Graph Gamma Process Generalized Linear Dynamical Systems

Rahi Kalantari, Mingyuan Zhou
The University of Texas at Austin
Austin, TX 78712

July 24, 2020

Abstract

We introduce graph gamma process (GGP) linear dynamical systems to model real-valued multivariate time series. For temporal pattern discovery, the latent representation under the model is used to decompose the time series into a parsimonious set of multivariate sub-sequences. In each sub-sequence, different data dimensions often share similar temporal patterns but may exhibit distinct magnitudes, and hence allowing the superposition of all sub-sequences to exhibit diverse behaviors at different data dimensions. We further generalize the proposed model by replacing the Gaussian observation layer with the negative binomial distribution to model multivariate count time series. Generated from the proposed GGP is an infinite dimensional directed sparse random graph, which is constructed by taking the logical OR operation of countably infinite binary adjacency matrices that share the same set of countably infinite nodes. Each of these adjacency matrices is associated with a weight to indicate its activation strength, and places a finite number of edges between a finite subset of nodes belonging to the same node community. We use the generated random graph, whose number of nonzero-degree nodes is finite, to define both the sparsity pattern and dimension of the latent state transition matrix of a (generalized) linear dynamical system. The activation strength of each node community relative to the overall activation strength is used to extract a multivariate sub-sequence, revealing the data pattern captured by the corresponding community. On both synthetic and real-world time series, the proposed nonparametric Bayesian dynamic models, which are initialized at random, consistently exhibit good predictive performance in comparison to a variety of baseline models, revealing interpretable latent state transition patterns and decomposing the time series into distinctly behaved sub-sequences.

1 Introduction

Linear dynamical systems (LDSs) have been widely used to model real-valued time series (Kalman, 1960; West and Harrison, 1997; Ghahramani and Roweis, 1999; Ljung, 1999), with diverse applications such as financial time series analysis (Carvalho and Lopes, 2007) and movement trajectory modeling (Gao et al., 2016; Zhang et al., 2017). They have become standard tools in optimal filtering, smoothing, and control (Imani and Braga-Neto, 2018; Hardt et al., 2018; Koyama, 2018). An LDS consists of two main blocks, including an observation model, which assumes that the observations are

translated from their latent states via a linear system with added Gaussian noise, and a transition block, which is represented by a Markov chain that linearly transforms a latent state from time $t-1$ to time t with added Gaussian noise. The transition block plays an important role in capturing the underlying dynamics of the data. An LDS, which has limited representation power due to its linear assumption, allows one to examine the temporal trajectory of each latent dimension to understand the role played by the corresponding latent factor. While it is often considered to be simple to interpret, its interpretability often quickly deteriorates as its latent state dimension increases.

To enhance the representation power of LDSs, in particular, to model non-linear behaviors of the time series and improve their interpretability, one may consider switching LDSs (Fox et al., 2009; Linderman et al., 2017; Nassar et al., 2018), which learn how to divide the time series into separate temporal segments and fit them by switching between different LDSs. Important parameters include the number of different LDSs, their latent state dimensions, and the transition mechanism from one LDS to another. While nonparametric Bayesian techniques have been applied to switching LDSs to learn the number of LDSs that is needed, the latent state dimensions often stay as tuning parameters to be set (Fox et al., 2009; Nassar et al., 2018). Moreover, switching LDSs do not allow different LDSs to share latent states, making it difficult to capture smooth transitions between different temporal patterns, and false positives/negatives and delays in detecting the switching points will also compromise their performance. In addition, existing optimal smoothing and filtering techniques developed for LDSs, such as Kalman filtering (Kalman, 1960), require non-trivial modifications before being able to be applied to switching LDSs (Murphy, 1998).

Moving beyond switching LDSs where different LDSs neither share their latent states nor overlap in time, we propose the graph gamma process (GGP) LDS that encourages forming multiple LDSs that can share their latent states and co-occur at the same time. GGP-LDS uses a flexible combination of multiple LDSs to fit the observation at any given time point, allowing smooth transitions between different dynamical patterns across time. A notable feature of GGP-LDS is that existing optimal filtering and smoothing techniques developed for a canonical LDS can be readily applied to GGP-LDS without any modification. Therefore, GGP-LDS can serve as a plug-in replacement of the LDS in an existing system. The introduced nonparametric Bayesian construction in GGP-LDS will support S latent states that are shared by K different types of LDSs, each of which is characterized by its own pattern of activation probabilities imposed on a $S \times S$ sparse state-transition matrix, and allow both S and K to grow without bound. This unique construction is realized by modeling the sparsity structure of the $S \times S$ state-transition matrix as the adjacency matrix of a directed random graph, which is resulted from the logical OR operation over K latent binary adjacency matrices, each of which is drawn according to the interaction strengths between the states (nodes) of a type of LDS (node community). Each latent binary adjacency matrix can be considered as a specific type of state community, which describes a particular type of relationship between latent states, *i.e.*, for a given state of the current time, which states of the previous states will directly influence it, and which states of the future time will be directly influenced by it. While a latent state is associated with all communities, the association strengths can clearly differ. Note that the sparsity pattern of the

state-transition matrix is determined by the logical OR of these community-specific binary adjacency matrices. Therefore, to facilitate interpretation and visualization, one can hard assign a state to a community whose binary adjacency matrix best explains how this state is being influenced by the states of the previous time, or to a community that best explains how this state is influencing the states of the next time.

GGP-LDS allows approximating complex nonlinear dynamics by activating a certain combination of communities to model a particular type of linear dynamics at any given time, and using smooth transitions between overlapping communities to model smooth transitions between distinct linear dynamics. The characteristics of each community can be visualized by reconstructing the observations using the inferred latent representation and a community-specific reweighted latent state-transition matrix, where the weights are determined by the activation strength of that community relative to the combined activation strength of all communities. In addition to modeling real-valued time series, we show how to generalize the observation layer of GGP-LDS to model overdispersed count time series. It is noteworthy to mention that while the LDS (Kalman, 1963) has been chosen as the transition and observation model of GGP-LDS, the proposed GGP can potentially be applied to many other nonlinear systems that have a latent state transition module (Johnson et al., 2016).

2 Nonparametric Bayesian Modeling

For LDSs, let us denote $\boldsymbol{y}_t \in \mathbb{R}^V$ and $\boldsymbol{x}_t \in \mathbb{R}^S$ as the observed data and latent state vectors, respectively, at time $t \in \{1, \ldots, T\}$, $\mathbf{D} = (\boldsymbol{d}_1, \ldots, \boldsymbol{d}_S) \in \mathbb{R}^{V \times S}$ as the observation factor loading matrix, and both $\boldsymbol{\Phi} \in \mathbb{R}^{V \times V}$ and $\boldsymbol{\Lambda} = \text{diag}(\lambda_1, \ldots, \lambda_S)$ as precision (inverse covariance) matrices. Inspired by Kalantari et al. (2018), we first modify the usual LDS hierarchical model by utilizing a spike-and-slab construction (Mitchell and Beauchamp, 1988; Ishwaran and Rao, 2005; Zhou et al., 2009), which imposes binary mask $\mathbf{Z} \in \{0, 1\}^{S \times S}$ element-wise on the real-valued latent state transition matrix $\mathbf{W} \in \mathbb{R}^{S \times S}$ as

$$\boldsymbol{y}_t \sim \mathcal{N}(\mathbf{D}\boldsymbol{x}_t, \boldsymbol{\Phi}^{-1}),\ \boldsymbol{x}_t \sim \mathcal{N}\left((\mathbf{W} \odot \mathbf{Z})\, \boldsymbol{x}_{t-1}, \boldsymbol{\Lambda}^{-1}\right).$$

For count observations, we modify the Gaussian distribution based observation layer with the negative binomial (NB) distribution by letting $\boldsymbol{y}_t \sim \text{NB}\left(\eta, \sigma(\mathbf{D}\boldsymbol{x}_t)\right),\ \eta \sim \text{Gamma}(\alpha_\eta, 1/\beta_\eta),\ \boldsymbol{x}_t \sim \mathcal{N}\left((\mathbf{W} \odot \mathbf{Z})\, \boldsymbol{x}_{t-1}, \boldsymbol{\Lambda}^{-1}\right)$, where $\sigma(x) := 1/(1 - e^{-x})$ is the sigmoid function.

The $K \times K$ latent state-transition matrix $\mathbf{W} \odot \mathbf{Z}$, in particular, the sparsity structure of $\mathbf{Z}$, plays an important role in determining the dynamical behaviors of the model. First, the nonzero locations in $\mathbf{Z}$ determine the temporal dependencies between the latent states (Kalantari et al., 2018). For example, if z_{ij}, the (i, j)th element of $\mathbf{Z}$, is zero, then at time t, x_{ti} will be independent of $x_{(t-1)j}$, and x_{tj} will not influence $x_{(t+1)i}$. Thus in what follows, we consider that there is a directed link (edge) from states (nodes) j to i if $z_{ij} = 1$.

Second, viewing $\mathbf{Z}$ as the adjacency matrix of a directed random graph and the states of the LDS as the graph nodes, we may introduce inductive bias to encourage its nodes to be formed

into overlapping communities, reflected by overlapping dense blocks along the diagonal of the adjacency matrix after appropriately rearranging the orders of the nodes. We may then view each community as an LDS, which forms its own state-transition matrix, using a submatrix of $\mathbf{W} \odot \mathbf{Z}$, to model the transitions between the corresponding subset of states. This construction allows approximating complex nonlinear dynamics by activating different communities at different levels to model a particular type of linear dynamics at any given time, and using smooth transitions between overlapping communities to model the smooth transitions between distinct linear dynamics.

To induce the structure of overlapping communities into $\mathbf{Z}$, the adjacency matrix of a directed random graph, and allow both the number of communities and number of nodes (dimension of $\mathbf{Z}$) to grow without bound, we propose the graph gamma process (GGP). A draw from the GGP consists of countably infinite latent communities, each of which is associated with a positive weight indicating the overall activation strength of the community. These communities all share the same set of countably infinite nodes (states) but place different weights on how strongly a node is associated with a community. We describe the detail in what follows.

2.1 Graph Gamma Process

Denote $Z(i,:)$ and $Z(:,i)$ as row i and column i of $\mathbf{Z}$, respectively. Since $\mathbb{E}[\boldsymbol{x}_t \,|\, \boldsymbol{x}_{t-1}, \mathbf{W}, \mathbf{Z}] = (\mathbf{W} \odot \mathbf{Z})\boldsymbol{x}_{t-1}$, we have

$$\mathbb{E}[x_{ti} \,|\, \boldsymbol{x}_{t-1}, \mathbf{W}, \mathbf{Z}] = (W(i,:) \odot Z(i,:))\boldsymbol{x}_{t-1}, \tag{1}$$

$$\mathbb{E}[\boldsymbol{x}_{t+1} \,|\, \boldsymbol{x}_t, \mathbf{W}, \mathbf{Z}] = (W(:,i) \odot Z(:,i))x_{ti} + \textstyle\sum_{j\neq i}^{\infty}(W(:,j) \odot Z(:,j))x_{tj}, \tag{2}$$

which means x_{ti} will be dependent on $\boldsymbol{x}_{t-1}$ if $Z(i,:)$ contains non-zero elements, and it will influence $\boldsymbol{x}_{t+1}$ if $Z(:,i)$ contains non-zero elements. To construct a nonparametric Bayesian model that removes the need to tune the hidden state dimension, our first goal is to allow $\mathbf{Z}$ to have an unbounded number of rows and columns, which means that the model can support countably infinite state-specific factors $\boldsymbol{d}_i$, with which the mean of $\boldsymbol{y}_t$ given $\boldsymbol{x}_t$ is factorized as $\mathbb{E}[\boldsymbol{y}_t \,|\, \boldsymbol{x}_t, \mathbf{D}] = \mathbf{D}\boldsymbol{x}_t = \sum_{t=1}^{\infty} \boldsymbol{d}_i x_{ti}$.

To achieve this goal, with $c_\rho > 0$ and $G_{0,\rho}$ defined as a finite and continuous base measure over a complete and separable metric space Ω, we first introduce a gamma process $G_\rho \sim \Gamma\mathrm{P}(c_\rho, G_{0,\rho})$ on the product space $\mathbb{R}^+ \times \Omega$, where $\mathbb{R}^+ := \{x : x > 0\}$, such that for each subset $A \subset \Omega$, we have $G_\rho(A) \sim \mathrm{Gamma}(G_{0,\rho}(A), 1/c_\rho)$. The Lévy measure of this gamma process can be expressed as $\nu(d\rho d\boldsymbol{d}) = \rho^{-1}e^{-c_\rho\rho}d\rho G_{0,\rho}(d\boldsymbol{d})$. A draw from this gamma process can be expressed as $G_\rho = \sum_{i=1}^{\infty}\rho_i d\boldsymbol{d}_i$, consisting of countably infinite atoms (factors) $\boldsymbol{d}_i$ with weights ρ_i. We view $\boldsymbol{d}_i$ as the factor loading vector for latent state i, and will make ρ_i determine the number of nonzero elements in $Z(i,:)$ and, consequently, how strongly x_{ti}, the activation of state i at time t, is influenced by $\boldsymbol{x}_{t-1}$ of the previous time. As the number of ρ_i that are larger than an arbitrarily small constant ϵ follows a Poisson distribution with a finite mean as $\gamma_{0,\rho}\int_\epsilon^\infty \rho^{-1}e^{-c_\rho\rho}d\rho$, where $\gamma_{0,\rho} := G_{0,\rho}(\Omega)$ is the mass parameter, this can be used to express the idea that only a finite number of elements in $\{x_{ti}\}_{i=1,\infty}$ at time t will be dependent on $\boldsymbol{x}_{t-1}$ of the previous time.

We further mark each ρ_i with a degenerate gamma random variable, changing the Lévy measure of the gamma process to that of a marked gamma process (Kingman, 1993) as $\nu(d\rho\, d\boldsymbol{d}\, d\tau) = \rho^{-1}e^{-c_\rho \rho}d\rho G_{0,\rho}(d\boldsymbol{d})\gamma_{0,\tau}\tau^{-1}e^{-c_\tau\tau}d\tau$; we express a draw from this marked gamma process as $G_{\rho,\tau} = \sum_{i=1}^{\infty}(\rho_i,\tau_i)\delta_{\boldsymbol{d}_i}$. We will make τ_i determine the random number of nonzero elements in $Z(:,i)$ and, consequently, how strongly x_{ti}, the factor score of state i at time t, will influence $\boldsymbol{x}_{t+1}$ of the next time point. As the number of τ_i that are larger than an arbitrarily small constant ϵ follows a Poisson distribution with a finite mean as $\gamma_{0,\tau}\int_\epsilon^\infty \tau^{-1}e^{-c_\tau\tau}d\tau$, this can be used to express the idea that only a finite number elements in $\{x_{ti}\}_{i=1,\infty}$ at time t will influence $\boldsymbol{x}_{t+1}$.

Given $G_{\rho,\tau} = \sum_{i=1}^{\infty}(\rho_i,\tau_i)\delta_{\boldsymbol{d}_i}$, we further define a gamma process $G_o \sim \Gamma\mathrm{P}(c_o, G_{\rho,\tau})$, with Lévy measure $\nu(drd\boldsymbol{\theta}d\boldsymbol{\psi}) = r^{-1}e^{-cr}drG_o(d\boldsymbol{\theta}d\boldsymbol{\psi})$, a draw from which is expressed as $G_o = \sum_{\kappa=1}^{\infty} r_\kappa\delta_{\{\boldsymbol{\theta}_\kappa,\boldsymbol{\psi}_\kappa\}}$. In this random draw, $r_\kappa \in \mathbb{R}_+$, reflecting the activation strength of community κ, is the weight of the κth atom $\{\boldsymbol{\theta}_\kappa,\boldsymbol{\psi}_\kappa\}$, where $\boldsymbol{\theta}_\kappa = (\theta_{1\kappa},\ldots,\theta_{\infty\kappa})^T$, $\boldsymbol{\psi}_\kappa = (\psi_{1\kappa},\ldots,\psi_{\infty\kappa})^T$, and $\theta_{i\kappa}$ and $\psi_{i\kappa}$, representing how strongly that node i is associated with community κ, are defined on ρ_i and τ_i, the weights of the atoms of the gamma process $G_{\rho,\tau}$, using

$$\theta_{i\kappa} \sim \mathrm{Gamma}(\rho_i, 1/e), \quad \psi_{i\kappa} \sim \mathrm{Gamma}(\tau_i, 1/f).$$

We refer to the hierarchical stochastic process constructed in this way as the GGP. We denote the mass parameter of the GGP as $\gamma_0 := \int G_o(d\boldsymbol{\theta}d\boldsymbol{\psi})$. Inherited from the property of a gamma process, the GGP has an inherent shrinkage mechanism that its number of atoms (node communities) with weights greater than $\epsilon > 0$ is a finite random number drawn from $\mathrm{Pois}(\gamma_0\int_\epsilon^\infty r^{-1}e^{-cr}dr)$.

