Neural Computing and Applications (2023) 35:3191–3202
https://doi.org/10.1007/s00521-022-07863-5

ORIGINAL ARTICLE

Conditional hybrid GAN for melody generation from lyrics

Yi Yu[1] · Zhe Zhang[1] · Wei Duan[1] · Abhishek Srivastava[2] · Rajiv Shah[2] · Yi Ren[3]

Received: 6 March 2022 / Accepted: 21 September 2022 / Published online: 8 October 2022

Abstract
Conditional sequence generation aims to instruct the generation procedure by conditioning the model with additional context information, which is an interesting research issue in AI and machine learning. Unfortunately, current state-of-the-art generative models for music fail to generate good melodies due to the discrete-valued property of music attributes. In this paper, we propose a novel conditional hybrid GAN (C-Hybrid-GAN) for melody generation from lyrics. Three discrete sequences corresponding to music attributes, namely pitch, duration, and rest, are separately generated by melody generation model conditioned on the same lyrics. Gumbel-Softmax is used to approximate the distribution of discrete-valued samples so as to directly generate discrete melody attributes. Most importantly, a hybrid structure is proposed, which contains three independent branches (each for one melody attribute) in the generator and one branch for distinguishing concatenated attributes in the discriminator. Relational memory core is exploited to model not only the dependency inside each sequence of attribute during the training of the generator, but also the consistency among three sequences of attributes during the training of the discriminator. Through extensive experiments using evaluation metrics, e.g., maximum mean discrepancy, average rest value, and MIDI number transition, we demonstrate that the proposed C-Hybrid-GAN outperforms the existing methods in melody generation from lyrics.

Keywords Melody generation from lyrics · GAN · AI music · Conditional sequence generation

1 Introduction

Generating melody from lyrics to compose a song has been a challenging research task in the field of artificial intelligence and music, which falls under the field of conditional discrete-valued sequence generation. This generation aims to develop generative models that can automatically predict melody with given lyrics, to match the desired lyricism, structure, and diversity in a way similar to music creativity of human. An earlier study by [1] has shown the feasibility of exploiting conditional long short-term memory—generative adversarial network (LSTM-GAN) for melody generation from lyrics. Although this state-of-the-art lyrics-conditioned melody generation method has demonstrated promising results compared with the traditional maximum likelihood estimation (MLE) method, it has the limitation in generating more plausible melodies due to failing to accurately model the discrete music attributes. On the one hand, the continuous-valued sequence, as the output of the generator in the GAN, is not in accordance with the discrete-valued music attributes. On the other hand, due to the quantization error, the generated music attributes could be associated with an improper discrete-valued music attribute, which would lead to a negative impact on melody generation.

To overcome the aforementioned disadvantage, in this work, the Gumbel-Softmax [2] is exploited to approximate the distribution of discrete-valued sequences. On this basis, a novel conditional hybrid generative adversarial network (C-Hybrid-GAN) is suggested to generate melodies from lyrics, where three discrete sequences of music attributes are separately generated by the melody generation model conditioned on the same lyrics. In particular, this paper contains several contributions: (i) A hybrid structure is proposed, which contains three independent branches (each

✉ Yi Yu
yiyu@nii.ac.jp

[1] Digital Content and Media Sciences Research Division, National Institute of Informatics and SOKENDAI, Chiyoda-ku, Tokyo 101-8430, Japan

[2] Indian Institute of Technology Delhi, Delhi 110016, India

[3] Zhejiang University, Hangzhou 310027, Zhejiang, China

for one melody attribute) in the generator and one branch for distinguishing concatenated attributes in the discriminator. (ii) The relational reasoning technique is exploited to model the dependency inside each sequence of music attribute during the training of the generator as well as the consistency among three sequences of music attributes during the training of the discriminator. (iii) This is the first work for discussing how to learn a model for discrete-valued sequence generation with multiple attributes by considering reasoning technique and Gumbel-Softmax in artificial intelligence (AI) music. Through the extensive experiments, we show that the proposed C-Hybrid-GAN outperforms the existing melody generation methods and has the capability of generating more natural and plausible melodies.

2 Related work

In the previous works, we have witnessed various methods to solve music generation such as autoregressive-based approach in [3]. The focus of this work is how to tackle melody generation from lyrics that is regarded as conditional discrete-valued sequence generation, aiming to imitate human creation of melody sequence conditioned on the same lyrics. Consequently, in this work, we mainly discuss melody generation from lyrics and GAN-based discrete sequence generation.

2.1 Melody generation from lyrics

Song composition from lyrics in [4] aimed to generate a melody when given Japanese lyrics, patterns of music rhythms, and harmony sequences. Some constraints are defined to associate syllables with notes. Melody generation is implemented by dynamic programming. The rhythmic patterns occurring in notes can be classified in [5]. Pitches most suitable for accompanying the lyrics are generated using n-gram models. Three stylistic categories such as nursery rhymes, folk songs, and rock songs are composed when given lyrics. ALYSIA songwriting in [6] is a lyrics-conditioned melody composition system based on a random forest model, which can model the pitch and rhythm of notes to determine the accompaniments for lyrics. When given Chinese lyrics, melody and exact alignment are predicted in a lyrics-conditioned melody composition system by [7], which is an end-to-end neural network model including RNN-based lyrics encoder, RNN-based context melody encoder, and a hierarchical RNN decoder. Large-scale Chinese language lyrics-melody dataset was built to evaluate the proposed learning model. In our initial work by [1], we not only built a large dataset consisting of 12,197 MIDI songs each with paired lyrics and melody, but also have verified the feasibility of melody generation from lyrics by LSTM-based deep generative model [8]. Moreover, some baselines and evaluation methods were established. Continuous-valued sequence is generated from the generator and quantized to the underlying representation of discrete-valued melody attributes (pitch, duration, rest). Lyrics2Song in [9] is a smartphone application, which can generate the melody when given the lyrics by exploiting an existing method of singing voice synthesis based on deep neural networks. SongMASS in [10] exploited the masked sequence to sequence (MASS) pre-training and attention-based alignment modeling for lyrics-to-melody and melody-to-lyrics generation.

2.2 GAN-based discrete sequence generation

Generative adversarial networks (GANs) in [11] were originally developed to generate continuous data [12], which have been applied successfully in the conditional sequence generation such as dialogue in [13], text-to-video in [14], and lyrics-to-melody in [1] generation. However, GANs have the limitation in generating discrete sequence due to the non-differentiable problem of the discrete-valued outputs from the generator. To overcome this disadvantage, existing works pay attention to two research lines: (i) policy gradient based on reinforcement learning and (ii) continuous approximation of the discrete distribution.

(i) Policy gradient based on reinforcement learning. SeqGAN by [15] models the generator as a stochastic policy in reinforcement learning, which avoids the generator differentiation problem by directly performing policy-gradient updates. RankGAN by [16] uses the ranking score as the rewards to learn the generator, which is optimized through the policy gradient method. LeakGAN by [17] addresses a mechanism of providing richer information from the discriminator to the generator by exploiting hierarchical reinforcement learning. In MaskGAN, [18] proposes the actor-critic GAN architecture that uses reinforcement learning to train the generator, where the in-filling technique may alleviate mode collapse.

