

MODELING MELODIC FEATURE DEPENDENCY WITH MODULARIZED VARIATIONAL AUTO-ENCODER

Yu-An Wang* Yu-Kai Huang* Tzu-Chuan Lin* Shang-Yu Su Yun-Nung Chen

National Taiwan University, Taipei, Taiwan

{b04902004, b04902131, b04705003, f05921117}@csie.ntu.edu.tw y.v.chen@ieee.org

ABSTRACT

Automatic melody generation has been a long-time aspiration for both AI researchers and musicians. However, learning to generate euphonious melodies has turned out to be highly challenging. This paper introduces 1) a new variant of variational autoencoder (VAE), where the model structure is designed in a modularized manner in order to model polyphonic and dynamic music with domain knowledge, and 2) a hierarchical encoding/decoding strategy, which explicitly models the dependency between melodic features. The proposed framework is capable of generating distinct melodies that sounds natural, and the experiments for evaluating generated music clips show that the proposed model outperforms the baselines in human evaluation.¹

Index Terms— Music Generation, VAE, Modularization

1. INTRODUCTION

Recently, in algorithmic music generation field, there are two main-streams: *symbolic-domain* and *audio-domain*. In symbolic-domain, the target is to generate music in standard MIDI format [1, 2, 3, 4, 5, 6, 7]. On the other hand, in audio-domain, the goal is to synthesize music waveform [8, 9, 10]. In this paper, we focus on *symbolic-domain* music generation.

Generating music is different from generating other modalities like images or natural languages. To create harmonic music, the order and combination of music elements in the temporal scale are of significant importance. There are two main modeling directions for music generation, one uses sequence modeling such as recurrent neural networks (RNN) [1, 5, 6, 11, 7], and another uses generative modeling such as variational autoencoders (VAE) [12] and generative adversarial networks (GAN) [13, 3]. To model the sequence-like attributes of music elements, we utilize variational auto-encoders (VAE) to generate polyphonic and dynamic music. Further extension of MusicVAE - Multitrack MusicVAE [14] and SeqGAN [15] was proposed to focus on multi-track music.

In this work, we utilized an integrated model, variational recurrent auto-encoders (VRAE) [16], incorporating the benefits of modeling long-term dependencies by recurrent units and generative nature of VAE, and further introduce a novel generative model for music generation. First, we propose the architecture of encoder so as to encode more information into the latent code. With domain knowledge, we modularize the encoder into two parts: the first part mainly focuses on the rhythm and the pitch of a note, while the second part

gathers information and puts the essence of the first part into contexts. Moreover, we use a hierarchical and recurrent approach, note unrolling, to model the dependency of music notes. This is the first work applying note unrolling in VAE model, and we find that it is suitable for the decoder to explicitly model the dependency of time, duration and pitch of music. Our main contributions are as follows:

- The proposed model incorporates domain knowledge of music by using a modularized framework for modeling various melodic features.
- This is the first work that well integrates the note-unrolling technique in VAE to model the dependency between melodic features for music generation.
- The proposed model is capable of generating natural music from the human perspective and achieves better performance than other generative models.

2. PROPOSED MODEL

In the proposed model, the gated recurrent unit (GRU) [17] is applied to formulate variational recurrent auto-encoders (VRAE) [16], considering its balance between performance and model size [18]. Given a datapoint x which depends on an unobserved latent variable z , the training object of VAE is evidence lower bound objective (ELBO),

$$\mathcal{L}_{\text{ELBO}}(x) = \mathbb{E}_{q_{\phi}(z|x)}[\log p_{\theta}(x|z)] - D_{KL}(q_{\phi}(z|x) \parallel p_{\theta}(z)).$$

The first term of ELBO can be viewed as the reconstruction loss, while the second one is the regularization term, which is the Kullback-Leibler (KL) divergence between the amortized inference distribution and the prior $p_{\theta}(z)$. The choice of these distribution is often a factorized Gaussian by its simplicity and computational efficiency. This work utilizes the normal distribution with a diagonal covariance matrix, $p_{\theta}(z) = \mathcal{N}(0, I)$. The whole objective can be optimized by gradient-based methods and reparametrization tricks with respect to the model parameters ϕ and θ .

