Deep Poisson gamma dynamical systems

Dandan Guo, **Bo Chen,**[*] **Hao Zhang**
National Laboratory of Radar Signal Processing
Collaborative Innovation Center of Information Sensing and Understanding
Xidian University, Xi'an, China
gdd_xidian@126.com, bchen@mail.xidian.edu.cn, zhanghao_xidian@163.com

Mingyuan Zhou
McCombs School of Business
The University of Texas at Austin
Austin, TX 78712, USA
mingyuan.zhou@mccombs.utexas.edu

Abstract

We develop deep Poisson-gamma dynamical systems (DPGDS) to model sequentially observed multivariate count data, improving previously proposed models by not only mining deep hierarchical latent structure from the data, but also capturing both first-order and long-range temporal dependencies. Using sophisticated but simple-to-implement data augmentation techniques, we derived closed-form Gibbs sampling update equations by first backward and upward propagating auxiliary latent counts, and then forward and downward sampling latent variables. Moreover, we develop stochastic gradient MCMC inference that is scalable to very long multivariate count time series. Experiments on both synthetic and a variety of real-world data demonstrate that the proposed model not only has excellent predictive performance, but also provides highly interpretable multilayer latent structure to represent hierarchical and temporal information propagation.

1 Introduction

The need to model time-varying count vectors $\boldsymbol{x}_1, ..., \boldsymbol{x}_T$ appears in a wide variety of settings, such as text analysis, international relation study, social interaction understanding, and natural language processing [1–9]. To model these count data, it is important to not only consider the sparsity of high-dimensional data and robustness to over-dispersed temporal patterns, but also capture complex dependencies both within and across time steps. In order to move beyond linear dynamical systems (LDS) [10] and its nonlinear generalization [11] that often make the Gaussian assumption [12], the gamma process dynamic Poisson factor analysis (GP-DPFA) [5] factorizes the observed time-varying count vectors under the Poisson likelihood as $\boldsymbol{x}_t \sim \text{Poisson}(\boldsymbol{\Phi}\boldsymbol{\theta}_t)$, and transmit temporal information smoothly by evolving the factor scores with a gamma Markov chain as $\boldsymbol{\theta}_t \sim \text{Gamma}(\boldsymbol{\theta}_{t-1}, \boldsymbol{\beta})$, which has highly desired strong non-linearity. To further capture cross-factor temporal dependence, a transition matrix $\boldsymbol{\Pi}$ is further used in Poisson–gamma dynamical system (PGDS) [7] as $\boldsymbol{\theta}_t \sim \text{Gamma}(\boldsymbol{\Pi}\boldsymbol{\theta}_{t-1}, \boldsymbol{\beta})$. However, these shallow models may still have shortcomings in capturing long-range temporal dependencies [8]. For example, if given $\boldsymbol{\theta}_t$, then $\boldsymbol{\theta}_{t+1}$ no longer depends on $\boldsymbol{\theta}_{t-k}$ for all $k \geq 1$.

Deep probabilistic models are widely used to capture the relationships between latent variables across multiple stochastic layers [4, 8, 13–16]. For example, deep dynamic Poisson factor analysis (DDPFA)

[*] Corresponding author

[8] utilizes recurrent neural networks (RNN) [3] to capture long-range temporal dependencies of the factor scores. The latent variables and RNN parameters, however, are separately inferred. Deep temporal sigmoid belief network (DTSBN) [4] is a deep dynamic generative model defined as a sequential stack of sigmoid belief networks (SBNs), whose hidden units are typically restricted to be binary. Although a deep structure is designed to describe complex long-range temporal dependencies, how the layers in DTSBN are related to each other lacks an intuitive interpretation, which is of paramount interest for a multilayer probabilistic model [15].

In this paper, we present deep Poisson gamma dynamical systems (DPGDS), a deep probabilistic dynamical model that takes the advantage of the hierarchical structure to efficiently incorporate both between-layer and temporal dependencies, while providing rich interpretation. Moving beyond DTSBN using binary hidden units, we build a deep dynamic directed network with gamma distributed nonnegative real hidden units, inferring a multilayer contextual representation of multivariate time-varying count vectors. Consequently, DPGDS can handle highly overdispersed counts, capturing the correlations between the visible/hidden features across layers and over times using the gamma belief network [15]. Combing the deep and temporal structures shown in Fig. 1(a), DPGDS breaks the assumption that given $\boldsymbol{\theta}_t$, $\boldsymbol{\theta}_{t+1}$ no longer depends on $\boldsymbol{\theta}_{t-k}$ for $k \geq 1$, suggesting that it may better capture long-range temporal dependencies. As a result, the model can allow more specific information, which are also more likely to exhibit fast temporal changing, to transmit through lower layers, while allowing more general information, which are also more likely to slowly evolve over time, to transmit through higher layers. For example, as shown in Fig. 1(b) that is learned from GDELT2003 with DPGDS, when analyzing these international events, the factors at lower layers are more specific to discover the relationships between the different countries, whereas those at higher layers are more general to reflect the conflicts between the different areas consisting of several related countries, or the ones occurring simultaneously, and the latent representation $\boldsymbol{\theta}_t$ at a lower layer varies more intensely than that at a higher layer.

Distinct from DDPFA [8] that adopts a two-stage inference, the latent variables of DPGDS can be jointly trained with both a Backward-Upward–Forward-Downward (BUFD) Gibbs sampler and a sophisticated stochastic gradient MCMC (SGMCMC) algorithm that is scalable to very long multivariate time series [17–21]. Furthermore, the factors learned at each layer can refine the understanding and analysis of sequentially observed multivariate count data, which, to the best of our knowledge, may be very challenging for existing methods. Finally, based on a diverse range of real-world data sets, we show that DPGDS exhibits excellent predictive performance, inferring interpretable latent structure with well captured long-range temporal dependencies.

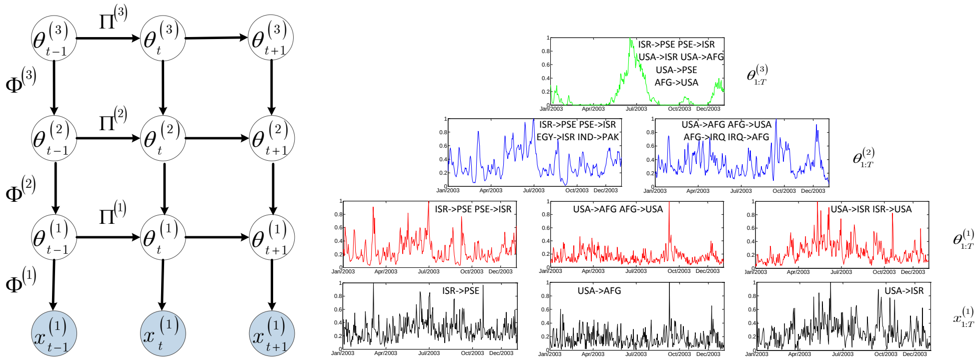
2 Deep Poisson gamma dynamic systems

Shown in Fig. 1(a) is the graphical representation of a three-hidden-layer DPGDS. Let us denote $\theta \sim \text{Gam}(a, c)$ as a gamma random variable with mean a/c and variance a/c^2. Given a set of V-dimensional sequentially observed multivariate count vectors $\boldsymbol{x}_1, ..., \boldsymbol{x}_T$, represented as a $V \times T$ matrix $\mathbf{X}$, the generative process of a L-hidden-layer DPGDS, from top to bottom, is expressed as

$$\boldsymbol{\theta}_t^{(L)} \sim \text{Gam}\left(\tau_0 \boldsymbol{\Pi}^{(L)} \boldsymbol{\theta}_{t-1}^{(L)}, \tau_0\right), \cdots, \ \boldsymbol{\theta}_t^{(l)} \sim \text{Gam}\left(\tau_0 (\boldsymbol{\Phi}^{(l+1)} \boldsymbol{\theta}_t^{(l+1)} + \boldsymbol{\Pi}^{(l)} \boldsymbol{\theta}_{t-1}^{(l)}), \tau_0\right), \cdots,$$
$$\boldsymbol{\theta}_t^{(1)} \sim \text{Gam}\left(\tau_0 (\boldsymbol{\Phi}^{(2)} \boldsymbol{\theta}_t^{(2)} + \boldsymbol{\Pi}^{(1)} \boldsymbol{\theta}_{t-1}^{(1)}), \tau_0\right), \ \boldsymbol{x}_t^{(1)} \sim \text{Pois}\left(\delta_t^{(1)} \boldsymbol{\Phi}^{(1)} \boldsymbol{\theta}_t^{(1)}\right), \tag{1}$$

where $\boldsymbol{\Phi}^{(l)} \in \mathbb{R}_+^{K_{l-1} \times K_l}$ is the factor loading matrix at layer l, $\boldsymbol{\theta}_t^{(l)} \in \mathbb{R}_+^{K_l}$ the hidden units of layer l at time t, and $\boldsymbol{\Pi}^{(l)} \in \mathbb{R}_+^{K_l \times K_l}$ a transition matrix of layer l that captures cross-factor temporal dependencies. We denote $\delta_t^{(1)} \in \mathbb{R}_+$ as a scaling factor, reflecting the scale of the counts at time t; one may also set $\delta_t^{(1)} = \delta^{(1)}$ for $t = 1, ..., T$. We denote $\tau_0 \in \mathbb{R}_+$ as a scaling hyperparameter that controls the temporal variation of the hidden units. The multilayer time-varying hidden units $\boldsymbol{\theta}_t^{(l)}$ are well suited for downstream analysis, as will be shown below.

DPGDS factorizes the count observation $\boldsymbol{x}_t^{(1)}$ into the product of $\delta_t^{(1)}$, $\boldsymbol{\Phi}^{(1)}$, and $\boldsymbol{\theta}_t^{(1)}$ under the Poisson likelihood. It further factorizes the shape parameters of the gamma distributed $\boldsymbol{\theta}_t^{(l)}$ of layer l at time t into the sum of $\boldsymbol{\Phi}^{(l+1)} \boldsymbol{\theta}_t^{(l+1)}$, capturing the dependence between different layers, and $\boldsymbol{\Pi}^{(l)} \boldsymbol{\theta}_{t-1}^{(l)}$, capturing the temporal dependence at the same layer. At the top layer, $\boldsymbol{\theta}_t^{(L)}$ is only



(a) (b)

Figure 1: Graphical model and illustration for a three-hidden-layer deep Poisson Gamma Dynamical System (DPGDS). (a) The generative model; (b) Visualization of data and latent factors learned from GDELT2003, with the black, red, blue and green lines denoting the observed data, temporal trajectories of example latent factors at layer 1, 2, 3, respectively.

dependent on $\boldsymbol{\Pi}^{(L)}\boldsymbol{\theta}_{t-1}^{(L)}$, and at $t=1$, $\boldsymbol{\theta}_1^{(l)} \sim \text{Gam}\left(\tau_0 \boldsymbol{\Phi}^{(l+1)}\boldsymbol{\theta}_1^{(l+1)}, \tau_0\right)$ for $l = 1, \ldots, L-1$ and $\boldsymbol{\theta}_1^{(L)} \sim \text{Gam}\left(\tau_0 \nu_k^{(L)}, \tau_0\right)$. To complete the hierarchical model, we introduce K_l factor weights $\boldsymbol{\nu}^{\,(l)} = (\nu_1^{(l)}, ..., \nu_{K_l}^{(l)})$ in layer l to model the strength of each factor, and for $l = 1, ..., L$, we let

$$\boldsymbol{\pi}_k^{(l)} \sim \text{Dir}(\nu_1^{(l)}\nu_k^{(l)}, ..., \nu_{k-1}^{(l)}\nu_k^{(l)}, \xi^{(l)}\nu_k^{(l)}, \nu_{k+1}^{(l)}\nu_k^{(l)} ..., \nu_{K_l}^{(l)}\nu_k^{(l)}), \;\; \nu_k^{(l)} \sim \text{Gam}(\tfrac{\gamma_0}{K_l}, \beta^{(l)}). \tag{2}$$

Note that $\boldsymbol{\pi}_k^{(l)}$ is the k^{th} column of $\boldsymbol{\Pi}^{(l)}$ and $\pi_{k_1 k_2}^{(l)}$ can be interpreted as the probability of transiting from topic k_2 of the previous time to topic k_1 of the current time at layer l.