(ii) Continuous approximation of the discrete distribution. In TextGAN, [19] utilizes a kernelized discrepancy metric to map high-dimensional latent feature distributions of real and synthetic sentences, with the aim of mitigating the model collapse. Instead of applying standard GAN objective, FM-GAN by [20] suggests to match the latent feature distributions of real and synthetic sentences exploiting the feature-movers distance. In ARAE, [21] utilizes the adversarial autoencoder to transform the

discrete data into a continuous latent space for GAN training. In GAN for sequences of discrete elements by [2] and RelGAN by [22], Gumbel-Softmax approaches are suggested to approximate the discrete-valued distribution for continuous-valued distribution. There are also some transformer-based adversarial learning models by [23] and [24] to generate discrete-valued symbolic music sequence.

2.3 Novelty of this study

Melody attributes are actually discrete in nature. In this work, we focus on conditional discrete-valued melody generation from lyrics by extending our previous work by [1]. Here, we propose a hybrid GAN structure for learning multiple melody attributes which contains two novel techniques to improve the quality of lyrics-conditioned melody generation: (i) Relational reasoning technique is applied to modeling not only dependency inside each sequence of music attributes, but also consistency among three sequences of music attributes during the training stage. (ii) Gumbel-Softmax technique is utilized to approximate the discrete-valued distribution of music attributes in a conditional hybrid GAN.

3 Conditional GAN

We propose an end-to-end deep generative model for generating melodies conditioned on lyrics. The proposed C-Hybrid-GAN model is trained by considering the alignment relationship between sequences of music attributes and their corresponding lyrics. It consists of two main components, as shown in Fig. 1: (i) a generator with three independent relational memory-based conditional sub-networks, and (ii) a discriminator based on single relational memory for long-term dependency modeling and for providing informative updates to the generator. The Gumbel-Softmax relaxation technique is exploited to train GAN for directly generating discrete-valued sequences. Particularly, a hybrid structure is used in the adversarial training stage, containing three independent branches for attributes in the generator and one branch for concatenating attributes in the discriminator. Relational memory is employed to model not only the dependency inside each sequence of attributes during the training of the generator, but also the consistency among three sequences of attributes during the training of the discriminator.

3.1 Relational memory core

Relational memory core (RMC) as a relational reasoning technique proposed by [25] is composed of a fixed set of memory slots and employs multi-head dot product attention (MHDPA), also known as self-attention between the memory slots suggested by [26], to enable interaction between them and facilitate long-term dependency modeling. [25] empirically shows that RMC is better-suited for tasks such as language modeling that benefits from relational reasoning across the sequential information as compared to LSTM. Formally, we suppose M_t represents the memory of the RMC module and x_t represents the input at time t. Let H represent the number of attention heads. For each head h, M_t is used to construct queries $Q_t^{(h)} = M_t W_q^{(h)}$, and its combination with x_t is used to construct keys $K_t^{(h)} = [M_t; x_t] W_k^{(h)}$ and values $V_t^{(h)} = [M_t; x_t] W_v^{(h)}$, where [;] represents the row-wise concatenation operation, and $W_k^{(h)}, W_v^{(h)}, W_q^{(h)}$ are weights. An attention weight is computed from $Q_t^{(h)}$ and $K_t^{(h)}$, and $\tilde{M}_{t+1}$ is computed as the product of attention weight and the value, as follows:

$$\begin{aligned} \tilde{M}_{t+1} &= [\tilde{M}_{t+1}^{(1)} : \cdots : \tilde{M}_{t+1}^{(H)}], \\ \tilde{M}_{t+1}^{(h)} &= softmax\left(\frac{Q_t^{(h)} (K_t^{(h)})^T}{\sqrt{d_k}}\right) V_t^{(h)}. \end{aligned} \tag{1}$$

where d_k is column dimension of the key $K_t^{(h)}$ and [:] represents column-wise concatenation. Then, the memory M_{t+1} is updated, and the output o_t is computed from $\tilde{M}_{t+1}$ and M_t by

$$M_{t+1} = f_{\theta_1}(\tilde{M}_{t+1}, M_t), \; o_t = f_{\theta_2}(\tilde{M}_{t+1}, M_t), \tag{2}$$

where f_{θ_1} and f_{θ_2} are parameterized functions consisting of skip connections, multi-layer perceptron, and gated operation.

3.2 Generator with three relational sub-networks

The role of the generator network is to generate a melody with given lyrics. The generator network is composed of three independent relational memory-based conditional sub-networks. Each sub-network is responsible for generating a sequence of a particular attribute conditioned on the lyrics, i.e., either a pitch sequence, $\hat{y}^p = [\hat{y}_1^p, \ldots, \hat{y}_T^p]$, a duration sequence, $\hat{y}^d = [\hat{y}_1^d, \ldots, \hat{y}_T^d]$, or a rest sequence, $\hat{y}^r = [\hat{y}_1^r, \ldots, \hat{y}_T^r]$. The key component of each sub-network is the RMC module.

We explain the generation process with the pitch attribute as an example, and the similar process applies to the

