ENERGY-BASED RECURRENT MODEL FOR STOCHASTIC MODELING OF MUSIC

Yingru Liu[†] *and Dongliang Xie*[‡] *and Xin Wang*[†]

† Stony Brook University, Stony Brook, NY, USA
‡ Beijing University of Posts and Telecommunications, Beijing, China

ABSTRACT

The aim of this work is to more accurately model the stochastic process of music-related data, which is essential for many AI applications in musicology. When music is naturally represented as a sequence of vectorized frames, existing models generally cannot well capture the correlation of the elements inside each frame. We propose an energy-based model called Chain Graphical Recurrent Neural Network (CGRNN) to explore the correlation of elements for more accurate modeling of the dynamics of music. In CGRNN, a probabilistic sub-structure named Conditional spike-and-slab Restricted Boltzmann Machine (C-ssRBM) is defined to better model the conditional covariance and joint distribution of elements in a frame. Besides, CGRNN is capable of tracking the evolution of music and extracting sparse features with an efficient design of temporal transition. With the estimated stochastic process of music, we further implement CGRNN to generate melodious music automatically. Extensive empirical evaluations of multiple unsupervised learning tasks are conducted on symbolic MIDI and audio sounds to demonstrate the performance of our model.

Index Terms— Automated Music Generation, Stochastic Deep Learning Model, Unsupervised Learning

1. INTRODUCTION

Music is an important media to express human emotion and feeling. In recent years, we have experienced an evolution in diverse music styles and trends. Music is an art of time, and music data are sequential in nature. An accurate modeling of the temporal dependency among music data is definitely crucial to understand the dynamics and regular patterns of melodies. Besides, an accurate temporal model has been shown to be essential for many AI applications in music [6]. Therefore, we intend to more accurately model the stochastic process of music-related data and apply the estimated dynamics for feature extraction and automated music generation.

Music can be recorded as either symbolic MIDI or audio waveform. Both types of data form the sequence of vectorized frames with strong correlation. In symbolic MIDI, the music information is transformed into a sequence of binary vectors, where each element of a vector indicates if there exists a specific pitch [3]. An example of symbolic MIDI is given in Fig. 1. In each time slot, a group of pitches are activated together to produce a chord or melody. The correlation among pitches is essential to estimate the statistic property of the symbolic MIDI. Audio waveform can also be organized as a sequence of frames, where each frame is formed as a vector containing a fixed number of samples taken within a time duration [4, 7]. The successive sampling points within a frame are impacted by physical properties of the instrument and are highly-correlated as well.

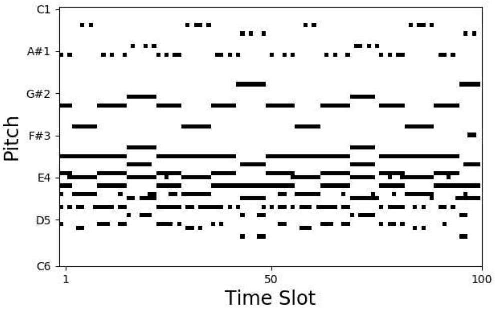

Fig. 1. Matrix representation of a symbolic MIDI file

A few stochastic deep learning models have been proposed to track the dynamics of sequential data for Natural Language Processing and Polyphonic Music Generation [3, 4, 7, 2]. However, the majority of these models assume a mixture model with a diagonal conditional covariance matrix to estimate the distribution of data within a frame, which makes them less practical to model the complicated correlation of music-related data.

To better estimate the joint distribution of vectorized frames, we first design a variant of ssRBM [5] named Conditional ssRBM (C-ssRBM) that can more flexibly model the complicated correlation among elements within each frame. Furthermore, we propose an efficient transition structure of Recurrent Neural Network (RNN) to more accurately track the dynamics of data sequences. As the topology of our model is equivalent to a Chain Graphical Model that has both directed and undirected links, we named the whole framework Chain Graphical Recurrent Neural Network (CGRNN).

The work is supported by the National Science Foundation (NSF) under the grants CNS-1526843 and ECCS-1731238.