Given a random draw from the GGP as $G_o = \sum_{\kappa=1}^{\infty} r_\kappa\delta_{\{\boldsymbol{\theta}_\kappa,\boldsymbol{\psi}_\kappa\}}$, we will let r_κ determine the overall activation strength of community κ, $\theta_{i\kappa}$ how strongly state i in community κ is influenced by the states of the previous time in the same community, and $\psi_{j\kappa}$ how strongly state j in community κ influences the states of the next time in the same community. To express this idea, for community κ parameterized by $\{r_\kappa,\boldsymbol{\theta}_\kappa,\boldsymbol{\psi}_\kappa\}$, we generate a community-specific sparse adjacency matrix, whose (i,j)th element is drawn as

$$z_{ij\kappa} \sim \mathrm{Bernoulli}(1 - e^{-r_\kappa\theta_{i\kappa}\psi_{j\kappa}}). \tag{3}$$

Thus from nodes j to i, community κ defines its own interaction probability, expressed as $p_{ij\kappa} = 1 - e^{-r_\kappa\theta_{i\kappa}\psi_{j\kappa}}$, and draws a binary edge $z_{ij\kappa}$ based on $p_{ij\kappa}$. While there are countably infinite nodes, in community κ, the total number of edges is a finite random number and hence the number of nodes with nonzero degrees is also finite.

Lemma 1. *The number of edges in community κ, expressed as $\sum_{i=1}^{\infty}\sum_{j=1}^{\infty} z_{ij\kappa}$, is finite.*

As in (3), whether $z_{ij\kappa} = 1$ or 0 is related to both the overall strength of community κ and how strongly nodes i and j are affiliated with community κ. Lemma 1, whose proof is deferred to the Appendix, suggests that we can extract a finite submatrix $\mathcal{Z}_\kappa := \{z_{ij\kappa}\}_{i,j\in\mathcal{S}_\kappa}$, where $\mathcal{S}_\kappa := \{i : \sum_j z_{ij\kappa} + \sum_j z_{ji\kappa} > 0\}$ is the set of nodes with non-zero degrees in community κ. We

consider $\mathcal{S}_\kappa$ as the nodes activated by community κ and $\mathcal{Z}_\kappa$ as its nonempty graph adjacency matrix. Thus under the proposed GGP construction, different communities could overlap in the nodes belonging to their respective nonempty graph adjacency matrices, which means it is possible that $\mathcal{S}_\kappa \cap \mathcal{S}_{\kappa'} \neq \emptyset$ for $\kappa \neq \kappa'$. If $\mathcal{S}_\kappa \cap \mathcal{S}_{\kappa'} = \emptyset$, then we consider communities κ and κ' as two non-overlapping communities.

Our previous analysis tells that whether $z_{ij} = 1$ determines not only whether state i at a given time will be dependent of the states of the previous time, but also whether state j at a given time will influence the states of the next time. To express the idea that whether $z_{ij} = 1$ is collectively decided by all countably infinite communities, whose nonempty adjacency matrices could overlap in their selections of nodes, we take the OR operation over all elements in $\{z_{ij\kappa}\}_\kappa$ to define the adjacency matrix of the full model as

$$z_{ij} = \vee_{\kappa=1}^{\infty} z_{ij\kappa}, \tag{4}$$

which means $z_{ij} = 1$ if at least one $z_{ij\kappa} = 1$, indicating community κ places a directed edge from nodes j to i, and $z_{ij} = 0$ otherwise. In a matrix format, we have

$$\mathbf{Z} = \vee_{\kappa=1}^{\infty} \mathbf{Z}^{(\kappa)},$$

where $\mathbf{Z}^{(\kappa)}$ represents the graph adjacency matrix of community κ, whose (i, j)th element is $z_{ij\kappa}$.

We note that marginalizing out $\{z_{ij\kappa}\}_\kappa$, we can directly draw the graph adjacency matrix defined in (4) as

$$\mathbf{Z} \sim \text{Bernoulli}(1 - e^{-\sum_{\kappa=1}^{\infty} r_\kappa \boldsymbol{\theta}_k \boldsymbol{\psi}_k^T}), \tag{5}$$

which can also be equivalently generated under the Bernoulli-Poisson link (Zhou, 2015) as

$$\mathbf{Z} = \delta(\mathbf{M} \geq 1),\ \mathbf{M} = \textstyle\sum_{\kappa=1}^{\infty} \mathbf{M}_\kappa,\ \mathbf{M}_\kappa \sim \text{Pois}(r_\kappa \boldsymbol{\theta}_\kappa \boldsymbol{\psi}_\kappa^T),$$

where $\delta(\cdot)$ returns one if the condition is true and zero otherwise. While the graph defined by $\mathbf{Z}$ has countably infinite nodes, the total number of edges is finite and hence the number of nodes with nonzero degrees is also finite; the proof of the following Lemma is deferred to the Appendix.

Lemma 2. *The number of edges in* $\mathbf{Z}$*, expressed as* $\sum_{i=1}^{\infty} \sum_{j=1}^{\infty} z_{ij}$*, is finite.*

In summary, the GGP uses a gamma process to support countably infinite node communities in the prior, and another marked gamma process to support countably infinite number of nodes (states) shared by these communities. The adjacency matrix of the generated random graph from the GGP can be either viewed as taking the OR operation over all community-specific binary adjacency matrices, or viewed as thresholding a latent count matrix that aggregates the activation strengths across all communities for each node pair. Under this model construction, with the inherent shrinkage mechanisms of the gamma processes, only a finite number of communities will contain

edges between the nodes, and the nonempty communities overlap with each other on their selections of nonzero-degree nodes, the total number of which across all communities is finite.

2.2 Overlapping Community based Model Interpretation and Visualization

As in (5), the sparsity pattern $\mathbf{Z}$ of the latent state-transition matrix can be linked to a latent count matrix whose expectation is $\sum_\kappa r_\kappa \boldsymbol{\theta}_\kappa \boldsymbol{\psi}_\kappa^T$. Thus we can measure the activation strength of community k relative to the combined activation strength of all communities using

$$\mathbf{A}_\kappa := \frac{r_\kappa \boldsymbol{\theta}_\kappa \boldsymbol{\psi}_\kappa^T}{\sum_{\kappa'=1}^K r_{\kappa'} \boldsymbol{\theta}_{\kappa'} \boldsymbol{\psi}_{\kappa'}^T}. \tag{6}$$

Due to the shrinkage property of the gamma process, only a small number of r_κ are encourage to have non-negligible values. Therefore, there is often only a parsimonious set of active communities in the posterior. Since $\sum_\kappa \mathbf{A}_\kappa$ is a matrix whose elements are all equal to one, we can use the $\mathbf{A}_\kappa$'s to decompose the reconstruction of the time series under the proposed GGP-LDS into different sub-sequences, the κ^{th} of which reveals the type of temporal patters captured by community κ.

More specifically, given the model parameters and latent state representation $\{\boldsymbol{x}_t\}_t$, taken from a posterior sample inferred by GGP-LDS, we define

$$\hat{\boldsymbol{y}}_t^{(\kappa)} = \mathbf{D}\hat{\boldsymbol{x}}_t^{(\kappa)}, \quad \hat{\boldsymbol{x}}_t^{(\kappa)} = [(\mathbf{W} \odot \mathbf{Z}) \odot \mathbf{A}_\kappa]\boldsymbol{x}_{t-1}. \tag{7}$$

Note we have $\mathbb{E}[\boldsymbol{x}_t \,|\, \mathbf{W}, \mathbf{Z}, \boldsymbol{x}_{t-1}] = \sum_\kappa \hat{\boldsymbol{x}}_t^{(\kappa)}$ and $\mathbb{E}[\boldsymbol{y}_t \,|\, \mathbf{D}, \mathbf{W}, \mathbf{Z}, \boldsymbol{x}_{t-1}] = \sum_\kappa \hat{\boldsymbol{y}}_t^{(\kappa)}$. Thus we can consider $\hat{\boldsymbol{y}}_{1:T}^{(\kappa)}$ as a sub-sequence in the data space and $\hat{\boldsymbol{x}}_{1:T}^{(\kappa)}$ as a sub-sequence in the latent space, both extracted according to the relative strength of community κ. Our experiments show that different dimensions of sub-sequence $\hat{\boldsymbol{y}}_t^{(\kappa)}$ often resemble each other in temporal patterns, but differ from each other in magnitudes, which is the reason why the aggregation of these sub-sequences, expressed as $\sum_\kappa \hat{\boldsymbol{y}}_t^{(\kappa)}$, can exhibit distinct temporal behaviors at different data dimensions. Note one may also define $\hat{\boldsymbol{x}}_t^{(\kappa)} = (\mathbf{W} \odot \mathbf{Z}^{(\kappa)})\boldsymbol{x}_{t-1}$, which would lead to somewhat different community-specific sub-sequences, whose superposition no longer reconstructs $\mathbb{E}[\boldsymbol{y}_t \,|\, \mathbf{D}, \mathbf{W}, \mathbf{Z}, \boldsymbol{x}_{t-1}]$ but may help highlight transitional time points between different linear dynamics.

To visualize the overlapping community structure of the latent states inferred by the proposed GGP-(G)LDS, we order the communities in decreasing oder based on their overall activation strength, which, for example, can be measured by either $\|\mathbf{Z}_\kappa\|_0 = \sum_i \sum_j z_{ij\kappa}$ or $\|\mathbf{M}_\kappa\|_1 = \sum_i \sum_j m_{ij\kappa}$. In this paper, we rank the activation strength according to $\|\mathbf{M}_\kappa\|_1$. As in (1), $Z(i,:)$ determines how latent state i at the current time is influenced by the latent states of the previous time, and $Z(i,:)$ can be generated by thresholding latent counts $M(i,:) = \sum_\kappa M_\kappa(i,:)$. Thus we can we map row i to community

$$\pi_{row}(i) = \arg\max_\kappa \textstyle\sum_j m_{ij\kappa},$$

the primary community via which latent state i is influenced by the latent states of the previous time. Similarly, as in (2), $Z(:,j)$ determines how latent state j at the current time is influencing the

latent states of the next time, we can map column j to community

$$\pi_{col}(j) = \arg\max_\kappa \sum_i m_{ij\kappa},$$

the primarily community via which latent state j influences the the latent states of the next time. To help visualize $\mathbf{Z}$, we rearrange its rows such that the rows mapped to the same community according to $\pi_{row}(i)$ are placed in the same row block, and we rearrange the columns in the same way according to $\pi_{col}(j)$.

For the rows inside the same row block (*i.e.*, sharing the same $\pi_{row}(i)$), we rearrange them in decreasing order based on $\sum_j m_{ij\pi_{row}(i)}$; we follow the same way to rearrange the columns inside each column block according to $\sum_i m_{ij\pi_{col}(j)}$.

Note that $m_{ij\kappa}$ are augmented count variables in a posterior sample, which are introduced to provide closed-form Gibbs sampling update equations, as described below.

2.3 Hierarchical Model and Bayesian Inference

To facilitate implementation, we truncate the GGP by setting K as an upper-bound of the number of communities, and S as an upper-bound of the number of states (nodes). We set $\gamma_{0,\rho} = \gamma_{0,\tau} = \gamma_0$. We make the scales of $\theta_{i\kappa}$ and $\psi_{j\kappa}$ change with κ to increase model flexibility. The hierarchical model of the truncated GGP-LDS is expressed as

$$\begin{aligned}
&\boldsymbol{y}_t \sim \mathcal{N}(\mathbf{D}\boldsymbol{x}_t,\, \mathbf{\Phi}^{-1}), \quad \mathbf{\Phi} \sim \text{Wishart}(\mathbf{V}, V+2),\\
&\boldsymbol{x}_t \sim \mathcal{N}\left[(\mathbf{W}\odot\mathbf{Z})\boldsymbol{x}_{t-1},\, \text{diag}(\lambda_1,\ldots,\lambda_S)^{-1}\right],\\
&\boldsymbol{d}_s \sim \mathcal{N}\left(\mathbf{0},\, \mathbf{I}_V/\sqrt{V}\right), \quad \lambda_s \sim \text{Gamma}\left(a, 1/b\right),\\
&w_{ij} \sim \mathcal{N}(0, \varphi_{ij}^{-1}), \quad \varphi_{ij} \sim \text{Gamma}\left(\alpha_0, 1/\beta_0\right),\\
&z_{ij} = \vee_{\kappa=1}^K z_{ij\kappa}, \quad z_{ij\kappa} = \delta(m_{ij\kappa} \geq 1),\\
&m_{ij\kappa} \sim \text{Pois}\left(r_\kappa\theta_{i\kappa}\psi_{j\kappa}\right), \quad r_\kappa \sim \text{Gamma}(\gamma_0/K, 1/c),\\
&\theta_{i\kappa} \sim \text{Gamma}\left(\rho_i, 1/e_\kappa\right), \quad \rho_i \sim \text{Gamma}\left(\gamma_0/S, 1/c_\rho\right),\\
&\psi_{j\kappa} \sim \text{Gamma}\left(\tau_j, 1/f_\kappa\right), \quad \tau_j \sim \text{Gamma}\left(\gamma_0/S, 1/c_\tau\right),
\end{aligned} \tag{8}$$

where $e_\kappa, f_\kappa \sim \text{Gamma}\left(\alpha_0, 1/\beta_0\right)$ and $\boldsymbol{x}_0 \sim \mathcal{N}(\boldsymbol{m}_0, \mathbf{H}_0)$. As in Lemma 2, the total number of nonzero elements in $\mathbf{Z}$ has a finite expectation. Thus if the GGP truncation levels K and S are set large enough, it is expected for some state i that $\sum_j z_{ij} = 0$, which means its corresponding row in $\mathbf{Z}$ has no nonzero elements, and/or $\sum_j z_{ji} = 0$, which means its corresponding column in $\mathbf{Z}$ has no nonzero elements. If node i has zero degree that $\sum_j z_{ij} = \sum_j z_{ji} = 0$, then x_{ti} will neither be dependent on $\boldsymbol{x}_{t-1}$ nor influence $\boldsymbol{x}_{t+1}$, which means $\{x_{ti}\}_t$, the factor scores of state i, capture only the non-dynamic noise component of the data. Moreover, the proposed model will penalize the total energy captured by zero-degree node (state) i, expressed as $\sum_{t=1}^T x_{ti}^2$ if it is a zero-degree node (see Appendix B for more details).

To accommodate count data that are often overdispersed, we replace the Gaussian distribution based observation layer of GGP-LDS, *i.e.*, the first row of (8), with a negative binomial distribution based observation layer as

$$\boldsymbol{y}_t \sim \mathrm{NB}\left(\eta, \sigma\left(\mathbf{D}\boldsymbol{x}_t\right)\right), \quad \eta \sim \mathrm{Gamma}\left(\alpha_\eta, 1/\beta_\eta\right), \quad \alpha_\eta, \beta_\eta \sim \mathrm{Gamma}\left(\alpha_0, 1/\beta_0\right), \tag{9}$$

while keeping the other parts of the model unchanged; we refer to the modified model for count observation as GGP generalized LDS (GGP-GLDS). Due to the use of the negative binomial distribution, we may also refer to the modified model as GGP negative binomial dynamical system (GGP-NBDS).

We perform Bayesian inference via Gibbs sampling. Exploiting a variety of data augmentation and marginalization techniques developed for discrete data (Zhou and Carin, 2013; Zhou, 2015) , we provide closed-form Gibbs sampling updated equations for all model parameters, as described in detail in Appendix C. Unless specified otherwise, we consider 6000 Gibbs sampling iterations, treat the first 3000 samples as burnin, and collect one sample per 60 iterations afterwards, resulting in a collection of 50 posterior MCMC samples that are used to predict the means and uncertainty of future observations. In the following section, we provide a review of related work, where we compare the proposed models against a variety of dynamical systems, including switching LDSs and autoregressive, nonparametric, and deep neural network based models.

3 Related Work

The main challenges for modeling time series include providing accurate forecast, having the ability to process missing data, and learning interpretable latent representation such that latent structures (*e.g.*, clusters) can be translated into meaningful data temporal patterns, representing different time series behaviors, without the need to perform data specific tuning. A time series method addressing all these challenges is desired to have the following properties: 1) The underlying model is neither over- nor under-parameterized, which will result in over- or under-fitting, respectively, and hence poor performance in forecasting (Liu and Hauskrecht, 2015; Kalantari et al., 2018). 2) The model learns a parsimonious set of latent states such that the induced latent structures can explain the underlying patterns of the time series without the need of manual tuning. Inducing too many (overlapping) clusters of the latent states makes the model difficult to interpret, while imposing heavy regularization can enhance interpretability but hinder prediction. A good model will be able to balance interpretability and predictive performance. 3) The model is capable of modeling non-linear dynamics exhibited by the time series. 4) The model is capable of capturing latent state transition behaviors and the uncertainty of latent parameters. In this section, we review related works and discuss whether they satisfy the aforementioned properties.

