability of our method to capture *stationary and transition dynamics*, comparing with the linear interpolation mixture model and with a continuous state space model.

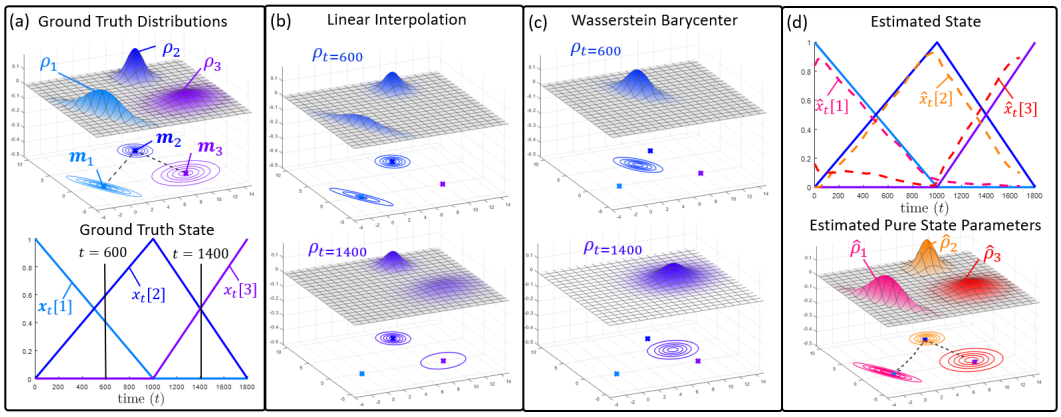

Figure 1: (a) Three Gaussian distributions ρ_1, ρ_2, ρ_3 each representing distinct activities with corresponding means $\boldsymbol{m}_1, \boldsymbol{m}_2, \boldsymbol{m}_3$ that are marked ('x') in all plots as reference points. The time series is drawn from a time-varying distribution according to the ground truth state vector as the system transitions linearly from ρ_1 to $\rho_2, t = 1, ..., 1000$, then continues to $\rho_3, t = 1000, ..., 1800$. (b) Under the linear interpolation state-transition model, the PDF at select times of $t = 600, 1400$ are linear combinations of ρ_1, ρ_2, ρ_3. (c) Alternatively, the proposed displacement-interpolation transition model between pure-states given by the Wasserstein barycenter translates the mass between pure states. (d) Following the Wasserstein barycentric model for time series, our proposed method accurately recovers both the pure state distributions and state vector from the observed time series.

Outline: Sec. 2 provides an overview of the Wasserstein distance, barycenter, and the associated geometry for Gaussian distributions. We then formalize our problem statement and estimation problem in Sec. 3. Sec. 4 discusses the model parameters, covering the dynamical simplex state-space model in Sec. 4.1 and the pure-state parameters in Sec. 4.2. Sec. 5 discusses the optimization of our model parameters leveraging geometric properties of the Wasserstein distance for Gaussians. Finally, Sec. 6 evaluates and discusses the advantages of our model in the context of human activity data.

2 Technical Background

A core component to our approach is to model the intermediate transition states of a time series using the Wasserstein barycenter (Agueh and Carlier, 2011) of probability distributions, which generalizes the displacement-interpolation framework (McCann, 1997) beyond two distributions. We refer the works of (Peyré and Cuturi, 2019) and (Villani, 2009) for a detailed discussion on these concepts.

Consider the space of all Borel probability measures over $\mathbb{R}^d$ with finite second moment. The squared Wasserstein-2 distance for two distributions ρ_1, ρ_2 with squared Euclidean ground cost is defined as,

$$\mathcal{W}_2^2(\rho_1, \rho_2) = \inf_{M \in \Pi(\rho_1, \rho_2)} \int_{\mathbb{R}^d \times \mathbb{R}^d} \|\boldsymbol{\alpha} - \boldsymbol{\beta}\|_2^2 \, M(d\boldsymbol{\alpha}, d\boldsymbol{\beta}), \tag{1}$$

where $\Pi(\rho_1, \rho_2)$ is the set of all joint distributions with marginals ρ_1, ρ_2, and M is the optimal transport plan, the element that minimizes the total transportation cost. When these measures are Gaussian, parameterized by their mean vectors $\boldsymbol{m}_i \in \mathbb{R}^d$, and symmetric positive-definite covariance matrices $\boldsymbol{S}_i \in Sym_+^d$, the squared Wasserstein-2 distance has a closed form solution (Takatsu, 2011),

$$\mathcal{W}_2^2\left(\rho_1(\boldsymbol{m}_1, \boldsymbol{S}_1), \rho_2(\boldsymbol{m}_2, \boldsymbol{S}_2)\right) = \underbrace{\|\boldsymbol{m}_1 - \boldsymbol{m}_2\|_2^2}_{\mathcal{E}^2(\boldsymbol{m}_1, \boldsymbol{m}_2)} + \underbrace{\operatorname{tr}\left(\boldsymbol{S}_1 + \boldsymbol{S}_2 - 2\left(\boldsymbol{S}_1^{\frac{1}{2}} \boldsymbol{S}_2 \boldsymbol{S}_1^{\frac{1}{2}}\right)\right)}_{\mathcal{B}^2(\boldsymbol{S}_1, \boldsymbol{S}_2)}. \tag{2}$$

This distance decomposes into sum of the squared Euclidean distance between mean vectors, $\mathcal{E}^2(\boldsymbol{m}_1, \boldsymbol{m}_2)$, and the squared Bures distance (Bhatia et al., 2017) between covariance matrices, $\mathcal{B}^2(\boldsymbol{S}_1, \boldsymbol{S}_2)$. Thus, the contributions of the mean and covariance to the Wasserstein distance between Gaussians are decoupled, a property that is uncommon for Gaussian distribution distances (Nagino and Shozakai, 2006) and has important implications for optimization and barycenter computation.

The Wasserstein barycenter extends the notion of the weighted average of points in $\mathbb{R}^d$ using the Euclidean distance to the space of probability distributions with the Wasserstein distance. Given a set of K measures and the barycentric coordinate vector on the K-simplex, $\boldsymbol{x} \in \Delta^K$, the Wasserstein

barycenter is the measure that minimizes this weighted Wasserstein distance to the set of measures,

$$\rho_B\left(\boldsymbol{x}, \{\rho_k\}_{k=1}^K\right) = \underset{\rho \in \mathcal{P}_2(\mathbb{R}^d)}{\operatorname{argmin}} \sum_{k=1}^K \boldsymbol{x}[k] \mathcal{W}_2^2(\rho_k, \rho). \tag{3}$$

When ρ_k are Gaussian distributions, ρ_B defined in (3) is itself Gaussian (Agueh and Carlier, 2011) with parameters $\boldsymbol{m}_B, \boldsymbol{S}_B$. Again, because of the decomposition of the Wasserstein distance in (2), the Wasserstein barycentric problem in (3) can be solved separately for its components,

$$\boldsymbol{m}_B = \underset{\boldsymbol{m} \in \mathbb{R}^d}{\operatorname{argmin}} \sum_{k=1}^K \boldsymbol{x}[k] \mathcal{E}^2(\boldsymbol{m}_k, \boldsymbol{m}), \qquad \boldsymbol{S}_B = \underset{\boldsymbol{S} \in Sym_+^d}{\operatorname{argmin}} \sum_{k=1}^K \boldsymbol{x}[k] \mathcal{B}^2(\boldsymbol{S}_k, \boldsymbol{S}). \tag{4}$$

The optimal mean can be computed in closed-form: $\boldsymbol{m}_B = \sum_k \boldsymbol{x}[k] \boldsymbol{m}_k$. The optimal covariance matrix can be solved via the fixed-point iteration proposed in Álvarez Esteban et al. (2016).

3 Problem Formulation

Model Definition. Our DWB model's data-generating process is illustrated in Fig. 2. First, the *pure-state emission parameters* $\boldsymbol{\Theta} \equiv \{(\boldsymbol{m}_k, \boldsymbol{S}_k)\}_{k=1}^K$, define a Gaussian distribution ρ_k for each pure state k. Second, the *state vector* $\boldsymbol{x}_t$ defines the system state at t and lies on the simplex. Given $\boldsymbol{\Theta}$ and $\boldsymbol{x}_t$, we can form a time-varying Gaussian distribution $\rho_{B_t}(\boldsymbol{x}_t, \boldsymbol{\Theta}) \equiv N(\boldsymbol{m}_{B_t}, \boldsymbol{S}_{B_t})$ for each t, which is a barycentric combination of the K pure Gaussians using weights $\boldsymbol{x}_t$ via (4). We can write the states for an entire sequence as $\boldsymbol{X}$, comprised of an initial state and a sequence of simplex-valued state vectors, denoted $\boldsymbol{X} \equiv \{\boldsymbol{x}_0, \{\boldsymbol{x}_t\}_{t=1}^T\}$.