Finally, we place Dirichlet priors on the factor loadings and draw other parameters from a noninformative gamma prior: $\boldsymbol{\phi}_k^{(l)} = (\phi_{1k}^{(l)}, ..., \phi_{K_{l-1}k}^{(l)}) \sim \text{Dir}(\eta^{(l)}, ..., \eta^{(l)})$, and $\delta_t^{(1)}, \xi^{(l)}, \beta^{(l)} \sim \text{Gam}(\epsilon_0, \epsilon_0)$. Note that imposing Dirichlet distributions on the columns of $\boldsymbol{\Pi}^{(l)}$ and $\boldsymbol{\Phi}^{(l)}$ not only makes the latent representation more identifiable and interpretable, but also facilitates inference, as will be shown in the next section. Clearly when $L = 1$, DPGDS reduces to PGDS [7]. In real-world applications, a binary observation can be linked to a latent count using the Bernoulli-Poisson link as $b = 1(n \geq 1), n \sim \text{Pois}(\lambda)$ [22]. Nonnegative-real-valued matrix can also be linked to a latent count matrix via a Poisson randomized gamma distribution as $x \sim \text{Gam}(n, c), n \sim \text{Pois}(\lambda)$ [23].

Hierarchical structure: To interpret the hierarchical structure of (1), we notice that $\mathbb{E}\left[\boldsymbol{x}_t^{(1)} \,|\, \boldsymbol{\theta}_t^{(l)}, \{\boldsymbol{\Phi}^{(p)}\}_{p=1}^{l}\right] = \left[\prod_{p=1}^{l} \boldsymbol{\Phi}^{(p)}\right] \boldsymbol{\theta}_t^{(l)}$ if the temporal structure is ignored. Thus it is straightforward to interpret $\boldsymbol{\phi}_k^{(l)}$ by projecting them to the bottom data layer as $\left[\prod_{t=1}^{l-1} \boldsymbol{\Phi}^{(t)}\right] \boldsymbol{\phi}_k^{(l)}$, which are often quite specific at the bottom layer and become increasingly more general when moving upwards, as will be shown below in Fig. 5(a).

Long-range temporal dependencies: Using the law of total expectations on (1), for a three-hidden-layer DPGDS shown in Fig. 1(a), we have

$$\begin{aligned}\mathbb{E}[\boldsymbol{x}_t^{(1)} \,|\, \boldsymbol{\theta}_{t-1}^{(1)}, \boldsymbol{\theta}_{t-2}^{(2)}, \boldsymbol{\theta}_{t-3}^{(3)}]/\delta_t^{(1)} = \boldsymbol{\Phi}^{(1)}\boldsymbol{\Pi}^{(1)}\boldsymbol{\theta}_{t-1}^{(1)} + \boldsymbol{\Phi}^{(1)}\boldsymbol{\Phi}^{(2)}[\boldsymbol{\Pi}^{(2)}]^2\boldsymbol{\theta}_{t-2}^{(2)} \\ + \boldsymbol{\Phi}^{(1)}\boldsymbol{\Phi}^{(2)}(\boldsymbol{\Pi}^{(2)}\boldsymbol{\Phi}^{(3)} + \boldsymbol{\Phi}^{(3)}\boldsymbol{\Pi}^{(3)})[\boldsymbol{\Pi}^{(3)}]^2\boldsymbol{\theta}_{t-3}^{(3)},\end{aligned} \tag{3}$$

which suggests that $\{\boldsymbol{\Pi}^{(l)}\}_{l=1}^{L}$ play the role of transiting the latent representation across time and, different from most existing dynamic models, DPGDS can capture and transmit long-range temporal information (often general and change slowly over time) through its higher hidden layers.

3 Scalable MCMC inference

In this paper, in each iteration, across layers and times, we first exploit a variety of data augmentation techniques for count data to "backward" and "upward" propagate auxiliary latent counts, with which

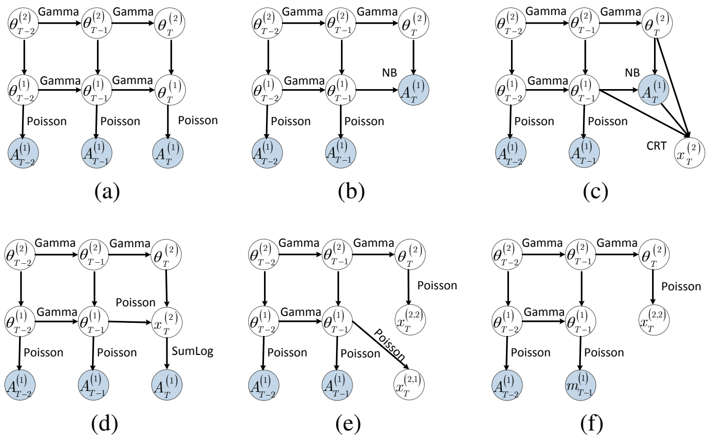


Figure 2: Graphical representation of the model and data augmentation and marginalization based inference scheme. (a) An alternative representation of layer $l = 1$ using the relationships between the Poisson and multinomial distributions; (b) A negative binomial distribution based representation that marginalizes out the gamma from the Poisson distributions, corresponding to (4) for $t = T$; (c) An equivalent representation that introduces CRT distributed auxiliary variables, corresponding to (5); (d) An equivalent representation using **P3**, corresponding to (6); (e) An equivalent representation obtained by using **P1**, corresponding to (7); (f) A representation obtained by repeating the same augmentation-marginalization steps described in (a).

we then “downward” and “forward” sample latent variables, leading to a Backward-Upward–Forward-Downward Gibbs sampling (BUFD) Gibbs sampling algorithm.

3.1 Backward and upward propagation of latent counts

Different from PGDS that has only backward propagation for latent counts, DPGDS have both backward and upward ones due to its deep hierarchical structure. To derive closed-form Gibbs sampling update equations, we exploit three useful properties for count data, denoted as **P1**, **P2**, and **P3** [7, 24], respectively, as presented in the Appendix. Let us denote $x \sim \mathrm{NB}(r, p)$ as the negative binomial distribution with probability mass function $P(x = k) = \frac{\Gamma(k+r)}{k!\Gamma(r)} p^k (1-p)^r$, where $k \in \{0, 1, \ldots\}$. First, we can augment each count $x^{(1)}_{vt}$ in (1) into the summation of K_1 latent counts that are smaller or equal as $x^{(1)}_{vt} = \sum_{k=1}^{K_1} A^{(1)}_{vkt}$, $A^{(1)}_{vkt} \sim \mathrm{Pois}(\delta^{(1)}_t \phi^{(1)}_{vk} \theta^{(1)}_{kt})$, with $A^{(1)}_{\cdot kt} = \sum_{v=1}^{V} A^{(1)}_{vkt}$. Since $\sum_{v=1}^{V} \phi^{(1)}_{vk} = 1$ by construction, we also have $A^{(1)}_{\cdot kt} \sim \mathrm{Pois}(\delta^{(1)}_t \theta^{(1)}_{kt})$, as shown in Fig. 2(a). We start with $\boldsymbol{\theta}^{(1)}_T$ at the last time point T, as none of the other time-step factors depend on it in their priors. Via **P2**, as shown in Fig. 2(b), we can marginalize out $\theta^{(1)}_{kT}$ to obtain

$$A^{(1)}_{\cdot kT} \sim \mathrm{NB}\left[\tau_0 \left(\sum\nolimits_{k_2=1}^{K_2} \phi^{(2)}_{kk_2} \theta^{(2)}_{k_2 T} + \sum\nolimits_{k_1=1}^{K_1} \pi^{(1)}_{kk_1} \theta^{(1)}_{k_1, T-1}\right),\ g(\zeta^{(1)}_T)\right], \tag{4}$$

where $\zeta^{(1)}_T = \ln(1 + \frac{\delta^{(1)}_T}{\tau_0})$ and $g\left(\zeta\right) = 1 - \exp\left(-\zeta\right)$.

In order to marginalize out $\boldsymbol{\theta}^{(1)}_{T-1}$, as shown in Fig. 2(c), we introduce an auxiliary variable following the Chines restaurant table (CRT) distribution [24] as

$$x^{(2)}_{kT} \sim \mathrm{CRT}\left[A^{(1)}_{\cdot kT},\ \tau_0 \left(\sum\nolimits_{k_2=1}^{K_2} \phi^{(2)}_{kk_2} \theta^{(2)}_{k_2 T} + \sum\nolimits_{k_1=1}^{K_1} \pi^{(1)}_{kk_1} \theta^{(1)}_{k_1, T-1}\right)\right]. \tag{5}$$

As shown in Fig. 2(d), we re-express the joint distribution over $A^{(1)}_{\cdot kT}$ and $x^{(2)}_{kT}$ according to **P3** as

$$A^{(1)}_{\cdot kT} \sim \mathrm{SumLog}(x^{(2)}_{kT}, g(\zeta^{(1)}_T)),\ \ x^{(2)}_{kT} \sim \mathrm{Pois}\left[\zeta^{(1)}_T \tau_0 \left(\sum\nolimits_{k_2=1}^{K_2} \phi^{(2)}_{kk_2} \theta^{(2)}_{k_2 T} + \sum\nolimits_{k_1=1}^{K_1} \pi^{(1)}_{kk_1} \theta^{(1)}_{k_1, T-1}\right)\right], \tag{6}$$

where the sum-logarithmic (SumLog) distribution is defined as in Zhou and Carin [24]. Via **P1**, as in Fig. 2(e), the Poisson random variable $x_{kT}^{(2)}$ in (6) can be augmented as $x_{kT}^{(2)} = x_{kT}^{(2,1)} + x_{kT}^{(2,2)}$, where

$$x_{kT}^{(2,1)} \sim \text{Pois}(\zeta_T^{(1)} \tau_0 \sum\nolimits_{k_1=1}^{K_1} \pi_{kk_1}^{(1)} \theta_{k_1,T-1}^{(1)}), \;\; x_{kT}^{(2,2)} \sim \text{Pois}(\zeta_T^{(1)} \tau_0 \sum\nolimits_{k_2=1}^{K_2} \phi_{kk_2}^{(2)} \theta_{k_2T}^{(2)}). \tag{7}$$

It is obvious that due to the deep dynamic structure, the count at layer two $x_{kT}^{(2)}$ is divided into two parts: one from time $T-1$ at layer one, while the other from time T at layer two. Furthermore, $\zeta_T^{(1)}$ is the scaling factor at layer two, which is propagated from the one at layer one $\delta_T^{(1)}$. Repeating the process all the way back to $t=1$, and from $l=1$ up to $l=L$, we are able to marginalize out all gamma latent variables $\{\mathbf{\Theta}\}_{t=1,l=1}^{T,L}$ and provide closed-form conditional posteriors for all of them.