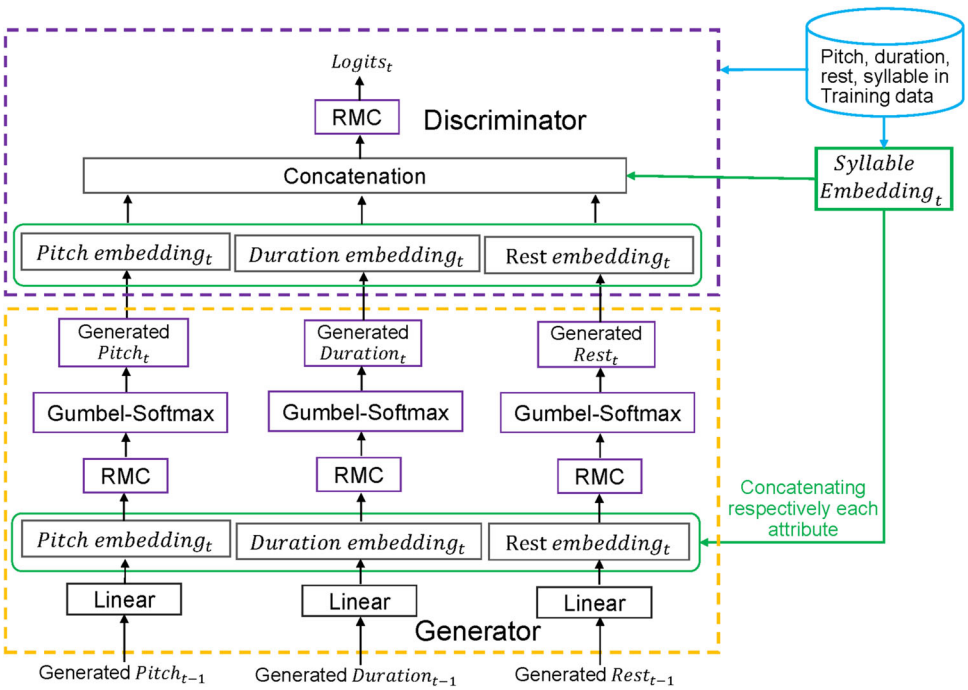

Fig. 1 Architecture of conditional hybrid GAN

other two music attributes (duration, rest). At each time step t, the input to the sub-network is one-hot encoded representation of the pitch attribute generated during the previous time step $\hat{y}^p_{t-1} \in \mathbb{R}^{100}$ and the embedded lyrics syllable $x_t \in \mathbb{R}^{20}$. During the forward pass of the sub-network, $\hat{y}^p_{t-1}$ is passed through a linear layer to obtain a dense representation of the pitch attribute, which is then concatenated with x_t and passed through a fully connected (FC) layer with the ReLU activation. The output of the FC layer and the RMC memory M_{t-1} are further passed through the RMC layer, whose output is further passed through a linear layer to obtain the output logits $o_t \in \mathbb{R}^{100}$. The Gumbel-Softmax operation is performed on o_t to obtain the one-hot approximation of the pitch attribute $\hat{y}^p_t \in \mathbb{R}^{100}$. $\hat{y}^p_0 \sim Uniform(0, 1)$ is used for the initial time step.

Since sequences with length $T = 20$ are utilized in our model, we repeat this process for 20 steps and generate the pitch sequence $\hat{y}^p = [\hat{y}^p_1, \hat{y}^p_2, \ldots, \hat{y}^p_T]$, $\hat{y}^p_t \in \mathbb{R}^{100}, 1 \le t \le 20$. The other two sub-networks, respectively, follow the same procedure to generate a duration sequence $\hat{y}^d = [\hat{y}^d_1, \hat{y}^d_2, \ldots, \hat{y}^d_T]$, $\hat{y}^d_t \in \mathbb{R}^{12}, 1 \le t \le 20$ and a rest sequence $\hat{y}^r = [\hat{y}^r_1, \hat{y}^r_2, \ldots, \hat{y}^r_T]$, $\hat{y}^r_t \in \mathbb{R}^7, 1 \le t \le 20$.

In the generator network, the embedding dimensions of the pitch, duration, and rest are set to 32, 16, and 8, respectively. In the pitch sub-network, the fully connected layer following the embedding layer uses the ReLU activation with 64 units. The RMC layer following the fully connected layer uses a single memory slot with the head size set to 64, the number of heads set to 2, and the number of blocks set to 2. In the duration sub-network, the fully connected layer following the embedding layer uses the ReLU activation with 32 units. The RMC layer following the fully connected layer uses a single memory slot with the head size set to 32, the number of heads set to 2, and the number of blocks set to 2. In the rest sub-network, the fully connected layer following the embedding layer uses the ReLU activation with 16 units. The RMC layer following the fully connected layer uses a single memory slot with the head size set to 16, the number of heads set to 2, and the number of blocks set to 2.

3.3 Gumbel-Softmax

Training GANs for the generation of discrete data faces a non-differentiable problem due to discrete-valued output from the generator. The gradient of the generator loss $\frac{\partial loss_G}{\partial \theta_G}$ cannot be back propagated to the generator via the discriminator, and hence generator parameters θ_G cannot be updated. To overcome this issue, we apply the Gumbel-Softmax relaxation technique. Using the generator sub-network responsible for generating the pitch attribute as an example, we explain more about the non-differentiability issue. In our dataset, the number of distinct MIDI numbers is 100. At time step t, the output logits obtained from the generator sub-network are denoted as $o_t \in \mathbb{R}^{100}$. Then, we can obtain the next one-hot encoded pitch attribute y^p_{t+1} by sampling:

$$y^p_{t+1} \sim softmax(o_t). \tag{3}$$

Here, $softmax(o_t)$ represents the multinomial distribution on the set of all possible MIDI numbers. Because the sampling operation in Eqn. (3) is not differentiable, this implies the presence of a step function at the output of the sub-network. Since the derivative of a step function is 0 almost everywhere, $\frac{\partial loss_G}{\partial \theta^p_G} = 0$, this is the non-differentiability issue which is to be mitigated by applying the Gumbel-Softmax relaxation. The Gumbel-Softmax relaxation defines a continuous distribution over the simplex that can approximate samples from a categorical distribution [27, 28]. Applying Gumbel-Softmax, we can reparameterize the sampling in Eq. (3) as

$$\hat{y}^p_{t+1} = softmax(\beta(o_t + g_t)), \tag{4}$$

where $\beta > 0$ is a tunable parameter called *inverse temperature*, $g^{(i)}_t$ is a random variable from the *i.i.d* standard Gumbel distribution, i.e., $g^{(i)}_t = -\log(-\log U^{(i)}_t)$ with $U^{(i)}_t \sim Uniform(0, 1)$. Now, $\hat{y}^p_{t+1}$ in Eqn. (4) is differentiable w.r.t. o_t, and we use it instead of y^p_{t+1} as the input to the discriminator.

3.4 Discriminator with single relation network

The discriminator has a relational memory-based network. Its role is to distinguish between the generated sequence and the real sequence conditioned on the lyrics. At each time step t, the input to the discriminator network is the one-hot encoded representation of each music attribute (either real or generated), i.e., the pitch attribute $y^p_t \in \mathbb{R}^{100}$, duration attribute $y^d_t \in \mathbb{R}^{12}$, and rest attribute $y^r_t \in \mathbb{R}^7$ and the embedded representation of lyrics syllable $x_t \in \mathbb{R}^{20}$.

Initially, during the discriminator network forward pass, each music attribute, y^p_t, y^d_t or y^r_t, is independently passed through a linear layer to obtain a dense representation. The dense representations of all three music attributes are concatenated together with x_t to form a syllable conditioned triplet of music attributes $\{y^p_t, y^d_t, y^r_t\}$. We then pass the syllable conditioned triplet of music attributes $\{y^p_t, y^d_t, y^r_t\}$ through a dense layer with the ReLU activation. The outputs of the dense layer and the RMC memory M_{t-1} are passed through the RMC layer, whose output is passed through a linear layer with a single unit to obtain the output logits $o_t \in \mathbb{R}$.

Since the length of sequences is $T = 20$, we repeat this process for 20 steps and generate a sequence of output logits $o = [o_1, o_2, \ldots, o_T]$. We then use the mean of o for the loss computation. Let o and $\hat{o}$ represent the output logits obtained for real and generated music attributes conditioned on the same lyrics, which are passed through the discriminator, respectively. Then, the discriminator loss is given by

$$loss_D = \log sigmoid\left(\frac{1}{T}\sum_{t=1}^{T} o_t - \frac{1}{T}\sum_{t=1}^{T} \hat{o}_t\right). \tag{5}$$

Here, we use the relativistic standard GAN (RSGAN) loss function in [29]. Intuitively, the loss function in Eqn. (5) directly estimates the average probability that a real melody is more realistic than a generated melody. We simply set the generator loss as $loss_G = -loss_D$.

In the discriminator network, the embedding dimensions of pitch, duration, and rest are set to 32, 16, and 8 respectively. The fully connected layer following the embedding layer uses the ReLU activation with 64 units. The RMC layer following the fully connected layer contains a single memory slot with the head size, the number of heads, and the number of blocks set to 64, 2, and 2, respectively.