978-1-5386-9552-4/19/$31.00 ©2019 IEEE
DOI 10.1109/ICME.2019.00049

The major contributions of this paper are two-fold:

- We propose CGRNN to accurately capture the stochastic process of music, and conduct extensive experiments to demonstrate the performance.
- We further apply CGRNN for automated music generation. Perceptual evaluation shows that our model is capable of generating melodious music without human knowledge.

The rest of this paper is organized as follows. In Section 2, we first establish the formulation of stochastic modeling problem to music. In Section 3, the whole framework of CGRNN is proposed and the details of C-ssRBM are described in Section 4. After that, experiments are presented in Section 5 and the final conclusion is given in Section 6.

2. PROBLEM FORMULATION

The sequence of data in music is represented as $V_{1:N} = [V_1, \cdots, V_N]$. For the symbolic MIDI, a $V_n \in \{0,1\}^{128}$ is a binary vector where each element represents a specific note. For the audio waveform, $V_n \in \mathbb{R}^L$ is a real-value vector that contains L consecutive points of the signal [4]. To model the dynamics of music, we need to estimate the stochastic process of the sequential representation of symbolic MIDI and audio sounds. Given a set of music data $\{V_{1:N}^{(i)}\}$, we look for a stochastic model $\widehat{P}(V_{1:N})$ with system parameters θ such that the distance between $\widehat{P}(V_{1:N})$ and the distribution of the observed data is minimized. This is equivalent to training a model by maximizing the log-likelihood as follows:

$$\text{maximize} \sum_i \log \widehat{P}(V_{1:N}^{(i)}), \text{ w.r.t } \widehat{P}(V_{1:N}),\ \theta. \quad (1)$$

In a music sequence, the n_{th} point V_n is a high-dimensional vector, and there exists strong correlation among its elements. Furthermore, the distribution of V_n is influenced by previous frames. Despite that these properties are important for the accurate modeling of music data, they are often not well considered in the literature work. The goal of our work is to propose a flexible model that can effectively capture the inter-dependency among data elements within each vector V_n and an efficient transition structure to catch the dependency between V_n and $V_{1:n-1}$.

3. CHAIN GRAPHICAL RECURRENT NEURAL NETWORK (CGRNN)

To well represent the stochastic process of the music, we propose a deep-learning probablistic model named Chain Graphical Recurrent Neural Network (CGRNN). Built as a deep stochastic generative model whose topology is equivalent to a chain graphical model [1], CGRNN can explore the correlation of multi-dimensional data as well as track the temporal evolution of time series to improve the performance of stochastic modeling.

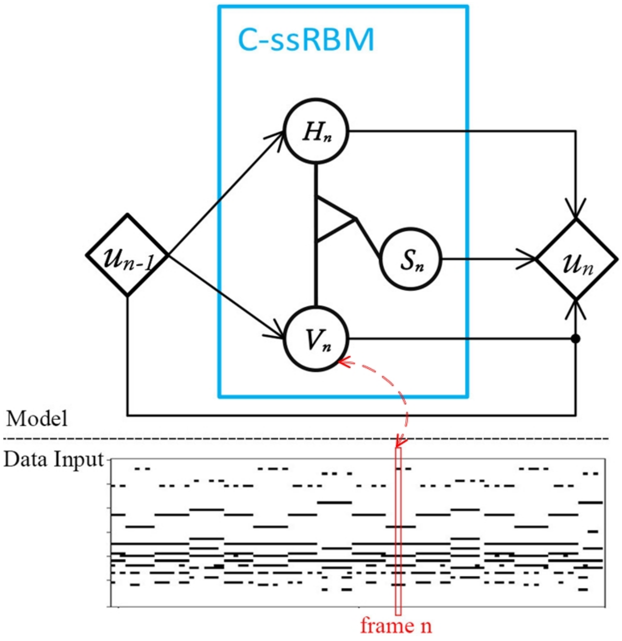

Fig. 2. Model Structure of CGRNN

The temporal unfolding of CGRNN in time slot n is shown in Fig. 2. Quadrangle nodes denote the computational function of transition and circle nodes represent random vectors. In each time slot n, there are three types of random vectors to model the specific distribution of data taken during the n_{th} time slot. V_n denotes the input frame, which is generally formed as a vector. H_n is a binary random vector applied to represent the discrete latent state of the stochastic process, while the continuous latent state vector S_n is introduced to model the covariance matrix of the input vector. H_n, S_n and V_n are connected by undirected links that indicate their probabilistic dependency. The joint distribution of the three random vectors is given by a subgraph C-ssRBM, which will be proposed later. Two types of temporal transitions are considered. One is driven by V_n, which is deterministic with a given input. The other is the probabilistic transitions H_n and S_n which are the latent states sampled from C-ssRBM. u_n denotes the nonlinear transition.

