HDP-Flow: Generalizable Bayesian Nonparametric Model for Time Series State Discovery

Sana Tonekaboni[1,2] **Tina Behrouzi**[2] **Addison Weatherhead**[2] **Emily B. Fox**[3] **David Blei**[4] **Anna Goldenberg**[2]

[1]Eric and Wendy Schmidt Center, Broad Institute of MIT and Harvard, Cambridge, MA, USA
[2]Department of Computer science and Vector Institute of AI, University of Toronto, Toronto, ON, Canada
[3]Department of Statistics and Computer Science, Stanford University, Stanford, CA, USA
[4]Department of Statistics and Computer Science, Columbia University, New York, NY, USA

Abstract

We introduce HDP-Flow, a Bayesian nonparametric (BNP) model for unsupervised state discovery in dynamic, non-stationary time series data. Unlike prior work that assumes fixed states, HDP-Flow models evolving datasets with unknown and variable latent states. By integrating the adaptability of BNP models with the expressive power of normalizing flows, HDP-Flow effectively models dynamic, non-stationary patterns, while learning transferable states across datasets with well-calibrated uncertainty. We propose a scalable variational algorithm to enable efficient inference, addressing the limitations of traditional sampling-based BNP methods. HDP-Flow outperforms existing approaches in latent state identification and provides probabilistic insight into state distributions and transition dynamics. Evaluating HDP-Flow across two wearable datasets demonstrates the transferability of states across diverse subpopulations, validating its robustness and generalizability. We demonstrate that HDP-Flow outperforms existing nonparametric models in latent state identification, particularly in the face of non-stationary states. In most cases, it even performs better than models that have prior information about the number of states. Additionally, we show that HDP-Flow's variational inference algorithm successfully scales to long time series, where sampling-based inference fails, showcasing the model's practical utility for large-scale analyses.

1 INTRODUCTION

Unsupervised modeling of latent states in time series can reveal the underlying processes that generate the data. For example, in healthcare, physiological metrics such as heart rate and respiratory rate can be used to infer the underlying health state of a patient, allowing the identification, prediction or tracking of various health conditions [Pantelopoulos and Bourbakis, 2009, Nazaret et al., 2023]. Unsupervised representation learning methods have successfully encoded time series data to capture underlying states [Franceschi et al., 2019, Tonekaboni et al., 2020, Zhang et al., 2022, Yu and Qin, 2022, Zhou et al., 2023]. However, these methods often require prior knowledge of the number of states and cannot adapt to evolving conditions. In real-world scenarios, the number and distribution of states can change over time. For instance, the emergence of a new disease would increase the representation of a previously unrepresented state. Models that can adapt to these changes and accommodate a potentially unbounded number of states are essential for many applications. Bayesian nonparametric (BNP) models offer a solution to this problem [Orbanz and Teh, 2010, Hjort et al., 2010, Lorek et al., 2022, Orbanz and Teh, 2010], but often rely on overly simplistic assumptions for real-world time series data. In particular, while allowing for an unbounded number of states, these models assume simple parametric state descriptions.

In this paper, we introduce a BNP sequence model called HDP-Flow. HDP-Flow combines nonparametric modeling of state dynamics with the expressivity of deep generative modeling, all while ensuring computational efficiency. There are three main components to HDP-Flow: (1) To model state dynamics, HDP-Flow builds on the hierarchical Dirichlet process hidden Markov model (HDP-HMM) [Teh et al., 2006]. Specifically, the sticky HDP-HMM [Fox et al., 2011], a sequence model that enables the number of states to adapt to the observed data and learns realistic transitions by encouraging state persistence. (2) To capture the intricate structure of real-world time series, HDP-Flow integrates the sticky HDP-HMM with conditional normalizing flows [Papamakarios et al., 2017], enabling modeling of complex state-specific emissions. (3) To capture non-stationarity within states, HDP-Flow introduces a time-conditioning

mechanism that tracks state duration and conditions the observation distribution on the number of time steps within the state. This enables modeling trends, periodicity, and other forms of non-stationarity. Traditional HDP-HMMs, with their Markov assumption and static emission distributions, fail to capture such non-stationary states.

Most BNP models rely on sampling-based methods for inference, such as Markov chain Monte Carlo [Neal, 2000] and Gibbs sampling [Teh et al., 2004], which can become computationally intractable when analyzing long time series across large cohorts. To address this limitation, we employ an efficient stochastic variational inference (SVI) algorithm based on black-box variational inference (BBVI) [Ranganath et al., 2014]. This approach enables effective handling of the complex distributions and dependencies inherent in the generative process of HDP-Flow, making it scalable for large-scale applications.

We evaluate HDP-Flow on both real and simulated datasets, comparing the learned states to those of other nonparametric and parametric models. HDP-Flow consistently outperforms nonparametric models in identifying latent states, demonstrating exceptional accuracy in settings with non-stationary emissions. It also bests other models in approximating the true data distribution within each state. Additionally, we test the generalizability of HDP-Flow across two cohorts, demonstrating its ability to adapt to new datasets and provide insights into physiological changes in humans. Finally, when applied to long time series data of human activities, we showcase the superior scalability of HDP-Flow's SVI algorithm compared to sampling-based inference methods.

2 BACKGROUND

Nonparametric HMM This work builds on and contributes to the field of Bayesian nonparametric (BNP) sequence models. BNP models offer a flexible way to represent data where the number of underlying patterns is unknown, achieving this by defining probability distributions over infinite-dimensional spaces [Orbanz and Teh, 2010]. The hierarchical Dirichlet process (HDP) [Orbanz and Teh, 2010] is a prominent BNP approach that models grouped data where the number of groups (or clusters) is unknown. It uses a Dirichlet process (DP) to model the data within each group k. Critically, the HDP links these DPs with a shared base distribution G_0, also governed by a DP, allowing the discovery of patterns shared across groups [Teh et al., 2004]:

$$G_0 \sim DP(\gamma, H_\lambda) \quad G_k \sim DP(\alpha, G_0). \tag{1}$$

The parameter γ controls how tightly data points cluster around the DP mean, while α determines deviations from the base distribution H_λ. Equation 2 reformulates the HDP using the stick-breaking process [Sethuraman, 1994].

$$\begin{aligned} G_0 &= \sum_{k=1}^{\infty} \beta_k \delta_{\theta_k}, \quad \beta \sim GEM(\gamma) \quad \theta_k \overset{iid}{\sim} H_\lambda \\ G_k &= \sum_{j=1}^{\infty} \pi_{k,j} \delta_{\theta_j}, \quad \pi_k \sim DP(\alpha, \beta). \end{aligned} \tag{2}$$

Here, the variables $\{\theta_k\}_{k=1}^{\infty}$ and $\{\beta_k\}_{k=1}^{\infty}$ parameterize the location and the corresponding probability mass of each group. The indicator function δ_{θ_k} evaluates to zero everywhere, except for $\delta_{\theta_k}(\theta_k) = 1$. The variables $\{\beta_k\}_{k=1}^{\infty}$ are sampled from the GEM distribution [Johnson et al., 1997, Pitman and Yor, 1997], following a procedure resembling the recursive breaking of a unit-length stick via $\beta_k = \beta' \prod_{i=1}^{k-1}(1-\beta_i)$, where $\beta' \sim Beta(1, \gamma)$.

The hierarchical Dirichlet process hidden Markov model (HDP-HMM) [Teh et al., 2006] is a Bayesian nonparametric extension of the hidden Markov model (HMM) that uses an HDP to model its state distributions and state transitions. The top level DP determines the global distribution of states, while the draws G_k from the base distribution G_0 determine the transition probabilities from each state k. The parameters $\pi_{i,j}$ of the stick-breaking process can be interpreted as the probability of transitioning from state i to j. The sequence of latent states $z_t \in \{1, 2, \cdots\}$ and observations in the HDP-HMM are then modelled as:

$$z_t \sim \pi_{z_{t-1}} \quad x_t \sim p(x_t|\theta_{z_t}). \tag{3}$$

To encourage self-transitions in HDP-HMM, the transition distributions can be modeled as:

$$\pi_k \sim DP\left(\alpha + \kappa, \frac{\alpha\beta + \delta\kappa}{\alpha + \kappa}\right). \tag{4}$$

The modifications of Equation 2 to Equation 4 (introduced as the sticky HDP-HMM [Fox et al., 2011]) encourages state persistence by an amount proportional to κ. HDP-Flow inherits the scaffolding of its latent variables from this model.

Normalizing Flows Normalizing flows (NFs) are powerful density estimation models that learn complex distributions using a series of invertible transformations. By stacking multiple flow functions $f(\cdot) : \mathbb{R}^D \to \mathbb{R}^D$ such that $x = f(u)$ and $u = f^{-1}(x)$, we can transform a simple distribution $p_u(u)$ into a complex target distribution $p_x(x)$ [Kobyzev et al., 2021]. Using the change of variables formula, we can compute the density $p_x(x)$ as follows:

$$p_x(x) = p_u(f^{-1}(x)) \times \left| det\left(\frac{\partial f^{-1}(x)}{\partial x}\right)\right|. \tag{5}$$

Specifically, HDP-Flow employs conditional masked autoregressive flows (MAFs) [Papamakarios et al., 2017],

which are NFs with transformation layers built as an autoregressive neural network. MAF models the joint distribution of data dimensions as the product of the conditionals $p(x) = \prod_j p(x_j|x_{0:j-1})$. The conditional distribution of each dimension $\{j\}_{j=1}^{D}$ is a transform of the latent distribution of u set to a unit-variance Gaussian, and is modelled as: $p(x_j|x_{0:j-1}) = \mathcal{N}(\mu_j, \sigma_j^2)$. The transform parameters μ_j and σ_j are estimated as a function of $x_{0:j-1}$ using networks denoted as f_μ and f_σ. The flexibility of NFs and the well-defined estimation of the density makes them an efficient and suitable model for estimating the distribution of observations in HDP-Flow.

Inference in Bayesian models The hierarchical structure of HDP models complicates inference. While sampling strategies exist [Teh et al., 2006, Van Gael et al., 2008, Neal, 2003], variational inference (VI) offers a faster, deterministic alternative for approximating probabilities. VI is particularly advantageous for large, high-dimensional datasets where sampling methods become computationally prohibitive [Blei and Jordan, 2004, 2006]. Previous VI methods exist for HMMs [Foti et al., 2014, Johnson and Willsky, 2014], but the HDP-HMM's hierarchical structure requires specialized approaches. Zhang et al. [2016] introduces a VI algorithm specific to the dependencies of HDP-HMM. However, it does not generalize to complex emissions like in HDP-Flow. Alternatively, Black-box VI (BBVI) [Ranganath et al., 2014, Archer et al., 2015] offers a more general solution, avoiding model-specific calculations. We employ a stochastic extension of BBVI to handle HDP-Flow's complex dependencies while ensuring scalability.

3 HDP-FLOW

3.1 HDP-FLOW GENERATIVE PROCESS

The generative process of HDP-Flow is illustrated in Figure 1. This graphical model shows a time series sample $X^{(i)} \in \mathbb{R}^{T\times D}$ with $i \in [0, N]$, where N is the total number of samples. Each $\mathbf{x}_t$ represents a D-dimensional observation at time $t \in [0, T]$, where the total number of time steps T depends on the sample. [1]

In this latent variable model constitutes of (1) *local latent variables* $Z = \{z_t, \forall t \in [1, \dots, T]\}$ that indicate the underlying state of a sample at each time step t, (2) A set of *global latent variables* $\{\beta, \pi_k, \kappa_k, \theta_k\}$ that represent the characteristics of the states, with $k = 1, 2, \dots$ indexing the potentially infinite number of states. The variable β represents the overall density of each state k and π_k is the transition probability from state k to all possible infinite states. The self-transition parameter κ_k is the added weight to transition probabilities that prevents unrealistically fast

[1] For clarity, we omit the sample index i throughout the remainder of the paper.