4 Experiments

In this section, we discuss the experimental setup and results to demonstrate the feasibility of our proposed C-Hybrid-GAN. The melody-lyrics dataset in [1] is utilized in our experiment, which contains 13,251 sequences, each consisting of 20 syllables aligned with the triplet of music attributes $\{y^p_t, y^d_t, y^r_t\}$. The dataset is split into training, validation and test sets with the ratio of 8:1:1.

[25] empirically showed that RMC is better suited for tasks such as language modeling that benefits from relational reasoning across the sequential information as compared to LSTM. Moreover, [22] showed that GAN with RMC and Gumbel-Softmax outperforms other existing state-of-the-art generative models in terms of sample quality and diversity in text generation. In these research reports, we have seen that GAN with RMC and Gumbel-Softmax performs best. This work aims to discuss the effectiveness of melody generation from lyrics where three discrete sequences corresponding to music attributes, namely pitch, duration, and rest, are separately generated by GAN with RMC and Gumbel-Softmax when given lyrics. Therefore, to evaluate our proposed architecture, we use Self-BLEU in [30] to measure the diversity of generated samples and maximum mean discrepancy (MMD) in [31] to measure the quality of generated samples. The effect of lyrics conditioning is also investigated.

In addition, four competitive methods are implemented to compare with the proposed C-Hybrid-GAN as follows: TBC-LSTM-MLE: It contains a lyrics-conditioned LSTM-based generator which is composed of three branches of identical and independent lyrics-conditioned LSTM-based

sub-networks, each responsible for generating an attribute of a melody and trained with the MLE objective. This corresponds to a method without Gumbel-Softmax and RMC. TBC-LSTM-GAN: It is similar to TBC-LSTM-MLE, except that GAN is used and the Gumbel-Softmax technique is exploited to train a GAN for discrete-valued sequence generation. It corresponds to a method with Gumbel-Softmax but without RMC. C-Hybrid-MLE: It is similar to TBC-LSTM-MLE except that RMC-based generator is used. It corresponds to a method with RMC but without Gumbel-Softmax. C-Hybrid-GAN: The proposed method uses both RMC and Gumbel-Softmax. C-LSTM-GAN in [1]: It contains a deep LSTM generator and a deep LSTM discriminator both conditioned on lyrics, without using RMC and Gumbel-Softmax. As Gumbel-Softmax and RMC are mainly involved in the proposed C-Hybrid-GAN, their impacts are further investigated as the ablation study, and the results of TBC-LSTM-MLE and TBC-LSTM-GAN are shown in Table 2.

4.1 Experimental setup

We use the Adam [32] optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.99$ and perform gradient clipping if the norm of the gradients exceeds 5. Initially, the generator network is pre-trained with the MLE objective for 40 epochs using a learning rate of 10^{-2}. And then, adversarial training is performed for 120 epochs with a learning rate of 10^{-2} for both the generator and discriminator. Each step of adversarial training is composed of a single discriminator step and a single generator step. The batch size is set to 512 and a maximum temperature $\beta_{\max} = 1000$ is used during the adversarial training. The configuration of generator and discriminator is summarized in Table 1.

4.2 Diversity evaluation of generated sequences

We use the Self-BLEU score by [30] as a metric to measure the diversity of melodies generated by our proposed model. The value of the Self-BLEU score ranges between 0 and 1 with a smaller value of Self-BLEU implying a higher sample diversity hence a less chance of mode collapse in the GAN model. Intuitively, the Self-BLEU score measures how a generated melody sample is similar to the rest of the generated melody samples. With respect to our proposed model, to compute the Self-BLEU score, we first combine the pitch, duration, and rest sequences generated by each generator sub-network to form a sequence of music attributes, i.e., a melody. As an example, assume the sequences of pitches, durations, and rests generated by each corresponding sub-network are $\hat{p} = [\hat{p}_1, \ldots, \hat{p}_T]$, $\hat{d} = [\hat{d}_1, \ldots, \hat{d}_T]$, $\hat{r} = [\hat{r}_1, \ldots, \hat{r}_T]$, respectively. Then, we can represent a melody as $\hat{n} = [\hat{p}_1\hat{d}_1\hat{r}_1, \ldots, \hat{p}_T\hat{d}_T\hat{r}_T]$.

To compute the Self-BLEU score, we regard one generated melody as the hypothesis and the rest of the generated melodies as the references. We calculate the BLEU score for every generated melody and define the average BLEU score as the Self-BLEU metric. The results of Self-BLEU are shown in Fig. 2. During the adversarial training, Self-BLEU values of our C-Hybrid-GAN architecture reach the peak around 45 epochs, decrease until 100 epochs, and then approach to the stability. The results indicate that the diversity of generated melody samples gets better with the decrease in Self-BLEU and keeps unchanged from 100 epochs to 150 epochs.

In Fig. 3, we show sheet music corresponding to three lyrics chosen randomly from the test data. From the figure, we can observe that the proposed model generates diverse melodies.

Table 1 Configuration of the generator and discriminator

	Input	Layer 1 : FC(Linear)	Layer 2 : FC (ReLU)	Layer 3 : RMC				Layer 4 : FC (Linear)
		# units	# units	# memory slots	# heads	Head size	# blocks	# units
Generator								
Pitch	100	32	64	1	2	64	2	100
Duration	12	16	32	1	2	32	2	12
Rest	7	8	16	1	2	16	2	7
Discriminator								
Pitch	100	32	64	1	2	64	2	1
Duration	12	16						
Rest	7	8						