CGRNN is operated as a recurrent model. For the n_{th} frame input, transition u_{n-1} from the previous time slot is included to modulate the distributions of H_n and V_n. After that, CGRNN generates a new non-linear feedback u_n from H_n, S_n, $V_n$and u_{n-1}. In our application, we parameterize u_n as the hidden output of a Recurrent Neural Network (RNN):

$$u_n = \text{RNN}(\text{Relu}(V_n), H_n \odot S_n, u_{n-1}),$$

where Relu() denotes multiple layers of Relu units and $\odot$ is element-wise multiplication. As Relu unit is capable of generating sparse representation of input, it is essential for constructing a deep neural network. In addition, H_n in $H_n \odot S_n$ serves as a binary mask to control the effort of S_n in the temporal transition. Thus $H_n \odot S_n$ forms a sparse representation of the latent state activity and is appealing to use in deep learning models for the advantages such as information disentangling and efficient variable-size representation [8]. Therefore, we take the concatenation of $\text{Relu}(V_n)$ and $H_n \odot S_n$ as the input of RNN to improve the performance.

The overall structure of CGRNN provides an estimation of $\widehat{P}(V_{1:N})$ for a given time series. The log likelihood of the input sequence is given by:

$$\begin{aligned}
&\mathcal{L}(V_{1:N}) = \log \widehat{P}(V_{1:N}) \\
&= \log \Big\{ \sum_{H_*, S_*} \widehat{P}(V_{1:N}|H_{0:N-1}, S_{0:N-1}) \widehat{P}(H_{0:N-1}, S_{0:N-1}) \Big\} \\
&\geq \mathbb{E}_{\widehat{P}(H_{0:N-1}, S_{0:N-1})} \Big\{ \sum_{n=1}^{N} \log \widehat{P}(V_n|V_{0:n-1}, H_{0:N-1}, S_{0:N-1}) \Big\} \\
&\approx \mathbb{E}_{\widehat{P}(H_{1:N-1}, S_{1:N-1})} \Big\{ \sum_{n=1}^{N} \log \widehat{P}(V_n|u_{n-1}) \Big\}, \qquad (2)
\end{aligned}$$

where the undirected subgraph C-ssRBM is incorporated to define $\widehat{P}(V_n|u_{n-1})$. We have applied the approximation $\widehat{P}(V_n|V_{0:n-1}, H_{0:N-1}, S_{0:N-1}) \approx \widehat{P}(V_n|u_{n-1})$ and omitted the dependency between V_n and future latent states $\{H_{n+1:N-1}, S_{n+1:N-1}\}$ for two reasons. First, the information in the future is generally intractable for online applications. Second, music data are generally represented as a long sequence, thus it is time consuming to consider the dependency on the latent states in the future. Therefore, the approximation given by $\widehat{P}(V_n|u_{n-1})$ is an simple way to make CGRNN an online algorithm.

CGRNN is trained by substituting Eq. (2) into the maximizing log-likelihood problem (1). The training process is divided into two steps. In the first step, samples $\{H_{1:N-1}, S_{1:N-1}\}$ are generated according to the expected distribution $\widehat{P}(H_{1:N-1}, S_{1:N-1})$. This step is simplified with our proposed C-ssRBM. As an energy-based undirected graphical model, it provides the joint distribution of $\{V_n, H_n, S_n\}$ and the exact conditional distributions. Combining the conditional distributions with Gibbs sampling, CGRNN is able to generate the latent states that follow $\widehat{P}(H_{1:N-1}, S_{1:N-1})$ induced by the model description. In the second step of training, a temporal stack of C-ssRBMs is optimized by maximizing the likelihood function $\log \widehat{P}(V_n|u_{n-1})$ through Contrastive Divergence [9].

