Frontiers of Information Technology & Electronic Engineering
www.jzus.zju.edu.cn; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

Style-conditioned music generation with Transformer-GANs*

Weining WANG†, Jiahui LI, Yifan LI, Xiaofen XING†‡

School of Electronic and Information Engineering, South China University of Technology, Guangzhou 510600, China

†E-mail: wnwang@scut.edu.cn; xfxing@scut.edu.cn

Received May 21, 2023; Revision accepted Oct. 29, 2023; Crosschecked Jan. 3, 2024

Abstract: Recently, various algorithms have been developed for generating appealing music. However, the style control in the generation process has been somewhat overlooked. Music style refers to the representative and unique appearance presented by a musical work, and it is one of the most salient qualities of music. In this paper, we propose an innovative music generation algorithm capable of creating a complete musical composition from scratch based on a specified target style. A style-conditioned linear Transformer and a style-conditioned patch discriminator are introduced in the model. The style-conditioned linear Transformer models musical instrument digital interface (MIDI) event sequences and emphasizes the role of style information. Simultaneously, the style-conditioned patch discriminator applies an adversarial learning mechanism with two innovative loss functions to enhance the modeling of music sequences. Moreover, we establish a discriminative metric for the first time, enabling the evaluation of the generated music's consistency concerning music styles. Both objective and subjective evaluations of our experimental results indicate that our method's performance with regard to music production is better than the performances encountered in the case of music production with the use of state-of-the-art methods in available public datasets.

Key words: Music generation; Style-conditioned; Transformer; Music emotion
https://doi.org/10.1631/FITEE.2300359 **CLC number:** TP39

1 Introduction

As an artistic form that expresses ideas, emotions, and thoughts, musical composition has always been highly regarded. Music can express a rich variety of content, including emotional and aesthetic as well as cultural, historical, and religious aspects, through some elements such as melody, rhythm, harmony, and lyrics. Music generation has been an attractive topic for scientists for a long time. From rule-based generation to deep learning based generation, great progress has been made in automatic music generation.

Researchers are committed to studying two fundamental attributes in deep learning based music generation: structural awareness and interpretive ability (Wang L et al., 2023). Concerned with learning long musical sequences, the study of structural awareness aims to train the model to recognize the varying degrees of repetition and variation between notes (Huang CZA et al., 2019; Ren et al., 2020; Yu et al., 2022; Zhang XY et al., 2022). The study of interpretive ability aims to explore how the model can be controlled (Luo et al., 2020), facilitating the style transfer (Brunner et al., 2018) and style-conditioned generation (Wang WP et al., 2022).

Music style is the representative and unique appearance presented by a musical work, and it is one of the most salient qualities of music. Introducing style information into music generation can not only improve the quality and diversity of generated music (Hung et al., 2021; Wang WP et al., 2022), but also

‡ Corresponding author
* Project supported by the Natural Science Foundation of Guangdong Province in China (No. 2021A1515011888)
ORCID: Weining WANG, https://orcid.org/0009-0006-0589-8157; Xiaofen XING, https://orcid.org/0000-0002-0016-9055

provide users with more personalized and customized music experiences. For example, the affective aspect of music, which can be seen as a kind of music style, has been studied in emotion-based music generation (Sulun et al., 2022). To generate music with the desired emotion, some researchers combined music theory at music generation and adjusted music elements generated by the model (Mou et al., 2023). Some research extracts additional information like rhythms and chords from music to enable style-based generation (Choi et al., 2020; Jiang et al., 2020; Huang YS and Yang, 2020). With the introduction of additional information, the quality of the generated music has improved. However, the vague definition of style limits the interpretive ability of such methods. By designing additional modules, some studies introduce style information in the model latent space (Mao et al., 2018; Roberts et al., 2018; Lim et al., 2020) to enhance the interpretive ability. However, since these methods are mostly based on style transfer, they can generate only short music rather than full music from scratch. Recently, researchers have introduced style information in data representation, thus enabling the generation of full music from scratch (Hung et al., 2021; Wang WP et al., 2022). In these works, the addition of control tokens to the music sequence results in an improvement in the quality and diversity of the generated music. Nevertheless, the style information is not used in the output layer of the model, resulting in the generated music not expressing obvious style characteristics. Therefore, without combining research on interpretive ability with structural awareness, the performance of existing models is limited.

In this paper, we focus on the combination of interpretive ability and structural awareness of the model, and further on conditioning with style-related tokens and generating full music from scratch. A style-conditioned linear Transformer is proposed to embed music style information in the top layer of the linear Transformer to adjust the output. Considering the structural awareness of the model, the style-conditioned patch discriminator with two novel losses is proposed to enhance the modeling of music sequences through an adversarial learning mechanism under the guidance of style information. We also design a specific objective metric to evaluate the generation effect of the model with style control. Our main contributions are as follows:

1. We propose a model for music generation under style control, namely style-conditioned Transformer-GANs (SCTG). Style information is effectively used to enhance the interpretive ability of the model and a style-conditioned patch discriminator is designed to consider structural awareness.

2. We design a style-conditioned linear Transformer in SCTG, which models musical instrument digital interface (MIDI) event sequences and emphasizes the role of style information. It embeds style information in the output module as a solution.

3. A style-conditioned patch discriminator is proposed behind the linear Transformer in SCTG, which introduces an adversarial learning mechanism under style condition to enhance the structural awareness and interpretive ability. Two specific losses, the music style category loss $\mathrm{Loss}_{\mathrm{Cls}}$ (for interpretive ability) and the music style information adversarial loss $\mathrm{Loss}_{\mathrm{Gan}}$ (for structural awareness), are variously designed in the discriminator.

In addition, we define a metric to evaluate the stylistic similarity between the generated music and the music in the original dataset. Unlike conventional metrics that are calculated across the entire dataset, our newly proposed metric emphasizes the style consistency.

We perform experiments on two publicly available datasets and the experimental results demonstrate that our model achieves the best results in both objective and subjective evaluations.

2 Related works

2.1 Rule-based music generation

A rule-based music generation model is a non-adaptive model that generates music based on theoretical knowledge. The performance of rule-based generative models depends heavily on the creativity of the researcher, the depth of understanding of musical concepts, and how the musical structure is expressed according to the corresponding variables (Kaliakatsos-Papakostas et al., 2020; Wang L et al., 2023). Supper (2001) proposed a syntactic rule-based formula for generating musical rhythms that focuses on the structure and redundancy of the music and controls some global structure of the rhythmic sequence of notes. Delgado et al. (2009) used rules based on musical knowledge, introduced

users' emotional input, and proposed a bottom–up approach to design a two-level structure for music generation in a modular way. In addition, other researchers designed music generation systems based on similarity rule constraints (Leach and Fitch, 1995) and rhythm rule constraints (Herremans and Chew, 2019).

Rule-based music generation models can effectively restrict the exploration region; however, music is organized by many levels of abstract structures and rules. These models cannot effectively capture deeper levels of knowledge. In addition, rule-based music generation models limit the diversity of outputs to some extent. When the number of rules increases, the ambiguity of the model increases.

2.2 Deep learning methods for music generation

With the development of deep learning, many deep learning models are being gradually used in music generation. Unlike rule-based music generation models, deep learning based music generation models can learn the distribution and relevance of samples from an arbitrary music corpus and generate music that represents the music style of the corpus by prediction (e.g., predicting the pitch of the next note of a melody).

Generative adversarial nets (GANs) (Goodfellow et al., 2020) perform well in generating high-quality images, and some researchers have used them in music generation (Yang et al., 2017; Liu and Yang, 2018; Trieu and Keller, 2018; Jhamtani and Berg-Kirkpatrick, 2019). Among them, MuseGAN (Dong et al., 2018) learns piano-roll for multi-track music and proposes the construction of connections between tracks to accomplish independent generation within tracks, global generation between tracks, and composite generation. However, this method is prone to produce over-fragmented notes. BinaryMuseGAN (Dong and Yang, 2018) introduces binary neurons as input to the generator, which makes the discriminator to learn decision boundaries more easily to reduce over-fragmented notes. Although GANs can learn polyphonic piano-roll well, they are difficult to employ when it comes to training and modeling musical score sequences.