3.2 Backward-upward–forward-downward Gibbs sampling

Sampling auxiliary counts: This step is about the "backward" and "upward" pass. Let us denote $Z_{\cdot kt}^{(l)} = \sum_{k_l=1}^{K_l} Z_{k_l kt}^{(l)}$, $Z_{\cdot k,T+1}^{(l)} = 0$, and $x_{kt}^{(1,1)} = x_{vt}^{(1)}$. Working backward for $t = T, ..., 2$ and upward for $l = 1, ..., L$, we draw

$$(A_{k1t}^{(l)}, ..., A_{kK_lt}^{(l)}) \sim \text{Multi}\left(x_{kt}^{(l,l)}; \frac{\phi_{k1}^{(l)} \theta_{1t}^{(l)}}{\sum_{k_l=1}^{K_l} \phi_{kk_l}^{(l)} \theta_{k_lt}^{(l)}}, ..., \frac{\phi_{kK_l}^{(l)} \theta_{K_lt}^{(l)}}{\sum_{k_l=1}^{K_l} \phi_{kk_l}^{(l)} \theta_{k_lt}^{(l)}} \right), \tag{8}$$

$$x_{kt}^{(l+1)} \sim \text{CRT}\left[A_{\cdot kt}^{(l)} + Z_{\cdot k,t+1}^{(l)}, \tau_0 \left(\sum\nolimits_{k_{l+1}=1}^{K_{l+1}} \phi_{kk_{l+1}}^{(l+1)} \theta_{k_{l+1}t}^{(l+1)} + \sum\nolimits_{k_l=1}^{K_l} \pi_{kk_1}^{(l)} \theta_{k_1,t-1}^{(l)} \right) \right]. \tag{9}$$

Note that via the deep structure, the latent counts $x_{kt}^{(l+1)}$ will be influenced by the effects from both of time $t-1$ at layer l and time t at layer $l+1$. With $p_1 := \sum_{k_l=1}^{K_l} \pi_{kk_l}^{(l)} \theta_{k_l,t-1}^{(l)}$ and $p_2 := \sum_{k_{l+1}=1}^{K_{l+1}} \phi_{kk_{l+1}}^{(l+1)} \theta_{k_{l+1}t}^{(l+1)}$, we can sample the latent counts at layer l and $l+1$ by

$$(x_{kt}^{(l+1,l)}, x_{kt}^{(l+1,l+1)}) \sim \text{Multi}\left(x_{kt}^{(l+1)}, p_1/(p_1+p_2), p_2/(p_1+p_2) \right), \tag{10}$$

and then draw

$$(Z_{k1t}^{(l)}, ..., Z_{kK_lt}^{(l)}) \sim \text{Multi}\left(x_{kt}^{(l+1,l)}; \frac{\pi_{k1}^{(l)} \theta_{1,t-1}^{(l)}}{\sum_{k_l=1}^{K_l} \pi_{kk_l}^{(l)} \theta_{k_l,t-1}^{(l)}}, ..., \frac{\pi_{kK_l}^{(l)} \theta_{K_l,t-1}^{(l)}}{\sum_{k_l=1}^{K_l} \pi_{kk_l}^{(l)} \theta_{k_l,t-1}^{(l)}} \right). \tag{11}$$

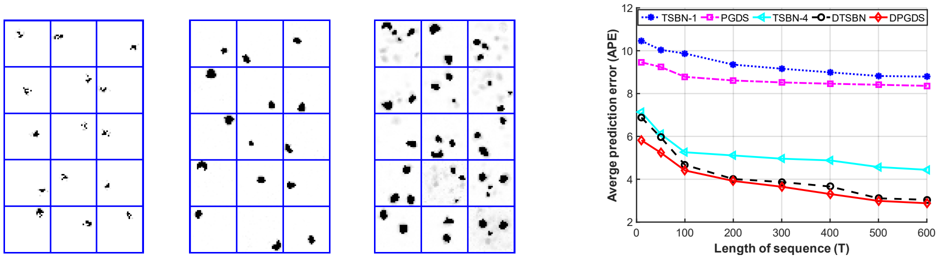
Sampling hidden units $\boldsymbol{\theta}_t^{(l)}$ and calculating $\zeta_t^{(l)}$: Given the augmented latent count variables, working forward for $t = 1, ..., T$ and downward for $l = L, ..., 1$, we can sample

$$\begin{aligned}\theta_{kt}^{(l)} \sim \text{Gamma}\Big[& A_{\cdot kt}^{(l)} + Z_{\cdot k(t+1)}^{(l)} + \tau_0 \Big(\sum\nolimits_{k_{l+1}=1}^{K_{l+1}} \phi_{kk_{l+1}}^{(l+1)} \theta_{k_{l+1}t}^{(l+1)} + \sum\nolimits_{k_l=1}^{K_l} \pi_{kk_l}^{(l)} \theta_{k_2,t-1}^{(l)} \Big), \\ & \tau_0 \big(1 + \zeta_t^{(l-1)} + \zeta_{t+1}^{(l)} \big) \Big],\end{aligned} \tag{12}$$

where $\zeta_t^{(0)} = \frac{\delta_t^{(1)}}{\tau_0}$ and $\zeta_t^{(l)} = \ln\left(1 + \zeta_t^{(l-1)} + \zeta_{t+1}^{(l)}\right)$. Note if $\delta_t^{(1)} = \delta^{(1)}$ for $t = 1, ..., T$, then we may let $\zeta^{(l)} = -\mathbf{W}_{-1}(-\exp(-1-\zeta^{(l-1)})) - 1 - \zeta^{(l-1)}$, where the function $\mathbf{W}_{-1}$ is the lower real part of the Lambert W function [7,25]. From (12), we can find that the conditional posterior of $\boldsymbol{\theta}_t^{(l)}$ is parameterized by not only both $\mathbf{\Phi}^{(l+1)} \boldsymbol{\theta}_t^{(l+1)}$ and $\mathbf{\Pi}^{(l)} \boldsymbol{\theta}_{t-1}^{(l)}$, which represent the information from layer $l+1$ (downward) and time $t-1$ (forward), respectively, but also both $A_{\cdot,:,t}^{(l)}$ and $Z_{\cdot,:,t+1}^{(l)}$, which record the message from layer $l-1$ (upward) in (8) and time $t+1$ (backward) in (11), respectively. We describe the BUFD Gibbs sampling algorithm for DPGDS in Algorithm 1 and provide more details in the Appendix.

3.3 Stochastic gradient MCMC inference

Although the proposed BUFD Gibbs sampling algorithm for DPGDS has closed-form update equations, it requires processing all time-varying vectors at each iteration and hence has limited scalability [26]. To allow for scalable inference, we apply the topic-layer-adaptive stochastic gradient



(a) (b)

Figure 3: Results on the bouncing ball data set. (a) Shown in the first to third columns are the top fifteen latent factors learned by a three-hidden-layer DPGDS at layers 1, 2, and 3, respectively; (b) The average prediction errors as a function of the sequence length for various algorithms.

Riemannian (TLASGR) MCMC algorithm described in Cong et al. [27] and Zhang et al. [26], which can be used to sample simplex-constrained global parameters [28] in a mini-batch based manner. It improves its sampling efficiency via the use of the Fisher information matrix (FIM) [29], with adaptive step-sizes for the latent factors and transition matrices of different layers. More specifically, for $\boldsymbol{\pi}_k^{(l)}$, column k of the transition matrix $\boldsymbol{\Pi}^{(l)}$ of layer l, its sampling can be efficiently realized as

$$\begin{aligned}\left(\boldsymbol{\pi}_k^{(l)}\right)_{n+1} = & \Bigg[\left(\boldsymbol{\pi}_k^{(l)}\right)_n + \frac{\varepsilon_n}{M_k^{(l)}}\left[\left(\rho\tilde{\boldsymbol{z}}_{:k\cdot}^{(l)} + \boldsymbol{\eta}_{:k}^{(l)}\right) - \left(\rho\tilde{z}_{\cdot k\cdot}^{(l)} + \eta_{\cdot k}^{(l)}\right)\left(\boldsymbol{\pi}_k^{(l)}\right)_n\right] \\ & + \mathcal{N}\left(0, \frac{2\varepsilon_n}{M_k^{(l)}}\left[\operatorname{diag}(\boldsymbol{\pi}_k^{(l)})_n - (\boldsymbol{\pi}_k^{(l)})_n(\boldsymbol{\pi}_k^{(l)})_n^T\right]\right)\Bigg]_\angle,\end{aligned} \tag{13}$$

where $M_k^{(l)}$ is calculated using the estimated FIM, both $\tilde{\boldsymbol{z}}_{:k\cdot}^{(l)}$ and $\tilde{z}_{\cdot k\cdot}^{(l)}$ come from the augmented latent counts $Z^{(l)}$, $[.]_\angle$ denotes a simplex constraint, and $\boldsymbol{\eta}_{:k}^{(l)}$ denotes the prior of $\boldsymbol{\pi}_k^{(l)}$. The update of $\boldsymbol{\Phi}^{(l)}$ is the same with Cong et al. [27], and all the other global parameters are sampled using SGNHT [20]. We provide the details of the SGMCMC for DPGDS in Algorithm 2 in the Appendix.

4 Experiments

In this section, we present experimental results on a synthetic dataset and five real-world datasets. For a fair comparison, we consider PGDS [7], GP-DPFA [5], DTSBN [4], and GPDM [11] that can be considered as a dynamic generalization of the Gaussian process latent variable model of Lawrence [30], using the code provided by the authors. Note that as shown Schein et al. [7] and Gan et al. [4], PGDS and DTSBN are state-of-the-art count time series modeling algorithms that outperform a wide variety of previously proposed ones, such as LDS [12] and DRFM [31]. The hyperparameter settings of PGDS, GP-DPFA, GPDM, TSBN, and DTSBN are the same as their original settings [4, 5, 7, 11]. For DPGDS, we set $\tau_0 = 1, \gamma_0 = 100, \eta_0 = 0.1$ and $\epsilon_0 = 0.1$. We use $[K^{(1)}, K^{(2)}, K^{(3)}] = [200, 100, 50]$ for both DPGDS and DTSBN and $K = 200$ for PGDS, GP-DPFA, GPDM, and TSBN. For PGDS, GP-DPFA, GPDM, and DPGDS, we run 2000 Gibbs sampling as burn-in and collect 3000 samples for evaluation. We also use SGMCMC to infer DPGDS, with 5000 collection samples after 5000 burn-in steps, and use 10000 SGMCMC iterations for both TSBN and DTSBN to evaluate their performance.