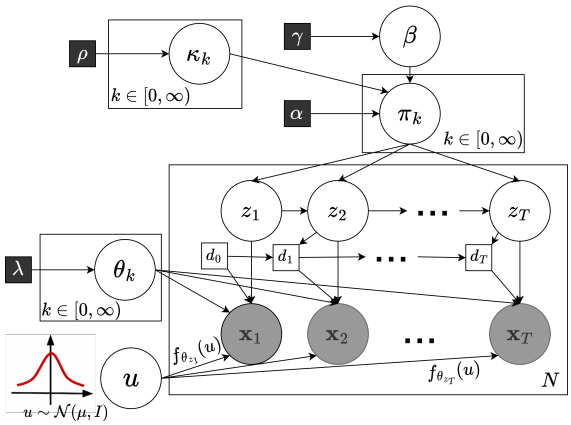

Figure 1: Description of HDP-Flow generative process. The variables $\{\mathbf{x}_t, \forall t \in [1, \dots, T]\}$ are the observations and the local latent variables $\{z_t, \forall t \in [1, \dots, T]\}$ are the underlying state of the sample at all t. The global latent variables $\{\beta, \pi_k, \kappa_k, \theta_k\}$ characterize the global properties of the states, where k is the index for infinite states. The parameters $\{\gamma, \alpha, \rho, \lambda\}$ of the priors are visualized with dark boxes, and d_t is the deterministic variable measuring the number of steps z_t has persisted.

transitions among states. Similar to Fox et al. [2011], this parameter is sampled from a distribution $Beta(\rho_1, \rho_2)$, but instead of using the same sticky parameter for all states, HDP-Flow samples the parameter κ_k independently for each state. This gives HDP-Flow the flexibility to model different self-transition behaviors across different states. The parameters θ_k determine the distribution of observations in state k and are sampled from the base distribution H_λ of the top-level DP. The variable d_t is a deterministic function of latent states which measures how many consecutive steps the sample has been in state z_t. The distribution of observations $p(\mathbf{x}|z_t, d_t)$ at any time t is conditioned on d_t as well as the underlying state, which enables modelling non-stationarity. The generative process is summarized in Equations 6 to 12.

$$\beta \sim GEM(\gamma) \tag{6}$$

$$\theta_k \sim H_\lambda, \tag{7}$$

$$\kappa_k \sim Beta(\rho_1, \rho_2) \tag{8}$$

$$\pi_k \sim DP(\alpha + \kappa_k, \frac{\alpha\beta + \kappa_k\delta_k}{\alpha + \kappa_k}) \tag{9}$$

$$z_t \sim \pi_{z_{t-1}} \tag{10}$$

$$d_t = \begin{cases} d_{t-1} + 1 & \text{if } z_t = z_{t-1} \\ 1 & \text{else.} \end{cases} \tag{11}$$

$$\mathbf{x}_t \sim p(\mathbf{x}|z_t, d_t) \tag{12}$$

The joint likelihood of HDP-Flow variables is shown in Equation 13. Here, the *hyperparameters* $\Theta = \{\alpha, \gamma, \lambda, \rho\}$

are the parameters of the priors over the latent variables (dark squares in Figure 1).

$$p(X, Z, \theta, \pi, \beta, \kappa) = p(\beta|\gamma) \prod_{k}^{\infty} p(\pi_k|\beta, \alpha, \kappa_k) p(\theta_k|\lambda) \times p(\kappa_k|\rho) \prod_{t=1}^{T} p(z_t|z_{t-1}, \pi) p(\mathbf{x}_t|z_t, d_t). \tag{13}$$

What distinguishes HDP-Flow is its approach to estimating the data distribution at any time step t. It incorporates conditional masked autoregressive flows (MAFs) [Papamakarios et al., 2017] into the nonparametric process. When extending MAF to the HDP-Flow setting, each dimension $j \in [D]$ of a sample at time t is modeled conditionally on the preceding dimensions, i.e. $p(x_{j,t}|\mathbf{x}_{0:j-1,t})$. MAFs transform a standard Gaussian into a Normal distribution $\mathcal{N}(\mu_j, \sigma_j^2)$ to model the conditional as $p(x_{j,t}|\mathbf{x}_{0:j-1,t}) = \mathcal{N}(\mu_j, \sigma_j^2)$. The parameters μ_j and σ_j are functions of the preceding observations $\mathbf{x}_{0:j-1,t}$ and are estimated by neural networks f_μ and f_σ. By chaining these conditionals, MAF can model arbitrarily complex data distributions $p(\mathbf{x}_t)$ as a product of the Gaussian conditionals.

HDP-Flow uses state-specific MAF functions $f_{\mu(\theta_{z_t})}$ and $f_{\sigma(\theta_{z_t})}$ where the parameters of the transformation functions are determined by the state z_t at time t. In essence, where $z_t = k$, the function parameters are set as the parameters $\theta_k \sim H_\gamma$. This differs from traditional HDP-HMMs where θ_k directly models the data distribution. To capture non-stationarity, HDP-Flow additionally conditions the observation distribution $p(\mathbf{x}_t|z_t, d_t)$ on d_t. This is achieved by incorporating d_t as an input to the mapping transforms within the conditional MAFs as:

$$x_{j,t} \sim \mathcal{N}(\mu_j, \sigma_j^2), \quad \begin{aligned} \mu_j &= f_{\mu(\theta_{z_t})}(x_{1:j-1,t}, g(d_t)) \\ \sigma_j &= f_{\sigma(\theta_{z_t})}(x_{1:j-1,t}, g(d_t)). \end{aligned} \tag{14}$$

To model various types of non-stationarities, HDP-Flow applies a non-linear function $g(\cdot)$ to the duration variable d_t. Standard activation functions like ReLU can capture trends within states. However, for datasets with periodic patterns (like ECG or EEG), a specialized activation function is needed. HDP-Flow incorporates the activation function $g(x) = sin(x)^2$ [Ziyin et al., 2020] for its periodic inductive bias, enabling it to model cyclical patterns within states.

3.2 VARIATIONAL INFERENCE FOR HDP-FLOW

A key algorithmic challenge for HDP-Flow is performing approximate inference, i.e. estimating the posterior over the global and local variables given observations $\mathbf{X}_{train}$. In large datasets of long time series, most existing sampling-based inference algorithms struggle with long time series due to their repeated reliance on the memory-intensive forward-backward (FB) algorithm.

To address this limitation, HDP-Flow employs stochastic variational inference (SVI) for scalable posterior approximation. While closed-form SVI exists for HDP-HMMs [Zhang et al., 2016], it still relies on FB estimation, creating a bottleneck for long sequences. Moreover, HDP-Flow's exact posterior is heavily conditioned, making closed-form approximations difficult. We instead adopt black-box variational inference (BBVI) [Ranganath et al., 2014] with a mean-field assumption, extending it to HDP-Flow's hierarchical and temporal setting.

HDP-Flow's mean-field factorized variational posterior over all global and local latent variables $W = \{Z, \theta, \beta, \pi, \kappa\}$ is shown in Equation 15. The variational posterior of each variable is modeled independently with a family of distributions similar to the prior. The infinite number of states are truncated to a large value K for the posteriors. Note that the truncation is only for the variational approximation and not the generative model. The set Θ^* contains all variational parameters of the factorized distributions: (1) the Dirichlet concentration parameters for $q(\beta)$ and $q(\pi_k)$, (2) the probabilities of the categorical distribution of the states $q(z_t)$, (3) the mean and variance for $q(\theta_k)$, and (4) the concentration parameters of the Beta distribution of $q(\kappa_k)$:

$$q(Z, \theta, \beta, \pi, \kappa|\Theta^*) = \underbrace{q(\beta)}_{\text{Dirichlet}} \prod_{t} \underbrace{q(z_t^{(i)})}_{\text{Categorical}} \prod_{k} \underbrace{q(\theta_k)}_{\text{Gaussian}} \times \prod_{k} \underbrace{q(\pi_k)}_{\text{Dirichlet}} \prod_{k} \underbrace{q(\kappa_k)}_{\text{Beta}}. \tag{15}$$

We dynamically update the κ parameters during training to enhance performance. The parameters are scaled by a factor of $1 + 0.1 \cdot \text{epoch}$, allowing the model to adapt over time. This approach enables the model to start with broader state representations and refine them progressively, balancing exploration and stability in non-stationary environments.

The VI objective is to find the variational parameters Θ^* such that the posterior $q(W)$ closely approximates the true posterior $p(W|\mathbf{X}_{train})$ by maximizing the evidence lower bound (ELBO), $\mathbb{E}_{q(W)}[\log p(\mathbf{X}_{train}, W) - \log q(W)]$.

Algorithm 1 details the SVI approach we use for HDP-Flow. We compute noisy gradients of the ELBO with the Rao-Blackwellized estimator [Casella and Robert, 1996] and using Monte Carlo samples $W[s] \sim q(W)$ for $s \in [1, \ldots, S]$. The gradients $\hat{\nabla}_{\theta_i^*} \mathcal{L}$ with respect to each parameter $\theta_i^* \in \Theta^*$ of the variational posteriors is a function of all components of the log joint (Eq. 13) that include terms from the i^{th} factor. See Equation 24 in Appendix 8 for derivations of the gradients based on the hierarchical structure of the global and local variables. To further enhance estimation, we incorporate control variates to reduce variance [Ranganath et al., 2014]. Additionally, we employ an adaptive per-component learning rate during parameter updates. This detail is crucial

for HDP-Flow as the parameterizations of its probability distributions have varying scales.

Algorithm 1 Stochastic BBVI for HDP-Flow

Input: $\mathbf{X}_{train}$, $p(x, W)$ (Eq. 13), $q(W)$ (Eq. 15)
repeat
  for all $X^{(i)} \in \mathbf{X}_{train}$ **do**
    for $s = 1$ **to** S **do**
      $W[s] \sim q(Z, \theta, \beta, \pi, \kappa|\Theta^*)$
    end for
    for all $\theta^* \in \Theta^*$ **do**
      Estimate control variate a_{θ^*} [Ranganath et al., 2014];
      $\nabla_{\theta^*}\mathcal{L} = \frac{1}{S}\sum_{s=1}^{S} \nabla_{\theta^*} \log q(W[s]) \times (\log p(x, W[s]) - a_{\theta^*} \log q(W[s]))$ (Eq. 24);
      $\rho_{\theta^*} \leftarrow$ Update adaptive learning rate
      $\theta^* \leftarrow \theta^* + \rho_{\theta^*} \nabla_{\theta^*}$
    end for
  end for
until ∇ELBO$\leq \epsilon$

To compute the log joint probabilites $p(\beta|\gamma)$ and $p(\pi_k|\alpha, \beta, \kappa)$ under the infinite states of the non-parametric priors, we employ a degree-L weak-limit approximation. This technique expresses a DP as the limit of finite-dimensional Dirichlet distributions as the dimensions tend to infinity [Ishwaran and Zarepour, 2002, Teh et al., 2006]. Using the weak-limit theorem, we impose a finite Dirichlet prior over the variables β and π_k as shown in Equation 16. Importantly, the approximation order L can be different and significantly larger than the posterior truncation value K.

$$\begin{aligned} p(\beta|\gamma) &\approx Dir(\gamma/L, \ldots, \gamma/L) \\ p(\pi_k|\beta, \alpha) &\approx Dir(\alpha\beta_1, \ldots, \alpha\beta_L). \end{aligned} \tag{16}$$

Finally, the NF allows us to estimate the log likelihood of observations $p(\mathbf{x}_t|z_t, d_t)$ in closed form,

$$\begin{aligned} \log p(\mathbf{x}_t|z_t, d_t) &= \log p_{\mathbf{u}}(f_{\theta_{z_t}}(\mathbf{x}_t, d_t)) - \log \left| det \left(\frac{\partial \mathbf{x}_t}{\partial \mathbf{u}_t} \right) \right| \\ &= \log p_{\mathbf{u}}(f_{\theta_{z_t}}(\mathbf{x}_t, d_t)) + \sum_i \log \sigma_{i,t}. \end{aligned} \tag{17}$$

Posterior predictive estimation The posterior predictive distribution will help assess generalization. It is the distribution of new, unseen samples $\mathbf{X}_{test}$ given the data we've already seen $\mathbf{X}_{train}$. We can estimate the likelihood of a new sample $\tilde{X} \in \mathbf{X}_{test}$ by integrating over the learned posterior of the global variables as shown in Equation 18:

$$\begin{aligned} p(\tilde{X}|\mathbf{X}_{train}) &= \int_{\beta,\pi,\kappa,\theta} p(\tilde{X}|\beta, \pi, \kappa, \theta) p(\beta, \pi, \kappa, \theta|\mathbf{X}_{train}) \\ &\approx \int_{\beta,\pi,\kappa,\theta} p(\tilde{X}|\beta, \pi, \kappa, \theta) q(\beta, \pi, \kappa, \theta) \\ &\approx \mathbb{E}_{\beta,\pi,\kappa,\theta\sim q} p(\tilde{X}|\beta, \pi, \kappa, \theta). \end{aligned} \tag{18}$$

Knowing the global structure of the generative process of HDP-Flow also enables us to estimate the most likely underlying state $\tilde{Z}$ for a newly observed time series sample $\tilde{X} \in \mathbf{X}_{test}$ using Equation 19.

$$\begin{aligned} p(\tilde{Z}|\tilde{X}, \mathbf{X}_{train}) &= \int_{\beta,\pi,\kappa,\theta} p(\tilde{Z}|\tilde{X}, \beta, \pi, \kappa, \theta) \\ &\quad \times p(\beta, \pi, \kappa, \theta|\mathbf{X}_{train}) \\ &\approx \mathbb{E}_{\beta,\pi,\kappa,\theta\sim q} p(\tilde{Z}|\tilde{X}, \beta, \pi, \kappa, \theta). \end{aligned} \tag{19}$$

To evaluate Equations 18 and 19, we employ the FB algorithm. The k^{th} forward message of the FB algorithm $\mathbf{f}_t(k)$ at time t measures $p(\tilde{X}_{0:t}, z_t = k|\beta, \pi, \kappa, \theta)$. Hence, we can estimate the likelihood of a series of observations $p(\tilde{X}|\beta, \pi, \kappa, \theta)$ as the marginal of the last time step, and find the most likely sequence of underlying states for $\tilde{X}$ using the Viterbi algorithm (details in Appendix 9).