The encoder first encodes an input sequence $\mathbf{x} = (x_1, \dots, x_T)$ as a normal distribution $\mathcal{N}(z | \mu(x, \phi), \Sigma(x, \phi))$. Then the RNN decoder generates an output sequence given the sampled latent vector z from the normal distribution of the encoder. Different from previous described, VRAE can effectively learn to represent an output sequence due to the objective constraints on the KL divergence. Therefore, VRAE is able to embed richer semantic information into the latent space Z than traditional sequence modeling approaches. This paper builds a music-generation framework on top of VRAE and allows us to incorporate domain knowledge for the target task by the modular components.

The proposed framework integrates the advantages from the prior work and proposes a novel model with better flexibility and

¹The first three authors have equal contributions.

¹The source code is available at https://github.com/MiuLab/MVAE_music.

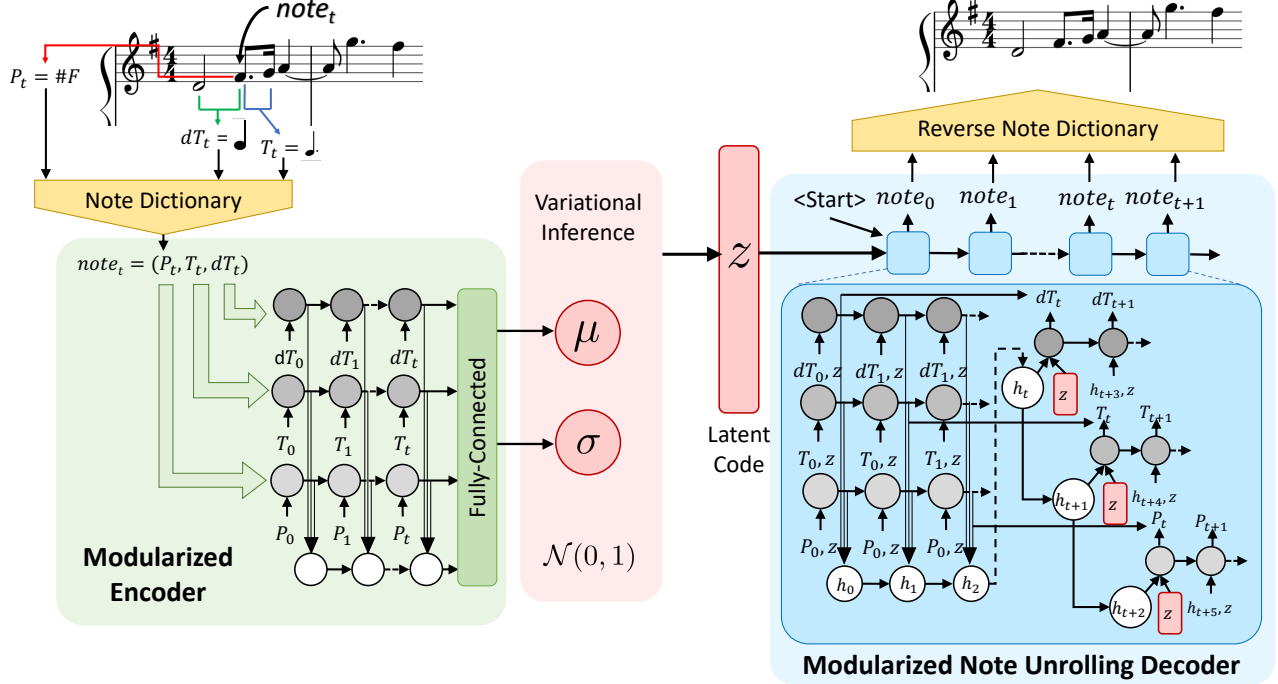


Fig. 1. The illustration of the proposed modularized framework, where the VAE architecture is embedded with a modularized encoder and a note-unrolling decoder, and the note event representations are realized by a note dictionary and a reverse note dictionary.

performance, which is illustrated in Figure 1. Considering that BachProp [19] used the normalized note representation of music, we use similar note representation, called note event representation, which is capable of translating music with minimal distortion.

In the previous work, MusicVAE [14] utilized VAE models to generate music. Moreover, MusicVAE applied a hierarchical decoder on each measure so that MusicVAE can learn the long-term structure of music. However, the hierarchical framework was limited to generate only 4/4 time music piece due to its decoder’s structure. In contrast, our model is not constrained to generate specific time signature music by maintaining the structure between notes by note unrolling, demonstrating better flexibility and practical usage.