4.1 Synthetic dataset

Following the literature [1, 4], we consider sequences of different lengths, including $T = 10, 50, 100, 200, 300, 400, 500$ and 600, and generate 50 synthetic bouncing ball videos for training, and 30 ones for testing. Each video frame is a binary-valued image with size 30×30, describing the location of three balls within the image. Both TSBN and DTSBN model it with the Bernoulli likelihood, while both PGDS and DPGDS use the Bernoulli-Poisson link [22].

As shown in Fig. 3(b), the average prediction errors of all algorithms decrease as the training sequence length increases. A higher-order TSBN, TSBN-4, performs much better than the first-order TSBN

Table 1: Top-M results on real-world text data

Model	Top-M	GDELT ($T = 365$)	ICEWS ($T = 365$)	SOTU ($T = 225$)	DBLP ($T = 14$)	NIPS ($T = 17$)
GPDPFA	MP	0.611 ±0.001	0.607 ±0.002	0.379 ±0.002	**0.435** ±0.009	0.843 ±0.005
	MR	0.145 ±0.002	0.235 ±0.005	0.369 ±0.002	0.254 ±0.005	0.050 ±0.001
	PP	0.447 ±0.014	0.465 ±0.008	0.617 ±0.013	0.581 ±0.011	0.807 ±0.006
PGDS	MP	0.679 ±0.001	0.658 ±0.001	0.375 ±0.002	0.419 ±0.004	0.864 ±0.004
	MR	**0.150** ±0.001	**0.245** ±0.005	0.373 ±0.002	0.252 ±0.004	0.050 ±0.001
	PP	0.420 ±0.017	0.455 ±0.008	0.612 ±0.018	0.566 ±0.008	0.802 ±0.020
GPDM	MP	0.520±0.001	0.530 ±0.002	0.274 ±0.001	0.388 ±0.004	0.355 ±0.008
	MR	0.141 ±0.001	0.234 ±0.001	0.261 ±0.002	0.146 ±0.005	0.050 ±0.001
	PP	0.362±0.021	0.185±0.017	0.587 ±0.016	0.509 ±0.008	0.384 ±0.028
TSBN	MP	0.594 ±0.007	0.471 ±0.001	0.360 ±0.001	0.403 ±0.012	0.788 ±0.005
	MR	0.124 ±0.001	0.158 ±0.001	0.275 ±0.001	0.194 ±0.001	0.050 ±0.001
	PP	0.418 ±0.019	0.445 ±0.031	0.611 ±0.001	0.527 ±0.003	0.692 ±0.017
DTSBN-2	MP	0.439 ±0.001	0.475 ±0.002	0.370 ±0.004	0.407 ±0.003	0.756 ±0.001
	MR	0.134 ±0.001	0.208 ±0.001	0.361 ±0.001	0.248 ±0.007	0.050 ±0.001
	PP	0.391 ±0.001	0.446 ±0.001	0.587 ±0.027	0.522 ±0.005	0.737 ±0.004
DTSBN-3	MP	0.411 ±0.001	0.431 ±0.001	**0.450** ±0.008	0.390 ±0.002	0.774 ±0.002
	MR	0.141 ±0.001	0.189 ±0.001	0.274 ±0.001	0.252 ±0.004	0.050 ±0.001
	PP	0.367 ±0.011	0.451 ±0.026	0.548 ±0.013	0.510 ±0.006	0.715 ±0.009
DPGDS-2	MP	0.688 ±0.002	0.659 ±0.001	0.379 ±0.002	0.430 ±0.009	0.867 ±0.008
	MR	0.149 ±0.001	0.242 ±0.007	0.373 ±0.001	0.254 ±0.005	0.050 ±0.001
	PP	0.443 ±0.025	0.473 ±0.012	0.622 ±0.014	0.582 ±0.007	0.814 ±0.035
DPGDS-3	MP	**0.689** ±0.002	0.660 ±0.001	0.380 ±0.001	0.431 ±0.012	**0.887** ±0.002
	MR	**0.150** ±0.001	0.244 ±0.003	**0.374** ±0.002	**0.255** ±0.004	0.050 ±0.001
	PP	**0.456** ±0.015	**0.478** ±0.024	**0.628** ±0.021	**0.600** ±0.001	**0.839** ±0.007

does, suggesting that using high-order messages can help TSBN better pass useful information. As discussed above, since a deep structure provides a natural way to propagate high-order information for prediction, it is not surprising to find that both DTSBN and DPGDS, which are both multi-layer models, have exhibited superior performance. Moreover, it is clear that the proposed DPGDS consistently outperforms DTSBN under all settings.

Another advantage of DPGDS is that its inferred deep latent structure often has meaningful interpretation. As shown in Fig. 3(a), for the bouncing ball data, the inferred factors at layer one represent points or pixels, those at layer two cover larger spatial contiguous regions, some of which exhibit the shape of a single bouncing ball, and those at layer three are able to capture multiple bouncing balls. In addition, we show in Appendix B the one-step prediction frames of different models.

4.2 Real-world datasets

Besides the binary-valued synthetic bouncing ball dataset, we quantitatively and qualitatively evaluate all algorithms on the following real-world datasets used in Schein et al. [7]. The State-of-the-Union (SOTU) dataset consists of the text of the annual SOTU speech transcripts from 1790 to 2014. The Global Database of Events, Language, and Tone (GDELT) and Integrated Crisis Early Warning System (ICEWS) are both datasets for international relations extracted from news corpora. Note that ICEWS consists of undirected pairs, while GDELT consists of directed pairs of countries. The NIPS corpus contains the text of every NIPS conference paper from 1987 to 2003. The DBLP corpus is a database of computer science research papers. Each of these datasets is summarized as a $V \times T$ count matrix, as shown in Tab. 1. Unless specified otherwise, we choose the top 1000 most frequently used terms to form the vocabulary, which means we set $V = 1000$ for all real-data experiments.

4.2.1 Quantitative comparison

For a fair and comprehensive comparison, we calculate the precision and recall at top-M [4,5,31,32], which are calculated by the fraction of the top-M words that match the true ranking of the words and appear in the top-M ranking, respectively, with $M = 50$. We also use the Mean Precision (MP) and Mean Recall (MR) over all the years appearing in the training set to evaluate different models. As another criterion, the Predictive Precision (PP) shows the predictive precision for the final year, for which all the observations are held out. Similar as previous methods [4,5], for each corpus, the entire data of the last year is held out, and for the documents in the previous years we randomly partition the words of each document into 80% / 20% in each trial, and we conduct five random trials to report the sample mean and standard deviation. Note that to apply GPDM, we have used Anscombe transform

[33] to preprocess the count data to mitigate the mismatch between the data and model assumption. The results on all five datasets are summarized in Tab. 1, which clearly show that the proposed DPGDS has achieved the best performance on most of the evaluation criteria, and again a deep model often improves its performance by increasing its number of layers. To add more empirical study on scalability, we have also tested the efficiency of our model on a GDELT data (from 2001 to 2005, temporal granularity of 24 hrs, with a total of 1825 time points), which is not too large so that we can still run DPGDS-Gibbs and GPDM. As shown in Fig. 4, we present how various algorithms progress over time, evaluated with MP. It takes about 1000s for DTSBN and DPGDS-SGMCMC to converge, 3.5 hrs for DPGDS-Gibbs, 5 hrs for GPDM. Clearly, our DPGDS-SGMCMC is scalable and clearly outperforms both DTSBN and GPDM. We also present in Appendix C the results of DPGDS-SGMCMC on a very long time series, on which it becomes too expensive to run a batch learning algorithm.

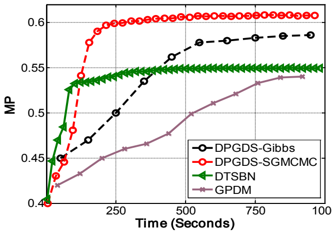


Figure 4: MP as a function of time for GDELT.

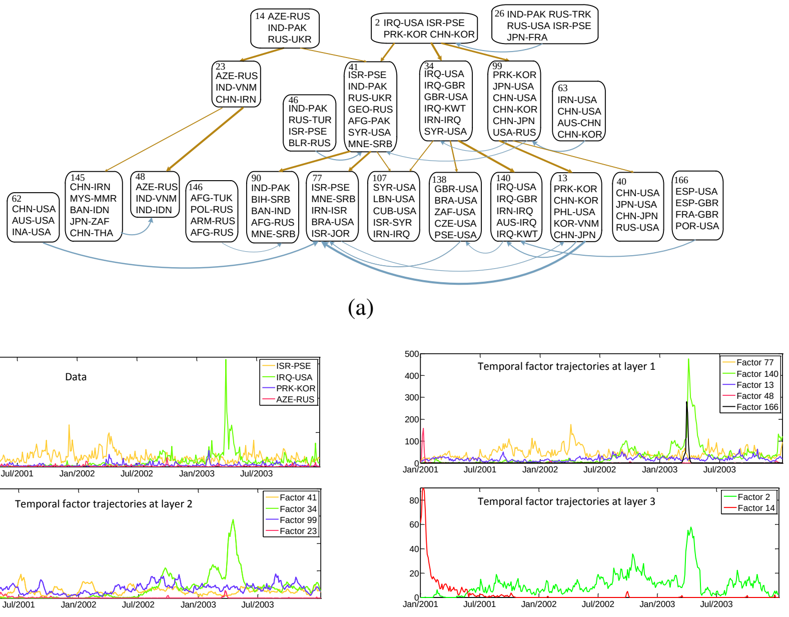
4.2.2 Exploratory data analysis

Compared to previously proposed dynamic systems, the proposed DPGDS, whose inferred latent structure is simple to visualize, provides much richer interpretation. More specifically, we may not only exhibit the content of each factor (topic), but also explore both the hierarchical relationships between them at different layers, and the temporal relationships between them at the same layer. Based on the results inferred on ICEWS 2001-2003 via a three hidden layer DPGDS, with the size of 200-100-50, we show in Fig. 5 how some example topics are hierarchically and temporally related to each other, and how their corresponding latent representations evolve over time.

