Hierarchical Deep Generative Models for Multi-Rate Multivariate Time Series

Zhengping Che [*1] **Sanjay Purushotham** [*1] **Guangyu Li** [*1] **Bo Jiang** [1] **Yan Liu** [1]

Abstract

Multi-Rate Multivariate Time Series (MR-MTS) are the multivariate time series observations which come with various sampling rates and encode multiple temporal dependencies. State-space models such as Kalman filters and deep learning models such as deep Markov models are mainly designed for time series data with the same sampling rate and cannot capture all the dependencies present in the MR-MTS data. To address this challenge, we propose the Multi-Rate Hierarchical Deep Markov Model (MR-HDMM), a novel deep generative model which uses the latent hierarchical structure with a learnable switch mechanism to capture the temporal dependencies of MR-MTS. Experimental results on two real-world datasets demonstrate that our MR-HDMM model outperforms the existing state-of-the-art deep learning and state-space models on forecasting and interpolation tasks. In addition, the latent hierarchies in our model provide a way to show and interpret the multiple temporal dependencies.

1. Introduction

Multivariate time series (MTS) analysis (Hamilton, 1994; Reinsel, 2003) has attracted a lot of attention in machine learning, signal processing, and other related areas, due to its impact and usefulness in many real world applications such as healthcare, climate, and financial forecasting. State-space models such as Kalman filters (Kalman et al., 1960) and hidden Markov models (Rabiner, 1989) have been developed to model MTS and have shown promising results on prediction tasks such as forecasting and interpolation. However, in many applications, the MTS observations usually come from multiple sources and are often characterized by various sampling rates. For example, in healthcare, vital signs such as heart rate are sampled frequently, while lab results such as pH are measured infrequently; in finance, the stock prices are sampled daily or even more frequently, while macro-economic data such as employment, GDP are sampled monthly or quarterly. Such time series observations with either regular or irregular sampling rates are termed as *Multi-Rate Multivariate Time Series (MR-MTS)* data. Modeling the MR-MTS using state-space models is challenging since MR-MTS naturally comes with multiple temporal dependencies and these dependencies may not have direct relationship to the sampling rates. That is, the long and short-term temporal dependencies may be associated with a few or all the time series data with different sampling rates. Capturing these temporal dependencies is important as they model the underlying data generation mechanism, and they impact the interpolation and forecasting tasks. Upsampling or downsampling MR-MTS to a single rate time series cannot address this challenge, since these simple techniques may artifically introduce or remove some naturally occurring dependencies present in MR-MTS. For example, forward/backward imputation will introduce long-term dependencies. Therefore, building models which can capture multiple temporal dependencies directly from the MR-MTS data is still an open problem in the time series analysis field.

Deep learning models such as recurrent neural networks (RNNs) (Hochreiter & Schmidhuber, 1997) have emerged as successful models for time series analysis (Graves et al., 2013; Mikolov et al., 2010) and sequence modeling applications (Socher et al., 2011; Xu et al., 2015). While deep discriminative models (Hermans & Schrauwen, 2013; Martens & Sutskever, 2011; Pascanu et al., 2013; Chung et al., 2016) have been shown to model complex non-linear temporal dependencies present in MTS, deep generative models (Gan et al., 2015; Rezende et al., 2014) have become more popular since they are intuitive, interpretable and are more powerful than their discriminative counterparts (Durbin & Koopman, 2012) and they capture the data generation process. Despite their success with single-rate time series data, the existing deep generative models are not suitable for modeling MR-MTS as they are not designed to capture multiple temporal dependencies from different sampling rates.

Recently, latent hierarchical structure learning based on deep learning models have led to remarkable ad-

[*]Equal contribution [1]Department of Computer Science, University of Southern California, Los Angeles, California, United States. Correspondence to: Zhengping Che, Sanjay Purushotham, Guangyu Li, Bo Jiang, Yan Liu <{zche,spurusho,guangyul,boj,yanliu.cs}@usc.edu>.

Proceedings of the 35^{th} International Conference on Machine Learning, Stockholm, Sweden, PMLR 80, 2018. Copyright 2018 by the author(s).

vances in capturing temporal dependencies from sequential data (El Hihi & Bengio, 1995; Chung et al., 2016; Koutnik et al., 2014). Motivated by these models, we propose a novel deep generative model termed as **M**ulti-**R**ate **H**ierarchical **D**eep **M**arkov **M**odel **(MR-HDMM)**, which learns multiple temporal dependencies directly from MR-MTS by jointly modeling time series with different sampling rates. MR-HDMM learns the latent hierarchical structures along with learnable switches and captures the data generation process of MR-MTS. It simultaneously learns a inference network and a generative model by leveraging a structured variational approximation parameterized by recurrent neural networks to mimic the posterior distribution. The data generation process of MR-HDMM can automatically infer the hierarchical structures directly from data, which is extremely helpful for downstream tasks such as interpolation and forecasting.

In summary, we develop a first-of-a-kind novel deep generative model called MR-HDMM to systematically capture the multiple temporal dependencies present in MR-MTS by using hierarchical latent structures and learnable switches. In addition, we also propose a new structured inference network for MR-HDMM. A comprehensive and systematic evaluation of the MR-HDMM model is conducted on two real-world datasets to demonstrate the state-of-the-art performance in forecasting and interpolation tasks. Finally, we interpret the learnt latent hierarchies from MR-HDMM to study the captured temporal dependencies.

2. Related Work

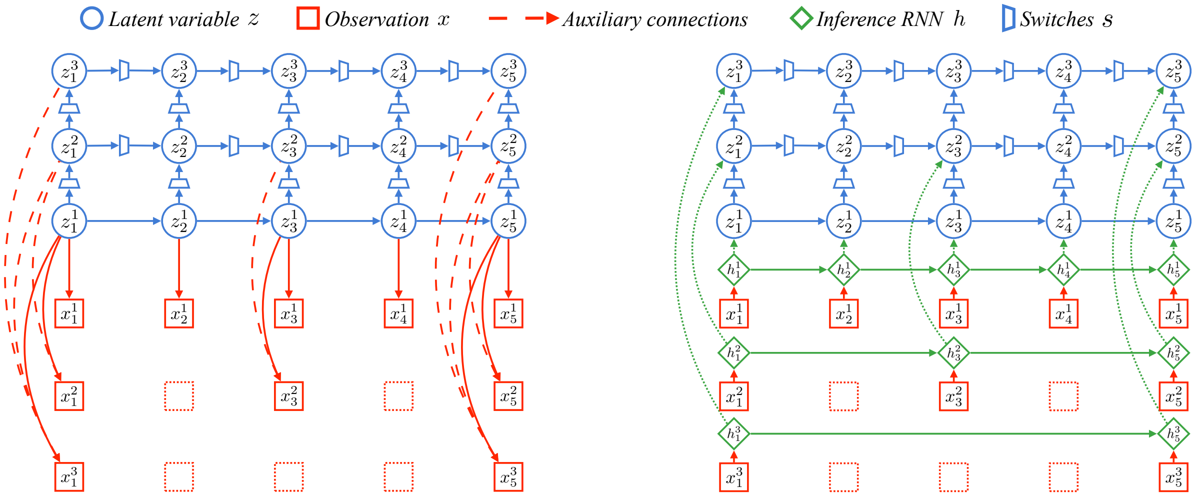
State-space models such as Kalman filters (KF) (Kalman et al., 1960), and hidden Markov models (HMMs) (Rabiner, 1989) have been widely used in various time series applications such as speech recognition (Rabiner, 1989), atmospheric monitoring (Houtekamer & Mitchell, 2001), and robotic control (Negenborn, 2003). These approaches successfully model regularly sampled (i.e. sampled at the same frequency/rate) time series data, however, they cannot be directly used for MR-MTS as they cannot simultaneously capture the multiple temporal dependencies present in MR-MTS. To handle MR-MTS with state-space models, researchers have extended KF models and proposed multi-rate Kalman filters (MR-KF) (Armesto et al., 2008; Safari et al., 2014). MR-KF approaches either fuse the data with different sampling rates or fuse the estimates for KFs trained on each sampling rate. Many of these MR-KF approaches aim to improve the estimates for the highest sampled rate data and do not focus on capturing the multiple temporal dependencies present in MR-MTS. Moreover, the linear transition and emission functionality of the MR-KF models limits their usability on complex real-world data.