4. CONDITIONAL UNDIRECTED SUBGRAPH

To effectively model the distribution of the data frame in CGRNN, we propose an undirected graphical model named *conditional* spike-and-slab RBM (C-ssRBM), which takes the feedback u_{n-1} from previous time slots as the condition and applies it to modulate the mean and covariance parameters of conventional ssRBM [5]. We first describe how C-ssRBM works when inputs are continuous, and then extend it to model data with binary inputs. We further introduce our methods in efficiently evaluating CGRNN in the presence of C-ssRBM at the end of this section.

4.1. C-ssRBM for Continous Inputs

C-ssRBM is an energy-based graphical model and applied to estimate the distribution of data at the time slot n. It has undirected links among the continuous visible layer V_n, the binary latent state H_n and the continuous latent state S_n. H_n is generally called "spike" state and S_n is called "slab" state. The joint distribution of $\{V_n, H_n, S_n\}$ on the condition of the feedback u_{n-1} is given by Boltzmann distribution with the corresponding energy function defined as

$$\widehat{P}(V_n, H_n, S_n|u_{n-1}) = \frac{\exp(E(V_n, H_n, S_n|u_{n-1}))}{Z_n(u_{n-1})}, \qquad (3)$$

$$\begin{aligned}
E(V_n, H_n, S_n|u_{n-1}) &= \frac{1}{2} S_n^T \text{diag}(\alpha) S_n - V_n^T W (S_n \odot H_n) \\
&+ \frac{1}{2} V_n^T \Big(\sum_{H_{ni} \in H} \Phi_i H_{ni} + \Lambda + \text{diag}\big(U_1 u_{n-1}\big) \Big) V_n \\
&- (b_h + U_2 u_{n-1})^T H_n - (b_v + U_3 u_{n-1})^T V_n \\
&+ \alpha^T \text{diag}(\mu^2) H_n - S_n^T \text{diag}(\alpha \odot \mu) H_n, \qquad (4)
\end{aligned}$$

where $Z_n(u_{n-1})$ is called partition function. The symbols $\odot$, $(\cdot)^2$ and $\text{diag}(\cdot)$ denote respectively the element-wise multiplication, the element-wise square, and a diagonal matrix with the argument vector on its diagonal. $\{U_*, W, b_v, b_h\}$ are weight and bias parameters. $\{\Phi_*\}$ and Λ are diagonal positive matrices to insert the positive definite constraints of the conditional covariance matrices of visible layer. α and μ are the variance and mean parameters of S_n.

By marginalizing Eq. (3) with respect to the spike and slab variables, the distribution of input is given by

$$\widehat{P}(V_n|u_{n-1}) = \frac{\exp\Big(- \mathcal{F}_n(V_n|u_{n-1}) \Big)}{Z_n(u_{n-1})}, \qquad (5)$$

where the free-energy $\mathcal{F}_n(V_n|u_{n-1})$ is described as

$$\begin{aligned}
\mathcal{F}_n(V_n|u_{n-1}) &= \frac{1}{2} V_n^T \widetilde{\Lambda}_n V_n - \widetilde{b}_v^T V_n - \frac{1}{2} \sum_{\alpha_i \in \alpha} \log(2\pi\alpha_i^{-1}) \\
&- \sum \text{softplus}\Big(\frac{1}{2} \text{diag}(\alpha)^{-1} (W^T V_n)^2 - \frac{1}{2} V_n^T \{\Phi_i\} V_n \\
&+ W^T V_n \odot \mu + \widetilde{b}_h \Big).
\end{aligned}$$

Comparing Eq. (4) with the energy function of ssRBM [5], there are three additional terms. The first two terms $(U_2 u_{n-1})^T H_n$ and $(b_v + U_3 u_{n-1})^T V_n$ introduce time-variant biases for the conditional distributions of H_n and V_n. The third item $\frac{1}{2} V_n^T \cdot \text{diag}(U_1 u_{n-1}) \cdot V_n$ is applied to modulate the covariance matrix with the feedback.