Different from LDSs that employ latent state-transition blocks, autoregressive models that directly regress on the past observations are also commonly used to model real-valued time series data (Harrison et al., 2003; Davis et al., 2016; Saad and Mansinghka, 2018), with a wide range

of applications, such as health care and epidemic modeling (Li et al., 2010; Kennedy and Turley, 2011). In addition to LDSs and autoregressive models, there also exist several other types of parametric models (Barber et al., 2011). To achieve good performance on a given time series, these parametric models in general require searching over a large set of possible parameter settings or model configurations via cross validation, or appropriately regularizing the training.

A variety of regularization techniques have been proposed for parameteric time series models. Charles et al. (2011) introduce a sparsity constraint on the update equations of Kalman filtering to enforce the latent state vectors to be sparse, but need to assume that all the other model parameters, including the observation and transition matrices and noise covariance matrices, are known. Städler et al. (2013) encourage a hidden Markov model (HMM) to have sparse state-specific inverse covariance matrices by imposing L_1-penalties on their elements. Liu and Hauskrecht (2015) propose to regularize the state transition matrix by penalizing its nuclear norm and imposing multivariate Laplacian priors over its rows. Siddiqi et al. (2010) propose a solution on reduced-rank HMM by relaxing the assumptions of a spectral learning algorithm by learning a K-dimensional subspace and finding the mapping between the high dimensional and low dimensional spaces.

Another set of time-series models are nonparametric Bayesian switching LDSs (Fox et al., 2009; Linderman et al., 2017), in which every temporal segment of the time series is fitted by one LDS. These models are focused on finding a mixture of LDSs, which are used to fit different time series segments, and a switching mechanism between different LDSs is learned to model the transitions between segments. Switching LDSs, however, may not provide satisfactory predictive performance on test data, as false switching, missed switching, and delayed switching could all compromise their predictions. Chiuso and Pillonetto (2010) design another type of nonparametric Bayesian models that identify sparse linear systems. Unlike the proposed GGP-LDS, it assumes no latent state transitions and models each observation as a linear combination of previous observations and some external input.

In addition, there are models that use the hierarchical Dirichlet process (Teh et al., 2006) priors over the states in hidden Markov models (Johnson and Willsky, 2013; Fox et al., 2009). There are also models that perform clustering on the time series use a Pitman-Yor process based mixture prior on non-linear state-space models (Nieto-Barajas et al., 2014), and Dirichlet process mixtures (Caron et al., 2008) for modeling noise distributions. These models are not fully nonparametric as they typically have some parametric assumptions as part of the model such as having a fixed number of hidden states or imposing explicit specifications of the underlying temporal dynamics, such as seasonality and trends.

Chiuso and Pillonetto (2010) design a nonparametric Bayesian model to identify sparse linear systems. It assumes no latent transitions and believes each observation is a linear combination of previous observations plus some external input. Saad and Mansinghka (2018) introduce a recurrent Chinese restaurant process based mixture to capture temporal dependencies and a hierarchical prior to discover groups of time series whose underlying dynamics are modeled jointly. This model is able to cluster the observations to a set of trajectories with similar behaviors, although it is prone

to creating unnecessary clusters as if the same pattern repeats with different magnitudes in two different segments of the observation, these two segments are likely to be assigned to two different clusters. This may result in many unnecessary clusters for high dimensional and/or lengthy data.

Another widely used type of time series models are autoregressive models (Harrison et al., 2003; Davis et al., 2016; Saad and Mansinghka, 2018). There also exist several other parametric models, such as Barber et al. (2011), that provide additional tools to model time series. Most of these parametric models require searching over a large set of possible parameter settings with cross validation or model configurations to achieve satisfactory performance.

On the other side of the spectrum, we have a set of neural network based solutions, such as recurrent neural network based systems (Flunkert et al., 2017; Lai et al., 2018), which are quite flexible but suffer from several major limitations. They often provide point estimate of their parameters, without uncertainty estimation, and often have the problem of interpretability and need considerable amount of data to reliably train the model.

In this paper, we introduce a nonparametric Bayesian hierarchical model to address the aforementioned challenges. The proposed model is based on the LDS and GGP. The proposed GGP is designed to learn overlapping communities of latent states such that each community models a behavior which can be described with a linear system. Instead of assigning one community (LDS) to one segment of the observed trajectory, our model allows multiple communities to be used at any time point. This allows the model to break the sophisticated behavior in a trajectory to a weighted combination of simpler behaviors modeled by different linear systems, and it helps to model the nonlinearities of the data using multiple linear systems and smooth transitions between them.

4 Experimental Results

The code is provided at: https://github.com/GGPGLDS/GGP_GLDS. In this section, we will demonstrate the interpretablity of GGP-(G)LDS and its predictive performance on several different datasets. Due to the inherent shrinkage mechanisms of the GGP, we find that the proposed nonparametric Bayesian model is not sensitive to the choice of the truncation levels S and K as long as they are set large enough. For all the datasets in this section, we truncate them at $K = 16$ and $S = 30$, which are found to be large enough to accommodate all nonempty node communities, with interpretable latent representation and good predictive performance. Our Gibbs sampling based inference is not sensitive to initialization, allowing us to randomly initialize the model parameters. In this paper, we set $\gamma_0 = \alpha_0 = \beta_0 = c = c_\rho = c_\tau = 1$ for all experiments. We set $a = 1$ and $b = 0.1$ for all experiments (except for all visualizations, we set $b = 1$ to encourage sparser latent state-transition matrices), encouraging λ_s^{-1} to be small and hence encouraging the latent state representation vector to be constituted more by the autoregressive components and less by the white noise, generated by $\mathcal{N}\left[\mathbf{0}, \text{diag}(\lambda_1, \ldots, \lambda_S)^{-1}\right]$.

For GGP-(G)LDS, we measure the performance of t-step prediction, for $t = 1, \ldots, 10$. In addition, 1-step-at-a-time predictive performance will be provided by using Kalman filter to update the state

$\boldsymbol{x}_t$ after observing $\boldsymbol{y}_t$, while other model parameters, including $\mathbf{D}$, $\boldsymbol{\Phi}$, $\mathbf{W}$, $\mathbf{Z}$, and $\boldsymbol{\lambda}$, stay the same to make the next step prediction. Note that 1-step-at-a-time prediction via Kalman filter is only an option readily available for an LDS based system (not including switching LDS). Denoting $\hat{\boldsymbol{y}}_t$ as the t-step prediction of a given model, we measure its t-step predictive performance using mean absolute error (MAE) for real time series as

$$\text{MAE}_t = \sum_{v=1}^{V} |y_{dt} - \hat{y}_{dt}| / V, \tag{10}$$

and use mean absolute relative error (MARE) for count time series as

$$\text{MARE}_t = \sum_{v=1}^{V} \frac{|y_{dt} - \hat{y}_{dt}|}{V(y_{dt} + 1)}. \tag{11}$$

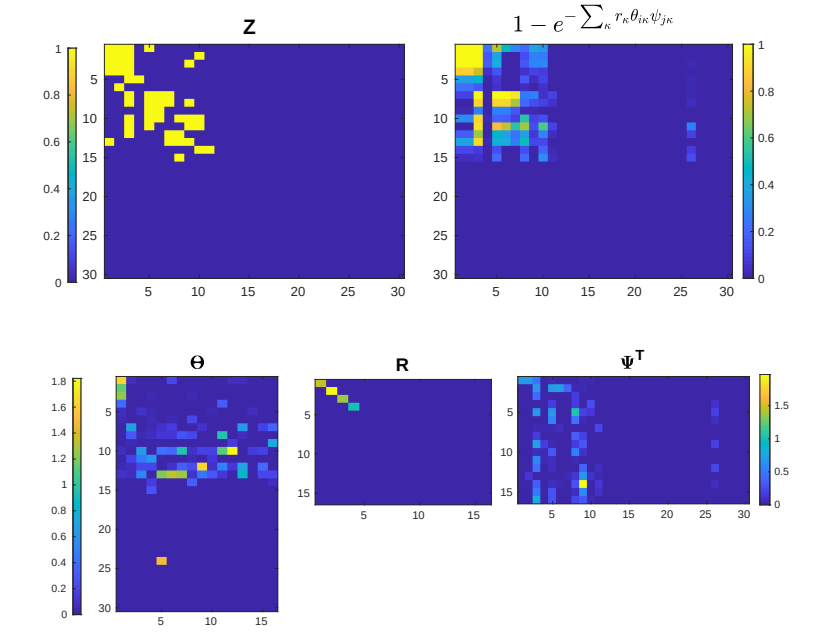
We compare the predictive performance of GGP-GLDS with these of several representative time series models, whose description and parameter setup for each dataset are described in detail below. For each dataset, we consider tuning important parameters for each competing algorithm. Notably for GGP-(G)LDS, when evaluating predictive performance, we simply use a same set of non-informative hyperparameters across all datasets and initialize all learnable parameters at random. Additional experiments on a synthetic dataset (the FitzHugh-Nagumo model) and a real dataset (closing stock price of twelve companies) will also be provided in the Appendix.

4.1 Lorenz Attractor

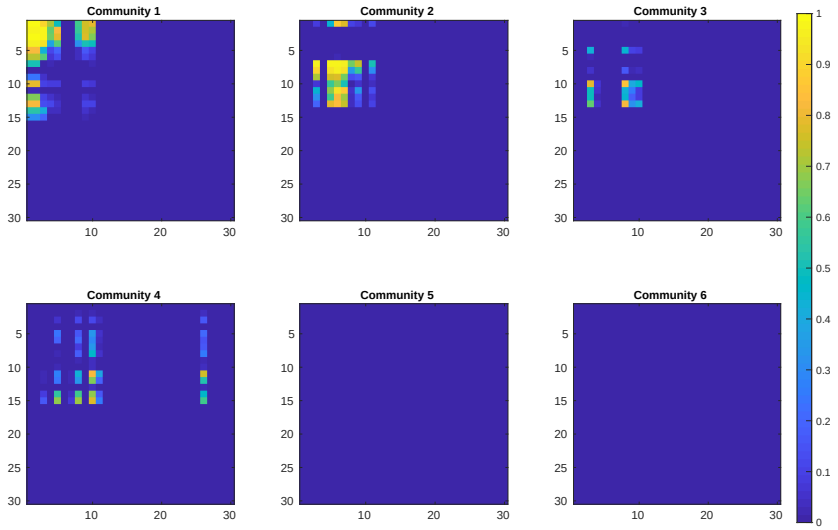
To demonstrate the performance of GGP-LDS on a dataset that has an underlying nonlinear dynamical pattern, we consider the Lorenz Attractor. We show how GGP-LDS finds an interpretable approximation to the generated time series with nonlinear dynamics. The Lorenz system is a classical nonlinear differential equation with three independent variables, defined as

$$\frac{dx_1}{dt} = \alpha(x_2 - x_1), \quad \frac{dx_2}{dt} = x_1(\beta - x_3) - x_2, \quad \frac{dx_3}{dt} = x_1 x_2 - \gamma x_3.$$

There exist approximate solutions for this differential equation (Hernandez et al., 2018; Linderman et al., 2017; Nassar et al., 2018). A linear approximation will be very useful as we can leverage for this non-linear system many canonical algorithms developed for filtering and smoothing on linear systems. To show how our model approximates the latent states, we generate numerical solutions of the Lorenz system with a randomly generated initial state, $\alpha = 1$, $\beta = 2$, $\gamma = 1$, and $T = 578$ time points. The original generated variables under the Lorenz system have three dimensions (x_1, x_2, and x_3). We treat them as latent variables and use a randomly generated 10×3 matrix to map them to a 10-dimensional observation space. We use this $10 \times T$ observed data with added white Gaussian noise to train both GGP-LDS and a variety of baseline models.



(a)



(b)

Figure 1: (a): Visualization of a GGP-LDS inferred posteriori sample on a Lorenz Attractor synthesized time series. Top Left: $\mathbf{Z}$ from this posterior sample, where the rows and columns are separately reordered with the method described in Section 2.2; Top Right: The inferred activation probability of $\mathbf{Z}$; Bottom Left: $\mathbf{\Theta}$, where $\theta_{i\kappa}$ shows how strongly state i is influenced by the states of the previous time due to its association with community κ; Bottom Middle: $\mathbf{R}$, whose diagonal elements show the activation strength of different communities; Bottom Right: $\mathbf{\Psi}^T$, where $\psi_{\kappa,j}$ shows how strongly state j influences the states of the next time due to its association with community κ.

(b): Relative activation strength $\mathbf{A}_\kappa$, as defined in (6), of the top six communities; note that 4 active communities formed over 15 active latent states are inferred by GGP-LDS while the truncation levels of the GGP are set as $K = 16$ and $S = 30$.

Fig. 1(a) illustrates a single posterior sample of GGP-LDS, focusing on the inferred graph adjacency matrix, and the underlying activation probabilities of the edges of the graph adjacency matrix. More specifically, in the top row, we show on the left the graph adjacency matrix $\mathbf{Z}$, whose rows and columns have been separately reordered following the description in Section 2.2, and on the right the underlying edge activation probabilities. In the bottom row, we show $\mathbf{\Theta} = (\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_K) \in \mathbb{R}_+^{S \times K}$, $\mathbf{R} = \text{diag}(r_1, \ldots, r_K) \in \mathbb{R}_+^{K \times K}$, and $\mathbf{\Psi}^T = (\boldsymbol{\psi}_1, \ldots, \boldsymbol{\psi}_K)^T \in \mathbb{R}_+^{K \times S}$, where $\theta_{i\kappa}$ shows the affiliation strength of $x_{(t+1)i}$ to the κ^{th} community (κ^{th} LDS) and $\psi_{\kappa,j}$ shows the association strength of x_{tj} to the κ^{th} LDS.

It can be observed how the shrinkage property of the gamma process $G_{\rho,\tau}$ has been effective in sparsifying the rows of $\mathbf{\Theta}$ and columns of $\mathbf{\Psi}^T$, with unnecessary elements being shrunk towards zero. In addition, it can be seen that each active row of $\mathbf{\Theta}$, or active column of $\mathbf{\Psi}^T$ can potentially be a member of several different communities. The shrinkage property of the GGP G_o drives many elements of r_k towards zero and hence helps the model to pick which types of LDSs to be utilized. This is equivalent to say that the model infers which of these associations should be amplified or suppressed in expressing the underlying dynamics of the data. Moreover, for the $\mathbf{\Theta}$ matrix, it has 7 members (rows) associated with community one, which implies there are 7 corresponding states at time $t+1$ that will be influenced by $\boldsymbol{x}_t$ of the previous time due to their associations with community one, and $\mathbf{\Psi}^T$ shows that it has 4 members (columns) associated with community one, which implies that there are 4 corresponding states at time t that will influence $\boldsymbol{x}_{t+1}$ of the next time due to their associations with community one. Thus the transition matrix of the first member of overlapping LDSs will be the 7×4 block shown on the top left corner, as shown in both $\mathbf{Z}$ and its corresponding probability matrix in Fig. 1(a).

Out of $K = 16$ (truncation level) possible communities, we show in Fig. 6(a) the top six formed communities, extracted from the inferred transition matrix for Lorenz Attractor, in six different subplots; we display each of these six communities using its relative strength defined in (6). It is shown in Fig. 1(b) that our nonparametric Bayesian model finds four communities in total to model the underlying pattern of the data. The number of linear solutions that our model has discovered is similar to that of Nassar et al. (2018), in which a tree based stick breaking process has been used as the prior. Moreover, it can be observed from Figs. 1(a) and 1(b) that these 4 active communities are formed over 15 active states. Note for GGP-LDS, we have truncated its number of communities at $K = 16$ and that of states at $S = 30$. The results in Fig. 1 have demonstrated the ability of GGP-LDS in inferring a parsimonious set of active communities and states to model the time series.

Fig. 2 shows how each community can reconstruct the observed data. Each row corresponds to a data dimension of the observed time series. The first three columns show how the three strongest communities contribute to data reconstruction, while the last column shows the superpositions of the first three columns and compares them against the observed time series. It can be seen from Fig. 2 that the different dimensions of each community specific sub-sequence share similar temporal patterns, but may exhibit clearly different magnitudes.