HDP-Flow's state- and duration-dependent observations $p(\mathbf{x}_t|z_t, d_t)$ require us to modify the traditional FB algorithm. We must explicitly account for the probability of a state transition or persistence at each time step. This modification is expressed in Equation 20 and implemented efficiently using matrix calculations.

$$\begin{aligned} \mathbf{f}_t(z_t) = &\sum_{z_{t-1} \neq z_t} \mathbf{f}_{t-1}(z_{t-1}) p(z_t|z_{t-1}) p(\mathbf{x}_t|z_t, d = 1) + \\ &\mathbf{f}_{t-1}(z_{t-1}) p(z_t|z_{t-1}) p(\mathbf{x}_t|z_t, d = d_{t-1} + 1). \end{aligned} \tag{20}$$

During inference, we approximate the FB algorithm to accommodate HDP-Flow's state dynamics. At each step, the state with the highest likelihood updates the duration variable d. This modification, essential for batch inference, does not affect training, while the standard FB algorithm remains applicable in real-time streaming settings.

4 EVALUATION

We evaluate the performance of HDP-Flow in identifying the underlying state of various time series datasets against Bayesian and non-Bayesian benchmark models. [2]

4.1 BASELINES

We benchmark HDP-Flow against three categories of models (More details on implementation in Appendix 11):

[2]Code available at `https://github.com/sanatonek/HDP-Flow.git`

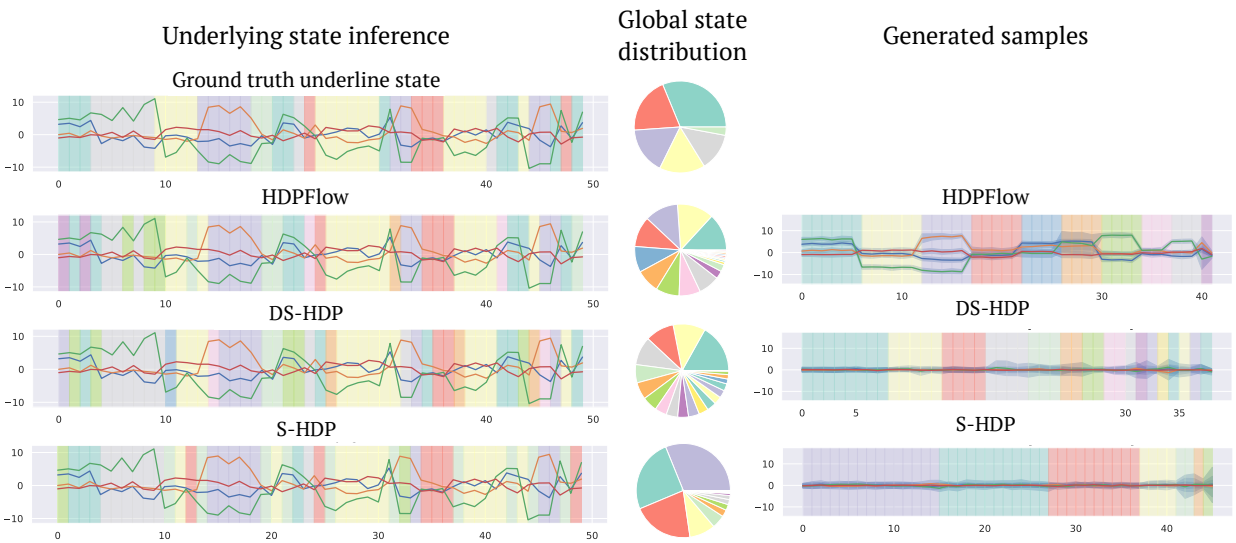

Figure 2: The ground truth vs. BNP model inferences on simulated data III. The first row presents the ground truth underlying state for a test sample (left), and the distribution of states in the training data (middle). Each subsequent row presents results from a different BNP model. The left column shows the inferred state sequences. The middle column shows each model's estimated global state distribution. The right column depicts samples generated by the BNP models, with states as background colors and state duration reflecting their estimated probabilities.

1. Nonparametric HMMs: The sticky HDP-HMM (**S-HDP**) [Fox et al., 2011] and the disentangled sticky HDP-HMM (**DS-HDP**) [Zhou et al., 2020]. For both baselines, We use the augmented autoregressive HMM (ARHMM) implementation that models within-state dynamics by estimating the emission distribution $p(\mathbf{x}_t|z_t, \mathbf{x}_{t-1})$.
2. Unsupervised parametric sequential models: A flow-based continuous HMM (**HMM-Flow**) [Lorek et al., 2022] that uses NF to estimate emission probabilities, and a recurrent neural network (**RNN**) to learn representations that are then clustered to find the states. Both models need the number of states to be specified a priori.
3. Supervised model: An RNN (**RNN sup.**) trained with all state labels. This model shows the best achievable performance on all datasets.

4.2 DATASETS

We studied datasets with varying degrees of complexity for a thorough comparison (more details are in Appendix 10):

Simulated dataset I (static): This dataset consists of 3-dimensional time series samples with 4 different underlying states. The state transitions are governed by an HMM with fixed transition probabilities, and in each state, observations are drawn from a Gaussian $p(x_t) \sim \mathcal{N}(\mu_{z_t}, I)$, where μ_k is fixed for each state.

Simulated dataset II (dynamic): For a more complex setup, the sequence of states in this dataset are generated from a sticky HDP-HMM, with 6 states and different self-transitions. The states are non-stationary with emission for each state k defined as $x_t = a_k t + b_k + \epsilon_t$. State-specific parameter a_k determines the non-stationary trend for each state and ϵ_t is Gaussian noise.

Simulated dataset III (dynamic): Samples for this dataset are directly sampled from the HDP-Flow prior.

CPAP: The CPAP Pressure and Flow Dataset [Guy et al., 2022] that measures differential pressure measurements from a CPAP breathing mask. Participants were instructed to breathe at varying rates, from slow to very fast breathing. The time series consists of 4 signals, and we concatenate different breathing levels for each subject.

Human Activity Recognition (HAR): The UCI HAR dataset [Reyes Ortiz et al., 2012] consists of wearable data from 30 individuals performing six basic activities. These activities and the postural transitions create 12 underlying states. The signals have 6 features, collected at a rate of 50Hz that we down-sample to get on average 1K time steps.

4.3 RESULTS

Our results demonstrate HDP-Flow's strength to accurately identify latent states, learn the global distribution of states, and accurately model the data distribution of each state.

Learning latent states We assess the performance of all models in learning the underlying states in time series. This is measured by the Hamming distance between the true and estimated state sequences, equivalent to the normalized count of mismatches between the predictions and ground truths. To find the one-to-one mapping between predicted and ground truth states for all baselines, we use the Hungarian algorithm [Kuhn, 1955] that maps the indices of the

	Simulated data I		Simulated data II		Simulated data III		HAR		CPAP	
	Hamming	NLL	Hamming	NLL	Hamming	NLL	Hamming	NLL	Hamming	NLL
HDP-Flow	**0.14**±0.04	**270.8**±13.9	**0.25**±0.05	224.8±69.7	**0.38**±0.07	**245.7**±100.4	**0.59**±0.04	433.7±107.2	**0.72**±0.17	1722.0±1647.2
DS-HDP	0.17±0.04	283.6±15.0	0.42±0.12	**165.3**±42.8	0.58±0.09	332.5±18.7	**0.59**±0.06	**-4327.8**±598.1	0.84±0.12	**-31.18**±1427.6
S-HDP	0.25±0.10	327.6±25.6	0.65±0.16	217.3±27.7	0.58±0.09	395.6±20.0	0.66±0.04	-4163.4±575.9	0.74±0.07	671.9±1881.4
RNN	0.76±0.16	N/A	0.86±0.11	N/A	0.83±0.08	N/A	0.92±0.05	N/A	**0.53**±0.25	N/A
HMM-Flow	0.57±0.10	5057±1561	**0.24**±0.08	3480±1176	0.46±0.08	2133±290	0.62±0.05	1779±251	0.54±0.08	26782±57225
RNN Sup.	0.001±0.00	N/A	0.18±0.00	N/A	0.32±0.07	N/A	0.43±0.14	N/A	0.51±0.22	N/A

Table 1: Performance on simulated datasets, measured by the Hamming distance and the posterior predictive likelihood. Standard deviations are reported across samples, and best results with statistical significance are highlighted.

estimated state sequence to the set of indices that maximize the overlap with the true sequence. We present all results on learning the latent states in Table 1. The RNN Sup. baseline provides a measure of the difficulty of inferring the underlying states, serving as a proxy for the best achievable performance assuming access to all state labels.

HDP-Flow consistently outperforms all BNP baselines in learning the latent states on different datasets. Also, it outperforms parametric baselines like HMM-Flow in all datasets except for CPAP. This is notable because the parametric baselines are given the number of states, which BNP models learn on their own. The HAR70+ dataset is an example of a large real dataset with approximately 6K time steps per sample, that highlights the importance of scalability in time series settings. BNP baselines with sampling-based inference fail to train on this dataset, since every sampling step requires approximation of the FB algorithm for long samples. The SVI algorithm of HDP-Flow allows it to scale well to this setting and perform close to a supervised setup (Results in Appendix 15). The left column of Figure 2 shows how the BNP baselines estimate the latent states in the simulated dataset III. The first row shows a sample with the ground truth underlying states and the rest show the estimated state by all BNP models. States are indicated by the background colors, matched such that the same color indicates the same state across all models. Plots for other datasets are in Appendix 16.

We also show that the estimated posterior over the states provide a calibrated probabilistic estimate of states. Figure 3 illustrates the calibration error (ECE) [Naeini et al., 2015] for posterior state probabilities and true states in 2 simulated datasets. These results highlight the model's ability to capture state uncertainty across varying data distributions.

Learning the global posterior Population-level state characteristics are described by the global variables. The posterior distribution $q(\beta)$ reflects the prevalence of each state, allowing us to identify the emergence of new states. The posterior distribution $q(\theta_k)$ defines the data distribution in each state and can be used to generate state-specific samples. The right column of Figure 2 shows time series samples generated from each of the BNP models for simulated dataset III. Each line is one of the time series features

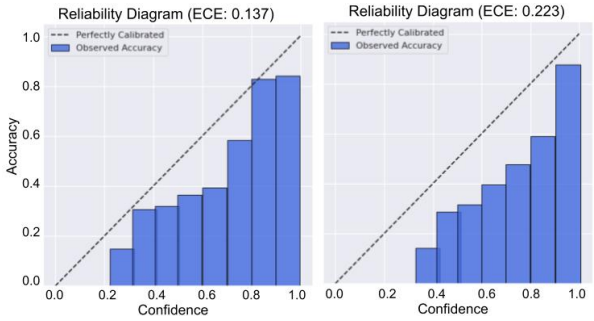

Figure 3: Reliability plots and Expected Calibration Error (ECE) for Simulated Data I and II. Bars closer to the diagonal dotted line indicate better calibration.

and the underlying state is the background color. The length of each generated state is based on the estimated global probability of that state (determined by β_k). Similar to before, the colors are matched to the same state across all baselines. The samples generated by HDP-Flow accurately match the data distribution of each state, apparent by comparing the generated sample to the test samples under each state. It indicates that the posterior has learned the underlying structure effectively. In contrast, generated samples from other BNP baselines are not accurate representations for the states.

We also measure the posterior predictive likelihood of unseen samples under the learned global distribution. We report the negative log likelihood of the posterior predictive in Table 1 as NLL. Despite not learning the posterior over the observations accurately (as shown in Figure 2), S-HDP and DS-HDP achieve better NLL values than HDP-Flow on some datasets. The reason for this is their autoregressive (AR) structure, which allows them to approximate the emission distribution $p(\mathbf{x}_t|z_t, \mathbf{x}_{t-1})$ conditioned on the previous observation. These models learns to set the emission distribution at time t to a value very similar to observation at $t-1$. As a result, NLL values will be very low. This strategy works well in time series where features have very small changes over time, like the HAR dataset (Figure 16.7), however, in datasets where signal changes are more significant, like the case of Simulated data I, the AR model no longer has this advantage.

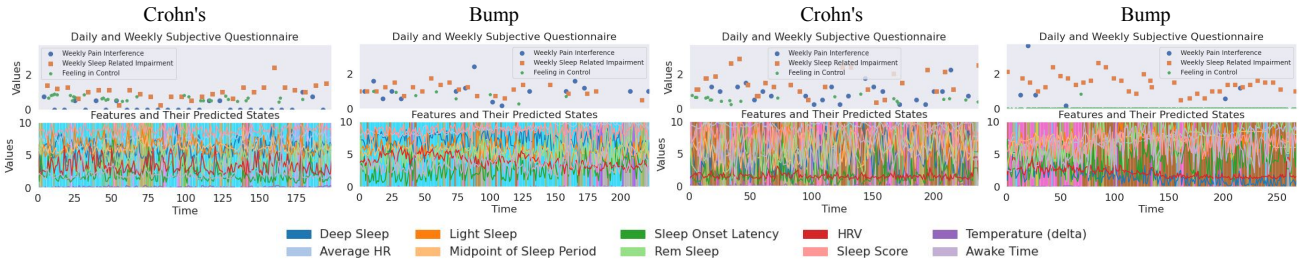

Figure 4: Plot of Beta distribution matching for the Sleep-Related Impairment subjective labels against other features and their state transitions. The Sleep Impairment responses in the two left and two right paired datasets are statistically similar within each pair but significantly different between the pairs based on beta distribution Bayes testing.