By applying the Bayes law, we further obtain the conditional distributions of the visible units as

$$\widehat{P}(V_n|S_n,H_n,u_{n-1}) = \mathcal{N}\Big(C_n^{v|s,h}\Big[W(S_n\odot H_n)+\widetilde{b}_v\Big], \\ C_n^{v|s,h}\Big),$$

$$\widehat{P}(V_n|H_n,u_{n-1}) = \mathcal{N}\Big(C_n^{v|h}\Big[W(\mu\odot H)+\widetilde{b}_v\Big], C_n^{v|h}\Big),$$

The time-varying biases for input are given by $\widetilde{b}_v = b_v + U_3 u_{n-1}$. The covariance matrices corresponding to $\widehat{P}(V_n|S_n,H_n,u_{n-1})$ as well as $\widehat{P}(V_n|H_n,u_{n-1})$ are given as

$$C_n^{v|s,h} = \Big[\sum_{H_{ni}\in H}\Phi_i H_{ni}+\widetilde{\Lambda}_n\Big]^{-1},$$

$$C_n^{v|h} = \Big[(C_n^{v|s,h})^{-1} - W\text{diag}(H_n/\alpha)W^T\Big]^{-1}.$$

where $\widetilde{\Lambda}_n = \text{diag}(U_1 u_{n-1})+\Lambda$. Through $\widetilde{b}_v$ and $\widetilde{\Lambda}_n$, the feedback u_{n-1} can modulate both the mean value and covariance in the conditional distributions of input. Therefore, C-ssRBM is capable of estimating the essential statistics of V_n based on the history transition. Recall that H_n is a binary random vector and $C_n^{v|s,h}$ should be positive definite. The activation of H_n trims the covariance, which results in a non-diagonal matrix $C_n^{v|h}$. Hence, $P(V_n|H_n,u_{n-1})$ can model high-order partial correlation between elements of the input vector V_n.

4.2. C-ssRBM for Binary Inputs

The spike and slab structure also provides an efficient mechanism to utilize the conditional correlations among binary inputs [5]. Therefore, we define a binary C-ssRBM to capture the conditional dependency between different elements of the binary frame vector V_n at time n for the symbolic MIDI. The corresponding free-energy function of the binary C-ssRBM is given by

$$\mathcal{F}_n(V_n|u_{n-1}) = -\widetilde{b}_v^T V_n - \frac{1}{2}\sum_{\alpha_i\in\alpha}\log(2\pi\alpha_i^{-1}) \\ -\sum\text{softplus}\Big(W^T V_n\odot\mu+\frac{1}{2}\text{diag}(\alpha)^{-1}(W^T V_n)^2+\widetilde{b}_h\Big).$$

4.3. Evaluation of CGRNN

C-ssRBMs for the continuous and binary observations provides CGRNN with an integrated structure to capture the conditional covariance of the input vector in each time slot, which is less considered in existing stochastic neural models [3, 4, 7, 2]. However, C-ssRBM also induces the challenge of approximating the intractable partition functions $Z_n(u_{n-1})$ when evaluating the model performance. In this paper, we consider two methods for the approximation.

In the first method, we apply the anneal importance sampling (AIS) [13], a standard approximation algorithm to find the partition functions for the family of Restricted Boltzmann Machines. The second method takes the corresponding lower bound of Neural Variational Inference and Learning (NVIL) [10], which is given as

$$\log\widehat{P}(V_n|u_{n-1}) \geq -\mathcal{F}_n(V_n|u_{n-1}) \\ -\frac{1}{2}\log\ \mathbb{E}_{q_n(V_n)}\Big(\frac{q_n(Z_n|V_n)\exp(-\mathcal{F}_n)}{q_n(Z_n)q_n(V_n|Z_n)}\Big),$$

where $q_n(Z_n|V_n)$, $q_n(V_n|Z_n)$ and $q_n(Z_n)$ are defined by a variational autoencoder.