Recently, researchers have resorted to deep learning models (Chung et al., 2016; Krishnan et al., 2015; Gan et al., 2015) to model the non-linear temporal dynamics of real-world and sequential data. Discriminative models such as hierarchical recurrent neural network (El Hihi & Bengio, 1995), hierarchical multiscale recurrent neural network (HM-RNN) (Chung et al., 2016), and phased long short-term memory (PLSTM) (Neil et al., 2016) have been proposed to capture temporal dependencies of sequential data. However, these discriminative models do not capture the underlying data generation process and therefore are not suited for forecasting and interpolation tasks. Deep generative models (Rezende et al., 2014; Krishnan et al., 2015; Gan et al., 2015) have been developed to model the data generation process of the complex time series data. Krishnan et al. (2015) proposed deep Kalman filter, a nonlinear state-space model, by marrying the ideas of deep neural networks with Kalman filters. Fraccaro et al. (2016) introduced stochastic recurrent neural network (SRNN) which glued a RNN with a state space model together to form a stochastic and sequential neural generative model. Even though these deep generative models are the state-of-the-art approaches to obtain the underlying data generation process, they are not designed to capture all the temporal dependencies of MR-MTS. None of the existing deep learning models or state-space models can be directly used for modeling MR-MTS. Thus, in this work, we develop a deep generative model which leverages the properties of the above discriminative and generative models, to model the data generation process of MR-MTS while also capturing the multiple temporal dependencies using a latent hierarchical structure.

3. Our Model

In this section, we present our proposed Multi Rate-Hierarchical Deep Markov Model (MR-HDMM). We first clarify the notations and definitions used in this paper.

Notations Given a MR-MTS of L different sampling rates and length T, we use a vector $\boldsymbol{x}_t^l \in \mathbb{R}^{D_l}$ to represent the time series observations of lth rate at time t. Here $l = 1, \ldots, L$, $t = 1, \ldots, T$, and D_l is the dimension of time series with lth rate. The L sampling rates are in descending order, i.e., $l = 1$ and $l = L$ refer to the highest and lowest sampling rates. To make the notations succinct, we use $\boldsymbol{x}_{t:t'}^{l:l'}$ to denote all observed time series of lth to l'th rates and from time t to t'. We use $\theta_{(.)}$ and $\phi_{(.)}$ to denote the parameter sets for generation model p_θ and inference network q_ϕ respectively. we use L layers of RNNs in the inference network to model MR-MTS of L different sampling rates. We use L_{HS}, the number of hidden layers in both generation model and inference network, to control the depth of the learnt hierarchical structures. In the rest of this paper we take $L_{HS} = L$ for model simplicity, but in practice they are not tied. The latent states or variables are denoted by $\boldsymbol{z}$, $\boldsymbol{s}$ and $\boldsymbol{h}$. Their superscript and subscript respectively indicate the corresponding layer(s) and the time step(s) (e.g., $\boldsymbol{z}_{1:T}^{1:L}$, $\boldsymbol{s}_t^{2:L}$, $\boldsymbol{h}_t^l$).


Latent variable z
Observation x
Auxiliary connections
Inference RNN h
Switches s
z_1^3 z_2^3 z_3^3 z_4^3 z_5^3
z_1^2 z_2^2 z_3^2 z_4^2 z_5^2
z_1^1 z_2^1 z_3^1 z_4^1 z_5^1
h_1^1 h_2^1 h_3^1 h_4^1 h_5^1
h_1^2 h_3^2 h_5^2
h_1^3 h_5^3
x_1^1 x_2^1 x_3^1 x_4^1 x_5^1
x_1^2 x_3^2 x_5^2
x_1^3 x_5^3

(a) Generation model.

(b) Inference network.

Figure 1. Generation model and structured inference network (with the *filtering* setting) of our proposed MR-HDMM for MR-MTS. The switches on incoming edges to a node (z_t^l) are the same, which is shown as s_t^l in Figure 2.

Figure 1 illustrates our MR-HDMM model which consists of the generation model and inference network. MR-HDMM captures the underlying data generation process by using the variational inference methods (Rezende et al., 2014; Kingma & Welling, 2013) and learns the *latent hierarchical structures* using *learnable switches* and *auxiliary connections* to adaptively encode the dependencies across the hierarchies and the timestamps. In particular, the switches use an *update-and-reuse* mechanism to control the updates of the latent states of a layer based on their previous states (i.e., utilizing temporal information) and the lower latent layers (i.e., utilizing the hierarchy). The switch triggers an update of the current states if it gets enough information from lower-level states, otherwise it reuses the previous states. Thus, the higher-level states act as summarized representations over the lower-level states and the switches help to propagate the temporal dependencies. The auxiliary connections (dashed lines in Figure 1(a)) between MR-MTS of different sampling rates and different latent layers help the model effectively capture the short-term and long-term temporal dependencies. Without the auxiliary connections, the higher-rate time series may mask the multi-scale dependencies present in the lower-rate time series data while propagating dependencies through bottom-up connections. Note that, the auxiliary connections are not related to the sampling rate of MR-MTS, and the sampling rate of higher-rate variable need not be a multiple of sampling rate of the lower-rate variable. Due to the flexibility of auxiliary connections, our MR-HDMM can also handle irregularly sampled time series data or missing data. We can a) zero-out the missing data points in the inference network and remove the corresponding auxiliary connections in the generation model during training, and b) interpolate missing values by adding auxiliary connections in the well-trained model.

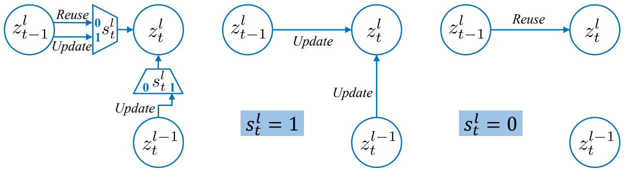


Figure 2. The switch mechanism for updating the latent states $\boldsymbol{z}_t^l$ in MR-HDMM. Left: The switch structure; Middle: Switch on ($s_t^l = 1$); Right: Switch off ($s_t^l = 0$).

3.1. Generation Model

Figure 1(a) shows the generation model of our MR-HDMM. The generation process of our MR-HDMM follows the **transition** and **emission** framework, which is obtained by applying deep recurrent neural networks to non-linear continuous state space models. The generation model is carefully designed to incorporate the switching mechanism and auxiliary connections in order to capture the multiple temporal dependencies present in MR-MTS.