Variational auto-encoder (VAE) is essentially a compression algorithm for encoders and decoders that has been able to analyze and generate information. This model can be used in music generation to analyze pitch distribution and rhythmic variation information (Jiang and Wang, 2019; Lousseief and Sturm, 2019; Rivero et al., 2020). MIDI-VAE (Brunner et al., 2018) constructs pitch, intensity, and instrument sequences for musical compositions with three pairs of encoder/decoder sharing a potential space. MusicVAE (Roberts et al., 2018) uses hierarchical decoders to effectively improve the modeling of long sequences and enhance the structure of the generated music. MIDI-Sandwich2 (Liang et al., 2019) improves the refinement method of BinaryMuseGAN (Dong and Yang, 2018) to transform the binary input problem into a multi-label classification problem. Using the Kullback–Leibler (KL) divergence, VAE-based models can be forced to follow any probability distribution, and thus they provide greater flexibility in choosing the prior distribution of latent variables. However, there are some difficulties involved in employing these models to model long sequences, and they require splitting up the complete music.

Based on the attention mechanism, Transformer (Vaswani et al., 2017) is good at modeling long sequences, widely used in the field of natural language processing. In recent years, Transformer has been widely used in music generation (Huang CZA et al., 2019; Yu et al., 2022). Unlike GAN-based music generation which treats music as piano-roll, Transformer-based music generation is usually designed to encode music as a MIDI event sequence first (Huang YS and Yang, 2020; Hsiao et al., 2021), which restricts music to a symbolic domain and enhances the structure. Considering the structural awareness, Transformer-based models can learn multiple levels of features of MIDI event sequences well (Wu XC et al., 2020; Shih et al., 2022), thus generating higher-quality music clips. At the same time, some studies have noticed that the training approach that relies solely on the model to predict the next token cannot guarantee the overall coordination of the generated music fragments. Therefore, to further emphasize the structural awareness, some research proposes the introduction of adversarial mechanisms into this task (Muhamed et al., 2021; Zhang N, 2023). Such methods are based on the Gumbel-Softmax technique (Jang et al., 2017) to accomplish the sampling of the discrete output of the Transformer generator, and employ the design of an inverse temperature parameter that decreases with training time to ensure

the stability of the training (Nie et al., 2019).

However, these models do not consider music style information and do not include the interpretive ability as one of the focuses of their studies. We emphasize the importance of music style information and combine the study of structural awareness with the interpretive ability.

2.3 Style-based music generation

Style information is important in music modeling and automatic generation. In the study of interpretive ability, some researchers consider emotion as a type of style and use emotion information to control the music generation process. There are studies that constrain emotional classification in hidden layers (Ferreira and Whitehead, 2021) to enable the generation of music with a specific emotion. In Emopia (Hung et al., 2021), music emotion labels are added in the header of MIDI event sequences to distinguish variations in the symbol domain. Using a linear layer, Sulun et al. (2022) concatenated music emotion embedding with the original note sequence to affect the model. At the same time, research shows that labeling music style information through similar methods effectively enhances the similarity between the generated music and the music in the training data (Wang WP et al., 2022). Although these works proved that the introduction of style information can help the model learn effectively, they did not consider the structural awareness of the model. Thus, we consider both interpretive ability and structural awareness, effectively using the style information for automatic music generation. We conduct our study based on both emotion style and composer style, and experimentally demonstrate the effectiveness of this combination.

3 Method

Music style information plays a crucial role in the model's ability to learn specific styles. We introduce our music generation model SCTG in this section. We design the style-conditioned linear Transformer to embed style information in the output module of the generation model. We also propose the style-conditioned patch discriminator, which calculates the discriminant loss and distinguishes the style of the generated music under the guidance of style labels. Our style-conditioned linear Transformer and style-conditioned patch discriminator both emphasize the use of music style information, resulting in the generated music that follows the data distribution in the note sequence and effectively expresses the corresponding style. The overall architecture of our SCTG model is shown in Fig. 1. As seen in Fig. 1, "tokens with labels" represents how we introduce style information into MIDI events, and will be described in Section 3.1. The style-conditioned generator is our style-conditioned linear Transformer that generates complete musical composition from scratch based on a specified target style, and will be described in Section 3.2. The "discrete MIDI scores" shows how we discretize tokens related to MIDI scores. The style-conditioned patch discriminator introduces adversarial learning mechanisms to enhance the interpretive ability and structural awareness of the model, and will be described in Section 3.3. The [CLS] stands for the special [CLS] token, like in BERT (Devlin et al., 2019).

3.1 Data representation

To model symbolic music generation, we need a data representation method for MIDI format. There are many representation methods discussed in previous works (Waite et al., 2016; Wu SL and Yang, 2020; Liao et al., 2022). Since there has been no standard thus far, we design an event-based data representation method with inserted style information (Fig. 2). The details of our representation vocabulary are shown in Table 1. This design includes the following key points:

1. Group musical information

Our token sequences contain the musical information in MIDI scores (Oore et al., 2020) and consider the idea of "token grouping" in CP (Hsiao et al., 2021) to reduce the length of token sequences. Particularly, as shown in Fig. 2b, there are four score-related tokens ([Bar], [Position], [Pitch], and [Duration]) containing information about musical composition, including beat advancement information, note pitch change information, and note duration. The relevant elements of descriptive information pertaining to these tokens are shown in Table 1.

2. Introduce style information

Our token sequences contain not only the musical information but also the style information. Particularly, we have two special tokens ([Class] and [CP]) in our representation, which are shown in

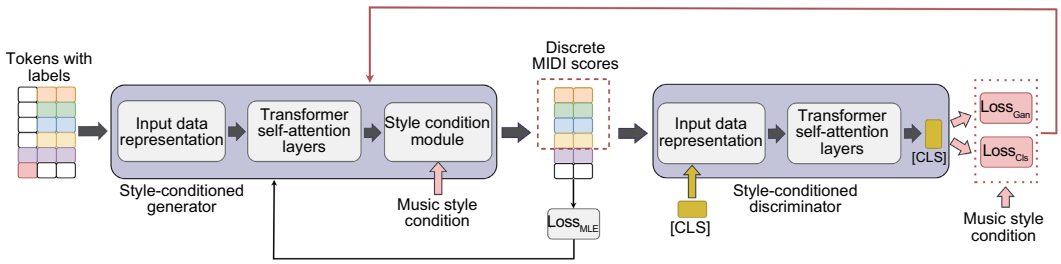

Fig. 1 Overall architecture of our music generation model SCTG

Table 1 Information about tokens used in SCTG

Token type	Description	Vocabulary size	Embedding size	Used in discriminator or not
[Bar]	Begin/Continue a bar	2+1	32	Yes
[Position]	Position in a bar	16+1	128	Yes
[Pitch]	MIDI note numbers	86+1	256	Yes
[Duration]	Duration time	64+1	256	Yes
[CP/Class]	CP or class	2	32	No
[Style]	Emotion/Composer label	4+1/8+1	64	No

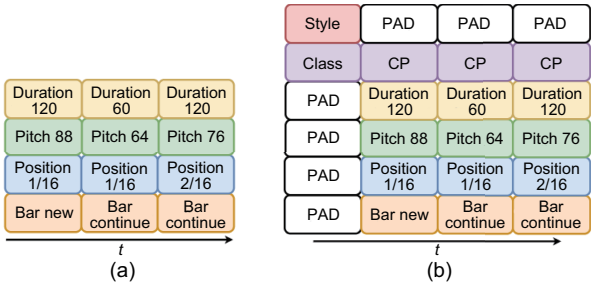

Fig. 2 Our method for introducing music style information into a MIDI event sequence: (a) original MIDI events; (b) MIDI events with style information. References to color refer to the online version of this figure

purple in Fig. 2b. They represent the current compound word that includes the music style information or MIDI score information. According to the method in compound word with style (CPS) (Wang WP et al., 2022), we introduce music style information in the header of each token sequence, as shown in Fig. 2. We insert the style token [Style] in the header and the rest tokens are set to [PAD], while the style token is set to [PAD] in the later.

Therefore, our data representation has the ability to introduce style information as input to our later generator and discriminator.