It is preferable to use AIS than the NVIL lower bound in evaluating the partition function, as the bound could be very loose if the proposal distribution $q_n(V_n)$ given by the variational autoencoder is far different from the distribution of our model [10]. However, the performance of AIS for continuous input is unstable. As sampling is done in the whole real-valued space, there is no upper bound of the generated samples if the difference between the proposal distribution of the algorithm and $\widehat{P}(V_n|u_{n-1})$ given by CGRNN is large. Therefore, we approximate $Z_n(u_{n-1})$ inside the log-likelihood by AIS for binary input but by NVIL lower bound for continuous input.

5. EXPERIMENTS ON MUSIC AND SOUNDS

In this section, we evaluate the performance of CGRNN through experiments over two music datasets, the Lakh Midi dataset [12] with over 19000 clean MIDI files and the IDMT-SMT-Audio-Effects dataset [14] consisting of 55044 WAV files of single bass and guitar notes with different audio effects. We first perform quantitative evaluations with unsupervised learning metrics, and then analyze the qualitative properties of CGRNN visually. We further apply our model to the automated music generation and discuss the advantages of our model. The implementation of the experiments can be accessed in `https://github.com/liu2231665/Project-dl4s`.

5.1. Data Preprocessing

In both datasets, the ratio of train, valid and test sets is 0.9/0.05/0.05. We only perform some simple format transformations for the raw data as follows:

$*$ **Lakh:** Each midi file is segmented by one minute. Short files with lengths less than one minute are removed. A segment is transferred into piano-rolls and forms a sequence of data frames. Each data frame is a 128-dimensional **binary vector** that indicates the activated pitches in the interval of 0.25 seconds.

$*$ **Audio Effect:** The audio file is sampled at the frequency of 11.025 kHz and reshaped as a 147-by-150 matrix with **continuous values**. Each row of the matrix is a 150-dimensional vector that represents 150 successive sampling points. The 147 rows of the whole matrix record a sequence of data frames. We also normalize the data using the global mean and standard deviation computed from the training set.

All models are evaluated in time domain without extracting spectrograms.

5.2. Stochastic Modeling

To verify the performance of CGRNN on modeling the stochastic process of music data, we compare our model with the baseline and state-of-art models on stochastic modeling: RNN [4], VRNN [4], STORN [2], SRNN [7] and RNN-RBM [3]. STORN, VRNN and SRNN are sequential variational autoencoders (VAE), which are trained by evidence lower bound (ELBO). RNN-RBM is an energy-based TRBM model, which is trained by Contrastive Divergence (CD).

We use log-likelihood and recursive prediction error as performance metrics. The log-likelihood (LL) is frequently used in unsupervised learning tasks, but exact LL is intractable except for RNN, and therefore either the approximation or the lower bound is often applied. As the comparison using only the log-likelihood is rough, we also evaluate the error of recursive prediction in these datasets. For Lakh dataset, we use the average accuracy (ACC%) of frame-wise pitch prediction [11] as the metric. For Audio Effect dataset, we use the frame-wise rooted mean square error (RMSE) between the predicted signal and the ground-true one. The results are listed in Table 1.

Table 1. Modeling and reconstruction performance of various models

	Lakh Midi		Audio Effects	
	LL	ACC%	LL	RMSE
RNN-I	-6.82	40.43	308.72	5.14
RNN-II	-14.12	8.98	298.85	4.85
STORN	≥ -7.06	39.58	≥ 253.24	5.61
VRNN	≥ -3.53	81.83	≥ 456.18	1.80
SRNN-s	≥ -3.41	79.01	≥ 323.10	4.16
SRNN-s+res	≥ -4.72	63.85	≥ 398.56	4.08
SRNN-f	≥ -3.29	81.37	≥ 411.44	2.29
RNN-RBM	≈ -6.09	60.83	≈ -141.8	3.27
RNN-ssRBM	≈ -5.76	86.28	≥ 310.42	1.71
CGRNN	$\geq \mathbf{-2.70}$	**87.88**	$\geq \mathbf{503.80}$	**0.50**

Our model has the best unsupervised learning performances in both the datasets. Compared with the second best model, CGRNN increases LL by $\mathbf{18}\%$ and accuracy by $\mathbf{2}\%$ in the midi data. For the audio data, the improvement is significant. RMSE is reduced from 1.71 to 0.50. The approximated lower bound of LL is increased by $\mathbf{10}\%$.