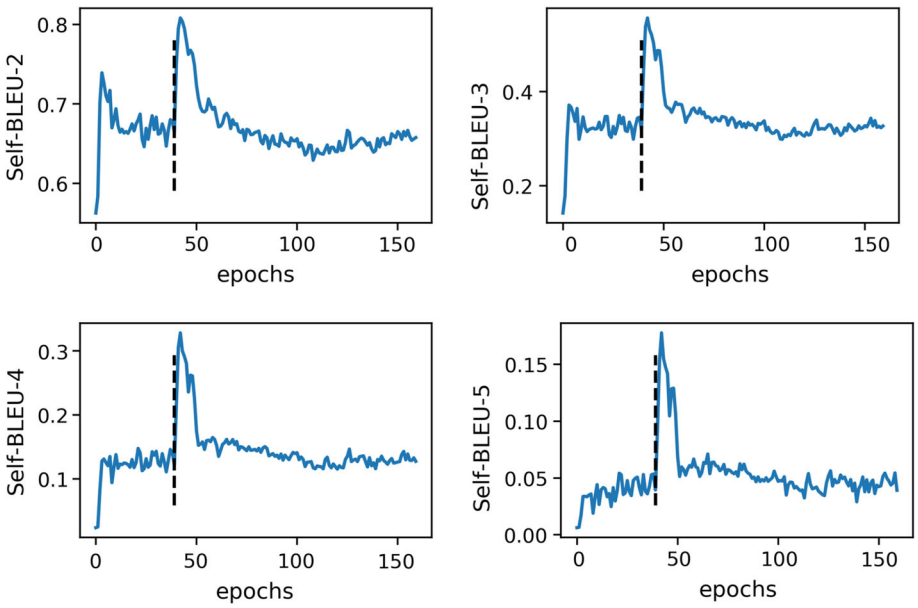

Fig. 2 Training curves of self-BLEU scores on testing dataset

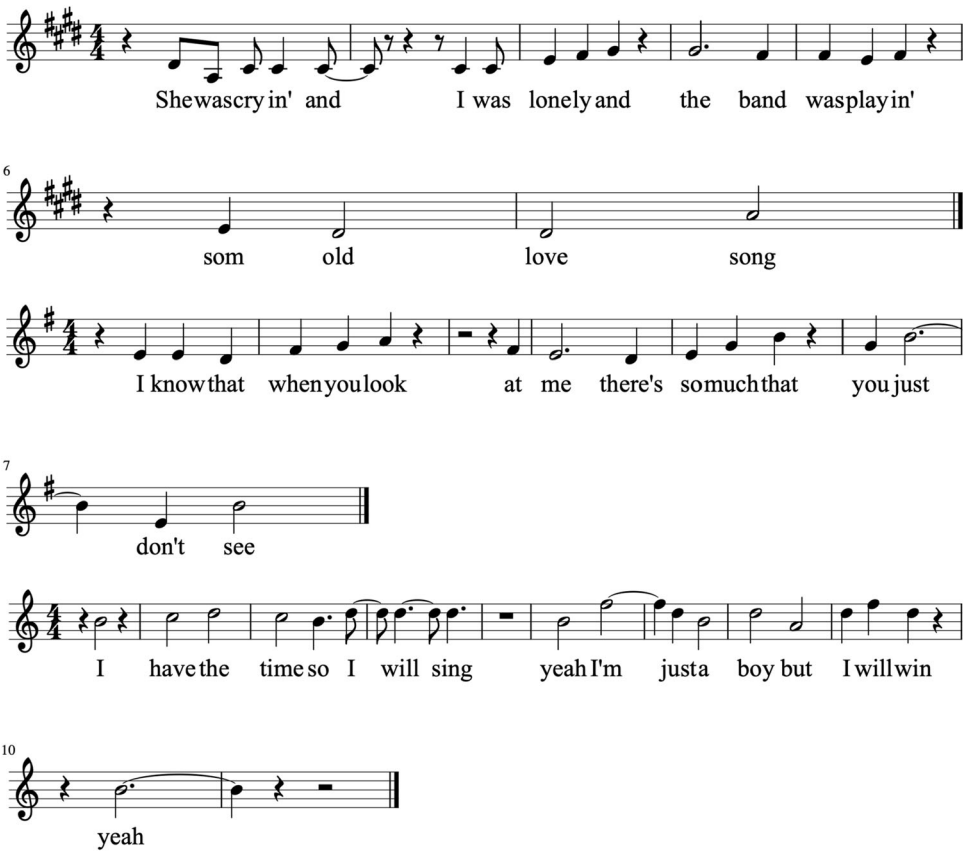

Fig. 3 Sheet music for test lyrics samples

4.3 Quality evaluation of generated sequences

The quality of generated melodies is investigated using a MMD unbiased estimator in [31]. A smaller MMD value indicates a better performance. As shown in Fig. 4, at each epoch, the generator outputs a sequence of pitches, a sequence of durations, and a sequence of rests. Using these generated sequences of pitches, durations, and rests together with the corresponding real sequences, we can compute MMD values of pitch, duration, and rest, respectively. The sum of these three values is utilized to obtain the overall MMD of the testing set.

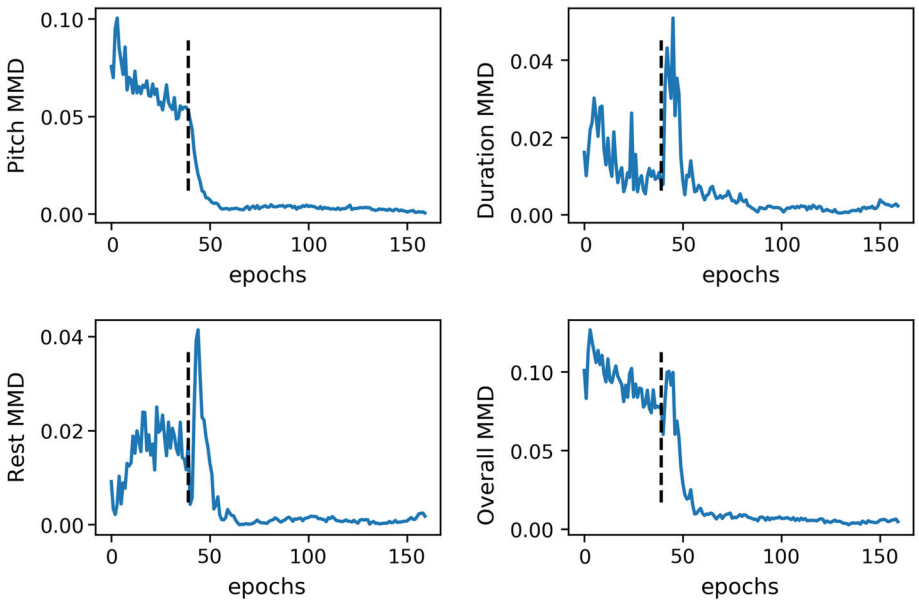

Fig. 4 Training curves of MMD scores on testing dataset

During the adversarial training, we can see that in Fig. 4, the sample quality, as measured by the MMD, first increases with the quick decrease in MMD value until 50 epochs and then starts to approach the stability and remains unchanged until 150 epochs. Each trend indicated from MMD values of pitch, duration, or rest is consistent with that of the other two and the overall trends of MMD value. The results demonstrate that the overall quality of generated melodies is high because it has a low value of MMD.

In Fig. 5, we show the sheet music corresponding to a sample lyrics at epoch 1, 40, and 160 respectively. From the figure, it is evident that the quality of the generated melody improves as the training progresses.

4.4 Effect of conditioning lyrics

To show the generated melodies are efficiently conditioned by lyrics, we follow the previous evaluation method proposed in [1], where the effect of conditioning lyrics on the generated note duration and rest duration is studied. Average note duration distance between generated sequences and sequences from ground truth dataset is calculated and shown in Fig. 6. Average rest duration distance between generated sequences and sequences from ground truth dataset is shown in Fig. 7. The subscripts $_{rs}$, $_{rn}$, and $_{rns}$, respectively denote "random songs," "random notes," and "random notes + songs." In this experiment, d is a real value, which is compared to the distribution of the random variables d_{rs}, d_{rn}, and d_{rns}, with $N = 1051$ (number of songs in testing set) and $T = 20$.

The three distributions are estimated using 10,000 samples for each random variable. As shown in Figs. 6 and 7, in each case, d is statistically lower than the mean value, indicating that the generator learns useful correlation between syllable embeddings and note/rest durations. For a detailed evaluation method of lyrics conditioning, please refer to [1].

4.5 Comparison with state-of-the-art methods

To study if C-Hybrid-GAN can generate sequences that resemble the same distribution as training samples, quantitative evaluation is performed to compare existing state-of-the-art approaches following the previous quantitative measurements in [1], for example, 2-MIDI numbers repetitions, 3-MIDI numbers repetitions, MIDI numbers span, the number of unique MIDI, the number of notes without rest, average rest value in a song, and song length. More detailed descriptions for these measurements can be found in [1].