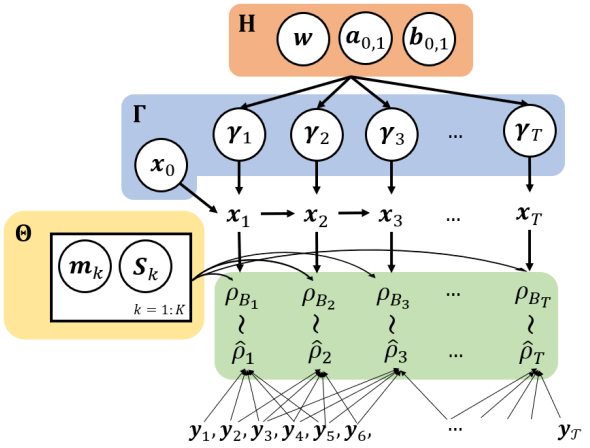

Figure 2: Graphical model diagram of our proposed dynamic Wasserstein barycenter (DWB) time series model. For each window of data, indexed by t, our model forms an emission distribution ρ_{B_t} that is the Wasserstein barycenter given known pure-state Gaussian parameters $\boldsymbol{\Theta} = \{\boldsymbol{m}_k, \boldsymbol{S}_k\}_{k=1}^K$ and time-varying weight vector $\boldsymbol{x}_t$. The simplex-valued state sequence $\{\boldsymbol{x}_t\}_{t=1}^T$ is drawn from a random walk using Beta-mixture draws $\boldsymbol{\gamma}_t$ (Sec. 4.1), with hyperparameters $\boldsymbol{H} = \{\boldsymbol{w}, \boldsymbol{a}_{0,1}, \boldsymbol{b}_{0,1}\}$ and initial state $\boldsymbol{x}_0$. Random variables are denoted by circles. Figure shown has $n = 2$, $\delta = 1$ Colors correspond to terms in (9).

Data Preprocessing. We are given a vector-valued time series of observations $\boldsymbol{y}_\tau \in \mathbb{R}^d, \tau = 1, ..., \mathcal{T}$. Instead of modeling this data directly, to improve smoothness we model the empirical distribution of sliding windows of $2n+1$ samples (Aghabozorgi et al., 2015) strided by δ samples. We retain only windows with complete data, with start times corresponding to $\tau = 1, (\delta+1), (2\delta+1), ..., \lfloor \frac{(\mathcal{T}-(2n+1))}{\delta} \rfloor \delta + 1$, which we index sequentially as $t \in \{1, 2, \dots T\}$. A window indexed at t corresponds to a window centered at $\tau = (\delta(t-1) + n + 1)$, which provides an estimates of the underlying distribution at $\boldsymbol{y}_\tau$. For each window location t, we compute an unbiased *empirical* Gaussian distribution $\hat{\rho}_t = \mathcal{N}(\boldsymbol{m}_t, \boldsymbol{S}_t)$ where,

$$\boldsymbol{m}_t = \frac{1}{2n+1} \sum_{i=1}^{2n+1} \boldsymbol{y}_{\delta(t-1)+i}, \qquad \boldsymbol{S}_t = \frac{1}{2n} \sum_{i=1}^{2n+1} (\boldsymbol{y}_{\delta(t-1)+i} - \boldsymbol{m}_t)(\boldsymbol{y}_{\delta(t-1)+i} - \boldsymbol{m}_t)^T. \tag{5}$$

Minimizing the Wasserstein distance between this sequence of empirical distributions $\hat{\rho}_t$ and the model-predicted distributions ρ_{B_t} drives our model's parameter learning.

Estimation Objective. In practice, we are given an observed sequence of empirical distributions $\{\hat{\rho}_t\}_{t=1}^T$ and a desired number of states K. We wish to estimate the state sequence $\boldsymbol{X}$ and emission parameters $\boldsymbol{\Theta}$. We pose the estimation of $\boldsymbol{X}$ and $\boldsymbol{\Theta}$ as the solution to an optimization problem seeking to balance fidelity to a prior model on the parameters of interest with the desire to minimize the time integrated Wasserstein distance between the predicted and observed distributions:

$$\hat{\boldsymbol{X}}, \hat{\boldsymbol{\Theta}} = \underset{\boldsymbol{X}, \boldsymbol{\Theta}}{\operatorname{argmin}} - \log\left(p(\boldsymbol{X}) p(\boldsymbol{\Theta})\right) + \lambda \sum_{t=1}^T \mathcal{W}_2^2(\hat{\rho}_t, \rho_{B_t}(\boldsymbol{x}_t, \boldsymbol{\Theta})). \tag{6}$$

The scalar weight $\lambda > 0$ trades off the model's fit to data (measured by the Wasserstein distance) with the probability of the state sequence $\boldsymbol{X}$ and pure-state emission parameters $\boldsymbol{\Theta}$ under assumed *prior* distributions. Our chosen priors $p(\boldsymbol{X})$ and the $p(\boldsymbol{\Theta})$ are covered in the following section.

4 Model Parameter Priors

4.1 Prior on Simplex States over Time

Here we develop the *transition* model that generates the sequence of state vectors $\boldsymbol{x}_0, \boldsymbol{x}_1, \ldots \boldsymbol{x}_T$. We assume a first-order Markovian structure: $p(\boldsymbol{X}) = p(\boldsymbol{x}_0) \prod_{t=1}^{T} p(\boldsymbol{x}_t | \boldsymbol{x}_{t-1})$. Recall that each state vector lies on the K-dimensional *simplex*. The geometry of the state space in the case of $K = 3$ states is shown in Fig. 3, where $\boldsymbol{x}_t$ lies in the convex hull of the three simplex vertices, the unit coordinate vectors $\boldsymbol{e}_1, \boldsymbol{e}_2, \boldsymbol{e}_3$. Each vertex is associated with a pure-state in our problem. For a more general K-state problem, this is generalized to the $K-$dimensional simplex, denoted Δ^K, in a straightforward manner.

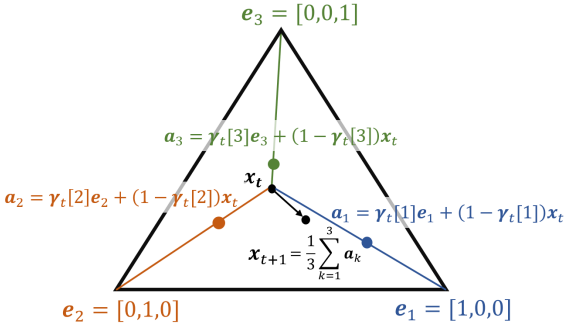

Figure 3: Given current state $\boldsymbol{x}_t$, we transition to next state $\boldsymbol{x}_{t+1}$ by averaging K step-to-vertex updates. For each $k = 1, ..., K$, step-length $\boldsymbol{\gamma}_t[k] \in [0, 1]$ represents a proportional step from $\boldsymbol{x}_t$ to simplex vertex $\boldsymbol{e}_k$.

To ensure that each new state $\boldsymbol{x}_t$ lies in the simplex, we define its update using a K-dimensional "innovations" vector, $\boldsymbol{\gamma}_t$. As seen in Fig. 3, we imagine taking K different steps from the previous state $\boldsymbol{x}_{t-1}$. Each step (indexed by k) moves toward vertex $\boldsymbol{e}_k$ with proportional step length $\boldsymbol{\gamma}_t[k] \in [0, 1]$. A zero-length step ($\boldsymbol{\gamma}_t[k] = 0$) leaves the state at its previous value $\boldsymbol{x}_{t-1}$ while a full step ($\boldsymbol{\gamma}_t[k] = 1$) jumps to the vertex $\boldsymbol{e}_k$. Unlike prior methods (Nguyen and Volkov, 2020), we repeat this process for each of the K components and average their results to achieve the next state $\boldsymbol{x}_t$,

$$\boldsymbol{x}_t = (1 - \tfrac{1}{K} \textstyle\sum_{k=1}^{K} \boldsymbol{\gamma}_t[k])\boldsymbol{x}_{t-1} + \tfrac{1}{K}\boldsymbol{\gamma}_t. \tag{7}$$

By construction, (7) delivers a valid state $\boldsymbol{x}_t$ that lies in the K-simplex.

Inspired by ideas from dynamical Bayesian nonparametric models (Ren et al., 2008), a suitable prior over innovations $\boldsymbol{\gamma}_t$ on the domain $[0, 1]$ is the Beta distribution (Yates and Goodman, 2005). We draw independent innovation values for each component (indexed by K) as IID across time according to a two-component Beta *mixture*. The first component (index 0) captures *stationary* behavior and the second component (index 1) captures *transitions* between pure states:

$$p\left(\{\boldsymbol{\gamma}_t\}_{t=1}^{T}\right) = \prod_{t=1}^{T}\prod_{k=1}^{K} \boldsymbol{w}[k]\text{Beta}\left(\boldsymbol{\gamma}_t[k]; \boldsymbol{a_0}[k], \boldsymbol{b_0}[k]\right) + (1 - \boldsymbol{w}[k])\,\text{Beta}\left(\boldsymbol{\gamma}_t[k]; \boldsymbol{a}_1[k], \boldsymbol{b}_1[k]\right). \tag{8}$$

This Beta-mixture prior for $\boldsymbol{\gamma}_t$ allows flexibility in how $\boldsymbol{x}_t$ evolves on the simplex. By requiring that the Beta parameters of each component are larger than one ($\boldsymbol{a}[k] > 1, \boldsymbol{b}[k] > 1$), we induce uni-modal distributions on $[0, 1]$. For the stationary component (index 0), we expect innovations close to zero, which means we should set $\boldsymbol{b}_0[k] \gg \boldsymbol{a}_0[k]$. We fix $\boldsymbol{a}_0[k] = 1.1, \boldsymbol{b}_0[k] = 20$ in all experiments. For the transition component (index 1) of each state k, we allow Beta parameters $\boldsymbol{a}_1[k], \boldsymbol{b}_1[k]$ as well as the mixture weight $\boldsymbol{w}[k]$ to be *learnable* hyperparameters, denoted $\boldsymbol{H} = \{\boldsymbol{w}, \boldsymbol{a}_1, \boldsymbol{b}_1\}$. To prevent mode collapse we constrain $\boldsymbol{a}_1[k] > 1.1$, $\frac{\boldsymbol{a}_1[k]}{\boldsymbol{a}_1[k]+\boldsymbol{b}_1[k]} > 0.15$, and $\boldsymbol{w}[k] \in [0.01, 0.99]$.