5.3. Qualitative Evaluations

We further study other features of CGRNN, including the patterns of hidden activations, the learned conditional covariances and the quality of input reconstruction from hidden activations. These properties are hard to evaluate quantitatively. Therefore, we provide the visualization of some representative samples. As the performances of RNN, RNN-RBM and STORN are relatively worse, we focus the comparison on VRNN, SRNN-s, RNN-ssRBM and CGRNN.

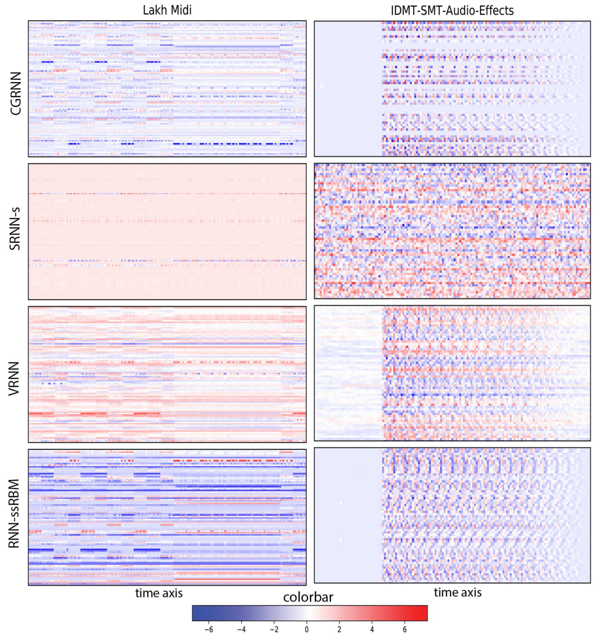

Fig. 3. The hidden activations of various models. The horizon is time axis and the vertical axis is the latent state vector.

Hidden activations: The hidden activation can reflect how well the stochastic model captures the temporal evolution of data sequence. It can also be applied as a feature input for other deep learning models. For VRNN and SRNN-s which have merely continuous latent states, we define the hidden activation as the conditional mean of the latent states with given inputs. For CGRNN and RNN-ssRBM, $H_n \odot S_n$ could produce a sparse representation. Therefore, we use $\mathbb{E}(H_n \odot S_n|V_n, u_{n-1})$ as the hidden activation. According to Fig. 3, the hidden activation of SRNN-s is chaotic and CGRNN has the most sparse activations. $\mathbb{E}(Z_n|V_n, u_{n-1})$ of VRNN for the audio wave has a noisy fluctuation at the beginning, when the corresponding part of the audio wave is blank.

Conditional Precisions: The covariance matrix $C_n^{v|h}$ and its inverse precision matrix $(C_n^{v|h})^{-1}$ of CGRNN provide a method to model the high-order partial correlation among data. The precision matrices at different time slot n of the CGRNN and RNN-ssRBM are displayed in Fig. 4. Without the efficient transition structure, the precision matrix of RNN-ssRBM consistently degenerates into simple diagonal matrices. In contrast, CGRNN effectively captures the evolution of the correlation over time. At the time slot 0, the matrix is diagonal. At the later time slots, the wider diagonal lines reflect the correlation between adjacent elements within each frame.

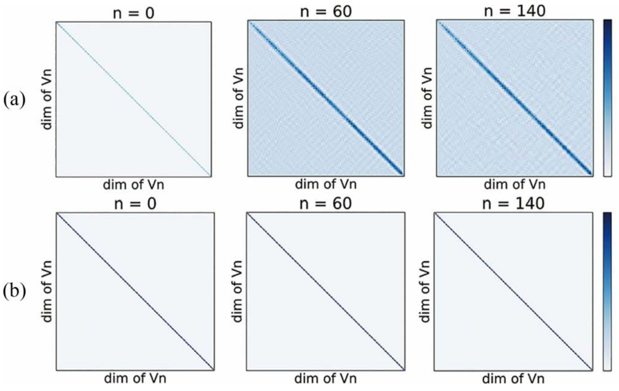

Fig. 4. (a) The conditional precision matrices of CGRNN. (b) The conditional precision matrices of RNN-ssRBM.