Table 2 shows the results related to quantitative evaluation of generated melodies. It is very obvious that the overall performance of the proposed C-Hybrid-GAN outperforms other competitive methods in most aspects. For pitch-related attributes such as MIDI number span and the number of unique MIDI numbers, the proposed C-Hybrid-GAN method is closest to the ground truth. In addition, for metrics on temporal attributes such as average rest value and the number of notes without rest, C-Hybrid-GAN is also closest to the ground truth. The results illustrate that both Gumbel-Softmax and RMC contribute to the melody generation from lyrics and each of them is meaningful.

Besides metrics discussed in Table 2, the distribution of the transitions between MIDI numbers is a very important attribute for quantitatively measuring generated melodies. Figure 8 shows the distributions of the transitions for the

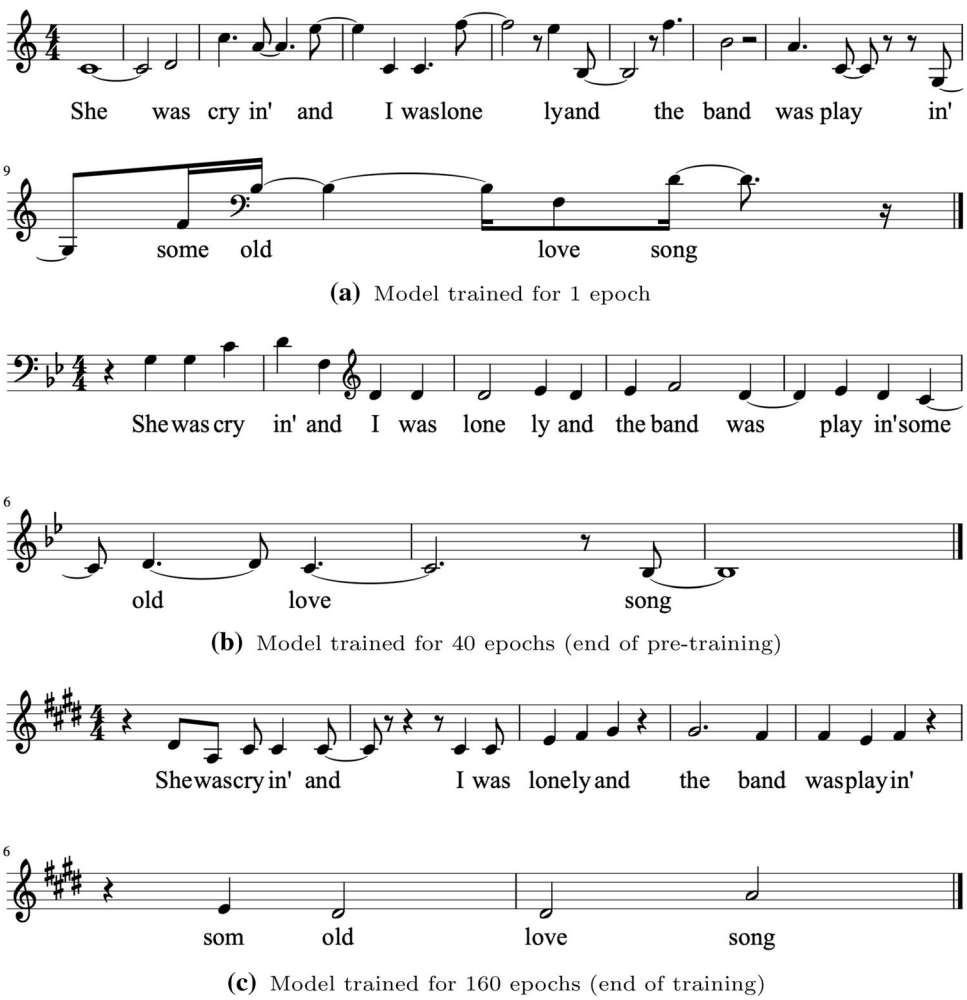

Fig. 5 Sheet music for a sample lyrics at different stages of training

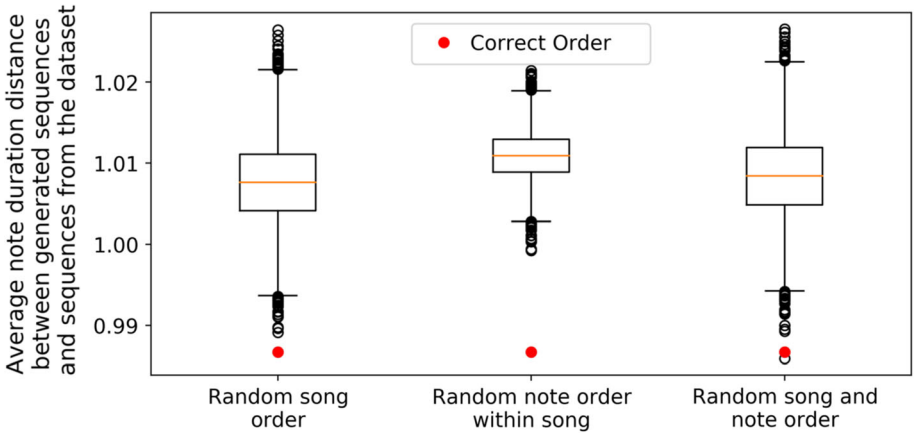

Fig. 6 Note duration attribute: $d = 0.9867$ is highlighted in red in each boxplot. Mean values are $\mu_{rs} = 1.0076$, $\mu_{rn} = 1.0108$, and $\mu_{rns} = 1.0083$, respectively

melodies generated by our model C-Hybrid-GAN, C-Hybrid-MLE, and C-LSTM-GAN. According to the occurrence of MIDI number transition in the figures, it is very clear that the proposed C-Hybrid-GAN model can better capture the distribution of MIDI number transition. This confirms that our proposed model outperforms other competitive methods and best approximates the MIDI number transition in the ground truth.