Transition We design the transition process of the latent state $\boldsymbol{z}$ to capture the *hierarchical structure* for multiple temporal dependencies with learnable binary switches $\boldsymbol{s}$. For each non-bottom layer $l > 1$ and time step $t \geq 1$, we use a binary switch state s_t^l to control the updates of the corresponding latent states $\boldsymbol{z}_t^l$, as shown in Figure 2. s_t^l is obtained based on the values of the previous latent states $\boldsymbol{z}_{t-1}^l$ and the lower layer latent states $\boldsymbol{z}_t^{l-1}$ by a deterministic mapping $s_t^l = \mathbb{I}\left(g_{\theta_s}(\boldsymbol{z}_{t-1}^l, \boldsymbol{z}_t^{l-1}) \geq 0\right)$. When the switch is on (i.e., *update* operation, $s_t^l = 1$), $\boldsymbol{z}_t^l$ is updated based on $\boldsymbol{z}_{t-1}^l$ and $\boldsymbol{z}_t^{l-1}$ through a learnt transition distribution. We use a multivariate Gaussian distribution $\mathcal{N}\left(\boldsymbol{\mu}_t^l, \boldsymbol{\Sigma}_t^l | \boldsymbol{z}_{t-1}^l, \boldsymbol{z}_t^{l-1}; \theta_z\right)$ with mean and covariance given by $(\boldsymbol{\mu}_t^l, \boldsymbol{\Sigma}_t^l) = g_{\theta_z}(\boldsymbol{z}_{t-1}^l, \boldsymbol{z}_t^{l-1})$ as the transition distribution. When the switch is off (i.e., *reuse* operation, $s_t^l = 0$), $\boldsymbol{z}_t^l$ will be drawn from the same distribution as its previous states $\boldsymbol{z}_{t-1}^l$, which is $\mathcal{N}\left(\boldsymbol{\mu}_{t-1}^l, \boldsymbol{\Sigma}_{t-1}^l\right)$. Note, unlike Chung et al. (2016), we do not copy the previous state since our latent states are stochastic. The latent states of the first layer ($\boldsymbol{z}_{1:T}^1$) are always updated at each time step. In our model, g_{θ_s} is parameterized by a multilayer perceptron (MLP), and g_{θ_z} is parameterized by gated recurrent units (GRU) (Chung et al., 2014) to capture the temporal dependencies. With this *update-or-reuse* transition mechanism, higher latent layers tend to capture longer-term temporal dependencies through the bottom-up connections in the latent layers.

Emission Multi-rate multivariate observation $\boldsymbol{x}$ needs to be generated from $\boldsymbol{z}$ in the emission process. In order to embed the multiple temporal dependencies in the generated MR-MTS, we introduce *auxiliary connections* (denoted by the dashed lines in Figure 1(a)) from the higher latent layers to the lower rate time series. That is, time series of lth rate at time t (i.e., $\boldsymbol{x}_t^l$) is generated from all latent states up to lth layer $\boldsymbol{z}_t^{1:l}$ through emission distribution $\Pi\left(\boldsymbol{x}_t^l|\boldsymbol{z}_t^{1:l};\theta_x\right)$. The choice of emission distribution Π is flexible and depends on the data type. Multinomial distribution is used for categorical data, and Gaussian distribution is used for continuous data. Since all the data in our tasks are continuous, we use Gaussian distribution where the mean $\boldsymbol{\mu}^{(\boldsymbol{x})}{}_t^l$ and covariance $\boldsymbol{\Sigma}^{(\boldsymbol{x})}{}_t^l$ are determined by $g_{\theta_x}(\boldsymbol{z}_t^{1:l})$, which is parameterized by an MLP.

To summarize, the overall generation process is described in Algorithm 1. The parameter set of generation model is $\theta = \{\theta_x, \theta_z, \theta_s\}$. Given this, the joint probability of MR-MTS and the latent states/switches can be factorized by the following Equation (1).

$$
\begin{aligned}
& p_\theta\left(\boldsymbol{x}_{1:T}^{1:L}, \boldsymbol{z}_{1:T}^{1:L}, \boldsymbol{s}_{1:T}^{2:L}|\boldsymbol{z}_0^{1:L}\right) \\
=& p_\theta\left(\boldsymbol{x}_{1:T}^{1:L}|\boldsymbol{z}_{1:T}^{1:L}\right) p_\theta\left(\boldsymbol{z}_{1:T}^{1:L}, \boldsymbol{s}_{1:T}^{2:L}|\boldsymbol{z}_0^{1:L}\right) \\
=& \prod_{t=1}^{T} p_\theta\left(\boldsymbol{x}_t^{1:L}|\boldsymbol{z}_t^{1:L}\right) \cdot \prod_{t=1}^{T} p_\theta\left(\boldsymbol{z}_t^{1:L}, \boldsymbol{s}_t^{2:L}|\boldsymbol{z}_{t-1}^{1:L}\right) \\
=& \prod_{t=1}^{T}\prod_{l=1}^{L} p_{\theta_x}\left(\boldsymbol{x}_t^l|\boldsymbol{z}_t^{1:l}\right) \cdot \prod_{t=1}^{T} p_{\theta_z}\left(\boldsymbol{z}_t^1|\boldsymbol{z}_{t-1}^1\right) \\
& \cdot \prod_{t=1}^{T}\prod_{l=2}^{L} p_{\theta_s}\left(s_t^l|\boldsymbol{z}_{t-1}^l, \boldsymbol{z}_t^{l-1}\right) p_{\theta_z}\left(\boldsymbol{z}_t^l|\boldsymbol{z}_{t-1}^l, \boldsymbol{z}_t^{l-1}, s_t^l\right) \quad (1)
\end{aligned}
$$

In order to obtain the parameters of MR-HDMM, we need to maximize the log marginal likelihood of all MR-MTS data points, which is the summation of the log marginal likelihood $\mathcal{L}(\theta) = \log p_\theta\left(\boldsymbol{x}_{1:T}^{1:L}|\boldsymbol{z}_0^{1:L}\right)$ of each MR-MTS data point $\boldsymbol{x}_{1:T}^{1:L}$. The log marginal likelihood of one data point can be achieved by integrating out all possible $\boldsymbol{z}$ and $\boldsymbol{s}$ in Equation (1). Since $\boldsymbol{s}$ are deterministic binary variables, integrating them out can be done straightforwardly by taking their values in the likelihood. However, stochastic variable $\boldsymbol{z}$ cannot be analytically integrated out. Thus, we resort to the well-known variational principle (Jordan, 1998) and introduce our inference network below.

Algorithm 1 Generation model of MR-HDMM

```
Initialize $\boldsymbol{z}_0^{1:L} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$
for $t = 1, \ldots, T$ do
    $\left(\boldsymbol{\mu}_t^1, \boldsymbol{\Sigma}_t^1\right) = g_{\theta_z}(\boldsymbol{z}_{t-1}^1)$
    $\boldsymbol{z}_t^1 \sim \mathcal{N}\left(\boldsymbol{\mu}_t^1, \boldsymbol{\Sigma}_t^1\right)$ {Transition of the first layer.}
    for $l = 2, \cdots, L$ do
        $s_t^l = \mathbb{I}\left(g_{\theta_s}(\boldsymbol{z}_{t-1}^l, \boldsymbol{z}_t^{l-1}) \geq 0\right)$
        $\left(\boldsymbol{\mu}_t^l, \boldsymbol{\Sigma}_t^l\right) = \begin{cases} g_{\theta_z}(\boldsymbol{z}_{t-1}^l, \boldsymbol{z}_t^{l-1}) & \text{if } s_t^l = 1 \\ \left(\boldsymbol{\mu}_{t-1}^l, \boldsymbol{\Sigma}_{t-1}^l\right) & \text{otherwise.} \end{cases}$
        $\boldsymbol{z}_t^l \sim \mathcal{N}\left(\boldsymbol{\mu}_t^l, \boldsymbol{\Sigma}_t^l\right)$ {Transition of other layers.}
    end for
    for $l = 1, \cdots, L$ do
        $\left(\boldsymbol{\mu}^{(\boldsymbol{x})}{}_t^l, \boldsymbol{\Sigma}^{(\boldsymbol{x})}{}_t^l\right) = g_{\theta_x}(\boldsymbol{z}_t^{1:l})$
        $\boldsymbol{x}_t^l \sim \mathcal{N}\left(\boldsymbol{\mu}^{(\boldsymbol{x})}{}_t^l, \boldsymbol{\Sigma}^{(\boldsymbol{x})}{}_t^l\right)$ {Emission.}
    end for
end for
```