In Fig. 3(a), the red trajectory in the left plot represents the Lorenz Attractor synthesized 3D

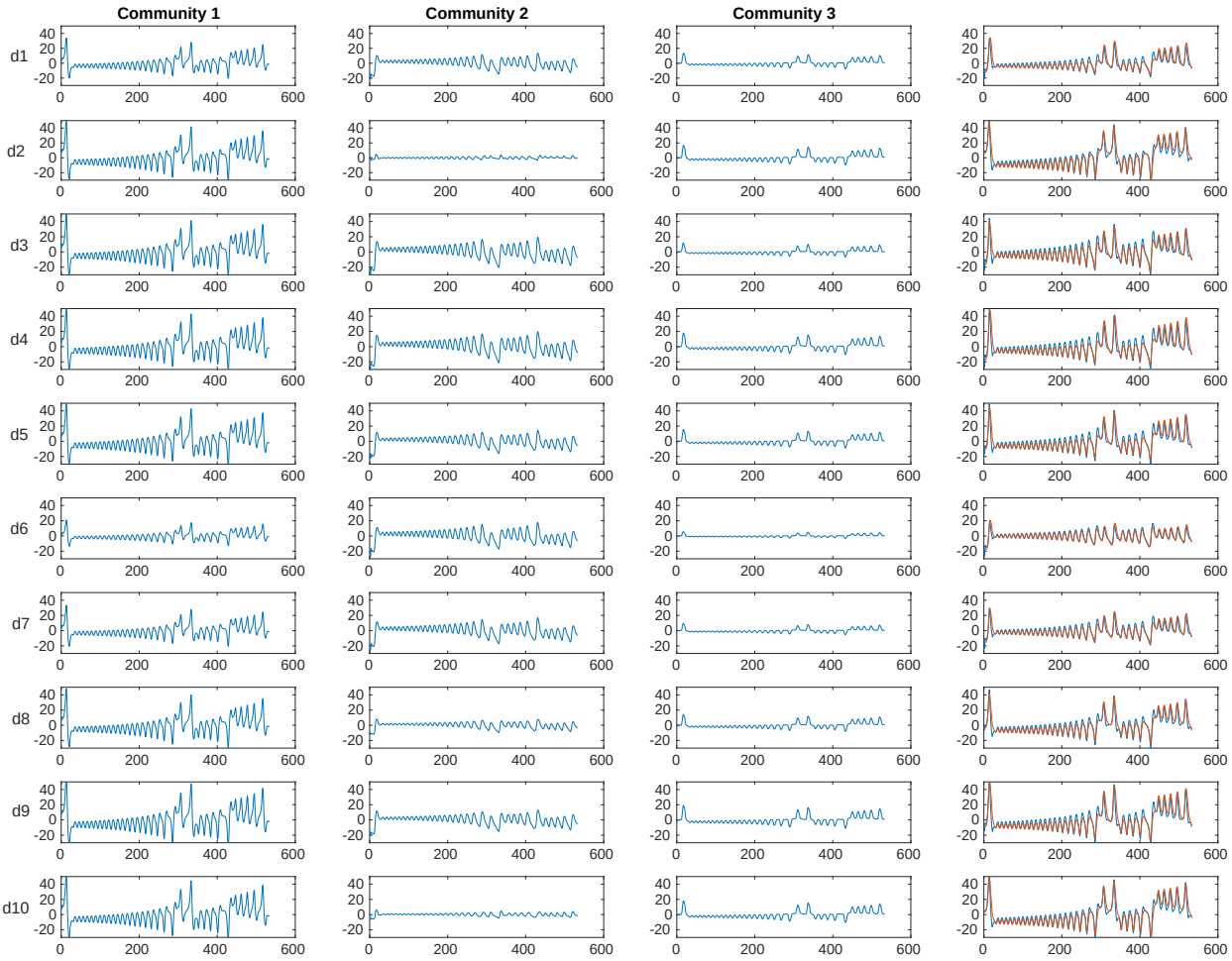
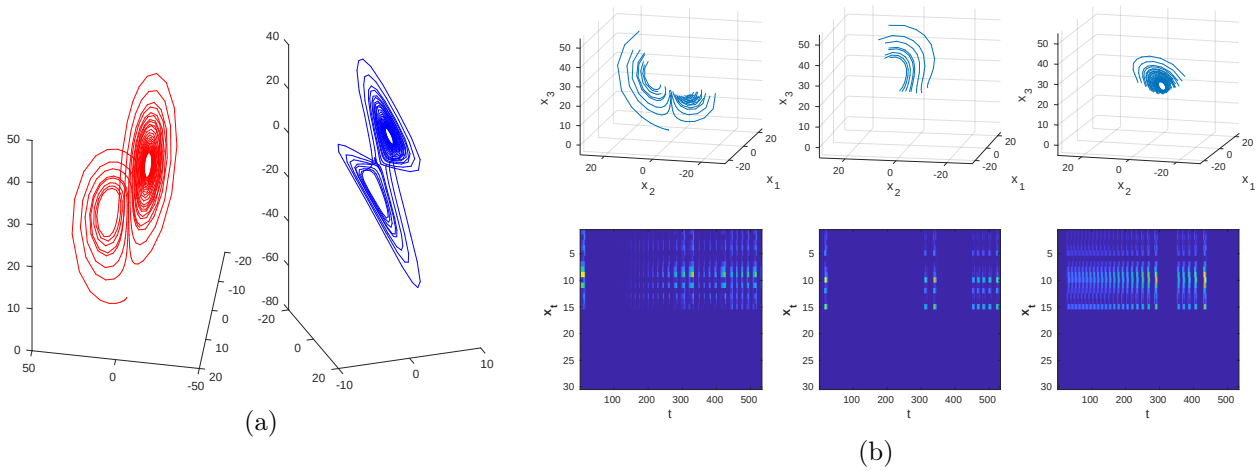


Figure 2: Visualization of community specific sub-sequences decomposed from the Lorenz Attractor based time series reconstructed by GGP-LDS. The first three columns show the sub-sequences of the three strongest communities, with each row showing one of the 10 data dimensions. The last column shows the superposition of the three sub-sequences shown in the first three columns, where the observed time series is also included for comparison (highlighted in red).

time series that is used as the latent state representation to generate the observed 10D time series, and the blue trajectory in the right plot illustrates a 3D representation of the latent dynamics of GGP-LDS trained on this 10D time series. More specifically, the blue trajectory is the visualization of the inferred community-specific latent sub-sequences $(\hat{\boldsymbol{x}}^{(1)}_{1:T}, \hat{\boldsymbol{x}}^{(2)}_{1:T}, \hat{\boldsymbol{x}}^{(3)}_{1:T})$, where $\hat{\boldsymbol{x}}^{(\kappa)}_{1:T}$, defined as in (7), is the latent sub-sequence extracted according to the relative strength of the κ^{th} strongest community to the aggregation of all communities, as illustrated in Figs. 1(b) and 2 and described in detail in Algorithm 1. It can be seen from Fig. 3(a) that the latent dynamics (*e.g.*, moving between two spirals) of GGP-LDS, visualized in 3D based on its inferred sub-sequences of its three strongest communities, are closely synchronized with the underlying dynamics of the Lorenz Attract synthesized 3D time series (a video showing how the red and blue trajectories move synchronously with each other has been included). This shows that our model infers a close linear approximation to the underlying nonlinear dynamics.



(a) (b)

Figure 3: (a): The red trajectory shown on the left is synthesized by a Lorenz Attractor and used as the 3D latent state sequence to generate $\boldsymbol{y}_{1:T}$, a 10D time series observation. Training GGP-LDS on $\boldsymbol{y}_{1:T}$, the blue trajectory shown on the right is a 3D visualization of the inferred latent dynamics based on $(\hat{\boldsymbol{x}}^{(1)}_{1:T}, \hat{\boldsymbol{x}}^{(2)}_{1:T}, \hat{\boldsymbol{x}}^{(3)}_{1:T})$, the sub-sequences of the three strongest communities decomposed from the reconstructed time series by GGP-LDS.

(b): The bottom row visualizes the inferred latent states $\boldsymbol{x}_{1:T}$ of GGP-LDS, which are assigned into three non-overlapping clusters via the K-means algorithm, and the top row visualizes the corresponding segments of the Lorenz Attractor synthesized 3D time series.

We provide another visualization of the latent dynamics inferred by GGP-LDS in Fig. 3(b). Instead of decomposing the time series into sub-sequences, we now cluster it in time according to the inferred latent states $\boldsymbol{x}_t$. In the bottom row of Fig. 3(b), the $\boldsymbol{x}_t$'s are partitioned into three non-overlapping clusters with the K-means algorithm, which means each $\boldsymbol{x}_t$ is assigned to one of the three clusters. In the top row of Fig. 3(b), the same cluster assignment is applied to segment the Lorenz Attractor time series into three sequences that do not overlap in time. It is clear that the segmentation points based on the $\boldsymbol{x}_t$'s inferred by GGP-LDS well align with the switching points between different linear dynamics, demonstrating the ability of GGP-LDS to seamlessly transit between different temporal patterns, each of which is modeled by adjusting the activation strengths of different latent state communities that can co-occur at the same time.

In addition to these qualitative analyses, we quantitatively compare GGP-LDS and a variety of baseline algorithms on their predictive performance on the same 10D time series $\boldsymbol{y}_t$, generated by adding Gaussian noise to $\mathbf{D}\boldsymbol{x}_t$, where $\boldsymbol{x}_{1:T}$ is a Lorenz Attractor synthesized 3D time series. Below we list the parameter settings of the baseline algorithms; for each competing algorithm, we choose its setting on each dataset that provides it the best overall prediction performance over 10 forecast horizons, except for SGLDS and GGP-LDS that are insensitive to hyperparameter settings under their respective nonparametric Bayesian constructions:

- LDS of Ghahramani and Hinton (1996), inferred by EM, with $K = 6$.
- $rLDS_g$ and $rLDS_r$ of Liu and Hauskrecht (2015), with $K = 8$ and random initialization.
- SGLDS of Kalantari et al. (2018), with the number of hidden states truncated at $K = 30$.

Table 1: Lorenz Attractor predictive performance. The best result and the results that are not considered as statistically different are highlighted in bold.

Mean absolute error for 10 forecast horizons										
Algorithm	$t=1$	$t=2$	$t=3$	$t=4$	$t=5$	$t=6$	$t=7$	$t=8$	$t=9$	$t=10$
LDS	$11.12_{(1.10)}$	$13.76_{(2.47)}$	$18.54_{(2.78)}$	$22.34_{(2.90)}$	$18.81_{(3.18)}$	$13.12_{(3.30)}$	$9.68_{(2.87)}$	$8.54_{(2.37)}$	$7.21_{(2.51)}$	$7.54_{(2.48)}$
rLDSg	$8.12_{(0.90)}$	$12.76_{(1.31)}$	$17.33_{(1.98)}$	$20.52_{(1.61)}$	$15.62_{(2.05)}$	$10.54_{(3.21)}$	$8.62_{(3.28)}$	$9.42_{(3.65)}$	$7.46_{(3.16)}$	$6.28_{(3.49)}$
rLDSr	$12.46_{(1.72)}$	$19.22_{(3.72)}$	$21.51_{(4.22)}$	$25.32_{(3.67)}$	$18.21_{(3.11)}$	$11.51_{(4.16)}$	$13.21_{(4.21)}$	$10.21_{(4.11)}$	$8.57_{(3.95)}$	$6.91_{(3.18)}$
SGLDS	$8.84_{(1.32)}$	$10.43_{(1.66)}$	$14.51_{(2.43)}$	$15.32_{(3.61)}$	$16.31_{(3.24)}$	$15.32_{(3.83)}$	$12.21_{(3.94)}$	$9.13_{(4.24)}$	$9.57_{(3.95)}$	$7.91_{(3.68)}$
TrSLDS	$5.21_{(0.62)}$	$5.76_{(0.98)}$	$6.23_{(1.31)}$	$7.45_{(1.42)}$	$\mathbf{5.31_{(1.19)}}$	$\mathbf{5.12_{(1.53)}}$	$\mathbf{4.21_{(1.24)}}$	$\mathbf{2.31_{(1.18)}}$	$\mathbf{2.57_{(1.45)}}$	$\mathbf{5.78_{(1.11)}}$
Multi-output GP	$11.52_{(1.58)}$	$15.35_{(1.73)}$	$16.21_{(2.21)}$	$19.08_{(2.36)}$	$17.21_{(3.79)}$	$12.37_{(3.77)}$	$8.38_{(3.98)}$	$8.21_{(2.98)}$	$6.31_{(3.21)}$	$7.21_{(3.98)}$
FB Prophet	$5.57_{(1.31)}$	$11.82_{(1.45)}$	$13.42_{(1.98)}$	$15.21_{(2.04)}$	$16.26_{(1.76)}$	$9.41_{(1.86)}$	$8.78_{(2.01)}$	$7.66_{(1.91)}$	$6.54_{(2.14)}$	$6.72_{(2.23)}$
DeepAR	$9.42_{(0.26)}$	$10.21_{(0.31)}$	$16.22_{(0.54)}$	$16.42_{(1.28)}$	$15.24_{(1.21)}$	$11.21_{(1.61)}$	$13.25_{(2.08)}$	$12.83_{(2.83)}$	$14.21_{(3.01)}$	$16.25_{(3.21)}$
TRCRP	$5.66_{(0.86)}$	$7.91_{(1.01)}$	$11.23_{(1.35)}$	$15.37_{(2.31)}$	$16.21_{(2.42)}$	$9.68_{(2.68)}$	$8.21_{(2.98)}$	$6.85_{(2.71)}$	$5.63_{(2.38)}$	$7.35_{(2.81)}$
GGP-LDS (10 steps)	$\mathbf{2.12_{(0.84)}}$	$\mathbf{3.76_{(1.87)}}$	$\mathbf{4.77_{(2.68)}}$	$\mathbf{5.04_{(3.16)}}$	$\mathbf{4.83_{(3.32)}}$	$\mathbf{4.50_{(3.27)}}$	$\mathbf{4.15_{(3.34)}}$	$4.14_{(3.51)}$	$4.60_{(3.66)}$	$\mathbf{5.24_{(3.86)}}$
GGP-LDS (1 step)	$2.10_{(0.52)}$	$0.37_{(0.23)}$	$0.32_{(0.17)}$	$0.41_{(0.18)}$	$0.40_{(0.24)}$	$0.64_{(0.27)}$	$0.84_{(0.28)}$	$0.81_{(0.25)}$	$0.66_{(0.23)}$	$0.57_{(0.23)}$

- TrSLDS of Nassar et al. (2018), with its tree depth set as 2 and dimension of latent states for each LDS as 4.
- Multi-outputGP of Alvarez and Lawrence (2009), setting: 1. multi-gp option, 2. kerneltype: ggwhite, 3. optimizer: scg, 4. nlf = 1.
- FB Prophet of Taylor and Letham (2018), default setting.
- Deep AR of Flunkert et al. (2017), default setting.
- TRCRP of Saad and Mansinghka (2018), with the Markov chain order set as $p = 10$.

As $t = 445$ is one of the switching time points at which $\boldsymbol{x}_t$ moves from one spiral to another, we choose $\boldsymbol{y}_{1:445}$ for training. This set up can measure how well an algorithm detects and responds to changes in the underlying dynamics. The predictive performance of each algorithm is measured by mean absolute error defined in 10 over a horizon of 10 time points. The results are presented in Table 1. As shown in Table 1, most of the competing algorithms are not making good predictions following the switching point, likely because they expect that the trajectory will keep following the same spiral pattern before the switching point. In reality, the trajectory quickly switches to the other spiral pattern for a few steps before coming back to the same spiral pattern observed before $t = 445$. To further illustrate this point, we pick three different dimensions of the 10D time series $\boldsymbol{y}_t$, and show in Fig. 4 the prediction of four different algorithms, including SGLDS, TrLDS, TCRCP, and the proposed GGP-LDS, on these three dimensions over a horizon of 10 time points. It is evident from Fig. 4 that at the switching point, SGLDS, TrLDS, and TCRCP all fail to detect the transition from one spiral to another. More specifically, SGLDS closely follows the pattern of the same spiral, TrSLDS is experiencing delays in switching to the correct LDS that better fits the second spiral, and TRCRP creates wrong patterns.