2.1. Data Representation

In order to represent more dynamic and complex music, our proposed model use the *note event* originally presented in BachProp[19] as a basic unit of music, which can separate into three attributes:

$$\{\mathbf{note}_t = (dT_t, T_t, P_t)\}, \quad (1)$$

where t indicates t -th note event, dT represents starting time difference of note events between \mathbf{note}_{t-1} and \mathbf{note}_t , and T, P represent the duration and pitch of one note event, respectively. By setting the value of $dT_t = 0$, the note \mathbf{note}_t and the previous note \mathbf{note}_{t-1} can be pressed at the same time, producing a polyphonic structure like chords and mixed chorus. The top-left part of Figure 1 illustrates the note dictionary.

Previous work, such as MusicVAE [14] and DeepBach [20], discretized time into sixteenth notes. In contrast, note event representation allows the model to learn beats which are not multiples of sixteenth notes and have the freedom of pressing multiple notes at same time to generate polyphonic music. The attributes are indexed by the appearance order in dataset.

2.2. Shared Embedding

From the human knowledge of music theory, we know different beats and pitches have certain relationships. For example, C, G and B, #F are in C and B major, respectively. A triplet quarter note with duration $\frac{1}{6}$ and three sixteenth notes with duration $\frac{3}{16}$ have close duration value. In this work, we project the discrete representation of each melodic feature into an embedding space to model these relationships before feeding into the encoder and the decoder. Empirically, using shared embeddings for each feature in the encoder and the decoder can reduce parameters while having same performance.

2.3. Modularized Encoder

Naïvely, we can use only one large GRU taking the concatenation of dT_t, T_t and P_t ’s embedding vector as input. However, rhythm and pitch are different category of attributes in music theory, which means it is possible that once encoder gets rhythm mixed up with pitch-related information, it could not understand this music piece well. As a result, we propose a modularized encoder encouraging them to focus on encoding different melodic features separately and extracting cleaner information.

The proposed modularized encoder is illustrated in Figure 1, which consists of four GRUs. The first three GRU encoders extract respective messages from different melodic features: time(dT), duration(T) and pitch(P), without inter-connections among one another. To combine the extracted information from these encoders, the context GRU takes output vectors from the above three encoders at each time step as input and integrates information. Further, we concatenate the the latent vectors at the last step from all four GRU encoder modules and pass it through fully-connected layers to get the variational parameters of the distribution over latents, which is the mean μ and standard deviation σ .

Formally, the proposed modularized encoder can be represented

as the following mathematical forms:

$$\begin{aligned} h_t^{dT} &= \text{GRU}_{dT}(h_{t-1}^{dT}, dT_t), \\ h_t^T &= \text{GRU}_T(h_{t-1}^{dT}, T_t), \\ h_t^P &= \text{GRU}_P(h_{t-1}^P, P_t), \\ h_t^C &= \text{GRU}_C(h_{t-1}^C, [h_t^{dT}; h_t^T; h_t^P]), \end{aligned}$$

where GRU_C is the context GRU that gathers information from the lower-level modules GRU_{dT} , GRU_T and GRU_P . Finally, we have μ and σ by transformation with a few linear layers:

$$\begin{aligned} v &= W_1[h_T^{dT}; h_T^T; h_T^P; h_T^C] + b_1, \\ \mu &= W_\mu v + b_\mu, \\ \sigma &= \exp\left(\frac{1}{2}(W_\sigma v + b_\sigma)\right). \end{aligned} \quad (2)$$

2.4. Modularized Note-Unrolling Decoder

In the field of music theory, rhythm, modes and tone are combined with dependency to create a melody. Rhythm stands for time and duration to some degree, while the lowness and highness of a tone represents pitch. All attributes of a note for a chord ($[dT, T, P]$) are combined with some relation to fit that chord. Considering the nature of music composition described above, we should model the dependency between attributes of chords explicitly while decoding.

The previous work [19] models the dependency relations between dT , T and P by decomposing the joint probability of three attributes in a note event into a product of conditional probabilities, which can be written as:

$$\begin{aligned} p(dT_t, T_t, P_t | \text{note}_{1:t-1}) &= p(dT_t | \text{note}_{1:t-1}) \\ &\times p(T_t | dT_t, \text{note}_{1:t-1}) \\ &\times p(P_t | dT_t, T_t, \text{note}_{1:t-1}). \end{aligned} \quad (3)$$

The design of note unrolling follows music domain knowledge, where the note attributes are often conditioned on other attributes. Figure 1 illustrates the concept of note unrolling and the design of the proposed hierarchical decoder, which contains total 7 GRUs: three for modeling attribute-specific contexts, one for combining multiple attributes as a contextual module, and three for generating associated note attributes. Furthermore, we utilize residual skip connection from upper-level modules to lower-level modules, since our modularized decoder have many GRUs, skip connection can avoid gradient vanishing caused by back propagation through too many layers. The hierarchical decoder separates the generation procedure into subsequent three process: time, duration and pitch; it is taught to output dT_t depending on previous notes $\text{note}_{1:t-1}$. After that, the second step depends on dT_t and $\text{note}_{1:t-1}$ to generate T_t . Finally, the network depends on dT_t , T_t and $\text{note}_{1:t-1}$ to generate P_t .