3.2. Inference Network

We design our inference network to mimic the structure of the generative model. The goal is to obtain an objective which can be optimized easily and which can make the model parameter learning amenable. Instead of directly maximizing $\mathcal{L}(\theta)$ w.r.t θ, we build an inference network with a tractable distribution q_ϕ, and maximize the variational *evidence lower bound (ELBO)* $\mathcal{F}(\theta, \phi) \leq \mathcal{L}(\theta)$ with respect to both θ and ϕ. Note, ϕ is the parameter set of the inference network which will is formally defined at the end of this section. The lower bound can be written as (please refer to the supplementary materials for full derivation):

$$
\begin{aligned}
&\mathcal{F}(\theta, \phi) = \mathbb{E}_{q_\phi}\left[\log p_\theta\left(\boldsymbol{x}_{1:T}^{1:L}|\boldsymbol{z}_{0:T}^{1:L}\right)\right] \\
&- D_{\mathrm{KL}}\left(q_\phi\left(\boldsymbol{z}_{1:T}^{1:L}, \boldsymbol{s}_{1:T}^{2:L}|\boldsymbol{x}_{1:T}^{1:L}, \boldsymbol{z}_0^{1:L}\right)\middle\| p_\theta\left(\boldsymbol{z}_{1:T}^{1:L}, \boldsymbol{s}_{1:T}^{2:L}|\boldsymbol{z}_0^{1:L}\right)\right)
\end{aligned} \tag{2}
$$

where the expectation of the first term is under $q_\phi\left(\boldsymbol{z}_{1:T}^{1:L}|\boldsymbol{x}_{1:T}^{1:L}, \boldsymbol{z}_0^{1:L}\right)$. To get a tight bound and an accurate estimate from our MR-HDMM, we need to properly design a new inference network as using the existing inference networks from SRNN (Fraccaro et al., 2016) or DMM (Krishnan et al., 2015) is not applicable for MR-MTS. In the following, we show how we design the inference network (Figure 1(b)) to obtain a good structured approximation to the posterior. First, we maintain the Markov properties of $\boldsymbol{z}$ in the inference network, which leads to the factorization:

$$
q_\phi\left(\boldsymbol{z}_{1:T}^{1:L}, \boldsymbol{s}_{1:T}^{2:L}|\boldsymbol{x}_{1:T}^{1:L}, \boldsymbol{z}_0^{1:L}\right) = \prod_{t=1}^{T} q_\phi\left(\boldsymbol{z}_t^{1:L}, \boldsymbol{s}_t^{2:L}|\boldsymbol{z}_{t-1}^{1:L}, \boldsymbol{x}_{1:T}^{1:L}\right) \tag{3}
$$

We then leverage the hierarchical structure and inherit the switches from the generation model into the

Table 1. Comparison of structured inference networks.

Inference network	Implemented with	RNN output	Captured in $\boldsymbol{h}_t^l$	Variational approximation for $\boldsymbol{z}_t^l$
filtering	forward RNN	$\boldsymbol{h}^{\text{forward}}$	$\boldsymbol{x}_{1:t}^l$	$q_\phi\left(\boldsymbol{z}_t^l\vert\boldsymbol{z}_{t-1}^l, \boldsymbol{z}_t^{l-1}, s_t^l, \boldsymbol{x}_{1:t}^{1:L}\right)$
smoothing	backward RNN	$\boldsymbol{h}^{\text{backward}}$	$\boldsymbol{x}_{t:T}^l$	$q_\phi\left(\boldsymbol{z}_t^l\vert\boldsymbol{z}_{t-1}^l, \boldsymbol{z}_t^{l-1}, s_t^l, \boldsymbol{x}_{t:T}^{1:L}\right)$
bi-direction	bi-directional RNN	$\left[\boldsymbol{h}^{\text{forward}}, \boldsymbol{h}^{\text{backward}}\right]$	$\boldsymbol{x}_{1:T}^l$	$q_\phi\left(\boldsymbol{z}_t^l\vert\boldsymbol{z}_{t-1}^l, \boldsymbol{z}_t^{l-1}, s_t^l, \boldsymbol{x}_{1:T}^{1:L}\right)$

inference network. That is, the same g_{θ_s} from the generation model is used in the inference network, i.e., $q_\phi\left(s_t^l|\boldsymbol{z}_{t-1}^l, \boldsymbol{z}_t^{l-1}, \boldsymbol{x}_{1:T}^{1:L}\right) = q_{\phi_s}\left(s_t^l|\boldsymbol{z}_{t-1}^l, \boldsymbol{z}_t^{l-1}\right) = p_{\theta_s}\left(s_t^l|\boldsymbol{z}_{t-1}^l, \boldsymbol{z}_t^{l-1}\right)$. Then, for each term in the righthand side of Equation (3) and for all $t = 1, \cdots, T$, we have:

$$
\begin{aligned}
& q_\phi\left(\boldsymbol{z}_t^{1:L}, \boldsymbol{s}_t^{2:L}|\boldsymbol{z}_{t-1}^{1:L}, \boldsymbol{x}_{1:T}^{1:L}\right) \\
=& q_\phi\left(\boldsymbol{z}_t^1|\boldsymbol{z}_{t-1}^1, \boldsymbol{x}_{1:T}^{1:L}\right) \\
& \cdot \prod_{l=2}^{L} q_\phi\left(s_t^l|\boldsymbol{z}_{t-1}^l, \boldsymbol{z}_t^{l-1}, \boldsymbol{x}_{1:T}^{1:L}\right) q_\phi\left(\boldsymbol{z}_t^l|\boldsymbol{z}_{t-1}^l, \boldsymbol{z}_t^{l-1}, s_t^l, \boldsymbol{x}_{1:T}^{1:L}\right) \\
=& q_\phi\left(\boldsymbol{z}_t^1|\boldsymbol{z}_{t-1}^1, \boldsymbol{x}_{1:T}^{1:L}\right) \\
& \cdot \prod_{l=2}^{L} p_{\theta_s}\left(s_t^l|\boldsymbol{z}_{t-1}^l, \boldsymbol{z}_t^{l-1}\right) q_\phi\left(\boldsymbol{z}_t^l|\boldsymbol{z}_{t-1}^l, \boldsymbol{z}_t^{l-1}, s_t^l, \boldsymbol{x}_{1:T}^{1:L}\right) \qquad (4)
\end{aligned}
$$