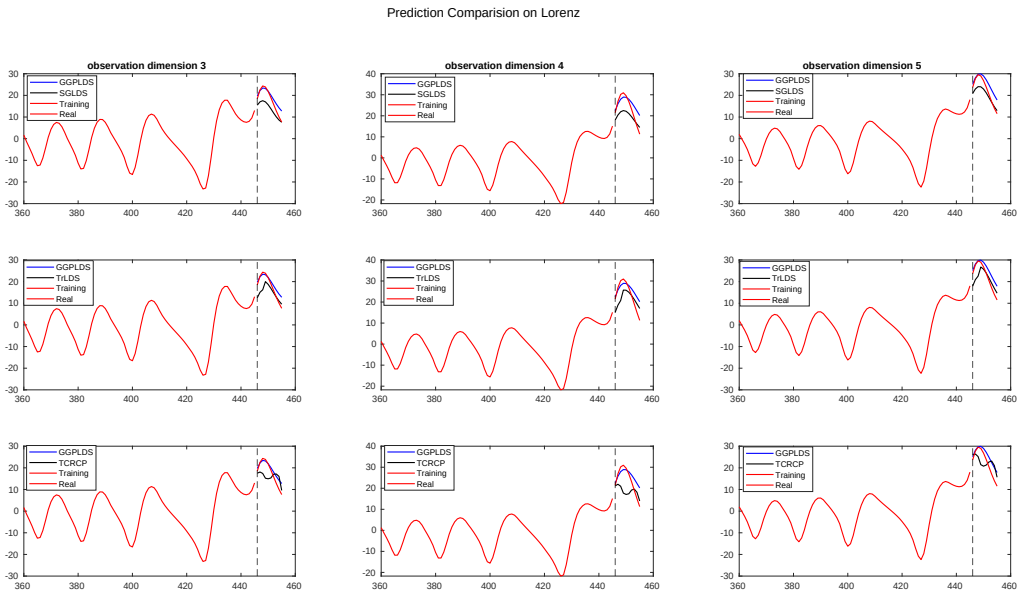


Figure 4: Comparison of predictive performance of the GGPLDS to SGLDS, TrSLDS and TRCRP on Lorenz attractor data. Each column of the figure represents one dimension of the observation (out of 10) and each row contain the visual comparison of GGP-LDS with one of those aforementioned algorithms.

4.2 FitzHugh-Nagumo

We provide another set of visualization of GGP-LDS by considering the FitzHugh-Nagumo (FHN) model, which can be formulated by the following differential equations:

$$\frac{dv}{dt} = v - \frac{v^3}{3} - w + I,$$
$$\frac{dw}{dt} = c \times (v + a - bw)\,.$$

We set $I = 0.3$, $a = 0.7$, $b = 0.8$, and $c = 0.7$. We train our model with a trajectory whose starting point is picked randomly. The trajectory consists of 800 time points. The observation model is linear and Gaussian where $\boldsymbol{d}_i \sim \mathcal{N}\left([0,0]^T, \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}\right)$ and $\boldsymbol{\Phi} = 0.01 * \mathbf{I}_2$, where $\mathbf{I}_2$ is an identity matrix of dimension 2×2. We provide a set of plots analogous to those in Figs 1(a), 1(b), and 3(a) for Lorenz Attractor. It can be seen in Fig. 5 how the latent trajectories have been recovered by $(\hat{\boldsymbol{x}}_{c_1}, \hat{\boldsymbol{x}}_{c_2})$ that represent the activation strength of the two strongest communities (c_1, c_2).

4.3 Pedestrians' Trajectories

We analyze a dataset that records by camera the 3D motions of pedestrians and their interactions, downloaded from `https://motchallenge.net/` and available in the provided code repository. For visualization purpose, we use only 2 dimensional data for each pedestrian (x, y). We select six pedestrians over 120 time points to train our model. The next 10 time points are used to measure

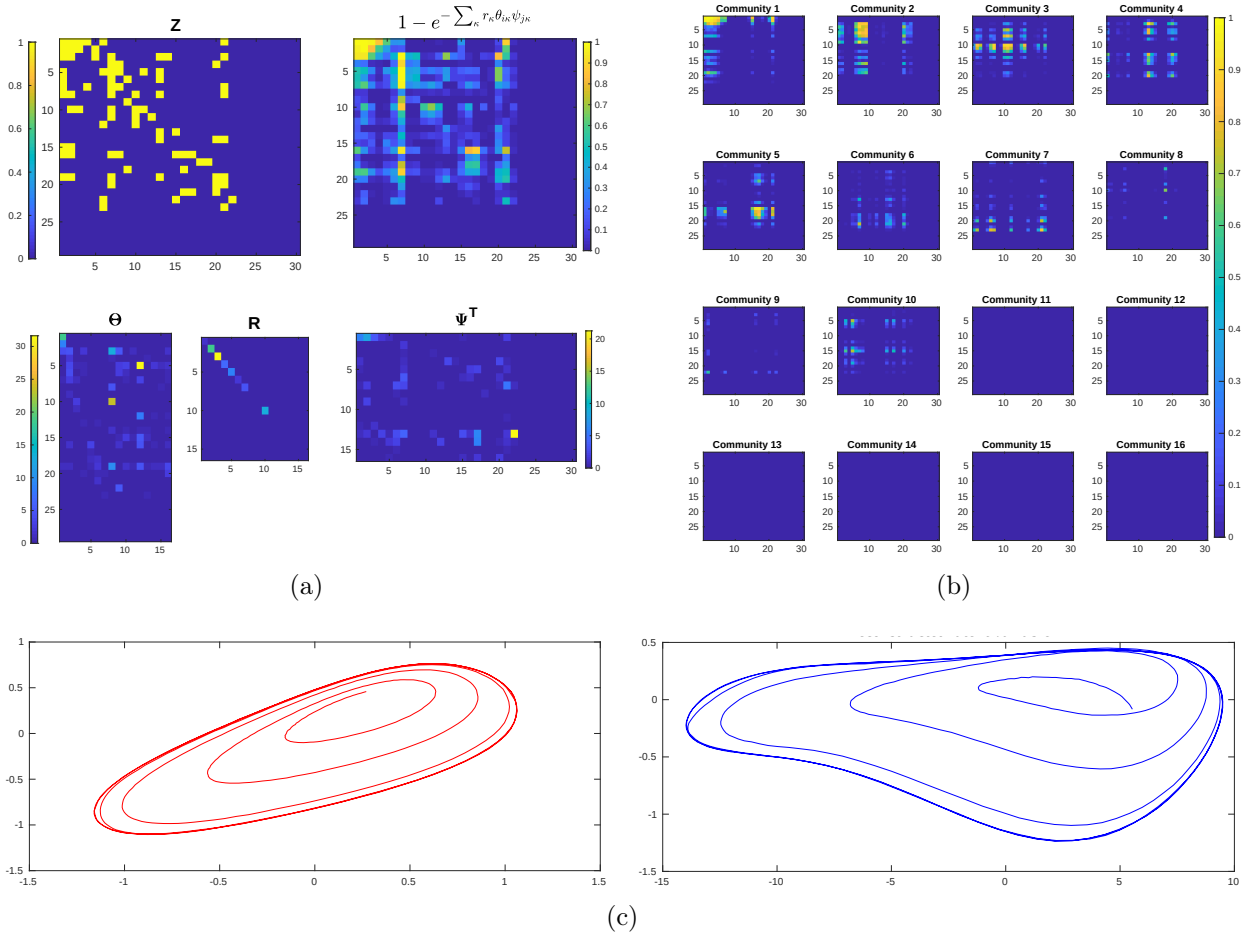


Figure 5: Visualization of a posterior sample of GGP-LDS applied to FitzHugh-Nagumo, where (a), (b), and (c) are analogous plots to Figs. 1(a), 1(b), and 3(a), respectively.

the predictive performance of various algorithms.

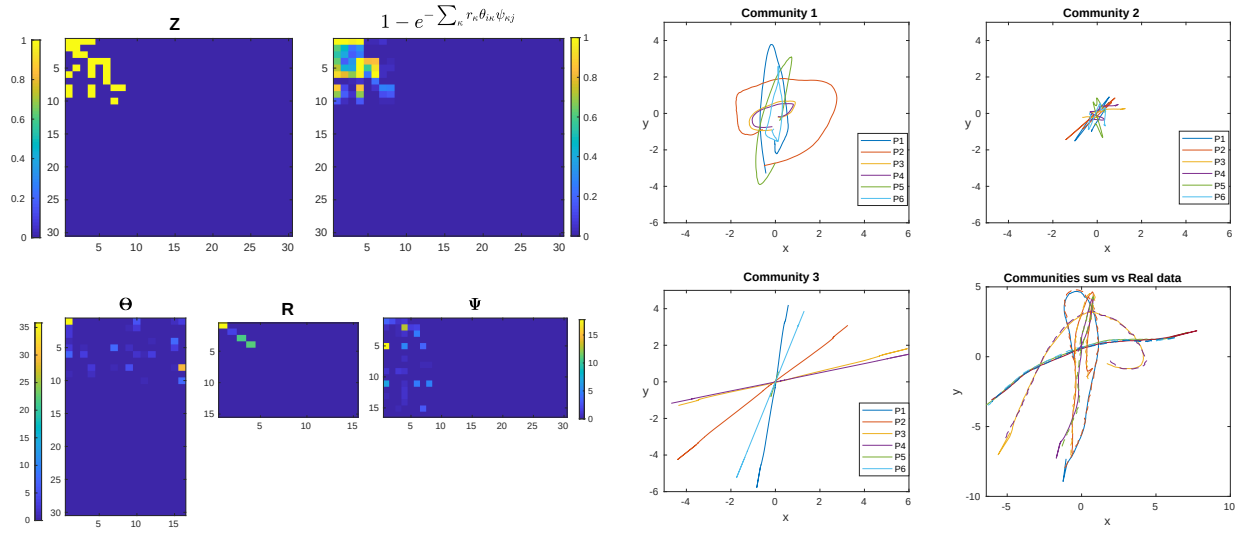
Table 2 compares the predictive performance of various algorithms on this dataset. In most of the 10 forecast horizons our model has outperformed the other competing models. The parameter settings of various algorithms are the same as those in Section 4.1 for Lorenz Attractor, with the following exceptions:

- LDS of Ghahramani and Hinton (1996), inferred by EM, with $K = 8$.
- TrSLDS of Nassar et al. (2018), with its tree depth set as 2 and dimension of latent states for each LDS as 2.

Fig. 6(a) provides the interpretation of the latent factors for this dataset, analogous to Fig. 1(a) used to provide interpretation of the latent structure inferred from Lorenz Attractor. Fig. 6(b), analogous to Fig. 2 for Lorenz Attractor, represents the reconstruction of all 6 pedestrians' trajectories using the three strongest communities. A total of four communities is inferred by GGP-LDS to model the underlying pattern of the data. Fig. 6(b) shows how each community can decompose

Table 2: Analogous table to Table 1 for the Pedestrians dataset

Mean absolute error for 10 forecast horizons										
Algorithm	$t=1$	$t=2$	$t=3$	$t=4$	$t=5$	$t=6$	$t=7$	$t=8$	$t=9$	$t=10$
LDS	$0.21_{(0.09)}$	$0.31_{(0.17)}$	$1.10_{(0.41)}$	$1.27_{(0.46)}$	$1.55_{(0.66)}$	$1.48_{(0.65)}$	$1.52_{(0.62)}$	$1.62_{(0.61)}$	$1.60_{(0.59)}$	$1.76_{(0.71)}$
rLDSg	$0.19_{(0.07)}$	$0.29_{(0.12)}$	$0.41_{(0.19)}$	$0.71_{(0.16)}$	$1.11_{(0.23)}$	$1.08_{(0.59)}$	$1.12_{(0.65)}$	$1.27_{(0.58)}$	$1.37_{(0.61)}$	$1.41_{(0.70)}$
SGLDS	$0.17_{(0.12)}$	$0.24_{(0.16)}$	$0.28_{(0.23)}$	$0.42_{(0.21)}$	$0.54_{(0.34)}$	$0.66_{(0.31)}$	$0.71_{(0.41)}$	$0.87_{(0.44)}$	$0.97_{(0.55)}$	$1.02_{(0.68)}$
TrSLDS	$0.16_{(0.10)}$	$0.26_{(0.12)}$	$0.32_{(0.18)}$	$0.40_{(0.21)}$	$0.52_{(0.28)}$	$0.60_{(0.29)}$	$0.69_{(0.32)}$	$0.79_{(0.39)}$	$0.79_{(0.41)}$	$0.92_{(0.48)}$
Multi-output GP	$0.22_{(0.11)}$	$0.28_{(0.16)}$	$0.42_{(0.21)}$	$0.60_{(0.20)}$	$0.68_{(0.21)}$	$0.89_{(0.17)}$	$1.03_{(0.23)}$	$1.15_{(0.28)}$	$1.41_{(0.32)}$	$1.40_{(0.30)}$
FB Prophet	$0.17_{(0.05)}$	$0.26_{(0.08)}$	$0.33_{(0.11)}$	$0.40_{(0.12)}$	$0.56_{(0.14)}$	$0.65_{(0.14)}$	$0.87_{(0.13)}$	$0.86_{(0.17)}$	$1.04_{(0.23)}$	$1.12_{(0.20)}$
DeepAR	$0.22_{(0.06)}$	$0.21_{(0.04)}$	$0.32_{(0.04)}$	$0.42_{(0.12)}$	$0.64_{(0.21)}$	$0.81_{(0.27)}$	$0.85_{(0.31)}$	$0.98_{(0.36)}$	$1.03_{(0.45)}$	$1.15_{(0.51)}$
TRCRP	$0.17_{(0.09)}$	$0.19_{(0.10)}$	$0.29_{(0.12)}$	$\mathbf{0.27_{(0.16)}}$	$\mathbf{0.33_{(0.21)}}$	$\mathbf{0.38_{(0.15)}}$	$0.61_{(0.22)}$	$0.66_{(0.28)}$	$0.71_{(0.32)}$	$0.92_{(0.38)}$
GGP-LDS (10 steps)	$\mathbf{0.14_{(0.11)}}$	$\mathbf{0.20_{(0.16)}}$	$\mathbf{0.26_{(0.21)}}$	$0.31_{(0.26)}$	$0.38_{(0.32)}$	$0.44_{(0.37)}$	$\mathbf{0.52_{(0.44)}}$	$\mathbf{0.59_{(0.50)}}$	$\mathbf{0.67_{(0.57)}}$	$\mathbf{0.75_{(0.64)}}$
GGP-LDS (1 step)	$0.11_{(0.08)}$	$0.13_{(0.09)}$	$0.14_{(0.11)}$	$0.14_{(0.11)}$	$0.15_{(0.12)}$	$0.17_{(0.13)}$	$0.19_{(0.14)}$	$0.21_{(0.16)}$	$0.23_{(0.18)}$	$0.26_{(0.21)}$



(a) (b)

Figure 6: Visualization of a posterior sample of GGP-LDS applied to the Pedestrian dataset, where (a) and (b) are analogous plots to Figs. 1(a) and 2, respectively.

the observed data in 2 dimensions (x, y) into a community-specific sub-sequence. The last subplot superposes the first three sub-sequences and compares it against the true trajectory (shown in dashed lines).

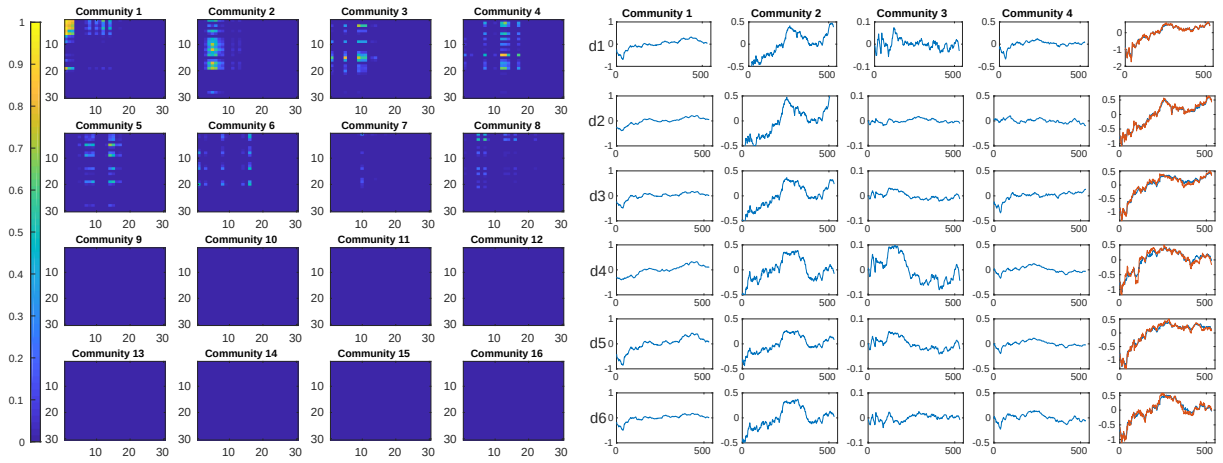
It is interesting to see how each community can create one type of motion (*e.g.*, straight movement, circular trajectory, and spiral movement). It is evident that regardless of the property of motion, such as "turn direction," "radius of circular motion," or "direction of straight," the trajectories of the same nature have appeared in a same community-specific sub-sequence. It can be seen in the figure that some of the communities have a very small contribution for some of the pedestrians' trajectory reconstruction since those pedestrians did not use that specific pattern in their recorded walking.