In Fig. 5(a), we select two large-weighted topics at the top hidden layer and move down the network to include any lower-layer topics that are connected to them with sufficiently large weights. For each topic, we list all its terms whose values are larger than 1% of the largest element of the topic. It is interesting to note that topic 2 at layer three is connected to three topics at layer two, which are characterized mainly by the interactions of Israel (ISR)-Palestinian Territory (PSE), Iraq (IRQ)-USA-Iran (IRN), and North Korea (PRK)-South Korea (KOR)-USA-China (CHN)-Japan (JPN), respectively. The activation strength of one of these three interactions, known to be dominant in general during 2001-2003, can be contributed not only by a large activation of topic 2 at layer three, but also by a large activation of some other topic of the same layer (layer two) at the previous time. For example, topic 41 of layer two on "ISR-PSE, IND-PAK, RUS-UKR, GEO-RUS, AFG-PAK, SYR-USA, MNE-SRB" could be associated with the activation of topic 46 of layer two on "IND-PAK, RUS-TUR, ISR-PSE, BLR-RUS" at the previous time; and topic 99 of layer two on "PRK-KOR, JPN-USA, CHN-USA, CHN-KOR, CHN-JPN, USA-RUS" could be associated with the activation of topic 63 of layer two on "IRN-USA, CHN-USA, AUS-CHN, CHN-KOR" at the previous time.

Another instructive observation is that topic 140 of layer one on "IRQ-USA, IRQ-GBR, IRN-IRQ, IRQ-KWT, AUS-IRQ" is related not only in hierarchy to topic 34 of the higher layer on "IRQ-USA, IRQ-GBR, GBR-USA, IRQ-KWT, IRN-IRQ, SYR-USA," but also in time to topic 166 of the same layer on "ESP-USA, ESP-GBR, FRA-GBR, POR-USA," which are interactions between the member states of the North Atlantic Treaty Organization (NATO). Based on the transitions from topic 13 on "PRK-KOR" to both topic 140 on "IRQ-USA" and 77 on "ISR-PSE," we can find that the ongoing Iraq war and Israeli–Palestinian relations regain attention after the six-party talks [7].

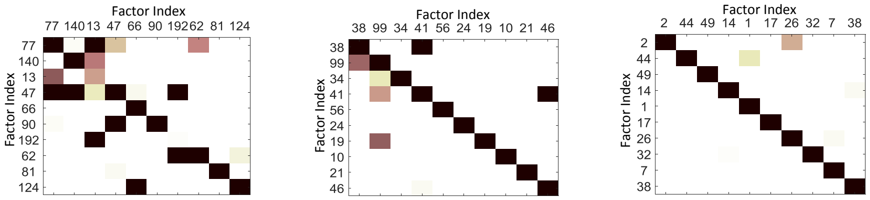
To get an insight of the benefits attributed to the deep structure, how the latent representations of several representative topics evolve over days are shown in Fig. 5(b). It is clear that relative to these temporal factor trajectories at the bottom layer, which are specific for the bilateral interactions between two countries, these from higher layers vary more smoothly, whose corresponding high-layer topics capture the multilateral interactions between multiple closely related countries. Similar phenomena have also been demonstrated in Fig. 1(b) on GDELT2003. Moreover, we find that a spike of the temporal trajectory of topic 166 (NATO) appears right before a one of topic 140 (Iraq war), matching the above description in Fig. 5(a). Also, topic 14 of layer three and its descendants, including topic 23 of layer two and topic 48 at layer one are mainly about a breakthrough between RUS and Azerbaijan (AZE), coinciding with Putin's visit in January 2001. Additional example results for the topics and their hierarchical and temporal relationships, inferred by DPGDS on different datasets, are provided in the Appendix.



(a)

(b)

Figure 5: Topics and their temporal trajectories inferred by a three-hidden-layer DPGDS from the ICEWS 2001-2003 dataset (best viewed in color). (a) Some example topics that are hierarchically or temporally related; (b) The temporal trajectories of some inferred latent topics.



(a)

(b)

(c)

Figure 6: Learned transition structure on ICEWS 2001-2003 from the same DPGDS depicted in Fig. 5. Shown in (a)-(c) are transition matrices for layers 1, 2 and 3, respectively, with a darker color indicating a larger transition weight (between 0 and 1).

In Fig. 6, we also present a subset of the transition matrix $\mathbf{\Pi}^{(l)}$ in each layer, corresponding to the top ten topics, some of which have been displayed in Fig. 5(b). The transition matrix $\mathbf{\Pi}^{(l)}$ captures the cross-topic temporal dependence at layer l. From Fig. 6, besides the temporal transitions between the topics at the same layer, we can also see that with the increase of the layer index l, the transition matrix $\mathbf{\Pi}^{(l)}$ more closely approaches a diagonal matrix, meaning that the feature factors become more likely to transit to themselves, which matches the characteristic of DPGDS that the topics in higher layers have the ability to cover longer-range temporal dependencies and contain more general information, as shown in Fig. 5(a). With both the hierarchical connections between layers and dynamic transitions at the same layer, distinct from the shallow PGDS, DPGDS is equipped with a larger capacity to model diverse temporal patterns with the help of its deep structure.

5 Conclusions

We propose deep Poisson gamma dynamical systems (DPGDS) that take the advantage of a probabilistic deep hierarchical structure to efficiently capture both across-layer and temporal dependencies. The inferred latent structure provides rich interpretation for both hierarchical and temporal information propagation. For Bayesian inference, we develop both a Backward-Upward–Forward-Downward Gibbs sampler and a stochastic gradient MCMC (SGMCMC) that is scalable to long multivariate count/binary time series. Experimental results on a variety of datasets show that DPGDS not only exhibits excellent predictive performance, but also provides highly interpretable latent structure.

Acknowledgements

D. Guo, B. Chen, and H. Zhang acknowledge the support of the Program for Young Thousand Talent by Chinese Central Government, the 111 Project (No. B18039), NSFC (61771361), NSFC for Distinguished Young Scholars (61525105) and the Innovation Fund of Xidian University. M. Zhou acknowledges the support of Award IIS-1812699 from the U.S. National Science Foundation.

References

[1] I. Sutskever and G. E. Hinton, "Learning multilevel distributed representations for high-dimensional sequences," in *AISTATS*, 2007.

[2] C. Wang, D. Blei, and D. Heckerman, "Continuous time dynamic topic models," in *UAI*, 2008, pp. 579–586.

[3] M. Hermans and B. Schrauwen, "Training and analysing deep recurrent neural networks," in *NIPS*, 2013, pp. 190–198.

[4] Z. Gan, C. Li, R. Henao, D. E. Carlson, and L. Carin, "Deep temporal sigmoid belief networks for sequence modeling," in *NIPS*, 2015, pp. 2467–2475.

[5] A. Acharya, J. Ghosh, and M. Zhou, "Nonparametric Bayesian factor analysis for dynamic count matrices," in *AISTATS*, 2015.

[6] L. Charlin, R. Ranganath, J. Mcinerney, and D. M. Blei, "Dynamic Poisson factorization," in *ACM*, 2015, pp. 155–162.

[7] A. Schein, M. Zhou, and H. Wallach, "Poisson–gamma dynamical systems," in *NIPS*, 2016.

[8] C. Y. Gong and W. Huang, "Deep dynamic Poisson factorization model," in *NIPS*, 2017.

[9] H. R. Rabiee, H. R. Rabiee, H. R. Rabiee, H. R. Rabiee, H. R. Rabiee, H. R. Rabiee, and H. R. Rabiee, "Recurrent Poisson factorization for temporal recommendation," in *KDD*, 2017, pp. 847–855.

[10] Z. Ghahramani and S. T. Roweis, "Learning nonlinear dynamical systems using an EM algorithm," in *NIPS*, 1999, pp. 431–437.

[11] J. M. Wang, A. Hertzmann, and D. M. Blei, "Gaussian process dynamical models," in *NIPS*, 2006.

[12] R. E. Kalman, "Mathematical description of linear dynamical systems," *Journal of The Society for Industrial and Applied Mathematics, Series A: Control*, vol. 1, no. 2, pp. 152–192, 1963.

[13] R. M. Neal, "Connectionist learning of belief networks," *Artificial Intelligence*, vol. 56, no. 1, pp. 71–113, 1992.

[14] R. Ranganath, L. Tang, L. Charlin, and D. M. Blei, "Deep exponential families," in *AISTATS*, 2014, pp. 762–771.

[15] M. Zhou, Y. Cong, and B. Chen, "The Poisson gamma belief network," in *NIPS*, 2015, pp. 3043–3051.

[16] R. Henao, Z. Gan, J. T. Lu, and L. Carin, "Deep Poisson factor modeling," in *NIPS*, 2015, pp. 2800–2808.

[17] Y. A. Ma, T. Chen, and E. B. Fox, "A complete recipe for stochastic gradient MCMC," in *NIPS*, 2015, pp. 2917–2925.

[18] M. Welling and Y. W. Teh, "Bayesian learning via stochastic gradient Langevin dynamics," in *ICML*, 2011, pp. 681–688.

[19] S. Patterson and Y. W. Teh, "Stochastic gradient Riemannian Langevin dynamics on the probability simplex," in *NIPS*, 2013, pp. 3102–3110.

[20] N. Ding, Y. Fang, R. Babbush, C. Chen, R. D. Skeel, and H. Neven, "Bayesian sampling using stochastic gradient thermostats," in *NIPS*, 2014, pp. 3203–3211.

[21] C. Li, C. Chen, D. Carlson, and L. Carin, "Preconditioned stochastic gradient Langevin dynamics for deep neural networks," in *AAAI*, 2016, pp. 1788–1794.

[22] M. Zhou, "Infinite edge partition models for overlapping community detection and link prediction," in *AISTATS*, 2015, pp. 1135–1143.

[23] M. Zhou, Y. Cong, and B. Chen, “Augmentable gamma belief networks,” *Journal of Machine Learning Research*, vol. 17, no. 163, pp. 1–44, 2016.

[24] M. Zhou and L. Carin, “Negative binomial process count and mixture modeling,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 2, pp. 307–320, 2015.

[25] R. M. Corless, G. H. Gonnet, D. E. G. Hare, D. J. Jeffrey, and D. E. Knuth, “On the LambertW function,” *Advances in Computational Mathematics*, vol. 5, no. 1, pp. 329–359, 1996.

[26] H. Zhang, B. Chen, D. Guo, and M. Zhou, “WHAI: Weibull hybrid autoencoding inference for deep topic modeling,” in *ICLR*, 2018.

[27] Y. Cong, B. Chen, H. Liu, and M. Zhou, “Deep latent Dirichlet allocation with topic-layer-adaptive stochastic gradient Riemannian MCMC,” in *ICML*, 2017.

[28] Y. Cong, B. Chen, and M. Zhou, “Fast simulation of hyperplane-truncated multivariate normal distributions,” *Bayesian Anal.*, vol. 12, no. 4, pp. 1017–1037, 2017.

[29] M. A. Girolami and B. Calderhead, “Riemann manifold Langevin and Hamiltonian Monte Carlo methods,” *Journal of The Royal Statistical Society Series B-statistical Methodology*, vol. 73, no. 2, pp. 123–214, 2011.

[30] N. D. Lawrence, “Probabilistic non-linear principal component analysis with gaussian process latent variable models,” *Journal of Machine Learning Research*, vol. 6, pp. 1783–1816, 2005.

[31] S. Han, L. Du, E. Salazar, and L. Carin, “Dynamic rank factor model for text streams,” in *NIPS*, 2014, pp. 2663–2671.

[32] P. Gopalan, F. J. R. Ruiz, R. Ranganath, and D. M. Blei, “Bayesian nonparametric Poisson factorization for recommendation systems,” in *AISTATS*, 2014.