Thus, the inference network can be factorized by Equation (3) and (4). Note, we also can factorize generative model based on Equation (1). Given these, we further factorize the ELBO in Equation (2) as a summation of expectations of conditional log likelihood and KL divergence terms over time steps and hierarchical layers:

$$
\begin{aligned}
& \mathcal{F}(\theta, \phi) = \sum_{t=1}^{T}\sum_{l=1}^{L} \mathbb{E}_{\mathcal{Q}^*\left(\boldsymbol{z}_t^{1:l}\right)} \log p_{\theta_x}\left(\boldsymbol{x}_t^l|\boldsymbol{z}_t^{1:l}\right) \\
& + \sum_{t=1}^{T} \mathbb{E}_{\mathcal{Q}^*\left(\boldsymbol{z}_{t-1}^1\right)} D_{\mathrm{KL}}\left(q_\phi\left(\boldsymbol{z}_t^1|\boldsymbol{x}_{1:T}^{1:L}, \boldsymbol{z}_{t-1}^1\right)\middle\|p_\theta\left(\boldsymbol{z}_t^1|\boldsymbol{z}_{t-1}^1\right)\right) \\
& + \sum_{t=1}^{T}\sum_{l=2}^{L} \mathbb{E}_{\mathcal{Q}^*\left(\boldsymbol{z}_{t-1}^1, \boldsymbol{z}_t^{l-1}\right)} \\
& \quad D_{\mathrm{KL}}\left(q_\phi\left(\boldsymbol{z}_t^l|\boldsymbol{x}_{1:T}^{1:L}, \boldsymbol{z}_{t-1}^l, \boldsymbol{z}_t^{l-1}\right)\middle\|p_\theta\left(\boldsymbol{z}_t^1|\boldsymbol{z}_{t-1}^1, \boldsymbol{z}_t^{l-1}\right)\right) \qquad (5)
\end{aligned}
$$

where $\mathcal{Q}^*(\cdot)$ denotes the marginal distribution of $(\cdot)$ from q_ϕ. The details about the factorization and the marginalized distribution are provided in the supplementary materials.

Parameterization of inference network We parameterize the inference network and construct the variational approximation $q_\phi\left(\boldsymbol{z}_t^l|\boldsymbol{z}_{t-1}^l, \boldsymbol{z}_t^{l-1}, s_t^l, \boldsymbol{x}_{1:T}^{1:L}\right)$ used in Equation 5 by deep learning models. First, we use L RNNs to capture MR-MTS with L different sampling rates such that each rate is modeled by one RNN model separately. Second, to obtain lth latent states $\boldsymbol{z}_t^l$ of the inference network at time step t, we not only use the previous latent states $\boldsymbol{z}_{t-1}^l$ and the lower layer latent states $\boldsymbol{z}_t^{l-1}$ but also take the lth RNN output denoted by $\boldsymbol{h}_t^l$ as an input. Third, we reuse the same latent state distribution and switch mechanism from the generation model to generate $\boldsymbol{z}$ of the inference network. To be more specific, $\boldsymbol{z}_t^l$ is drawn from a multivariate normal distribution, where the mean and covariance are reused from those of $\boldsymbol{z}_{t-1}^l$ if $s_t^l = 1$ and $l > 1$, otherwise the mean and covariance are modeled by gated recurrent units (GRU) with input $\left[\boldsymbol{h}_t^l, \boldsymbol{z}_{t-1}^l, \boldsymbol{z}_t^{l-1}\right]$. The choice of the RNN models for $\boldsymbol{h}_t^l$ affects what and how the information at other time steps is considered in the approximation at time t, i.e. the form of $q_\phi\left(\boldsymbol{z}_t^l|\boldsymbol{z}_{t-1}^l, \boldsymbol{z}_t^{l-1}, s_t^l, \boldsymbol{x}_{1:T}^{1:L}\right)$. Inspired by Krishnan et al. (2016), we construct the variational approximation in three settings (*filtering, smoothing, bi-direction*) for forecasting and interpolation tasks. In *filtering* setting, we only consider the information up to time t (i.e., $\boldsymbol{x}_{1:t}^{1:L}$) using forward RNNs. By doing this, we have $\boldsymbol{h}_t^l = \boldsymbol{h}_t^{l\,\text{forward}} = RNN^{\text{forward}}\left(\boldsymbol{h}_{t-1}^{l\,\text{forward}}, \boldsymbol{x}_t^l\right)$, and thus $q_\phi\left(\boldsymbol{z}_t^l|\boldsymbol{z}_{t-1}^l, \boldsymbol{z}_t^{l-1}, s_t^l, \boldsymbol{x}_{1:T}^{1:L}\right) = q_\phi\left(\boldsymbol{z}_t^l|\boldsymbol{z}_{t-1}^l, \boldsymbol{z}_t^{l-1}, s_t^l, \boldsymbol{x}_{1:t}^{1:L}\right)$. The filtering setting does not use future information, so it is suitable for forecasting task at future time step $t' > T$. For interpolation tasks, we can use backward RNNs to utilize the information after time t (i.e., $\boldsymbol{x}_{t:T}^{1:L}$) with $\boldsymbol{h}_t^l = \boldsymbol{h}_t^{l\,\text{backward}} = RNN^{\text{backward}}\left(\boldsymbol{h}_{t+1}^{l\,\text{backward}}, \boldsymbol{x}_t^l\right)$, or bi-directional RNNs to utilize information across all time steps, which is $\boldsymbol{x}_{1:T}^{1:L}$, at any time t with $\boldsymbol{h}_t^l = \left[\boldsymbol{h}_t^{l\,\text{forward}}, \boldsymbol{h}_t^{l\,\text{backward}}\right]$. These two models lead to *smoothing* and *bidirection* settings, respectively. We summarize the three inference networks in Table 1. We use ϕ_h and ϕ_z to denote the parameter sets related to $\boldsymbol{h}$ and $\boldsymbol{z}$ respectively and use $\phi = \{\phi_h, \phi_z, \phi_s = \theta_s\}$ to represent the parameter set of the inference network.

3.3. Learning the Parameters

We jointly learn the parameters (θ, ϕ) of the generative model p_θ and the inference network q_ϕ by maximizing the ELBO in Equation (5). The main challenge in the optimization is obtaining the gradients of all the terms under the correct expectation i.e, $\mathbb{E}_{\mathcal{Q}^*}$. We use stochastic backpropagation (Kingma & Welling, 2013) for estimating all these gradients and train the model by stochastic gradient descent (SGD) approaches. We employ ancestral sampling techniques to obtain the samples $\boldsymbol{z}$. That is, we draw all samples $\boldsymbol{z}$ in a sequential way from time 1 to T and from layer 1 to L. Given the samples from previous layer $l-1$ or previous time $t-1$, the new samples at time t and layer l will be distributed according to the marginal distribution $\mathcal{Q}^*$. Notice