3.2 Style-conditioned linear Transformer

The interpretive ability of the model is a focus of automatic music generation research. Our goal is to develop a model that can accurately represent the sequence information, while demonstrating interpretive ability. We introduce the style-conditioned linear Transformer to address the limitations of earlier research on music style-conditioned generation. Based on the linear Transformer (Katharopoulos et al., 2020), we embed style information in the output module as a solution. Unlike previous models that put only special tokens in the sequence, we embed style information directly into the hidden space of the model. Concatenated with output features of the model, the style information feature can modulate the output of the whole sequence. The structure of our model is shown in Fig. 3. The input sequence is our MIDI events with style information, described in Section 3.1. The tokens on the output module represent the output events at current moment t.

1. Input data representation

As input to the Transformers, tokens in a sequence are represented by an embedding vector $\boldsymbol{x}_t \in \mathbb{R}^d$, and then a positional embedding vector is added (Ke et al., 2021). We combine the embedding vectors $\boldsymbol{e}_t$ of the compound word sequence w_t, then obtain the input vectors $\boldsymbol{x}_t$ by a projection matrix $\boldsymbol{W}_{\text{in}}$, and finally add the positional embedding vector to obtain the input of the sequence. The details are given in the following:

$$\begin{cases} \boldsymbol{e}_{t,k} = \text{Embedding}_k(w_{t,k}),\ k = 1, 2, \ldots, K, \\ \boldsymbol{x}_t = \boldsymbol{W}_{\text{in}}\ [\boldsymbol{e}_{t,1} \oplus \boldsymbol{e}_{t,2} \oplus \ldots \oplus \boldsymbol{e}_{t,K}], \\ \overrightarrow{\boldsymbol{x}}_t = \text{Positional encoding}(\boldsymbol{x}_t), \end{cases} \tag{1}$$

where Embedding_k involves the use of lookup tables.

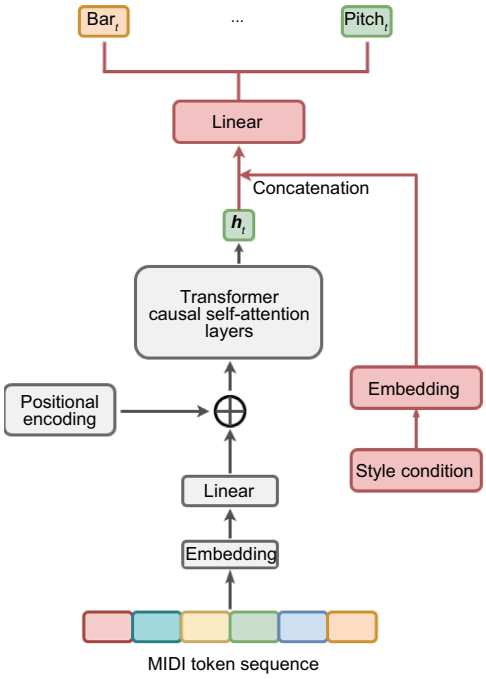

Fig. 3 Architecture of our style-conditioned linear Transformer

In essence, the input here can be considered as the compressed information representing the compound word sequence. This compressed information can aggregate the music information in different dimensions and facilitate the modeling of the whole sequence. Different embedding sizes are used according to the vocabulary size of the token type, as shown in Table 1.

2. Style condition module

In the output module of the Transformer, to control the output features, we first embed the music style information w_{s} into a style vector $\boldsymbol{e}_{\mathrm{s}}$, then combine it with the hidden vector of the model, and finally use different feedforward headers for the output. Specifically, at the t^{th} time step, the output process can be summarized as follows:

$$\begin{cases} \boldsymbol{h}_t = \text{Self-attn}\left(\overrightarrow{\boldsymbol{x}}_{t-1}\right), \\ \boldsymbol{e}_{\mathrm{s}} = \text{Embedding}_{\mathrm{style}}(w_{\mathrm{s}}), \\ \boldsymbol{h}_t^{\mathrm{out}} = \boldsymbol{W}_{\mathrm{out}}\left[\boldsymbol{h}_t \oplus \boldsymbol{e}_{\mathrm{s}}\right], \\ \widehat{w_{t,k}} = \text{Sample}_k(\text{softmax}\left(\boldsymbol{W}_k \boldsymbol{h}_t^{\mathrm{out}}\right)), \\ \qquad k = 1, 2, \ldots, K, \end{cases} \tag{2}$$

where $\boldsymbol{W}_1$–$\boldsymbol{W}_K$ represent the K feedforward heads that output the K tokens in the compound word, and Self-attn$(\cdot)$ stands for the causal self-attention layers. Therefore, the style information in this output process will influence the whole sequence, ultimately making the generated music express the style characteristics.

3. Discretize output for discriminator

We insert a discriminator after the generator in the training process as a tool to guide our style-conditioned generator, to mitigate the effects of exposure bias (Muhamed et al., 2021; Zhang N, 2023) and enhance the structural awareness of the model. However, the output of the generator in the training phase is continuous, while the discriminator input needs to be discrete. If we directly discretize the output of the generator, the gradient of the discriminator loss is forbidden from propagating to the generator. Therefore, it is notoriously difficult to generate discrete sequences from a continuous output. Several studies have proposed strategies to solve this problem, such as Gumbel-Softmax (Jang et al., 2017). The Gumbel-Softmax method uses k Gumbel distributions with location 0 and scale 1, g_i ($i \in \{1, 2, \ldots, k\}$), and draws samples y from the categorical distribution using

$$y = \text{one-hot}\left(\arg\max_i \left(g_i + \log \pi_i\right)\right), \tag{3}$$

where each π_i is a categorical distribution that we want to discretize. Then, we can obtain a continuous differential approximation to the categorical distribution parameterized by the softmax function:

$$y = \text{softmax}((1/\tau)(\pi + g)), \tag{4}$$

where τ is a temperature parameter that regulates how close y is to the categorical (lower τ) versus to the uniform distribution (higher τ). We let this parameter decay during the training process to maintain the stability of the training, which is the same as the approach adopted in Muhamed et al. (2021).

3.3 Style-conditioned patch discriminator

In this subsection, we propose a style-conditioned patch discriminator based on the linear Transformer. We innovatively introduce adversarial training mechanisms and style information into the model to enhance structural awareness and interpretive ability. We also propose two novel losses in our discriminator, the music style category loss Loss_{Cls} and the music style information adversarial loss Loss_{Gan}. Loss_{Gan} is used to enhance the structural awareness and Loss_{Cls} is used to enhance the interpretive ability.

In the task of automatic music generation, we aim for the model to generate music of a specific style according to our needs while ensuring high-quality, smooth transitions. A Transformer decoder models the entire sequence by learning the correlation between tokens within the sequence as follows:

$$p\left(x_t \mid x_{<t}\right), \tag{5}$$

where x_t is the element of a sequence to be predicted at time step t, and it is predicted based on all the previous information. The process is then performed using maximum likelihood estimation (MLE) (Williams and Zipser, 1989) for constraint. However, this training approach usually performs poorly in the face of long sequences like music sequences and suffers from exposure bias (Muhamed et al., 2021; Zhang N, 2023), and it thus cannot be optimized in terms of the dimensionality of the entire music sequence. This problem can be effectively alleviated by introducing the discriminative object of a GAN (Goodfellow et al., 2020). Therefore, we propose the style-conditioned patch discriminator, which introduces the GAN object under the style condition. Our discriminator enhances the structural awareness of the music generation model and the interpretive ability as well. The structure of the discriminator is shown in Fig. 4.