2.5. Training and Generation

We optimize our model by RMSprop optimizer with learning rate 10^{-4} and a batch size of 128. Due to strong autoregressive characteristics in RNN, VRAE tends to ignore the latent distribution, so-called *posterior collapse* issue. To mitigate this problem, KL annealing is applied [21] to allow the model to encode more information into the latent code \mathbf{z} at first and then gradually fit the prior as the weight approached 1.

In the inference stage, we sample k -dimension latent code \mathbf{z} from the standard normal distribution $\mathcal{N}(0, I_k)$ as input of the decoder, where k indicates the dimension of latent vector. Then, we can generate music pieces with various length of notes by the proposed hierarchical decoder with all the techniques mentioned above.

3. EXPERIMENTS

3.1. Setup

The experiments are performed on three diverse benchmark datasets: Nottingham, Piano-midi.de and JSB Chorales, which are frequently used for music generation [11, 19]. Nottingham dataset contains 1037 midi files of folk music; Piano-midi.de dataset contains 333 midi files of classical music; JSB Chorales dataset contains 382 midi files of chorales of J.S Bach. To better validate the capability of modeling the music diversity in our generative model, we merge three into one large dataset for training. The hidden size of all GRUs in our model is set to 512. Considering the difficulty of modeling long songs in the dataset, the midi files are cut into 100 note segments (about 15 seconds) with stride equal to 50. Then we randomly rearrange tonality of each segment by $[-3, +3]$ for data augmentation.

3.2. Baseline

We compare our proposed model two baselines, BachProp [19] and modularized autoencoder.

- BachProp: the model similar to our proposed model without an encoder and variational approximation on latent distribution, hence the comparison can highlight the importance of our modularized encoder.
- Modularized autoencoder: the model similar to our model without the objective of KL divergence constraint on the latent distribution z , which could show the importance of the variational inference.

Note that we do not compare with MusicVAE [14], because it cannot handle the music that is not 4/4 time signature contained by our dataset, making the comparison inapplicable. Another advantage of our model is the flexibility of modeling dynamic music with different time signatures compared with MusicVAE.

3.3. Human Evaluation

To measure the performance of the proposed model, we conduct human evaluation, the procedure is designed as following. First, the recruited raters are asked about their level of expertise. Second, the raters are requested to give the Likert scale scores from 1 to 6 to measure whether the music is human-composed (higher score) or machine-generated (lower score) for each given 100-note midi file. Finally, We collect 85 scores for each model.

3.4. Results

The human evaluation results are shown in Table 1, where we perform the significance test to validate the improvement. The improvement achieved by our model compared to BachProp and modularized autoencoder is both statistically significant using the single-tailed t-test with $\alpha < 0.01$ marked as \dagger .

It tells that our model is capable of generating music according to the designated encoded \mathbf{z} and, thus, has more flexibility and diversity. In contrast with BachProp, it has no information from the latent code \mathbf{z} and cannot maintain a consistent structure in an output sequence. Thus, its scores are much lower than our proposed model

Table 1. Experimental results of reconstruction loss (**Rec.**), KL divergence loss (**KL**) for each model with different settings.

| Model | Rec. | KL | Human Score | |
|-------------------------|--------|--------|--------------------|----------|
| | | | μ | σ |
| BachProp [19] | 240.16 | — | 3.51 | 1.61 |
| Modularized autoencoder | 20.79 | — | 2.77 | 1.65 |
| <i>Proposed model</i> | | | | |
| w/o note unrolling | 85.88 | 264.00 | 3.22 [†] | 1.73 |
| w/ note unrolling | 73.19 | 30.37 | 4.24 ^{†‡} | 1.54 |
| Real Data | — | — | 4.34 | 1.55 |

by a large margin. When comparing with modularized autoencoders, it hard codes \mathbf{z} for each song in the dataset on the latent space. However, without KL loss during training, the decoder cannot model the meaning of a randomly sampled \mathbf{z} at the inference phase. The human evaluation results show that our proposed model can obtain an informative latent code \mathbf{z} using VAE and outperform other baseline generative models. The achieved performance is close to the real data scores, which are the upperbound of our music generation model.