Algorithm 2 Learning MR-HDMM with stochastic back-propagation and SGD

```
Require: $\mathcal{X}$: a set of MR-MTS of $L$ sampling rates; Initial $(\theta, \phi)$
while not converged do
    Choose a random minibatch of MR-MTS $\mathcal{X}' \subset \mathcal{X}$
    for each sample $\boldsymbol{x}_{1:T}^{1:L} \in \mathcal{X}'$ do
        Compute $\boldsymbol{h}_{1:T}^{1:L}$ by inference network $\phi_h$ on input $\boldsymbol{x}_{1:T}^{1:L}$
        Sample $\widehat{\boldsymbol{z}_0^{1:L}} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$
        for $t = 1, \cdots, T$ do
            Estimate $\boldsymbol{\mu}_t^{1(\phi)}, \boldsymbol{\Sigma}_t^{1(\phi)}$ by $\phi_z$, and $\boldsymbol{\mu}_t^1, \boldsymbol{\Sigma}_t^1$ by $\theta_z$,
            given samples $\widehat{\boldsymbol{z}_{t-1}^1}$ and $\boldsymbol{h}_t^1$
            Based on $\boldsymbol{\mu}_t^{1(\phi)}, \boldsymbol{\Sigma}_t^{1(\phi)}, \boldsymbol{\mu}_t^1, \boldsymbol{\Sigma}_t^1$, compute the gradient
            of $D_{\mathrm{KL}}\left(q_\phi\left(\boldsymbol{z}_t^1|\cdot\right) \middle\| p_\theta\left(\boldsymbol{z}_t^1|\cdot\right)\right)$
            Sample $\widehat{\boldsymbol{z}_t^1} \sim \mathcal{N}\left(\boldsymbol{\mu}_t^{1(\phi)}, \boldsymbol{\Sigma}_t^{1(\phi)}\right)$
            for $l = 2, \cdots, L$ do
                Compute $s_t^l$ by $\theta_s$ from samples $\widehat{\boldsymbol{z}_{t-1}^l}$ and $\widehat{\boldsymbol{z}_t^{l-1}}$
                Estimate $\boldsymbol{\mu}_t^{l(\phi)}, \boldsymbol{\Sigma}_t^{l(\phi)}$ by $\phi_z$, and $\boldsymbol{\mu}_t^l, \boldsymbol{\Sigma}_t^l$ by $\theta_z$,
                given samples $\widehat{\boldsymbol{z}_{t-1}^l}, \widehat{\boldsymbol{z}_t^{l-1}}, s_t^l$, and $\boldsymbol{h}_t^l$
                Based on $\boldsymbol{\mu}_t^{l(\phi)}, \boldsymbol{\Sigma}_t^{l(\phi)}, \boldsymbol{\mu}_t^l, \boldsymbol{\Sigma}_t^l$, compute the gradi-
                ent of $D_{\mathrm{KL}}\left(q_\phi\left(\boldsymbol{z}_t^l|\cdot\right) \middle\| p_\theta\left(\boldsymbol{z}_t^l|\cdot\right)\right)$
                Sample $\widehat{\boldsymbol{z}_t^l} \sim \mathcal{N}\left(\boldsymbol{\mu}_t^{l(\phi)}, \boldsymbol{\Sigma}_t^{l(\phi)}\right)$
            end for
            Compute the gradient of $\log p_{\theta_x}\left(\boldsymbol{x}_t^l|\widehat{\boldsymbol{z}_t^{1:l}}\right)$
        end for
    end for
    Update $(\theta, \phi)$ using all gradients
end while
```

that all terms of $D_{\mathrm{KL}}\left(q_\phi\left(\boldsymbol{z}_t^l|\cdot\right) \middle\| p_\theta\left(\boldsymbol{z}_t^l|\cdot\right)\right)$ in Equation (5) are KL divergences between two multivariate Gaussian distributions, and $p_{\theta_x}\left(\boldsymbol{x}_t^l|\boldsymbol{z}_t^{1:l}\right)$ is also a multivariate Gaussian distribution. Thus, all the required gradients can be estimated analytically from the samples drawn in our proposed way. Algorithm 2 shows the overall learning procedure.

4. Experiments

We conducted experiments on two real-world datasets - the MIMIC-III healthcare dataset and the USHCN climate dataset - and answer the following questions: (a) How does our proposed model perform when compared to the existing state-of-the-art approaches? (b) To what extent, are the proposed learnable hierarchical latent structure and auxiliary connections useful to model the data generation process? (c) How do we interpret the hierarchy learned by the proposed model? In the remainder of this section, we will describe the datasets, methods, empirical results and interpretations to answer the above questions.

4.1. Datasets and Experimental Design

MIMIC-III dataset MIMIC-III is a public de-identified dataset collected at Beth Israel Deaconess Medical Center from 2001 to 2012 (Johnson et al., 2016). It contains over 58,000 hospital admission records of 38,645 adults and 7,875 neonates. For our experiments, we chose 10,709 adult admission records and extracted 62 temporal features from the first 72 hours. These features had one of the three sampling rates of 1 hour, 4 hours and 12 hours. To fill-in any missing entries in our dataset we used forward or linear imputation similar to Che et al. (2016). To ensure fair comparison, we only evaluate and compare all the models on the original time-series (i.e. non-imputed data). Our main tasks on the MIMIC-III dataset are forecasting on time series with all rates, and interpolation of the low-rate time series values.

USHCN climate dataset The U.S. Historical Climatology Network Monthly (USHCN) dataset (Menne et al., 2010) is publicly available and consists of daily meteorological data of 54 stations in California spanning from 1887 to 2009. It has five climate variables for each station: a) daily maximum temperature, b) daily minimum temperature, c) whether it was a snowy day or not, d) total daily precipitation, and e) daily snow precipitation. We preprocessed this dataset to extract daily climate data for 100 consecutive years starting from 1909. To get multi-rate time series data, we extract 208 features and split all features into 3 groups with sampling rates of 1 day, 5 days, and 10 days respectively. This public dataset has been carefully processed by National Oceanic and Atmospheric Administration (NOAA) to ensure quality control and it has no missing entries. Our tasks on this dataset are climate forecasting on all features and interpolation on 5-day and 10-day sampled data.

Tasks We use the proposed MR-HDMM on two prediction tasks: multi-rate time series forecasting and low-rate time series interpolation. Since both datasets have 3 different sampling rates, we use HSR/MSR/LSR to denote high/medium/low sampling rate respectively.

- *Forecasting*: Predict the future multivariate time series based on its history. For MIMIC-III dataset, we predict the last 24 hrs time series based on the first (previous) 48 hours time series data. In USHCN dataset, we forecast the climate for the next 30 days based on the observations of the previous year.
- *Interpolation*: Fill-in the low rate time series based on co-evolving higher rate time series data. For MIMIC-III dataset, we down-sampled 8 features from MSR to LSR and then performed interpolation task by up-sampling these 8 features back to MSR. For USHCN dataset, the interpolation task involved up-sampling the MSR and LSR features to HSR features, i.e. up-sample 5-day and 10-day data to 1-day. We demonstrate in-sample interpolation (i.e. interpolation within training dataset) and out-sample interpolation (i.e. interpolation in the testing dataset) on the MIMIC-III dataset and in-sample interpolation on the USHCN dataset.

Baselines We compare MR-HDMM with several strong baselines in these two tasks. Additionally, to show the advantage of *learnable hierarchical latent structure* and *auxiliary connections*, we simplify MR-HDMM into two other models for comparison: (a) Multi-Rate Deep Markov Models (MR-DMM) which removes the hierarchical structure in latent space; (b) Hierarchical Deep Markov Models (HDMM) which drops the auxiliary connections between the lower-rate time series and higher level latent layers. MR-DMM and HDMM are discussed in the supplementary materials.

For forecasting tasks, we compare MR-HDMM with the following baseline models:

- *Single-rate*: Kalman Filters (KF), Vector Auto-Regression (VAR), Long-Short Term Memory (LSTM) (Hochreiter & Schmidhuber, 1997), Phased-LSTM (PLSTM) (Neil et al., 2016), Deep Markov Models (DMM) (Krishnan et al., 2015) and Hierarchical Multiscale Recurrent Neural Networks (HM-RNN) (Chung et al., 2016).
- *Multi-rate*: Multiple Kalman Filters (MKF) (Drolet et al., 2000), Multi-rate Kalman Filters (MR-KF) (Safari et al., 2014), Multi-Rate Deep Markov Models (MR-DMM) and Hierarchical Deep Markov Models (HDMM).