In nonparametric models, the parameter β represents an estimate of the global state distribution. The pie charts on Figure 2 compares the estimated posteriors for each model to the ground-truth distribution (top row). Among baselines, S-HDP is more conservative in introducing new states. DS-HDP identifies the existing states but also adds additional states with low probability. HDP-Flow learns fewer additional states, and matches the distribution of existing states closest to the ground truth global distribution. The additional states do not hinder the performance of HDP-Flow because its mean-field assumption models the global state distribution and transitions independently. Finally, the nonparametric nature of BNP models allows them to identify new states which highlights their flexibility to adapt and assign increasing probability to emerging states as more data becomes available.

5 CLINICAL CASE-STUDY

Wearable devices capture rich, longitudinal health data in real-world settings, offering insights into patient well-being beyond the clinic. However, high variability, sensor noise, and scarce annotations make interpretation challenging, and many physiological patterns and signal variations remain poorly understood. The ability of HDP-Flow to identify latent states in time series data without much prior information on the distribution or number of states makes it particularly valuable for wearable healthcare applications. It provides a principled way to discover and adaptively refine latent states. Here, we show that HDP-Flow extracts meaningful states from wearable data and that these states generalize across datasets, enabling the creation of a growing repository for interpretable health monitoring across studies.

Datasets: The Stress in Crohn's dataset tracked 112 patients using Oura ring data to assess stress monitoring for symptom prediction, alongside surveys on flare-ups, medical history, and treatments[3]. Similarly, the BUMP study [Goodday et al., 2022] monitored 431 pregnant participants, of which we use 256 (see Appendix 10.1 for inclusion criteria), capturing physiological and psychological changes.

Experiments: Unlike wearable datasets for HAR, which are collected in controlled environments with well-defined states, wearable data for these studies capture complex, uncontrolled dynamics with many underlying factors and no clear state definitions. Here, we demonstrate an exploratory analysis of learned latent states in this real-world dataset.

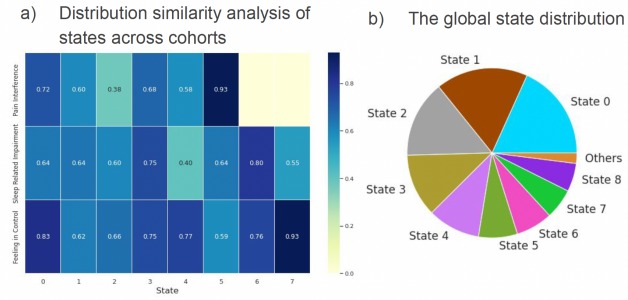

Figure 5: State-wise Distribution of Paired Bump and Crohn's Data. a) Heatmap showing the ratio of paired individuals with similar beta distributions for Sleep Related Impairment, Pain Interference, and Feeling in Control across states. b) The distribution of predicted states of Crohn's data.

We first train HDP-Flow on the Crohn's disease population and analyze the distribution of states it identifies (Figure 5.b). To interpret these states and assess whether they capture similar concepts across populations, we leverage subjective measures from survey data. To quantify the consistency of state distributions across Crohn's and BUMP populations, we perform a Beta-distribution Bayes Factor analysis (detailed in Supplementary Section 14). Figure 5.a illustrates the proportion of paired individuals between the two datasets exhibiting similar beta distributions, where higher values indicate greater cross-population alignment. This analysis also helps characterize state-specific patterns; state 5 predominantly captures pain, while state 7 aligns with feelings of control, a key indicator of stress.

This is further evident in Figure 4, where the dominant states for each individual (across both cohorts) correspond

[3]https://clinicaltrials.gov/study/NCT04809194

to patterns in sleep impairment subjective measures. This suggests that the learned states capture structured relationships between subjective assessments and that this structure transfers effectively to a different population with a distinct distribution of observations. As shown in the bottom panel, wearable data can be highly noisy, making it challenging to extract meaningful signals. In Appendix 15.1, we further show the correlation between input features and probabilistic states, highlighting the consistency of wearable signals. These findings highlight the potential of HDP-Flow in uncovering latent patterns in complex, real-world scenarios.

6 CONCLUSION

We present HDP-Flow, a Bayesian nonparametric model for unsupervised latent state modeling in time series. By unifying the adaptability of Bayesian nonparametrics with the expressive power of conditional normalizing flows, HDP-Flow captures non-stationary and evolving states in uncontrolled environments with minimal prior knowledge all while maintaining an efficient variational inference for modeling complex real-world time series dynamics. Our results demonstrate superior performance in learning latent states and highlight the transferability of the states across sub-populations. However, this flexibility also presents a common challenge in Bayesian nonparametrics: determining the optimal state granularity for structured tasks. Careful tuning of priors is crucial to balance model growth and avoid unnecessary complexity. Although HDP-Flow is computationally more intensive than standard deterministic neural networks, its Bayesian framework provides a structured representation of latent states, uncertainty estimates, and a generative understanding of observations; making it a powerful tool for inference and modeling in evolving time series.

Acknowledgements

Resources used in preparing this research were provided, in part, by the Province of Ontario, the Government of Canada through CIFAR, and companies sponsoring the Vector Institute. This work was supported in part by ONR Grant N00014-22-1-2110, NSF Grant 2205084, and the Stanford Institute for Human-Centered Artificial Intelligence (HAI). EBF is a Chan Zuckerberg Biohub – San Francisco Investigator.

References

Evan Archer, Il Memming Park, Lars Buesing, John Cunningham, and Liam Paninski. Black box variational inference for state space models. *Workshop track of International Conference on Learning Representations*, 2015.

David M Blei and Michael I Jordan. Variational methods for the Dirichlet process. In *Proceedings of the twenty-first International Conference on Machine learning*, page 12, 2004.

David M Blei and Michael I Jordan. Variational inference for Dirichlet process mixtures. *Bayesian Analysis*, 1(1): 121–143, 2006.

David M Blei, Alp Kucukelbir, and Jon D McAuliffe. Variational inference: A review for statisticians. *Journal of the American statistical Association*, 112(518):859–877, 2017.

George Casella and Christian P Robert. Rao-blackwellisation of sampling schemes. *Biometrika*, 83 (1):81–94, 1996.

James M Dolezal, Andrew Srisuwananukorn, Dmitry Karpeyev, Siddhi Ramesh, Sara Kochanny, Brittany Cody, Aaron S Mansfield, Sagar Rakshit, Radhika Bansal, Melanie C Bois, et al. Uncertainty-informed deep learning models enable high-confidence predictions for digital histopathology. *Nature communications*, 13(1):6572, 2022.

Nick Foti, Jason Xu, Dillon Laird, and Emily Fox. Stochastic variational inference for hidden Markov models. *Advances in Neural Information Processing Systems*, 27, 2014.

Emily B Fox, Erik B Sudderth, Michael I Jordan, and Alan S Willsky. A sticky HDP-HMM with application to speaker diarization. *The Annals of Applied Statistics*, pages 1020–1056, 2011.

Jean-Yves Franceschi, Aymeric Dieuleveut, and Martin Jaggi. Unsupervised scalable representation learning for multivariate time series. *Advances in Neural Information Processing Systems*, 32, 2019.

Jakob Gawlikowski, Cedrique Rovile Njieutcheu Tassi, Mohsin Ali, Jongseok Lee, Matthias Humt, Jianxiang Feng, Anna Kruspe, Rudolph Triebel, Peter Jung, Ribana Roscher, et al. A survey of uncertainty in deep neural networks. *Artificial Intelligence Review*, 56(Suppl 1): 1513–1589, 2023.

Andrew Gelman, John B Carlin, Hal S Stern, and Donald B Rubin. *Bayesian data analysis*. Chapman and Hall/CRC, 1995.

Sarah M Goodday, E Karlin, A Brooks, C Chapman, Daniel R Karlin, Luca Foschini, E Kipping, M Wildman, M Francis, H Greenman, et al. Better understanding of the metamorphosis of pregnancy (bump): protocol for a digital feasibility study in women from preconception to postpartum. *NPJ Digital Medicine*, 5(1):40, 2022.

Ella Guy, Jennifer Knopp, and Geoff Chase. Cpap pressure and flow data from a local trial of 30 adults at the university of canterbury, 2022.

Nils Lid Hjort, Chris Holmes, Peter Müller, and Stephen G Walker. *Bayesian nonparametrics*. Cambridge University Press, 2010.

Eyke Hüllermeier and Willem Waegeman. Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods. *Machine learning*, 110(3): 457–506, 2021.

Hemant Ishwaran and Mahmoud Zarepour. Exact and approximate sum representations for the dirichlet process. *Canadian Journal of Statistics*, 30(2):269–283, 2002.

Matthew Johnson and Alan Willsky. Stochastic variational inference for Bayesian time series models. In *International Conference on Machine Learning*, pages 1854–1862. PMLR, 2014.

Norman Lloyd Johnson, Samuel Kotz, and Narayanaswamy Balakrishnan. *Discrete Multivariate Distributions*, volume 165. Wiley New York, 1997.

Robert E Kass and Adrian E Raftery. Bayes factors. *Journal of the american statistical association*, 90(430):773–795, 1995.

Ivan Kobyzev, Simon J.D. Prince, and Marcus A. Brubaker. Normalizing flows: An introduction and review of current methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(11):3964–3979, Nov 2021. ISSN 1939-3539. doi: 10.1109/tpami.2020.2992934. URL http://dx.doi.org/10.1109/TPAMI.2020.2992934.

Harold W Kuhn. The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1-2): 83–97, 1955.

Aleksej Logacjov and Astrid Ustad. HAR70+. UCI Machine Learning Repository, 2023. DOI: https://doi.org/10.24432/C5CW3D.

Pawel Lorek, Rafal Nowak, Tomasz Trzcinski, and Maciej Zieba. Flowhmm: Flow-based continuous hidden markov models. *Advances in Neural Information Processing Systems*, 35:8773–8784, 2022.

Mahdi Pakdaman Naeini, Gregory Cooper, and Milos Hauskrecht. Obtaining well calibrated probabilities using bayesian binning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 29, 2015.

Achille Nazaret, Sana Tonekaboni, Gregory Darnell, Shirley You Ren, Guillermo Sapiro, and Andrew C Miller. Modeling personalized heart rate response to exercise and environmental factors with wearables data. *NPJ Digital Medicine*, 6(1):207, 2023.

Radford M Neal. Markov chain sampling methods for dirichlet process mixture models. *Journal of computational and graphical statistics*, 9(2):249–265, 2000.

Radford M Neal. Slice sampling. *The Annals of Statistics*, 31(3):705–767, 2003.

Andrew A Neath and Joseph E Cavanaugh. The bayesian information criterion: background, derivation, and applications. *Wiley Interdisciplinary Reviews: Computational Statistics*, 4(2):199–203, 2012.

Peter Orbanz and Yee Whye Teh. Bayesian nonparametric models. *Encyclopedia of machine learning*, 1:81–89, 2010.

Alexandros Pantelopoulos and Nikolaos G Bourbakis. A survey on wearable sensor-based systems for health monitoring and prognosis. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 40(1):1–12, 2009.

George Papamakarios, Theo Pavlakou, and Iain Murray. Masked autoregressive flow for density estimation. *Advances in Neural Information Processing Systems*, 30, 2017.

Jim Pitman and Marc Yor. The two-parameter poisson-dirichlet distribution derived from a stable subordinator. *The Annals of Probability*, pages 855–900, 1997.

Rajesh Ranganath, Sean Gerrish, and David Blei. Black box variational inference. In *Artificial Intelligence and Statistics*, pages 814–822. PMLR, 2014.

Jorge Reyes Ortiz, Davide Anguita, Alessandro Ghio, Luca Oneto, and Xavier Parra. Human activity recognition using smartphones, 2012.

Gerard Salton and Christopher Buckley. Term-weighting approaches in automatic text retrieval. *Information processing & management*, 24(5):513–523, 1988.

Jayaram Sethuraman. A constructive definition of dirichlet priors. *Statistica Sinica*, pages 639–650, 1994.

Yee Teh, Michael Jordan, Matthew Beal, and David Blei. Sharing clusters among related groups: Hierarchical Dirichlet processes. *Advances in Neural Information Processing Systems*, 17, 2004.

Yee Whye Teh, Michael I Jordan, Matthew J Beal, and David M Blei. Hierarchical dirichlet processes. *Journal of the American Statistical Association*, 101(476):1566–1581, 2006.

Sana Tonekaboni, Danny Eytan, and Anna Goldenberg. Unsupervised representation learning for time series with temporal neighborhood coding. In *International Conference on Learning Representations*, 2020.

Filippo Valdettaro and Aldo A Faisal. Offline bayesian aleatoric and epistemic uncertainty quantification and posterior value optimisation in finite-state mdps. In *The*

40th Conference on Uncertainty in Artificial Intelligence, 2024.