As mentioned in Sec. 3, the simplex-state $\boldsymbol{x}_t$ represents the barycentric mixing weights used to compute the model-predicted distribution of the data. In our current formulation, this sequence of states is *deterministic* according to (7), given the initial state and the sequence of innovation vectors. Since these innovations are the random variables of interest, it is convenient to replace $\boldsymbol{X}$ in (6) with $\boldsymbol{\Gamma} \equiv \left\{\boldsymbol{x}_0, \{\boldsymbol{\gamma}_t\}_{t=1}^{T}\right\}$ resulting in the estimation problem,

$$\hat{\boldsymbol{\Gamma}}, \hat{\boldsymbol{\Theta}}, \hat{\boldsymbol{H}} = \underset{\boldsymbol{\Gamma}, \boldsymbol{\Theta}, \boldsymbol{H}}{\text{argmin}} \underbrace{-\log\left(p_{\boldsymbol{H}}(\boldsymbol{\Gamma})\; p(\boldsymbol{\Theta})\right) + \lambda \textstyle\sum_{t=1}^{T} \mathcal{W}_2^2(\hat{\rho}_t, \rho_{B_t}(\boldsymbol{\Gamma}, \boldsymbol{\Theta}))}_{F\left(\boldsymbol{\Gamma}, \boldsymbol{\Theta}, \boldsymbol{H}, \{\hat{\rho}_t\}_{t=1}^{T}\right)}. \tag{9}$$

In our implementation, we are indifferent to the initial state and therefore set $p(\boldsymbol{x}_0)$ as uniform over the simplex. The coloration in (9) is linked to the associated parameters in Fig. 2.

4.2 Prior on Pure-State Emission Parameters

The final component to (9) is $p(\boldsymbol{\Theta})$, the prior on the pure-state emission parameters. Using a reference normal distribution $\mathcal{N}(\boldsymbol{m}_0, \sigma_0 \boldsymbol{I})$, we can define a probability density function over the space of all Gaussian distributions derived from the Wasserstein distance to this reference distribution,

$$\begin{aligned} p(\boldsymbol{m}, \boldsymbol{S}) =& \kappa(s, \sigma_0) \exp\left(-\frac{1}{2s^2} \mathcal{W}_2^2\left((\boldsymbol{m}, \boldsymbol{S}), (\boldsymbol{m}_0, \sigma_0^2 \boldsymbol{I})\right)\right) \\ =& \kappa(s, \sigma_0) \exp\left(-\frac{1}{2s^2} \left\|\boldsymbol{q} - \boldsymbol{q}_0\right\|_2^2\right), \end{aligned} \tag{10}$$

where s is a scalar hyperparameter that controls the variance of this prior. A simple calculation shows that (10) is equivalent to a multivariate Gaussian distribution over $\boldsymbol{q} \equiv [\boldsymbol{m}, \boldsymbol{\omega}] \in \mathbb{R}^{2d}$, the joint space of means $\boldsymbol{m}$ and the eigenvalues $\boldsymbol{\omega}$ of the covariance matrices $\boldsymbol{S} \in Sym_+^d$. This Gaussian has mean $\boldsymbol{q}_0 = [\boldsymbol{m}_0, \sigma_0, ..., \sigma_0]$ and covariance equal to $s\boldsymbol{I}$. Since the eigenvalues of $\boldsymbol{S}$ must be positive, it follows that the normalizing constant needed for (10) to be a valid distribution given the normal CDF function Φ is $\kappa(s, \sigma_0) = \left(\left(2\pi s^2\right) \Phi\left(\frac{\sigma_0}{s}\right)\right)^{-d}$. We assume that the pure-state distributions parameters are mutually independent. To ensure that this prior scales similarly to the other terms in (9), all of which scale with the length of the time series T, we set $\log\left(p(\boldsymbol{\Theta})\right) = T \sum_{k=1}^{K} \log\left(p(\boldsymbol{m}_k, \boldsymbol{S}_k)\right)$.

5 Model Estimation

Algorithm 1: Dynamical Wasserstein Barycenter (DWB) Time-Series Estimation

Input:
$\boldsymbol{y}_\tau, \tau = 1 \ldots \mathcal{T}$: Time series observations
K: Number of pure states

Hyperparameters:
n: Window size, δ: Window stride
λ: Weight on data-fit term
s: Variance on prior for $\boldsymbol{\Theta}$
(μ_0, σ_0): Mean, var. of $p(\boldsymbol{\Theta})$ reference dist.
η: Convergence threshold

Output:
$\boldsymbol{\Theta} = \left\{\{\boldsymbol{m}_k, \boldsymbol{S}_k\}_{k=1}^K\right\}$: Pure-state emission params
$\boldsymbol{\Gamma} = \left\{\boldsymbol{x}_0, \{\boldsymbol{\gamma}_t\}_{t=1}^T\right\}$: Initial state and innovations
$\boldsymbol{X} = \{\boldsymbol{x}_t\}_{t=1}^T$: Wasserstein barycentric state vector (Computed from $\boldsymbol{\Gamma}$ via (7))
$\boldsymbol{H} = \{\boldsymbol{w}, \boldsymbol{a}_1, \boldsymbol{b}_1\}$: Beta mixture parameters for transition dynamics

for $t = 1, ..., T$ where $T = \lfloor \frac{(\mathcal{T} - (2n+1))}{\delta} \rfloor + 1$ **do**
  $\boldsymbol{m}_t = \frac{1}{(2n+1)} \sum_{i=1}^{2n+1} \boldsymbol{y}_{\delta(t-1)+i}$; // Preprocessing of windowed
  $\boldsymbol{S}_t = \frac{1}{2n} \sum_{i=1}^{2n+1} (\boldsymbol{y}_{\delta(t-1)+i} - \boldsymbol{m}_t)(\boldsymbol{y}_{\delta(t-1)+i} - \boldsymbol{m}_t)^T$; // empirical distributions
  $\hat{\rho}_t = \mathcal{N}(\boldsymbol{m}_t, \boldsymbol{S}_t)$
end
$c^{(0)} = F\left(\boldsymbol{\Gamma}^{(0)}, \boldsymbol{\Theta}^{(0)}, \boldsymbol{H}^{(0)}, \{\hat{\rho}_t\}_{t=1}^T\right)$; // Cost function F defined in (9)
do
  $\boldsymbol{\Gamma}^{(n+1)}, \boldsymbol{H}^{(n+1)} = \text{argmin}_{\boldsymbol{\Gamma}, \boldsymbol{H}} F\left(\boldsymbol{\Gamma}^{(n)}, \boldsymbol{\Theta}^{(n)}, \boldsymbol{H}^{(n)}, \{\hat{\rho}_t\}_{t=1}^T\right)$; // Adam
  $\boldsymbol{\Theta}^{(n+1)} = \text{argmin}_{\boldsymbol{\Theta}} F\left(\boldsymbol{\Gamma}^{(n+1)}, \boldsymbol{\Theta}^{(n)}, \boldsymbol{H}^{(n+1)}, \{\hat{\rho}_t\}_{t=1}^T\right)$; // Riemannian line search
  $c^{(n+1)} = F\left(\boldsymbol{\Gamma}^{(n+1)}, \boldsymbol{\Theta}^{(n+1)}, \boldsymbol{H}^{(n+1)}, \{\hat{\rho}_t\}_{t=1}^T\right)$
while $(c^{(n)} - c^{(n+1)}) > \eta$;

Given a desired number of states K and a multivariate time series dataset $\boldsymbol{y}$, Alg. 1 details the steps needed to learn all parameters of our DWB model: $\boldsymbol{\Gamma}$, the initial state and innovations sequence that drive the dynamical state model; $\boldsymbol{\Theta}$, the pure-state emission distribution means and covariance matrices; and $\boldsymbol{H}$, the hyperparameters governing transition dynamics on the simplex.

The algorithm performs coordinate descent (updating some variables while fixing others) to optimize the objective function in (9). We chose this structure because the update to $\boldsymbol{\Theta}$ is able to exploit specialized optimization structure. Gradient descent methods are used to implement each minimization step in Alg. 1 taking advantage of auto-differentiation in PyTorch (Paszke et al., 2017). The runtime cost of each step in Alg. 1 is $\mathcal{O}(TKd^3)$, where d is the dimension of each observed data vector $\boldsymbol{y}_t$.