For interpolation task, we compare MR-HDMM with the following baseline models:

- *Imputation methods*: Mean imputation (Simple-Mean), Cubic Spline (CubicSpline) (De Boor et al., 1978), Multiple Imputations by Chained Equations (MICE) (White et al., 2011), MissForest (Stekhoven & Bühlmann, 2011), SoftImpute (Mazumder et al., 2010).
- *Deep learning models*: Deep Markov Models (DMM), Multi-Rate Deep Markov Models (MR-DMM) and Hierarchical Deep Markov Models (HDMM).

4.2. Evaluation and Implementation Details

We show the evaluation results of our MR-HDMM on the following: (a) Forecasting: we generate the next latent state using the learned transition distribution and then generate observations from these new latent states; (b) Interpolation: we use the mode of the approximated posterior in the generation model to generate the unseen data in low-rate time series. (c) Inference: we take multi-rate time series as the input to obtain the approximate posterior of latent states.

For generation model in MR-HDMM, we use multivariate Gaussian with diagonal covariance for both emission distribution and transition distribution. We parameterized the emission mapping g_{θ_x} by a 3-layer MLP with `ReLU` activations, the transition mapping g_{θ_z} by gated recurrent unit (GRU), and mapping g_{θ_s} by a 3-layer MLP with `ReLU` activations on the hidden layers and linear activations on the output layer. For inference networks, we adopt *filtering* setting for forecasting and *bidirection* setting for interpolation from Table 1 with 3-layer GRUs. To update θ_s, we replace the sign function with a sharp sigmoid function during training, and use the indicator function during validation. The single-rate baseline models cannot handle multi-rate data directly, and we up-sample all the lower rate data into higher rate data using linear interpolation. We use the `stats-toolbox` (Seabold & Perktold, 2010) in python for the VAR model implementation. We use `pykalman` (Duckworth, 2013) to implement all the KF-based models. The implementation details of the KF-based methods are discussed in the supplementary materials. For LSTM and PLSTM model, we use one layer with 100 neurons to model the time-series, and then apply a soft-max regressor on top of the last hidden state to do regression.

To ensure a fair comparison, we use roughly the same amount of parameters for all models. For experiments on USHCN dataset, train/valid/test sets were split as 70/10/20. For experiments on MIMIC-III, we used 5-fold cross validation (train on 3 folds, validate on another fold and test on the remaining fold) and report the average Mean Squared Error (MSE) of 5 runs for both forecasting and interpolation tasks. Note that, we train all the deep learning models with the Adam optimization method (Kingma & Ba, 2014) and use validation set to find the best weights, and report the results on the held-out test set. All the input variables are normalized to be of 0 mean and 1 standard deviation.

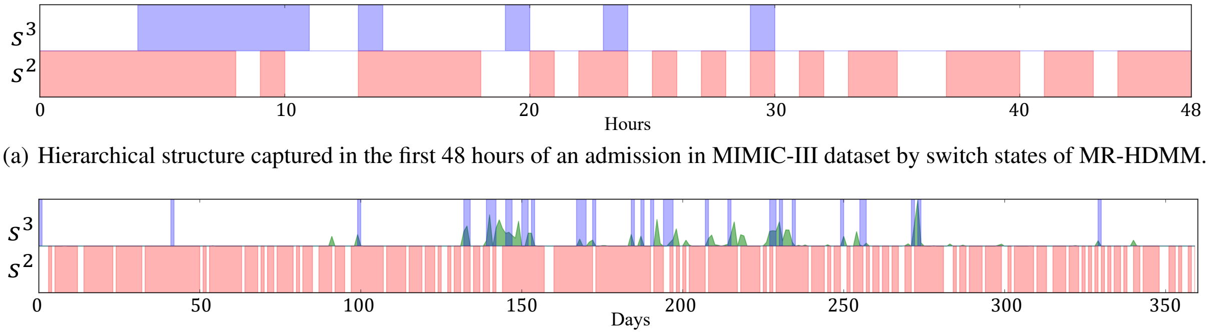
4.3. Quantitative Results

Forecasting Table 2 and 3 respectively show the forecasting results on MIMIC-III and USHCN datasets in terms of MSE. Our proposed MR-HDMM outperforms all the competing multi-rate latent space models by at least 5% and beats the single-rate models by at least 15% on both datasets with all features. Our model also performs the best on single-rate HSR and MSR forecasting tasks, and performs well on the LSR forecasting task on MIMIC-III and USHCN datasets.

Table 2. Forecasting results (MSE) on MIMIC-III.

	All	HSR	MSR	LSR
KF	1.91×10^{18}	3.34×10^{18}	8.38×10^{9}	1.22×10^{6}
VAR	1.233	1.735	0.779	0.802
DMM	1.530	1.875	1.064	1.070
HM-RNN	1.388	1.846	0.904	**0.713**
LSTM	1.512	1.876	1.006	1.036
PLSTM	1.244	1.392	1.030	1.056
MKF	2.05×10^{18}	3.58×10^{18}	3.63×10^{4}	9.54×10^{2}
MR-KF	1.691	2.289	0.944	0.860
MR-DMM	1.061	1.192	0.723	1.065
HDMM	1.047	1.168	0.702	1.076
MR-HDMM	**0.996**	**1.148**	**0.678**	0.911

Interpolation Table 4 shows the interpolation results on the two datasets. Since VAR and LSTM cannot be directly



(a) Hierarchical structure captured in the first 48 hours of an admission in MIMIC-III dataset by switch states of MR-HDMM.

(b) Hierarchical structure (red & blue blocks) captured along with precipitation time series (green curve) in the one-year observation in USHCN dataset by switch states of MR-HDMM.

Figure 3. Interpretable latent space learned by MR-HDMM model.

Table 3. Forecasting results (MSE) on USHCN.

	All	HSR	MSR	LSR
KF	1.236	1.254	1.190	1.148
VAR	2.415	2.579	1.921	1.748
DMM	0.795	0.608	0.903	0.877
HM-RNN	0.692	0.594	1.151	**0.775**
LSTM	0.849	0.688	0.934	0.928
PLSTM	0.813	0.710	0.870	0.915
MKF	1.212	1.082	1.727	1.518
MR-KF	0.628	0.542	0.986	0.799
MR-DMM	0.667	0.611	0.847	0.875
HDMM	0.626	0.568	0.815	0.836
MR-HDMM	**0.591**	**0.541**	**0.742**	0.795

Table 4. Interpolation results (MSE) on MIMIC-III and USHCN.

	MIMIC-III		USHCN
	In-sample	Out-sample	In-sample
Simple-Mean	3.812	3.123	0.987
CubicSpline	3.713	3.212×10^4	0.947
MICE	3.747	7.580×10^2	0.670
MissForest	3.863	3.027	0.941
SoftImpute	3.715	3.086	0.759
DMM	3.714	3.027	0.782
MR-DMM	3.710	3.021	0.696
HDMM	3.790	3.100	0.750
MR-HDMM	**3.582**	**2.921**	**0.626**

used for the interpolation task, we focus on evaluating generative models and imputation methods. From Table 4, we observe that our proposed model outperforms the baselines and the competing multi-rate latent space models by a large margin on all the interpolation tasks on these two datasets.

Table 5. Lower bound of log-likelihood of generative models. Higher values are better.

	DMM	MR-DMM	HDMM	MR-HDMM
MIMIC-III	−1.54	2.62	10.54	**15.27**
USHCN	2.37	14.37	17.25	**33.62**

Inference We also compare the lower bound of log-likelihood of all generative models in Table 5. The higher lower bound value indicates a better fitted model given the training data. Our MR-HDMM model achieves the best performance on both datasets.