Jurgen Van Gael, Yunus Saatci, Yee Whye Teh, and Zoubin Ghahramani. Beam sampling for the infinite hidden Markov model. In *Proceedings of the 25th international conference on Machine learning*, pages 1088–1095, 2008.

Jiaxin Yu and S. Joe Qin. Latent state space modeling of high-dimensional time series with a canonical correlation objective. *IEEE Control Systems Letters*, 6:1–1, 2022. doi: 10.1109/LCSYS.2022.3183895.

Aonan Zhang, San Gultekin, and John Paisley. Stochastic variational inference for the HDP-HMM. In *Artificial Intelligence and Statistics*, pages 800–808. PMLR, 2016.

Xiang Zhang, Ziyuan Zhao, Theodoros Tsiligkaridis, and Marinka Zitnik. Self-supervised contrastive pre-training for time series via time-frequency consistency. *Advances in Neural Information Processing Systems*, 35: 3988–4003, 2022.

Ding Zhou, Yuanjun Gao, and Liam Paninski. Disentangled sticky hierarchical Dirichlet process hidden Markov model. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 612–627. Springer, 2020.

Linqi Zhou, Michael Poli, Winnie Xu, Stefano Massaroli, and Stefano Ermon. Deep latent state space models for time-series generation. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 39658–39680. PMLR, 2023. URL `https://proceedings.mlr.press/v202/zhou23i.html`.

Liu Ziyin, Tilman Hartwig, and Masahito Ueda. Neural networks fail to learn periodic functions and how to fix it. *Advances in Neural Information Processing Systems*, 33: 1583–1594, 2020.

HDP-Flow: Generalizable Bayesian Nonparametric Model for Time Series State Discovery (Supplementary Material)

Sana Tonekaboni[1,2] **Tina Behrouzi**[2] **Addison Weatherhead**[2] **Emily B. Fox**[3] **David Blei**[4] **Anna Goldenberg**[2]

[1]Eric and Wendy Schmidt Center, Broad Institute of MIT and Harvard, Cambridge, MA, USA
[2]Department of Computer science and Vector Institute of AI, University of Toronto, Toronto, ON, Canada
[3]Department of Statistics and Computer Science, Stanford University, Stanford, CA, USA
[4]Department of Statistics and Computer Science, Columbia University, New York, NY, USA

7 HDP-FLOW VARIATIONAL POSTERIOR DISTRIBUTION

The mean-field factorized variational posterior of HDP-Flow over the global and local latent variables $W = \{Z, \theta, \beta, \pi, \kappa\}$ is shown below, where the posterior on each variable is modelled independently, and with a family of distribution similar to the prior. The infinite number of states are truncated to $k \in [1, \cdots, L]$ to simplify the variational posterior.

$$\begin{aligned} q(Z, \theta, \beta, \pi, \kappa | \Theta^*) = & \underbrace{q(\beta)}_{Dirichlet} \prod_t \underbrace{q(z_t^{(i)})}_{Categorical} \prod_k \underbrace{q(\theta_k)}_{Gaussian} \\ & \times \prod_k \underbrace{q(\pi_k)}_{Dirichlet} \prod_k \underbrace{q(\kappa_k)}_{Beta} \end{aligned} \tag{21}$$

The distribution of the global state probabilities and the transition probability for each state is modelled with a Dirichlet distribution with L categories. The variational distribution over each latent state z_t is a categorical distribution, following the mean-field assumption.

8 STOCHASTIC BBVI GRADIENT ESTIMATIONS

BBVI uses the Rao-Blackwell estimator for the gradients, which under the mean field assumption becomes:

$$\begin{aligned} \nabla_{\theta^*} \mathcal{L} = \mathbb{E}_{q_1} \ldots \mathbb{E}_{q_i} [& \sum_{j=1}^{i} \nabla_{\theta^*} \log q_j(z_j | \theta_j^*) (\log p(x, z) \\ & - \sum_{j=1}^{i} \log q_j(z_j | \theta_j^*))] \end{aligned} \tag{22}$$

$\nabla_{\theta^*} \mathcal{L}$ as the gradient of the ELBO with respect to θ_i^* , p_i are the components of the log joint that include terms form the ith factor, and $\mathbb{E}_{q_i}$ is the expectation with respect to the set of latent variables that appear in the complete conditional for z_i. Let p_i be the components of the joint that does not include terms from the i-th factor respectively. We can write the gradient with respect to the i-th factor's variational parameters as:

$$\nabla_{\theta_i^*} \mathcal{L} = \mathbb{E}_{q_i} [\nabla_{\theta^*} \log q_i(z_i | \theta_i^*) (\log p_i(x, z_i) - \log q_i(z_i | \theta_i^*))] \tag{23}$$

Using this derivation, the joint distribution defined in Equation 13 and the factorized variation posterior (21), the gradient of each of the variational parameters $\Theta^*_\beta, \Theta^*_\theta, \Theta^*_\pi, \Theta^*_\kappa, \Theta^*_Z$ that are the parameters of $q(\beta), q(\theta), q(\pi), q(\kappa), q(Z)$ respectively is calculated in Equation 24. Note that these estimations take into account the unique sequential and hierarchical dependencies of HDP-Flow distribution.

$$
\begin{aligned}
\hat{\nabla}_{\Theta^*_\beta}\mathcal{L} =& \frac{1}{S}\sum_{s=1}^{S}\nabla\log q(\beta_s)\Big(\log p(\beta_s)+ \\
& \sum_{k=1}^{L}\log p(\pi_{k,s}|\beta_s) - \log q(\beta_s)\Big) \\
\hat{\nabla}_{\Theta^*_{\theta_k}}\mathcal{L} =& \frac{1}{S}\sum_{s=1}^{S}\nabla\log q(\theta_{k,s})\Big(\log p(\theta_{k,s})+ \\
& \sum_{t=1}^{T}\log p(x_t|z_{t,s},\theta_s)\delta_{(z_{t,s}=k)} - \log q(\theta_{k,s})\Big) \\
\hat{\nabla}_{\Theta^*_{\pi_k}}\mathcal{L} =& \frac{1}{S}\sum_{s=1}^{S}\nabla\log q(\pi_{k,s})\Big(\log p(\pi_{k,s})+ \\
& \sum_{t=1}^{T}\log p(z_{t,s}|\theta_s,\pi_s,\kappa_s)\delta_{(z_{t-1}=k)} - \log q(\pi_{k,s})\Big) \\
\hat{\nabla}_{\Theta^*_{\kappa_k}}\mathcal{L} =& \frac{1}{S}\sum_{s=1}^{S}\nabla\log q(\kappa_{k,s})\Big(\log p(\kappa_{k,s})+ \\
& \sum_{k=1}^{L}\log p(\pi_{k,s}|\beta_s) - \log q(\kappa_{k,s})\Big) \\
\hat{\nabla}_{\Theta^*_{z_t}}\mathcal{L} =& \frac{1}{S}\sum_{s=1}^{S}\nabla\log q(z_{t,s})\Big(\log p(z_{t,s}|\pi_s,\kappa_s)+ \\
& \log p(x_t|z_{t,s},\theta_s) - \log q(z_{t,s})\Big).
\end{aligned}
\tag{24}
$$

9 POSTERIOR PREDICTIVE ESTIMATION

In order to measure the posterior likelihood of new samples and to estimate the underlying states for these sample, we use the forward meassages of FB algorithm. The k^{th} forward message of the FB algorithm at time t, $\mathbf{f}_t(k)$, estimates the joint likelihood of the observations upto time t, and the state z_t:

$$
\mathbf{f}_t(k) = p(\tilde{x}_{0:t}, z_t = k|\beta,\pi,\kappa,\theta) \tag{25}
$$

Therefore, the likelihood of a series of observations $p(\tilde{X}|\beta,\pi,\kappa,\theta,\mathbf{X})$ is the marginal of the last time step.

$$
\begin{aligned}
\mathbf{f_t}(k) =& p(x_t|z_t = k)\sum_{z_{t-1}=0}^{k}\mathbf{f_{t-1}}(z_{t-1})p(z_t|z_{t-1}) \\
\mathbf{f_t}(k) =& p(x_1, x_2, \ldots, x_t, z_t = k)
\end{aligned}
\tag{26}
$$

Typically, the forward probability vectors at each step are normalized so that the entries sum to 1. A scaling factor is thus introduced, and as a result, the product of the scaling factors is the total probability for observing the given events irrespective of the final states:

$$\hat{\mathbf{f}}_t(k) = c_t^{-1} p(x_t|z_t = k) \sum_{z_{t-1}=0}^{k} \hat{\mathbf{f}}_{t-1}(z_{t-1}) p(z_t|z_{t-1})$$
$$p(\tilde{X}_{0:T}) = p(x_1, x_2, \ldots, x_T) = \prod_{t=1}^{T} c_t \tag{27}$$

To estimate the negative log likelihood of an unobserved time series sample $\tilde{x}$, we need to estimate the posterior likelihood as follows:

$$\begin{aligned} NLL &= -\log(p(\tilde{X}|\mathbf{X})) \\ &= -\log \int_{\beta,\pi,\kappa,\theta} (p(\tilde{X}|\beta,\pi,\kappa,\theta) p(\beta,\pi,\kappa,\theta|\mathbf{X}) \\ &= -\log \int_{\beta,\pi,\kappa,\theta} p(\tilde{X}|\beta,\pi,\kappa,\theta) q(\beta,\pi,\kappa,\theta) \\ &= -\log \mathbb{E}_{\beta,\pi,\kappa,\theta\sim q} p(\tilde{X}|\beta,\pi,\kappa,\theta) \end{aligned} \tag{28}$$

10 DATASETS

Simulated dataset I This dataset consists of 3-dimensional time series samples with 4 underlying states. The state transitions are governed by an HMM with the following fixed transition probabilities:

$$\pi = \begin{bmatrix} 0.8 & 0.1 & 0.05 & 0.05 \\ 0.1 & 0.8 & 0.1 & 0. \\ 0.05 & 0.1 & 0.8 & 0.05 \\ 0.05 & 0.05 & 0.0 & 0.9 \end{bmatrix}$$

Here the ij'th element is the probability of moving from state i to state j. The emission probability of each state is a Normal Gaussian $\mathcal{N}(\mu_{z_t}, I)$, where μ_{z_t} is fixed for each state and defined as $\begin{bmatrix} 0 & 1 & 2 \\ 5 & 6 & 1 \\ 5 & 5 & 5 \\ 9 & 12 & 11 \end{bmatrix}$, where the the i'th row is the mean vector for state i. The observations are drawn iid from the distribution and don't depend on time.

Simulated dataset II This dataset consists of 4-dimensional time series samples with 6 underlying states. The sequence of states for each sample are determined by a sticky HDP-HMM, with a fixed global state distribution, transition probabilities, and self-transition parameters. This dataset is designed to have a non-stationary emission where within each state k, the data distribution is $x_t = a_k t + b_k + \epsilon$, with ϵ Gaussian noise. The matrix of all a_k and b_k are as follows:

$$A = \begin{bmatrix} 0 & 0 & 0 & 0.2 \\ 0.3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0.2 & -0.1 & 0 \\ 0 & 0.5 & 0 & 0 \end{bmatrix} \quad B = \begin{bmatrix} 8 & 8 & 8 & 8 \\ -3 & -3 & -5 & -5 \\ 5 & 0 & 2 & 0 \\ 2 & 0 & 2 & 0 \\ 0 & 2 & -3 & 0 \\ -4 & -4 & -4 & -4 \end{bmatrix}$$

Simulated dataset III To increase complexity in the experiments, we used a second simulated dataset with additional temporal dynamics. This dataset is generated from a generative model with a similar structrue to HDP-Flow, where the number of states are finite (set to 6) and all latent variables are fixed. The code to generate this dataset is included as part of the supplementary material.

Human Activity Recognition (HAR 70+): This dataset contains 18 fit-to-frail older-adult subjects (70-95 years old) wearing wearable sensors during a semi-structured free-living protocol [Logacjov and Ustad, 2023]. With 6 features and an average of 5K time steps per sample, it is a good example of long time series found in real-world applications.

10.1 BUMP AND CROHNS' DATASETS

The inclusion criteria for both Bump and Crohn's datasets require participants to have less than 60% missing data during their participation and at least 35 recorded data points. Additionally, participants from the Bump dataset with missing information were excluded. The Crohn's dataset consists of a 62% female population, with the demographic distribution shown in Figure 10.1.