[33] F. J. Anscombe, “The transformation of Poisson, binomial and negative-binomial data,” *Biometrika*, vol. 35, no. 3/4, pp. 246–254, 1948.

[34] D. B. Dunson and A. H. Herring, “Bayesian latent variable models for mixed discrete outcomes,” *Biostatistics*, vol. 6, no. 1, pp. 11–25, 2005.

[35] M. Zhou, L. Hannah, D. B. Dunson, and L. Carin, “Beta-negative binomial process and Poisson factor analysis,” in *AISTATS*, 2012, pp. 1462–1471.

[36] M. Zhou, “Nonparametric Bayesian negative binomial factor analysis,” *Bayesian Anal.*, vol. 13, no. 4, pp. 1061–1089, 2018.

Supplementary material for deep Poisson gamma dynamical systems

Dandan Guo, Bo Chen, Hao Zhang, and Mingyuan Zhou

A Details of inference via Gibbs sampling for DPGDS

Inference for the DPGDS shown in (1) is challenging, as neither the conjugate prior nor closed-form maximum likelihood estimate is known for the shape parameter of a gamma distribution. Although seemingly difficult, by generalizing the data augmentation and marginalization techniques, we are able to derive a backward-upward and then forward-downward Gibbs sampling algorithm, making it simple to draw random samples to represent the posteriors of model parameters. We marginalize over $\Theta^{(1:L)}$ by performing a "ackward" and "upward" filters, starting with $\boldsymbol{\theta}_T^{(1)}$. We repeatedly exploit the following three properties:

Property 1 (P1): if $y_{\cdot kt} = \sum_{n=1}^{N} y_n$, where $y_n \sim \mathrm{Pois}(\theta_n)$ are independent Poisson-distributed random variables, then $(y_1, ..., y_n) \sim \mathrm{Multi}\left(y_{\cdot}, \frac{\theta_1}{\sum_{n=1}^{N}\theta_n}, ..., \frac{\theta_N}{\sum_{n=1}^{N}\theta_n}\right)$ and $y_{\cdot} \sim \mathrm{Pois}(\sum_{n=1}^{N}\theta_n)$ [34, 35].

Property 2 (P2): $y \sim \mathrm{Pois}(c\theta)$, where c is a constant, and $\theta \sim \mathrm{Gam}\left(a, b\right)$ then $y \sim \mathrm{NB}\left(a, \frac{c}{c+b}\right)$ is a negative binomial–distributed random variable. We can equivalently parameterize it as $y \sim \mathrm{NB}\left(a, g\left(\zeta\right)\right)$, where $g\left(\zeta\right) = 1 - \exp\left(-\zeta\right)$ is the Bernoulli–Poisson link [22] and $\zeta = \ln\left(1 + \frac{c}{b}\right)$.

Property 3 (P3): if $y \sim \mathrm{NB}\left(a, g\left(\zeta\right)\right)$ and $l \sim \mathrm{CRT}\left(y, a\right)$ is a Chinese restaurant table-distributed random variable, then y and l are equivalently jointly distributed as $y \sim \mathrm{SumLog}\left(l, g\left(\zeta\right)\right)$ and $l \sim \mathrm{Pois}\left(a\zeta\right)$ [24].

A.1 Forward-downward sampling

Sampling transition matrix $\boldsymbol{\Pi}^{(l)}$: The alternative model specification, with Θ marginalized out, assumes that $\left(Z_{1kt}^{(l)}, ..., Z_{K_l,k,t}^{(l)}\right) \sim \mathrm{Multi}\left(x_{kt}^{(l+1,l)}, \left(\pi_{1k}^{(l)}, ..., \pi_{K_l k}^{(l)}\right)\right)$. Therefore, via the Dirichlet-multinomial conjugacy, we have

$$(\boldsymbol{\pi}_k^{(l)}|-) \sim \mathrm{Dir}(\nu_1^{(l)}\nu_k^{(l)} + Z_{1k\cdot}^{(l)}, ..., \nu_{K_l}^{(l)}\nu_k^{(l)} + Z_{K_l k\cdot}^{(l)}) \ . \tag{14}$$

Sampling loading factor matrix $\boldsymbol{\Phi}^{(l)}$: Given these latent counts, via the Dirichlet-multinomial conjugacy, we have

$$(\boldsymbol{\phi}_k^{(l)}|-) \sim \mathrm{Dir}(\eta^{(l)} + A_{1k\cdot}^{(l)}, ..., \eta^{(l)} + A_{K_{l-1}k\cdot}^{(l)}) \ . \tag{15}$$

Sampling $\delta_t^{(1)}$: Via the gamma-Poisson conjugacy, we have

$$(\delta_t^{(1)}|-) \sim \mathrm{Gam}\left(\varepsilon_0 + \sum_{v=1}^{V} x_{vt}^{(1)}, \varepsilon_0 + \sum_{k=1}^{K_1}\theta_{kt}^{(1)}\right), \text{ if } \delta_t^{(1)} \neq \delta_{t'}^{(1)} \text{ for } t \neq t'; \tag{16}$$

$$(\delta^{(1)}|-) \sim \mathrm{Gam}\left(\varepsilon_0 + \sum_{t=1}^{T}\sum_{v=1}^{V} x_{vt}^{(1)}, \varepsilon_0 + \sum_{t=1}^{T}\sum_{k=1}^{K_1}\theta_{kt}^{(1)}\right), \text{ if } \delta_t^{(1)} = \delta^{(1)} \text{ for all } t. \tag{17}$$

Sampling $\beta^{(l)}$:

$$(\beta^{(l)}|-) \sim \mathrm{Gam}\left(\varepsilon_0 + \gamma_0, \varepsilon_0 + \sum_{k=1}^{K_l}\nu_k^{(l)}\right) \tag{18}$$

Sampling $v_k^{(l)}$ and $\xi^{(l)}$:

$$(Z_{k1t}^{(l)}, ..., Z_{kK_lt}^{(l)}\,|-) \sim \mathrm{Multi}\left(x_{kt}^{(l+1,l)}; \frac{\pi_{k1}^{(l)}\theta_{1,t-1}^{(l)}}{\sum_{k_l=1}^{K_l}\pi_{kk_l}^{(l)}\theta_{k_l,t-1}^{(l)}}, ..., \frac{\pi_{kK_l}^{(l)}\theta_{K_l,t-1}^{(l)}}{\sum_{k_l=1}^{K_l}\pi_{kk_l}^{(l)}\theta_{k_l,t-1}^{(l)}}\right), \tag{19}$$

To obtain closed-form conditional posterior for $v_k^{(l)}$ and $\xi^{(l)}$, we start with

$$(Z^{(l)}_{1k\cdot}, \cdots, Z^{(l)}_{kk\cdot}, \cdots, Z^{(l)}_{K_1 k\cdot}) \sim \text{DirMult}(Z^{(l)}_{\cdot k\cdot}, (v^{(l)}_1 v^{(l)}_k, \cdots, \xi^{(l)} v^{(l)}_k, \cdots, v^{(l)}_K v^{(l)}_k)), \tag{20}$$

where $Z^{(l)}_{k_1 k\cdot} = \sum_{t=1}^{T} Z^{(l)}_{k_1 k t}$ and $Z^{(l)}_{\cdot k\cdot} = \sum_{t=1}^{T} \sum_{k_1=1}^{K_l} Z^{(l)}_{k_1 k t}$. Following Zhou [36], we draw a beta-distributed auxiliary variable:

$$(q_k^{(l)} \,|\, -) \sim \text{Beta}(Z^{(l)}_{\cdot k\cdot}, \nu_k^{(l)}(\xi^{(l)} + \textstyle\sum_{k_1 \neq k} \nu^{(l)}_{k_1})). \tag{21}$$

Consequently, we have

$$P(Z^{(l)}_{kk\cdot}, q_k^{(l)}) \propto \text{NB}(Z^{(l)}_{kk\cdot}; \xi^{(l)} \nu_k^{(l)}, q_k^{(l)}) \quad \text{and} \quad P(Z^{(l)}_{k_1 k\cdot}, q_k^{(l)}) \propto \text{NB}(Z^{(l)}_{k_1 k\cdot}; v^{(l)}_{k_1} \nu_k^{(l)}, q_k^{(l)}) \tag{22}$$

for $k_1 \neq k$. Next, we introduce the following auxiliary variables:

$$(h^{(l)}_{kk} \,|\, -) \sim \text{CRT}(Z^{(l)}_{kk\cdot}, \xi^{(l)} \nu_k^{(l)}) \quad \text{and} \quad (h^{(l)}_{k_1 k} \,|\, -) \sim \text{CRT}(Z^{(l)}_{k_1 k\cdot}, \nu^{(l)}_{k_1} \nu^{(l)}_k) \tag{23}$$

for $k_1 \neq k$. We can then re-express the joint distribution over the variable in (22) and (23) as

$$Z^{(l)}_{kk\cdot} \sim \text{SumLog}(h^{(l)}_{kk}, q_k^{(l)}) \quad \text{and} \quad Z^{(l)}_{k_1 k\cdot} \sim \text{SumLog}(h^{(l)}_{k_1 k}, q_k^{(l)}) \tag{24}$$

and

$$h^{(l)}_{kk} \sim \text{Pois}(-\xi^{(l)} \nu^{(l)}_k \ln(1 - q^{(l)}_k)) \quad \text{and} \quad h_{k_1 k} \sim \text{Pois}(-\nu^{(l)}_{k_1} \nu^{(l)}_k \ln(1 - q^{(l)}_k)). \tag{25}$$

Then, via the gamma-Poisson conjugacy, we have

$$(\xi^{(l)} | -) \sim \text{Gam}\left(\frac{\gamma_0}{K_l} + \sum_{k=1}^{K_l} h^{(l)}_{kk},\ \beta^{(l)} - \sum_{k=1}^{K_l} \nu^{(l)}_k \ln(1 - q^{(l)}_k)\right). \tag{26}$$

Note that when $l = L$ and $t = 1$, we have $\boldsymbol{\theta}_1^{(L)} \sim \text{Gam}\left(\tau_0 \nu_k^{(L)}, \tau_0\right)$ and $m^{(L)}_{k1} \sim \text{Pois}\left(\tau_0(\zeta_2^{(L)} + \zeta_1^{(L-1)}) \theta^{(L)}_{k1}\right)$, where $m^{(1)}_{k1} = A^{(1)}_{\cdot k1} + Z^{(1)}_{\cdot k2}$. So we can sample $(x^{(L+1)}_{k1} \,|\, -) \sim \text{CRT}(m^{(l)}_{k1}, \tau_0 \nu_k^{(l)})$. Via **P3**, We can further get $x^{(L+1)}_{k1} \sim \text{Pois}(\zeta_1^{(L)} \tau_0 \nu_k^{(L)})$.