1. Adversarial training mechanisms

Specifically, we extract the sequence of four tokens related to MIDI scores from the output of the generator, obtained through the Gumbel-Softmax technique. The types of tokens to be input into the discriminator are detailed in Table 1. Taking inspiration from the visual Transformer (Dosovitskiy et al., 2021), we first separate the sequence into patches of a certain length, with each patch undergoing conversion into a single feature vector **pk**. Next, similar to BERT (Devlin et al., 2019), we concatenate the vectors with the [CLS] token and input them into the same attention module used by the generator model. To be specific, the input to the discriminator undergoes the following processing steps:

$$\begin{cases} \boldsymbol{e}_{d,k} = \text{Embedding}_k(w_{g,k}),\ k = 1, 2, \ldots, K, \\ \boldsymbol{c}_{d,t} = \boldsymbol{W}_{\text{in}}[\boldsymbol{e}_{d,1} \oplus \boldsymbol{e}_{d,2} \oplus \ldots \oplus \boldsymbol{e}_{d,K}], \\ \boldsymbol{p}_{k,t} = \text{Patch}(\boldsymbol{c}_{d,t}), \\ \boldsymbol{x}_{d,t} = \boldsymbol{p}_{k,t} \oplus \mathbf{Cls}_{\text{v}}, \end{cases} \tag{6}$$

where $w_{g,k}$ represents the output token of the generator.

2. Two losses with style

At the output of the discriminator, we variously construct the music style category loss Loss_{Cls} and the music style information adversarial loss Loss_{Gan}, where the output vector $\mathbf{Cls}_{\text{v}}$ of the [CLS] token passes through a music style classification layer to obtain the music style category loss. $\mathbf{Cls}_{\text{v}}$ combines with the style information feature and passes through a linear layer to obtain the music style information adversarial loss. Finally, the overall objective for the discriminator in this training stage is

$$\begin{cases} \text{Loss}^{\text{D}} = \alpha\,\text{Loss}^{\text{D}}_{\text{Cls}} + \beta\,\text{Loss}^{\text{D}}_{\text{Gan}}, \\ \text{Loss}^{\text{D}}_{\text{Cls}} = -\mathbb{E}_{x\sim p_{\text{data}}(x)}[\log P(y_{\text{style}}|x)] \\ \qquad\qquad -\mathbb{E}_{z\sim p_z(z)}[\log P(y_{\text{style}}|G(z))], \\ \text{Loss}^{\text{D}}_{\text{Gan}} = -\mathbb{E}_{x\sim p_{\text{data}}}(x)[\log D(x)] \\ \qquad\qquad -\mathbb{E}_{z\sim p_z(z)}[\log(1 - D(G(z)))], \end{cases} \tag{7}$$

where α and β control the ratio of the two losses (we analyze their roles in Section 5), y_{style} is the music style label, and G and D represent our generator and discriminator respectively. The overall objective for the generator in this training stage is

$$\begin{cases} \text{Loss}^{\text{G}} = \text{Loss}_{\text{MLE}} + \alpha\,\text{Loss}^{\text{G}}_{\text{Cls}} + \beta\,\text{Loss}^{\text{G}}_{\text{Gan}}, \\ \text{Loss}^{\text{G}}_{\text{Cls}} = -\mathbb{E}_{z\sim p_z(z)}[\log P(y_{\text{style}}|G(z))], \\ \text{Loss}^{\text{G}}_{\text{Gan}} = -\mathbb{E}_{z\sim p_z(z)}[\log D(G(z))], \end{cases} \tag{8}$$

where Loss_{MLE} is the maximum likelihood.

3.4 Style-conditioned evaluation metrics

We use some existing objective metrics to evaluate the effectiveness of the music generation model. Moreover, we propose a novel metric named style distance (SD) to evaluate the stylistic consistency of the generated music to the music in the original dataset.

1. Surface-level objective metrics

To evaluate the generated music, we use some conventional objective metrics (Dong et al., 2018; Yang and Lerch, 2020), i.e., pitch range (PR), number of unique pitch classes used (NPC), and scale consistency (SC), which are computed by MusPy (Dong et al., 2020). These metrics are computed for both real (training set) data and generated data. Then we calculate the differences between the metrics of the generated data and real data, and use |PR|, |NPC|, and |SC| to represent them. The smaller the values, the higher the similarity to the music in the datasets that the generated music is characterized by.

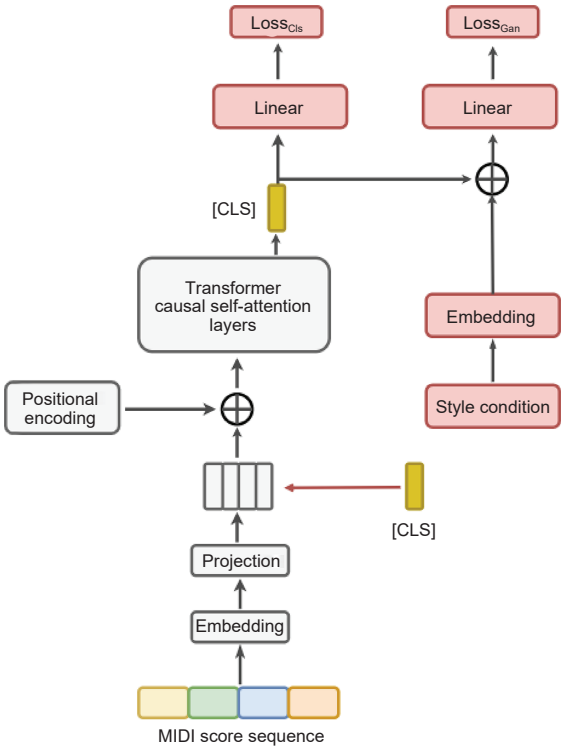

Fig. 4 Architecture of our style-conditioned patch discriminator

2. Style distance

A metric to evaluate the style consistency of the generated music is not available from the existing methods discussed in the literature. Thus, we propose the metric SD to evaluate the consistency of the generated music in terms of music style.

The surface-level objective metrics (|PR|, |NPC|, and |SC|) mentioned above can calculate only the similarity between the original music in datasets and all generated music (regardless of the style) as a whole. It is impossible to observe through these metrics whether the generated music maintains style consistency. However, we find that the values of PR, NPC, and SC vary according to the style classes. For example, in this study, we choose two datasets with music style labels, EMOPIA (with emotion style label) (Hung et al., 2021) and Pianst8 (with composer style label, https://zenodo.org/record/5089279). As shown in Tables 2 and 3, the average values of music in different classes are different in these metrics. In the EMOPIA dataset, music from group (Q1, Q2) can be distinguished by comparing PR with (Q3, Q4), while (Q2, Q4) can be distinguished by comparing NPC and SC with (Q1, Q3). In the Pianst8 dataset, music from Yiruma has the highest PR while Herbie Hancock's has the lowest.

Table 2 Surface-level objective metrics of music in the EMOPIA dataset

Emotion style	PR	NPC	SC
Q1 (HVHA)	55.79	8.50	0.963
Q2 (LVHA)	56.56	8.74	0.939
Q3 (HVLA)	44.59	7.96	0.962
Q4 (LVLA)	44.23	8.13	0.969
All data	50.34	8.34	0.958

PR: pitch range; NPC: number of unique pitch classes used; SC: scale consistency; HVHA: high valence high arousal; HVLA: high valence low arousal; LVHA: low valence high arousal; LVLA: low valence low arousal

Table 3 Surface-level objective metrics of music in the Pianst8 dataset

Composer style	PR	NPC	SC
Richard Clayderman	57.20	7.06	0.999
Yiruma	69.34	9.93	0.941
Herbie Hancock	46.91	8.16	0.979
Ludovico Einaudi	66.89	11.95	0.781
Hisaishi Joe	58.13	7.35	0.998
Ryuichi Sakamoto	59.07	10.66	0.921
Bethel Music	56.29	10.63	0.895
Hillsong Worship	55.78	10.29	0.919
All data	59.89	9.78	0.911

PR: pitch range; NPC: number of unique pitch classes used; SC: scale consistency

Therefore, we propose that the numerical difference between the generated music and the training data on these metrics should be calculated separately based on music style, followed by adding them up as an evaluation metric. We name this metric SD. With the three objective metrics (PR, NPC, and SC) mentioned above, the details of SD are as follows:

$$\mathrm{SD}=\sum_{n=1}^{N}\sum_{d=1}^{3}|\mathrm{Metric}_{d,n}-\mathrm{metric}_{d,n}|, \tag{9}$$

where N represents the total number of music styles, $\mathrm{Metric}_{d,n}$ represents the value of the d^{th} metric of the n^{th} music style in the original datasets, and $\mathrm{metric}_{d,n}$ represents the value of the generated music. The smaller the values, the greater the similarity to the music in the datasets characterizing the generated music.