4.4 Stock Price

This dataset contains 12 companies' relative closing price $(P_t - P_{t-1})$ over the course of three years. These 12 companies have been selected from four different industries, and the stock closing prices of

Table 3: Analogous table to Table 1 for the Stock Price dataset

Mean absolute error for 10 forecast horizons										
Algorithm	$t=1$	$t=2$	$t=3$	$t=4$	$t=5$	$t=6$	$t=7$	$t=8$	$t=9$	$t=10$
LDS	$0.07_{(0.01)}$	$0.06_{(0.02)}$	$0.07_{(0.01)}$	$0.07_{(0.02)}$	$0.08_{(0.02)}$	$0.08_{(0.02)}$	$0.09_{(0.02)}$	$0.08_{(0.02)}$	$0.09_{(0.03)}$	$0.10_{(0.02)}$
rLDSg	$0.03_{(0.07)}$	$0.03_{(0.01)}$	$0.04_{(0.19)}$	$0.04_{(0.01)}$	$0.06_{(0.01)}$	$0.06_{(0.02)}$	$0.06_{(0.02)}$	$0.07_{(0.02)}$	$0.08_{(0.02)}$	$0.09_{(0.02)}$
SGLDS	$0.05_{(0.01)}$	$0.05_{(0.01)}$	$0.06_{(0.02)}$	$0.06_{(0.02)}$	$0.06_{(0.03)}$	$0.08_{(0.02)}$	$0.08_{(0.03)}$	$0.09_{(0.03)}$	$0.09_{(0.03)}$	$0.10_{(0.04)}$
TrSLDS	$0.03_{(0.01)}$	$0.04_{(0.01)}$	$0.04_{(0.02)}$	$\mathbf{0.05_{(0.02)}}$	$0.06_{(0.03)}$	$0.07_{(0.02)}$	$0.07_{(0.03)}$	$0.08_{(0.03)}$	$0.08_{(0.03)}$	$0.09_{(0.03)}$
Multi-output GP	$0.09_{(0.03)}$	$0.08_{(0.03)}$	$0.10_{(0.03)}$	$0.13_{(0.03)}$	$0.11_{(0.03)}$	$0.10_{(0.02)}$	$0.12_{(0.03)}$	$0.12_{(0.03)}$	$0.13_{(0.03)}$	$0.14_{(0.03)}$
FB Prophet	$\mathbf{0.01_{(0.02)}}$	$\mathbf{0.03_{(0.01)}}$	$0.06_{(0.01)}$	$0.08_{(0.02)}$	$0.07_{(0.01)}$	$0.08_{(0.02)}$	$0.09_{(0.02)}$	$0.08_{(0.02)}$	$0.09_{(0.03)}$	$0.10_{(0.02)}$
DeepAR	$0.05_{(0.01)}$	$0.05_{(0.02)}$	$0.06_{(0.02)}$	$0.06_{(0.03)}$	$0.07_{(0.02)}$	$0.08_{(0.03)}$	$0.08_{(0.03)}$	$0.09_{(0.03)}$	$0.09_{(0.04)}$	$0.10_{(0.03)}$
TRCRP	$0.03_{(0.01)}$	$0.04_{(0.02)}$	$0.04_{(0.01)}$	$0.06_{(0.01)}$	$0.07_{(0.02)}$	$\mathbf{0.05_{(0.02)}}$	$\mathbf{0.04_{(0.02)}}$	$\mathbf{0.03_{(0.01)}}$	$\mathbf{0.05_{(0.01)}}$	$0.07_{(0.02)}$
GGP-LDS (10 steps)	$0.03_{(0.01)}$	$\mathbf{0.03_{(0.01)}}$	$\mathbf{0.03_{(0.01)}}$	$0.05_{(0.02)}$	$\mathbf{0.05_{(0.01)}}$	$\mathbf{0.05_{(0.01)}}$	$0.05_{(0.01)}$	$0.05_{(0.02)}$	$0.06_{(0.02)}$	$\mathbf{0.07_{(0.02)}}$
GGP-LDS (1 step)	$0.02_{(0.01)}$	$0.02_{(0.01)}$	$0.02_{(0.01)}$	$0.03_{(0.01)}$	$0.02_{(0.01)}$	$0.02_{(0.01)}$	$0.02_{(0.01)}$	$0.02_{(0.01)}$	$0.04_{(0.01)}$	$0.05_{(0.02)}$



(a) (b)

Figure 7: Visualization of GGP-LDS applied to stock closing prices. (a) Inferred communities, analogous plot to Fig. 1(b) (b) Inferred community-specific sub-sequences of the four strongest communities and their superpositions, analogous plot to Fig. 2.

the ones in the same industry share similar temporal behaviors. Table 4 compares the predictive performance of various algorithms on this dataset. In most of the 10 forecast horizons our model has outperformed the other competing models. The parameter settings of various algorithms are the same as those in Section 4.1 for Lorenz Attractor, with the following exceptions:

- LDS of Ghahramani and Hinton (1996), inferred by EM, with $K = 12$.
- $rLDS_g$ of Liu and Hauskrecht (2015), with $K = 14$ and random initialization.
- TrSLDS of Nassar et al. (2018), with its tree depth set as 1 and dimension of latent states for each LDS as 8.

Fig. 7(a) shows the formed communities. Eight non-zero communities has been formed with the first four communities having at least one non-overlapping member, while the next four communities do not have members that are exclusive to them. Having four major communities, Fig. 7(b) shows how each of these major communities can contribute to reconstruct the observed data. Rows of

Table 4: Prediction of daily new Covid-19 cases in the U.S. given daily observations from March 6 to May 23, 2020.

Mean absolute relative error for 6 forecast horizons						
Algorithm	$t=1$	$t=2$	$t=3$	$t=4$	$t=5$	$t=6$
Year 2020	May 24	May 25	May 26	May 27	May 28	May 29
TrSLDS	$6.68_{(2.31)}$	$7.21_{(3.28)}$	$6.18_{(3.08)}$	$5.25_{(2.42)}$	$1.87_{(2.8)}$	$8.85_{(3.27)}$
UCLA Covid19	$\mathbf{3.87_{(1.68)}}$	$4.86_{(1.51)}$	$5.23_{(2.25)}$	$3.23_{(2.27)}$	$2.87_{(1.96)}$	$6.61_{(4.18)}$
UT-Austin Covid19	$6.32_{(2.51)}$	$5.38_{(2.73)}$	$6.26_{(3.25)}$	$3.78_{(2.13)}$	$3.16_{(2.02)}$	$7.95_{(3.93)}$
TRCRP	$7.66_{(3.81)}$	$6.27_{(3.25)}$	$7.22_{(3.36)}$	$4.21_{(2.86)}$	$2.31_{(1.37)}$	$\mathbf{6.11_{(3.25)}}$
GGP-GLDS (6 steps)	$4.78_{(2.95)}$	$\mathbf{4.47_{(2.27)}}$	$\mathbf{3.71_{(1.12)}}$	$\mathbf{1.68_{(5.67)}}$	$\mathbf{0.69_{(0.42)}}$	$9.50_{(18.01)}$

Fig. 7(b) correspond to 6 stocks out of 12. The first four columns of the figure describe how the four strongest detected communities contributed to reconstruct the data in each dimension. There are two noticeable observations in Fig. 7(b). First, it can be seen that each community represents similar behavior for all 6 selected stocks in the figure, and these behaviors are distinct from one community to another. Second, it is evident that if a behavior represented by a community does not play a significant role in reconstruction of the data in a specific dimension, that community contribution will be trivial. As an example, community 3's role in reconstructing the second stock is trivial, while playing a much more significant role to reconstruct the observation of the fourth stock.

4.5 Daily new COVID-19 Cases

This dataset contains the daily reported COVID-19 cases in each of the 50 States and District of Columbia from the beginning of March to end of May 2020. For predictive performance comparison, we have compared our model against two models which have been specifically developed to understand the COVID-19 pandemic: Zou et al. (2020) use a differential equation based epidemic model and use differential equation solver to find the model parameters; Woody et al. (2020) utilize the social distancing data as covariates and a statistical model negative binomial regression model utilizing these covariates to predict the intensity of pandemic in the future. Table 4 shows the 6-steps predictive performance of GGP-GLDS on this dataset. For the competing algorithms in Table 4, we list the parameter settings below:

- TrSLDS of Nassar et al. (2018), with its tree depth set as 1 and dimension of latent states for each LDS as 5.
- TRCRP of Saad and Mansinghka (2018), with the Markov chain order set as $p = 6$.
- UCLA-SuEIR of Zou et al. (2020)
- UT-Mobility of Woody et al. (2020)

Similar to previous visualizations, the first row of Fig. 8 shows the relative strengths of four discovered communities that have at least one non-overlapping member, with the last column of the first row representing $1 - e^{-\sum_\kappa r_\kappa \theta_{i\kappa} \psi_{j\kappa}}$. The relative strength of a community is defined as in

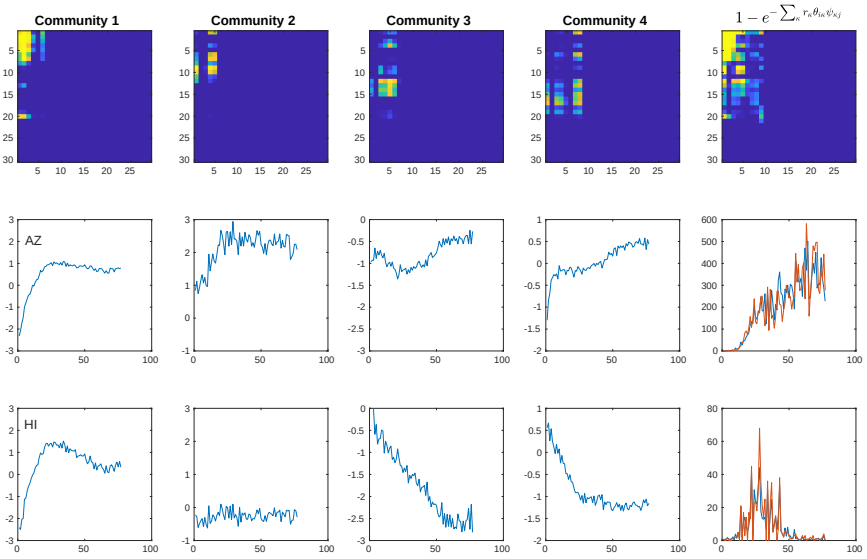


Figure 8: Visualization of GGP-GLDS (NBDS) applied to U.S. COVID-19 daily cases. Row 1 shows the relative activation strengths of the four strongest communities, analogous to Fig. 1(b); Rows 2 and 3 show the inferred community-specific sub-sequences of the four strongest communities and the expected counts given their superpositions.

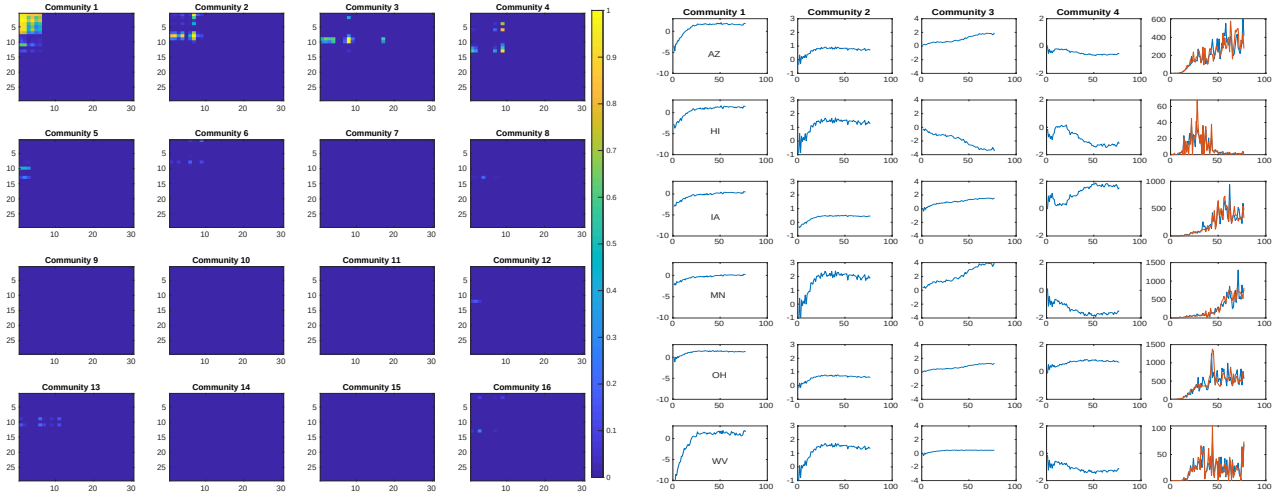
(6). The second row shows the trajectories for Arizona (AZ), with the trajectory in each of its first four columns showing a community-specific sub-sequence for AZ, which is the corresponding row of $\mathbf{D}\hat{\boldsymbol{x}}_{1:T}^{(\kappa)}$, where $\hat{\boldsymbol{x}}_t^{(\kappa)} = [(\mathbf{W} \odot \mathbf{Z}) \odot \mathbf{A}_\kappa]\boldsymbol{x}_{t-1}$ is defined as in (7); the last column of the second row compares the trajectory obtained by

$$\hat{\boldsymbol{y}}_t^{(\kappa)} := \eta e^{(\sum_{\kappa=1}^{4} \mathbf{D}\hat{\boldsymbol{x}}_t^{(\kappa)})}$$

against the true trajectory of AZ, which shows how discovered communities can reconstruct the observed time series at different data dimensions. Shown in the third row are analogous plots for Hawaii (HI).

Fig. 9(a) shows all discovered communities, out of which the first four communities have at least one non-overlapping member unique to themselves. Please note that the discovered communities and adjacency matrix of the posterior sample in Fig. 9 are different from those in Fig. 8. We intentionally choose two different samples from 2 different runs with different random seeds to show the consistency of interpretation across different runs. Although these two posterior samples come from two independent Markov chains initialized with different random seeds, and consequently different adjacency matrices, the algorithm still finds 4 communities with at least one non-overlapping member, and Figs. 9(b) and 8 show that the interpretation of the communities stay consistent from one run to another. This consistency of interpretation helps to show that no specific method for model parameter initialization is required for the proposed GGP-GLDS.

Similar to previous visualizations, there are two noticeable observations from Figs. 8 and 9(b). First, it can be seen how each community represents a distinct behavior shared by all data dimensions



(a) (b)

Figure 9: Visualization of a posterior sample (different from the sample used for Fig. 8) of GGP-GLDS (NBDS) applied to U.S. daily new COVID-19 cases. (a): Inferred communities, analogous to Fig. 1(b); (b): Inferred community-specific sub-sequences of the four strongest communities and their superpositions, analogous to Fig. 2.

at different magnitudes. Second, a behavior represented by a community may play a negligible role in reconstructing the data in a specific dimension if its corresponding magnitude is small.

5 Discussion and Conclusion

We introduce the graph gamma process (GGP) to form an infinite dimensional transition model with a finite random number of nonzero-degree nodes and a finite random number of nonzero-edge communities over these nodes. The GGP is used to promote sparsity on the state-transition matrix of a (generalized) linear dynamical system, and encourage forming overlapping communities among the nonzero-degree nodes of the graph. The model is designed such that each node community models a behavior described with a linear dynamical system. Instead of assigning one behavior to a temporal segment of an observation trajectory, it allows the whole time series to be viewed as a superposition of distinctly behaved trajectories, each of which is modeled by one of the discovered overlapping communities. These overlapping communities can be activated at different levels at different time points to support the superposition to have sophisticated temporal behavior, allowing smooth transitions between different linear dynamics to approximate nonlinear dynamical behaviors.