Updates to $\boldsymbol{\Gamma}$, $\boldsymbol{H}$ via Adam. The Adam optimizer (Kingma and Ba, 2017) is used to solve the $\boldsymbol{\Gamma}$, $\boldsymbol{H}$ problem on line 8 of Alg. 1. To ensure that $\boldsymbol{\gamma}_t \in [0, 1]$ for $t = 1, \ldots, T$. we clamp these parameters to $[\epsilon, 1 - \epsilon]$ for $\epsilon = 1e^{-6}$. The initial state vector is clamped and normalized to stay on the simplex in a similar manner and the parameters of $\boldsymbol{H}$ are clamped as mentioned in Sec. 4.1.

Updates to $\boldsymbol{\Theta}$ via natural Riemannian geometry. The pure-state emission parameters $\boldsymbol{\Theta}$ define the mean and covariance parameters for K Gaussian distributions. While a variety of methods each based on different geometries have been proposed for optimizing Gaussian parameters (Lin, 2019; Hosseini and Sra, 2015; Arsigny et al., 2007), in this work we choose to leverage the geometry of Gaussian distributions under the Wasserstein distance (Malagò et al., 2018). From the decomposition in (2), we see that optimization for $\boldsymbol{\Theta} \equiv \{(\boldsymbol{m}_k, \boldsymbol{S}_k)\}_{k=1}^K$ under Wasserstein geometry can be carried out over a Riemannian product manifold $\left(\mathbb{R}^d \times Sym_+^d\right)$ (Hu et al., 2020) with standard Euclidean geometry on $\mathbb{R}^d$, and Wasserstein-Bures geometry on Sym_+^d (Malagò et al., 2018; Takatsu, 2011; Bhatia et al., 2017). Therefore, we estimate $\boldsymbol{\Theta}$ over this Riemannian product manifold using a gradient descent line search algorithm (Absil et al., 2008). The supplement provides further details and experimental results demonstrating improved optimization speeds compared to standard Euclidean geometry.

6 Real World Results

6.1 Datasets and Evaluation Procedures

Datasets. Our work is motivated by applications in human activity accelerometry where "pure" states correspond to atomic actions such as walking, running, or jumping. We evaluate our algorithm on two datasets where smooth transitions between states are observable and the number of states is known.

Beep Test (BT, proprietary): 46 subjects run between two points to a metronome with increasing frequency. In this setting the subject alternates between running and standing thus we estimate a two state model. Data is captured from a three-axis accelerometer sampled at 100 Hz.

Microsoft Research Human Activity (MSR, Morris et al. (2014)): 126 subjects perform exercises in a gym setting. Exercises vary among subjects covering strength, cardio, cross-fit, and static exercises. Each time series is truncated to five minutes. We set K to the number of labeled discrete states in the truncated time series (range: 2 to 7). The three-axis accelerometer is sampled at 50 Hz.

	BT	MSR
n	100	250
δ	25	125
λ	100	100
s	1.0	1.0
η	1e-4	1e-4

Table 1: Model hyperparameters

Available labels. All models are trained in *unsupervised* fashion: each method is provided only the 3-axis accelerometer signal $\boldsymbol{y}$ and desired number of states K as input. While some ground-truth state annotations are available, each timestep is labeled as belonging exclusively to one discrete state. This assumes *instantaneous* transition between pure states and belies the underlying gradual transitions (e.g. acceleration from stand to run) that actually occur in the data stream, which our method is designed for. Because annotations that properly characterize the gradual transition between states are not available, we evaluate performance based on how well a given model's predicted emission distribution fits the observed data over the whole time series.

Performance metrics: We measure data fit quality using both the average Wasserstein error (11), akin to the model-fit term in (9), as well as the negative log likelihood (12) of all samples in each window given the model's inferred barycentric distribution for that window.

$$e_W = \frac{1}{T}\sum_{t=1}^{T} \mathcal{W}_2^2(\hat{\rho}_t, \rho_{B_t}), \quad (11) \qquad e_{nll} = \frac{1}{T(2n+1)}\sum_{t=1}^{T}\sum_{i=1}^{2n+1} -\log\left(\rho_{B_t}(\boldsymbol{y}_{\delta(t-1)+i})\right). \quad (12)$$

Baseline methods: To the best of our knowledge, the problem of characterizing continuous transitions among discrete pure states is largely unexplored. Most relevant are continuous state space models which identify a continuous latent state, but not in a manner that identifies pure-states of the system. Therefore, we compare our proposed DWB model a continuous-state deep neural state space (DSS) model. Additionally, we baseline the Wasserstein barycentric interpolation model against the linear interpolation model given by discrete-state Gaussian mixture models (GMM).

GMM Linear interpolation baseline. Under the linear interpolation model, each timestep's emission distribution is a Gaussian mixture of the pure states, $\rho_{G_t} = \sum_{k=1}^K \boldsymbol{x}_t[k]\mathcal{N}(\boldsymbol{m}_k, \boldsymbol{S}_k)$. We highlight that ρ_{B_t} and ρ_{G_t} are equivalent when the $\boldsymbol{x}_t$ is in a pure-state, thus the difference between models

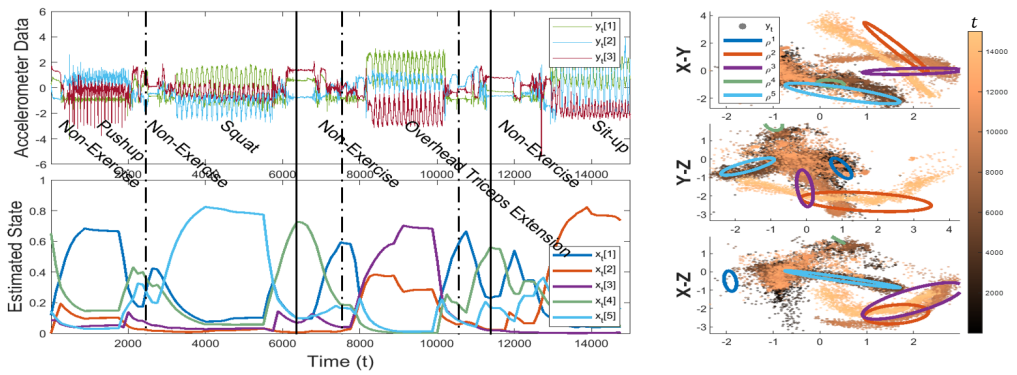

Figure 4: (top left) Three-axis accelerometer of a sample MSR time series consisting of 5 actions. (right) Estimated pure-state Gaussian distributions projected onto each pairwise axes. (bot left) The Wasserstein barycentric weights are given by the estimated system state.

lies in the *transition regions*. The GMM model is implemented in our framework by replacing ρ_{B_t} with ρ_{G_t} in Eqs. (9), (11), and (12). To compute the Wasserstein distance between a Gaussian and Gaussian mixture, we use the upper bound in Chen et al. (2018) for fast gradient-based parameter estimation, but use Monte-Carlo estimation (Sriperumbudur et al., 2010) for more precise evaluation.

Deep neural State Space (DSS, (Krishnan et al., 2016)). The DSS method uses neural networks to parameterize the transition dynamics and the emission distribution of the latent state. Unlike our DWB approach, continuous state space models do not identify pure-states of the system. Thus, post-processing of the latent state space would be required to identify such pure states under these frameworks. For our DWB model, the maximum number of parameters required for state transitions and emissions for the MSR dataset is $p = 91$ ($\mathcal{O}(d^2K)$). Therefore, we evaluate the DSS model under two different settings, one where *both* the transition and emission networks are given a comparable number of parameters to the DWB model ($p = 94$) and one with many more ($p \approx 88,000$) parameters. Exact configuration details for DSS are provided in the supplement.

Hyperparameters and initialization. Hyperparameter choices are documented in Tab. 1, with window size n chosen to capture 2-5 seconds of real-time activity in each dataset. We set $s = 1.0$ and $\lambda = 100$ according to the parameter selection study later in Sec. 6.2. We set $\boldsymbol{m}_0$ to the mean of the observed data, and σ_0 to the average eigenvalue of the set of covariance matrices obtained from a K-component GMM fit to the data using expectation-maximization via code from Pedregosa et al. (2011). Our $\boldsymbol{\Theta}$ estimation problem is non-convex, so the solution is dependent on initialization. To ensure fair comparison, we use the same initialization for each method: a time-series clustering method (Cheng et al., 2020a) that applies matched-filtered change point detection (Cheng et al., 2020b) for the Wasserstein distance. Additional details regarding optimization and initialization are provided in the supplement.

6.2 Experimental Results and Analysis

Qualitative evaluation. Fig. 4 shows for one exemplary MSR time series how our model can estimate pure-state emission distributions for the five states (*right*) and capture the system state in both the stationary and transition periods over time (*bot left*). Of interest are the segments of "Non-Exercise" that have significant contributions from "Pushup" (e.g. *dashed* lines at $t = 2.5k, 7.5k, 10.5k$). The data shows these regions are distinct from the pure "Non-Exercise" regions (e.g. solid lines at $t = 6.5k, 11.5k$): the data mean and variance appear to be intermediate values between "Pushup"and "Non-Exercise" pure states, showing the model's ability to identify gradual transitions. Results from other MSR time series are included in the supplement.