3. Style-related objective metrics

To quantify how effectively the generation result is influenced by the style condition, we use a pre-trained music style classification model, MidiBERT-Piano (Chou et al., 2021), for evaluation. We first use our trained model to generate samples per class and use the assigned style labels as the target class of

the samples. Then the classification model is used to make style prediction on the generated samples. The higher the classification accuracy (CA), the better the generated music.

4 Experiments and results

4.1 Datasets

In our experiments, we choose two datasets for music style-conditioned generation training, EMOPIA (Hung et al., 2021) and Pianst8. EMOPIA is widely used in music emotion research and is well suited for emotion-based music generation. Pianst8, on the other hand, is rich in composer styles and is often used in music understanding. Here, it is very suitable for composer-style-based music generation.

1. Emotion-style dataset

EMOPIA is originally built for the study of musical emotions, and it has a total of 1087 clips of four emotion classes. It conceptualizes emotion in a two-dimensional space defined by valence and arousal. Considering a simple four-class taxonomy corresponding to the four quadrants (4Q) of Russell's Circumplex Model of Affect (Zhong et al., 2019), the four emotion classes are high valence high arousal (HVHA), high valence low arousal (HVLA), low valence high arousal (LVHA), and low valence low arousal (LVLA). In this study, we consider each emotion as a music style.

2. Composer-style dataset

Pianst8 consists of original piano music performed by eight composers. They are Richard Clayderman, Yiruma, Herbie Hancock, Ludovico Einaudi, Hisaishi Joe, Ryuichi Sakamoto, Bethel Music, and Hillsong Worship. The dataset contains a total of 411 pieces, with the number of pieces per composer being fairly balanced. In this study, we consider each composer as a music style.

4.2 Experimental setting

In this study, we compare our model with two state-of-the-art models (Hung et al., 2021; Sulun et al., 2022). These models are all implemented based on linear Transformer (Katharopoulos et al., 2020). All generator models are with 8 attention heads and 12 attention layers, while our patch discriminator is with 8 attention heads and 8 attention layers. The maximum input length of our generator is set to 513, and the head labels are cut at the output and fed into the discriminator; thus, the maximum input length of the discriminator is set to 512. The patch size is set to 16. In addition, the two hyperparameters of the discriminator, α and β, are set to 0.75 and 1.25, respectively. All experiments are conducted on a GeForce RTX 2080 Ti graphics card with 11 GB.

In the data pre-processing stage, we first transform all the data of EMOPIA and Pianst8 into the MIDI score format and insert music style labels in the header, as described in Section 3. After segmentation by the abovementioned fixed sequence length, we obtain, as our training data, a total of 924 segments of EMOPIA and 1186 segments of Pianst8.

4.3 Results

4.3.1 Objective evaluation

We generate many pieces of music from music generation models to perform the objective evaluation. In the emotion-style-based generation experiment, we generate 100 pieces of music for each emotion style, amounting to a total of 400 pieces. In the composer-style-based generation experiment, we generate 50 pieces of music for each style, amounting to a total of 400 pieces.

1. Music generation based on emotion style

We consider emotion as a kind of music style and compare our model with the two emotion-conditioned music generation models (Hung et al., 2021; Sulun et al., 2022). The experiments are conducted on the EMOPIA dataset. For result evaluation, we use the conventional objective metrics |PR|, |NPC|, |SC|, and CA, and our newly proposed metric SD, as described in Section 3.4. The results are presented in Table 4.

As shown in Table 4, our model obtains a superior performance in comparison with the existing models. Although the model proposed in Sulun et al. (2022) achieves higher CA than Emopia (Hung et al., 2021) by using sentiment control input augmentation, there is also a large gap between the generated music for each sentiment and the original dataset with a higher SD. In comparison, our model exhibits the best performance in all metrics, which means that the music generated by our model is not only similar to the original data in terms of surface-level objective metrics (|PR|, |NPC|, and |SC|), but also

Table 4 Performance comparison of objective evaluation

Dataset	Model	|PR| ↓	|NPC| ↓	|SC| ↓	SD ↓	CA ↑
EMOPIA	Emopia (Hung et al., 2021)	2.07	1.54	0.0012	16.62	52.5%
	Sulun et al. (2022)'s	4.52	1.63	0.0044	24.89	63.0%
	SCGT (ours)	**0.95**	**1.34**	**0.0012**	**12.45**	**69.5%**
Pianst8	Emopia (Hung et al., 2021)	1.39	**0.88**	0.043	33.47	29.5%
	Sulun et al. (2022)'s	1.44	1.41	0.038	39.05	54.0%
	SCGT (ours)	**1.30**	0.90	**0.036**	**29.37**	**67.0%**

Best results are in bold. |PR|: difference in pitch range; |NPC|: difference in the number of unique pitch classes used; |SC|: difference in scale consistency; SD: style distance; CA: classification accuracy

conforms to the corresponding emotion with a high CA.

Moreover, in the analysis of musical emotions, the arousal of music can be easily observed based on note density and note length (Livingstone et al., 2010). The note density is defined as the number of notes per beat, and the note length is defined as the average note length in the beat unit. As shown in Figs. 5a and 5b, the note lengths are generally longer in the low-arousal group (Q3, Q4), and the high-arousal group (Q1, Q2) has more dynamic (higher values in note density) than the low-arousal group. Thus, the high-arousal group (Q1, Q2) and low-arousal group (Q3, Q4) in the original dataset can be effectively distinguished. Similarly, in Figs. 5c and 5d, the music generated by our model is similar to the original dataset in terms of note density and note length, thus facilitating effective distinguishing between the high-arousal and low-arousal groups.

Furthermore, to evaluate the emotional expression ability of the music generated by these models, we use the pre-trained classification model MidiBERT-Piano to classify the generated music into four emotion classes. The classification results are shown in Figs. 6a–6c. As can be seen, the music generated by our method basically obtains higher classification accuracy on each emotion style, with an overall CA of 69.5%, meaning that the music we generate can better express the specific music emotion.

2. Music generation based on composer style

We also conduct experiments on the Pianst8 dataset for music generation based on composer styles. Similarly, we compare the results with those corresponding to the two existing models.

As shown in Table 4, our model still achieves a better performance than the other two models. Our model has the best results in all metrics except for the 0.02 difference in |NPC| compared to the existing model. Similar to the results on the EMOPIA dataset, although the results derived in Sulun et al. (2022) show an improvement in the CA of the generated music, the derived music is characterized by a higher SD, and performs poorly on the |NPC| and |PR|. Our model outperforms it by 13% in term of CA, and is closer to the original dataset in terms of other metrics. In addition, although our model dose not perform optimally on |NPC|, it achieves significantly higher performance on our proposed SD, indicating that SD is better at effectively measuring style consistency.

To evaluate the composer style of music generated by these models, we classify the generated music into eight composer classes. The details of the classification results are given in Figs. 6d–6f. The music generated by our method basically obtains a higher classification accuracy on each composer style, with an overall CA of 67.0%. Furthermore, the results show that as the number of music styles increases, Emopia is not able to generate the music of a specified style well and the overall CA is only 29.5%. Sulun et al. (2022)'s model also has a significant reduction in CA, while our model reduces it by only 1.5%. As the number of music styles increases, our model is still able to generate music with good differentiation. This indicates that our model can be applied to generate music with more styles, and the number of styles is not an important factor affecting the performance of our model.