4.4. Discussion

In all our experiments, MR-HDMM outperforms other generative models by a significant margin. Considering that all the deep generative models have the same amount of parameters, this improvement empirically demonstrates the effectiveness of our proposed *learnable latent hierarchical structure* and *auxiliary connections*. In Figure 3(a) and 3(b), we visualize the latent hierarchical structure of MR-HDMM learned from the first 48 hours of an admission in MIMIC-III dataset and one-year climate observations in USHCN dataset. A color block indicates that the latent state z_t^l is updated from z_{t-1}^l and z_t^{l-1} (*update*), while the white block indicates z_t^l is generated from the same distribution of z_{t-1}^l (*reuse*). As expected, the higher latent layers tend to update less frequently and capture the long-term temporal dependencies. To understand learned hierarchical structure more intuitively, we also show *precipitation* time series from USCHN dataset along with learned switches in Figure 3(b). We observe that the higher latent layer tends to update along with the precipitation, which is reasonable since *precipitation* makes significant changes to the underlying weather condition which is captured by the higher latent layer.

5. Summary

We proposed the Multi-Rate Hierarchical Deep Markov Model (MR-HDMM) - a novel deep generative model for forecasting and interpolation tasks on multi-rate multivariate time series (MR-MTS) data. MR-HDMM models the data generation process by learning a latent hierarchical structure using auxiliary connections and learnable switches to capture the temporal dependencies. Empirically we showed that our proposed model outperforms the existing single-rate and multi-rate models on healthcare and climate datasets.

Acknowledgments

This work is supported in part by NSF Research Grant IIS-1254206 and IIS-1539608, and MURI grant W911NF-11-1-0332. The views and conclusions are those of the authors and should not be interpreted as representing the official policies of the funding agency, or the U.S. Government.

References

Armesto, L., Tornero, J., and Vincze, M. On multi-rate fusion for non-linear sampled-data systems: Application to a 6d tracking system. *Robotics and Autonomous Systems*, 56(8):706–715, 2008.

Che, Z., Purushotham, S., Cho, K., Sontag, D., and Liu, Y. Recurrent neural networks for multivariate time series with missing values. *arXiv preprint arXiv:1606.01865*, 2016.

Chung, J., Gulcehre, C., Cho, K., and Bengio, Y. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

Chung, J., Ahn, S., and Bengio, Y. Hierarchical multiscale recurrent neural networks. *arXiv preprint arXiv:1609.01704*, 2016.

De Boor, C., De Boor, C., Mathématicien, E.-U., De Boor, C., and De Boor, C. *A practical guide to splines*, volume 27. Springer-Verlag New York, 1978.

Drolet, L., Michaud, F., and Côté, J. Adaptable sensor fusion using multiple kalman filters. In *Intelligent Robots and Systems, 2000.(IROS 2000). Proceedings. 2000 IEEE/RSJ International Conference on*, volume 2, pp. 1434–1439. IEEE, 2000.

Duckworth, D. pykalman, an implementation of the kalman filter, kalman smoother, and em algorithm in python. https://pykalman.github.com, 2013.

Durbin, J. and Koopman, S. J. *Time series analysis by state space methods*, volume 38. OUP Oxford, 2012.

El Hihi, S. and Bengio, Y. Hierarchical recurrent neural networks for long-term dependencies. In *NIPS*, 1995.

Fraccaro, M., Sønderby, S. K., Paquet, U., and Winther, O. Sequential neural models with stochastic layers. In *NIPS*, 2016.

Gan, Z., Li, C., Henao, R., Carlson, D. E., and Carin, L. Deep temporal sigmoid belief networks for sequence modeling. In *NIPS*, 2015.

Graves, A., Mohamed, A.-r., and Hinton, G. Speech recognition with deep recurrent neural networks. In *ICASSP*, 2013.

Hamilton, J. D. *Time series analysis*, volume 2. Princeton university press Princeton, 1994.

Hermans, M. and Schrauwen, B. Training and analysing deep recurrent neural networks. In *Advances in neural information processing systems*, pp. 190–198, 2013.

Hochreiter, S. and Schmidhuber, J. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

Houtekamer, P. L. and Mitchell, H. L. A sequential ensemble kalman filter for atmospheric data assimilation. *Monthly Weather Review*, 129(1):123–137, 2001.

Johnson, A., Pollard, T., Shen, L., Lehman, L., Feng, M., Ghassemi, M., Moody, B., Szolovits, P., Celi, L., and Mark, R. Mimic-iii, a freely accessible critical care database. *Scientific Data*, 2016.

Jordan, M. I. *Learning in graphical models*, volume 89. Springer Science & Business Media, 1998.

Kalman, R. E. et al. A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, 82 (1):35–45, 1960.

Kingma, D. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Kingma, D. P. and Welling, M. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Koutnik, J., Greff, K., Gomez, F., and Schmidhuber, J. A clockwork rnn. In *International Conference on Machine Learning*, pp. 1863–1871, 2014.

Krishnan, R. G., Shalit, U., and Sontag, D. Deep kalman filters. *arXiv preprint arXiv:1511.05121*, 2015.

Krishnan, R. G., Shalit, U., and Sontag, D. Structured inference networks for nonlinear state space models. *arXiv preprint arXiv:1609.09869*, 2016.

Martens, J. and Sutskever, I. Learning recurrent neural networks with hessian-free optimization. In *ICML*, 2011.

Mazumder, R., Hastie, T., and Tibshirani, R. Spectral regularization algorithms for learning large incomplete matrices. *Journal of machine learning research*, 11(Aug): 2287–2322, 2010.

Menne, M., Williams Jr, C., and Vose, R. Long-term daily and monthly climate records from stations across the contiguous united states. 2010.

Mikolov, T., Karafiát, M., Burget, L., Cernockỳ, J., and Khudanpur, S. Recurrent neural network based language model. In *Interspeech*, volume 2, pp. 3, 2010.

Negenborn, R. *Robot localization and Kalman filters*. PhD thesis, Utrecht University, 2003.

Neil, D., Pfeiffer, M., and Liu, S.-C. Phased lstm: Accelerating recurrent network training for long or event-based sequences. In *Advances in Neural Information Processing Systems*, pp. 3882–3890, 2016.

Pascanu, R., Mikolov, T., and Bengio, Y. On the difficulty of training recurrent neural networks. *ICML (3)*, 28:1310–1318, 2013.

Rabiner, L. R. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

Reinsel, G. C. *Elements of multivariate time series analysis*. Springer Science & Business Media, 2003.

Rezende, D. J., Mohamed, S., and Wierstra, D. Stochastic backpropagation and approximate inference in deep generative models. *arXiv preprint arXiv:1401.4082*, 2014.

Safari, S., Shabani, F., and Simon, D. Multirate multisensor data fusion for linear systems using kalman filters and a neural network. *Aerospace Science and Technology*, 39: 465–471, 2014.

Seabold, S. and Perktold, J. Statsmodels: Econometric and statistical modeling with python. In *Proceedings of the 9th Python in Science Conference*, 2010.

Socher, R., Lin, C. C., Manning, C., and Ng, A. Y. Parsing natural scenes and natural language with recursive neural networks. In *ICML*, 2011.

Stekhoven, D. J. and Bühlmann, P. Missforest—non-parametric missing value imputation for mixed-type data. *Bioinformatics*, 28(1):112–118, 2011.

White, I. R., Royston, P., and Wood, A. M. Multiple imputation using chained equations: issues and guidance for practice. *Statistics in medicine*, 30(4):377–399, 2011.

Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A. C., Salakhutdinov, R., Zemel, R. S., and Bengio, Y. Show, attend and tell: Neural image caption generation with visual attention. In *ICML*, 2015.