We further visualize the learned state vectors for both our model and the baseline over time for the BT data in Fig. 5, revealing the benefits of our approach for transition modeling. This dataset has two pure states (stand, run), and transition periods can be clearly seen where the subject accelerates and decelerates between each running segment (e.g. Fig. 5(a) $t = 750, 1300$). The GMM interpolation model in Fig. 5(b) identifies the alternating discrete states, however all of the transition regions appear identical, switching between states almost instantly. Only our Wasserstein barycentric model in Fig. 5(d) captures the varying rates of acceleration and deceleration in the transition regions. This

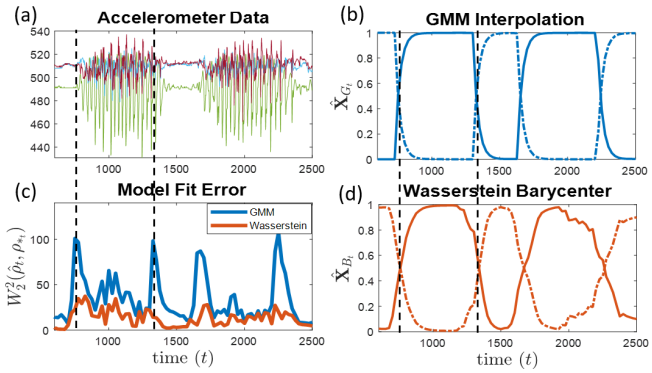

Figure 5: BT alternates between two discrete states (run and stand). (a) The transition between states are observable in the three-axis accelerometer. (b) The state model learned by the GMM linear model is akin to a binary switch in states whereas (c) the Wasserstein barycentric model closely tracks both gradual and quick state changes. (d) This difference is highlighted in the discrepancy in $\mathcal{W}_2^2(\hat{\rho}_t, \rho_{B_t})$ in the transition regions.

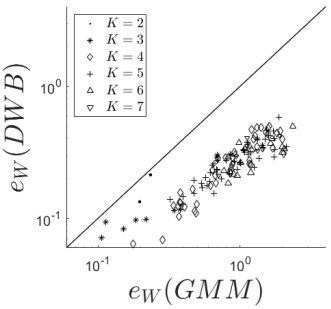

Figure 6: Comparison for each MSR time series between the GMM and Wasserstein barycentric interpolation model (DWB) under the e_W evaluation metric. Points below $y = x$ indicate a better data fit for the Wasserstein model.

discrepancy between the models is highlighted in Fig. 5(c), where the improvement of the Wasserstein relative to the GMM model is accentuated in the transition regions.

Quantitative comparison to GMM linear interpolation. Fig. 6 shows how our Wasserstein barycentric interpolation model improves data fit quality compared to the GMM linear interpolation model, as shown by the decrease in the e_W metric across all of the 126 MSR time series. Evaluation with respect to e_{nll} (lower is better) shows similar improvement with an average of 1.02 for our Wasserstein barycenter model versus 1.50 for the GMM model. Because the interpolated distributions ρ_{B_t} and ρ_{G_t} are equivalent when the system exists in a pure-state, the benefits of our displacement-interpolation barycentric model come from improved fit during the transition periods.

Quantitative comparison to Deep State Space (DSS). As shown in Tab. 2, when given a similar number of parameters, our DWB method outperforms the DSS model regardless of the evaluation metric chosen. When given many more parameters, DSS still has worse e_W but better e_{nll} scores. This difference can be attributed to the fact that the DSS objective is minimized with respect to e_{nll} whereas the DWB objective is minimized with respect to e_W. These results suggest that in terms of characterizing a time series' underlying data distribution, our method is competitive with deep learning methods. Our DWB approach has the added benefit of learning the pure-states of the system, something that would require additional post-processing of the latent space for the DSS and other continuous state space models.

	DWB $p = 91$	DSS $p = 2(94)$	DSS $p \approx 2(88k)$
e_W	**0.27**	4.34	3.07
e_{nll}	1.02	1.49	**-0.508**

Table 2: Evaluation of DWB and DSS with 94 and 80k parameters on MSR dataset. DWB performs best according to e_W, and better than DSS when comparable number of parameters are used under e_{nll}.

Ablation study of the dynamics prior. We also consider a variation of our model using a (non-mixture) single Beta distribution prior for γ_t with parameters $\boldsymbol{a}[k] = 1.1, \boldsymbol{b}[k] = 3.0, \lambda = 10, s = 2.0$. Because this uses a single Beta for learning both stationary and transition dynamics, the model is more sluggish in adapting to fast changing states as seen in the supplement. Under this configuration, the MSR dataset has an average $e_W = 0.50$ compared to the average $e_W = 0.27$ shown in Fig. 6 for the Beta-mixture learnable prior. A sample plot is included in the supplement.

Riemannian optimization. The supplement provides experimental results demonstrating improved optimization speed for estimating $\boldsymbol{\Theta}$ using the Riemannian product manifold discussed in Sec. 5 compared to using the standard Euclidean geometry.

Hyperparameter sensitivity. We explore the sensitivity of the results to variations in two key hyperparameters: the scalar λ that controls the strength of the data term during learning and the pure state variance s, whose inverse also plays the role of a regularization parameter in (9). From Tab. 3, increasing λ generally improves the model fit as is expected from (9). For $\lambda \geq 100$, there is an

optimal value $s = 1.0$ for the MSR dataset which corresponds to "reasonable" pure state distributions seen in Fig. 7. For s too small, the distributions are constrained to the centroid of the data. For s too large, the pure state distributions become disjoint from the data themselves, a result allowed by the simplicial structure of the model. In this regime the the barycentric state vector moves away from the vertices towards the interior of the simplex causing more perceived uncertainty between states.

Sensitivity to initialization. The supplement includes a plot showing similar results to Fig. 6 obtained when initializing $\boldsymbol{\Theta}$ using ground truth activity labels, suggesting our chosen initialization is unbiased.

7 Conclusions and Future Work

Addressing recent trends in technology where the sampling rate of sensors can capture both stationary and transient behaviors of the system, we propose a dynamical Wasserstein barycentric model (DWB) to learn both pure-state emission distributions and the time-varying state vector under a displacement-interpolation transition model between states in an unsupervised setting. For applications such as human activity recognition where transitions are often gradual, the displacement-interpolation given by the Wasserstein barycenter fits data more accurately than the mixture transition model commonly used in the literature. The proposed method can be applied to a wide range of time-series problems including segmentation, clustering, classification, and estimation.

As further contributions, we provide a dynamical state-evolution model of barycentric weight vectors over time. Inspired by previous work in state-space and Bayesian domains, this simplex random walk applies to other temporal modeling applications requiring simplex-valued representations. We also show how tailoring the optimization geometry to the problem leads to improved convergence speeds.

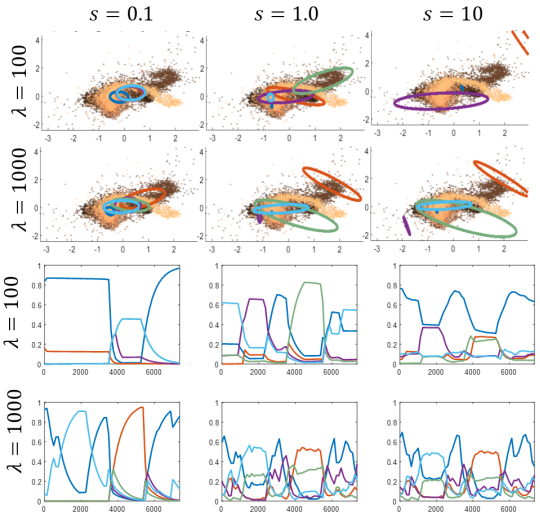

Figure 7: (a) Estimated pure state distributions and (b) estimated state for MSR data varying λ, s

$\lambda \backslash s$	0.01	0.1	1.0	10	100
0.1	596	601	581	395	434
1.0	597	593	455	389	378
10	572	533	284	203	214
100	521	362	**139**	157	173
1000	410	175	128	137	137

Table 3: Hyperparameter sensitivity results. Mean e_W for 25 MSR time series varying λ, s. Reported results in this paper use $\lambda = 100, s = 1.0$.

Limitations. Due to the need to estimate pure-state distributions in a time-sensitive manner, our method is primarily useful for applications where the data is low-dimensional and densely sampled. We have assumed knowledge of the true number of states; in practice at least an upper bound might be known but model size selection remains an open problem. Moreover, we have made a strong assumption that all distributions are Gaussian primarily to leverage the associated geometry and simplify the barycenter computation and optimization.

Future Work. Because the Wasserstein barycenter in (3) is defined for any set of distributions with finite second moment (Peyré and Cuturi, 2019), the displacement-interpolation data model outlined in this work can in principle be extended to non-Gaussian distributions. Constructing tractable algorithms based on non-parametric pure state models is certainly an interesting task. We also observe that the simplex random walk is amenable to natural extensions. For example, in our work, we assume that the transition parameters $\boldsymbol{\gamma}_t$ are IID over time. However, by coupling these parameters, we can obtain higher-order smoothness in the simplex-state vector's trajectory.