The study of interpretive ability aims to turn complex models into controllable interfaces for interactive music performance (Jiang et al., 2020). In this study, our model offers a friendly style control interface, demonstrating a strong interpretive ability. As shown in Table 4, our model obtains the highest CA values, 69.5% with the EMOPIA dataset and 67.0% with the Pianst8 dataset. These results

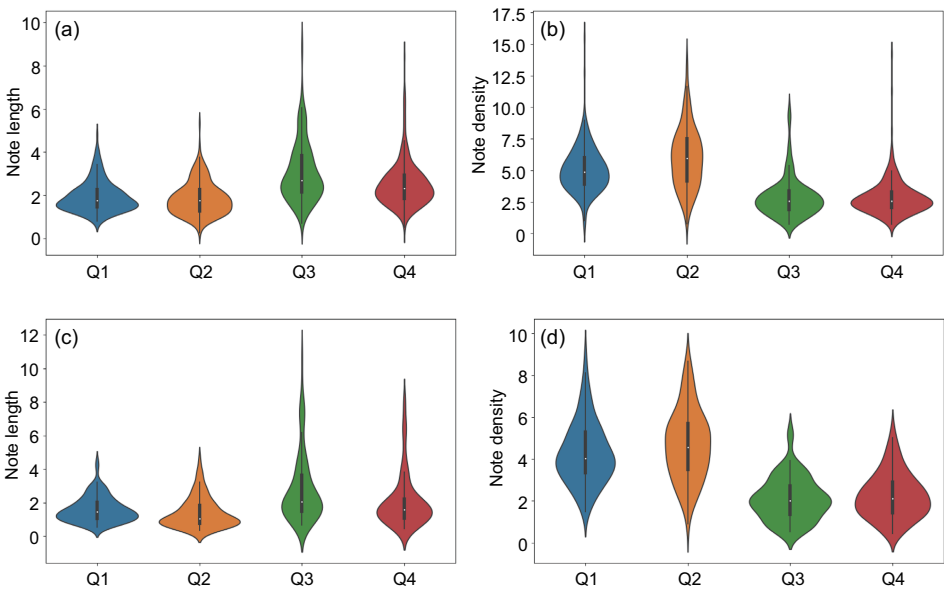

Fig. 5 Note length and note density distributions of the original dataset EMOPIA and the generated data: (a) note length of the original data; (b) note density of the original data; (c) note length of the generated data; (d) note density of the generated data

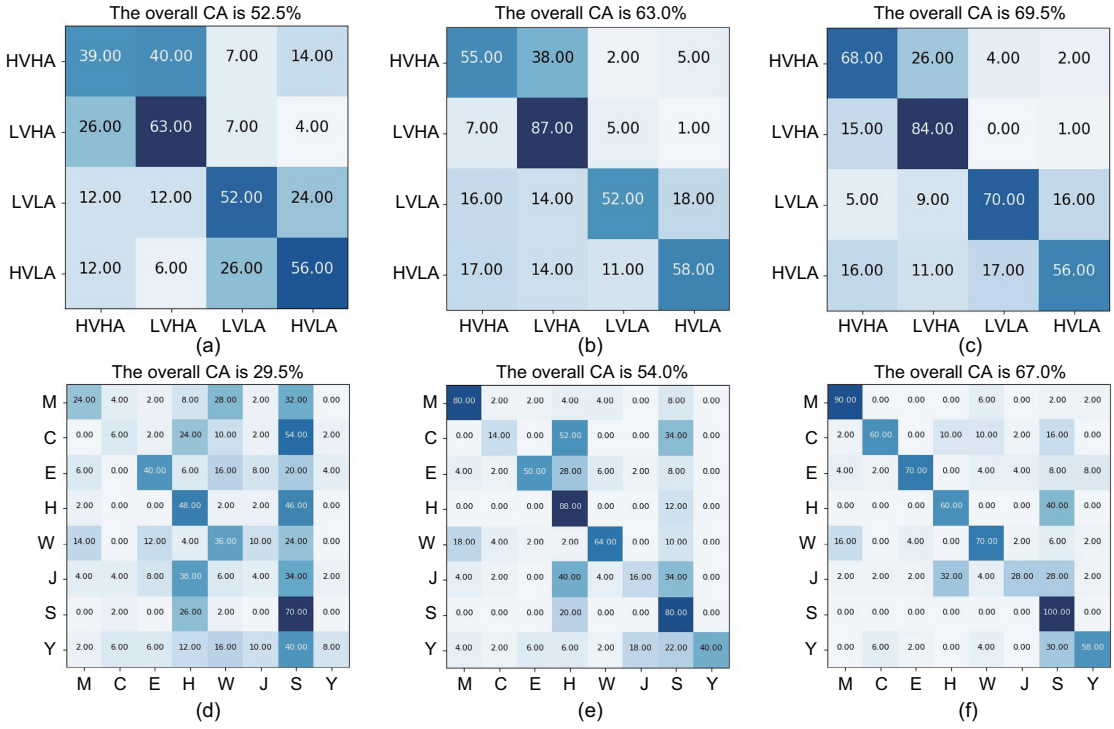

Fig. 6 Confusion tables for emotion- and composer-style classification of the generated music: (a) classification results of Emopia on the EMOPIA dataset; (b) classification results of Sulun et al. (2022)'s on the EMOPIA dataset; (c) classification results of SCTG (ours) on the EMOPIA dataset; (d) classification results of Emopia on the Pianst8 dataset; (e) classification results of Sulun et al. (2022)'s on the Pianst8 dataset; (f) classification results of SCTG (ours) on the Pianst8 dataset. CA: classification accuracy; HVHA: high valence high arousal; HVLA: high valence low arousal; LVHA: low valence high arousal; LVLA: low valence low arousal; M: Bethel Music; C: Richard Clayderman; E: Ludovico Einaudi; H: Herbie Hancock; W: Hillsong Worship; J: Hisaishi Joe; S: Ryuichi Sakamoto; Y: Yiruma

indicate that our model can better control the style of the generated music.

4.3.2 Subjective evaluation

We conduct a user study to further analyze our method. We setup the subjective evaluation of our model as Emopia (Hung et al., 2021). Participants are asked to evaluate models for both emotion-style-based and composer-style-based music generation. Specifically, for emotion-style-based models, there are 12 randomly generated samples, with one per model for each of the emotion classes. Similarly, there are 24 samples for composer-style-based models. Each participant is asked to rate the samples on a five-point Likert scale with respect to: (1) humanness (H, how well it sounds like a piece played by human); (2) richness (R, is the content interesting); and (3) overall musical quality (O). In total, 10 subjects participate in the survey.

We present the average scores of participants given to each model in Table 5. As shown in Table 5, the music generated by our model is preferred by participants, from which it would be reasonable to infer that it outperforms the existing models, implying a potential for its practical use.

Table 5 Performance comparison of subjective evaluation

Dataset	Model	H	R	O
EMOPIA	Emopia (Hung et al., 2021)	3.13	3.39	3.25
	Sulun et al. (2022)'s	3.46	3.17	3.52
	SCTG (ours)	**3.65**	**3.60**	**3.75**
Pianst8	Emopia (Hung et al., 2021)	3.22	3.34	3.15
	Sulun et al. (2022)'s	3.39	3.45	3.44
	SCTG (ours)	**3.92**	**3.81**	**3.96**

Best results are in bold. H: humanness; R: richness; O: overall musical quality

5 Ablation experiments

Based on the CP-Transformer (Hsiao et al., 2021), we innovatively embed style condition and implement a style-conditioned patch discriminator. We also design two losses related to the style condition in the discriminator. In this section, we perform ablation experiments to study their roles.

5.1 Without style-conditioned generator or discriminator

Based on the CP-Transformer (Hsiao et al., 2021), we implement our style-conditioned generator through embedding the style condition. We also design a style-conditioned patch discriminator to guide the generator. To study their roles, we separately implement them with style-conditioned generator and style-conditioned patch discriminator representatives, respectively. We conduct experiments on both EMOPIA and Pianst8 datasets, and the results of each metric are shown in Table 6.

As shown in Table 6, when we replace CP-Transformer with our style-conditioned generator, the CA of the generated music improves significantly. This indicates that the music style information in the model successfully helps it discriminate music styles and enables it to generate music of a specific style. After we add the style-conditioned patch discriminator, the generated music has a better generality and is more similar to the music in the original dataset. Except for |NPC| and |PR|, the complete SCTG performs the best on all metrics, especially on SD and CA far better than others. This means that our SCTG is effectively at generating music with good style consistency.