3.5. Effectiveness of Note-Unrolling Decoder

The note-enrolling mechanism was first proposed by BachProp on the sequence prediction model, and they claimed that there is dependency between dT, T and P for one note according to human knowledge [19]. However, they did not illustrate or analyze how note unrolling is better than original approaches. Therefore, we perform an ablation test to explicitly verify whether the note unrolling decoder brings the benefit on generating pleasant music pieces. From Table 1, adding note unrolling can decrease the reconstruction loss and significantly improve the human scores. The difference between their results is significant with $\alpha < 0.01$ marked as [‡], demonstrating the effectiveness of modeling music structures based on music domain knowledge in our proposed model.

3.6. Analysis of Latent Space

Our dataset is composed of multiple music characteristics including folk music, classical music, and chorales of J.S Bach. In order to check 1) whether our model can learn diverse characteristics in the same latent space and 2) whether our model can capture the difference about characteristics and encode into the informative codes, we perform the following analysis.

3.6.1. Interpolation Distribution

To further analyze whether our proposed approach is capable of modeling diverse music characteristics, we sample two datapoints, A and B, compute the interpolation points between them using their latent codes, and further check whether our model can smoothly model their distribution. We verify the smoothness of the latent space distribution by computing Hamming distance between every interpolation points and datapoints. Undoubtedly, we find that as the interpolation point goes from data point A to data point B, its Hamming distance to A starts increasing but decreasing in distance to B. Note that we only show the Hamming distance to A due to the tendency of two curves is almost symmetric.

The results are shown in Figure 2, where our proposed model has a more smooth curve than the baseline autoencoder, implying

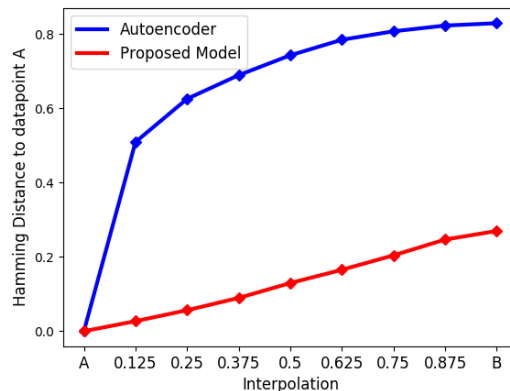


Fig. 2. Average distance between two random datapoints on Z .

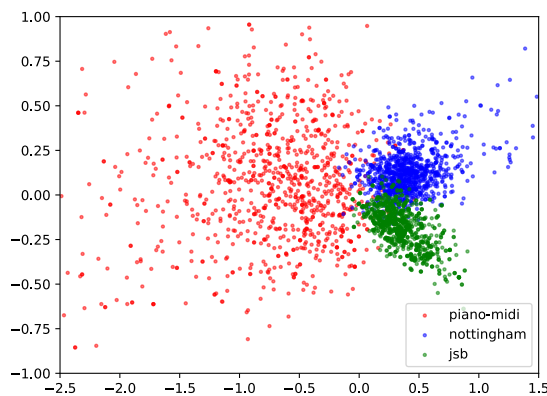


Fig. 3. Visualization on the latent space via PCA, where three different types of music are separated in Z .

that the interpolation points between two data points are meaningful. To sum up, our proposed model can handle the diverse inputs by constraining the distribution using KL, while the autoencoder models diverse characteristics in multiple separate spaces.

3.6.2. Visualization

To further analyze whether our model can effectively capture the distinct features among different characteristics in music, we project the learned latent codes of different datasets into 2-dimensions by PCA. The results are shown in Figure 3, where \mathbf{z} of different type of music are separated in the latent space. The result demonstrates that our encoder can surely encode music into an informative latent code.

4. CONCLUSION

This paper presents a VAE model that incorporates a modularized encoder and a modularized decoder in the framework to better generate realistic melodies. The modularized encoder is capable of encoding the latent information, and a note unrolling decoder models the melodic dependency between note attributes. Also, the proposed note event representations bring the better flexibility. The experiments are conducted in a merged dataset with diverse characteristics in music, demonstrating the superior performance of our proposed model for all evaluation scenarios: human evaluation and latent space analysis.