Finally, the only barrier for making (9) a true likelihood is building a probabilistic model for the model fit based on the Wasserstein distance to the observed data. This has proven difficult to implement as normalization factor in (10) is dependent on the reference distribution. However, with this modification, we can use posterior analysis to properly assess uncertainty in the model and aid in setting the model parameters including the total number of states K, which currently is assumed to be known a-priori.

8 Acknowledgements

This research was sponsored by the U.S. Army DEVCOM Soldier Center, and was accomplished under Cooperative Agreement Number W911QY-19-2-0003. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Army DEVCOM Soldier Center, or the U.S. Government. The U. S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

We also acknowledge support from the U.S. National Science Foundation under award HDR-1934553 for the Tufts T-TRIPODS Institute. Shuchin Aeron is supported in part by NSF CCF:1553075, NSF RAISE 1931978, NSF ERC planning 1937057, and AFOSR FA9550-18-1-0465. Michael C. Hughes is supported in part by NSF IIS-1908617. Eric L. Miller is supported in part by NSF grants 1934553, 1935555, 1931978, and 1937057.

References

P.-A. Absil, R. Mahony, and R. Sepulchre. 2008. *Optimization algorithms on matrix manifolds*. Princeton University Press, Princeton, N.J. ; Woodstock. OCLC: ocn174129993.

Saeed Aghabozorgi, Ali Seyed Shirkhorshidi, and Teh Ying Wah. 2015. Time-series clustering – A decade review. *Information Systems* 53 (Oct. 2015), 16–38. https://doi.org/10.1016/j.is.2015.04.007

Martial Agueh and Guillaume Carlier. 2011. Barycenters in the Wasserstein Space. *SIAM Journal on Mathematical Analysis* 43, 2 (Jan. 2011), 904–924. https://doi.org/10.1137/100805741

Vincent Arsigny, Pierre Fillard, Xavier Pennec, and Nicholas Ayache. 2007. Geometric Means in a Novel Vector Space Structure on Symmetric Positive-Definite Matrices. *SIAM J. Matrix Anal. Appl.* 29, 1 (Jan. 2007), 328–347. https://doi.org/10.1137/050637996

Rajendra Bhatia, Tanvi Jain, and Yongdo Lim. 2017. On the Bures-Wasserstein distance between positive definite matrices. *arXiv:1712.01504 [math]* (Dec. 2017). http://arxiv.org/abs/1712.01504 arXiv: 1712.01504.

H. Bi, M. Perello-Nieto, R. Santos-Rodriguez, and P. Flach. 2021. Human Activity Recognition Based on Dynamic Active Learning. *IEEE Journal of Biomedical and Health Informatics* 25, 4 (April 2021), 922–934. https://doi.org/10.1109/JBHI.2020.3013403 Conference Name: IEEE Journal of Biomedical and Health Informatics.

Yoosoon Chang, Robert K. Kaufmann, Chang Sik Kim, J. Isaac Miller, Joon Y. Park, and Sungkeun Park. 2020. Evaluating trends in time series of distributions: A spatial fingerprint of human effects on climate. *Journal of Econometrics* 214, 1 (Jan. 2020), 274–294. https://doi.org/10.1016/j.jeconom.2019.05.014

Yoosoon Chang, Chang Sik Kim, and Joon Y. Park. 2016. Nonstationarity in time series of state densities. *Journal of Econometrics* 192, 1 (May 2016), 152–167. https://doi.org/10.1016/j.jeconom.2015.06.025

Yongxin Chen, Tryphon T. Georgiou, and Allen Tannenbaum. 2018. Optimal transport for Gaussian mixture models. *arXiv:1710.07876 [cs, math]* (Jan. 2018). http://arxiv.org/abs/1710.07876 arXiv: 1710.07876.

Kevin C. Cheng, Shuchin Aeron, Michael C. Hughes, Erika Hussey, and Eric L. Miller. 2020a. Optimal Transport Based Change Point Detection and Time Series Segment Clustering. *arXiv:1911.01325 [cs, eess]* (Feb. 2020). http://arxiv.org/abs/1911.01325 arXiv: 1911.01325.

Kevin C. Cheng, Eric L. Miller, Michael C. Hughes, and Shuchin Aeron. 2020b. On Matched Filtering for Statistical Change Point Detection. *IEEE Open Journal of Signal Processing* 1 (2020), 159–176. https://doi.org/10.1109/OJSP.2020.3035070 Conference Name: IEEE Open Journal of Signal Processing.

Philip I. Davies and Nicholas J. Higham. 2000. Numerically Stable Generation of Correlation Matrices and Their Factors. *BIT Numerical Mathematics* 40, 4 (Dec. 2000), 640–651. https://doi.org/10.1023/A:1022384216930

Zoubin Ghahramani and Geoffrey E. Hinton. 2000. Variational Learning for Switching State-Space Models. *Neural Computation* 12, 4 (2000), 831–864. https://doi.org/10.1162/089976600300015619

Reshad Hosseini and Suvrit Sra. 2015. Manifold Optimization for Gaussian Mixture Models. *arXiv:1506.07677 [cs, math, stat]* (June 2015). http://arxiv.org/abs/1506.07677 arXiv: 1506.07677.

Jiang Hu, Xin Liu, Zai-Wen Wen, and Ya-Xiang Yuan. 2020. A Brief Introduction to Manifold Optimization. *Journal of the Operations Research Society of China* 8, 2 (June 2020), 199–248. https://doi.org/10.1007/s40305-020-00295-9

R. E. Kalman. 1960. A New Approach to Linear Filtering and Prediction Problems. *Journal of Basic Engineering* 82, 1 (March 1960), 35–45. https://doi.org/10.1115/1.3662552

Diederik P. Kingma and Jimmy Ba. 2017. Adam: A Method for Stochastic Optimization. *arXiv:1412.6980 [cs]* (Jan. 2017). http://arxiv.org/abs/1412.6980 arXiv: 1412.6980.

Rahul G. Krishnan, Uri Shalit, and David Sontag. 2016. Structured Inference Networks for Nonlinear State Space Models. *arXiv:1609.09869 [cs, stat]* (Dec. 2016). http://arxiv.org/abs/1609.09869 arXiv: 1609.09869.

Zhenhua Lin. 2019. Riemannian Geometry of Symmetric Positive Definite Matrices via Cholesky Decomposition. *arXiv:1908.09326 [math, stat]* (Aug. 2019). https://doi.org/10.1137/18M1221084 arXiv: 1908.09326.

Scott Linderman, Matthew Johnson, Andrew Miller, Ryan Adams, David Blei, and Liam Paninski. 2017. Bayesian Learning and Inference in Recurrent Switching Linear Dynamical Systems. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*. 914–922. https://proceedings.mlr.press/v54/linderman17a.html

Luigi Malagò, Luigi Montrucchio, and Giovanni Pistone. 2018. Wasserstein Riemannian Geometry of Positive Definite Matrices. *arXiv:1801.09269 [math, stat]* (Sept. 2018). http://arxiv.org/abs/1801.09269 arXiv: 1801.09269.

Robert J. McCann. 1997. A Convexity Principle for Interacting Gases. *Advances in Mathematics* 128, 1 (June 1997), 153–179. https://doi.org/10.1006/aima.1997.1634

Dan Morris, T. Scott Saponas, Andrew Guillory, and Ilya Kelner. 2014. RecoFit: using a wearable sensor to find, recognize, and count repetitive exercises. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, Toronto Ontario Canada, 3225–3234. https://doi.org/10.1145/2556288.2557116

Goshu Nagino and Makoto Shozakai. 2006. Distance Measure Between Gaussian Distributions for Discriminating Speaking Styles. (2006), 4.

Tuan-Minh Nguyen and Stanislav Volkov. 2020. On a class of random walks in simplexes. *Journal of Applied Probability* 57, 2 (June 2020), 409–428. https://doi.org/10.1017/jpr.2020.19 arXiv: 1709.00174.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in PyTorch. (2017), 4.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12, 85 (2011), 2825–2830. http://jmlr.org/papers/v12/pedregosa11a.html

Gabriel Peyré and Marco Cuturi. 2019. Computational Optimal Transport: With Applications to Data Science. *Foundations and Trends® in Machine Learning* 11, 5-6 (Feb. 2019), 355–607. https://doi.org/10.1561/2200000073 Publisher: Now Publishers, Inc.

L. R. Rabiner. 1989. A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. IEEE* 77, 2 (Feb. 1989), 257–286. https://doi.org/10.1109/5.18626 Conference Name: Proceedings of the IEEE.

Lu Ren, David B. Dunson, and Lawrence Carin. 2008. The dynamic hierarchical Dirichlet process. In *Proceedings of the 25th international conference on Machine learning - ICML '08*. ACM Press, Helsinki, Finland, 824–831. https://doi.org/10.1145/1390156.1390260

Walter Rudin. 1976. *Principles of mathematical analysis* (3d ed ed.). McGraw-Hill, New York.