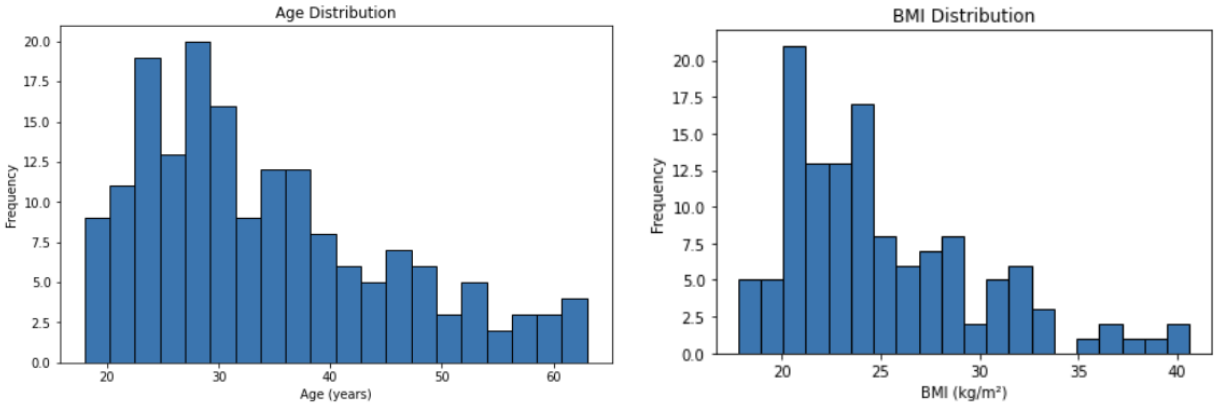

Figure 10.1: Age and BMI distribution of Crohn's data

Physiological wearable input features include Nighttime Mean Heart Rate and Heart Rate Variability (HRV), calculated using RMSSD, which provide insights into cardiovascular function. Sleep metrics, including the duration of deep sleep, REM sleep, and awake time, help assess overall sleep structure and efficiency. Body Temperature Shift, representing deviations from an individual's baseline, offers insights into physiological changes. The Midpoint of Sleep Period, measured in seconds from sleep onset to its midpoint, helps identify patterns and potential disruptions in sleep timing. Sleep Onset Latency, which quantifies the time taken to transition from wakefulness to sleep, serves as a key indicator of sleep efficiency and potential disorders. Lastly, the Sleep Score provides a comprehensive evaluation of sleep quality and quantity by analyzing factors such as sleep stages, restfulness, and timing.

Subjective features that HDP-Flowis not trained on were analyzed to check whether changes in the distribution of input physiological signals relate to the distribution of subjective measures. To identify overlapping survey questions between the Bump and Crohn's datasets, we used TF-IDF Salton and Buckley [1988] vectorization and cosine similarity. First, both sets were transformed into numerical TF-IDF vectors, capturing word importance while minimizing the impact of common terms. Then, cosine similarity was computed between each pair of questions from the two datasets. A similarity threshold of 0.7 was applied to identify closely related question pairs. Based on this analysis, we selected the following set of features:

- Feeling in Control: A daily feature based on the question, "Right now, do you feel in control?" Responses range from 0 to 100, normalized to 0-1 for analysis.
- Weekly Sleep Impairment: Assessed through 9 questions, e.g. "I had problems during the day because of poor sleep.", with responses ranging from "Not at all" (0) to "Very much" (4), capturing the extent of sleep-related difficulties.
- Pain Interference: Evaluates how pain impacts daily life, including participation in social activities, day-to-day tasks, work, and household chores.

11 BASELINES

11.1 HDP-FLOW

We use the pytorch-ts implementation of MAF [1] in HDP-Flow. The models are all trained on CPU machines, and with a 8 hour limit for the training. For evaluation, data is split into train, validation and test cohort. All results in the paper are

[1] https://github.com/zalandoresearch/pytorch-ts

Dataset	α	γ	ρ_1	ρ_2	H	MADE size
Simulated I	4	4	1	3	$\mathcal{N}(0, 3I)$	2
Simulated II	4	2	0.2	0.6	$\mathcal{N}(0, 2I)$	1
Simulated III	6	2	0.5	1	$\mathcal{N}(0, 4I)$	2
HAR	6	4	1	3	$\mathcal{N}(0, 3I)$	2
CPAP	2	2	0.01	0.3	$\mathcal{N}(0, 4I)$	2
HAR70	6	2	0.2	0.6	$\mathcal{N}(0, 3I)$	2
Crohns' data	4	2	0.8	0.2	$\mathcal{N}(0, I)$	2

Table 2: Optimal hyper parameters selected for HDP-Flow for each dataset

reported for the test set, and the validation set is used to choose the parameters of the prior distribution $\Theta = \{\alpha, \gamma, \rho, \lambda\}$. We estimate the posterior predictive over the unobserved samples in the validation set for tuning the parameters of our prior.

Variation inference algorithm can potentially converge to local optima, as a result not yielding the best estimate of the posterior. To overcome this, we use the estimated ELBO to pick the best converged posterior out of 5 runs for each experiment.

The final parameters used for each set of experiments are shown in Table 2

As discussed in 3, we use a MAF to model the data distribution. [2] This method utilizes Masked Autoencoders for Density Estimation (MADE) blocks. Our generative model uses 1 MADE block, with 1 hidden layer, and we treat the size of the hidden layer as one of our parameters.

11.2 S-HDP-HMM AND DS-HDP-HMM

We use implementations of both the S-HDP-HMM and DS-HDP-HMM from Zhou et al. [2020], including the gibbs sampling routine. For S-HDP-HMM, a $gamma(1, 1)$ prior is placed on γ (the concentration parameter on the higher level DP). A value x is sampled from $gamma(\alpha_a, 1/\alpha_b)$ and a value y is sampled from $beta(c_1, c_2)$. These define the initial value of κ (self transition weight) and α (lower level DP concentration parameter) as follows: $\kappa = x * y$, $\alpha = x - \kappa$. For the DS-HDP-HMM model, we place a $beta(\rho_0, \rho_1)$ prior on the κ values, where $\rho_0 = \frac{v_0}{v_1^3}$ and $\rho_1 = \frac{(1-v_0)\rho_0}{v_0}$, where $v_0 \sim Unif(0, 1)$ and $v_1 \sim Unif(0, 1)$. γ is initialized the same way as above described for the S-HDP-HMM, and $\alpha \sim gamma(1, 10)$. In both DS-HDP-HMM and S-HDP-HMM, we use the implementation from Zhou et al. [2020] for the AR-HMM emission, which models the emission distribution as follows: $y_t \sim \mathcal{N}(A_{z_t} y_{t-1}, \Sigma_{z_t})$ where y_t is the observed vector at time t, and A and Σ matrices are learned for each state. A Matrix Normal prior is placed on A_j given Σ_j as follows:

$$
\begin{aligned}
p(A_j|\Sigma_j) = &\frac{1}{(2\pi)^{\frac{d^2}{2}}|V|^{d/2}|\Sigma|^{d/2}} \times \\
&exp\bigg(-\frac{1}{2} tr[(A_j - M)^\top)\Sigma_j^{-1}(A_j - M)V^{-1}] \bigg)
\end{aligned}
\tag{29}
$$

and an Inverse Wishart prior is placed on Σ_j as follows:

$$
\begin{aligned}
p(\Sigma_j) = &\frac{|S_0|^{n_0/2}}{2^{n_0/2}\Gamma_d(n_0/2)} |\Sigma_j|^{-(n_0+d+1)/2} \\
&exp\bigg(-\frac{1}{2} tr(\Sigma_j^{-1} S_0) \bigg)
\end{aligned}
\tag{30}
$$

where $\Gamma_d()$ is the multivariate gamma function, d is the dimension of the data, M is a $d \times d$ 0 matrix and $n_0 = d + 2$. $V = v * I_{d \times d}$ and $S_0 = s * \bar{\Sigma}$ (where $\bar{\Sigma}$ is the emperical covariance matrix of the train data). The final choice of hyper parameters chosen for S-HDP-HMM for all datasetes can be found in Table 3, and for DS-HDP-HMM in Table 4

[2] We use the implementation of MAF provided on

Dataset	α_a	α_b	c_1	c_2	v	s
Simulated I	2	1	1	1	0.1	0.75
Simulated II	2	1	2	1	1	0.75
Simulated III	2	1	2	1	0.1	1.0
HAR	2	1	2	1	1	0.75
HAR70	N/A	N/A	N/A	N/A	N/A	N/A
CPAP	1	1	1	1	1	1

Table 3: Best set of hyper parameters (based on validation loss) for each dataset for S-HDP-HMM

Dataset	v	s
Simulated I	0.1	0.75
Simulated II	0.1	0.75
Simulated III	0.1	1
HAR	1	0.75
HAR70	N/A	N/A
CPAP	1	0.75

Table 4: Best set of hyper parameters (based on validation loss) for each dataset for DS-HDP-HMM

11.3 HMM-FLOW

We use the implementation[3] provided by Lorek et al. [2022]. For all datasets, we train for 100 epochs, using Q training (see Lorek et al. [2022] for more details), learning rate of 0.01. The number of hidden states for the HMM is set to the true number of hidden states for each dataset.

11.4 SUPERVISED RNN

This architecture consists of an LSTM along with a linear classifier which takes in the LSTM's hidden state and predicts the state class. The model is trained end to end. Each model is trained for 100 epochs, with a dropout rate of 0.50 in the LSTM. We vary learning rate and the number of layers in the LSTM, and report the best choice for each dataset (chosen according to the lowest validation loss).

Dataset	LR	# Layers
Simulated I	0.01	4
Simulated II	0.01	4
Simulated III	0.005	2
HAR	0.005	2
HAR70	0.01	4
CPAP	0.01	4

Table 5: Best set of hyper parameters (based on validation loss) for each dataset for Supervised RNN

12 COMPUTATIONAL ANALYSIS

Gibbs sampling is indeed a major computational bottleneck in Bayesian nonparametric models, particularly when applied to long sequences. The inference complexity of HDP-HMM variants is $\mathcal{O}(N(TK^2 + TL^2) + NK)$ (see Zhou et al. [2020] for more details), and addressing this challenge was a key motivation for adopting variational inference in our approach. By using a mean-field approximation within the black-box variational inference (BBVI) framework, we reduce the overall

[3] https://github.com/tooploox/flowhmm

complexity to $\mathcal{O}(NS(TL + TdL))$, where the first term accounts for the variational posterior updates, and the second term reflects the cost of evaluating the MAF-based emission likelihood. This reduction translates into significant efficiency gains in practice. To ensure a fair comparison, we allocated a maximum training time of 20 hours for all Gibbs-based models or until convergence. In practice, these models consistently reached the time limit without converging. In contrast, our model typically converges well within this time frame. Table 6 shows runtime of HDPFlow on different datasets.

Dataset	Sim I	Sim II	Sim III	HAR	CPAP
Runtime	4	4.5	6	2.5	8

Table 6: HDPFlow train time until convergence (measured by hours) on CPU for different datasets

13 UNCERTAINTY MEASURES

Our interpretable probabilistic model enables uncertainty estimation, providing insights into model reliability for new samples and states. Distinguishing different types of uncertainty is crucial, especially in clinical applications Hüllermeier and Waegeman [2021], Valdettaro and Faisal [2024], Gawlikowski et al. [2023].We compute multiple uncertainty metrics based on the inferred state probabilities γ and posterior sample likelihood, capturing epistemic (model-related), aleatoric (data-related), and robustness-based (perturbation-based) uncertainties.

Aleatoric Uncertainty (Variance of Log-Likelihood): Aleatoric uncertainty represents inherent data noise. It is estimated by the variance of posterior likelihoods across MC samples:

$$\text{log\_like\_var} = \text{Var}(\text{posterior\_like}, \text{across MC samples})$$

A higher variance suggests greater ambiguity in the data, limiting confidence in inferred states. Based on log_like_var, we identified 4 patients in the Crohn's dataset and 3 participants in the Bump dataset with high aleatoric uncertainty. These individuals were excluded from the analysis.

State Uncertainty (Variance of γ): We quantify state uncertainty following Blei et al. [2017]:

$$\text{gamma\_var} = \text{Var}(\gamma, \text{across MC samples})$$

This metric reflects the variance of inferred state probabilities γ across Monte Carlo (MC) samples. Higher variance indicates greater disagreement in state assignments, suggesting increased uncertainty in state estimation.

Credible Interval Width (Bayesian Uncertainty): This Bayesian uncertainty measure Gelman et al. [1995] provides interval-based state probability estimates $U - L$, where U, L are the 97.5th and 2.5th percentile bounds of the posterior distribution. A wider interval indicates higher uncertainty in state estimates.

Incorporating gamma_var and credible interval uncertainty estimates to exclude uncertain detected states reduced the Hamming distance by an average of 2-3%.

Uncertainty via Perturbation (Robustness Test): To assess the model's robustness to input noise, we introduce Gaussian perturbations to input features X and compute the variance in inferred states:

$$\text{perturbation\_variance} = \text{Var}(\gamma_{\text{perturbed}}, \text{dim} = 0)$$

Higher variance indicates that state assignments are sensitive to small input changes, suggesting reduced robustness.

Feature Dropout Robustness (Effect of Missing Features): following Dolezal et al. [2022], we evaluate robustness by randomly setting features to zero (dropout) and measuring the variance in inferred states:

$$\text{feature\_dropout\_variance} = \text{Var}(\gamma_{\text{dropout}}, \text{dim} = 0)$$

A high variance suggests that the model strongly depends on specific features, making it more dependent to missing data.