4.6 Subjective evaluation of generated sequence

To show if the generated sequence fits the lyrics-conditioning input, we also do subjective evaluation as shown in

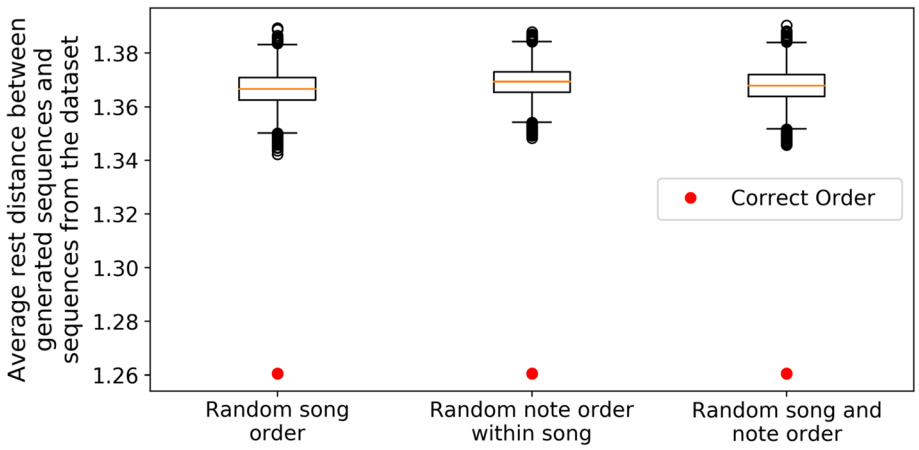

Fig. 7 Rest duration attribute: $d = 1.2605$ is highlighted in red in each boxplot. Mean values are $\mu_{rs} = 1.3666$, $\mu_{rn} = 1.3692$, and $\mu_{rns} = 1.3679$, respectively

Table 2 Metrics evaluation of attributes

	Ground truth	C-LSTM-GAN	C-Hybrid-MLE	C-Hybrid-GAN	TBC-LSTM-MLE	TBC-LSTM-GAN
2-MIDI repetitions	7.4	9.7	6.8	6.5	9.1	10.6
3-MIDI repetitions	3.8	2.2	2.8	2.7	2.1	3.0
MIDI span	10.8	7.7	12.7	12.0	13.7	12.1
Unique MIDI number	5.9	5.1	6.0	6.1	6.2	6.1
Average rest value	0.8	0.6	1.4	0.7	1.1	0.8
Non-rest note number	15.6	16.7	12.7	16.1	12.7	15.9
Song length	43.3	39.2	60.9	39.1	51.0	41.4

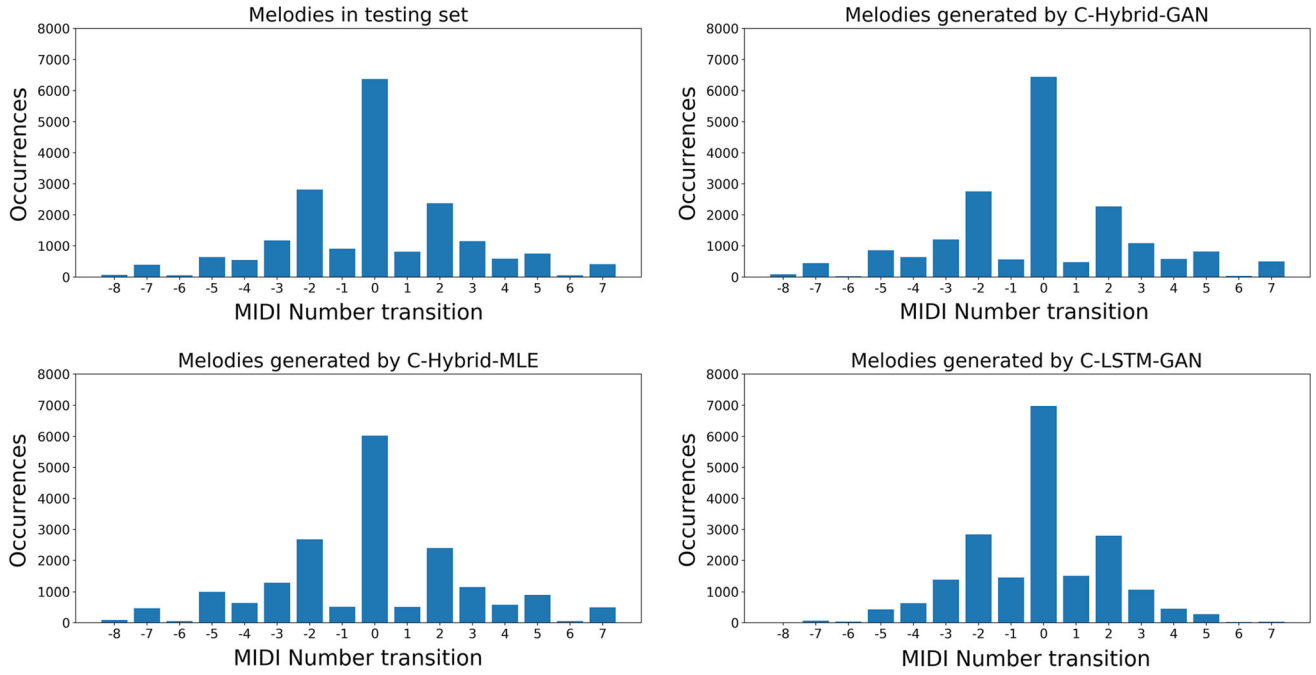

Fig. 8 Distribution of transitions

Fig. 9 to grade the effectiveness. In the subjective evaluation, we use three different lyrics to generate 18 melodies, which are available at the link.[1] The different methods in this subjective evaluation can generate music attributes (pitch, duration, and rest) when given the same lyrics. By exploiting the alignment relationship between music attributes and lyrics, all corresponding melodies are synthesized by a female voice produced by synthesizer V. As we have released these melodies for research purpose, the subjects can locally play and listen to these melodies, and anonymously give scores without submitting personal information. In particular, seven subjects without music knowledge were invited to listen to these melodies. Each melody around 15 s is played 3 times in a random order. Thus, each subject listened to 18 melodies for 3 times in this evaluation. The first play is taken to enable subjects to get used to the type of melodies, which is not used for calculating the scores. Following the existing works, three kinds of subjective measurements are used as evaluation metrics: (1) how about the entire melody? (2) how about the rhythm? (3) does the melody fit the lyrics well? Subjects are asked to give a score from 1 to 5 (where 1 corresponds to "very bad," 2 to "bad," 3 to "OK," 4 to "good," and 5 to "very good") after they listened to each melody. Subjective evaluation results are shown in Fig. 9. Generally, it can be seen that the overall result of the proposed method C-Hybrid-GAN is the closest to that of the ground truth, especially for melody and rhythm scores. Moreover, from these measurement scores, we can also see there still are gaps between melodies generated by our model and the ones from human composition (ground truth), which tells us there is much space we can investigate to improve the capability of neural melody generation. These melodies are sung by the synthesizer V software, which might be a reason resulting in the low scores of the overall performances (less than 4). A better singing synthesizer could improve the quality of melodies.

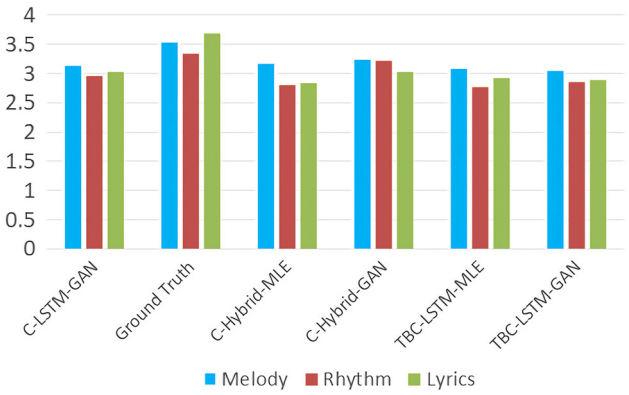

Fig. 9 Subjective evaluation

5 Conclusion

Melody generation from lyrics has been an interesting research topic in the area of artificial intelligence and music. The paired sequences of lyrics and melody should contain rhythm, syllabic patterns, story-like progression, relatable topics, and a catchy tune. The goal of this task is to design generative models that can automatically infer melodies when given lyrics in a way similar to the human way. However, current state-of-the-art generative models have the incapability of generating music because it is difficult to model discrete-valued music attributes.