Table 6 Results of ablation experiments for each module in SCTG

Dataset	Model	|PR| ↓	|NPC| ↓	|SC| ↓	SD ↓	CA ↑
EMOPIA	CP-Transformer	2.07	1.54	0.0012	16.62	52.5%
	Style-conditioned generator	3.36	1.73	0.0039	20.66	65.3%
	CP-Transformer + style-conditioned patch discriminator	1.31	1.46	0.0051	13.92	57.5%
	SCTG (ours)	**0.95**	**1.34**	**0.0012**	**12.45**	**69.5%**
Pianst8	CP-Transformer	1.39	0.88	0.043	33.47	29.5%
	Style-conditioned generator	5.22	**0.78**	0.043	46.61	54.0%
	CP-Transformer + style-conditioned patch discriminator	**1.18**	0.79	0.044	29.80	35.5%
	SCTG (ours)	1.30	0.90	**0.036**	**29.37**	**67.0%**

Best results are in bold. |PR|: difference in pitch range; |NPC|: difference in the number of unique pitch classes used; |SC|: difference in scale consistency; SD: style distance; CA: classification accuracy

5.2 Without Loss_{Cls} or Loss_{Gan} in the discriminator

In our style-conditioned patch discriminator, we design two losses related to style condition. We separately remove them to study their roles. Similarly, we conduct experiments on both the EMOPIA and Pianst8 datasets. The results are shown in Table 7.

As shown in Table 7, the model performs poorly with the removal of either loss. Higher |PR| and higher |SC| indicate that the similarity between the generated music and the original dataset decreases. Higher SD and lower CA indicate that the style consistency of the generated music becomes worse. Therefore, for our style-conditioned patch discriminator, both losses are important for the performance.

6 Conclusions

Music, as a form of artistic expression, is part of human culture and history. Automatic music generation based on style will explore the richness of human-created music and generate music with rich emotions and distinctive style characteristics. In this paper, we have proposed a novel music generation model, SCTG, which can generate a complete musical composition from scratch based on a specified target style. We innovatively embedded style information in our proposed style-conditioned linear Transformer. We also designed a style-conditioned patch discriminator with two innovative losses to enhance the interpretive ability and structural awareness of the model. In addition, to evaluate the style consistency, we defined a discriminative metric for the first time. Extensive experiments on two public datasets showed the effectiveness of our approach. Our model achieved the best in the objective evaluation, especially in the two metrics for measuring the style consistency (SD and CA). Furthermore, the results of the subjective evaluation proved that the music generated by our model was preferred by participants.

Moreover, as a kind of music style, emotion has been widely studied in previous research. Our method can also be used directly in this field to generate music with specific emotion. Furthermore, unlike most studies dealing with style transfer, our method allows users to specify styles and completely generate a full song from scratch. The benefit deriving from the introduction of style information in each module of the model is that our model can effectively control the style characteristics of the generated music.

However, there are relatively few music datasets with style information. We will build a richer dataset in the future and continue to explore style-conditioned music generation. In addition, we will further investigate the influence of style information on the inference stage of the model and introduce music style information in this stage.

Contributors

Weining WANG and Jiahui LI designed the research and processed the data. Jiahui LI and Yifan LI drafted the paper. Weining WANG and Xiaofen XING helped organize the paper. Weining WANG and Xiaofen XING revised and finalized the paper.

Compliance with ethics guidelines

All the authors declare that they have no conflict of interest.

Data availability

The data that support the findings of this study are available from the corresponding author upon reasonable request. The code and some generated music examples are

Table 7 Results of ablation experiments for Loss_{Cls} and Loss_{Gan} in SCTG

Dataset	Model	|PR| ↓	|NPC| ↓	|SC| ↓	SD ↓	CA ↑
EMOPIA	Without Loss_{Cls}	2.45	1.57	0.0022	16.35	67.8%
	Without Loss_{Gan}	3.08	1.73	0.0051	19.57	68.0%
	SCTG (ours)	**0.95**	**1.34**	**0.0012**	**12.45**	**69.5%**
Pianst8	Without Loss_{Cls}	3.58	0.88	0.036	31.42	61.3%
	Without Loss_{Gan}	3.46	**0.56**	0.051	32.11	56.5%
	SCTG (ours)	**1.30**	0.90	**0.036**	**29.37**	**67.0%**

Best results are in bold. |PR|: difference in pitch range; |NPC|: difference in the number of unique pitch classes used; |SC|: difference in scale consistency; SD: style distance; CA: classification accuracy

shared in https://github.com/li-car-fei/SCTG.

References

Brunner G, Konrad A, Wang YY, et al., 2018. MIDI-VAE: modeling dynamics and instrumentation of music with applications to style transfer. Proc 19th Int Society for Music Information Retrieval Conf, p.747-754.

Choi K, Hawthorne C, Simon I, et al., 2020. Encoding musical style with Transformer autoencoders. Proc 37th Int Conf on Machine Learning, p.1899-1908.

Chou YH, Chen IC, Chang CJ, et al., 2021. MidiBERT-Piano: large-scale pre-training for symbolic music understanding. https://doi.org/10.48550/arXiv.2107.05223

Delgado M, Fajardo W, Molina-Solana M, 2009. Inmamusys: intelligent multiagent music system. *Expert Syst Appl*, 36(3):4574-4580. https://doi.org/10.1016/j.eswa.2008.05.028

Devlin J, Chang MW, Lee K, et al., 2019. BERT: pre-training of deep bidirectional Transformers for language understanding. Proc Conf of the North American Chapter of the Association for Computational Linguistics, p.4171-4186. https://doi.org/10.18653/v1/N19-1423

Dong HW, Yang YH, 2018. Convolutional generative adversarial networks with binary neurons for polyphonic music generation. Proc 19th Int Society for Music Information Retrieval Conf, p.190-196.

Dong HW, Hsiao WY, Yang LC, et al., 2018. MuseGAN: multi-track sequential generative adversarial networks for symbolic music generation and accompaniment. Proc 32nd AAAI Conf on Artificial Intelligence, Article 5.

Dong HW, Chen K, McAuley JJ, et al., 2020. MusPy: a toolkit for symbolic music generation. Proc 21st Int Society for Music Information Retrieval Conf, p.101-108.

Dosovitskiy A, Beyer L, Kolesnikov A, et al., 2021. An image is worth 16x16 words: Transformers for image recognition at scale. Proc 9th Int Conf on Learning Representations.

Ferreira LN, Whitehead J, 2021. Learning to generate music with sentiment. Proc 20th Int Society for Music Information Retrieval Conf, p.384-390.

Goodfellow I, Pouget-Abadie J, Mirza M, et al., 2020. Generative adversarial networks. *Commun ACM*, 63(11):139-144. https://doi.org/10.1145/3422622

Herremans D, Chew E, 2019. MorpheuS: generating structured music with constrained patterns and tension. *IEEE Trans Affect Comput*, 10(4):510-523. https://doi.org/10.1109/TAFFC.2017.2737984

Hsiao WY, Liu JY, Yeh YC, et al., 2021. Compound word Transformer: learning to compose full-song music over dynamic directed hypergraphs. Proc 35th AAAI Conf on Artificial Intelligence, p.178-186. https://doi.org/10.1609/aaai.v35i1.16091

Huang CZA, Vaswani A, Uszkoreit J, et al., 2019. Music Transformer: generating music with long-term structure. Proc 7th Int Conf on Learning Representations.

Huang YS, Yang YH, 2020. Pop music Transformer: beat-based modeling and generation of expressive pop piano compositions. Proc 28th ACM Int Conf on Multimedia, p.1180-1188. https://doi.org/10.1145/3394171.3413671

Hung HT, Ching J, Doh S, et al., 2021. EMOPIA: a multi-modal pop piano dataset for emotion recognition and emotion-based music generation. Proc 22nd Int Society for Music Information Retrieval Conf, p.318-325.

Jang E, Gu SX, Poole B, 2017. Categorical reparameterization with Gumbel-Softmax. Proc 5th Int Conf on Learning Representations.

Jhamtani H, Berg-Kirkpatrick T, 2019. Modeling self-repetition in music generation using generative adversarial networks. Proc 36th Int Conf on Machine Learning.

Jiang JY, Wang ZQ, 2019. Stylistic melody generation with conditional variational auto-encoder. Available from https://www.cs.cmu.edu/∼epxing/Class/10708-19/assets/project/final-reports/project8.pdf [Accessed on Oct. 28, 2023].