Both dropout sensitivity (≈ 0.003) and noise sensitivity (≈ 0.004) are relatively small, indicating that HDP-Flowis fairly robust to input perturbations. As expected, the perturbation sensitivity for the Bump data, when the model is trained on Crohn's data, is higher, with dropout sensitivity (≈ 0.005) and noise sensitivity (≈ 0.007). Without sleep features, both sensitivities increased to 0.01.

14 BETA BAYES FACTOR ANALYSIS

To assess whether two independent samples originate from the same Beta distribution, we approximate the Bayes Factor (BF) Kass and Raftery [1995] using the Bayesian Information Criterion (BIC) Neath and Cavanaugh [2012]. This approach efficiently estimates distributional differences by balancing model complexity and goodness of fit. Unlike the commonly used Gaussian distribution, we adopt the Beta distribution as it better captures the bounded nature of subjective symptom responses and accommodates a wide range of distribution shapes. This is particularly beneficial when sample sizes are limited and the assumptions of the Central Limit Theorem may not hold. Given two independent samples, X_1 and X_2 (normalized by the maximum value of questionnaire responses), we define the Null hypothesis H_0 as both samples being drawn from a single Beta distribution.

To test these hypotheses, we fit Beta distributions to the combined dataset, yielding parameters $(\alpha_{\text{comb}}, \beta_{\text{comb}})$ and each sample separately, yielding parameters (α_1, β_1) and (α_2, β_2). The fitting procedure is performed using maximum likelihood estimation (MLE), constrained to the interval $[0, 1]$.

The log-likelihood of a Beta-distributed sample S with shape parameters α and b is given by:

$$\log P(X|\alpha, \beta) = \sum_i \log \text{Beta}(X_i|\alpha, \beta), \tag{31}$$

where $\text{Beta}(X_i|\alpha, \beta)$ denotes the probability density function (PDF) of the Beta distribution. We compute $\log L_{\text{comb}}$: Log-likelihood of the combined dataset and $\log L_{\text{sep}}$ which is sum of log-likelihoods for the separate distributions.

To balance model complexity and fit, we compute the BIC for each model using:

$$BIC = k \log(n) - 2 \log L, \tag{32}$$

where k is the number of parameters in the model, n is the sample size, and $\log L$ is the log-likelihood of the fitted model. The BIC values are computed as:

$$BIC_{\text{comb}} = 2 \log(|X_1| + |X_2|) - 2 \log L_{\text{comb}}, \tag{33}$$

$$BIC_{\text{sep}} = 4 \log(|X_1| + |X_2|) - 2 \log L_{\text{sep}}. \tag{34}$$

Since direct computation of the Bayes Factor requires marginal likelihood estimation, which is computationally expensive, we approximate it using the difference in BIC values:

$$BF \approx e^{(BIC_{\text{comb}} - BIC_{\text{sep}})/2}. \tag{35}$$

The computed Bayes Factor provides a quantitative measure of evidence for model selection:

- **If** $BF > 10$: Strong evidence **against** H_0, suggesting that S_1 and S_2 are drawn from distinct Beta distributions.
- **If** $BF \leq 10$: Insufficient evidence to reject H_0.

This implementation enables efficient hypothesis testing while preserving the interpretability of Bayesian model selection.

15 SUPPLEMENTARY RESULTS

15.1 BUMP AND CROHN'S CROSS COHORT ANALYSIS

Figure 16.3 illustrates the consistency in wearable signal distributions and their correlation with states across the Bump and Crohn's datasets. While the z-score correlation of Awake Time remains nearly identical between the two datasets, relative

	HAR 70+	
	Hamming	NLL
HDP-Flow	**0.28±0.06**	**5219.0±1106.6**
DS-HDP	—	—
S-HDP	—	—
RNN	0.56 ± 0.14l	N/A
HMM-Flow	**0.28±0.07**	40121.6±4341.1
RNN Sup.	0.27 ± 0.08	N/A

Table 7: Performance on real-world datasets, measured by the Hamming distance, and the posterior predictive likelihood. Standard deviation are reported across samples, and best results with statistical significance are highlighted.

differences emerge in specific correlations: Body Temperature varies in states 0, 2, 3, 4, and 7, HRV differs in states 4 and 5, and Heart Rate shows variation in states 3 and 9. These features are particularly important as they exhibit distinct behavior in pregnant women, a pattern that is reflected in the Bump data in this figure. As a result, HDP-Flow serves as a powerful framework for tracking and interpreting wearable data across cohorts, identifying new states or shifts in state distribution, and detecting significant changes in physiological signals.

16 SUPPLEMENTARY FIGURES

This section provides similar visualizations as the ones present in the paper for all datasets. In Figure 16.1, compared to the simulated datasets I and II (3), this dataset is more complex. However, calibration (ECE = 0.2328) remains reasonable, though some confidence bins show larger deviations from perfect calibration.

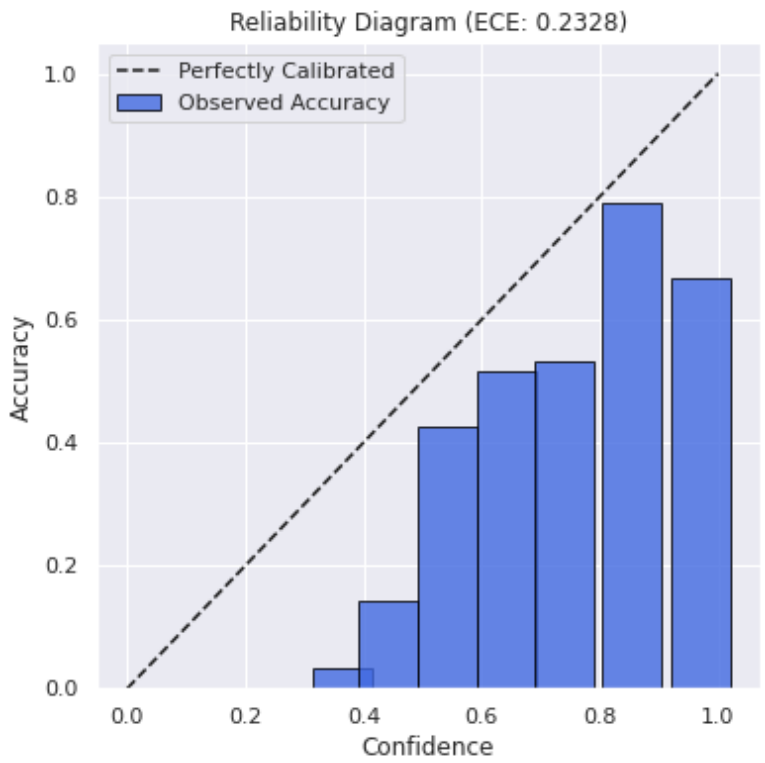

Figure 16.1: Reliability Plot of Simulated data III

In the rest of the figures, the first row presents the ground truth underlying state for a test sample (left), and distribution of states in the training data (middle). Each subsequent row presents corresponding results from a different model (HDP-Flow

and baselines). The left column shows the inferred state sequences for a test sample, indicated by the background color. The middle column shows each model's estimated global state distribution. The right column depicts samples generated by the BNP models, with states as background colors and state duration reflecting their estimated probabilities. All colors are matched for each state.

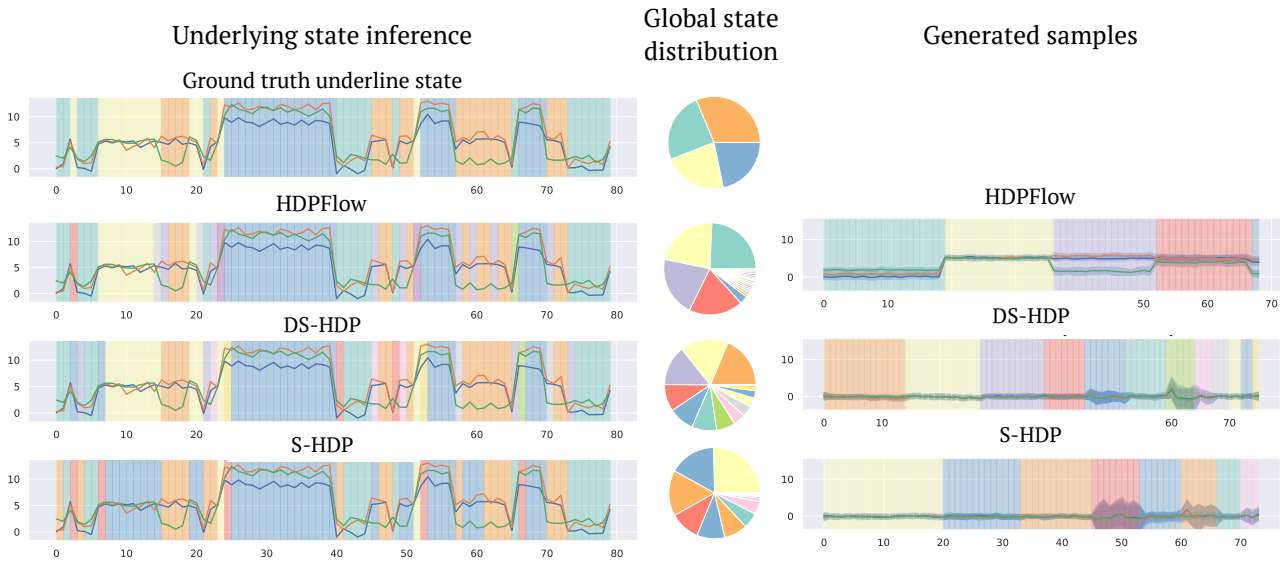

Figure 16.2: Ground Truth vs. BNP models inference on **Simulated Data I**.

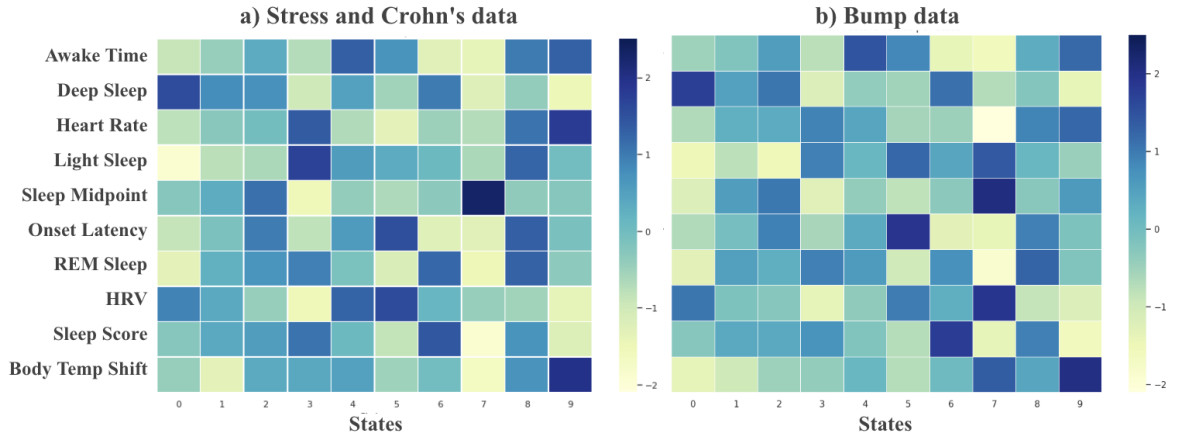

Figure 16.3: Z-scored mean values of physiological features across predicted states in both the Crohn's and Bump datasets.

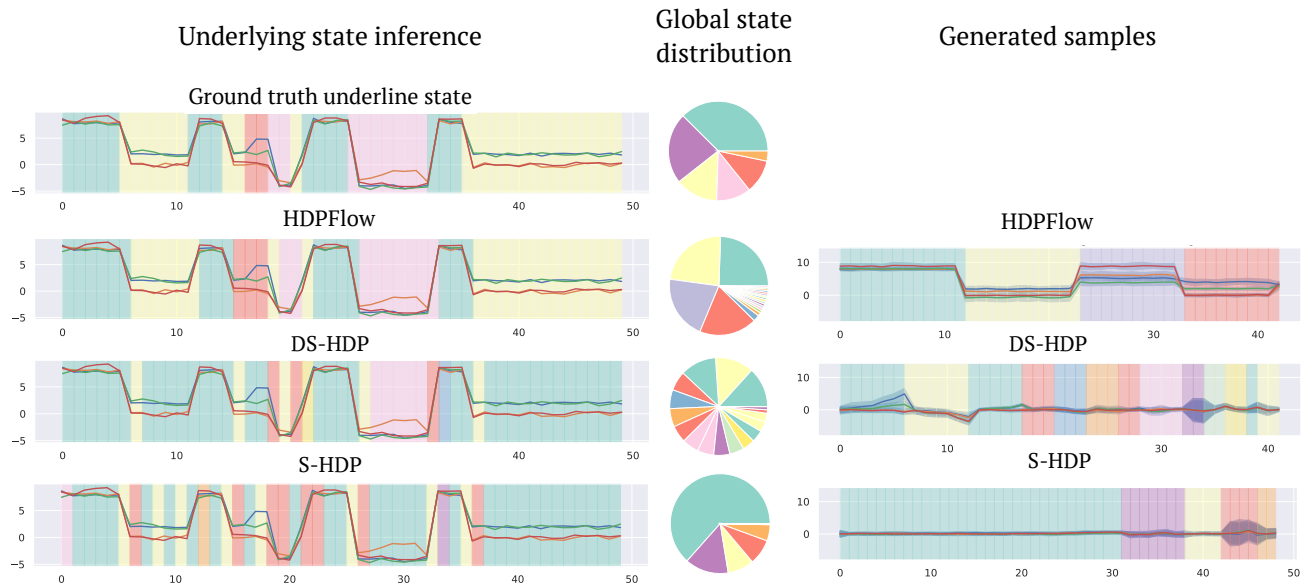

Figure 16.4: Ground Truth vs. BNP models inference on **Simulated Data II**.