5. REFERENCES

- [1] Ian Simon and Sageev Oore, “Performance rnn: Generating music with expressive timing and dynamics,” 2017.
- [2] Ian Simon, Adam Roberts, Colin Raffel, Jesse Engel, Curtis Hawthorne, and Douglas Eck, “Learning a latent space of multitrack measures,” *arXiv preprint arXiv:1806.00195*, 2018.
- [3] Li-Chia Yang, Szu-Yu Chou, and Yi-Hsuan Yang, “Midinet: A convolutional generative adversarial network for symbolic-domain music generation,” *arXiv preprint arXiv:1703.10847*, 2017.
- [4] Hao-Wen Dong and Yi-Hsuan Yang, “Convolutional generative adversarial networks with binary neurons for polyphonic music generation,” *arXiv preprint arXiv:1804.09399*, 2018.
- [5] Gino Brunner, Yuyi Wang, Roger Wattenhofer, and Jonas Wiesendanger, “Jambot: Music theory aware chord based generation of polyphonic music with lstms,” in *Tools with Artificial Intelligence (ICTAI), 2017 IEEE 29th International Conference on*. IEEE, 2017, pp. 519–526.
- [6] Huanru Henry Mao, Taylor Shin, and Garrison Cottrell, “Deepj: Style-specific music generation,” in *Semantic Computing (ICSC), 2018 IEEE 12th International Conference on*. IEEE, 2018, pp. 377–382.
- [7] Daniel D Johnson, “Generating polyphonic music using tied parallel networks,” in *International Conference on Evolutionary and Biologically Inspired Music and Art*. Springer, 2017, pp. 128–143.
- [8] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu, “Wavenet: A generative model for raw audio,” *arXiv preprint arXiv:1609.03499*, 2016.
- [9] Jesse Engel, Cinjon Resnick, Adam Roberts, Sander Dieleman, Douglas Eck, Karen Simonyan, and Mohammad Norouzi, “Neural audio synthesis of musical notes with wavenet autoencoders,” *arXiv preprint arXiv:1704.01279*, 2017.
- [10] Soroush Mehri, Kundan Kumar, Ishaan Gulrajani, Rithesh Kumar, Shubham Jain, Jose Sotelo, Aaron Courville, and Yoshua Bengio, “SAMPLERNN: An unconditional end-to-end neural audio generation model,” *arXiv preprint arXiv:1612.07837*, 2016.
- [11] Nicolas Boulanger-Lewandowski, Yoshua Bengio, and Pascal Vincent, “Modeling temporal dependencies in high-dimensional sequences: Application to polyphonic music generation and transcription,” *arXiv preprint arXiv:1206.6392*, 2012.
- [12] Diederik P Kingma and Max Welling, “Auto-encoding variational bayes,” *arXiv preprint arXiv:1312.6114*, 2013.
- [13] Hao-Wen Dong, Wen-Yi Hsiao, Li-Chia Yang, and Yi-Hsuan Yang, “Musegan: Symbolic-domain music generation and accompaniment with multi-track sequential generative adversarial networks,” *arXiv preprint arXiv:1709.06298*, 2017.
- [14] Adam Roberts, Jesse Engel, Colin Raffel, Curtis Hawthorne, and Douglas Eck, “A hierarchical latent vector model for learning long-term structure in music,” *arXiv preprint arXiv:1803.05428*, 2018.
- [15] Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu, “Seqgan: Sequence generative adversarial nets with policy gradient,” in *AAAI*, 2017, pp. 2852–2858.
- [16] Otto Fabius and Joost R van Amersfoort, “Variational recurrent auto-encoders,” *arXiv preprint arXiv:1412.6581*, 2014.
- [17] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio, “Learning phrase representations using rnn encoder-decoder for statistical machine translation,” *arXiv preprint arXiv:1406.1078*, 2014.
- [18] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio, “Empirical evaluation of gated recurrent neural networks on sequence modeling,” *arXiv preprint arXiv:1412.3555*, 2014.
- [19] Florian Colombo and Wulfram Gerstner, “Bachprop: Learning to compose music in multiple styles.,” *CoRR*, 2018.
- [20] Gaëtan Hadjeres, François Pachet, and Frank Nielsen, “Deepbach: a steerable model for bach chorales generation,” *arXiv preprint arXiv:1612.01010*, 2016.
- [21] Samuel R Bowman, Luke Vilnis, Oriol Vinyals, Andrew M Dai, Rafal Jozefowicz, and Samy Bengio, “Generating sentences from a continuous space,” *arXiv preprint arXiv:1511.06349*, 2015.