This paper focuses on time-series data modeling using a nonparametric Bayesian hierarchical model. The main challenges for modeling the time series include accurate forecast, missing data imputation, and meaningful clustering of the latent variables of the model such that these clusters can be translated to meaningful underlying patterns of the data, which represent different behaviors in a dataset without requiring data specific customization. The main goal of this paper is to

develop a nonparametric Bayesian model to find a flexible solution that provides interpretable latent representation and have good predictive performance. The nonparametric Bayesian construction of the model introduced in this paper prevents the underlying model from being over- and under-parameterized, which can result in over- or under-fitting, respectively, and hence poor performance in forecasting. In addition, the model learns a parsimonious graph among the infinite dimensional latent variables such that the induced clusters by community forming among non-zero-degree states can explain the underlying patterns of the data without the need of manual tuning. The nonparametric Bayesian construction and shrinkage property of the GGP introduced in this paper avoid inducing too many latent communities over too many latent states that will make the model difficult to interpret. Our experiments show that GGP-(G)LDS creates good balance between interpretability and predictive performance. Using the formed overlapping communities over the inferred set of latent states, the model is capable of modeling non-linearity in the data by creating multiple linear systems which try to learn the underlying non-linear behavior of the data. The model can break the sophisticated behavior in a trajectory to the combination of simpler behaviors modeled by linear dynamical systems, and it helps to model the nonlinearity of the data using multiple linear dynamical systems and smooth transition between them. The linear approximation of data with nonlinear dynamics enables the proposed GGP-LDS to leverage existing smoothing and filtering algorithms readily available to canonical LDSs.

Furthermore, this model can be seen as a generalization to existing nonparametric Bayesian linear dynamical systems from two different perspectives. First, the formed communities with overlapping members will translate to the soft switching concept among latent communities (or different LDSs) such that the strength of each LDS at any given time can change using the overlapping states which have replaced the hard switching among the different LDSs in switching LDSs. The hard switch is prone to picking a false LDS at the switching point, and results in wrong interpretation and deteriorated predictive performance. Second, unlike switching LDSs, the proposed model does not make any assumption on the dimensions of latent states, which are often set to be different in switching LDSs for different segments of the trajectory.

In addition, we show by an example how the proposed model with minor change in observation layer can turn to a model for generalized linear models (GLM) which helps us to model overdispersed count data, which often seriously violates the Gaussian assumption on observations. That adaptability helps us to model the trajectories of daily new COVID-19 cases in each State of the US by training the model on multivariate count observations across the 50 States and D.C., and hope its interpretability and predictive performance could help decision makers in pattern discovery, resource assignment, and predictive analytics related tasks. An on-going effort is using the proposed negative binomial dynamical system to jointly model the number of daily COVID-19 cases and deaths across all the States at the U.S. and provide forecast for future COVID-19 cases and deaths.

References

M. Alvarez and N. D. Lawrence. Sparse convolved gaussian processes for multi-output regression. In *Advances in neural information processing systems*, pages 57–64, 2009.

D. Barber, A. T. Cemgil, and S. Chiappa. *Bayesian time series models*. Cambridge University Press, 2011.

F. Caron, M. Davy, A. Doucet, E. Duflos, and P. Vanheeghe. Bayesian inference for linear dynamic models with dirichlet process mixtures. *IEEE Transactions on Signal Processing*, 56(1):71–84, 2008.

C. M. Carvalho and H. F. Lopes. Simulation-based sequential analysis of Markov switching stochastic volatility models. *Computational Statistics & Data Analysis*, 51(9):4526–4542, 2007.

A. Charles, M. S. Asif, J. Romberg, and C. Rozell. Sparsity penalties in dynamical system estimation. In *Information Sciences and Systems (CISS), 2011 45th Annual Conference on*, pages 1–6. IEEE, 2011.

A. Chiuso and G. Pillonetto. Learning sparse dynamic linear systems using stable spline kernels and exponential hyperpriors. In *Advances in Neural Information Processing Systems*, pages 397–405, 2010.

R. A. Davis, P. Zang, and T. Zheng. Sparse vector autoregressive modeling. *Journal of Computational and Graphical Statistics*, 25(4):1077–1096, 2016.

V. Flunkert, D. Salinas, and J. Gasthaus. Deepar: Probabilistic forecasting with autoregressive recurrent networks. *arXiv preprint arXiv:1704.04110*, 2017.

E. Fox, E. B. Sudderth, M. I. Jordan, and A. S. Willsky. Nonparametric bayesian learning of switching linear dynamical systems. In *NIPS*, pages 457–464. 2009.

Y. Gao, E. W. Archer, L. Paninski, and J. P. Cunningham. Linear dynamical neural population models through nonlinear embeddings. In *Advances in neural information processing systems*, pages 163–171, 2016.

Z. Ghahramani and G. E. Hinton. Parameter estimation for linear dynamical systems. Technical report, 1996.

Z. Ghahramani and S. T. Roweis. Learning nonlinear dynamical systems using an EM algorithm. In *NIPS*, pages 431–437, 1999.

M. Hardt, T. Ma, and B. Recht. Gradient descent learns linear dynamical systems. *The Journal of Machine Learning Research*, 19(1):1025–1068, 2018.

L. Harrison, W. D. Penny, and K. Friston. Multivariate autoregressive modeling of fmri time series. *Neuroimage*, 19(4):1477–1491, 2003.

D. Hernandez, A. K. Moretti, Z. Wei, S. Saxena, J. Cunningham, and L. Paninski. A novel variational family for hidden nonlinear markov models. *arXiv preprint arXiv:1811.02459*, 2018.

M. Imani and U. M. Braga-Neto. Particle filters for partially-observed boolean dynamical systems. *Automatica*, 87:238–250, 2018.

H. Ishwaran and J. S. Rao. Spike and slab variable selection: frequentist and Bayesian strategies. *Annals of statistics*, pages 730–773, 2005.

M. Johnson, D. K. Duvenaud, A. Wiltschko, R. P. Adams, and S. R. Datta. Composing graphical models with neural networks for structured representations and fast inference. In *Advances in neural information processing systems*, pages 2946–2954, 2016.

M. J. Johnson and A. S. Willsky. Bayesian nonparametric hidden semi-markov models. *Journal of Machine Learning Research*, 14(Feb):673–701, 2013.

R. Kalantari, J. Ghosh, and M. Zhou. Nonparametric bayesian sparse graph linear dynamical systems. *arXiv preprint arXiv:1802.07434*, 2018.

R. E. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering*, 82(Series D):35–45, 1960.

R. E. Kalman. Mathematical description of linear dynamical systems. *Journal of the Society for Industrial and Applied Mathematics, Series A: Control*, 1(2):152–192, 1963.

C. E. Kennedy and J. P. Turley. Time series analysis as input for clinical predictive modeling: Modeling cardiac arrest in a pediatric icu. *Theoretical Biology and Medical Modelling*, 8(1):40, 2011.

J. F. C. Kingman. *Poisson Processes*. Oxford University Press, 1993.

S. Koyama. Projection smoothing for continuous and continuous-discrete stochastic dynamic systems. *Signal Processing*, 144:333–340, 2018.

G. Lai, W.-C. Chang, Y. Yang, and H. Liu. Modeling long-and short-term temporal patterns with deep neural networks. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 95–104. ACM, 2018.

J. Li, C. Hu, D. Xu, J. Xiao, and H. Wang. Application of time-series autoregressive integrated moving average model in predicting the epidemic situation of newcastle disease. In *World Automation Congress (WAC), 2010*, pages 141–144. IEEE, 2010.

S. Linderman, M. Johnson, A. Miller, R. Adams, D. Blei, and L. Paninski. Bayesian learning and inference in recurrent switching linear dynamical systems. In *AISTATS*, volume 54, pages 914–922, Fort Lauderdale, FL, USA, 20–22 Apr 2017.

Z. Liu and M. Hauskrecht. A regularized linear dynamical system framework for multivariate time series analysis. In *AAAI*, pages 1798–1804, 2015.

L. Ljung. *System Identification: Theory for the User, 2nd edition*. Prentice Hall, 1999.

T. J. Mitchell and J. J. Beauchamp. Bayesian variable selection in linear regression. *Journal of the American Statistical Association*, 83(404):1023–1032, 1988.

K. P. Murphy. Switching Kalman filters. 1998.

J. Nassar, S. Linderman, M. Bugallo, and I. M. Park. Tree-structured recurrent switching linear dynamical systems for multi-scale modeling. *arXiv preprint arXiv:1811.12386*, 2018.

L. E. Nieto-Barajas, A. Contreras-Cristán, et al. A bayesian nonparametric approach for time series clustering. *Bayesian Analysis*, 9(1):147–170, 2014.

N. G. Polson and J. G. Scott. Default Bayesian analysis for multi-way tables: A data-augmentation approach. *arXiv preprint arXiv:1109.4180*, 2011.

N. G. Polson, J. G. Scott, and J. Windle. Bayesian inference for logistic models using pólya–gamma latent variables. *Journal of the American statistical Association*, 108(504):1339–1349, 2013.

F. Saad and V. Mansinghka. Temporally-reweighted chinese restaurant process mixtures for clustering, imputing, and forecasting multivariate time series. In *International Conference on Artificial Intelligence and Statistics*, pages 755–764, 2018.

S. Siddiqi, B. Boots, and G. Gordon. Reduced-rank hidden markov models. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 741–748, 2010.

N. Städler, S. Mukherjee, et al. Penalized estimation in high-dimensional hidden markov models with state-specific graphical models. *The Annals of Applied Statistics*, 7(4):2157–2179, 2013.

S. J. Taylor and B. Letham. Forecasting at scale. *The American Statistician*, 72(1):37–45, 2018.

Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei. Hierarchical Dirichlet processes. *J. Amer. Statist. Assoc.*, 101, 2006.

M. E. Tipping. Sparse bayesian learning and the relevance vector machine. *Journal of machine learning research*, 1(Jun):211–244, 2001.

M. West and J. Harrison. *Bayesian Forecasting and Dynamic Models (2Nd Ed.)*. Springer-Verlag New York, Inc., New York, NY, USA, 1997. ISBN 0-387-94725-6.

S. Woody, M. G. Tec, M. Dahan, K. Gaither, M. Lachmann, S. Fox, L. A. Meyers, and J. G. Scott. Projections for first-wave covid-19 deaths across the us using social-distancing measures derived from mobile phones. *medRxiv*, 2020.

H. Zhang, R. Ayoub, and S. Sundaram. Sensor selection for Kalman filtering of linear dynamical systems: Complexity, limitations and greedy algorithms. *Automatica*, 78:202–210, 2017.

M. Zhou. Infinite edge partition models for overlapping community detection and link prediction. In *AISTATS*, pages 1135–1143, 2015.

M. Zhou. Softplus regressions and convex polytopes. *arXiv preprint arXiv:1608.06383*, 2016.

M. Zhou and L. Carin. Negative binomial process count and mixture modeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(2):307–320, 2013.

M. Zhou, H. Chen, J. Paisley, L. Ren, G. Sapiro, and L. Carin. Non-parametric Bayesian dictionary learning for sparse image representations. In *NIPS*, 2009.

M. Zhou, L. Li, D. Dunson, and L. Carin. Lognormal and gamma mixed negative binomial regression. In *Proceedings of the... International Conference on Machine Learning. International Conference on Machine Learning*, volume 2012, page 1343. NIH Public Access, 2012.

D. Zou, L. Wang, P. Xu, J. Chen, W. Zhang, and Q. Gu. Epidemic model guided machine learning for covid-19 forecasts in the united states. *medRxiv*, 2020.

A Graphical Model

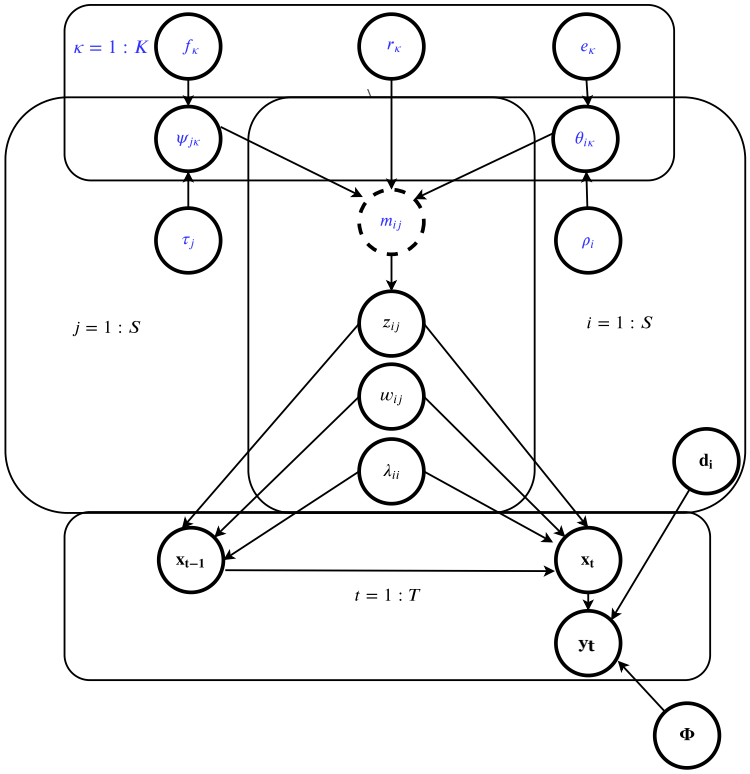


Figure 10: Graphical description of GGP-LDS.

B Proofs

Proof of Lemma 1. We first notice $z_{ij\kappa}$ can be equivalently generated under the Bernoulli-Poisson link (Zhou, 2015) as

$$z_{ij\kappa} = \delta(m_{ijk} \geq 1),\ m_{ij\kappa} \sim \text{Pois}(r_\kappa \theta_{i\kappa} \psi_{j\kappa}).$$

As $\sum_i \sum_j z_{ij\kappa} \leq \sum_i \sum_j m_{ij\kappa}$, it is sufficient to prove $\mathbb{E}[\sum_i \sum_j m_{ij\kappa}]$ is finite, which is true as $\mathbb{E}[\sum_i \sum_j m_{ij\kappa}] = r_\kappa \mathbb{E}[\sum_i \theta_{i\kappa}]\mathbb{E}[\sum_j \psi_{j\kappa}] = r_\kappa \gamma_\rho \gamma_\tau/(ef)$. □

Proof of Lemma 2. Following the proof of Lemma 1, with model properties, $\mathbf{Z} = \delta(\mathbf{M} \geq 1),\ \mathbf{M} \sim \text{Pois}(\sum_{k=1}^\infty r_\kappa \boldsymbol{\theta}_\kappa \boldsymbol{\psi}_\kappa^T)$, we have $\mathbb{E}[\sum_i \sum_j z_{ij}] \leq \mathbb{E}[\sum_i \sum_j \sum_\kappa \gamma_\kappa \theta_{i\kappa} \psi_{j\kappa}] = \mathbb{E}[\sum_\kappa r_\kappa \gamma_\rho \gamma_\tau/(ef)] = \gamma_0 \gamma_\rho \gamma_\tau/(cef)$. □

Below we show the total energy captured by all non-dynamic states is small: Due to the

normal-gamma construction, for state i we have

$$P(x_{1i}, \ldots, x_{Ti} \,|\, a, b) = \int_0^\infty \left[\prod_{t=1}^T \mathcal{N}(x_{ti}; 0, \lambda_i^{-1})\right] \text{Gamma}(\lambda_i; a, 1/b)\, d\lambda_i$$
$$= \frac{b^a \Gamma(a+\frac{T}{2})}{(2\pi)^{\frac{T}{2}} \Gamma(a)} \left(b + \tfrac{1}{2}\sum_{t=1}^T x_{ti}^2\right)^{-\left(a+\frac{T}{2}\right)}.$$

Similar to the analysis in Tipping (2001), if we let $a = b = 0$, we obtain an improper prior as $p(\sum_{t=1}^T x_{ti}^2) \propto (\sum_{t=1}^T x_{ti}^2)^{-T/2}$, which is sharply peaked at $\sum_{t=1}^T x_{ti}^2 = 0$. Thus choosing small a and b, the proposed model will penalize the total energy captured by node (state) i, expressed as $\sum_{t=1}^T x_{ti}^2$, if it is a zero-degree node.