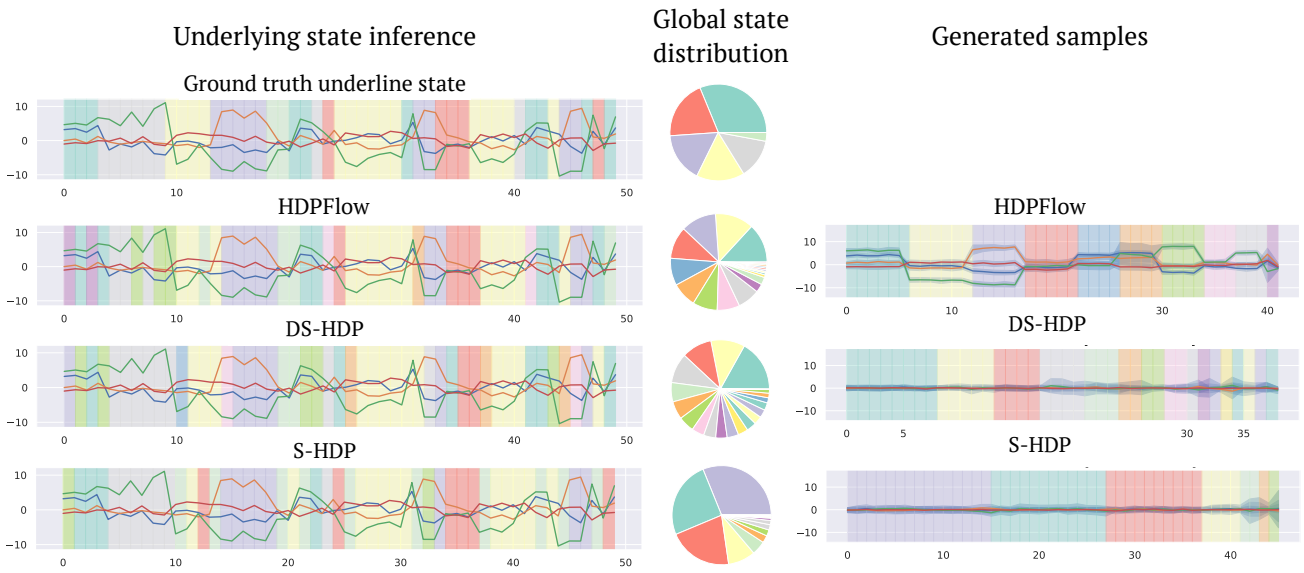

Figure 16.5: Ground Truth vs. BNP models inference on **Simulated data III**.

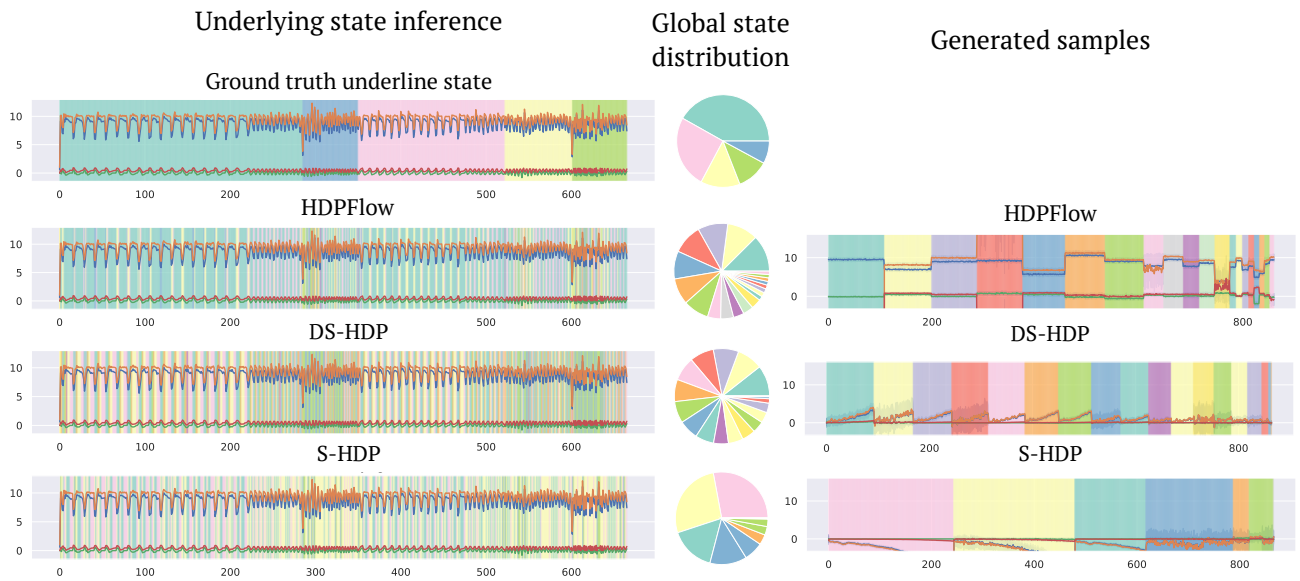

Figure 16.6: Ground Truth vs. BNP models inference on **CPAP** dataset.

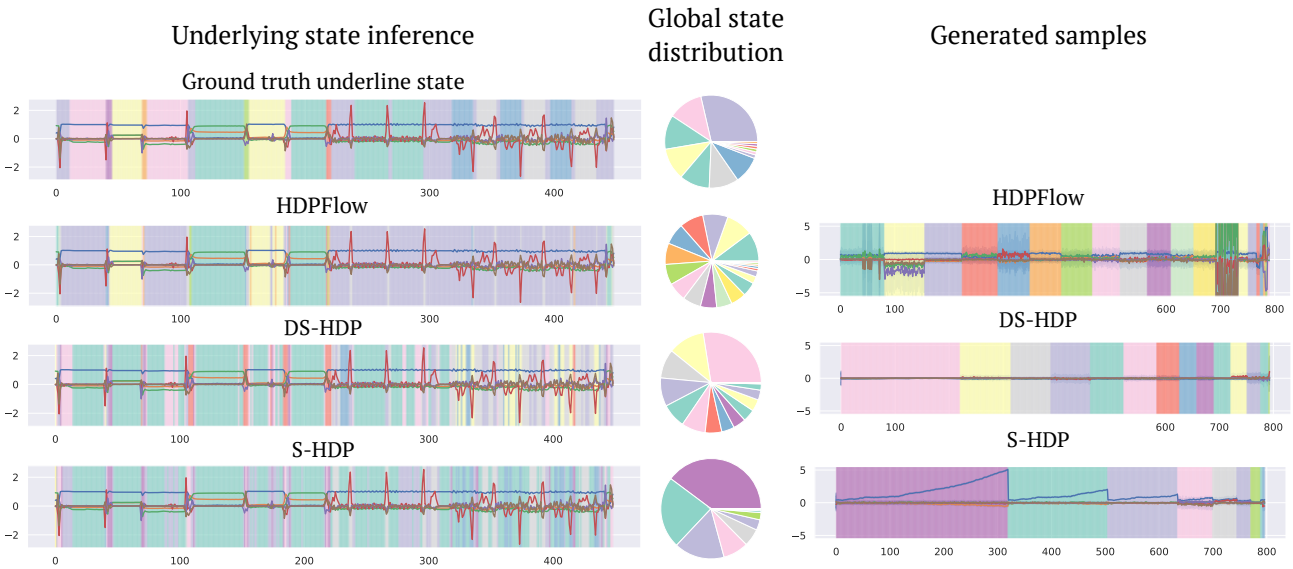

Figure 16.7: Ground Truth vs. BNP models inference on **HAR** dataset.

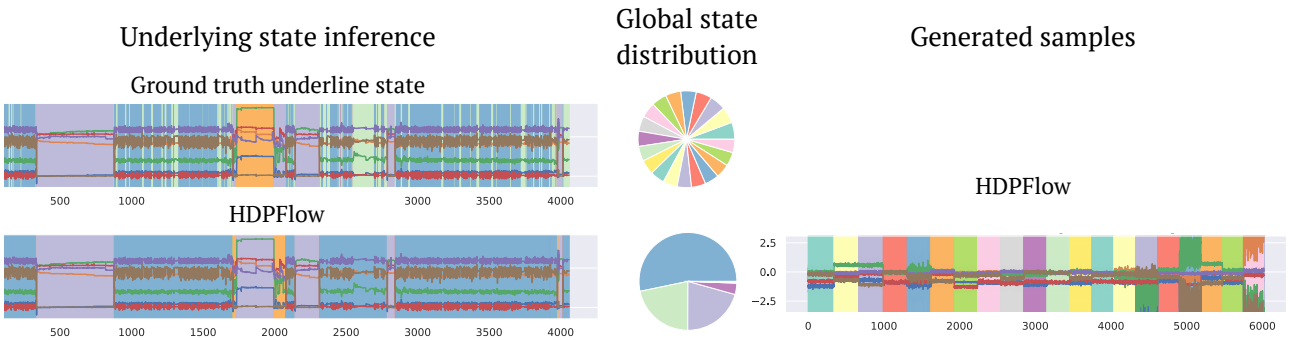

Figure 16.8: Ground Truth vs. BNP models inference on **HAR70** dataset.

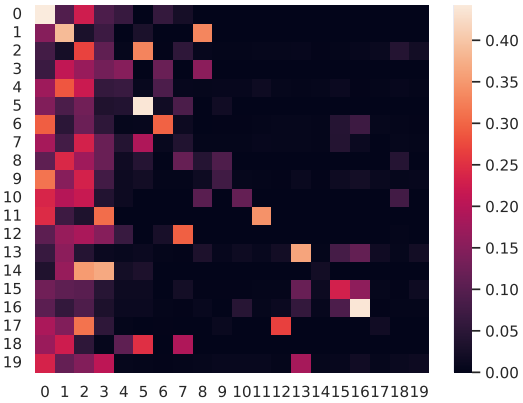

Figure 16.9: Transition probabilities learned for the **Simulated dataset I** by HDP-Flow. The states are ordered from high to low probability determined by β.

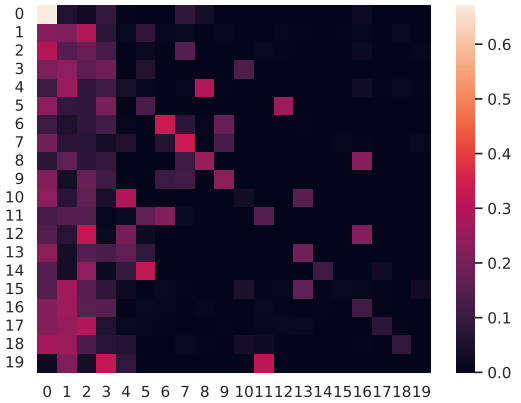

Figure 16.10: Transition probabilities learned for the **Simulated dataset II** by HDP-Flow. The states are ordered from high to low probability determined by β.

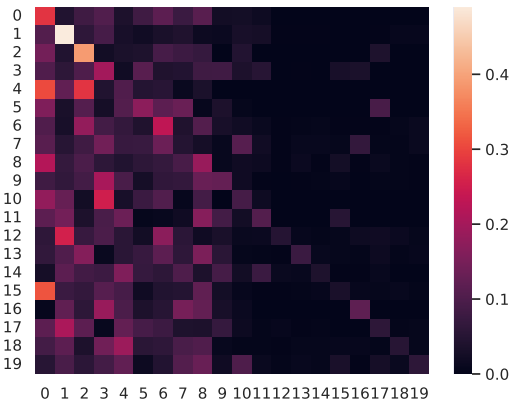

Figure 16.11: Transition probabilities learned for the **Simulated dataset III** by HDP-Flow. The states are ordered from high to low probability determined by β.

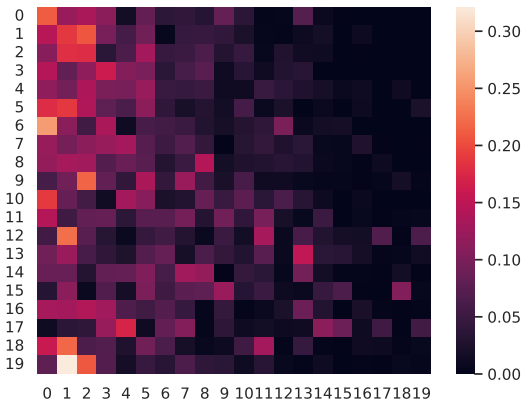

Figure 16.12: Transition probabilities learned for the **HAR** by HDP-Flow. The states are ordered from high to low probability determined by β.