In this paper, we have proposed a novel conditional hybrid generative adversarial network for generating melodies from lyrics. Three independent discrete-valued sequences containing pitch, duration, and rest are exploited to learn lyrics-conditioned melody generation. In particular, the relational reasoning method is employed to learn the dependency inside each sequence of a specific attribute during the training stage of the generator as well as the consistency among all sequences of attributes during the training stage of the discriminator. To avoid the problem of non-differentiability in GANs for discrete data generation, we exploit the Gumbel-Softmax to approximate the distribution of discrete-valued sequences. Through extensive experiments of melody generation from lyrics including the diversity and quality of generated melody samples, the effect of lyrics-based context conditioning, and the comparison with existing works, we indicate that the proposed C-Hybrid-GAN outperforms the existing cutting-edge methods in lyrics-conditioned melody generation with multiple music attributes.

Data Availability The datasets generated during and/or analyzed during the current study are available in [1] repository, https://github.com/yy1lab/Lyrics-Conditioned-Neural-Melody-Generation.

Declarations

Conflict of interest All authors declare that they have no conflicts of interest.

References

1. Yu Y, Srivastava A, Canales S (2021) Conditional lstm-gan for melody generation from lyrics. ACM Trans Multimed Comput Commun Appl 17(1):1–20

[1] https://drive.google.com/file/d/1ozUVA5suGAERP9sgdc5q3NKkRXj3jkhE/view.

2. Kusner MJ, Hernández-Lobato JM (2016) Gans for sequences of discrete elements with the gumbel-softmax distribution. arXiv:1611.04051
3. Chi W, Kumar P, Yaddanapudi S, Suresh R, Isik U (2020) Generating music with a self-correcting non-chronological autoregressive model. arXiv:2008.08927
4. Fukayama S, Nakatsuma K, Sako S, Nishimoto T, Sagayama S (2010) Automatic song composition from the lyrics exploiting prosody of Japanese language. In: Sound and music computing conference
5. Monteith K, Martinez TR, Ventura D (2012) Automatic generation of melodic accompaniments for lyrics. In: International conference on computational creativity, pp 87–94
6. Ackerman M, Loker D (2017) Algorithmic songwriting with Alysia. In: Computational intelligence in music, sound, art and design, pp 1–16
7. Bao H, Huang S, Wei F, Cui L, Wu Y, Tan C, Piao S, Zhou M (2019) Neural melody composition from lyrics. In: International conference on natural language processing and Chinese computing, pp 499–511
8. Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9(8):1735–1780
9. Liu A, Mei Y, Zhu Q, Zhu Z, Cai Z, Xie Z, Zhang M, Zhang S, Xiao J (2020) Lyrics2song: an automatic song generator for lyrics input. In: IEEE conference on multimedia information processing and retrieval, pp 388–391
10. Sheng Z, Song K, Tan X, Ren Y, Ye W, Zhang S, Qin T (2021) Songmass: automatic song writing with pre-training and alignment constraint. In: AAAI conference on artificial intelligence, vol 35, pp 13798–13805
11. Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y (2014) Generative adversarial nets. Adv Neural Inf Process Syst 27:2672–2680
12. Zhou D, Zhang H, Li Q, Ma J, Xu X (2022) Coutfitgan: learning to synthesize compatible outfits supervised by silhouette masks and fashion styles. In: IEEE transactions on multimedia, pp 1–15
13. Tuan Y-L, Lee H-Y (2019) Improving conditional sequence generative adversarial networks by stepwise evaluation. IEEE/ACM Trans Audio Speech Langu Process 27(4):788–798
14. Deng K, Fei T, Huang X, Peng Y (2019) Irc-gan: introspective recurrent convolutional gan for text-to-video generation. In: International joint conference on artificial intelligence, pp 2216–2222
15. Yu L, Zhang W, Wang J, Yu Y (2017) Seqgan: sequence generative adversarial nets with policy gradient. In: AAAI conference on artificial intelligence, vol 31, pp 2852–2858
16. Lin K, Li D, He X, Zhang Z, Sun M-T (2017) Adversarial ranking for language generation. Adv Neural Inf Process Syst 30:5998–6008
17. Guo J, Lu S, Cai H, Zhang W, Yu Y, Wang J (2018) Long text generation via adversarial training with leaked information. In: AAAI conference on artificial intelligence, vol 32, pp 5141–5148
18. Fedus W, Goodfellow I, Dai AM (2018) Maskgan: better text generation via filling in the_____. arXiv:1801.07736
19. Zhang Y, Gan Z, Fan K, Chen Z, Henao R, Shen D, Carin L (2017) Adversarial feature matching for text generation. In: International conference on machine learning, pp 4006–4015
20. Chen L, Dai S, Tao C, Zhang H, Gan Z, Shen D, Zhang Y, Wang G, Zhang R, Carin L (2018) Adversarial text generation via feature-mover's distance. Adv Neural Inf Process Syst 31:4671–4682
21. Zhao J, Kim Y, Zhang K, Rush A, LeCun Y (2018) Adversarially regularized autoencoders. In: International conference on machine learning, pp 5902–5911
22. Nie W, Narodytska N, Patel A (2018) Relgan: relational generative adversarial networks for text generation. In: International conference on learning representations
23. Zhang N (2020) Learning adversarial transformer for symbolic music generation. In: IEEE transactions on neural networks and learning systems, pp 1–10
24. Muhamed A, Li L, Shi X, Yaddanapudi S, Chi W, Jackson D, Suresh R, Lipton ZC, Smola AJ (2021) Symbolic music generation with transformer-gans. In: AAAI conference on artificial intelligence, vol 35, pp 408–417
25. Santoro A, Faulkner R, Raposo D, Rae J, Chrzanowski M, Weber T, Wierstra D, Vinyals O, Pascanu R, Lillicrap T (2018) Relational recurrent neural networks. Adv Neural Inf Process Syst 31:7310–7321
26. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin I (2017) Attention is all you need. Adv Neural Inf Process Syst 30:5998–6008
27. Jang E, Gu S, Poole B (2016) Categorical reparameterization with gumbel-softmax. arXiv:1611.01144
28. Maddison CJ, Mnih A, Teh YW (2016) The concrete distribution: a continuous relaxation of discrete random variables. arXiv:1611.00712
29. Jolicoeur-Martineau A (2018) The relativistic discriminator: a key element missing from standard gan. arXiv:1807.00734
30. Zhu Y, Lu S, Zheng L, Guo J, Zhang W, Wang J, Yu Y (2018) Texygen: a benchmarking platform for text generation models. In: ACM SIGIR conference on research & development in information retrieval, pp 1097–1100
31. Smola A, Gretton A, Song L, Schölkopf B (2007) A hilbert space embedding for distributions. In: International conference on algorithmic learning theory, pp 13–31
32. Kingma DP, Ba J (2014) Adam: a method for stochastic optimization. arXiv:1412.6980