Reconstruction of samples: The reconstruction quality from the hidden activation to data input is essential for a deep generative model. All the four models are able to reconstruct the main melodies of a given piano-rolls from the hidden activations, with some noise in the jumpy pitches. For reconstructing the audio waves of the guitar and bass sound, only CGRNN is able to generate an accurate sound with an acceptable level of noise. The reconstructions of VRNN, SRNN and RNN-ssRBM are submerged into noises.

5.4. Music and Sound Generation

CGRNN can benefit a broad range of applications in multimedia. As another example, we implement CGRNN and train the model with the Lakh dataset for automated music generation. This is one of the applications of AI in the musicology.

Perceptually, CGRNN is capable of generating melodious polyphonic music without human knowledge. The generated music consists of abundant chords and rhythms, which is not observed from existing methods [6, 3]. Unlike the recent work [6] that seeks to generate multi-instrument music, we focus on the generation of single-track music with higher quality and large variation of rhythm. Nevertheless, CGRNN can also provide a more accurate temporal model for existing methods to generate better multi-track songs.

6. CONCLUSION

In this paper, we propose a deep learning model called CGRNN for more accurate stochastic modeling of music. The experiments using two challenging datasets demonstrate that CGRNN can achieve a significant improvement on music modeling over multiple state-of-art stochastic models. By learning the stochastic process of music represented as symbolic MIDI and audio signal, CGRNN can be applied to a broad range of real-world applications on music information retrieval. We will investigate these in our future study.

7. REFERENCES

[1] D. Barber. Graphical models. In *Bayesian Reasoning and Machine Learning*, chapter 4, pages 58–76. 2012.

[2] J. Bayer and C. Osendorfer. Learning stochastic recurrent networks. *arXiv preprint arXiv:1411.7610*, 2014.

[3] N. Boulanger-Lewandowski, Y. Bengio, and P. Vincent. Modeling temporal dependencies in high-dimensional sequences: Application to polyphonic music generation and transcription. In *ICML*, pages 1159–1166, 2012.

[4] J. Chung, K. Kastner, L. Dinh, K. Goel, A. Courville, and Y. Bengio. A recurrent latent variable model for sequential data. In *NIPS*, pages 2980–2988, 2015.

[5] A. Courville, G. Desjardins, J. Bergstra, and Y. Bengio. The spike-and-slab RBM and extensions to discrete and sparse data distributions. *IEEE Trans. Pattern Anal. Mach. Intell.*, 36(9):1874–1887, 2014.

[6] H. Dong, W. Hsiao, L. Yang, and Y. Yang. Musegan: Multi-track sequential generative adversarial networks for symbolic music generation and accompaniment, 2018.

[7] M. Fraccaro, S. Sønderby, U. Paquet, and O. Winther. Sequential neural models with stochastic layers. In *NIPS*, pages 2199–2207, 2016.

[8] X. Glorot, A. Bordes, and Y. Bengio. Deep sparse rectifier neural networks. In *AISTATS*, pages 315–323, 2011.

[9] G. Hinton. A practical guide to training restricted boltzmann machines. In *Neural networks: Tricks of the trade*, pages 599–619. 2012.

[10] V. Kuleshov and S. Ermon. Neural variational inference and learning in undirected graphical models. In *NIPS*, pages 6737–6746. 2017.

[11] G. E. Poliner and D. P. W. Ellis. A discriminative model for polyphonic piano transcription. *EURASIP Journal on Advances in Signal Processing*, 2006.

[12] C. Raffel. Learning-based methods for comparing sequences, with applications to audio-to-midi alignment and matching. PhD Thesis. 2016.

[13] R. Salakhutdinov and I. Murray. On the quantitative analysis of deep belief networks. In *ICML*, pages 872–879, 2008.

[14] M. Stein, J. Abeßer, C. Dittmar, and G. Schuller. Automatic detection of audio effects in guitar and bass recordings. In *Audio Engineering Society Convention 128*, 2010.