Next, because $x_k^{(L+1)}$ also depends on $\nu_k^{(L)}$, we introduce

$$n_k^{(l)} = h^{(l)}_{kk} + \sum_{k_1 \neq k} h^{(l)}_{k_1 k} + \sum_{k_2 \neq k} h^{(l)}_{k k_2} \tag{27}$$

for $l = 1, \ldots, L-1$ and

$$n_k^{(L)} = h^{(L)}_{kk} + \sum_{k_1 \neq k} h^{(L)}_{k_1 k} + \sum_{k_2 \neq k} h^{(L)}_{k k_2} + x^{(L+1)}_{k1}. \tag{28}$$

Then, via **P1**, we have

$$n_k^{(l)} \sim \text{Pois}(\nu_k^{(l)} \rho_k^{(l)}), \tag{29}$$

where

$$\rho_k^{(l)} = -\ln(1 - q_k^{(l)})(\xi^{(l)} + \sum_{k_1 \neq k} \nu^{(l)}_{k_1}) - \sum_{k_2 \neq k} \ln(1 - q^{(l)}_{k_2}) \nu^{(l)}_{k_2} \tag{30}$$

for $l = 1, \ldots, L-1$ and

$$\rho_k^{(L)} = -\ln(1 - q_k^{(L)})(\xi^{(L)} + \sum_{k_1 \neq k} \nu^{(L)}_{k_1}) - \sum_{k_2 \neq k} \ln(1 - q^{(L)}_{k_2}) \nu^{(L)}_{k_2} + \zeta^{(L)} \tau_0. \tag{31}$$

Finally, via the gamma-Poisson conjugacy, we have

$$(\nu_k^{(l)} | -) \sim \text{Gam}\left(\frac{\gamma_0}{\beta^{(l)}} + n_k^{(l)}, \beta^{(l)} + \rho_k^{(l)}\right). \tag{32}$$

Algorithm 1 Backward-Upward–Forward-Downward Gibbs sampling for DPGDS

```
for iter = 1 : $B_L$ + $C_L$ do Gibbs sampling do
    \ ⋆ Collect local information
    Backward-upward Gibbs sampling for $\{A_{vkt}^{(l)}\}_{v,k,t}$; $\{x_{kt}^{(l+1)}\}_{k,t}$; $\{x_{kt}^{(l+1,l)}\}_{k,t}$ ; $\{x_{kt}^{(l+1,l+1)}\}_{k,t}$;
    $\{Z_{k_1k_2t}^{(l)}\}_{k_1,k_2,t}$ with (8)-(11);
    Backward-upward calculating for $\{\zeta_t^{(l)}\}_t$;
    Forward-downward Gibbs sampling for $\{\boldsymbol{\theta}_t^{(l)}\}_t$ with (12);
    Sampling $\boldsymbol{\delta}^{(1)}$ with (16) or (17);
    \ ⋆ Update global parameters
    for $l = 1, 2, \cdots, L \quad and \quad k = 1, 2, \cdots, K_L$ do
        Update $\{\boldsymbol{\pi}_k^{(l)}\}_k$ from (14); Update $\{\boldsymbol{\phi}_k^{(l)}\}_k$ from (15); Update $\beta^{(l)}, \xi^{(l)}, \{\nu_k^{(l)}\}_k$ according to
        (26), (18), and (32);
    end for
end for
```

A.2 SGMCMC for DPGDS

Although the Gibbs sampling algorithm for DPGDS has closed-form update equations discussed above, it requires handling all time-varying vectors in each iteration and hence has limited scalability [26]. To allow for tractable and scalable inference, in Section 3.3, we propose a SGMCMC method to infer the DGPDS using TLASGR-MCMC [27] to update $\{\mathbf{\Pi}^{(l)}\}_{l=1}^L$. In this section, we discuss how to update the other global parameters in detail, as described in Algorithm in 2.

Sample the transmission matrix $\{\mathbf{\Pi}^{(l)}\}_{l=1}^L$:

$$\left(\boldsymbol{\pi}_k^{(l)}\right)_{n+1} = \Bigg[\left(\boldsymbol{\pi}_k^{(l)}\right)_n + \frac{\varepsilon_n}{M_k^{(l)}}\Big[\left(\rho\tilde{\boldsymbol{z}}_{:k\cdot}^{(l)} + \boldsymbol{\eta}_{:k}^{(l)}\right) - \left(\rho\tilde{z}_{\cdot k\cdot}^{(l)} + \eta_{\cdot k}^{(l)}\right)\left(\boldsymbol{\pi}_k^{(l)}\right)_n\Big] + \mathcal{N}\left(0, \frac{2\varepsilon_n}{M_k^{(l)}}\Big[\text{diag}(\boldsymbol{\pi}_k^{(l)})_n - (\boldsymbol{\pi}_k^{(l)})_n(\boldsymbol{\pi}_k^{(l)})_n^T\Big]\right)\Bigg]_\angle . \tag{33}$$

Sample the hierarchical topics $\{\mathbf{\Phi}^{(l)}\}_{l=1}^L$: In DPGDS, the prior and likelihood of $\{\mathbf{\Phi}^{(l)}\}_{l=1}^L$ resemble those for $\{\mathbf{\Pi}^{(l)}\}_{l=1}^L$, so we also apply the TLASGR MCMC sampling algorithm on it as

$$\left(\boldsymbol{\phi}_k^{(l)}\right)_{n+1} = \Bigg[\left(\boldsymbol{\phi}_k^{(l)}\right)_n + \frac{\varepsilon_n}{P_k^{(l)}}\Big[\left(\rho\tilde{\boldsymbol{A}}_{:k\cdot}^{(l)} + \eta_0^{(l)}\right) - \left(\rho\tilde{A}_{\cdot k\cdot}^{(l)} + K_{l-1}\eta_0^{(l)}\right)\left(\boldsymbol{\phi}_k^{(l)}\right)_n\Big] + \mathcal{N}\left(0, \frac{2\varepsilon_n}{P_k^{(l)}}\Big[\text{diag}(\boldsymbol{\phi}_k^{(l)})_n - (\boldsymbol{\phi}_k^{(l)})_n(\boldsymbol{\phi}_k^{(l)})_n^T\Big]\right)\Bigg]_\angle , \tag{34}$$

where $M_k^{(l)}$ and $P_k^{(l)}$ are calculated using the estimated FIM, $\tilde{\boldsymbol{z}}_{:k\cdot}$, $\tilde{z}_{\cdot k\cdot}^{(l)}$, $\tilde{\boldsymbol{A}}_{:k\cdot}^{(l)}$, and $\tilde{A}_{\cdot k\cdot}^{(l)}$ come from the augmented latent counts $\mathbf{Z}^{(l)}$ and $\mathbf{A}^{(l)}$, $\boldsymbol{\eta}_{:k}^{(l)}$ and $\eta_0^{(l)}$ denote the prior of $\boldsymbol{\pi}_k^{(l)}$ and $\boldsymbol{\phi}_k^{(l)}$, and $[\cdot]_\angle$ denotes a simplex constraint; more details about TLASGR-MCMC for DLDA can be found in Cong et al. [27].

For other global variables, Λ_g, containing $\{\xi^{(l)}\}_{l=1}^L$ and $\{v_k^{(l)}\}_{l=1,k=1}^{L,K_l}$ (the hyper-parameter $\{\beta^{(l)}\}_{l=1}^L$ is set to 1 here), we find that it is enough to use a first-order SGMCMC method to sample them. Considering the efficiency and the performance, we use the stochastic gradient Nose-Hoover thermostat (SGNHT) to update all these variables, which has the potential advantage of making the system jump out of local models easier and reach the equilibrium state faster. Specifically, the dynamic system are defined by the following stochastic differential equations:

$$d\Lambda_g = \boldsymbol{p}dt, d\boldsymbol{p} = \boldsymbol{f}(\Lambda_g) - \tau\boldsymbol{p}dt + \sqrt{2A}\mathcal{N}(0, dt) \tag{35}$$

$$d\tau = (\frac{1}{n}\boldsymbol{p}^T\boldsymbol{p} - 1)dt \tag{36}$$

where $\boldsymbol{p}$ simulate the momenta in a system and τ is called the thermostat variable which ensures the system temperature to be constant. The stochastic force $\boldsymbol{f}(\Lambda_g) = -\nabla_{\Lambda_g} U(\Lambda_g)$, where $U(\Lambda_g)$ is the negative log-posterior of a Bayesian model, is calculated on a mini-batch subset of data or the other global parameters. Note that given the appropriate initial values of $\Lambda_g, \tau, \boldsymbol{p}, A$, it is only need to calculate the $\boldsymbol{f}(\Lambda_g)$ to update the Λ_g, which will be given.

Calculate the stochastic force of $v_k^{(l)}$:

$$U\left(\nu_k^{(l)}\right) = -\sum_{k=1}^{K_l} \log p\left(\boldsymbol{\pi}_k^{(l)} | \zeta^{(l)}, \nu_k^{(l)}\right) - \log p\left(\nu_k^{(l)} | \frac{\gamma_0}{K_l}, \beta^{(l)}\right), \tag{37}$$

$$\begin{aligned}\nabla_{\nu_k^{(l)}} U\left(\nu_k^{(l)}\right) = &-\left[\sum_{k_1=1}^{K_l}\left(\nu_{k_1}^{(l)}\right)\log\left(\pi_{k_1 k}^{(l)}\right) + \sum_{k_2=1}^{K_l}\left(\nu_{k_2}^{(l)}\right)\log\left(\pi_{k k_2}^{(l)}\right) + \left(\zeta^{(l)} - 4\nu_k^{(l)}\right)\log \pi_{kk}^{(l)}\right] \\ &- \frac{\left(\frac{\gamma_0}{K_l} - 1\right)}{\nu_k^{(l)}} + \beta^{(l)}.\end{aligned} \tag{38}$$

Calculate the stochastic force of $\xi^{(l)}$:

$$U\left(\xi^{(l)}\right) = -\sum_{k=1}^{K_l} \log p\left(\pi_k^{(l)} | \xi^{(l)}\right) - \log p\left(\xi^{(l)} | \varepsilon_0, \varepsilon_0\right), \tag{39}$$