Bharath K. Sriperumbudur, Arthur Gretton, Kenji Fukumizu, Bernhard Schölkopf, and Gert R. G. Lanckriet. 2010. Hilbert Space Embeddings and Metrics on Probability Measures. *Journal of Machine Learning Research* 11, 50 (2010), 1517–1561. http://jmlr.org/papers/v11/sriperumbudur10a.html

Asuka Takatsu. 2011. Wasserstein geometry of Gaussian measures. *Osaka Journal of Mathematics* 48, 4 (Dec. 2011), 1005–1026. https://doi.org/10.18910/4973 Publisher: Osaka University and Osaka City University, Departments of Mathematics.

Cédric Villani. 2009. *Optimal transport: old and new*. Number 338 in Grundlehren der mathematischen Wissenschaften. Springer, Berlin. OCLC: ocn244421231.

Greg Welch. 1997. An Introduction to the Kalman Filter. (1997), 16.

Roy D. Yates and David J. Goodman. 2005. *Probability and Stochastic Processes; a Friendly Introduction for Electrical and Computer Engineers* (2nd ed ed.). John Wiley & Sons, Hoboken, NJ.

Pedro C. Álvarez Esteban, E. del Barrio, J. A. Cuesta-Albertos, and C. Matrán. 2016. A fixed-point approach to barycenters in Wasserstein space. *arXiv:1511.05355 [math, stat]* (April 2016). http://arxiv.org/abs/1511.05355 arXiv: 1511.05355.

9 Supplement

9.0.1 Data Acquisition

The Microsoft Research Human Activity (MSR) dataset[2] (Morris et al., 2014) was obtained under a CDLA-Permissive license. The time series included in this submission have been truncated to the first 5 minutes of activity.

The Beep Test (BT) dataset is a proprietary dataset and thus is not included in this submission and is unfortunately not available to share with the public. It was collected under approval by an affiliated institution's Institutional Review Board (IRB). All data has been deidentified before it was shared with study authors. All study authors are approved by their institutional IRB to use this deidentified wearable sensor data for research purposes.

9.0.2 Compute Time and Resources

Results were obtained using 4 CPU cores with 16 GB RAM, replicated for each time series on an internal cluster using slurm. On average convergence was reached in 6 hours.

9.1 Optimization Simulations

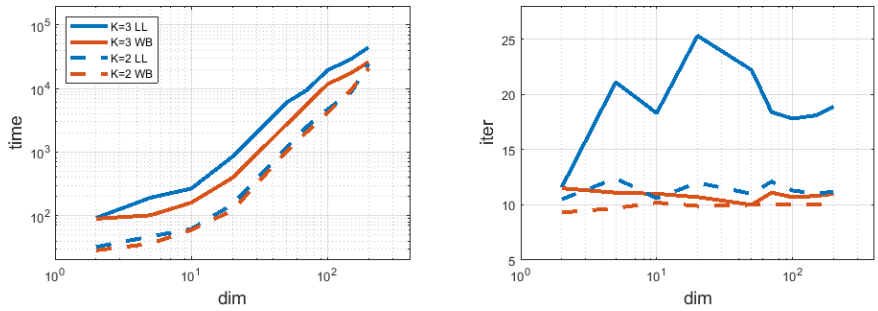

Figure 8: (left) Total time required for convergence optimizing simulated data outlined in Sec. 9.1 across data dimensions ranging from 2 to 200. Optimization with respect to Wasserstein-Bures geometry (red) for PSD matrices converges much faster than using Euclidean geometry (blue) parameterized by the Cholesky (LL) decomposition for both 2-state (dotted), and 3-state (solid) problems. (right) total number of line-search iterations required to reach convergence. Both methods converge to similar results.

To motivate the choice of the Wasserstein-Riemannian manifold for optimization, we compare the optimization speeds in terms of wall-clock time and number of line search iterations with the more standard Euclidean geometry for covariance matrices (Arsigny et al., 2007) parameterized by the Cholesky decomposition. We evaluate performance on a set of simulated time series following our data model similar to that shown in Fig. 1 from the main paper where the system state linearly interpolates between two pure states over the course of 100 time steps. We vary the dimensionality of the data ranging from $d = 2$ to 200, and number of states for $K = 2, 3$, repeating the experiment 10 times for each case.

To construct the data, we start by creating a random Gaussian distribution by generating a random mean vector $\boldsymbol{m}_1 \sim \mathcal{N}(0, \boldsymbol{I})$, and a random PSD matrix $\boldsymbol{S}_1$ generated by the method described in (Davies and Higham, 2000) with eigenvalues given by $\Lambda = \text{diag}([\lambda, ...\lambda_d]$ where $\lambda_i \sim U[0.5, 1.5]$. A second Gaussian distribution is generated a set distance away such that $\mathcal{W}_2^2((\boldsymbol{m}_1, \boldsymbol{S}_1), (\boldsymbol{m}_2, \boldsymbol{S}_2)) = 5$ where $\mathcal{E}^2(\boldsymbol{m}_1, \boldsymbol{m}_2) = 1$, and $\mathcal{B}^2(\boldsymbol{S}_1, \boldsymbol{S}_2) = 4$. This is achieved by generating a random vector in the tangent space and traveling the specified distance, along the geodesic on the manifold in the tangent direction. Specifically, for a random tangent vector to the mean $\boldsymbol{v} \in R^d$, we set $\boldsymbol{m}_2 = \boldsymbol{m}_1 + \frac{1}{\|\boldsymbol{v}\|_2}\boldsymbol{v}$, and for a random symmetric matrix $\boldsymbol{V}$, $\boldsymbol{S}_2 = \exp^{\mathcal{B}}_{\boldsymbol{S}_1}\left(\frac{2}{g^{\mathcal{B}}_{\boldsymbol{S}_1}(\boldsymbol{V},\boldsymbol{V})}\boldsymbol{V}\right)$, where $\exp^{\mathcal{B}}_{\boldsymbol{S}}$ and $g^{\mathcal{B}}_{\boldsymbol{S}}$ are the corresponding exponential map and Riemannian metric on the Wasserstein-Bures Manifold (Takatsu,

[2] https://msropendata.com/datasets/799c1167-2c8f-44c4-929c-227bf04e2b9a

2011). For the three-state example, this process is repeated to generate a third Gaussian distribution such that $\mathcal{W}_2^2((\boldsymbol{m}_2, \boldsymbol{S}_2), (\boldsymbol{m}_3, \boldsymbol{S}_3)) = 5$.

For the simulated data, the state interpolates between $\boldsymbol{e}_1$ to $\boldsymbol{e}_2$ over 100 equi-spaced steps. The three-state problem, continues on and interpolates from $\boldsymbol{e}_2$ to $\boldsymbol{e}_3$ over an additional 100 steps. The empirical Gaussian distributions at each of these intermediate points is generated from $20d$ samples drawn from ρ_{B_t} according to the Wasserstein barycentric model described in the main paper. Since this optimization choice only pertains to $\boldsymbol{\Theta}$, for the purposes of this experiment we fix $\boldsymbol{X}$ as the ground truth thus eliminating the need to estimate $\boldsymbol{\Gamma}, \boldsymbol{H}$.

Fig. 8 shows that estimating the for $\boldsymbol{\Theta}$ using Wasserstein Riemannian geometry for Gaussians has improved performance in terms of converge speed in terms of both wall-clock and number of line-search iterations needed to converge. We found no significant difference in the solutions to which the two methods converged.

Algorithm 2: Riemannian Manifold Line Search (Absil et al., 2008)

Given : $(\mathcal{M}, g_p^{\mathcal{M}}(\cdot,\cdot))$: Riemannian manifold
$f(p)$: Smooth scalar function on the manifold
$\exp_p^{\mathcal{M}}$: Exponential map
α_0: Initial step
β: Sufficient decrease
c: Contraction factor
η: Convergence threshold

while $f\left(p^{(n)}\right) - f\left(p^{(n+1)}\right) > \eta$ **do**
  $\alpha = \alpha_0$
  while
  $\left(f\left(p^{(n)}\right) - f\left(exp_{p^{(n)}}^{\mathcal{M}}\left(\alpha\, gradf\left(p^{(n)}\right)\right)\right)\right) > \alpha\beta g_{p^{(n)}}^{\mathcal{M}}\left(gradf\left(p^{(n)}\right), gradf\left(p^{(n)}\right)\right)$ **do**
    $\alpha = c\alpha$
  end
  $p^{(n+1)} = \exp_{p^{(n)}}^{\mathcal{M}}\left(\alpha\, \text{grad} f\left(p^{(n)}\right)\right)$
end

9.2 Parameter Initialization and Optimization Parameters

Model Params	Initialization	Constraint	Description
$\boldsymbol{\gamma}_t[k]$	$1e-6$	$[0,1]$	State innovation
$\boldsymbol{x}_0[k]$	$\frac{1}{K}$	$\sum_k \boldsymbol{x}_0[k] = 1$	Initial state
$(\boldsymbol{a}_1[k], \boldsymbol{b}_1[k])$	$(10, 20)$	$\boldsymbol{a}_1[k] > 1.1$ $\frac{\boldsymbol{a}_1[k]}{\boldsymbol{a}_1[k]+\boldsymbol{b}_1[k]} > 0.15$	Beta prior for transition dynamics
w	0.5	[0.01, 0.99]	Weight for Beta mixture prior
$(\mu_k, \boldsymbol{S}_k)$	Time series clustering given (Cheng et al., 2020b)	$\mu_k \in \mathbb{R}^d$ $\boldsymbol{S}_k \in Sym_+^d$	Pure state distribution

Table 4: Initialization and constraints of learned model parameters. Time index, $t = 1, ..., T$ and pure-state index $k = 1, ..., K$

The initialization and constraints for the learned model parameters are provided in Table 4. Other fixed parameters for the algorithm are included in Table 5.