C Bayesian Inference via Gibbs Sampling

1) ***Sample ω_{vt} for GGD-GLDS.*** Using data augmentation for negative binomial regression, as in Zhou et al. (2012) and Polson et al. (2013), we denote ω_{vt} as a random variable drawn from the Polya-Gamma (PG) distribution (Polson and Scott, 2011) as $\omega_{vt} \sim \text{PG}\left(y_{vt} + \eta,\ 0\right)$, under which we have $\mathbb{E}_{\omega_{vt}}\left[\exp(-\omega_{vt}(\xi_{vt})^2/2)\right] = \cosh^{-(y_{vt}+\eta)}(\xi_{vt}/2)$. Since $\boldsymbol{y}_t \sim \text{NB}(\eta, \sigma(\mathbf{D}\boldsymbol{x}_t))$, the likelihood of $\boldsymbol{\xi}_t := \mathbf{D}\boldsymbol{x}_t$ can be expressed as

$$\mathcal{L}(\boldsymbol{\xi}_t) \propto \prod_{v=1}^V \frac{(e^{\xi_{vt}})^{y_{vt}}}{(1+e^{\xi_{vt}})^{y_{vt}+\eta}}$$
$$= \prod_{v=1}^V \frac{2^{-(y_{vt}+\eta)} \exp(\frac{y_{vt}-\eta}{2}\xi_{vt})}{\cosh^{y_{vt}+\eta}(\xi_{vt}/2)}$$
$$\propto \prod_{v=1}^V \exp\left(\frac{y_{vt}-\eta}{2}\xi_{vt}\right) \mathbb{E}_{\omega_{vt}}\left[\exp[-\omega_{vt}(\xi_{vt})^2/2]\right].$$

Combining the likelihood $\mathcal{L}(\xi_{vt}, \omega_{vt}) \propto \exp\left(\frac{y_{vt}-\eta}{2}\xi_{vt}\right)\exp[-\omega_{vt}(\xi_{vt})^2/2]$ and the prior, we sample auxiliary PG random variables ω_{vt} as

$$(\omega_{vt} \,|\, -) \sim \text{PG}\left(y_{vt} + \eta,\ \mathbf{D}(v,:)\boldsymbol{x}_t\right), \tag{12}$$

where $\mathbf{D}(v,:)$ denotes the vth row of $\mathbf{D}$. Note to sample from the PG distribution, a fast and accurate approximate sampler of Zhou (2016) that matches the first two moments of the true distribution, with the truncation level set as five, is used in this paper.

2) ***Sample $\boldsymbol{x}_t$ for GGP-GLDS.*** Denote $\boldsymbol{\Omega}_t = \text{diag}\left(\omega_{1t}, \ldots, \omega_{Vt}\right)$. Given ω_{vt}, we sample $\boldsymbol{x}_t$ as

$$(\boldsymbol{x}_t \,|\, -) \sim \mathcal{N}(\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t), \tag{13}$$

where, for if $1 \leq t \leq T-1$, we have

$$\begin{aligned}
\boldsymbol{\Sigma}_t &= (\mathbf{D}^T \boldsymbol{\Omega}_t \mathbf{D} + \boldsymbol{\Lambda} + (\mathbf{W} \odot \mathbf{Z})^T \boldsymbol{\Lambda} (\mathbf{W} \odot \mathbf{Z}))^{-1}, \\
\boldsymbol{\mu}_t &= \boldsymbol{\Sigma}_t (\mathbf{D}^T \frac{\boldsymbol{y}_t - \eta}{2} + \boldsymbol{\Lambda} (\mathbf{W} \odot \mathbf{Z}) \boldsymbol{x}_{t-1} + (\mathbf{W} \odot \mathbf{Z})^T \boldsymbol{\Lambda} \boldsymbol{x}_{t+1}),
\end{aligned}$$

and

$$\begin{aligned}
\boldsymbol{\Sigma}_T &= (\mathbf{D}^T \boldsymbol{\Omega}_T \mathbf{D} + \boldsymbol{\Lambda})^{-1}, \\
\boldsymbol{\mu}_T &= \boldsymbol{\Sigma}_T \left(\mathbf{D}^T \frac{\boldsymbol{y}_T - \eta}{2} + \boldsymbol{\Lambda} (\mathbf{W} \odot \mathbf{Z}) \boldsymbol{x}_{T-1} \right).
\end{aligned}$$

3) ***Sample $\boldsymbol{x}_t$ for GGP-LDS.*** We sample $\boldsymbol{x}_t$ as

$$(\boldsymbol{x}_t \,|\, -) \sim \mathcal{N}(\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t), \tag{14}$$

where, for if $1 \leq t \leq T-1$, we have

$$\begin{aligned}
\boldsymbol{\Sigma}_t &= (\mathbf{D}^T \boldsymbol{\Phi} \mathbf{D} + \boldsymbol{\Lambda} + (\mathbf{W} \odot \mathbf{Z})^T \boldsymbol{\Lambda} (\mathbf{W} \odot \mathbf{Z}))^{-1}, \\
\boldsymbol{\mu}_t &= \boldsymbol{\Sigma}_t (\mathbf{D}^T \boldsymbol{\Phi} \mathbf{y}_t + \boldsymbol{\Lambda} (\mathbf{W} \odot \mathbf{Z}) \boldsymbol{x}_{t-1} + (\mathbf{W} \odot \mathbf{Z})^T \boldsymbol{\Lambda} \boldsymbol{x}_{t+1}),
\end{aligned}$$

and

$$\begin{aligned}
\boldsymbol{\Sigma}_T &= (\mathbf{D}^T \boldsymbol{\Phi} \mathbf{D} + \boldsymbol{\Lambda})^{-1}, \\
\boldsymbol{\mu}_T &= \boldsymbol{\Sigma}_T \left(\mathbf{D}^T \boldsymbol{\Phi} \mathbf{y}_T + \boldsymbol{\Lambda} (\mathbf{W} \odot \mathbf{Z}) \boldsymbol{x}_{T-1} \right), \\
\boldsymbol{\Sigma_0} &= (\mathbf{H_0} + (\mathbf{W} \odot \mathbf{Z})^T \boldsymbol{\Lambda} (\mathbf{W} \odot \mathbf{Z}))^{-1}, \\
\boldsymbol{\mu}_0 &= \boldsymbol{\Sigma}_0 (\mathbf{H_0} \boldsymbol{m}_0 + (\mathbf{W} \odot \mathbf{Z})^T \boldsymbol{\Lambda} \boldsymbol{x}_1).
\end{aligned}$$

4) ***Sample $\boldsymbol{d}_s$.*** We sample $\boldsymbol{d}_s$, the sth column of $\mathbf{D}$, as

$$(\boldsymbol{d}_s \,|\, -) \sim \mathcal{N}(\boldsymbol{m}_s, \mathbf{E}_s), \tag{15}$$

where for GGP-GLDS, we have

$$\begin{aligned}
\mathbf{E}_s &= \left(\sum_{t=1}^{T} x_{st}^2 \boldsymbol{\Omega}_t + \sqrt{V} \mathbf{I}_V \right)^{-1}, \\
\boldsymbol{m}_s &= \mathbf{E}_s \left[\sum_{t=1}^{T} x_{st} \left(\frac{\boldsymbol{y}_t - \eta}{2} - \boldsymbol{\Omega}_t \sum_{i \in \{1,\ldots,S\} \setminus s} \boldsymbol{d}_i x_{it} \right) \right].
\end{aligned}$$

and for GGP-LDS, we have

$$\mathbf{E}_s = \left(\sum_{t=1}^{T} x_{st}^2 \mathbf{\Phi} + \sqrt{V} \mathbf{I}_V \right)^{-1},$$
$$\boldsymbol{m}_s = \mathbf{E}_s \mathbf{\Phi} \left[\sum_{t=1}^{T} x_{st} \left(\boldsymbol{y}_t - \sum_{i \in \{1,\ldots,S\} \backslash s} \boldsymbol{d}_i x_{it} \right) \right].$$

5) ***Sample η for GGP-GLDS.*** Using the data augmentation techniques for the NB distribution (Zhou and Carin, 2013), we first sample an auxiliary random variable following the Chinese restaurant table (CRT) distribution and then sample η as

$$(l_{vt}^{(3)} \,|\, -) \sim \mathrm{CRT}\left(y_{vt}, \eta\right), \tag{16}$$
$$(l^{(4)} \,|\, -) \sim \mathrm{CRT}\left(\sum_{v=1}^{V} \sum_{t=1}^{T} l_{vt}^{(3)}, \alpha_\eta \right) \tag{17}$$
$$(\alpha_\eta \,|\, -) \sim \mathrm{Gamma}\Big(\alpha_0 + l^{(4)}, \frac{1}{\beta_0 - \ln(1-\tilde{p})}\Big), \tag{18}$$
$$(\beta_\eta | -) \sim \mathrm{Gamma}\Big(\alpha_0 + \alpha_\eta, \frac{1}{\beta_0 + \eta}\Big), \tag{19}$$
$$(\eta \,|\, -) \sim \mathrm{Gamma}\Big(\alpha_\eta + \sum_{v=1}^{V} \sum_{t=1}^{T} l_{vt}^{(3)}, \frac{1}{\beta_\eta + \sum_{v=1}^{V} \sum_{t=1}^{T} \zeta_{vt}}\Big), \tag{20}$$

where $\zeta_{vt} := \ln\left(1 + e^{\mathbf{D}(v,:)\boldsymbol{x}_t}\right)$ and $\tilde{p} := \frac{\sum_{v=1}^{V} \sum_{t=1}^{T} \zeta_{vt}}{\beta_\eta + \sum_{v=1}^{V} \sum_{t=1}^{T} \zeta_{vt}}$.

6) ***Sample $\mathbf{\Phi}^{-1}$ for GGP-LDS***. We sample $\mathbf{\Phi}^{-1}$ as

$$(\mathbf{\Phi}^{-1} \,|\, -) \sim \mathrm{InverseWishart}(G + \mathbf{V}, V + 2 + T), \tag{21}$$

where $G = G_1 - G_2 + G_3 - G_4$, $G_1 = \sum_{t=1}^{T} \boldsymbol{y}_t \boldsymbol{y}_t^T$, $G_2 = \sum_{t=1}^{T} \boldsymbol{y}_t \boldsymbol{x}_t^T \mathbf{D}^T$, $G_3 = \sum_{t=1}^{T} \mathbf{D} \mathbf{x_t} \boldsymbol{x}_t^T \mathbf{D}^T$, and $G_4 = \sum_{t=1}^{T} \mathbf{D} \boldsymbol{x}_t \mathbf{y_t^T}$.

7) ***Sample $\mathbf{W}$.*** We sample w_{ij} as

$$(w_{ij} \,|\, -) \sim \mathcal{N}(\mu_{ij}, \tau_{ij}), \tag{22}$$

where

$$\tau_{ij} = (z_{ij}\lambda_i T_j + \varphi_{ij})^{-1}, \quad \mu_{ij} = \tau_{ij} z_{ij} \lambda_i Q_{ij},$$
$$T_j := \sum_{t=1}^{T} x_{j(t-1)}^2, \quad Q_{ij} := \sum_{t=1}^{T} x_{it}^{-j} x_{j(t-1)},$$
$$x_{it}^{-j} := x_{it} - \sum_{s \in \{1,\ldots,S\} \setminus j} w_{is} z_{is} x_{s(t-1)}.$$

It is noteworthy to mention when $z_{ij} = 0$, it is equivalent to sample w_{ij} from the prior $\mathcal{N}\left(0, \varphi_{ij}^{-1}\right)$.

8) ***Sample* $\mathbf{Z}$.** We sample z_{ij}, the (i,j)th element of $\mathbf{Z}$, as

$$(z_{ij} \,|\, -) \sim \text{Bernoulli}[p_{ij1}/(p_{ij1} + p_{ij0})], \tag{23}$$

where

$$p_{ij1} := e^{-\frac{1}{2}(w_{ij}^2 T_j \lambda_i - 2 w_{ij} \lambda_i Q_{ij})}(1 - e^{-\sum_\kappa r_\kappa \theta_{i\kappa} \psi_{j\kappa}}),$$
$$p_{ij0} := e^{-\sum_\kappa r_\kappa \theta_{i\kappa} \psi_{j\kappa}}.$$

9) ***Sample* m_{ij}.** We sample augmented count m_{ij} as

$$(m_{ij} \,|\, -) \sim z_{ij} \text{Pois}_+ \left(\sum_\kappa r_\kappa \theta_{i\kappa} \psi_{j\kappa} \right), \tag{24}$$

where $x \sim \text{Pois}_+(\lambda)$ is a truncated Poisson distribution with $P(x = k) = (1 - e^{-\lambda})^{-1} \lambda^k e^{-\lambda}/k!$ for $k \in \{1, 2, 3, ...\}$, which can be efficiently sampled from using a rejection sampler (Zhou, 2015).

10) ***Sample* $m_{ij\kappa}$.** We sample the latent counts, representing how strongly a potential transition from state j to state i would be associated with the κ^{th} community, as

$$(m_{ij1} \ldots m_{ijK} \,|\, -) \sim \text{Multinomial}\left(m_{ij}, \left[\frac{r_1 \theta_{i1} \psi_{j1}}{\sum_\kappa r_\kappa \theta_{i\kappa} \psi_{j\kappa}}, \ldots, \frac{r_K \theta_{iK} \psi_{jK}}{\sum_\kappa r_\kappa \theta_{i\kappa} \psi_{j\kappa}} \right] \right). \tag{25}$$

11) ***Sample* $\theta_{i\kappa}$.** We sample $\theta_{i\kappa}$ as

$$(\theta_{i\kappa} \,|\, -) \sim \text{Gamma}\left(\rho_i + \sum_{j=1}^{S} m_{ij\kappa}, \frac{1}{e_\kappa + r_\kappa \sum_{j=1}^{S} \psi_{j\kappa}} \right). \tag{26}$$

12) ***Sample*** $\psi_{j\kappa}$**.** We sample $\psi_{j\kappa}$ as

$$(\psi_{j\kappa} \,|\, -) \sim \text{Gamma}\left(\tau_j + \sum_{i=1}^{S} m_{ij\kappa}, \frac{1}{f_\kappa + r_\kappa \sum_{i=1}^{S} \theta_{i\kappa}}\right). \tag{27}$$

13) ***Sample*** r_κ**.** We sample r_κ as

$$(r_\kappa \,|\, -) \sim \text{Gamma}\left(\frac{\gamma_0}{K} + \sum_{i=1}^{S}\sum_{j=1}^{S} m_{ij\kappa}, \frac{1}{c + \sum_{i=1}^{S}\sum_{j=1}^{S} \theta_{i\kappa}\psi_{j\kappa}}\right). \tag{28}$$

14) ***Sample*** ρ_i**.** In order to sample ρ_i, we use a data augmentation technique introduced by Zhou and Carin (2013) for the negative binomial distribution. Specifically, we augment the model with CRT counts and then sample ρ_i as

$$(l_{i\kappa}^{(1)} \,|\, -) \sim \text{CRT}\left(\sum_{j=1}^{S} m_{ij\kappa}, \rho_i\right), \tag{29}$$

$$(\rho_i \,|\, -) \sim \text{Gamma}\left(\frac{\gamma_0}{S} + \sum_{\kappa=1}^{K} l_{i\kappa}^{(1)}, \frac{1}{c_\rho - \sum_{\kappa=1}^{K} \ln(1 - p_{i\kappa}^{(1)})}\right), \tag{30}$$

$$p_{i\kappa}^{(1)} := \frac{r_\kappa \sum_{j=1}^{S} \psi_{j\kappa}}{e_\kappa + r_\kappa \sum_{j=1}^{S} \psi_{j\kappa}}.$$

15) ***Sample*** τ_j**.** We sample τ_j as

$$(l_{j\kappa}^{(2)} \,|\, -) \sim \text{CRT}\left(\sum_{i=1}^{S} m_{ij\kappa}, \tau_j\right) \tag{31}$$

$$(\tau_j \,|\, -) \sim \text{Gamma}\left(\frac{\gamma_0}{S} + \sum_{\kappa=1}^{K} l_{j\kappa}^{(2)}, \frac{1}{c_\tau - \sum_{\kappa=1}^{K} \ln(1 - p_{j\kappa}^{(2)})}\right), \tag{32}$$

$$p_{j\kappa}^{(2)} := \frac{r_\kappa \sum_{i=1}^{S} \theta_{i\kappa}}{e_\kappa + r_\kappa \sum_{i=1}^{S} \theta_{i\kappa}}.$$

16) ***Sample*** $e_\kappa, f_\kappa, \lambda_i, \varphi_{ij}$**.** We sample these parameters as

$$(e_\kappa|-) \sim \text{Gamma}\left(\alpha_0 + \sum_{i=1}^{S} \rho_i, \frac{1}{\beta_0 + \sum_{i=1}^{S} \theta_{i\kappa}}\right), \tag{33}$$

$$(f_\kappa|-) \sim \text{Gamma}\left(\alpha_0 + \sum_{j=1}^{S} \tau_j, \frac{1}{\beta_0 + \sum_{j=1}^{S} \psi_{j\kappa}}\right), \tag{34}$$

$$(\lambda_i \,|-) \sim \text{Gamma}\Big(a + T/2, \frac{1}{b + \sum_{t=1}^{T}(x_{it} - (W \odot Z)_i \boldsymbol{x}_{t-1})^2/2)}\Big), \tag{35}$$

$$(\varphi_{ij} \,|-) \sim \text{Gamma}\big(\alpha_0 + 1/2, 1/(w_{ij}^2/2 + \beta_0)\big), \tag{36}$$

where $(W \odot Z)_i$ is the ith row of $\mathbf{W} \odot \mathbf{Z}$.