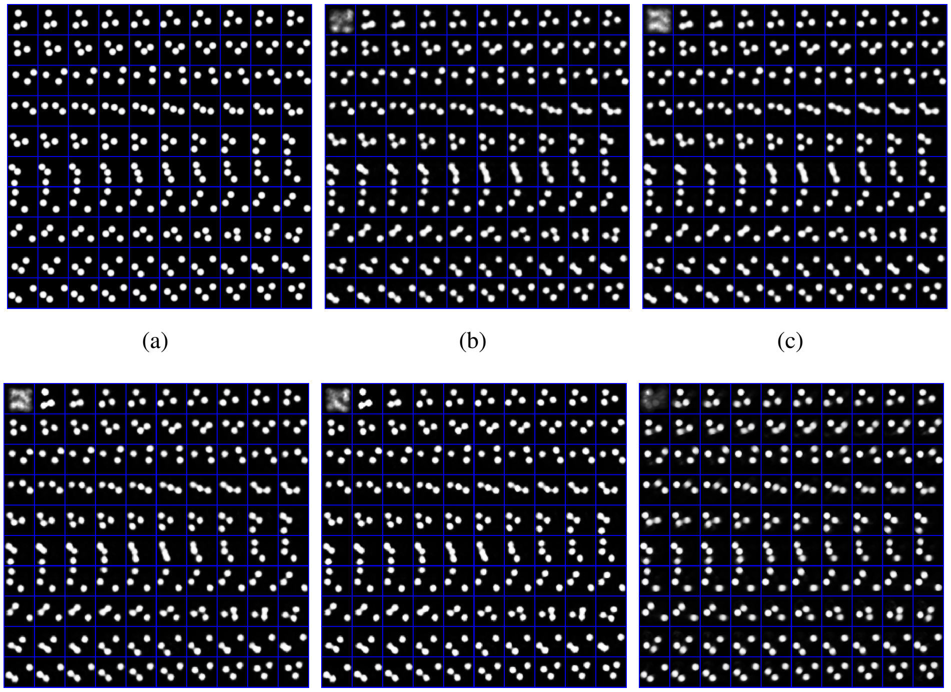
$$\nabla_{\xi^{(l)}} U\left(\xi^{(l)}\right) = -\sum_{k=1}^{K_l} \nu_k^{(l)} \log\left(\pi_{kk}^{(l)}\right) - \frac{(\varepsilon_0 - 1)}{\xi^{(l)}} + \varepsilon_0. \tag{40}$$

Algorithm 2 Stochastic-gradient MCMC for DPGDS

```
Input: Data mini-batches; Output: Global parameters of DPGDS.
  for $i = 1, 2, \cdots$ do
    \⋆ Collect local information
    Backward-upward Gibbs sampling on the $i$th mini-batch for $\{A_{vkt}^{(l)}\}_{v,k,t}$; $\{x_{kt}^{(l+1)}\}_{k,t}$; $\{x_{kt}^{(l+1,l)}\}_{k,t}$ ; $\{x_{kt}^{(l+1,l+1)}\}_{k,t}$; $\{Z_{k_1 k_2 t}^{(l)}\}_{k_1,k_2,t}$ with (8)-(11);
    Backward-upward calculating for $\{\zeta_t^{(l)}\}_t$;
    Forward-downward Gibbs sampling for $\{\boldsymbol{\theta}_t^{(l)}\}_t$ with (12);
    Sampling $\boldsymbol{\delta}^{(1)}$ with (16) or (17);
    \⋆ Update global parameters
    for $l = 1, 2, \cdots, L \quad and \quad k = 1, 2, \cdots, K_L$ do
      Update $M_k^{(l)}$ according to Cong et al. [27], and then $\{\boldsymbol{\phi}_k^{(l)}\}_k$ with (34); Update $M_k^{(l)}$ according to [27], and then $\{\boldsymbol{\pi}_k^{(l)}\}_k$ with (33);
    end for
    Update $\xi^{(l)}$, $\{\nu_k^{(l)}\}_k$, and $\beta^{(l)}$ with SGNHT [20]
  end for
```

B Results on Bouncing ball

In Fig. 7, we show the original data and the one-step prediction frames of five different algorithms. The frames in each subplot is arranged by time from left to right and top to bottom. We find that the most difficult prediction is the frames that describe how the balls move after the collision, such as observing the fourth row and ninth row. We find that comparing with the original data, a good model means that two balls can be separated soon after the collision, while a bad model means that two balls have unreasonable trajectories. According to this action mechanism, we can see that DPGDS outperforms the others.



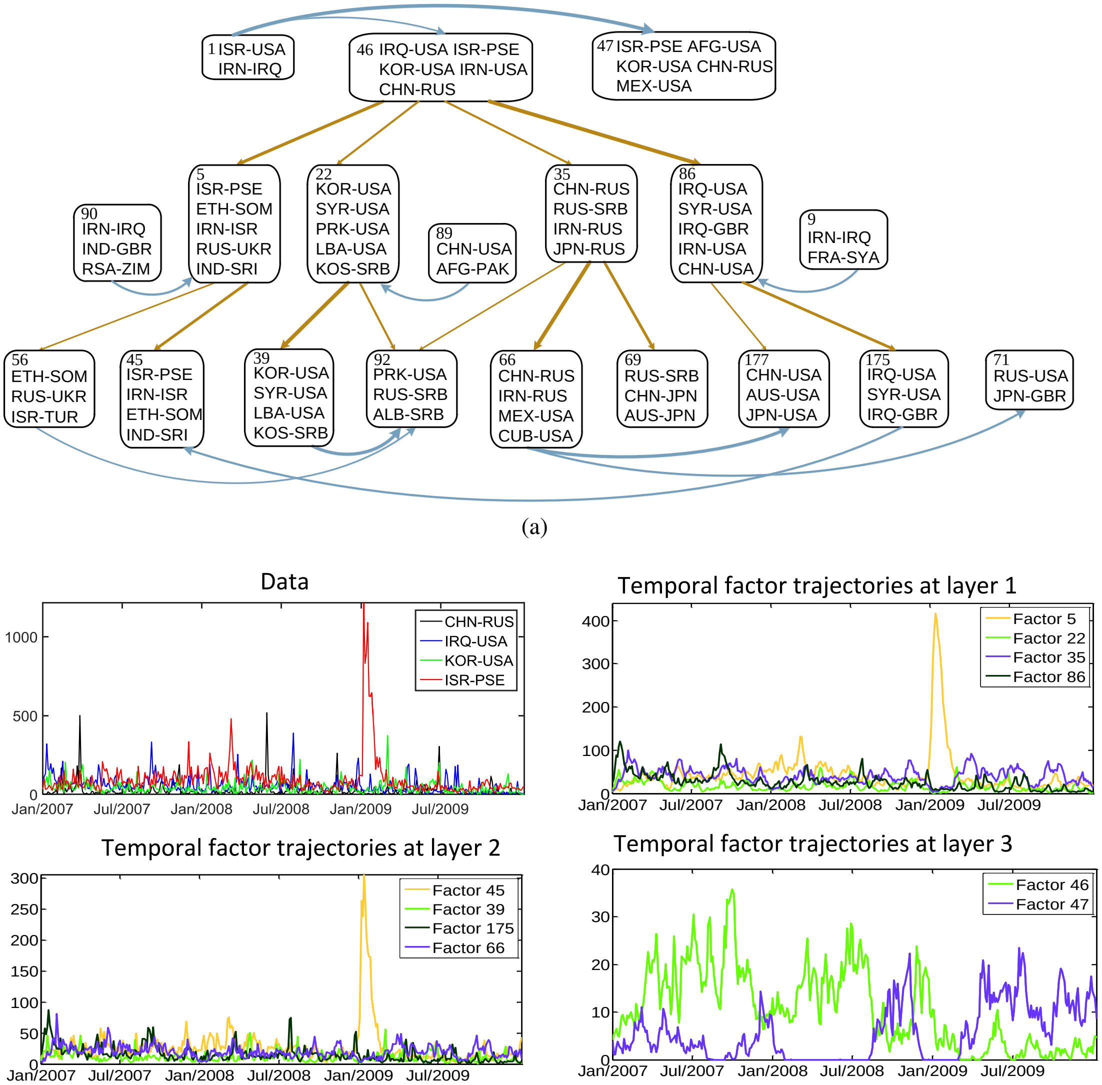
(a) (b) (c)

(d) (e) (f)

Figure 7: (a) The original data and the one-step prediction results on the bouncing ball date set by (b) TSBN, (c) PGDS, (d) TSBN-4, (e) DTSBN, (f) DPGDS, and .

C Results on ICEWS 2007-2009

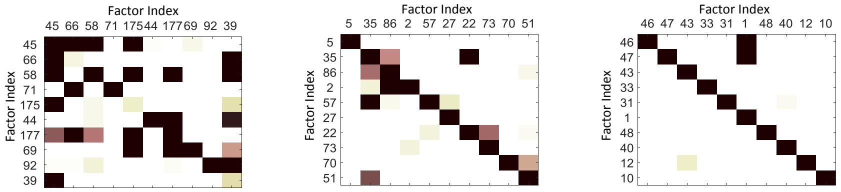
In order to understand the DPGDS better, based on the results inferred on ICEWS 2007-2009 via a three-hidden-layer DPGDS, with the size of 200-100-50, we show in Fig. 8 how some example topics are hierarchically and temporally related to each other, and how their corresponding latent representations evolve over time. Similar findings and conclusions can be reached according to Fig. 8 like ICEWS 2001-2003 in Figs. 5 and 6. In Fig. 9, we also present a subset of the transition matrix $\mathbf{\Pi}^{(l)}$ in each layer, corresponding to the top ten topics, some of which have been displayed in Fig. 8.



(a)

(b)

Figure 8: Topics and their temporal trajectories inferred by a three-layer DPGDS from the ICEWS 2007-2009 dataset. (a) Some example topics that are hierarchically or temporally related; (b) The temporal trajectories of some inferred latent topics.

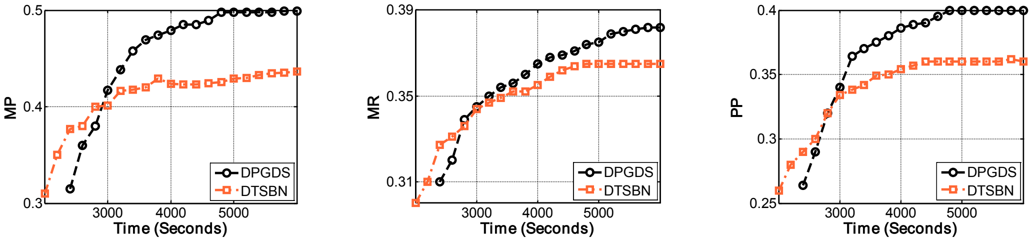


(a) (b) (c)

Figure 9: Learned transition structure on ICEWS 2007-2009 from the same DPGDS depicted in Fig. 8. Shown in (a)-(c) are transition matrices for layers 1, 2 and 3, respectively, with a darker color indicating a larger transition weight (between 0 and 1).

D Results on GDELT 2015-2018

To add more empirical study on scalability, we have collected GDELT data from February 2015 to July 2018 (temporal granularity of 15 mins), resulting in a count matrix with $V = 1000$ and $T \approx 120,000$. For such a long time series, Backward-Upward–Forward-Downward Gibbs sampling for DPGDS is impractical to run as a single iteration takes nearly 3000 seconds. GPDM is trained with a batch algorithm, which is also too time consuming to run for this dataset. However, by taking short sequences at random locations from the data, we can run both DTSBN [4] and the proposed DPGDS using SGMCMC. Here, we use $[K^{(1)}, K^{(2)}, K^{(3)}] = [200, 100, 50]$ for both DPGDS and DTSBN and choose the length of each short sequence to be $T = 60$. As shown in Fig. 10, we present how DTSBN and the proposed DPGDS progress over time, evaluated with MP, MR and PP. It takes about 6000s for DTSBN and DPGDS-SGMCMC to converge. Clearly, our DPGDS-SGMCMC is scalable and clearly outperforms DTSBN.



(a) (b) (c)

Figure 10: Shown in (a)-(c) are MP, MR, PP, respectively, as the function of time for GDELT 2015 -2018.