For the optimization parameters, we use a learning rate of $2e-3$ for the ADAM optimization (Kingma and Ba, 2017) of $\boldsymbol{\Gamma}, \boldsymbol{H}$ and a convergence criterion of $\eta = 0.05$.

The Riemannian line search used to estimate $\boldsymbol{\Theta}$ is given in Alg. 2. Our implementation sets $\alpha_0 = 1e-1$, $\beta = 1e-10$ $c = 0.5$, $\eta = 0.05$

Model Params	Value	Description
$\boldsymbol{m}_0$	$\frac{1}{T}\sum_{t=1}^{T}\boldsymbol{y}_t$	Used as reference distribution mean for $\boldsymbol{\Theta}$
σ_0	average eigenvalue of covariance matrices of K-component GMM fit to $\boldsymbol{y}_t$ (Pedregosa et al., 2011)	$\sigma_0\mathbf{I}$ used as reference distribution covariance for $\boldsymbol{\Theta}$
η	0.05	Convergence threshold
$(\boldsymbol{a}_0, \boldsymbol{b}_0)$	(1.1, 20)	Beta prior for stationary dynamics
	10	# fixed point iterations for covariance Wasserstein barycenter computation (Álvarez Esteban et al., 2016)
	5000	# Monte Carlo samples used to compute Wasserstein distance between GMM and Gaussian (Sriperumbudur et al., 2010)

Table 5: Value of fixed algorithm parameters.

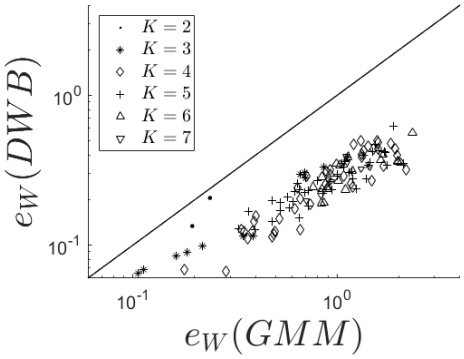

Figure 9: Evaluation comparison between the GMM and Wasserstein barycenter model for the MSR data when using the ground truth discrete labels to initialize the pure-state distribution parameters. Results shown here are close to that shown in the main paper using an unsupervised approach for parameter initialization.

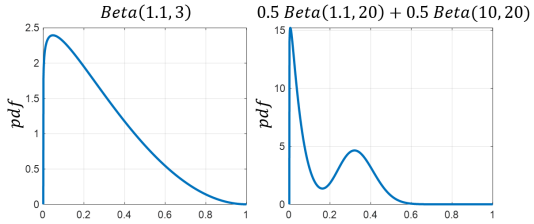

Figure 10: Sample pdfs of single a single component and two-component Beta mixture. Beta distributions are defined on the domain $[0, 1]$ and are uni-modal for parameters $a, b > 1$, The Beta mixture allows us to separately model the stationary and transition dynamics.

9.3 Additional Results

9.3.1 MSR Results with Ground Truth Initialization

In order to stay true to the unsupervised nature of our problem, in our real-world experiments, we initialize the pure-state Gaussian model parameters using the time-series segmentation algorithm using the unsupervised methods described in (Cheng et al., 2020a). Since our problem is non-convex, poor initialization could lead to local minimum. To ensure that our reported results are not a result of biased initialization, we also run the same experiments initializing $\boldsymbol{\Theta}$ according to the sample mean and covariance matrices of each activity given the ground truth discrete labels of the MSR data. As shown in Fig. 9, the results given by this ground truth initialization do not deviate much from Fig. 6 from the main paper where the average absolute difference between the two initialization methods for $e_W(DWB)$ and $e_W(GMM)$ are 0.011 and 0.022 respectively.

9.3.2 Innovation Prior Ablation Study

Here Fig. 11 shows the comparison when using fixed $\boldsymbol{a}, \boldsymbol{b}$ parameters for a single-modal Beta distribution for the prior on $\boldsymbol{\gamma}_t$ versus using the two-component Beta mixture with learnable $\boldsymbol{a}, \boldsymbol{b}, w$ parameters for the transition component as specified in the main paper in Sec. 4.1.

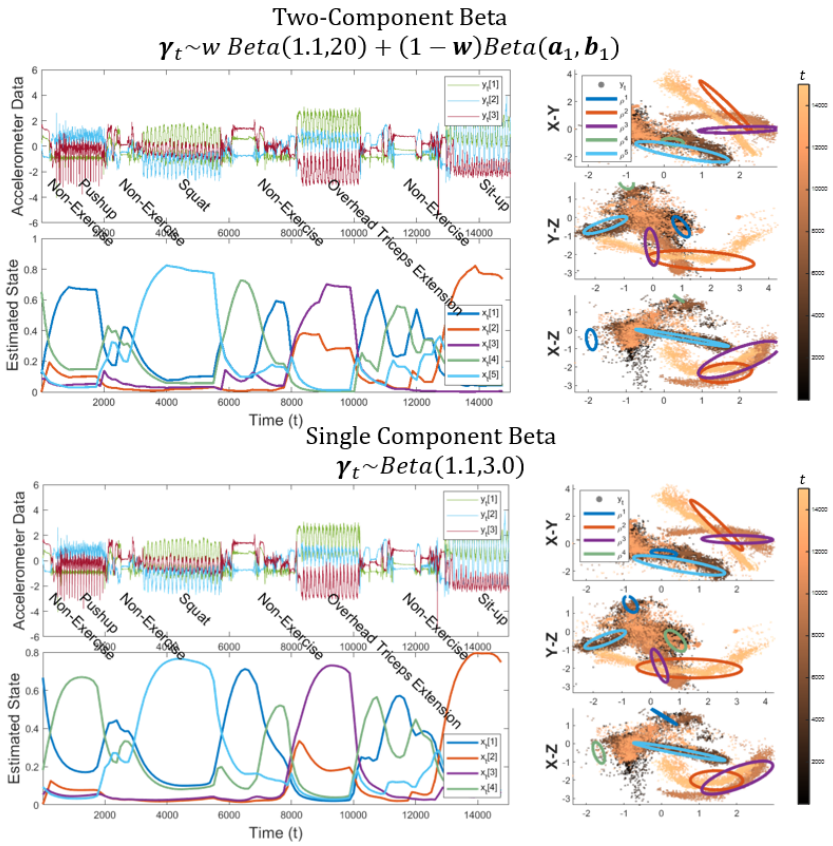

Figure 11: Comparison of model output using a single-component fixed prior versus a two-component learnable prior for the innovations γ_t. While the pure state results in the right column are comparable between the two approaches, the state-variable is more sluggish in adapting to faster transitions in the single-component model.

When using a fixed single component Beta prior, there is a single mode to learn the system dynamics in both transition regions (where one expects more rapid changes in state) and stationary regions (where changes are much smaller). Because of this compromise, the transitions between states are more sluggish while compared to the two-component Beta model. The learned pure-state distribution parameters do not differ much but the overall model fit is improved in the two-component model due to better tracking in the transition regions. For the two-component Beta prior, the average across all MSR datasets is $e_W = 0.50$ compared to the average $e_W = 0.27$ for the single-component fixed Beta model.

9.3.3 Deep State Space Benchmark

We emphasize that in addition to the characterization of the underlying distribution shown in the above evaluation, our DWB approach identifies a set of discrete pure states in the system and yields a directly-interpretable per-sample state representation; the Wasserstein barycentric mixing weights for each pure-state is directly given by the corresponding component of the simplex-state vector. In contrast, there is no direct interpretation of the latent state and no equivalent comparison to the learned pure-state in the DSS model. Additional ad hoc processing (e.g. clustering) of the DSS latent state space would be required to achieve this, which is beyond the scope of our present paper. Tab. 6 highlights additional similarities and differences between the two models.

As mentioned, we run the DSS with two parameter settings. The first uses 2 hidden layers each with 5 neurons for a total of 94 learned parameters for each transmission and emission networks. The second uses the default parameters included with the code[3] which contains 3 hidden layers for

[3] https://github.com/clinicalml/dmm

	DWB	DSS
State Space	K-simplex	R^n
State Transition Dynamics	Beta mixture	Gaussian distribution
State Transition Parameters	Learnable Beta parameters	Neural network parameterizing mean and diagonal covariance of Gaussian
Emission distribution model	Gaussian with full covariance	Gaussian with diagonal covariance
Number of learned parameters	$O(d^2K)$ d =data dimension $K = \#$ clusters	$O(m^2)$ m =NN hidden layer size

Table 6: Comparison between DWB and DSS models

the transition and emission networks with a hidden state dimension of 200 for a total of for each neural network. In both cases, the latent space has dimension (K-1) (to match the dimensionality of the K-simplex), an RNN of 2 layers and 600 nodes each is used as the variational approximation network, and training occurs over 1000 epochs with a learning rate of 0.008. Plots of the variational lower bound show convergence under these conditions