Jiang JY, Xia GG, Carlton DB, et al., 2020. Transformer VAE: a hierarchical model for structure-aware and interpretable music representation learning. Proc IEEE Int Conf on Acoustics, Speech and Signal Processing, p.516-520. https://doi.org/10.1109/ICASSP40776.2020.9054554

Kaliakatsos-Papakostas M, Floros A, Vrahatis MN, 2020. Artificial intelligence methods for music generation: a review and future perspectives. In: Yang XS (Ed.), Nature-Inspired Computation and Swarm Intelligence. Academic Press, Amsterdam, p.217-245. https://doi.org/10.1016/B978-0-12-819714-1.00024-5

Katharopoulos A, Vyas A, Pappas N, et al., 2020. Transformers are RNNs: fast autoregressive transformers with linear attention. Proc 37th Int Conf on Machine Learning, p.5156-5165.

Ke GL, He D, Liu TY, 2021. Rethinking positional encoding in language pre-training. Proc 9th Int Conf on Learning Representations.

Leach J, Fitch J, 1995. Nature, music, and algorithmic composition. *Comput Music J*, 19(2):23-33. https://doi.org/10.2307/3680598

Liang X, Wu JM, Cao J, 2019. MIDI-Sandwich2: RNN-based hierarchical multi-modal fusion generation VAE networks for multi-track symbolic music generation. https://doi.org/10.48550/arXiv.1909.03522

Liao YK, Yue W, Jian YQ, et al., 2022. MICW: a multi-instrument music generation model based on the improved compound word. Proc IEEE Int Conf on Multimedia and Expo Workshops, p.1-10. https://doi.org/10.1109/ICMEW56448.2022.9859531

Lim YQ, Chan CS, Loo FY, 2020. Style-conditioned music generation. Proc IEEE Int Conf on Multimedia and Expo, p.1-6. https://doi.org/10.1109/ICME46284.2020.9102870

Liu HM, Yang YH, 2018. Lead sheet generation and arrangement by conditional generative adversarial network. Proc 17th IEEE Int Conf on Machine Learning and Applications, p.722-727. https://doi.org/10.1109/ICMLA.2018.00114

Livingstone SR, Mühlberger R, Brown AR, et al., 2010. Changing musical emotion: a computational rule system for modifying score and performance. *Comput Music J*, 34(1):41-64. https://doi.org/10.1162/comj.2010.34.1.41

Lousseief E, Sturm BLT, 2019. MahlerNet: unbounded orchestral music with neural networks. Proc Nordic Sound and Music Computing Conf and the Interactive Sonification Workshop, p.58-64. https://doi.org/10.5281/zenodo.3755968

Luo J, Yang XY, Ji SL, et al., 2020. MG-VAE: deep Chinese folk songs generation with specific regional styles. Proc 7th Conf on Sound and Music Technology, p.93-106. https://doi.org/10.1007/978-981-15-2756-2_8

Mao HH, Shin T, Cottrell G, 2018. DeepJ: style-specific music generation. Proc IEEE 12th Int Conf on Semantic Computing, p.377-382. https://doi.org/10.1109/ICSC.2018.00077

Mou LT, Sun YH, Tian YH, et al., 2023. MemoMusic 3.0: considering context at music recommendation and combining music theory at music generation. Proc IEEE Int Conf on Multimedia and Expo Workshops, p.296-301. https://doi.org/10.1109/ICMEW59549.2023.00057

Muhamed A, Li L, Shi XJ, et al., 2021. Symbolic music generation with Transformer-GANs. Proc 35th AAAI Conf on Artificial Intelligence, p.408-417. https://doi.org/10.1609/aaai.v35i1.16117

Nie WL, Narodytska N, Patel A, 2019. RelGAN: relational generative adversarial networks for text generation. Proc 7th Int Conf on Learning Representations.

Oore S, Simon I, Dieleman S, et al., 2020. This time with feeling: learning expressive musical performance. *Neur Comput Appl*, 32(4):955-967. https://doi.org/10.1007/s00521-018-3758-9

Ren Y, He JZ, Tan X, et al., 2020. PopMAG: pop music accompaniment generation. Proc 28th ACM Int Conf on Multimedia, p.1198-1206. https://doi.org/10.1145/3394171.3413721

Rivero D, Ramírez-Morales I, Fernandez-Blanco E, et al., 2020. Classical music prediction and composition by means of variational autoencoders. *Appl Sci*, 10(9): 3053.

Roberts A, Engel J, Raffel C, et al., 2018. A hierarchical latent vector model for learning long-term structure in music. Proc 35th Int Conf on Machine Learning, p.4364-4373.

Shih YJ, Wu SL, Zalkow F, et al., 2022. Theme Transformer: symbolic music generation with theme-conditioned Transformer. *IEEE Trans Multimed*, 25: 3495-3508. https://doi.org/10.1109/TMM.2022.3161851

Sulun S, Davies MEP, Viana P, 2022. Symbolic music generation conditioned on continuous-valued emotions. *IEEE Access*, 10:44617-44626. https://doi.org/10.1109/ACCESS.2022.3169744

Supper M, 2001. A few remarks on algorithmic composition. *Comput Music J*, 25(1):48-53. https://doi.org/10.1162/014892601300126106

Trieu N, Keller RM, 2018. JazzGAN: improvising with generative adversarial networks. Proc 6th Int Workshop on Musical Metacreation.

Vaswani A, Shazeer N, Parmar N, et al., 2017. Attention is all you need. Proc 31st Int Conf on Neural Information Processing Systems, p.6000-6010.

Waite E, Eck D, Roberts A, et al., 2016. Project magenta: generating long-term structure in songs and stories. Available from https://github.com/magenta/magenta/issues/1438 [Accessed on Oct. 28, 2023].

Wang L, Zhao ZY, Liu HW, et al., 2023. A review of intelligent music generation systems. https://doi.org/10.48550/arXiv.2211.09124

Wang WP, Li XB, Jin C, et al., 2022. CPS: full-song and style-conditioned music generation with linear transformer. Proc IEEE Int Conf on Multimedia and Expo Workshops, p.1-6. https://doi.org/10.1109/ICMEW56448.2022.9859286

Williams RJ, Zipser D, 1989. A learning algorithm for continually running fully recurrent neural networks. *Neur Comput*, 1(2):270-280. https://doi.org/10.1162/neco.1989.1.2.270

Wu SL, Yang YH, 2020. The jazz Transformer on the front line: exploring the shortcomings of AI-composed music through quantitative measures. Proc 21st Int Society for Music Information Retrieval Conf, p.142-149.

Wu XC, Wang CY, Lei QY, 2020. Transformer-XL based music generation with multiple sequences of time-valued notes. https://doi.org/10.48550/arXiv.2007.07244

Yang LC, Lerch A, 2020. On the evaluation of generative models in music. *Neur Comput Appl*, 32(9):4773-4784. https://doi.org/10.1007/s00521-018-3849-7

Yang LC, Chou SY, Yang YH, 2017. MidiNet: a convolutional generative adversarial network for symbolic-domain music generation. Proc 18th Int Society for Music Information Retrieval Conf, p.324-331.

Yu BT, Lu PL, Wang R, et al., 2022. Museformer: Transformer with fine- and coarse-grained attention for music generation. Proc 36th Conf on Neural Information Processing Systems, p.1376-1388.

Zhang N, 2023. Learning adversarial transformer for symbolic music generation. *IEEE Trans Neur Netw Learn Syst*, 34(4):1754-1763. https://doi.org/10.1109/TNNLS.2020.2990746

Zhang XY, Zhang JC, Qiu Y, et al., 2022. Structure-enhanced pop music generation via harmony-aware learning. Proc 30th ACM Int Conf on Multimedia, p.1204-1213. https://doi.org/10.1145/3503161.3548084

Zhong K, Qiao TW, Zhang LQ, 2019. A study of emotional communication of emoticon based on Russell's Circumplex Model of Affect. Proc 8th Int Conf on Human-Computer Interaction, p.577-596. https://doi.org/10.1007/978-3-030-23570-3_43